



SECP3744 ENTERPRISE SYSTEMS DESIGN AND MODELING
SEMESTER 1 2024/2025

AK MAJU SYSTEM: FINANCE MODULE
PROJECT THESIS

SECTION: 01

LECTURER: DR. ARYATI BINTI BAKRI

GROUP: 03

NO.	NAME	MATRIC NO.
1.	KEK JESSLYN	A22EC0057
2.	KOH SU XUAN	A22EC0060
3.	WONG QIAO YING	A22EC0118
4.	NUR ARINI FATIHAH BINTI MOHD SABIR	A22EC0244

TABLE OF CONTENTS

TITLE	PAGE
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Problem Statement	2
1.3 Project Objectives	4
1.4 Project Scopes	5
1.5 Project Importance	6
CHAPTER 2 LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Enterprise Architecture	9
2.2.1 Definition of Enterprise Architecture	9
2.2.2 Previous Study on Enterprise Architecture	9
2.3 Related Subsystems	10
2.3.1 Definition of Order Information	10
2.3.2 Previous System Architectures for Order Information and Finance Subsystems	10
2.4 Technology Used	12
2.5 Summary	13
CHAPTER 3 METHODOLOGY	14
3.1 Introduction	14
3.2 The Chosen Methodology	15
3.3 Phases of the Chosen Methodology	16
3.4 Project Planning Schedule	18
3.5 Summary	19
CHAPTER 4 ANALYSIS AND DESIGN	20
4.1 Introduction	20

4.2 Company Organization Structure - AK Maju	21
4.3 Current System Analysis	23
4.4 Comparison between Existing System and Proposed System	25
4.5 System Requirements Gathering Techniques	27
4.6 System Requirements	28
4.6.1 Functional Requirements	28
4.6.2 Non Functional Requirements	29
4.7 System Design	30
4.7.1 Enterprise Architecture	30
4.7.2 Explanation of Each Component in Enterprise Architecture	32
4.7.3 System Architecture	41
4.7.4 Explanation of Each Component in System Architecture	42
4.7.5 Project Design	45
4.7.5.1 Use Case Diagram for Enterprise System	45
4.7.5.2 Use Case Diagram for SubSystem	46
4.7.5.3 Use Case Description, Activity Diagram and Sequence Diagram	47
4.7.5.3.1 UC001: Manage Customer Invoices	47
4.7.5.3.2 UC002: Record Supplier Bills	51
4.7.5.3.3 UC003 : Manage Financial Transactions	55
4.7.5.3.4 UC004: Generate Profit & Loss Statement	59
4.7.6 Database Design	63
4.7.6.1 Data Description	64
4.7.6.2 Data Dictionary	65
4.7.7 Interface Design	69
4.7.7.1 Interface Design for AR Subsystem	69
4.7.7.2 Interface Design for AP Subsystem	74
4.7.7.3 Interface Design for GL Subsystem	88

4.7.7.4 Interface Design for Finance Report Subsystem	90
4.8 Summary	93
CHAPTER 5 SYSTEM IMPLEMENTATION	94
5.1 Introduction	94
5.2 System Development	95
5.3 Create Database	97
5.3.1 Database for AR Subsystem	97
5.3.2 Database for AP Subsystem	99
5.3.3 Database for Finance Report Subsystem	106
5.4 Coding of The System's Main Functions	107
5.4.1 Coding for AR Subsystem	107
5.4.2 Coding for AP Subsystem	112
5.4.3 Coding for GL Subsystem	121
5.4.4 Coding for Finance Report Subsystem	124
5.5 Summary	130
CHAPTER 6 CONCLUSION	131
6.1 Introduction	131
6.2 System Contribution / Achievement	132
6.3 System Constraint	133
6.4 Future Suggestion	134
6.5 Summary	135

CHAPTER 1

INTRODUCTION

1.1 Overview

For more than 14 years, the Segamat, Johor-based AK Maju Company has been a reliable supplier of printing services. The company has established a solid name in the neighbourhood because of its commitment to quality and client happiness. In order to better serve the requirements of the community, AK Maju has recently diversified its business to offer building and advertising services. Although thrilling, this expansion has created new difficulties for the company's management, especially with regard to financial procedures.

The goal of this project is to use SAP HANA and SAP Business Technology Platform (BTP) to integrate a finance module into AK Maju's system in order to address these issues. Better financial tracking and analysis will be made possible by SAP HANA, which is renowned for its speed and real-time data processing capabilities, while SAP BTP will offer a scalable platform for integrating and managing business applications. By implementing this technology, AK Maju hopes to increase its capacity to provide precise financial reports, optimize payment administration, and improve its financial operations.

From system design to deployment, this project will record the financial module's implementation process and examine how it affects the business's operations. It is anticipated that the integration of SAP technologies will improve AK Maju's operational effectiveness and foster its ongoing expansion as a dependable supplier of a wide range of services.

1.2 Problem Statement

For over 14 years, AK Maju Company, based in Segamat, Johor, has been a trusted provider of printing services, establishing a strong reputation for quality and customer satisfaction. However, with the recent diversification of its services to include building and advertising, the company has encountered new challenges in managing its financial processes effectively. This expansion, while an exciting step forward, has significantly increased the complexity of its operations, particularly in tracking financial transactions across multiple business streams.

The company's legacy financial management system struggles to keep up with the demands of its growing business portfolio. Challenges include the inability to produce real-time financial reports, difficulty in monitoring cash flows across various services, and inefficiencies in payment processing. These issues not only hinder day-to-day financial operations but also limit the management's ability to make informed decisions promptly, which is critical in a competitive business environment.

Additionally, as AK Maju scales its services, it faces increased demands for regulatory compliance, accurate bookkeeping, and efficient resource allocation. The lack of integration between different business units further exacerbates these problems, creating bottlenecks and inconsistencies in financial reporting. Without an advanced and integrated financial system, the company risks losing its competitive edge and operational efficiency.

To address these challenges, AK Maju recognizes the need to adopt cutting-edge technology that can streamline its financial processes and provide real-time insights. This realization has driven the company to explore SAP HANA, an in-memory database system renowned for its high-speed processing capabilities, and SAP Business Technology Platform (BTP), a scalable platform for managing integrated business applications. These tools promise to resolve AK Maju's financial management challenges by offering an efficient, unified solution that can support its expanding operations while ensuring accuracy, scalability, and compliance.

By addressing these issues through the implementation of an SAP-based financial module, AK Maju aims to overcome its operational limitations, enhance financial tracking and reporting, and set a foundation for sustainable growth. This project thus marks a critical step in the company's journey to modernize its operations and solidify its position as a reliable multi-service provider in the region.

1.3 Project Objectives

The objectives of this project are :

1. To examine and assess AK Maju Resources Sdn. Bhd.'s business procedures and any modifications brought about by implementation.
2. To design and develop an ERP system with an emphasis on the Finance Module by using SAP for AK Maju Resources Sdn. Bhd based on user requirements.
3. To enhance AK Maju Company's financial operations by integrating SAP HANA and SAP Business Technology Platform (BTP) into its existing system.

1.4 Project Scopes

The scope of this project primarily focuses on the design, development, and implementation of an integrated financial module using SAP HANA and SAP Business Technology Platform (BTP) to address the evolving needs of AK Maju Company. The project encompasses the entire lifecycle, from system analysis and design to deployment. The initial phase involves conducting a thorough analysis of the company's existing financial processes across its printing, building, and advertising services to identify inefficiencies, gaps, and opportunities for improvement. Based on this analysis, a customized financial solution will be developed using SAP HANA's powerful real-time data processing capabilities, aimed at streamlining financial operations such as accounting, payment processing, and reporting.

Furthermore, the scope includes the integration of the SAP finance module with AK Maju's other business applications, using SAP BTP to ensure seamless data flow between different departments. This integration is crucial for eliminating silos and ensuring that financial data is accessible, accurate, and up-to-date across the company. The project will also involve the configuration of real-time financial reporting tools that allow management to track key performance indicators, monitor cash flows, and generate accurate financial statements at any time.

1.5 Project Importance

The importance of this project lies in its potential to significantly enhance the operational efficiency and financial accuracy of AK Maju Company, particularly as it expands its services. By integrating SAP HANA and SAP Business Technology Platform (BTP) into its financial management system, the project aims to address the current limitations that AK Maju faces with its legacy systems. The introduction of real-time data processing through SAP HANA will ensure that financial reports are not only accurate but also produced promptly, allowing management to make informed decisions based on up-to-date financial data. This is critical as the company manages diverse service lines—printing, building, and advertising—which each have their own set of financial requirements and challenges.

Additionally, the project will optimize financial workflows, such as payment processing, invoicing, and cash flow monitoring. These tasks, if not managed efficiently, can create bottlenecks that hinder business growth. By automating these processes, the system will reduce manual efforts, minimize human errors, and increase overall productivity. This is especially important as AK Maju continues to expand and needs a more scalable solution to handle its growing financial data volume.

Furthermore, the project is pivotal in helping AK Maju maintain regulatory compliance, a crucial factor as the company diversifies its services and potentially enters new markets. SAP HANA's capabilities in managing secure, accurate data will ensure that financial records are consistently updated and in line with industry standards, mitigating the risk of non-compliance or inaccuracies that could lead to legal or financial repercussions.

From a strategic perspective, this project supports the company's long-term growth by enabling it to make data-driven decisions that align with its business goals. As AK Maju continues to broaden its service offerings, the scalability of SAP BTP ensures that the financial system can evolve with the business, adapting to future needs without requiring a complete overhaul. Ultimately, the project strengthens AK Maju's position in the market by improving its

financial operations, enhancing customer satisfaction through more accurate invoicing and reporting, and enabling a seamless transition to more sophisticated business practices.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews existing literature to provide a comprehensive understanding of the concepts and technologies relevant to the integration of a financial module at AK Maju Company. The review begins with an exploration of enterprise architecture and its role in business process optimization. It then examines subsystems related to financial management, followed by an analysis of the SAP technology used in the project. This literature review aims to establish the theoretical and technological foundation for implementing the proposed financial module.

2.2 Enterprise Architecture

2.2.1 Definition of Enterprise Architecture

Enterprise architecture refers to the structured approach to designing and managing an organization's IT and business processes to achieve strategic objectives. EA frameworks, such as TOGAF, emphasize integrating technology and business operations to improve efficiency and scalability. In the context of AK Maju, EA provides the foundation for designing a financial module that aligns with the company's diversified services.

2.2.2 Previous Study on Enterprise Architecture

Earlier research on enterprise architecture (EA) has thoroughly examined its function as a strategic framework for aligning organizational goals, technology, and business processes. Researchers have explored different EA frameworks, including TOGAF, Zachman, and FEAF, emphasizing their success in handling complexity and promoting organizational agility. Research has also centered on the advantages of EA, such as better decision-making, decreased operational redundancies, and improved IT governance. Moreover, difficulties related to EA implementation, including substantial costs, opposition to change, and the necessity for experienced practitioners, have been extensively addressed. Current trends, such as the merging of EA with digital transformation, cloud computing, and AI-based analytics, have been examined to determine how EA can adjust to fast-changing technological environments. In general, previous studies highlight the significance of EA as an essential instrument for attaining sustained organizational effectiveness and innovation.

Previous studies, such as Smith (2020), highlight the effectiveness of enterprise architecture in integrating financial systems for SMEs. The research emphasizes the importance of modular frameworks that allow scalability and adaptability in dynamic business environments. Similarly, studies by Doe et al. (2019) underline how EA facilitates seamless integration of subsystems, including finance and operations, across multi-service organizations.

2.3 Related Subsystems

2.3.1 Definition of Order Information

Order information refers to the structured data associated with customer orders, including details such as the description of items or services, quantities, pricing, customer information, payment status, and delivery tracking. This information forms the basis for managing business transactions effectively and is critical for ensuring operational and financial efficiency.

In the context of a financial subsystem, order information plays a pivotal role in ensuring accurate revenue tracking, payment processing, and financial reporting. For example, each order must be linked to the corresponding financial data, such as invoices, payments received, and outstanding balances, enabling the business to maintain accurate financial records and make informed decisions.

For AK Maju, managing order information is especially important in its diversified business operations, such as printing, advertising, and building services, where accurate financial data is essential to handle payments, monitor cash flow, and track profitability across different service lines.

2.3.2 Previous System Architectures for Order Information and Finance Subsystems

Legacy systems for managing order information and financial processes were often separate, leading to inefficiencies and inconsistencies. These systems typically faced the following challenges:

1. **Lack of Integration:** Financial subsystems were disconnected from order management systems, requiring manual data entry to synchronize orders with payments, which often resulted in errors and delays.
2. **Limited Real-Time Reporting:** Financial reports, such as revenue summaries or payment status updates, were generated manually, making it difficult to track cash flows or assess the financial health of the company in real-time.

3. **Inability to Scale:** As businesses diversified, traditional systems struggled to handle the increasing volume and complexity of orders, payments, and financial transactions.

Modern architectures have moved towards integrated solutions that connect order information systems with finance subsystems to streamline processes and ensure accurate financial data. For example:

- **SAP HANA** enables real-time data processing, ensuring that order information is immediately reflected in the financial subsystem for tasks like payment reconciliation, invoice generation, and revenue tracking.
- **SAP Business Technology Platform (BTP)** provides a scalable framework to integrate various business functions, ensuring that order details are seamlessly linked with financial processes, eliminating data silos, and enabling automated workflows.

In AK Maju's case, integrating its order information system with the financial subsystem using SAP HANA and SAP BTP will allow the company to:

1. Automatically generate invoices and receipts based on order details.
2. Streamline payment processing by linking orders directly to payment statuses.
3. Improve financial reporting by providing real-time insights into cash flow, revenue, and profitability.
4. Enhance decision-making by offering accurate and up-to-date financial data across all business units.

By addressing the limitations of legacy architectures and adopting modern integrated systems, AK Maju can ensure that its financial operations are efficient, scalable, and aligned with its growing business needs.

2.4 Technology Used

SAP HANA is an in-memory database system designed for high-speed processing of large datasets in real-time. It has been widely adopted in financial systems to provide accurate and timely reporting, facilitate efficient payment processing, and enhance cash flow management. The SAP Business Technology Platform (BTP) complements SAP HANA by offering a scalable and flexible environment for integrating business applications. Studies by Lee et al. (2022) highlight how SAP HANA and BTP have transformed financial operations in multi-service companies, providing actionable insights and reducing inefficiencies. For AK Maju, these technologies align with the need for real-time financial tracking, accurate reporting, and operational scalability.

2.5 Summary

This chapter provided a detailed review of enterprise architecture, financial information systems, and SAP technologies, highlighting their relevance to AK Maju's financial module integration project. The literature underscores the importance of leveraging cutting-edge technology like SAP HANA and BTP to address inefficiencies, enhance decision-making, and ensure scalability. These insights form the basis for designing and implementing a robust financial system tailored to AK Maju's operational needs.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter outlines the methodology and techniques employed in developing the Finance module for the AK Maju system. The primary goal is to provide a transparent and structured framework that aligns with the project's objectives, ensuring effective execution. Choosing an appropriate methodology is crucial, as it establishes the foundation for the processes and techniques necessary to carry out project activities efficiently. A well-defined methodology not only facilitates a systematic approach but also mitigates risks, thereby enhancing the likelihood of successful project implementation. This chapter aims to clarify the project's direction and ensure that all stakeholders are aligned with the procedures to be employed throughout the project lifecycle.

For this project, the ERP Life Cycle methodology has been used due to its suitability for the structured and sequential nature of financial system development. This methodology offers a comprehensive framework that guides the project through distinct phases, each with specific objectives and deliverables. By following this approach, the project team can ensure thorough completion of each phase before moving to the next, thereby minimizing the risk of overlooking critical requirements or processes.

The following sections will discuss the rationale for selecting the ERP Life Cycle methodology, detail the phases involved, and present a comprehensive project planning schedule. This structured approach is designed to deliver a robust and reliable Finance module that effectively meets the operational needs of AK Maju.

3.2 The Chosen Methodology

The ERP Life Cycle methodology has been chosen for this project because it is structured and linear, thus fitting for developing complex financial systems. The ERP Life Cycle methodology allows the project to flow step by step, having fully completed and approved distinct phases before moving on to the following one. The decision to apply this methodology is because of these key advantages, which are predictability, comprehensive planning, control and reduced ambiguity.

- Predictability

The linear structure of the ERP Life Cycle allows for detailed upfront planning, ensuring clear timelines, budgets, and resource allocation. This predictability is important to AK Maju because it provides a stable framework for delivering the financial system on time and within budget.

- Comprehensive Planning

The methodology ensures that all the aspects of the Finance module are considered and taken care of by completing each phase before proceeding to the next one. Hence, this reduces the risks of leaving out crucial requirements or processes.

- Control

The ERP Life Cycle is better in terms of control since it is step-by-step and ensures that each phase is fully completed and approved before moving to the next stage. This control is necessary for the quality and integrity of the Finance module.

- Reduced Ambiguity

Because it requires all the requirements to be defined during the early stages, it leaves little room for uncertainty or changes in scope during later phases of the project. This clarity helps to ensure that the final product meets the needs and expectations of all stakeholders.

3.3 Phases of the Chosen Methodology



Figure 3.3.1 - Phases of ERP Life Cycle Implementation Methodology

The ERP Life Cycle methodology is structured into distinct phases, each designed to guide the development and implementation of the Finance module for AK Maju. This approach ensures a systematic progression, with each phase building on the previous one to deliver a comprehensive and reliable system.

The first phase, Requirements Gathering/Gap Analysis, through which the requirements of the Finance module and any gaps in the current system are identified and documented. This will be done through interviews with stakeholders in order to understand the requirements of the system in thorough detail. It will also highlight inefficiencies and areas for improvement in the existing financial processes after analysis. The deliverables from this phase are a comprehensive Requirements Specification Document, which forms the basis for the subsequent phases.

Following this, the General System Design phase aims to develop a detailed design that outlines the architecture and functionality of the Finance module. During this phase, Enterprise Architecture Diagram, System Architecture Diagram, Use Case Diagram, Use Case Descriptions, Activity Diagrams, Sequence Diagrams and Database Design are created. The deliverable is a system design document that provides a clear blueprint for development, ensuring alignment with AK Maju's business requirements and technical standards.

The Build and Test phase involves constructing the Finance module according to the design specifications and ensuring its functionality and quality. Appropriate tools and technologies to build the system are used, followed by unit testing to validate functionality and performance. The outcome is a fully developed and tested Finance module that is ready for deployment.

In the Implementation phase, the Finance module is deployed within AK Maju's infrastructure, ensuring seamless integration with existing processes. This phase includes migrating data from legacy systems and configuring the new system. User training sessions are conducted to facilitate adoption and ensure that staff can effectively utilize the new system. The deliverable is a successfully implemented Finance module.

Finally, the Stabilization and Production Support phase ensures the system operates smoothly post-implementation and provides ongoing support. The system performance is monitored, any issues that arise are addressed and user feedback is gathered for continuous improvement. Necessary updates and enhancements are implemented to keep the system aligned with AK Maju's evolving needs.

3.4 Project Planning Schedule

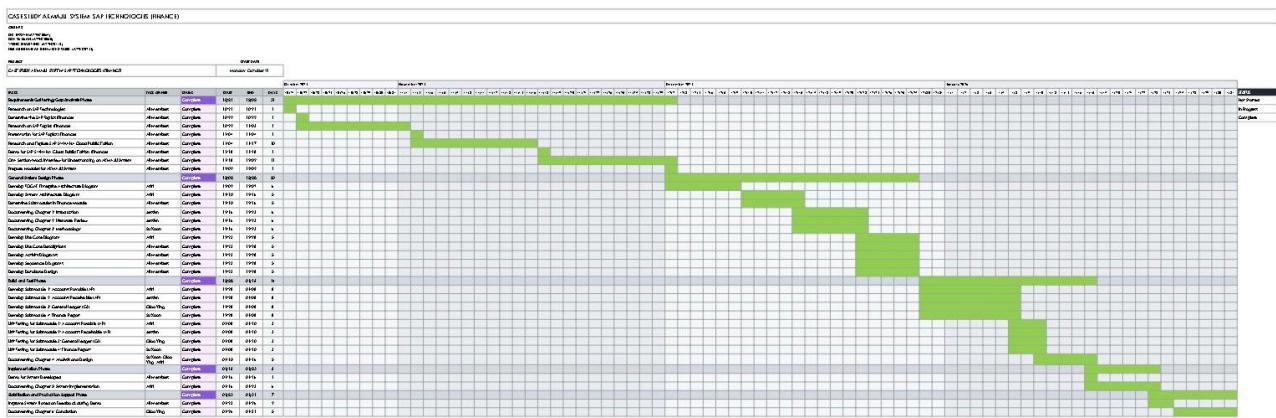


Figure 3.4.1 - Gantt Chart for Project Planning Schedule

Detailed View: + 3.4 Project Planning Schedule

3.5 Summary

This chapter presented the project's methodology, which combines structured practices to ensure the development of the AK Maju Finance module is both systematic and aligned with organizational needs. By adhering to the ERP Life Cycle Implementation methodology, the project is positioned to deliver a Finance module that is not only functional and meets user requirements but also enhances AK Maju's operational performance. The structured approach outlined in this chapter provides a clear roadmap for the project, ensuring that all stakeholders are aligned and that the project is executed efficiently and effectively.

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Introduction

This chapter focuses on the analysis and design of the proposed Finance module for AK Maju. It begins by examining the company's organizational structure to provide context for the system's implementation. A detailed analysis of the current manual system is conducted to identify inefficiencies and areas for improvement. The chapter then compares the existing processes with the proposed automated system, highlighting expected benefits. System requirements are gathered through various techniques, including interviews, to ensure alignment with AK Maju's needs. Finally, the chapter presents a comprehensive system design, detailing architecture, components, and database design, to deliver a robust and efficient solution.

4.2 Company Organization Structure - AK Maju



Figure 4.2.1 - Company Organization Structure of AK Maju Resources Sdn. Bhd.

AK Maju Resources Sdn. Bhd. is organized to efficiently manage its diverse service offerings, including design signboards, printing, advertising and more. The company is led by Encik Noor Azam Khalid and comprises approximately 10 staff members. The organizational structure includes three administrators who oversee project coordination and ensure effective communication across departments. This setup supports AK Maju's operations in Segamat, Johor, and extends its services to regions such as Johor Bahru, Melaka, Negeri Sembilan, and Selangor. The structure is designed to facilitate seamless service delivery and maintain high standards of quality across all operational areas.

4.3 Current System Analysis

Our analysis reveals that AK Maju's current Finance system is heavily reliant on manual processes, utilizing spreadsheets and physical documentation. While this approach was adequate for a smaller organization, it now presents significant limitations as the company expands and diversifies its operations.

1. Manual Data Entry

Financial data, including invoices, receipts, and sales figures, are manually entered into spreadsheets. This process is not only time-consuming but also prone to errors and inconsistencies, leading to potential delays in order processing, quotation generation, and inventory management.

2. Fragmented Systems

Key financial processes, such as accounts payable, accounts receivable, and reporting, operate independently without integration. This fragmentation results in data silos, hindering real-time cash flow tracking and affecting decision-making and customer satisfaction.

3. Limited Reporting Capabilities

The creation of financial reports is a manual and labor-intensive task, limiting the ability to access information instantly. This hampers real-time performance monitoring and informed decision-making, impacting the company's ability to respond swiftly to market changes.

4. Inefficient Payment Processing

Payment and reconciliation processes are slow, often causing delays in both receiving and making payments. This inefficiency affects cash flow management and overall financial health.

5. Lack of Automation

The absence of an integrated system and automation necessitates a more streamlined approach to handle financial operations efficiently. Improvements are needed to enhance accuracy, streamline workflows, and improve reporting capabilities.

Overall, AK Maju's existing system is inadequate to support its growing business portfolio. Implementing a modern system capable of managing complex financial data and transactions while providing real-time financial visibility is crucial for the company's future growth. The proposed system aims to reduce manual effort and minimizing errors, thereby enhancing operational efficiency and customer service.

4.4 Comparison between Existing System and Proposed System

The proposed Finance module, using SAP HANA and SAP BTP will provide several advantages compared to the existing system:

Table 4.4.1 - Advantages of Proposed System

Feature	Existing System	Proposed System (SAP HANA and SAP BTP)
Data Entry	Data entry is done manually by users which could raise human error such as wrong, duplicate, or missing data. This process requires more time and additional effort to correct them.	Automated data capture systems made possible by making use of technologies like optical character recognition (OCR) and intelligent input fields. These solutions save time and increase production by lowering the possibility of errors and guaranteeing data correctness.
System Integration	Departments operate in silos hence no data integration thus leading to slow and inefficient data exchange and delays.	Provide real-time data processing, allowing users to manage daily operations instantly with just a few clicks.
Reporting	Reports are generated manually using Microsoft Excel which require manually data collection and validation from different sources. This leads to delayed and outdated issued reports.	Provide real-time reporting through advanced analytics and dashboards. User has access to up-to-date and precise data
Payment Processing	Manual and time consuming Require multiple manual steps,	Automate payment workflows such as validating invoices, approving

	including generating manual invoices, approving transactions, etc. These are very time consuming and error-prone.	transactions, and executing payments. This enhances accuracy and shortens delays, thus leading to more efficient payment processing.
Automation	Data entry, approvals, and reporting are handled manually largely resulting in higher operating expenses.	High degree of automation for daily and repetitive tasks help to increase productivity, and decrease human intervention, thus freeing up employees for higher-value activities.
Real Time	Since real-time data is not supported, accessing and analyzing the latest data is challenging.	Since real-time data is supported, users can generate the latest up-to-date report instantly to monitor key metrics.

As a result, The new system will help AK Maju to become more efficient and reliable in their financial reporting.

4.5 System Requirements Gathering Techniques

In order to guarantee that the new finance module completely meets AK Maju's requirements, we used a two approach to gather comprehensive system requirements:

1. Interviews: To understand the requirements, expectations, and difficulties that key stakeholders—such as management, accounting personnel, and other users—had with the current procedures, we conducted in-depth interviews with them. The group employed semi-structured interviews, which allowed for flexibility in the conversation while still staying within predetermined subject boundaries.
2. Document Analysis: In order to completely comprehend the current procedures and pinpoint areas in which the new system will be required, we examined all of the company's records, including financial statements, invoices, reports, and other papers.

Below is the list of questions asked in the interviews:

1. How does the process go from order receipt to product delivery?
2. Do the three different customer types—government agencies, walk-in customers, and online customers—have different workflows?
3. How much does the design concept cost?
4. Can you give the organisational chart of a company?
5. Who is the system's intended user?
6. How can we determine the material's cost?
7. Does the system have a particular theme?
8. Is the order report recorded on a weekly or monthly basis?
9. Which invoice components would you like to see included?
10. Is the cost measured on a weekly or monthly basis?

4.6 System Requirements

4.6.1 Functional Requirements

1. User Management: With secure login features, the system should be able to handle staff and administrator users, roles, and permissions.
2. Processing of Invoices: The system will automatically create invoices for various services, such as advertising, construction, and printing.
3. Payment Processing: In addition to offering payment reconciliation capabilities, the system must process payments effectively and in a variety of formats.
4. Data Entry: Able to handle a wide range of transaction types and enter data.
5. Management of the General Ledger (GL): Establish a strong system for managing the accounts and their balances.
6. Reporting and Analytics: In addition to customisable dashboards, the system must include real-time financial reports, cash flow analysis, and other financial data.
7. Audit Trail: A record of all user transactions and activities will be kept by the system.

4.6.2 Non Functional Requirements

1. Performance: All processes should respond quickly to the system's real-time updates.
2. Security: Appropriate authentication procedures and other security measures must be used by the system to safeguard financial data.
3. Scalability: As the company grows, the system should be able to accommodate more users and higher transaction volumes.
4. Availability: The system must be accessible around-the-clock, with little downtime.
5. Usability: An intuitive user interface that requires little training for users to navigate and comprehend.
6. Reliability: The system must process data accurately, consistently, and correctly each and every time.

4.7 System Design

4.7.1 Enterprise Architecture

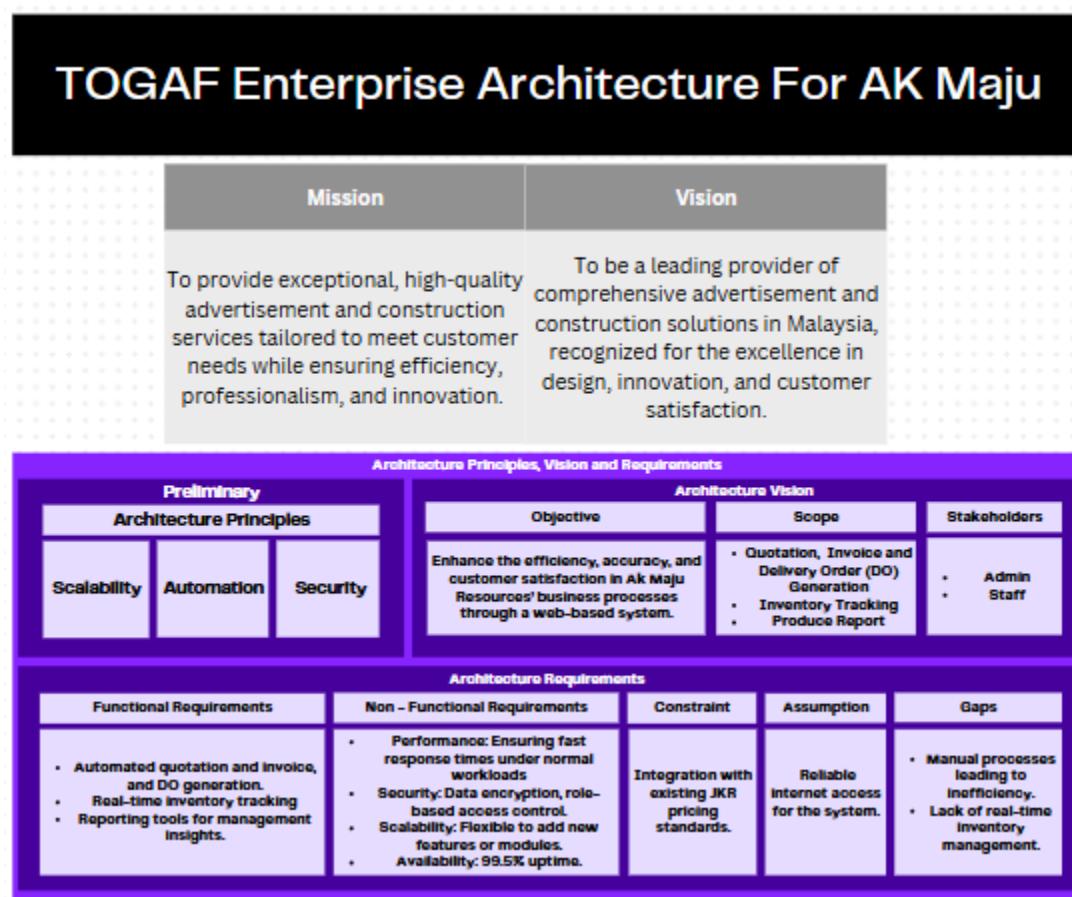


Figure 4.7.1.1 - Enterprise Architecture

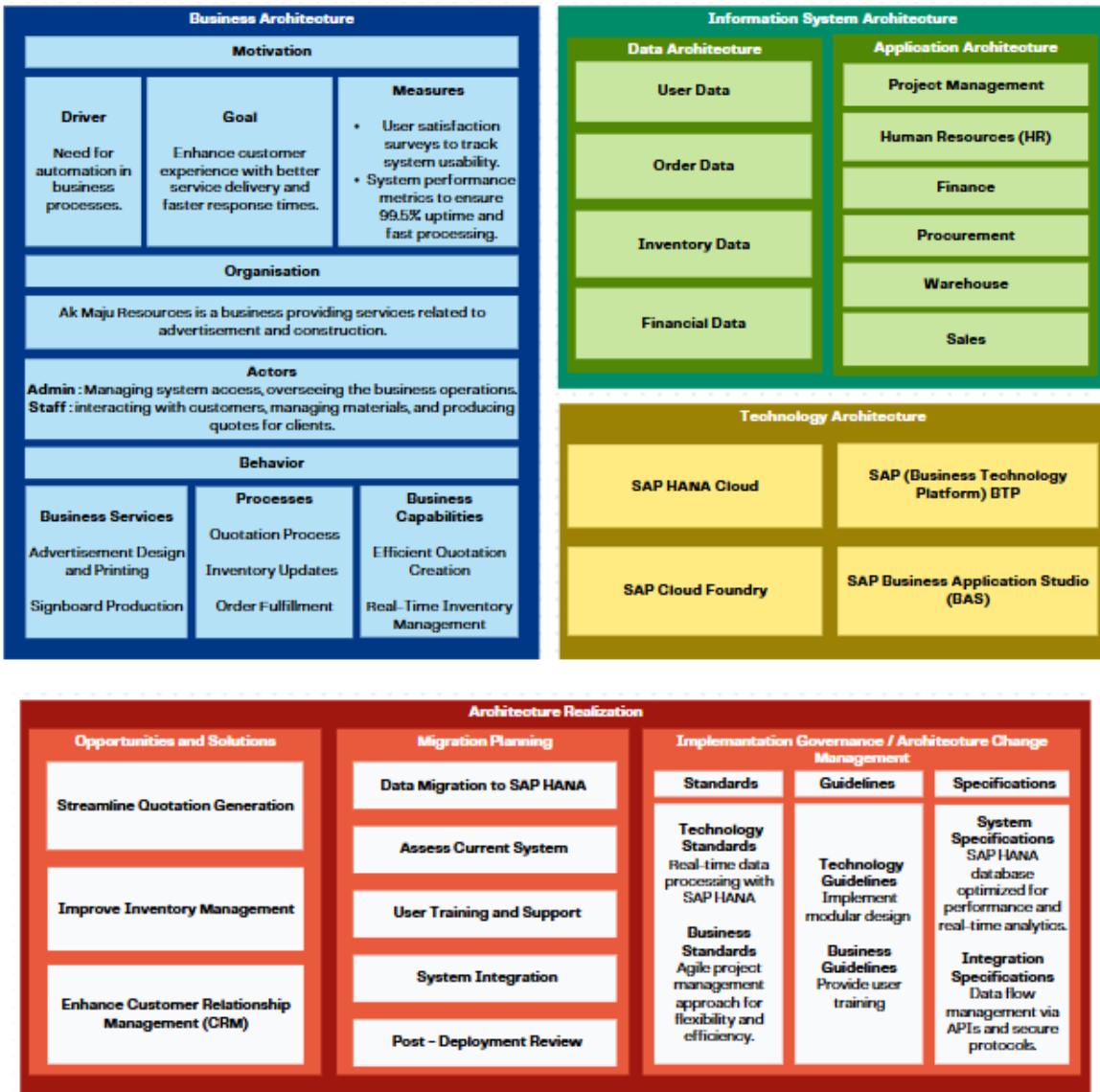


Figure 4.7.1.2 - Enterprise Architecture (continued)

The AK Maju TOGAF Enterprise Architecture (Figure 4.7.1.1 & Figure 4.7.1.2) outlines the management framework for the company's core departments. It encompasses essential areas such as project management, human resources, financial operations, procurement, warehouse logistics, and sales. This comprehensive structure ensures that all facets of the business are interconnected, with defined processes and technology to support each function. The architecture fosters collaboration between departments, streamlines operations, and provides a blueprint for future development. By implementing this EA, AK Maju can ensure that its IT systems align with business strategies, enabling better decision-making and resource management.

4.7.2 Explanation of Each Component in Enterprise Architecture

Table 4.7.2.1 - Components in Enterprise Architecture

Components	Explanation
Mission (Figure 4.7.1.1)	The mission of AK Maju is to deliver top-notch advertisement and construction services that address client specific preferences. They show their commitment to high quality service and customer satisfaction across all projects.
Vision (Figure 4.7.1.1)	AK Maju envisions becoming a leading provider of advertisement and construction solutions in Malaysia. This vision reflects their direction to build their industry standards and build trust with their customers.
Architecture Principle (Figure 4.7.1.1)	<p>The architecture principles form the foundation for designing the enterprise system.</p> <ul style="list-style-type: none"> ● Scalability: The system is built to support additional business requirements as the company expands and will need to handle more customers. ● Automation: The new system automates non-human tasks to save time while making fewer mistakes. ● Security: The data is protected while preserving secure transactions to gain customer trust.
Architecture Vision (Figure 4.7.1.1)	<p>This part describes the major goals and scope of the system.</p> <ul style="list-style-type: none"> ● Objective: The system aims to improve the way AK Maju's resources manage their business processes through a web-based platform. ● Scope: The system manages AK Maju's business tasks which are generating quotations, invoices, and delivery

	<p>orders, as well as inventory tracking and creating reports.</p> <ul style="list-style-type: none"> ● Stakeholders: Admin and staff will use this system to complete their daily tasks easier than the previous manual system.
Architecture Requirements (Figure 4.7.1.1)	<p>The requirements describe what the system needs to achieve.</p> <ol style="list-style-type: none"> Functional Requirements: <ul style="list-style-type: none"> ○ Automate manual tasks which generate quotations, invoices, and delivery orders. ○ Assure instant access to updated inventory. ○ Provide reporting solutions to help the company understand how well their business performed. Non-Functional Requirements: <ul style="list-style-type: none"> ○ Performance: Ensure the system performs well, even during peak usage. ○ Security: Protect data from unauthorized access and ensure safe transactions. ○ Reliability: Maintain high availability, with a 99.5% uptime goal. Constraint: <ul style="list-style-type: none"> ○ The system must integrate with existing JKR (Jabatan Kerja Raya) standards to meet compliance requirements. Assumption: <ul style="list-style-type: none"> ○ Reliable internet access will be available for the system to function smoothly. Gaps: <ul style="list-style-type: none"> ○ Current manual processes are slow and prone to errors, which the system aims to eliminate.

Business Architecture (Figure 4.7.1.2)	
Motivation	<p>Driver: AK Maju recognizes the need to simplify and speed up their business processes. Automation is key to reducing repetitive tasks and avoiding delays.</p> <p>Goal: The company wants to improve how customers experience their services. By making their processes smoother and faster, they aim to keep customers happy and provide services more quickly.</p> <p>Measures: They plan to track progress by using customer feedback, like satisfaction surveys, to check if the system is meeting their expectations. Additionally, they will monitor technical performance to make sure the system stays reliable with minimal downtime.</p>
Organisation	<p>Overview: AK Maju is a company that provides services such as advertisement and construction. Their operations include creating advertisement designs, printing, and manufacturing items like signboards.</p>
Actors	<p>Admin: The admin is responsible for managing the system, making sure everything runs smoothly, and overseeing the company's operations.</p> <p>Staff: The staff handle customer interactions, update inventory, and create quotes for clients while managing materials needed for production.</p>
Behaviour	<p>Business Services: The company provides services that revolve around creating and</p>

	<p>delivering advertisements, printing materials, and producing signboards tailored to client needs.</p> <p>Processes:</p> <p>These are the tasks involved in their daily operations, which include:</p> <ul style="list-style-type: none"> • Managing the quotation process for customers. • Updating inventory levels to keep track of materials. • Fulfilling customer orders efficiently. <p>Business Capabilities:</p> <p>To achieve their goals, AK Maju focuses on:</p> <ul style="list-style-type: none"> • Creating quotes for clients quickly and accurately. • Keeping their inventory records up to date in real time to avoid any shortages. • Managing their resources and processes in a way that ensures everything flows smoothly.
--	---

Data Architecture (Figure 4.7.1.2)

User Data	This refers to all the information about the people using the system. It includes details like login credentials, contact information, and access levels. For AK Maju, this could mean records for admins and staff managing the system.
Order Data	This covers all the data related to customer orders. It keeps track of quotations, invoices, delivery details, and order statuses. This ensures the company has a clear picture of what's been requested and completed.
Inventory Data	This part focuses on tracking materials and stock levels. It helps the team know what's available and what needs restocking. By

	keeping this information up to date, AK Maju can avoid delays caused by material shortages.
Financial Data	This includes all the monetary records, such as payments received, expenses, account receivables, and account payables. It's the backbone of the financial system, ensuring the company's finances are accurate and easy to manage.
Application Architecture (Figure 4.7.1.2)	
Project Management	This application helps the team organize and keep track of their projects. It's used to set deadlines, assign tasks, and monitor progress. For AK Maju, this ensures their advertisement and construction projects run smoothly without delays or confusion.
Human Resources	This part is all about managing employee-related information and processes. It includes tracking staff details, managing leave requests, and handling payroll. This system helps AK Maju ensure their employees are supported and administrative tasks are handled efficiently.
Finance	The finance managing everything related to money. It helps track invoices, payments, account receivables, and account payables. This ensures AK Maju has a clear picture of their financial health and can manage budgets effectively.
Procurement	This component focuses on purchasing materials or supplies needed for projects. It helps streamline the ordering process, track supplier information, and ensure the right materials are bought at the right time. For AK Maju, this is essential for keeping projects on schedule.
Warehouse	The warehouse application keeps track of inventory levels and

	materials stored. It ensures that the company knows what they have in stock, what's been used, and what needs restocking. This prevents delays caused by running out of materials during projects.
Sales	The sales application is used to manage customer interactions, track orders, and generate quotations. It helps the team provide accurate pricing and maintain a record of all client transactions. For AK Maju, this ensures customer needs are met efficiently and professionally.
Technology Architecture (Figure 4.7.1.2)	
SAP HANA Cloud	SAP HANA Cloud serves as the backbone of the system, acting as the database where all critical information—like financial transactions, customer data, and inventory—are stored and managed. It's designed to handle large amounts of data quickly, ensuring that users can access the information they need without delays. This platform also simplifies database management by allowing updates and queries to be performed in real time, which is essential for accurate reporting and decision-making.
SAP BTP	SAP BTP is like the foundation that supports the entire system. It provides the tools and services needed to connect different parts of the system, such as databases, applications, and APIs. With SAP BTP, the system can operate smoothly across various environments, offering flexibility for scaling or adding new features in the future. For example, it enables integration between the financial modules, ensuring seamless communication between account receivable, account payable, and reporting functionalities.
SAP Cloud Foundry	Cloud Foundry is the environment where the applications are

	deployed and run. By running on a cloud platform, Cloud Foundry allows the system to remain accessible anytime, anywhere, which is particularly useful for remote work or on-the-go users. It also ensures that updates can be rolled out without disrupting the system's operations.
SAP BAS	SAP BAS is the tool used to build and customize the system. This development environment simplifies the process of creating user-friendly interfaces and ensures that all components are well-integrated with SAP services. With its easy-to-use features, SAP BAS helps developers focus on building a system that meets the needs of the users while saving time on repetitive tasks.
Architecture Realization (Figure 4.7.1.2)	
Opportunities Solutions	<p>This section focuses on the practical benefits the system will bring to AK Maju and how the identified solutions address their challenges:</p> <ol style="list-style-type: none"> Streamline Quotation Generation Generating quotations manually can be time-consuming and error-prone. With the new system, this process becomes automated, ensuring accuracy and saving time. This allows staff to focus on more important tasks, like customer engagement and project management. Improve Inventory Management Keeping track of inventory manually can lead to inefficiencies like overstocking or shortages. The system introduces real-time inventory tracking, making it easier to monitor stock levels, reduce waste, and ensure materials are always available when needed.

	<p>3. Enhance Customer Relationship Management (CRM) By integrating a CRM feature, AK Maju can better manage interactions with clients, ensuring their needs are met efficiently. This includes tracking past orders, managing customer feedback, and building stronger relationships through personalized service.</p>
Migration Planning	<p>To transition to the new system smoothly, specific steps are needed:</p> <ol style="list-style-type: none"> Data Migration to SAP HANA Existing data—like customer information, invoices, and inventory details—will be transferred to the SAP HANA database. This ensures all historical data is available in the new system without starting from scratch. Assess Current System Before migrating, the current system is carefully evaluated to identify gaps, inefficiencies, and areas for improvement. This ensures the new solution addresses all the pain points. User Training and Support Employees will receive hands-on training to familiarize themselves with the system. Ongoing support will also be provided to address any issues or questions, helping users feel confident in using the new tools. System Integration and Post-Deployment Review Once the system is live, it will be integrated with existing tools to ensure a seamless flow of information. A review will follow after deployment to fix any glitches, collect user feedback, and make improvements where needed.

<p>Implementation Governance / Architecture Change Management</p>	<p>To ensure everything runs smoothly and stays on track, the project is guided by specific standards, guidelines, and specifications:</p> <ol style="list-style-type: none"> 1. Standards <ul style="list-style-type: none"> ○ Technology Standards: Real-time data processing with SAP HANA ensures that information is always up-to-date and accessible. ○ Business Standards: An agile project management approach is followed, allowing flexibility to adapt to changes and ensuring the project meets deadlines effectively. 2. Guidelines <ul style="list-style-type: none"> ○ Technology Guidelines: A modular design is implemented to make the system more adaptable and scalable in the future. ○ Business Guidelines: User training is emphasized, ensuring that employees can use the system confidently and independently. 3. Specifications <ul style="list-style-type: none"> ○ System Specifications: SAP HANA is optimized for high performance and real-time analytics, making it ideal for AK Maju's needs. ○ Integration Specifications: APIs and secure protocols are used to manage data flow between components, ensuring that all parts of the system communicate effectively and securely.
---	---

4.7.3 System Architecture

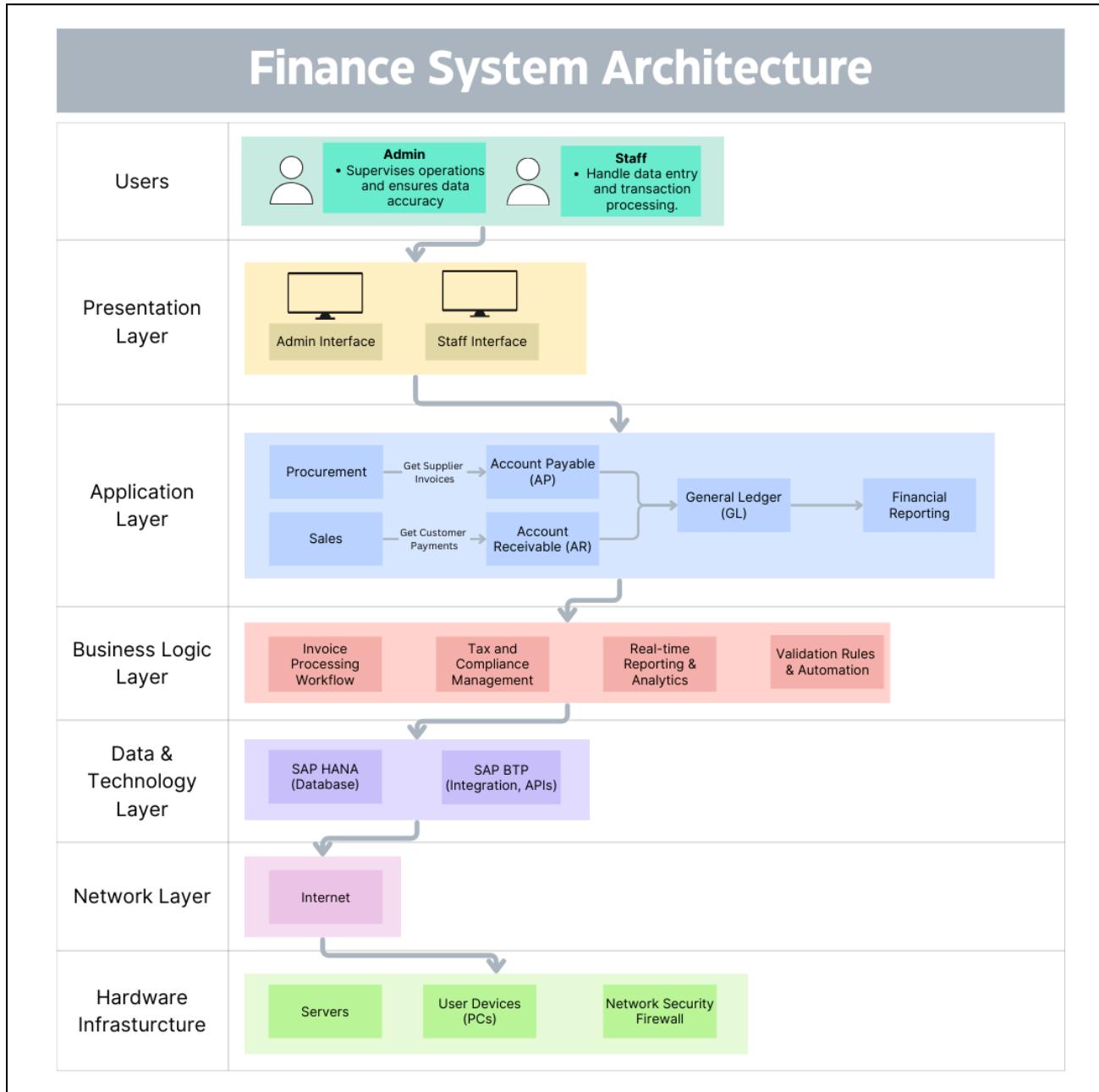


Figure 4.7.3.1 - System Architecture Diagram for Finance Module in AK Maju System

The System Architecture (Figure 4.7.3.1) focuses on the financial subsystem of AK Maju. This architecture is designed to manage and streamline financial operations by incorporating four key modules, which are Account Receivable (AR), Account Payable (AP), General Ledger (GL), and Finance Reports. Each module addresses specific aspects of financial management. AR tracks incoming payments and outstanding invoices, while AP manages outgoing payments and vendor interactions. The GL acts as the central repository for all financial transactions, ensuring accuracy and compliance. Finally, Finance Reports provide actionable insights and summaries of financial performance. This structured system architecture ensures efficient workflows, robust data management, and comprehensive financial oversight.

4.7.4 Explanation of Each Component in System Architecture

Table 4.7.4.1 - Components in System Architecture

Component	Explanation
Users Layer	
Admin	The admin oversees operations, ensures data accuracy.
Staff	Staff members handle data entry and transaction processing
Presentation Layer	
Admin Interface	The admin interface provides access to all functionalities in the finance system.
Staff Interface	The staff interface also provides access to all functionalities in the finance system.
Application Layer	
Procurement	Manages supplier data, invoices, and purchase orders then sends them to the AP system.

AP	Receives supplier invoices and handles payment to supplier.
Sales	Manages all sales data, customer information, and sales payments then sends them to the AR system.
AR	Receives invoices from the sales department and manages customer payments.
GL	Core of accounting, responsible for tracking all financial transactions.
Financial Reporting	For generating detailed financial reports.
Business Logic Layer	
Invoice Processing Workflow	Manages the processing of invoices from generation to payment.
Tax and Compliance Management	Ensure the system follows tax and legal regulations.
Real-Time Reporting and Analytics	Provides real-time financial insights through dashboards.
Validation Rules and Automation	Handles data validation, workflow automation and other similar requirements.
Data & Technology Layer	
SAP HANA (Database)	In-memory database system that will hold the data and processing of financial data.
SAP BTP (Integration, APIs)	Provides the platform for seamless integration of various modules and applications, creating APIs and ensuring scalability.
Network Layer	
Internet	The system will be accessible over the internet and will need to adhere

	to security measures.
Hardware Infrastructure	
Servers	Physical or cloud based servers for system and data storage.
User Devices	Personal computers or other devices used by AK Maju staff.
Network Security Firewalls	To protect the server and network.

4.7.5 Project Design

4.7.5.1 Use Case Diagram for Enterprise System

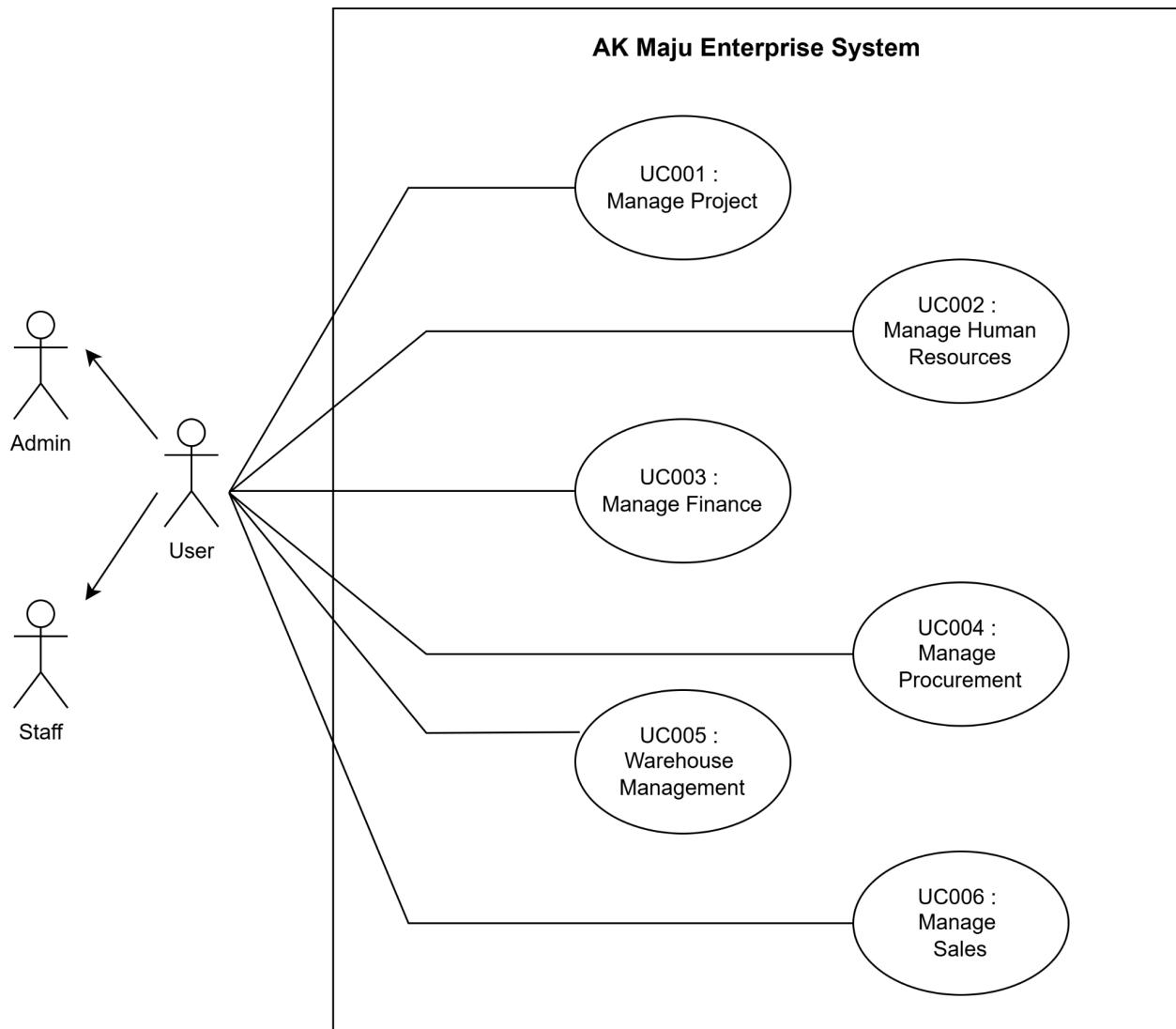


Figure 4.7.5.1.1 - Use Case Diagram for AK Maju Enterprise System

4.7.5.2 Use Case Diagram for SubSystem

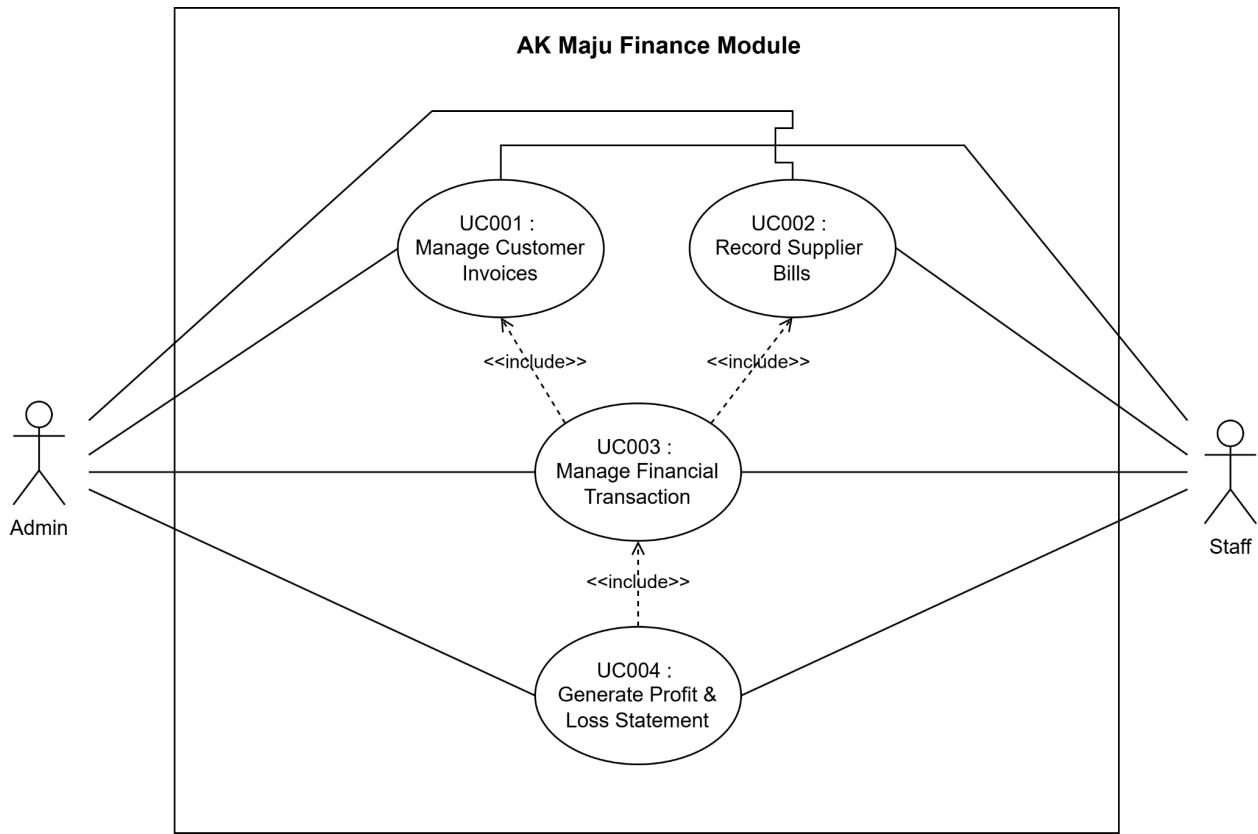


Figure 4.7.5.2.1 - Use Case Diagram for Finance Module in AK Maju Enterprise System

4.7.5.3 Use Case Description, Activity Diagram and Sequence Diagram

4.7.5.3.1 UC001: Manage Customer Invoices

Table 4.7.5.3.1.1 - Use Case Description for AR Subsystem -

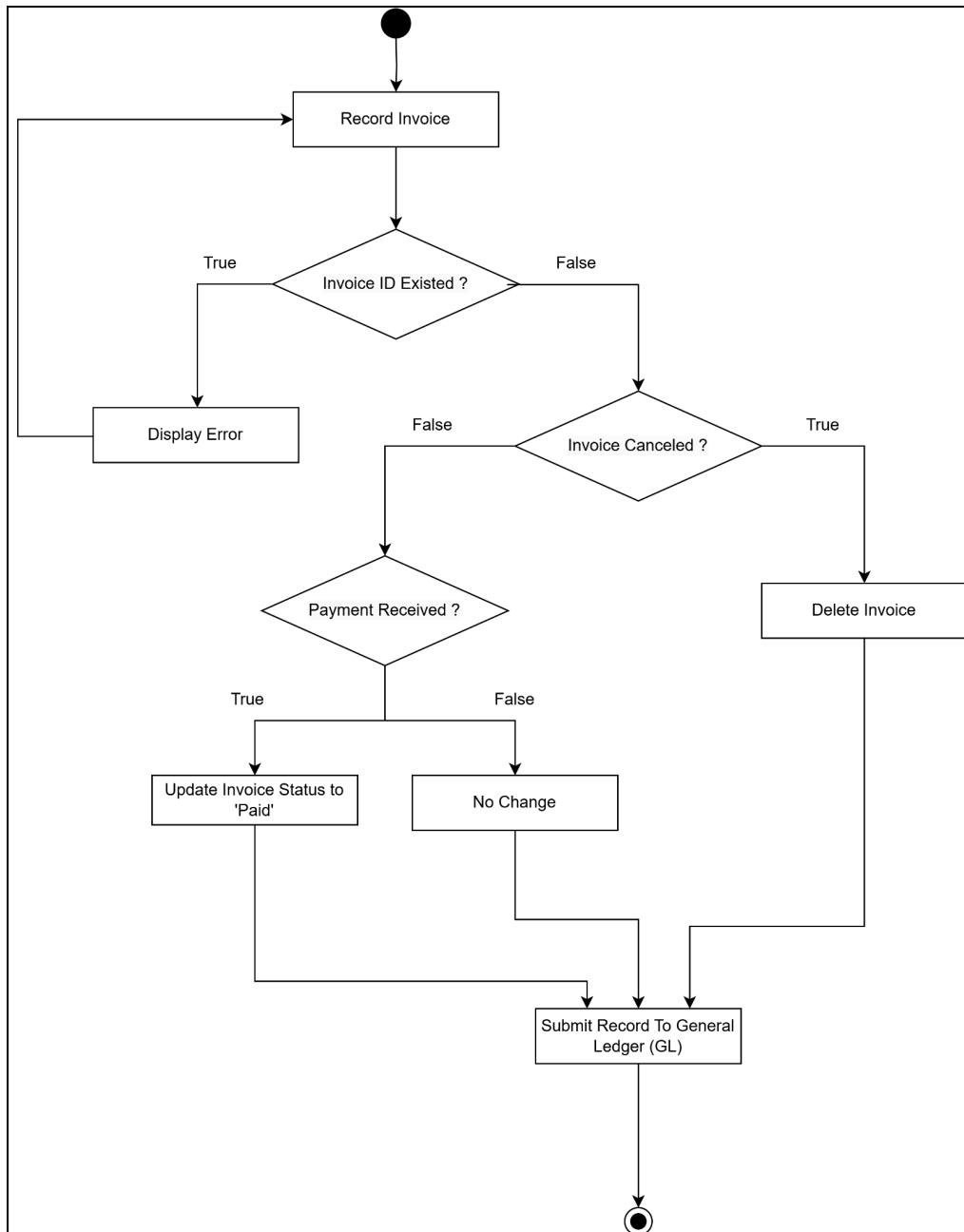
UC001: Manage Customer Invoices

Use case: Manage Customer Invoices	
ID: UC001	
Actors:	<ol style="list-style-type: none">1. Admin2. Staff
Preconditions:	<ul style="list-style-type: none">• The admin/staff must be logged into the system with appropriate permissions.
Flow of events:	<ol style="list-style-type: none">1. Admin/Staff enters invoice details into the system.2. The system validates the input and saves the invoice information into the database.3. Admin/Staff selects an existing invoice to delete.4. Once deleted, the system removes the invoice from the database.5. Admin/Staff selects an invoice marked as "Pending" to update its payment status.6. The Admin/Staff specifies the new status as either "Fully Paid" or "Not Fully Paid" based on the customer's payment.7. The system saves the updated status in the database.8. Admin/Staff generates a daily account receivable report summarizing invoice records and payment statuses to submit to the General Ledger module for further processing.
Exception flow:	E1. Error During Invoice Recording (branches from step 1) <ol style="list-style-type: none">1. The system identifies that the entered invoice ID matches an existing record in the database.

- | |
|---|
| <ol style="list-style-type: none">2. The system notifies the Admin/Staff with a message: "<i>Invoice ID already exists.</i>"3. Return to step 1. |
|---|

Postconditions:

- New invoices are stored in the database and visible in the system.
- Deleted invoices are removed from the database, and the invoice list reflects the changes.
- Updated invoice statuses are correctly stored in the database and displayed in the system.



**Figure 4.7.5.3.1.1 - Activity Diagram for AR SubSystem -
UC001: Manage Customer Invoices**

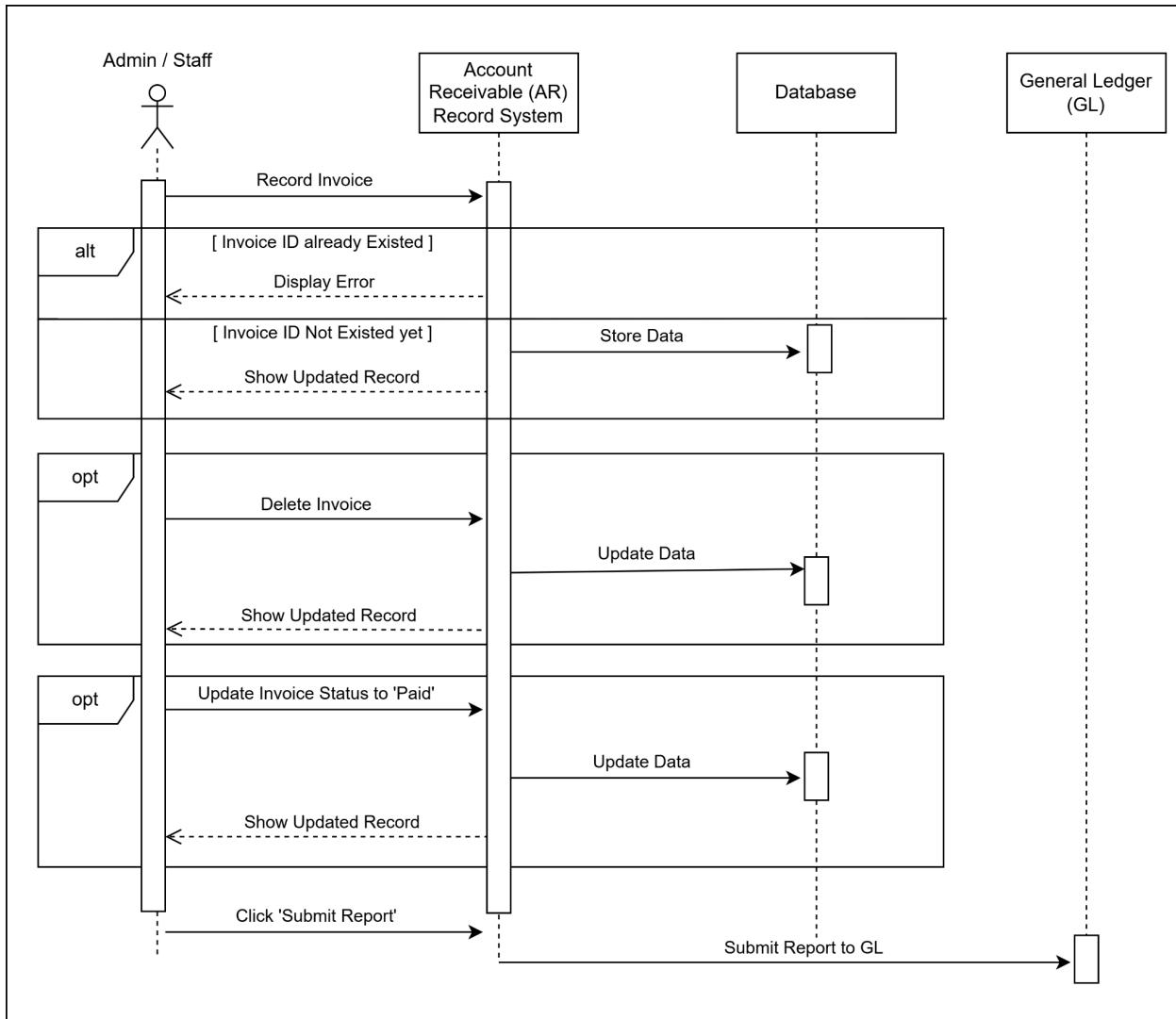


Figure 4.7.5.3.1.2 - Sequence Diagram for Account Receivable (AR) SubSystem - UC001: Manage Customer Invoices

4.7.5.3.2 UC002: Record Supplier Bills

Table 4.7.5.3.2.1 - Use Case Description for AP Subsystem -

UC002: Record Supplier Bills

Use case: Record Supplier Bills
ID: UC002
Actors: 1. Admin 2. Staff
Preconditions: <ul style="list-style-type: none">● The admin/staff must be logged into the system with appropriate permissions.
Flow of events: <ol style="list-style-type: none">1. Admin/Staff enter the supplier details at the supplier page (eg. supplier name, supplier contact information, and more related information) of the system. The supplier id representing each supplier will be generated automatically upon creating the supplier.2. The created supplier and related details will be saved into the database and displayed in the supplier display list.3. Admin/Staff enter the invoice details at the invoice page (eg. supplier name selected from the supplier drop down menu, invoice amount, payment due date, and invoice status either paid, pending or overdue) of the system. The invoice id representing each invoice will be generated automatically upon creating the invoice.4. The created invoice and related details will be saved into the database and displayed in the invoice display list.5. Admin/Staff enter the payment details for the corresponding invoice at the payment page (eg. invoice id selected from the invoice drop down menu, payment amount, and payment date) of the system. The payment id representing each payment will be generated automatically upon creating the payment.6. The created payment and related details will be saved into the database and displayed in the payment display list.

7. In the supplier page, admin/staff can delete any unwanted supplier from the display list by clicking the button “Delete”.
8. A message will prompt out to confirm the actor’s action on deleting the supplier and then it will update the database by removing the supplier that had been deleted.
9. This delete function is the same goes to invoice and payment page where admin/staff can delete unwanted invoice as well as payment.
10. In the supplier page, admin/staff can edit any selected supplier’s information by clicking the button “Edit” at the particular supplier page.
11. After editing the information, admin/staff click on the “Save” button then the modified information will be updated in the supplier display list as well as in the database.
12. This edit function is the same goes to invoice and payment page where admin/staff can edit the selected invoice as well as payment.
13. At the supplier, invoice and payment page, admin/staff can export the list to excel file by clicking the “Export” button.

Exception flow:

1. If admin/staff selected the invoice with status “PAID” or “OVERDUE”, an error message will prompt out saying that only invoice with status “PENDING” or “PENDING (HALF)” can be selected to make payment.
2. If the admin/staff entered the payment amount that is more than the amount entered when creating the invoice previously, an error message will prompt out saying that the entered amount had exceeded the amount entered in the invoice.
3. When creating a payment, if admin/staff did not make full payment amount for that particular invoice, a new invoice will be created with the leftover amount and the status is “PENDING (HALF)”. This created invoice will be added in the database and also in the invoice display list.

Postconditions:

- Created supplier, invoice and payment are added into the display list and database.
- Deleted supplier, invoice and payment are removed from the display list and database.
- Edited supplier, invoice and payment information are updated in the display list and database.

- After making full payment for that particular invoice, the status of that invoice will be changed into “PAID” and updated in the invoice display list as well as database.
- An excel file is preparing to be downloaded upon clicking the export button at the supplier, invoice or payment page.

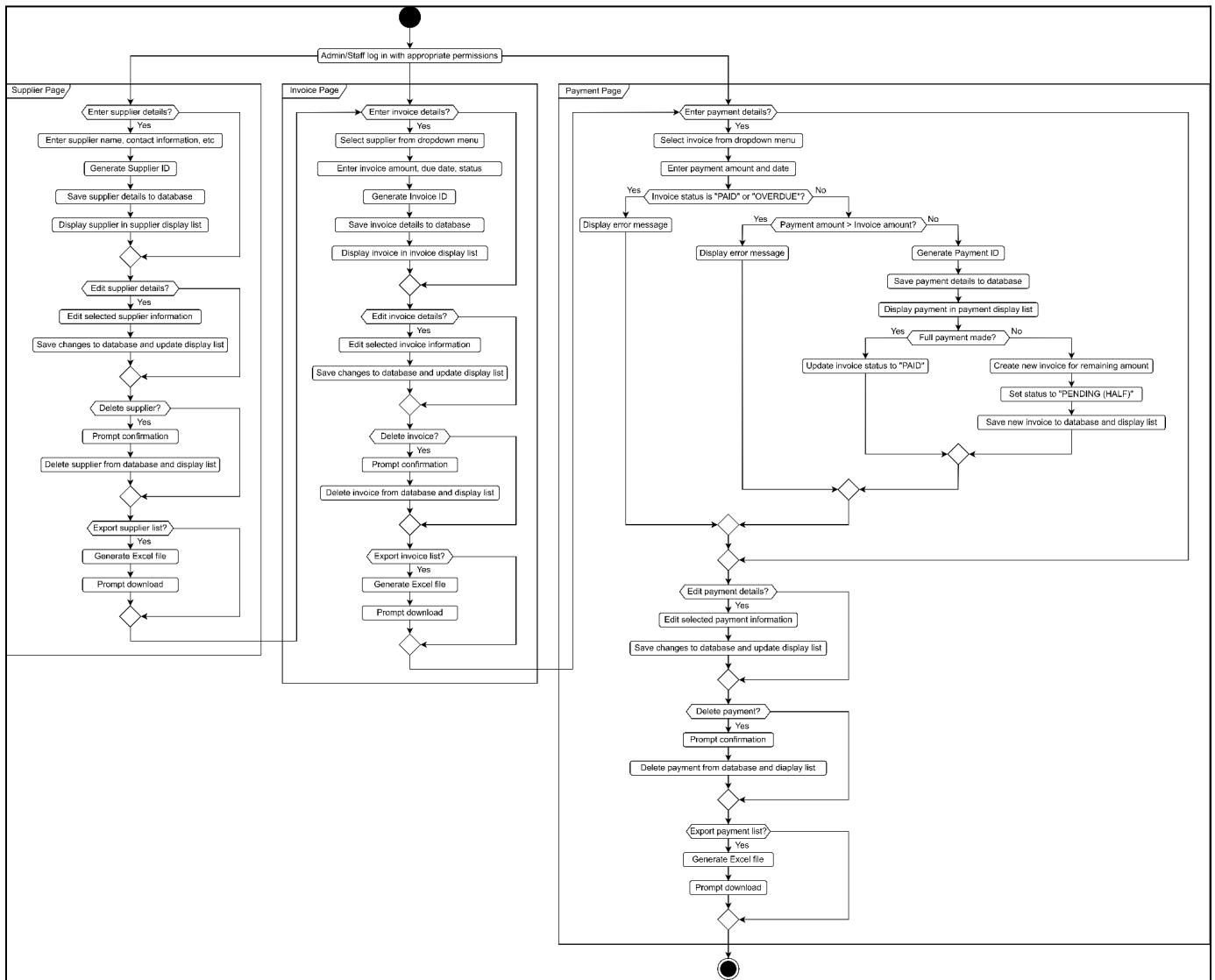


Figure 4.7.5.3.2.1 - Activity Diagram for Accounts Payable (AP) SubSystem - UC002: Record Supplier Bills

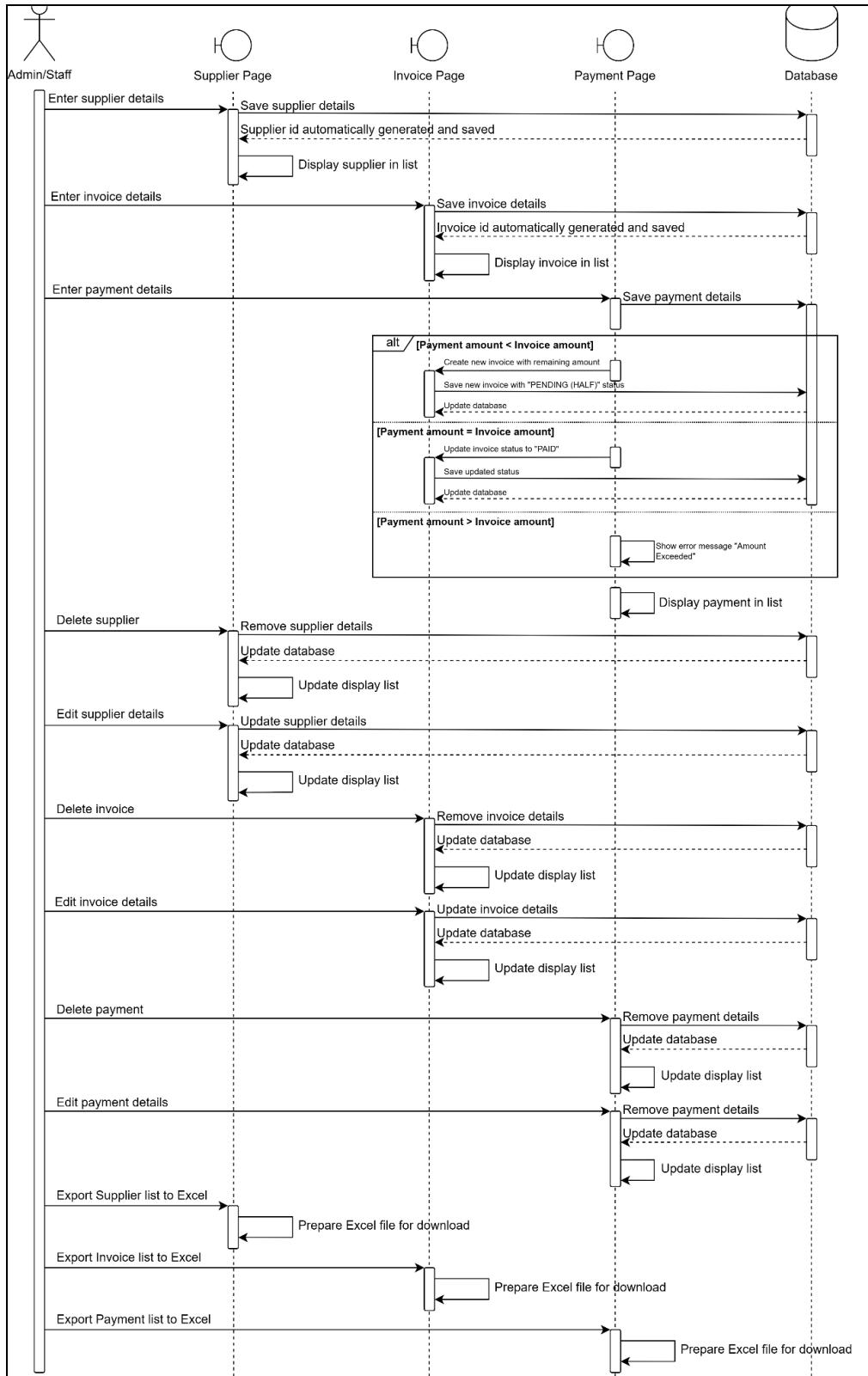


Figure 4.7.5.3.2.2 - Sequence Diagram for Accounts Payable (AP) SubSystem - UC002: Record Supplier Bills

4.7.5.3.3 UC003 : Manage Financial Transactions

Table 4.7.5.3.3.1 - Use Case Description for GL Subsystem -

UC003 : Manage Financial Transactions

Use case: Manage Financial Transactions
ID: UC003
Actors:
1. Admin
Preconditions:
<ul style="list-style-type: none">● The admin must be logged into the system with appropriate permissions.● The transaction data for the specified month and year is available and stored in the SAP HANA database.● SAP BTP is connected to the SAP HANA database.● The General Ledger template and chart of accounts are predefined.
Flow of events:
<ol style="list-style-type: none">1. The Admin logs into the SAP system and enters the year and month for the ledger generation.2. The system queries the SAP HANA database for transaction data corresponding to the specified year and month.3. The system checks the retrieved data for completeness and consistency (e.g., missing or invalid entries).4. If validation fails, an error message is displayed to the Admin for corrective action.5. The system processes the validated data, categorizing transactions based on the predefined GL chart of accounts.6. A structured General Ledger for the specified period is created.7. The system generates a PDF version of the General Ledger using the predefined report template.8. The Admin is notified of the successful ledger generation and provided with a download link for the PDF

Exception flow:

1. If no transaction data is found for the selected year and month, the system notifies the Admin and suggests verifying the input or ensuring the data exists in the SAP HANA database.
2. If data validation fails due to incomplete or inconsistent records, the system halts the process, generates an error report, and prompts the Admin to correct the data before restarting.
3. If the system cannot connect to the SAP HANA database, it notifies the Admin to check the network or database status and retry the process.
4. If the PDF ledger generation fails, the system logs the error and informs the Admin to verify the template or escalate the issue to technical support.

Postconditions:

- A PDF General Ledger for the specified month and year is generated and stored in the system.
- The PDF is ready for download or sharing.

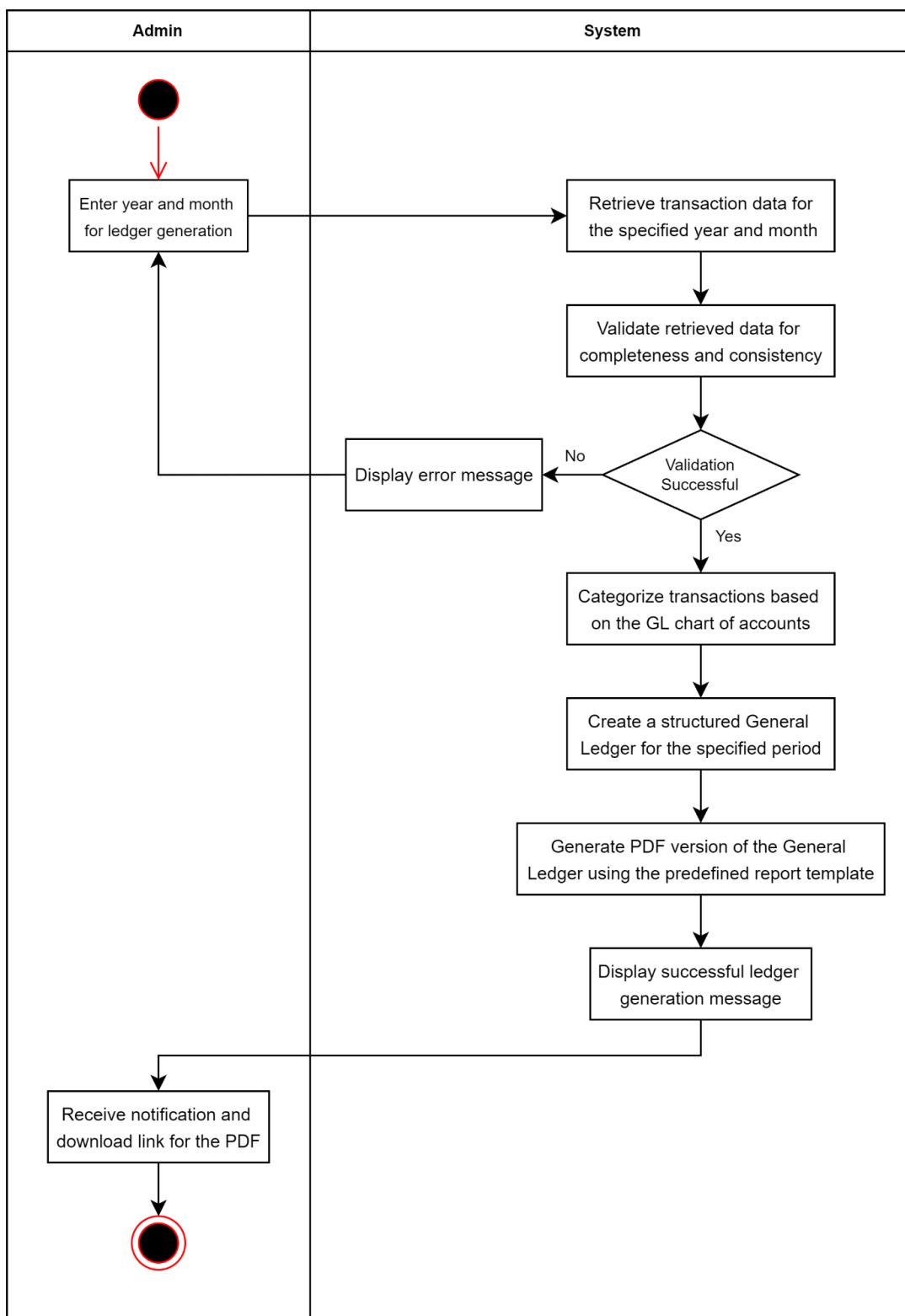


Figure 4.7.5.3.3.1 - Activity Diagram for General Ledger (GL) SubSystem - UC003: Manage Financial Transactions

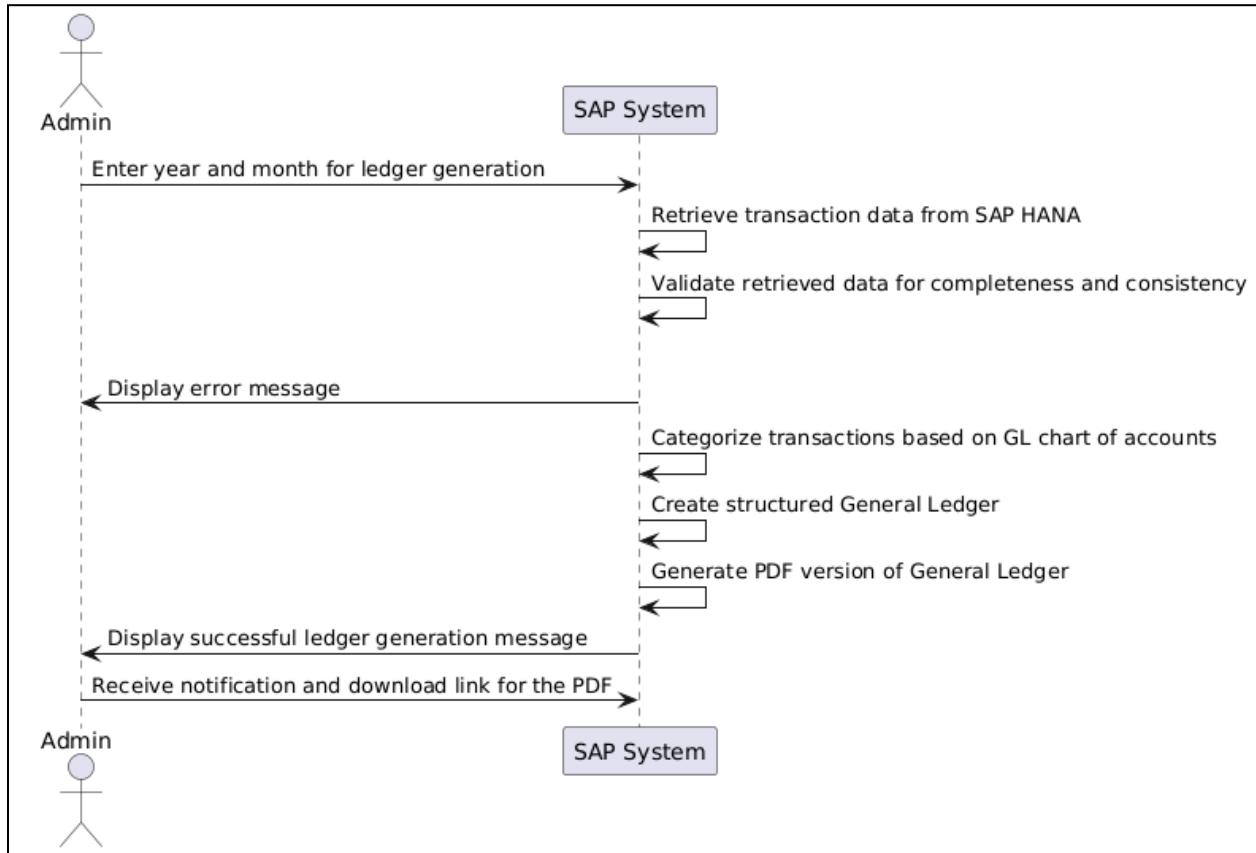


Figure 4.7.5.3.3.2 - Sequence Diagram for General Ledger (GL) SubSystem - UC003: Manage Financial Transactions

4.7.5.3.4 UC004: Generate Profit & Loss Statement

Table 4.7.5.3.4.1 - Use Case Description for Finance Report SubSystem -

UC004: Generate Profit & Loss Statement

Use case: Generate Profit & Loss Statement
ID: UC004
Actors:
<ol style="list-style-type: none"> 1. Admin 2. Staff
Preconditions:
<ul style="list-style-type: none"> • The admin/staff must be logged into the system with appropriate permissions. • Financial data must be available in the system for the selected reporting period.
Flow of events:
<ol style="list-style-type: none"> 1. The system presents options for selecting the report type. 2. The admin/staff fills in the required report type. 3. The system presents options for selecting the reporting period. 4. The admin/staff fills in the desired reporting period. 5. The admin/staff submits the selected option. 6. The system validates the entered reporting period. 7. If the reporting period is invalid: <ol style="list-style-type: none"> a. The system displays an error message. b. The admin/staff fills in the corrected reporting period. c. Return to step 5 8. If the reporting period is valid: <ol style="list-style-type: none"> a. The system retrieves the relevant financial data. b. The system calculates the total revenue, total expenses and net profit/loss. c. The system generates the profit & loss statement for the selected reporting period. 9. The admin/staff views and prints the generated statement.
Alternative flow:

A1. Cancel Operation (can occur after any step in main flow)

1. Admin/staff cancels the operation.
2. System terminates the use case.
3. Use case ends.

Exception flow:

E1. Invalid Reporting Period (branches from step 6)

1. System detects invalid reporting period.
2. System displays an error message.
3. Return to step 4

Postconditions:

- A profit & loss statement is generated and available for viewing and printing.

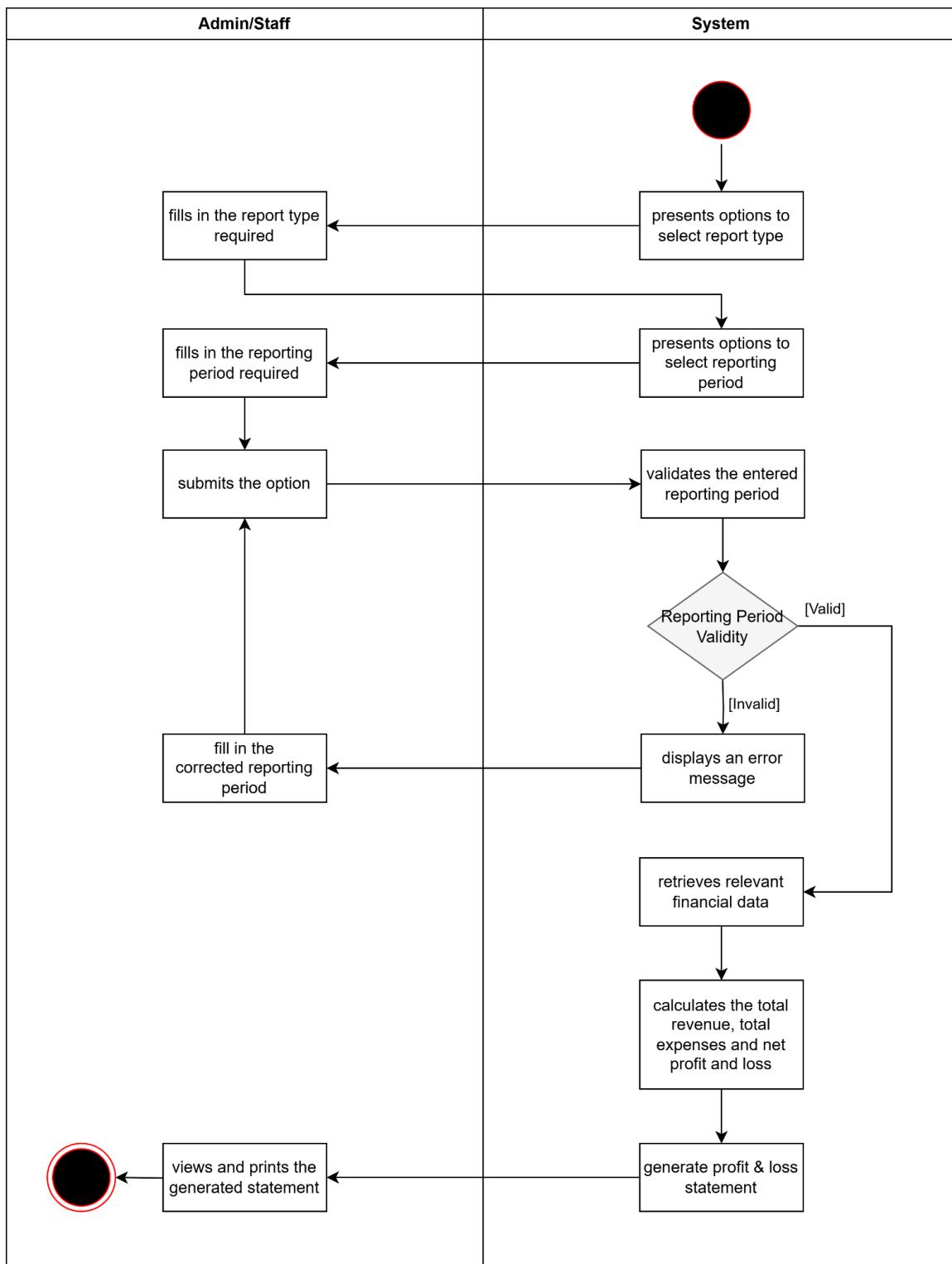


Figure 4.7.5.3.4.1 - Activity Diagram for Finance Report SubSystem - UC004: Generate Profit & Loss Statement

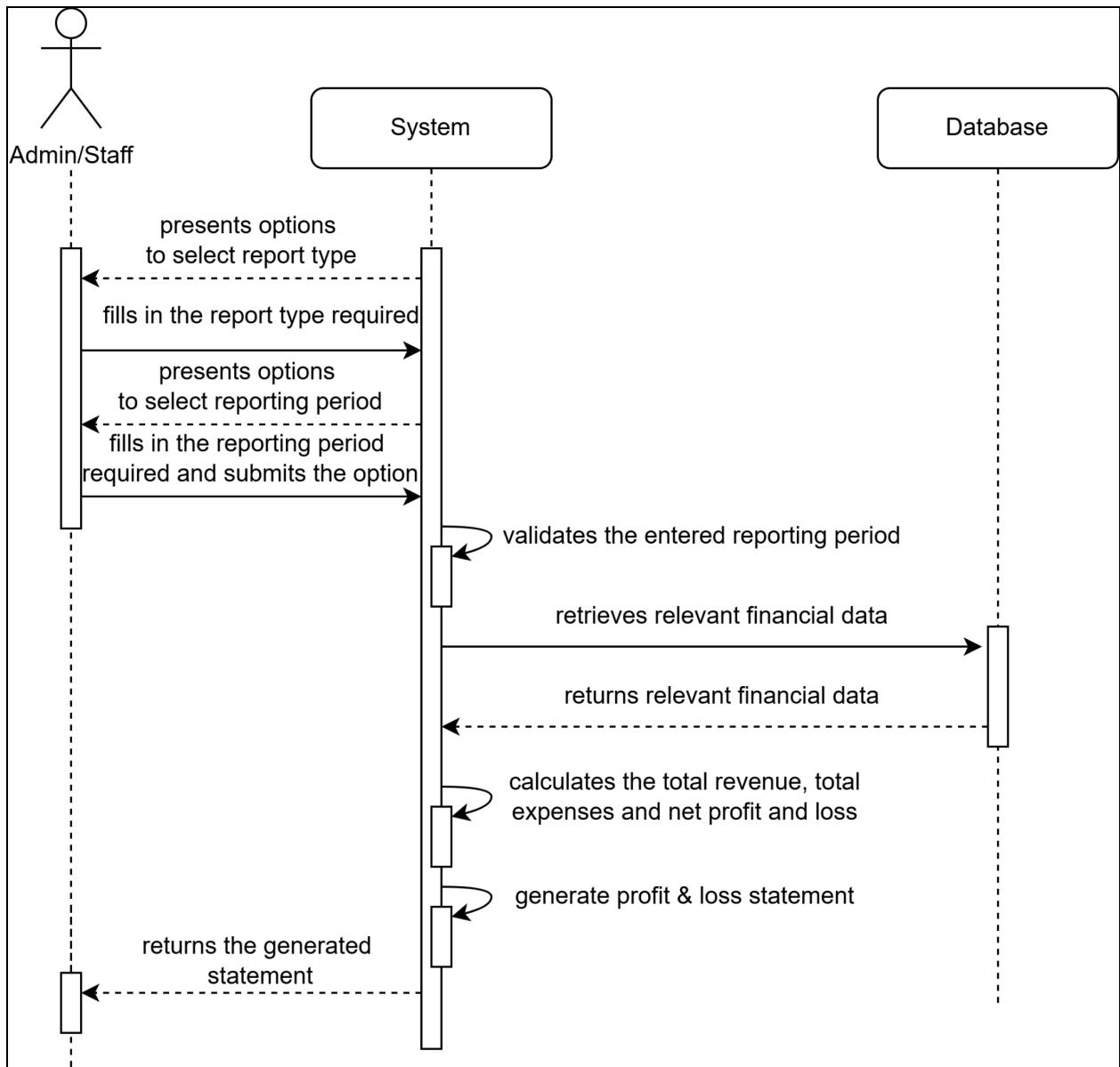


Figure 4.7.5.3.4.2 - Sequence Diagram for Finance Report SubSystem -
UC004: Generate Profit & Loss Statement

4.7.6 Database Design

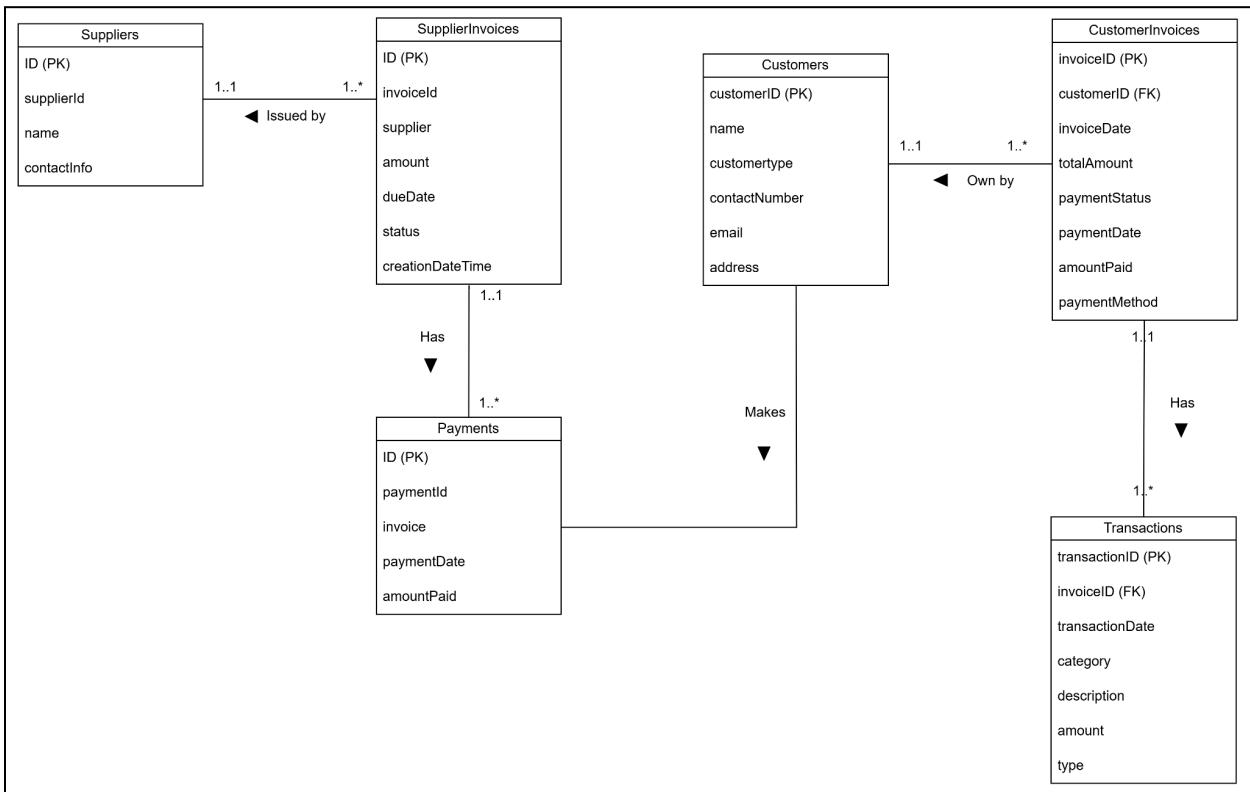


Figure 4.7.6.1 - Entity Relationship Diagram (ERD) of Finance System

The ERD represents the structure of the financial system, showcasing the relationships between key entities involved in supplier and customer transactions. The diagram consists of six main entities which are Suppliers, SupplierInvoices, Payments, Customers, CustomerInvoices, and Transactions. Each supplier can issue multiple supplier invoices, and each invoice can have multiple payments recorded. A customer must have at least one customer invoice, and each invoice can have multiple associated transactions to track financial activities. Payments made towards supplier invoices ensure proper reconciliation, while customer invoices record sales and payments received. These entities will then be utilized by the General Ledger subsystem for further financial analysis, ensuring accurate record-keeping, tracking outstanding balances, and generating comprehensive financial reports.

4.7.6.1 Data Description

Table 4.7.6.1.1 - Description of Entities in the Database

No.	Entity Name	Description
1.	CustomerInvoices	This entity represents the invoices issued to customers, capturing the details necessary for tracking payments and managing accounts receivable.
2.	Customers	This entity stores the customers' information who purchase items from the company and own at least one invoice.
3.	SupplierInvoices	This entity represents the invoices issued by the suppliers, capturing the details for managing accounts payable and highlighting the amount that needed to be paid. Each invoice is associated with a supplier.
4.	Suppliers	This entity stores the suppliers list and their information that supplies items to the company and owns at least one invoice.
5.	Payments	This entity records the payment that had been made to the supplier and also the amount that had been paid. Each payment is associated with an invoice.
6.	Transactions	This entity records individual financial transactions, tracking the monetary movements with details such as amount, category and transaction type. Each transaction is associated with an invoice.

4.7.6.2 Data Dictionary

Table 4.7.6.2.1 - Data Dictionary for CustomerInvoices Entity

Attribute Name	Type	Description
invoiceID	varchar	A unique identifier for each invoice.
customerID	varchar	A reference to the CustomerID in the Customers entity, linking the invoice to a specific customer.
invoiceDate	date	The date when the invoice was issued to the customer.
totalAmount	decimal(10,2)	The total amount billed on the invoice.
paymentStatus	varchar	Indicates whether the invoice has been paid, partially paid, or remains unpaid.
paymentDate	date	The date when payment was received.
amountPaid	decimal(10,2)	The amount paid by the customer for the invoice. This may be less than the TotalAmount for partial payments.
paymentMethod	varchar	The method used by the customer to make the payment.

Table 4.7.6.2.2 - Data Dictionary for Customers Entity

Attribute Name	Type	Description
customerID	varchar	A unique identifier for each customer.
name	varchar	The name of the customer (either individual or company).
customertype	varchar	The type of customer (e.g., "Individual," "Government", "Organisation").
contactNumber	varchar	The customer's phone number for communication purposes.
email	varchar	The customer's email address for correspondence.
address	varchar	The physical address of the customer for record-keeping and billing purposes.

Table 4.7.6.2.3 - Data Dictionary for SupplierInvoices Entity

Attribute Name	Type	Description
ID	varchar	A unique identifier in UUID for each supplier invoice which is used for linkage purposes.
invoiceId	varchar	A unique identifier in a user readable format and is automatically generated upon creation.
supplier	varchar	Associate to the Suppliers entity.
amount	decimal(15,2)	The amount that needs to be paid to the corresponding supplier for each invoice.
dueDate	date	The payment due date of the invoice which

		format is in YYYY-MM-DD.
status	varchar	The status of the invoice either in PAID, PENDING, PENDING (HALF) for those that have leftover amount to be paid or OVERDUE.
creationDateTime	timestamp	The date and time upon invoice creation.

Table 4.7.6.2.4 - Data Dictionary for Suppliers Entity

Attribute Name	Type	Description
ID	varchar	A unique identifier in UUID for each supplier which is used for linkage purposes.
supplierId	varchar	A unique identifier in a user readable format and is automatically generated upon creation.
name	varchar	The name of the supplier.
contactInfo	varchar	The contact information such as email of the supplier.

Table 4.7.6.2.5 - Data Dictionary for Payments Entity

Attribute Name	Type	Description
ID	varchar	A unique identifier in UUID for each payment which is used for linkage purposes.
paymentId	varchar	A unique identifier in a user readable format and is automatically generated upon creation.
invoice	varchar	Associate to the SupplierInvoices entity.

paymentDate	date	The payment date for the particular invoice which format is in YYYY-MM-DD.
amountPaid	decimal(15,2)	The amount that had been paid to the corresponding supplier for each invoice.

Table 4.7.6.2.6 - Data Dictionary for Transactions Entity

Attribute Name	Type	Description
transactionID	varchar	Unique identifier for each transaction.
transactionDate	date	Date of the transaction in YYYY-MM-DD format.
category	varchar	Classification of the transaction based on their category.
description	varchar	Detailed description of the transaction.
amount	decimal(10,2)	Monetary value of the transaction.
type	varchar	Transaction classification based on their type, either INCOME or EXPENSE.

4.7.7 Interface Design

4.7.7.1 Interface Design for AR Subsystem

The screenshot shows a standard web-based application interface for managing invoices. At the top left, there is a dropdown menu labeled "Standard". To its right is a toolbar with icons for "Search" (magnifying glass), "Create" (plus sign), "Delete" (trash can), and other settings. Below the toolbar is a table with the following columns: "Invoice ID", "Customer ID", "Invoice Date", "Total Amount (MYR)", "Payment Status", and "Amount Paid". There are six rows of data, each representing an invoice entry. The first row's "Invoice ID" field, containing "INV1167", is highlighted with a blue border.

Invoice ID	Customer ID	Invoice Date	Total Amount (MYR)	Payment Status	Amount Paid
INV1167	CUST07	Jan 10, 2025	65.00	Not Fully Paid	50.00 >
INV1178	CUST08	Jan 11, 2025	100.00	Fully Paid	100.00 >
INV1180	CUST08	Jan 11, 2025	100.00	Fully Paid	100.00 >
INV1185	CUST03	Jan 15, 2025	300.00	Fully Paid	300.00 >
INV1189	CUST06	Jan 15, 2025	600.00	Pending	600.00 >

Figure 4.7.7.1.1 - Main Page for AR Subsystem

This figure showcases the main interface where users can view a comprehensive list of invoices. It provides a clear and organized layout, making it easy to navigate through various invoice entries.

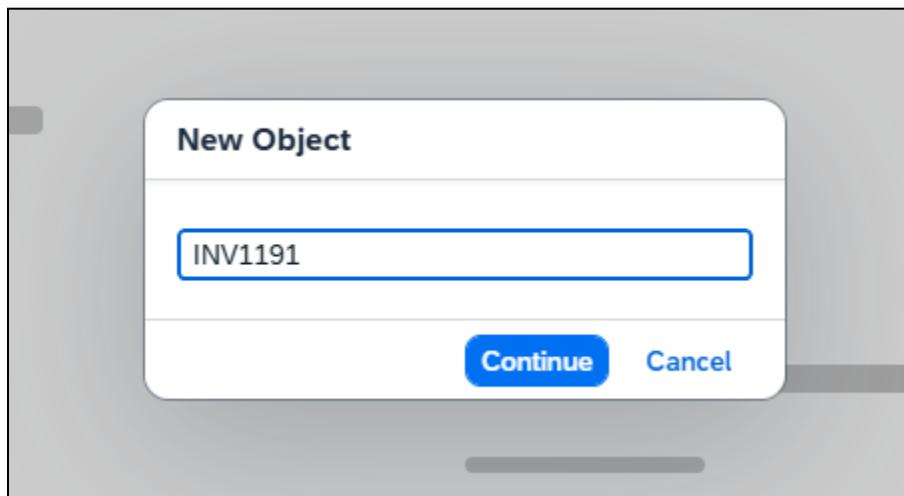


Figure 4.7.7.1.2 - System Prompt to Enter Invoice ID

After clicking the 'create' button, users are prompted to enter an invoice ID.

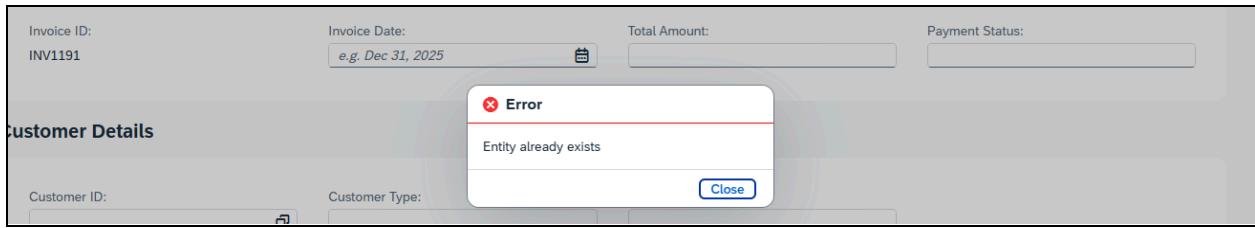


Figure 4.7.7.1.3 - Exception Flow if Invoice ID Already Exist

This figure illustrates the system's response when a user attempts to enter a duplicate invoice ID. It helps prevent errors by alerting users to the existing entry, ensuring data integrity.

Figure 4.7.7.1.4 - Invoice Form to Record Invoice Details

Here, users can input mandatory details for the invoice received from the sales department. The form is designed to ensure that all necessary information is captured efficiently.

INV1191

Invoice Details Customer Details Payment Details

Invoice ID: INV1191	Invoice Date: Jan 24, 2025	Total Amount: 500.00	Payment Status: Pending
------------------------	-------------------------------	-------------------------	----------------------------

Customer Details

Customer ID: CUST06	Customer Type: Government	Email: health@selangor.gov.my
Customer Name: Jabatan Kesihatan Selangor	Contact Number: 012-3344556	Address: Kompleks Bangunan Kerajaan

Payment Details

Payment Date: -	Payment Method / Term Of Payment: -	Amount Paid: -
--------------------	--	-------------------

Figure 4.7.7.1.5 - Invoice Details after Created

This figure displays the invoice details once they have been successfully created. Users can quickly verify the information before finalizing the entry.

INV1191

Invoice Details Customer Details Payment Details

Invoice ID: INV1191	Invoice Date: Jan 24, 2025	Total Amount: 500.00	Payment Status: Fully Paid
------------------------	-------------------------------	-------------------------	--------------------------------------

Customer Details

Customer ID: CUST06	Customer Type: Government	Email: health@selangor.gov.my
Customer Name: Jabatan Kesihatan Selangor	Contact Number: 012-3344556	Address: Kompleks Bangunan Kerajaan

Payment Details

Payment Date: Jan 24, 2025	Payment Method / Term Of Payment: Local Order	Amount Paid: 500.00
-------------------------------	--	------------------------

Draft **Save** Discard Draft

Figure 4.7.7.1.6 - Edit Invoice Form

In this page, users can modify existing invoice details by clicking the ‘edit’ button. The interface allows for easy adjustments to keep records up-to-date.

INV1191

Invoice Details	Customer Details	Payment Details
Invoice ID: INV1191	Invoice Date: Jan 24, 2025	Total Amount: 500.00
		Payment Status: Fully Paid
Customer Details		
Customer ID: CUST06	Customer Type: Government	Email: health@selangor.gov.my
Customer Name: Jabatan Kesihatan Selangor	Contact Number: 012-3344556	Address: Kompleks Bangunan Kerajaan
Payment Details		
Payment Date: Jan 24, 2025	Payment Method / Term Of Payment: Local Order	Amount Paid: 500.00

Figure 4.7.7.1.7 - Invoice Details after Editing

This figure shows the updated invoice details following any edits. It ensures that any changes are clearly reflected for user review.

Standard

	Invoice ID	Customer ID	Invoice Date	Total Amount (MYR)	Payment Status	Amount Paid	
<input type="checkbox"/>	INV1167	CUST07	Jan 10, 2025	65.00	Not Fully Paid	50.00	>
<input type="checkbox"/>	INV1178	CUST08	Jan 11, 2025	100.00	Fully Paid	100.00	>
<input type="checkbox"/>	INV1180	CUST08	Jan 11, 2025	100.00	Fully Paid	100.00	>
<input type="checkbox"/>	INV1185	CUST03	Jan 15, 2025	300.00	Fully Paid	300.00	>
<input checked="" type="checkbox"/>	INV1189	CUST06	Jan 15, 2025	600.00	Pending	600.00	>
<input type="checkbox"/>	INV1191	CUST06	Jan 24, 2025	500.00	Fully Paid	500.00	>

Figure 4.7.7.1.8 - Invoice is Selected to Delete

In this figure, a specific invoice is highlighted for deletion.

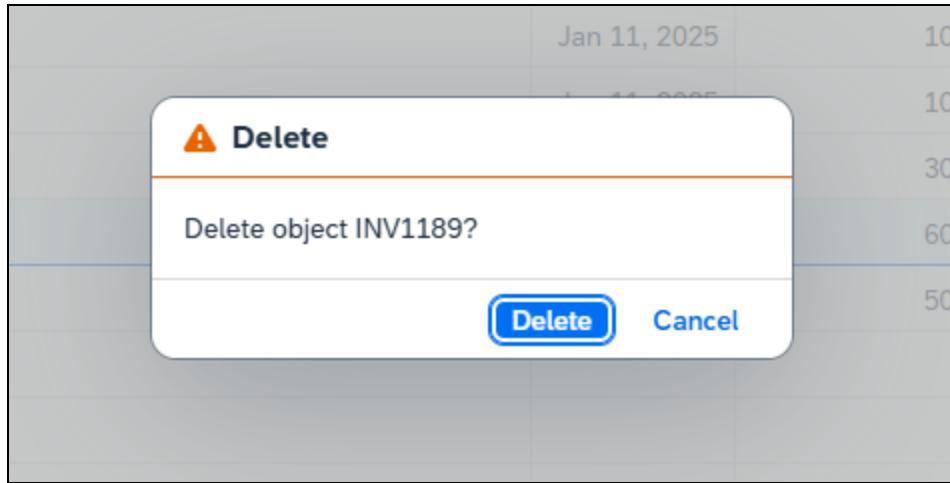


Figure 4.7.7.1.9 - Confirmation Prompt to Delete Invoice

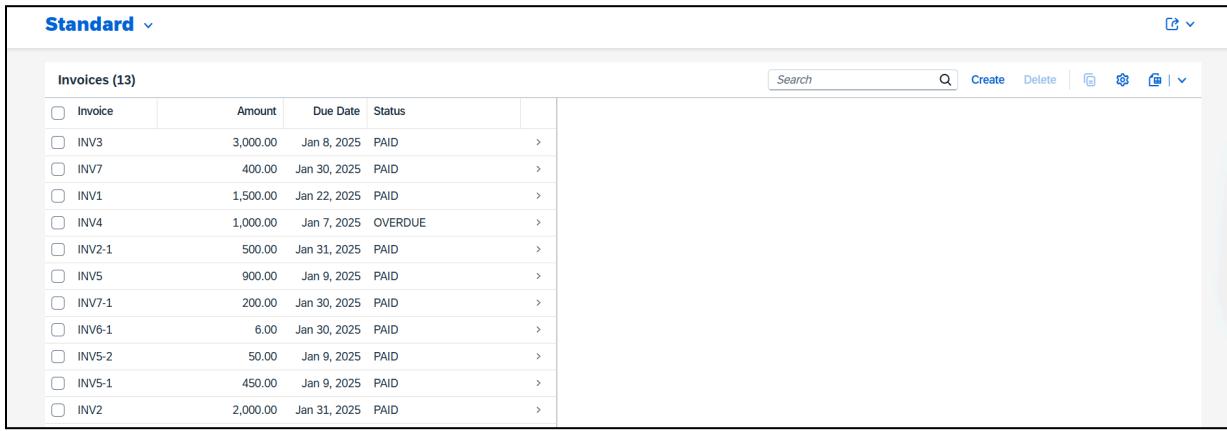
Before deleting an invoice, users receive a confirmation prompt. This safeguard helps prevent accidental deletions, ensuring users are intentional with their actions.

Standard Edit ▾						
	Invoice ID	Customer ID	Invoice Date	Total Amount (MYR)	Payment Status	Amount Paid
<input type="checkbox"/>	INV1167	CUST07	Jan 10, 2025	65.00	Not Fully Paid	50.00 >
<input checked="" type="checkbox"/>	INV1178	CUST08	Jan 11, 2025	100.00	Fully Paid	100.00 >
<input type="checkbox"/>	INV1180	CUST08	Jan 11, 2025	100.00	Fully Paid	100.00 >
<input type="checkbox"/>	INV1185	CUST03	Jan 15, 2025	300.00	Fully Paid	300.00 >
<input type="checkbox"/>	INV1191	CUST06	Jan 24, 2025	500.00	Fully Paid	500.00 >

Figure 4.7.7.1.10 - Invoice Form to Record Invoice

After an invoice is successfully deleted, this figure presents the updated main page. It reflects the changes in real-time, keeping users informed of their actions.

4.7.7.2 Interface Design for AP Subsystem

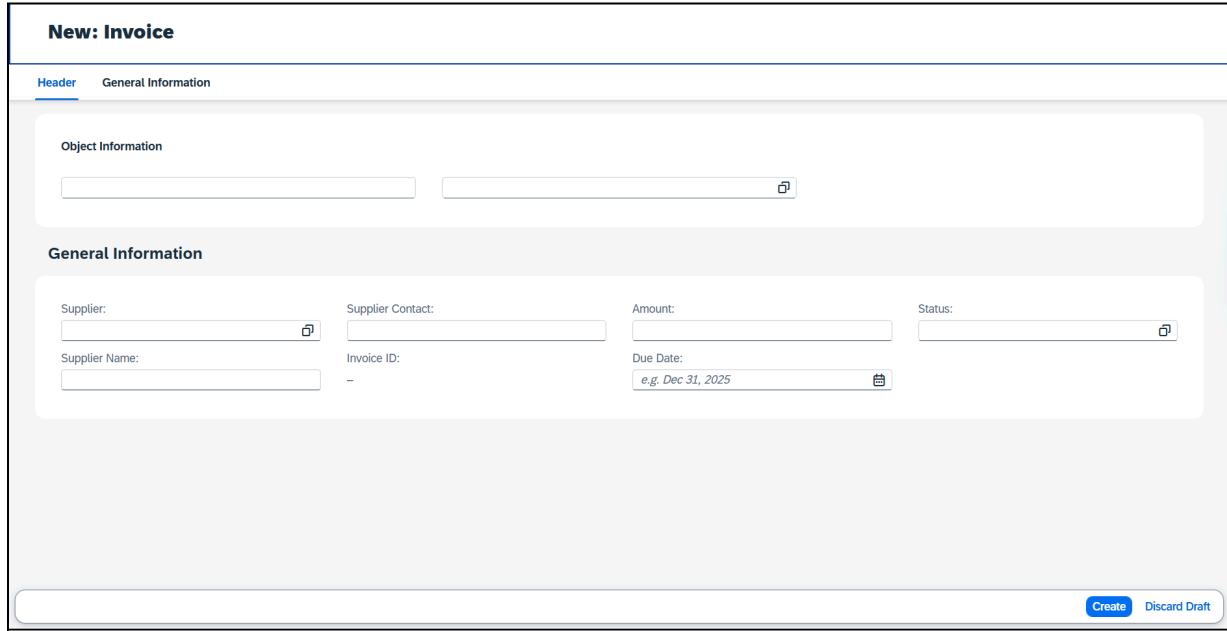


The screenshot shows a standard web interface for managing supplier invoices. At the top, there's a header with a dropdown menu labeled "Standard". Below the header is a search bar with a magnifying glass icon and buttons for "Create", "Delete", and other actions. The main content area is titled "Invoices (13)". It contains a table with columns: "Invoice", "Amount", "Due Date", and "Status". The table lists 13 invoices, each with a checkbox in the first column and a link icon in the last column. The invoices are as follows:

Invoice	Amount	Due Date	Status
INV3	3,000.00	Jan 8, 2025	PAID
INV7	400.00	Jan 30, 2025	PAID
INV1	1,500.00	Jan 22, 2025	PAID
INV4	1,000.00	Jan 7, 2025	OVERDUE
INV2-1	500.00	Jan 31, 2025	PAID
INV5	900.00	Jan 9, 2025	PAID
INV7-1	200.00	Jan 30, 2025	PAID
INV6-1	6.00	Jan 30, 2025	PAID
INV5-2	50.00	Jan 9, 2025	PAID
INV5-1	450.00	Jan 9, 2025	PAID
INV2	2,000.00	Jan 31, 2025	PAID

Figure 4.7.7.2.1 - Main Page for Supplier Invoices

This figure showcases the main interface where users can view a comprehensive list of invoices from each supplier. It provides a clear and organized layout, making it easy to navigate through various supplier invoice entries.



The screenshot shows a form titled "New: Invoice" for recording a new invoice. The form is divided into sections: "Header" and "General Information".

Header: General Information

Object Information: (Two empty text input fields)

General Information:

Supplier:	Supplier Contact:	Amount:	Status:
Supplier Name:	Invoice ID:	Due Date:	
(Text input field)	(Text input field)	(Text input field)	(Text input field)
e.g. Dec 31, 2025			

At the bottom right of the form are two buttons: "Create" and "Discard Draft".

Figure 4.7.7.2.2 - Invoice Form to Record Invoice

This figure showcases the form for invoice creation.

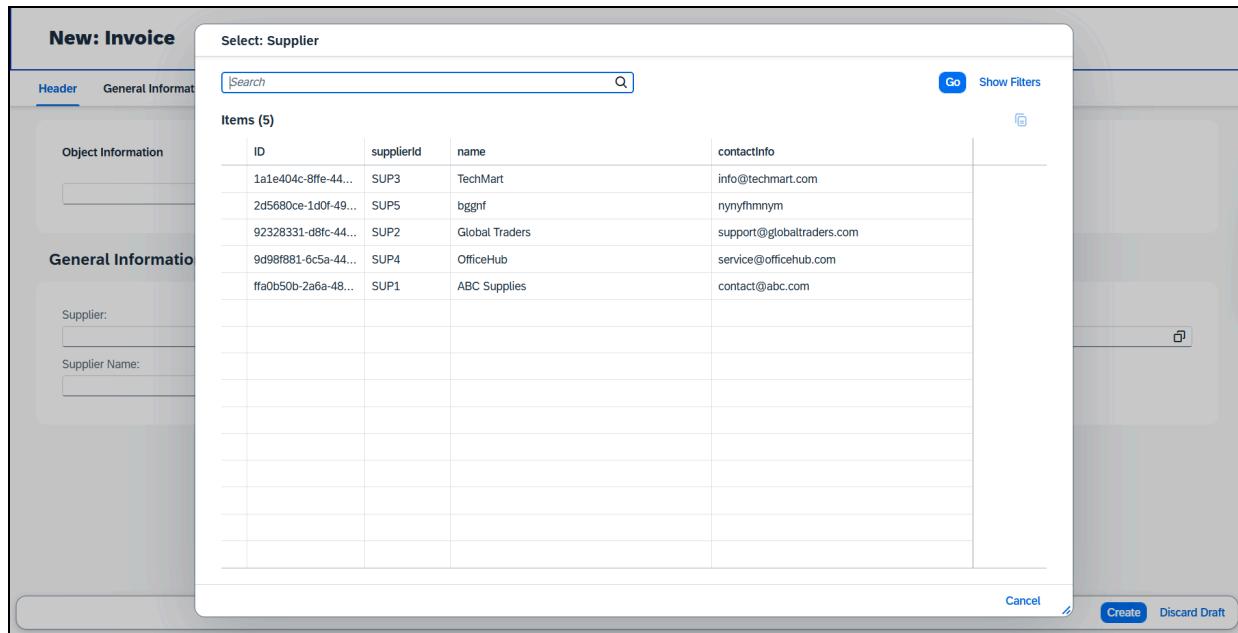


Figure 4.7.7.2.3 - Dropdown Menu for Supplier Selection

This figure showcases the dropdown menu of the suppliers where users can choose the supplier for each invoice.



Figure 4.7.7.2.4 - The Details for Invoice from Supplier

This figure showcases the interface of the details of supplier invoice.

The screenshot shows a software interface for managing invoices. At the top, there's a header with the title 'TechMart' and a status 'PAID'. Below the header, there are two tabs: 'Header' and 'General Information', with 'General Information' being the active tab. The main area is titled 'General Information' and contains several input fields:

- Supplier: A dropdown menu showing the ID '1a1e404c-8ffe-4453-bc93-09d5b5f336c1'.
- Supplier Contact: An input field containing 'info@techmart.com'.
- Amount: An input field showing '3,000.00'.
- Status: A dropdown menu showing 'PAID'.
- Supplier Name: An input field showing 'TechMart'.
- Invoice ID: An input field showing 'INV3'.
- Due Date: An input field showing 'Jan 8, 2025'.

At the bottom right of the form, there are two buttons: 'Save' and 'Discard Draft'.

Figure 4.7.7.2.5 - Edit Page for a Particular Invoice

This figure showcases the edit form for a particular invoice after clicking the edit button at the upper right corner.

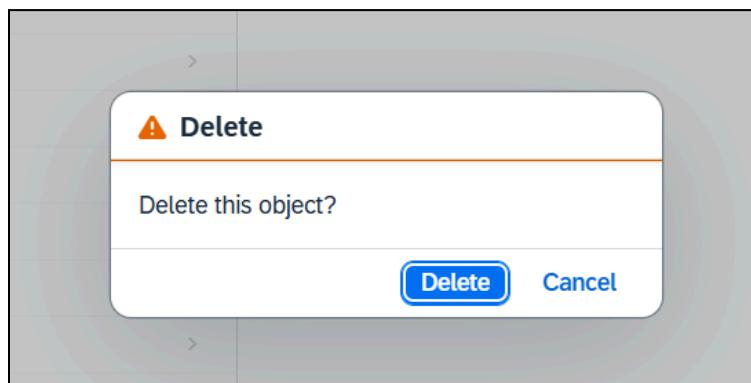


Figure 4.7.7.2.6 - Confirmation Message After Clicking Delete Button

This figure shows the confirmation message when the user is trying to delete a supplier invoice to avoid any regrets.

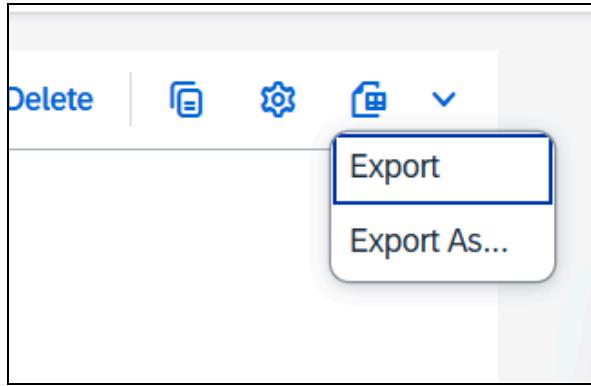


Figure 4.7.7.2.7 - Selection for Export Method In Invoices Page

This figure shows the selection of export where users can choose to direct export as xlsx format without any requirement or “Export as” which users can tick for any requirement.

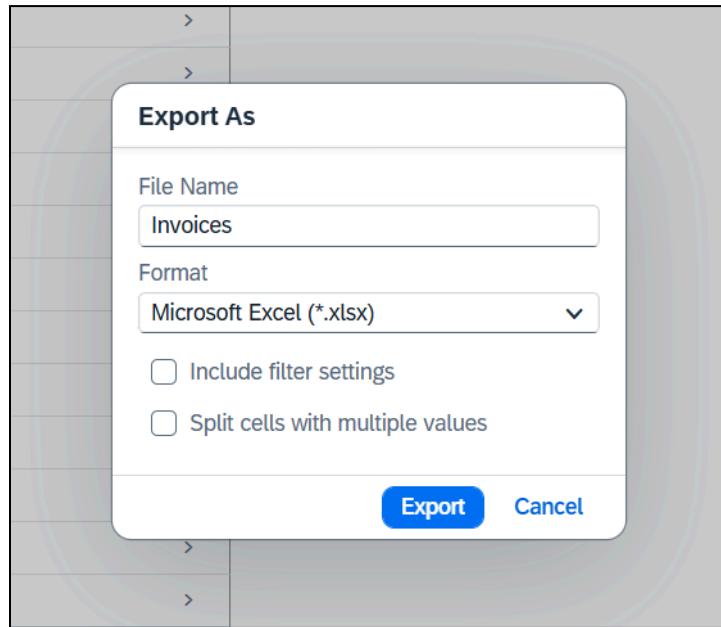


Figure 4.7.7.2.8 - Interface Popping Up After Selecting “Export As” In Invoices Page

This figure showcases the interface when users choose to “Export As” and they can tick for their requirement whether to include the filter settings or split cells with multiple values.

Suppliers (5)			
	Supplier ID	Supplier Name	Supplier Contact Info
<input type="checkbox"/>	SUP3	TechMart	info@techmart.com >
<input type="checkbox"/>	SUP5	bgnf	nynyfhmnym >
<input type="checkbox"/>	SUP2	Global Traders	support@globaltraders.com >
<input type="checkbox"/>	SUP4	OfficeHub	service@officehub.com >
<input type="checkbox"/>	SUP1	ABC Supplies	contact@abc.com >

Figure 4.7.7.2.9 - Main Page for Suppliers

This figure showcases the main interface where users can view a comprehensive list of suppliers. It provides a clear and organized layout, making it easy to navigate through various supplier entries.

New: Supplier

General Information

Supplier ID:	Supplier Name:	Supplier Contact Infomation:
—	<input type="text"/>	<input type="text"/>

Create **Discard Draft**

Figure 4.7.7.2.10 - Supplier Form to Add New Supplier

This figure showcases the form for new supplier creation.

The screenshot shows a web-based application interface for managing suppliers. At the top, the title "TechMart" is displayed. In the top right corner, there are three buttons: "Edit" (highlighted in blue), "Delete", and a refresh icon. Below the title, the section "General Information" is visible. Under this section, there are three fields: "Supplier ID" with the value "SUP3", "Supplier Name" with the value "TechMart", and "Supplier Contact Infomation" with the value "info@techmart.com".

Figure 4.7.7.2.11 - The Details for a Particular Supplier

This figure showcases the interface of the details of the supplier.

The screenshot shows the same "TechMart" application interface, but it is in edit mode. The top right corner now shows "Draft" with a dropdown arrow and a refresh icon. The "General Information" section is identical to Figure 4.7.7.2.11, with "Supplier ID" as "SUP3", "Supplier Name" as "TechMart", and "Supplier Contact Infomation" as "info@techmart.com". At the bottom right of the form, there are two buttons: "Save" (highlighted in blue) and "Discard Draft".

Figure 4.7.7.2.12 - Edit Page for a Particular Supplier

This figure showcases the edit form for a particular supplier after clicking the edit button at the upper right corner.

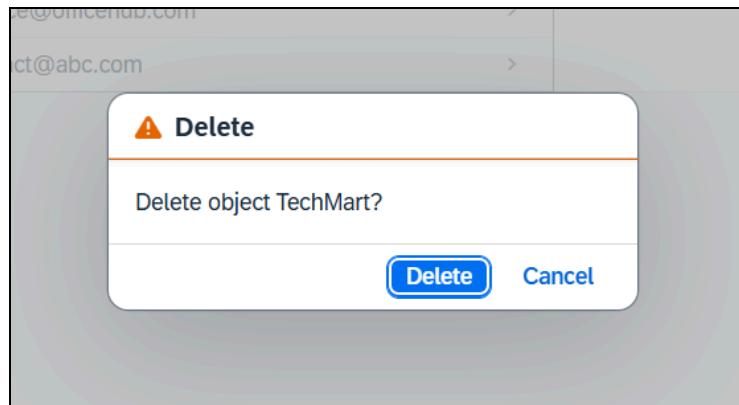


Figure 4.7.7.2.13 - Confirmation Message After Clicking Delete Button

This figure shows the confirmation message when the user is trying to delete a supplier to avoid any regrets.

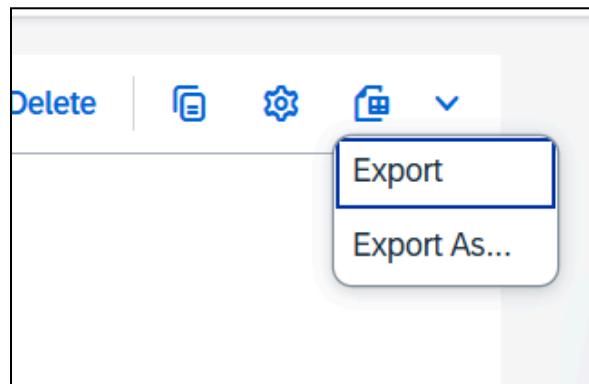


Figure 4.7.7.2.7.14 - Selection for Export Method In Suppliers Page

This figure shows the selection of export where users can choose to direct export as xlsx format without any requirement or “Export as” which users can tick for any requirement.

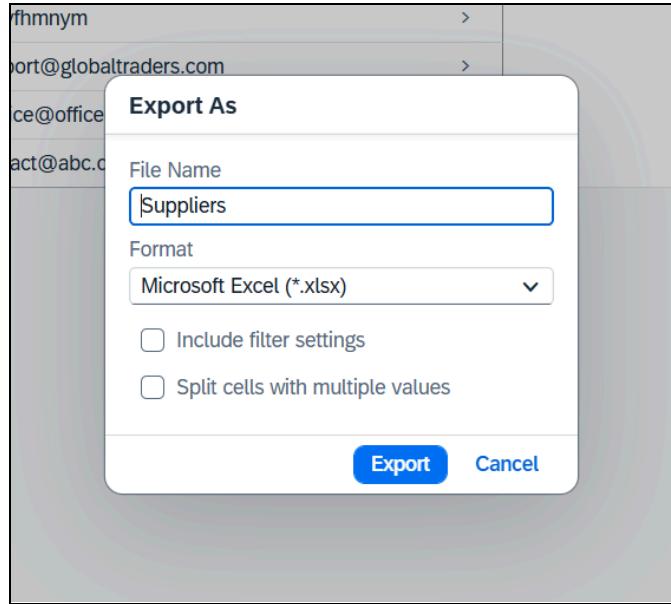


Figure 4.7.7.2.15 - Interface Popping Up After Selecting “Export As” In Suppliers Page

This figure showcases the interface when users choose to “Export As” and they can tick for their requirement whether to include the filter settings or split cells with multiple values.

Payments (11)					Search	Create	Delete	Print	Export	Import	Help
<input type="checkbox"/>	Payment ID	Invoice ID	Payment Date	Amount Paid							
<input type="checkbox"/>	PAY8	INV6	Jan 16, 2025	5.00	>						
<input type="checkbox"/>	PAY1	INV1	Jan 11, 2025	1,500.00	>						
<input type="checkbox"/>	PAY5	INV5	Jan 11, 2025	450.00	>						
<input type="checkbox"/>	PAY7	INV5-2	Jan 11, 2025	50.00	>						
<input type="checkbox"/>	PAY9	INV6-1	Jan 16, 2025	6.00	>						
<input type="checkbox"/>	PAY10	INV7-1	Jan 16, 2025	200.00	>						
<input type="checkbox"/>	PAY4	INV2-2	Jan 11, 2025	250.00	>						
<input type="checkbox"/>	PAY10	INV7	Jan 16, 2025	200.00	>						
<input type="checkbox"/>	PAY6	INV5-1	Jan 11, 2025	400.00	>						
<input type="checkbox"/>	PAY2	INV2	Jan 11, 2025	1,500.00	>						

Figure 4.7.7.2.16 - Main Page for Payments

This figure showcases the main interface where users can view a comprehensive list of payments. It provides a clear and organized layout, making it easy to navigate through various payment entries.

New: Payment

General Information

Invoice:	Payment ID:	Amount Paid:
<input type="text"/>	-	<input type="text"/>
Invoice ID:	Payment Date:	
-	<input type="text"/> e.g. Dec 31, 2025	<input type="button"/>

Create **Discard Draft**

Figure 4.7.7.2.17 - Payment Form to Record Payment

This figure showcases the form for payment records that had been made.

New: Payment

Select: Invoice

Items (13)						
ID	invoiceId	amount	creationDateTime	dueDate	status	
101f4c79-862a-49...	INV3	3,000.00	Jan 11, 2025, 12:09:13AM	Jan 8, 2025	PAID	<input type="button"/>
1d61adae-e89e-4...	INV7	400.00	Jan 16, 2025, 2:52:13PM	Jan 30, 2025	PAID	<input type="button"/>
320b879d-9093-4...	INV1	1,500.00	Jan 11, 2025, 12:06:03AM	Jan 22, 2025	PAID	<input type="button"/>
374cb334-f1ac-44...	INV4	1,000.00	Jan 11, 2025, 12:09:31AM	Jan 7, 2025	OVERDUE	<input type="button"/>
3f9715f4-6e6e-4e...	INV2-1	500.00	Jan 11, 2025, 12:40:53AM	Jan 31, 2025	PAID	<input type="button"/>
418a1e0a-e45e-4...	INV5	900.00	Jan 11, 2025, 1:01:49AM	Jan 9, 2025	PAID	<input type="button"/>
4309d594-3d76-4...	INV7-1	200.00	Jan 16, 2025, 2:53:07PM	Jan 30, 2025	PAID	<input type="button"/>
4fc741a5-fc19-44...	INV6-1	6.00	Jan 16, 2025, 2:37:07PM	Jan 30, 2025	PAID	<input type="button"/>
7e2918b7-f53d-45...	INV5-2	50.00	Jan 11, 2025, 1:03:47AM	Jan 9, 2025	PAID	<input type="button"/>
8cd2cab0-7a10-4f...	INV5-1	450.00	Jan 11, 2025, 1:02:29AM	Jan 9, 2025	PAID	<input type="button"/>
93530131-d4c2-4...	INV2	2,000.00	Jan 11, 2025, 12:08:05AM	Jan 31, 2025	PAID	<input type="button"/>
988f51bf-a3ac-44...	INV6	11.00	Jan 16, 2025, 2:35:44PM	Jan 30, 2025	PAID	<input type="button"/>
bcf9cf60-459f-45b...	INV2-2	250.00	Jan 11, 2025, 12:42:46AM	Jan 31, 2025	PAID	<input type="button"/>

Create **Discard Draft**

Figure 4.7.7.2.18 - Dropdown Menu for Invoice Selection

This figure showcases the dropdown menu of the invoices where users can choose the invoice for each payment.

The screenshot shows a payment details page with the following fields:

Invoice:	Payment ID:	Amount Paid:
988f51bf-a3ac-4405-8157-b11f92dd7992	PAY8	5.00
Invoice ID:	Payment Date:	
INV6	Jan 16, 2025	

Buttons at the top right include Edit, Delete, and a dropdown menu.

Figure 4.7.7.2.19 - The Details for Payment

This figure showcases the interface of the details of payment that had been made.

The screenshot shows an edit form for a payment, with the following fields:

Invoice:	Payment ID:	Amount Paid:
988f51bf-a3ac-4405-8157-b11f92dd7992	PAY8	5.00
Invoice ID:	Payment Date:	
INV6	Jan 16, 2025	

Buttons at the bottom right include Save and Discard Draft.

Figure 4.7.7.2.20 - Edit Page for a Particular Payment

This figure showcases the edit form for a particular payment after clicking the edit button at the upper right corner.

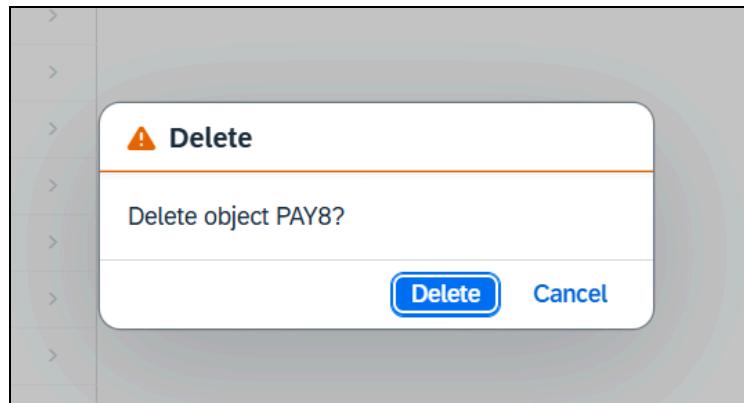


Figure 4.7.7.2.21 - Confirmation Message After Clicking Delete Button

This figure shows the confirmation message when the user is trying to delete a payment record to avoid any regrets.

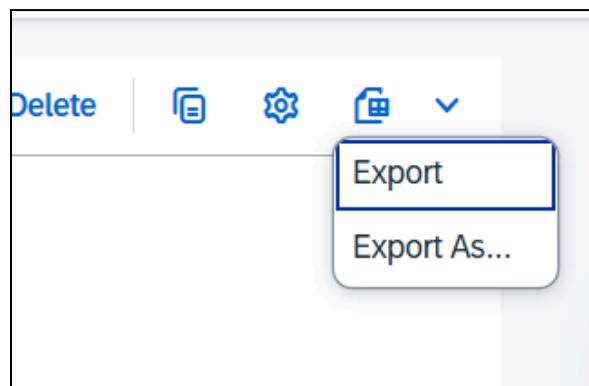


Figure 4.7.7.2.22 - Selection for Export Method In Payments Page

This figure shows the selection of export where users can choose to direct export as xlsx format without any requirement or “Export as” which users can tick for any requirement.

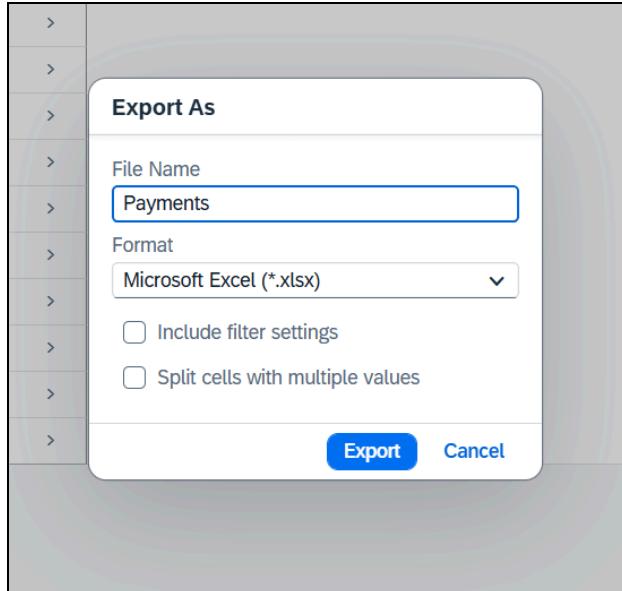


Figure 4.7.7.2.23 - Interface Popping Up After Selecting “Export As” In Payments Page

This figure showcases the interface when users choose to “Export As” and they can tick for their requirement whether to include the filter settings or split cells with multiple values.

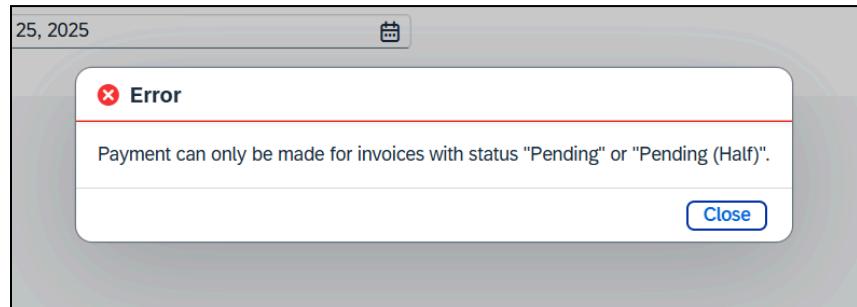


Figure 4.7.7.2.24 - Error Message Popping Up When User Select Invalid for Payment

This figure showcases the error message popping up when a user chooses the invoice that has status of “PAID” or “OVERDUE” which is not available for payment making while only invoice with status of “PENDING” or “PENDING (HALF)” can be selected.

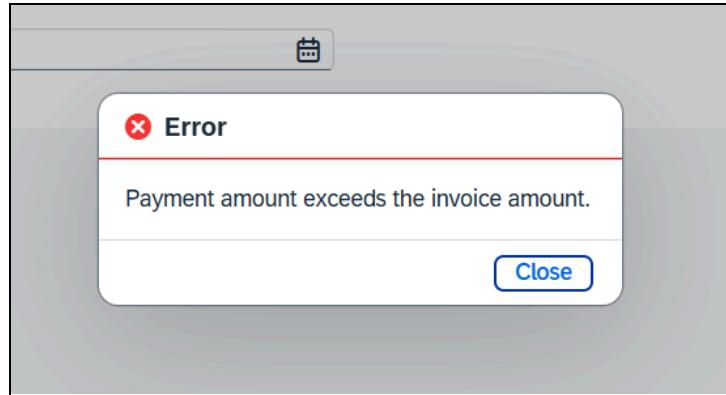


Figure 4.7.7.2.25 - Error Message Popping Up When User Enters Payment Amount That Exceeds The Amount Recorded In That Particular Invoice

This figure showcases the error message popping up when a user enters the payment amount that exceeds the amount that had been recorded previously during invoice creation.

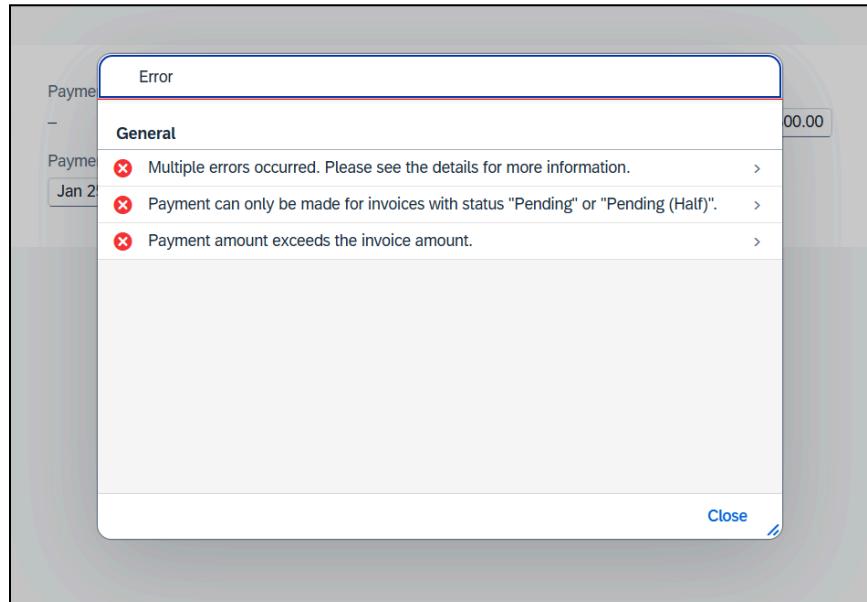


Figure 4.7.7.2.26 - Multiple Error Message Popping Up When a User did Multiple Error

This figure showcases the multiple error messages popping up when a user did multiple errors such as selecting an invoice that is invalid for payment making as well as entering a payment amount that is too big than what had been recorded before.

Invoices (2)			
Invoice	Amount	Due Date	Status
INV8-1	5.00	Jan 31, 2025	PENDING (HALF)
INV8	50.00	Jan 31, 2025	PAID

Figure 4.7.7.2.27 - Invoice Creation for Payment Amount that is not Full

This figure showcases the invoice(eg. INV8-1) that had been automatically created when the user did not pay the full amount of a particular invoice and the amount will be the leftover amount that is needed to be paid as well as the status will be PENDING (HALF).

4.7.7.3 Interface Design for GL Subsystem

The screenshot shows a user interface for generating a general ledger. At the top left is the logo 'ak maju' and the text 'AK MAJU General Ledger'. Below this is a section titled 'Choose Year and Month' containing two dropdown menus: 'Year' set to '2025' and 'Month' set to 'JAN'. At the bottom right is a blue button labeled 'Generate General Ledger'.

Figure 4.7.7.3.1 - Year and Month Selection to Generate General Ledger

This figure showcases the General Ledger Generation Page. In this page, users must choose a year and month for ledger generation. After that, the ledger will be generated when the Generate General Ledger button is clicked.

The screenshot shows the generated general ledger for the month of January 2025. At the top left is the 'ak maju PRINTWORKS' logo. On the right, there is company information for 'AK MAJU RESOURCES SDN. BHD.' including address, phone number, email, and company number. The main area displays a table of financial records:

Date	Category	Description	Debit	Credit
1/15/2025	Sales Revenue	Service Income	-	RM 36000.00
1/10/2025	Operating Expense	Office Rent	RM 5500.00	-
1/1/2025	Sales Revenue	Product Sales	-	RM 62000.00
1/5/2025	Cost of Goods	Raw Materials	RM 38000.00	-
1/20/2025	Marketing	Advertising	RM 4500.00	-
Net Income			RM 50000.00	

At the bottom center is a blue 'PRINT' button.

Figure 4.7.7.3.2 - Generated General Ledger

This figure showcases the generated General Ledger. It displays the financial records of within the month. The expenses are recorded in the column “Debit” whereas the incomes are recorded in the column “Credit”. Users can then print the general ledger by clicking the “PRINT” button.

General Ledger



AK MAJU RESOURCES SDN. BHD.
No. 39 & 41 Jalan Utama 3/2, Pusat Komersial Sri Utama,
Segamat, Johor, Malaysia - 85000
07-33310717, 010-2218224
akmaju.jo@gmail.com
Company No.: 1088436 K

Date	Category	Description	Debit	Credit
1/15/2025	Sales Revenue	Service Income	-	RM 36000.00
1/10/2025	Operating Expense	Office Rent	RM 5500.00	-
1/1/2025	Sales Revenue	Product Sales	-	RM 62000.00
1/5/2025	Cost of Goods	Raw Materials	RM 38000.00	-
1/20/2025	Marketing	Advertising	RM 4500.00	-
Net Income			RM 50000.00	

Print 1 page

Destination Save as PDF

Pages All

Layout Landscape

More settings ▼

Save Cancel

Figure 4.7.7.3.3 - Generated General Ledger Printing Page

This figure showcases the printing page of the generated General Ledger. Years can choose to save as PDF or print a physical copy by connecting to the printers.

4.7.7.4 Interface Design for Finance Report Subsystem

Report Options

Report Type: Yearly Monthly
Year: 2024

Generate P&L Statement

Transaction List

Transaction Records

Transaction ID	Date	Category	Description	Amount	Type
T001	Jan 1, 2024	Sales Revenue	Product Sales	50,000.00	INCOME
T002	Jan 2, 2024	Cost of Goods	Raw Materials	-30,000.00	EXPENSE
T003	Jan 3, 2024	Operating Expense	Office Rent	-5,000.00	EXPENSE
T004	Jan 5, 2024	Sales Revenue	Service Income	25,000.00	INCOME
T005	Jan 7, 2024	Operating Expense	Utilities	-2,000.00	EXPENSE
T006	Jan 10, 2024	Other Income	Interest Income	1,000.00	INCOME
T007	Jan 15, 2024	Operating Expense	Salaries	-15,000.00	EXPENSE
T008	Jan 20, 2024	Sales Revenue	Product Sales	35,000.00	INCOME
T009	Jan 25, 2024	Marketing	Advertising	-3,000.00	EXPENSE
T010	Jan 26, 2024	Operating Expenses	Equipment Maintenance	1,500.00	EXPENSE

Figure 4.7.7.4.1 - Main Page for Finance Report SubSystem

The main page for Finance Report Subsystem shows mainly two sections for generating Profit & Loss statements. At the top, users can find the Report Options section where they select their preferred report type (Yearly or Monthly) and specify the reporting period (Year and/or Month). Below, a Transaction List displays all financial transactions in a tabulated format, showing the transaction records and their key details such as Transaction ID, Date, Category, Description, Amount (color-coded for income and expenses) and Transaction Type. This provides a straightforward way for the admin and staff to view financial data and generate P&L statements.

 <p>AK MAJU RESOURCES SDN. BHD. No. 39 & 41 Jalan Usama 3/2, Pusat Komersial Sri Usama, Segamat, Johor, Malaysia 85000 07-4891117, 010-3218224 akmaju.printworks.com Company No.: 1088438 K</p>
PROFIT & LOSS STATEMENT
PERIOD: 2024
REVENUE
SALES REVENUE RM 911000.00
OTHER INCOME RM 11800.00
TOTAL REVENUE RM 922800.00
EXPENSES
OPERATING EXPENSE RM 93200.00
COST OF GOODS RM 398500.00
MARKETING RM 24000.00
TOTAL EXPENSES RM 515700.00
NET PROFIT RM 407100.00
PRINT

Figure 4.7.7.4.2 - Preview of Yearly P&L Statement for a selected reporting year

Upon selecting the preferred report type as yearly, the user can then specify the reporting year and select the 'Generate P&L Statement' button. This leads the user to a new tab showing the generated P&L statement for preview.

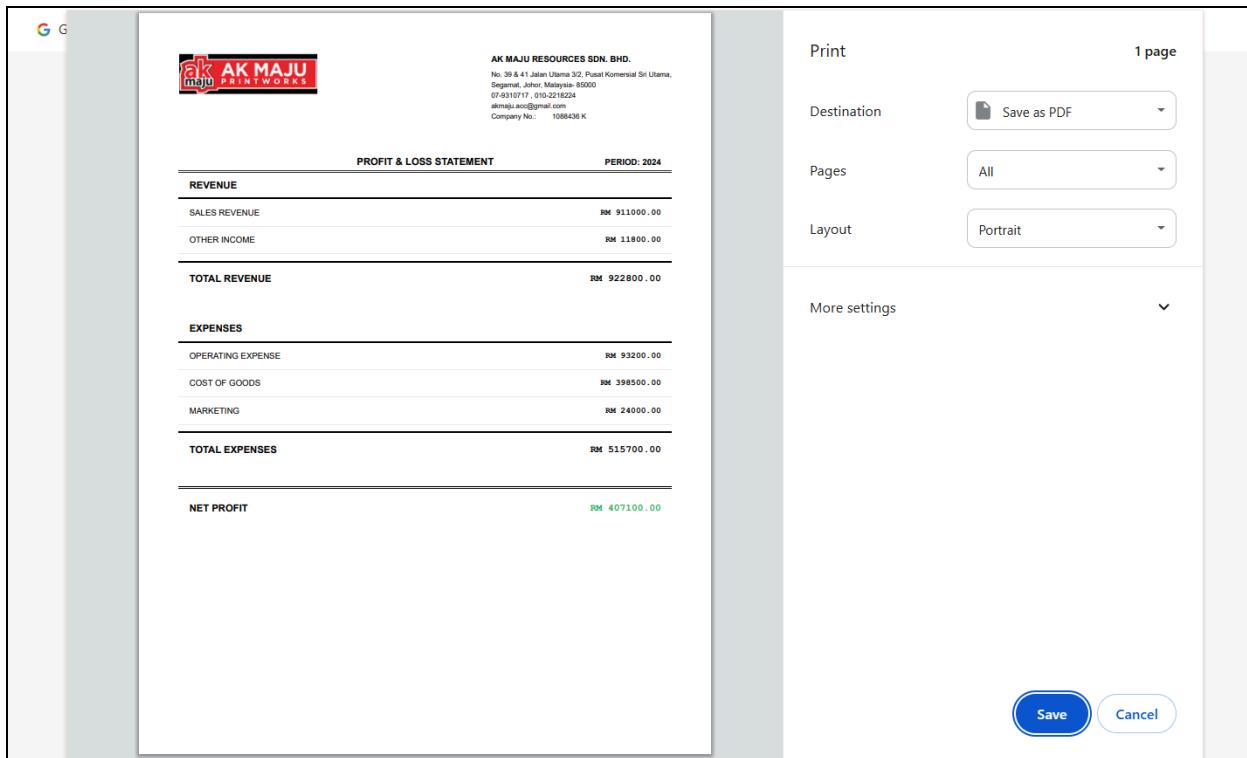


Figure 4.7.7.4.3 - Print or Save for Yearly P&L Statement for a selected reporting year

After clicking the 'Print' button, the user can either save the P&L statement as PDF or print it directly for future reference.

Figure 4.7.7.4.4 - Main Page for Finance Report SubSystem after Selecting Monthly Report Type

Upon selecting the preferred report type as monthly, the user can then specify both the year and month for the P&L statement generation. The subsequent steps follow the same process as the yearly report, where clicking the 'Generate P&L Statement' button opens a new tab for preview and is followed by options to print or save as PDF.

4.8 Summary

To summarize, Chapter 4 covers the analysis and design process for developing a financial system tailored to the needs of AK Maju Resources Sdn. Bhd. The chapter identifies the functional and non-functional requirements essential for managing AR, AP, GL, and financial reporting. It emphasizes the importance of replacing manual processes with automated workflows to enhance efficiency, accuracy, and reliability. The system's scope, objectives, and key functionalities are outlined to ensure the design meets the company's operational goals.

In conclusion, this chapter provides a detailed framework for designing the system, including enterprise architecture, database structures, workflows, and subsystem designs. Visual representations like data flow diagrams (DFDs) and interface designs ensure user-friendly navigation and logical data management. Challenges such as maintaining security and scalability while ensuring compatibility with existing tools were addressed effectively. Overall, this chapter establishes a solid foundation for implementing a financial system that will streamline AK Maju's operations.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction

Chapter 5 outlines the system implementation process for the financial management system developed for AK Maju Resources Sdn. Bhd. This chapter provides a comprehensive view of how the system was designed and developed, focusing on its modular architecture and team-based development strategy. By leveraging SAP Business Application Studio (BAS), SAP HANA Cloud, and the SAP CAP framework, the development process emphasized efficiency, scalability, and user-friendliness.

The chapter details the project's development environment setup, including the tools and frameworks used to streamline backend and frontend tasks. It also explains the structured workflow the team followed, from creating database schemas to coding the core functionalities. Each subsystem was developed with its own database structure and main functions, ensuring seamless data handling and efficient financial reporting.

5.2 System Development

To setup the development environment, five phases were involved which are :

Phase 1: Setting Up the SAP BTP Trial and Subaccount

The setup process began with subscribing to a 90-day SAP Business Technology Platform (SAP BTP) trial, which provided access to essential tools and services. Once access was granted, we created a sub account in SAP BTP. This subaccount acted as the central hub for managing all project resources, configurations, and service instances throughout the development process.

Phase 2: Adding SAP BAS and SAP HANA Cloud Instances

After setting up the subaccount, we followed official SAP tutorials to guide the initial development steps. As part of this phase, SAP Business Application Studio (BAS) and SAP HANA Cloud instances were added to the subaccount. SAP BAS was chosen as the primary development environment for coding and managing the application, while SAP HANA Cloud was selected as the database platform for secure and scalable data storage.

Phase 3: Creating a Development Project and Connecting to Cloud Foundry

Next, we launched SAP BAS and created a new development project using the SAP Cloud Application Programming (CAP) Model. CAP provided pre-built functionalities that simplified backend development. To enable the application to run effectively, we established a connection between SAP BAS and Cloud Foundry. Configuring Cloud Foundry ensured the project had a stable runtime environment to support system operations.

Phase 4: Configuring SAP HANA Cloud and Binding to the Project

In this phase, an SAP HANA Cloud instance was created, specifically setting up an HDI (HANA Deployment Infrastructure) container to serve as the primary database for the system. The HDI container was then bound to the project in SAP BAS, creating a seamless integration

between the development environment and the database. This integration ensured that all data operations, such as storage and retrieval, could be executed reliably.

Phase 5: Customizing the SAP CAP Code

Finally, the SAP CAP code was tailored to align with the project's subsystems and specific use cases. Adjustments were made to ensure that the core modules—accounts receivable, accounts payable, general ledger, and financial reporting—operated according to the defined requirements. By leveraging the flexibility of SAP CAP, the project team created interfaces and services optimized for the functionality and user experience required by the financial system.

By breaking down the environment setup into these phases, the team established a robust and efficient foundation for development while ensuring alignment with project goals and requirements.

5.3 Create Database

5.3.1 Database for AR Subsystem

```
10 entity Invoices : managed {
11     key invoiceID : String(100);
12     customerID : String(100);
13     customer : Association to Customers on customer.customerID = $self.customerID;
14     invoiceDate : Date;
15     totalAmount : Price;
16     paymentStatus: String(20);
17     paymentDate : Date;
18     amountPaid : Price;
19     paymentMethod: String(20);
20 };
```

Figure 5.3.1.1 - invoices.cds File

Figure 5.3.1.1 represents the core database entity for the Account Receivable subsystem, which is the Invoices entity. This file outlines the structure and attributes of the invoices, defining key elements such as invoice ID, customer details, invoice date, total amount, payment status, payment date, amount paid, and payment method. The Invoices entity serves as the foundation for managing all invoice-related data in the Account Receivable module. By clearly defining these attributes, this entity ensures that invoice data is stored consistently and can be retrieved efficiently for various financial operations.

The screenshot shows a software interface for managing CSV files. At the top, there's a header bar with icons for back, forward, and search, followed by the text "workspace (Workspace)". Below the header is a sidebar with various icons: a list icon, a document icon, a magnifying glass icon, a person icon with a "28" count, and a refresh/circular arrow icon. The main area displays a CSV file named "app_interaction_invoices.csv". The file path is shown as "AR_System > gen > db > src > gen > data > app_interaction_invoices.csv". The content of the CSV file is as follows:

	invoiceID	customerID	invoiceDate	totalAmount	paymentStatus	paymentDate	amountPaid	paymentMethod
1	INV1167	CUST07	2025-01-10	65.00	Not Fully Paid	2025-01-10	50.00	Cash
2	INV1178	CUST08	2025-01-11	100.00	Fully Paid	2025-01-11	100.00	Bank Transfer
3	INV1180	CUST08	2025-01-11	100.00	Fully Paid	2025-01-11	100.00	Bank Transfer
4	INV1185	CUST03	2025-01-15	300.00	Fully Paid	2025-01-16	300.00	Cash
5	INV1189	CUST06	2025-01-15	600.00	Fully Paid	2025-01-16	600.00	Local Order

Figure 5.3.1.2 - CSV File Database for AR Subsystem

Figure 5.3.1.2 illustrates the CSV file database designed for the Account Receivable system. This file contains the raw data that populates the Invoices entity in the database. Each record in the CSV file corresponds to a unique invoice, including details such as customer information, payment status, and amounts. The data stored in this CSV file feeds directly into the Invoices entity, ensuring that all invoice-related information is systematically organized and readily available for processing. This design facilitates smooth data integration and simplifies data import/export operations within the finance system.

5.3.2 Database for AP Subsystem

```
1  namespace app.interactions;
2
3  using {
4  |   cuid
5  } from '@sap/cds/common';
6
7  type SupplierID : String(10);
8  type InvoiceID : String(10);
9  type PaymentID : String(10);
10 type Price : Decimal(15, 2); // Adjusted to align with the schema
11 type Status : String(20); // Added type for Invoice Status
12
13 entity Suppliers : cuid {
14   key ID : UUID;
15   supplierId : SupplierID @readonly;
16   name : String(100); // Renamed from `Name` to `name` for consistency
17   contactInfo : String(200); // Renamed from `ContactInfo` to `contactInfo`
18 }
19
20 entity Invoices : cuid {
21   key ID : UUID;
22   invoiceId : InvoiceID @readonly;
23   supplier : Association to Suppliers; // Adjusted from SupplierID
24   amount : Price;
25   creationDateTime : Timestamp @cds.on.insert : $now;
26   dueDate : Date;
27   status : Status; // Added status field for Invoice
28 };
29
30 entity Payments : cuid {
31   key ID : UUID;
32   paymentId : PaymentID @readonly;
33   invoice : Association to Invoices; // Adjusted from InvoiceID
34   paymentDate : Date;
35   amountPaid : Price; // Used 'Price' type for consistency
36 };
```

Figure 5.3.2.1 - interaction.cds File for Three Entities such as Suppliers, Invoices (owned by Supplier) and Payments

Figure 5.3.2.1 represents the core database entity for the Account Payable subsystem, which consists of three entities such as Suppliers, Invoices (owned by Supplier) and Payments. Five types of variables such as SupplierID, InvoiceID, PaymentID, Price and Status are defined earlier to ensure the consistency which is all in string type. In the Suppliers entity, it consists of four variables such as ID which is in UUID form, supplierId, name and contactInfo. This entity is used to record the supplier's details. In the Invoices entity, it consists of seven variables as shown. One of the variable named supplier is used as association to the Suppliers entity which will later be used when calling the supplier information during invoice creation. Same goes to the variable named invoice in the Payments entity. All the variables named ID in these three entities which format are in UUID form are used as a unique identifier for a particular supplier, invoice and payment. The Invoices entity is used to record the invoice's details while Payments entity is used to record the payment's details.

The screenshot shows the SAP HANA Database Explorer interface. On the left, there is a sidebar with a tree view of database objects under 'Filter Instances'. The main area displays the schema for the table 'APP_INTERACTIONS_SUPPLIERS' in the schema 'FINANCEMODULE_HDI_FINANCEMODULE_DB_DEPLOYER_1'. The table has four columns: ID (NVARCHAR(36), Key, Not Null, Default NULL), SUPPLIERID (NVARCHAR(10)), NAME (NVARCHAR(100)), and CONTACTINFO (NVARCHAR(200)).

	Name	SQL Data Type	Key	Not Null	Default	Comment
1	ID	NVARCHAR(36)		X	NULL	
2	SUPPLIERID	NVARCHAR(10)			NULL	
3	NAME	NVARCHAR(100)			NULL	
4	CONTACTINFO	NVARCHAR(200)			NULL	

Figure 5.3.2.2 - Variables in Suppliers Entity and their corresponding Data Type as shown in Database

Figure 5.3.2.2 represents the variables name in Suppliers Entity and their corresponding data type as shown in SAP HANA Database Explorer.

The screenshot shows the SAP HANA Database Explorer interface. On the left, there's a sidebar with 'Filter Instances' and a tree view of database objects: Public Syms, Remote S., Job Schec, Sequence, Synonyms, Table Type, Tables, Tasks, Triggers, Views, and SharedDevK. Below this is a search bar for 'Search Table'. The main area is titled 'Raw Data' and shows a table titled 'Rows (5)'. The table has columns: ID, SUPPLIERID, NAME, and CONTACTINFO. The data is as follows:

	ID	SUPPLIERID	NAME	CONTACTINFO
1	ffa0b50b-2a6a-4867-af2b-96fa32c1f50	SUP1	ABC Supplies	contact@abc.com
2	92328331-d8fc-44a1-a3f0-bbf6b6899862	SUP2	Global Traders	support@globaltraders.com
3	1a1e404c-8ffe-4453-bc93-09d5b5f336c1	SUP3	TechMart	info@techmart.com
4	9d98ff81-6c5a-446e-8bc1-816b76210560	SUP4	OfficeHub	service@officehub.com
5	2d5680ce-1d0f-4919-b9ad-f69d7695c7e3	SUP5	bggnf	nynyfhmnym

Figure 5.3.2.3 - Data recorded in Suppliers Entity as shown in Database

Figure 5.3.2.3 represents all the data recorded in Suppliers Entity as shown in SAP HANA Database Explorer. All this data is recorded by using the create form and it will automatically be saved inside this database.

The screenshot shows the SAP HANA Database Explorer interface. On the left, there's a sidebar with icons for Public Syms, Remote S., Job Schec., Sequence, Synonyms, Table Type, Tables, Tasks, Triggers, Views, and SharedDevK. Below this is a search bar labeled 'Search Tables' with a magnifying glass icon. The main area has tabs for 'Filter Instances', 'Table Name', and 'Schema'. The 'Table Name' tab is selected, showing the table 'APP_INTERACTIONS_INVOICES' under the schema 'FINANCEMODULE_HDI_FINANCMODULE_DB_DEPLOYER_1'. There are also 'Open Data' and 'Columns', 'Indexes', 'Properties', and 'Runtime Information' tabs. The central part of the screen displays a table structure with columns: Name, SQL Data Type, Key, Not Null, Default, and Comment. The table has 7 rows:

Name	SQL Data Type	Key	Not Null	Default	Comment
1 ID	NVARCHAR(36)	1	X	NULL	
2 INVOICEID	NVARCHAR(10)			NULL	
3 SUPPLIER_ID	NVARCHAR(36)			NULL	
4 AMOUNT	DECIMAL(15,2)			NULL	
5 DUEDATE	DATE			NULL	
6 STATUS	NVARCHAR(20)			NULL	
7 CREATIONDATETIME	TIMESTAMP			NULL	

Figure 5.3.2.4 - Variables in Invoices (owned by supplier) Entity and their corresponding Data Type as shown in Database

Figure 5.3.2.4 represents the variables name in Invoices (owned by supplier) Entity and their corresponding data type as shown in SAP HANA Database Explorer.

The screenshot shows the SAP HANA Database Explorer interface. On the left, there is a sidebar with various database objects like Public Syms, Remote S, Job Sched, Sequence, Synonyms, Table Type, Tables, Tasks, Triggers, Views, and SharedDevk. The main area displays the 'Raw Data' tab for the 'APP_INTERACTIONS_INVOICED' table. The table has 15 rows of data. The columns are: ID, INVOICEID, SUPPLIER_ID, AMOUNT, DUEDATE, STATUS, and CREATIONDATETIME. The data includes various invoice IDs (e.g., INV3, INV4, INV1, INV2, INV2-1, INV2-2, INV5, INV5-1, INV5-2, INV6, INV6-1, INV7, INV7-1, INV8, INV8-1), supplier IDs (e.g., 1a1e404c-bffe-4453-bc93-09d5b5f336c1, 9d98f881-6c5a-446e-8bc1-816b76210560), amounts (e.g., 3000.00, 1000.00, 1500.00, 2000.00, 500.00, 250.00, 900.00, 450.00, 50.00, 11.00, 6.00, 400.00, 200.00, 50.00, 5.00), due dates (e.g., 2025-01-08, 2025-01-07, 2025-01-22, 2025-01-31, 2025-01-31, 2025-01-31, 2025-01-09, 2025-01-09, 2025-01-09, 2025-01-30, 2025-01-30, 2025-01-31, 2025-01-31, 2025-01-31), statuses (e.g., PAID, OVERDUE, PAID, PENDING (HALF)), and creation dates (e.g., 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-10, 2025-01-24).

ID	INVOICEID	SUPPLIER_ID	AMOUNT	DUEDATE	STATUS	CREATIONDATETIME
1	10114c79-862a-497e-b147-705c5bd928f1	INV3	1a1e404c-bffe-4453-bc93-09d5b5f336c1	3000.00	2025-01-08	PAID
2	374cb334-f1ac-44ac-9ca4-b679fedc02f8	INV4	9d98f881-6c5a-446e-8bc1-816b76210560	1000.00	2025-01-07	OVERDUE
3	320b879d-9093-406e-ac9e-35f1d715bcbe	INV1	ffab650b-2a6a-4867-a72b-95bfba32cf1f0	1500.00	2025-01-22	PAID
4	93530131-d4c2-4054-a8e3-66666d30cb26	INV2	92328331-d8fc-44a1-a3f0-bbf6b6899862	2000.00	2025-01-31	PAID
5	3f9715f4-e6e6-4e8e-b2f191a6b7609d933	INV2-1	92328331-d8fc-44a1-a3f0-bbf6b6899862	500.00	2025-01-31	PAID
6	bdf9cf60-459f-45bf-a01a-19956b9a5c67	INV2-2	92328331-d8fc-44a1-a3f0-bbf6b6899862	250.00	2025-01-31	PAID
7	418a1ea0-e45e-4d74-a282-9d2b8a4e225c	INV5	1a1e404c-bffe-4453-bc93-09d5b5f336c1	900.00	2025-01-09	PAID
8	8cd2cab0-7a10-4fc5-afab-64ab3040da11	INV5-1	1a1e404c-bffe-4453-bc93-09d5b5f336c1	450.00	2025-01-09	PAID
9	7e2918b7-f53d-4577-b460-10880266c32f	INV5-2	1a1e404c-bffe-4453-bc93-09d5b5f336c1	50.00	2025-01-09	PAID
10	988f51bf-a3ac-4405-8157-b11f92dd7992	INV6	1a1e404c-bffe-4453-bc93-09d5b5f336c1	11.00	2025-01-30	PAID
11	4fc741a5-fc19-448c-b434-3c29e3aec7a	INV6-1	1a1e404c-bffe-4453-bc93-09d5b5f336c1	6.00	2025-01-30	PAID
12	1d61adae-e89e-4bba-a024-87c378e7a30c	INV7	9d98f881-6c5a-446e-8bc1-816b76210560	400.00	2025-01-30	PAID
13	4309d594-3d76-4760-a9d9-76cfca334f7	INV7-1	9d98f881-6c5a-446e-8bc1-816b76210560	200.00	2025-01-30	PAID
14	f45c1a7e-4bc0-4d9e-baa7-0a07f1891c479	INV8	1a1e404c-bffe-4453-bc93-09d5b5f336c1	50.00	2025-01-31	PAID
15	2b97f66a-ad7c-4892-8612-a0f07c07e9e7	INV8-1	1a1e404c-bffe-4453-bc93-09d5b5f336c1	5.00	2025-01-31	PENDING (HALF)

Figure 5.3.2.5 - Data recorded in Invoices (owned by supplier) Entity as shown in Database

Figure 5.3.2.5 represents all the data recorded in Invoices (owned by supplier) Entity as shown in SAP HANA Database Explorer. All this data is recorded by using the create form and it will automatically be saved inside this database.

The screenshot shows the SAP HANA Database Explorer interface. On the left, there is a sidebar with various database objects: Public Syms, Remote S, Job Schec, Sequence, Synonyms, Table Type, Tables, Tasks, Triggers, Views, and SharedDev. Below this is a search bar labeled "Search Tables". The main area displays the schema of the "APP_INTERACTIONS_PAYMENTS" table. At the top, there are tabs for "Table Name" and "Schema". The "Table Name" tab is active, showing the table name "APP_INTERACTIONS_PAYMENTS", the schema "FINANCEMODULE_HDI_FINANCEMODULE_DB_DEPLOYER_1", and a "Open Data" button. Below these are tabs for "Columns", "Indexes", "Properties", and "Runtime Information". The "Columns" tab is selected, displaying a table with five columns:

	Name	SQL Data Type	Key	Not Null	Default	Comment
1	ID	NVARCHAR(36)		X	NULL	
2	PAYMENTID	NVARCHAR(10)			NULL	
3	INVOICE_ID	NVARCHAR(36)			NULL	
4	PAYMENTDATE	DATE			NULL	
5	AMOUNTPAID	DECIMAL(15,2)			NULL	

Figure 5.3.2.6 - Variables in Payments Entity and their corresponding Data Type as shown in Database

Figure 5.3.2.6 represents the variables name in Payments Entity and their corresponding data type as shown in SAP HANA Database Explorer.

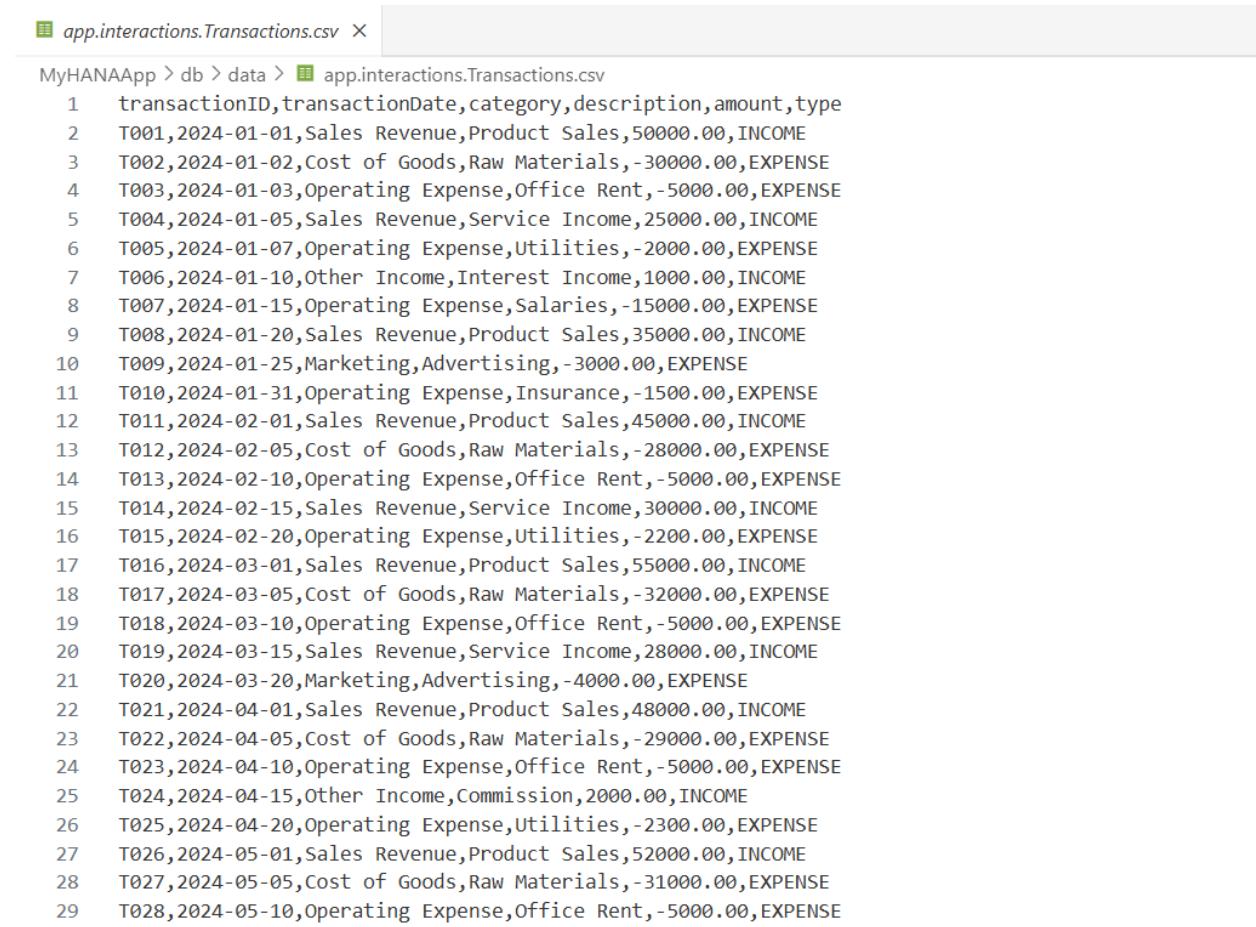
The screenshot shows the SAP HANA Database Explorer interface. On the left, there is a sidebar with various database objects like Public Syms, Remote S., Job Sched, Sequence, Synonyms, Table Types, Tables, Tasks, Triggers, and Views. Below that is a search bar labeled 'Search Tables' with a magnifying glass icon. The main area is titled 'Raw Data' and shows a table with 12 rows of data. The table has columns: ID, PAYMENTID, INVOICE_ID, PAYMENTDATE, and AMOUNTPAID. The data is as follows:

	ID	PAYMENTID	INVOICE_ID	PAYMENTDATE	AMOUNTPAID
1	07d92d6e-30e1-490d-a089-95bbcd9baef1	PAY1	320b879d-9093-406e-ac9e-35f1d715bcb	2025-01-11	1500.00
2	d7548f3f-d394-436f-9ca8-f183bc572038	PAY2	93530131-d4c2-4054-a8e3-68666d930cb26	2025-01-11	1500.00
3	d83e9a6a-709a-40e0-8a86-5160aee5b919	PAY3	3f9715f4-6e6e-4e8e-b21f-91ab7609d693	2025-01-11	250.00
4	acced62f-a1b1-4308-828c-19de7a50383f	PAY4	bdf9cf60-459f-45bf-a01a-19956b9a5c67	2025-01-11	250.00
5	2881d695-6567-4598-b09a-d62272267cba	PAY5	418a1ea0-e45e-4d74-a282-9d2b8aae225c	2025-01-11	450.00
6	b533208e-26c3-45ad-beef-d651a100d275	PAY6	8cd2cab0-7a10-4fc5-afa6-640b3040da11	2025-01-11	400.00
7	57ea5e7a-888d-4e4c-af14-4b09cad00078	PAY7	7e2918b7-f53d-4577-b460-10880266c32f	2025-01-11	50.00
8	04e4a569-e02f-40e6-83d3-5366fafcd03e	PAY8	988f51bfa3ac-4405-8157-b11f92dd7992	2025-01-16	5.00
9	8cb1d46e-128e-42aa-869e-69b1028c7ccf	PAY9	4fc741a5-fc19-448c-b434-3c29e3eca7a	2025-01-16	6.00
10	bd6ede73-f08-4f0d-860a-8b4f15e5f3e	PAY10	1d61adae-e89e-4bba-a024-87c378e7a30c	2025-01-16	200.00
11	a33fb4be-d1cf-4a5d-b974-abda78b6e2cb	PAY10	4309d594-3d76-4760-ae9d-78cfca334f7	2025-01-16	200.00
12	10e155ae-2f4b-44ad-a4bc-214b166e61e7	PAY10	f45c1a7e-4bc0-4d9e-baa7-0af71891c479	2025-01-25	45.00

Figure 5.3.2.7 - Data recorded in Payments Entity as shown in Database

Figure 5.3.2.7 represents all the data recorded in Payments Entity as shown in SAP HANA Database Explorer. All this data is recorded by using the create form and it will automatically be saved inside this database.

5.3.3 Database for Finance Report Subsystem



The screenshot shows a CSV file named "Transactions.csv" with 29 rows of transaction data. The columns are transactionID, transactionDate, category, description, amount, and type. The data includes various financial entries such as Sales Revenue, Cost of Goods, Operating Expense, and Other Income.

transactionID	transactionDate	category	description	amount	type
T001	2024-01-01	Sales Revenue	Product Sales	50000.00	INCOME
T002	2024-01-02	Cost of Goods	Raw Materials	-30000.00	EXPENSE
T003	2024-01-03	Operating Expense	Office Rent	-5000.00	EXPENSE
T004	2024-01-05	Sales Revenue	Service Income	25000.00	INCOME
T005	2024-01-07	Operating Expense	Utilities	-2000.00	EXPENSE
T006	2024-01-10	Other Income	Interest Income	1000.00	INCOME
T007	2024-01-15	Operating Expense	Salaries	-15000.00	EXPENSE
T008	2024-01-20	Sales Revenue	Product Sales	35000.00	INCOME
T009	2024-01-25	Marketing	Advertising	-3000.00	EXPENSE
T010	2024-01-31	Operating Expense	Insurance	-1500.00	EXPENSE
T011	2024-02-01	Sales Revenue	Product Sales	45000.00	INCOME
T012	2024-02-05	Cost of Goods	Raw Materials	-28000.00	EXPENSE
T013	2024-02-10	Operating Expense	Office Rent	-5000.00	EXPENSE
T014	2024-02-15	Sales Revenue	Service Income	30000.00	INCOME
T015	2024-02-20	Operating Expense	Utilities	-2200.00	EXPENSE
T016	2024-03-01	Sales Revenue	Product Sales	55000.00	INCOME
T017	2024-03-05	Cost of Goods	Raw Materials	-32000.00	EXPENSE
T018	2024-03-10	Operating Expense	Office Rent	-5000.00	EXPENSE
T019	2024-03-15	Sales Revenue	Service Income	28000.00	INCOME
T020	2024-03-20	Marketing	Advertising	-4000.00	EXPENSE
T021	2024-04-01	Sales Revenue	Product Sales	48000.00	INCOME
T022	2024-04-05	Cost of Goods	Raw Materials	-29000.00	EXPENSE
T023	2024-04-10	Operating Expense	Office Rent	-5000.00	EXPENSE
T024	2024-04-15	Other Income	Commission	2000.00	INCOME
T025	2024-04-20	Operating Expense	Utilities	-2300.00	EXPENSE
T026	2024-05-01	Sales Revenue	Product Sales	52000.00	INCOME
T027	2024-05-05	Cost of Goods	Raw Materials	-31000.00	EXPENSE
T028	2024-05-10	Operating Expense	Office Rent	-5000.00	EXPENSE

Figure 5.3.3.1 - CSV File Database for Finance Report SubSystem

The CSV File Database for Finance Report SubSystem is named as Transactions entity and primarily involved in data retrieval processes in which financial transactions are retrieved from the database and utilized for generating P&L statements based on the selected reporting period. To integrate with the subsystems in the Finance module, this database file will associate with the Invoice entity in a relation in which each Transactions instance will be held by exactly one Invoice instance.

5.4 Coding of The System's Main Functions

5.4.1 Coding for AR Subsystem

Table 5.4.1.1 - Coding and Description of AR Subsystem

Description	Coding of The System's Main Function
<p>The UI.FieldGroup annotations are used to organize related fields into logical groups for better display and interaction within the user interface.</p> <ul style="list-style-type: none"> ● Payment Details Field Group ● Customer Details Field Group ● Invoice Details Field Group 	<pre> 1 using CatalogService as service from '../../srv/interaction_srv'; 2 3 annotate service.Invoices_Header with @(4 5 UI.FieldGroup #GeneratedGroup3 : { 6 \$Type : 'UI.FieldGroupType', 7 Data : [8 { 9 \$Type : 'UI.DataField', 10 Label : 'Payment Date', 11 Value : paymentDate, 12 }, 13 { 14 \$Type : 'UI.DataField', 15 Label : 'Payment Method / Term Of Payment', 16 Value : paymentMethod, 17 }, 18 { 19 \$Type : 'UI.DataField', 20 Label : 'Amount Paid', 21 Value : amountPaid, 22 }, 23], 24 }, 25 26 UI.FieldGroup #GeneratedGroup2 : { 27 \$Type : 'UI.FieldGroupType', 28 Data : [29 { 30 \$Type : 'UI.DataField', 31 Label : 'Customer ID', 32 Value : customerID, 33 }, 34 { 35 \$Type : 'UI.DataField', 36 Label : 'Customer Name', 37 Value : customer.name, 38 }, 39 { 40 \$Type : 'UI.DataField', 41 Label : 'Customer Type', 42 Value : customer.customertype, 43 }, 44 { 45 \$Type : 'UI.DataField', 46 Label : 'Contact Number', 47 Value : customer.contactNumber, 48 }, </pre>

```
49  {
50      $Type : 'UI.DataField',
51      Label : 'Email',
52      Value : customer.email,
53  },
54  {
55      $Type : 'UI.DataField',
56      Label : 'Address',
57      Value : customer.address,
58  ],
59 },
60 },
61
62 //in the invoice details
63 UI.FieldGroup #GeneratedGroup : {
64     $Type : 'UI.FieldGroupType',
65     Data : [
66         {
67             $Type : 'UI.DataField',
68             Label : 'Invoice ID',
69             Value : invoiceID,
70         },
71         {
72             $Type : 'UI.DataField',
73             Label : 'Invoice Date',
74             Value : invoiceDate,
75         },
76         {
77             $Type : 'UI.DataField',
78             Label : 'Total Amount',
79             Value : totalAmount,
80         },
81         {
82             $Type : 'UI.DataField',
83             Label : 'Payment Status',
84             Value : paymentStatus,
85         },
86     ],
87 },
88 
```

The **UI.Facets** annotation enables tab-like navigation between the different field groups:

- **Invoice Details Facet:**
Displays the invoice-specific details grouped in the GeneratedGroup.
- **Customer Details Facet:**
Shows the customer-related fields grouped in the GeneratedGroup2.
- **Payment Details Facet:**
Focuses on payment-related fields from GeneratedGroup3.

These facets allow users to view invoice information in an organized and segmented way.

```

90 |     UI.Facets : [
91 |     {
92 |         $Type : 'UI.ReferenceFacet',
93 |         Label : 'Invoice Details',
94 |         Target : '@UI.FieldGroup#GeneratedGroup',
95 |     },
96 |     {
97 |         $Type : 'UI.ReferenceFacet',
98 |         Label : 'Customer Details',
99 |         Target : '@UI.FieldGroup#GeneratedGroup2',
100 |    },
101 |    {
102 |        $Type : 'UI.ReferenceFacet',
103 |        Label : 'Payment Details',
104 |        Target : '@UI.FieldGroup#GeneratedGroup3',
105 |    },
106 |    ...
107 | ]

```

The **UI.LineItem** annotation is used to configure the columns displayed in a table for the invoice header list:

- Fields like Invoice ID, Customer ID, Invoice Date, Total Amount (MYR), Payment Status, and Amount Paid are included.
- These columns make it easier for users to view and compare invoice details at a glance.

```

108  UI.LineItem : [
109    {
110      $Type : 'UI.DataField',
111      Label : 'Invoice ID',
112      Value : invoiceID,
113    },
114    {
115      $Type : 'UI.DataField',
116      Label : 'Customer ID',
117      Value : customerID,
118    },
119    {
120      $Type : 'UI.DataField',
121      Label : 'Invoice Date',
122      Value : invoiceDate,
123    },
124    {
125      $Type : 'UI.DataField',
126      Label : 'Total Amount (MYR)',
127      Value : totalAmount,
128    },
129    {
130      $Type : 'UI.DataField',
131      Label : 'Payment Status',
132      Value : paymentStatus,
133    },
134    {
135      $Type : 'UI.DataField',
136      Label : 'Amount Paid',
137      Value : amountPaid,
138    },
139  ],
140 );

```

The **Common.ValueList** annotation enables a dropdown feature for the Customer ID field. When selecting a customer, the dropdown fetches customer data from the Customers collection.

This feature minimizes manual entry errors and improves efficiency when creating or editing invoices.

```
143     annotate service.Invoices_Header with {
144         customerID @Common.ValueList : {
145             $Type : 'Common.ValueListType',
146             CollectionPath : 'Customers',
147             Parameters : [
148                 {
149                     $Type : 'Common.ValueListParameterInOut',
150                     LocalDataProperty : customerID,
151                     ValueListProperty : 'customerID',
152                 },
153                 {
154                     $Type : 'Common.ValueListParameterInOut',
155                     LocalDataProperty : customer.name,
156                     ValueListProperty : 'name',
157                 },
158                 {
159                     $Type : 'Common.ValueListParameterInOut',
160                     LocalDataProperty : customer.customertype,
161                     ValueListProperty : 'customertype',
162                 },
163                 {
164                     $Type : 'Common.ValueListParameterInOut',
165                     LocalDataProperty : customer.contactNumber,
166                     ValueListProperty : 'contactNumber',
167                 },
168                 {
169                     $Type : 'Common.ValueListParameterInOut',
170                     LocalDataProperty : customer.email,
171                     ValueListProperty : 'email',
172                 },
173                 {
174                     $Type : 'Common.ValueListParameterInOut',
175                     LocalDataProperty : customer.address,
176                     ValueListProperty : 'address',
177                 },
178             ],
179         }
180     };
```

5.4.2 Coding for AP Subsystem

Table 5.4.2.1 - Coding and Description of AP Subsystem

Description	Coding of The System's Main Function
Supplier Invoices Page	
<p>The UI.HeaderInfo annotations are used to define how the entity is displayed in the supplier invoices list</p>	<pre> 1 using PayableService as service from '../../../../../srv/interaction_svr'; 2 3 annotate service.Invoices with @(4 UI.HeaderInfo: { 5 Title: { 6 \$Type: 'UI.DataField', 7 Value: supplier.name, 8 }, 9 TypeName: 'Invoice', 10 TypeNamePlural: 'Invoices', 11 Description: { Value: status } 12 }, </pre>

The **UI.FieldGroup** annotations are used to organize related fields into logical groups for better display and interaction within the user interface.

- Invoice and corresponding supplier's details field group in the Supplier Invoices Page

```

13     UI.FieldGroup #GeneralInfo: {
14         $Type: 'UI.FieldGroupType',
15         Data: [
16             {
17                 $Type: 'UI.DataField',
18                 Label: 'Supplier',
19                 Value: supplier_ID
20             },
21             {
22                 $Type: 'UI.DataField',
23                 Label: 'Supplier Name',
24                 Value: supplier.name
25             },
26             {
27                 $Type: 'UI.DataField',
28                 Label: 'Supplier Contact',
29                 Value: supplier.contactInfo
30             },
31             {
32                 $Type: 'UI.DataField',
33                 Label: 'Invoice ID',
34                 Value: invoiceId
35             },
36             {
37                 $Type: 'UI.DataField',
38                 Label: 'Amount',
39                 Value: amount
40             },
41             {
42                 $Type: 'UI.DataField',
43                 Label: 'Due Date',
44                 Value: dueDate
45             },
46             {
47                 $Type: 'UI.DataField',
48                 Label: 'Status',
49                 Value: status
50             }
51         ]
52     }

```

<p>The UI.Facets annotation enables tab-like navigation between the different field groups:</p> <ul style="list-style-type: none"> • General Information <p>Facet: Displays the supplier invoice-specific details grouped in the GeneralInfo.</p> <p>This facet allows users to view supplier invoice information in an organized and segmented way.</p>	<pre> 53 UI.Facets: [54 { 55 \$Type: 'UI.ReferenceFacet', 56 ID: 'GeneralInfo', 57 Label: 'General Information', 58 Target: '@UI.FieldGroup#GeneralInfo', 59 } 60], </pre>
<p>The UI.LineItem annotation is used to configure the columns displayed in a table for the supplier invoice header list:</p> <ul style="list-style-type: none"> • Fields like Invoice, Amount, Due Date, and Status are included. • These columns make it easier for users to view and compare supplier invoice details at a glance. 	<pre> 61 UI.LineItem: [62 { 63 \$Type: 'UI.DataField', 64 Label: 'Invoice', 65 Value: invoiceId, 66 }, 67 { 68 \$Type: 'UI.DataField', 69 Label: 'Amount', 70 Value: amount, 71 }, 72 { 73 \$Type: 'UI.DataField', 74 Label: 'Due Date', 75 Value: dueDate, 76 }, 77 { 78 \$Type: 'UI.DataField', 79 Label: 'Status', 80 Value: status, 81 } 82], 83); 84 </pre>

The **Common.ValueList** annotation enables a dropdown feature for the Supplier and status field:

- When selecting a supplier, the dropdown fetches supplier data from the Suppliers collection
- The dropdown menu for status showing the three type of status that can be selected such as PAID, PENDING and OVERDUE

This feature minimizes manual entry errors and improves efficiency when creating or editing supplier invoices.

```

85  annotate service.Invoices with {
86    supplier @Common.ValueList: {
87      $Type: 'Common.ValueListType',
88      CollectionPath: 'Suppliers',
89      Parameters: [
90        {
91          $Type: 'Common.ValueListParameterInOut',
92          LocalDataProperty: supplier_ID, // Maps the association to Suppliers
93          ValueListProperty: 'ID' // Supplier ID in Suppliers entity
94        },
95        {
96          $Type: 'Common.ValueListParameterInOut',
97          LocalDataProperty: supplier.supplierId, // Maps the association to Suppliers
98          ValueListProperty: 'supplierId' // Supplier ID in Suppliers entity
99        },
100       {
101         $Type: 'Common.ValueListParameterInOut',
102         LocalDataProperty: supplier.name, // Maps the association to Suppliers
103         ValueListProperty: 'name' // ID field in Suppliers
104       },
105       {
106         $Type: 'Common.ValueListParameterInOut',
107         LocalDataProperty: supplier.contactInfo, // Maps the association to Suppliers
108         ValueListProperty: 'contactInfo' // ID field in Suppliers
109       }
110     ]
111   };
112 };
113
114  annotate service.Invoices with {
115    status @Common.ValueList: {
116      $Type: 'Common.ValueListType',
117      CollectionPath: 'StatusOptions', // Define the value list for status
118      Parameters: [
119        {
120          $Type: 'Common.ValueListParameterInOut',
121          LocalDataProperty: status, // Bind the association itself
122          ValueListProperty: 'Status', // Map to the Employees' GUID
123        }
124      ]
125    };
126  };

```

Suppliers Page

The **UI.HeaderInfo** annotations are used to define how the entity is displayed in the suppliers list

```

1  using PayableService as service from '../../../../../srv/interaction_srv';
2
3  annotate service.Suppliers with @(
4    UI.HeaderInfo: {
5      Title: {
6        $Type: 'UI.DataField',
7        Value: name,
8      },
9      TypeName: 'Supplier',
10     TypeNamePlural: 'Suppliers'
11   },

```

The **UI.FieldGroup** annotations are used to organize related fields into logical groups for better display and interaction within the user interface.

- Supplier details field group in the Suppliers Page

```

12   UI.FieldGroup #GeneralInfo: {
13     $Type: 'UI.FieldGroupType',
14     Data: [
15       {
16         $Type: 'UI.DataField',
17         Label: 'Supplier ID',
18         Value: supplierId
19       },
20       {
21         $Type: 'UI.DataField',
22         Label: 'Supplier Name',
23         Value: name
24       },
25       {
26         $Type: 'UI.DataField',
27         Label: 'Supplier Contact Information',
28         Value: contactInfo
29       }
30     ],
31   },

```

The **UI.Facets** annotation enables tab-like navigation between the different field groups:

- General Information**
Facet: Displays the supplier-specific details grouped in the GeneralInfo.

```

32   UI.Facets: [
33   {
34     $Type: 'UI.ReferenceFacet',
35     ID: 'GeneralInfo',
36     Label: 'General Information',
37     Target: '@UI.FieldGroup#GeneralInfo',
38   }
39 ],

```

This facet allows users to view supplier information in an organized and segmented way.

The **UI.LineItem** annotation is used to configure the columns displayed in a table for the supplier header list:

- Fields like Supplier ID, Supplier Name, and Supplier Contact Info are included.
- These columns make it easier for users to view and compare supplier details at a glance.

```

40     UI.LineItem: [
41         {
42             $Type: 'UI.DataField',
43             Label: 'Supplier ID',
44             Value: supplierId,
45         },
46         {
47             $Type: 'UI.DataField',
48             Label: 'Supplier Name',
49             Value: name,
50         },
51         {
52             $Type: 'UI.DataField',
53             Label: 'Supplier Contact Info',
54             Value: contactInfo,
55         }
56     ],
57 );

```

Payments Page

The **UI.HeaderInfo** annotations are used to define how the entity is displayed in the payments list

```

1  using PayableService as service from '../../../../../srv/interaction_svr';
2
3  annotate service.Payments with @(
4      UI.HeaderInfo: {
5          Title: {
6              $Type: 'UI.DataField',
7              Value: paymentId,
8          },
9          TypeName: 'Payment',
10         TypeNamePlural: 'Payments'
11     },

```

The **UI.FieldGroup** annotations are used to organize related fields into logical groups for better display and interaction within the user interface.

- Payment and corresponding invoice's details field group in the Payments Page

```

12     UI.FieldGroup #GeneratedGroup : [
13         $Type : 'UI.FieldGroupType',
14         Data : [
15             {
16                 $Type : 'UI.DataField',
17                 Label : 'Invoice',
18                 Value : invoice_ID,
19             },
20             {
21                 $Type : 'UI.DataField',
22                 Label : 'Invoice ID',
23                 Value : invoice.invoiceId,
24             },
25             {
26                 $Type : 'UI.DataField',
27                 Label : 'Payment ID',
28                 Value : paymentId,
29             },
30             {
31                 $Type : 'UI.DataField',
32                 Label : 'Payment Date',
33                 Value : paymentDate,
34             },
35             {
36                 $Type : 'UI.DataField',
37                 Label : 'Amount Paid',
38                 Value : amountPaid,
39             },
40         ],
41     },

```

The **UI.Facets** annotation enables tab-like navigation between the different field groups:

- **General Information Facet:** Displays the payment-specific details grouped in the GeneratedGroup.

This facet allows users to view

```

42     UI.Facets : [
43         {
44             $Type : 'UI.ReferenceFacet',
45             ID : 'GeneratedFacet1',
46             Label : 'General Information',
47             Target : '@UI.FieldGroup#GeneratedGroup',
48         },
49     ],

```

payment information in an organized and segmented way.	
<p>The UI.LineItem annotation is used to configure the columns displayed in a table for the payment header list:</p> <ul style="list-style-type: none"> Fields like Payment ID, Invoice ID, Payment Date, and Amount Paid are included. These columns make it easier for users to view and compare payment details at a glance. 	<pre> 50 UI.LineItem : [51 { 52 \$Type : 'UI.DataField', 53 Label : 'Payment ID', 54 Value : paymentId, 55 }, 56 { 57 \$Type : 'UI.DataField', 58 Label : 'Invoice ID', 59 Value : invoice.invoiceId, 60 }, 61 { 62 \$Type : 'UI.DataField', 63 Label : 'Payment Date', 64 Value : paymentDate, 65 }, 66 { 67 \$Type : 'UI.DataField', 68 Label : 'Amount Paid', 69 Value : amountPaid, 70 }, 71], 72); </pre>

The **Common.ValueList** annotation enables a dropdown feature for the Invoice field. When selecting an invoice, the dropdown fetches supplier invoice data from the SupplierInvoices collection.

This feature minimizes manual entry errors and improves efficiency when creating or editing payments.

```
74  annotate service.Payments with {
75      invoice @Common.ValueList : {
76          $Type : 'Common.ValueListType',
77          CollectionPath : 'Invoices',
78          Parameters : [
79              {
80                  $Type : 'Common.ValueListParameterInOut',
81                  LocalDataProperty : invoice_ID,
82                  ValueListProperty : 'ID',
83              },
84              {
85                  $Type : 'Common.ValueListParameterDisplayOnly',
86                  ValueListProperty : 'invoiceId',
87              },
88              {
89                  $Type : 'Common.ValueListParameterDisplayOnly',
90                  ValueListProperty : 'amount',
91              },
92              {
93                  $Type : 'Common.ValueListParameterDisplayOnly',
94                  ValueListProperty : 'creationDateTime',
95              },
96              {
97                  $Type : 'Common.ValueListParameterDisplayOnly',
98                  ValueListProperty : 'dueDate',
99              },
100             {
101                 $Type : 'Common.ValueListParameterDisplayOnly',
102                 ValueListProperty : 'status',
103             },
104         ],
105     };
106 }
```

5.4.3 Coding for GL Subsystem

Table 5.4.3.1 - Coding and Description of GL Subsystem

Description	Coding of The System's Main Function
<p>_generateYears(): The current year and the four years prior are generated as an array, which is then returned as an object with a key (the year) and text (the year as a string).</p> <p>_generateMonths(): Creates an array of objects that each include a key (the month number) and text (the month abbreviation), signifying the twelve months of the year.</p>	<pre> 18 _generateYears: function() { 19 const currentYear = new Date().getFullYear(); 20 return Array.from({ length: 5 }, (_, i) => ({ 21 key: currentYear - i, 22 text: (currentYear - i).toString() 23 })); 24 }, 25 26 _generateMonths: function() { 27 return [28 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 29 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC' 30].map((month, index) => ({ key: index + 1, text: month })); 31 }, </pre>
<p>The onGeneratePress() function creates a filter string to collect transaction data from an OData service after retrieving the chosen year and month from the view and verifying that a year (and month, if appropriate) has been selected. It responds to the potential that no transactions were located and shows the relevant messages. The</p>	<pre> 33 onGeneratePress: async function() { 34 try { 35 const viewModel = this.getView().getModel("viewModel"); 36 const selectedYear = this.byId("yearSelect").getSelectedKey(); 37 const selectedMonth = viewModel.getProperty("/isMonthly") ? 38 this.byId("monthSelect").getSelectedKey() : null; 39 40 if (!selectedYear) { 41 MessageBox.error("Please select a year"); 42 return; 43 } 44 if (viewModel.getProperty("/isMonthly") && !selectedMonth) { 45 MessageBox.error("Please select a month"); 46 return; 47 } 48 49 // Fetch transactions using OData V4 50 const oModel = this.getOwnerComponent().getModel(); 51 let filterStr = `year(transactionDate) eq \${selectedYear}`; 52 if (selectedMonth) { 53 filterStr += ` and month(transactionDate) eq \${selectedMonth}`; 54 } 55 56 const transactions = await oModel.bindList("/Transactions", null, null, null, 57 { \$filter: filterStr 58 }).requestContexts(0, 1000); 59 60 const data = transactions.map(context => context.getObject()); 61 console.log("Retrieved Transactions: ", data); 62 63 if (!data.length) { 64 MessageBox.information("No transactions found for the selected period."); 65 return; 66 } 67 68 this._showLedger(data, selectedYear, selectedMonth); 69 } catch (error) { 70 MessageBox.error("Error generating General Ledger: " + error.message); 71 } 72 }, </pre>

transaction data is passed to the `_showLedger()` function for additional display if data is found. Message boxes are used to capture and display process errors.

For the specified transactions, the `_showLedger()` function creates a printable ledger that is structured for viewing in a new browser window. Based on the chosen year and month, it calculates the period and formats monetary figures in Malaysian Ringgit (MYR). As it iterates over transactions, it computes the total sums of debits and credits and generates table rows including formatted dates, categories, descriptions, and debit/credit values. The resulting rows are then combined into an HTML table for display.

A printable General Ledger for AK MAJU RESOURCES SDN. BHD is produced by this HTML template. It consists of:

- Company Letterhead: Shows

```

74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
    _showLedger: function(transactions, year, month) {
      const printWindow = window.open('', '_blank');

      const formatMYR = amount => `RM ${parseFloat(amount).toFixed(2)}`;
      const getMonthName = monthNum => [
        'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN',
        'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC'
      ][monthNum - 1];

      const period = month ? `${getMonthName(month)} ${year}` : year;
      let totalDebit = 0, totalCredit = 0;

      const rows = transactions.map(t => {
        const debit = t.amount < 0 ? formatMYR(Math.abs(t.amount)) : "-";
        const credit = t.amount > 0 ? formatMYR(t.amount) : "-";
        if (t.amount < 0) totalDebit += Math.abs(t.amount);
        else totalCredit += t.amount;
      });

      return `
        <tr>
          <td>${new Date(t.transactionDate).toLocaleDateString()}</td>
          <td>${t.category}</td>
          <td>${t.description}</td>
          <td>${debit}</td>
          <td>${credit}</td>
        </tr>`;
      }).join('');
    }
  
```

```

212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
<body>
  <h2>General Ledger</h2>
  <div class="container">
    <div class="letterhead">
      <div class="logo-section">
        
      </div>
      <div class="company-details">
        <div class="company-name">AK MAJU RESOURCES SDN. BHD.</div>
        <div>No. 38 & 41 Jalan Utama 3/2, Pusat Komersial Sri Utama,</div>
        <div>Segamat, Johor, Malaysia- 85000</div>
        <div>07-9310717 , 010-2218224</div>
        <div>akmaju.acc@gmail.com</div>
        <div>Company No.: <span style="margin-left: 20px;">1088436 K</span></div>
      </div>
    </div>
  </div>

```

<p>the name, address, phone number, logo, and contact information.</p> <ul style="list-style-type: none"> • Transaction details (Date, Category, Description, Debit, and Credit) are listed in the Table of Transactions. • Calculation of Net Income: Shows the net income (total credit less total debit). • Print Button: By pressing a button, users can print the ledger. <p>Transaction rows are dynamically inserted using the \${rows} placeholder, and correct currency formatting is guaranteed by \${formatMYR(netIncome)}.</p>	<pre> 229 <!--<p>Period: \${period}</p>--> 230 <table> 231 <thead> 232 <tr> 233 <th>Date</th> 234 <th>Category</th> 235 <th>Description</th> 236 <th>Debit</th> 237 <th>Credit</th> 238 </tr> 239 </thead> 240 <tbody> 241 \${rows} 242 </tbody> 243 <tfoot> 244 <tr class="net-income"> 245 <td colspan="3">Net Income</td> 246 <td colspan="2">\${formatMYR(netIncome)}</td> 247 </tr> 248 </tfoot> 249 </table> 250 <button class="print-button no-print" onclick="window.print()">Print</button> 251 </div> 252 </pre>
<p>A printable General Ledger is created and shown in a new browser window by these two lines.</p>	<pre> 255 printWindow.document.write(html); 256 printWindow.document.close(); </pre>

5.4.4 Coding for Finance Report Subsystem

Table 5.4.4.1 - Coding and Description of Finance Report Subsystem

Description	Coding of The System's Main Function
<p>The onInit: function () is an initialization function that establishes the view model with essential configurations for the Finance Report interface. It includes the report type selection (monthly/yearly), generation of year options for the past 5 years, and month selection options (JAN to DEC).</p>	<pre> 19 onInit: function () { 20 const viewModel = new JSONModel({ 21 isMonthly: false, 22 years: this._generateYears(), 23 months: this._generateMonths() 24 }); 25 this.getView().setModel(viewModel, "viewModel"); 26 }, 27 28 _generateYears: function() { 29 const currentYear = new Date().getFullYear(); 30 const years = []; 31 for (let i = 0; i < 5; i++) { 32 years.push({ 33 key: currentYear - i, 34 text: (currentYear - i).toString() 35 }); 36 } 37 return years; 38 }, 39 40 _generateMonths: function() { 41 const months = [42 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 43 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC' 44]; 45 return months.map((month, index) => ({ 46 key: index + 1, 47 text: month 48 })); 49 </pre>
<p>The onGeneratePress: async function() is the P&L statement generation process that handles the core functionality of the system. It performs period selection validation, retrieves transaction data based on the selected period, checks for data availability, and processes the data for P&L statement generation.</p>	<pre> 56 onGeneratePress: async function() { 57 try { 58 const viewModel = this.getView().getModel("viewModel"); 59 const selectedYear = this.byId("yearSelect").getSelectedKey(); 60 const selectedMonth = viewModel.getProperty("isMonthly") ? 61 this.byId("monthSelect").getSelectedKey() : null; 62 63 if (!selectedYear) { 64 MessageBox.error("Please select a year"); 65 return; 66 } 67 68 if (viewModel.getProperty("/isMonthly") && !selectedMonth) { 69 MessageBox.error("Please select a month"); 70 return; 71 } 72 73 // Get all transactions 74 const oModel = this.getOwnerComponent().getModel(); 75 let filterStr = `year(transactionDate) eq \${selectedYear}`; 76 if (selectedMonth) { 77 filterStr += ` and month(transactionDate) eq \${selectedMonth}`; 78 } 79 80 // Fetch transactions using OData V4 81 const transactions = await oModel.bindList("/Transactions", null, null, null, { 82 \$filter: filterStr 83 }).requestContexts(0, 1000); // Adjust the number based on your data size </pre>

	<pre> 85 // Convert contexts to plain objects 86 const data = transactions.map(context => context.getObject()); 87 88 // Check if data exists 89 if (!data data.length === 0) { 90 const period = selectedMonth ? 91 `\${this._generateMonths()[selectedMonth - 1].text} \${selectedYear}` : 92 selectedYear; 93 94 MessageBox.information(95 `No transactions found for the selected period: \${period}`, 96 { 97 title: "No Data Available", 98 actions: [MessageBox.Action.OK], 99 emphasizedAction: MessageBox.Action.OK, 100 onClose: function() { 101 } 102 } 103); 104 return; 105 } 106 107 // Generate and show P&L statement 108 const pData = this._generatePLData(data, selectedYear, selectedMonth); 109 this._showPLStatement(pData); 110 111 } catch (error) { 112 MessageBox.error("Error generating P&L statement: " + error.message); 113 } 114 </pre>
The <code>_generatePLData: function()</code> is the data processing function that organizes transaction data into the P&L statement format. It separates and groups transactions by category (income and expenses), calculates sectional totals, and determines the final net profit or loss.	<pre> 116 _generatePLData: function(transactions, year, month) { 117 // Get month name function 118 const getMonthName = (monthNum) => { 119 const months = [120 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 121 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC' 122]; 123 return months[monthNum - 1]; 124 }; 125 126 // Group transactions by category 127 const incomeTransactions = transactions.filter(t => t.type === 'INCOME'); 128 const expenseTransactions = transactions.filter(t => t.type === 'EXPENSE'); 129 130 // Calculate income by category 131 const incomeByCategory = this._groupTransactionsByCategory(incomeTransactions); 132 const expensesByCategory = this._groupTransactionsByCategory(expenseTransactions); 133 134 // Calculate totals 135 const totalIncome = object.values(incomeByCategory).reduce((sum, amount) => sum + amount, 0); 136 const totalExpenses = object.values(expensesByCategory).reduce((sum, amount) => sum + Math.abs(amount), 0); 137 138 return { 139 period: month ? `\${getMonthName(month)} \${year}` : year.toString(), 140 sections: [141 { 142 title: "Revenue", 143 items: object.entries(incomeByCategory).map(([category, amount]) => ({ 144 description: category, 145 amount: amount 146 })), 147 total: totalIncome 148 }, 149 { 150 title: "Expenses", 151 items: object.entries(expensesByCategory).map(([category, amount]) => ({ 152 description: category, 153 amount: Math.abs(amount) // Make expenses positive for display 154 })), 155 total: totalExpenses 156 } 157], 158 netIncome: totalIncome - totalExpenses 159 }; 160 }, 161 162 _groupTransactionsByCategory: function(transactions) { 163 return transactions.reduce((groups, transaction) => { 164 const category = transaction.category; 165 if (!groups[category]) { 166 groups[category] = 0; 167 } 168 groups[category] += parseFloat(transaction.amount); 169 }, {}); 170 }, </pre>

The [showPLStatement: function\(\)](#) creates a formatted P&L statement with company letterhead, statement header, revenue, expenses sections and net profit calculation. The statement is styled with CSS to ensure proper formatting and layout. The function also implements color-coding for amounts where income values are displayed in green and expense values in red to enhance the readability of the P&L statement.

```
_showPLStatement: function(plData) {
  // Function to format currency in MYR (just changing the symbol)
  const formatMYR = (amount) => {
    return `RM ${Math.abs(amount).toFixed(2)}`;
  };

  const printWindow = window.open('', '_blank');

  const logoPath = sap.ui.require.toUrl("financereport/images/company-logo.png");

  const html = `
    <!DOCTYPE html>
    <html>
      <head>
        <title>PROFIT & LOSS STATEMENT - ${[plData.period.toUpperCase()]}</title>
        <style>
          body {
            font-family: Arial, sans-serif;
            margin: 40px auto;
            max-width: 800px;
            padding: 20px;
            background-color: #f5f5f5;
          }
          .container {
            background-color: white;
            padding: 40px;
            border-radius: 8px;
            box-shadow: 0 2px 4px rgba(0,0,0,0.1);
          }
          .header {
            padding-bottom: 5px;
            border-bottom: 3px double #333;
            position: relative;
          }
          .header-line {
            display: flex;
            justify-content: center;
            align-items: center;
            padding: 0 15px;
            position: relative;
          }
          .statement-title {
            font-size: 14px;
            font-weight: bold;
            position: absolute;
            left: 50%;
            transform: translateX(-50%);
          }
          .period {
            font-size: 12px;
            margin-left: auto;
            font-color: black;
            font-weight: bold;
          }
        </style>
      </head>
      <body>
        <div class="container">
          <div class="header">
            <div class="header-line">
              <div class="statement-title">${plData.period}</div>
              <div class="period">${plData.period}</div>
            </div>
          </div>
          <div class="section">
            <div class="section-title">${plData.period}</div>
            <div class="item-description">${plData.item}</div>
            <div class="total-label">Total</div>
            <div class="net-income">${plData.netIncome}</div>
            <div class="item">${plData.item}</div>
          </div>
        </div>
      </body>
    </html>
  `;
  printWindow.document.write(html);
  printWindow.print();
  printWindow.close();
}
```

```

254     margin: 0;
255     display: flex;
256     justify-content: space-between;
257     padding: 12px 15px;
258     border-bottom: 1px solid #eee;
259     font-size: 14px;
260   }
261   .item:hover {
262     background-color: #f8f9fa;
263   }
264   .total {
265     font-weight: bold;
266     display: flex;
267     justify-content: space-between;
268     padding: 15px;
269     background-color: #f8f9fa;
270     border-top: 2px solid #333;
271     margin-top: 10px;
272     font-size: 14px;
273   }
274   .amount {
275     font-family: 'Courier New', monospace;
276     font-weight: bold;
277     font-size: 12px;
278   }
279   .net-income {
280     font-size: 1.2em;
281     font-weight: bold;
282     border-top: 3px double #333;
283     padding: 20px 15px;
284     margin-top: 30px;
285     display: flex;
286     justify-content: space-between;
287     background-color: #f8f9fa;
288     font-size: 14px;
289   }
290   .net-income .amount {
291     color: ${plData.netIncome >= 0 ? '#27ae60' : '#c0392b'};
292   }
293   .total .amount,
294   .net-income .amount {
295     font-size: 14px;
296   }
297   .print-button {
298     display: block;
299     margin: 20px auto;
300     padding: 6px 16px;
301     background-color: #0854a0;
302     color: white;
303     border: 1px solid #0854a0;
304     border-radius: 0.25rem;
305     cursor: pointer;
306     font-size: 0.875rem;
307     font-family: "72","72full",Arial,Helvetica,sans-serif;
308     line-height: 1.42857;
309     text-align: center;
310     text-shadow: none;
311     box-shadow: 0 0 0.0625rem transparent;
312     transition: all 0.125s ease-in;
313     min-width: 80px;
314     height: 32px;
315     text-transform: uppercase;
316     font-weight: normal;
317   }
318   .print-button:hover {
319     background-color: #0a6ed1;
320     border-color: #0a6ed1;
321     box-shadow: 0 0 0.0625rem transparent;
322   }
323   .print-button:active {
324     background-color: #0854a0;
325     border-color: #0854a0;
326     color: #ffffff;
327   }
328   .print-button:focus {
329     outline: none;
330     box-shadow: 0 0 0 2px #ffffff, 0 0 0 4px #0854a0;
331   }
332   .company-info {
333     text-align: center;
334     margin-bottom: 20px;

```

```

335 |     color: #7f8c8d;
336 |
337 | }
338 |
339 | .letterhead {
340 |   display: flex;
341 |   justify-content: space-between;
342 |   align-items: flex-start;
343 |   margin-bottom: 30px;
344 |   padding-bottom: 20px;
345 | }
346 |
347 | .company-logo {
348 |   text-align: left;
349 |   font-size: 10px;
350 |   line-height: 1.5;
351 |   flex: 0 0 auto;
352 | }
353 |
354 | .company-name {
355 |   font-weight: bold;
356 |   font-size: 12px;
357 |   margin-bottom: 5px;
358 | }
359 |
360 | .company-logo {
361 |   width: 200px;
362 |   height: auto;
363 |   object-fit: contain;
364 | }
365 |
366 | .logo-section {
367 |   flex: 0 0 auto;
368 | }
369 |
370 | @media print {
371 |   body {
372 |     margin: 0;
373 |     background-color: white;
374 |   }
375 |
376 |   .container {
377 |     box-shadow: none;
378 |     padding: 0;
379 |   }
380 |
381 |   .no-print {
382 |     display: none;
383 |   }
384 |
385 |   .section-title {
386 |     background-color: white;
387 |     border-bottom-color: black;
388 |   }
389 |
390 |   .total, .net-income {
391 |     background-color: white;
392 |     border-color: black;
393 |   }
394 |
395 | }
396 |
397 | </style>
398 | </head>
399 |
400 | <body>
401 | <div class="container">
402 |   <div class="letterhead">
403 |     <div class="logo-section">
404 |       
405 |     </div>
406 |     <div class="company-details">
407 |       <div class="company-name">AK MAJU RESOURCES SDN. BHD.</div>
408 |       <div>No. 39 & 41 Jalan Utama 3/2, Pusat Komersial Sri Utama,</div>
409 |       <div>Segamat, Johor, Malaysia- 85000</div>
410 |       <div>07-9310717 , 010-2218224</div>
411 |       <div>akmaju.acc@gmail.com</div>
412 |       <div>Company No.: <span style="margin-left: 20px;">1088436 K</span></div>
413 |     </div>
414 |   </div>
415 |
416 |   <div class="header">
417 |     <div class="header-line">
418 |       <span class="statement-title">PROFIT & LOSS STATEMENT</span>
419 |       <span class="period">PERIOD: ${pldata.period.toUpperCase()}</span>
420 |     </div>
421 |   </div>
422 |
423 |   ${pldata.sections.map(section => `<div class="section"><div class="section-title">${section.title.toUpperCase()}</div>`)}
424 |
425 | </div>

```

```

415     ${section.items.map(item => ` 
416       <div class="item">
417         <span class="item-description">${item.description.toUpperCase()}</span>
418         <span class="amount">${formatMYR(item.amount)}</span>
419       </div>
420     `).join('')}
421   <div class="total">
422     <span class="total-label">TOTAL ${section.title.toUpperCase()}</span>
423     <span class="amount">${formatMYR(section.total)}</span>
424   </div>
425 `),
426 `).join(''))
427 <div class="net-income">
428   <span>NET PROFIT</span>
429   <span class="amount">${formatMYR(plData.netIncome)}</span>
430 </div>
431 <button class="print-button no-print" onclick="window.print()">PRINT</button>
432 </div>
433 </body>
434   `;
435 
436 
437 printWindow.document.write(html);
438 printWindow.document.close();
439 });
440 );
441 );

```

5.5 Summary

In summary, Chapter 5 showcases the system implementation process for AK Maju's financial management system, covering every aspect from the development environment setup to coding. The system, designed to cater to four financial modules AR, AP, GL, and Finance Report, serves as a robust framework for managing financial data. The AR subsystem focuses on invoices, ensuring the efficient handling of payment statuses and customer details, while the AP subsystem is designed to streamline vendor payments. The GL subsystem integrates these modules by maintaining accurate and balanced financial records, and the Finance Report subsystem transforms the data into actionable insights for decision-making.

The collaborative effort of the team ensured that each subsystem was developed with precision, utilizing the SAP ecosystem for smooth data processing and a user-friendly experience. The coding of main functions, including annotations for organizing fields and configuring user interfaces, highlights the depth of customization achieved during development. This chapter emphasizes the technical components of the project, ultimately presenting a system that aligns with AK Maju's operational needs.

CHAPTER 6

CONCLUSION

6.1 Introduction

This study has carefully examined some major benefits of integrated solutions in the contemporary context that play a very crucial role in mitigating the operational inefficiencies linked with legacy systems. In an effort to simplify processes, remove data silos, and make workflows easy, connectivity between order information systems and finance subsystems can be materialized using SAP HANA and SAP Business Technology Platform (BTP). The thesis has focused on how such technologies can revolutionize business operations for organizations like AK Maju by providing practical insights into the benefits, limitations, and potential improvements that could be made in exploiting such solutions.

6.2 System Contribution / Achievement

The integration of SAP HANA with SAP BTP has yielded excellent results for AK Maju's business processes. The business can ensure that order details are reflected in real time in its financial subsystem through the activation of real-time data processing. This advancement has streamlined payment reconciliation processes and reduced human error by allowing the automatic creation of invoices and receipts based on order data. In addition, the true real-time reporting capability brings a lot of financial visibility for immediate insight into key variables: cash flow, revenue, and profitability.

The improvement in decision-making is another notable accomplishment. The employees now have access to latest and accurate financial data, which enables AK Maju's management team to make well-informed decisions that support the company's expansion plans. Additionally, SAP BTP's scalability guarantees that the system can accommodate AK Maju's changing requirements, facilitating its entry into new product categories and markets.

6.3 System Constraint

The study has revealed a number of limitations related to the implementation and operation of the integrated system, notwithstanding the significant contributions. The substantial upfront costs associated with switching from a traditional system to SAP HANA and SAP BTP are one of the main obstacles. This covers costs for infrastructure improvements, software licenses, and the expert services needed for deployment. Furthermore, training for employees is necessary to guarantee they are prepared to utilise the new system efficiently, which might be resource- and time-intensive.

The second disadvantage is that operational outages may happen during the time of changeover, which might interfere with business continuity. Of course, dependence on cloud infrastructure does bring forth data security and regulatory compliance issues, at least for those industries dealing in sensitive financial information. These would then emphasize the need for prudent preparation and management of risks throughout the implementation process to minimize disruptions and ensure a seamless transition.

6.4 Future Suggestion

A few suggestions are proposed to address the identified constraints and further enhance the system. Firstly, AK Maju can investigate the SAP Business Technology Platform's sophisticated analytics and artificial intelligence (AI) features. These tools are able to improve decision-making processes by enabling automated fraud detection, real-time market trend research, and predictive financial modelling.

Secondly, by connecting customer data with order and financial information, integrating customer relationship management (CRM) technologies with the current system would offer a more thorough understanding of business operations. As a result, AK Maju would be able to provide individualised services while staying cost-effective.

Finally, AK Maju could put strong cybersecurity measures in place, like encryption, multi-factor authentication, and frequent audits, to solve data security issues. Working together with SAP's security specialists could assist guarantee that the system respects applicable laws and preserves the accuracy of financial data.

6.5 Summary

In conclusion, AK Maju's decision to integrate SAP HANA with SAP BTP is revolutionary. It assists the business in increasing productivity, automating procedures, and making wiser financial choices. Even while there are obstacles, such expenses and security issues, these may be handled with careful preparation and ongoing advancements. With a robust, scalable, and effective financial system that underpins its expansion and competitive advantage, AK Maju is well-positioned for long-term success after implementing these contemporary solutions.