VIETNAM NATIONAL UNIVERSITY, HANOI

VNU- International School

Global Information System

Decision Support System for Small-Scale Retailers

Class:	**INS307401**	
Lecturer:	**Ms. Michael Omar**	
Group:	**A**	
Group member:	**Đào Thu Phương**	**21070714**
	Đậu Trúc Linh	**21070369**
	Phạm Thị Xuân Quỳnh	**21070388**
	Dương Thanh Thưởng	**21070184**
	Kiều Văn Bách	**21070512**

Ha Noi, Dec 2024

Participants

Evaluation Criteria	Phạm Thị Xuân Quỳnh	Đậu Trúc Linh	Đào Thu Phương	Dương Thanh Thưởng	Kiều Văn Bách
Attends group meetings regularly and arrives on time.	A	A	A	A	A
Contributes meaningfully to group discussions.	A	A	A	A	A
Completes group assignments on time.	A	A	A	A	A
Prepares work in a quality manners.	A	A	A	A	A
Demonstrates a cooperative and supportive attitude.	A	A	A	A	A
Contributes significantly to the success of the project.	A	A	A	A	A
TOTALS	A	A	A	A	A

Acknowledgement

We would like to express our sincere gratitude to Ms. Michael Omar for her invaluable guidance and support throughout our research. Her expertise and insights were instrumental in selecting the appropriate topic and tools for our study on Global Information Systems. We are grateful for her mentorship and the time she dedicated to reviewing our work, providing constructive feedback, and shaping our research direction.

We would also like to thank our friends who provided us with valuable information and insights related to retail operations and system implementation. Their inputs and discussions enriched our understanding of the local context, allowing us to incorporate relevant perspectives into our project on developing a Decision Support System (DSS) for a small retailer.

Their support and encouragement were invaluable, and we are grateful for their willingness to share their knowledge and experiences.

Collectively, the contributions of Ms. Michael Omar and our friends have been crucial in shaping the success of our research project. We are deeply grateful for their support and collaboration throughout this journey.

Executive Summary

In this report, we present the development and implementation of a Decision Support System (DSS) for the VNU-IS canteen, aimed at addressing operational inefficiencies in inventory management, sales forecasting, and supplier collaboration. Our proposed system leverages modern technologies, including MySQL, Python (Flask), and React.js, to streamline processes, enhance decision-making, and improve customer satisfaction.

Our primary objective is to enable real-time inventory tracking, accurate sales forecasting, and proactive supplier management. The system includes key features such as automated alerts for low-stock and overstock situations, demand forecasting models, and supplier performance analysis. These functionalities are designed to reduce waste, prevent stockouts, and align inventory levels with customer demand.

We expect the system to deliver several benefits, including: Automating manual processes to reduce errors and workload while improving response times.Providing data-driven insights for better planning, resource allocation, and inventory management. Ensuring product availability and responsiveness to demand fluctuations, thereby enhancing the customer experience.

By conducting a thorough feasibility analysis and integrating scalable technologies, we have planned to develop a DSS that aligns with the canteen's strategic goals.

Table of Contents

List of abbreviations

VNU	Vietnam National University, Hanoi
IS	International School
DSS	Decision Support Systems
SMART	Specific, measurable, achievable, relevant, and time-bound
POS	Point of sale
ERD	Entity Relationship Diagram
GDIS	Global Decision Support Information System
WBS	Work Breakdown Structure
ROI	Return on Investment
BCR	Benefit-Cost Ratio
ML	Machine Learning
APIs	Application Programming Interfaces
IT	Information Technology
KPIs	Key Performance Indicators
ID	Identifier

List of Figures

List of tables

Introduction

1. Problem Statement

The canteens at Vietnam National University, International School (VNU-IS) are an important facility, catering to students, faculty, and staff with snacks, beverages, and light meals. Its role is to support academic and social activities. However, operational inefficiencies in inventory management and sales forecasting limit the canteen's ability to meet customer expectations and operate effectively.

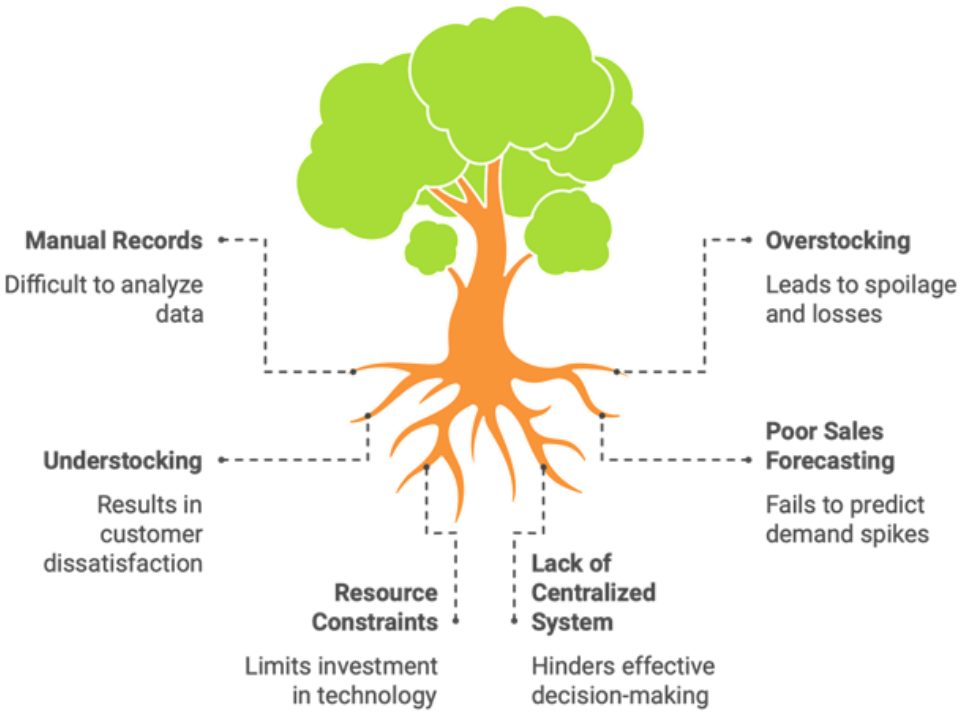

Figure a.1. Current problems of VNUIS Canteen

One of the core issues lies in the fragmented and inefficient management of sales and inventory data. Without a centralized digital system, the canteen relies on manual records and spreadsheets generated from POS, making it difficult to analyze consumption patterns or identify high-demand products. Overstocking is a frequent issue for items such as drinks and packaged snacks, leading to spoilage, storage constraints, and financial losses. Conversely, understocking of popular items like fresh meals and beverages during peak times, such as lunch breaks or special events, often results in customer dissatisfaction and missed revenue opportunities.

Additionally, the canteen struggles with accurately forecasting sales, particularly during periods of fluctuating demand. Seasonal variations, such as increased

consumption during exam seasons or reduced activity during holidays, are not effectively predicted. Event-driven surges, such as those during school fairs or competitions, further exacerbate the problem, leaving the canteen unable to meet demand spikes. These forecasting limitations are compounded by resource constraints, including limited storage space and tight financial budgets, which restrict investments in advanced technologies or specialized staff to improve decision-making.

These problems create several negative outcomes. Too much stock means wasted money on items that do not sell while running out of popular products loses sales and disappoints customers. Without a clear understanding of what items are needed and when it is also harder for the canteen to improve or offer new products.

By implementing a Decision Support System (DSS), the canteen can address these issues effectively. The DSS would use historical and real-time sales data to improve inventory tracking and forecasting, enabling better decision-making. Automated alerts for low-stock and overstock situations would help reduce waste and prevent shortages. Additionally, demand forecasting models would account for seasonal trends and special events, ensuring the canteen is well-prepared to meet customer needs. This system would transform the canteen into a more efficient and customer-focused operation, contributing to a better experience for the VNU-IS community.

2. Objectives

In the context of Global Decision Support Information Systems (GDIS), objectives define the goals or purposes that the system aims to achieve in supporting decision-making across geographically dispersed organizations or international operations. These objectives are aligned with the organization's strategic, tactical, and operational needs and ensure consistency, efficiency, and responsiveness in a global context.

In this section, we outline specific objectives for the proposed Decision Support System (DSS) to improve the canteen's efficiency and customer satisfaction. These objectives are categorized into primary, secondary, and long-term goals, each defined according to the SMART criteria. SMART is an acronym that defines important criteria for setting ideal goals and objectives to manage employees or a company [1]. SMART objectives must be specific, measurable, achievable, relevant and time-bound.

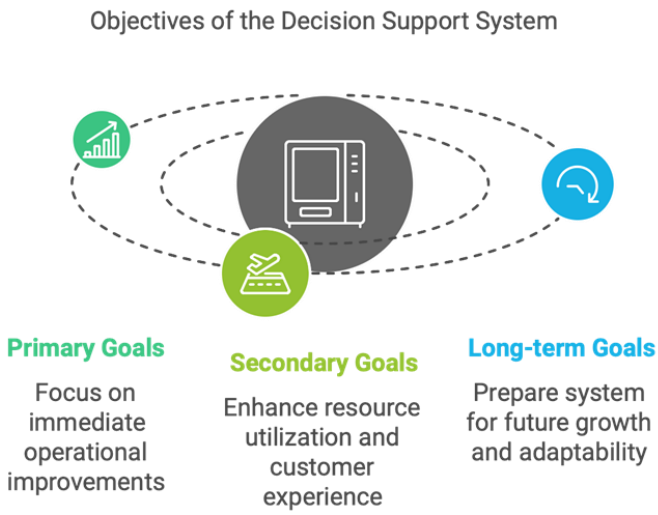

Figure a.2. Objectives of the Decision Support System

2.1. Primary goals

Primary goals address the most critical challenges, such as optimizing inventory management, preventing stockouts, and improving sales forecasting accuracy, which is essential for immediate operational improvements within the first 3–6 months.

- Reduce overstocking by 30% within six months and prevent stockouts, ensuring 95% availability of high-demand products during peak periods.
- Improve demand forecast accuracy by using historical and real-time sales data.
- Provide weekly automated forecasts to align inventory with seasonal and event-driven demand.
- Implement real-time alerts for low stock levels, cutting response times by 50%.
- Automate 80% of inventory tracking, reducing manual errors and staff workload.

2.2. Secondary goals

Secondary goals focus on enhancing resource utilization, and customer experience, providing actionable insights, supporting the primary goals, and adding value over a medium-term timeframe of 6 - 12 months.

- Decrease product unavailability complaints by 50% within three months and ensure popular items are 95% available during busy periods.
- Generate monthly reports on sales, inventory, and wastage, and identify high-demand items within the first month.

2.3. Long-term goals

Long-term goals aim to prepare the system for future growth and adaptability over a longer period.

- Boost annual revenue by 15% through better stock availability and reduced waste.
- Adapt the system to manage new menu items.

Chapter 1 - System Overview

1.1. Cross-Functional Areas

The implementation of the Decision Support System (DSS) for the canteens at VNU-IS integrates and optimizes multiple critical business functions. These include inventory management, sales, and procurement. By centralizing data and automating key processes, the DSS enhances efficiency, improves decision-making, and fosters collaboration across these functions. In this section, we will discuss how these business functions operated before the DSS and how the system integrates and supports them to improve processes.

1.1.1. Functional Areas

The implementation of the Decision Support System (DSS) for the VNU-IS canteen integrates and optimizes the core functions of sales, inventory, and procurement. These functions work collaboratively to ensure the efficient operation of the canteen, meeting customer demands while minimizing waste and operational inefficiencies. Below is a detailed explanation of each functional area and its role within the DSS framework.

1.1.1.1. Sales

The Sales Department is responsible for generating revenue by selling a company's products or services. It focuses on understanding customer needs, maintaining relationships, and driving transactions to meet organizational targets [3].

In the context of the DSS, the Sales department plays a critical role by providing accurate and real-time data on transactions, which directly influences inventory and procurement decisions.

The sales team inputs detailed transaction data, including items sold, quantities, prices, and timestamps, into the system. Sales data is analyzed to identify patterns and trends, such as peak sales periods or popular products. Sales reports are shared with the inventory team to synchronize stock levels with customer demand.

Impact of DSS on Sales:

- Automates data collection via the Point-of-Sale (POS) system, reducing manual entry errors.
- Enhances decision-making by providing real-time sales analytics and trends.
- Improves customer satisfaction through better availability of high-demand products.

1.1.1.2. Inventory

The Inventory Department manages the stock of raw materials, work-in-progress goods, and finished products to ensure the organization has the right quantity of items available at the right time and place [4].

It acts as the link between sales and procurement, providing insights into stock levels, product availability, and replenishment needs. The DSS revolutionizes inventory management by providing real-time updates, automated alerts for low-stock situations, and integration with demand forecasting models. These capabilities enable the canteen to minimize waste, reduce stockouts, and maintain a balanced inventory that aligns with customer demand.

1.1.1.3. Procurement

The Procurement Department is responsible for sourcing and acquiring goods and services needed by the organization, ensuring cost-effectiveness and quality [5].

It involves evaluating replenishment requests, placing orders with suppliers, and ensuring the timely delivery of products. The DSS supports procurement by providing data-driven insights into supplier performance, such as delivery times and pricing, as well as optimizing reorder quantities based on historical and forecasted demand. This ensures that procurement decisions are aligned with inventory requirements and budgetary constraints, ultimately improving the canteen's supply chain efficiency.

By integrating these three functional areas, the DSS creates a centralized platform that fosters collaboration and improves data flow across departments. Real-time data sharing ensures that sales, inventory, and procurement teams are aligned, enabling proactive decision-making and reducing operational bottlenecks. This integration not only enhances the canteen's ability to meet customer needs but also contributes to long-term sustainability by reducing waste and improving resource utilization.

1.1.2. Current Process

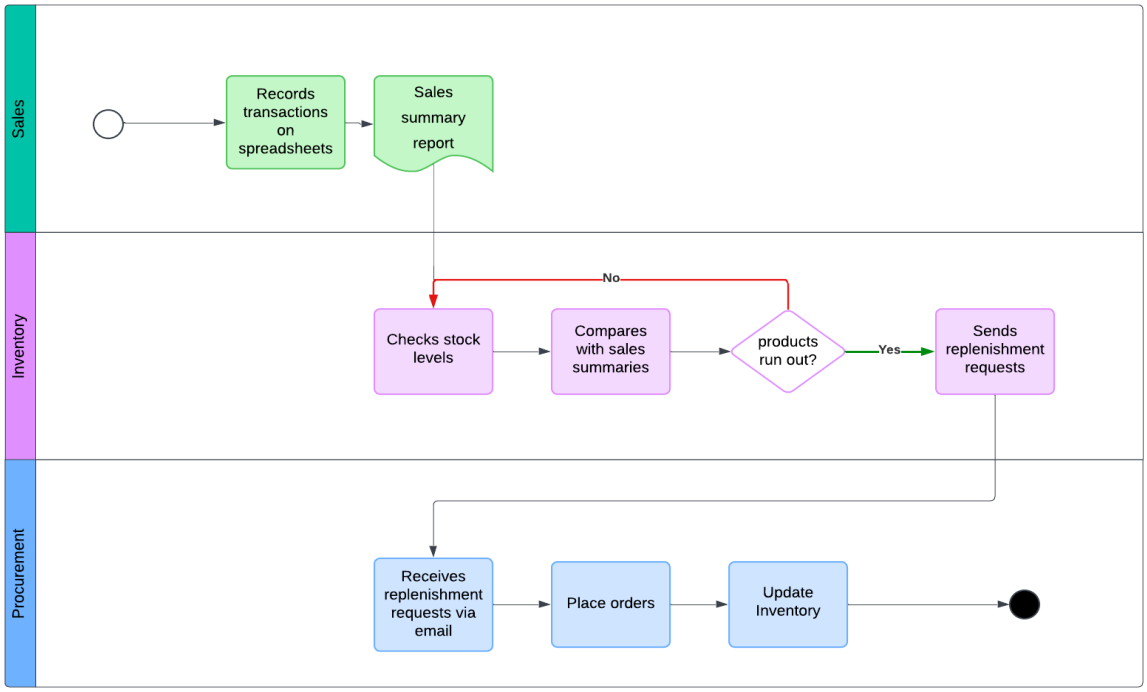

Figure 1.1.2. Swimlane Diagram for Current Process

This diagram represents a Swimlane Diagram for the workflow of managing sales, inventory, and procurement in our business context. The three key actors—Sales, Inventory, and Procurement departments—are organized into distinct swimlanes to clarify responsibilities and interactions. Below is a detailed breakdown of the diagram:

Table 1.1.2. Swimlane Diagram for Current Process Description

Actor	Activity	Details
Sales	Record transactions in spreadsheets	Store information about sold products, quantities, prices, and revenues, typically through Excel.
	Generate sales summary reports	Compile sales data into reports to identify trends and share with the inventory department.
Inventory	Check stock levels	Periodically monitor the quantity of remaining products in stock.
	Compare with sales summaries	Match inventory data with sales reports to detect surplus or shortages of products.

	Decision: Are products out of stock?	If products are out of stock, send a replenishment request. If not, continue monitoring sales data.
	Send replenishment requests	Send detailed requests for the type and quantity of products to be restocked to the procurement department via email.
Procurement	Receive replenishment requests via email	Accept requests from the inventory department, verify the information, and confirm the replenishment requirements.
	Place orders	Contact suppliers to order products, ensuring reasonable prices and delivery terms.
	Update inventory	Upon receiving the goods, update the stock levels in the inventory management system.

Limitations in the current :

The current cross-functional process as depicted in the swimlane diagram, exhibits several limitations that hinder operational efficiency.

First, the reliance on manual data entry and spreadsheet-based record-keeping in the Sales department increases the risk of human errors and delays in data processing. This method is time-consuming and prone to inconsistencies, particularly when generating sales summary reports.

In the Inventory department, stock monitoring and decision-making are reactive rather than proactive. The process of periodically checking stock levels and comparing with sales summaries fails to provide real-time insights, resulting in delayed replenishment requests. This leads to frequent stockouts of high-demand items or overstocking of low-demand products, causing customer dissatisfaction and waste.

Furthermore, the email-based communication between Inventory and Procurement for replenishment requests creates delays and inefficiencies in the procurement process. The lack of automation in placing orders, verifying deliveries, and updating

inventory records slows down the replenishment cycle, making it difficult to respond to fluctuations in demand.

Overall, the current process lacks integration, real-time data visibility, and predictive capabilities, which are essential for achieving seamless coordination and operational optimization.

1.1.3. Proposed new process

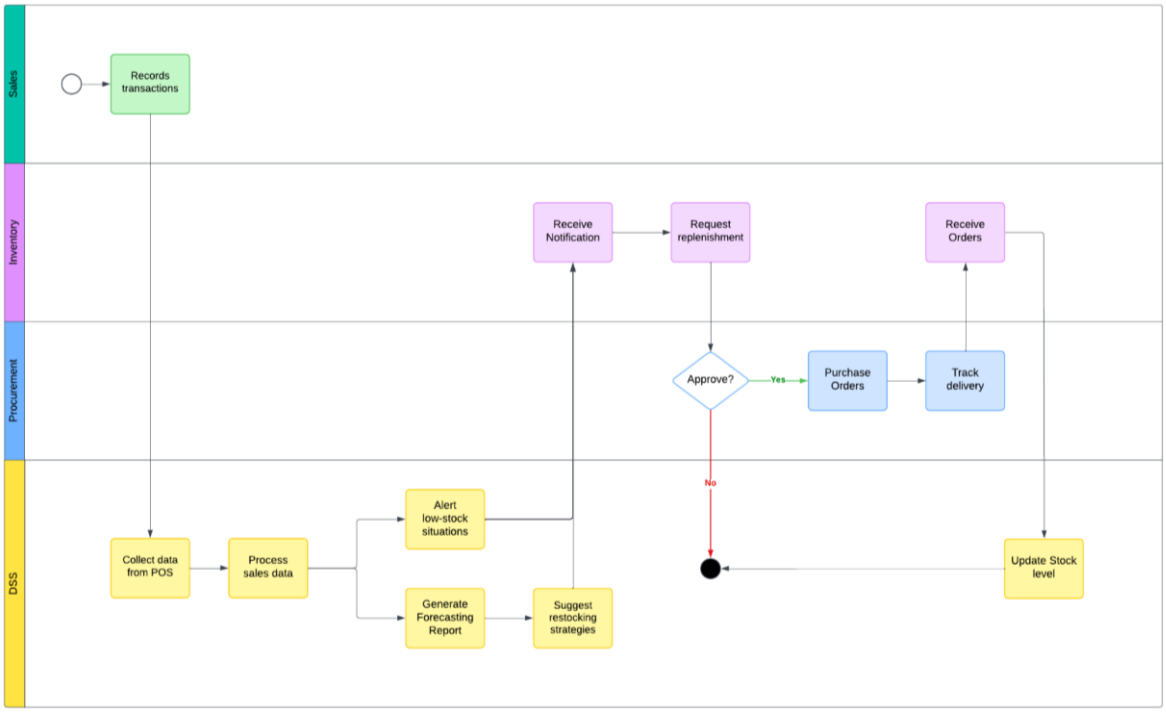

Figure 1.1.3. Swimlane Diagram for Proposed New Process

The detailed description of the updated swimlane diagram with the integration of a Decision Support System (DSS) into the existing sales, inventory, and procurement workflow:

Table 1.1.3. Swimlane Diagram for Proposed New Process Description

Actor	**Activity**	**Details**
Sales	**Record transactions**	Input sales data into the system, capturing details such as items sold, quantities, prices, and timestamps.
DSS	**Collect data from POS**	Automatically retrieve sales data from the Point-of-Sale (POS) system for processing.

	Process sales data	Analyze sales patterns, trends, and frequency of transactions to identify insights for inventory and procurement.
	Alert low-stock situations	Notify the inventory department when stock levels drop below predefined thresholds.
	Generate forecasting reports	Create predictive models for future demand based on historical sales and trends.
	Suggest restocking strategies	Recommend optimal replenishment quantities and timelines to avoid overstocking or stockouts.
Inventory	**Receive notification**	Receive alerts or notifications from the DSS about low-stock situations or forecasted shortages.
	Request replenishment	Create and send replenishment requests to the procurement department.
	Receive orders	Verify and accept the stock received from suppliers, ensure accuracy in quantity and quality
	Update stock levels	Update inventory records in the system to reflect the new stock availability.
Procurement	**Approve replenishment requests**	Evaluate and approve replenishment requests based on budget, priorities, and supplier availability.
	Purchase	Place purchase orders with suppliers for the

	orders	required stock.
	Track delivery	Monitor supplier delivery timelines and logistics to ensure timely restocking.

The updated process with the integration of the Decision Support System (DSS) introduces significant improvements to the workflow of managing sales, inventory, and procurement, addressing the limitations of the previous manual approach. The new system automates critical tasks and centralizes data, enhancing operational efficiency and decision-making.

In the Sales function, transaction data is now input directly into the DSS, capturing essential details such as items sold, quantities, prices, and timestamps. This automation eliminates manual errors and ensures real-time availability of accurate sales data. The DSS collects this data from the Point-of-Sale (POS) system, processes it to analyze patterns and trends, and provides actionable insights for inventory and procurement teams.

The Inventory function benefits significantly from the DSS integration. Automated notifications are generated when stock levels drop below predefined thresholds. These alerts enable inventory staff to act proactively by creating and sending replenishment requests to the procurement department. Additionally, the DSS generates forecasting reports based on historical and real-time sales data, helping inventory teams anticipate demand and adjust stock levels accordingly. By suggesting optimal restocking strategies, the system minimizes overstocking and stockouts, ensuring a consistent supply of high-demand products.

In the Procurement function, replenishment requests are now evaluated and approved based on data-driven insights provided by the DSS. This ensures that orders align with budget constraints and operational priorities. The system tracks supplier performance, monitors delivery timelines, and updates inventory levels upon receipt of goods, streamlining the entire procurement process.

Overall, the integration of the DSS transforms the workflow into a highly automated and data-driven system. It reduces inefficiencies, improves coordination among departments, and supports proactive decision-making, ultimately enhancing the canteen's ability to meet customer needs while optimizing resources.

1.2. Proposed key features

Our proposed Decision Support System (DSS) for the VNU-IS canteen aims to address the operational challenges identified earlier, focusing on improving inventory management, sales forecasting, supplier collaboration, and overall decision-making. Below is a detailed list of system features and how they address the canteen's needs.

1.2.1. Inventory Management

Efficient inventory management is one of the core functionalities of the DSS. The system will help the canteen manage stock levels effectively with the following features:

- Real-time stock tracking: The DSS will automatically update stock levels after each transaction, ensuring that staff have up-to-date information.
- Expiry monitoring: The system will flag items that are close to expiration, reducing waste and ensuring food safety.
- Inventory reporting: Staff can generate daily, weekly, or monthly reports on stock usage, wastage, and turnover rates for better planning.

These features will prevent overstocking and stockouts, minimize waste, and ensure the availability of high-demand items.

1.2.2. Supplier Management

To ensure the timely restocking of products, and help the canteen make cost-effective procurement decisions, we proposed these features for the DSS system:

- Supplier performance tracking: The system will track each supplier's delivery time (LeadTime) and pricing (PriceFromSupplier) to evaluate their reliability.
- Procurement planning: The DSS will recommend the best reorder quantities based on historical sales and forecasted demand.

1.2.3. Sales Management

To better understand customer preferences and increase revenue, the DSS will include:

- Top-selling products report: The DSS will generate a list of the best-selling items, helping the canteen adjust its product offerings.
- Revenue analysis: Staff can view sales trends over time, including high and low sales periods, to make data-driven decisions.

These features will allow the canteen to focus on popular products, adjust pricing strategies, and better meet customer expectations.

1.2.4. Forecasting & Demand Planning

Demand forecasting is critical for managing inventory efficiently. The DSS will provide:

- Sales forecasting models: Based on historical sales data, the system will predict future demand using statistical methods or machine learning.
- Dynamic inventory adjustments: The system will recommend changes to inventory levels to prepare for seasonal peaks (e.g., exams or holidays).
- Forecast accuracy analysis: The system will compare predicted demand with actual sales, allowing continuous improvement in forecasting accuracy.

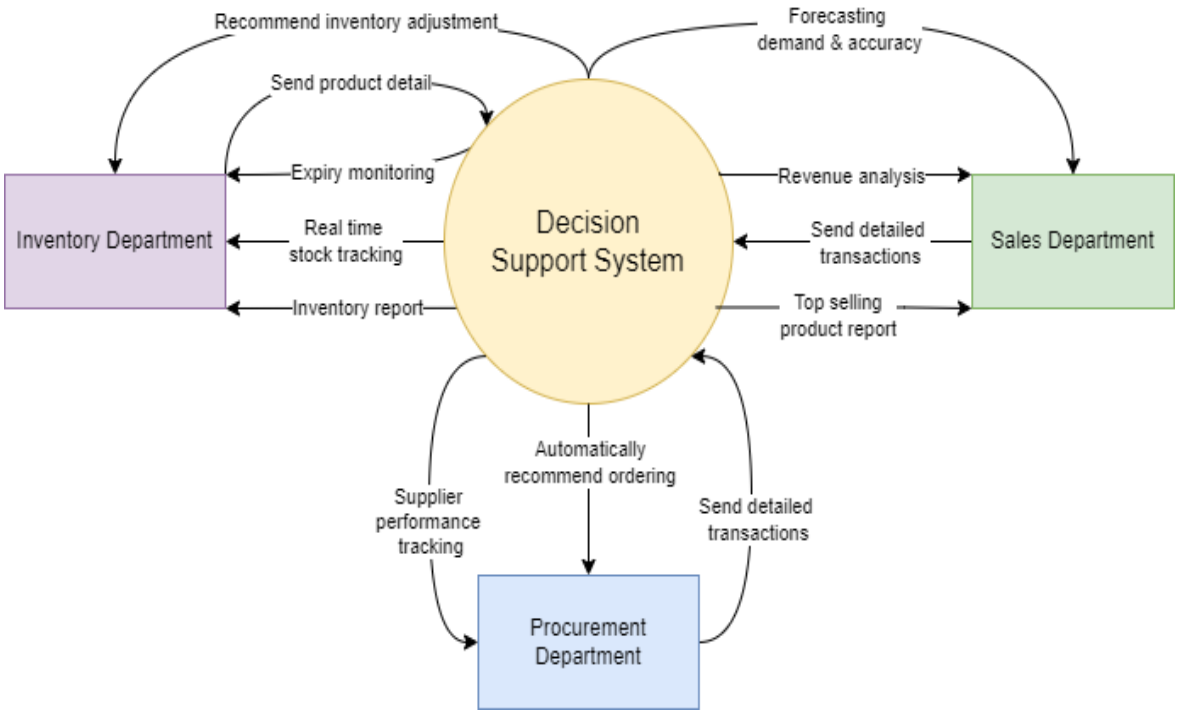

Figure 1.2.4. Context Diagram

Chapter 2 - System Design

2.1. Entity Relationship Diagram(ERD)

An entity relationship diagram (ER diagram or ERD) is a visual representation of how items in a database relate to each other [2]. ERDs are a specialized type of flowchart that conveys the relationship types between different entities within a system.

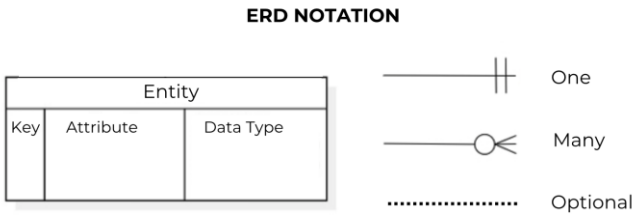

Figure 2.1.1. ERD Notation

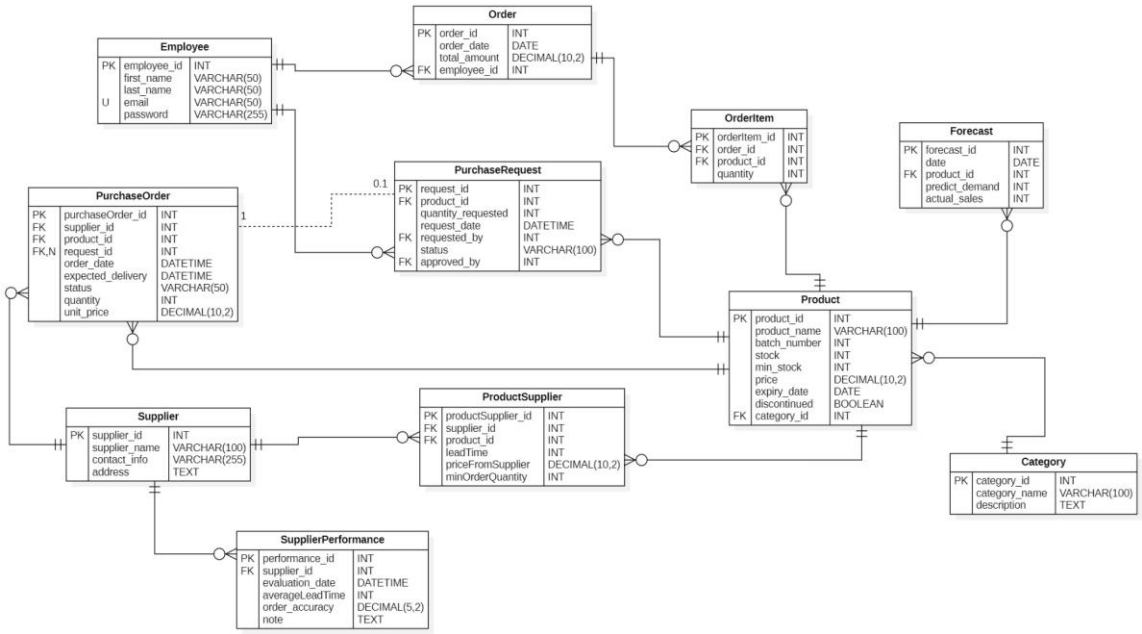

Figure 2.1.2. Entity Relationship Diagram

We designed the *Supplier* table to serve as the central repository for storing information about all suppliers. This table includes key details such as the unique identifier for each supplier (*SupplierID*), supplier name (*SupplierName*), contact information (*ContactInfo*), and address (*Address*). By storing this information, we enable the system to efficiently search for, retrieve, and select appropriate suppliers for each purchase order.

We structured the *Product* table to store detailed information about products, including their unique identifiers (*ProductID*), names (*ProductName*), current stock

levels (*Stock*), and the minimum stock levels (*MinStock*) required to prevent stockouts. Additionally, this table manages product pricing (*Price*) and expiry dates (*ExpiryDate*) for perishable items. The *Discontinued* column indicates the product's status, specifying whether it is still available for sale. We linked this table to the *Category* table, allowing products to be grouped into specific categories.

To monitor the relationship between suppliers and products, we created the *ProductSupplier* table as a bridge table connecting *Supplier* and *Product*. This table records information such as delivery lead times (*LeadTime*) and the price offered by each supplier for specific products (*PriceFromSupplier*). Through this table, we enable the system to track which products each supplier can provide, the delivery times, and the associated costs. Furthermore, we included the *MinOrderQuantity* column to capture the minimum order quantity required by each supplier.

We implemented the *PurchaseRequest* table to document restocking requests made by warehouse staff. Each request is identified by a unique *RequestID*, linked to *ProductID* to specify the product that needs restocking, and includes a *QuantityRequested* column to log the quantity required. Each request also records the creation date (*RequestDate*) and the staff member who created it (*Requested_by*), which we linked to the *Employee* table. The approval process for these requests is represented through the *Status* column (Pending, Approved, Rejected) and the *Approved_by* column, which identifies the employee who approved or rejected the request.

We designed the *PurchaseOrder* table to manage purchase orders sent to suppliers. A purchase order can be created directly or derived from an approved *PurchaseRequest* (linked via *RequestID*). This is reflected in an optional one-to-one relationship between *PurchaseOrder* and *PurchaseRequest*. Each purchase order is uniquely identified by a *PurchaseOrderID* and linked to a supplier (*SupplierID*) to specify the recipient. This table also records details such as the quantity ordered (*Quantity*), unit price (*UnitPrice*), order date (*OrderDate*), and expected delivery date (*ExpectedDelivery*). We use the *Status* column to monitor the order processing stages, including statuses such as Ordered, Delivered, or Cancelled.

We developed the *SupplierPerformance* table to independently evaluate the overall performance of each supplier. Instead of linking performance metrics to specific purchase orders, this table aggregates data to record average metrics such as average lead times (*AverageLeadTime*) and order accuracy (*OrderAccuracy*). These metrics enable us to assess the quality of supplier services. Each record in this table is uniquely identified by a *PerformanceID* and linked to a supplier via *SupplierID*. To

support longitudinal tracking, we included *EvaluationDate* and *Notes* columns for performance evaluations and additional observations.

Finally, we created the *Employee* table to manage information about employees participating in the system. Each employee is identified by a unique *EmpID* and associated with details such as first and last names (*FirstName*, *LastName*), email address (*EmpEmail*), and login credentials (*EmpPassword*). Employees can create purchase requests (*PurchaseRequest*) and approve them.

2.2. Use cases diagram

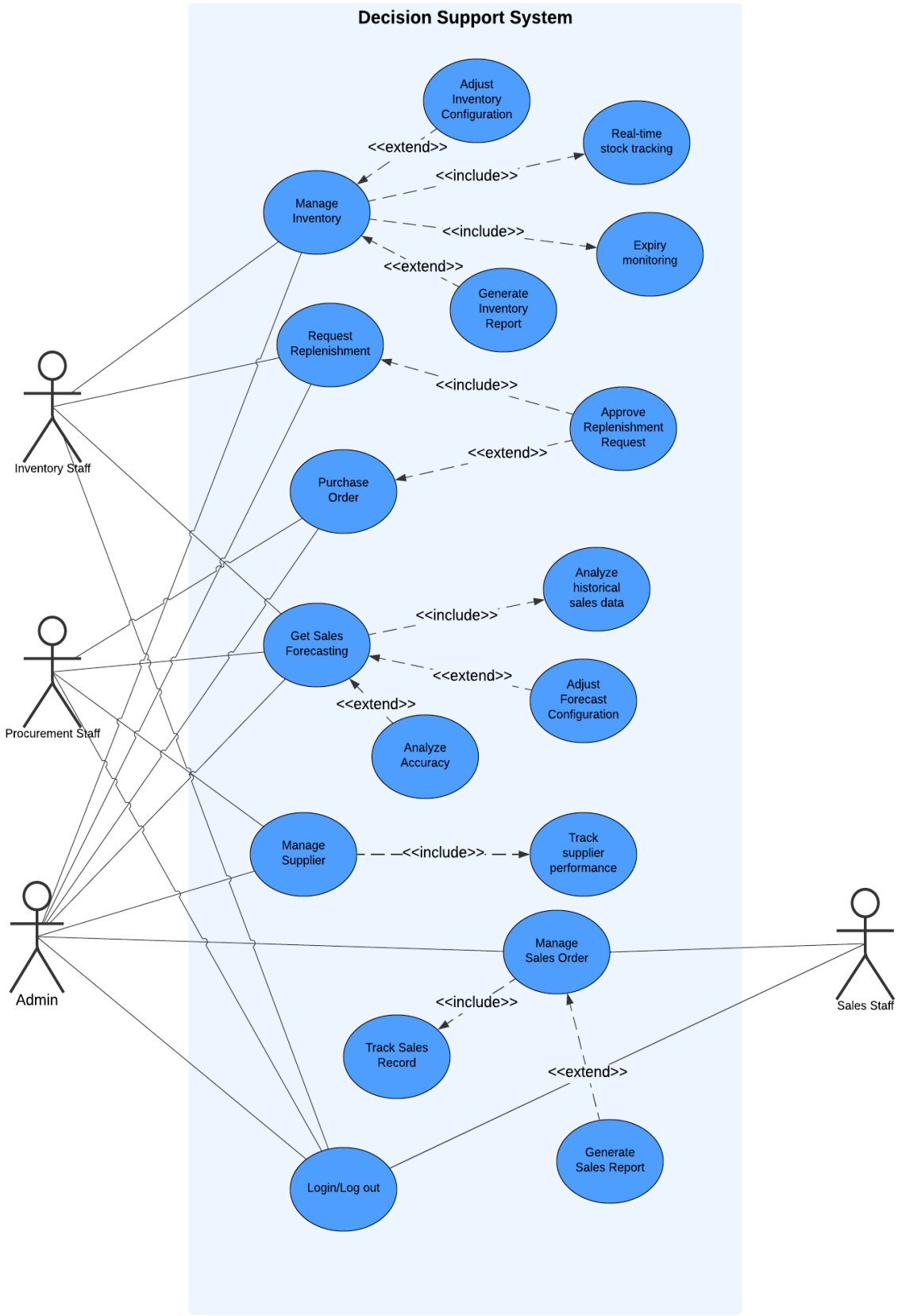

Figure 2.2.1. Use cases Diagram

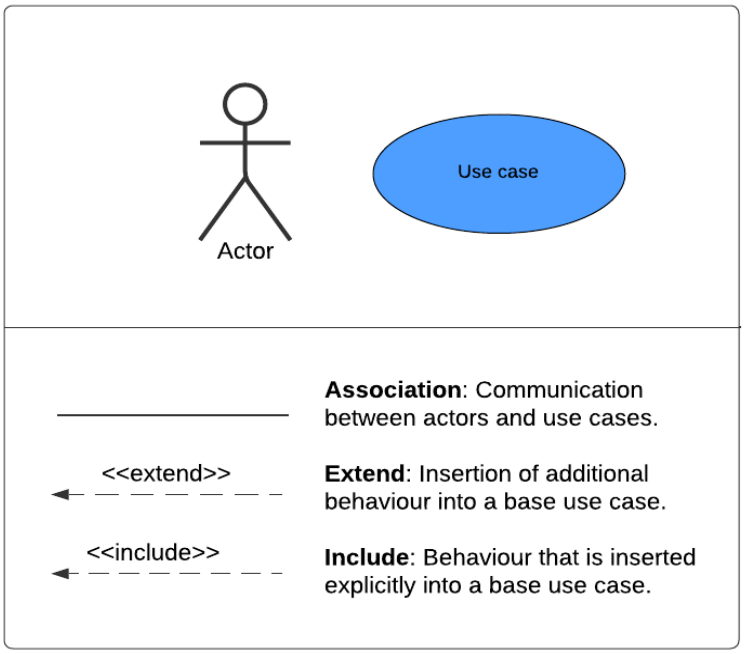

Figure 2.2.2. Use case diagram notations

2.2.1. Real-time stock tracking

Table 2.2.1. Real-time stock tracking Use case Description

Use Case Name	Real-time stock tracking
Actor(s)	Inventory staff, Admin
Description	I want the DSS system to track the inventory in real time
Priority	High
Trigger	• User initiates a request to view real-time stock levels for a product. • System automatically updates stock data in response to stock in/out events.
Pre-Condition(s):	• Inventory database is up-to-date. • Network connection is active. • User has valid login credentials.

Post-Condition(s):	• Users view accurate real-time stock information. • System records the query in the logs.
Basic Flow	• User selects or enters a product code to check stock levels. • If any stock updates are required, the system synchronizes the latest data. • System displays the real-time stock information to the user.
Alternative Flow	
Exception Flow	• The user entered an incorrect product code or the system could not connect to the database. • The system displays the error “Invalid product code” or “Unable to retrieve inventory data”.

2.2.2. Get sales forecasting

Table 2.2.2. Get sales forecasting Use case Description

Use Case Name	Get sales forecasting
Actor(s)	Inventory staff, Procurement staff, Admin
Description	I want the DSS system to forecast the sales (daily/ weekly/ monthly)
Priority	High
Trigger	• The system runs a recurring schedule (day/ week/ month) to predict sales based on updated data.

Pre-Condition(s):	• Required data is available • Forecasting model is set up • User has authorized to access the system
Post-Condition(s):	• Forecast result created successfully • Store and update data • Sent notification
Basic Flow	• User selects or enters the forecast page.
Alternative Flow	• User selects or enters the dashboard page..
Exception Flow	

2.2.3. Supplier performance tracking

Table 2.2.3. Supplier performance tracking Use case Description

Use Case Name	Supplier performance tracking
Actor(s)	Procurement staff, Admin
Description	I want the DSS system to track the suppliers based on their performance
Priority	High
Trigger	• The system runs on a regular schedule (weekly or monthly) to collect and evaluate supplier performance.
Pre-Condition(s):	• Required Data is available • System updates delivery status successfully

	• User has authorized to access the system
Post-Condition(s):	• Supplier performance report generated • Store and update data • Sent notification
Basic Flow	• User selects or enters the supplier management page.
Alternative Flow	
Exception Flow	

2.3. Workflows

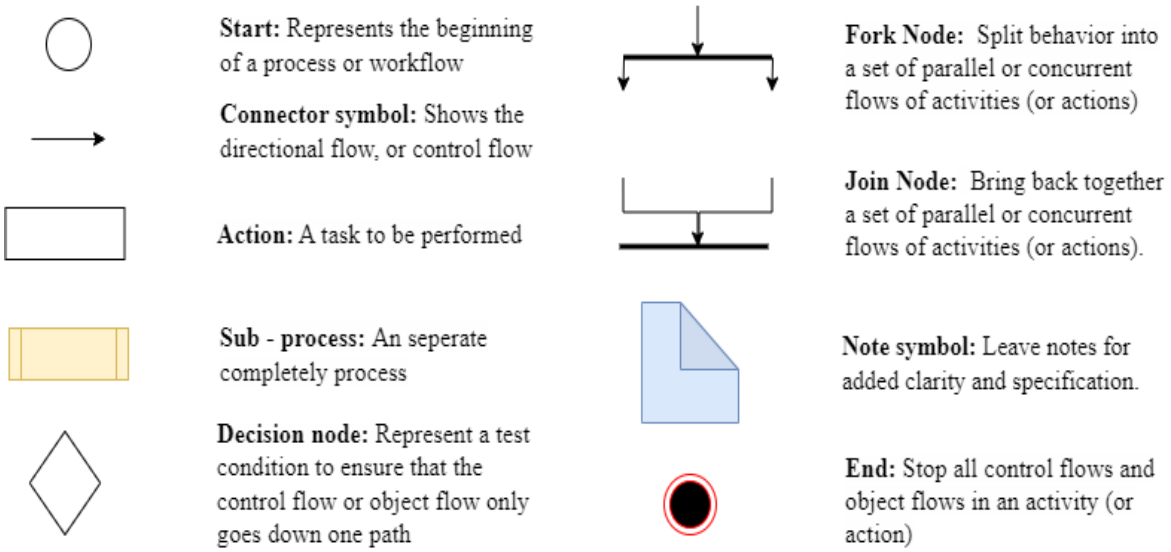

Figure 2.3. Workflow Diagram Notation

2.3.1. Feature Real-time stock tracking

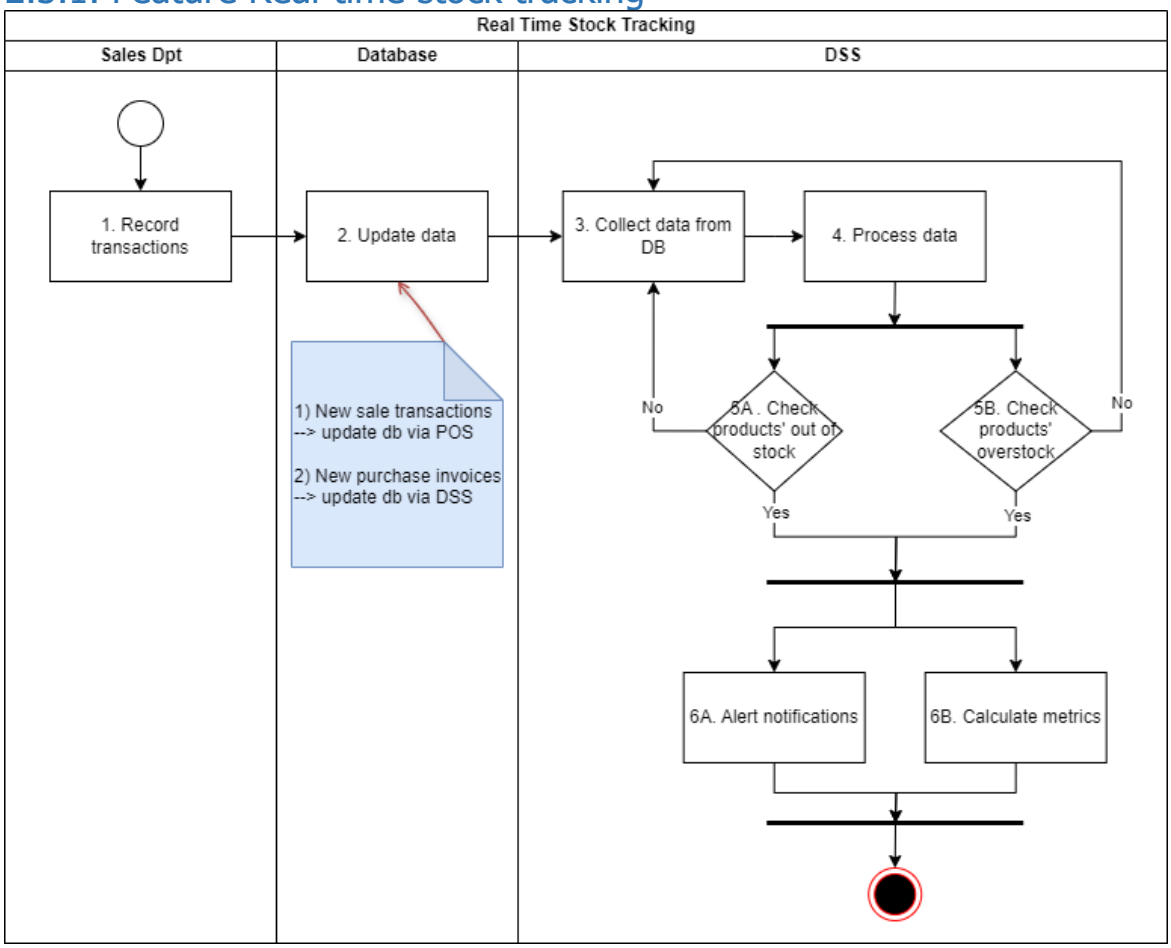

Figure 2.3.1. Real Time Stock Tracking Workflow

Table 2.3.1. Real Time Stock Tracking Workflow Description

Step	Description
1	Sales Department send record transactions
2	Database update data via POS
3	DSS collect data from database
4	DSS process data
5	DSS performs concurrently: 5A. Check if the product is out of stock? - If not, back to step 3. - If yes, go to step 6. 5B. Check if the product is over stock? - If not, back to step 3. - If yes, go to step 6.
6	DSS performs concurrently: ● 6A. Alert notification ● 6B. Calculate metrics

2.3.2. Feature Expiring data monitoring

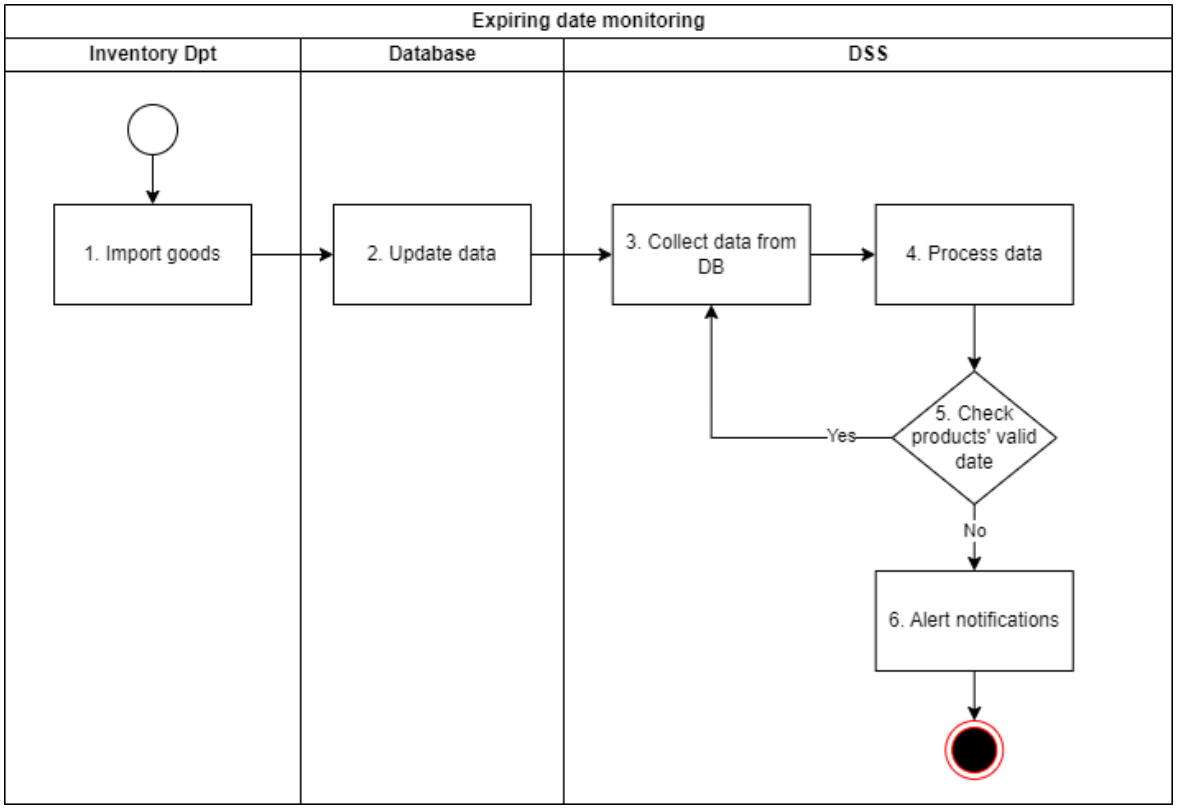

Figure 2.3.2. Expiring Data Monitoring Workflow

Table 2.3.2. Expiring Data Monitoring Workflow Description

Step	Description
1	Inventory Department imports goods.
2	Database update data
3	DSS collect data
4	DSS process data
5	DSS check if product's data is valid? - If yes, back to step 3. - If not, go to step 6.
6	DSS alert notifications.

2.3.3. Feature Recommend inventory adjustment

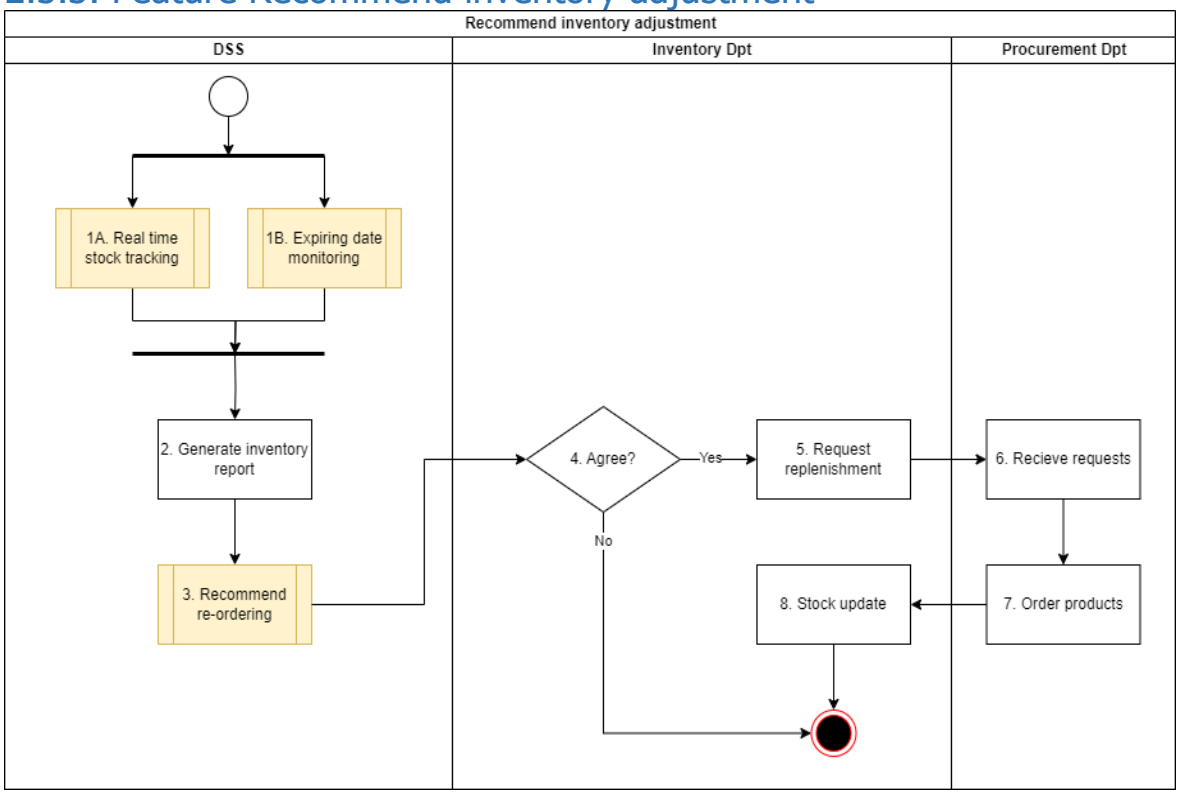

Figure 2.3.3. Recommend Inventory Adjustment Workflow

Table 2.3.3. Recommend Inventory Adjustment Workflow Description

Step	Description
1	DSS performs concurrently: • 1A. Real time stock tracking (Sub - process) • 1B. Expiring date monitoring (Sub - process)
2	DSS generate inventory report
3	DSS recommend re - ordering (Sub - process)
4	Inventory Department agree or not with the recommendations? - If you disagree, end the process. - If you agree, go to step 5.
5	Inventory Department requests replenishment.
6	Procurement Department receives requests.
7	The Procurement Department orders products.
8	Inventory Department updates stock.

2.3.4. Feature Recommend Reordering

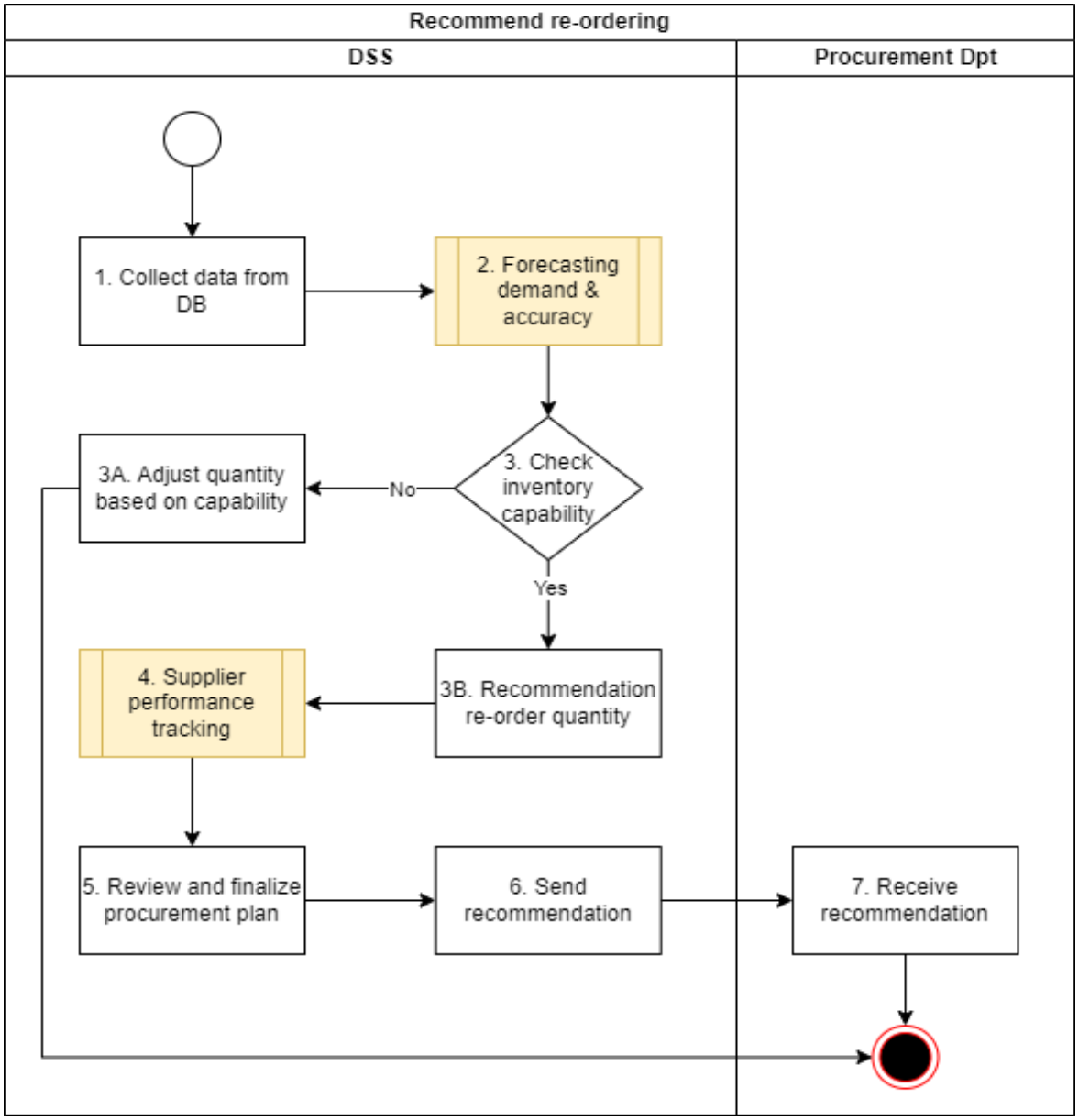

Figure 2.3.4. Recommend Re-ordering Workflow

Table 2.3.4. Recommend Re-ordering Workflow Description

Step	Description
1	DSS collects data from the database
2	DSS forecast demand & accuracy (Subprocess)
3	Does DSS check inventory capability? - If not, ⇒ 3A. Adjust quantity based on capability - If yes, ⇒ 3B. Recommendation re-order quantity
4	Supplier performance tracking (Subprocess)
5	DSS reviews and finalize the procurement plan
6	DSS send recommendation

7	Procurement Department receives the recommendation

2.3.5. Feature Forecasting demand & Accuracy

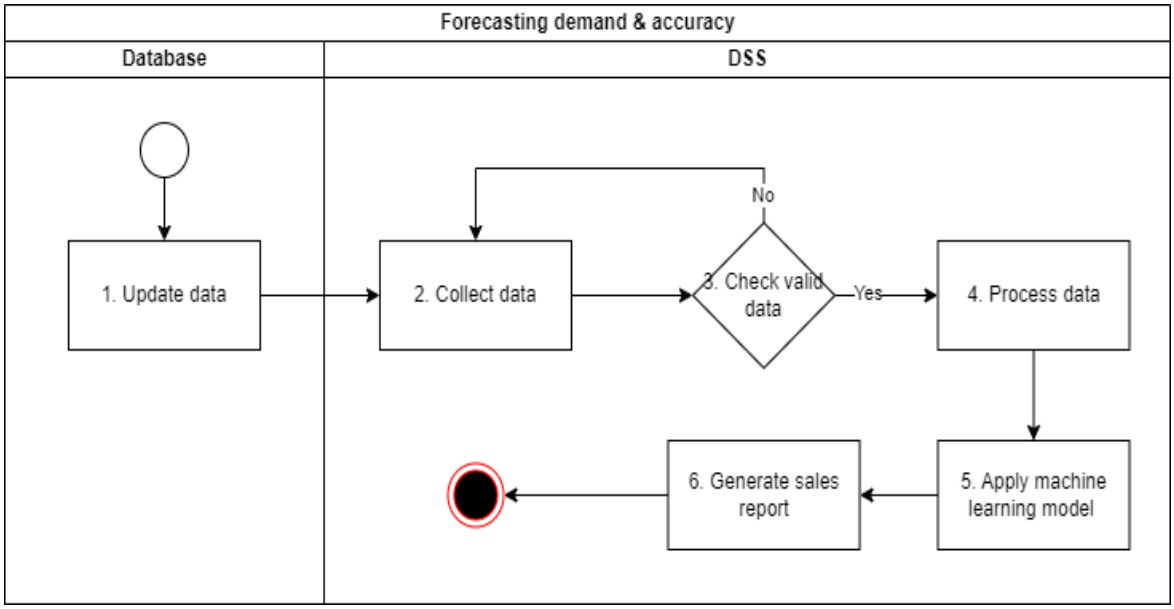

Figure 2.3.5. Forecasting Demand & Accuracy Workflow

Table 2.3.5. Forecasting Demand & Accuracy Workflow Description

Step	Description
1	Database update data
2	DSS collect data
3	Check if the data is valid. - If not, back to step 2. - If yes, go to step 4.
4	DSS process data.
5	DSS applies machine learning model
6	DSS generates sales report

2.3.6. Feature Supplier performance tracking

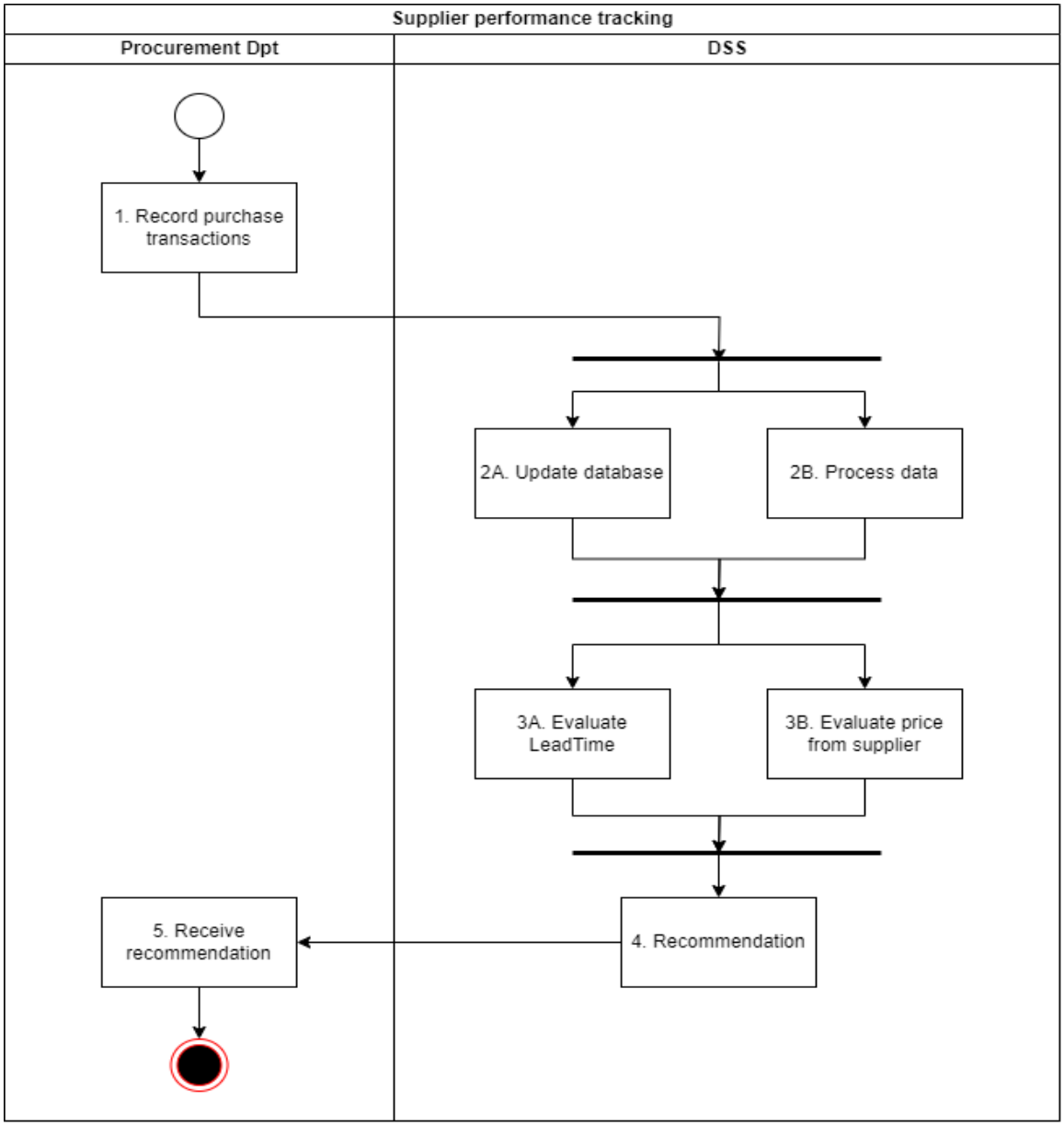

Figure 2.3.6. Supplier Performance Tracking Workflow

Table 2.3.6. Supplier Performance Tracking Workflow Description

Step	Description
1	Procurement Department record purchase transactions
2	DSS performs concurrently: • 2A. Update database • 2B. Process data
3	DSS performs concurrently: • 3A. Evaluate Lead Time • 3B. Evaluate price from supplier
4	DSS give recommendations
5	Procurement Department receive recommendations

2.4. Prototyping

2.4.1. POS

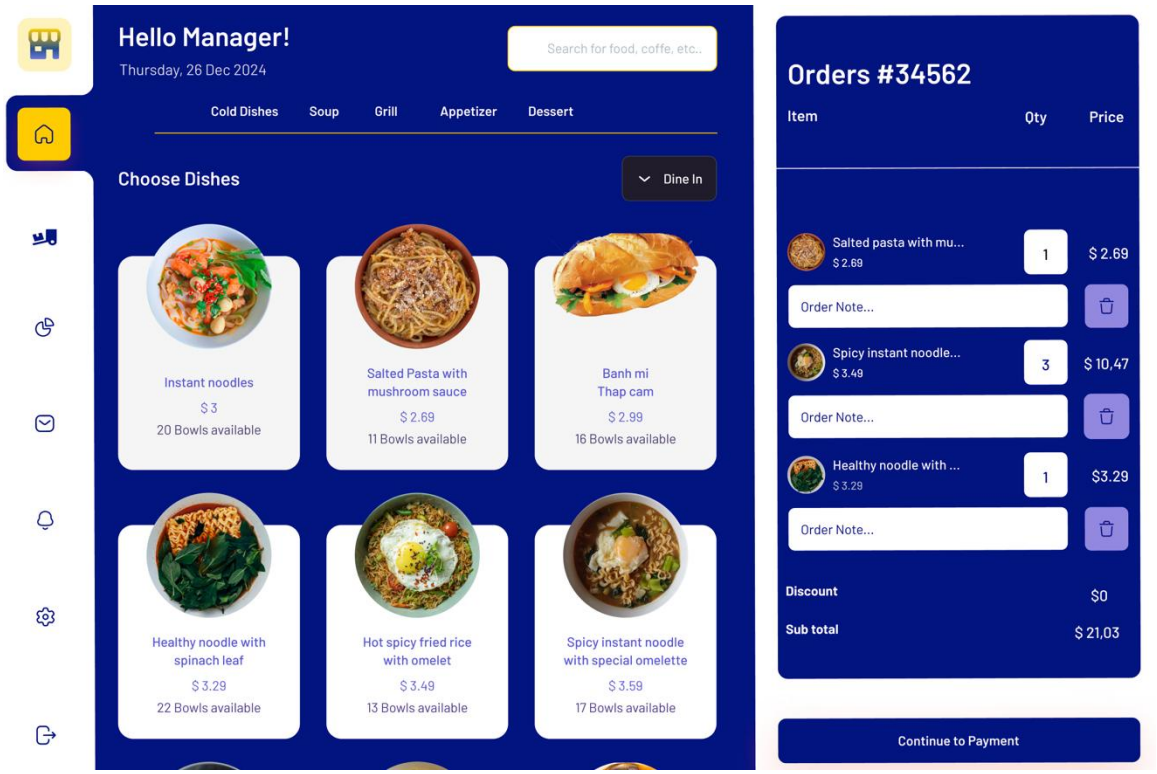

Figure 2.4.1. POS Interface

Table 2.4.1. POS Interface Description

Header Section	The search bar allows users to search for specific food items or drinks
Sidebar Navigation	The sidebar on the left contains icons representing key functions, such as going back, menu management, reports, and system settings. These icons are simple and easy to recognize.
Food Categories	- A horizontal navigation bar below the header provides categories such as Cold Dishes, Soup, Grill, Appetizer, and Dessert. - "Dine In" button is available for selecting the dining mode.
Menu	The central area showcases the available dishes with appealing images, dish names, remaining quantities (e.g., "16 Bowls available"), and prices (e.g., "$2.99").

Order Panel	- The right-hand section displays the current order list, include: + Dish name, quantity, and price. + Order Note + A trash icon to remove the item from the list. - The order total is summarized at the bottom, with details such as Discount, Subtotal, and the overall Total. - A prominent "Continue to Payment" button at the bottom allows users to proceed to the payment step.

2.4.2. Inventory

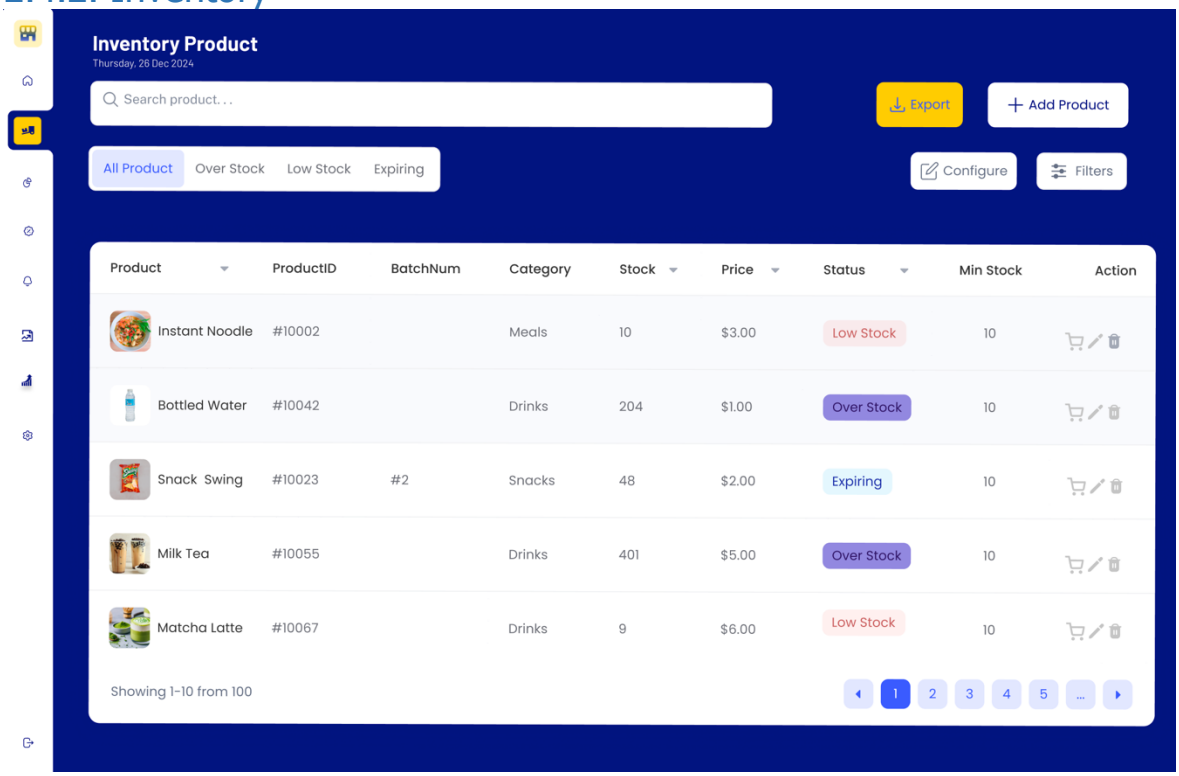

Figure 2.4.2. Inventory Product Description

Table 2.4.2. Inventory Product Dashboard Description

Main Dashboard		
Search Bar	Enables quick product searches by name or ProductID	
Product Filters	**All Products**	Displays the full inventory list.
	Overstock	Highlights products exceeding optimal stock levels.

	Low Stock	Focuses on items that are below the minimum stock threshold.
	Expiring	Lists items nearing their expiration date.
Export Button	Exports inventory data in multiple formats (e.g., CSV, Excel) for reporting or external analysis.	
Add Product Button	Opens a form to add new inventory items with comprehensive details.	

Product Table: Displays a detailed overview of inventory in a structured tabular format.			
Columns	**Product**	Includes product name and thumbnail image for easy identification.	
	ProductID	Unique identifier for each product.	
	BatchNum	Batch-specific tracking details.	
	Category	Classifies products (e.g., Meals, Drinks, Snacks).	
	Stock	Current stock level.	
	Price	Per-unit pricing.	
	Status	**Low Stock**	Indicates products needing urgent replenishment.
		Overstock	Highlights surplus items.
		Expiring	Flags items approaching expiration.
	Min Stock	Minimum stock level required to trigger restocking alerts.	
Action Buttons	**Cart Icon**	Opens a Procurement Request Form for submitting restocking requests.	

	Pencil Icon	Accesses the Product Information Form for updating product details such as category, price, stock, and batch.
	Trash Icon	Allows removal of obsolete or expiring items from the inventory.

Forms	
Procurement Request Form	Includes fields for product name, desired quantity, and additional notes for smooth restocking workflows.
Product Information Form	Enables updating of product attributes such as name, category, batch number, stock, and price.

Advanced Features	
Pagination	Provides controls to navigate through large inventories, ensuring usability even with extensive datasets.
Sidebar Navigation	Simplifies access to various system features and settings.
Export	Facilitates exporting inventory data for external reporting, auditing, or integration with other systems.
Add Product	Streamlines the addition of new items with user-friendly fields for detailed product information.

2.4.3. Replenishment Request Form

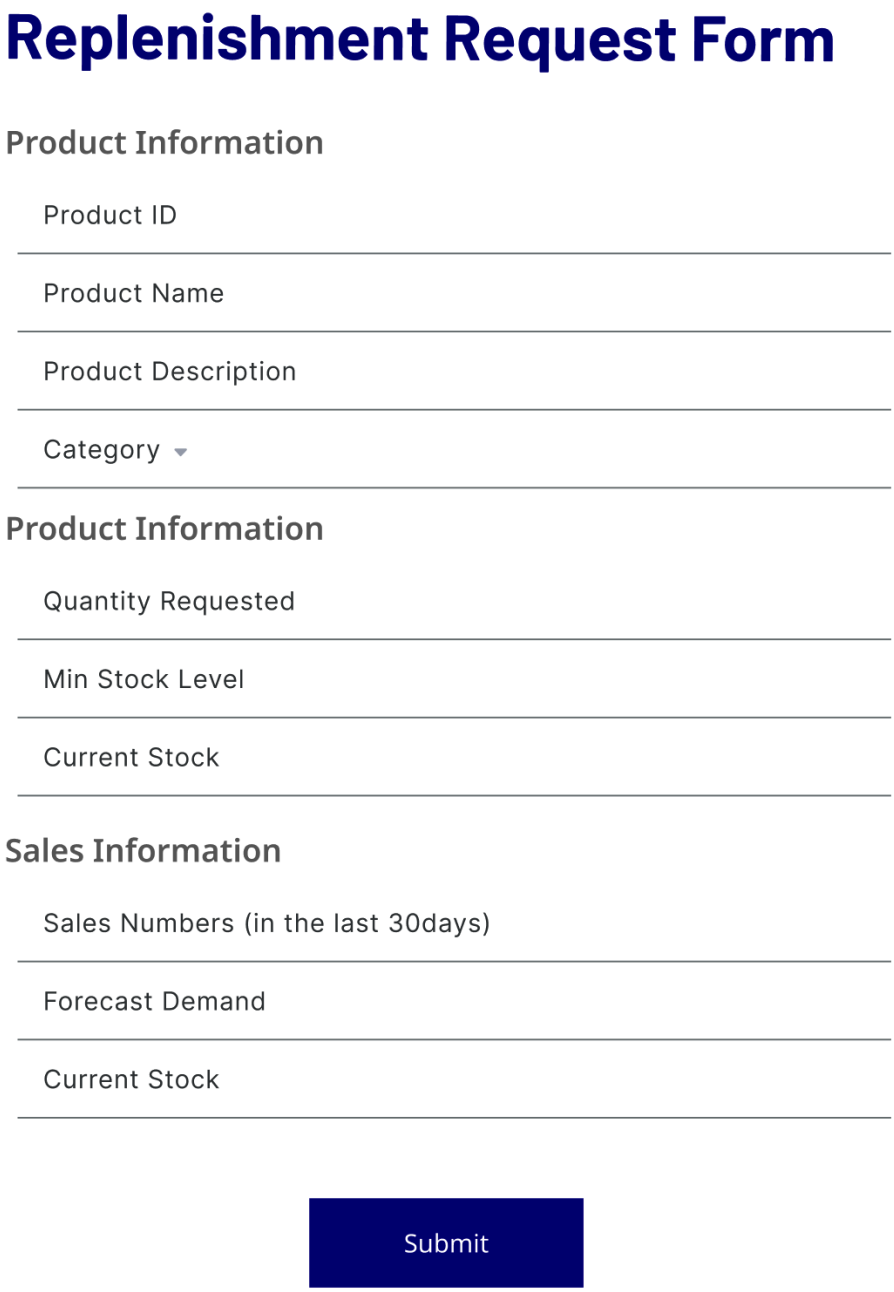

Figure 2.4.3. Replenishment Request Form

2.4.3.1. Product Information

Purpose: collects specific details about the requested product.

- **Product ID**: A unique identifier for the product.
- **Product Name**: The name of the product being requested, helping procurement staff confirm its identity along with the ID.
- **Product Description**: A free-text field where users can provide additional details about the product, such as specifications, versions, or distinguishing features.

- **Category**: A dropdown menu to classify the product into predefined categories (e.g., Electronics, Office Supplies, Raw Materials). This categorization facilitates prioritization and routing of the request.

2.4.3.2. Inventory Information

Purpose: the current stock status of the product.

- **Quantity Requested**: Specifies the exact number of items required
- **Min Stock Level**: The minimum stock threshold set to prevent stockouts.
- **Current Stock**: The number of items currently available in inventory.

2.4.3.3. Sales Information

Purpose: focuses on the product's consumption trends and future demand.

- **Sales Numbers (Last 30 Days)**: Shows the product's sales performance in the past month, helping the team analyze consumption trends.
- **Forecast Demand**: Predicts the expected demand for the product in the near future, aiding in stock planning without over ordering.
- **Current Stock (Reiterated)**: Reiterates the current stock information to ensure both inventory and sales teams have easy access to this critical detail.

2.4.3.4. Submission Section

- Send the request to the procurement team.
- Log the request in a centralized system.
- Notify relevant stakeholders for approval or follow-up actions.

2.4.4. Replenishment/ Request Replenishment

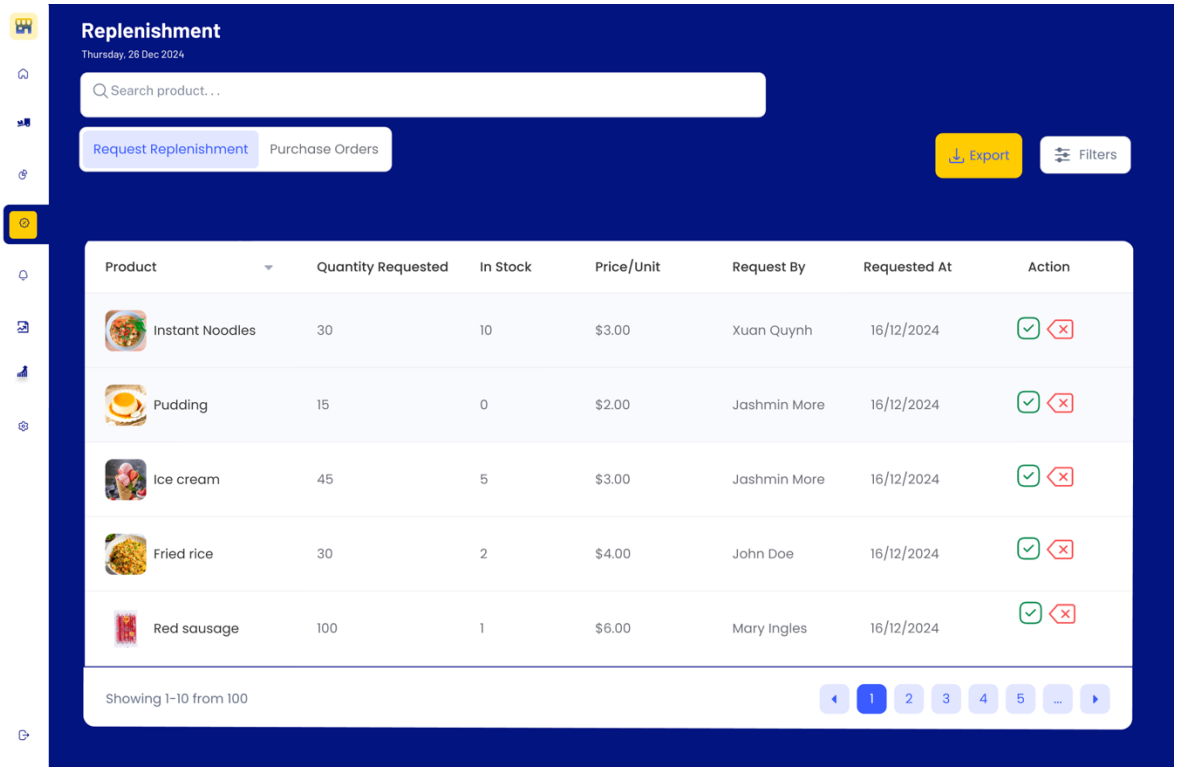

Figure 2.4.4.Replenishment/ Request Replenishment Dashboard

Table 2.4.4. Replenishment/ Request Replenishment Dashboard Description

Main Dashboard			
Header Section	**Title and Date**		
	Navigation Tabs	**Replenishment**	Include pending approvals, rejected orders, and completed requests.
		Purchase Orders	
	Search bar	Quickly find products by name, ID, or category.	
	Actions	**Request Replenishment**	Initiates a new inventory request.
		Export	Enables data export in various formats (Excel, CSV) for reporting or analysis.
		Filters	Refine the displayed data by criteria such as requestor, date, category, or urgency.

Replenishment Requests Table			
Columns	**Product**	Lists requested items	
	Quantity Requested	The number of units required	
	In Stock	Displays current inventory levels	
	Price/Unit	The cost of each product	
	Requested By	The requestor's name	
	Requested At	The date the request was made	
	Action	**Approve**	A green checkmark button for approving requests instantly
		Reject	A red cross button to deny requests

2.4.5. Purchase Order Form

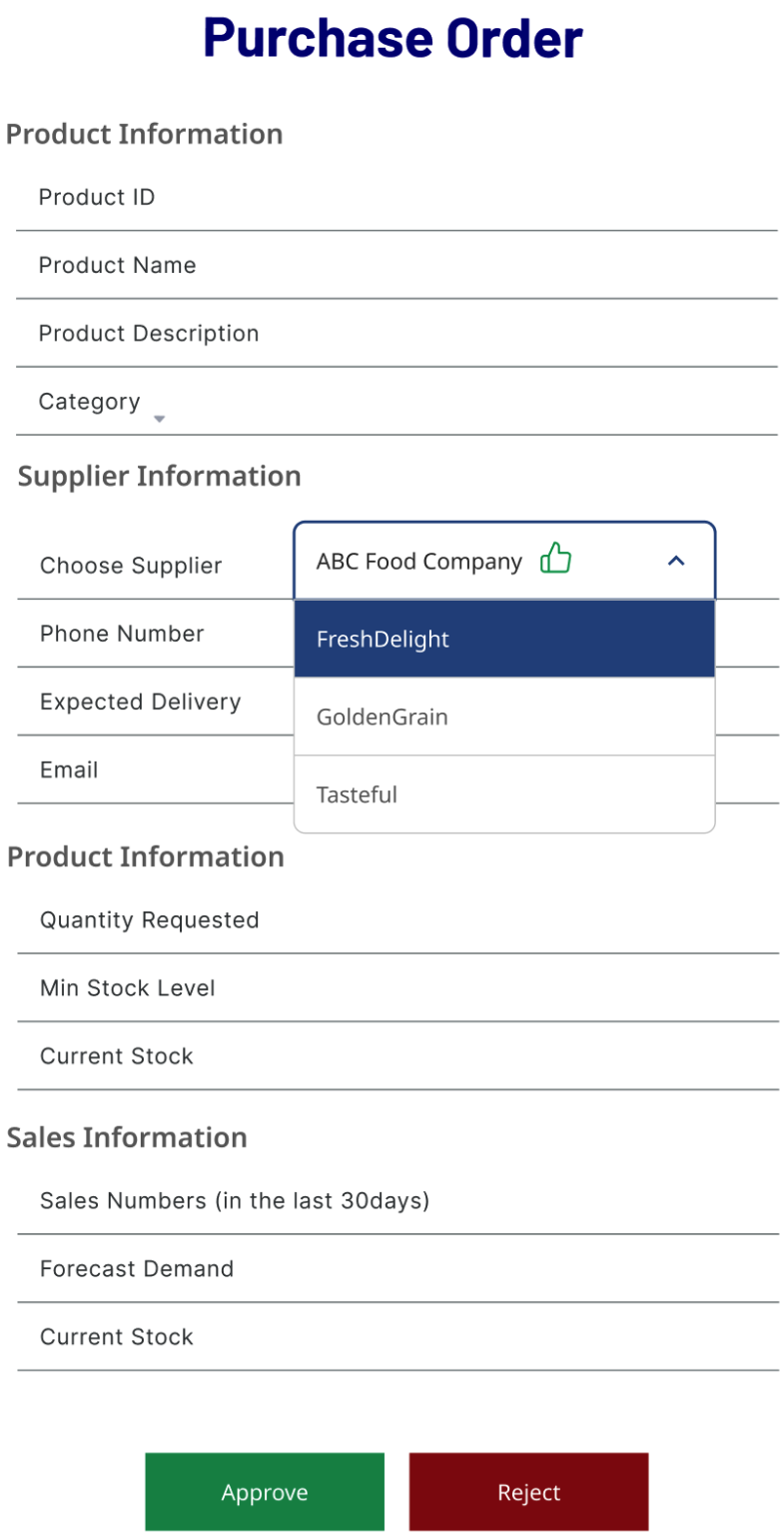

Figure 2.4.5. Purchase Order Form

2.4.5.1. Product Information

Purpose: captures essential details about the product being ordered.

- **Product ID**: A unique identifier for accurate tracking and reference.
- **Product Name**: The name of the product for easy identification.
- **Product Description**: Optional field for additional product details, such as specifications or characteristics.
- **Category**: A dropdown or input field for product classification (e.g., Snacks, Drinks, Meals).

2.4.5.2. Supplier Information

Purpose: ensures the order is sent to the correct supplier.

- **Choose Supplier**: A dropdown menu to select a supplier from predefined options, a thumbs-up: preferred or vetted suppliers.
- **Phone Number**: Supplier's contact number for quick communication.
- **Expected Delivery**: A field to specify the estimated delivery date.
- **Email**: The supplier's email address for sending purchase orders or inquiries.

2.4.5.3. Inventory Replenishment Details

Purpose: focuses on stock and replenishment needs.

- **Quantity Requested**: The number of units being ordered.
- **Min Stock Level**: The minimum stock threshold to avoid shortages.
- **Current Stock**: The existing inventory level.

2.4.5.4. Sales Information

Purpose: provides data insights to support ordering decisions.

- **Sales Numbers (in the last 30 days)**: Highlights recent sales figures to reflect demand trends.
- **Forecast Demand**: Predicted future demand based on sales patterns or forecasting models.
- **Current Stock**: Reinforces the inventory levels as a reference for replenishment.

2.4.5.5. Action Buttons

- **Approve (Green)**: Approves the purchase order for processing.
- **Reject (Red)**: Rejects the purchase order, possibly prompting a review or adjustment.

2.4.6. Replenishment/ Purchase Orders

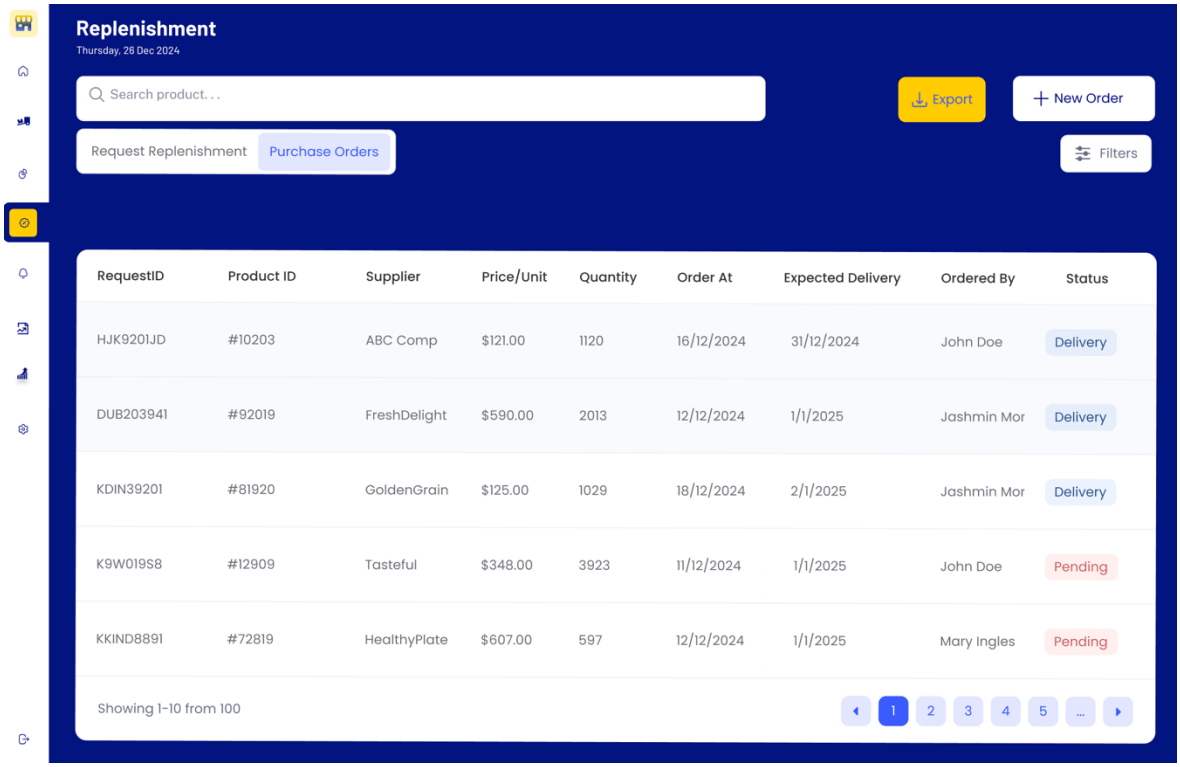

Figure 2.4.6. Replenishment/ Purchase Orders

Table 2.4.6. Replenishment/ Purchase Orders Description

Order Tracking	**Request ID**	A unique identifier for each order.	
	Product ID	Associated with the product being ordered.	
	Supplier	The vendor fulfilling the order.	
	Price/Unit	The cost per unit of the product.	
	Quantity	The number of units ordered.	
	Order At	The date the order was placed.	
	Expected Delivery	The projected delivery date to help teams plan stock management.	
	Ordered By	Identifies the person responsible for placing the order, ensuring accountability.	
	Status	**Delivery**	The order has been dispatched or delivered.

		Pending	The order is awaiting confirmation or further processing.
Order Creation	**New Order**: allows for creating purchase orders		
Actions	**Search Bar**	Quickly locates orders using keywords like product names or request IDs.	
	Filters	The display to prioritize orders by supplier, order status, delivery timelines, or other criteria, ensuring efficient management of high-priority items.	
	Export	To download purchase order data in formats like Excel or CSV.	

2.4.7. Forecast Dashboard

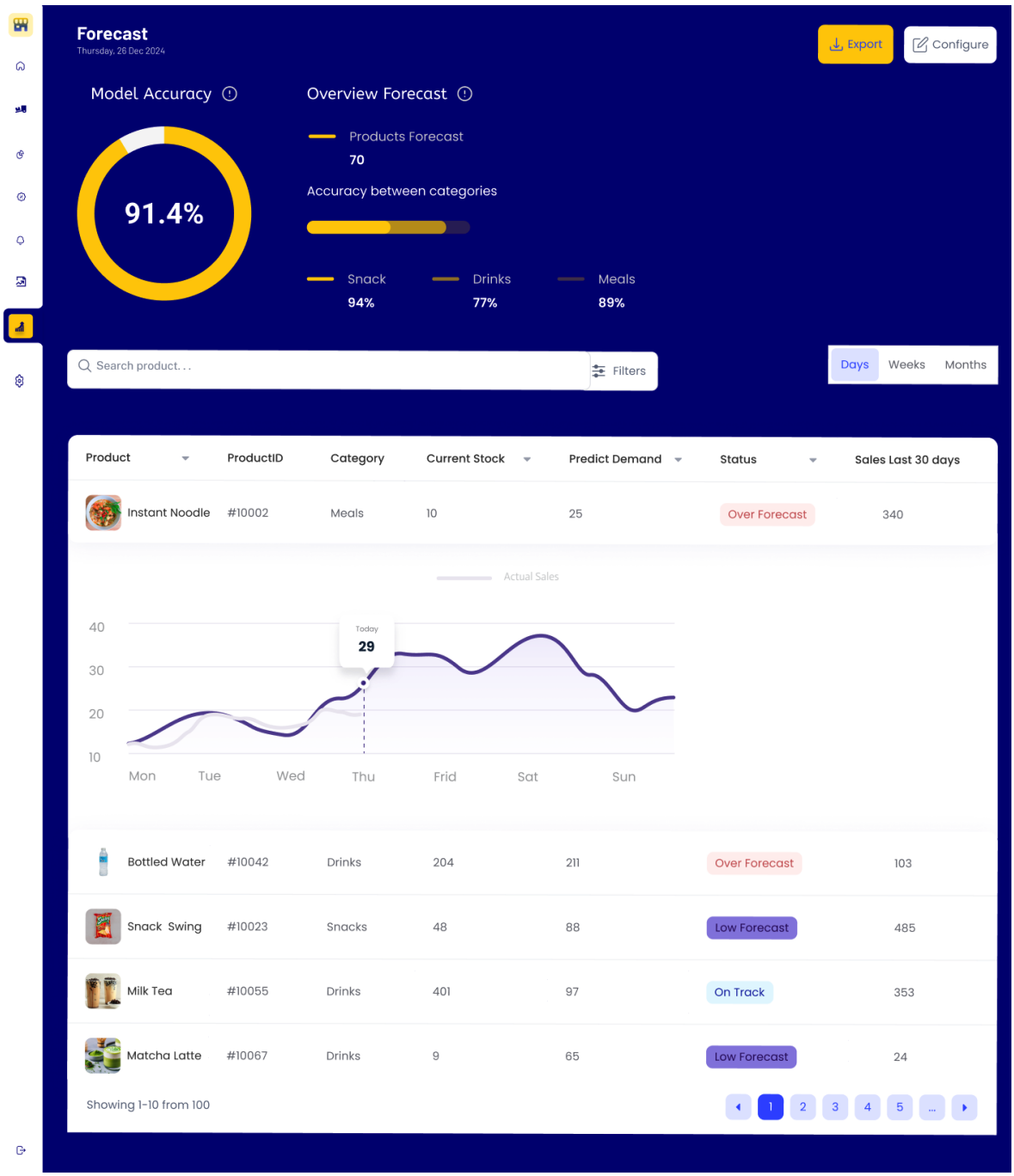

Figure 2.4.7. Forecast Dashboard

Table 2.4.7.1. Forecast Dashboard Header Section Description

Header Section	
Title & Date	Displays the title "Forecast" and the current date.
Model Accuracy	A circular chart indicates the overall accuracy of the forecasting model, such as 91.4%, demonstrating how well the system predicts demand across all products.

Forecast Overview	• Provides a summary of the number of products forecasted (e.g., 70 items) and accuracy for specific categories • Helps pinpoint the strengths and weaknesses of the forecasting model within different product groups.	
Action Buttons	**Export**	For downloading forecast data.
	Configure	Adjust forecasting parameters such as timeframes, product categories, or demand thresholds.
	Search Bar	Allows users to quickly search for specific products, focusing on items of interest.
	Filters	Customizable options enable narrowing down displayed data

Table 2.4.7.2. Forecast Dashboard Forecast details Table Description

Forecast Details Table		
Product	Displays product images and names for quick identification.	
Product ID	A unique identifier for each product to facilitate tracking and cross-referencing.	
Category	Groups products into relevant categories (e.g., Snacks, Drinks, Meals) for trend analysis.	
Current Stock	Displays the current inventory levels for each product.	
Predicted Demand	Shows the expected demand for a product over a specific period, helping determine.	
Status	**Over Forecast**	Predicted demand exceeds available stock.
	Low Forecast	Predicted demand is significantly below stock.
	On Track	Inventory levels align well with predicted demand.
Sales Last 30 Days	Shows the total units sold in the last month, providing a reference for understanding demand trends.	

Table 2.4.7.3. Forecast Dashboard Visual Analytics & Trends Description

Visual Analytics & Trends	
Sales Trends Chart	• A dynamic chart visualizes sales trends for selected products over time (e.g., Monday through Sunday). • Displays actual sales and predicted demand, enabling a comparison of real-world performance against forecasts. • Hovering over data points reveals detailed figures for specific days (e.g., 29 units on Thursday).
Timeframe Customization	Toggle between Days, Weeks, and Months to adjust the forecasting horizon for different periods.

2.4.8. Suppliers Management Dashboard

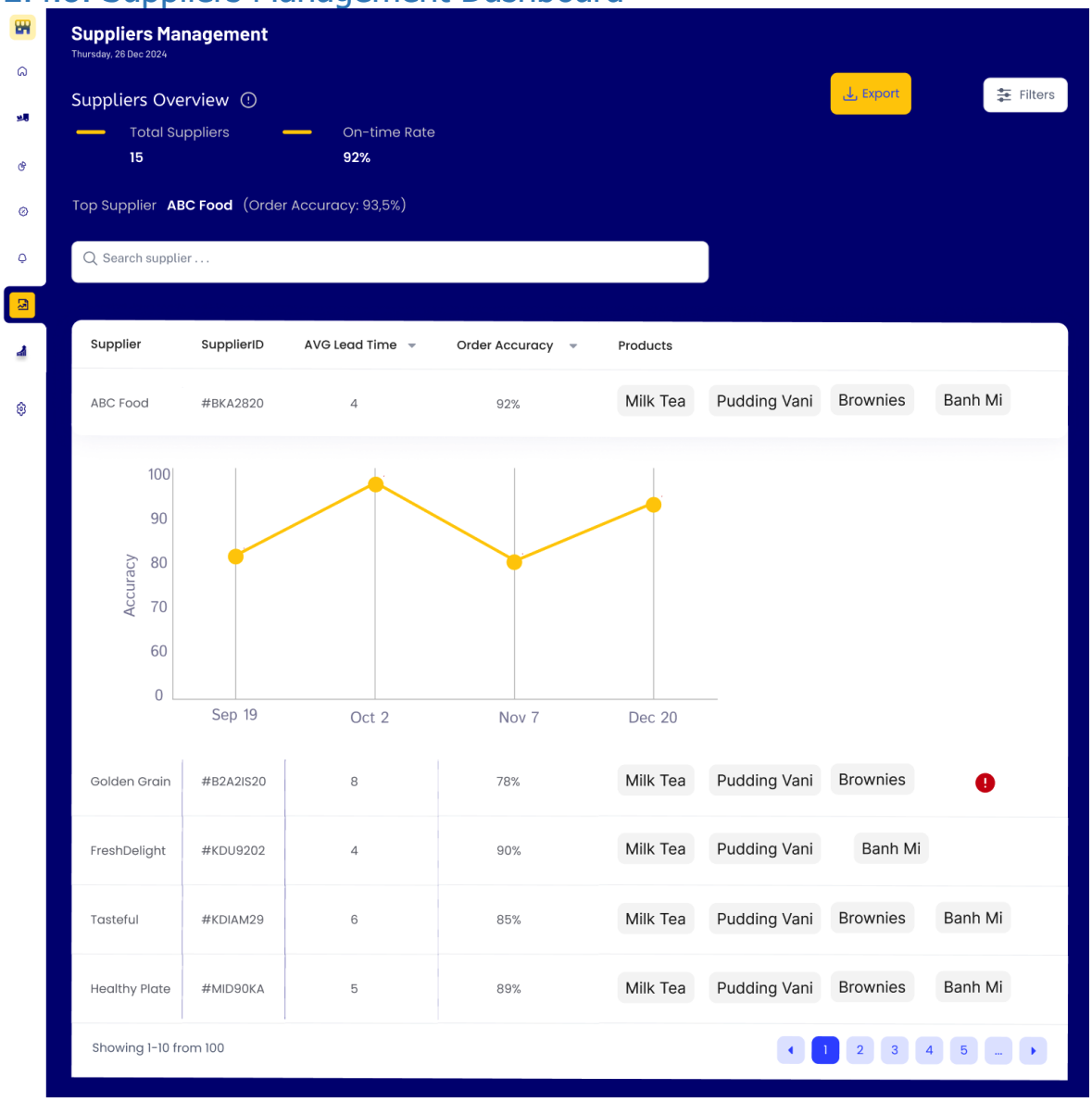

Figure 2.4.8. Suppliers Management Dashboard

Table 2.4.8.1. Suppliers Management Dashboard Suppliers Overview & Actions Description

Suppliers Overview: a quick summary of supplier performance metrics	
Total Suppliers	The total number of suppliers under management.
On-Time Rate	The percentage of orders delivered on time, reflecting supplier punctuality.
Top Supplier	Highlights the best-performing supplier based on accuracy metrics
Actions	
Search Bar	Provides quick access to supplier data by searching for specific names or IDs.
Filters	Enables customizable filtering options to narrow the dataset by criteria such as accuracy, lead time, or product category.
Export	Enables downloading supplier performance data for external analysis or reporting.

Table 2.4.8.2. Suppliers Management Dashboard Supplier Performance Table & Performance Trend Chart Description

Supplier Performance Table: offers detailed insights into supplier metrics.	
Supplier Name and ID	Lists suppliers along with their unique identifiers.
AVG Lead Time	Displays the average order lead time (in days).
Order Accuracy	The percentage of orders completed correctly and delivered on time.
Products	Identifies the main products supplied by each vendor.

Performance Flag	Suppliers with accuracy below 80% are flagged with a red exclamation mark (!), signaling underperformance and prompting review.
Performance Trend Chart: A dynamic line chart illustrates performance trends for individual suppliers	
Trend Chart	• Tracks accuracy over time for a selected supplier • Visualizes performance trends on key dates • Assists in identifying whether a supplier's performance is improving, stable, or declining.

2.4.9. Canteen Performance Monitoring Dashboard Overview

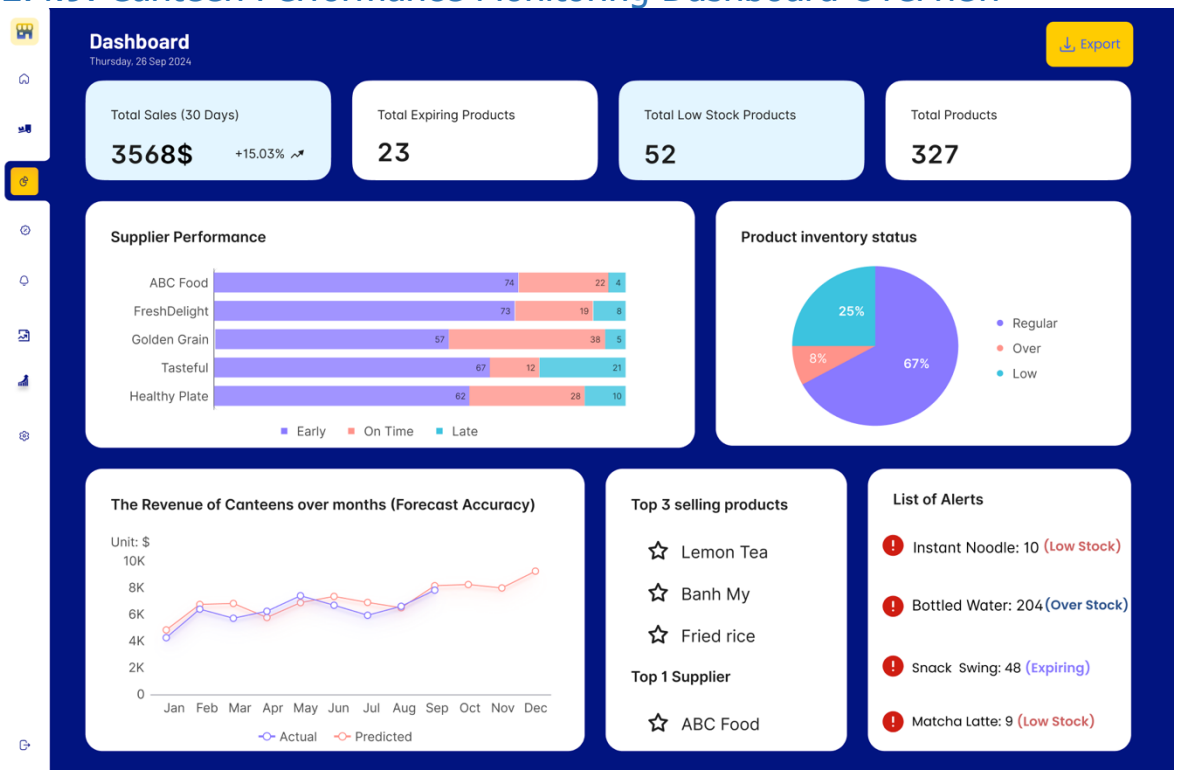

Figure 2.4.9. Canteen Performance Monitoring Dashboard Overview

Table 2.4.9.1. Canteen Performance Monitoring Dashboard Overview Description

High-Level Summary: features four key performance indicators (KPIs)	
Total Sales (30 Days)	Reflects revenue over the last month
Total Expiring Products	Displays items nearing expiry, encouraging waste reduction strategies.

Total Low Stock Products	Identifies items at risk of stock depletion.
Total Products	Shows the overall count of unique inventory items.
Supplier Performance: A bar chart categorizes supplier delivery performance	
Early Deliveries	Orders delivered ahead of schedule.
On-Time Deliveries	Orders delivered as planned.
Late Deliveries	Orders delivered late, potentially disrupting operations.
Product Inventory Status: A pie chart illustrates inventory distribution	
Regular Stock	Items maintained at optimal levels.
Overstocked	Items exceeding requirements, which may result in unnecessary costs or space constraints.
Low Stock	Items needing urgent replenishment.
Revenue Forecast Accuracy: A line chart compares Actual vs. Predicted Revenue, offering insights into DSS forecasting precision	
Key Insights	
Top 3 Selling Products	Identifies popular items like Lemon Tea, Banh Mi, and Fried Rice, guiding inventory prioritization.
Top Supplier	Recognizes the best-performing supplier (e.g., ABC Food), based on metrics such as delivery punctuality and order accuracy.
Alerts	
Low Stock Alerts	Examples include Instant Noodle (10 units) and Bottled Water (204 units).
Expiring Products	Items like Snack Swing (48 units) nearing expiry.

Forecast Warnings	Examples such as Matcha Latte (9 units) suggest insufficient stock relative to projected demand.

2.4.10. Product Details

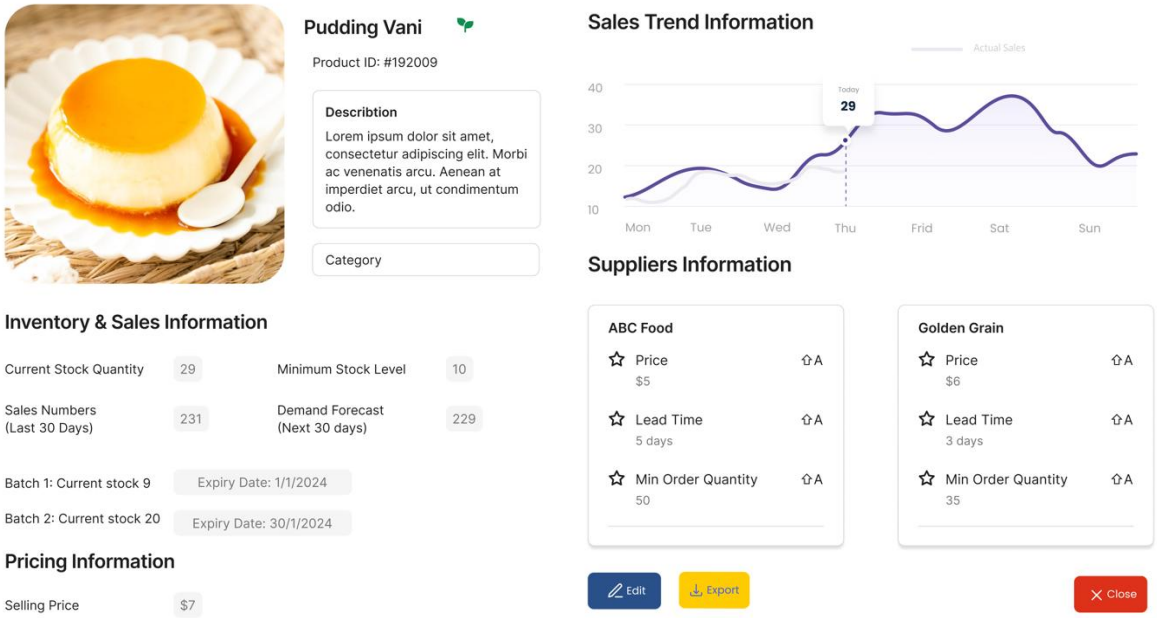

Figure 2.4.10. Product Details Management Interface

Product Management Interface: provide detailed information and support effective product management. The main sections include:

Table 2.4.10. Product Details Management Interface Description

Product Information	**Image**	Image of the product.
	Product Name & ID	Easily identify the product with its name and unique ID.
	Description	A brief overview of the product's key features.
Inventory & Sales Information	**Current Stock Quantity** and **Minimum Stock Level.**	
	Sales Numbers (Last 30 Days) and **Demand Forecast (Next 30 Days)**.	
	Current stock and **Expiry dates**	
Pricing Information	Displays the current retail price of the product.	

Sales Trend Information	A chart showing sales trends over time, aiding in tracking performance.	
Supplier Information	• Lists potential suppliers with detailed information on pricing, lead time, and minimum order quantities. • Enables comparison to select the most suitable supplier.	
Quick Actions	**Edit**	Update product details.
	Export	Download detailed data for reporting or record-keeping purposes.
	Close	Exit the interface.

Chapter 3 - Implementation

3.1. Technology Stack

3.1.1. Technology Stack

A technology stack, or tech stack, is a combination of software tools, programming languages, frameworks, and libraries that developers use to build and run applications [7]. Selecting the appropriate tech stack is crucial, as it influences the application's performance, scalability, and maintainability. A well-chosen tech stack can streamline development, enhance efficiency, and ensure the final product aligns with business objectives [8].

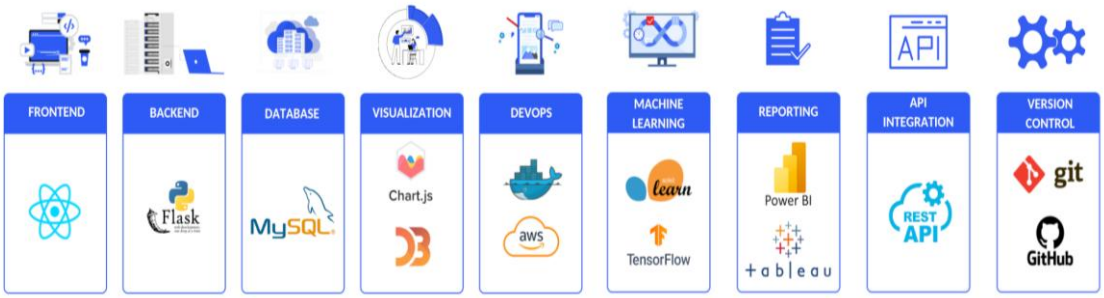

Figure 3.1.1. Technology Stack

3.1.2. Justification of the chosen technology stack

We propose a technology stack for the Decision Support System (DSS) that ensures the system is scalable, reliable, and easy to maintain. The selected tools and platforms focus on addressing the needs of the VNU-IS canteen while supporting the system's functionalities.

We use **MySQL** as the database management system because it is a reliable and cost-effective solution for managing structured data such as products, orders, suppliers, and forecasts. MySQL's relational capabilities allow us to handle the relationships between data efficiently. It is also open source, which reduces costs, and is widely supported, making it a practical choice for our system.

For the **backend**, we choose Python with the Flask framework. Flask is lightweight and flexible, making it easy for us to build and maintain APIs that connect the front end and database. Python's powerful libraries like Pandas help us analyze data, while frameworks such as Scikit-learn and TensorFlow allow us to implement machine learning models for demand forecasting. Python's versatility and community support ensure that the system remains efficient and easy to upgrade.

We select **React.js** for the **front end** to build a user-friendly and interactive interface. React's modular structure allows us to create reusable components for features like dashboards and inventory tracking. It works well with visualization tools like Chart.js and D3.js, helping us display real-time data trends such as stock levels and sales performance in a clear and intuitive way.

For **data visualization**, we integrate tools like Chart.js or D3.js to create graphs and charts that help the staff understand key metrics such as revenue trends and forecast accuracy. These visualizations are customizable and integrate seamlessly with React.

To generate reports and provide business insights, we propose using **Power BI** or **Tableau**. These tools allow us to connect directly to the MySQL database and create interactive reports and dashboards. With their drag-and-drop interfaces, they make it easy for staff to view data without needing technical skills.

Finally, we host the system on Amazon Web Services (AWS) to ensure high reliability and scalability. AWS services like RDS for databases and EC2 for backend hosting help us manage resources efficiently. We also use Docker for containerization, ensuring that the development and production environments are consistent and easy to deploy.

For communication between components, we build **RESTful APIs**. These APIs let the frontend, backend, and database interact efficiently and allow us to expand the system in the future. We use Git and GitHub for version control to track code changes and support collaborative development.

Table 3.1.2. Chosen Technology & The Reasons

Category	**Technology**	**Reasons**
Database	MySQL	Relational structure, scalability, and cost-effectiveness
Backend	Python (Flask)	Lightweight, flexible, and supports analytics integration.
Frontend	React.js	Interactive, modular, and responsive UI
Visualization	Chart.js / D3.js	Real-time, customizable dashboards
Machine Learning	Scikit-learn, TensorFlow	Advanced forecasting models
Reporting	Power BI / Tableau	Advanced reporting and business intelligence

Hosting	AWS	Scalable, reliable, and cost-efficient
API Integration	RESTful APIs	Standardized, scalable communication
Version Control	Git, GitHub	Team collaboration and code versioning
Containerization	Docker	Consistency and scalability

By choosing this technology stack, we ensure that the DSS is reliable, scalable, and tailored to meet the needs of the VNU-IS canteen. This stack will help us address the challenges of inventory management, sales forecasting, and supplier collaboration while improving decision-making for the canteen.

3.1.3. Deployment Documentation

3.1.3.1. Overview

Table 3.1.3.1. Deployment Documentation Overview

Project Name	VNU-IS Canteen Decision Support System (DSS)
Project Type	Decision Support System
Project Start Date	2024-12-01
Project End Date	2024-12-23
Project Owner	VNU-IS Canteen Management
Category of Project	Inventory and Sales Management
Project Manager/Department	IT Department

Objective: Provide detailed guidance for deploying the Decision Support System (DSS) at the VNU-IS canteen, including installation, configuration, testing, and operations.
Audience: Technical team, system administrators, and staff involved in deployment.

3.1.3.2. Pre-Deployment Requirements

Table 3.1.3.2. Pre-Deployment Requirements

	Server	8-core CPU, 16GB RAM, 500GB SSD

Hardware Requirements	Client Devices	Computers or tablets supporting modern browsers
	Network	Stable connection with at least 50Mbps speed.
Software Requirements	Backend	Python (Flask framework), Scikit-learn, TensorFlow
	Frontend	React.js, Chart.js/D3.js
	Database	MySQL 8.0
	Hosting	AWS EC2 and RDS
	Additional Tools	Docker, GitHub, Power BI/Tableau
Permissions & Access	Database Access	MySQL admin account
	Cloud Access	AWS account with administrative privileges
	Version Control	GitHub repository with push/pull rights

3.1.3.3. Deployment Process

Table 3.1.3.3. Deployment Process

Environment Setup	Server Configuration	1. Install Docker 2. Create containers for backend, frontend, and database 3. Set up AWS EC2 instance and RDS database instance
	Database Initialization	● Import tables and sample data from the SQL file ● Validate relationships between tables (based on the designed ERD)
	Backend Deployment	1. Clone the repository from GitHub. 2. Set up a Python virtual environment:

		`` python -m venv venv source venv/bin/activate pip install -r requirements.txt ``` 3. Start the Flask server: ``` flask run ```
	Frontend Deployment	1. Clone the frontend repository 2. Install dependencies: ``` npm install ``` 3. Run the application: ``` npm start ```
Configuration	Environment Variables	• Configure backend .env file: ``` DB_HOST=<your-database-host> DB_USER=<your-username> DB_PASS=<your-password> ``` • Set up the frontend to connect to the backend API
	Integration	• Connect the backend with the database • Test API endpoints using tools like Postman
Testing	Unit Testing	• Test backend and frontend modules • Use pytest or unittest for Python
	Integration Testing	Ensure correct data flow between frontend, backend, and database
	User Acceptance Testing(UAT)	• Conduct trials with canteen staff • Gather feedback on interface and functionality

3.1.3.4. Post-Deployment Steps

Table 3.1.3.4. Post-Deployment Steps

Monitoring	• Use AWS CloudWatch for performance monitoring • Set alerts for issues like CPU overuse or low disk space
Training	• Organize training sessions for staff

	• Provide user manuals and guides
Maintenance	• Perform weekly data backups to AWS S3 • Regularly update dependencies

3.1.3.5. Troubleshooting

Table 3.1.3.5. Troubleshooting

Issue	Resolution
Data synchronization failure	Check backend logs (/logs/error.log) and database connection.
The frontend application not running	Verify API URL configuration and browser console logs.
System downtime	Restart containers and check server resources.

3.1.3.6. Timeline

Table 3.1.3.6. Timeline

Phase	Start Date	End Date	Responsible Party
Preparation Phase	1st Dec, 2024	14th Dec, 2024	Project Manager
Testing Phase	15th Jan, 2025	27th Jan, 2025	QA Team
Deployment Phase	28th Jan, 2025	4th Feb, 2025	IT Team
Post-Deployment Phase	None	None	Support Team

3.1.3.7. Risk & Mitigation Plans

Table 3.1.3.7. Risk & Mitigation Plans

Risk	Mitigation Plan
Technology incompatibility	Conduct extensive compatibility testing and use standard protocols during integration.

Conduct extensive compatibility testing and use standard protocols during integration.	Create a detailed budget plan and conduct periodic cost reviews.
Project delays due to resource shortages	Allocate sufficient resources upfront and maintain a buffer in the project schedule.
Non-compliance with legal/data security	Regular audits and adherence to relevant data protection standards.
Adaptation issues among staff	Provide extensive training and ongoing support during the transition.
Shifts in market needs or competition	Continuously analyze market trends and update the system to match changing requirements.

3.1.3.8. Supporting Documentation

- References
  - System Requirements Document (SRD): Detailed specifications of hardware, software, and network requirements.
  - System Architecture Diagram: Visual representation of backend, frontend, database, and cloud services interaction.
  - Test Cases Document: Comprehensive list of unit, integration, and UAT test scenarios.
- Checklists

Table 3.1.3.8. Checklists

Pre-deployment Checklist	Verify server setup and Docker installation. Initialize the database and verify sample data import. Confirm all team members have access to required platforms.
Post-deployment Checklist	Validate data synchronization across all modules. Test API endpoints and frontend functionality. Conduct performance tests using AWS CloudWatch.

- Training Materials
  - User manuals for canteen staff.
  - Video tutorials on system navigation.
  - Quick reference guide for troubleshooting common issues.

3.1.3.9. Demo Deployment

The primary objective of this part was to deploy a simplified Decision Support System (DSS) to testing. Specifically, the prototype was designed to demonstrate core functionalities, including inventory monitoring, sales forecasting and it was not intended to be fully operational.

The DSS prototype was developed with several essential features:

- Forecast Dashboard: A visually interactive dashboard was created to provide insights into sales forecasts. It displays:
  - A 30-day aggregated sales forecast for the entire inventory.
  - Individual 7-day sales forecasts for specific products, allowing granular analysis.

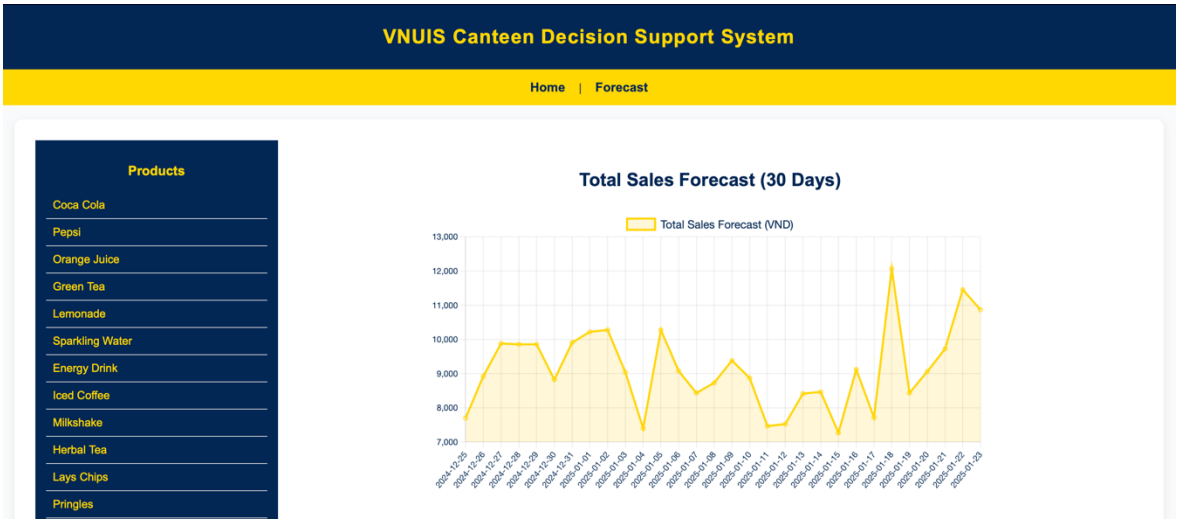

Figure 3.1.2. Total Sales Forecast Demo

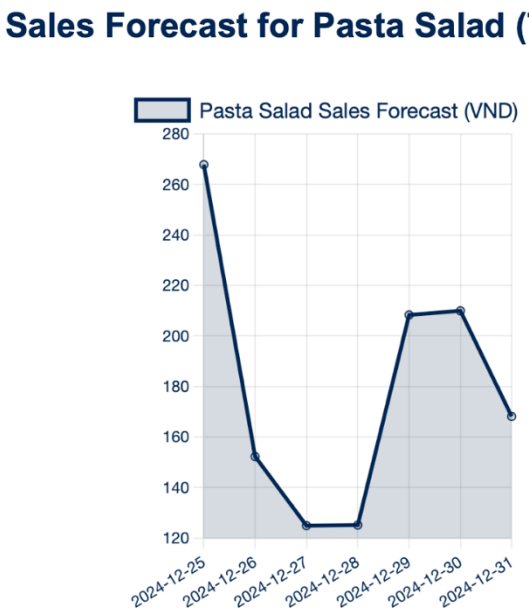

Figure 3.1.3. Sales Forecast for Specific Product Demo

- Inventory Monitoring: The system monitors inventory and provides critical alerts, including:
  - Expiring Soon: Products with expiration dates within the next 7 days.
  - Low Stock: Products with stock below a threshold.
  - Overstock: Products with excessive stock levels, ensuring optimal inventory management.

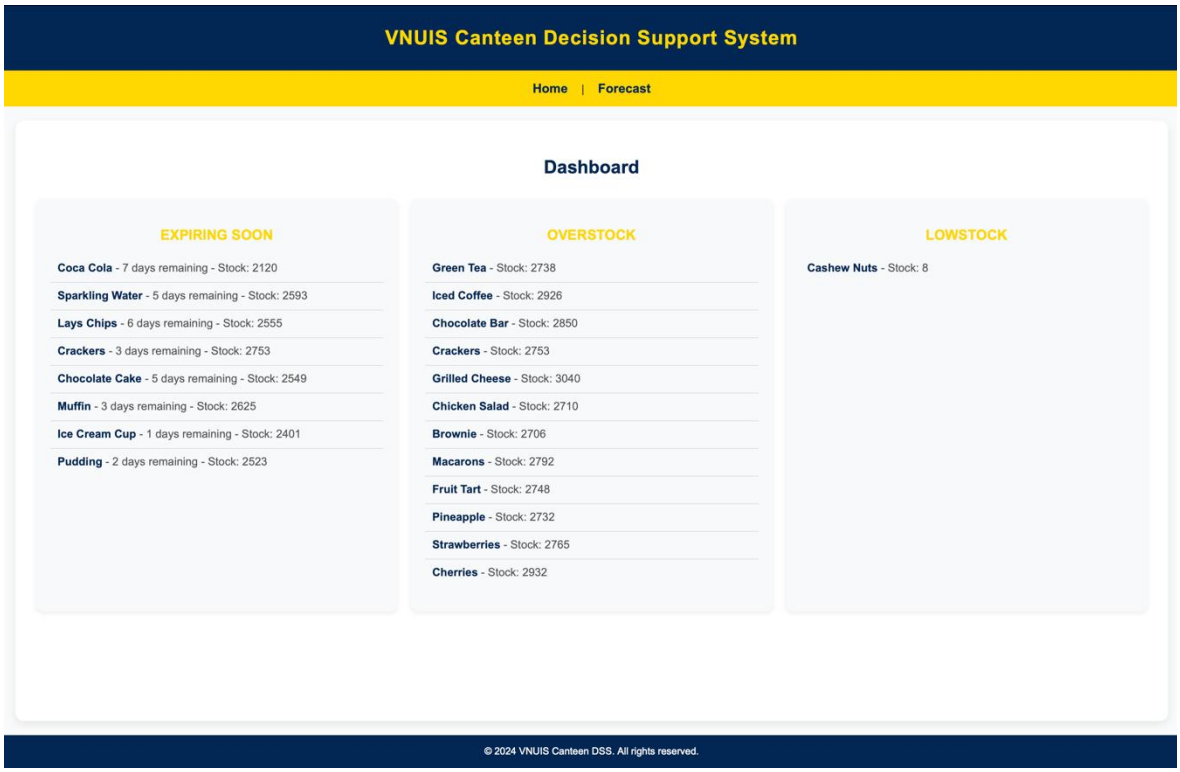

Figure 3.1.4. Inventory Monitoring Demo

In this part, we developed a sales forecasting process utilizing data from a MySQL database and the XGBoost regression model. This process consists of two primary components: forecasting total sales for the next 30 days and forecasting sales for individual products over the next 7 days.

First, we used SQLAlchemy to connect to the database and retrieve data. Total daily sales data was extracted from the Order table, then processed to generate features such as day, month, and year. The XGBoost Regressor model was trained on these features to predict total sales for the upcoming 30 days based on real-time data. The forecasting results were formatted into JSON for seamless integration with the user interface.

Next, we extracted detailed sales data for individual products from the *OrderItem* table, joining it with the *Product* and *Order* tables to calculate daily revenue for each product. Each product was processed and trained separately using an individual XGBoost model, with similar features as the total sales forecast. After training, the model was used to predict sales for each product over the next 7 days.

This process leverages real data from the database to simulate a forecasting system that closely resembles real-world applications. However, as this is a prototype system, the data might not yet be comprehensive enough for the model to achieve optimal performance.

3.2. Feasibility Analysis

A Feasibility Analysis is a comprehensive assessment conducted to determine the practicality and potential success of a proposed project. This analysis is crucial as it helps identify potential challenges and risks early in the project lifecycle, enabling informed decision-making and efficient resource allocation [6]. For this project, Feasibility Analysis ensures that the selected technologies align with the team's capabilities and project goals. It identifies cost-benefit scenarios, ensuring financial sustainability and operational effectiveness. This evaluation aids in proactively addressing challenges, minimizing risks, and optimizing resources, laying the groundwork for efficient implementation and improved decision-making in the canteen's operations.

3.2.1. Technical Feasibility

The technical feasibility of the proposed Decision Support System (DSS) involves evaluating the technologies used in the tech stack based on popularity, power, applicability, and the team's technical proficiency.

Table 3.2.1. Technical Feasibility

Technology	Popularity	Power	Applicability	Conclusion
MySQL	Widely used in the industry	Handles complex queries, supports transactions	Manages sales, inventory, and procurement records	Feasible
Python (Flask)	Top global programming language	Supports APIs, ML models, and data processing	Builds APIs, connects databases, forecasts demand	Highly Feasible
React.js	Popular for web development	Fast, modular, scalable UI framework	Creates interactive dashboards	Feasible
Chart.js, D3.js, Power BI, Tableau	Industry standards	Real-time dashboards, custom reports	Visualizes sales, stock levels, and forecasts	Highly Feasible
AWS (RDS, EC2, S3)	Leading cloud platform	Scalable, reliable, secure	Hosts the application, manages storage and backups	Extremely Feasible
Scikit-learn, TensorFlow	Widely used in AI projects	Advanced ML models, prediction algorithms	Forecasts sales and predicts inventory needs	Feasible

In any software project, ensuring that team members have sufficient technical skills is critical to project success. The Decision Support System (DSS) project needs specialized skills in key technologies such as Database Management, Backend Development, Frontend Development, Cloud Storage and Machine Learning. To evaluate the technical capabilities of our team members, we used Radar Chart, an effective visualization tool, to compare skills across different roles.Each team member has a specialized role, and their technical abilities are evaluated based on the technologies required in the project. Below is a list of roles and associated skills:

Project Manager		
Skills	**Have**	**Expect**
Project Planning	5	8
Risk Management	5	8
Communication	7	8
Decision Making	7	9
Leadership	5	8
Time management	9	8

Figure 3.2.1.1. Project Manager Radar Chart

For *Project Manager*, the skills furthest from the expected level are Project Planning and Risk Management, with a gap of 3 points (current: 5, expected: 8). The closest skill to the expected level is Time Management, where the current level (9) exceeds the expected level (8).

Backend Developer		
Skills	**Have**	**Expect**
API Development	6	8
Data Processing	7	8
System Security	7	9
Performance Optimization	6	8
Testing	7	8
Integration Support	7	8

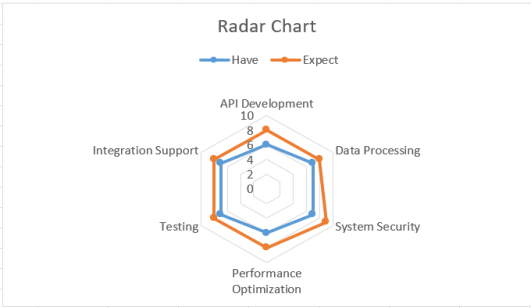

Figure 3.2.1.2. Backend Developer Radar Chart

For *Backend Developer*, the skills furthest from the expected level are API Development, System Security and Performance Optimization, with a gap of 2 points (current: 6, expected: 8 and current: 7, expected: 9). The closest skill to the expected level is Time Data Processing, where the current level (7) exceeds the expected level (8).

Database Administrator		
Skills	**Have**	**Expect**
Database Design	8	8
SQL Query Management	8	9
Data Security	7	8
Backup & Restore	6	8
Performance Monitoring	7	8
Backend Support	7	8

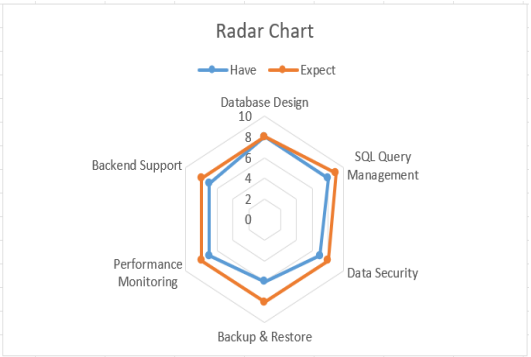

Figure 3.2.1.3. Database Administrator Radar Chart

For *Database Administrator*, the skill furthest from the expected level is Backup & Restore, with a gap of 2 points (current: 6, expected: 8). The closest skill to the expected level is Database Design where the current level (8) exceeds the expected level (8).

DevOps Engineer		
Skills	**Have**	**Expect**
System Deployment	7	9
Dockerization	7	8
CI/CD Pipelines	6	7
System Monitoring	8	8
System Security	7	9
Performance Optimization	7	8

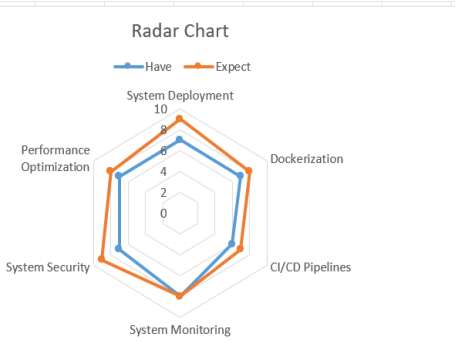

Figure 3.2.1.4. DevOps Engineer Radar Chart

For *DevOps Engineer*, the skills furthest from the expected level are System Deployment and System Security, with a gap of 2 points (current: 6, expected: 8). The closest skill to the expected level is System Monitoring, where the current level (8) exceeds the expected level (8).

Frontend Developer		
Skills	**Have**	**Expect**
UI Design	7	9
Data Feedback	7	8
Responsive Design	6	8
Performance Optimization	7	8
Testing	7	8
Teamwork Skills	8	9

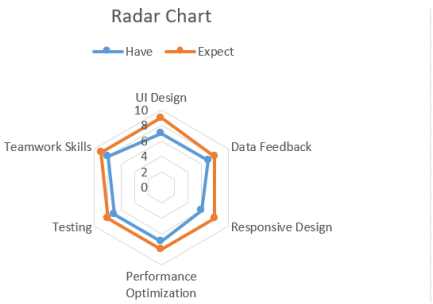

Figure 3.2.1.5. Frontend Developers Radar Chart

For *Frontend Developers*, the skills furthest from the expected level are UI Design and Responsive Design, with a gap of 2 points (current: 6, expected: 8 and current: 7, expected: 9). The closest skills to the expected level are Data Feedback, Performance Optimization and Teamwork Skills where the current level (7) exceeds the expected level (8) and the current level (8) exceeds the expected level (9).

Our assessment reveals that while technologies align perfectly with our requirements, there are manageable gaps in our team's expertise, particularly in areas like API Development, System Security, and UI Design. Despite these challenges, we are confident that they can be addressed through focused training, resource allocation, and leveraging our team's adaptability. For instance, while our backend developers need to strengthen their API Development skills, they are already proficient in Flask, which provides a strong foundation for learning advanced functionalities. To mitigate the identified gaps, we plan to organize internal training sessions, provide access to online learning platforms, and encourage collaborative knowledge sharing among team members. These efforts will not only enhance our technical capabilities but also ensure that the project remains on track. In conclusion, while there are areas for improvement, we believe that our team possesses the necessary foundation to execute

this project successfully. With the selected technologies and our planned measures to bridge skill gaps, we are confident in the technical feasibility of our DSS project.

3.2.2. Economic Feasibility

The Economic Feasibility analysis for the Decision Support System (DSS) Project for the Canteen Chain at VNU-IS serves as a critical evaluation of the project's viability in terms of costs and benefits over a three-year period. This assessment examines whether the anticipated economic and operational benefits justify the investment in hardware, software, employee training, and other related expenses. The primary goal of this analysis is to determine if the project can generate significant value, both financially and operationally, to enhance the efficiency and decision-making capabilities of the university’s canteen management system.

The proposed DSS aims to streamline daily operations, optimize resource allocation, and provide data-driven insights for improved decision-making. By comparing the total costs incurred with the quantified benefits over time, this study highlights the potential for achieving a positive return on investment (ROI) while ensuring sustainable operations.

Table 3.2.2.1. Cost Analysis

COST-BENEFIT ANALYSIS

PROPOSED PRODUCT / INITIATIVE / SERVICE	DATE CONDUCTED	COMPLETED BY
Decision Support System (DSS) Project for the Canteen Chain at VNU-IS	01/12/2024	Quynh Pham

QUANTITATIVE ANALYSIS

COST CATEGORY	YEAR 1	YEAR 2	YEAR 3	TOTAL
Hardware				
Servers (VPS Hosting)	$ 1.200	$ 1.200	$ 1.200	$ 3.600
Networking Equipment				
- Routers	$ 150	$ -	$ -	$ 150
- Wifi/Internet	$ 480	$ 480	$ 480	$ 1.440
- Cables/Racks	$ 50	$ -	$ -	$ 50
Computers/Keyboards/Mouse/CPU/ Monitor	$ 3.500	$ -	$ -	$ 3.500
Securities(Firewall Device)	$ 800	$ -	$ -	$ 800
Software				
Securities				
- Antivirus/ antimalware software	$ 200	$ 200	$ 200	$ 600
- IDS/IPS software	$ 700	$ 700	$ 700	$ 2.100
- VPN	$ 120	$ 120	$ 120	$ 360
- Firewall/UTM software	$ 700	$ 700	$ 700	$ 2.100
Tools				
- Power BI / Tableau	$ 840	$ 840	$ 840	$ 2.520
- Version Control System (e.g. GitHub, GitLab)	$ 100	$ 100	$ 100	$ 300
- Protyping Tools (e.g. Figma)	$ 720	$ 720	$ 720	$ 2.160
- Jira Software	$ 500	$ 500	$ 500	$ 1.500
- Microsoft Office/ Windows	$ 1.400	$ 1.400	$ 1.400	$ 4.200
Furniture and Fixtures				
Conference Tables/ Desks/ Chairs	$ 2.500	$ -	$ -	$ 2.500
Lightings/ Boards/ Cabinets	$ 1.000	$ -	$ -	$ 1.000
Location	$ 8.600	$ 8.600	$ 8.600	$ 25.800
Project Organization/ Support Costs				
Planning Project	$ 500	$ -	$ -	$ 500
Contract Negotiations	$ 200	$ -	$ -	$ 200
Maintainance	$ 1.000	$ 1.000	$ 1.000	$ 3.000
Operating Costs/ Other Costs				
Domain Name	$ 20	$ 20	$ 20	$ 60
Documents	$ 50	$ 50	$ 50	$ 150
Training Employees	$ 800	$ 800	$ 800	$ 2.400
Labor Salaries	$ 100.000	$ 100.000	$ 100.000	$ 300.000
TOTAL COSTS	$ 126.130	$ 117.430	$ 117.430	$ 360.990

Table 3.2.2.2. Quantitative Benefits

QUANTITATIVE BENEFITS

BENEFITS CATEGORY	YEAR 1	YEAR 2	YEAR 3	TOTAL
Economic Benefit				
Cost Reduction	$ 20.000	$ 25.000	$ 30.000	$ 75.000
Operational Benefits				
Compliance and Risk Reduction	$ 10.000	$ 12.000	$ 14.000	$ 36.000
Decision Support	$ 12.000	$ 15.000	$ 18.000	$ 45.000
Time Savings	$ 15.000	$ 20.000	$ 25.000	$ 60.000
Operation Efficiency	$ 25.000	$ 30.000	$ 35.000	$ 90.000
Resource Optimization	$ 8.000	$ 10.000	$ 12.000	$ 30.000
Technological Benefits				
Improved Decision Support System	$ 15.000	$ 18.000	$ 22.000	$ 55.000
Data Accuracy	$ 8.000	$ 10.000	$ 12.000	$ 30.000
Infrastructure Optimization	$ 10.000	$ 12.000	$ 15.000	$ 37.000
TOTAL BENEFITS	$ 123.000	$ 152.000	$ 183.000	$ 458.000

Table 3.2.2.3. Economic Feasibility Metrics

Economic Feasibility Metrics	
Metric	**Value**
Total Costs (3 years)	$ 360.990
Total Benefits (3 years)	$ 458.000
Net Economic Benefit	$ 97.010
Benefit-Cost Ratio (BCR)	1.27 (Feasible)
Payback Period	Year 2

Based on the detailed cost-benefit analysis:

- The **total cost** of the project over three years amounts to **$360,990**, while the **total benefits** are projected at **$458,000**, resulting in a **net economic benefit** of **$97,010**.
- The **Benefit-Cost Ratio (BCR)** of **1.27** demonstrates that for every dollar invested, the project yields $1.27 in benefits, confirming the project's economic viability.
- The **payback period** occurs within the **second year**, ensuring early recovery of the initial investment.
- Beyond financial gains, the project delivers operational efficiencies, improved decision-making, resource optimization, and technological advancements, which further enhance its value.

In conclusion, the **DSS Project for the Canteen Chain at VNU-IS** is economically feasible and offers substantial benefits that outweigh its costs. This project not only ensures a solid return on investment but also lays the foundation for long-term operational success and improved service quality across the canteen chain.

3.2.3. Operational Feasibility

3.2.3.1. Current Process Analysis

The existing processes at the VNU-IS canteen heavily rely on manual operations, creating significant inefficiencies. Sales data is recorded using spreadsheets, which is time-consuming and prone to errors. Inventory monitoring is conducted periodically, leading to reactive decision-making instead of a proactive approach. Communication between departments, particularly between Inventory and Procurement, is carried out via email, causing delays in replenishment requests and order processing. Additionally, the lack of demand forecasting tools limits the canteen's ability to respond to seasonal or event-driven fluctuations, resulting in frequent stock outs or overstocking issues.

3.2.3.2. Staff Usability

Skill Level: Most staff are familiar with basic tools such as spreadsheets and email but have limited exposure to automated systems. This creates a learning curve for adopting the Decision Support System (DSS).

Ease of Use: The DSS will provide a user-friendly interface with intuitive features such as real-time notifications, automated reports, and visual dashboards. These enhancements are designed to reduce manual workload and improve accuracy.

Adaptability: While there may be initial resistance to change, the DSS's ability to streamline processes and improve operational outcomes is expected to encourage staff acceptance. A comprehensive training program will ensure a smooth transition to the new system.

3.2.3.3. Process Optimization with DSS

The integration of the DSS will address current inefficiencies by:

- **Real-Time Inventory Management:** Automating stock level updates and alerting staff of low-stock situations to prevent stockouts.
- **Automated Sales Forecasting:** Providing data-driven predictions to handle seasonal and event-driven demand fluctuations more effectively.
- **Streamlined Communication:** Replacing email-based communication with integrated workflows for inventory and procurement processes, enabling quicker response times.
- **Improved Decision Support:** Offering data visualizations and actionable insights to support better decision-making regarding stock replenishment and product offerings.

3.2.3.4. Staff Training and Support Plan

To ensure successful adoption of the DSS, the following measures will be implemented:

- **Role-Specific Training:** Separate training sessions will be provided for sales, inventory, and procurement staff to familiarize them with their respective modules in the DSS.
- **Continuous Support:** A helpdesk and regular workshops will be established to address staff concerns and improve usability over time.
- **Knowledge Resources:** Online tutorials and user guides will be made available for ongoing learning.

3.2.3.5. Customer Impact

The proposed system will enhance customer satisfaction by:

- Ensuring consistent availability of high-demand products through improved inventory management.
- Reducing transaction delays with streamlined processes.
- Offering better product choices based on sales trend analysis.

3.2.4. Schedule Feasibility

Analyzing the feasibility of the project timeline is an essential part of ensuring that the project objectives can be completed on time and efficiently. This section provides an overview of evaluating the timeline feasibility based on the project scope, available resources, and factors affecting the implementation within the expected deadlines. The analysis will be based on the *Work Breakdown Structure (WBS)* and *Gantt Chart*, which help assess the completion of key tasks and identify potential risks.

The project has been divided into 6 major phases to ensure efficient planning and execution. A detailed *WBS* provides a structured view of all project tasks and their relationships.

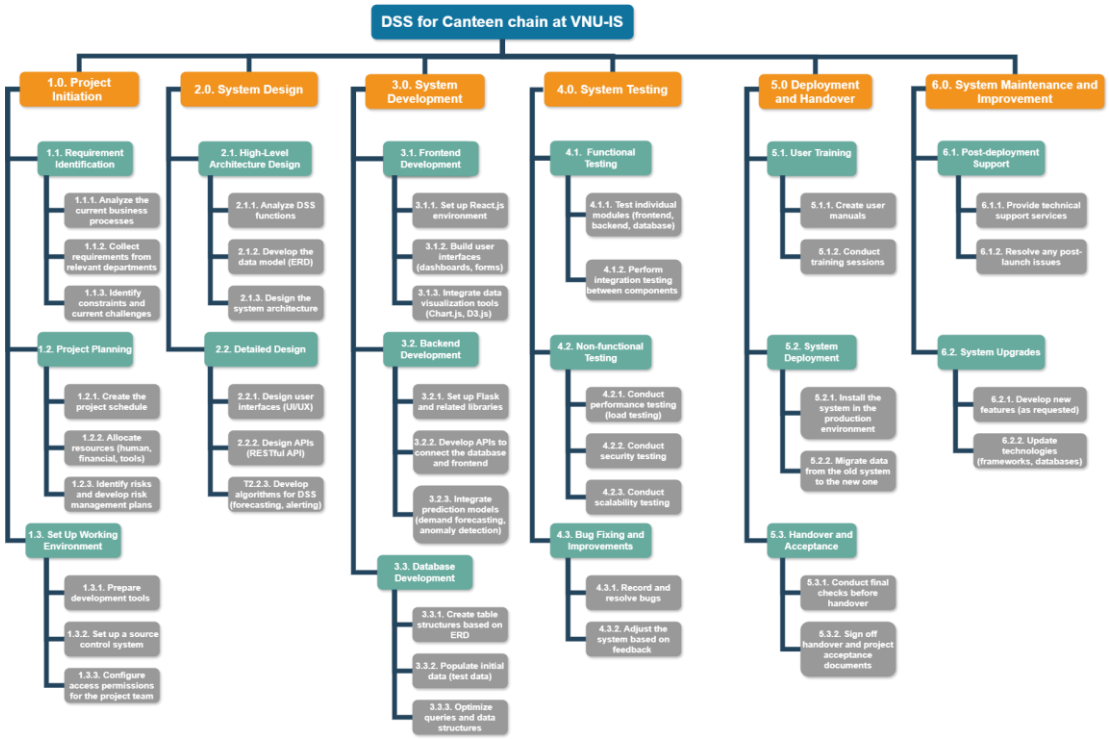

Figure 3.2.4.1. DSS for VNU-IS Canteens WBS

A *Gantt Chart* was created using Jira to track the project's progress, with each task assigned specific timeframes, highlighting milestones and dependencies.

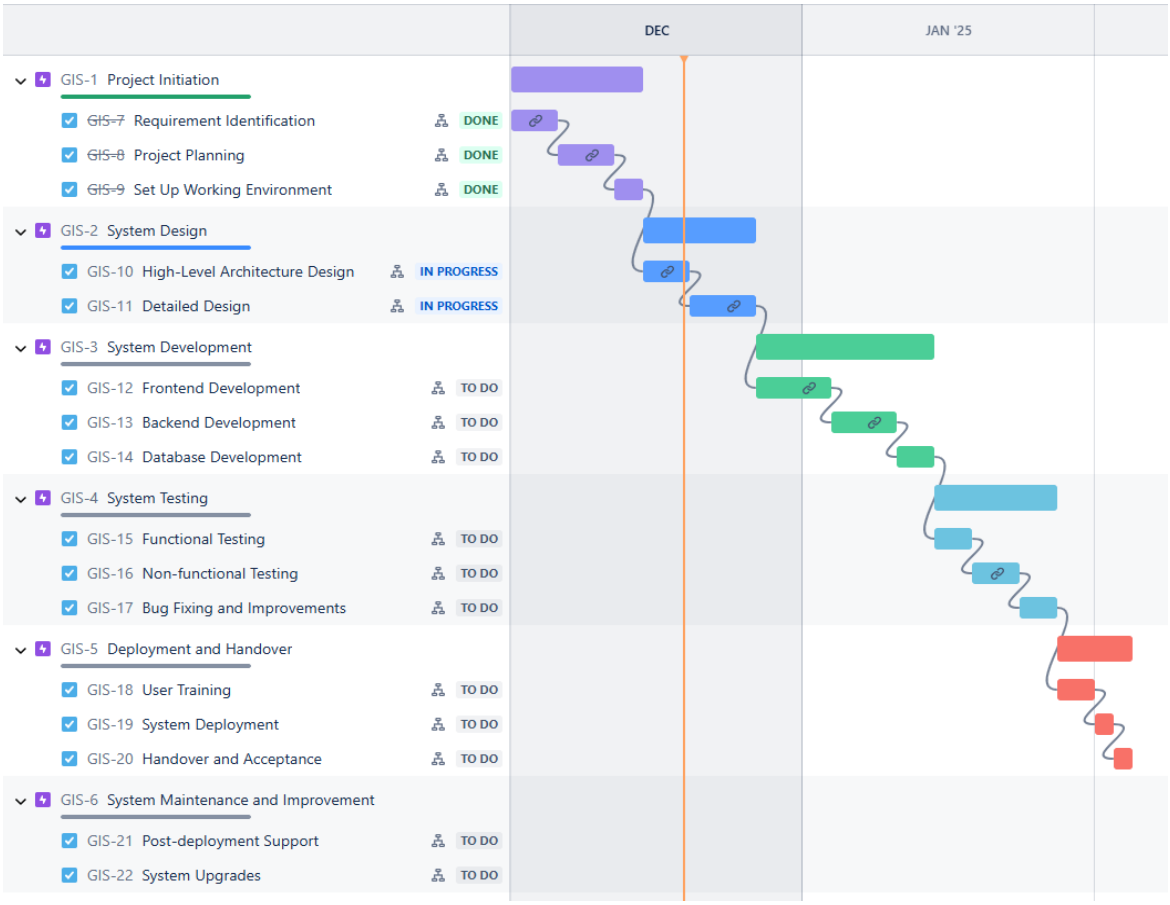

Figure 3.2.4.2. DSS for VNU-IS Canteens Gantt Chart

Based on the analysis from the WBS and Gantt Chart, the feasibility of the project timeline can be assessed through the following factors:

- **Task Durations:** The durations for each task have been realistically estimated based on the team's capabilities. Key tasks like development and testing have been appropriately scheduled to ensure that progress is not delayed.
- **Dependencies:** Dependencies between tasks are clearly defined in the Gantt Chart, ensuring no conflicts or omissions during project execution.
- **Risk Management:** Potential risks, such as delays in development, have been anticipated and mitigated. Specifically, hidden buffer time allows for unexpected delays, and regular progress meetings will help address issues early.

With the detailed analysis through WBS and Gantt Chart, **the project timeline is deemed feasible,** as long as risks are effectively managed, and team performance remains stable. Continuous monitoring through regular progress meetings and timely adjustments will help ensure that the project is completed on schedule and meets the expected quality.

Conclusion

The development of the Decision Support System (DSS) for VNU-IS Canteens addresses critical operational inefficiencies in inventory management, sales forecasting, and customer satisfaction. With the identified challenges of fragmented data management, overstocking, understocking, and unreliable demand forecasts, the DSS was strategically designed to enhance decision-making processes and align with the canteen's operational objectives.

Key achievements of the DSS include:

- Enhanced Inventory Management:
  - The DSS uses historical and real-time sales data to provide precise inventory tracking.
  - Automated alerts for overstock and low-stock scenarios help reduce financial losses due to spoilage and ensure popular items are always available.
  - Optimized stock levels directly contribute to efficient storage use and cost reduction.
- Accurate Sales Forecasting:
  - By integrating demand forecasting models that account for seasonal trends, event-driven surges, and historical patterns, the DSS ensures preparedness for fluctuating demands.
  - Improved forecasting minimizes resource wastage during low-activity periods and maximizes revenue opportunities during peak times like exams or school events.
- Improved Customer Satisfaction:
  - Ensuring the availability of high-demand items and avoiding stock outs creates a seamless and reliable customer experience.
  - Data insights empower the canteen to introduce or enhance product offerings based on customer preferences and consumption patterns.
- Alignment with SMART Objectives:
  - The DSS achieves specific goals by addressing clearly defined inventory and sales challenges.
  - Its performance is measurable through reduced spoilage, improved stock turnover rates, and enhanced forecast accuracy.
  - The system is achievable within the constraints of the canteen's budget and existing infrastructure.

- It is relevant to the strategic aim of providing better service to the VNU-IS community while maintaining cost-efficiency.
- Finally, the DSS's impact is time-bound, with measurable improvements observable in the short term and scalability for long-term benefits.

In the context of Global Decision Support Information Systems (GDIS), this project demonstrates how a well-designed DSS can address local decision-making needs while maintaining global standards of efficiency and adaptability. By enhancing the canteen’s operations, the DSS provides a model that other facilities within the VNU-IS system can adopt and benefit from.

In conclusion, implementing this Decision Support System marks an important step toward improving the efficiency and customer service of the VNU-IS canteens. The system not only solves current challenges but also paves the way for sustainable growth and innovation, ensuring the canteens can effectively meet future demands.

References

[1] bdc, "What are SMART objectives?," *BDC.ca*, Jun. 21, 2022. https://www.bdc.ca/en/articles-tools/entrepreneur-toolkit/templates-business-guides/glossary/smart-objectives (accessed Dec. 13, 2024).

[2] I. Belcic and C. Stryker, "What is an Entity Relationship Diagram? | IBM," *www.ibm.com*, Jun. 19, 2024. https://www.ibm.com/think/topics/entity-relationship-diagram (accessed Dec. 16, 2024).

[3] P. Kotler and K. L. Keller, "(PDF) Marketing Management," *ResearchGate*, Jan. 1994. https://www.researchgate.net/publication/40349508_Marketing_Management (accessed Dec. 13, 2024).

[4] Chopra, S., & Meindl, P. , "Chopra, S., & Meindl, P. (2016). Supply Chain Management Strategy, Planning, and Operation (6th ed.). Pearson. - References - Scientific Research Publishing," *www.scirp.org*, 2016. https://www.scirp.org/reference/referencespapers?referenceid=3203314 (accessed Dec. 13, 2024).

[5] R. M. Monczka, R. B. Handfield, L. C. Giunipero, and J. L. Patterson, *Purchasing and Supply Chain Management*. Routledge, 2015.

[6] iCertGlobal, "Feasibility Study and Its Importance in Project Management," *icertglobal.com*, Sep. 11, 2022. https://www.icertglobal.com/feasibility-study-in-project-management/detail? (accessed Dec. 20, 2024).

[7] S. Ellis, "What is a Tech Stack? Definition, Types, Benefits | Enterprise Tech News EM360," *em360tech.com*, Aug. 25, 2023. https://em360tech.com/tech-article/tech-stack-definition (accessed Dec. 15, 2024).

[8] M. Watson, "What Is A Tech Stack and Why Do You Need It - Full Scale," *Full Scale*, Oct. 30, 2023. https://fullscale.io/blog/what-is-a-tech-stack-and-why-you-need-it/ (accessed Dec. 20, 2024).