



Enterprise System Project Case Study

Exita Manufacturing

Group 4 - Crafters

Programme	: Bachelor of Computer Science (Data Engineering)
Subject Code	: SECP3774
Subject Name	: ENTERPRISE SYSTEMS DESIGN AND MODELING (WBL)
Session-Sem	: 24/25-1

Prepared by : 1) THEVAN RAJU A/L JEGANATH
2) NURUL ERINA BINTI ZAINUDDIN

3) ALIATUL IZZAH BINTI JASMAN

Section : 03

Lecturer : Prof Madya Dr Roliana Binti Ibrahim

Date : 24 January 2025

Table of Contents

Abstract.....	7
Chapter 1: Introduction.....	8
1.1 Overview.....	8
1.2 Problem Statement.....	8
1.3 Project Objectives.....	8
1.4 Project Scope.....	9
1.5 Project Importance.....	9
Chapter 2: Literature Review.....	10
2.1 Introduction.....	10
2.2 Enterprise Architecture.....	10
2.2.1 Evolution of Enterprise Architecture.....	10
2.2.2 Benefits of Enterprise Architecture.....	11
2.2.3 Challenges in Implementing Enterprise Architecture.....	11
2.2.4 Role of SAP in Enterprise Architecture.....	12
2.3 Related Subsystems.....	12
2.3.1 Reporting and Analytics Subsystem.....	12
2.3.2 Sales and Customer Management (CRM) Subsystem.....	13
2.3.3 Procurement and Inventory Management (SCM) Subsystem.....	14
2.3.4 Financial Management Subsystem.....	15
2.4 Technology Used.....	16
2.5 Summary.....	17
2.6 References.....	17
Chapter 3: Methodology.....	19
3.1 Introduction.....	19
3.2 The Chosen Methodology.....	19
3.3 Phases of the Chosen Methodology.....	20
3.3.1 TOGAF Phases.....	20
3.3.2 Waterfall Phases.....	21
3.4 Project Planning Schedule.....	22
3.5 Summary.....	23
Chapter 4: Analysis and Design.....	24
4.1 Introduction.....	24

4.2 Company Organisation Structure.....	24
4.3 Comparison of Existing Systems and Proposed Systems.....	24
4.3.1 Reporting and Analytics Subsystem.....	24
4.3.2 Sales and Customer Management (CRM) Subsystem.....	25
4.3.3 Procurement and Inventory Management (SCM) Subsystem.....	25
4.3.4 Financial Management Subsystem.....	26
4.4 System Requirements Gathering Technique.....	26
4.4.1 Analysis of Provided Case Study.....	26
4.4.2 Online Research on Industry Practices.....	26
4.5 System Requirements.....	27
4.5.1 Functional Requirements.....	27
4.5.2 Non-functional Requirements.....	28
4.5.3 Summary.....	28
4.6 System Design.....	29
4.6.1 Enterprise Architecture.....	29
4.6.2 Components Of Enterprise Architecture.....	30
4.6.3 System Architecture.....	32
4.6.4 Components Of System Architecture.....	33
4.7 Project Design.....	35
4.7.1 Use Case Diagram - Enterprise System.....	35
4.7.2 Detailed Use Case Diagram.....	36
4.7.2.1 Reporting and Analytics Subsystem.....	36
4.7.2.2 Sales and Customer Management (CRM) Subsystem.....	37
4.7.2.3 Procurement and Inventory Management (SCM) Subsystem.....	38
4.7.2.4 Financial Management Subsystem.....	39
4.7.3 Use Case Description.....	40
4.7.3.1 Reporting and Analytics Subsystem.....	40
4.7.3.1.1 Use Case Description: View Real-Time Dashboards.....	40
4.7.3.1.2 Use Case Description: Generate Report.....	42
4.7.3.1.3 Use Case Description: Analyze Trends with Predictive Analytics.....	44
4.7.3.2 Sales and Customer Management (CRM) Subsystem.....	45
4.7.3.2.1 Use Case Description: Track Customer Orders.....	45
4.7.3.2.2 Use Case Description: Manage Customer Profiles.....	46
4.7.3.2.3 Use Case Description: Automate Sales Invoices.....	47

4.7.3.2.4 Use Case Description: Forecast Sales Trends.....	48
4.7.3.3 Procurement and Inventory Management (SCM) Subsystem.....	49
4.7.3.3.1 Use Case Description: Track Inventory Levels.....	49
4.7.3.3.2 Use Case Description: Manage Supplier Information.....	51
4.7.3.3.3 Use Case Description: Automate Stock Replenishment.....	52
4.7.3.3.4 Use Case Description: Optimize Warehouse Locations.....	54
4.7.3.4 Financial Management Subsystem.....	56
4.7.3.4.1 Use Case: Process Accounts Payable/Receivable (AP/AR).....	56
4.7.3.4.2 Use Case: Generate Financial Reports.....	58
4.7.3.4.3 Use Case: Track Budget Allocation.....	60
4.7.3.4.4 Use Case: Automate Tax Calculations.....	62
4.7.4 Activity Diagram.....	64
4.7.4.1 Reporting and Analytics Subsystem.....	64
4.7.4.1.1 View Real-Time Dashboards.....	64
4.7.4.1.2 Generate Reports.....	65
4.7.4.1.3 Analyze Trends with Predictive Analytics.....	66
4.7.4.2 Sales and Customer Management (CRM) Subsystem.....	67
4.7.4.2.1 Track Customer Orders.....	67
4.7.4.2.2 Manage Customer Profiles.....	68
4.7.4.2.3 Automate Sales Invoices.....	69
4.7.4.2.4 Forecast Sales Trends.....	70
4.7.4.3 Procurement and Inventory Management (SCM) Subsystem.....	71
4.7.4.3.1 Track Inventory Levels.....	71
4.7.4.3.2 Manage Supplier Information.....	72
4.7.4.3.3 Automate Stock Replenishment.....	73
4.7.4.3.4 Optimize Warehouse Locations.....	74
4.7.4.4 Financial Management Subsystem.....	75
4.7.4.4.1 Process Accounts Payable/Receivable (AP/AR).....	75
4.7.4.4.2 Generate Financial Reports.....	76
4.7.4.4.3 Track Budget Allocation.....	77
4.7.4.4.4 Automate Tax Calculation.....	78
4.7.5 Sequence Diagram.....	79
4.7.5.1 Reporting and Analytics Subsystem.....	79
4.7.5.1.1 View Real-Time Dashboards.....	79

4.7.5.1.2 Generate Reports.....	80
4.7.5.1.3 Analyze Trends with Predictive Analytics.....	81
4.7.5.2 Sales and Customer Management (CRM) Subsystem.....	82
4.7.5.2.1 Track Customer Orders.....	82
4.7.5.2.2 Manage Customer Profiles.....	83
4.7.5.2.3 Automate Sales Invoices.....	84
4.7.5.2.4 Forecast Sales Trends.....	85
4.7.5.3 Procurement and Inventory Management (SCM) Subsystem.....	86
4.7.5.3.1 Track Inventory Levels.....	86
4.7.5.3.2 Manage Supplier Information.....	87
4.7.5.3.3 Automate Stock Replenishment.....	88
4.7.5.3.4 Optimize Warehouse Locations.....	89
4.7.5.4 Financial Management Subsystem.....	90
4.7.5.4.1 Process Accounts Payable/Receivable (AP/AR).....	90
4.7.5.4.2 Generate Financial Reports.....	91
4.7.5.4.3 Track Budget Allocation.....	92
4.7.5.4.4 Automate Tax Calculations.....	93
4.7.6 Centralized Database Design.....	94
4.7.7 Selected Sub-System Interface Design.....	95
4.7.7.1 Track Customer Orders.....	95
4.7.7.2 Manage Customer Profiles.....	97
4.7.7.3 Automate Sales Invoices.....	97
4.7.7.4 Forecast Sales Trends.....	98
4.8 Summary.....	99
Chapter 5: System Implementation.....	100
5.1 Introduction.....	100
5.2 System Development.....	100
5.2.1 Data Modeling.....	101
5.2.2 Service Definition.....	102
5.2.3 User Interface Development.....	102
5.2.4 Logic Implementation.....	103
5.3 Create Database.....	104
5.3.1 Data Scheme.....	104
5.3.2 Service Layer.....	105

5.3.3 Data Files.....	107
5.4 Coding of the system's main functions.....	108
5.4.1 Customer logic.....	108
5.4.2 Calculate Total Amount Logic.....	110
5.4.3 Automatic Invoice Generation Logic.....	112
5.4.4 index.html.....	114
5.4.5 launchpadPage.html.....	116
5.5 Sales Dashboard.....	119
5.5.1 Create Dashboard Page.....	122
5.5.2 Database of Dashboard.....	124
5.5.3 Sales and Management Dashboard.....	125
5.6 System Testing with Input and Output Screen.....	127
5.6.1 Track Customers Order.....	128
5.6.2 Automate Sales Invoice.....	130
5.6.3 Manage Customer Profile.....	131
5.6.4 Forecast Sales Trend.....	138
5.6.4.1 Filtering Data By Date.....	139
5.6.4.2 Filtering Data By State.....	140
5.6.4.3 Filtering Data By Sales Method.....	141
5.7 Summary Chapter.....	142
Chapter 6: Conclusion.....	143
6.1 Introduction.....	143
6.2 System Contribution/Achievement.....	143
6.3 System Constraint.....	144
6.4 Future Suggestion.....	145
6.5 Summary.....	146

Exita Manufacturing

Abstract

Exita Manufacturing, a mid-sized company specializing in industrial equipment production, faces operational inefficiencies due to isolated systems across its departments. These challenges hinder real-time data access, disrupt inter-departmental collaboration, and complicate decision-making. This project aims to design and implement a centralized Enterprise Resource Planning (ERP) system using the SAP cloud platform to address these issues and support the company's growth.

The proposed ERP solution integrates four critical modules: Reporting and Analytics, Sales and Customer Management (CRM), Procurement and Inventory Management (SCM), and Financial Management (ERP). These modules leverage SAP's cloud capabilities to enhance data visibility, streamline workflows, and automate key processes such as order management, inventory tracking, and financial operations.

A dual-methodology approach is employed to ensure effective implementation. The TOGAF framework provides a structured foundation for enterprise architecture, while the System Development Life Cycle (SDLC) facilitates systematic design, testing, and deployment. Key deliverables include a modular ERP system hosted on SAP's scalable cloud platform, with robust security mechanisms and user-friendly interfaces.

The expected outcomes of this project include improved operational efficiency, enhanced decision-making capabilities, and the ability to scale operations to support future business needs. This initiative positions Exita Manufacturing as a forward-thinking organization, leveraging SAP technology for competitive advantage.

Chapter 1: Introduction

1.1 Overview

Exita Manufacturing is a mid-sized industrial equipment company facing operational challenges due to fragmented systems across its departments. The lack of integration between production, sales, procurement, inventory, and finance systems leads to inefficiencies, delays, and miscommunication. To address these issues and support its expansion goals, Exita Manufacturing plans to implement a centralized Enterprise Resource Planning (ERP) system. By adopting SAP's cloud-based platform, the aim is to streamline the company's workflows, enhance data accessibility, and improve interdepartmental collaboration.

1.2 Problem Statement

The fragmented nature of Exita Manufacturing's systems has led to several critical challenges:

1. **Operational Inefficiency:** Repetitive manual tasks and lack of automation slow down processes.
2. **Communication Gaps:** Disconnected systems cause difficulty in collaboration and data sharing between departments.
3. **Limited Decision-Making Support:** Inconsistent, outdated data limits the ability to make informed strategic decisions.
4. **Scalability Issues:** Current systems are inadequate and insufficient to support the company's future growth and expansion.

These issues not only delay the company's operational performance but also prevent it from leveraging opportunities in a competitive market.

1.3 Project Objectives

The primary objectives of this project are:

1. To integrate business functions across departments into a centralized ERP system.
2. To automate repetitive processes and reduce manual effort.
3. To provide real-time access to accurate and consistent data.
4. To enhance decision-making through unified and centralized reporting and analytics.
5. To implement a scalable and secure ERP system to support future growth.

1.4 Project Scope

The scope of this project includes the implementation of four key ERP modules using SAP's cloud-based platform:

1. **Reporting and Analytics:** Provides unified reporting, real-time Key Performance Indicator (KPI) dashboards, and predictive analytics for strategic planning.
2. **Sales and Customer Management (CRM):** Facilitates order tracking, customer interaction management, and sales forecasting to improve customer relationships and revenue generation.
3. **Procurement and Inventory Management (SCM):** Enables supplier management, inventory tracking, and warehouse optimization to streamline the supply chain.
4. **Financial Management (ERP):** Enhances financial management by automating processes like accounts payable, accounts receivable, budgeting, and cost accounting for improved financial oversight.

The project will adopt a phased implementation approach to minimize operational disruptions while ensuring a smooth transition to the new system.

1.5 Project Importance

Implementing the ERP system is essential for Exita Manufacturing to achieve the following:

1. **Improved Efficiency:** Automation of processes will reduce manual effort and operational delays.
2. **Enhanced Collaboration:** A centralized system will bridge communication gaps across the different departments.
3. **Better Decision-Making:** Unified data and analytics will enable timely and informed decisions.
4. **Scalability:** The SAP cloud platform offers flexibility and scalability to support the company's future growth.
5. **Competitive Edge:** Leveraging modern ERP technology will position the company as an innovative and agile industry leader.

Chapter 2: Literature Review

2.1 Introduction

Enterprise systems (ES) are integral to modern business operations, enabling the integration of processes across departments. With advancements in technology, tools like SAP's ecosystem have revolutionized how businesses manage workflows, develop applications, and enhance employee skills. For Exita Manufacturing, implementing an ERP system is essential to address inefficiencies caused by siloed systems. This chapter reviews the evolution of enterprise architecture, subsystems relevant to the four selected modules, and SAP technologies applicable to this project.

2.2 Enterprise Architecture

Enterprise architecture (EA) is a framework that defines the structure and operation of an organization's IT systems to align them with business goals. It integrates processes, data, and technology across departments to ensure efficiency, scalability, and strategic alignment [13]. EA provides a holistic view of an organization's workflow, enabling better decision-making, resource optimization, and adaptability to changing market demands.

Frameworks like TOGAF guide the development of EA, offering structured methodologies to manage complexity, integrate systems, and ensure continuous improvement. Modern EA emphasizes flexibility and innovation, leveraging technologies such as cloud computing, artificial intelligence, and IoT to enhance operational capabilities [7].

2.2.1 Evolution of Enterprise Architecture

Enterprise architecture (EA) has evolved significantly over the past few decades. Early systems, such as Material Requirements Planning (MRP), were application-specific, addressing isolated functions like procurement and production. In the 1990s, the introduction of Enterprise Resource Planning (ERP) systems revolutionized EA by integrating core business functions such as finance, procurement, logistics, and inventory management into unified platforms. This shift aimed to address data silos, enabling businesses to centralize operations and improve efficiency [8][15].

Modern EA frameworks, including TOGAF, provide structured approaches for aligning IT systems with organizational goals. These frameworks help businesses design scalable, modular architectures that adapt to changing operational demands. Emerging technologies such as artificial intelligence (AI), cloud computing, and the Internet of Things (IoT) further enhance EA, enabling real-time decision-making and process automation [12][15].

2.2.2 Benefits of Enterprise Architecture

Enterprise architecture is pivotal in ensuring that IT systems support business objectives. Key benefits include:

- **Centralization of Data:** By integrating disparate systems, EA ensures that all departments access consistent, real-time information, which improves the company's decision-making [13].
- **Scalability:** Modular architectures, such as those enabled by SAP S/4HANA, allow businesses to expand operations seamlessly [15].
- **Collaboration:** Unified platforms foster better communication across departments by reducing redundancies and delays [5].
- **Operational Efficiency:** Automation of repetitive tasks through advanced EA designs frees up resources for strategic activities [12].

2.2.3 Challenges in Implementing Enterprise Architecture

Despite its benefits, the implementation of EA can be challenging:

- **Integration with Legacy Systems:** Many organizations face difficulties in aligning modern ERP platforms with outdated infrastructure. Middleware and adapters are often required to ensure compatibility [7].
- **Complexity:** The scope of EA implementation often spans multiple departments with distinct requirements, necessitating thorough planning and customization [5].
- **Implementation Cost:** High initial investments in technology, staff training, and change management can be barriers, particularly for smaller organizations [13].

2.2.4 Role of SAP in Enterprise Architecture

SAP has been at the forefront of enterprise architecture innovation. Its flagship ERP solution, SAP S/4HANA, integrates financial, operational, and logistical functions into a single platform. By leveraging advanced technologies such as AI, IoT, and predictive analytics, SAP enables organizations to optimize workflows, enhance decision-making, and achieve digital transformation [12][15]. Additionally, SAP's modular approach ensures scalability, allowing businesses to implement components like CRM, SCM, and analytics based on their priorities [8][15].

2.3 Related Subsystems

This section examines the functionalities and significance of the selected subsystems in addressing the operational challenges faced by Exita Manufacturing. By comparing existing solutions from prior studies with the proposed modules, this analysis highlights enhancements tailored to optimize performance, streamline processes, and support strategic decision-making within the organization.

2.3.1 Reporting and Analytics Subsystem

The Reporting and Analytics Subsystem is crucial for addressing operational challenges in manufacturing environments like Exita Manufacturing. This subsystem focuses on enhancing decision-making through advanced data analysis and visualization.

Functionality	Existing Systems	Our Subsystem
Dashboard	Provide centralized dashboards for real-time tracking of KPIs such as production efficiency and target sales.	Enhanced dashboards with industry-specific KPIs and real-time updates tailored to management needs.
Predictive Analytics	Basic demand forecasting and operational planning capabilities.	Advanced predictive analytics for more accurate demand of sales forecasting and operational planning.
Report Customization	Limited customization, offering generic reports.	Fully customizable reports for strategic decision-making and filtering systems.
Data Integration	Many systems face challenges in integrating data across departments, leading to inconsistencies.	Ensures seamless integration across departments, providing a unified and consistent data view.

Table 2.3.1 : Table of Related Subsystems of Reporting and Analytics Subsystem

Key Functions and Benefits:

- The subsystem provides centralized dashboards for real-time tracking of key performance indicators (KPIs), such as production efficiency, inventory levels, and sales performance. This capability allows management to monitor operational metrics continuously, facilitating timely decision-making [6].
- Enables predictive analytics for demand forecasting and operational planning.
- The subsystem offers customizable reports that support strategic decision-making tailored to management needs. This flexibility allows users to filter and analyze data based on specific criteria, improving the relevance and utility of reports for different managerial levels [3].

2.3.2 Sales and Customer Management (CRM) Subsystem

Functionality	Existing Systems	Our Subsystem
Process Automation	Automate order tracking, invoicing, and basic customer service.	Automates processes while integrating real-time updates for greater accuracy.
Personalization	Limited personalization for customer interactions.	Enhanced personalization tools to improve customer satisfaction and engagement.
Sales Forecasting	Relies on basic historical data for sales forecasting.	Combines historical data and real-time updates for more accurate revenue planning.
Customer Data Management	Struggles with maintaining up-to-date customer information, reducing CRM efficiency.	Maintains real-time updates of customer data to ensure more effective customer interactions.

Table 2.3.2 : Table of Related Subsystems of Sales and Customer Management (CRM) Subsystem

Key Functions and Benefits:

- The subsystem automates processes such as order tracking, invoicing, and customer service, reducing manual workloads. This streamlining enhances operational efficiency by minimizing time-consuming tasks [4].
- By utilizing advanced personalization tools, the subsystem improves customer satisfaction through tailored interactions and swift issue resolution. Personalized communication fosters stronger relationships with customers, as it addresses their specific needs and preferences [10].

- Improves sales forecasting using historical data and real-time updates, aiding in revenue planning.

2.3.3 Procurement and Inventory Management (SCM) Subsystem

Functionality	Existing Systems	Our Subsystem
Inventory Management	Just-in-time inventory practices to reduce holding costs.	Implements real-time inventory tracking across multiple warehouses to reduce redundancies.
Supplier Performance	Tracks supplier performance to ensure timely procurement.	Enhances supplier monitoring with analytics to prevent disruptions.
Visibility	Provides basic visibility into inventory flows, often limited to single warehouses.	Offers comprehensive tracking across warehouses, improving inventory management efficiency.
Demand Planning	Basic demand planning, often unable to adapt to real-time market fluctuations.	Advanced demand planning that incorporates real-time market data and analytics for accuracy.

Table 2.3.3 : Table of Related Subsystems of Procurement and Inventory Management (SCM) Subsystem

Key Functions and Benefits:

- The subsystem tracks supplier performance to ensure timely procurement and avoid disruptions in production. Enhanced analytics capabilities allow for proactive monitoring of supplier reliability and quality [4].
- Implements just-in-time inventory practices, optimizing stock levels and reducing holding costs.
- Enhances visibility into inventory movements across warehouses, minimizing redundancies and improving efficiency.

2.3.4 Financial Management Subsystem

Functionality	Existing Systems	Our Subsystem
Workflow Automation	Automates accounts payable/receivable and budgeting workflows to reduce manual errors.	Automates workflows while improving consistency and reducing errors further.
Reporting	Generates static financial reports.	Provides dynamic insights for proactive financial planning and resource allocation.
Budget Analysis	Tracks costs and analyzes budgets for basic decision-making.	Integrates real-time cost tracking and advanced budget analysis tools.
Compliance Management	Limited tools to ensure alignment with evolving financial regulations.	Includes regulatory compliance tools that automatically update and monitor current regulations.

Table 2.3.4 : Table of Related Subsystems of Financial Management Subsystem

Key Functions and Benefits:

- Centralizes accounts payable/receivable, budgeting, and financial reporting for accuracy and consistency.
- By automating financial workflows, the subsystem significantly reduces manual errors, which enhances operational efficiency. This automation allows finance teams to focus on strategic tasks rather than routine administrative duties [4].
- Supports cost tracking and budget analysis, enabling informed financial planning and resource allocation.

2.4 Technology Used

SAP Ecosystem for ERP Implementation

SAP provides a comprehensive suite of tools for ERP implementations, ensuring scalability, efficiency, and security. Key technologies relevant to Exita Manufacturing include:

1. **SAP S/4HANA:**
 - A next-generation ERP platform that integrates processes across all selected modules, ensuring real-time data access and operational efficiency.
 - Offers scalability for future business growth while maintaining system reliability [11].
2. **SAP Analytics Cloud (SAC):**
 - Enables advanced reporting and analytics, including predictive insights and visualization of KPIs [9].
 - Integrates seamlessly with SAP S/4HANA for a unified data environment.
3. **SAP Integrated Business Planning (IBP):**
 - Optimizes supply chain processes through demand forecasting and inventory planning.
 - Ensures efficient procurement and inventory management [1].
4. **SAP Build (formerly SAP AppGyver):**
 - A no-code platform for creating custom applications to enhance system functionalities.
 - Allows quick adaptation to specific business needs without extensive development efforts [2].
5. **SAP Fiori:**
 - Provides an intuitive user interface for employees to access ERP functionalities across devices [14].
 - Enhances productivity with role-based, simplified dashboards and workflows.

These technologies address Exita Manufacturing's specific needs by streamlining operations, enhancing decision-making, and providing a scalable foundation for future growth.

2.5 Summary

This chapter highlights the evolution of enterprise architecture, the significance of integrating subsystems into an ERP framework, and the role of SAP technologies in achieving operational excellence. By leveraging tools like SAP S/4HANA, SAC, and SAP Build, Exita Manufacturing can overcome inefficiencies, ensure real-time collaboration, and prepare for long-term scalability.

2.6 References

- [1] Acroplans. (2024). SAP Integrated Business Planning (IBP) for supply chain. LinkedIn. <https://www.linkedin.com/pulse/sap-integrated-business-planning-ibp-supply-chain-acroplans-0dep/>
- [2] AdrDen. (2022). First look into SAP Build & SAP Build Apps – QA. SAP Community. <https://community.sap.com/t5/technology-blogs-by-members/first-look-into-sap-build-sap-build-apps-qa/ba-p/13552093>
- [3] Davenport, T. H., & Harris, J. G. (2007). Competing on analytics: The new science of winning. Harvard Business Review Press.
- [4] FindMyCRM. (2024). Findmycrm.com. <https://doi.org/1029311/1717498416824>
- [5] Ferreira, D. R. (2013). Enterprise systems integration: A process-oriented approach. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-40796-3>
- [6] Kracklauer, A. H. (2004). Collaborative customer relationship management: Taking CRM to the next level. Springer Science & Business Media.
- [7] Li, Q., & Chen, Y. (2009). Modeling and analysis of enterprise and information systems: From requirements to realization. Springer-Verlag, Berlin Heidelberg.
- [8] Markus, M. L., & Tanis, C. (2000). The enterprise system experience—From adoption to success. In R. W. Zmud & M. F. Price (Eds.), *Framing the domains of IT management: Projecting the future through the past* (pp. 173-189). Pinnaflex Educational Resources.
- [9] MHundekari. (2024). Evaluating SAP Analytics Cloud's (SAC) classification with Smart Predict. SAP Community. <https://community.sap.com/t5/technology-blogs-by-sap/evaluating-sap-analytics-cloud-s-sac-classification-with-smart-predict/ba-p/13791397>
- [10] Salesforce. (2023). What is CRM. Salesforce. <https://www.salesforce.com/ap/crm/what-is-crm/>
- [11] Venkatesh Golla. (2020). S/4HANA Article Master - Listing article-Part1. SAP Community. <https://community.sap.com/t5/enterprise-resource-planning-blogs-by-members/s-4hana-article-master-listing-article-part1/ba-p/13464845>

- [12] Vom Brocke, J., Maaß, W., Buxmann, P., Maedche, A., Leimeister, J. M., & Pecht, G. (2018). Future work and enterprise systems. *Business & Information Systems Engineering*, 60(4), 357–366.
<https://doi.org/10.1007/s12599-018-0544-2>
- [13] Wang, W. Y. C., Pauleen, D., & Taskin, N. (2022). Enterprise systems, emerging technologies, and the data-driven knowledge organization. *Knowledge Management Research & Practice*, 20(1), 1–13.
<https://doi.org/10.1080/14778238.2022.2039571>
- [14] Wilding, S. (2020, March 6). SAP Fiori for web — Updated design guidelines for SAP Fiori 3. Medium.
<https://medium.com/@susanne.wilding/sap-fiori-for-web-updated-design-guidelines-for-sap-fiori-3-a1f731665a3>
- [15] Xu, L. D. (2011). Enterprise systems: State-of-the-art and future trends. *IEEE Transactions on Industrial Informatics*, 7(4), 630–639.

Chapter 3: Methodology

3.1 Introduction

Given the limited project duration of one month, a streamlined and structured approach is crucial for the successful implementation of an ERP system at Exita Manufacturing. This chapter outlines the chosen methodologies, TOGAF and the Waterfall model, and details their phases to ensure effective integration and deployment within the given timeframe.

3.2 The Chosen Methodology

1. TOGAF (The Open Group Architecture Framework):

- TOGAF was chosen because it provides a structured framework for aligning IT systems with the strategic goals of Exita Manufacturing.
- Its focus on scalability ensures the ERP system can adapt to future business needs.
- Emphasizing integration and stakeholder alignment, TOGAF ensures that the CRM, SCM, Financial Management, and Reporting modules work cohesively to meet business objectives.

2. Waterfall Model (System Development):

- The Waterfall model was selected due to its clear and sequential development process, which is critical given the project's tight one-month timeline.
- Ensuring each phase is completed before progressing minimizes the risk of rework and provides well-defined deliverables, making it ideal for this project.

Integration of TOGAF and Waterfall:

- TOGAF and the Waterfall model work in tandem to ensure both strategic and operational success.
- TOGAF provides the high-level architectural vision and roadmap for the ERP system, aligning it with long-term business goals.
- The Waterfall model delivers structured development, ensuring each module is designed, implemented, tested, and deployed systematically within the limited timeframe.

3.3 Phases of the Chosen Methodology

3.3.1 TOGAF Phases

1. Architecture Vision :

- Define the project's objectives, scope, and stakeholders.
- Identify high-level requirements for the four selected modules (CRM, SCM, Financial Management, Reporting).
- This phase sets the foundation for understanding the overall project direction and the business needs the ERP system must address.

2. Business Architecture :

- Map current business processes and align them with the proposed ERP system.
- Document workflows for integration.
- This phase will include business process modeling, ensuring a clear roadmap for how the system will enhance business operations.

3. Information Systems Architecture :

- Design data flow and communication mechanisms between the selected modules.
- Focus will be on establishing how information will move within the system and between different modules, ensuring smooth data exchange and process efficiency.

4. Technology Architecture :

- Finalize the selection of SAP technologies, such as SAP S/4HANA and SAP Fiori.
- Ensure the technological infrastructure aligns with business goals, focusing on scalability and integration.

5. Opportunities and Solutions :

- Identify potential risks, legacy system challenges, and strategies for mitigation.
- Define potential roadblocks and create contingency plans to overcome them during the development and implementation phases.

3.3.2 Waterfall Phases

1. Requirement Gathering and Analysis:

- Conduct interviews with stakeholders to gather module-specific requirements.
- Document functional and technical requirements.
- This phase involves gathering detailed information on the ERP system's needs, establishing clear requirements for each module.

2. System Design :

- Create architectural diagrams, workflows, and detailed specifications for system integration and module development.
- The system design phase provides a visual and structural blueprint for how the ERP system will be built and integrated, ensuring alignment with business goals.

3. Implementation :

- Develop the ERP system modules using SAP technologies.
- Implement Reporting and Analytics first, followed by CRM, SCM, and Financial Management.
- This phase involves building the actual system, coding, configuring the modules, and setting up the infrastructure.

4. Verification (Testing) :

- Conduct unit and integration testing for each module.
- Perform user acceptance testing (UAT) to ensure the system meets requirements.
- Verification ensures the system works as expected, meets all functional and technical requirements, and is ready for deployment.

5. Deployment :

- Deploy the ERP system in a single-phase rollout.
- Provide initial training and support for users.
- This phase focuses on making the system operational and providing users with the tools and knowledge they need to use it effectively.

6. Maintenance :

- Address any immediate post-deployment issues.
- Prepare the system for long-term monitoring and updates.
- After deployment, maintenance ensures the system operates smoothly, with any necessary tweaks or updates applied.

3.4 Project Planning Schedule

The condensed timeline is structured as follows:

Phase	Duration	Activities
Phase 1: Planning	Day 1-4	Stakeholder analysis, requirement gathering, and initial architecture.
Phase 2: Design	Day 5-7	Business process mapping, data flow design, and module specifications.
Phase 3: Development	Day 8-18	Module implementation using SAP technologies.
Phase 4: Testing	Day 19-24	System testing and user acceptance testing (UAT).
Phase 5: Deployment	Day 25-28	System rollout and user onboarding.
Phase 6: Maintenance	Day 29-30	Post-deployment support and system readiness check.

Table 3.4 : Table of project planning schedule

Phase	Duration	Activities
Architecture Vision	Day 1-2	Define objectives, scope, and stakeholders. Identify high-level requirements for four modules.
Business Architecture	Day 3-4	Map current business processes. Align workflows with the ERP system.
Information Systems Architecture	Day 5	Design data flow and communication mechanisms between modules.
Technology Architecture	Day 6	Finalize SAP technologies (e.g., SAP S/4HANA, SAP Fiori).
Opportunities & Solutions	Day 7	Identify risks, legacy system challenges, and mitigation strategies.

Table 3.5: TOGAF Project Planning Schedule

3.5 Summary

The methodologies ensure the project's success within the one-month timeline. TOGAF provides a strategic approach for aligning IT with business goals, while the Waterfall model ensures clear deliverables and systematic progress through the development stages. The phased plan minimises risks and ensures a functional, integrated ERP system tailored to Exita Manufacturing's needs.

Chapter 4: Analysis and Design

4.1 Introduction

This chapter will delve into the analysis and design of the ERP system project. The primary focus of this chapter is to certify the organisational structure, compare it with the existing system and proposed system, determine the system requirements technique, determine functional and non-functional system requirements, and determine the system design and project design. This architecture will also encompass the centralized database to ensure that each sub-system works properly with each other.

4.2 Company Organisation Structure

Exita Manufacturing operates by a structured organization to facilitate well-organized workflow and communication within various different departments. With the number of over 500 employees, the organization's structure is separated into its core function areas, such as Production, Sales, Finance, Procurement, Inventory, and Human Resources. Each of these core function areas is in charge of specific operational processes, which ensures clarity and accountability within the organisation.

4.3 Comparison of Existing Systems and Proposed Systems

In this section, we will compare the existing systems currently used by the industry in this era with the proposed ERP system for this project. The comparison will highlight each system's strengths and weaknesses.

4.3.1 Reporting and Analytics Subsystem

1. Existing Systems:

Most companies rely on different reporting tools and manual processes to generate insights, which often leads to inconsistencies and delays in integrating with their existing method. Traditional methods such as spreadsheets and standalone reporting software and applications require heavy manual input by employees making real-time data access challenging.

2. Proposed ERP System:

The proposed ERP system will be featured with a centralised Reporting and Analytics module that provides real-time integrated reporting. By integrating data across all the different departments of Exita Manufacturing, employees have the ability to access the accurate KPI seamlessly. This will enhance the strategic planning of the organization and faster decision-making.

3. Comparison:

- **Strengths of Proposed System:** Accessibility of real-time data, integrated reporting, and predictive business analytics.
- **Weaknesses of Existing Systems:** Data silos, manual processes, delayed reporting.

4.3.2 Sales and Customer Management (CRM) Subsystem

1. Existing Systems:

The existing CRM solutions often operate in isolation which often leads to incoherent customer interactions in the system and inaccurate or incomplete sales data. Companies often have to utilize multiple different tools for order tracking, customer service, and forecasting their business sales, which causes difficulty in the customer management process.

2. Proposed ERP System:

The proposed CRM subsystem in this project is to integrate the data of the sales and order management which will help in automating workflows especially features like order processing and customer communications during the purchasing process. It will also provide an integrated view of customer interactions that enables personalised service to the customer and additionally increases customer satisfaction.

3. Comparison:

- **Strengths of Proposed System:** Integrated customer interactions, automated workflows, improving business sales forecasting.
- **Weaknesses of Existing Systems:** Fragmentation, manual workflow, limited visibility into customer data.

4.3.3 Procurement and Inventory Management (SCM) Subsystem

1. Existing Systems:

Most organizations utilize various systems for procurement and inventory management which often leads to inefficiency of supply chain operations in the organization. These existing systems' inventory levels are often not visible in real-time access which results in stockouts or overstock problems.

2. Proposed ERP System:

The proposed SCM subsystem of this project will provide real-time visibility and access to inventory and procurement processes. It will be efficient in real-time inventory management and improve supplier relationship management by streamlining procurement operations with real-time organization data.

3. Comparison:

- **Strengths of Proposed System:** Real-time inventory tracking and access, integrated procurement processes and more efficient supplier management.
- **Weaknesses of Existing Systems:** Lack of visibility, and inefficient inventory management.

4.3.4 Financial Management Subsystem

1. Existing Systems:

Existing financial management systems which are often utilised by industries operate in silos, with multiple different applications for features like accounts payable, accounts receivable, and business budgeting. This scattering can lead to inconsistent display of financial data and requires manual effort in managing accounts.

2. Proposed ERP System:

The proposed ERP system will centralise all financial management functions with real-time organisation data and automated features for the invoice, payments, and budgeting processes. This proposed integration will ensure data consistency and provide extensive financial reporting capabilities for the organisation, which improves overall financial oversight.

3. Comparison:

- **Strengths of Proposed System:** Centralized financial management, automated workflows, enhanced financial reporting method.
- **Weaknesses of Existing Systems:** Data inconsistencies, requires manual management, limited reporting capabilities.

4.4 System Requirements Gathering Technique

To ensure the successful implementation of the ERP system for Exita Manufacturing, we employed the following techniques to gather system requirements:

4.4.1 Analysis of Provided Case Study

The case study provided by the instructor served as the primary source of information about the company's current systems, challenges, and operational needs. By carefully analyzing the case study:

- We identified inefficiencies such as data silos, lack of real-time reporting, and manual workflows.
- Detailed descriptions of Exita's departmental operations helped define the scope and objectives for each ERP module.

4.4.2 Online Research on Industry Practices

To supplement the insights from the case study, online research was conducted to understand best practices for ERP implementations in the manufacturing sector. Key findings included:

- The importance of integrating modules like CRM, SCM, and Financial Management for streamlined operations.
- Common features in ERP systems, such as automated reporting, inventory tracking, and predictive analytics.

- Insights into how SAP technologies (e.g., SAP S/4HANA, Analytics Cloud) are used to address similar challenges in other companies.

Consolidation of Requirements

The information from the case study and online research was synthesised to create a comprehensive set of system requirements. This approach ensured that:

- The solution addresses the specific challenges faced by Exita Manufacturing.
- Industry best practices are incorporated into the system design.

This dual approach of referring to the case study and conducting online research provided a strong foundation for defining both functional and non-functional requirements for the ERP system.

4.5 System Requirements

4.5.1 Functional Requirements

The functional requirements focus on the capabilities the ERP system must deliver to address Exita Manufacturing's operational needs.

1. Reporting and Analytics Subsystem

- Provide real-time dashboards for monitoring KPIs, such as production efficiency and sales performance.
- Enable predictive analytics to forecast demand and inventory needs.
- Generate custom financial, sales, and inventory reports.

2. Sales and Customer Management (CRM) Subsystem

- Automate customer order tracking and invoicing processes.
- Manage customer interactions and maintain a centralized customer database.
- Facilitate sales forecasting using historical and real-time data.

3. Procurement and Inventory Management (SCM) Subsystem

- Track supplier performance and maintain a supplier database.
- Monitor inventory levels in real-time and trigger automated stock replenishment.
- Optimize warehouse operations by tracking stock transfers and locations.

4. Financial Management Subsystem

- Automate accounts payable (AP) and accounts receivable (AR) workflows.
- Provide tools for budgeting, cost tracking, and financial forecasting.
- Ensure compliance with financial reporting standards and generate regulatory reports.

4.5.2 Non-functional Requirements

The non-functional requirements define the system's performance, reliability, and usability criteria.

1. Scalability

- The system should support future business growth by accommodating increased data volume and new modules.

2. Performance

- Ensure that reports are generated within 5 seconds for real-time decision-making.
- Support at least 100 concurrent users without performance degradation.

3. Security

- Implement role-based access controls to restrict sensitive data access.
- Use TLS/SSL encryption for secure data transmission.

4. Availability

- Guarantee 99.9% uptime, with minimal downtime for maintenance.
- Provide backup and disaster recovery mechanisms to prevent data loss.

5. User Experience

- Deliver an intuitive and responsive interface accessible on desktops and mobile devices.
- Provide role-based dashboards tailored to user responsibilities (e.g., managers, sales staff).

6. Integration

- Seamlessly integrate with existing third-party tools or legacy systems, if any, for data migration.
- Use APIs to enable real-time communication between modules.

4.5.3 Summary

These requirements ensure that the ERP system addresses Exita Manufacturing's operational challenges while meeting industry standards for performance, security, and usability. The functional requirements target core business processes, while the non-functional requirements focus on system robustness and scalability.

4.6 System Design

4.6.1 Enterprise Architecture

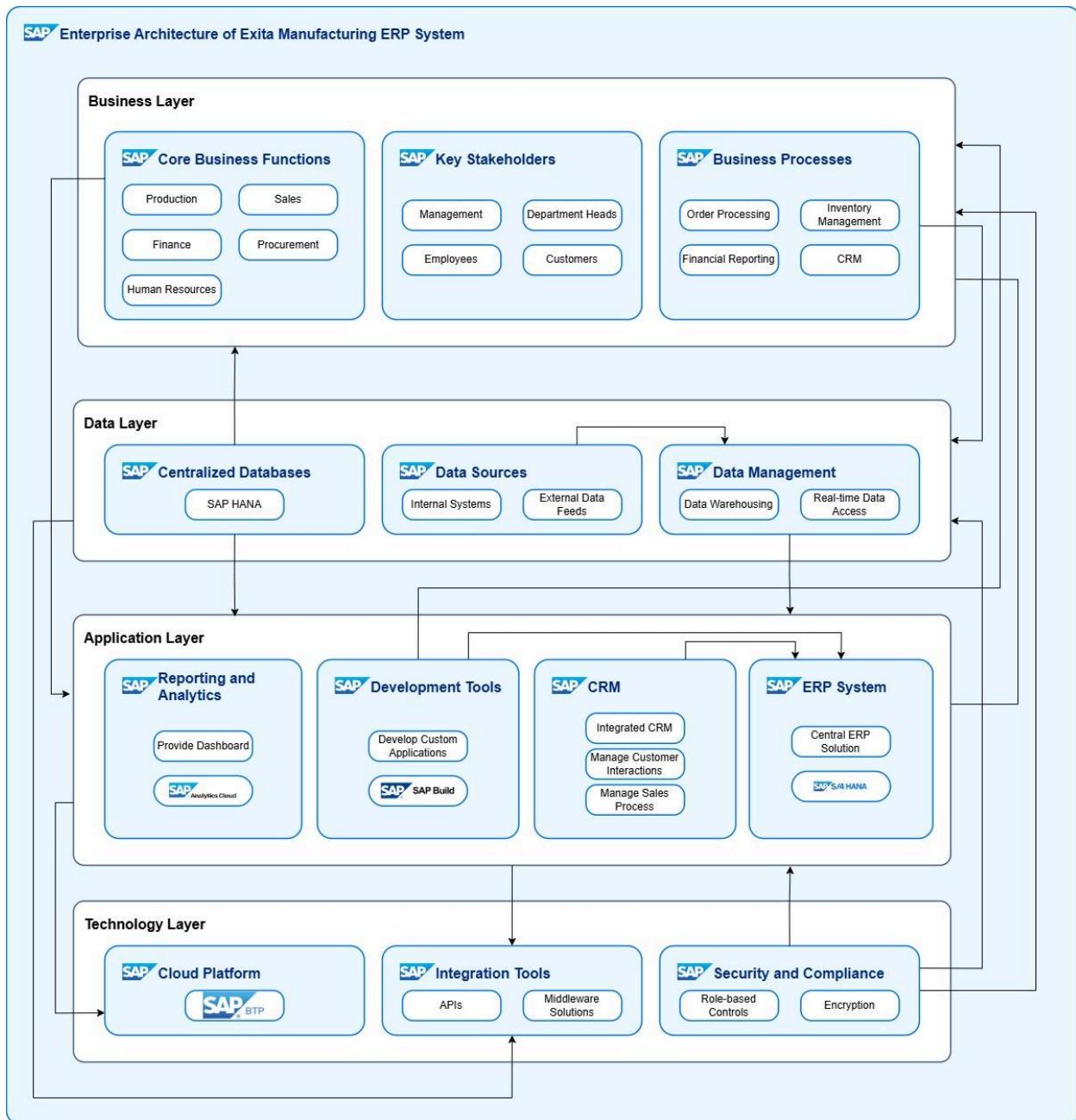


Figure 4.6.1 : Enterprise Architecture of Exita Manufacturing ERP System

4.6.2 Components Of Enterprise Architecture

Vision

The vision of enterprise architecture of Exita Manufacturing's ERP System is to create a highly integrated and agile organization that access with real-time data and featured with advanced analytics to enhance decision-making and improve customer satisfaction of Exita Manufacturing.

Mission

The mission is to deploy an extensive enterprise architecture that links technical aspects to business aims while boosting operations and cultural innovation throughout every operational division. Through this implementation Exita Manufacturing achieves both market-ready responsiveness and data-based operational efficiency.

Layers

1. Business Layer

- **Core Business Functions:** Defines the primary operational areas, such as Production, Sales, Finance, Procurement, and Human Resources.
- **Key Stakeholders:** Identifies the individuals and groups involved in the decision-making processes, including management, department heads, and employees.
- **Business Processes:** Defines the key processes that drive everyday operations, including Order Processing, Inventory Management, Financial Reporting, and Customer Relationship Management.

2. Data Layer

- **Centralized Database:** Utilizes SAP HANA for real-time data processing and analytics.
- **Data Sources:** Utilizes both internal and external data sources to ensure comprehensive data availability.
- **Data Management:** Focuses on data warehousing and real-time data access to support reporting and analytics needs.

3. Application Layer

- **ERP System:** Centralizes operations through SAP S/4HANA to integrate all functional areas.
- **Reporting and Analytics:** Employs SAP Analytics Cloud for real-time insights, sales forecasting and KPI tracking.
- **Development Tools:** Utilizes SAP Build for creating custom applications tailored to business requirements.
- **Customer Relationship Management:** Integrates CRM functionalities within the ERP system to enhance customer interactions with Exita Manufacturing.

4. Technology Layer

- **Cloud Platform:** Leverages SAP Business Technology Platform (BTP) for multiple SAP development applications and integration.

- **Integration Tools:** Implements APIs and middleware solutions to facilitate seamless data exchange across systems.
- **Security and Compliance:** Ensures robust security measures, including role-based access controls and data encryption, to protect sensitive information.

Values

- **Innovation:** Ability to adapt to new technologies and processes that enhance efficiency and effectiveness.
- **Collaboration:** Fostering teamwork across departments to achieve centralized goals and improve communication.
- **Agility:** Ability to adapt quickly to changing market conditions and customer needs through flexible processes and systems.
- **Customer Focus:** Prioritizing customer satisfaction by providing efficient personalized services.

Strategic Goals

- **Enhance Operational Efficiency:** The organization needs to optimize procedures throughout its entire departmental structure for both savings and increased operational effectiveness.
- **Improve Data Accessibility:** Real-time data availability must extend to stakeholders who need it for their important decisions.
- **Support Growth:** The system should provide expandability and operational adaptability in order to support future business development and market growth.
- **Facilitate Data-Driven Decision-Making:** By employing advanced analytics organizations can achieve data-driven strategic decisions which guide business growth.

Summary

The components of Exita Manufacturing's enterprise architecture create a holistic framework that aligns the organization's technology and requirements with its strategic objectives. This architecture aims to position Exita Manufacturing for long-term success in a competitive marketplace by focusing on integration, innovation, and efficiency, this architecture aims to position Exita Manufacturing for long-term success in a competitive marketplace.

4.6.3 System Architecture

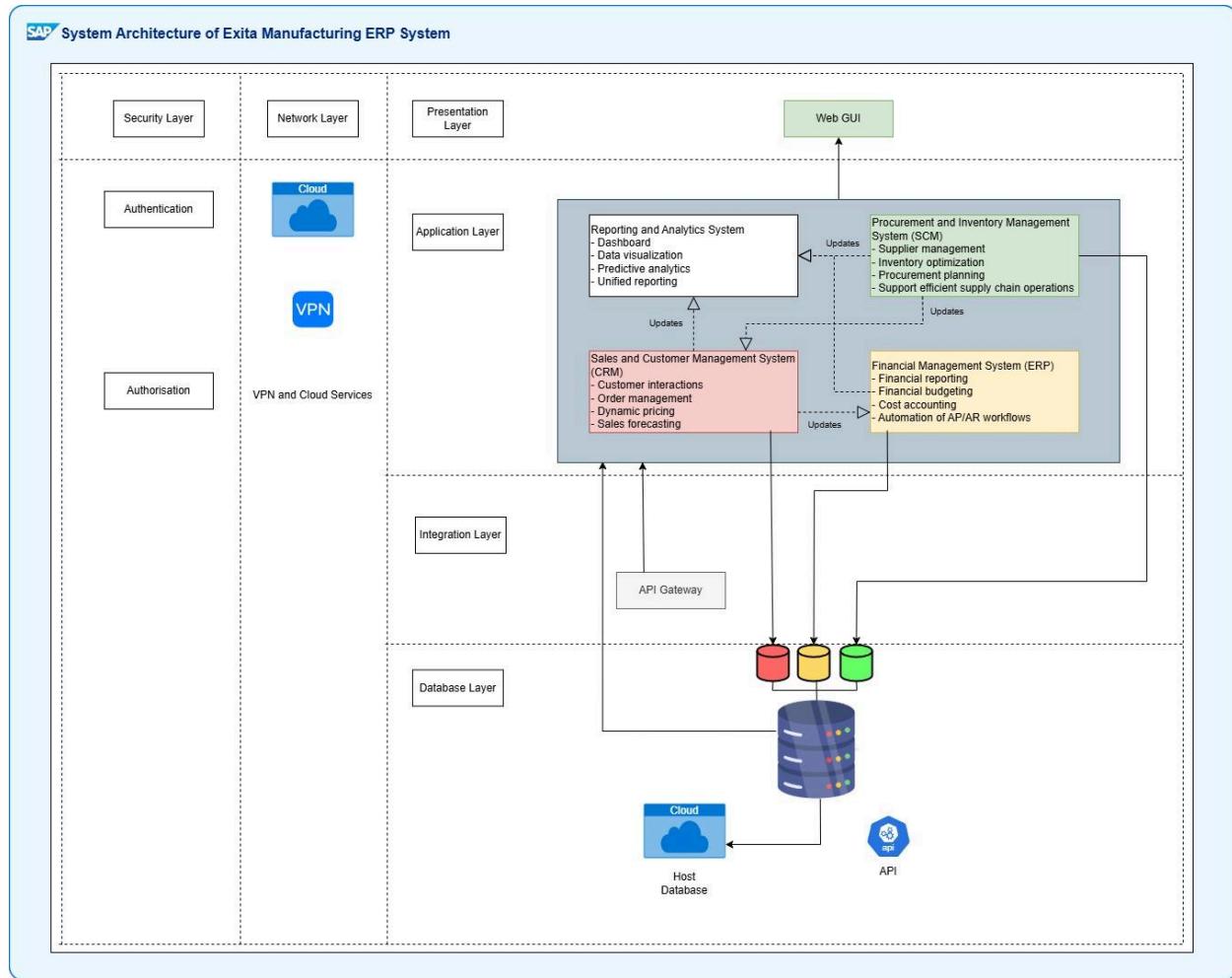


Figure 4.6.3 : System Architecture of Exita Manufacturing ERP System

4.6.4 Components Of System Architecture

1. Database Layer

- **Database Management:**
 - **SAP HANA:** The central database that facilitates real-time data processing and analytics from Sales and Customer Management System, Financial Management System and Procurement and Inventory Management System.
 - **Data Warehousing:** Consolidation of data from various subsystems for reporting and analysis.
- **APIs:**
 - **API Integration:** Use of Application Programming Interfaces (APIs) to allow for real-time data access and ensure that all systems are synchronized.

2. Application Layer

- **Financial Reporting System (ERP):**
 - **SAP S/4HANA:** Central ERP solution integrating core business functions such as Sales, Finance, Procurement, and Human Resources.
- **Reporting and Analytics:**
 - **SAP Analytics Cloud:** Provides dashboards and visualizations for real-time insights into key performance indicators (KPIs).
- **Development Tools:**
 - **SAP Build:** Used for creating custom applications and workflows tailored to business processes.
- **Sales and Customer Management System (CRM):**
 - Integrated customer relationship management capabilities that enhance customer interactions and sales management.

3. Presentation Layer

- **User Interface:**
 - **Web GUI:** A web-based interface for users to interact with the ERP system, ensuring accessibility across devices.

4. Integration Layer

- **API Gateway:**
 - Facilitates communication between various systems and modules, ensuring seamless data exchange.
- **Middleware:**
 - Tools that integrate legacy systems with the new ERP solution, enabling smooth data flow and interoperability.

5. Network Layer

- **Cloud Infrastructure:**
 - **SAP Business Technology Platform (BTP):** The underlying cloud infrastructure that supports application hosting and data storage.
- **VPN:**
 - Virtual Private Network solutions to ensure secure access to the ERP system for remote users.
 - **VPN Cloud Services:** Utilization of cloud-based VPN solutions to provide secure, scalable, and flexible remote access to the ERP system.

6. Security Layer

- **Authentication:**
 - Role-based access controls to ensure that only authorized users can access sensitive data and functionalities.
- **Authorization:**
 - The permissions and restrictions for each user role, ensuring that individuals can only perform actions that align with their responsibilities. This includes read, write, and administrative permissions tailored to specific user needs.

Summary

The aspects of Exita Manufacturing's ERP system architecture combine to operate harmoniously and enhance operational and strategic organizational achievements. Every architectural layer acts independently to provide secure timely authorized access for efficient information delivery that supports productive decision making processes and improved business output.

The architecture system delivers complete satisfaction of business and technical needs thereby allowing Exita Manufacturing to execute efficiently in an intense market framework.

4.7 Project Design

4.7.1 Use Case Diagram - Enterprise System

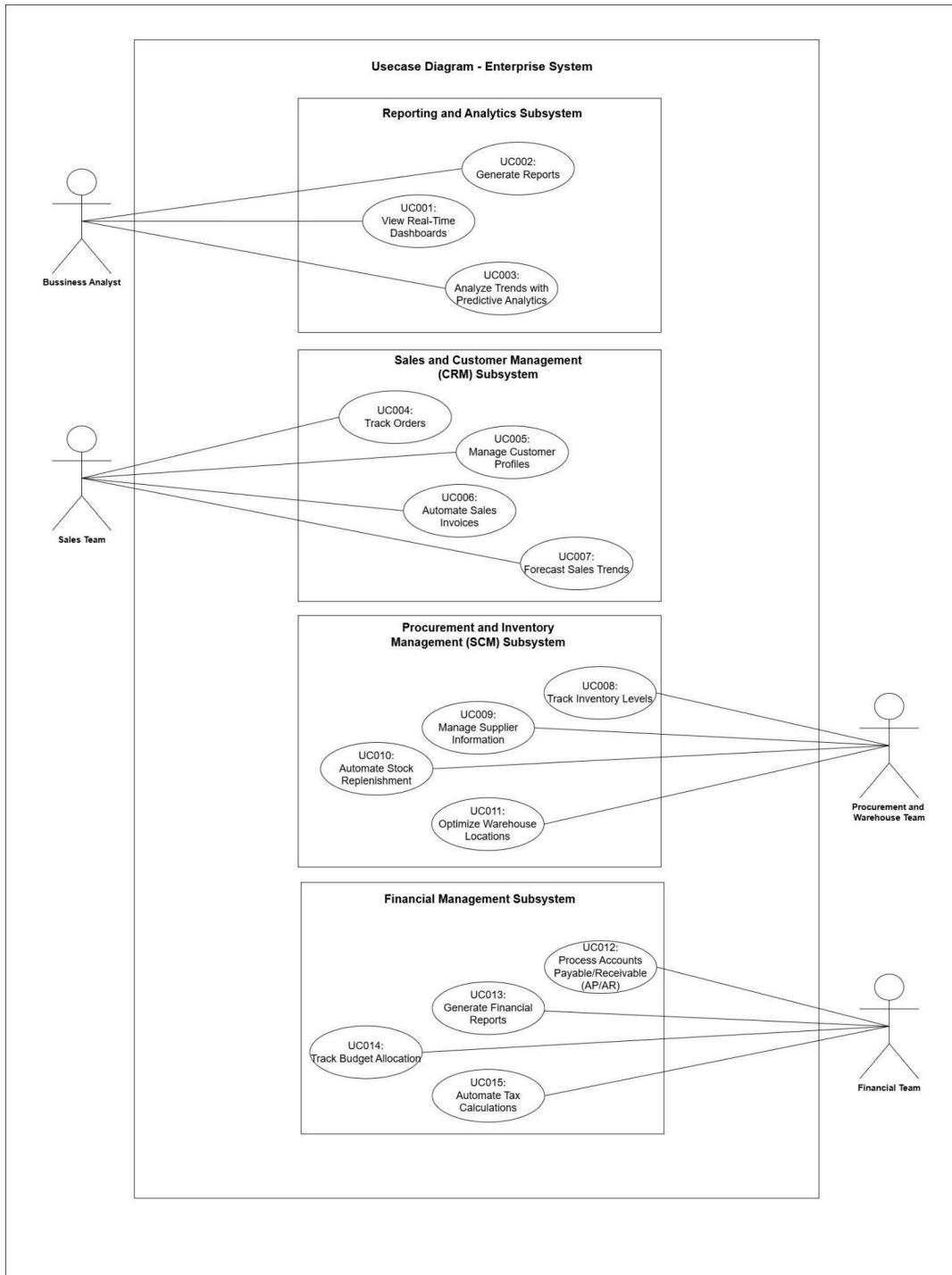


Figure 4.7.1: Use Case Diagram for Enterprise System

4.7.2 Detailed Use Case Diagram

4.7.2.1 Reporting and Analytics Subsystem

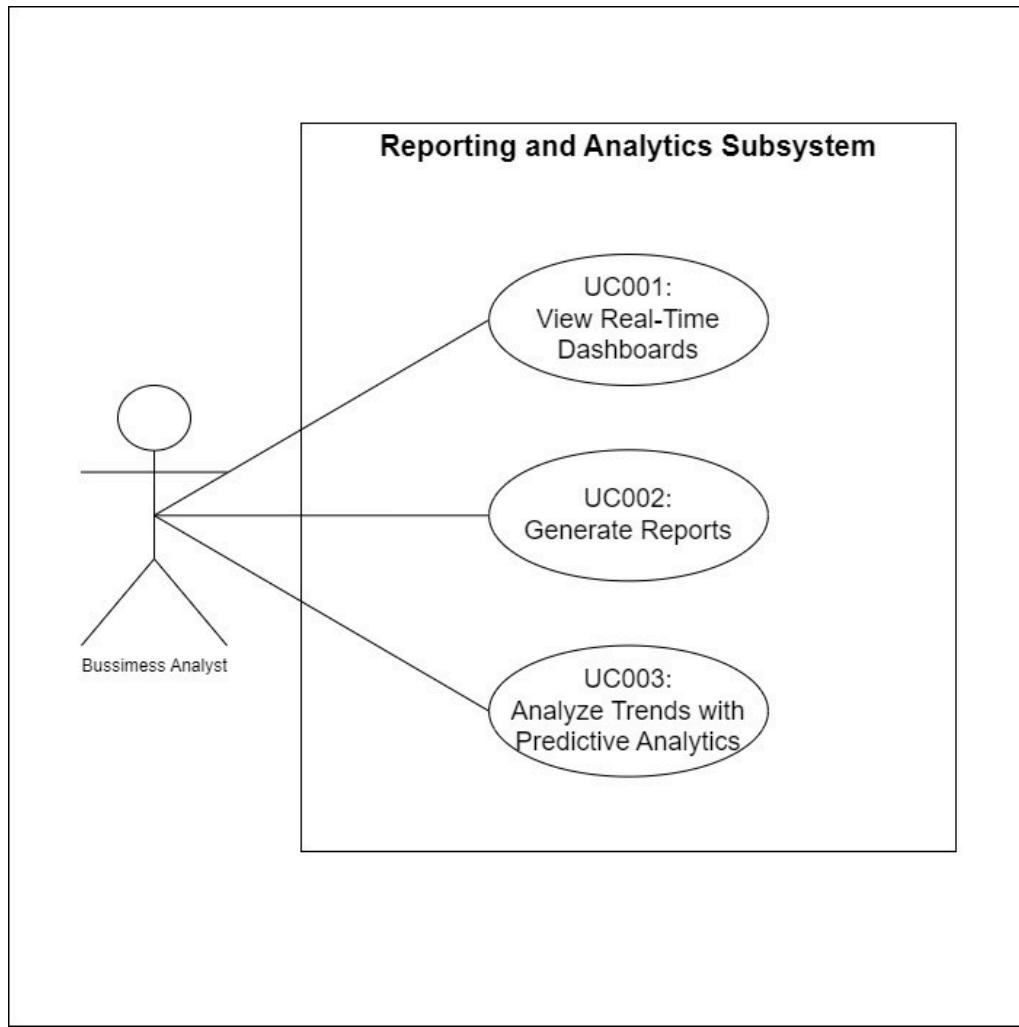


Figure 4.7.2.1: Use Case Diagram for Reporting and Analytics Subsystem

4.7.2.2 Sales and Customer Management (CRM) Subsystem

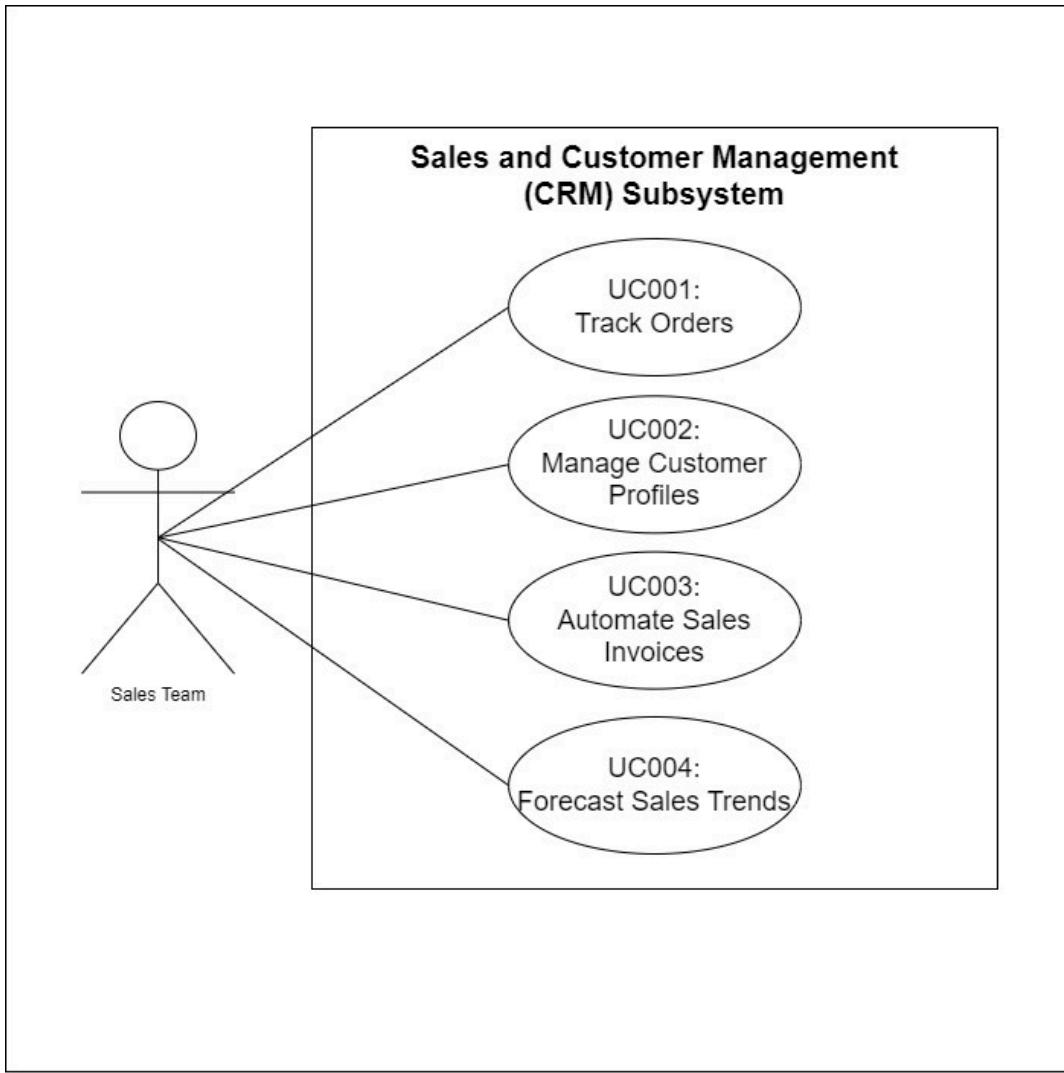


Figure 4.7.2.2: Use Case Diagram for Sales and Customer Management (CRM) Subsystem

4.7.2.3 Procurement and Inventory Management (SCM) Subsystem

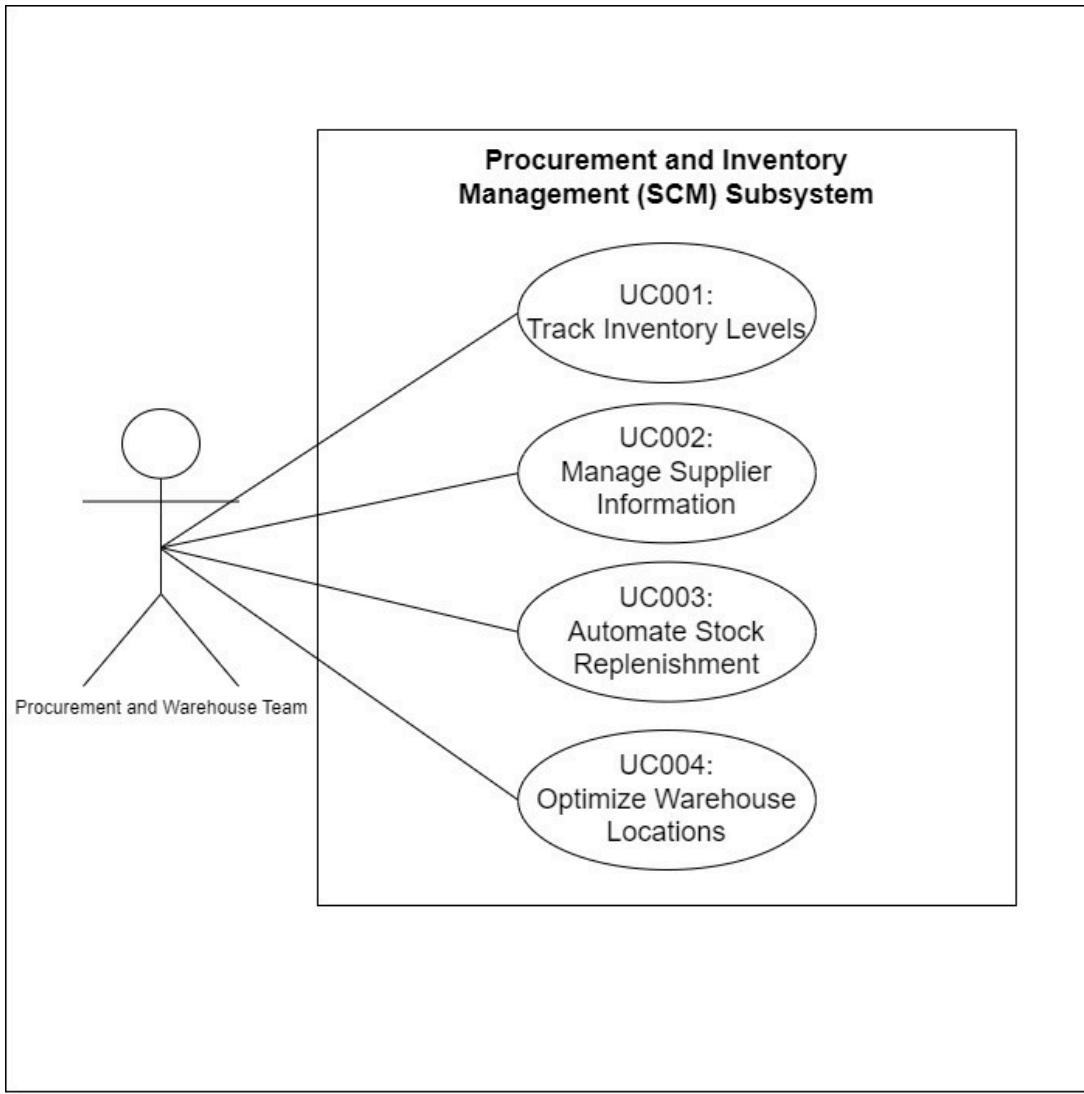


Figure 4.7.2.3: Use Case Diagram for Procurement and Inventory Management (SCM) Subsystem

4.7.2.4 Financial Management Subsystem

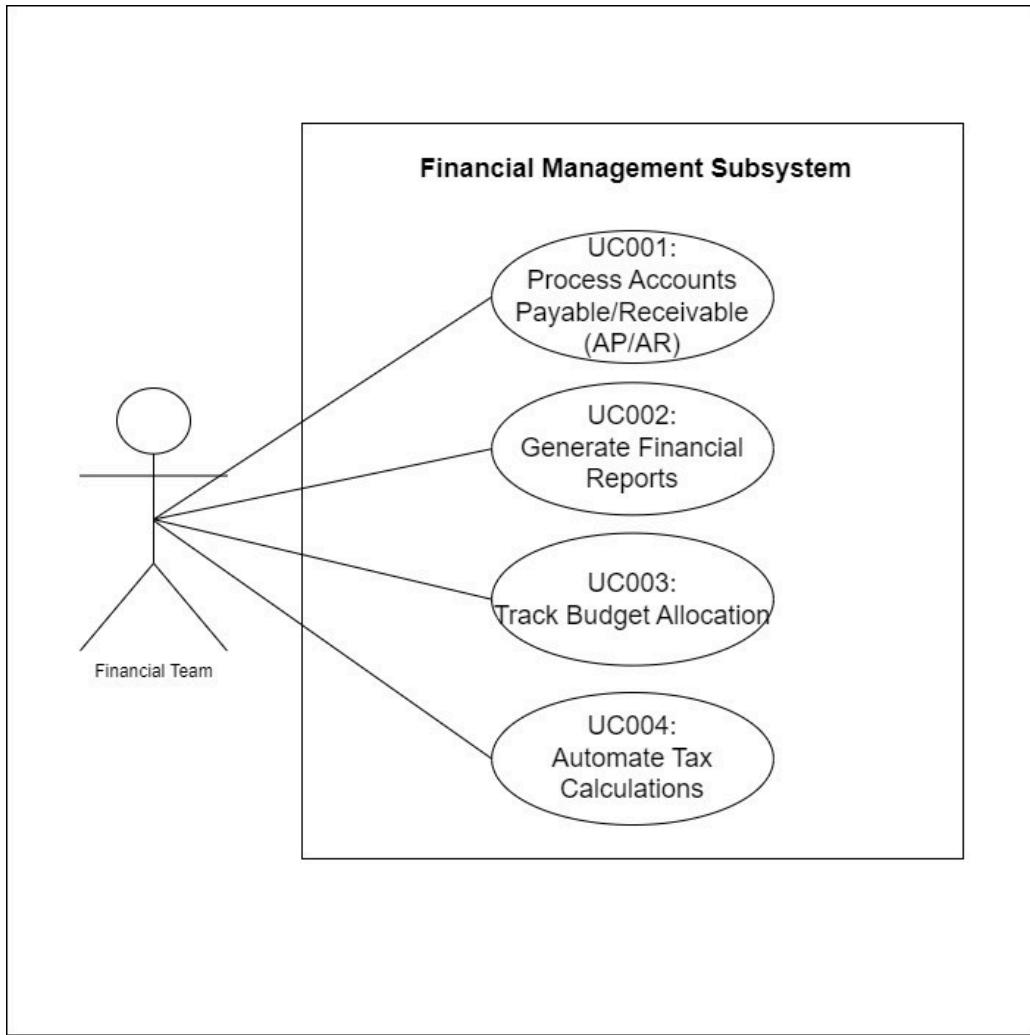


Figure 4.7.2.3: Use Case Diagram for Financial Management Subsystem

4.7.3 Use Case Description

4.7.3.1 Reporting and Analytics Subsystem

4.7.3.1.1 Use Case Description: View Real-Time Dashboards

Use Case ID:	UC001					
Use Case Name:	View Real-Time Dashboards					
Scenario:	<p>A business analyst needs to monitor key performance indicators (KPIs) such as sales performance, inventory levels, and financial summaries in real-time. The analyst logs into the ERP system, navigates to the Reporting and Analytics module, and selects the "Dashboard" option. The system retrieves real-time data, updates the dashboard widgets, and displays the visual insights on the user interface.</p>					
Triggering Event:	<p>The actor requires real-time data to monitor business operations, triggered by management queries, operational reviews, or decision-making needs.</p>					
Brief Description:	<p>This use case describes the process by which an actor uses the ERP system to generate a report by accessing historical and real-time data from a centralized database. The system retrieves, processes, and presents the requested data in a structured and visual format to support analysis and decision-making.</p>					
Actors:	Business Analyst					
Precondition:	<ol style="list-style-type: none"> 1. The user has valid login credentials and access to the Reporting and Analytics module. 2. Real-time data integration is configured between all relevant departments in the ERP system. 					
Postcondition:	<ol style="list-style-type: none"> 1. A real-time dashboard is displayed with the latest data. 2. The actor can monitor, interact with, and customize dashboard widgets as required. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Reporting and Analytics module. 3. Selects the "Dashboard" option. 4. Customizes the dashboard by selecting or rearranging widgets. 5. Reviews the real-time KPIs displayed on the dashboard. 6. Uses drill-down functionality to explore specific metrics. </td> <td> <ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Reporting and Analytics module interface. 3. Displays the default dashboard layout with predefined widgets. 4. Updates the dashboard layout and saves customization preferences. 5. Retrieves live data from the integrated modules and updates the dashboard widgets. </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Reporting and Analytics module. 3. Selects the "Dashboard" option. 4. Customizes the dashboard by selecting or rearranging widgets. 5. Reviews the real-time KPIs displayed on the dashboard. 6. Uses drill-down functionality to explore specific metrics. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Reporting and Analytics module interface. 3. Displays the default dashboard layout with predefined widgets. 4. Updates the dashboard layout and saves customization preferences. 5. Retrieves live data from the integrated modules and updates the dashboard widgets. 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Reporting and Analytics module. 3. Selects the "Dashboard" option. 4. Customizes the dashboard by selecting or rearranging widgets. 5. Reviews the real-time KPIs displayed on the dashboard. 6. Uses drill-down functionality to explore specific metrics. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Reporting and Analytics module interface. 3. Displays the default dashboard layout with predefined widgets. 4. Updates the dashboard layout and saves customization preferences. 5. Retrieves live data from the integrated modules and updates the dashboard widgets. 					

		<p>accordingly.</p> <p>6. Fetches and displays additional data for the selected metric in a detailed view.</p>
Exception Description:	<p>System Downtime: The system notifies the actor about downtime and provides the last cached data (if available).</p> <p>Permission Denied: The system blocks access and displays an error message if the user lacks required permissions.</p> <p>Data Latency: If real-time data cannot be retrieved due to integration issues, the system displays a warning message and partially updates the dashboard.</p>	

4.7.3.1.2 Use Case Description: Generate Report

Use Case ID:	UC002					
Use Case Name:	Generate Report					
Scenario:	<p>A business analyst is tasked with preparing a sales performance report for the previous quarter. The analyst logs into the ERP system, navigates to the Reporting and Analytics module, selects the "Sales Report" template, and defines parameters such as the date range and specific sales regions. The system processes the request, generates the report with visual insights, and displays it for review and sharing.</p>					
Triggering Event:	<p>The actor requires insights or data for decision-making, prompted by a scheduled task, a request from management, or operational needs (e.g., end-of-month sales performance review)</p>					
Brief Description:	<p>This use case describes the process by which an actor uses the ERP system to generate a report by accessing historical and real-time data from a centralized database. The system retrieves, processes, and presents the requested data in a structured and visual format to support analysis and decision-making.</p>					
Actors:	Business Analyst					
Precondition:	<ol style="list-style-type: none"> 1. The user has access to the Reporting and Analytics module in the ERP system. 2. The system contains historical and real-time data integrated from all departments. 					
Postcondition:	<ol style="list-style-type: none"> 1. A report is generated and stored in the system. 2. The report is available for download, sharing, or further analysis. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Reporting and Analytics module. 3. Selects the type of report to generate (e.g., KPI Dashboard, Financial Report). 4. Specifies report parameters such as date range, department, or metrics to include. 5. Submits the report request to the system. 6. Reviews the generated report displayed by the system. 7. Saves, downloads, or shares the report based on their needs. </td> <td> <ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Reporting and Analytics module interface. 3. Displays the available report templates for selection. 4. Accepts the parameters provided by the actor and validates them. 5. Retrieves relevant data from the centralized database based on the specified parameters. 6. Processes the data to generate the report, including creating visualizations (charts, graphs, tables). 7. Displays the generated report on </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Reporting and Analytics module. 3. Selects the type of report to generate (e.g., KPI Dashboard, Financial Report). 4. Specifies report parameters such as date range, department, or metrics to include. 5. Submits the report request to the system. 6. Reviews the generated report displayed by the system. 7. Saves, downloads, or shares the report based on their needs. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Reporting and Analytics module interface. 3. Displays the available report templates for selection. 4. Accepts the parameters provided by the actor and validates them. 5. Retrieves relevant data from the centralized database based on the specified parameters. 6. Processes the data to generate the report, including creating visualizations (charts, graphs, tables). 7. Displays the generated report on 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Reporting and Analytics module. 3. Selects the type of report to generate (e.g., KPI Dashboard, Financial Report). 4. Specifies report parameters such as date range, department, or metrics to include. 5. Submits the report request to the system. 6. Reviews the generated report displayed by the system. 7. Saves, downloads, or shares the report based on their needs. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Reporting and Analytics module interface. 3. Displays the available report templates for selection. 4. Accepts the parameters provided by the actor and validates them. 5. Retrieves relevant data from the centralized database based on the specified parameters. 6. Processes the data to generate the report, including creating visualizations (charts, graphs, tables). 7. Displays the generated report on 					

		<p>the user interface.</p> <p>8. Provides options to save, download, or share the report</p>
Exception Description:	<p>System Downtime: The user is notified, and the operation is queued or rescheduled.</p> <p>Permission Denied: If the user lacks authorization, the system blocks access and logs the event</p>	

4.7.3.1.3 Use Case Description: Analyze Trends with Predictive Analytics

Use Case ID:	UC003	
Use Case Name:	Analyze Trends with Predictive Analytics	
Scenario:	Business Analyst employs the tools of predictive analytics to single out business trends in total business data for decision-making purposes.	
Triggering Event:	The business analyst receives a request from the management informing the business analyst that they are interested in getting a trend analysis of the sales for the last quarter.	
Brief Description:	This use case outlines how the business analyst is able to incorporate the work, finance and performance data into the ERP system and utilise analytic tools to assess the prior performance of the business and estimate future trends. The aim is to generate knowledge that can inform strategies and provide insights.	
Actors:	Business Analyst	
Precondition:	<ol style="list-style-type: none"> 1. The business analyst has information access to the ERP reporting and analytics sub-system. 2. Proper records of the past data exist in the business and at the same time are implemented effectively in the ERP system. 	
Postcondition:	<ol style="list-style-type: none"> 1. The business analyst produces a report on business trends and future forecasts. 2. Proprietary knowledge and best practices are shared with management for the purpose of decision-making. 	
Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Analytics module. 3. Accesses historical data relevant to the organization. 4. Selects specific data sets for analysis. 5. Applies chosen predictive analytics tools. 6. Reviews the generated analytical output and insights. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the analytics module. 2. Retrieves historical data from integrated sources. 3. Filters data based on user input criteria. 4. Executes selected predictive analytics algorithms on the filtered data. 5. Generates and displays the analytical output. 6. Stores the analytical report data for future reference.
Exception Description:	Data Unavailability: If the historical data is either missing or contains errors, the analyst will have to contact the IT department to sort out data-related problems.	

4.7.3.2 Sales and Customer Management (CRM) Subsystem

4.7.3.2.1 Use Case Description: Track Customer Orders

Use Case ID:	UC004					
Use Case Name:	Track Customer Orders					
Scenario:	<p>A sales representative needs to monitor and update the status of customer orders. The representative logs into the ERP system, navigates to the Sales and Customer Management module, selects the "Customer Orders" option, and searches for specific orders using filters like order ID, customer name, or date. The system retrieves and displays the relevant order details in real-time, allowing the representative to update the status or add remarks as needed.</p>					
Triggering Event:	<p>The actor requires order information for a customer query, operational updates, or to resolve an issue (e.g., delayed shipment).</p>					
Brief Description:	<p>This use case describes the process by which an actor tracks customer orders by accessing real-time data. The ERP system enables the actor to search, view, and manage the status of orders to ensure timely fulfillment and customer satisfaction.</p>					
Actors:	Sales Representative					
Precondition:	<ol style="list-style-type: none"> 1. The actor has valid credentials and access to the Sales and Customer Management module. 2. Order data is stored in the centralized ERP database. 					
Postcondition:	<ol style="list-style-type: none"> 1. The actor successfully retrieves the required customer order information. 2. Updates made to order status or remarks are stored in the system. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Customer Orders" option. 4. Inputs search criteria (e.g., order ID, customer name, or date range) to locate specific orders. 5. Reviews the displayed order details. 6. Updates the order status (e.g., "Processing," "Shipped") or adds remarks, if needed. 7. Confirms and submits the updates. </td> <td> <ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Customer Orders" option. 4. Processes the search criteria provided by the user. 5. Retrieves and displays relevant order details from the centralized database. 6. Validates updates made by the user (e.g., checks for valid statuses or remarks). 7. Saves the updates to the database and confirms </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Customer Orders" option. 4. Inputs search criteria (e.g., order ID, customer name, or date range) to locate specific orders. 5. Reviews the displayed order details. 6. Updates the order status (e.g., "Processing," "Shipped") or adds remarks, if needed. 7. Confirms and submits the updates. 	<ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Customer Orders" option. 4. Processes the search criteria provided by the user. 5. Retrieves and displays relevant order details from the centralized database. 6. Validates updates made by the user (e.g., checks for valid statuses or remarks). 7. Saves the updates to the database and confirms 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Customer Orders" option. 4. Inputs search criteria (e.g., order ID, customer name, or date range) to locate specific orders. 5. Reviews the displayed order details. 6. Updates the order status (e.g., "Processing," "Shipped") or adds remarks, if needed. 7. Confirms and submits the updates. 	<ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Customer Orders" option. 4. Processes the search criteria provided by the user. 5. Retrieves and displays relevant order details from the centralized database. 6. Validates updates made by the user (e.g., checks for valid statuses or remarks). 7. Saves the updates to the database and confirms 					

		successful changes to the user.
Exception Description:	<p>Invalid Order ID: The system notifies the actor if no order matches the search criteria.</p> <p>Permission Denied: If the actor lacks authorization to modify order status, the system blocks the action and logs the event.</p> <p>Data Integration Issue: The system displays a warning message if order details cannot be retrieved due to integration or database errors.</p>	

4.7.3.2.2 Use Case Description: Manage Customer Profiles

Use Case ID:	UC005	
Use Case Name:	Manage Customer Profiles	
Scenario:	<p>A sales representative needs to create or update customer information in the ERP system. The representative logs into the system, navigates to the Sales and Customer Management module, selects the "Customer Profiles" option, and either searches for an existing customer or chooses to add a new one. The system validates the inputs and saves the changes in the centralized database, ensuring the updated or new customer profile is accessible across all related modules.</p>	
Triggering Event:	<p>The actor is prompted to manage customer profiles due to operational needs, such as onboarding a new customer, updating contact details, or correcting existing information.</p>	
Brief Description:	<p>This use case describes how an actor manages customer profiles by adding, updating, or searching for customer information. The ERP system ensures that all customer data is stored and synchronized across modules for consistency and accessibility.</p>	
Actors:	Sales Representative	
Precondition:	<ol style="list-style-type: none"> 1. The actor has valid credentials and access to the Sales and Customer Management module. 2. The centralized database is operational and contains existing customer data. 	
Postcondition:	<ol style="list-style-type: none"> 1. A new customer profile is created or an existing profile is updated successfully. 2. The updated profile is synchronized across all relevant modules in the ERP system. 	
Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and 	<ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system.

	<p>Customer Management module.</p> <ol style="list-style-type: none"> 3. Selects the "Customer Profiles" option. 4. Searches for an existing customer profile or selects "Add New Customer." 5. Inputs or updates customer details (e.g., name, contact information, address). 6. Reviews the entered or updated information. 7. Confirms and submits the changes. 	<ol style="list-style-type: none"> 2. Loads the Sales and Customer Management module interface. 3. Displays the "Customer Profiles" section. 4. Processes the user's request to search or add a customer profile. 5. Retrieves existing customer data or validates new inputs. 6. Saves the new or updated customer information in the centralized database. 7. Confirms successful profile creation or update to the user.
Exception Description:	<p>Duplicate Customer Record: If the customer already exists, the system notifies the actor and prompts for merging or updating the profile.</p> <p>Validation Errors: The system highlights missing or invalid inputs, such as an incorrect phone number or incomplete address.</p> <p>Permission Denied: If the actor lacks authorization, the system blocks the action and logs the event.</p>	

4.7.3.2.3 Use Case Description: Automate Sales Invoices

Use Case ID:	UC006
Use Case Name:	Automate Sales Invoices
Scenario:	A sales representative completes an order and needs to generate an invoice automatically. The representative logs into the ERP system, navigates to the "Invoices" section in the Sales and Customer Management module, and selects the order for which the invoice is required. The system retrieves the order details, applies the necessary pricing and tax rules, and generates the invoice. The invoice is saved in the system and made available for download, email, or printing.
Triggering Event:	The actor completes a customer order or receives a request for invoicing.
Brief Description:	This use case describes how the ERP system generates sales invoices automatically by processing order data and applying pricing and tax rules. The system ensures accuracy and consistency in invoicing, reducing manual errors.
Actors:	Sales Representative
Precondition:	<ol style="list-style-type: none"> 1. The actor has valid credentials and access to the Sales and Customer Management module. 2. The system contains order details, including pricing, taxes, and payment terms.

Postcondition:	<ol style="list-style-type: none"> 1. An invoice is generated, stored in the system, and linked to the corresponding order. 2. The invoice is available for download, sharing, or printing. 	
Flow of Activities:	Actor <ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Invoices" section. 4. Searches for and selects the completed order for invoicing. 5. Confirms the invoice generation request. 6. Reviews the generated invoice for accuracy. 7. Downloads, emails, or prints the invoice as needed. 	System <ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Invoices" section. 4. Retrieves the selected order details, including pricing, taxes, and payment terms. 5. Processes the order data to generate the invoice. 6. Saves the generated invoice in the centralized database and links it to the corresponding order. 7. Provides the user with options to download, email, or print the invoice.
Exception Description:	<p>Invalid Order Selection: The system notifies the actor if the selected order is incomplete or invalid for invoicing.</p> <p>System Downtime: If the system is unavailable, the operation is queued, and the actor is notified to retry later.</p> <p>Data Validation Errors: If pricing or tax data is missing, the system alerts the actor and halts the invoicing process.</p>	

4.7.3.2.4 Use Case Description: Forecast Sales Trends

Use Case ID:	UC007
Use Case Name:	Forecast Sales Trends
Scenario:	A business analyst is tasked with predicting future sales trends to support strategic planning. The analyst logs into the ERP system, navigates to the "Analytics" module, and selects the "Sales Trends Forecast" option. The analyst defines parameters such as the time period, product categories, and regions. The system processes historical sales data, applies predictive analytics algorithms, and generates a visual representation of forecasted trends, which the analyst can review and export for reporting purposes.
Triggering:	The actor requires predictive insights for sales planning or strategic

Event:	decision-making.					
Brief Description:	This use case describes how an actor uses the ERP system's analytics capabilities to forecast future sales trends. The system retrieves historical sales data, applies predictive models, and presents visual insights to the user.					
Actors:	Sales Department Manager					
Precondition:	<ol style="list-style-type: none"> 1. The actor has valid credentials and access to the Analytics module in the ERP system. 2. Historical sales data is available in the centralized database. 3. Predictive analytics functionality is configured and operational. 					
Postcondition:	<ol style="list-style-type: none"> 1. Forecasted sales trends are generated and displayed to the actor. 2. The forecast can be exported or integrated into reports for further analysis. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Sales Trends Forecast" option. 4. Defines parameters such as time period, product categories, and regions for the forecast. 5. Submits the request to generate the forecast. 6. Reviews the visual representation of forecasted sales trends. 7. Exports the forecast as a report or integrates it into other analyses. </td> <td> <ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Sales Trends Forecast" option. 4. Validates and processes the parameters provided by the user. 5. Retrieves historical sales data from the centralized database. 6. Applies predictive analytics models to generate forecasted sales trends. 7. Displays the forecast in visual formats such as graphs and charts. 8. Provides options for exporting or integrating the forecast data. </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Sales Trends Forecast" option. 4. Defines parameters such as time period, product categories, and regions for the forecast. 5. Submits the request to generate the forecast. 6. Reviews the visual representation of forecasted sales trends. 7. Exports the forecast as a report or integrates it into other analyses. 	<ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Sales Trends Forecast" option. 4. Validates and processes the parameters provided by the user. 5. Retrieves historical sales data from the centralized database. 6. Applies predictive analytics models to generate forecasted sales trends. 7. Displays the forecast in visual formats such as graphs and charts. 8. Provides options for exporting or integrating the forecast data. 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system using valid credentials. 2. Navigates to the Sales and Customer Management module. 3. Selects the "Sales Trends Forecast" option. 4. Defines parameters such as time period, product categories, and regions for the forecast. 5. Submits the request to generate the forecast. 6. Reviews the visual representation of forecasted sales trends. 7. Exports the forecast as a report or integrates it into other analyses. 	<ol style="list-style-type: none"> 1. Authenticates the user's credentials and grants access to the system. 2. Loads the Sales and Customer Management module interface. 3. Displays the "Sales Trends Forecast" option. 4. Validates and processes the parameters provided by the user. 5. Retrieves historical sales data from the centralized database. 6. Applies predictive analytics models to generate forecasted sales trends. 7. Displays the forecast in visual formats such as graphs and charts. 8. Provides options for exporting or integrating the forecast data. 					
Exception Description:	<p>Insufficient Data: The system notifies the actor if there is inadequate historical data to generate a reliable forecast.</p> <p>System Downtime: The operation is queued, and the actor is notified if the system is temporarily unavailable.</p> <p>Invalid Parameters: The system highlights errors in user-defined parameters and prompts for corrections.</p>					

4.7.3.3 Procurement and Inventory Management (SCM) Subsystem

4.7.3.3.1 Use Case Description: Track Inventory Levels

Use Case ID:	UC008
---------------------	-------

Use Case Name:	Track Inventory Levels					
Scenario:	The Procurement and Warehouse Team monitors and manages the inventory in order to avoid both stockouts and unnecessary stock accumulation.					
Triggering Event:	A regular inventory review is scheduled when inventory levels fall below a predefined threshold.					
Brief Description:	This use case outlines the procedure of using the SCM subsystem in real-life how the Procurement and Warehouse Team monitor inventory levels. Another is that the intended team can track the number of stocks, check on supplies that are running low, and even evaluate the turnover of stock supplies for the purpose of making sound procurement decisions.					
Actors:	Procurement and Warehouse Team					
Precondition:	<ol style="list-style-type: none"> 1. The inventory items are correctly grouped and captured in the system so that efficient analysis can be made. 					
Postcondition:	<ol style="list-style-type: none"> 1. Stock status is well managed and changes are updated automatically. 2. Alerts to indicate low stock are added and passed on to certain employees. 3. The recursive reports on inventory levels and turnover rates are provided for further analysis. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Inventory Management module. 3. Views current stock levels for all items. 4. Monitors for any inventory alerts indicating low stock. 5. Analyzes inventory levels to identify trends or anomalies. 6. Updates inventory records if discrepancies are found or adjustments are necessary. </td> <td> <ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Inventory Management module interface. 3. Retrieves real-time inventory data from the database. 4. Generates alerts when inventory levels fall below predefined thresholds. 5. Updates the inventory levels in the system following any changes made by the actor. 6. Maintains a log of changes to ensure an audit trail of inventory adjustments. </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Inventory Management module. 3. Views current stock levels for all items. 4. Monitors for any inventory alerts indicating low stock. 5. Analyzes inventory levels to identify trends or anomalies. 6. Updates inventory records if discrepancies are found or adjustments are necessary. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Inventory Management module interface. 3. Retrieves real-time inventory data from the database. 4. Generates alerts when inventory levels fall below predefined thresholds. 5. Updates the inventory levels in the system following any changes made by the actor. 6. Maintains a log of changes to ensure an audit trail of inventory adjustments. 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Inventory Management module. 3. Views current stock levels for all items. 4. Monitors for any inventory alerts indicating low stock. 5. Analyzes inventory levels to identify trends or anomalies. 6. Updates inventory records if discrepancies are found or adjustments are necessary. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Inventory Management module interface. 3. Retrieves real-time inventory data from the database. 4. Generates alerts when inventory levels fall below predefined thresholds. 5. Updates the inventory levels in the system following any changes made by the actor. 6. Maintains a log of changes to ensure an audit trail of inventory adjustments. 					
Exception Description:	<ol style="list-style-type: none"> 1. Data Entry Errors: In the case when some mistakes are entered while the data is being updated the program alerts the user to enter full information before the information is saved. 2. Data Discrepancies: Whenever there are small differences between the physical inventory and the data of the system, the team has to embark on an 					

	investigation to find out about the differences.
--	--

4.7.3.3.2 Use Case Description: Manage Supplier Information

Use Case ID:	UC009					
Use Case Name:	Manage Supplier Information					
Scenario:	The Procurement and Warehouse team create, improves or alters supplier records in order to provide the most current information for procurement activities.					
Triggering Event:	A supplier provides new information and then the change is made or a new supplier emerges in the system.					
Brief Description:	This use case outlines for the Procurement and Warehouse Team to use in managing supplier information. It comprises registering new suppliers, modifying information of a supplier currently in the system and deleting a supplier who is no longer necessary. Supplier records retention is therefore paramount in procurement and inventory management.					
Actors:	Procurement and Warehouse Team					
Precondition:	1. There are existing supplier records in the system data.					
Postcondition:	1. The information on the suppliers is well-updated in the system database. 2. Supplier records that are no longer in use, are deleted or disconnected for future access if needed. 3. The team produces reports of the suppliers' performance and reliability.					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> 1. Logs into the ERP system with valid credentials. 2. Navigates to the Supplier Management module. 3. Views the existing list of suppliers. 4. Adds a new supplier if needed, entering all required information. 5. Updates existing supplier information as necessary. </td> <td> 1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Supplier Management module interface. 3. Displays the current supplier data in a user-friendly format. 4. Processes the addition of a new supplier and stores the information in the database. 5. Updates the supplier data in the system following any changes made by the actor. </td> </tr> </tbody> </table>	Actor	System	1. Logs into the ERP system with valid credentials. 2. Navigates to the Supplier Management module. 3. Views the existing list of suppliers. 4. Adds a new supplier if needed, entering all required information. 5. Updates existing supplier information as necessary.	1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Supplier Management module interface. 3. Displays the current supplier data in a user-friendly format. 4. Processes the addition of a new supplier and stores the information in the database. 5. Updates the supplier data in the system following any changes made by the actor.	
Actor	System					
1. Logs into the ERP system with valid credentials. 2. Navigates to the Supplier Management module. 3. Views the existing list of suppliers. 4. Adds a new supplier if needed, entering all required information. 5. Updates existing supplier information as necessary.	1. Authenticates the actor's credentials and grants access to the system. 2. Loads the Supplier Management module interface. 3. Displays the current supplier data in a user-friendly format. 4. Processes the addition of a new supplier and stores the information in the database. 5. Updates the supplier data in the system following any changes made by the actor.					
Exception Description:	1. Data Entry Errors: In the case when some mistakes are entered while the data is being entered – for example, wrong contact details – the program					

	alerts the user to enter the correct information before the information is saved.
--	---

4.7.3.3.3 Use Case Description: Automate Stock Replenishment

Use Case ID:	UC010					
Use Case Name:	Automate Stock Replenishment					
Scenario:	The Procurement and Warehouse Team automate the replenishment of stock levels based on predefined inventory thresholds.					
Triggering Event:	When the inventory levels drop from the predefined threshold.					
Brief Description:	This use case outlines the mechanism of the stock replenishment activity accomplished in the SCM subsystem. The actual inventory is managed in real-time, and the purchase order is processed automatically when the inventory is low, thereby helping the warehouse control the most optimum levels of inventory without having to intervene physically.					
Actors:	Procurement and Warehouse Team					
Precondition:	1. It has inventory thresholds set and maintained within the system.					
Postcondition:	1. Purchase orders go out to suppliers automatically when the inventory levels drop from the predefined threshold. 2. Inventory levels are adjusted right away in accordance with new orders by using updated inventories. 3. The team can monitor, control and review replenishment actions through reports.					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Receives automated alerts for low stock levels. 2. Reviews the generated purchase orders for accuracy and necessity.</td> <td>1. Retrieves real-time inventory data to assess current stock levels. 2. Monitors inventory levels continuously against predefined thresholds. 3. Triggers the replenishment process when inventory levels fall below thresholds. 4. Generates purchase orders based on inventory needs. 5. Sends approved purchase orders to designated suppliers.</td> </tr> </tbody> </table>	Actor	System	1. Receives automated alerts for low stock levels. 2. Reviews the generated purchase orders for accuracy and necessity.	1. Retrieves real-time inventory data to assess current stock levels. 2. Monitors inventory levels continuously against predefined thresholds. 3. Triggers the replenishment process when inventory levels fall below thresholds. 4. Generates purchase orders based on inventory needs. 5. Sends approved purchase orders to designated suppliers.	
Actor	System					
1. Receives automated alerts for low stock levels. 2. Reviews the generated purchase orders for accuracy and necessity.	1. Retrieves real-time inventory data to assess current stock levels. 2. Monitors inventory levels continuously against predefined thresholds. 3. Triggers the replenishment process when inventory levels fall below thresholds. 4. Generates purchase orders based on inventory needs. 5. Sends approved purchase orders to designated suppliers.					

		6. Updates inventory records to reflect changes after orders are placed.
Exception Description:	1. Supplier Issues: In case the supplier cannot meet the order requirements due to stock out, the system should alert the team to look for an alternative supplier.	

4.7.3.3.4 Use Case Description: Optimize Warehouse Locations

Use Case ID:	UC011					
Use Case Name:	Optimize Warehouse Locations					
Scenario:	The Procurement and Warehouse Team analyzes existing warehouse layouts and inventory data to minimize retrieval times and improve the efficiency of storage locations.					
Triggering Event:	Additional costs, or productivity losses due to typical problems such as a higher level of inventory handling and storage.					
Brief Description:	This use case outlines the procedures for improving locations within the warehouse. Members of the Procurement and Warehouse Team will compare the current location of inventories to determine their pick patterns and come up with changes to the effective lay down of inventories and use of available space to improve the existing control of stock levels and the search time.					
Actors:	Procurement and Warehouse Team					
Precondition:	<ol style="list-style-type: none"> 1. The existing warehouse layout and inventory data are available in the system data. 					
Postcondition:	<ol style="list-style-type: none"> 1. There is also improvement in the location of the warehouses in order to increase efficiency. 2. New arrangements of the layout and the position of the inventory warehouse are input into the system. 3. The team can produce reports regarding the changes to retrieve information. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Reviews the existing warehouse layout and configuration. 3. Analyzes inventory data to assess current storage efficiency. 4. Identifies inefficiencies in the warehouse setup, such as bottlenecks or underutilized space. 5. Creates an optimization plan to address identified inefficiencies. 6. Implements changes to the warehouse layout based on the optimization plan. 7. Monitors performance post-implementation to evaluate the effectiveness of the changes. </td> <td> <ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Retrieves comprehensive warehouse data, including layout and inventory statistics. 3. Displays warehouse data in an easily interpretable format to facilitate analysis. 4. Supports data analysis by providing tools for evaluating storage efficiency. 5. Updates the warehouse layout and inventory records in the system following any changes made by the actor. </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Reviews the existing warehouse layout and configuration. 3. Analyzes inventory data to assess current storage efficiency. 4. Identifies inefficiencies in the warehouse setup, such as bottlenecks or underutilized space. 5. Creates an optimization plan to address identified inefficiencies. 6. Implements changes to the warehouse layout based on the optimization plan. 7. Monitors performance post-implementation to evaluate the effectiveness of the changes. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Retrieves comprehensive warehouse data, including layout and inventory statistics. 3. Displays warehouse data in an easily interpretable format to facilitate analysis. 4. Supports data analysis by providing tools for evaluating storage efficiency. 5. Updates the warehouse layout and inventory records in the system following any changes made by the actor. 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system with valid credentials. 2. Reviews the existing warehouse layout and configuration. 3. Analyzes inventory data to assess current storage efficiency. 4. Identifies inefficiencies in the warehouse setup, such as bottlenecks or underutilized space. 5. Creates an optimization plan to address identified inefficiencies. 6. Implements changes to the warehouse layout based on the optimization plan. 7. Monitors performance post-implementation to evaluate the effectiveness of the changes. 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the system. 2. Retrieves comprehensive warehouse data, including layout and inventory statistics. 3. Displays warehouse data in an easily interpretable format to facilitate analysis. 4. Supports data analysis by providing tools for evaluating storage efficiency. 5. Updates the warehouse layout and inventory records in the system following any changes made by the actor. 					

Exception Description:	1. Data Inconsistencies: If there are differences between physical inventory and system records, further physical check has to be initiated by the user.
-------------------------------	---

4.7.3.4 Financial Management Subsystem

4.7.3.4.1 Use Case: Process Accounts Payable/Receivable (AP/AR)

Use Case ID:	UC012					
Use Case Name:	Process Accounts Payable/Receivable (AP/AR)					
Scenario:	<p>The finance team receives a supplier invoice for goods delivered. The team logs into the ERP system and enters the invoice details into the Accounts Payable module. The system validates the invoice against purchase orders, updates the accounts ledger, and schedules the payment based on the due date. Once the payment is made, the system records the transaction, updates cash flow data, and marks the invoice as paid.</p>					
Triggering Event:	<ol style="list-style-type: none"> 1. An invoice is received from a supplier or generated for a customer. 2. A scheduled payment deadline or receivable reminder is approaching 					
Brief Description:	<p>1. This use case describes how the Financial Management Subsystem automates the processing of accounts payable (AP) and accounts receivable (AR) tasks. The system records incoming and outgoing invoices, ensures compliance with financial regulations, and tracks the status of payments or receivables. It generates alerts for overdue items, updates financial records, and produces reports for management review</p>					
Actors:	Financial Team					
Precondition:	<ol style="list-style-type: none"> 1. The system has valid supplier and customer records. 2. Purchase orders and invoices are properly documented in the system. 3. The actor has the necessary permissions to access and modify AP/AR records 					
Postcondition:	<ol style="list-style-type: none"> 1. The accounts ledger is updated to reflect the processed transaction. 2. Payment or receivable records are updated and flagged as completed. 3. Financial reports are generated for review and compliance purposes 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Accounts Payable" or "Accounts Receivable" section. 3. Enters the details of the invoice or payment (e.g., supplier/customer, amount, due date). 4. Reviews and confirms the </td> <td> <ol style="list-style-type: none"> 7. Verifies the actor's credentials and grants access to the Financial Management module. 8. Retrieves existing data (e.g., purchase orders, past invoices, customer records). 9. Validates the invoice or payment details against the existing records (e.g., supplier/customer information). 10. Updates the accounts ledger </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Accounts Payable" or "Accounts Receivable" section. 3. Enters the details of the invoice or payment (e.g., supplier/customer, amount, due date). 4. Reviews and confirms the 	<ol style="list-style-type: none"> 7. Verifies the actor's credentials and grants access to the Financial Management module. 8. Retrieves existing data (e.g., purchase orders, past invoices, customer records). 9. Validates the invoice or payment details against the existing records (e.g., supplier/customer information). 10. Updates the accounts ledger 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Accounts Payable" or "Accounts Receivable" section. 3. Enters the details of the invoice or payment (e.g., supplier/customer, amount, due date). 4. Reviews and confirms the 	<ol style="list-style-type: none"> 7. Verifies the actor's credentials and grants access to the Financial Management module. 8. Retrieves existing data (e.g., purchase orders, past invoices, customer records). 9. Validates the invoice or payment details against the existing records (e.g., supplier/customer information). 10. Updates the accounts ledger 					

	<p>details entered.</p> <p>5. Initiates the payment (for AP) or records receipt of payment (for AR).</p> <p>6. Reviews updated financial records and generates required financial reports</p>	<p>with the new transaction.</p> <p>11. Sends alerts or reminders for overdue payments or receivables (if applicable).</p> <p>12. Generates and displays a summary of the updated financial records.</p> <p>13. Produces detailed reports on accounts payable/receivable status for download or sharing</p>
Exception Description:	<p>1. Data Discrepancy: If the invoice details do not match existing purchase orders, the system flags the issue and prompts the actor for correction.</p> <p>2. Payment Failure: If the payment cannot be processed (e.g., insufficient funds, incorrect account details), the system generates an alert and logs the error.</p> <p>3. Overdue Invoices: The system generates reminders for unpaid invoices or receivables</p>	

4.7.3.4.2 Use Case: Generate Financial Reports

Use Case ID:	UC013	
Use Case Name:	Generate Financial Reports	
Scenario:	<p>The finance team receives a supplier invoice for goods delivered. The team logs into the ERP system and enters the invoice details into the Accounts Payable module. The system validates the invoice against purchase orders, updates the accounts ledger, and schedules the payment based on the due date. Once the payment is made, the system records the transaction, updates cash flow data, and marks the invoice as paid.</p>	
Triggering Event:	<ol style="list-style-type: none"> 1. An invoice is received from a supplier or generated for a customer. 2. A scheduled payment deadline or receivable reminder is approaching 	
Brief Description:	<p>This use case describes how the Financial Management Subsystem automates the processing of accounts payable (AP) and accounts receivable (AR) tasks. The system records incoming and outgoing invoices, ensures compliance with financial regulations, and tracks the status of payments or receivables. It generates alerts for overdue items, updates financial records, and produces reports for management review.</p>	
Actors:	Financial Team	
Precondition:	<ol style="list-style-type: none"> 1. The system has valid supplier and customer records. 2. Purchase orders and invoices are properly documented in the system. 3. The actor has the necessary permissions to access and modify AP/AR records 	
Postcondition:	<ol style="list-style-type: none"> 1. The accounts ledger is updated to reflect the processed transaction. 2. Payment or receivable records are updated and flagged as completed. 3. Financial reports are generated for review and compliance purposes 	
Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Accounts Payable" or "Accounts Receivable" section. 3. Enters the details of the invoice or payment (e.g., supplier/customer, amount, due date). 4. Reviews and confirms the details entered. 5. Initiates the payment (for AP) or 	<ol style="list-style-type: none"> 1. Verifies the actor's credentials and grants access to the Financial Management module. 2. Retrieves existing data (e.g., purchase orders, past invoices, customer records). 3. Validates the invoice or payment details against the existing records (e.g., supplier/customer information). 4. Updates the accounts ledger with the new transaction. 5. Sends alerts or reminders for

	<p>records receipt of payment (for AR).</p> <p>6. Reviews updated financial records and generates required financial reports</p>	<p>overdue payments or receivables (if applicable).</p> <p>6. Generates and displays a summary of the updated financial records.</p> <p>7. Produces detailed reports on accounts payable/receivable status for download or sharing</p>
Exception Description:	<p>1. Data Discrepancy: If the invoice details do not match existing purchase orders, the system flags the issue and prompts the actor for correction.</p> <p>2. Payment Failure: If the payment cannot be processed (e.g., insufficient funds, incorrect account details), the system generates an alert and logs the error.</p> <p>3. Overdue Invoices: The system generates reminders for unpaid invoices or receivables</p>	

4.7.3.4.3 Use Case: Track Budget Allocation

Use Case ID:	UC014					
Use Case Name:	Track Budget Allocation					
Scenario:	<p>The finance team receives a supplier invoice for goods delivered. The team logs into the ERP system and enters the invoice details into the Accounts Payable module. The system validates the invoice against purchase orders, updates the accounts ledger, and schedules the payment based on the due date. Once the payment is made, the system records the transaction, updates cash flow data, and marks the invoice as paid.</p>					
Triggering Event:	<ol style="list-style-type: none"> 1. A department head requests an update on their allocated budget utilization. 2. Management requires a report on the overall company budget status for a specific period. 3. A scheduled task or audit necessitates tracking budget allocation and spending 					
Brief Description:	<p>This use case describes how the Financial Management Subsystem allows users to monitor and track budget allocations and expenditures across departments or projects. The system provides real-time insights into budget utilization, identifies overages or underutilization, and ensures compliance with financial planning goals</p>					
Actors:	Financial Team					
Precondition:	<ol style="list-style-type: none"> 1. Budget allocation data for all departments/projects is configured and updated in the system. 2. Financial transactions (e.g., expenditures) are accurately logged in the system. 3. The actor has valid access rights to view and track budget information 					
Postcondition:	<ol style="list-style-type: none"> 1. The system generates a detailed report comparing allocated budgets with actual expenditures. 2. Variances or anomalies in budget utilization are flagged for review. 3. The budget tracking report is stored in the system for future reference or export. 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Track Budget Allocation" function. 3. Specifies filters for the report (e.g., department, project, date range). 4. Submits the request to generate </td> <td> <ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the Financial Management module. 2. Retrieves budget allocation and expenditure data from the centralized database. 3. Processes the data to calculate variances between allocated </td> </tr> </tbody> </table>		Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Track Budget Allocation" function. 3. Specifies filters for the report (e.g., department, project, date range). 4. Submits the request to generate 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the Financial Management module. 2. Retrieves budget allocation and expenditure data from the centralized database. 3. Processes the data to calculate variances between allocated
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Track Budget Allocation" function. 3. Specifies filters for the report (e.g., department, project, date range). 4. Submits the request to generate 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the Financial Management module. 2. Retrieves budget allocation and expenditure data from the centralized database. 3. Processes the data to calculate variances between allocated 					

	<p>the budget tracking report.</p> <ol style="list-style-type: none"> 5. Reviews the generated report for insights and variances. 6. Saves, downloads, or shares the report as required 	<p>budgets and actual spending.</p> <ol style="list-style-type: none"> 4. Generates a report that includes summaries, visualizations (e.g., charts, graphs), and flagged variances. 5. Displays the report and provides options for export or sharing. 6. Logs the budget tracking activity for audit purposes
Exception Description:	<ol style="list-style-type: none"> 1. Data Inconsistencies: If discrepancies are found in the budget or expenditure data, the system flags them and notifies the actor. 2. Access Denied: The actor lacks sufficient privileges to access budget tracking data for certain departments or projects. 3. Outdated Data: The system prompts the actor to synchronize recent expenditure data if it is not updated 	

4.7.3.4.4 Use Case: Automate Tax Calculations

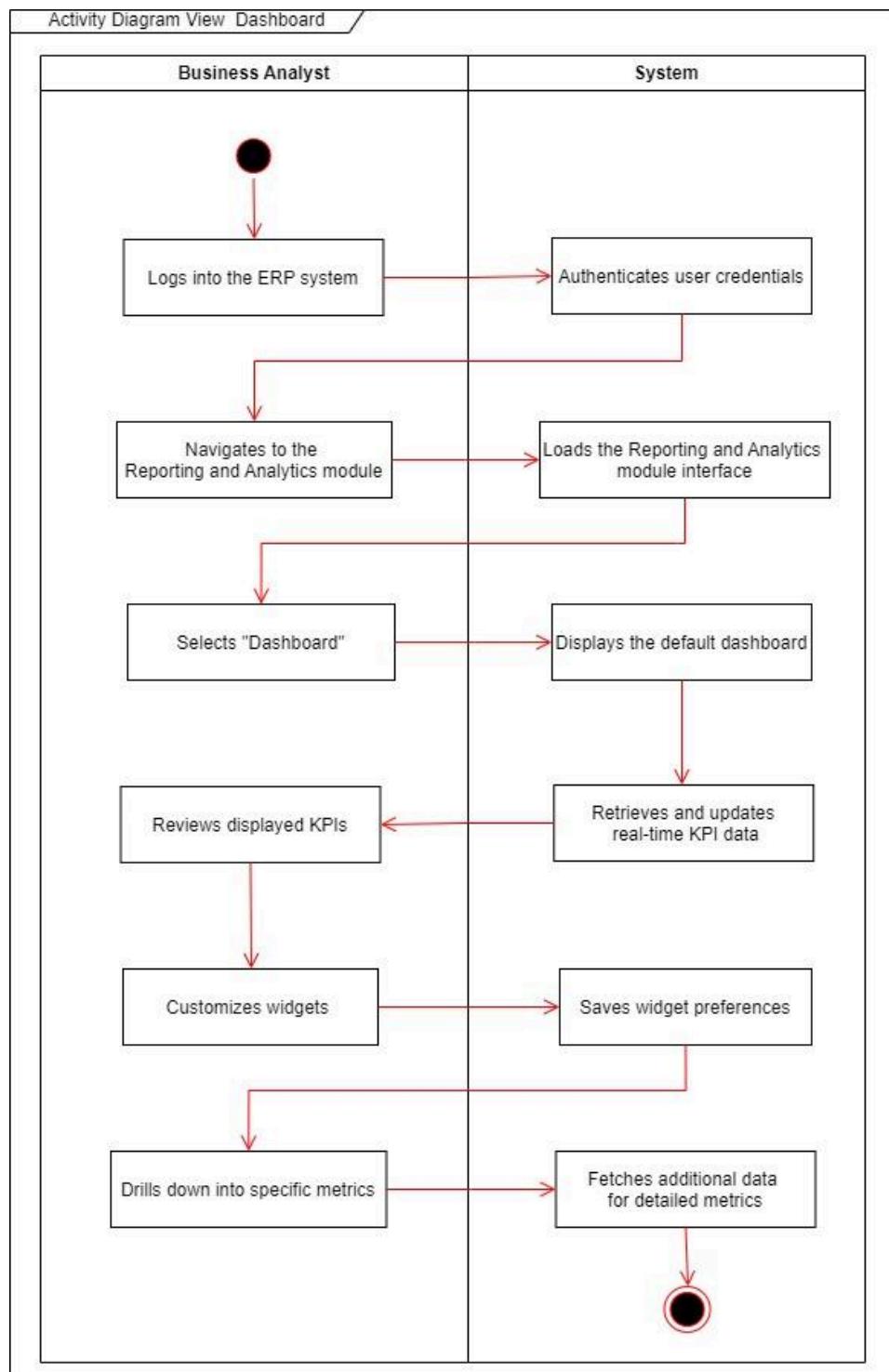
Use Case ID:	UC015					
Use Case Name:	Automate Tax Calculations					
Scenario:	<p>The finance team is preparing for quarterly tax submissions. A finance team member logs into the ERP system, navigates to the Financial Management module, and selects the "Automate Tax Calculations" feature. The system retrieves financial transaction data, applies predefined tax rules, calculates the tax amounts, and generates a detailed tax report. The report is reviewed and exported for submission to tax authorities</p>					
Triggering Event:	<ol style="list-style-type: none"> 1. A tax filing deadline is approaching, and the finance team initiates the tax calculation process. 2. A new invoice or financial transaction is processed that requires tax application. 3. Management requests a tax summary report for compliance or review. 					
Brief Description:	<p>This use case describes how the Financial Management Subsystem automatically calculates taxes for financial transactions, including accounts payable and receivable. The system applies appropriate tax rates (e.g., VAT, GST, corporate tax) based on predefined rules, generates tax summaries, and ensures compliance with local and international tax regulations</p>					
Actors:	Financial Team					
Precondition:	<ol style="list-style-type: none"> 1. Tax rates and rules are configured in the ERP system according to applicable regulations. 2. All financial transactions (e.g., sales, purchases) are logged and up-to-date in the system. 3. The finance team has valid access to the tax calculation feature in the Financial Management module 					
Postcondition:	<ol style="list-style-type: none"> 1. Taxes for all relevant financial transactions are calculated and stored in the system. 2. A detailed tax report is generated, showing taxable amounts, tax rates applied, and total tax liabilities. 3. The report is ready for export or submission to tax authorities 					
Flow of Activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Automate Tax Calculations" feature. 3. Specifies the tax calculation parameters (e.g., date range, transaction type, jurisdiction). </td> <td> <ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the Financial Management module. 2. Retrieves financial transaction data from the centralized database. 3. Applies tax rules and rates based </td> </tr> </tbody> </table>	Actor	System	<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Automate Tax Calculations" feature. 3. Specifies the tax calculation parameters (e.g., date range, transaction type, jurisdiction). 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the Financial Management module. 2. Retrieves financial transaction data from the centralized database. 3. Applies tax rules and rates based 	
Actor	System					
<ol style="list-style-type: none"> 1. Logs into the ERP system and navigates to the Financial Management module. 2. Selects the "Automate Tax Calculations" feature. 3. Specifies the tax calculation parameters (e.g., date range, transaction type, jurisdiction). 	<ol style="list-style-type: none"> 1. Authenticates the actor's credentials and grants access to the Financial Management module. 2. Retrieves financial transaction data from the centralized database. 3. Applies tax rules and rates based 					

	<ol style="list-style-type: none"> 4. Submits the request to calculate taxes. 5. Reviews the generated tax summary and report for accuracy. 6. Exports or submits the tax report to relevant stakeholders or tax authorities 	<p>on the jurisdiction, transaction type, and other parameters.</p> <ol style="list-style-type: none"> 4. Calculates tax amounts for each transaction and aggregates totals. 5. Generates a detailed tax report, including tax amounts, rates, and summaries by category. 6. Displays the report to the actor and provides options for export or submission. 7. Logs the tax calculation and report generation activity for audit purposes
Exception Description:	<ol style="list-style-type: none"> 1. Incorrect Tax Configuration: If tax rules or rates are missing or outdated, the system flags the issue and prompts the actor to update the settings. 2. Data Discrepancies: If financial transactions contain errors or inconsistencies, the system alerts the actor for resolution. 3. Access Denied: The actor lacks sufficient privileges to perform tax calculations for specific accounts or jurisdictions 	

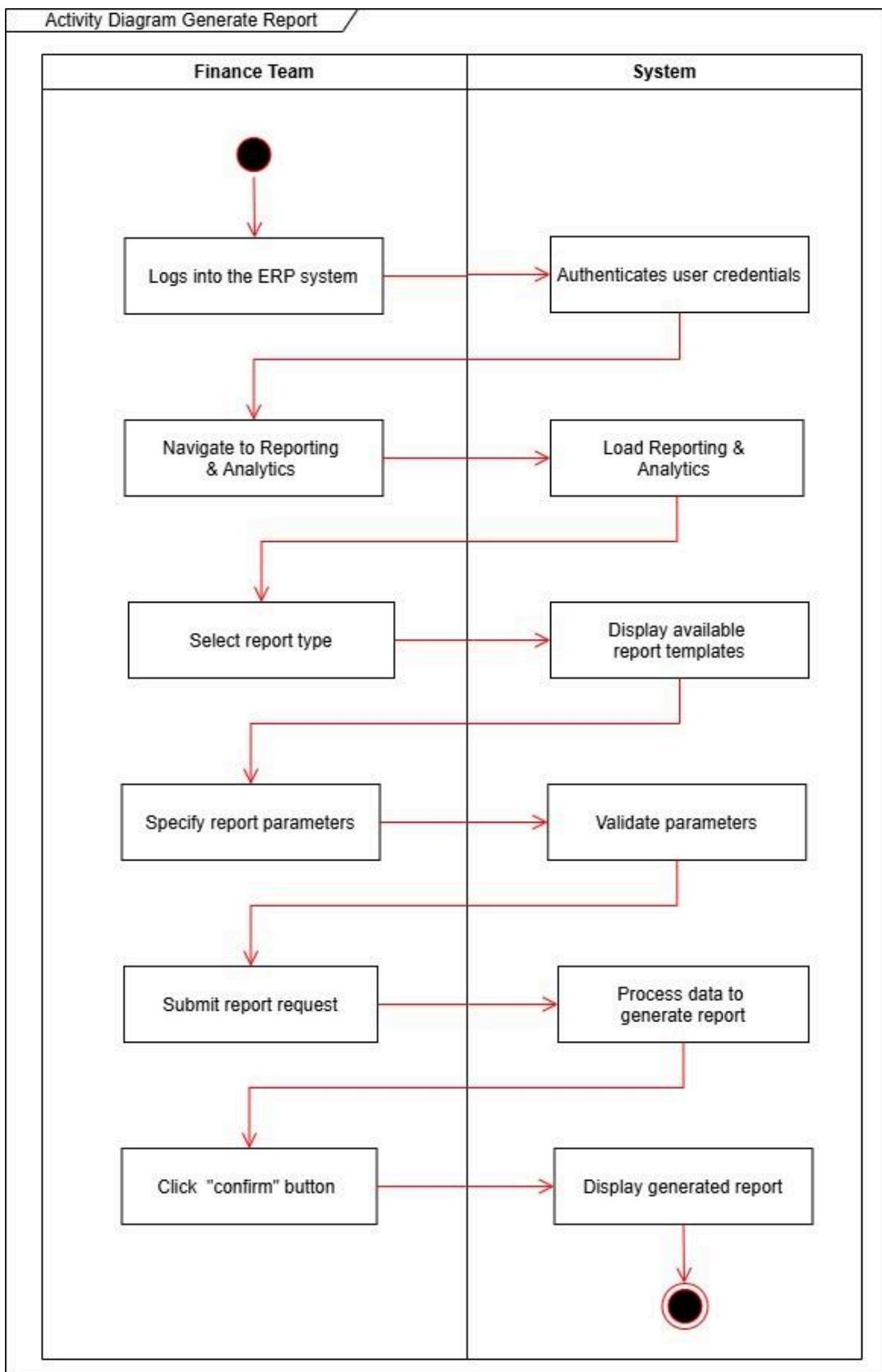
4.7.4 Activity Diagram

4.7.4.1 Reporting and Analytics Subsystem

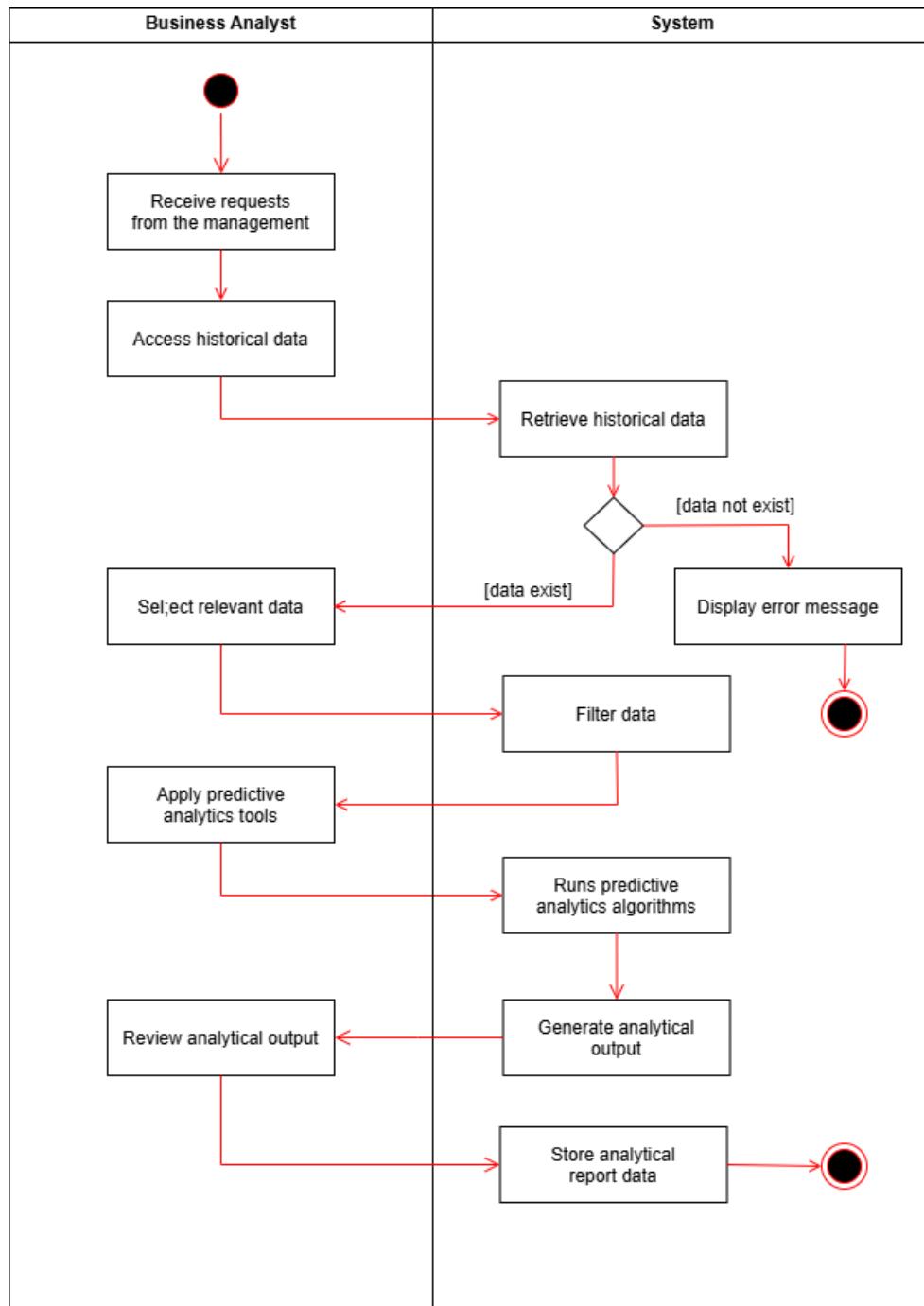
4.7.4.1.1 View Real-Time Dashboards



4.7.4.1.2 Generate Reports

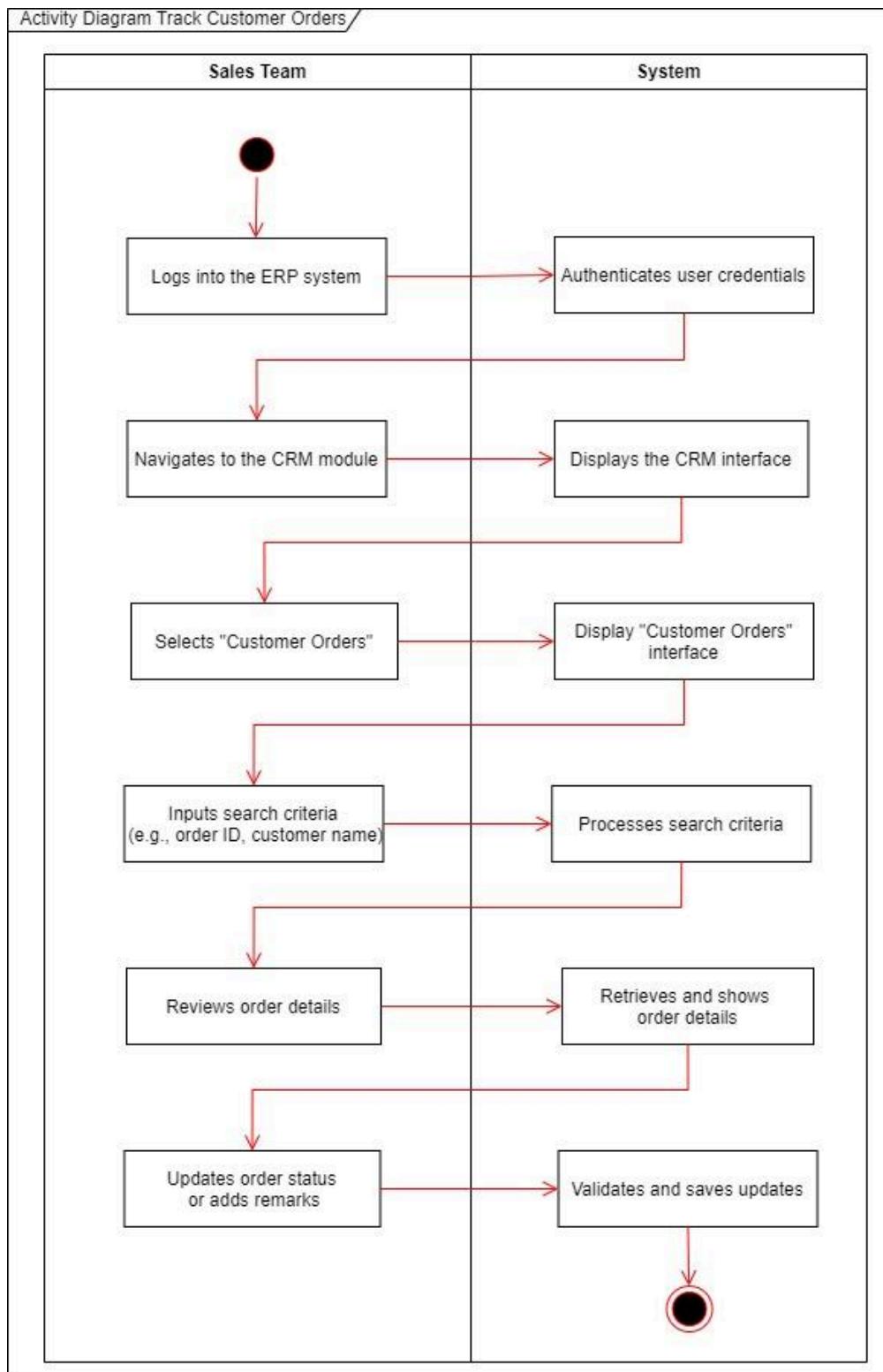


4.7.4.1.3 Analyze Trends with Predictive Analytics

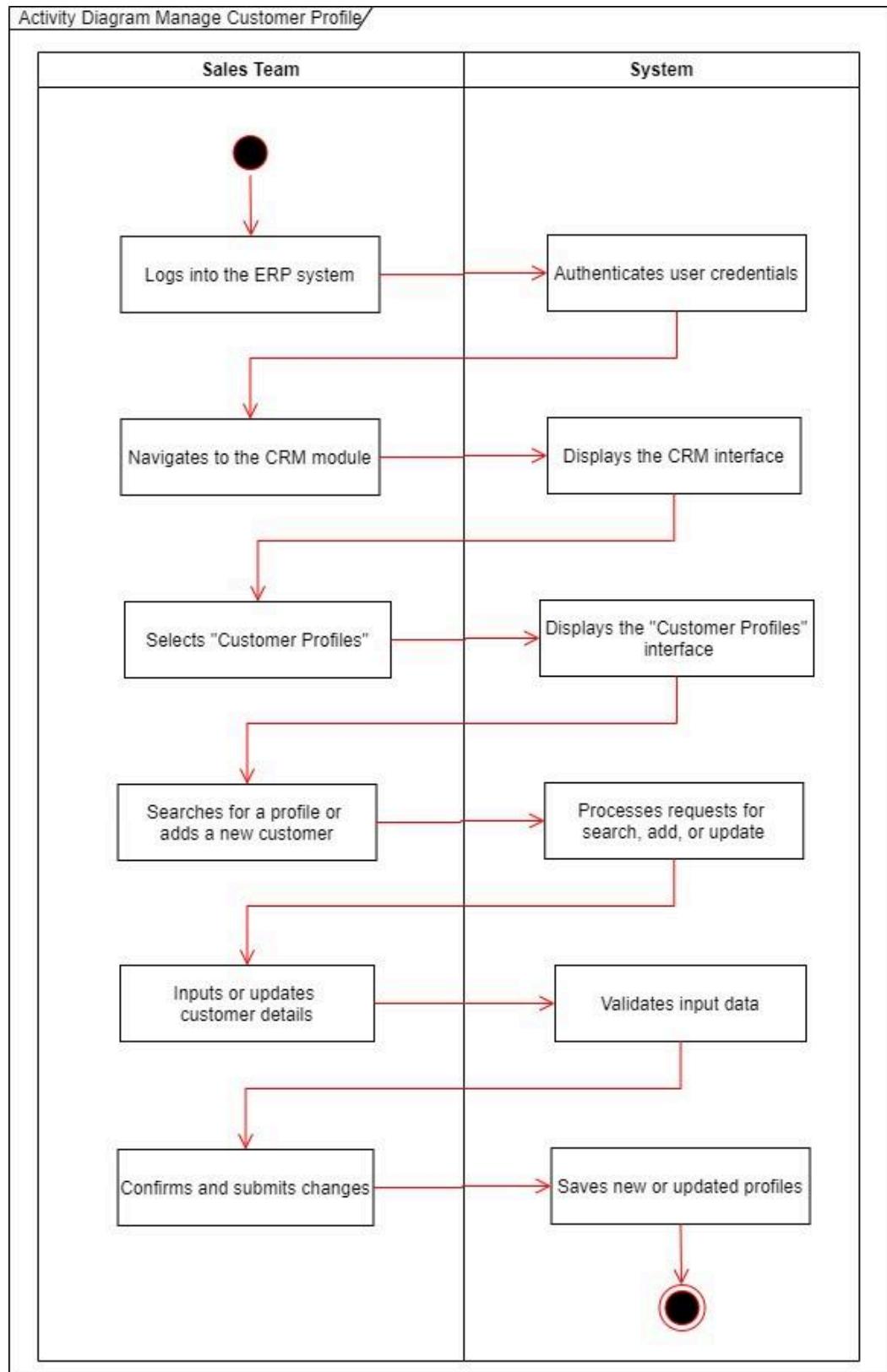


4.7.4.2 Sales and Customer Management (CRM) Subsystem

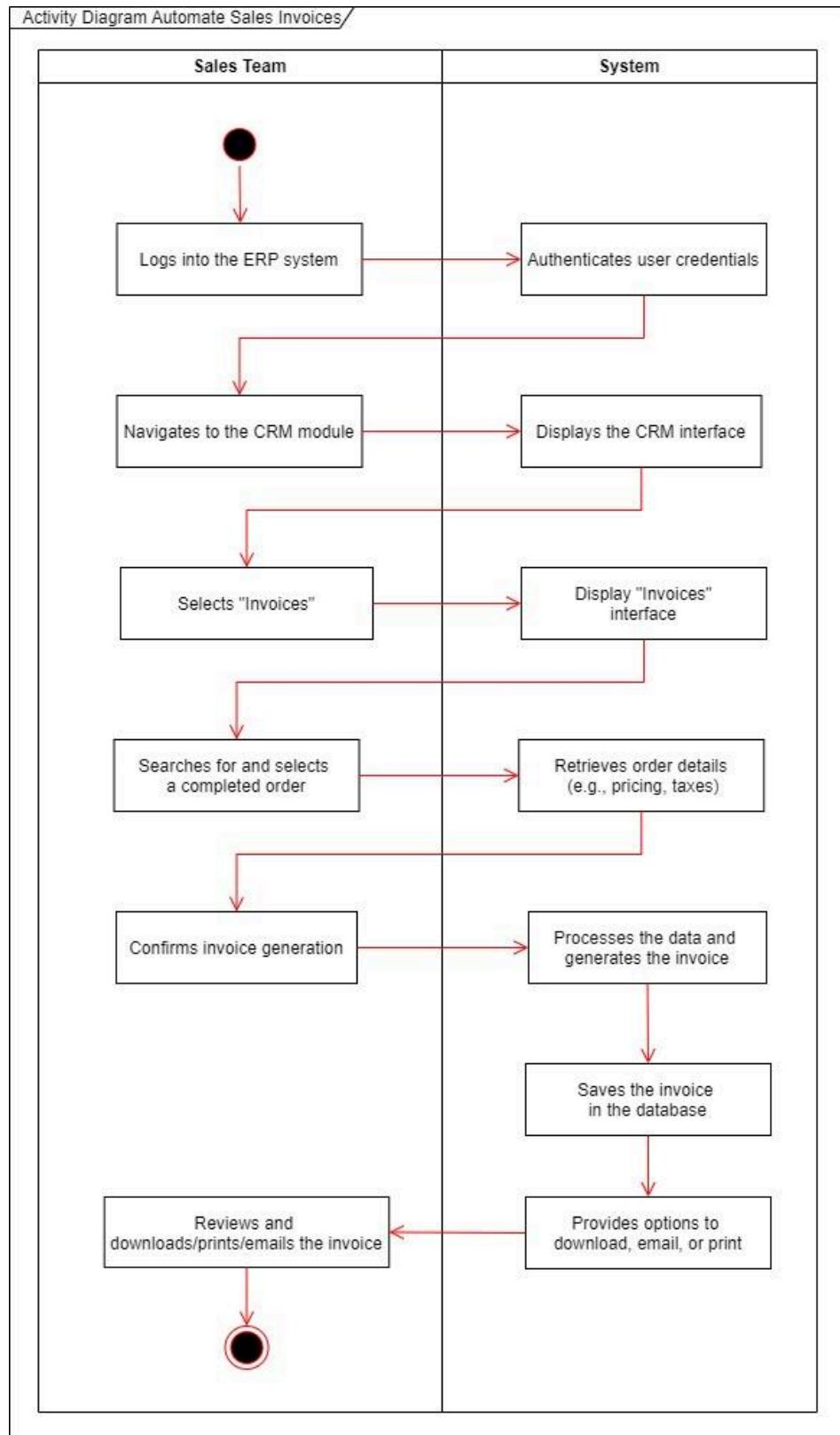
4.7.4.2.1 Track Customer Orders



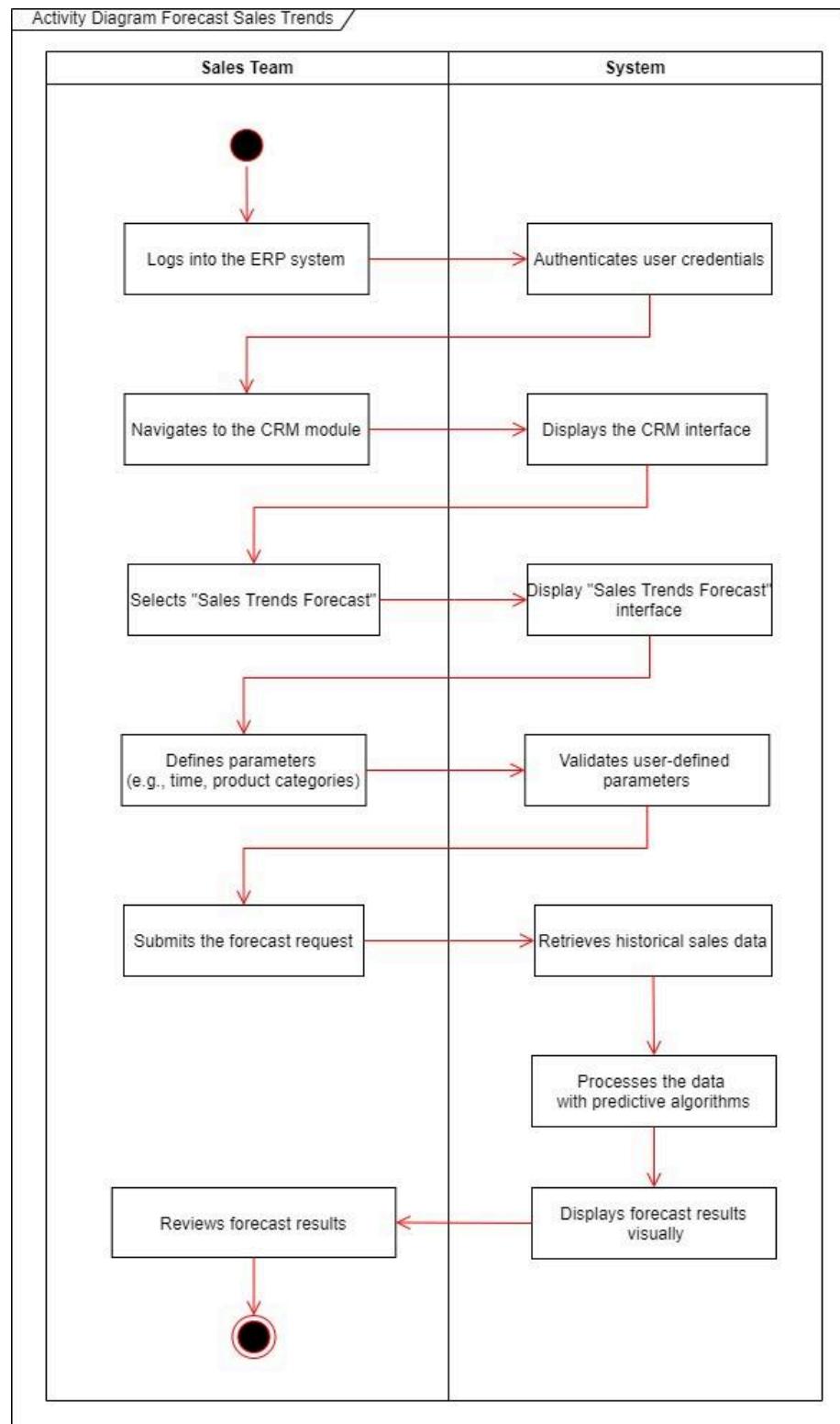
4.7.4.2.2 Manage Customer Profiles



4.7.4.2.3 Automate Sales Invoices

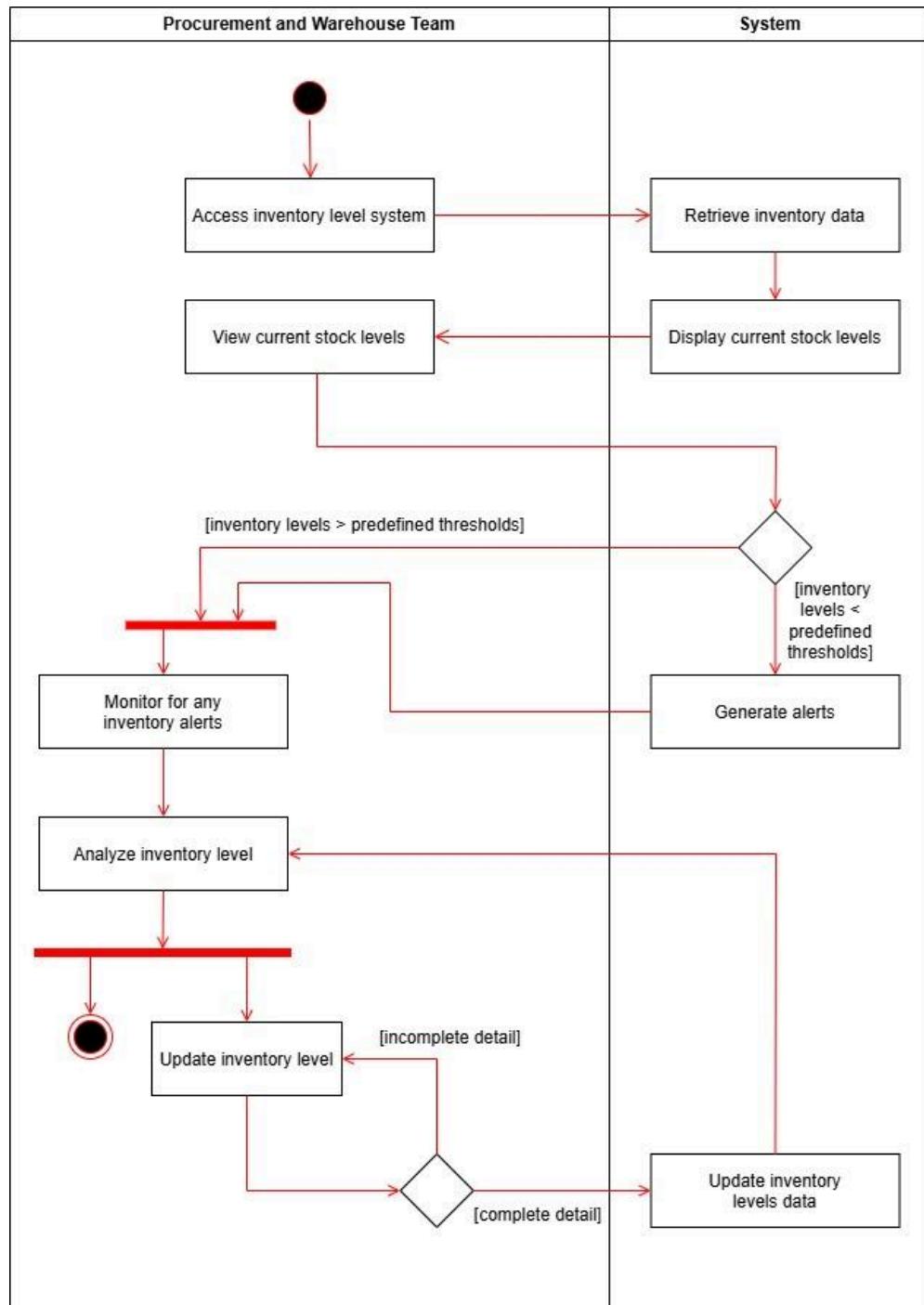


4.7.4.2.4 Forecast Sales Trends

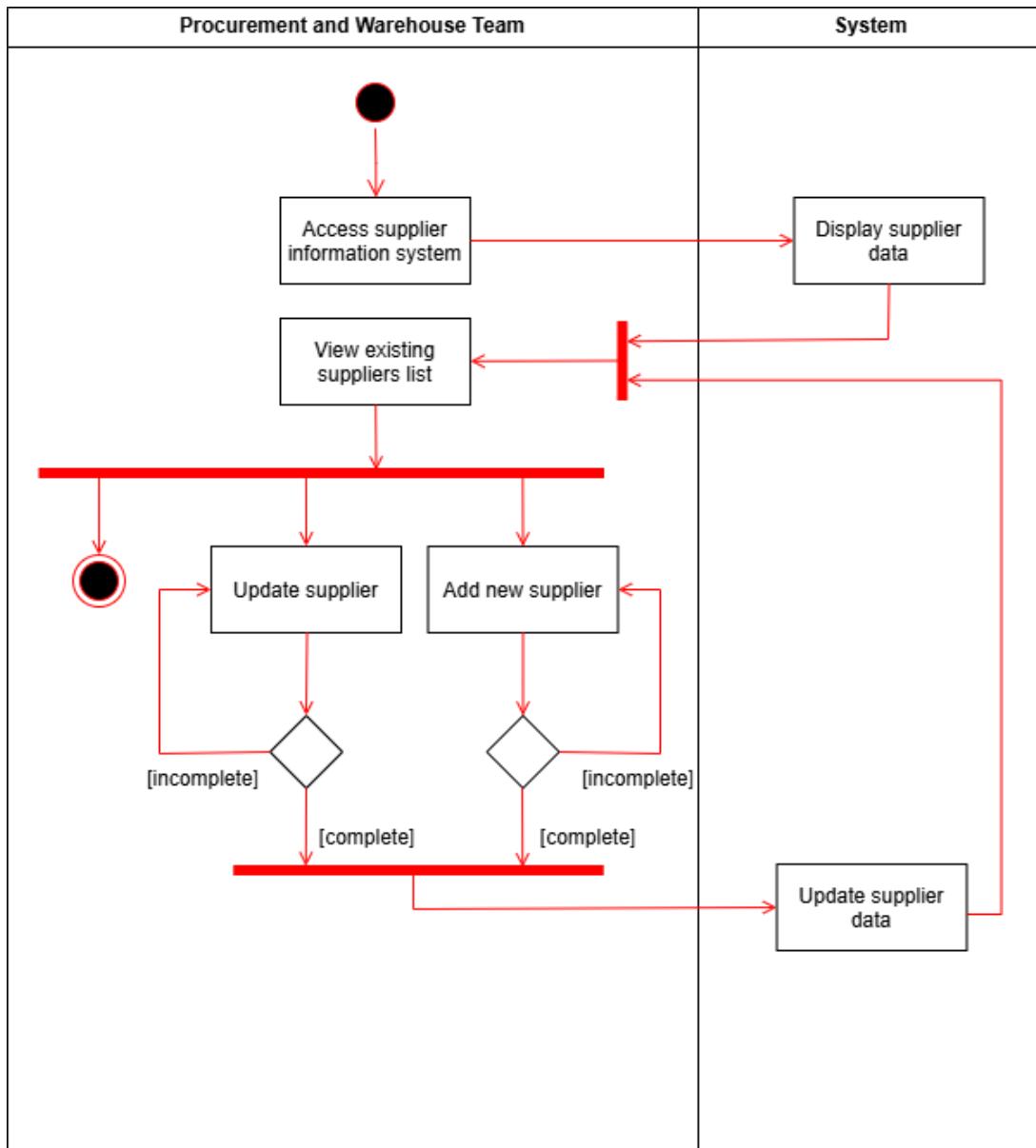


4.7.4.3 Procurement and Inventory Management (SCM) Subsystem

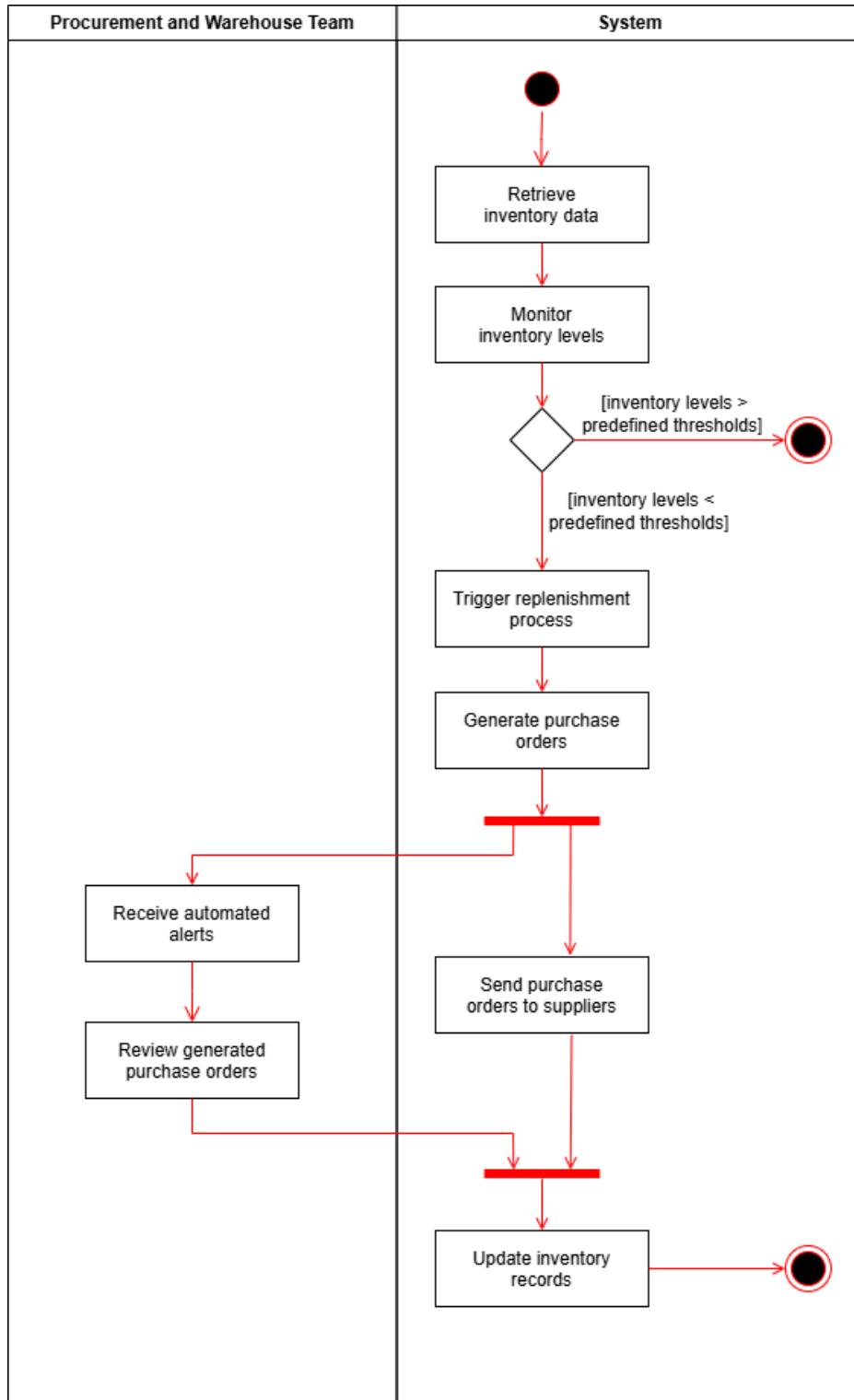
4.7.4.3.1 Track Inventory Levels



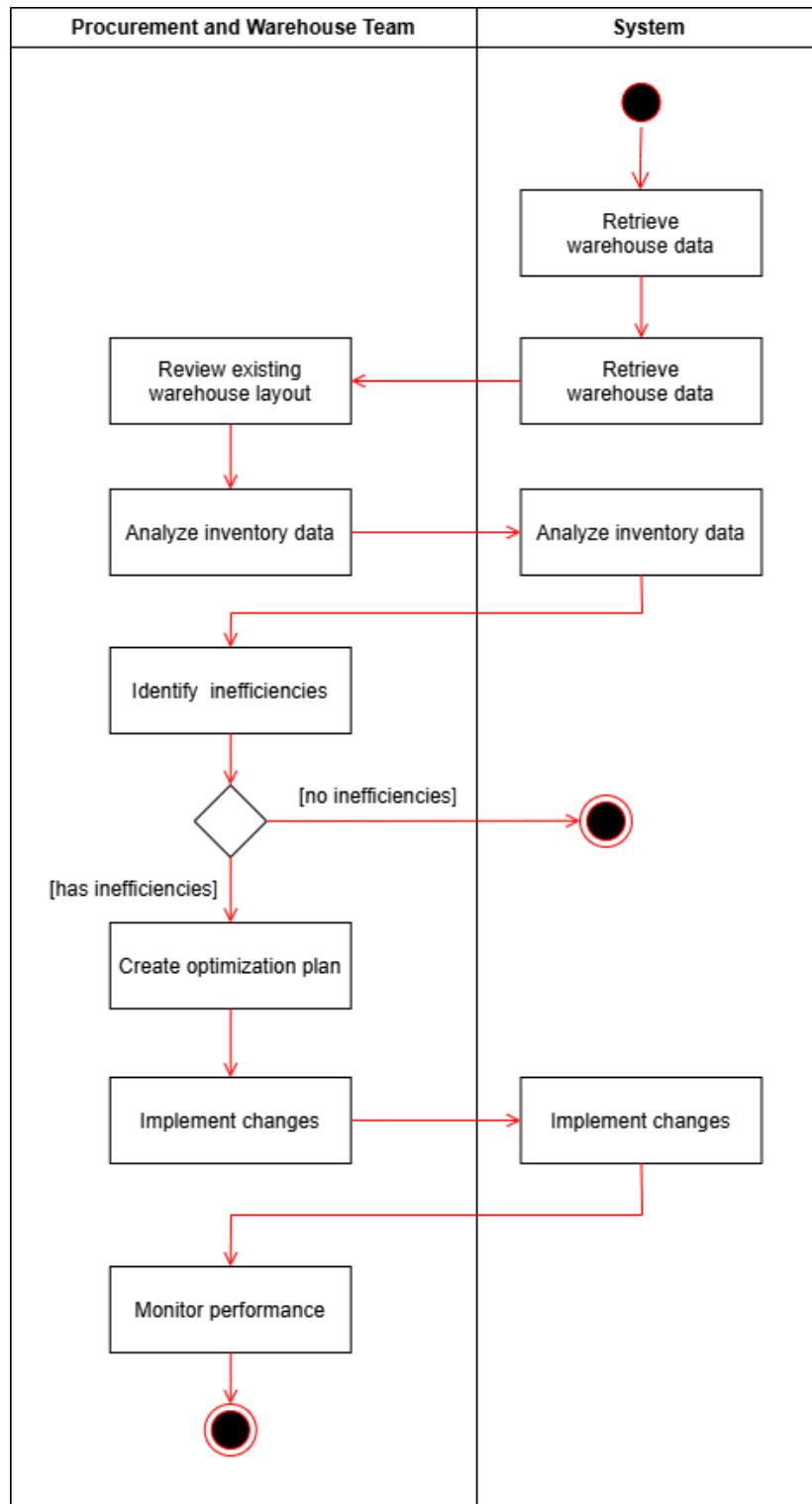
4.7.4.3.2 Manage Supplier Information



4.7.4.3.3 Automate Stock Replenishment

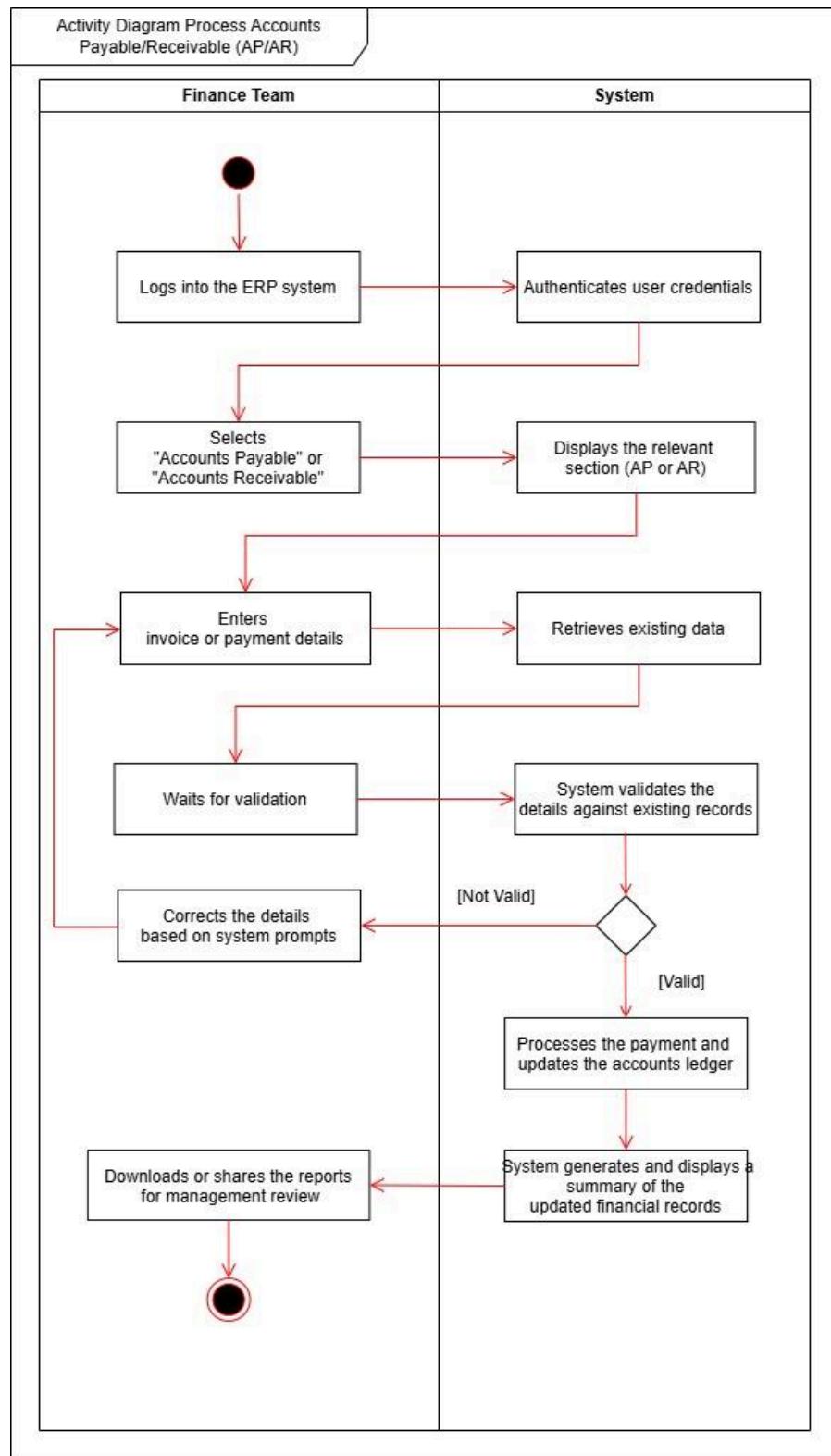


4.7.4.3.4 Optimize Warehouse Locations

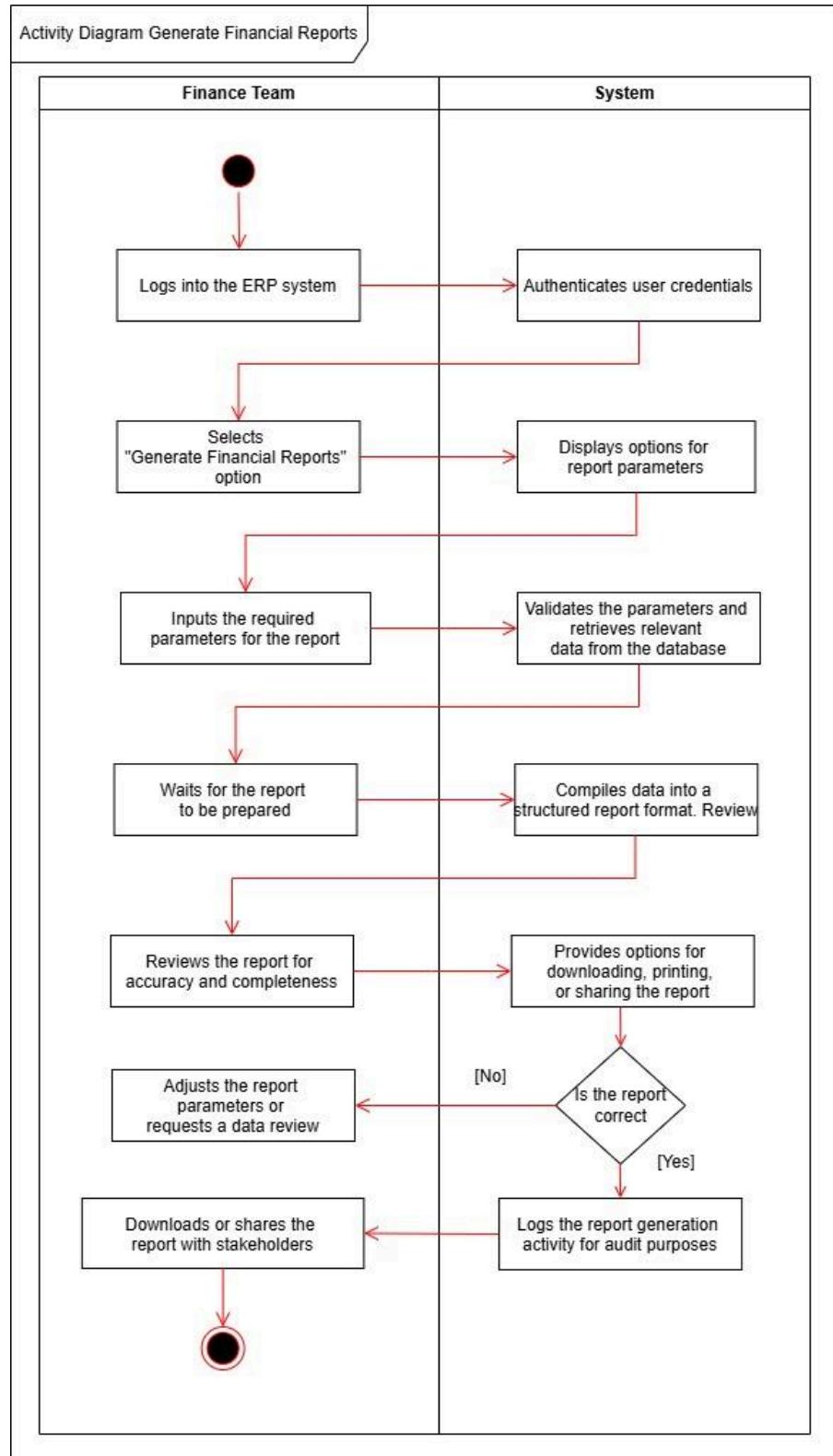


4.7.4.4 Financial Management Subsystem

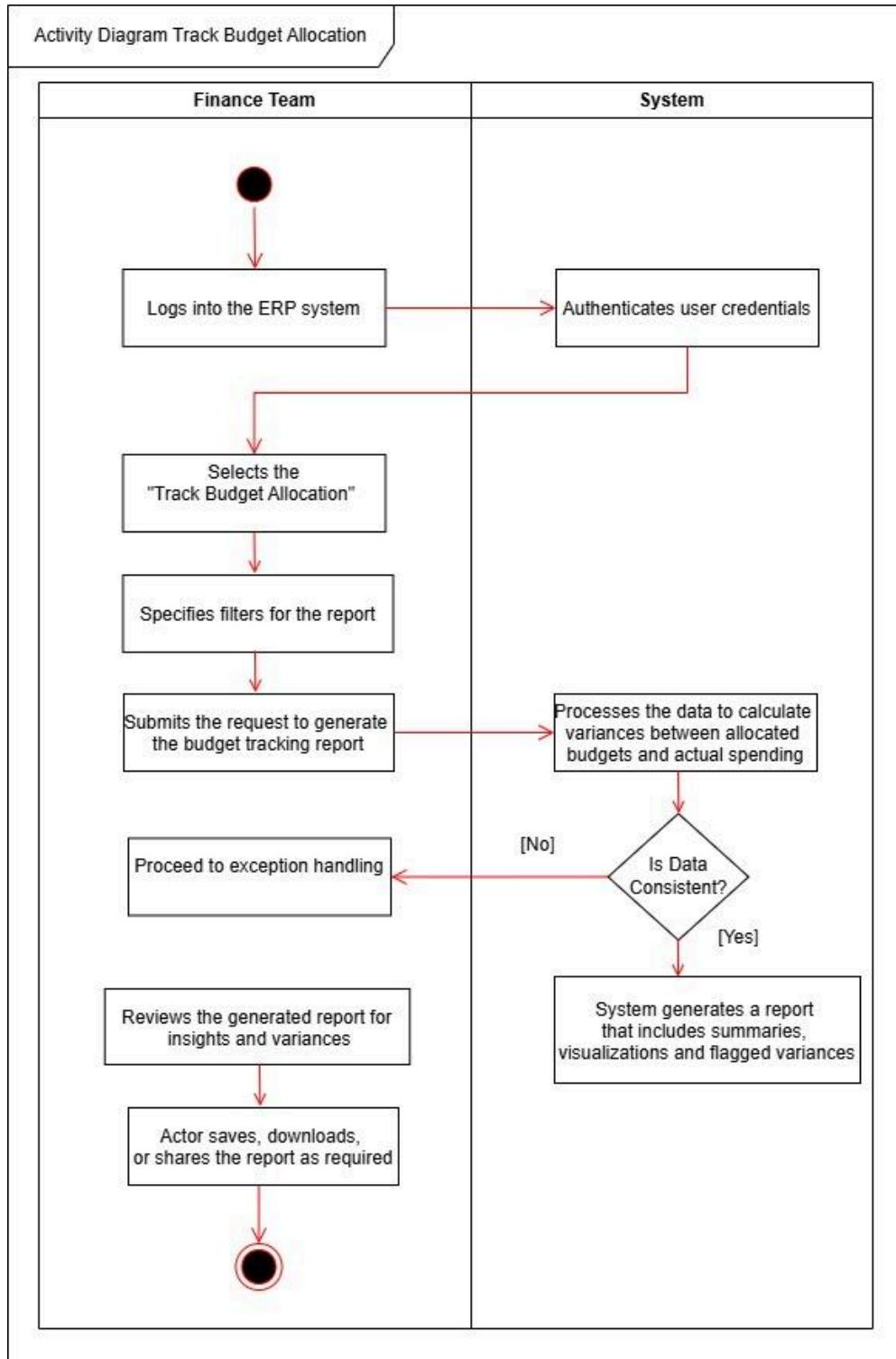
4.7.4.4.1 Process Accounts Payable/Receivable (AP/AR)



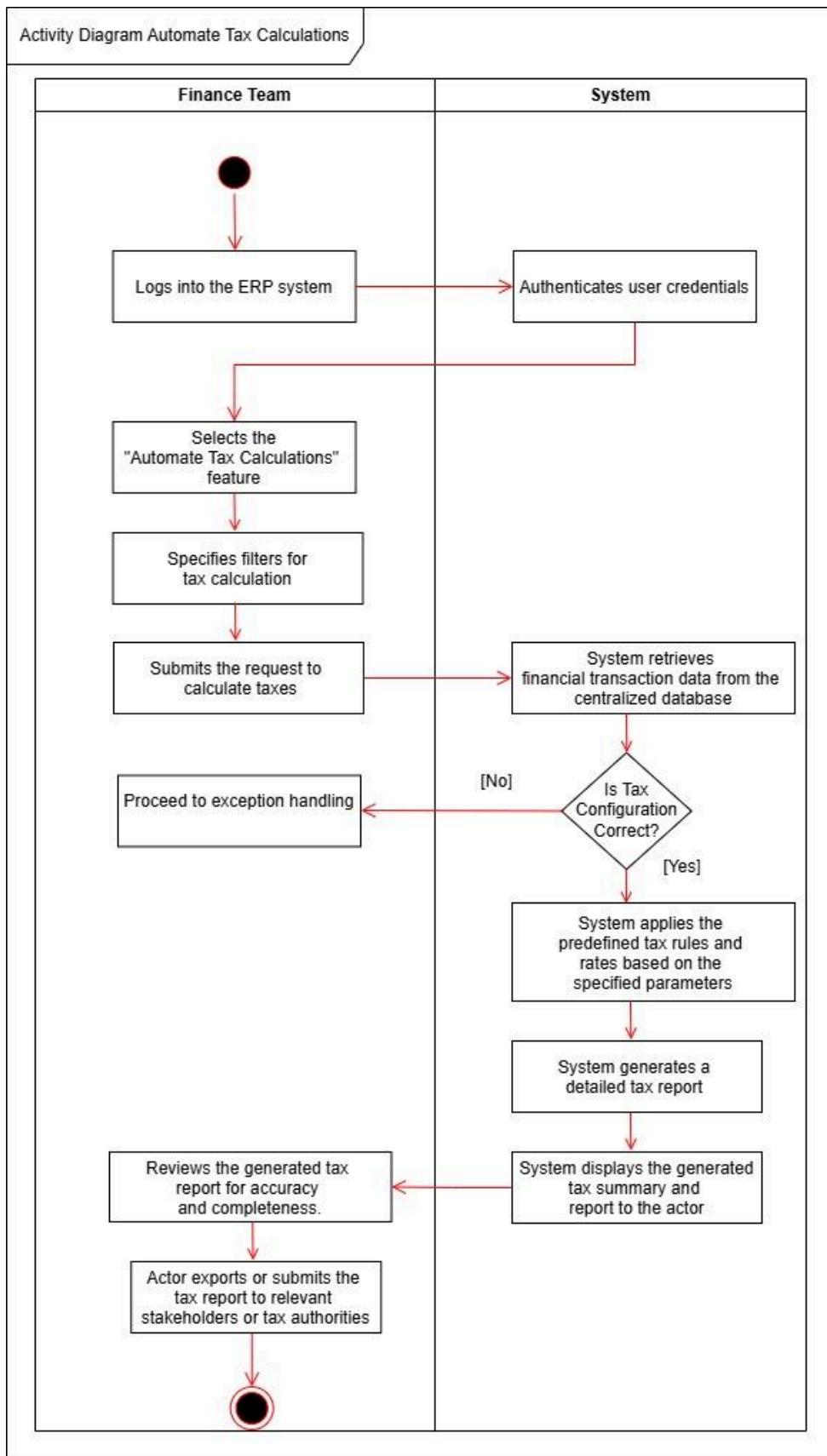
4.7.4.4.2 Generate Financial Reports



4.7.4.4.3 Track Budget Allocation



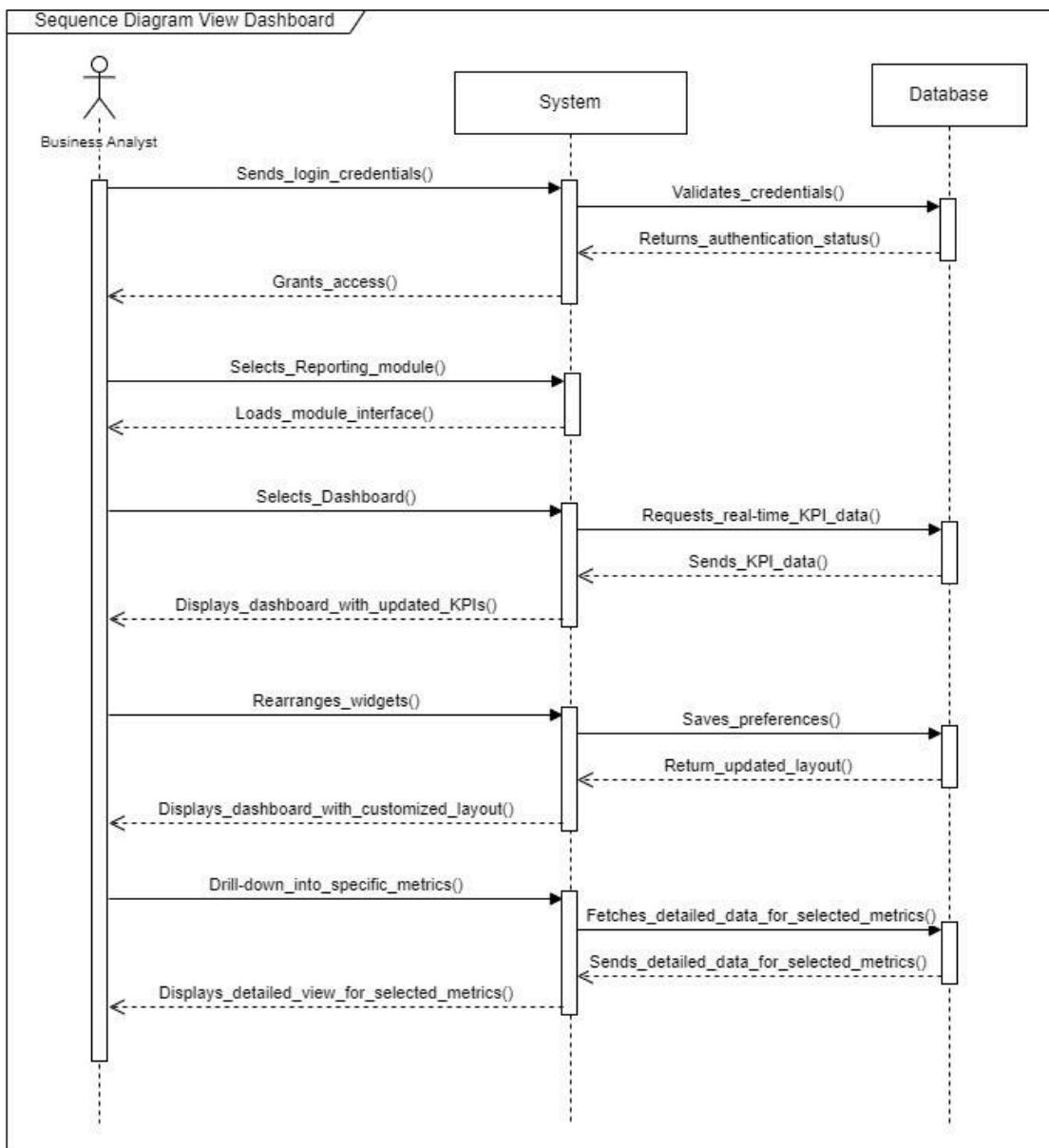
4.7.4.4 Automate Tax Calculation



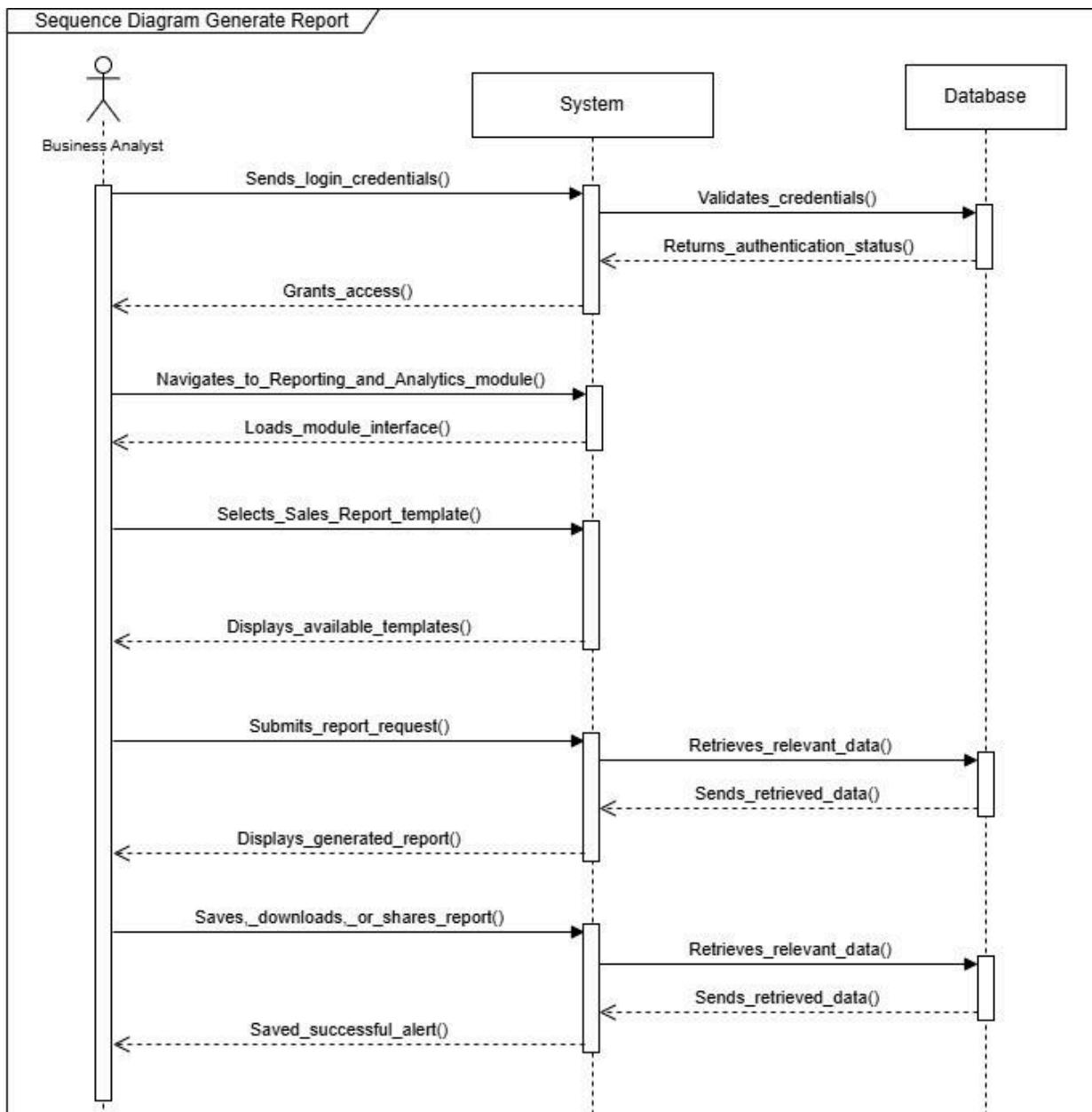
4.7.5 Sequence Diagram

4.7.5.1 Reporting and Analytics Subsystem

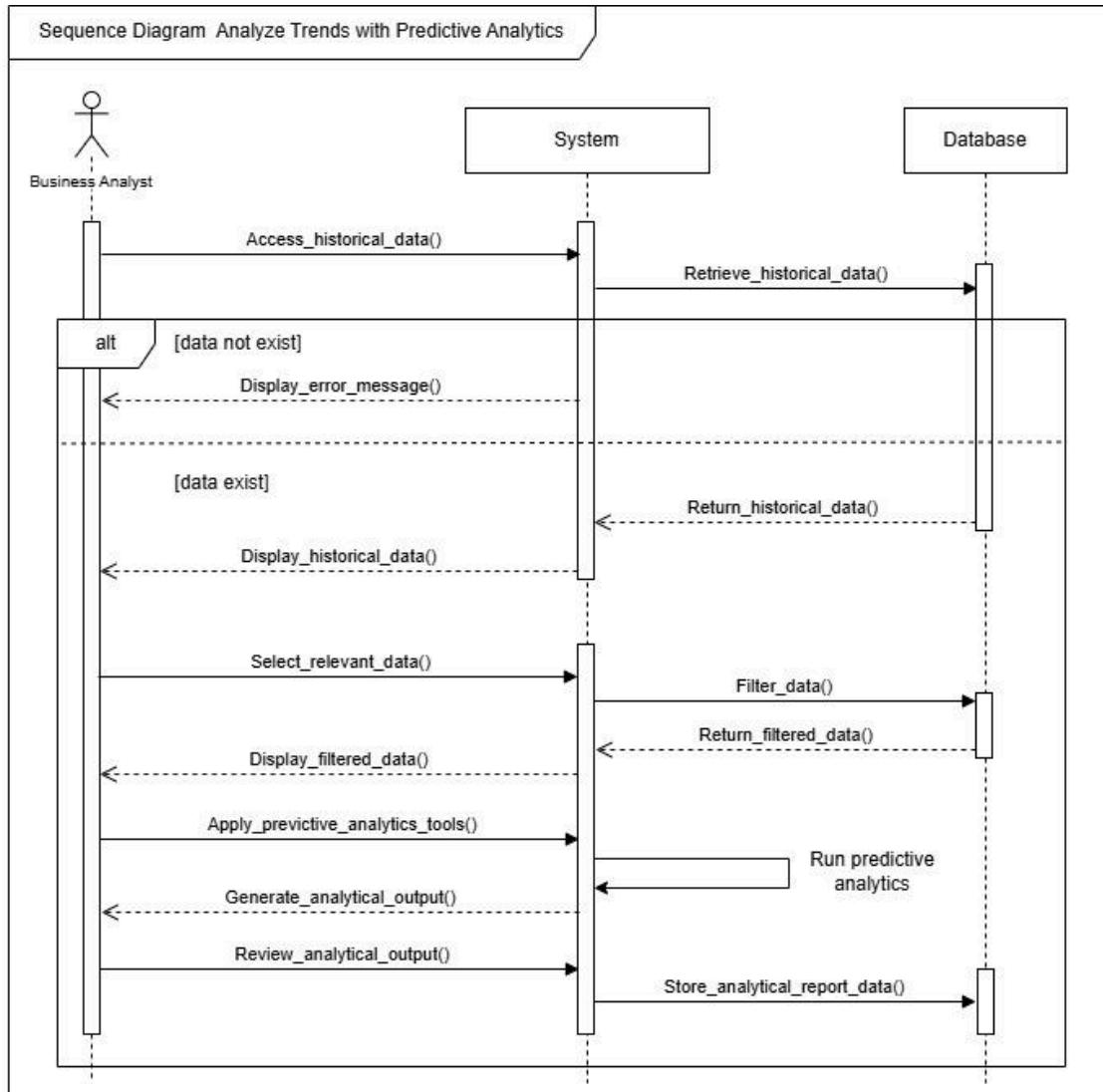
4.7.5.1.1 View Real-Time Dashboards



4.7.5.1.2 Generate Reports

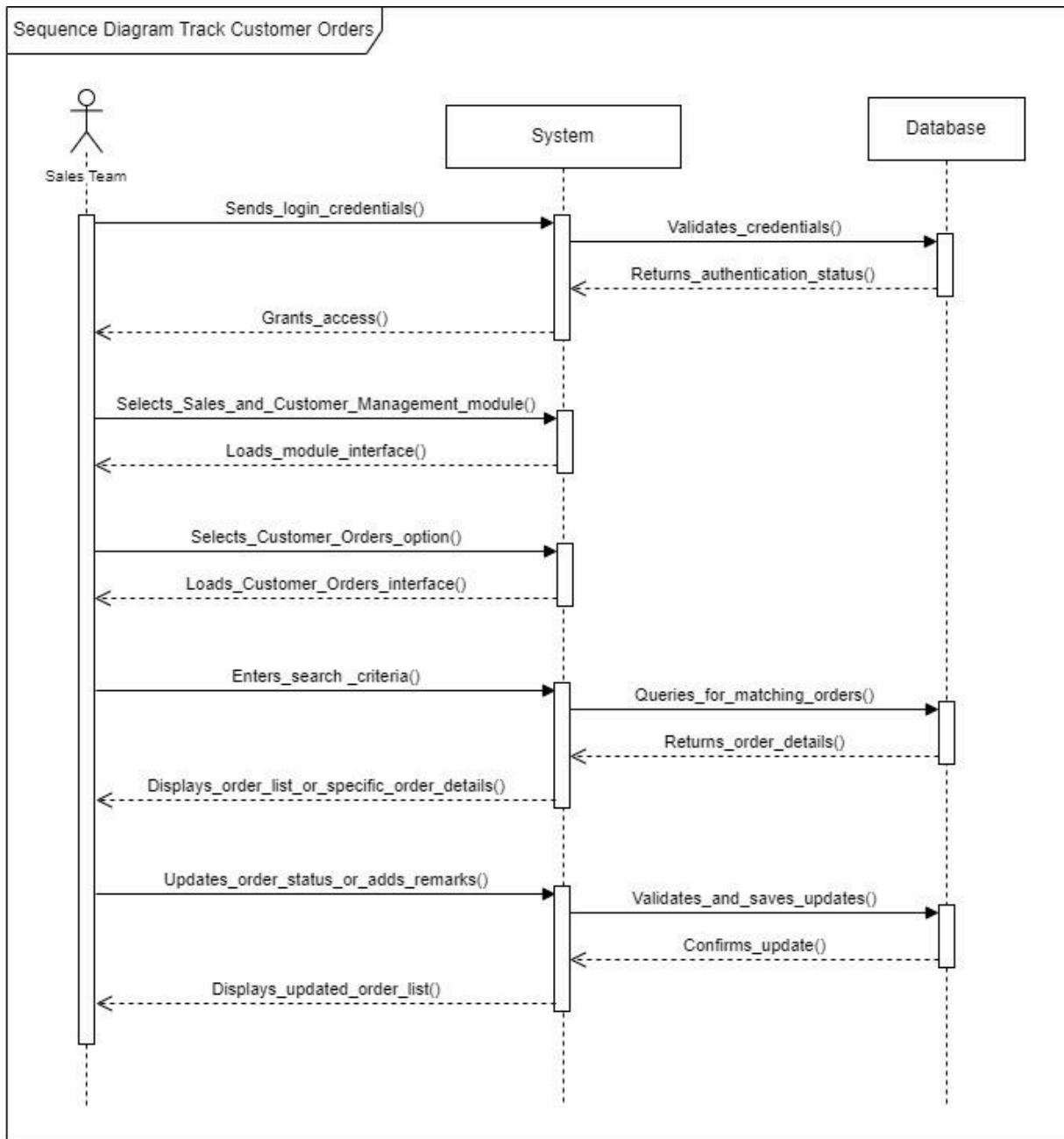


4.7.5.1.3 Analyze Trends with Predictive Analytics

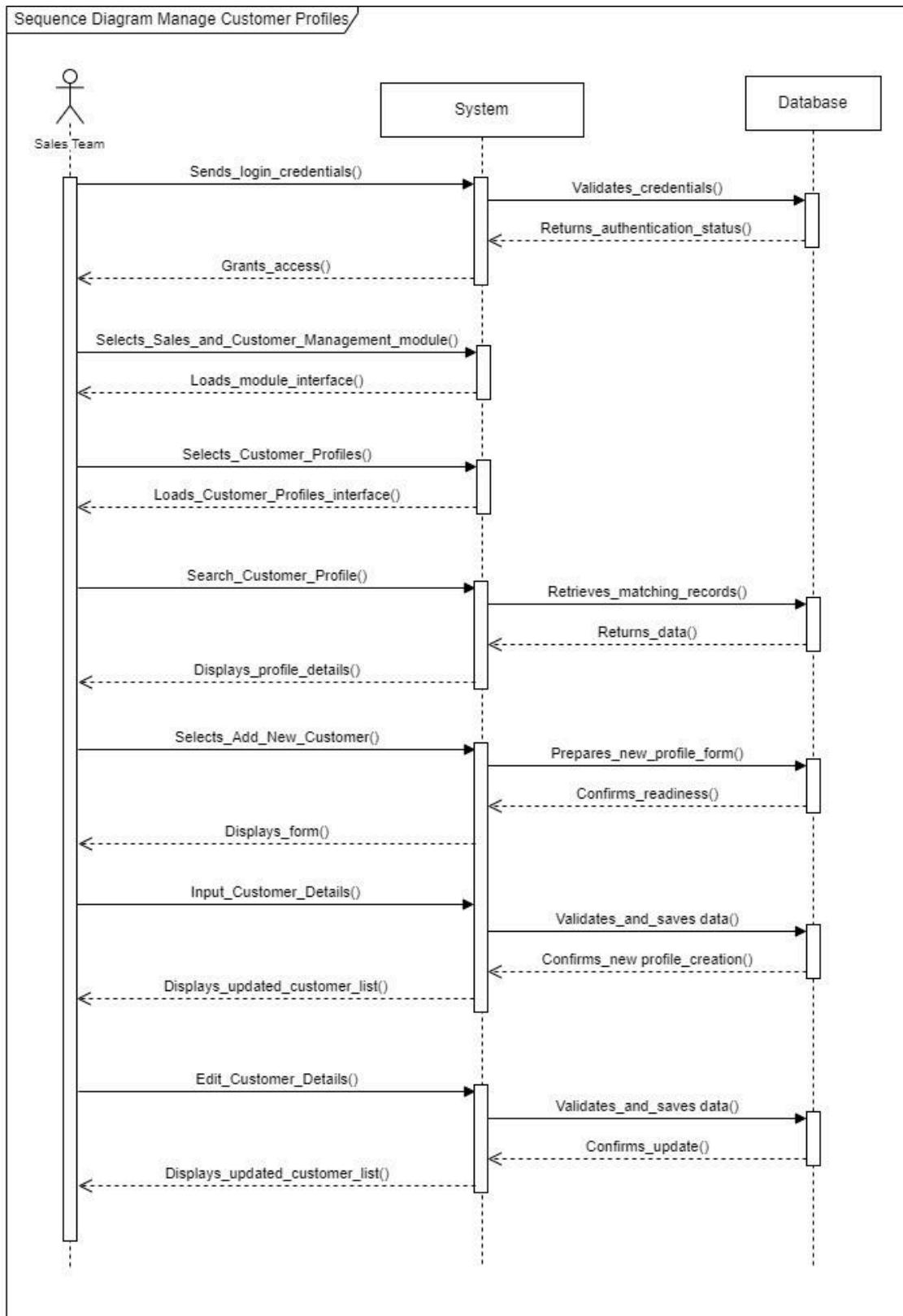


4.7.5.2 Sales and Customer Management (CRM) Subsystem

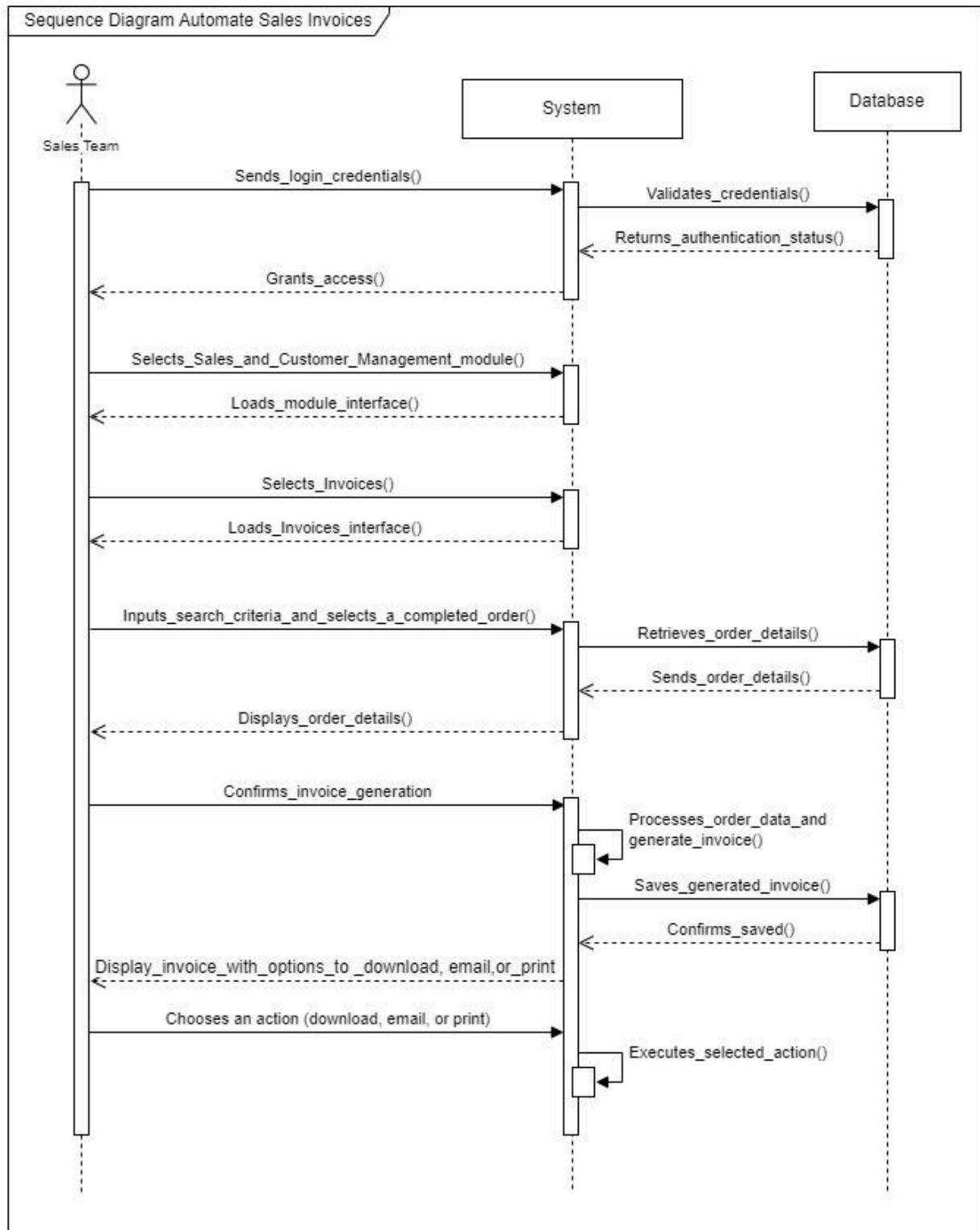
4.7.5.2.1 Track Customer Orders



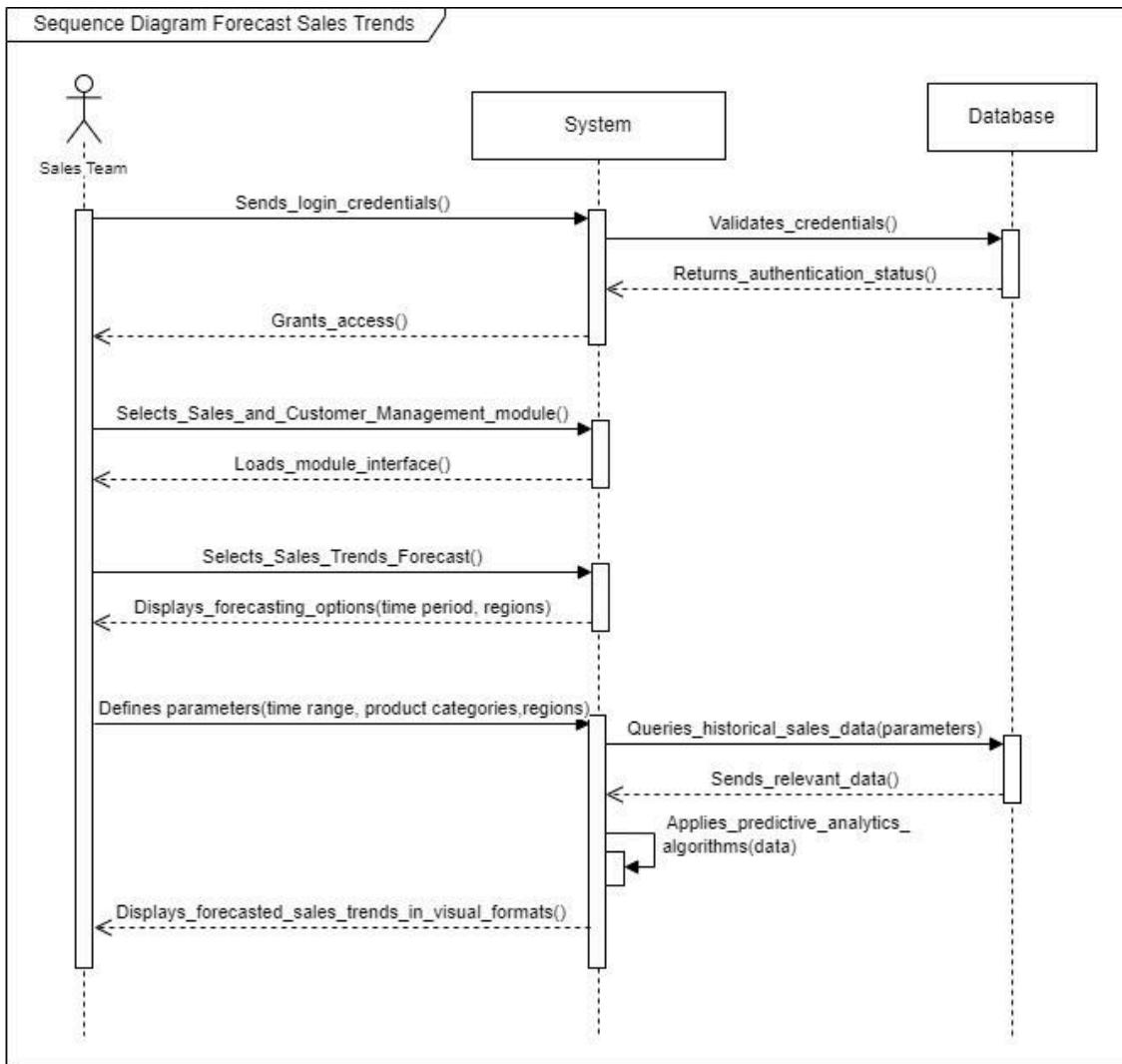
4.7.5.2.2 Manage Customer Profiles



4.7.5.2.3 Automate Sales Invoices

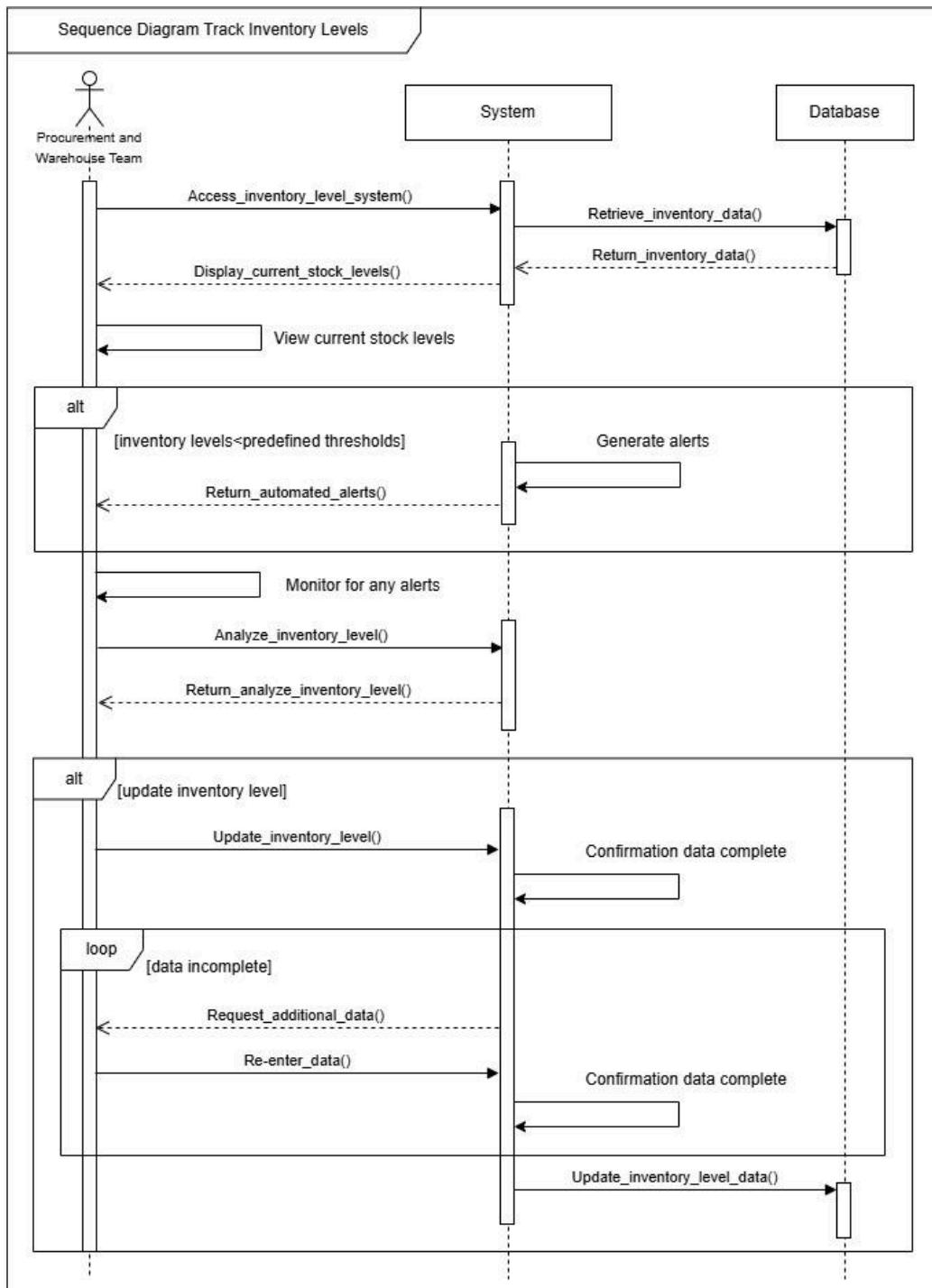


4.7.5.2.4 Forecast Sales Trends

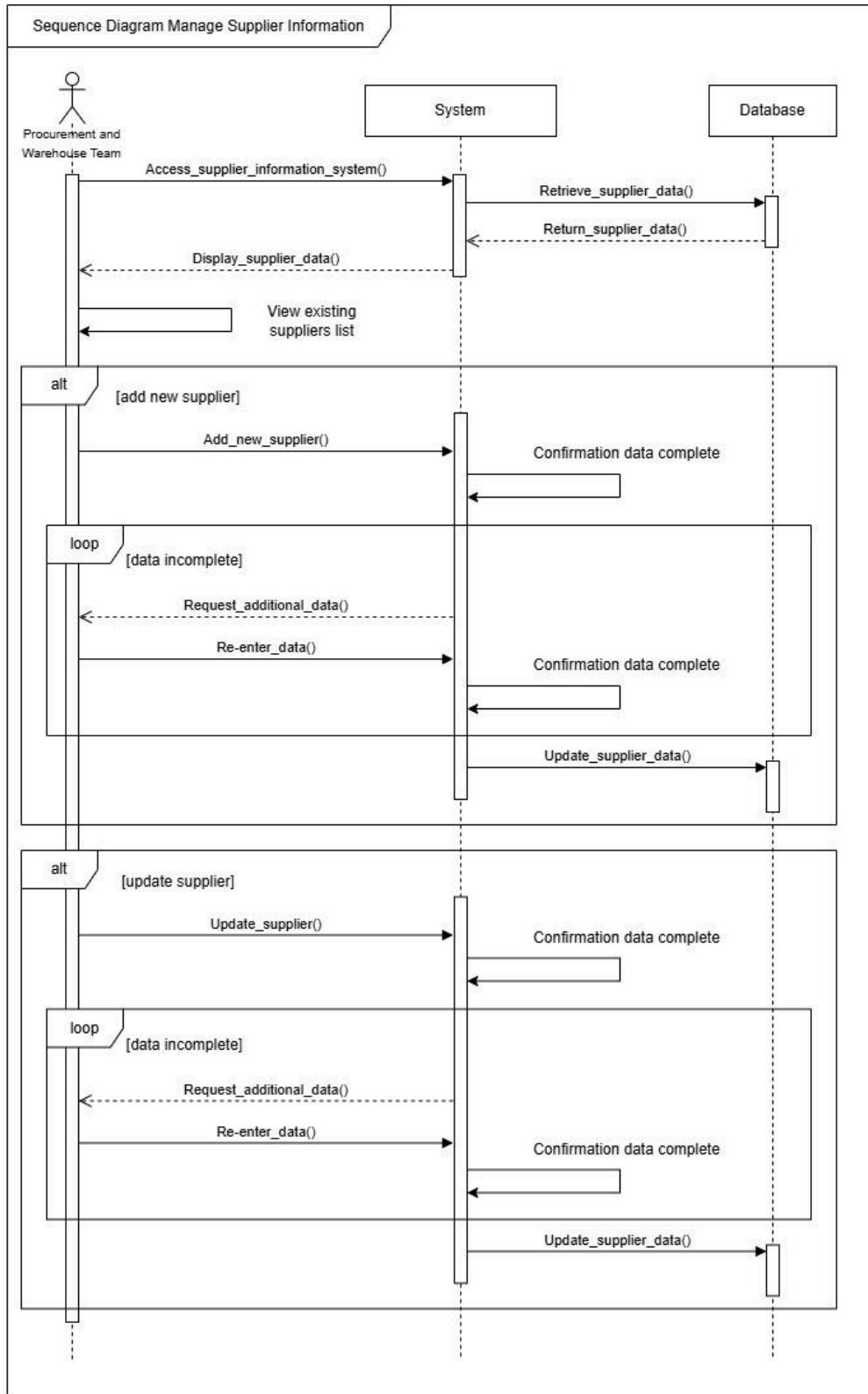


4.7.5.3 Procurement and Inventory Management (SCM) Subsystem

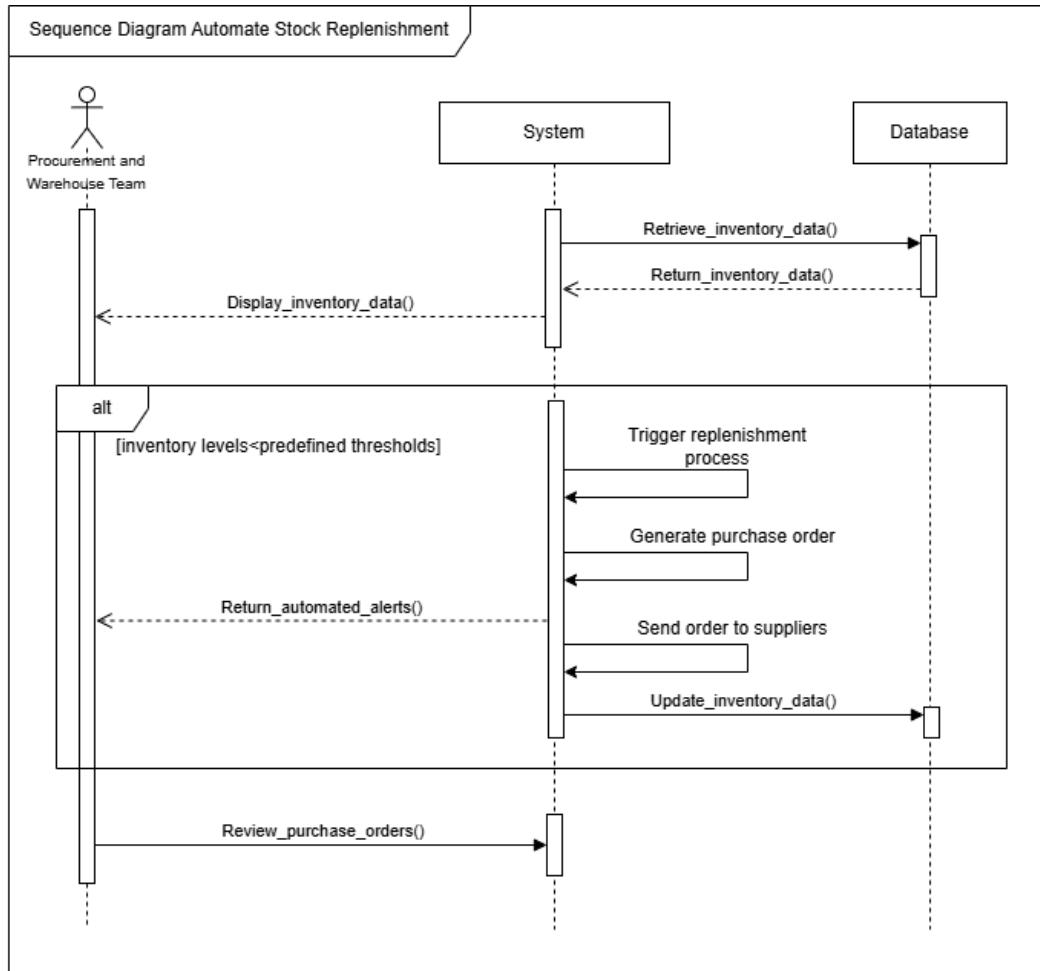
4.7.5.3.1 Track Inventory Levels



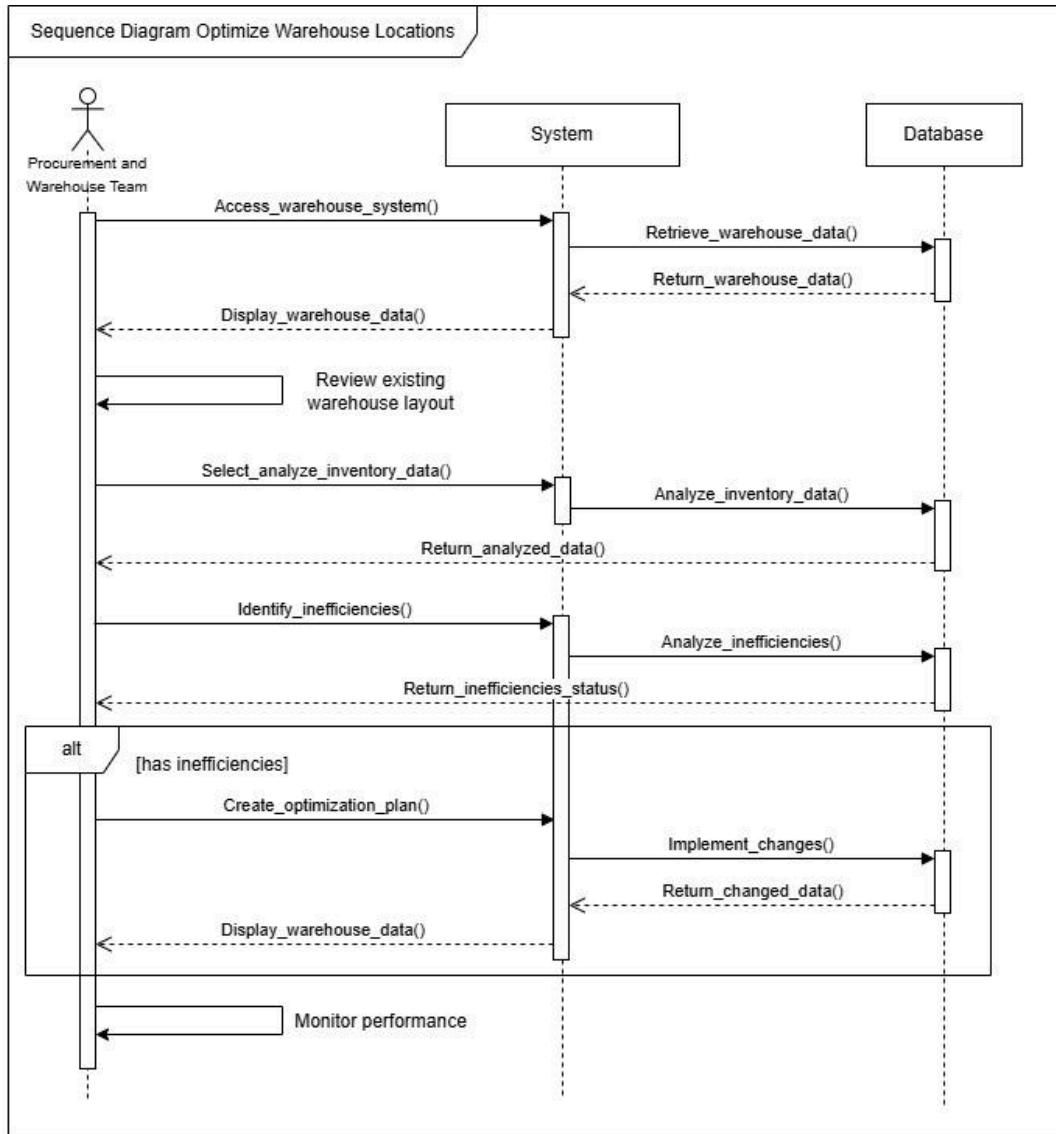
4.7.5.3.2 Manage Supplier Information



4.7.5.3.3 Automate Stock Replenishment

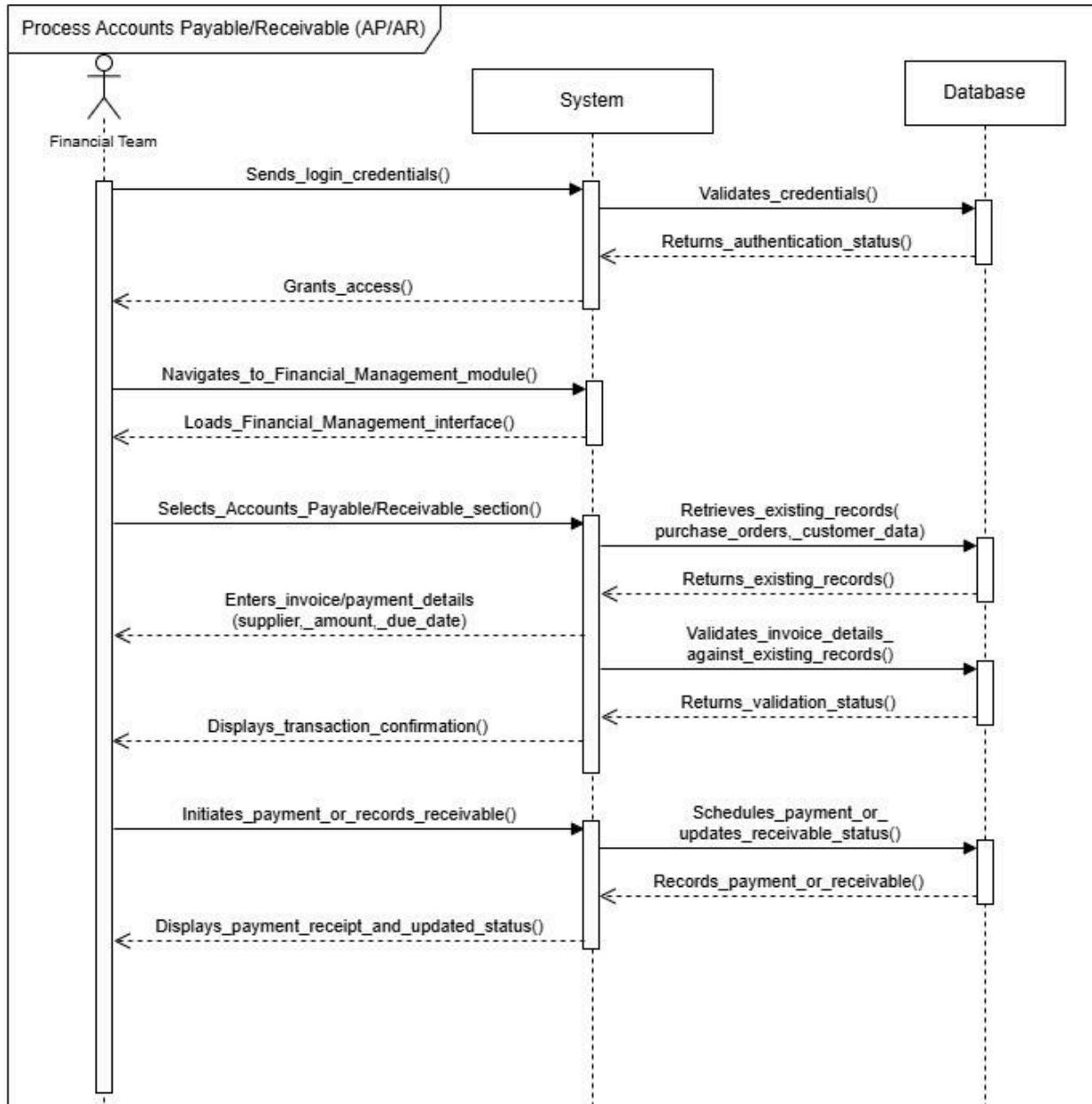


4.7.5.3.4 Optimize Warehouse Locations

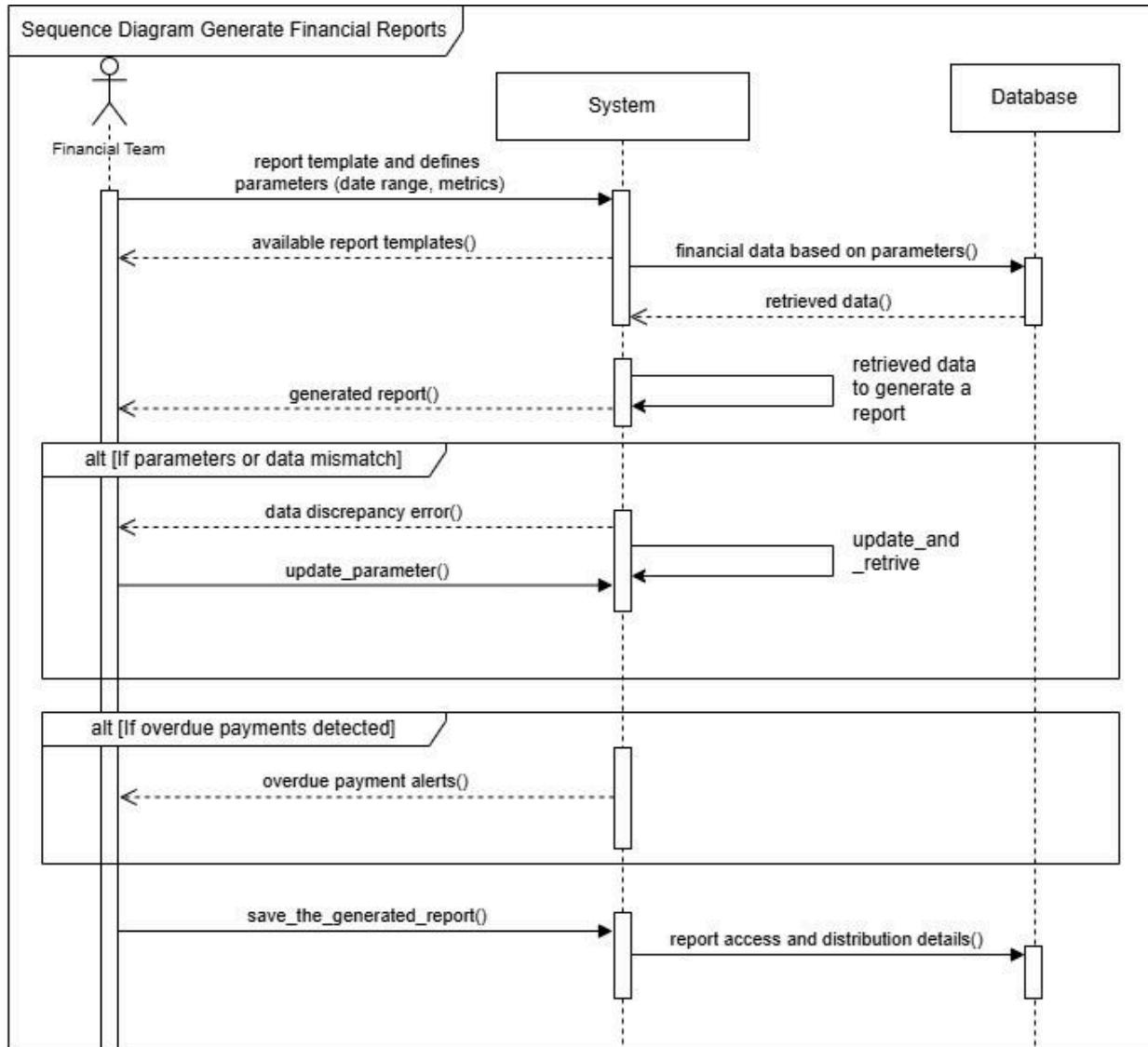


4.7.5.4 Financial Management Subsystem

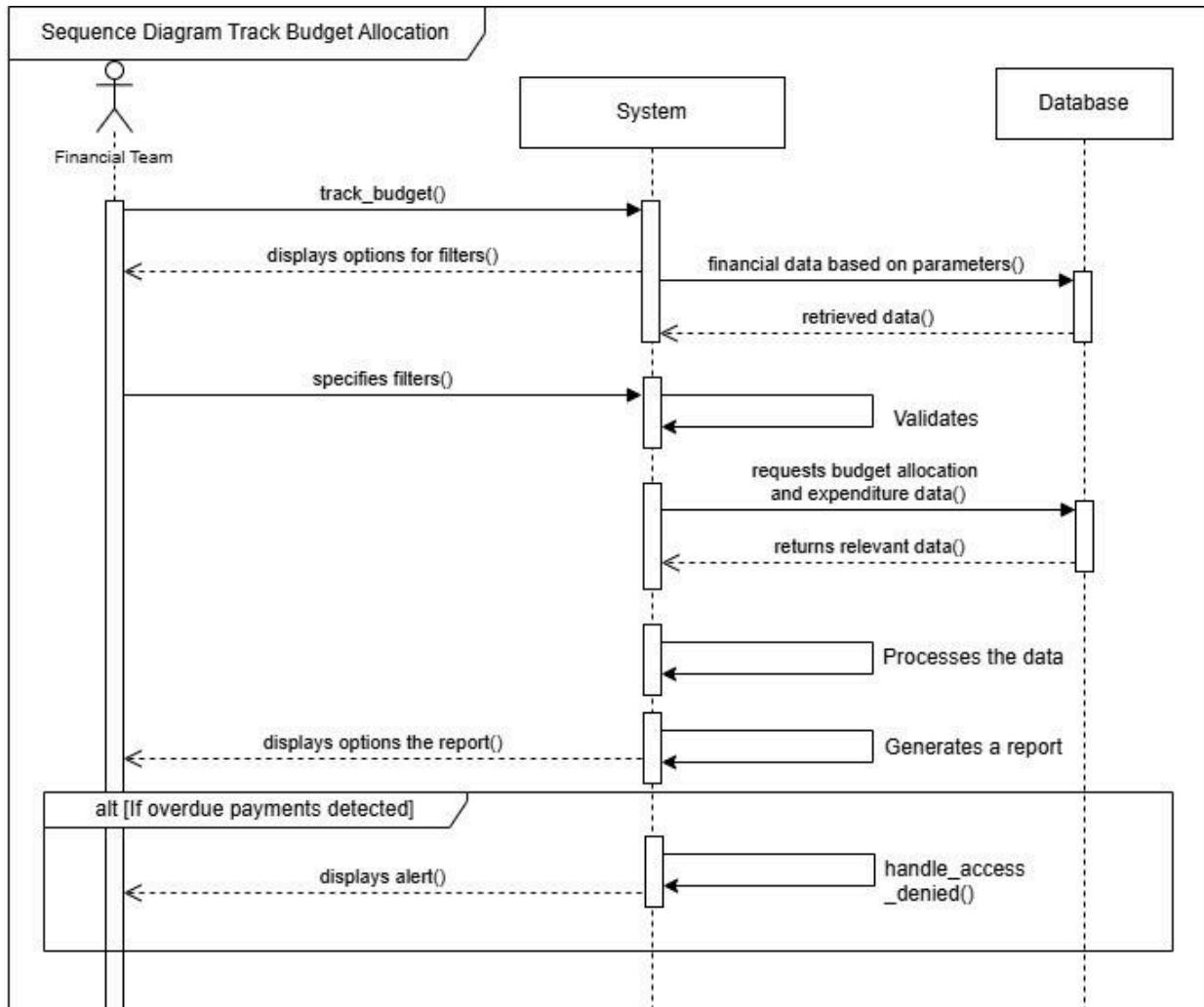
4.7.5.4.1 Process Accounts Payable/Receivable (AP/AR)



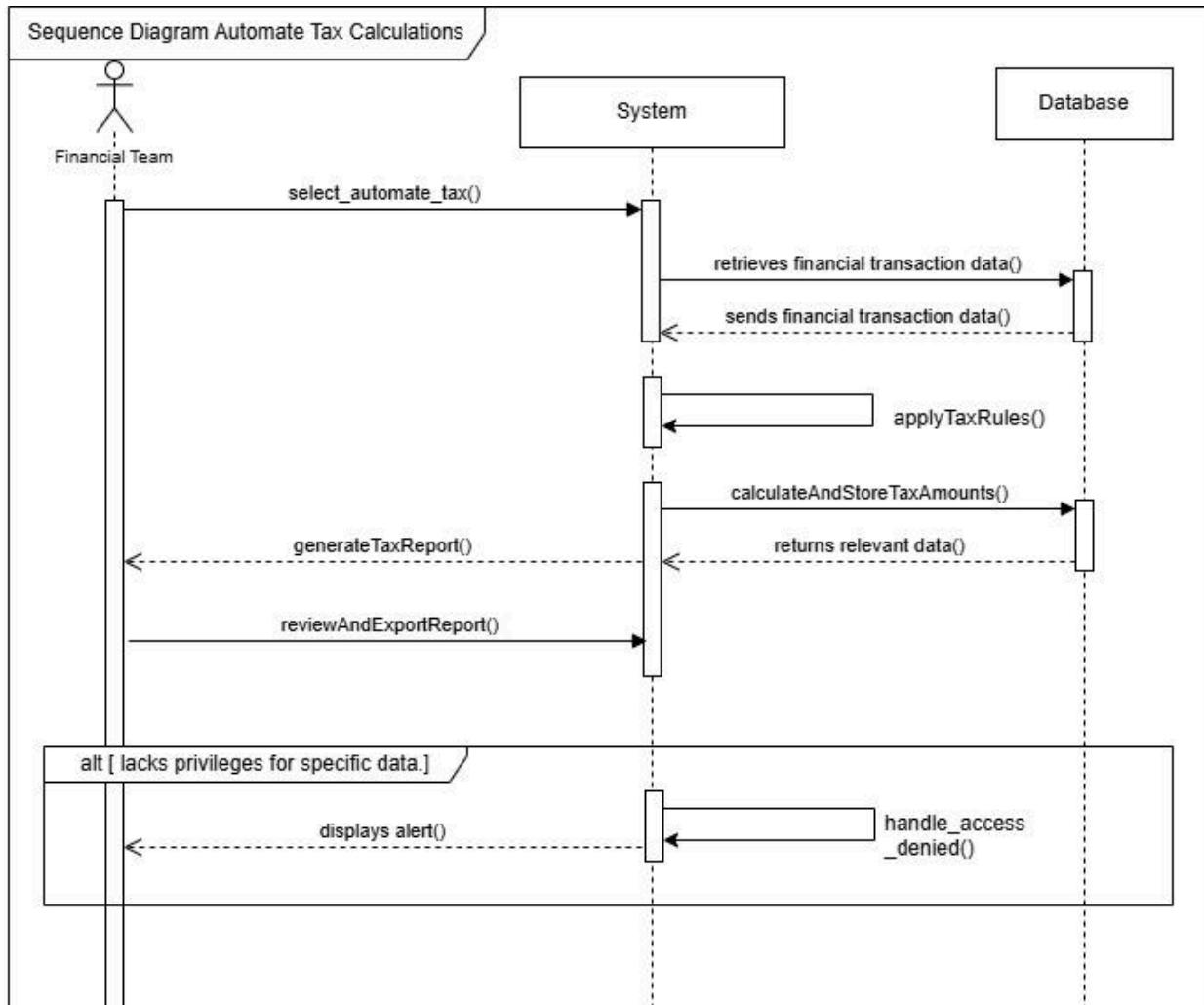
4.7.5.4.2 Generate Financial Reports



4.7.5.4.3 Track Budget Allocation



4.7.5.4.4 Automate Tax Calculations



4.7.6 Centralized Database Design



Figure 4.7.6: Centralized Database Design

4.7.7 Selected Sub-System Interface Design

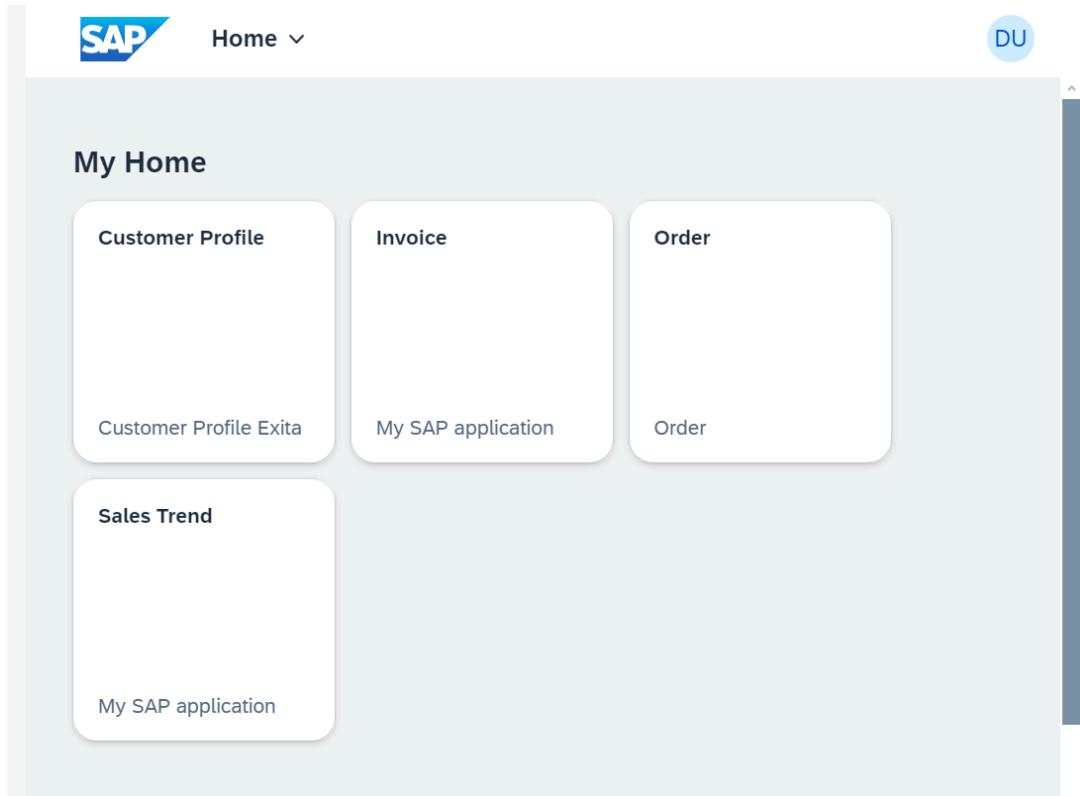


Image 4.7.7.1: Interface design of home screen

4.7.7.1 Track Customer Orders

Sales Orders (256,629)							Create		Create with Reference		Delivery Block		Billing Block		...		
Sales Or...	Sold-to Party	Customer Reference	Overall Status	Requested Delivery Date	Net Value	Document ...											
<input type="checkbox"/> 254438	KeepCycling (CACU_S05)	4500368574	In Process	06/19/2025	163,300.00	USD	06/19/2025	>									
<input type="checkbox"/> 254113	Cityscape Cycles (USCU_S14)	4500360590	In Process	06/18/2025	1,560.00	USD	06/18/2025	>									
<input type="checkbox"/> 253841	Bigmart (USCU_L09)	4500367335	Open	06/17/2025	21,300.00	USD	06/17/2025	>									
<input type="checkbox"/> 253840	G&M Bicycle (CACU_S01)	4500361656	In Process	06/17/2025	3,480.00	USD	06/17/2025	>									
<input type="checkbox"/> 253839	Bikepros (USCU_S17)	4500360589	In Process	06/17/2025	840.00	USD	06/17/2025	>									
<input type="checkbox"/> 253582	XYZ Cycles (CACU_S02)	4500368218	Completed	06/16/2025	0.00	USD	06/16/2025	>									
<input type="checkbox"/> 256693	Fit Cycles (USCU_S04)	4500393074	In Process	05/28/2025	24,180.00	USD	05/28/2025	>									
<input type="checkbox"/> 256692	Bikepros (USCU_S17)	4500390990	In Process	05/28/2025	32,200.00	USD	05/28/2025	>									
<input type="checkbox"/> 256691	Bigmart (USCU_L09)	4500390632	In Process	05/28/2025	2,100.00	USD	05/28/2025	>									
<input type="checkbox"/> 256690	Gogo Bikes (USCU_S16)	4500360569	In Process	05/28/2025	2,040.00	USD	05/28/2025	>									
<input type="checkbox"/> 256689	Custom Sports (USCU_S02)	4500355510	In Process	05/28/2025	3,430.00	USD	05/28/2025	>									
<input type="checkbox"/> 256466	MXBike (LACU_S05)	4500398332	In Process	05/27/2025	710.00	USD	05/27/2025	>									
<input type="checkbox"/> 256465	Dexon (USCU_L06)	4500396992	In Process	05/27/2025	7,100.00	USD	05/27/2025	>									
<input type="checkbox"/> 256464	KeepCycling (CACU_S05)	4500388832	In Process	05/27/2025	10,850.00	USD	05/27/2025	>									

Image 4.7.7.1.1: Interface design of track customer orders

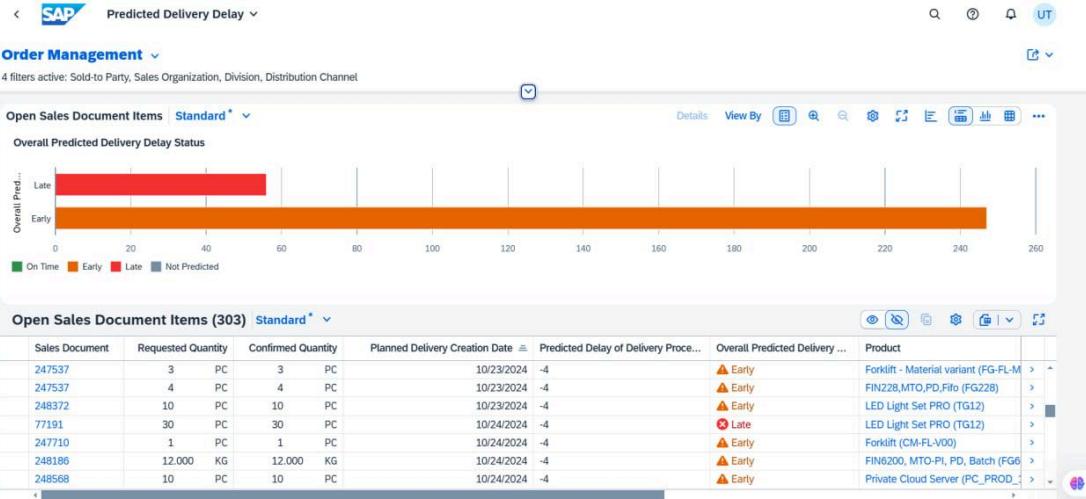


Image 4.7.7.1.2: Interface design of track customer orders

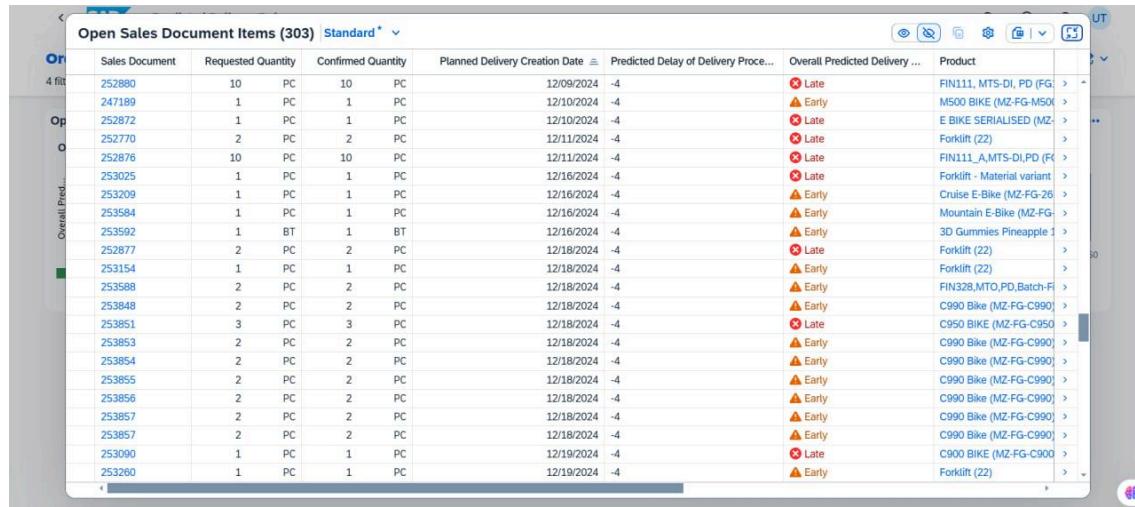
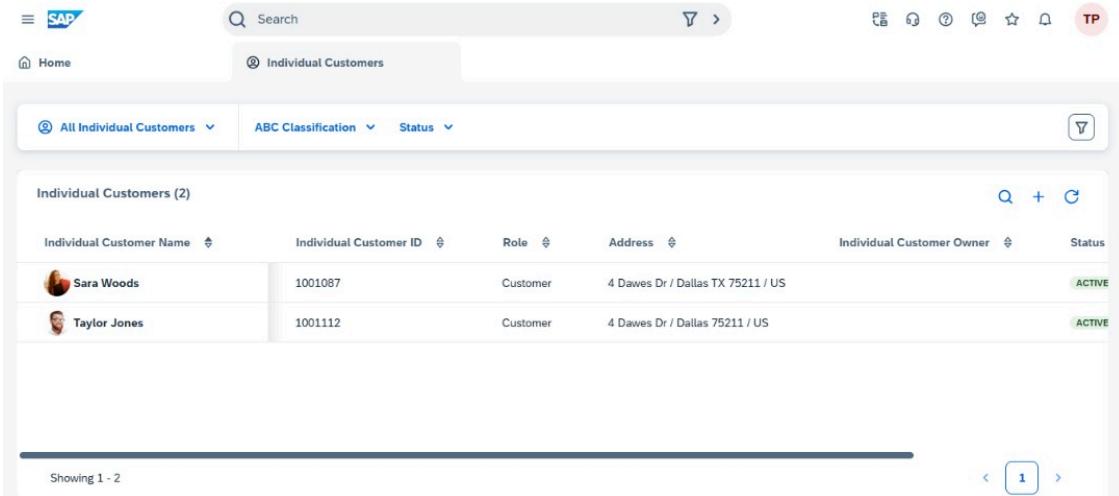


Image 4.7.7.1.3: Interface design of track customer orders

4.7.7.2 Manage Customer Profiles



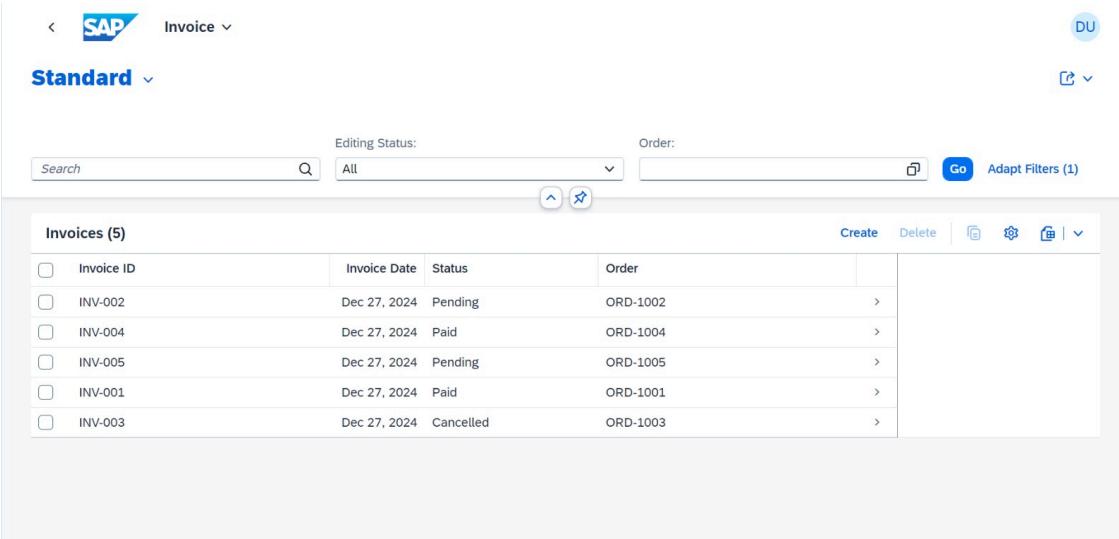
The screenshot shows the SAP interface for managing customer profiles. The top navigation bar includes the SAP logo, a search bar, and various icons. The main content area is titled "Individual Customers" and shows a list of "Individual Customers (2)". The columns are: Individual Customer Name, Individual Customer ID, Role, Address, Individual Customer Owner, and Status. Two entries are listed:

Individual Customer Name	Individual Customer ID	Role	Address	Individual Customer Owner	Status
Sara Woods	1001087	Customer	4 Dawes Dr / Dallas TX 75211 / US		ACTIVE
Taylor Jones	1001112	Customer	4 Dawes Dr / Dallas 75211 / US		ACTIVE

At the bottom, it says "Showing 1 - 2" and has navigation buttons for the first page (1) and the next page.

Image 4.7.7.2.1: Interface design of manage customer profile

4.7.7.3 Automate Sales Invoices



The screenshot shows the SAP interface for automating sales invoices. The top navigation bar includes the SAP logo, a search bar, and various icons. The main content area is titled "Invoices" and shows a list of "Invoices (5)". The columns are: Invoice ID, Invoice Date, Status, and Order. Five entries are listed:

Invoice ID	Invoice Date	Status	Order
INV-002	Dec 27, 2024	Pending	ORD-1002
INV-004	Dec 27, 2024	Paid	ORD-1004
INV-005	Dec 27, 2024	Pending	ORD-1005
INV-001	Dec 27, 2024	Paid	ORD-1001
INV-003	Dec 27, 2024	Cancelled	ORD-1003

Image 4.7.7.3.1: Interface design of automate sales invoices

New: Invoice

General Information

Invoice ID:*	Status:	Created By:	Modified By:
<input type="text"/>	<input type="text"/>	privileged	privileged
Invoice Date:	Created At:	Modified At:	Order:
<input type="text"/> e.g. Dec 31, 2024 <input type="button" value=""/>	Dec 27, 2024, 8:27:08 PM	Dec 27, 2024, 8:27:08 PM	<input type="text"/>

Create **Discard Draft**

Image 4.7.7.3.2: Interface design of automate sales invoices

4.7.7.4 Forecast Sales Trends



Image 4.7.7.4.1: Interface design of forecast sales trends

4.8 Summary

Chapter 4 outlines the comprehensive analysis and design of the proposed ERP system for Exita Manufacturing, focusing on bridging operational gaps and enhancing efficiency. The chapter systematically integrates organizational needs with technological solutions through:

1. System Requirements Gathering:

- Utilized the provided case study and online research to identify functional and non-functional requirements.
- Ensured the requirements address both core business processes and industry standards for security, performance, and scalability.

2. Comparison of Existing and Proposed Systems:

- Highlighted limitations in current systems, emphasizing how the proposed ERP will streamline workflows across four key subsystems: Reporting and Analytics, Sales and Customer Management (CRM), Procurement and Inventory Management (SCM), and Financial Management.

3. System and Enterprise Design:

Defined the enterprise architecture, including technology, business, and application layers, ensuring system interoperability and alignment with organizational goals.

- Presented the system architecture and its components, including data flow and modular design.

4. Project Design:

- Included use case and activity diagrams for all subsystems to illustrate workflows and user interactions.
- Designed a centralized database to ensure seamless data integration.
- Developed user interface designs for selected subsystems to enhance usability.

Conclusion:

The analysis and design process ensures the ERP system is well-equipped to address Exita Manufacturing's operational needs, with scalability and robustness to support long-term growth. This chapter bridges the gap between organizational goals and technological implementation, providing a solid foundation for system development.

Chapter 5: System Implementation

5.1 Introduction

The system implementation phase is a crucial step in ensuring that the proposed Enterprise Resource Planning (ERP) system is effectively deployed and operationalized to meet the objectives set forth in the earlier stages of the project. This chapter outlines the activities, tools, and processes involved in the implementation of the ERP system for Exita Manufacturing. It focuses on transforming the design specifications into a functional system through coding, database creation, module integration, and deployment.

The primary goal of this phase is to ensure that the system aligns with the functional and non-functional requirements identified during the analysis and design phases. Implementation activities include database development, coding of core system functions, integration of subsystems, and rigorous testing to validate system performance and usability.

Furthermore, this phase addresses the transition from existing systems to the new ERP platform, ensuring minimal disruptions to business operations. Key considerations include data migration, user training, and setting up the required infrastructure to support the new system. The successful implementation of the ERP system positions Exita Manufacturing to achieve enhanced operational efficiency, streamlined workflows, and improved decision-making capabilities.

5.2 System Development

The development of the ERP system began with the creation of a full-stack project using **SAP Build Code** within the **SAP Build Lobby**. This tool provided a seamless transition into **SAP Business Application Studio (BAS)**, enabling both low-code design and advanced development capabilities. Key activities included data modeling, service definition, user interface development, and logic implementation. Each step leveraged SAP's integrated tools to streamline the process and ensure the system was tailored to business requirements.

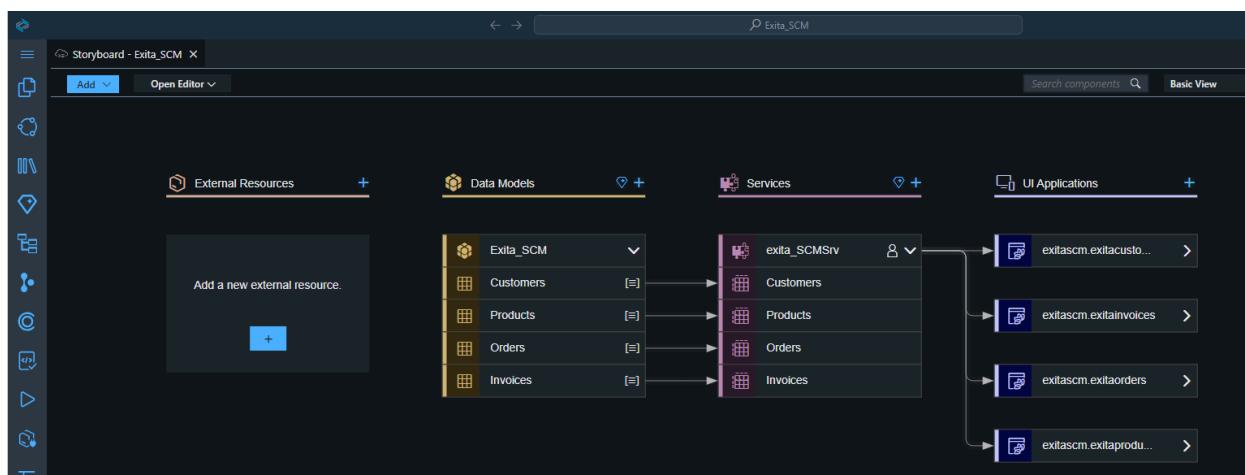


Image 5.2.1: Screenshot of storyboard of developing

5.2.1 Data Modeling

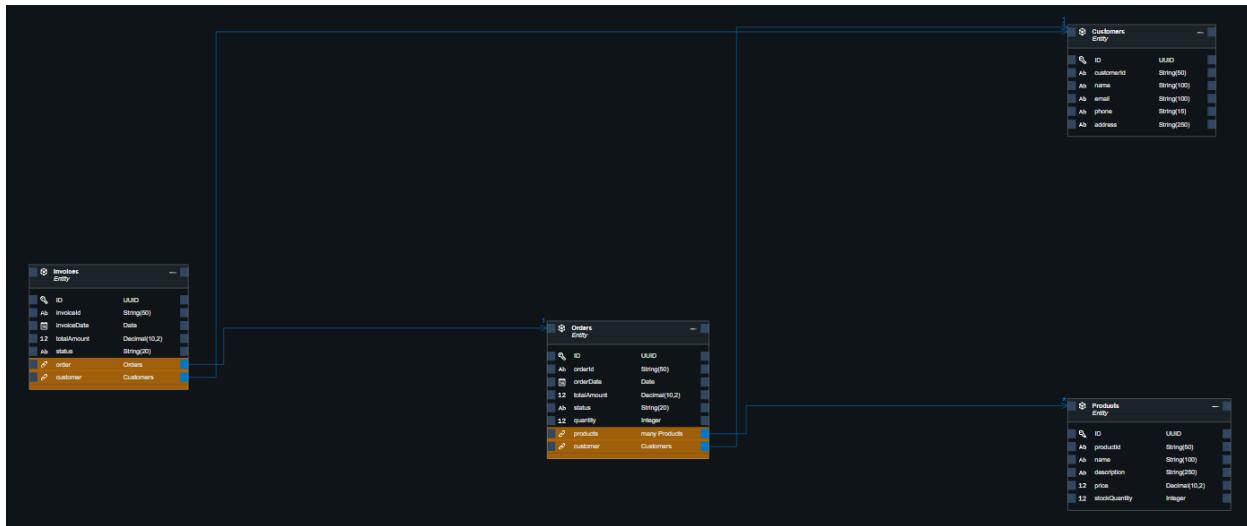


Image 5.2.1.1: Screenshot of data modeling

The initial step in system development involved designing and defining the data models to represent critical business entities. The data models were first constructed using **MySQL**, where tables such as Customers, Products, Orders, and Invoices were created to align with Exit Manufacturing's business processes. Relationships between these tables were established using primary and foreign keys, ensuring data integrity.

Once the data models were defined, they were imported into SAP Build Apps for further configuration and testing. The **Data Editor** in SAP Build Apps provided an intuitive interface to manage and visualize the data models.

Adding Sample Data

To facilitate testing, sample data was added to the entities using the following steps:

1. Navigating to the **Open Editor** menu and selecting **Sample Data**.
2. Selecting a data entity (e.g., Customers) and accessing the **INITIAL DATA** tab.
3. Setting the **Number of rows** to 5 and clicking **Add** to generate rows automatically.
4. Populating the rows with relevant sample data for each entity:
 - o **Customers**: Names, IDs, and contact details.
 - o **Products**: Product names, categories, and prices.
 - o **Orders**: Order dates, quantities, and customer references.
 - o **Invoices**: Invoice numbers, amounts, and related order details.

The data was validated to ensure consistency and accuracy, preparing it for integration with the system's services and UI.

5.2.2 Service Definition

Services were defined to enable interaction between the data models and the system's logic layer. This was achieved through the **Service Editor** in SAP Build Apps. For each data model, corresponding service endpoints were created, such as:

- Customers
- Products
- Orders
- Invoices

These services acted as an intermediary, facilitating CRUD (Create, Read, Update, Delete) operations for the data entities. They also provided the backend logic required for business processes, such as retrieving customer orders or calculating invoice totals.

5.2.3 User Interface Development

The development of the user interface (UI) focused on creating accessible and responsive applications for managing the ERP system's data. The process for building the first UI application included the following steps:

1. Adding the First UI Application:

Using the (+) button in the storyboard, a new UI application was created. Key details, such as the **Display Name** (e.g., "Customer Management"), **Data Source**, and **Description**, were provided to define the application.

2. Selecting Application and Page Templates:

The **Template-Based Responsive Application** option was chosen to ensure the UI was optimized for different devices. For the page layout, the **List Report Page** template was selected, offering a structured tabular view for managing entity data.

3. Linking to a Data Entity:

The appropriate data entity (e.g., Customers) was linked to the application, enabling it to dynamically display and manage data records.

4. Finalizing the Setup:

By clicking **Finish**, a functional UI application was generated, ready for customization and integration with system logic.

5.2.4 Logic Implementation

Business logic was implemented using the **Graphical Modeler** in SAP Build Apps. The process involved the following steps:

1. Selecting a specific service entity (e.g., Customers) and opening it in the **Graphical Modeler**.
2. Adding logic by selecting the **Add Logic** option and leaving default configurations.
3. Choosing the **Standard Event: Create** to define the behavior triggered when a new record was created.
4. Writing custom logic code directly in the editor, such as:
 - Assigning unique IDs for new records.
 - Calculating derived values (e.g., total order cost).
 - Validating input data.

This drag-and-drop interface, combined with custom logic coding, enabled efficient implementation of complex workflows without requiring extensive manual programming.

5.3 Create Database

5.3.1 Data Scheme



```
1  namespace Exita_SCM;
2  using { cuid } from '@sap/cds/common';
3
4  @assert.unique: { customerId: [customerId] }
5  entity Customers : cuid {
6    customerId: String(50) @mandatory;
7    name: String(100);
8    email: String(100);
9    phone: String(15);
10   address: String(250);
11 }
12
13 @assert.unique: { productId: [productId] }
14 entity Products : cuid {
15   productId: String(50) @mandatory;
16   name: String(100);
17   description: String(250);
18   price: Decimal(10,2);
19   stockQuantity: Integer;
20 }
21
22 @assert.unique: { orderId: [orderId] }
23 entity Orders : cuid {
24   orderId: String(50) @mandatory;
25   orderDate: Date;
26   totalAmount: Decimal(10,2);
27   status: String(20);
28   quantity: Integer;
29   products: Association to many Products;
30   customer: Association to Customers;
31 }
32
33 @assert.unique: { invoiceId: [invoiceId] }
34 entity Invoices : cuid {
35   invoiceId: String(50) @mandatory;
36   invoiceDate: Date;
37   totalAmount: Decimal(10,2);
38   status: String(20);
39   order: Association to Orders;
40   customer: Association to Customers;
41 }
42
```

Image 5.3.1.1: Screenshot of data scheme

The schema defines the database structure for the **Exita_SCM** system, written in **CDS (Core Data Services)**. Key entities include:

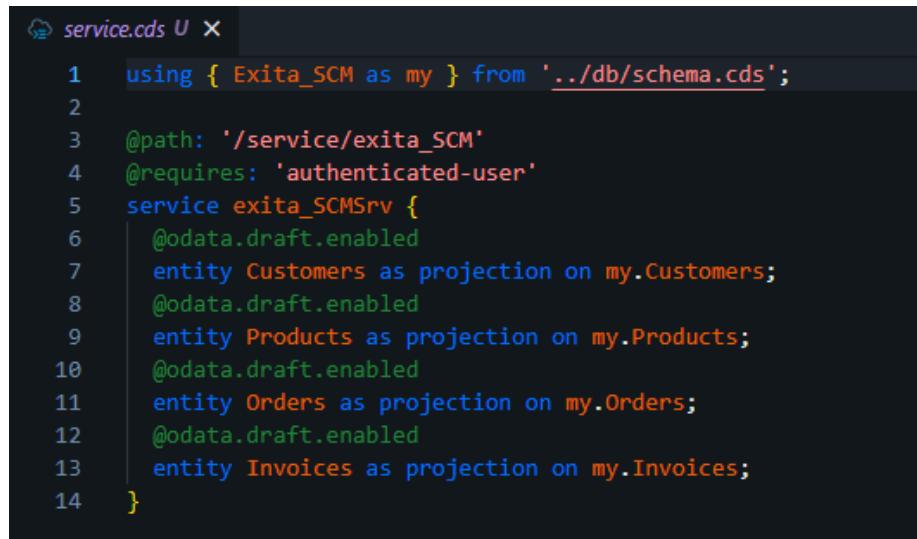
1. **Customers**
 - Attributes: customerId (unique), name, email, phone, address.
 - Stores customer details.
2. **Products**
 - Attributes: productId (unique), name, description, price, stockQuantity.
 - Manages product data.
3. **Orders**
 - Attributes: orderId (unique), orderDate, totalAmount, status, quantity.
 - Associations: Links to Products and Customers.
4. **Invoices**
 - Attributes: invoiceId (unique), invoiceDate, totalAmount, status.
 - Associations: Links to Orders and Customers.

Features:

- **Relationships:** Entities are linked using associations.
- **Validation:** `@assert.unique` ensures unique IDs.

5.3.2 Service Layer

This schema supports robust data management for the system.



```
service.cds U X
 1  using { Exita_SCM as my } from '../db/schema.cds';
 2
 3  @path: '/service/exita_SCM'
 4  @requires: 'authenticated-user'
 5  service exita_SCMSrv {
 6    @odata.draft.enabled
 7    entity Customers as projection on my.Customers;
 8    @odata.draft.enabled
 9    entity Products as projection on my.Products;
10    @odata.draft.enabled
11    entity Orders as projection on my.Orders;
12    @odata.draft.enabled
13    entity Invoices as projection on my.Invoices;
14  }
```

Image 5.3.2.1: Screenshot of service layer

This file defines the service layer for connecting the database to the application.

- **Namespace:** Imports Exita_SCM from schema.cds.
- **Service:** exita_SCMSrv, available at /service/exita_SCM. Requires authenticated user access.
- **Entity Projections:** Exposes **Customers**, **Products**, **Orders**, and **Invoices** for external use, with @odata.draft.enabled allowing draft data handling.

Purpose:

The service layer secures and standardizes access to database entities.

Exita_SCM-Customers.csv	
1	ID;customerID;name;email;phone;address
2	be3dd4eb-a544-4b39-925c-3996ac276609;CUST001;Ahmad Ali;ahmad.ali@example.my;+60123456789;123 Jalan Ampang, Kuala Lumpur, Malaysia
3	5ca11c6f-0894-45f7-acf6-cd9c91d6f84c;CUST002;Siti Nurhaliza;siti.nurhaliza@example.my;+60129876543;456 Jalan Bukit Bintang, Kuala Lumpur, Malaysia
4	45ce0fd1-89bd-4a21-bc04-f5bb4b54f764;CUST003;Tan Ah Kow;tan.ahkow@example.my;+60123459876;789 Jalan Raja Chulan, Kuala Lumpur, Malaysia
5	a0bc1e1d-792e-4c60-b52b-fc89d5aaa0c8;CUST004;Ravi Kumar;ravi.kumar@example.my;+60129873456;101 Jalan Tun Razak, Kuala Lumpur, Malaysia
6	6b6dbedb-0467-40e2-b0fa-c364ff6e812f;CUST005;Lim Mei Ling;lim.meiling@example.my;+60123451234;202 Jalan Sultan Ismail, Kuala Lumpur, Malaysia
7	13ffa527-8192-4d79-8c8f-d221e8b3b693;CUST006;Zainab Binti Zain;zainab.zain@example.my;+60129871234;303 Jalan Petaling, Kuala Lumpur, Malaysia
8	0edd4df6-42b2-44bd-9212-2e13bdf796ee;CUST007;Mohd Faizal;mohd.faizal@example.my;+60123456712;404 Jalan Imbi, Kuala Lumpur, Malaysia
9	9fd85d75-f9ad-e441-b7a9-2c71ed874b3b;CUST008;Chong Wei;chong.wei@example.my;+60129876512;505 Jalan Pudu, Kuala Lumpur, Malaysia
10	bafd6822-0c69-4462-b071-3f1dc4bc5499;CUST009;Aishah Binti Ahmad;aishah.ahmad@example.my;+60123459812;606 Jalan Cheras, Kuala Lumpur, Malaysia
11	f3e6a6c1-e44a-445f-87cb-cb0441c2a986;CUST010;Rajesh Singh;rajesh.singh@example.my;+60129873412;707 Jalan Kuching, Kuala Lumpur, Malaysia

Image 5.3.2.2: Screenshot of service layer

Exita_SCM-Invoices.csv	
1	ID;invoiceId;invoiceDate;totalAmount;status;order_ID;customer_ID
2	f9ec435-25af-4656-a7ea-a497909cb353;INV-001;2025-01-21;1500.00;paid;1ca42adc-9847-4ede-9a4a-cbc5e6c251a9;be3dd4eb-a544-4b39-925c-3996ac276609
3	d2e3424a-5d66-465a-9e6a-878b10ab89c1;INV-002;2025-01-21;2500.00;paid;64add1d-0111-4c83-b052-2979b7504b72;5ca11c6f-0894-45f7-acf6-cd9c91d6f84c
4	1cd1921f-e79f-451b-ae4c-ae989ad62fd4;INV-003;2025-01-21;1750.00;paid;9aa994ae-1b1d-4b6c-81c7-6135c3f1fd1;45ce0fd1-89bd-4a21-bc04-f5bb4b54f764
5	4f428c84-1985-4a54-a47f-e5d8f91ad070;INV-004;2025-01-21;3000.00;paid;3ec3f839-6032-4a5f-9ab3-c192f32ea1f8;a0bc1e1d-792e-4c60-b52b-fc89d5aaa0c8
6	643bdbb9-8346-439c-a341-6dee28f16ef9;INV-005;2025-01-21;2250.00;paid;e3e00e02-f212-427a-8405-e80f6bcc3f80;6b6dbedb-0467-40e2-b0fa-c364ff6e812f
7	7a3a0372-5ff7-4b27-az2d-2d0a9e662e01;INV-006;2025-01-21;4000.00;paid;faa86c88-4e06-4738-aeff-df23a9c28f08;13ffa527-8192-4d79-8c8f-d221e8b3b693
8	680f18d4-9395-4cd0-9ed2-5726a864b4bd;INV-007;2025-01-21;3200.00;paid;1566a0f9-1189-4dd8-ac2a-e231da282fa;0ed4df6e-42b2-44bd-9212-2e13b8f976ee
9	1bb612ab-a0c2-4b58-b360-e74867628b5e;INV-008;2025-01-21;2750.00;paid;812226a7-e976-4859-9d9d-12d20b2770f7;9fd85d75-f9ad-e441-b7a9-2c71ed8743b3
10	214dcde6-e987-4ba9-a019-0e2cdcb9bc52;INV-009;2025-01-21;1800.00;paid;f092958d-a4fe-41de-a214-34c14d26f522;bafd6822-0c69-4462-b071-3f1dc4bc5499
11	4918d822-bfe4-485f-8d69-f2ae8abfc9c6;INV-010;2025-01-21;3500.00;paid;3b480e34-5b1b-a27-8769-431990d448f1;f3e6a6c1-e44a-445f-87cb-cb0441c2a986

Image 5.3.2.3: Screenshot of service layer

Exita_SCM-Orders.csv	
1	ID;orderId;orderDate;totalAmount;status;quantity;products_ID;customer_ID
2	1ca42adc-9847-4ede-9a4a-cbc5e6c251a9;ORD001;2025-01-21;150.00;Pending;2;13e046fe-b9c5-4b93-8048-e38853cc2a;be3dd4eb-a544-4b39-925c-3996ac276609
3	64add1d-0111-4c83-b052-2979b7504b72;ORD002;2025-01-21;200.00;Shipped;1;3f3466a5-4dd2-4210-aec2-455f0320aa18;5ca11c6f-0894-45f7-acf6-cd9c91d6f84c
4	9aa994ae-1b1d-4b6c-81c7-6135c3f1fd1;ORD003;2025-01-21;300.00;Delivered;1;6c3e373d-3112-49d4-904b-5ee727d89f8c;45ce0fd1-89bd-4a21-bc04-f5bb4b54f764
5	3ec3f839-6032-4a5f-9ab3-c192f32ea1f8;ORD004;2025-01-21;250.00;Cancelled;1;b0441c3d-616c-4f3c-a6cb-77ce4f2dta9;a0bc1e1d-792e-4c60-b52b-fc89d5aaa0c8
6	e3e00e02-f212-427a-8405-e80f6bcc3f80;ORD005;2025-01-21;100.00;Pending;5;7395a294-9e49-44cc-be66-ca2166dbc9ae;6b6dbedb-0467-40e2-b0fa-c364ff6e812f
7	faa86c88-4e06-4738-aeff-df23a9c28f08;ORD006;2025-01-21;175.00;Shipped;2;eacc496-4a2f-9dea-8ca32be284ad;13ffa527-8192-4d79-8c8f-d221e8b3b693
8	1566a0f9-1189-4dd8-ac2a-e231da282fa;ORD007;2025-01-21;225.00;Delivered;4;8db4a418-9d65-42b8-9670-8e027808373a;0ed4df6e-42b2-44bd-9212-2e13b8f976ee
9	812226a7-e976-4859-9d9d-12d20b2770f7;ORD008;2025-01-21;275.00;Cancelled;1;ea2b34dc-30f2-4c6f-82fe-df7de38c2d84;9fd85d75-f9ad-e441-b7a9-2c71ed8743b3
10	f092958d-a4fe-41de-a214-34c14d26f522;ORD009;2025-01-21;125.00;Pending;3;b6e5a013-5768-4ed8-af37-4e049d551fa;bafd6822-0c69-4462-b071-3f1dc4bc5499
11	3b480e34-5b1b-a27-8769-431990d448f1;ORD010;2025-01-21;350.00;Shipped;2;09ae296f-503e-4af1-ab85-ba1b3d9da1c;f3e6a6c1-e44a-445f-87cb-cb0441c2a986

Image 5.3.2.4: Screenshot of service layer

Exita_SCm-Products.csv	
1	ID;productId;name;description;price;stockQuantity
2	13e046fe-b9c5-4b03-8d48-ec38853ccc2a;P001;Concrete;High-quality concrete for construction;75.50;100
3	3f3466a5-4dd2-4210-aec2-455f0320aa18;P002;Steel Beams;Durable steel beams for structural support;150.00;200
4	6c3e373d-3112-49d4-904b-5ee727d89f8c;P003;Bricks;Red bricks for building walls;0.50;5000
5	b8441cd3-616c-4fc3-a6cb-77cea4f2d1a9;P004;Cement;Premium cement for construction;10.00;300
6	7395a294-94e9-44cc-be66-ca2166dbc9ae;P005;Glass Windows;Tempered glass windows;120.00;50
7	eacc496-a62f-4e2b-9dea-8ca32be284ad;P006;Wooden Planks;High-quality wooden planks for flooring;25.00;150
8	8db4a418-9d65-42bb-9670-8e027808373a;P007;Roof Tiles;Durable roof tiles;2.00;1000
9	ea2b34dc-30f2-4c6f-82fe-df7de38c2db4;P008;Paint;Weather-resistant paint;15.00;400
10	b6esa013-5768-4ed8-af37-4e049d5515fa;P009;Insulation Material;High-efficiency insulation material;30.00;250
11	09ae296f-503e-4af1-ab85-ba1bb3d9da1c;P010;Pipes;PVC pipes for plumbing;5.00;600

Image 5.3.2.4: Screenshot of service layer

5.3.3 Data Files

The provided data files populate the database with sample records for testing and development. These files correspond to the entities defined in the schema:

1. **Exita_SCm-Customers.csv:** Contains customer data such as names, email addresses, phone numbers, and addresses.
2. **Exita_SCm-Invoices.csv:** Records invoice data, including invoice IDs, associated order IDs, total amounts, and statuses.
3. **Exita_SCm-Orders.csv:** Details order data, including IDs, order dates, quantities, total amounts, and customer associations.
4. **Exita_SCm-Products.csv:** Lists product data such as product IDs, names, descriptions, prices, and stock quantities.

Purpose:

These files ensure the system's functionality by providing realistic test data for operations and validations.

5.4 Coding of the system's main functions

5.4.1 Customer logic

```
1  /**
2  *
3  * @On(event = { "CREATE" }, entity = "exita_scmSrv.Customers")
4  * @param {Object} request - User information, tenant-specific CDS model, headers, and query parameters
5  */
6  module.exports = async function (request) {
7      const { Customers } = cds.entities;
8
9      // Extract data from the request
10     const { customerId, name, email, phone, address } = request.data;
11
12     // Validation functions
13     const validateCustomerId = (id) => id.startsWith("CUST") && /\d+/.test(id.slice(4));
14     const validatePhone = (phone) => phone.startsWith("+60") && /\+60\d{8,10}$.test(phone);
15     const validateEmail = (email) => /^[^s@]+@[^\s@]+\.[^\s@]+$.test(email);
16
17     // Validation checks
18     if (!customerId) {
19         return request.error(400, "Customer ID is mandatory");
20     }
21     if (!validateCustomerId(customerId)) {
22         return request.error(400, "Invalid Customer ID. Must start with 'CUST' followed by numbers.");
23     }
24
25     if (!name || name.trim() === "") {
26         return request.error(400, "Customer name is mandatory");
27     }
28 }
```

Image 5.4.1.1: Screenshot of code from Customer logic

```
if (!validateEmail(email)) {
    return request.error(400, "Invalid email format");
}

if (!phone) {
    return request.error(400, "Phone number is mandatory");
}
if (!validatePhone(phone)) {
    return request.error(400, "Invalid phone number. Must start with '+60' and contain 8-10 digits.");
}

if (!address || address.trim() === "") {
    return request.error(400, "Address is mandatory");
}

// Check if a customer with the same customerId already exists
const existingCustomer = await SELECT.one.from(Customers).where({ customerId });
if (existingCustomer) {
    return request.error(400, `Customer with ID ${customerId} already exists`);
}

// Insert the new customer into the database (assuming ID is auto-increment)
await INSERT.into(Customers).columns('customerId', 'name', 'email', 'phone', 'address').values(customerId, name, email, phone, address);

// Optionally, you can add additional logic here, such as logging or sending notifications
return request.success(`Customer ${customerId} created successfully`);
```

Image 5.4.1.2: Screenshot of code from Customer logic

This custom logic script is used to enforce data validation rules when creating or updating customer records in the system. It ensures that data integrity is maintained by validating phone numbers and email addresses.

Code Breakdown:

1. Email Validation:

- Checks if the email format is correct.
- If not, it returns an error message: "Invalid email format".

2. Phone Number Validation:

- Ensures a phone number is provided.
- Validates that it starts with **+60** and is between 8 to 10 digits.
- Returns an error if the phone number is not valid.

3. Address Validation:

- Checks if an address is provided.
- Returns an error if the address is missing.

4. Customer Existence Check:

- Looks up the database to see if a customer with the given ID already exists.
- If the customer exists, it returns a message indicating that the ID is already in use.

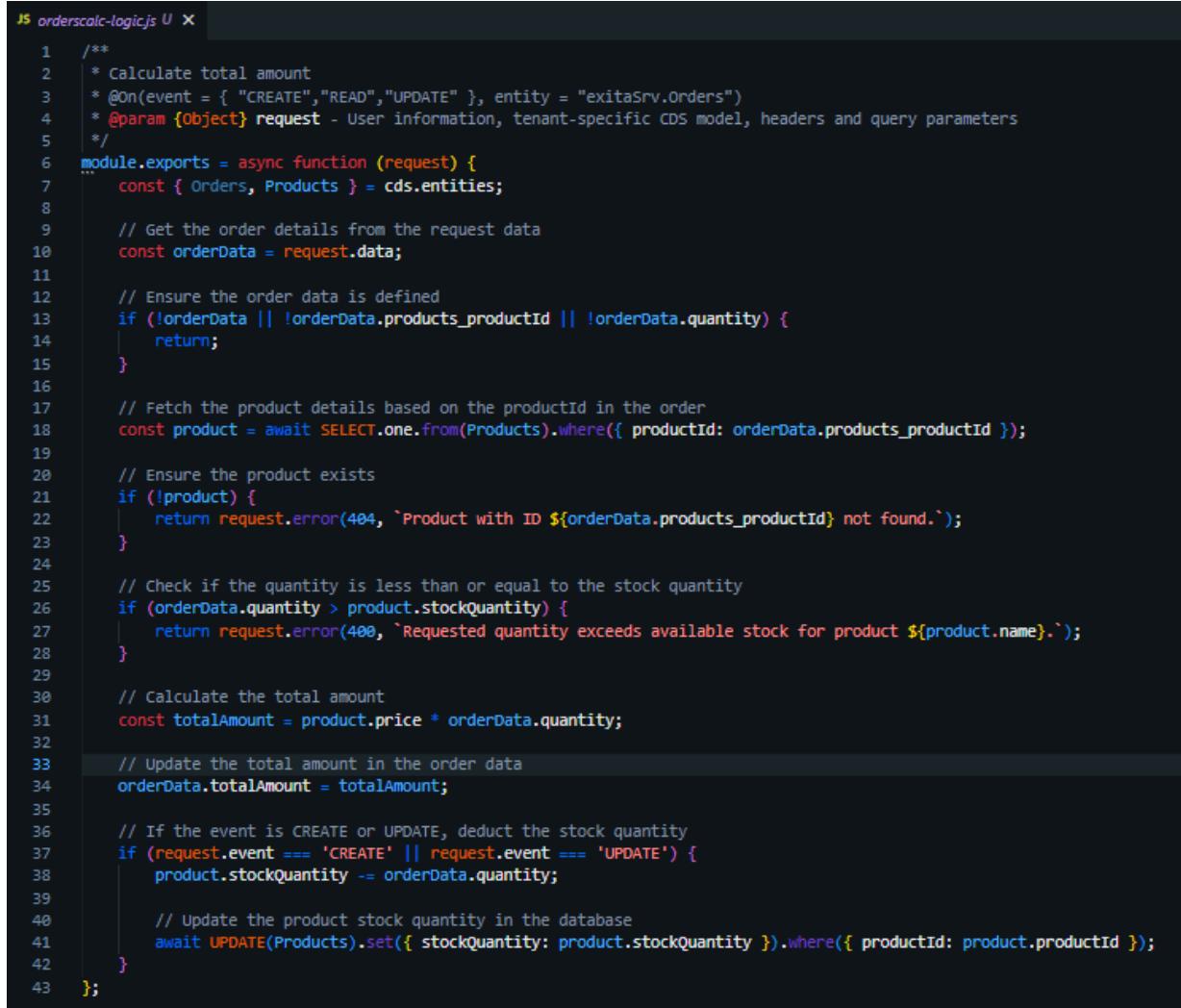
5. Insert New Customer:

- If the customer does not exist, it adds a new record to the database with the provided name, email, and phone.
- Optionally, it can log this action or send notifications.

Purpose:

This script safeguards the database by ensuring only correctly formatted data is accepted, enhancing the system's reliability and preventing potential issues caused by invalid input.

5.4.2 Calculate Total Amount Logic



```
JS orderscalc-logic.js U X
1  /**
2   * Calculate total amount
3   * @On(event = { "CREATE", "READ", "UPDATE" }, entity = "exitaSrv.Orders")
4   * @param {Object} request - User information, tenant-specific CDS model, headers and query parameters
5   */
6  module.exports = async function (request) {
7    const { Orders, Products } = cds.entities;
8
9    // Get the order details from the request data
10   const orderData = request.data;
11
12   // Ensure the order data is defined
13   if (!orderData || !orderData.products_productId || !orderData.quantity) {
14     return;
15   }
16
17   // Fetch the product details based on the productId in the order
18   const product = await SELECT.one.from(Products).where({ productId: orderData.products_productId });
19
20   // Ensure the product exists
21   if (!product) {
22     return request.error(404, `Product with ID ${orderData.products_productId} not found.`);
23   }
24
25   // Check if the quantity is less than or equal to the stock quantity
26   if (orderData.quantity > product.stockQuantity) {
27     return request.error(400, `Requested quantity exceeds available stock for product ${product.name}.`);
28   }
29
30   // Calculate the total amount
31   const totalAmount = product.price * orderData.quantity;
32
33   // Update the total amount in the order data
34   orderData.totalAmount = totalAmount;
35
36   // If the event is CREATE or UPDATE, deduct the stock quantity
37   if (request.event === 'CREATE' || request.event === 'UPDATE') {
38     product.stockQuantity -= orderData.quantity;
39
40     // Update the product stock quantity in the database
41     await UPDATE(Products).set({ stockQuantity: product.stockQuantity }).where({ productId: product.productId });
42   }
43};
```

Image 5.4.2.1: Screenshot of code from Calculate Total Amount Logic

Code Breakdown:

Event Handling

- The script listens for CREATE and UPDATE events on the exitaSRV.Orders entity.
- It executes logic to calculate the total amount and update product stock whenever these events are triggered.

Validation Rules

1. Order Data Validation:

- Ensures orderData.products_productId and orderData.quantity are provided in the request.
- If missing, it returns a 400 status error with the message: *"Invalid order data."*

2. Product Existence Check:

- Fetches the product details from the database using SELECT.one.
- If no matching product is found, it triggers a 404 error: *"Product with ID [productId] not found."*

3. Stock Quantity Validation:

- Compares the requested quantity (orderData.quantity) with the available stock (product.stockQuantity).
- If the quantity exceeds the stock, a 409 error is returned: *"Requested quantity exceeds available stock for product [product.name]."*

Error Handling

- All validation failures halt further processing and return appropriate HTTP error codes (400, 404, or 409) with detailed messages to inform the client.

Business Logic

1. Calculate Total Amount:

- Multiplies the product price (product.price) by the order quantity (orderData.quantity).
- Stores the result in orderData.totalAmount.

2. Stock Update:

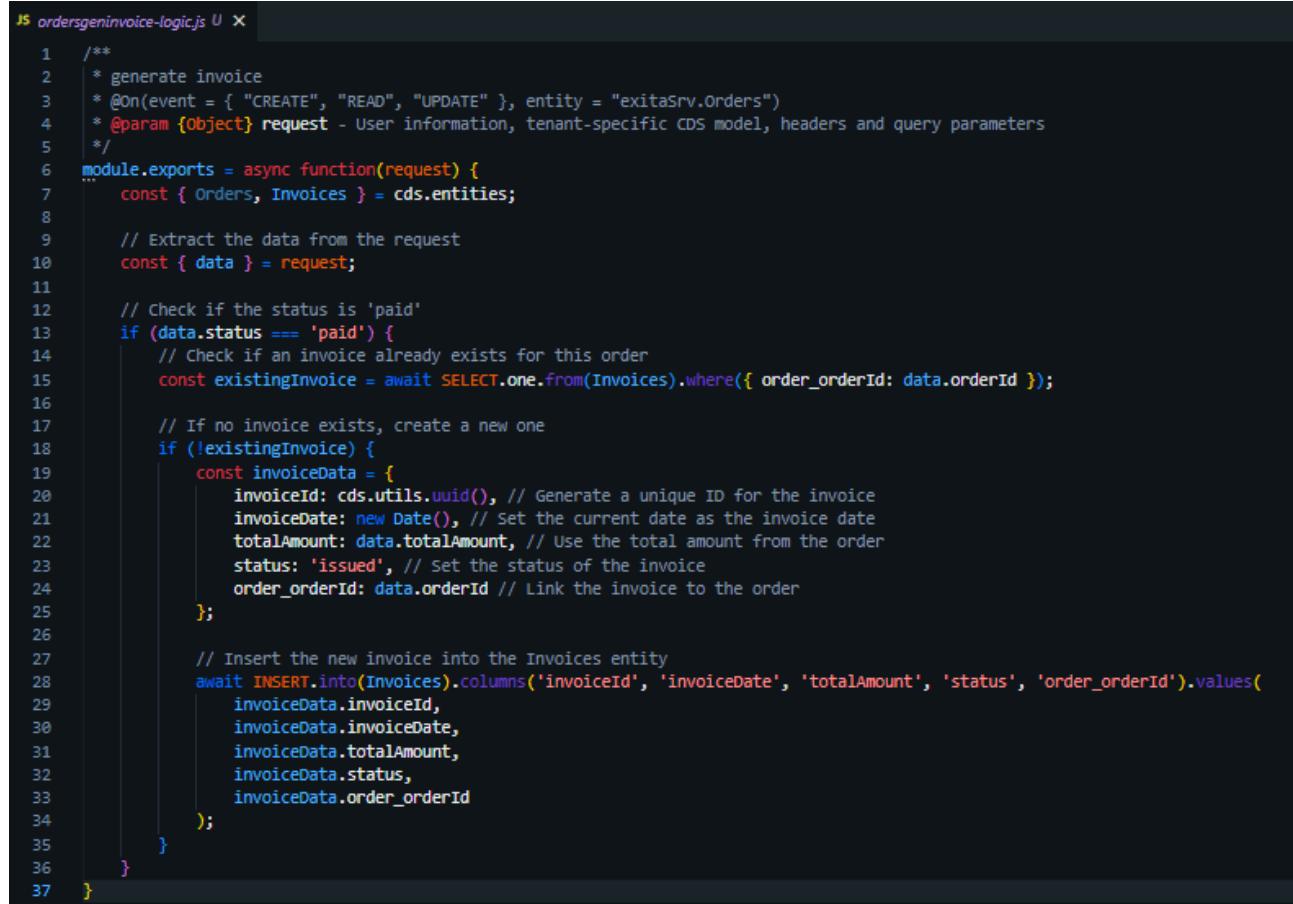
For CREATE and UPDATE events, deducts the ordered quantity from the product's stock (product.stockQuantity).

- Updates the database with the new stock value using UPDATE.

Purpose

- This script ensures accurate order processing while maintaining stock integrity.
- By validating inputs and restricting invalid operations, it safeguards database consistency and enhances the system's reliability.

5.4.3 Automatic Invoice Generation Logic



```
1  /**
2   * generate invoice
3   * @On(event = { "CREATE", "READ", "UPDATE" }, entity = "exitaSrv.Orders")
4   * @param {Object} request - User information, tenant-specific CDS model, headers and query parameters
5   */
6  module.exports = async function(request) {
7      const { Orders, Invoices } = cds.entities;
8
9      // Extract the data from the request
10     const { data } = request;
11
12     // Check if the status is 'paid'
13     if (data.status === 'paid') {
14         // Check if an invoice already exists for this order
15         const existingInvoice = await SELECT.one.from(Invoice).where({ order_orderId: data.orderId });
16
17         // If no invoice exists, create a new one
18         if (!existingInvoice) {
19             const invoiceData = {
20                 invoiceId: cds.utils.uuid(), // Generate a unique ID for the invoice
21                 invoiceDate: new Date(), // Set the current date as the invoice date
22                 totalAmount: data.totalAmount, // Use the total amount from the order
23                 status: 'issued', // Set the status of the invoice
24                 order_orderId: data.orderId // Link the invoice to the order
25             };
26
27             // Insert the new invoice into the Invoices entity
28             await INSERT.into(Invoice).columns('invoiceId', 'invoiceDate', 'totalAmount', 'status', 'order_orderId').values(
29                 invoiceData.invoiceId,
30                 invoiceData.invoiceDate,
31                 invoiceData.totalAmount,
32                 invoiceData.status,
33                 invoiceData.order_orderId
34             );
35         }
36     }
37 }
```

Image 5.4.3.1: Screenshot of code from Automatic Invoice Generation Logic

Code Breakdown:

Event Handling

- The script listens for CREATE, READ, and UPDATE events on the exitasrv.Orders entity.
- It triggers the logic whenever an order's status changes to "paid".

Validation Rules

1. Check for Paid Status:

- The logic only runs if the order's status is "paid".
- If the status is not "paid", the script does nothing.

2. Existing Invoice Check:

- Queries the Invoices entity to see if an invoice already exists for the given orderId.
- If an invoice is found, no new invoice is created.

3. Generate Invoice Data:

- Queries the Invoices entity to see if an invoice already exists for the given orderId.
- If an invoice is found, no new invoice is created.

4. Insert Invoice into Database:

- If no existing invoice is found:
 - Generates a unique invoiceId using cds.utils.uuid().
 - Sets the invoiceDate to the current date (new Date()).
 - Uses the totalAmount from the order data.
 - Sets the status of the invoice to "issued".
 - Links the invoice to the order using order_orderId.

5. Insert Invoice into Database:

- The new invoice is inserted into the Invoices entity using the INSERT operation.
- Fields populated include invoiceId, invoiceDate, totalAmount, status, and order_orderId.

Error Handling:

- The script assumes the database handles potential conflicts, such as duplicate invoiceId.
- Logic is designed to avoid duplicate invoices by checking for existing entries before creation.

Purpose:

- Automates the generation of invoices for orders marked as "paid".
- Ensures no duplicate invoices are created for the same order.
- Links each invoice to its corresponding order, improving traceability and system efficiency.

5.4.4 index.html

```
index.html ⑥
1  <!DOCTYPE HTML>
2  <html>
3
4  <head>
5      <title> Project Preview </title>
6      <script src="https://ui5.sap.com/1.129.2/resources/sap-ui-core.js"
7          id="sap-ui-bootstrap"
8          data-sap-ui-theme="sap_horizon"
9          data-sap-ui-bindingSyntax="complex"
10         data-sap-ui-xx-bindingSyntax="complex"
11         data-sap-ui-resourceroots='{"assets": "assets"}'
12         data-sap-ui-libs="sap.m,sap.uxap,sap.ui.layout,sap.f">
13     </script>
14     <link rel="stylesheet" href="assets/preview.css">
15 </head>
16
17 <body id="body" class="sapUiBody">
18     <script>
19         "use strict";
20         sap.ui.localResources("assets");
21         (function (jQuery, App, Page) {
22             sap.ui.getCore().attachInit(function () {
23                 sap.ui.define([
24                     "sap/ui/core/mvc/Controller",
25                     "sap/ui/model/json/JSONModel"
26                 ], function (Controller, JSONModel) {
27                     return Controller.extend("PreviewController", {
28                         init: function () {
29
30                             const data = {"Services": [{"ServiceName": "exita_SCMSrv", "DetailsURL": "./service/exita_SOM/", "Metadata": "./service/exita_SOM/$metadata", "Entities": [{"EntityName": "exita_SOM", "Type": "entity", "Properties": [{"Name": "id", "Type": "string", "Value": "1234567890"}, {"Name": "name", "Type": "string", "Value": "Project Preview"}]}]}];
31                             this.getView().setModel(new JSONModel(data));
32                         }
33                     });
34                 });
35                 var oApp = new App()
36                 var previewView = sap.ui.view({
37                     viewName: "assets.Preview",
38                     type: sap.ui.core.mvc.ViewType.XML
39                 });
40                 oApp.addPage(new Page({
41                     content: [previewView],
42                     showHeader: false
43                 })).placeAt("content");
44             });
45
46             }(jQuery, sap.m.App, sap.m.Page));
47         </script>
48         <div id="content"></div>
49     </body>
50 </html>
```

Image 5.4.4.1: Screenshot of code from index.html

Code Breakdown

Purpose

- Entry point for a SAPUI5 preview application, initializing libraries, setting up the view, and binding data.

Header Section

1. **SAPUI5 Framework**
 - Loads the SAPUI5 core library (1.129.2) with additional libraries and themes.
2. **Custom Styles**
 - Links preview.css for styling.

Body Section

1. **Main Content Wrapper**
 - <div id="content"></div> acts as the rendering container for the application.
2. **Application Initialization**
 - Ensures SAPUI5 core is loaded before logic execution.

Key Logic

1. **Controller Definition**
 - Loads a custom controller (PreviewController) and initializes a JSONModel with service metadata.
2. **View Setup**
 - Loads an XML view (assets.Preview) and wraps it in a sap.m.App container.
3. **Rendering**
 - Places the app in the content div.

Highlights

- **Dynamic Binding:** Uses JSONModel for metadata handling.
- **Modular MVC Design:** Separates logic (controller) and UI (view).
- **Responsive UI:** Leverages SAPUI5's robust layout libraries.

5.4.5 launchpadPage.html

```
launchpadPage.html U X
1  <!DOCTYPE html>
2  <html>
3  |  <head>
4  |  |  <meta http-equiv="X-UA-Compatible" content="IE=edge">
5  |  |  <meta charset="UTF-8">
6  |  |  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |  |  <title>{{appTitle}}</title>
8
9  |  <script>
10 |  |  window["sap-ushell-config"] = {
11 |  |  "defaultRenderer": "fiori2",
12 |  |  "renderers": {
13 |  |  |  "fiori2": {
14 |  |  |  |  "componentData": {
15 |  |  |  |  |  "config": {
16 |  |  |  |  |  |  "enableSearch": false
17 |  |  |  |  |  }
18 |  |  |  |  }
19 |  |  |  }
20 |  |  },
21 |  |  "applications": {
22 |  |  |  "exitascmexitacustomers-display": {
23 |  |  |  |  "title": "Exita Customers",
24 |  |  |  |  "description": "Customers",
25 |  |  |  |  "applicationType": "URL",
26 |  |  |  |  "navigationMode": "newWindowThenEmbedded",
27 |  |  |  |  "url": "./exita_scm.exita_customers/webapp",
28 |  |  |  |  "additionalInformation": "SAPUI5.Component=exitascm.exitacustomers"
29 |  |  |  },
30 |  |  |  "exitascmexitainvoices-display": {
31 |  |  |  |  "title": "Exita Invoices",
32 |  |  |  |  "description": "Invoices",
33 |  |  |  |  "applicationType": "URL",
34 |  |  |  |  "navigationMode": "newWindowThenEmbedded",
35 |  |  |  |  "url": "./exita_scm.exita_invoices/webapp",
36 |  |  |  |  "additionalInformation": "SAPUI5.Component=exitascm.exitainvoices"
37 |  |  |  },
38 |  |  |  "exitascmexitaorders-display": {
39 |  |  |  |  "title": "Exita Orders",
40 |  |  |  |  "description": "Orders",
41 |  |  |  |  "applicationType": "URL",
```

Image 5.4.5.1: Screenshot of code from launchpadPage.html

```

42     "navigationMode": "newWindowThenEmbedded",
43     "url": "./exita_scm.exita_orders/webapp",
44     "additionalInformation": "SAPUI5.Component=exitascm.exitaorders"
45   },
46   "exitascmexitaproducts-display": {
47     "title": "Exita Products",
48     "description": "Products",
49     "applicationType": "URL",
50     "navigationMode": "newWindowThenEmbedded",
51     "url": "./exita_scm.exita_products/webapp",
52     "additionalInformation": "SAPUI5.Component=exitascm.exitaproducts"
53   }
54 }
55 };
56 </script>
57 <link rel="stylesheet" href="assets/launchpad.css">
58 <script src="https://ui5.sap.com/1.129.2/test-resources/sap/ushell/bootstrap/sandbox.js"></script>
59 <script
60   src="https://ui5.sap.com/1.129.2/resources/sap-ui-core.js"
61   data-sap-ui-libs="sap.m, sap.ushell, sap.ffe.templates"
62   data-sap-ui-compatVersion="edge"
63   data-sap-ui-theme="sap_horizon"
64   data-sap-ui-frameOptions="allow"
65   data-sap-ui-bindingSyntax="complex"
66   data-sap-ui-xx-viewCache="false"
67 ></script>
68 <script>
69   sap.ui.getCore().attachInit(function() {
70     sap.ushell.Container.createRenderer("fiori2", true).then(function(oRenderer) {
71       oRenderer.placeAt('content');
72     });
73   });
74 </script>
75 </head>
76 <body class="sapUiBody" id="content"></body>
77 </html>

```

Image 5.4.5.2: Screenshot of code from launchpadPage.html

Code Breakdown

Purpose

- Implements a launchpad-like UI using SAP Fiori for navigating between multiple applications (Customers, Invoices, Orders, Products).

Header Section

1. **Meta Information**

- Defines charset, viewport, and compatibility for responsive and modern rendering.

2. **Launchpad Configuration**

- Configures sap-ushell-config for Fiori launchpad:
 - Sets fiori2 as the default renderer.
 - Lists applications (Customers, Invoices, Orders, Products) with their metadata, including URLs and SAPUI5 components.

3. **Scripts and Libraries**

- Loads sandbox.js for testing Fiori launchpad in a non-productive environment.
- Loads the SAPUI5 core library (sap-ui-core.js) and essential libraries (sap.m, sap.ushell, sap.fe.templates).
- Uses the sap_horizon theme for styling.

4. **Styles**

- Links launchpad.css for custom styling.

Body Section

1. **Launchpad Rendering**

- The <body> tag is given an ID (content) for rendering the Fiori launchpad.
- sap.ushell.Container.createRenderer initializes the fiori2 renderer and places it in the content div.

Key Features

- **Fiori Launchpad:** Provides a navigation hub for multiple SAPUI5 applications.
- **Application Configuration:** Predefines each application with a title, description, URL, and component metadata.
- **Responsive UI:** Ensures compatibility with modern browsers and devices.

5.5 Sales Dashboard

The purpose of this sales dashboard is to show the interface we built in SAP Analytics Cloud (SAC) using our ERP system data. The dashboard is designed for analyzing data from our ERP system and the primary aims to provide insights into sales performance, forecast future trends, and enable users to filter data dynamically.

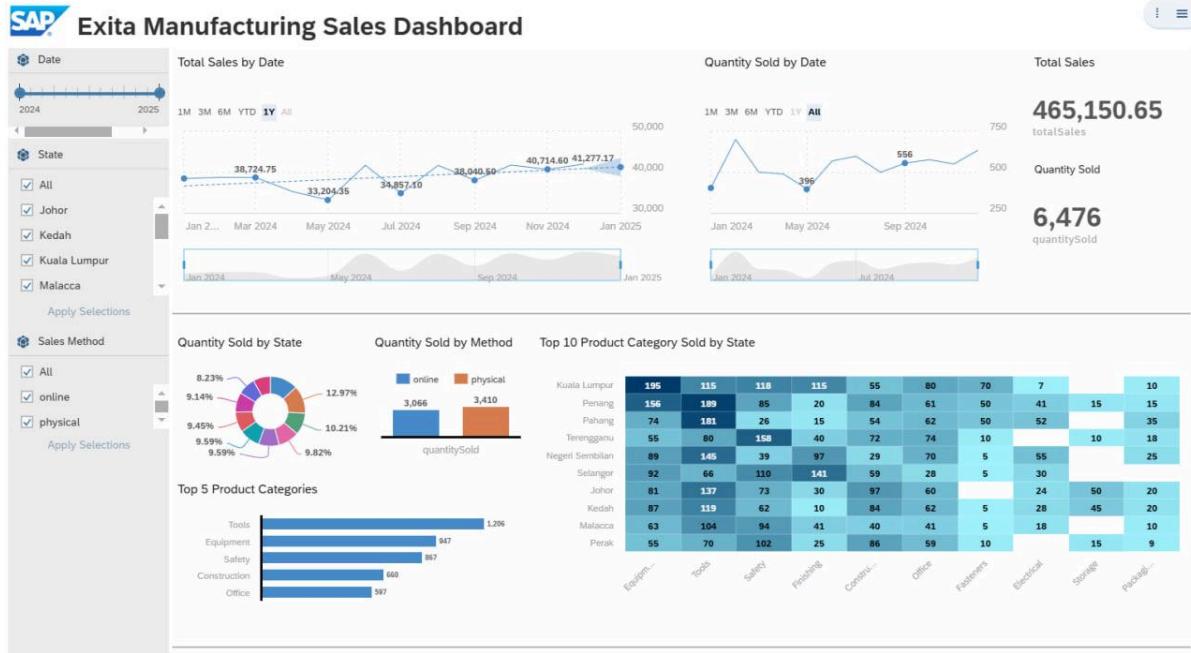


Image 5.5.1: Default sales dashboard



Image 5.5.2: Total sales by date with sales forecasting

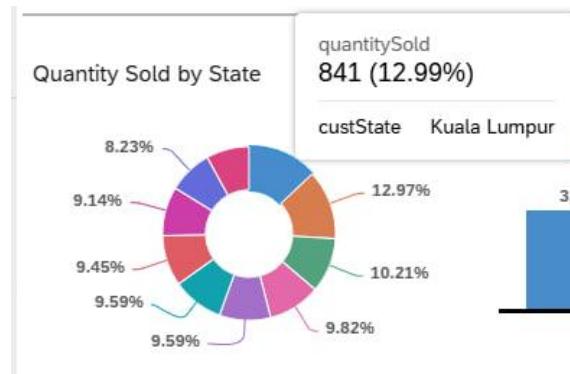


Image 5.5.2: Quantity product sold by state

Interface Breakdown

Purpose

The dashboard serves as a centralized platform for stakeholders to visualize and analyze sales data, facilitating informed decision-making and strategic planning. This primary aim is to provide visualized analytics for the user to seamlessly plan their decision-making.

Filtering Options (Left side)

The left side of the dashboard has filtering options for the user to allow them customize their data view and visualize dynamic analysis based on:

- Date Range: Enables use to filter data and select specific timeframes to analyze sales performance over different periods.
- State: Enables use to filter data to view sales data specific to particular states in Malaysia.
- Sales Method: Enables use to filter data based on the sales methods, such as physical sales or online transactions.

Body of the dashboard (Right side)

- **Total Sales by Date (Timeline Graph with Forecasting)**

This visualization shows the total sales amounts over time of the company in a timeline graph. The graph includes forecasting capabilities with upper confidence bound and lower confidence bound which will allow users to analyze future sales trends based on historical data.

- **Quantity Sold by Date (Timeline Graph)**

This visualization shows the timeline graph illustrates the quantity of products sold over specific dates which will provide insight into sales volume trends alongside total sales amounts.

- **Total Sales**

This displays the overall sales amount across the selected filters by offering a quick numerical snapshot of total revenue generated.

- **Quantity Sold**

Similar to total sales, this shows the overall quantity of products sold by quick numerical snapshot which will help users understand overall sales volume.

- **Quantity Sold by State (Donut Chart)**

A visually engaging donut chart presents the quantity of products sold by different states which allow users to seamlessly identify states with high and low sales performance.

- **Quantity Sold by Method (Bar Chart)**

This bar chart compares the quantities sold through different sales methods between online and physical and highlight trends in customer purchasing preferences.

- **Top 5 Product Categories (Bar Chart)**

This bar chart displays the five most popular product categories of the company based on sales volume. This visualization helps stakeholders identify key product lines that concur the sales revenue and help company decision-making.

- **Top 10 Product Categories by State (Heat Map)**

A heat map illustrates the top ten product categories sold in each state, showcasing the quantity of products sold. This visualization provides a geographic perspective on product performance, enabling targeted marketing strategies. The box coloured with dark blue shows the highest quantity purchased by the customer.

5.5.1 Create Dashboard Page

SAP Analytics Cloud offers robust features that allow users to create interactive stories and dashboards, facilitating in-depth analytics for organizations. These tools enable businesses to visualize data effectively, derive insights, and make informed decisions.

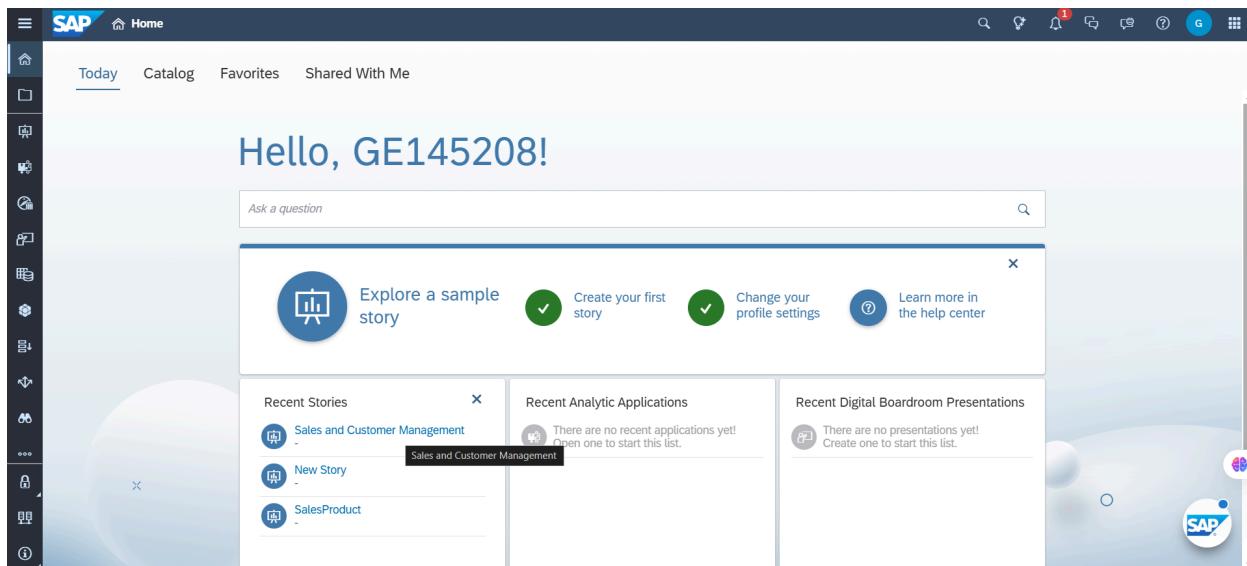


Image 5.5.1.1 : Main page of SAP Analytics Cloud

When users open SAP Analytics Cloud they see the main platform window. Users can find all platform features in one central area. To start using these functions users need to click the button marked “Create your first story”. The system will redirect users step-by-step in setting up their first analytical story and adding data visualizations alongside meaningful content.

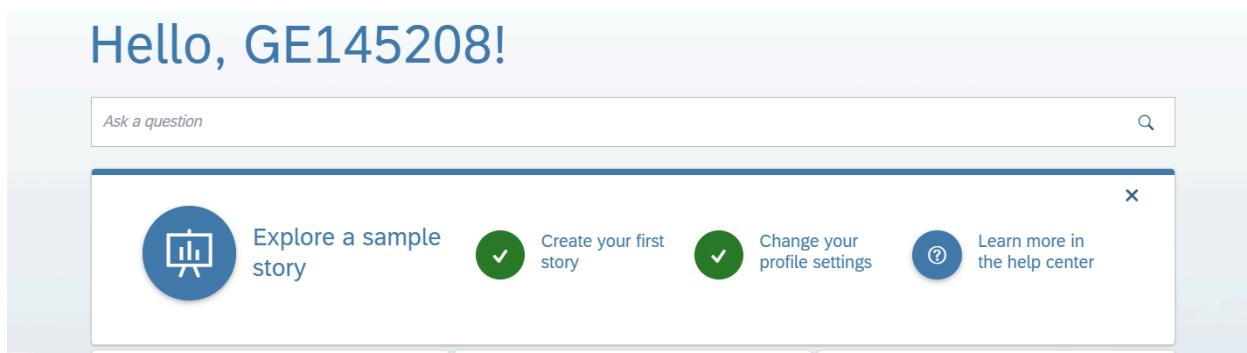
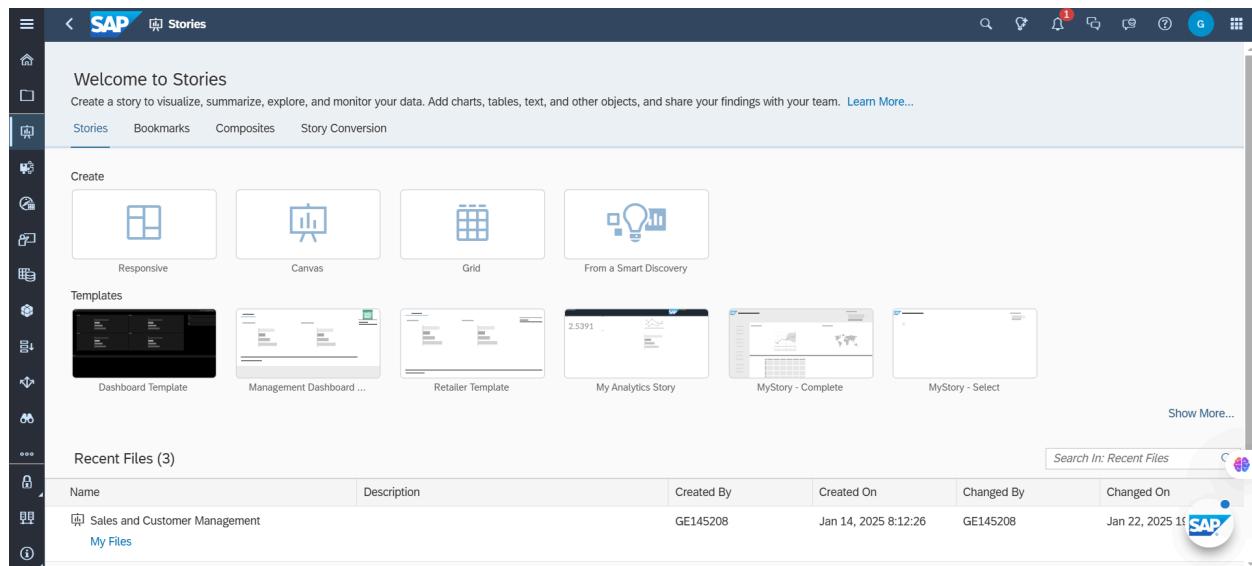


Image 5.5.1.2 : Features of the SAP Analytics Cloud

In the creation section of the platform, users can view multiple options tailored to their specific needs for creating their own dashboard. The SAP Analytics Cloud provides flexibility by allowing users to either start building their dashboard from scratch or choose to modify an existing dashboard template.



Welcome to Stories

Create a story to visualize, summarize, explore, and monitor your data. Add charts, tables, text, and other objects, and share your findings with your team. [Learn More...](#)

Stories Bookmarks Composites Story Conversion

Create

- Responsive
- Canvas
- Grid
- From a Smart Discovery

Templates

- Dashboard Template
- Management Dashboard ...
- Retailer Template
- 2.5391
- My Analytics Story
- MyStory - Complete
- MyStory - Select

Show More...

Recent Files (3)

Name	Description	Created By	Created On	Changed By	Changed On
Sales and Customer Management		GE145208	Jan 14, 2025 8:12:26	GE145208	Jan 22, 2025 10:15:00
My Files					

Search In: Recent Files

SAP

Image 5.5.1.3 : Options for creating story in SAP Analytics Cloud

5.5.2 Database of Dashboard

Image below shows the interface for setting up the dataset for building the dashboard interface. There is a option by connecting the SAP Analytics Cloud with SAP HANA databases. However due to the limitation of the trial, we used importing existing csv file into the dataset.

The screenshot shows the SAP Datasets interface. On the left, there is a sidebar with various icons for navigation. The main area has a title 'Welcome to Datasets' and a sub-section 'Create' with two options: 'From a CSV or Excel File' and 'From a Data Source'. Below this, there is a section for 'Recent Files (5)' with a table showing five entries. The table columns are 'Name', 'Description', 'Created By', 'Created On', 'Changed By', and 'Changed On'. The entries are: 'SalesCust' (Description: 'My Files'), 'SalesCustESDM' (Description: 'My Files'), 'fact_sales' (Description: 'My Files'), and 'dim_product' (Description: 'My Files'). The interface has a dark header with the SAP logo and a search bar.

Image 5.5.2.1 : Dataset interface

We can modify the data types of the attributes in our dataset and designate the measures and dimensions for further analytical processing.

The screenshot shows a detailed view of the 'fact_sales' dataset. On the left, there is a sidebar with various icons. The main area has a title 'Datasets | fact_sales' and a 'Create Transform' section with a table of data. The table has columns: salesId, date, custId, productId, quantitySold, totalSales, and salesMethod. The data shows 16 rows of sales records. To the right of the table is a 'Dataset Overview' panel. This panel shows the dataset name 'fact_sales', the number of rows (500), and the number of columns (7). It also shows the 'Output' section with measures: 'quantitySold' and 'totalSales', both set to 'SUM'. Below this is the 'Dimensions' section with dimensions: 'salesId', 'date', 'custId', 'productId', and 'salesMethod'. The interface has a dark header with the SAP logo and a search bar.

Image 5.5.1.2 : Sample dataset view of fact_sales

5.5.3 Sales and Management Dashboard

This is the main interface of the dashboard that we have created based on the attributes and the requirements of this ERP system.

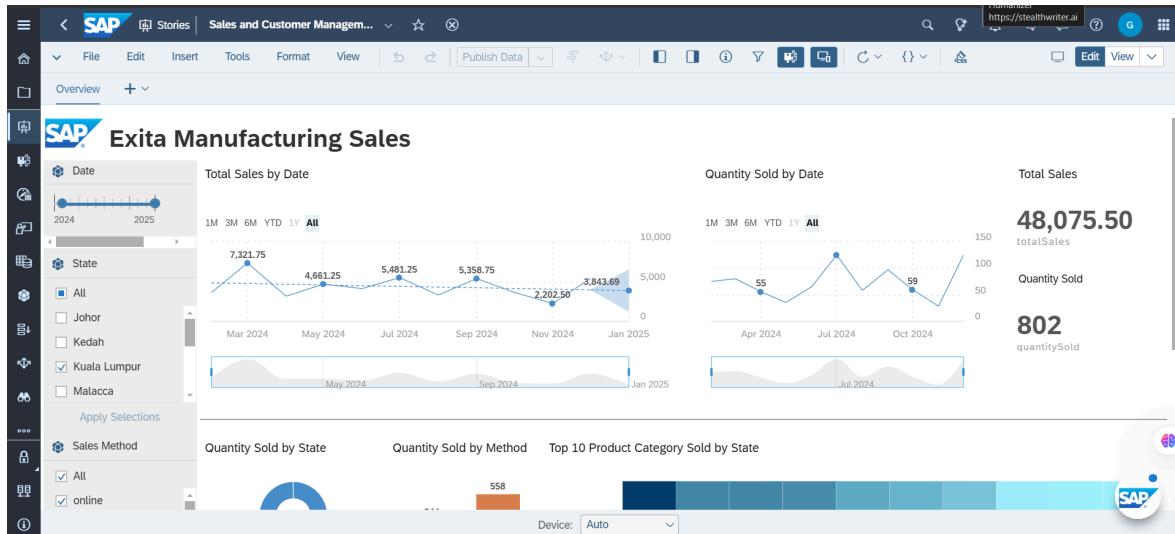


Image 5.5.3.1 : Dashboard interface

When users modify their sales method selection the system updates all data visualizations including total sales, number of items sold and how products perform. Image below shows the example of the dashboard where the data is filtered for sales of Kuala Lumpur, Johor, Kedah and Malacca and specifically visualizes data for physical sales method.



Image 5.5.1.2 : Dashboard of Kuala Lumpur, Johor, Kedah and Malacca with physical sales method

- **Conclusion**

This interactive design lets users work with data in real time as the user has the ability and options to see direct results instantly. This sales dashboard in SAP Analytics Cloud effectively combines various data visualizations to provide a clear and actionable overview of sales performance and help companies' decision making process. By utilizing filters and dynamic charts, users can delve and do more research deep into the data, uncovering valuable insights and solves potential problems that support strategic business decisions and forecasting efforts.

5.6 System Testing with Input and Output Screen

Home Screen

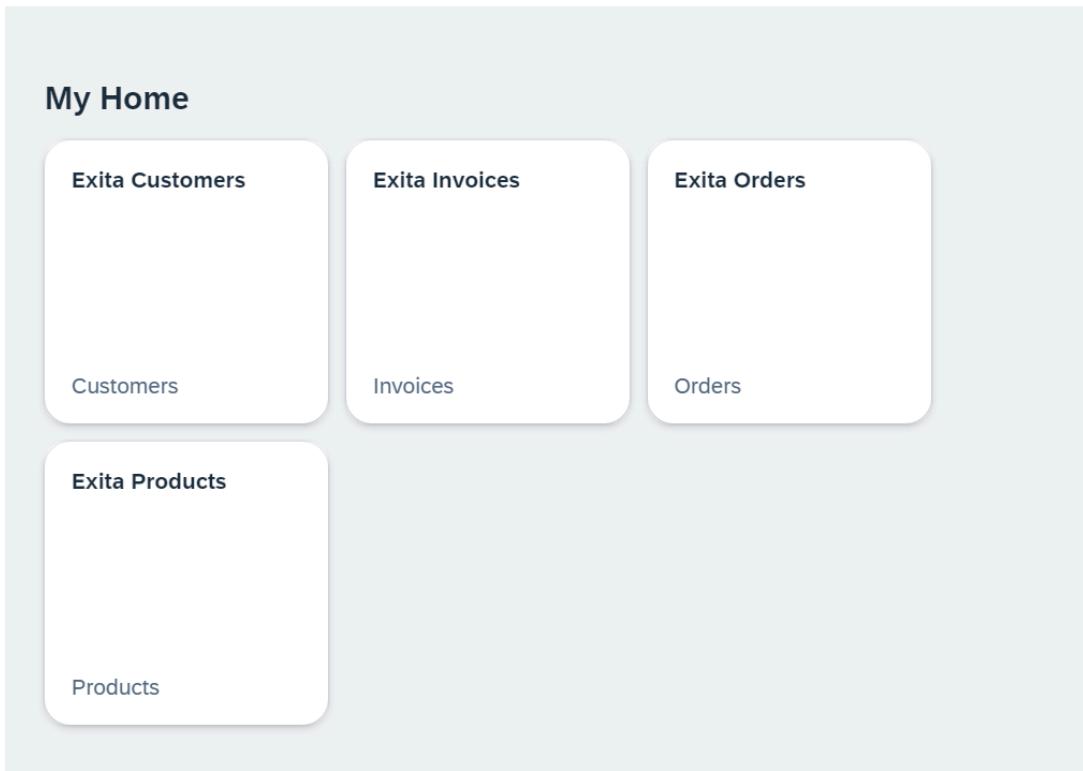
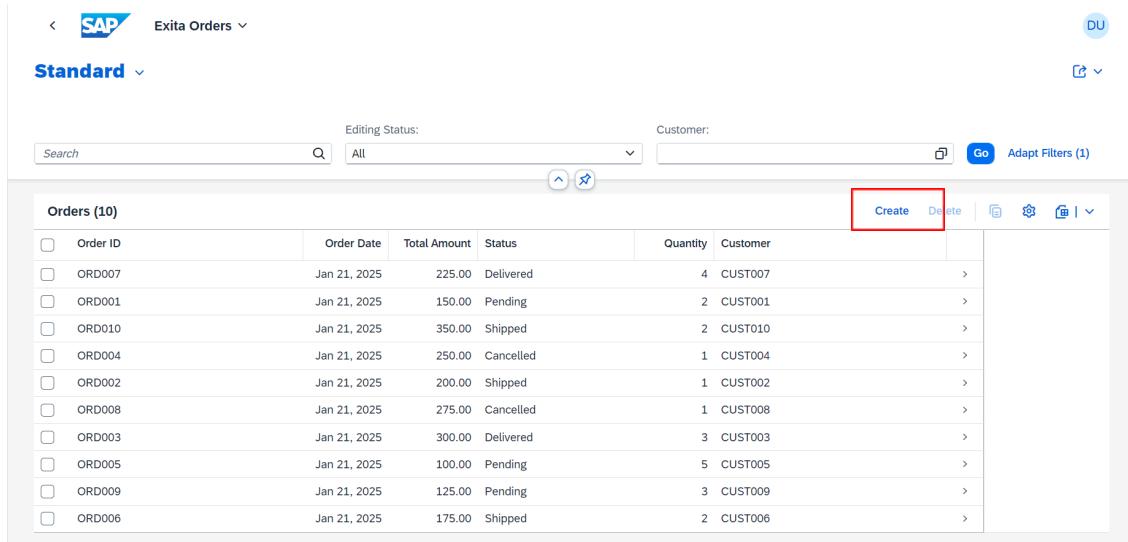


Image 5.6.1: Screenshot of system home screen

5.6.1 Track Customers Order

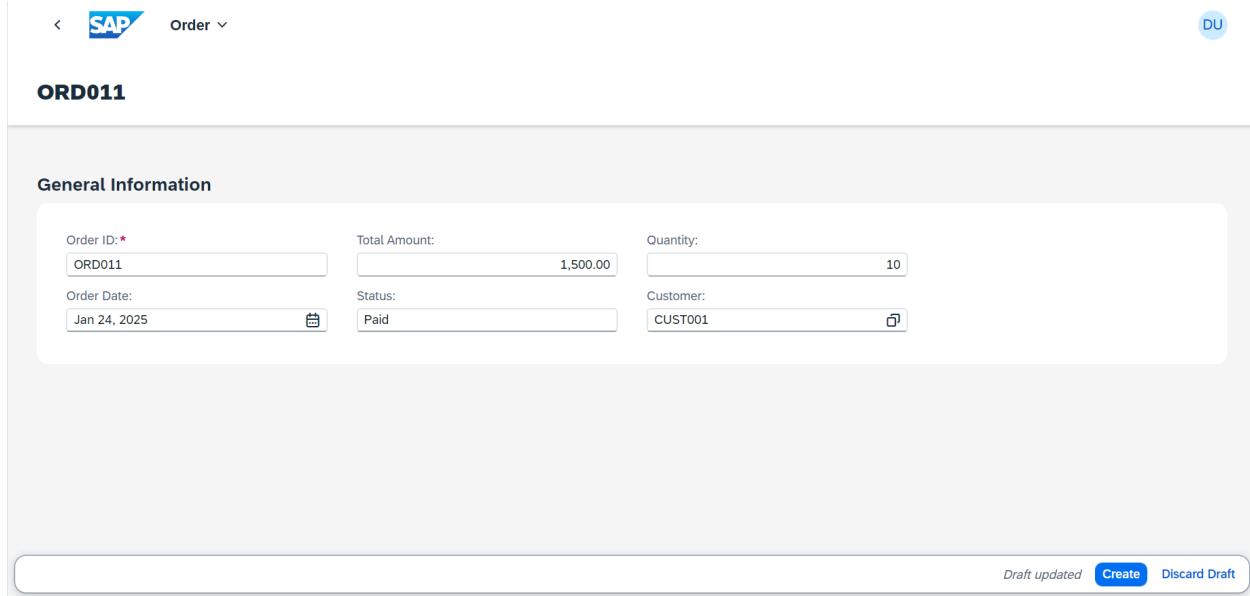
Create Order



Orders (10)						Create	Delete	Edit	Logistics	Details	Print	Help	Close
<input type="checkbox"/>	Order ID	Order Date	Total Amount	Status	Quantity	Customer							
<input type="checkbox"/>	ORD007	Jan 21, 2025	225.00	Delivered	4	CUST007	>						
<input type="checkbox"/>	ORD001	Jan 21, 2025	150.00	Pending	2	CUST001	>						
<input type="checkbox"/>	ORD010	Jan 21, 2025	350.00	Shipped	2	CUST010	>						
<input type="checkbox"/>	ORD004	Jan 21, 2025	250.00	Cancelled	1	CUST004	>						
<input type="checkbox"/>	ORD002	Jan 21, 2025	200.00	Shipped	1	CUST002	>						
<input type="checkbox"/>	ORD008	Jan 21, 2025	275.00	Cancelled	1	CUST008	>						
<input type="checkbox"/>	ORD003	Jan 21, 2025	300.00	Delivered	3	CUST003	>						
<input type="checkbox"/>	ORD005	Jan 21, 2025	100.00	Pending	5	CUST005	>						
<input type="checkbox"/>	ORD009	Jan 21, 2025	125.00	Pending	3	CUST009	>						
<input type="checkbox"/>	ORD006	Jan 21, 2025	175.00	Shipped	2	CUST006	>						

Image 5.6.1.1: Screenshot of create order

List of Orders. Click on "Create" to add a new order.



ORD011

General Information

Order ID:*	Total Amount:	Quantity:
ORD011	1,500.00	10
Order Date:	Status:	Customer:
Jan 24, 2025	Paid	CUST001

Draft updated [Create](#) [Discard Draft](#)

Image 5.6.1.2: Screenshot of order general information

Input Order details and click on "Create".

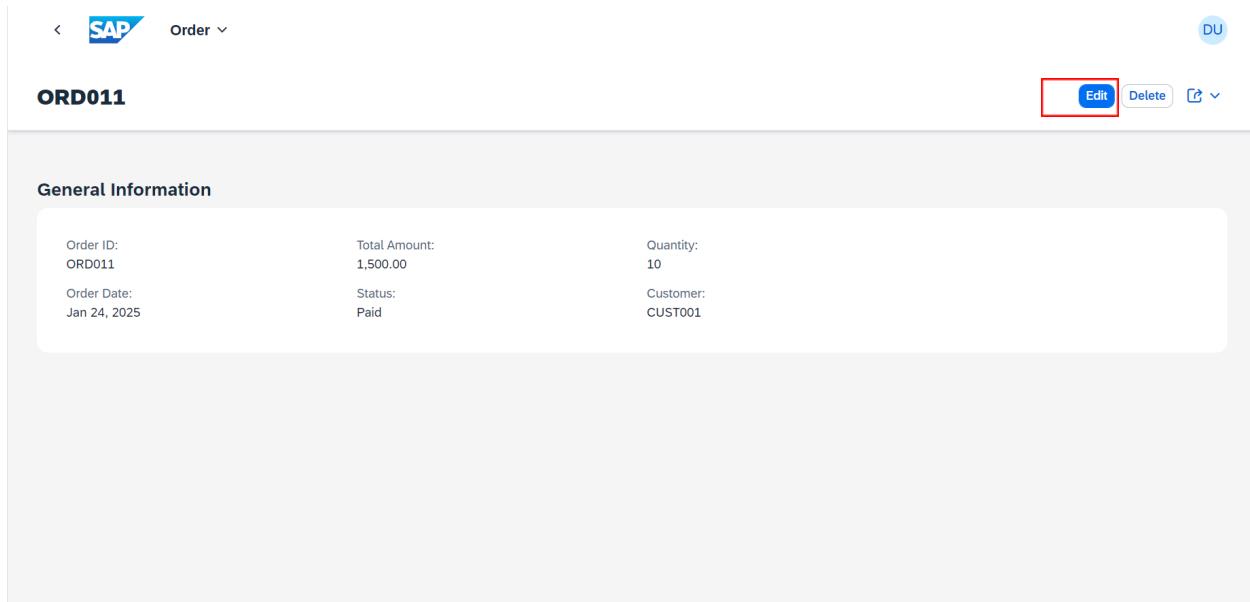


Image 5.6.1.3: Screenshot of generated order general information

New Order successfully created.

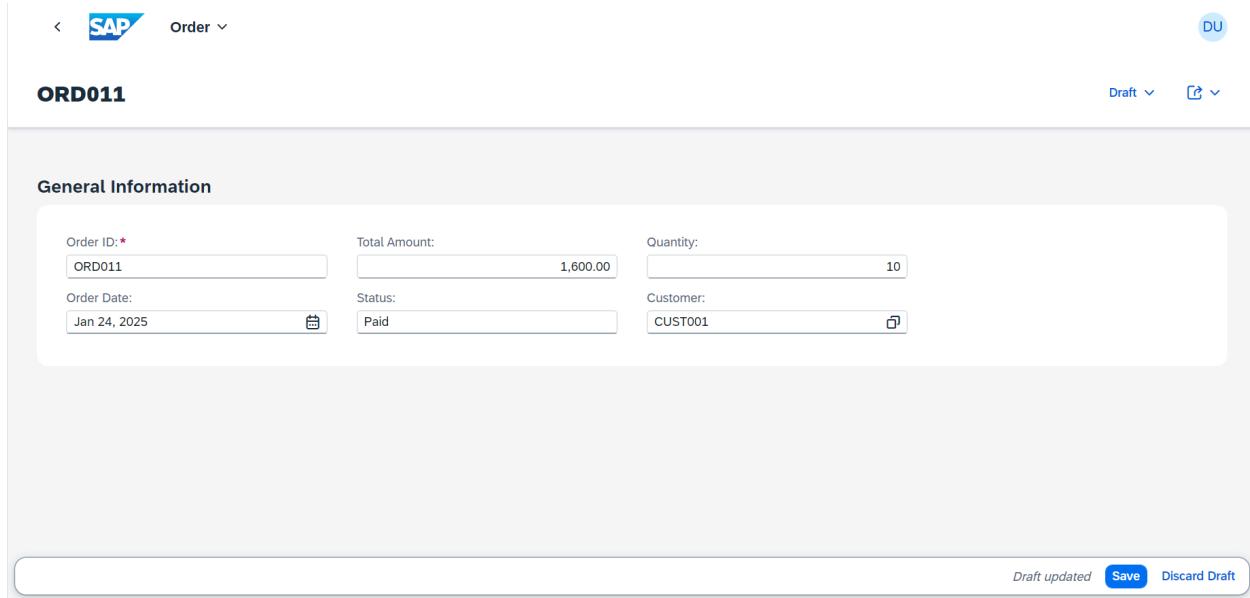
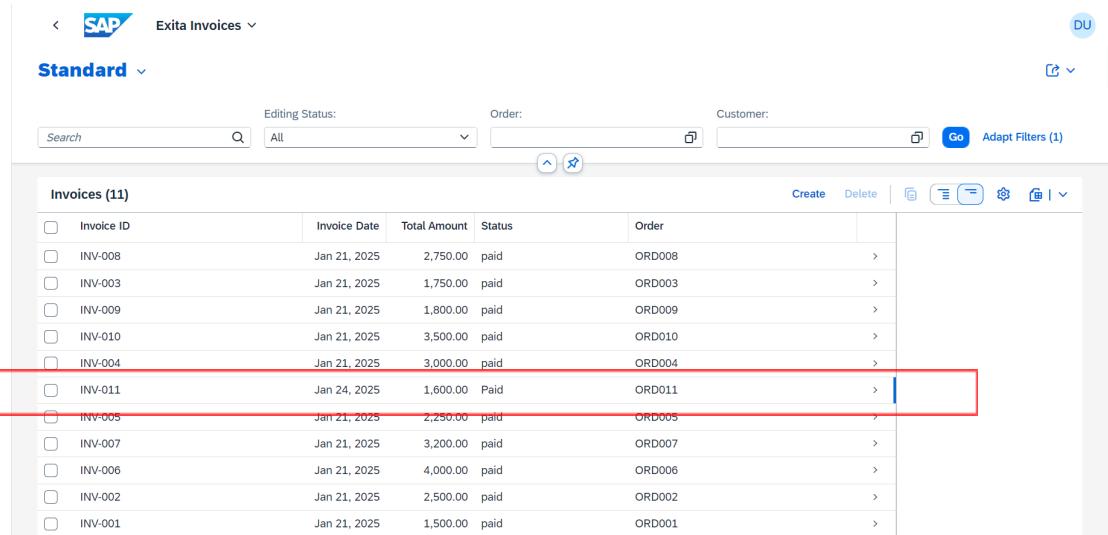


Image 5.6.1.2: Screenshot of modifying generated order general information

To edit an order you have to click on the “edit” button. Here, the total amount is edited.

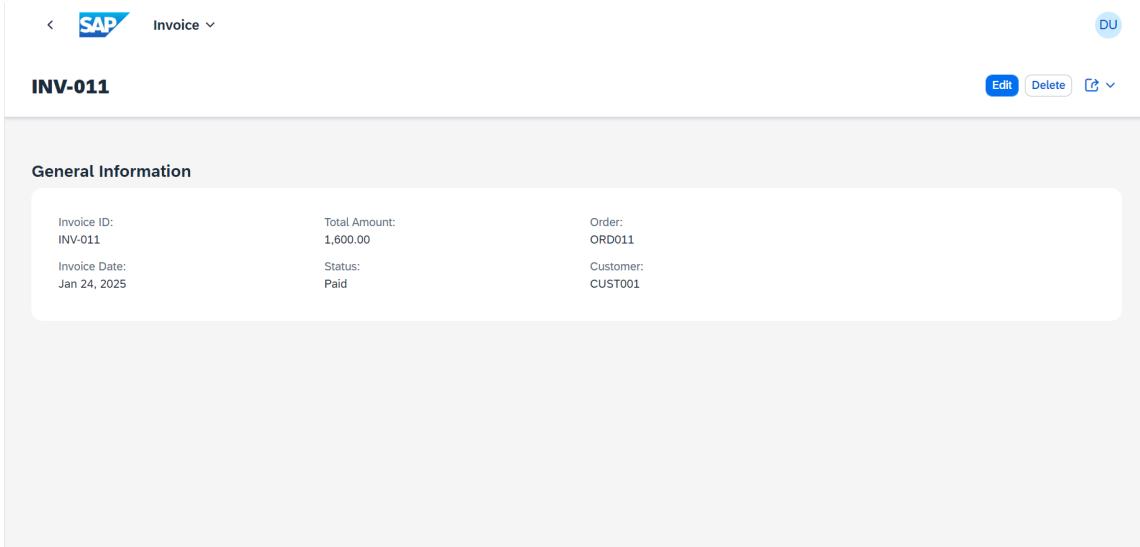
An order will go through many status like pending,paid,shipping and delivered. When the order status is “paid” the system will be triggered to generate a new invoice for that order.

5.6.2 Automate Sales Invoice



Invoice ID	Invoice Date	Total Amount	Status	Order
INV-008	Jan 21, 2025	2,750.00	paid	ORD008
INV-003	Jan 21, 2025	1,750.00	paid	ORD003
INV-009	Jan 21, 2025	1,800.00	paid	ORD009
INV-010	Jan 21, 2025	3,500.00	paid	ORD010
INV-004	Jan 21, 2025	3,000.00	paid	ORD004
INV-011	Jan 24, 2025	1,600.00	Paid	ORD011
INV-005	Jan 21, 2025	2,250.00	paid	ORD005
INV-007	Jan 21, 2025	3,200.00	paid	ORD007
INV-006	Jan 21, 2025	4,000.00	paid	ORD006
INV-002	Jan 21, 2025	2,500.00	paid	ORD002
INV-001	Jan 21, 2025	1,500.00	paid	ORD001

Image 5.6.2.1: Screenshot of automate sales invoice



General Information		
Invoice ID: INV-011	Total Amount: 1,600.00	Order: ORD011
Invoice Date: Jan 24, 2025	Status: Paid	Customer: CUST001

Image 5.6.2.2: Screenshot of general information of selected automate sales invoice

When the status of order is “paid” during creation or updation of an order automatically an invoice is generated.

5.6.3 Manage Customer Profile

Create Customer Profile

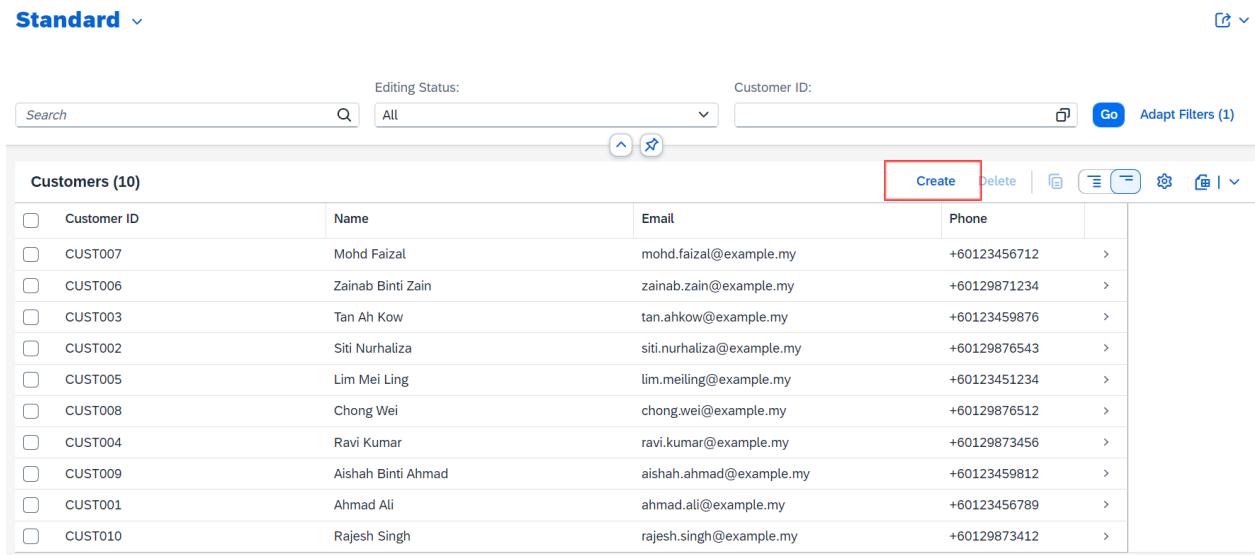


Image 5.6.3.1: Screenshot of create or edit customer profile

Click on “Create” to add new customer profile

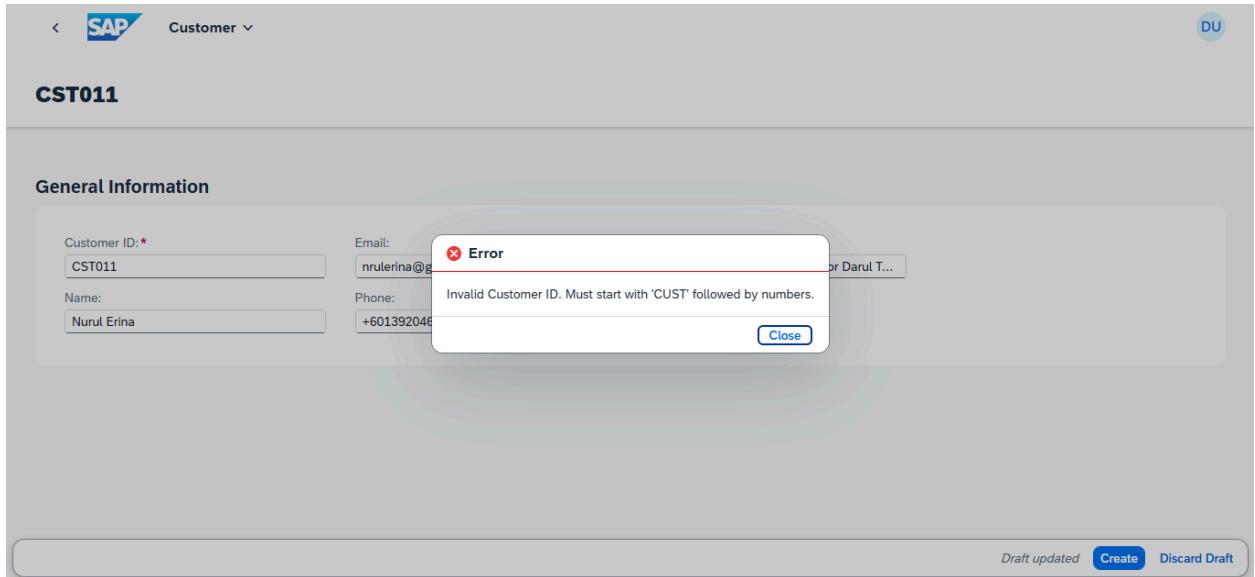


Image 5.6.3.2: Screenshot of error during create or edit customer profile

- The system has triggered an **error message**: "**Invalid Customer ID. Must start with 'CUST' followed by numbers.**"
- The provided Customer ID is "CST011", which does not comply with the required format.
- The error suggests that the Customer ID must strictly follow the format starting with '**CUST**' followed by a numerical sequence, indicating a validation rule in place for IDs.

This enforces a consistent naming convention for Customer IDs.

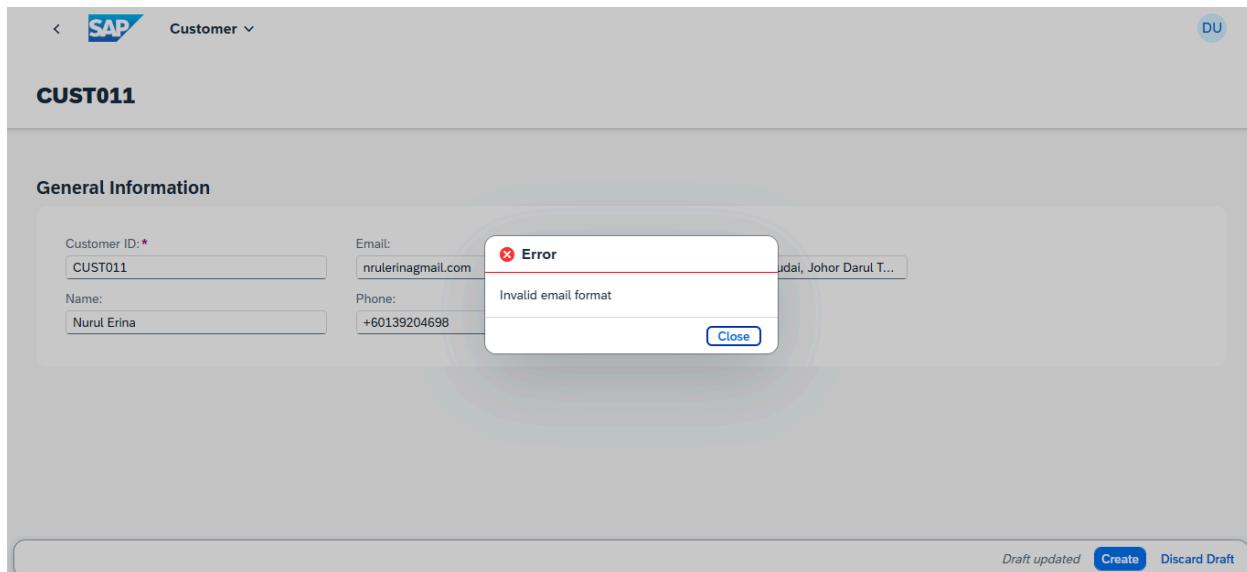


Image 5.6.3.3: Screenshot of error during create or edit customer profile

- The system shows an **error message**: "**Invalid email format.**"
- This error occurs because the entered email, "nuleinagmail.com," is missing the '@' symbol, which is a mandatory part of a valid email address format (e.g., "name@example.com").

The system enforces proper email formatting to ensure data accuracy and reliability.

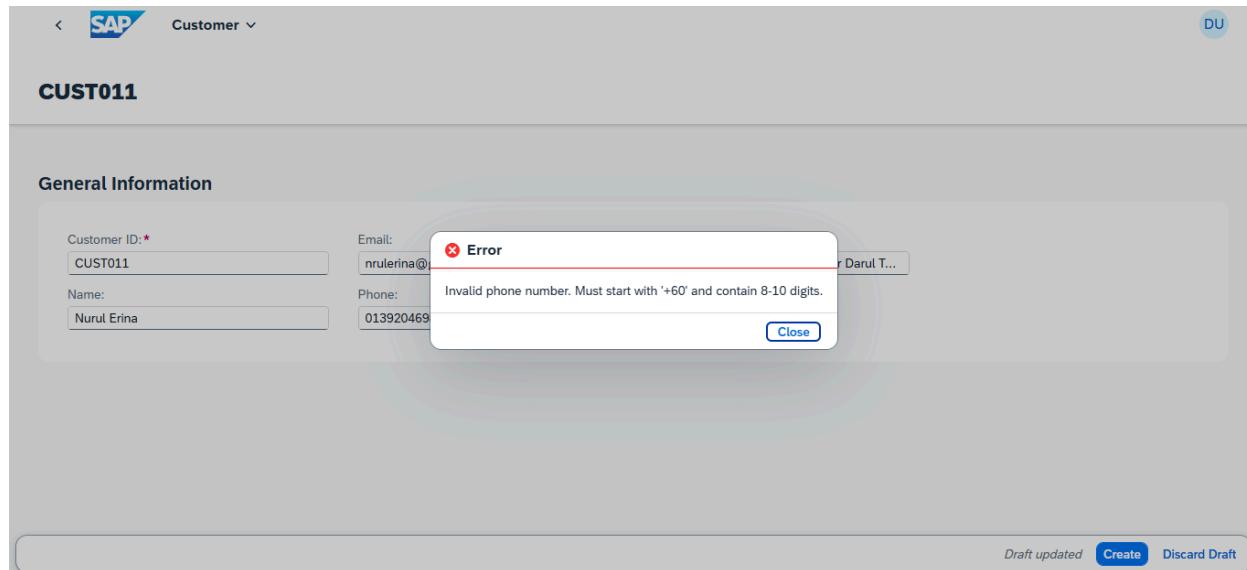


Image 5.6.3.4: Screenshot of error during create or edit customer profile

- The system has triggered an error message: **"Invalid phone number. Must start with '+60' and contain 8-10 digits."**
- The provided phone number is **"013920459"**, which does not comply with the required format.
- The error suggests that the phone number must strictly follow the format starting with **'+60'** (the country code for Malaysia) followed by a numerical sequence containing **8 to 10 digits**, indicating a validation rule in place for phone numbers.

This enforces a consistent and correct format for entering phone numbers.

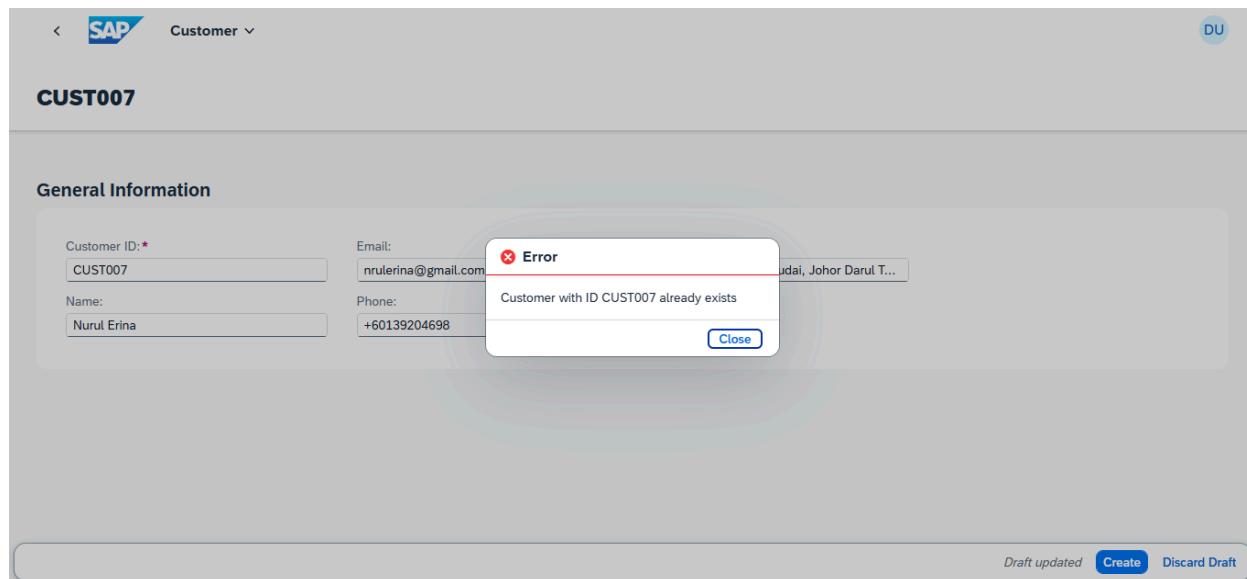


Image 5.6.3.5: Screenshot of error during create or edit customer profile

- The system has triggered an error message: "Customer with ID CUST007 already exists."
- The provided Customer ID, "CUST007", matches an existing record in the database.
- This error suggests that Customer IDs must be unique to ensure no duplicate entries are created.

The validation rule enforces data integrity by preventing the creation of multiple records with the same identifier.

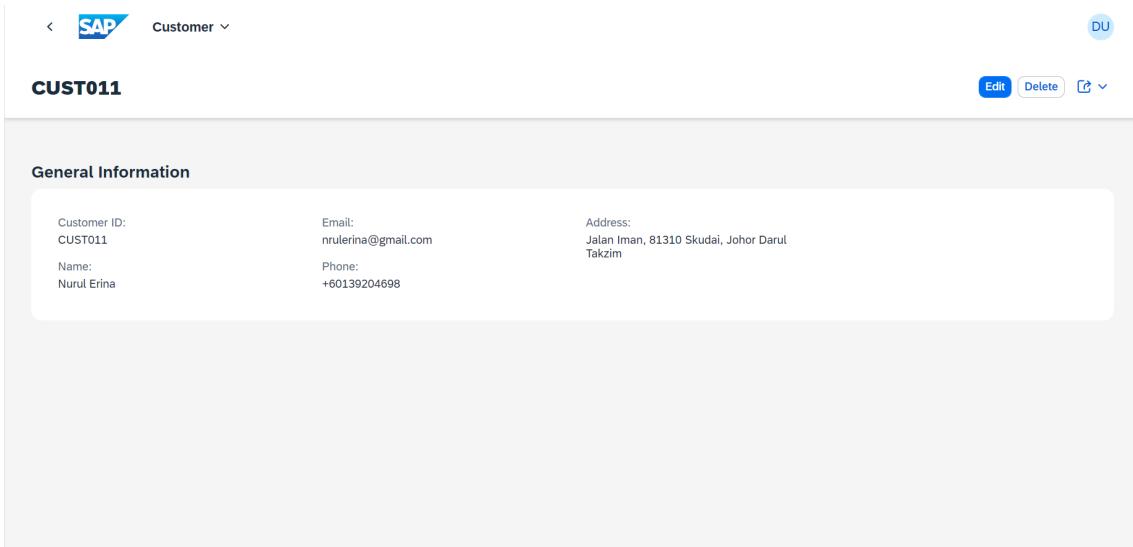


Image 5.6.3.6: Screenshot of successfully created customer profile

The customer record has been successfully created with the following details:

- Customer ID: CUST011
- Name: Nurul Erina
- Email: nrulerina@gmail.com
- Phone: +60139204698
- Address: Jalan Iman, 81310 Skudai, Johor Darul Takzim

This indicates that the input data complies with all validation rules, and the system has successfully processed the creation of the new customer entry.

<input type="checkbox"/>	Customer ID	Name	Email	Phone	>
<input type="checkbox"/>	CUST007	Mohd Faizal	mohd.faizal@example.my	+60123456712	>
<input type="checkbox"/>	CUST006	Zainab Binti Zain	zainab.zain@example.my	+60129871234	>
<input type="checkbox"/>	CUST003	Tan Ah Kow	tan.ahkow@example.my	+60123459876	>
<input type="checkbox"/>	CUST002	Siti Nurhaliza	siti.nurhaliza@example.my	+60129876543	>
<input type="checkbox"/>	CUST005	Lim Mei Ling	lim.meiling@example.my	+60123451234	>
<input type="checkbox"/>	CUST008	Chong Wei	chong.wei@example.my	+60129876512	>
<input type="checkbox"/>	CUST004	Ravi Kumar	ravi.kumar@example.my	+60129873456	>
<input type="checkbox"/>	CUST009	Aishah Binti Ahmad	aishah.ahmad@example.my	+60123459812	>
<input type="checkbox"/>	CUST001	Ahmad Ali	ahmad.ali@example.my	+60123456789	>
<input type="checkbox"/>	CUST011	Nurul Erina	nrulerina@gmail.com	+60139204698	>
<input type="checkbox"/>	CUST010	Rajesh Singh	rajesh.singh@example.my	+60129873412	>

Image 5.6.3.7: Screenshot of list of customers

Updated Customer Profiles List with new customer

Edit Customer

CUST011

General Information

Customer ID:*	Email:	Address:
CUST011	nrulerina@gmail.com	Jalan Nusaria, 81550 Gelang Patah, Joh...
Name:	Phone:	
Nurul Erina	+60139204698	

Draft updated **Save** **Discard Draft**

Image 5.6.3.8: Screenshot of edit customer profile

Address is edited.

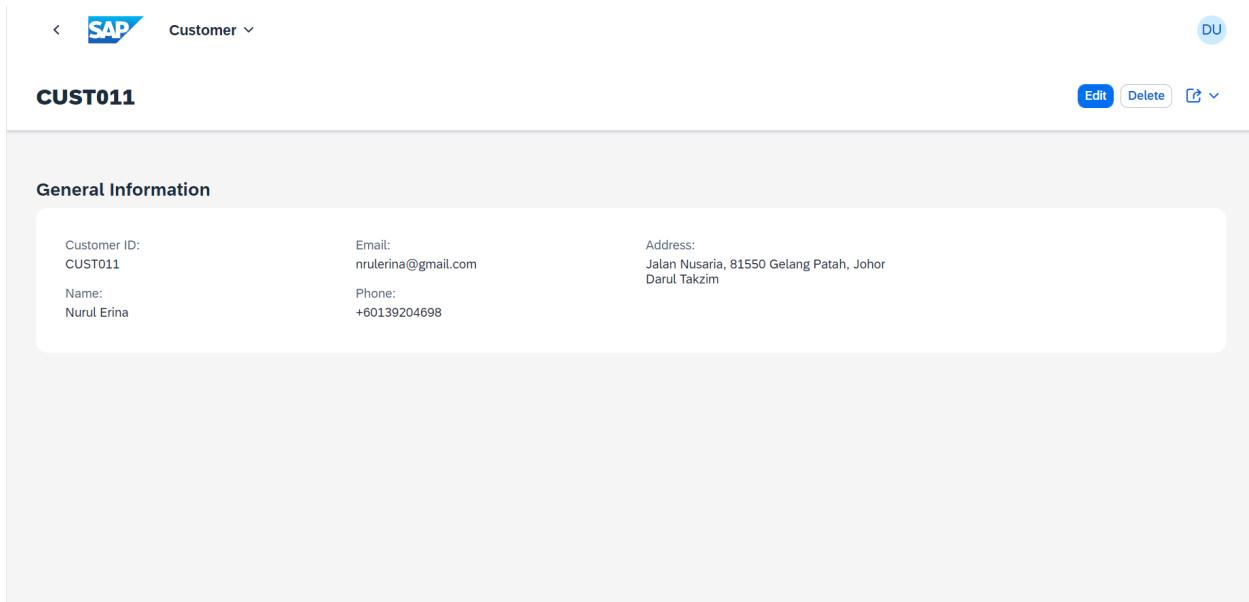


Image 5.6.3.9: Screenshot of successfully edited customer profile

Edited Customer Profile is saved.

Delete Customer

Customers (11)				
<input type="checkbox"/>	Customer ID	Name	Email	Phone
<input type="checkbox"/>	CUST007	Mohd Faizal	mohd.faizal@example.my	+60123456712 >
<input type="checkbox"/>	CUST006	Zainab Binti Zain	zainab.zain@example.my	+60129871234 >
<input type="checkbox"/>	CUST003	Tan Ah Kow	tan.ahkow@example.my	+60123459876 >
<input type="checkbox"/>	CUST002	Siti Nurhaliza	siti.nurhaliza@example.my	+60129876543 >
<input type="checkbox"/>	CUST005	Lim Mei Ling	lim.meiling@example.my	+60123451234 >
<input type="checkbox"/>	CUST008	Chong Wei	chong.wei@example.my	+60129876512 >
<input type="checkbox"/>	CUST004	Ravi Kumar	ravi.kumar@example.my	+60129873456 >
<input type="checkbox"/>	CUST009	Aishah Binti Ahmad	aishah.ahmad@example.my	+60123459812 >
<input type="checkbox"/>	CUST001	Ahmad Ali	ahmad.ali@example.my	+60123456789 >
<input checked="" type="checkbox"/>	CUST011	Nurul Erina	nrulerina@gmail.com	+60139204698 >
<input type="checkbox"/>	CUST010	Rajesh Singh	rajesh.singh@example.my	+60129873412 >

Image 5.6.3.10: Screenshot of customer list

CUST011 selected and click on 'Delete'

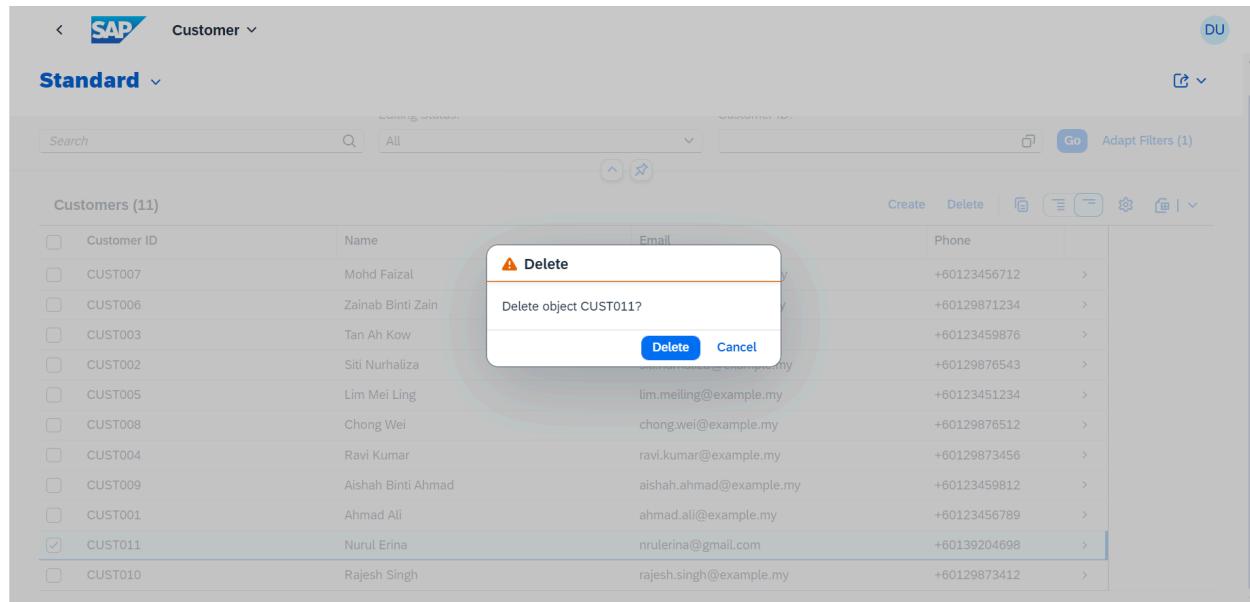


Image 5.6.3.11: Screenshot of confirmation of deleting customer data

The system asks for confirmation to delete the customer profile.

5.6.4 Forecast Sales Trend

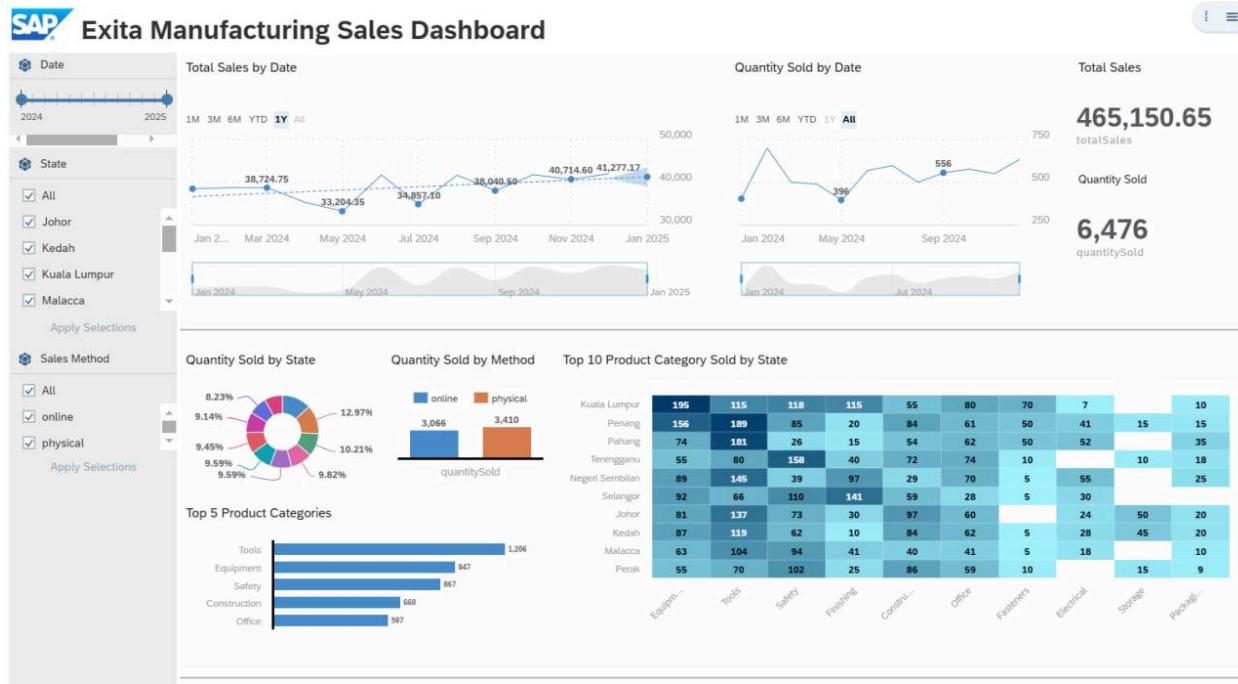


Image 5.6.4.1 : Default dashboard view

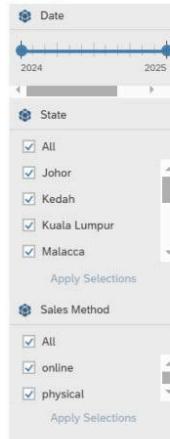


Image 5.6.4.2 : Dashboard filtering option

5.6.4.1 Filtering Data By Date

Users select the desired date of the data that they want to see in visualized form by filtering the 'Date' section at the left side of the dashboard as the image below. This feature allow users to modify the time frame selection to study specific sales patterns and business results.



Image 5.6.4.1.1 : Filtering data by date

When users modify their date selection the system updates all data visualizations including total sales, number of items sold and how products perform. Image below shows the example of the dashboard where the data is filtered for date from July 2024 until January 2025.

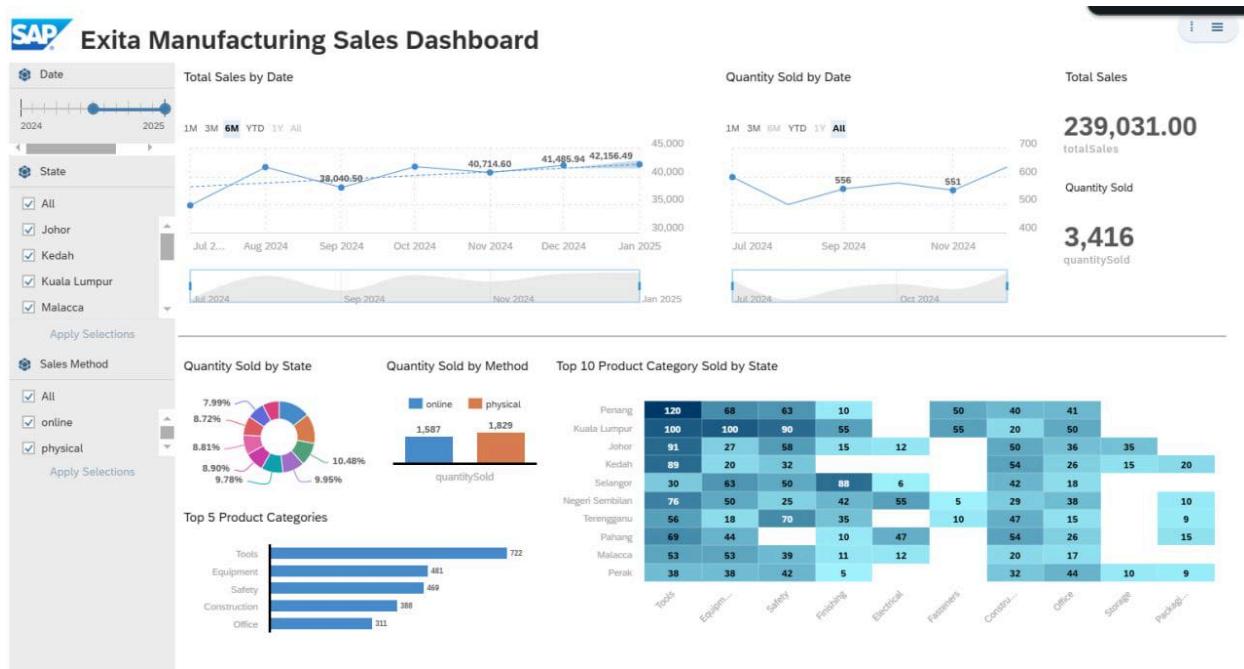


Image 5.6.4.1.2 : Dashboard of data from July 2024 until January 2025

5.6.4.2 Filtering Data By State

Users select the desired state of the data that they want to see in visualized form by filtering the 'State' section at the left side of the dashboard as the image below. This feature allows users to modify the dashboard by selecting one or multiple states to study specific sales patterns and business results in the specific states.

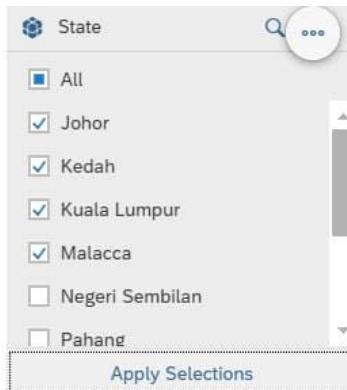


Image 5.6.4.2.1 : Filtering data by state

When users modify their state selection the system updates all data visualizations including total sales, number of items sold and how products perform. Image below shows the example of the dashboard where the data is filtered for sales of Kuala Lumpur, Johor, Kedah and Melacca.

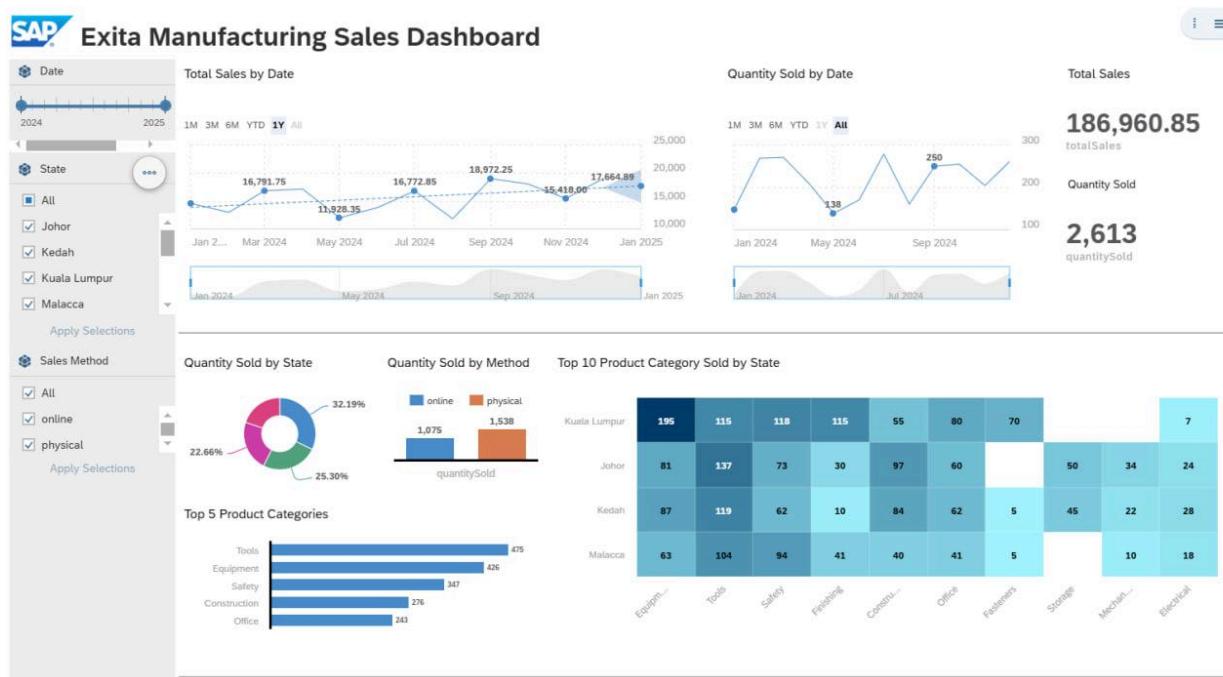


Image 5.6.4.2.2 : Dashboard of data from Kuala Lumpur, Johor, Kedah and Melacca

5.6.4.3 Filtering Data By Sales Method

Users select the desired sales method of the data that they want to see in visualized form by filtering the ‘Sales Method’ section at the left side of the dashboard as the image below and selecting between ‘online’, ‘physical’ or ‘all’. This feature allows users to modify the dashboard by selecting one or multiple sales methods to study specific sales patterns and business results based on the specific sales of the product by customers.

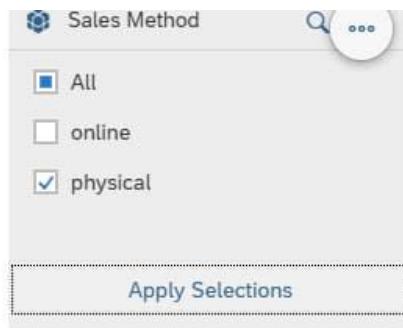


Image 5.6.4.3.1 : Filtering data by sales method

When users modify their sales method selection the system updates all data visualizations including total sales, number of items sold and how products perform. Image below shows the example of the dashboard where the data is filtered for sales of Kuala Lumpur, Johor, Kedah and Malacca and specifically visualizes data for physical sales method.

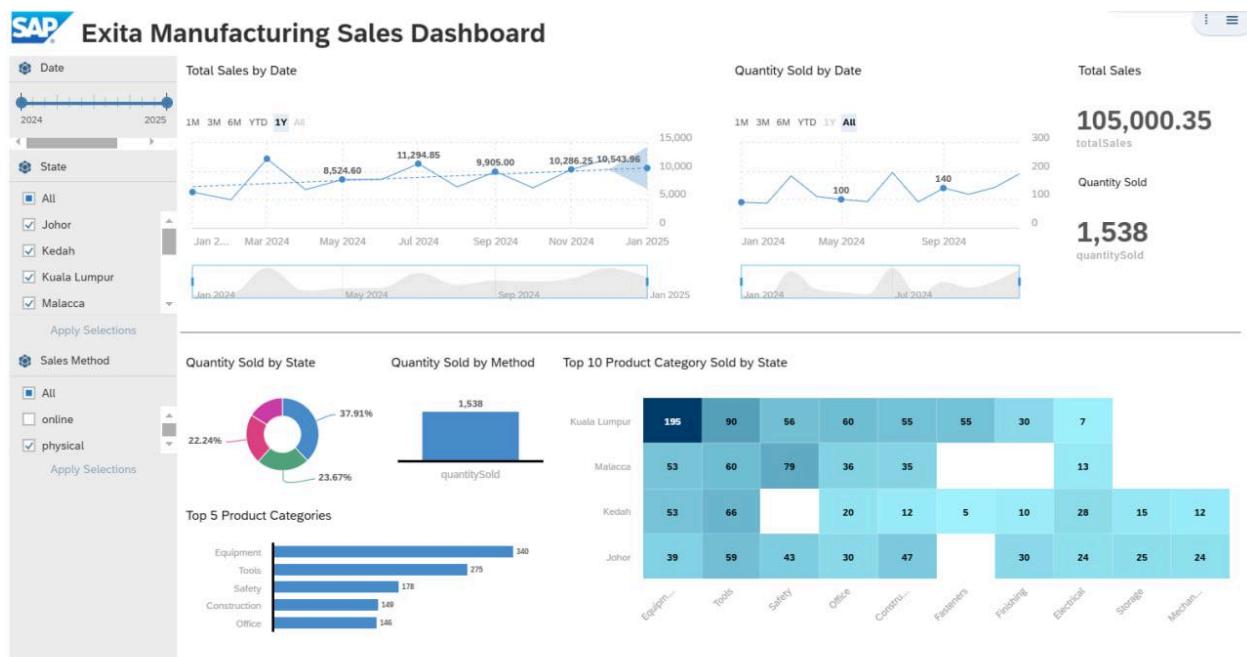


Image 5.6.4.3.2 : Dashboard of Kuala Lumpur, Johor, Kedah and Malacca with physical sales method

5.7 Summary Chapter

The implementation phase transformed the ERP system design into a functional solution using **SAP Build Code**, which operates based on **SAP Business Application Studio (BAS)** in the background. The development process prioritized scalability, efficiency, and user accessibility.

Key accomplishments include:

1. **Data Modeling:**
 - MySQL was used to define the initial data models, which were then configured in SAP Build Code.
 - Sample data was added and validated within the system to simulate real-world scenarios.
2. **Service Definition:**
 - Service endpoints were created to facilitate seamless backend interactions for CRUD operations across key entities like Customers, Products, and Orders.
3. **User Interface Development:**
 - UI applications were designed using template-based responsive layouts, ensuring usability across devices.
4. **Logic Implementation:**
 - Business logic was defined using the Graphical Modeler, enabling automated workflows such as order processing and real-time data updates.
5. **Testing and Validation:**
 - Functional, integration, and UI testing validated the system's ability to meet operational requirements.
 - User feedback was integrated to refine usability and resolve minor inconsistencies.

Challenges Addressed

- Minor data discrepancies during sample data entry were identified and corrected.
- Ensuring seamless alignment of UI and backend logic was achieved through iterative testing.

Final Assessment

The system is now ready for deployment, with all features fully tested and validated. The implementation process demonstrates that SAP Build Code effectively streamlines ERP system development while leveraging the robustness of SAP BAS in the background.

Chapter 6: Conclusion

6.1 Introduction

This chapter reflects on the design and partial implementation of the ERP system for Exita Manufacturing. The primary focus of the course was on enterprise architecture and system architecture design, ensuring that all modules would work seamlessly together. Due to the complexity of building a fully integrated enterprise system—typically undertaken by very large organizations—only one module was implemented within the scope of the project. The aim was to demonstrate the feasibility of the ERP system and highlight its potential to address the company's inefficiencies. The project centered on key modules: Reporting and Analytics, CRM, SCM, and Financial Management, with the understanding that full development would require significant resources and time.

6.2 System Contribution/Achievement

The project successfully demonstrated several key contributions:

1. Comprehensive Design:

- Designed a scalable ERP system architecture that aligns with Exita Manufacturing's goals, ensuring the integration of modules like CRM and SCM.
- Emphasized the seamless interaction between modules to enhance system efficiency and data consistency.

2. Effective Implementation:

- Developed and tested the Sales and Customer Management System, which included functionalities such as order tracking, customer profile management, automatic invoice generation, and a predictive sales dashboard.
- The system automated key processes like invoice generation and order tracking, reducing manual errors and improving operational efficiency.

3. Improved Processes:

- The implemented CRM module enabled better management of customer data and sales orders, streamlining communication and transaction tracking.
- Predictive sales analytics offered insights into future trends, helping the company plan for demand more accurately.

4. Foundational Model:

- Established a solid framework for scaling the ERP system by laying the groundwork for integrating additional modules (e.g., Financial Management, Reporting, and Analytics). This design ensures future scalability with minimal disruptions to existing operations.

6.3 System Constraint

1. Limited Scope:

- Due to time constraints, only one module (Sales and Customer Management) was developed. The remaining modules, while designed, were not implemented or tested. This limitation hindered the ability to assess the full potential of the integrated ERP system.

2. Resource Limitations:

- The project was developed within the SAP environment, which imposed certain limitations. One key challenge was the inability to retrieve data while filling out information in the system before saving it to the database. Specifically, when attempting to include logic for calculating total amounts and automatic invoice generation, the SAP UI5 framework ignored the logic when triggered. As a result, the functionality had to be adjusted to ensure that data was saved before being processed, limiting flexibility and functionality.

3. Testing Limitations:

- The system was tested in a simulated environment, preventing real-world validation. This limited the ability to gather actual user feedback and evaluate the system's robustness under real operational conditions.

4. Feature Constraints:

- Advanced features, such as a fully integrated predictive sales dashboard, were not included in the system due to limitations in the SAP environment. While the dashboard was created in SAP Analytics Cloud, it remained separate from the main system, limiting its integration and real-time use within the ERP.

6.4 Future Suggestion

To enhance the ERP system and overcome current limitations, the following suggestions are proposed:

1. Complete Module Implementation:

- Develop and integrate the remaining modules, including Reporting and Analytics, SCM, and Financial Management, to achieve the full potential of the ERP system. This will ensure end-to-end operational efficiency for Exita Manufacturing.

2. Enhance Data Handling:

- Address the limitations of the SAP UI5 environment by exploring alternative methods or tools that allow for more dynamic data retrieval and processing. This would enable features such as total amount calculations and real-time invoice generation without the need for saving to the database first.

3. Integrate Predictive Dashboard:

- Fully integrate the predictive sales dashboard from SAP Analytics Cloud into the main ERP system to provide real-time analytics and insights. This will help users make more informed decisions based on live data.

4. Expand Testing Scope:

- Conduct real-world testing with actual users and operational data to validate the system's effectiveness, gather valuable feedback, and identify areas for improvement.

5. Introduce Advanced Features:

- Incorporate advanced technologies such as AI for predictive analytics, IoT for real-time monitoring, and machine learning for automated process improvements. These features will enhance the system's decision-making capabilities and operational insights.

6. Provide Training and Support:

- Develop comprehensive training programs and user-friendly documentation to help employees understand and utilize the ERP system effectively. This will ensure smooth adoption and maximum benefit from the system.

6.5 Summary

In summary, this project successfully addressed the design and partial implementation of an ERP system tailored to Exita Manufacturing's needs. Despite the complexity of developing a fully integrated enterprise system, the project focused on enterprise architecture and system architecture design, ensuring that all modules would work cohesively in the future.

Through the development of the Sales and Customer Management module, the project demonstrated the potential to automate critical processes such as order tracking, customer management, and invoice generation. While constrained by time, resources, and limitations within the SAP environment, the system's design provides a scalable framework for integrating additional modules and features.

Although real-world testing and advanced functionalities like predictive analytics were not fully realized within the project scope, the achievements lay a solid foundation for future development. With further testing, integration, and enhancements, this ERP system has the potential to transform Exita Manufacturing's operations, driving long-term growth and competitiveness in the market.