Advanced Information Systems Analysis and Design Final Project Report

“Gotham Treats”

Lecturers:
Win Ce, S. Kom., M.M.
Dr. Ing. Rein Suadamara, S. Kom., M. Si.

Teaching Assistant:
Justin Darwin Sentosa

Report by:
Nadja Jasmine Donosepoetro - 2602119184
Dycinta Fedianka - 2602185942
Ajhe Ray Delrian - 2602176894
Nicholas Darmawan - 2602158550
Eliott Bastien Mathieu Rivet - 2602214791

Table of Contents

Chapter I: Project Introduction

1.1 Introduction

This final report is made to fulfil the requirements of the final project for the course: Advance Information Systems Analysis and Design (ISYS6676003). The final report includes a detailed explanation on the company profile of Gotham Treats, description on UML diagrams, and solution modelling for an inventory management system.

1.2. Company Overview

Gotham Treats is a small local B2C dessert bakeshop based in South Jakarta, Indonesia. Specialising in their own unique take on American classic sweets, Gotham Treats offers a delectable assortment of cookies, milks, dessert boxes, pies, and other sweet treats. Their culinary creations are heavily inspired by their deep appreciation for the goods found in New York City, aiming to bring a touch of that city's flavour and charm to the local market.

Operating within the retail industry, Gotham Treats caters to customers in the food and beverage sector, specifically dessert and bake sector. As a B2C (business-to-consumer) establishment, their primary focus is on selling products directly to individual consumers for personal use. In this case, Gotham Treats offers a delightful array of ice cream, freshly baked goods, and various related treats that satisfy even the most discerning sweet tooth.

What sets Gotham Treats apart is their commitment to quality and attention to detail. Each sweet treat is carefully crafted in small batches, ensuring that every item is made with the utmost care and precision. By baking their goodies fresh, Gotham Treats guarantees the highest level of flavour and freshness, ensuring a delightful experience for their customers.

Currently, Gotham Treats operates in two outlets located in Pakubuwono and Pantai Indah Kapuk (PIK). These strategic locations allow them to reach a wide customer base and cater to the local community's cravings for indulgent desserts. By providing convenient access to their delectable offerings, Gotham Treats has successfully created a local hub for dessert enthusiasts, serving as a go-to destination for New York-inspired treats.

Gotham Treats has four main operating locations: main office, central kitchen, and two retail outlets located in Pakubuwono and Pantai Indah Kapuk. The central kitchen is utilised as the main production location, where kitchen staff will produce the goods. The office is used for managerial activities that include purchasing, finance, marketing & communication activities. The two outlets serve as the retail location where the products will be sold after production.

In summary, Gotham Treats is a small local bakeshop in South Jakarta that brings the essence of New York City's sweets to the local market. With their commitment to quality, fresh ingredients, and a range of delectable treats, they have carved a niche for themselves in the retail industry. With their two outlets strategically positioned in prime locations, Gotham Treats continues to captivate customers and spread joy through their delightful creations.

1.3. Product Overview

Gotham Treats sells a variety of sweets with six product categories - cookies, dessert box, pudding, milk, pie and coffee. With a wide selection of unique flavors, these products provide customers with a delightful and distinctive tasting experience, enticing them to explore and indulge in the various offerings available.

Gotham Treats Menu	
Cookies	The Nolita The Chelsea The Upper East Side The Hells Kitchen The Greenwich Village The Soho The East Village
Dessert Box	The Little Italy - Tiramisu Dessert Box The Little Italy - Speculoos Dessert Box The Little Italy - Crunchy Milo Dessert Box The Little Italy - Es Teler Dessert Box
Pudding	The Broadway - Banana Pudding The Staten Island Pudding - Confetti Cake
Milk	The Dumbo Cereal Milk The Bronx Coffee Cereal Milk

	The Central Park Regal Cereal Milk
Pie	The Brooklyn Pie - Original The Brooklyn Pie Chocolate Rum
Coffee	Cold Brew Cold Brew Lite

The products are produced in the kitchen, one of the three main operating locations. When the products are sent to the outlet for retail, most of the products are sent as finished goods to later be sold in the outlet. For cookies and ice cream, the product is sent as work-in-progress, it is sent as dough, and the production will be continued for the baking process in the outlet.

1.4. Roles and Responsibilities Overview

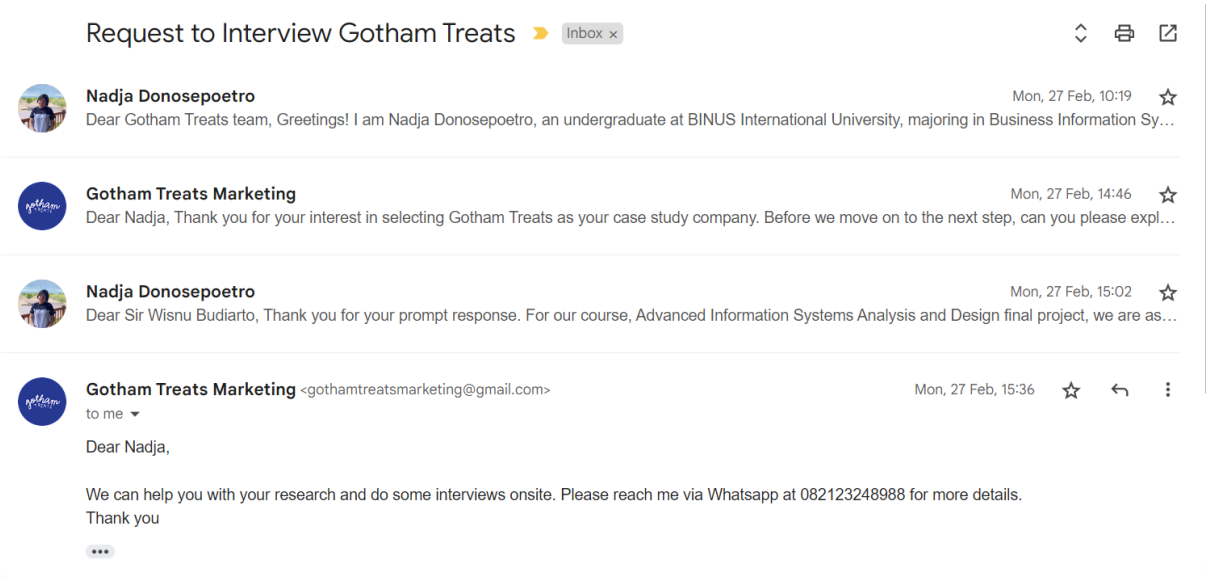
Within Gotham Treats, there are five actors involved with the proposed solution - Head of Operations, Purchasing Staff, Kitchen Staff, Outlet Staff and Finance Staff. These actors engage towards the system to ensure that product manufacturing and inventory is managed efficiently.

Actor	Location	Responsibilities
Head of Operations	Main Office, Central Kitchen, Outlet	• Decide on the make to stock quantity per day for each product every month for material requirement planning • Evaluate each incoming purchase request and send purchase orders to the vendor through form. • View quality control report once purchase order has arrived. • Submit payment request once purchase order is satisfied.
Purchasing Staff	Main Office	• Handles evaluating purchase Evaluate each incoming purchase request and send purchase order to vendor through form • Submit payment request once purchase order is satisfied
Kitchen Staff	Central Kitchen	• Update status and arrival of the

		items according to the purchase order ● Submit quality control report after purchase order item has been arrived into the central kitchen ● Create and monitor delivery list to track delivery items (inventory goods) and location from kitchen to outlet ● Manage goods inventor
Outlet Staff	Outlet	● When an incoming bulk order is made, they need to submit a purchase request ● Once purchase order is approved, they are able to submit payment request and view its details including status ● Outlet staff are able to view production planning and manage their current inventory ● With an incoming delivery of ingredients, they are able to monitor and keep up-to-date with delivery
Finance Staff	Main Office	● Evaluate the incoming payment request and do the necessary transactions ● Handle and execute the transaction and invoice ● Update the payment request status accordingly
Driver	Mobile	● Pick up products from kitchen ● Deliver products to outlet

1.5. Data Collection Evidence

To further understand and analyse the company's operations and the roles in the company, an interview and an observation was conducted. In late February, the group had emailed Gotham Treat's email to arrange an interview with the staff.



By exchanging emails, the group was able to schedule an interview time in early March to inquire regarding operations and gain an understanding on how Gotham Treats operate as a dessert bake shop.

We interviewed Sir Andi Buana, Head of Operations, who manages and supervises the production, outlet operations and inventory management of Gotham Treats. We were able to ask regarding his responsibilities and habits as Head of Operations as well as ask regarding the operations to get a greater understanding of how Gotham Treats operates. As he is Head of Operations, it is noted that he often moves around from the main office, central kitchen and outlet.

Aside from collecting data and insight from the interview, the group had also visited the outlet two times to conduct observation. We gained from our observations on how the outlet employees operate, bake and conduct their sales into the system.

Chapter II: Business Overview

2.1. Current Business Process

With Gotham Treats serving as a retail and dessert shop, their current business process is heavily following Porter's Value chain where there are five primary activities.

Inbound Logistics

Inbound logistics of the company include vendor management, purchase orders, shipping, and delivery, receiving delivery, storage and inventory management, and quality control. This requires establishing contacts with vendors, reaching agreements, and ensuring the timely supply of raw materials. When orders are submitted, transportation plans are prepared to get materials to the production plant based on production requirements. When the materials are delivered, they are examined for quality and appropriately stored. Quality control ensures that resources reach the production line, and quality control measures are employed throughout the process to maintain flavour, freshness, appearance, and safety standards.

Outbound Logistics

The outbound logistics of Gotham Treats comprise packaging the finished products, processing client orders, completing those orders by selecting the appropriate products from inventory, scheduling shipping and distribution, tracking shipments, and providing customer support. To prepare the packaged products for dispatch, several shipping methods are employed, and records are kept to monitor the process and ensure on-time delivery. Effective outbound logistics ensures that clients receive fresh and high-quality cookies on time, resulting in a successful customer experience and a stable distribution network.

Marketing & Sales

Gotham Treat's marketing and sales strategy includes market research, branding, product creation, advertising, sales channels, pricing, and customer relationship management. The company creates and innovates new products, employs a variety of advertising and promotional methods, selects sales channels and distribution networks, determines pricing strategies, and focuses on building strong customer connections. To identify its target clients and competitors, the company undertakes market research. In a competitive industry, these

tactics strive to promote customer happiness and loyalty while also increasing sales and creating demand.

Service

The company’s service activities involve offering an appealing product display, knowledgeable staff, sampling opportunities, customising possibilities, attractive gift wrapping, a speedy checkout procedure, post-purchase support, and opportunities for customer interaction and feedback are all examples of general customer service provided by a dessert company. By tastefully displaying the products, providing information and help, allowing consumers to test products such as cookies, milk, and ice cream. The company focuses on providing a delightful and tailored customer experience. The fast checkout procedure, post-purchase help, and engagement programs make it easier to build client loyalty and contentment.

Operations

Developing recipes, collecting materials and mixtures, planning production, mixing and preparing dough, baking, cooling, and packing, same-day sales and distribution, inventory management, and quality control are all part of a cookie company's general operations. The company develops and improves cookie, milk, ice cream and other product’s recipes, purchases quality ingredients, and creates a production plan to determine the volume and variety of cookies to be produced each day. Before being moulded into cookies and baked in ovens, the dough is produced with precise attention to temperature and baking times. The ingredients are precisely measured and mixed. Once the cookies have cooled, they are wrapped to keep them fresh and safe until delivery. The company develops sales channels and distribution systems in order to give freshly made cookies the same day and ensure speedy delivery to retail stores or other outlets. Inventory levels are regulated to avoid waste while meeting demand, and quality control processes are implemented at every stage of manufacture to ensure consistency in flavour, appearance, and texture. If these operations are carried out quickly, the company may provide customers with freshly baked cookies while maintaining high standards.

2.2. Current Application / Information System

The application / information system that is used by Gotham Treats currently has only two softwares: Mekari Talenta and Moka POS. As of now, Gotham Treats has two softwares that are used for the purpose of Human Resources system with Mekari Talenta and Point-of-Sale system with Moka POS.

In this instance, the Head of Operations uses Mekari Talenta to monitor and track employee attendance, performance and salary. While, outlet employees utilise Moka POS to input and enter in the sales made from the outlet in a certain period.

Specifically, in terms of managing inventory and production, there is no central system and instead, the outlet employees are seen with papers to document their production process and the products baked from their outlet. Furthermore, from those papers, they document it onto an Excel sheet on their laptop. With that said, Gotham Treats does not facilitate an inventory management and production system that is integrated digitally.

2.3. Assumption and Analysis

While designing the system and solution, we assumed that a new digital system covering all aspects of the organisation would be necessary. An operations system, a procurement system, and an inventory management system would be included in this system. By merging various systems into a single digital platform, the company may achieve a more streamlined, transparent, and efficient method for all of their regular business operations. We are aware that the company's basic operations, the manufacturing procedures in the kitchen and outlet, are critical. The design of our digital platform aims to improve the experience of kitchen and outlet employees by making it easier for them to execute duties such as delivery requests, inventory updates, and product manufacturing. Our system design is ultimately focused on the firm's production process, which comprises varied activities such as purchase orders, requests, payment processing, and more, all devoted to the manufacturing of the company's products.

2.4. Problems Identified

Inventory Management:

The absence of an inventory management system at Gotham Treats poses challenges such as manual tracking, inaccurate stock levels, inefficient reordering processes, limited insights, and operational inefficiencies. Gotham Treats currently tracks their inventory manually on paper, then the purchasing team will put it into a journal (app).

Sales Insight:

Gotham Treats also does not have sales insights on their system, which hampers their ability to improve performance and cater to customer needs. Without the means to analyse sales patterns, customer preferences, and product performance, Gotham Treats lacks crucial insights that could guide decision-making processes, such as optimising their offerings, identifying popular items, and tailoring their strategies to enhance customer satisfaction.

Managing Branches:

The issue for Gotham Treats as it grows its business is it is hard to manage sales and trends at each outlet. Finding patterns, trends, and differences in branch performance becomes challenging without a system in place to give centralised visibility and analysis of sales data across their locations. Due to this, Gotham Treats experience difficulty in managing their branches.

Chapter III: Future Overview of Business

3.1. Future Direction

Gotham Treats wants to open more locations for its outlets. Through this they hope to be able to reach a larger customer base and strengthen their brand recognition. To assure the success of each new office, they will concentrate on locating important areas and selecting the right markets. Apart from that, Gotham Treats will emphasise putting in place a centralised system as the number of branches increases. This technology will permit them to manage their operations more efficiently.

Gotham Treats is embarking on a strategic expansion plan to open more outlets, aiming to reach a broader customer base and strengthen brand recognition. By carefully selecting key areas and target markets, they aim to maximise their market presence and attract new customers while catering to existing demand. This expansion will allow Gotham Treats to offer their unique and delicious treats to a wider audience.

To ensure the success of each new branch, Gotham Treats recognizes the importance of efficient operations management. They will prioritise the implementation of a centralised system as they expand. This technology will provide centralised visibility and control over inventory management, sales tracking, and performance analysis across multiple locations. By utilising a unified system, Gotham Treats can maintain operational consistency, monitor branch performance, and make informed decisions to optimise resources and enhance customer satisfaction.

As Gotham Treats expands its outlets, they will also ramp up production and sales. With a larger network of branches, they will have the capacity to produce a greater volume and variety of their signature sweets. This expansion enables them to cater to diverse tastes, accommodate a larger customer base, and expand their market reach. To support increased production and sales, Gotham Treats will strengthen their inventory management and supply chain capabilities, establishing strong supplier relationships and implementing efficient ordering and replenishment processes.

In summary, Gotham Treats' strategic expansion plan involves opening more outlets, implementing a centralised system, and increasing production and sales. Through this

approach, they aim to reach more customers, strengthen their brand, and ensure operational excellence. By carefully selecting locations, utilising technology, and enhancing inventory management, Gotham Treats is poised to achieve growth and success in the competitive dessert market.

3.2. Proposed Solution

Our proposed solution for Gotham Treats is to implement a digitised inventory management system that automates inventory tracking, production monitoring, and procurement processes. This system will eliminate the need for manual paper-based tracking, ensuring accurate and real-time inventory information. It will enable the actors at Gotham Treats to easily update stock levels, automate reordering based on predefined thresholds, and streamline inventory management.

Additionally, we will integrate a comprehensive sales insights module into the system. This module will capture and analyse sales data, customer preferences, and product performance. By leveraging data analytics, Gotham Treats will gain valuable insights into sales patterns, popular items, and customer preferences. These insights will inform data-driven decision-making, enabling the optimization of offerings, tailored strategies, and improved customer satisfaction. Furthermore, we will provide a centralised system that offers visibility and analysis of sales data across all branches, enabling outlets and central management to access real-time sales information, identify trends, and compare branch performance. This centralised visibility will facilitate effective decision-making, resource allocation, and standardisation efforts across branches, addressing the challenges associated with managing multiple locations. Overall, our proposed solution will streamline operations, improve efficiency, and provide valuable data-driven support to help Gotham Treats thrive and grow.

3.2. Solution Implementation

In terms of implementing the solution of a digital inventory system to Gotham Treats, the programming language Java can be used as the main programming language for the system. With that, to store and retrieve data, a database management system (DBMS) can be used in effect to call the respective tables from a certain database onto the JFrame or frame viewed by the user.

While much of the data is stored on the XAMPP Database, data such as Order and Order history is able to be accessed through third party API. At the time we had interviewed, they did not have a system where they are paper-based. With the current observation on the business process, they use Moka POS as their main point-of-sale system held in the outlets. With the new inventory management system utilising order data, MOKA API can be used for the developers to integrate the system along with its data onto the inventory management system. Hence, with that API enabled in the system, data integration is possible and sales analytics is able to be viewed onto the end-user.

Chapter IV: Business Process Analysis

4.1. Value Proposition

	Operational Excellence	**Product Leadership**
Core Business Processes	With high customer demand, there is need to match with the supply	Create innovative products (flavours and bakeshop items) and explore the reach and sales in the market
Organization and Structure	There is a central authoritative with power, mainly the co-founders and the managers are in power	The organisation structure is organic and loose
Management Systems	There is standard operating procedure to ensure high quality is achieved.	Through product innovation, risk can be expected and it rewards people for innovation and creativity in its products.
Culture: Mindsets and Behaviour	The focus is on making the system effective and efficient.	The focus is on innovating and creating a breakthrough in the market.

4.2. Value Analysis

Porter's Value Chain Analysis of Gotham Treats

Supporting Activities

Firm Infrastructure focuses on management, finance and legal				Margin
Human Resource Management focuses on recruiting, managing and motivating employees				
Technology Department focuses on management information systems, hardware and IT				
Procurement focuses on purchasing the equipment				
Inbound Logistics	**Operations**	**Marketing & Sales**	**Service**	
Gotham Treats has its suppliers for obtaining raw material and handles inventory control.	Gotham Treats uses those raw materials to convert and create finished goods into production	Gotham Treats utilizes their social media platform to conduct marketing and perform point-of-sale onsite and online	Gotham Treats opens feedback and inquiries from customers, offering customer service.	

Primary Activities

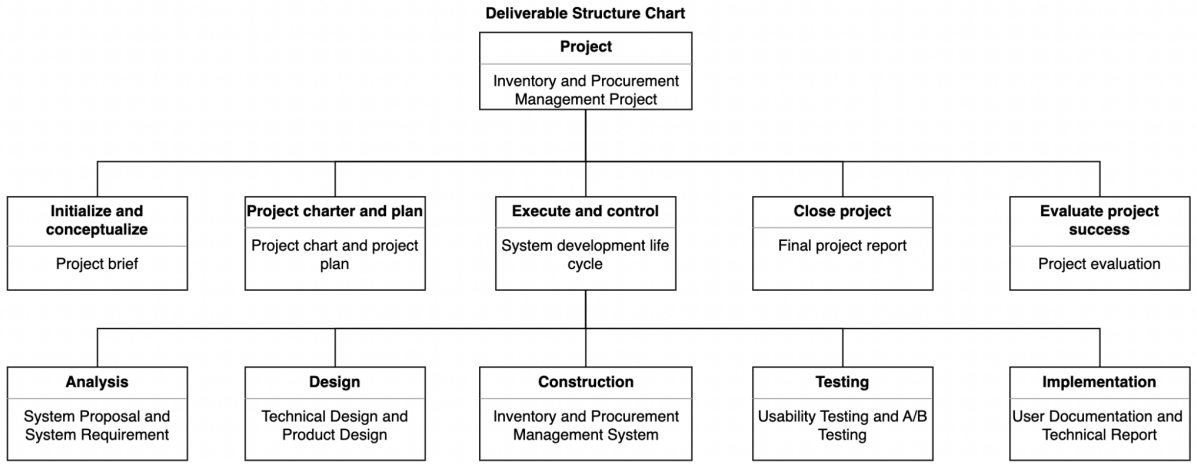
4.3. Primary and Supporting Activities & Values to Achieve

Primary Activity	**Values to Achieve**	
Inbound Logistics	Consistent Quality	Gotham Treats needs consistent quality control for raw materials inbound logistics to make sure their sold product stays consistent in quality.
	Time Efficient	Gotham Treats needs to ensure that the materials they order will be delivered in a timely manner, leaving them the right amount of time to go on with their operations.
	Cost Effective	Gotham Treats wants to purchase their raw materials at the best price as possible to maximise their profit.
Operations	Established Standard	Gotham Treats needs to have an established standard in procedure so the employees can produce a consistent day-to-day quality for their products.
	On-time	Gotham Treats needs to bake their cookies and make their ice cream on-time so every time a customer orders they are always ready.
	Effective	Gotham Treats needs to optimise processes, resources, and workflows to ensure efficient production, timely service, and seamless customer experiences.
Marketing & Sales	Customer Focus	In order to deliver the right goods or services to customers, the marketing and sales teams for Gotham Treats must work together to understand their needs and preferences.
	Lead Generation	By developing marketing strategies, producing content, and running advertisements to draw in potential clients, marketing teams generate leads. Sales teams strive to turn those leads into actual sales.
	Collaboration	To make sure that they are aligned in their approaches to generating revenue, both teams collaborate closely. Sales teams give feedback on the efficacy of marketing campaigns, while marketing teams supply sales teams with leads and customer insights.

Service	Customer Satisfaction	Gotham strives to consistently exceed customer expectations by providing exceptional products and services, resulting in a positive and delightful experience that leaves customers highly satisfied.

Supporting Activity	**Values to Achieve**	
Firm Infrastructure	System Robustness	Gotham Treats must have IT systems, processes and policies in place that are robust enough to support store operations reliably and consistently
HRM	Recruitment and Retention	Gotham Treats HR team must focus on recruiting and keeping top employees by fostering a great workplace culture, providing competitive pay and benefits, and giving competitive benefits.
	Development and Satisfaction	With training, coaching, and mentorship programs, Gotham Treats HR management seeks to enhance employees' professional development and career advancement inside the company, fostering employee engagement and satisfaction.
Technology	Data Collection	Gotham Treats can use technology to gather and analyse information on customer behaviour, sales trends, and inventory levels. Making informed decisions about menu items, pricing, and promotions is possible using this data.
	Efficiency by Integration	Gotham Treats must have integration of technology to streamline their operations for more efficiency and increasing sales. For example, technology to simplify their point-of-sales and speed up the ordering process.
Procurement	Reliable Quality	To produce excellent quality in their product output, Gotham Treats will need reliable, good quality machinery and tools at their disposal.
	Always in-stock	When there is a problem in Gotham Treats' tools and/or machinery, they will need to have a replacement as soon as possible so they can get back to their full work efficiency.

4.4. Projection of Business Scenario



4.5. Process Selection Matrix

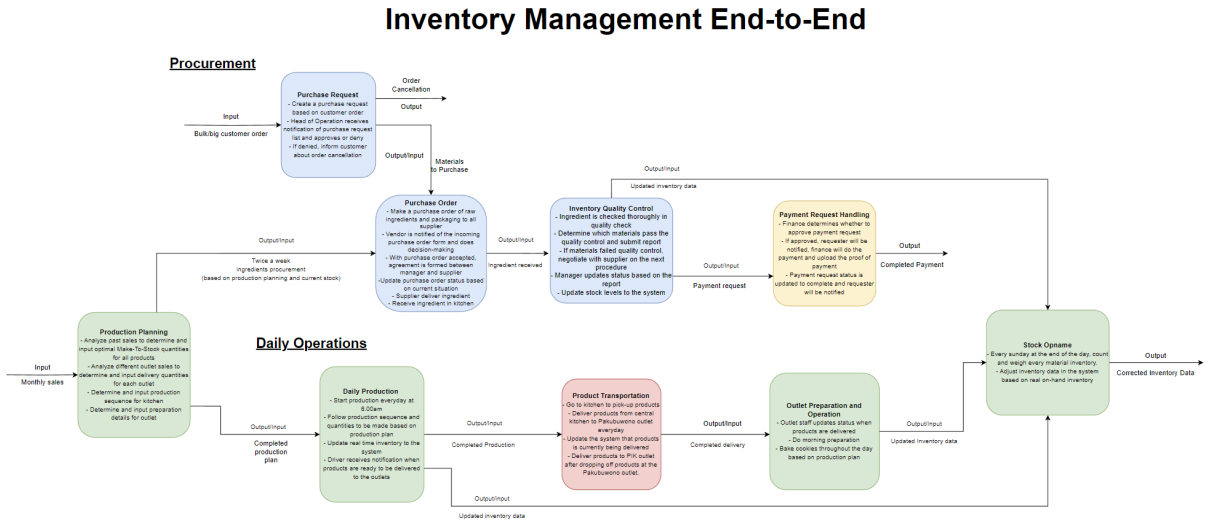
Classification	Type	BP	Activities	Business Scenario												
				Online		Request	Order	Inventory QC	Monthly	Weekly	Daily	Kitchen	Outlet	Social Media	E-commerce	Employees
Primary Activities	Inbound Logistics	4	Supplier Review	Supplier Research		1	2									
			Procuring Ingredient			Purchase Request	Purchase Order	3								
			Quality Control					Inventory Quality Control	4							
	Operations	4	Production Planning						Production Planning			5				
			Daily Production									Daily Production	6			
			Outlet Preparation and Operation							7			Outlet Preparation and Operation			
			Stock Opname							Stock Opname	8					
	Outbound Logistics	3	Ingredient Transportation			Requested Product Transportation					Product Transportation					
			Bulk Delivery	Bulk Delivery												
	Marketing and Sales	6	Point-of-Sale										Point of Sale		E-commerce Order	
			Product Display and Marketing											Digital Marketing		
			Sales/Discount Offers										Sales/Discount Offers Promotion			
			Brand Collaboration										Brand Collaboration			
			Event Management			Event Management										
	Service	3	Inquiry Handling										Inquiry Handling			
			Feedback Handling										Feedback Handling			
			Monthly Feedback Analysis			9			Monthly Feedback Analysis							
	Finance	1	Payment Request Handling			Payment Request Handling										
Secondary Activities	Firm Infrastructure	1	Equipment Maintenance									Equipment Maintenance				
	Finance	1	Transaction Management								Transaction Management					
	Human Resource Management	3	Employee Attendance													Employee Attendance
			Performance Report													Performance Report
			Employee Training													Employee Training
	Technology Department	1	IT Maintenance						IT Maintenance							
	Procurement	1	Procuring Equipment			Equipment Procurement										

Above is the Process Selection Matrix in which we are able to identify the business processes currently operating in Gotham Treats. The outlined red boxes are the business processes that are included in the solution of an inventory management system.

The businesses processes that will be affected by the inventory management system and lead those business processes to be more efficient and automated are as followed:

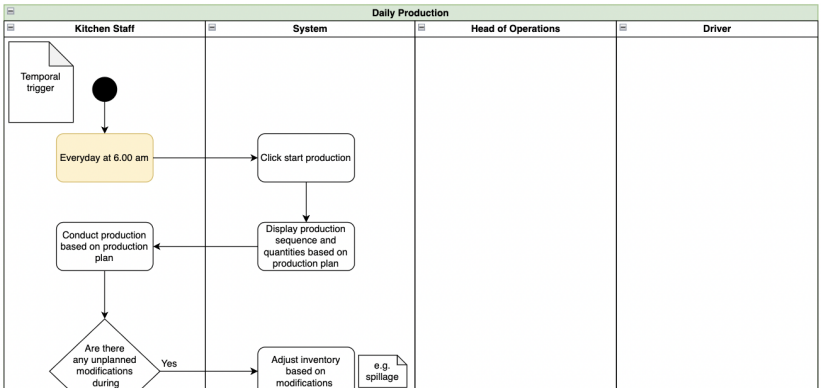
1. Purchase Request
2. Purchase Order
3. Inventory Quality Control
4. Production Planning
5. Daily Production
6. Outlet Preparation and Operation
7. Stock Opname
8. Product Transportation

4.6. End to End and Activity Diagram



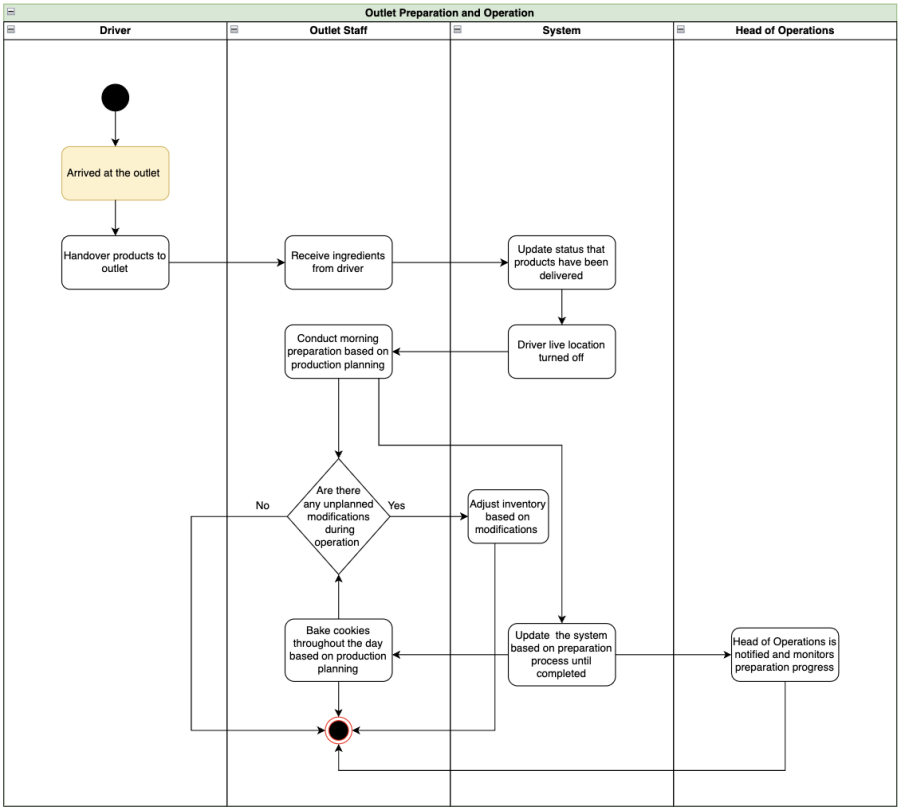
Our end-to-end business process begins with Production Planning, which is conducted monthly. It serves as the foundation for material requirements planning in procurement and day-to-day production operations. The process then progresses through various stages, ending in Stock Opname, which is carried out weekly. Stock Opname ensures that our recorded inventory aligns with the actual on-hand inventory.

1. Daily Production



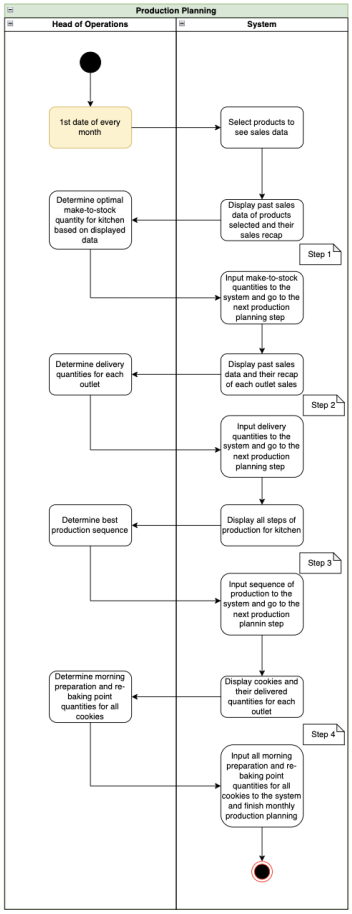
The activity diagram for the daily production process of Gotham Treats starts with a temporal trigger at 6 a.m. daily, initiating the production activities. It involves starting production, adjusting inventory if there are unplanned modifications, continuing the production process while ensuring quality assurance, finishing production, and notifying the driver that the materials can be picked up.

2. Outlet Preparation and Operation



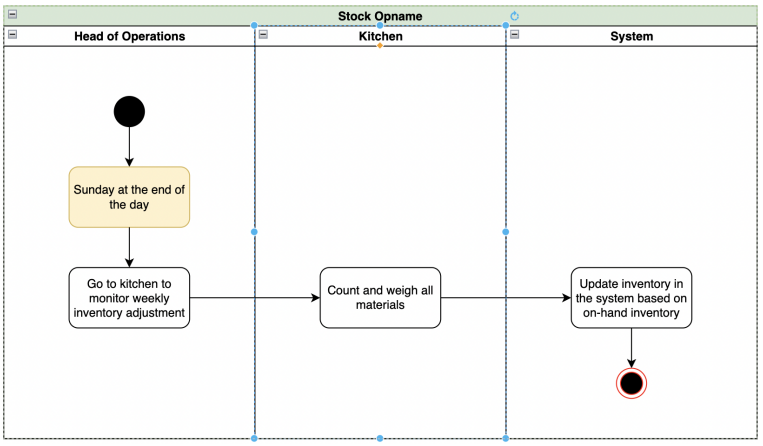
The activity diagram for outlet preparation and operation begins with the trigger of the driver arriving at the outlet and handing over the products to the outlet staff. The staff then updates the status in the system, confirming that the products have arrived. Next, based on the production planning, the outlet staff conducts morning preparation. After completing the preparation process, the staff updates the system to reflect the progress. If there are any unplanned modifications or changes in inventory, the staff adjusts it accordingly. Finally, the staff bakes the cookies according to the production planning.

3. Production Planning



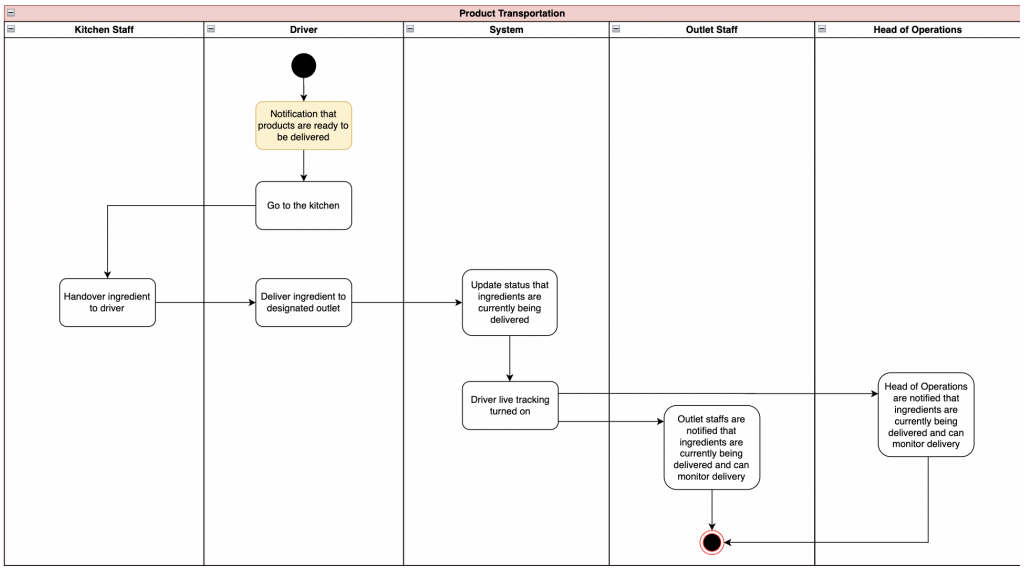
The activity diagram for the production process at Gotham Treats starts on the 1st date of every month and includes four key steps: determining the Make-to-Stock (MTS) quantity, establishing delivery quantities, defining the production sequence, and determining morning preparation and rebaking point quantities. The diagram concludes by inputting all the production data into the system for accurate record-keeping and future planning. This visual representation provides a clear overview of the sequential activities involved in the production process, ensuring efficient and well-coordinated operations at Gotham Treats.

4. Stock Opname



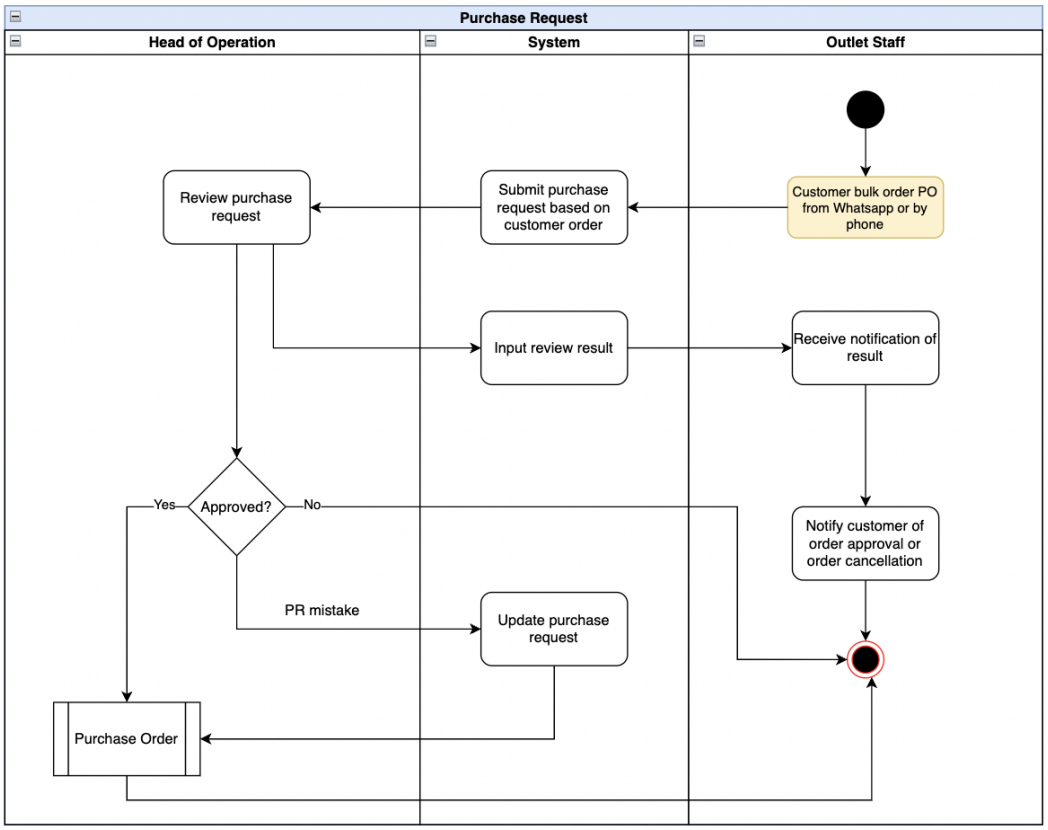
The activity diagram for Stock Opname begins with a temporal trigger on Sunday at the end of the day. The process involves the kitchen staff counting and weighing the materials in stock. They then update the inventory records based on the actual on-hand inventory. This activity allows the staff to reconcile the physical stock with the recorded stock levels, ensuring accurate and up-to-date inventory information.

5.Product Transportation



The activity diagram for Product Transportation begins with a trigger notification indicating that the products are ready to be delivered. Upon receiving the notification, the driver proceeds to the kitchen to collect the products. The driver then transports the products to the designated outlet. As the transportation process begins, the system updates the status to reflect that the ingredients are currently being delivered. The live tracking feature is activated, allowing real-time tracking of the delivery progress. Simultaneously, the Head of Operations (HoO) and the outlet staff are notified and granted access to the live tracking information. This enables the Head of Operations and Outlet Staff to monitor the delivery status.

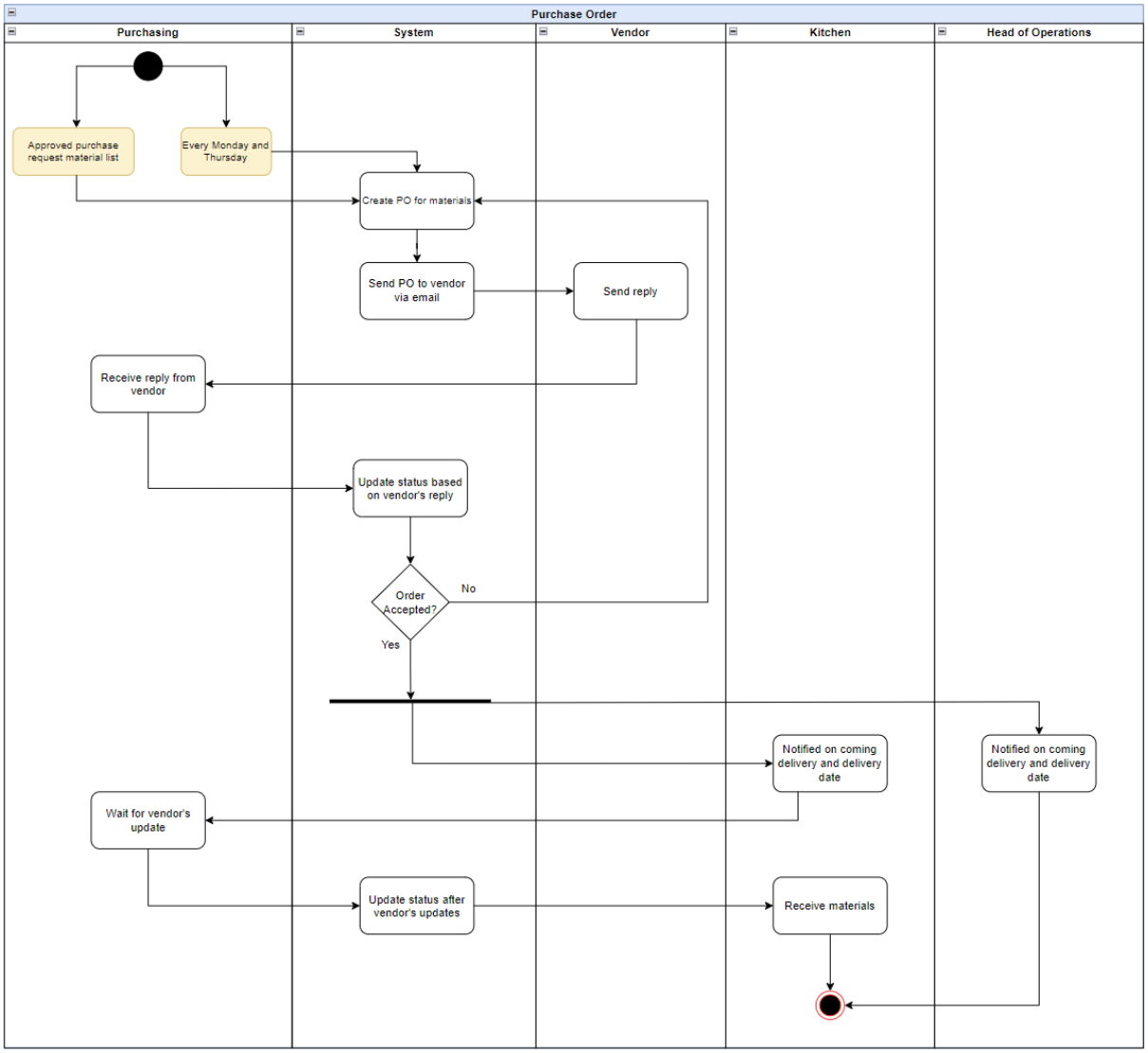
6. Purchase Request



The activity diagram for Purchase Request starts with a trigger from the outlet staff when they receive a customer's bulk order via WhatsApp or phone. The staff submits the purchase request to the system, then the Head of Operations (HoO) reviews the purchase request and either approves or rejects it. HoO inputs the result (approval or rejection) into the system. Upon receiving the notification of the result, the outlet staff informs the customer regarding the approval or cancellation of the requested items. If the purchase request is

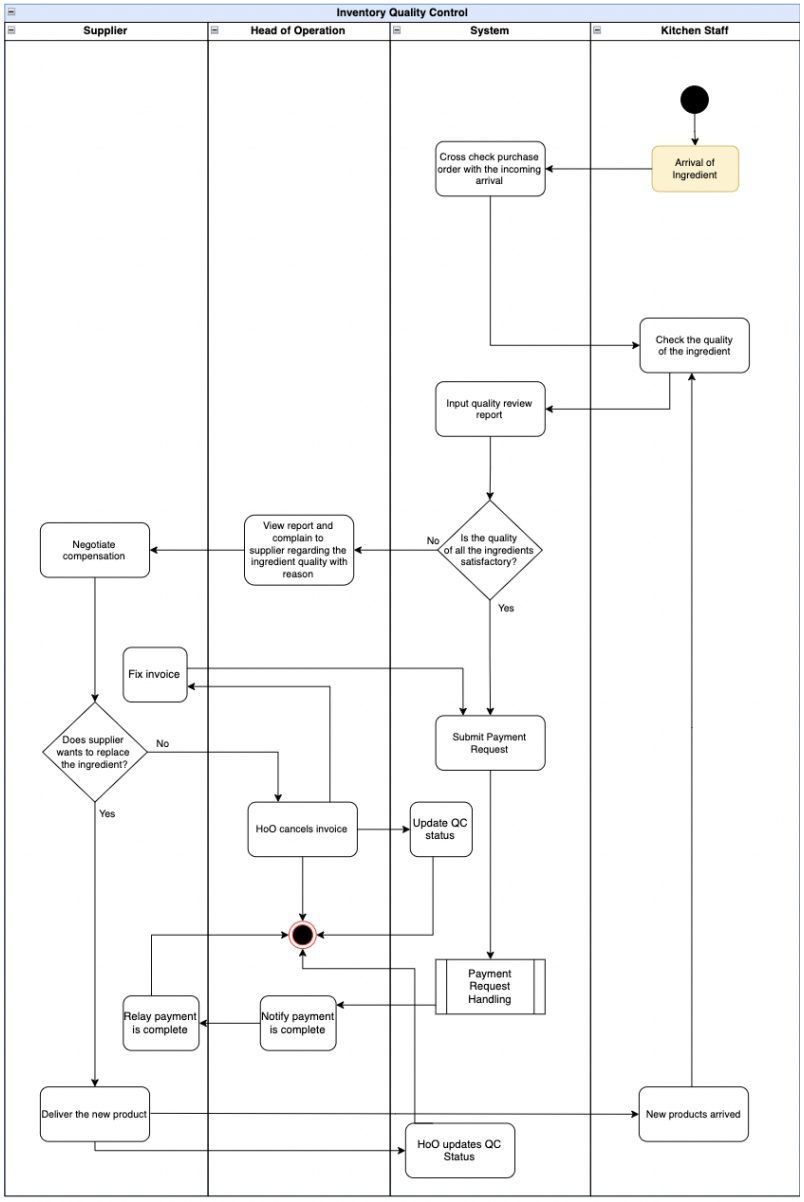
approved, the staff proceeds to create a purchase order, which serves as an official document for procuring the requested items.

7. Purchase Order



The activity diagram for Purchase Order begins with two triggers: either an approved Purchase Request list or a routine schedule of every Monday and Thursday. When triggered, the purchasing department creates a Purchase Order (PO) for the required materials, which is then sent to the vendor via email. The vendor receives the PO and replies to confirm or provide any updates regarding the order. Based on the vendor's reply, the system updates the status accordingly. If the order is accepted, the system notifies that the delivery is on its way to the kitchen and informs the Head of Operations (HoO). The system then waits for further updates from the vendor regarding the delivery. Finally, the kitchen receives the materials upon their arrival.

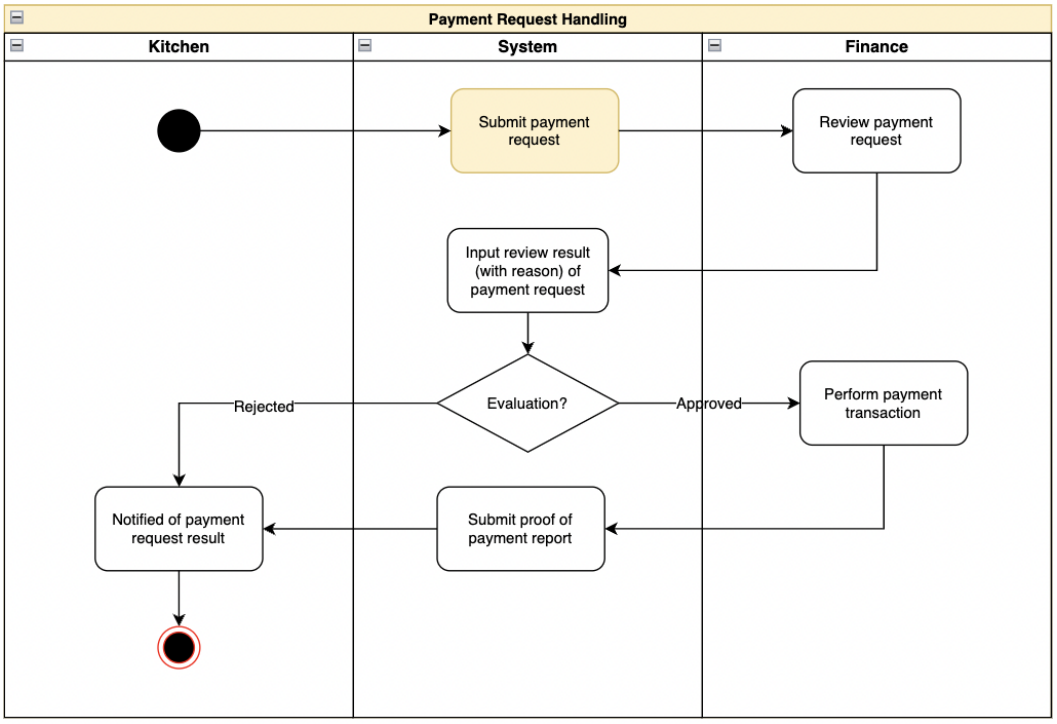
8. Inventory Quality Control



The activity diagram for Inventory Quality Control begins with the trigger of ingredient arrival to the kitchen staff. The staff first cross-checks the Purchase Order with the incoming arrival to ensure accuracy. They then proceed to check the quality of the materials and input the quality review report. If the quality satisfies the standards, the staff proceeds to submit a payment request and notifies the Head of Operations (HoO). Once the payment is completed, the staff relays the information. However, if the quality inspection does not pass, the staff initiates a complaint to the supplier and negotiates compensation. If the supplier

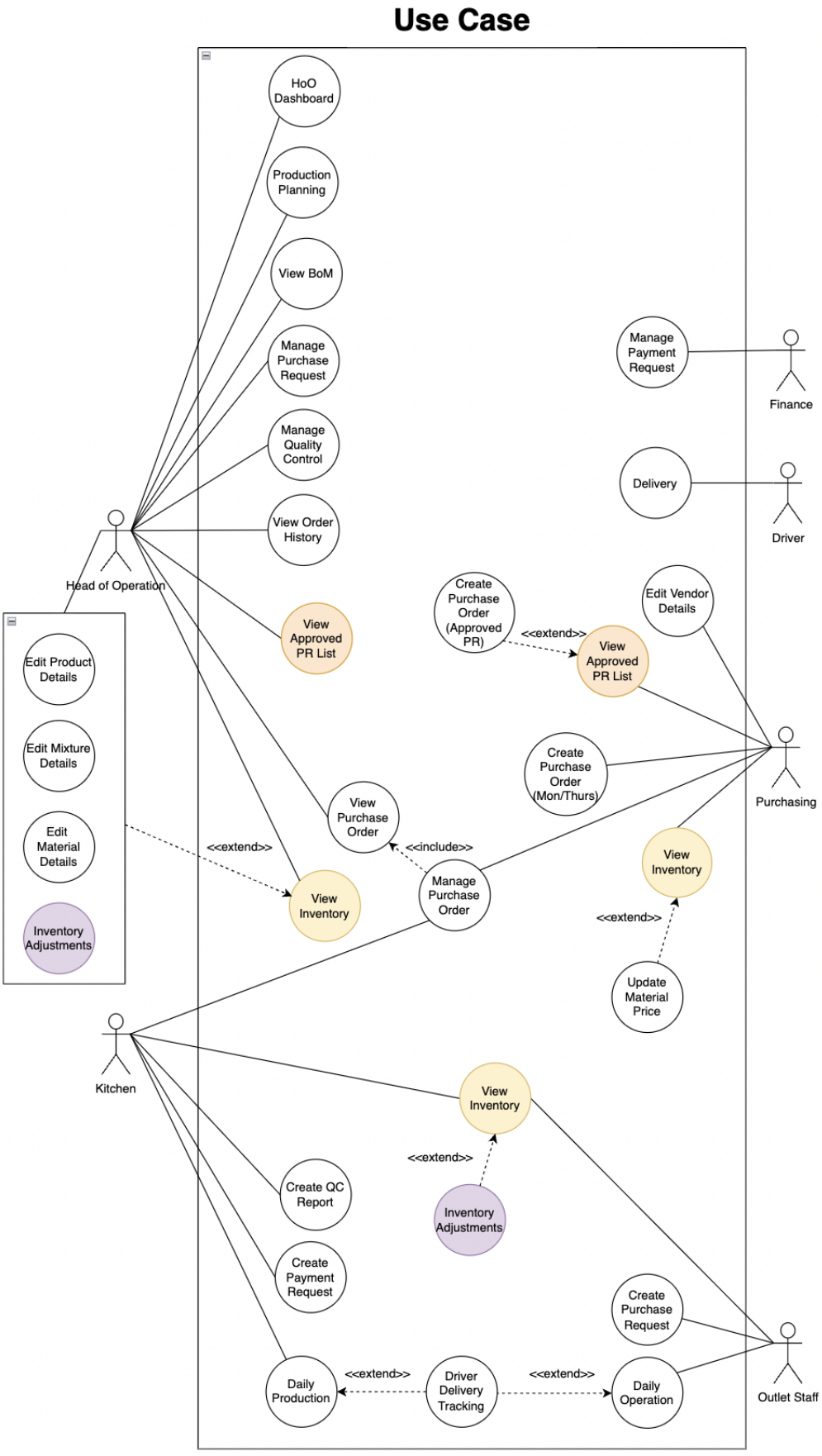
agrees to redeliver the ingredients, a new product is delivered and undergoes another quality check. If the supplier refuses to redeliver, the staff cancels the invoice.

9. Payment Request Handling

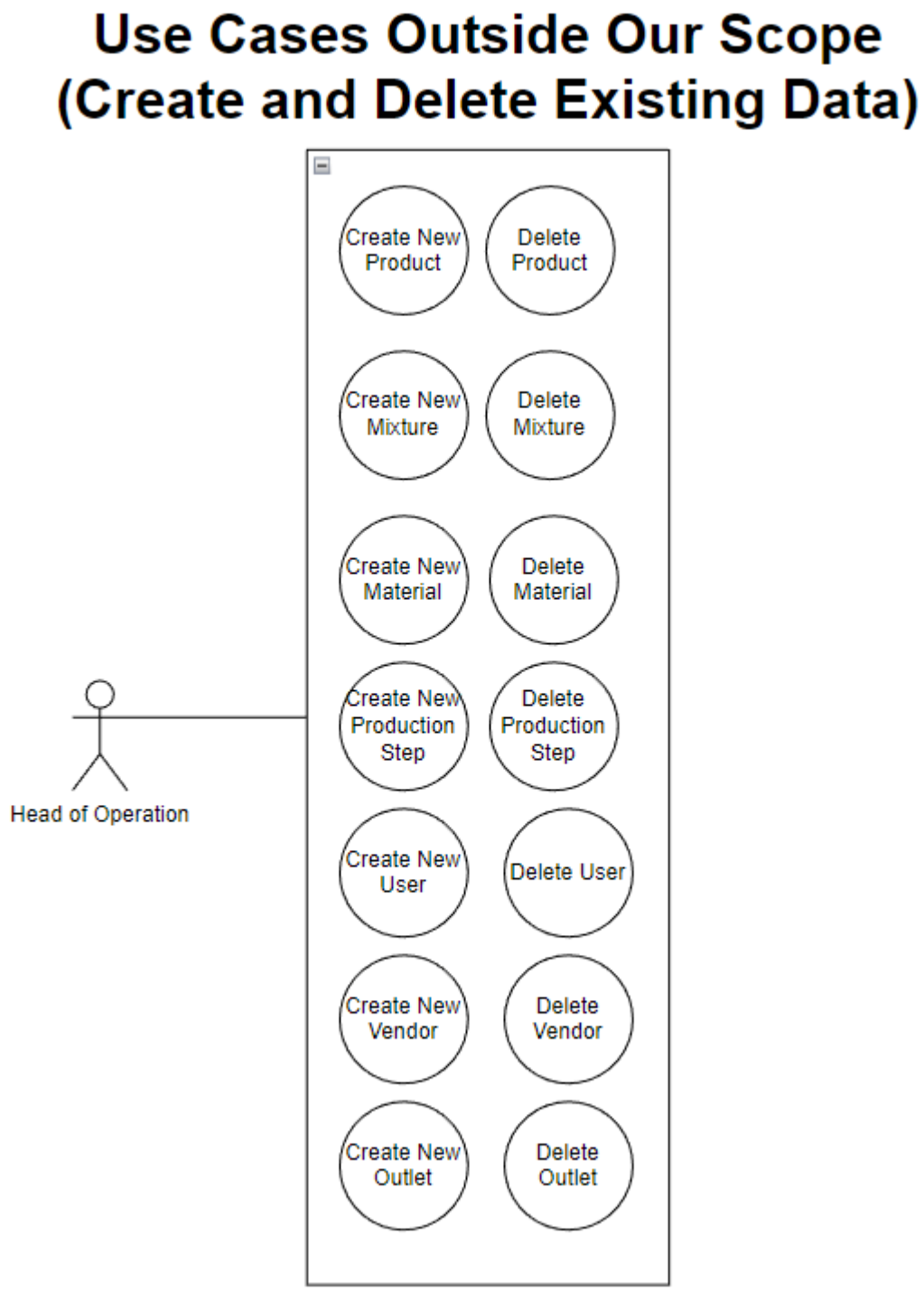


The activity diagram for Payment Request Handling begins with the trigger of the kitchen submitting a payment request in the system. The finance department then reviews the payment request and inputs the review result in the system. If the request is approved, the finance department proceeds to perform the payment transaction and submits the proof of payment. Finally, regardless of whether the request is rejected or approved, the kitchen is notified of the payment request result.

4.7. Use Case Diagram



We have tailored use cases for six different actors. The Head of Operations has access to 13 use cases, Kitchen has eight, Purchasing has eight, Outlet has five, Finance has one, and Driver has one. These use cases are designed to streamline operations and support specific tasks and responsibilities within each role.



We also created a list of use cases outside of our scope that acts as creating and deleting existing master data.

4.8. RACI/RASCI Model

The model below is a RACI Matrix model created for Gotham Treats business processes that we have designed. There are 40 use cases in our scope, and we have built the matrix model for all six actors that are involved with these use cases. The six actors are the Head of Operations (HoO), kitchen staff, finance staff, purchasing staff, outlet staff and driver.

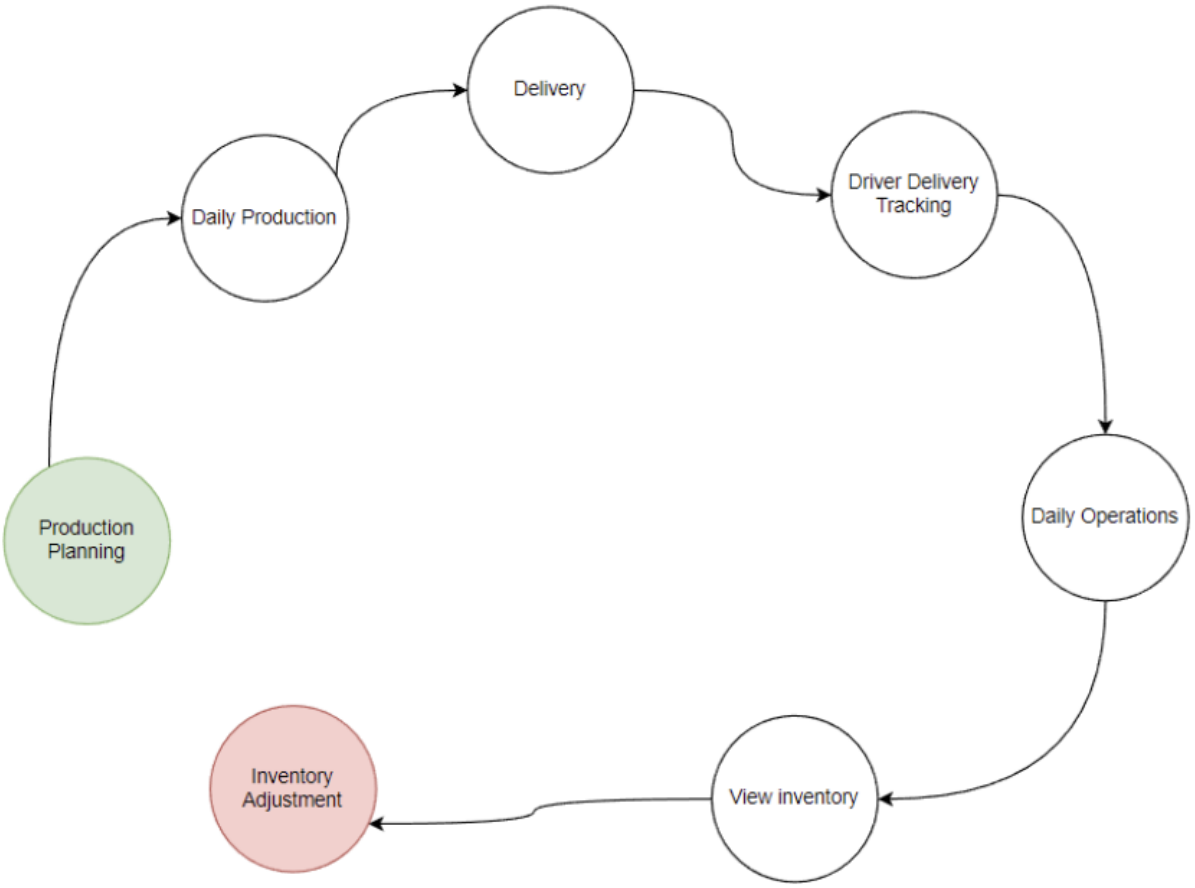
Link to Matrix Model:
https://docs.google.com/spreadsheets/d/12TbPG9JOHz3p6FYalHlAtd8PnHKT1P7MaF9eVnhvGs4/edit?usp=sharing

	Actors					
	Head of Operations	Kitchen	Finance	Driver	Purchasing	Outlet Staff
Use Cases						
Inventory						
View Inventory	R	R			R	R
Inventory adjustments	I	RA			C	RA
Update Material Price	I				RA	
Purchase Request						
Create Purchase Request	CI				CI	R
View approved PR list	R				R	
Manage Purchase Request	R/A				I	
Product						
Create New Product	R					
Delete Product	R					
Edit Product Details	R					

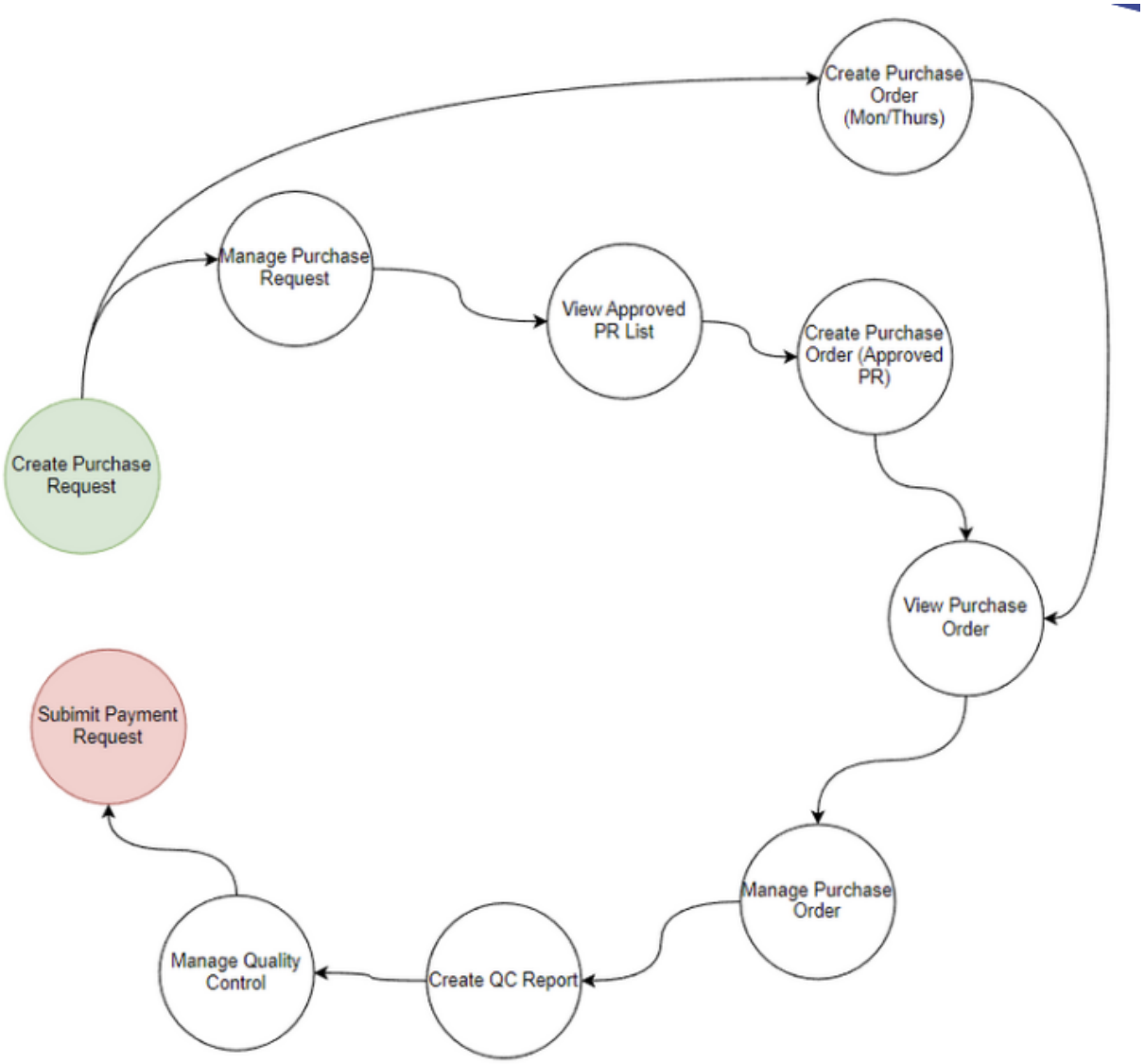
Mixture						
Create New Mixture	R					
Delete Mixture	R					
Edit Mixture Details	R					
Materials						
Create New Material	R					
Delete Material	R					
Edit Material Details	R					
Production						
Create New Production Step	R					
Delete Production Step	R					
Production Planning	R	I				I
User						
Create New User	R					
Delete User	R					
Vendor						
Create New Vendor	I				R	
Delete Vendor	R					
Edit Vendor Details	I				R	
Outlet						
Create New Outlet	R					
Delete Outlet	R					
Quality Control						
Create QC Report	I	R				

Manage Quality Control	R					
Payment Request						
Manage Payment Request			R			
Submit Payment Request		R	A			
Delivery						
Delivery				R		RA
Driver Delivery Tracking		I		RA		I
HoO Dashboard						
HoO Dashboard	RA					
Purchase Orders						
Create purchase order (Approved PR)	RA				RA	
Create purchase order (Mon & Thurs)	I				RA	
Manage Purchase Order		I			R	
View Purchase Order					R	
Daily						
Daily Operation	I					R
Daily Production	I	R		I		
Bill of Material						
View BOM	R					
Order						
View Order History	R					

4.9. Process Cycle



This is the process cycle of production and operation, commencing with Production Planning conducted on a monthly basis. It contains all relevant use cases in daily production and operation, concluding with Inventory Adjustment performed weekly as part of the Stock Opname business process.



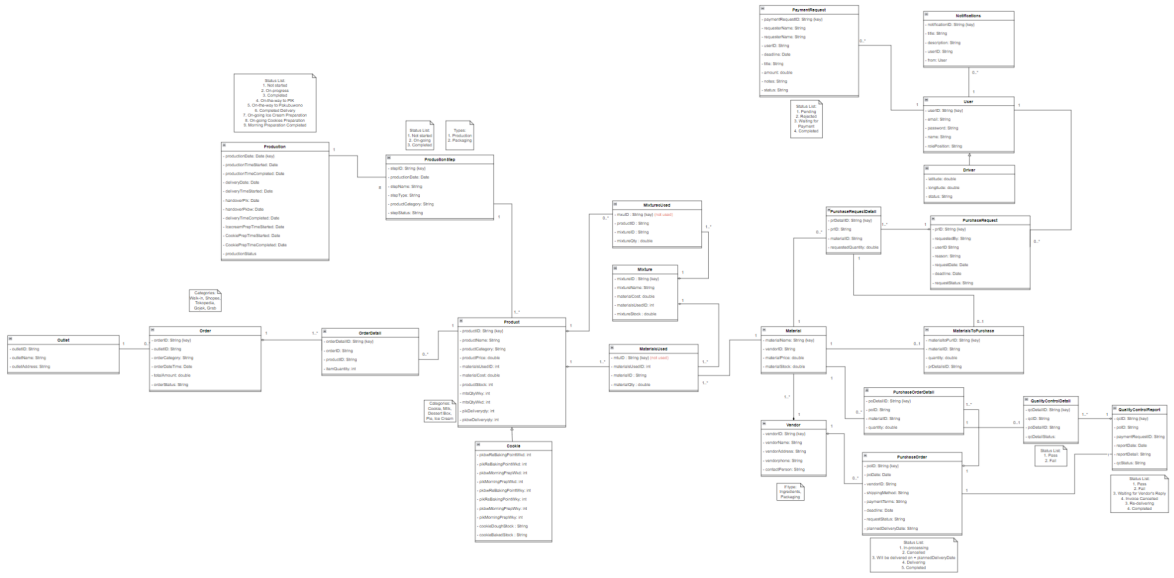
This is the process cycle of procurement, commencing with Create Purchase Request which is conducted when a customer orders in a bulk quantity. It contains all relevant use cases in the procurement process, concluding with Manage Payment Request after payment has been completed by Finance.

Chapter V: Solution Modelling

Link to all Models:

https://drive.google.com/file/d/1zlChAagIAX_BK-gJkKJS0S93mJDhm_Tp/view?usp=sharing

5.1. Domain Model Class Diagram



At the core of the domain model are the production-related classes, such as Production, ProductionStep, Product, Cookie, MaterialsUsed, and MixturesUsed. These classes capture the processes involved in producing the ice cream bake shop's products, including the steps, materials, and mixtures used. The ordering and customer management aspects are represented by classes like OrderDetail, Order, and Outlet. These classes handle the details of customer orders, such as the items ordered, quantities, and outlet locations.

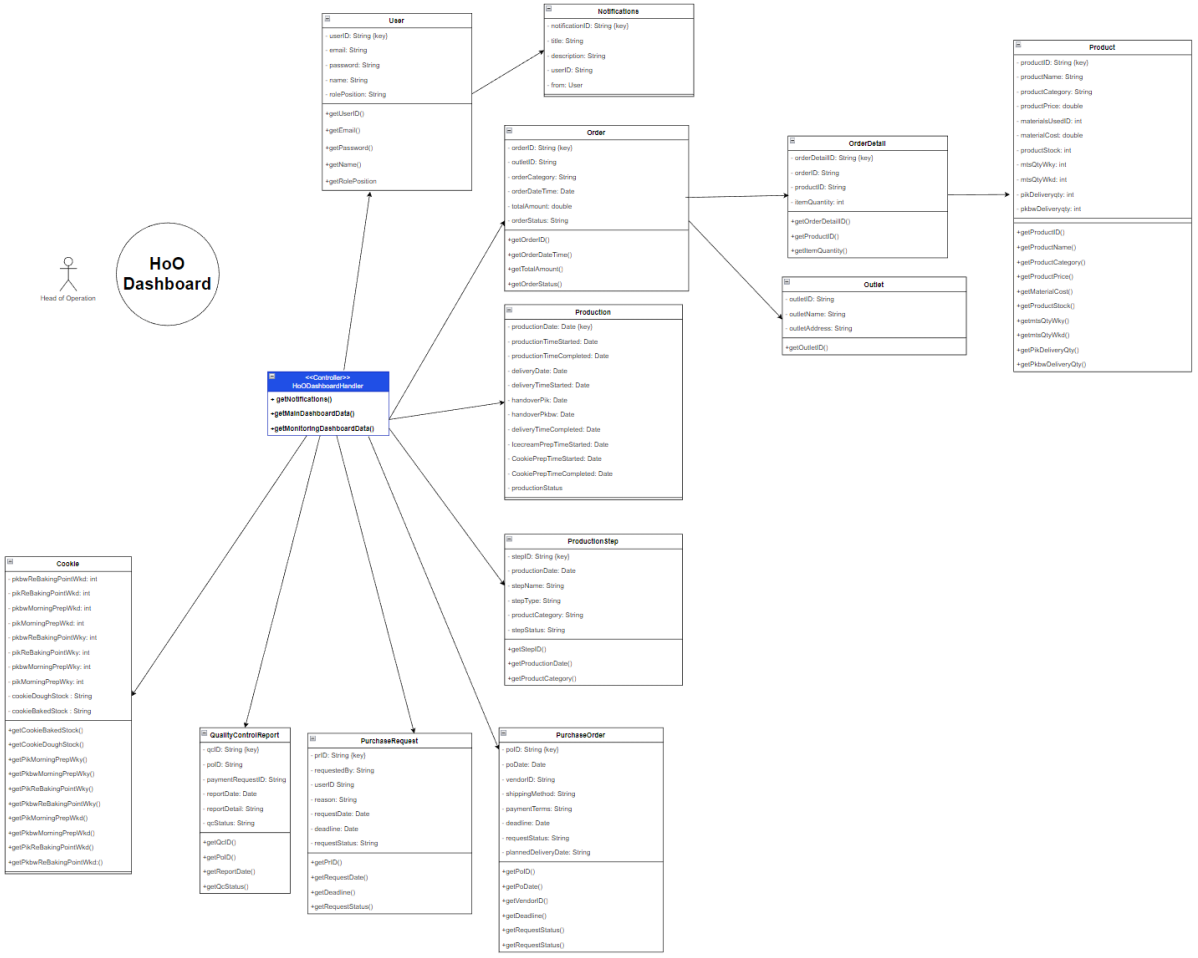
The management of materials, suppliers, and purchase-related processes is captured by classes such as Material, Mixture, MaterialsToPurchase, PurchaseOrder, PurchaseOrderDetail, Vendor, PurchaseRequest, and PurchaseRequestDetail. These classes handle the procurement and tracking of materials, generation of purchase orders, and interaction with suppliers. The quality control aspect is represented by classes like

QualityControlDetail and QualityControlReport. These classes handle the inspection and reporting of quality control activities during the production process.

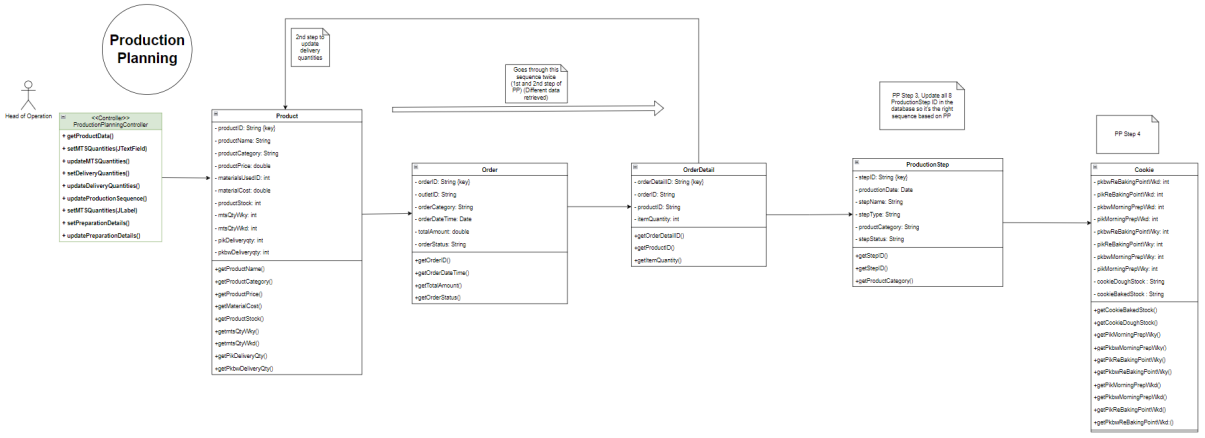
User management and authentication are handled by the User and Driver classes. User stores general user information, while Driver extends the User class to represent drivers associated with the ice cream bake shop.

Notifications and PaymentRequest classes handle communication and financial transactions within the system. Notifications keep users informed about important events, while PaymentRequest stores information about payment requests, including status, amount, and deadlines.

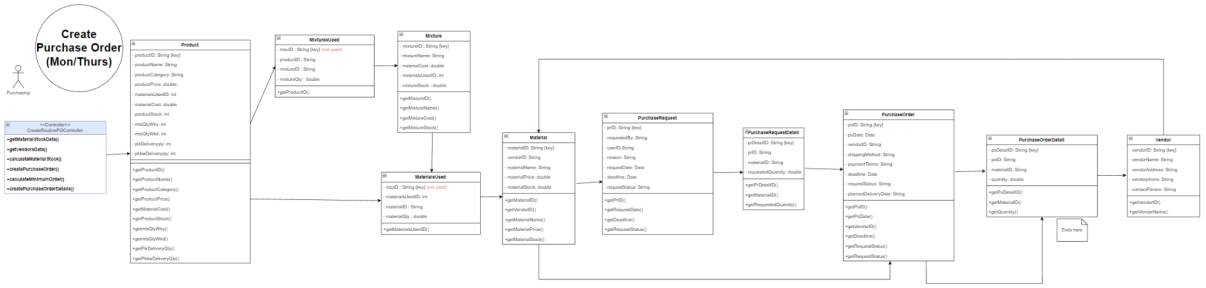
5.2. First Cut Class Diagram



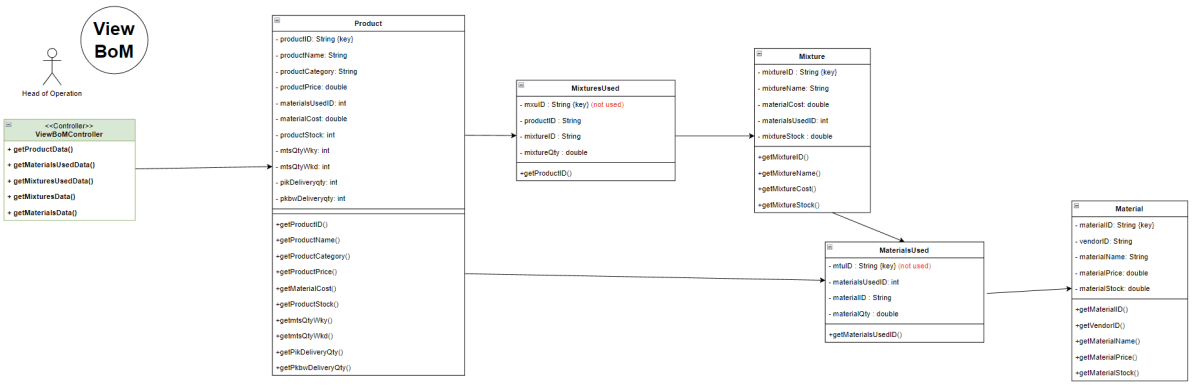
The HoO Dashboard first cut retrieves data from a lot of different database tables, which are User, Notification, Order, OrderDetail, Product, Outlet, Production, ProductionStep, PurchaseOrder, PurchaseRequest, QualityControlReport, and Cookie. This would be explained further in the sequence diagrams.



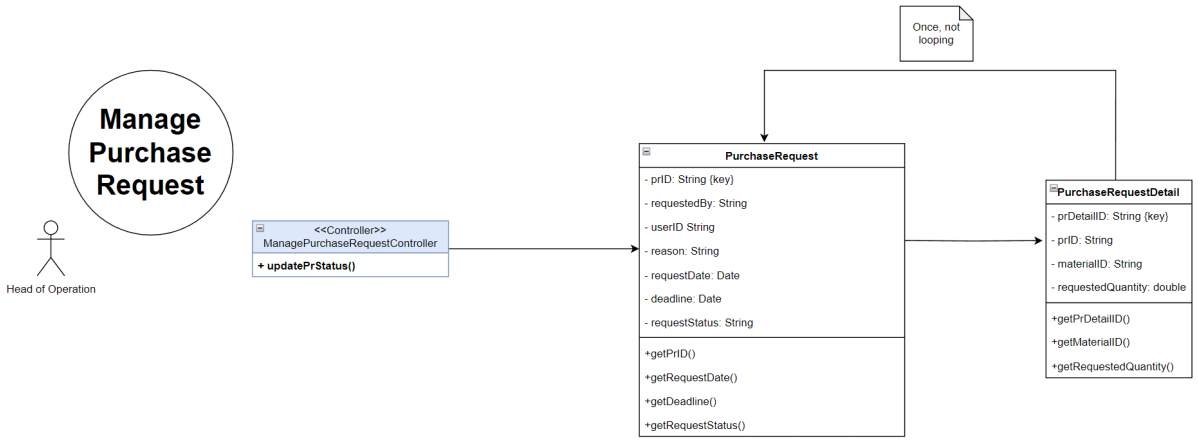
The Production Planning first cut retrieves data from Product, Orde, OrderDetail, ProductionStep, and Cookie. This first cut also updates data from Product, ProductionStep, and Cookie. This will be explained further in the sequence diagrams.



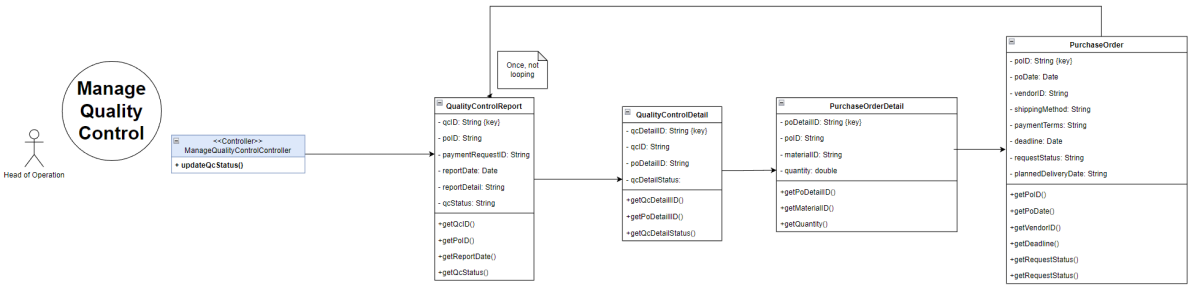
The Create Purchase Order (Mon/Thurs) first cut retrieves data from Product, MixturesUsed, MaterialsUsed, Mixture, Material, PurchaseRequest, PurchaseRequestDetail, PurchaseOrder, PurchaseOrderDetail, and Vendor. This will be explained further in the sequence diagrams



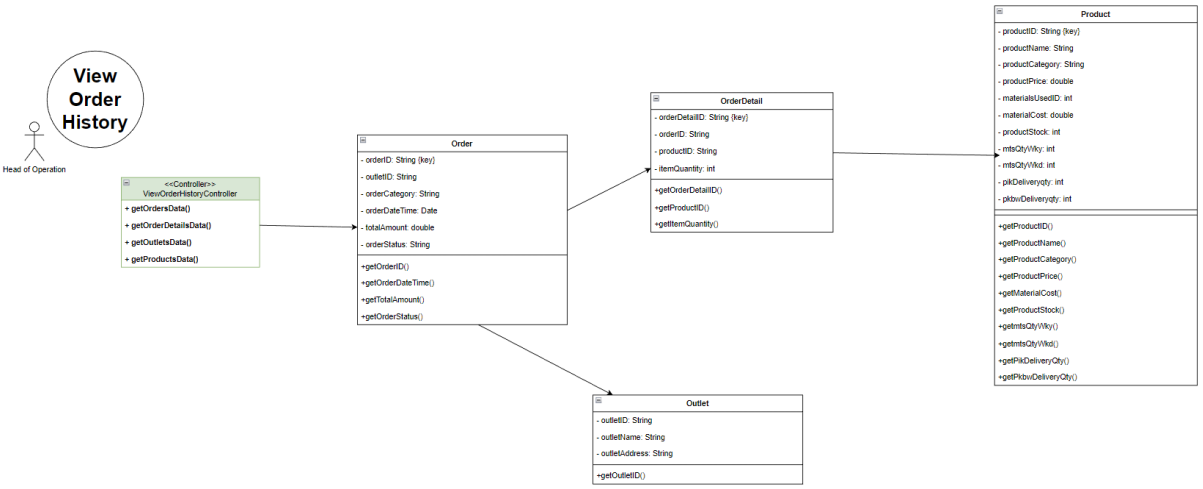
The View BoM first cut retrieves data from Product, and based on the ID, will get MaterialsUsed and MixturesUsed, which will get us to the Material Database, where materialCost will be calculated so we know how much is the material cost for our products.



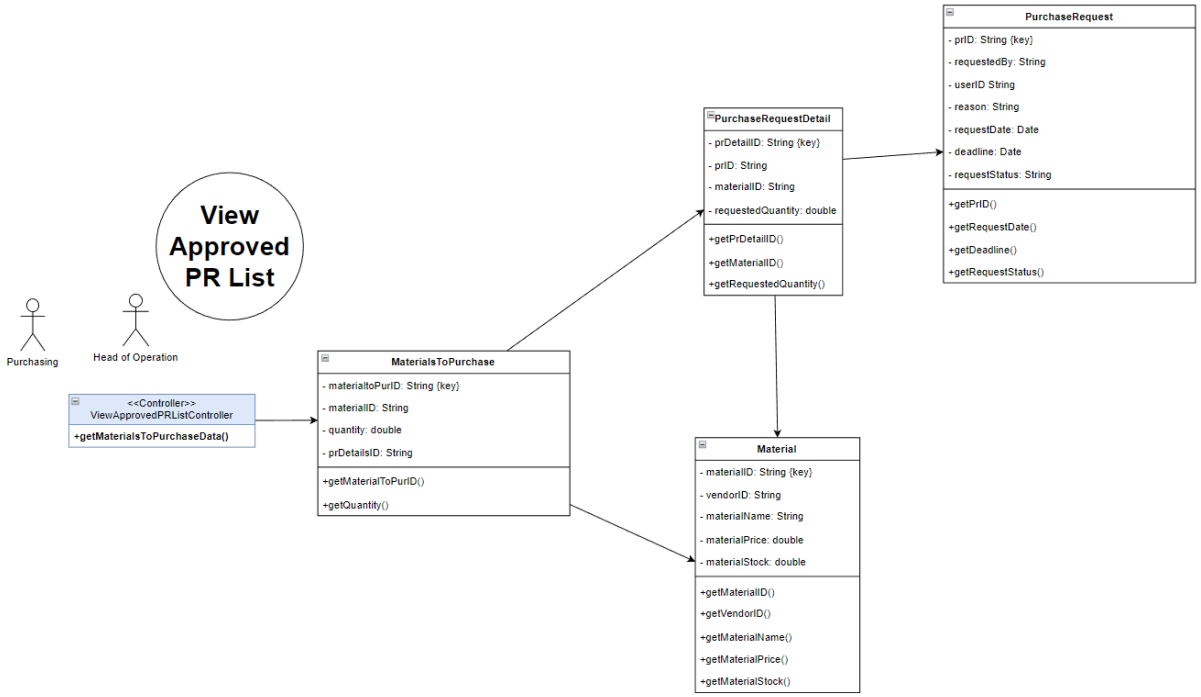
The Manage Purchase Request first cut will retrieve data from PurchaseRequest and PurchaseRequestDetail and update requestStatus attribute in PurchaseRequest.



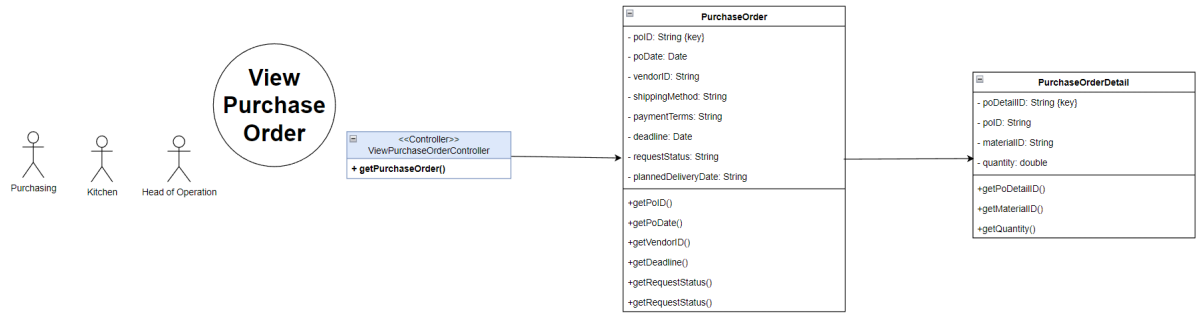
The Manage Quality Control first cut will retrieve data from QualityControlReport, QualityControlDetail, PurchaseOrderDetail, and PurchaseOrder. Then Manage Quality Control will update the status of QualityControlReport and every QualityControlDetail.



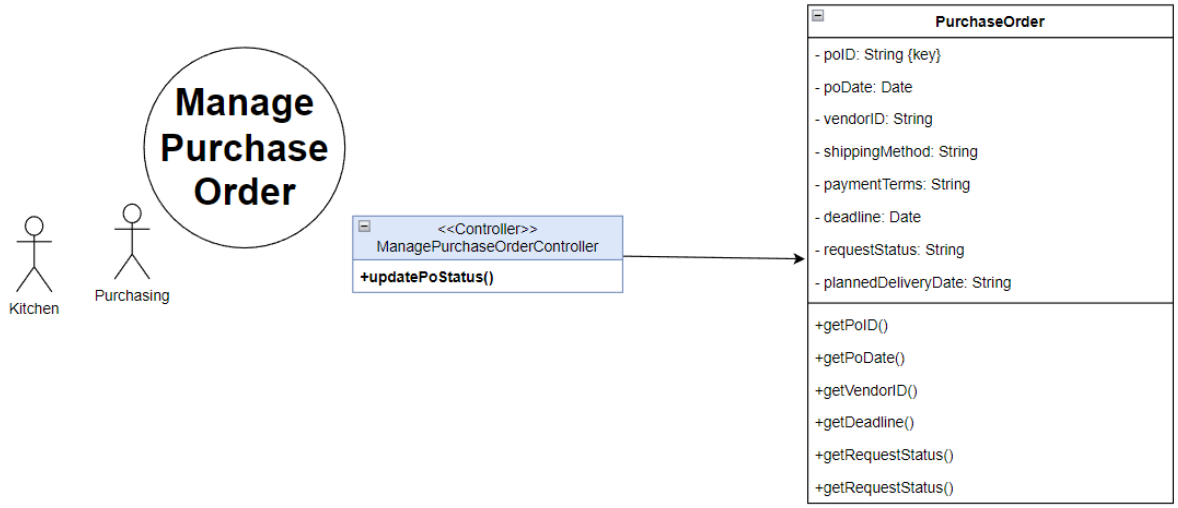
The View Order History first cut will retrieve data from Order, OrderDetail, Outlet, and Product.



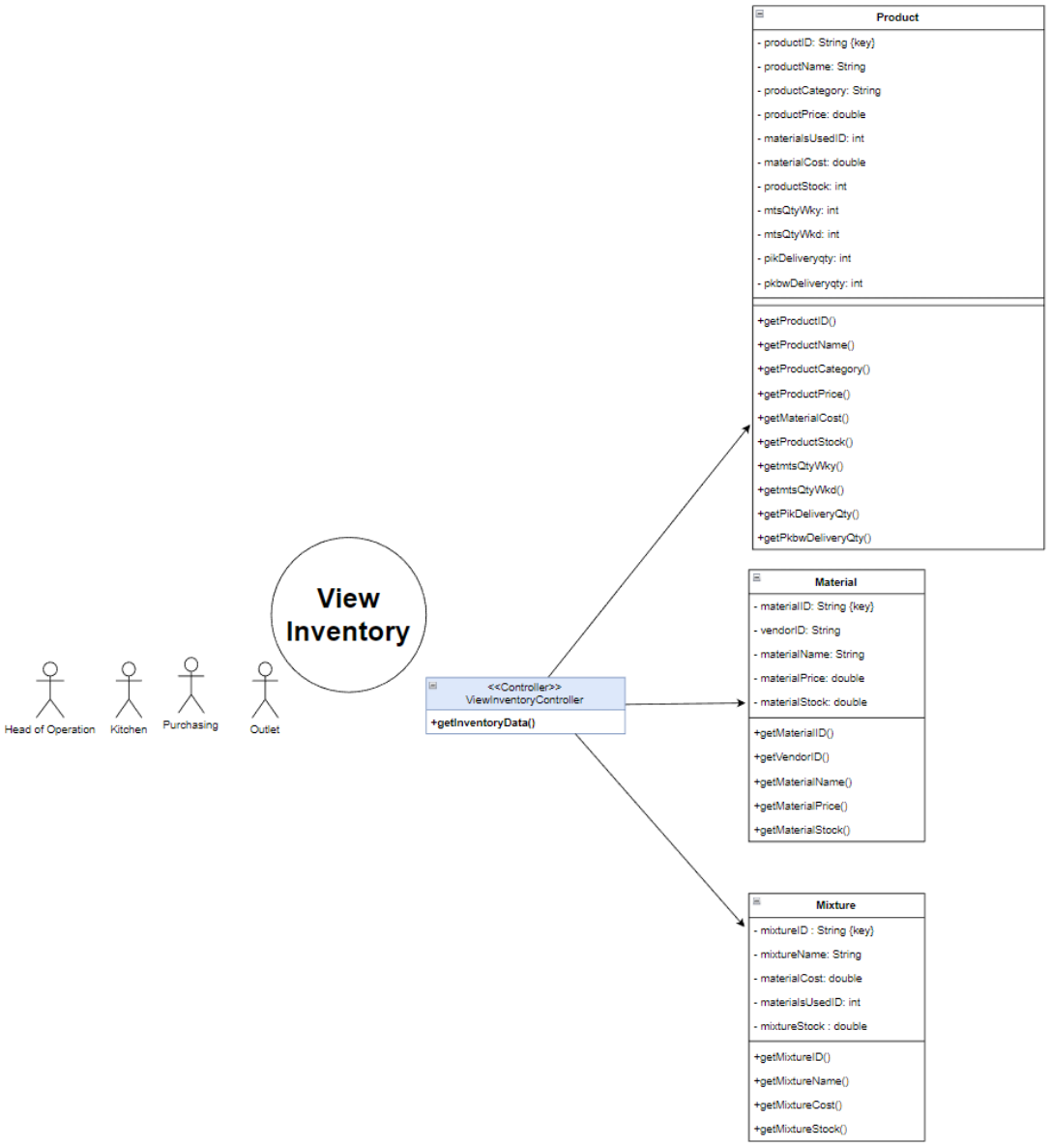
The View Approved PR List first cut will retrieve data from MaterialsToPurchase, PurchaseRequestDetail, Material, and Purchase Request.



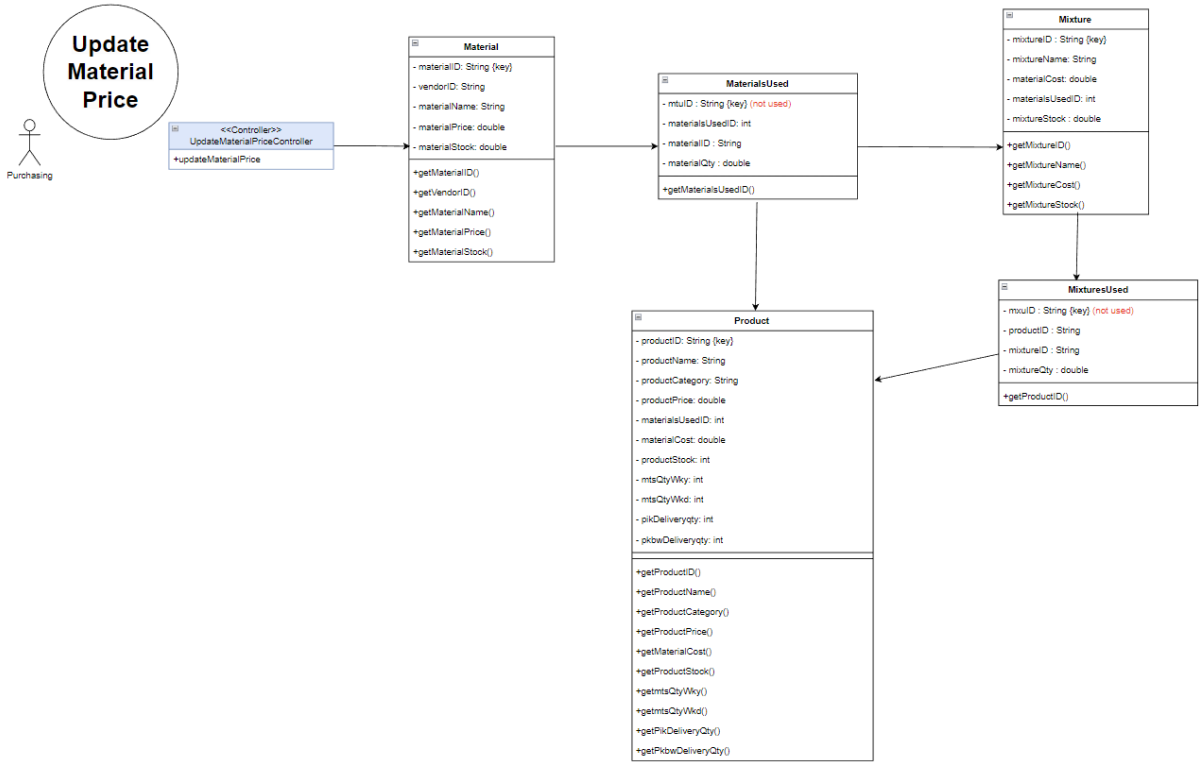
The View Purchase Order first cut will retrieve data from PurchaseOrder and PurchaseOrderDetail.



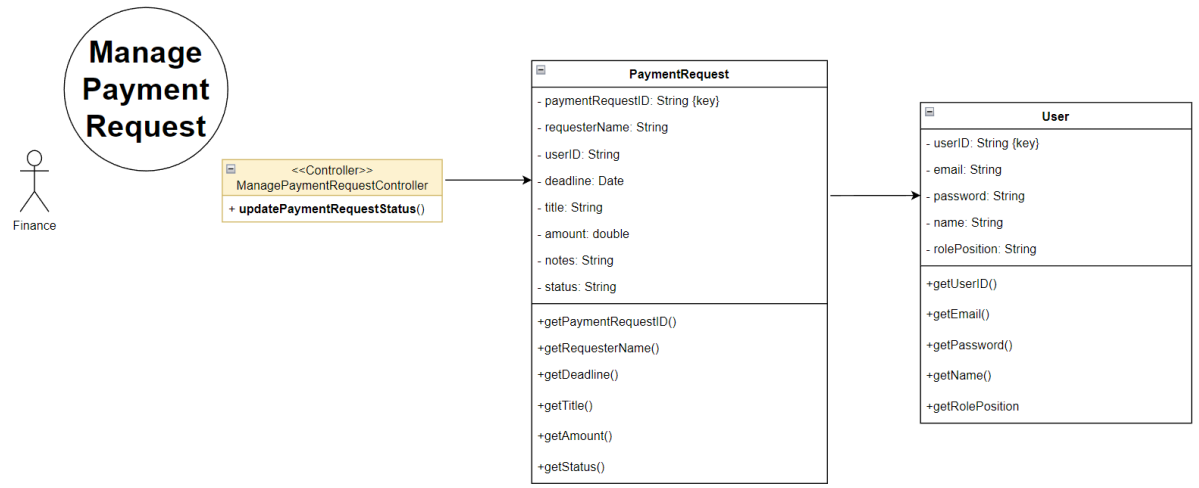
The Manage Purchase Order first cut will retrieve data from PurchaseOrder and update purchase order status.



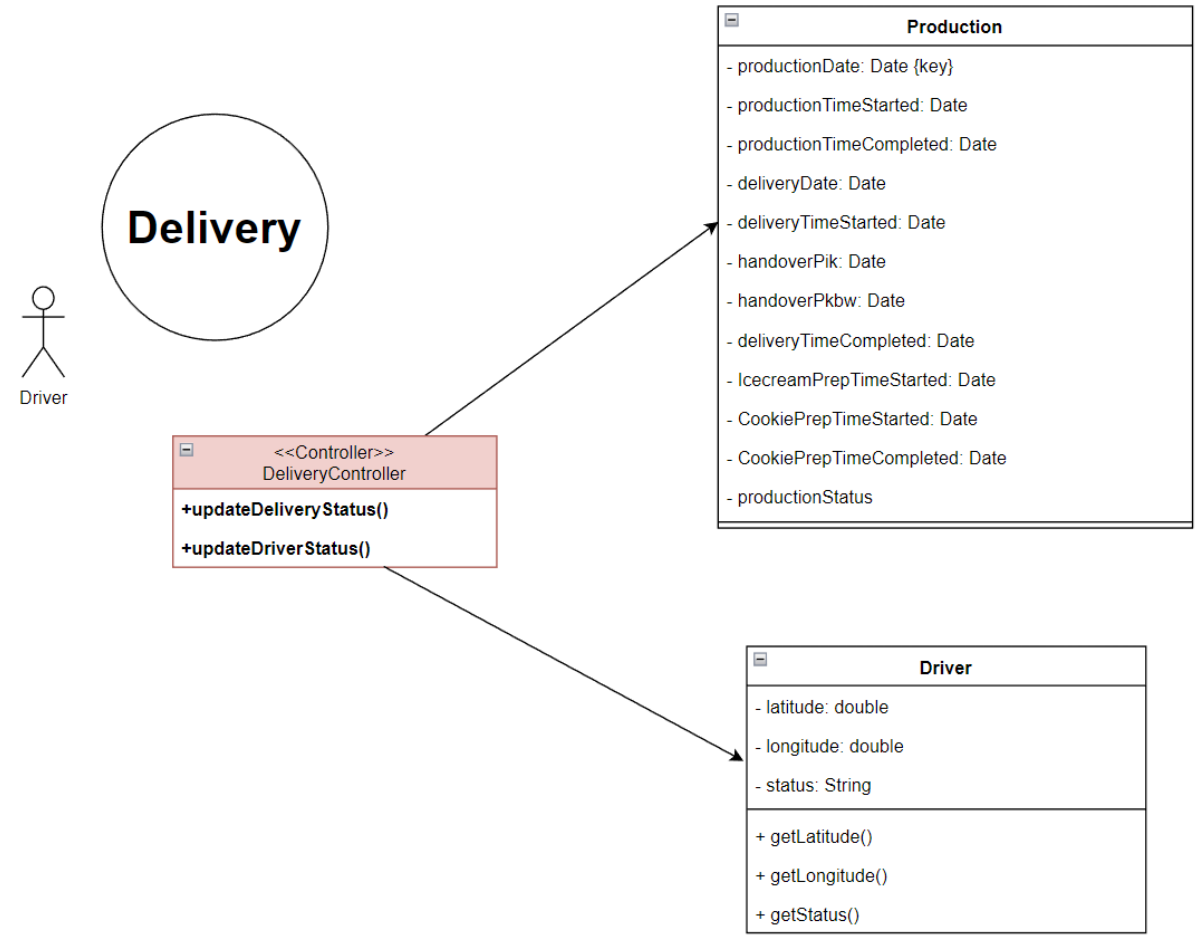
The View Inventory first cut will retrieve data from Product, Material, and Mixture.



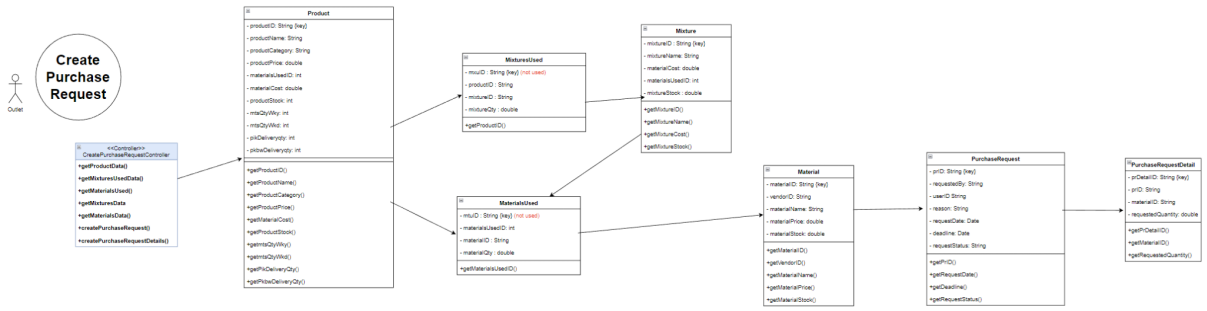
The Update Material Price first cut will retrieve data from Material and update the material price. This will trigger a domino effect of updates which will update mixture price and product price.



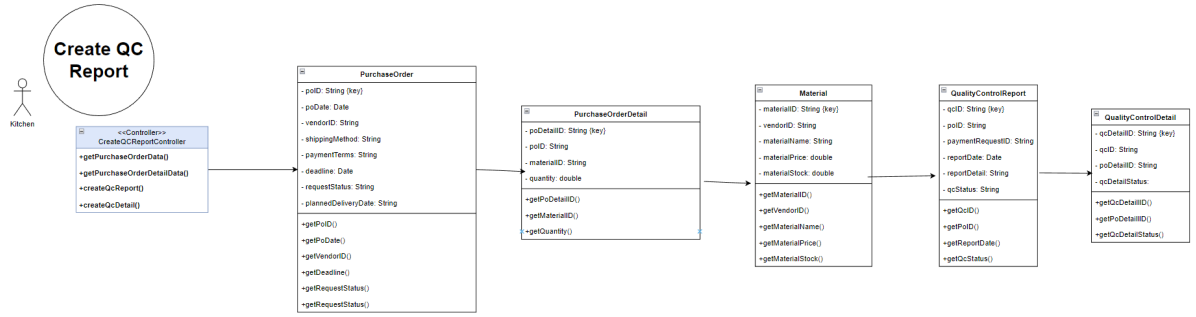
The Manage Payment Request first cut will retrieve data from PaymentRequest and User and updates payment request status.



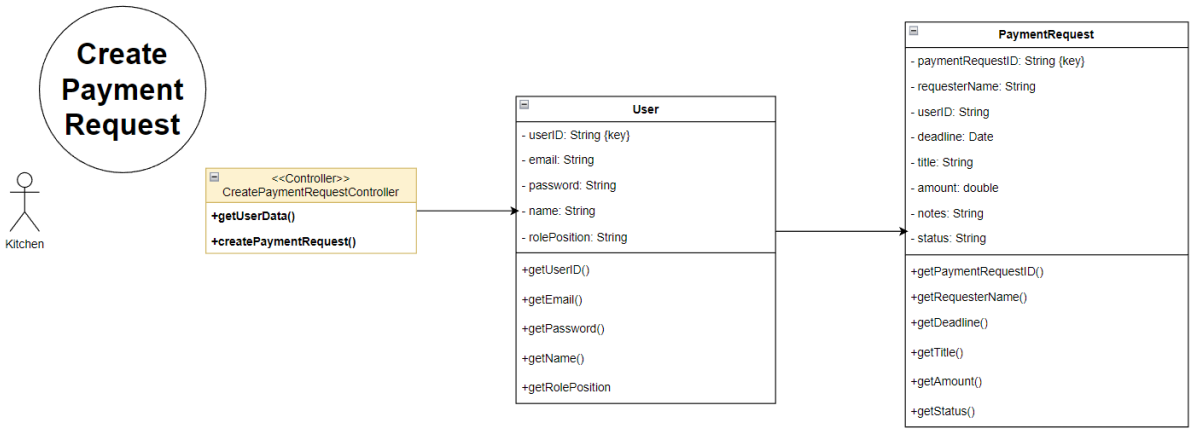
The Delivery first cut will retrieve and update data from Production and Driver.



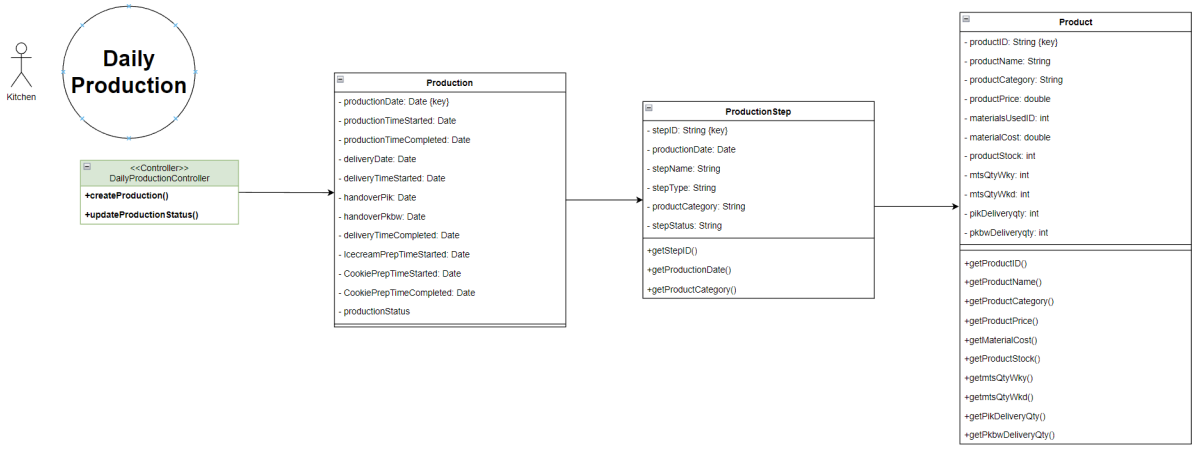
The Create Purchase Request first cut will retrieve data from Product, MaterialsUsed, MixturesUsed, Mixture, Material, and create data to PurchaseRequest and PurchaseRequestDetail.



The Create QC Report first cut will retrieve data from PurchaseOrder, PurchaseOrderDetail, Material, and will create data to QualityControlReport and QualityControlDetail.

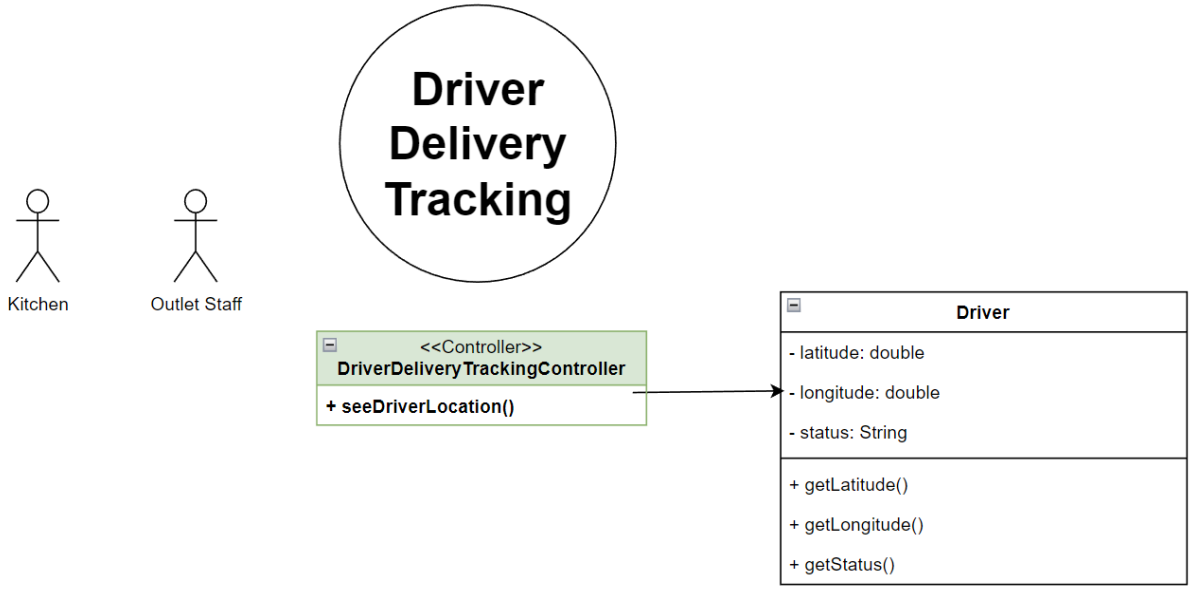


The Create Payment Request first cut will retrieve data from User and create data to PaymentRequest.

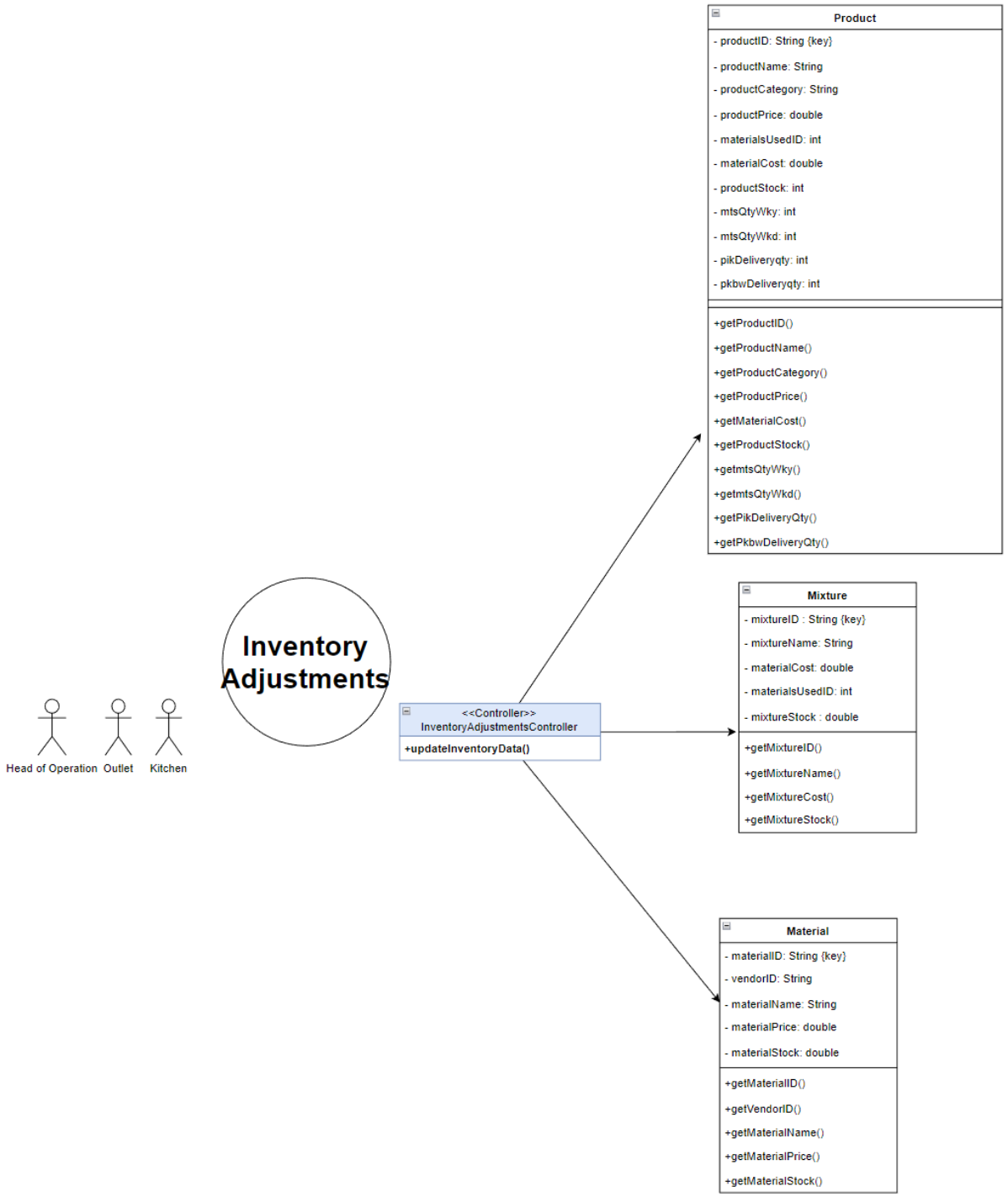


The Daily Production first cut will retrieve data from Production, ProductionStep, and Product. It will also update the data on Production and update status on ProductionStep.

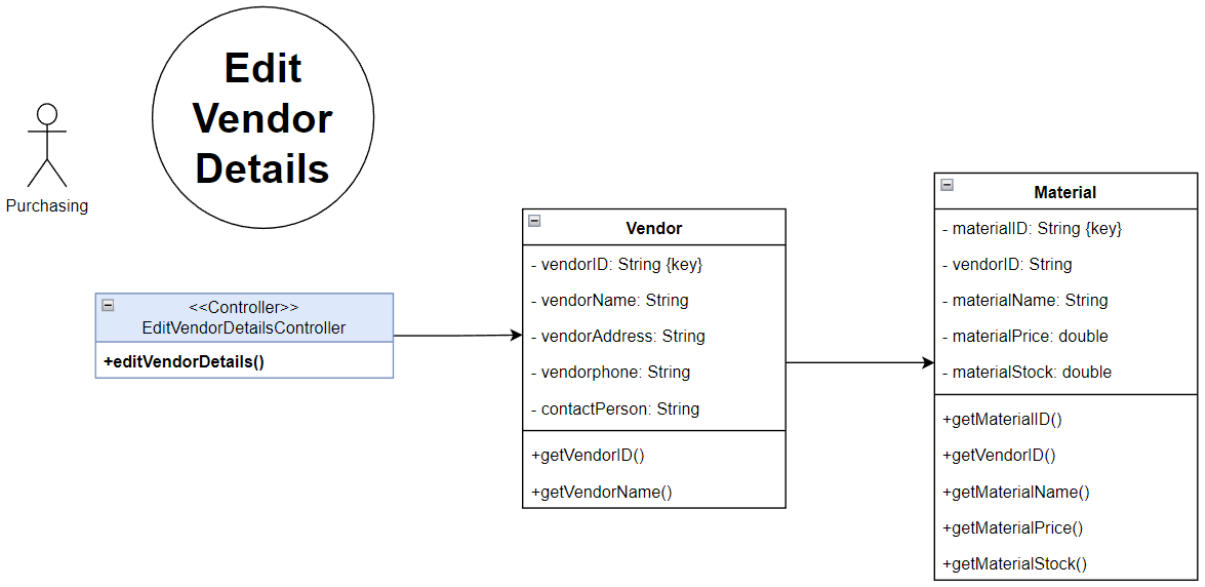
The Daily Operation first cut will retrieve and update data from Production and Cookie.



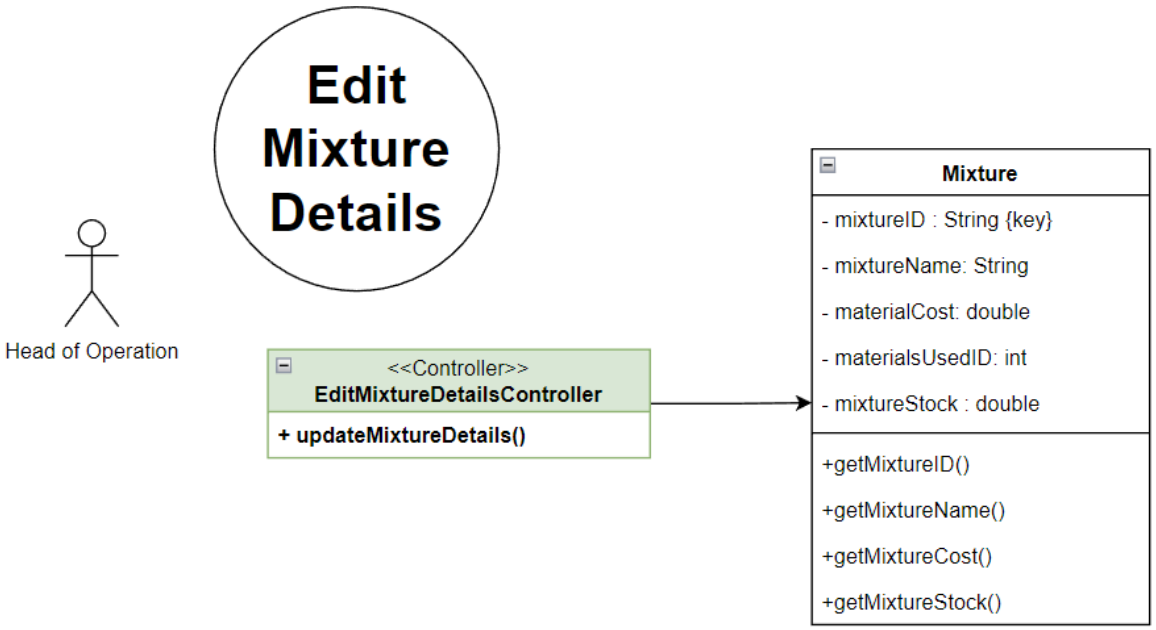
The Driver Delivery Tracking first cut will retrieve and update data from Driver.



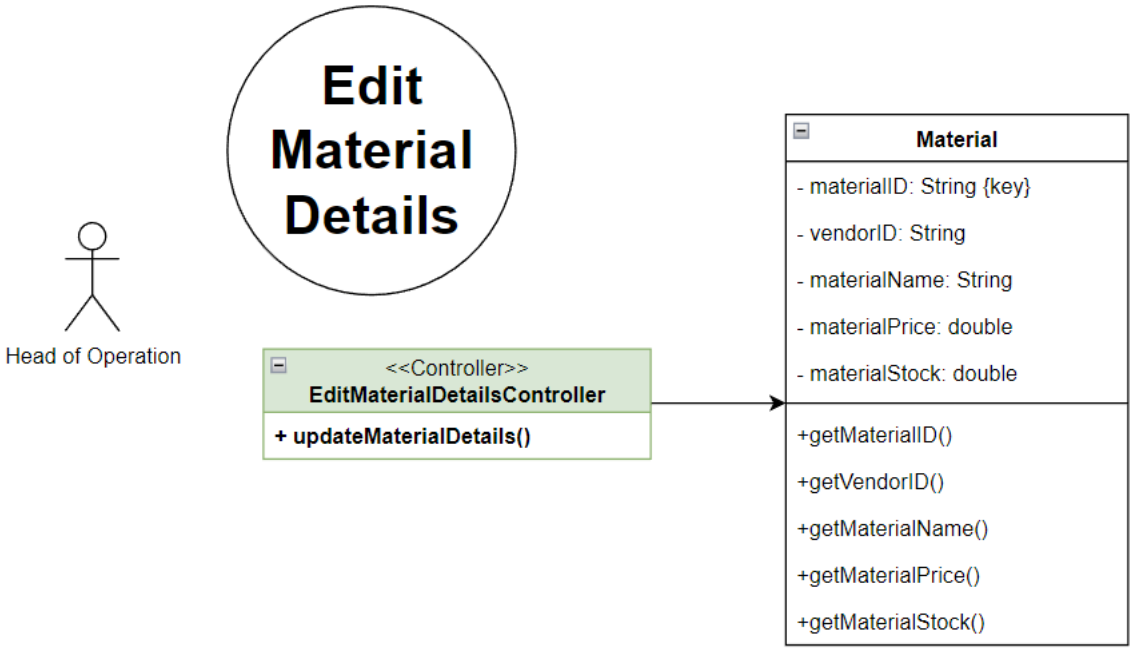
The Inventory Adjustments first cut will retrieve and update stock data on Product, Mixture, and Material.



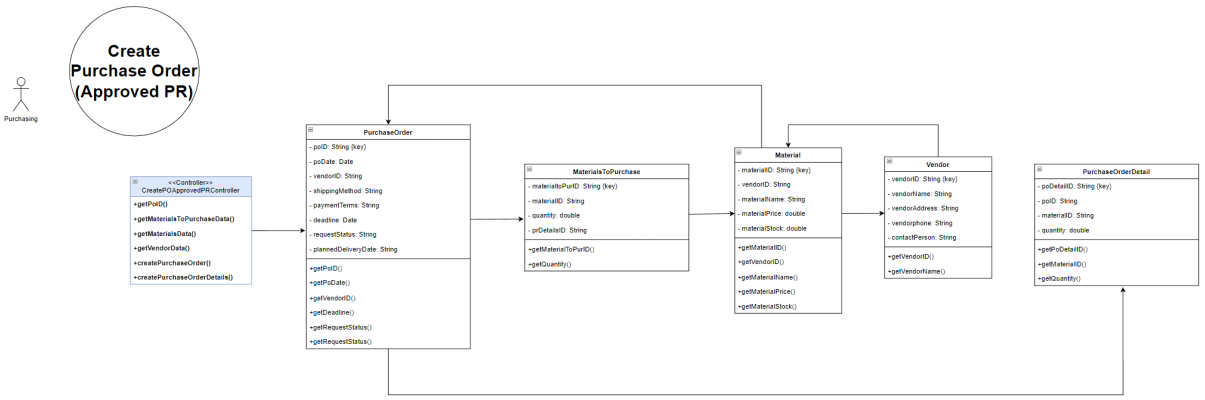
The Edit Vendor Details first cut will retrieve and update data from Vendor and Material.



The Edit Mixture Details first cut will retrieve and update data from Mixture.

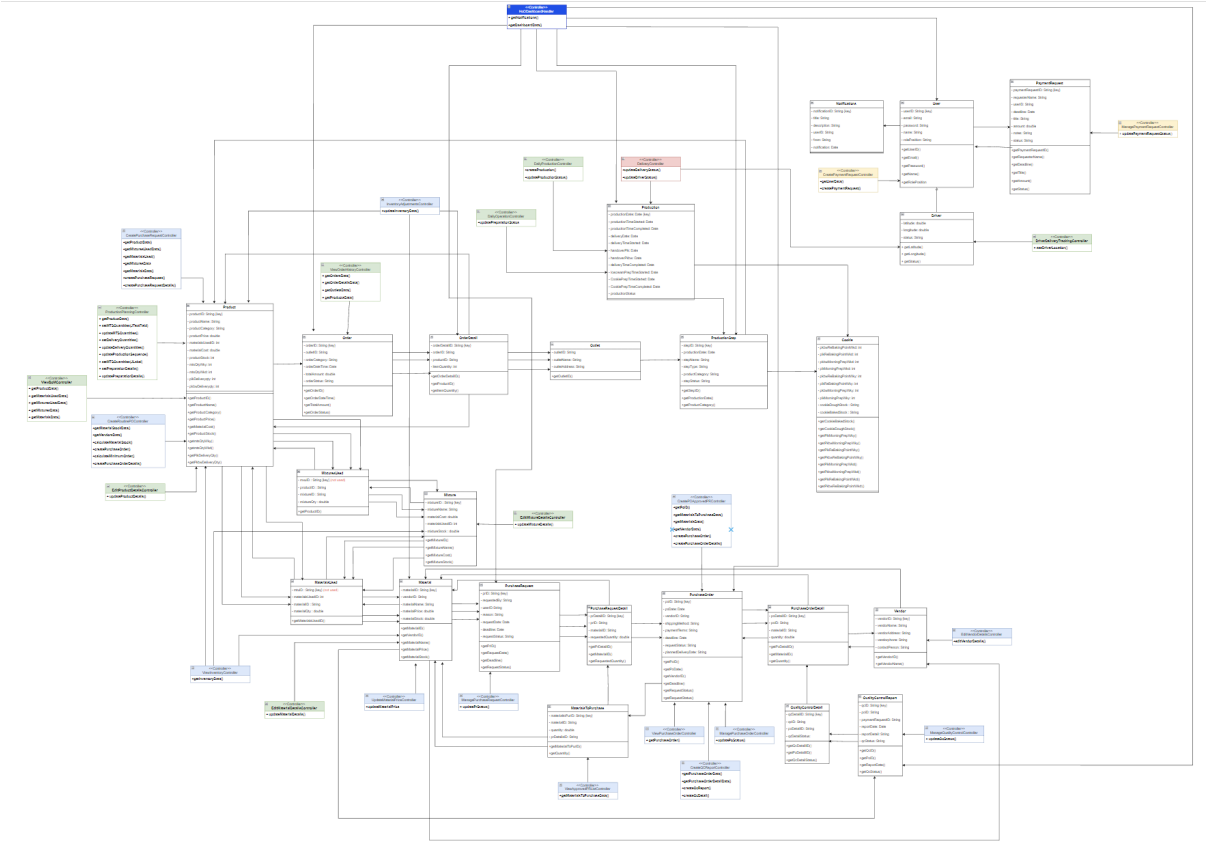


The Edit Material Details first cut will retrieve and update data from Material.



The Create Purchase Order (Approved PR) first cut will retrieve data from PurchaseOrder, MaterialsToPurchase, Material, and Vendor. Then it will create data to PurchaseOrder and PurchaseOrderDetail.

5.3. Revised Class Diagram



After making our first cuts, we consolidated them into a Revised Class Diagram, encompassing all classes utilised in the use cases, along with their respective handlers/controllers. The Revised Class Diagram maintains the connectivity and navigation feasibility of the first cut.

5.4. Use Case Description

HoO Dashboard

Use Case Name:	HoO Dashboard	
Scenario:	Head of operations logs in and views the dashboard.	
Triggering Event:	Log in successful	
Related Use Case:	-	
Brief Description:	Head of operations navigates to their dashboard after logging in, which serves as a visual interface providing information on what they need to do.	
Actors:	Head of operations	
Stakeholders:	HoO	
Preconditions:	- HoO must succeed login	
Postconditions:	- If other menus are open or called.	
Flow of Events:	Actor	System
	1. HoO logs into their account 2. HoO Clicks monitoring dashboard tab	1.1 System displays dashboard [During Production] 2.1 System displays production monitoring dashboard. [After Production] 2.2 System displays operation monitoring dashboard. 2.3 system closes menu.
Except Conditions	HoO failed to log into their account.	

Production Planning

Use Case Name:	Production Planning	
Scenario:	The HoO plans and changes the production plan for the kitchen and outlet staff for the month.	
Triggering Event:	The 1st date of every month (temporal event), the HoO will conduct the production planning for that month.	
Related Use Case:	Daily production, daily operation	
Brief Description:	The HoO opens a production planning menu to update any changes to a product's make to stock quantity and baking quantities of products to name a few. After making any input to the production plan, the HoO would confirm and save the new production plan for that month then it would update the daily production menu for the kitchen staff.	
Actors:	Head of Operation	
Stakeholders:	Kitchen staff	
Preconditions:	The HoO opens the production plan menu on the first day of every month.	
Postconditions:	The HoO completes the production plan and clicks on any other menu on his/her menu bar to exit.	
Flow of Events:	Actor	System
	1. The HoO logs into the system and enters the production plan on the first day of every month. 2. The HoO makes changes to the production plan. Inputting updated make-to-stock quantities of products and click next 3. The HoO confirms and moves to the 2nd step of the production plan. 4. The HoO inputs updated delivery quantities for each outlet and click next. 5. The HoO confirms and moves to the 3rd step of the production plan.	1.1 System opens and displays the production plan menu 1.2 System display products and their current make-to-stock quantities, past sales data, and recap statistics. 2.1 System updates any of the inputs in the production plan for MTS quantities. 2.2 Display a pop-up confirmation to go to the 2nd step of

Use Case Name:	Production Planning	
	6. The HoO Input updated sequence of production and click next. 7. The HoO inputs updated morning cookie preparation quantities for each outlet and clicks finish 8. The HoO exits the menu by going to another menu or logging out of the system.	production planning. 3.1 System saves the 1st step of the production plan. And opens the 2nd step of the production plan. 3.2 The system displays current delivery quantities of every product for each outlet, past sales data, and recap statistics of each outlet 4.1 System saves the 2nd step of the production plan. 4.2 The system displays a pop-up confirmation to go to the 3rd step of production planning 5.1 System saves the 2nd step of the plan. And opens the 3rd step of the production plan. 5.2 The system displays a list of all production steps for the kitchen. 6.1 The system display a pop-up confirmation to go to the 4th step of production planning. 6.2 The system saves the 3rd step of the production plan and opens the 4th step for display. 6.3 The system displays morning preparation quantities and

Use Case Name:	Production Planning	
		make-to-stock quantities of all cookies for each outlet. 7.1 The system display sa pop-up confirmation to finish monthly production planning 7.2 The system saves the final step of the production plan and opens a pop-up message. 7.3 The system displays a pop-up message that production planning has been completed. 8.1 System closes the production plan menu.
Except Conditions	-	

Purchase Order (Mon&Thurs)

Use Case Name:	Create purchase order (Mon & Thurs)
Scenario:	Creating a purchase order routinely every monday and thursday to stock up.
Triggering Event:	Temporal trigger: Monday & Thursday
Related Use Case:	- View approved PR list
Brief Description:	Purchasing creates purchase order(s) based on the products in the approved purchase request and choose the vendor accordingly. Purchasing will fill a form to create purchase orders.
Actors:	Purchasing
Stakeholders:	Purchasing
Preconditions:	- There must be an approved purchase request
Postconditions:	- Purchase order submitted

Use Case Name:	Create purchase order (Mon & Thurs)	
Flow of Events:	Actor	System
	1. Purchasing staff opens the purchase order list. 2. Purchasing staff user creates a purchase order 3. Purchasing staff picks the vendor. 4. Input materials to be ordered, purchase order detail, and email details and click OK. 5. Purchasing staff submits the purchase order. 6. Purchasing staff confirms submission of the purchase order.	1.1 System opens purchase order list menu. 2.1 System opens create purchase order menu. 2.2 System closes purchase order list menu. 2.3 System displays current inventory 2.4 System displays purchase order form. 3.1 System displays vendor's material list. 4.1 System saves purchase order form in menu. 4.2 Display made purchase order. 5.1 System displays purchase order view confirmation pop-up. 6.1 System displays a pop-up message of successful purchase order submission 6.2. System saves and sends a purchase order form through email to the vendor assigned. 6.3 System closes create purchase order menu. 6.4 System re-displays purchase order list menu.
Except Conditions	- Purchase request is rejected	

Products

Use Case Name:	Edit product details	
Scenario:	The HoO wishes to edit the price, stock and details of a product in the inventory.	
Triggering Event:	The HoO wishes to edit the price, stock and details of a product.	
Related Use Case:	View inventory	
Brief Description:	The HoO wishes to edit the price, stock and details of a product, this is done from accessing the inventory menu first.	
Actors:	Head of Operation	
Stakeholders:	Purchasing	
Preconditions:	The inventory menu must be opened first.	
Postconditions:	The new changes has been updated to the material	
Flow of Events:	Actor	System
	1. The HoO enters the inventory menu. 2. The HoO changes the details, price or stock of a selected product	1.1 The system displays the inventory menu. 2.1 The system opens a pop-up window to edit details, price and stock of a product.
Except Conditions	No edits or changes to be made.	

Mixture

Use Case Name:	Edit mixture details
Scenario:	The HoO wishes to edit the price, stock and details of a mixture in the inventory.
Triggering Event:	The HoO wishes to edit the price, stock and details of a

Use Case Name:	Edit mixture details	
	mixture.	
Related Use Case:	View inventory	
Brief Description:	The HoO wishes to edit the price, stock and details of a mixture, this is done from accessing the inventory menu first.	
Actors:	Head of Operation	
Stakeholders:	Purchasing	
Preconditions:	The inventory menu must be opened first.	
Postconditions:	The new changes has been updated to the material	
Flow of Events:	Actor	System
	1. The HoO enters the inventory menu. 2. The HoO changes the details, price or stock of a selected mixture	1.1 The system displays the inventory menu. 2.1 The system opens a pop-up window to edit details, price and stock of a mixture.
Except Conditions	No edits or changes to be made.	

Material

Use Case Name:	Edit material details
Scenario:	The HoO wishes to edit the price, stock and details of a material in the inventory.
Triggering Event:	The HoO wishes to edit the price, stock and details of a material.
Related Use Case:	View inventory
Brief Description:	The HoO wishes to edit the price, stock and details of a material, this is done from accessing the inventory menu first.
Actors:	Head of Operation

Use Case Name:	Edit material details	
Stakeholders:	Purchasing	
Preconditions:	The inventory menu must be opened first.	
Postconditions:	The new changes has been updated to the material	
Flow of Events:	Actor	System
	1. The HoO enters the inventory menu. 2. The HoO changes the details, price or stock of a selected material	1.1 The system displays the inventory menu. 2.1 The system opens a pop-up window to edit details, price and stock of a material.
Except Conditions	No edits or changes to be made.	

Inventory

Use Case Name:	View Inventory
Scenario:	Viewing the inventory menu, seeing products, ingredients and mixtures etc.
Triggering Event:	When staff members and HoO want to view the inventory menu.
Related Use Case:	Inventory adjustments, updating material price.
Brief Description:	All involved actors would view the inventory for various purposes. All actors would see all the stocks, prices and quantities of products, mixtures and ingredients to name a few.
Actors:	Purchasing, outlet, kitchen staff and HoO.

Use Case Name:	View Inventory	
Stakeholders:	-	
Preconditions:	The inventory menu bar has been pressed by the user.	
Postconditions:	A user exits by simply clicking any of the other sections by their menu bars.	
Flow of Events:	Actor	System
	1. Staff members or HoO opens the inventory menu by their menu bar. 2. Staff members or HoO closes the menu by clicking any other menu from their menu bar.	1.1 System displays list of items. 2.1 System closes menu. .
Except Conditions	-	

Use Case Name:	Update Material Price	
Scenario:	Purchasing updates the price of material if it changes.	
Triggering Event:	- Material price change	
Related Use Case:	View inventory	
Brief Description:	Purchasing will update the material price on the products & inventory page by clicking the edit button.	
Actors:	Purchasing	
Stakeholders:	-	
Preconditions:	- The material must exist.	
Postconditions:	Price of material changes in the products and inventory page.	
Flow of Events:	Actor	System
	1. Purchasing presses edit button from products & inventory page. 2. Purchasing set price (type in the	1.1 System displays pop up to set the price. 2.1 System closes pop

Use Case Name:	Update Material Price	
	price) and update	up and shows new updated price of material.
Except Conditions	There are no changes in the price of materials.	

Use Case Name:	Inventory Adjustments
Scenario:	Editing and updating the inventory stock.
Triggering Event:	The outlet or kitchen staff needs to update the stocks in the inventory due to any accidents or modifications occurring during the daily production or planning: - After the view inventory use case is called. The HoO conducts Stock Opname at the end of sundays every week: - After the view inventory use case is called.
Related Use Case:	View inventory
Brief Description:	The inventory adjustments use case is a case where the kitchen, outlet staff and HoO would manually adjust the stocks in the inventory system. Normally, if both staff were able to completely follow the production plans for their daily production and operations, the system would automatically adjust the changes in inventory after a staff member were to update the status of the activity towards completion. However, there are several reasons for staff and the HoO to adjust the stocks in the inventory system manually. If there were any accidents such as spillage of milk or other ingredients during daily production, a kitchen staff member would have to adjust and change the stock quantities in the inventory system due to the event. The same reasoning applies for the outlet staff members if anything were to happen during their daily operation preparation.

Use Case Name:	Inventory Adjustments	
	The Head of Operations conducts the Stock Opname activity every sunday at the end of the day.	
Actors:	Head of Operations, Kitchen and Outlet Staff,	
Stakeholders:	-	
Preconditions:	- The view inventory use case is called. Allowing the user to view the inventory and select items to update its stocks.	
Postconditions:	- Staff or HoO have inputted the changed quantities of items in the inventory. - The update button is pressed	
Flow of Events:	Actor	System
	1. Staff or HoO opens the view inventory menu. 2. Updates any changes to any of the items' stock in inventory. 3. Updates changes and closes adjustment/edit menu.	1.1 System displays list of items. 2.1 System updates changes in inventory to all servers. 3.1 System saves changes. 3.2 System closes menu.
Except Conditions	- For staff, adjusting inventory is not needed if no accidents or modifications were made during the daily production and operations.	

Quality Control

Use Case Name:	Create QC Report
Scenario:	Kitchen staff creating a QC report after products arrive.
Triggering Event:	- Materials arrive at kitchen - Kitchen conducts Quality control
Related Use Case:	-
Brief Description:	Kitchen creates a quality control report by filling in the QC form after doing a quality inspection on the materials that

Use Case Name:	Create QC Report	
	arrived at the kitchen.	
Actors:	Kitchen	
Stakeholders:	Kitchen, Head of Operations	
Preconditions:	- Materials has to arrive at kitchen - Kitchen has to do a quality inspection of the product	
Postconditions:	- Quality control report sent to the HoO where they will manage and send the report to the vendors.	
Flow of Events:	Actor	System
	1. Arrival of ingredient 2. Check quality of the ingredients 3. View report and send complaint to supplier	1.1 Cross check PO with the products arrived 2.1 Input quality review report (whether it is satisfactory or not)
Except Conditions	Products not arriving.	

Use Case Name:	Manage Quality Control
Scenario:	Head of Operations complained to the supplier regarding the QC reports and asked for a redelivery of the cancel invoice.
Triggering Event:	- Kitchen creates quality control report
Related Use Case:	-
Brief Description:	Head of operations views and manages the quality control reports, and complains to the supplier if there is an ingredient quality issue to later negotiate compensation where the supplier could deliver a new product if the invoice gets canceled.

Use Case Name:	Manage Quality Control	
Actors:	Head of operations	
Stakeholders:	Head of Operations,	
Preconditions:	- Quality control report has to be made - Kitchen has to do a quality inspection of the product	
Postconditions:	- Supplier deliver new product - HoO cancels invoice for the product with quality issue	
Flow of Events:	Actor	System
	1. View report and complain to supplier regarding product with quality issue 2. Negotiate compensation with supplier 3. Supplier deliver new products	3.1 System updates quality control status based on whether supplier wants to deliver new products/cancel invoice
Except Conditions	Quality control report not done.	

Payment Request

Use Case Name:	Manage payment request
Scenario:	Handling and updating payment requests.
Triggering Event:	When a new payment request enters the finance staff's notifications towards their list.
Related Use Case:	Submit payment request.
Brief Description:	This is a use case that allows the user to view and update payment requests. The finance staff uses this case to store, approve and reject payment requests sent by the kitchen staff.

Use Case Name:	Manage payment request	
Actors:	Finance staff	
Stakeholders:	Kitchen staff, HoO.	
Preconditions:	- Kitchen staff send payment requests, thus sending notification. - User presses the payment request menu on his/her menubar.	
Postconditions:	- Finance staff users have updated the payment requests by either approving or rejecting them.	
Flow of Events:	Actor	System
	1. Finance staff receive notification of payment request or simply opening his payment request list. 2. Staff user views the details of one payment request. 3. Staff approves or rejects the payment request. 4. Staff closes the list menu by opening another menu on his/her menubar.	1.1 System displays incoming notification pop-up. 1.2 System displays a list of payment requests. 2.1 System opens view payment request detail menu. 3.1 System updates payment requests by either approved or rejected. 3.2 System closes detail menu, displaying payment request list. 4.1 System closes the payment request list menu and displays a new menu.
Except Conditions	-	

Use Case Name:	Submit payment request
Scenario:	Submitting a payment request to the finance staff after conducting an inventory quality control on new ingredients by

Use Case Name:	Submit payment request	
	the supplier.	
Triggering Event:	The tested ingredients passing the company's inventory quality control tests.	
Related Use Case:	Inventory quality control, manage payment requests.	
Brief Description:	Once the kitchen staff has confirmed that the new ingredients passed the quality control test, a user from the staff begins logging into the system to create and submit a payment request to the finance staff to pay the supplier.	
Actors:	Kitchen Staff	
Stakeholders:	Finance staff, HoO.	
Preconditions:	- The ingredients or products have passed the company's inventory quality control tests in the kitchen.	
Postconditions:	- Payment request form is submitted.	
Flow of Events:	Actor	System
	1. A finance staff user logs into his system. 2. User enters the payment request list and views all the requests. 3. User creates a payment request by pressing the create button 4. User submits the payment request form. 5. User leaves payment request menu.	1.1 System displays logging system. 1.2 System closes logging system 2.1 System displays payment request list. 2.2 System opens preview menu on one of the payment requests. 2.3 System closes the preview menu and re-displays the payment request list. 3.1 System opens the create payment request form menu. 3.2 System saves changes on payment request form. 4.1 System sends

Use Case Name:	Submit payment request	
		payment request forms to finance staff through notification and calls to finance staff's payment request list. 4.2 Systems closes the menu and re-displays the payment request list menu. 5.1 System closes payment request list menu.
Except Conditions	- The ingredients or products do not pass the company's inventory quality control tests in the kitchen.	

Delivery

Use Case Name:	Delivery
Scenario:	Driver receives delivery requests from the kitchen staff to bring the manufactured products to both outlets.
Triggering Event:	Kitchen staff finishing daily production and completing the status of their production activity. - Sending a notification to the driver for pick-up.
Related Use Case:	Driver Delivery Tracking
Brief Description:	Driver waits for the kitchen staff to finish daily production. After daily production is completed, the driver will receive a notification for a request to conduct a delivery. Driver will constantly update status all up until completion of the delivery.
Actors:	Driver
Stakeholders:	Kitchen and outlet staff.
Preconditions:	- The daily production activity must be completed and its status must be updated by the kitchen staff for the delivery notification to activate.
Postconditions:	- The PIK outlet staff must update the delivery status to complete to end the use

Use Case Name:	Delivery	
	case process for the driver.	
Flow of Events:	Actor	System
	1. Driver receives notification of delivery request. 2. Driver begins the delivery process. Updating status at each drop-off point. 3. Driver requests for PIK outlet staff to update status to complete the delivery.	1.1 The system displays a notification of a new delivery request. 1.2 The system displays the delivery request list. 2.1 The system opens the view delivery request detail menu. 2.2 The system opens the delivery operation menu. 2.3 System updates status and turns on live location tracking. 2.4 System sends notification to outlet staff of driver's delivery status. 3.1 System updates status of delivery operation after receiving input from outlet staff. 3.2 System saves changes and updates delivery operation to completed status. 3.3 System closes menu.
Except Conditions	- If the daily production is not completed, this use case will not bed called for activation.	

Use Case Name:	Driver Delivery Tracking	
Scenario:	The outlet and kitchen staff view the live location of the driver for either pick-up or expecting delivery purposes.	
Triggering Event:	After the daily production is complete, the kitchen staff finishes the production sequence, automatically sends a delivery request which activates the live location tracking for the driver.	
Related Use Case:	Daily production, Daily operation.	
Brief Description:	After the daily production is complete, the kitchen staff finishes the production sequence, automatically sends a delivery request which activates the live location tracking for the driver. When the driver begins the delivery operation, the live location of the driver is relayed back to the screen of the outlet and kitchen staff's system. The outlet staff will get updates from the driver about the status of the delivery operation.	
Actors:	Outlet and kitchen staff.	
Stakeholders:	Driver	
Preconditions:	The daily production sequence must be completed first.	
Postconditions:	The outlet staff must update the delivery operation to be completed and close this menu.	
Flow of Events:	Actor	System
	1. Kitchen staff view the live location tracking menu. 2. Once the driver arrives and updates the status of the operation to start delivering the products, the kitchen staff is finished. 3. Outlet staff can view the live location and receive updates of the status from the driver from the same menu. 4. Outlet staff can complete the delivery operation after receiving the products from the driver.	1.1 System displays the live location tracking menu. 2.1 System would close the live location tracking menu for the Kitchen staff. 3.1 System keeps displaying the live location tracking menu for the outlet staff, notifying the outlet staff of any progress updates from status changes by the driver. 4.1 System notifies the outlet staff of the

Use Case Name:	Driver Delivery Tracking	
		arrival of the products. 4.2 System opens a pop-up window to complete the delivery operation after the staff opens the “update status” button from the live location tracking menu. 4.3 System updates the delivery operation to complete it. 4.4 System closes the live location tracking menu and opens the daily operation menu.
Except Conditions	The kitchen staff does not complete the daily production sequence.	

Purchase Orders

Use Case Name:	Create purchase order (Approved PR)	
Scenario:	Creating a purchase order based on the list of products in the approved purchase request.	
Triggering Event:	Purchase request is approved	
Related Use Case:	- View approved PR list	
Brief Description:	Purchasing creates purchase order(s) based on the products in the approved purchase request and choose the vendor accordingly. Purchasing will fill a form to create purchase orders.	
Actors:	Purchasing	
Stakeholders:	Purchasing	
Preconditions:	- There must be an approved purchase request	
Postconditions:	- Purchase order submitted	
Flow of Events:	Actor	System

Use Case Name:	Create purchase order (Approved PR)	
	1. Purchasing opens approved PR list 2. Purchasing fill in purchase order form according to materials/vendor 3. Purchasing submits purchase order	1.1 System displays PR with materials to buy 2.1 System shows list of materials that vendor offers 3.1 System displays confirmation of purchase order
Except Conditions	- Purchase request is rejected -	

Use Case Name:	Create purchase order (Mon & Thurs)	
Scenario:	Creating a purchase order routinely every monday and thursday to stock up.	
Triggering Event:	Temporal trigger: Monday & Thursday	
Related Use Case:	- View approved PR list	
Brief Description:	Purchasing creates purchase order(s) based on the products in the approved purchase request and choose the vendor accordingly. Purchasing will fill a form to create purchase orders.	
Actors:	Purchasing	
Stakeholders:	Purchasing	
Preconditions:	- There must be an approved purchase request	
Postconditions:	- Purchase order submitted	
Flow of Events:	Actor	System
	1. Purchasing staff opens the purchase order list. 2. Purchasing staff user creates a purchase order 3. Purchasing staff picks the vendor. 4. Input materials to be ordered,	1.1 System opens purchase order list menu. 2.1 System opens create purchase order menu. 2.2 System closes purchase order list

Use Case Name:	Create purchase order (Mon & Thurs)	
	purchase order detail, and email details and click OK. 5. Purchasing staff submits the purchase order. 6. Purchasing staff confirms submission of the purchase order.	menu. 2.3 System displays current inventory 2.4 System displays purchase order form. 3.1 System displays vendor's material list. 4.1 System saves purchase order form in menu. 4.2 Display made purchase order. 5.1 System displays purchase order view confirmation pop-up. 6.1 System displays a pop-up message of successful purchase order submission 6.2. System saves and sends a purchase order form through email to the vendor assigned. 6.3 System closes create purchase order menu. 6.4 System re-displays purchase order list menu.
Except Conditions	- Purchase request is rejected	

Use Case Name:	Manage purchase order
Scenario:	The purchasing staff wants to update, create or view the purchase orders from the purchasing order menu.
Triggering Event:	A purchasing staff user opens the purchasing order list menu to view the purchase orders.

Use Case Name:	Manage purchase order	
Related Use Case:	Create purchase orders (2 use cases), view purchase orders.	
Brief Description:	The purchasing staff manages all purchase orders for the company through the purchase orders list menu in the system. Here they can create, delete and view any of the purchase orders as they all would be saved here.	
Actors:	Purchasing staff	
Stakeholders:	HoO, Kitchen staff	
Preconditions:	A purchasing staff user must open the purchase order menu from his/her menu bar.	
Postconditions:	A purchasing staff user opens another menu from his/her menu bar.	
Flow of Events:	Actor	System
	1. Purchasing staff clicks on the purchase order menu from his/her menu bar. 2. Purchasing staff user either creates or views a purchase order based on his/her intentions. 3. Purchasing staff user goes to another menu or logs out of the system.	1.1 System displays purchase order list menu. 2.1. System will open another menu depending on the user's intentions. 2.2 System would close the purchase order list menu in response. 2.3 System will re-display purchase order list menu,. 3.1. System logs users out or displays another menu.
Except Conditions	-	

Use Case Name:	View purchase order
Scenario:	The purchasing staff views a purchase order from the list of purchase orders for inspection purposes.

Use Case Name:	View purchase order	
Triggering Event:	The purchasing staff user would open the view button for a select purchase order from the list menu.	
Related Use Case:	Manage purchase order, create purchase orders (2 use cases).	
Brief Description:	The purchasing staff would view the purchase order by pressing on the view button on a purchase order in the list menu. Normally, the purchasing staff would do this to inspect and check any purchase order based on the request of his superior or for other reasons.	
Actors:	Purchasing staff.	
Stakeholders:	-	
Preconditions:	The purchase order list menu must be called/open first.	
Postconditions:	The purchasing staff user clicks out of or exits the view purchase order menu.	
Flow of Events:	Actor	System
	1. Purchasing staff opens the purchase order list menu. 2. Purchasing staff press the view button on a purchase order to view the order. 3. Purchasing staff closes the view menu.	1.1 System displays the purchase order list menu. 2.1 System closes the purchase order list menu. 2.2 System opens the view menu of a purchase order. 3.1 System closes the view menu. 3.2 System re-displays the purchase order list menu.
Except Conditions	-	

Daily

Use Case Name:	Daily operation	
Scenario:	The outlet staff begins the daily operation after products arrive.	
Triggering Event:	Once the driver arrives and delivers the products to the outlet.	
Related Use Case:	Driver delivery tracking	
Brief Description:	After the driver delivers all the products to the outlets, the outlet staff will begin the daily operation. The outlet user logs into the system and enters the daily operation menu in his/her menu bar. The user would click the update button on his live location tracking menu to complete the delivery operation, thus showing the daily operation menu. The outlet staff user will press the "finish buttons" after completing each stage of the daily operation.	
Actors:	Outlet Staff	
Stakeholders:	-	
Preconditions:	The products must arrive at the outlet first before starting the daily operation.	
Postconditions:	The outlet staff user clicks out of the menu by going to another menu or logging out of the system.	
Flow of Events:	Actor	System
	1. The outlet staff updates the live location tracking to complete the delivery operation. 2. The outlet staff starts the daily operation. 3. The outlet staff progresses the daily operation by clicking on the "finish" buttons after completing each stage in the operation. 4. The outlet staff exits the menu by clicking another menu on his/her menubar.	1.1 System displays the live location tracking menu. 1.2 System opens a pop-up window to update the status of the live location tracking. 2.1 The System completes the delivery operation and opens the daily operation menu. 3.1 System updates the menu after each activation of the "finish" buttons. 4.1 System closes the

Use Case Name:	Daily operation	
		daily operation menu.
Except Conditions	The driver fails to deliver the products to the outlets.	

Use Case Name:	Daily production	
Scenario:	The kitchen staff conducts and completes the daily production sequence everyday beginning at 06:00.	
Triggering Event:	Everyday at 06:00 (temporal event), the kitchen staff clicks on the start production button.	
Related Use Case:	Driver delivery tracking	
Brief Description:	Everyday at 06:00, a kitchen staff user logs into the system and enters the daily production menu to begin the daily production sequence. The user would click the start production button and would follow accordingly to complete the production sequence, constantly updating the sequence until it is complete. After the sequence is complete, the system will automatically notify the driver with a delivery request.	
Actors:	Kitchen staff.	
Stakeholders:	Driver.	
Preconditions:	The production sequence will begin everyday at 06:00.	
Postconditions:	The staff must update and complete all the production sequences to begin the next stage.	
Flow of Events:	Actor	System
	1. Kitchen staff log into the system at 06:00. 2. Kitchen staff starts the daily production sequence by pressing the start production button. 3. Kitchen staff users update the sequence by clicking the “finish” button on each stage when they have completed it.	1.1 System displays the daily production menu. 2.1 System updates the daily production plan upon activation of the “start production” button. 3.1 System updates the daily production after

Use Case Name:	Daily production	
	4. After the production is complete, the kitchen staff finishes all the stages. 5. The kitchen staff can exit the menu after viewing the driver location.	each activation of the “finish” buttons. 4.1 System automatically sends delivery requests to the driver after the kitchen staff completes everything. 4.1 System calls live location tracking menu. 4.2 System opens a live location tracking menu if the user opens it. 4..2 System closes daily production menu as a response. 4.3 System re-display production plan menu after staff user is done viewing the location tracking. 5.1 System closes the daily production menu.
Except Conditions	-	

Purchasing Request

Use Case Name:	View approved PR list
Scenario:	The head of operations can view the his/her approved purchase request list. The purchasing staff can view all the materials that have been approved in the purchase request.
Triggering Event:	The head of operation and purchasing staff wishes to see the approved list.
Related Use Case:	Create purchase request,
Brief Description:	The head of operations will view the approved purchase request list after approving or rejecting the purchase

Use Case Name:	View approved PR list	
	requests. The purchasing staff also has access to the approved pr list as well.	
Actors:	Purchasing, Head of Operation	
Stakeholders:	-	
Preconditions:	The HoO must open the approved pr material list from the purchase request list menu. The purchasing staff as well.	
Postconditions:	The HoO and purchasing staff can exit the menu by gong to another menu.	
Flow of Events:	Actor	System
	1. The HoO can view the menu by clicking to it from the purchase request list menu. 2. The purchasing staff views it from his/her purchase request list menu.	1.1 The system displays the approved pr list menu.
Except Conditions	-	

Use Case Name:	Create purchase request
Scenario:	The outlet staff creates a purchase request to the Head of Operations after receiving a bulk order from a customer.
Triggering Event:	The outlet staff wants to create a purchase request by clicking the create button in the purchase request list menu.
Related Use Case:	Manage purchase request
Brief Description:	After receiving a bulk order from a customer either from the company's whatsapp or phone number, the outlet staff would log on to the system. After logging into the system, the outlet staff will head to the purchase request list menu and press the create button. The outlet staff would then fill in the purchase request form and submit the form which the HoO will receive through notification.
Actors:	Outlet staff
Stakeholders:	Head of Operations

Use Case Name:	Create purchase request	
Preconditions:	The purchase request list menu must be opened first.	
Postconditions:	The outlet staff will have to press either submit or cancel to close the menu.	
Flow of Events:	Actor	System
	1. Outlet staff user opens the purchase request list menu. 2. Outlet staff creates the purchase request. 3. Outlet staff submits the purchase request	1.1 System displays the purchase request list menu. 2.1 System opens the create purchase request menu. 2.2 System updates the purchase request. 2.3 System saves the created purchase request. 3.1 System submits the purchase request to HoO. 3.2 System closes the create purchase request menu. 3.3 System re-displays the purchase request list menu.
Except Conditions	The outlet staff does not receive any bulk orders.	

Use Case Name:	Manage purchase request
Scenario:	A purchasing staff user opens the purchasing order list menu to view the purchase orders.
Triggering Event:	An outlet staff user submits a purchase request.
Related Use Case:	Create Purchase request, create purchase order.
Brief Description:	The HoO receives all purchase requests for the company through the purchase request list menu in the system. Here they can create purchase orders, view any of the purchase requests to approve or reject.

Use Case Name:	Manage purchase request	
Actors:	Head of Operation	
Stakeholders:	Outlet Staff	
Preconditions:	Outlet staff submits a purchase request.	
Postconditions:	HoO clicks another menu or logs out of the system.	
Flow of Events:	Actor	System
	1. HoO opens the purchase request list menu. 2. HoO views one of the purchase requests 3. HoO approves or rejects the following purchase request. 4. HoO closes the menu by going to another menu or logging out of the system.	1.1 System displays the purchase request list menu. 2.1 System closes the purchase request list menu. 2.2 System opens the view purchase request menu in response. 2.3 System updates the purchase request by approved or rejected. 2.4 System sends notification to outlet staff of approved or rejected status. 2.5 System closes the view purchase request menu and re-displays the purchase request list menu. 4.1 System closes the menu by opening another or by logging the user out.
Except Conditions	The outlet staff does not submit any purchase request.	

Bill of Material

Use Case Name:	View BOM
Scenario:	The HoO views the bill of material of a product when

Use Case Name:	View BOM	
	conducting the production planning.	
Triggering Event:	During production planning, when the HoO wishes to view the bill of material by clicking on the view button on one of the products.	
Related Use Case:	-	
Brief Description:	To view the bill of material of a product, it is available in the production planning menu for the head of operations. The HoO would click on the view button of a product if he wishes to view the bill of material on the product.	
Actors:	Head of Operation	
Stakeholders:	Purchasing	
Preconditions:	The production planning menu must be open first.	
Postconditions:	The head of operation can click out of the view bill of the material menu.	
Flow of Events:	Actor	System
	1. The HoO opens the production plan menu. 2. The HoO views the bill of material of a select product. 3. The HoO exits from the view menu.	1.1 The system displays the production planning menu. 2.1 The system opens a pop-up window of the bill of material menu of a selected product. 3.1 The system closes the pop-up window and re-displays the production planning menu.
Except Conditions	The HoO does not wish to view the bill of material.	

Order

Use Case Name:	View Order History
Scenario:	The HoO views the history of all orders from customers in the

Use Case Name:	View Order History	
	outlets, to shopee, gojek and more.	
Triggering Event:	The HoO simply clicks on the order history menu in the menu bar.	
Related Use Case:	-	
Brief Description:	The Head of Operations can view the history of all of the orders made to the company in his or her menubar. It contains orders from walk-in outlets, to shopee and gojek and more.	
Actors:	Head of Operations	
Stakeholders:	-	
Preconditions:	The HoO must press the order history menu in his/her menubar.	
Postconditions:	The HoO can exit this menu by going to another menu.	
Flow of Events:	Actor	System
	1. The HoO enters the order history list menu. 2.The HoO selects on of the orders and views its details.	1.1 System displays order history menu. 2.1 System opens view order history menu.
Except Conditions	-	

Vendor

Use Case Name:	Edit Vendor Details
Scenario:	The purchasing staff wishes to edit the information detail of a vendor to keep accuracy.
Triggering Event:	A purchasing staff user opens the edit vendor button in the vendor list menu.
Related Use Case:	Purchase order
Brief Description:	A purchasing staff user wishes to edit the details of a recorded vendor in the system due to changes of new information either provided by the vendor or from other

Use Case Name:	Edit Vendor Details	
	sources. The staff would have to edit the details from the vendor list menu.	
Actors:	Purchasing staff	
Stakeholders:	Head of Operations	
Preconditions:	The vendor list menu must be open.	
Postconditions:	The purchasing staff closes the menu after updating the vendor's details.	
Flow of Events:	Actor	System
	1. Purchasing staff opens the vendor list menu. 2. Purchasing staff opens the menu to edit a vendor's details. 3. User finishes editing the details and saves.	1.1 The system displays the vendor list menu. 2.1 The system opens a pop-up window to edit the vendor's details. 3.1 The system inputs changes made by the user. 3.2 The system updates all changes done by the user. 3.3 The system closes the pop-up window and re-displays the vendor list menu. 4.1 example. Example.
Except Conditions	There is no need to edit the vendor's details.	

5.5. Event Table

Events	Type of Trigger	Trigger	Source	Use Case	Output	Destination
Head of Operations Monthly Production Planning	Temporal event	27th date of the month	System	Monthly Production Planning	Plan of production	System
View Bill Of Material of Products	State Event	When Monthly Production Planning use case is called, sometimes when Submit Purchase Request use case is called.	System	View BoM	Access to display	head of operations
Viewing Make To Stock Quantity	State Event	When the Kitchen Staffs are going to start daily product manufacturing and checks the system to see the determined quantity.	Staff	View MTS Quantity	Access to display	Kitchen Staff
Submitting a purchase request	State & Temporal Event	Incoming Bulk order or regularly twice a week		Submit Purchase Request	Purchase request	head of operation
Managing purchase requests	State Event	Submitted purchase request by Outlet Staff	Staff	Manage Purchase Request	Purchase request is managed	System
Viewing Submitted Purchase Requests	State Event	When Submit Purchase Request, Manage Purchase Request, and Send Purchase Order use cases are called.	Purchase Requester	View Purchase Request	Access to display	Outlet Staff
Send a Purchase Order in the process of procuring Ingredient	Temporal event	Twice a week based on approved incoming purchase request list	System	Send Purchase Order	Purchase order	Supplier
View a Purchase Order in the process of procuring Ingredient	State Event	The Head of Operations or Purchasing needs to view a purchase order to verify its status or content.	Head of Operations	View Purchase Order	Access to display	Head of operations
Update a Purchase Order in the process of procuring		Kitchen staff receive the delivery of goods and confirm that they				

5.6. Data Collection of Use Case

Classification	Business Process	Use Case	Data Collection		Literature Source	
			Interview	Observation	Website	SAP
Inventory		View Inventory	☑	☑	☑	☐
		Inventory Adjustment	☐	☐	☑	☑
		Update Material Price	☑	☐	☐	☑
Inventory and Procurement		Create New Product	☐	☐	☐	☐
		View approved PR list	☐	☐	☐	☑
		Manage purchase request	☐	☐	☐	☑
Purchase Order		Create purchase order (Approved PR)	☐	☐	☑	☑
		Create purchase order (Mon & Thurs)	☑	☐	☐	☐
		Manage Purchase Order	☑	☐	☐	☐
		View Purchase Order	☑	☐	☐	☑
Inventory Quality Control		Create QC Report	☐	☐	☐	☑
		Manage Quality Control	☐	☐	☐	☑
Payment Request		Submit Payment Request	☐	☐	☐	☑
		Manage Payment Request	☑	☐	☐	☑
Product		Create New Product	☐	☐	☐	☑
		Delete Product	☐	☐	☐	☑
		Edit Product Details	☐	☑	☐	☑
Mixture		Create New Mixture	☐	☐	☐	☑
		Delete Mixture	☐	☐	☐	☑
		Edit Mixture Details	☑	☑	☐	☑
Material		Create New Material	☐	☐	☐	☑
		Delete Material	☐	☐	☐	☑
		Edit Material Details	☑	☑	☐	☑
Production		Create New Production Step	☐	☐	☐	☑
		Delete Production Step	☐	☐	☐	☑
		Production Planning	☑	☑	☐	☑
User		Create New User	☐	☐	☐	☑
		Delete User	☐	☐	☐	☑
Vendor		Create New Vendor	☑	☐	☐	☑
		Delete Vendor	☐	☐	☐	☑
		Edit Vendor Details	☑	☐	☐	☑
Outlet		Create New Outlet	☑	☐	☐	☑
		Delete Outlet	☑	☐	☐	☑
Delivery		Delivery	☑	☐	☐	☑
		Driver Delivery Tracking	☑	☐	☐	☑
HoO Dashboard		HoO Dashboard	☐	☐	☐	☑
Daily		Daily Operation	☑	☐	☐	☑
		Daily Production	☑	☐	☐	☑
BoM		View BoM	☑	☑	☐	☑
Order		View Order History	☑	☑	☐	☑

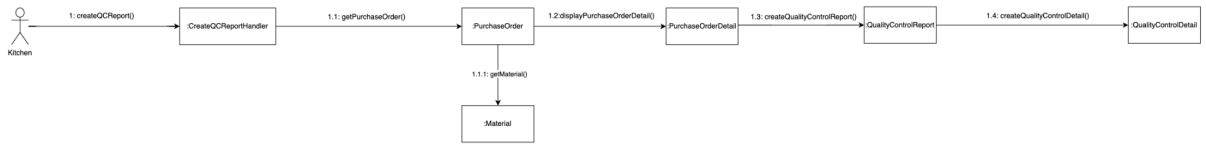
5.7. Difficulty Mapping of Use Case

Classification / Business Process	Use Case	Analysis			Design			Coding		
		Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
Inventory	View Inventory	☑	☐	☐	☐	☑	☐	☐	☑	☐
	Inventory Adjustment	☐	☑	☐	☐	☑	☐	☐	☑	☐
	Update Material Price	☑	☐	☐	☑	☐	☐	☑	☐	☐
Inventory and Procurement	Create New Product	☑	☐	☐	☑	☐	☐	☐	☑	☐
	View approved pr list	☐	☑	☐	☐	☑	☐	☐	☑	☐
	Manage purchase request	☑	☐	☐	☐	☑	☐	☐	☑	☐
Purchase Order	Create purchase order (Approved PR)	☐	☐	☑	☐	☑	☐	☐	☑	☐
	Create purchase order (Mon & Thurs)	☐	☐	☑	☐	☐	☑	☐	☐	☑
	Manage Purchase Order	☑	☐	☐	☐	☑	☐	☐	☑	☐
	View Purchase Order	☑	☐	☐	☑	☐	☐	☑	☐	☐
Inventory Quality Control	Create QC Report	☑	☐	☐	☑	☐	☐	☑	☐	☐
	Manage Quality Control	☑	☐	☐	☑	☐	☐	☑	☐	☐
Payment Request	Submit Payment Request	☑	☐	☐	☑	☐	☐	☑	☐	☐
	Manage Payment Request	☑	☐	☐	☐	☑	☐	☐	☑	☐
Product	Create New Product	☑	☐	☐	☑	☑	☐	☑	☐	☐
	Delete Product	☑	☐	☐	☑	☐	☐	☑	☐	☐
	Edit Product Details	☑	☐	☐	☑	☐	☐	☑	☐	☐
Mixture	Create New Mixture	☑	☐	☐	☐	☑	☐	☐	☑	☐
	Delete Mixture	☑	☐	☐	☑	☐	☐	☑	☐	☐
	Edit Mixture Details	☑	☐	☐	☑	☐	☐	☑	☐	☐
Material	Create New Material	☑	☐	☐	☐	☑	☐	☑	☐	☐
	Delete Material	☑	☐	☐	☑	☐	☐	☑	☐	☐
	Edit Material Details	☑	☐	☐	☑	☐	☐	☑	☐	☐
Production	Create New Production Step	☑	☐	☐	☐	☑	☐	☐	☑	☐
	Delete Production Step	☑	☐	☐	☑	☐	☐	☑	☐	☐
	Production Planning	☐	☐	☑	☐	☐	☑	☐	☐	☑
User	Create New User	☑	☐	☐	☑	☐	☐	☐	☑	☐
	Delete User	☑	☐	☐	☑	☐	☐	☑	☐	☐
Vendor	Create New Vendor	☑	☐	☐	☐	☑	☐	☐	☑	☐
	Delete Vendor	☐	☐	☐	☑	☐	☐	☑	☐	☐
	Edit Vendor Details	☑	☐	☐	☑	☐	☐	☑	☐	☐
Outlet	Create New Outlet	☑	☐	☐	☐	☑	☐	☐	☑	☐
	Delete Outlet	☑	☐	☐	☑	☐	☐	☑	☐	☐
Delivery	Delivery	☑	☐	☐	☐	☐	☑	☐	☐	☑
	Driver Delivery Tracking	☑	☐	☐	☐	☑	☐	☐	☑	☐
HoO Dashboard	HoO Dashboard	☐	☐	☑	☐	☐	☑	☐	☐	☑
Daily	Daily Operation	☑	☐	☐	☐	☑	☐	☑	☐	☐
	Daily Production	☑	☐	☐	☐	☑	☐	☑	☐	☐
BoM	View BoM	☐	☑	☐	☐	☑	☐	☐	☑	☐
Order	View Order History	☐	☐	☐	☑	☐	☐	☑	☐	☐

5.7. Communication Diagram

1. Create Quality Control Report

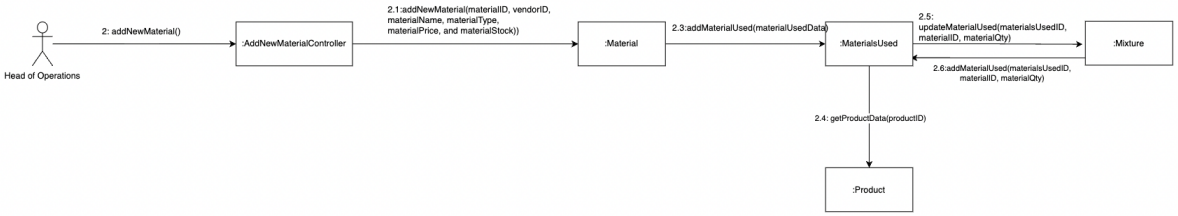
The communication diagram showcases the interaction and method invocations between the PurchaseOrder, Material, PurchaseOrderDetail, QualityControlReport, and QualityControlDetail classes in the "Create QC Report" use case. The PurchaseOrder class initiates the process by calling the method "getPurchaseOrder()", which communicates with the Material class through the method "getMaterial()" to retrieve the relevant material information associated with the purchase order. The PurchaseOrder class then communicates with the PurchaseOrderDetail class via the method "displayPurchaseOrderDetail()" to display detailed information about the purchase order. Subsequently, the QualityControlReport class is invoked through the method "createQualityControlReport()" to generate a quality control report based on the purchase order data. Finally, the QualityControlDetail class is involved through the method "createQualityControlDetail()" to create specific details related to the quality control report. This communication diagram demonstrates the sequential flow of method invocations, enabling the creation of quality control reports and associated details in the context of purchase order management.



2. Add New Material

The communication diagram illustrates the interaction and method invocations between the Material, MaterialsUsed, Product, and Mixture classes in the "Add New Material" use case. Starting with the Material class, the method "addNewMaterial()" is called with parameters such as material ID, vendor ID, material name, material type, material price, and material stock, allowing the addition of a new material. The MaterialsUsed class is then involved through the method "addMaterialUsed()" to add data related to the usage of materials. The Product class is communicated with using the method "getProductData()" to retrieve specific product data. Finally, the Mixture class is utilized through the methods "updateMaterialUsed()" and "addMaterialUsed()" to update and add material usage

information. This communication diagram demonstrates the flow of method invocations, facilitating the addition of new materials and managing material usage in the context of product mixtures.



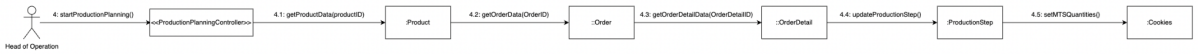
3. Create Purchase Request

The communication diagram outlines the interaction between the Product, Material, PurchaseRequestDetail, and PurchaseRequest classes for the "Create Purchase Request" use case. The Product class retrieves product data using the "getProductData(productName)" method, which is then passed to the Material class via the "getMaterialData(materialName)" method to obtain the relevant material information. The PurchaseRequestDetail class sets the purchase request details with the provided identifier using the "setPurchaseRequestDetail(prDetailID)" method, and finally, the PurchaseRequest class sets the purchase request using the provided identifier with the "setPurchaseRequest(prID)" method. This diagram demonstrates the sequential flow of method invocations, allowing for the creation of a purchase request with the necessary details.

4. Production Planning

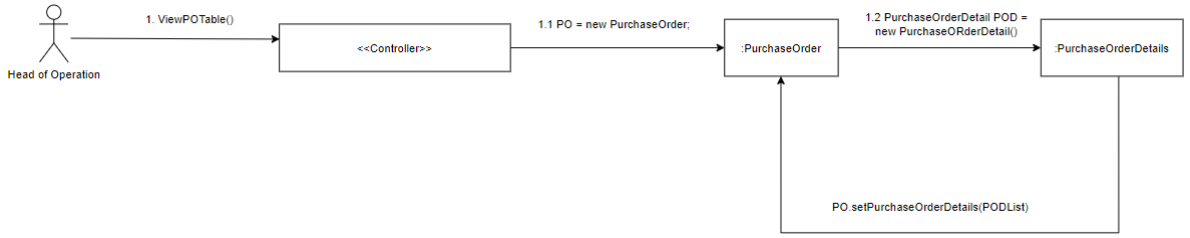
The communication diagram depicts the interaction and method invocations between the Product, Order, OrderDetail, ProductionStep, and Cookies classes in the "Production Planning" use case. The Product class retrieves product data, which is then passed to the Order class to obtain order data. The OrderDetail class retrieves specific order details, which are utilized by the ProductionStep class to update production steps. Finally, the ProductionStep class communicates with the Cookies class to set the quantities of cookies

required for production planning. This diagram illustrates the sequential flow of method invocations and the exchange of data, showcasing the coordination of steps in the production planning process.



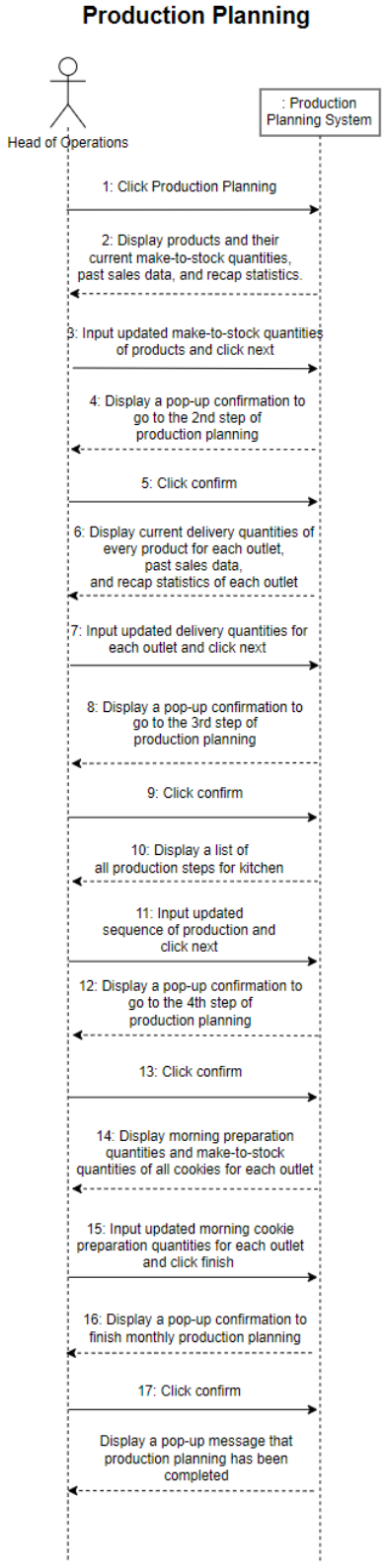
5. View Purchase Order

The communication diagram depicts the interaction between the PurchaseOrder and PurchaseOrderDetails classes for the "View Purchase Order" use case. The view class calls the viewPOTable() method from the controller, which communicates with the PurchaseOrder class through instantiating the class. We then instantiate the PurchaseOrderDetails class and we will put all the PurchaseOrderDetails into the PurchaseOrder by calling the setPurchaseOrderDetails() method from the PurchaseOrder class.

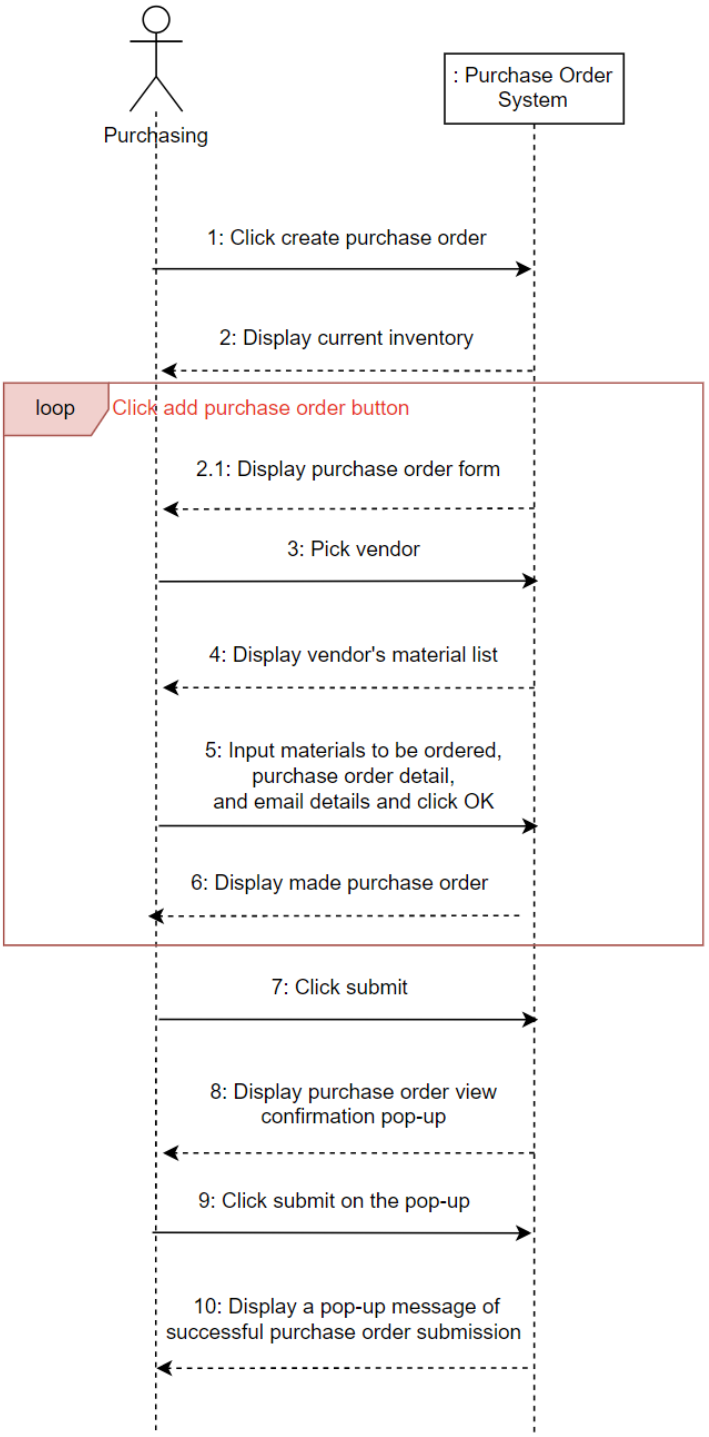


5.8. System Sequence Diagram

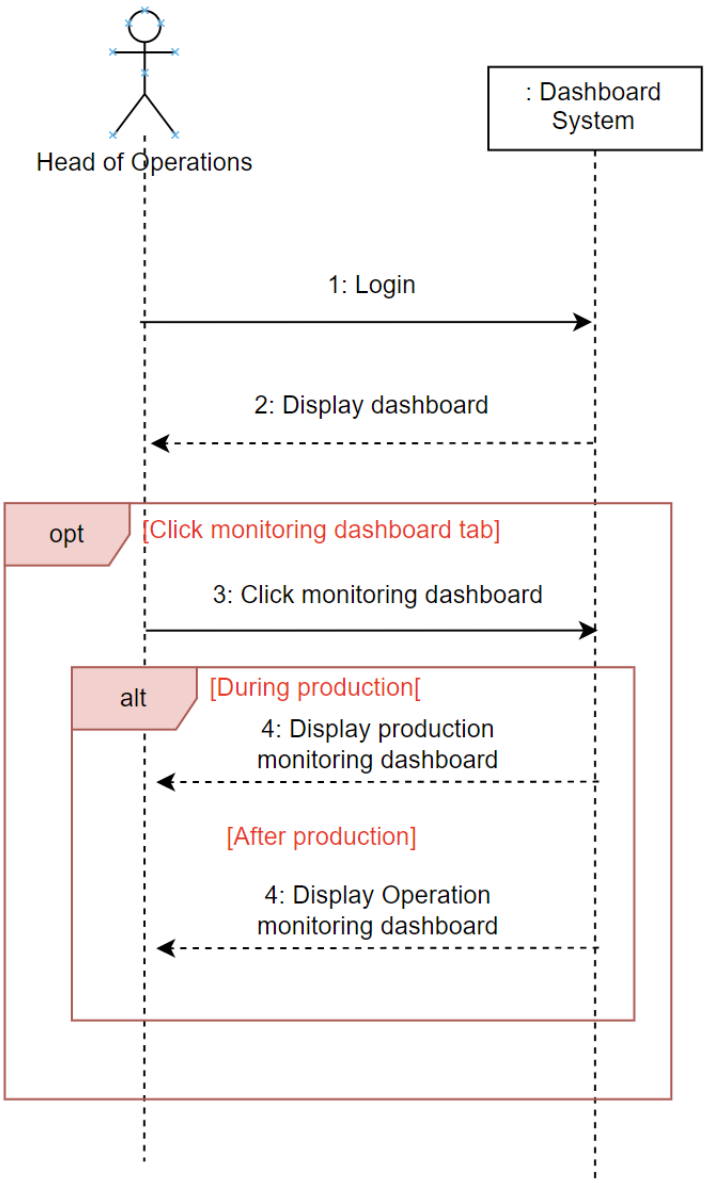
1. Production Planning



2. Create Purchase Orders (Monday / Thursday)

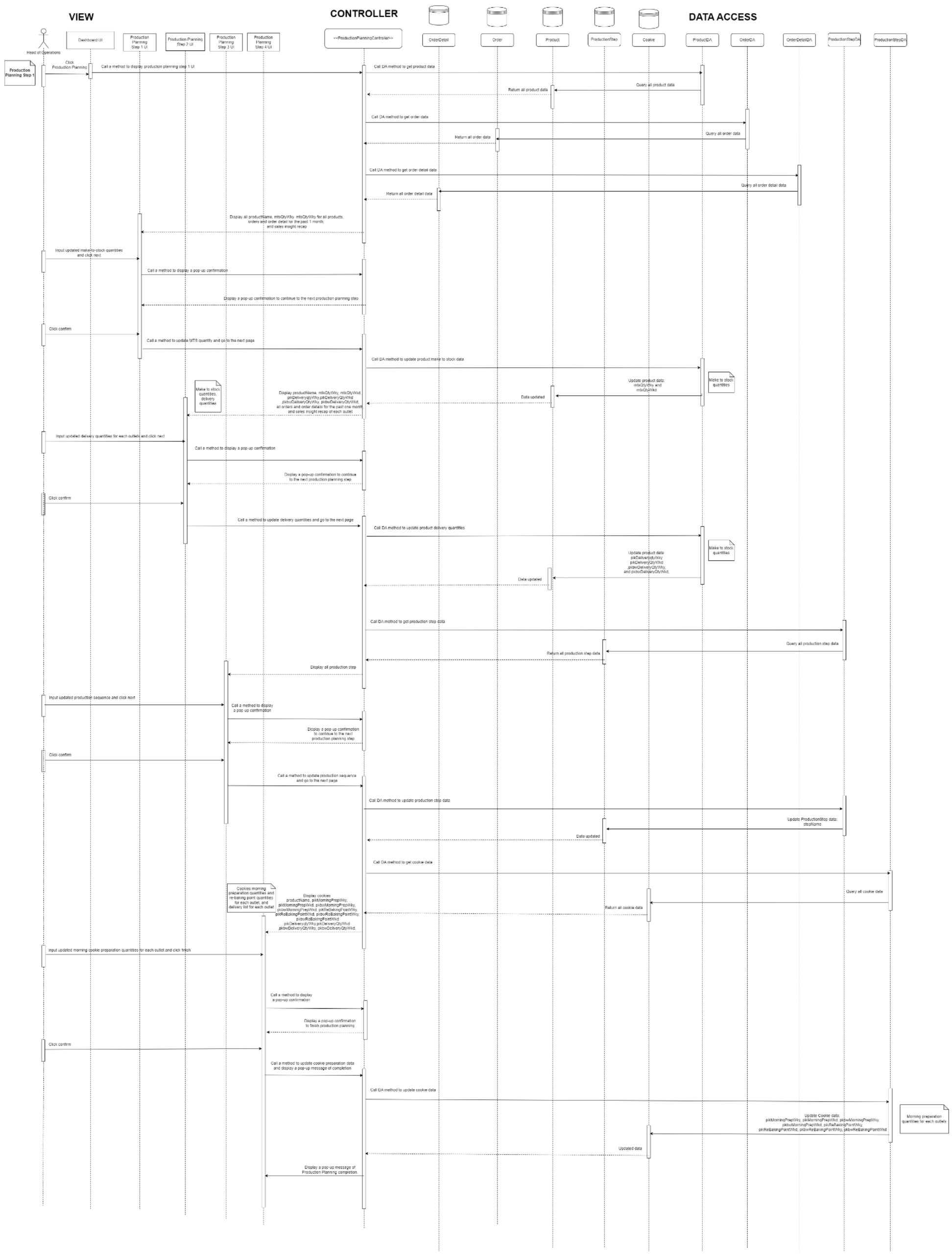


3. Head of Operations Dashboard


Head of Operations
: Dashboard System
1: Login
2: Display dashboard
opt
[Click monitoring dashboard tab]
3: Click monitoring dashboard
alt
[During production[
4: Display production monitoring dashboard
[After production]
4: Display Operation monitoring dashboard

5.9. Multilayer Diagram

1. Production Planning

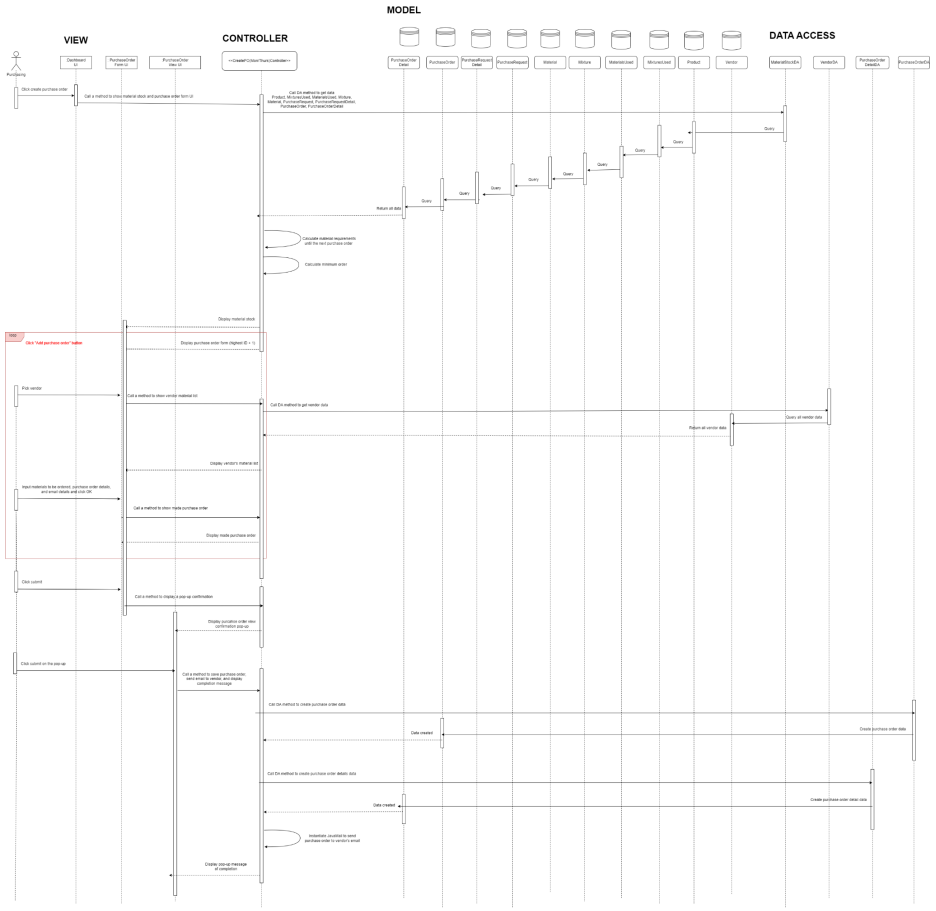


In the multilayer diagram of Production Planning, the Head of Operations reads product data specifically the Make to Stock quantities. After viewing sales insight, the Head of Operations is able to make decision making and determine the Make to Stock quantities both for the upcoming month for the two outlets and kitchen to produce and follow the quantity set by them. The Head of Operations also determines the production step sequence and chooses the sequence to produce the products from packaging and preparation of the goods.

The components involved in the multilayer are as followed:

- 5 UI Frame: Dashboard UI, Production Planning Step 1 UI, Production Planning Step 2 UI, Production Planning Step 3 UI, Production Planning Step 4 UI
- 1 Controller: ProductionPlanningController
- 5 Model: OrderDetail, Order, Product, ProductionStep, Cookie
- 5 Data Access: ProductDA, OrderDA, OrderDetailDA, ProductionStepDA, ProductionStepDA

2. Create Purchase Orders (Monday / Thursday)

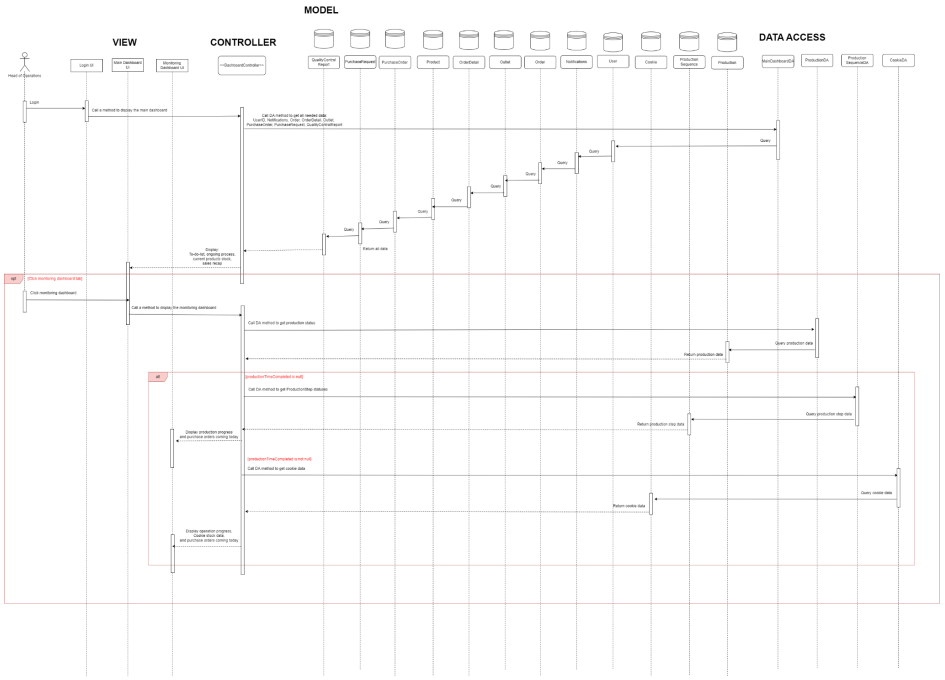


In the multilayer diagram of the Create Purchase Orders (Monday/Thursday), the Purchasing creates a purchase order every Monday and Thursday where they select the material and the quantity necessary to be restocked and supplied. In addition, with selecting the material(s) to be reported on the Purchase Order, the vendor details are accessed. By submitting a purchase order, the purchase order is saved onto the database and is viewed by the Purchasing.

The components involved in the multilayer are as followed:

- 3 UI Frame: Dashboard UI, Purchase Order Form UI, Purchase Order View UI
- 1 Controller: CreatePO(Mon/Thurs)Controller
- 10 Model: PurchaseOrderDetail, PurchaseOrder, PurchaseRequestDetail, PurchaseRequest, Material, Mixture, MaterialsUsed, MixturesUsed, Product, Vendor
- 4 Data Access: MaterialStockDA, VendorDA, PurchaseOrderDetailDA, PurchaseOrderDA

3. Head of Operations Dashboard



In the multilayer diagram of the Head of Operations Dashboard, the Head of Operations views from various models to see the upcoming tasks they must perform as well as see the incoming supply of inventory that will be delivered on that day. They are also able to monitor the current stock of the product as well as view sales analytics from order history made by the outlet. Additionally, they are able to access and view the Monitoring Dashboard UI where they are able to monitor the production sequence that was set from Production Planning and see the real-time status of each sequence.

The components involved in the multilayer are as followed:

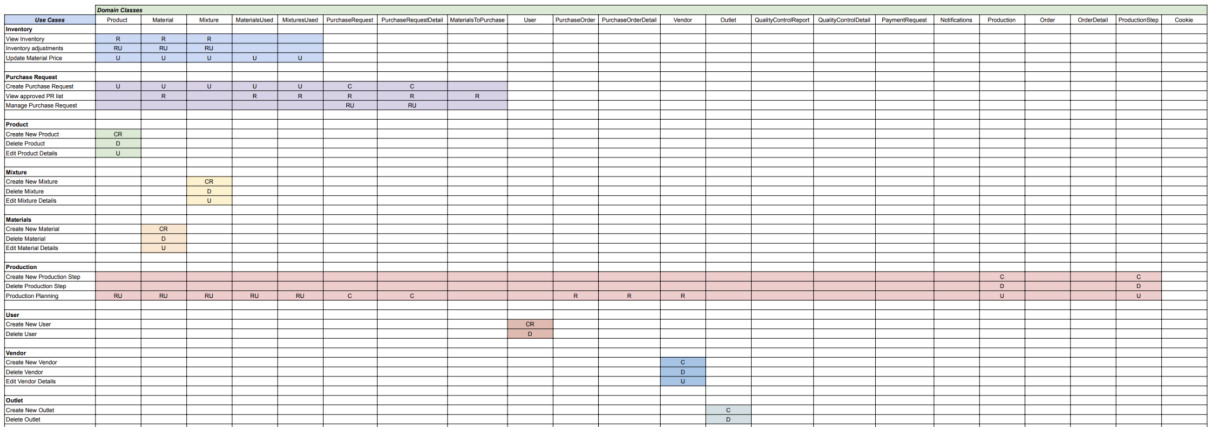
- 3 UI Frame: Login UI, Main Dashboard UI, Monitoring Dashboard UI
- 1 Controller: DashboardController

- 12 Model: QualityControlReport, PurchaseRequest, PurchaseOrder, Product, OrderDetail, Outlet, Order, Notifications, User, Cookie, ProductionSequence, Production
- 4 Data Access: MaterialStockDA, VendorDA, PurchaseOrderDetailDA, PurchaseOrderDA

5.10. CRUD Matrix

The figure below is a CRUD matrix model of our system design. The matrix is modelled for our 40 use cases and its relationship with our domain classes.

Link for matrix model:
https://docs.google.com/spreadsheets/d/12TbPG9JOHz3p6FYalHlAtd8PnHKT1P7MaF9eVnhvGs4/edit?usp=sharing



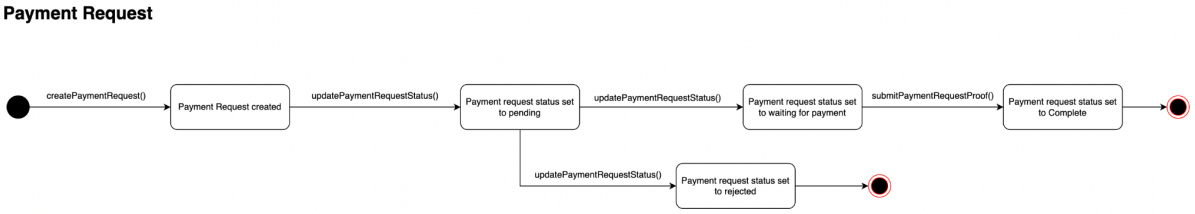
Use Cases	Domain Classes																					
	Product	Material	Mixture	MaterialsUsed	MixturesUsed	PurchaseRequest	PurchaseRequestDetail	MaterialsToPurchase	User	PurchaseOrder	PurchaseOrderDetail	Vendor	Outlet	QualityControlReport	QualityControlDetail	PaymentRequest	Notifications	Production	Order	OrderDetail	ProductionStep	Cookie
Inventory																						
View Inventory	R	R	R																			
Inventory adjustments	RU	RU	RU																			
Update Material Price	U	U	U	U	U																	
Purchase Request																						
Create Purchase Request	U	U	U	U	U	C	C															
View approved PR list		R		R	R	R	R	R														
Manage Purchase Request						RU	RU															
Product																						
Create New Product	CR																					
Delete Product	D																					
Edit Product Details	U																					
Mixture																						
Create New Mixture			CR																			
Delete Mixture			D																			
Edit Mixture Details			U																			
Materials																						
Create New Material		CR																				
Delete Material		D																				
Edit Material Details		U																				
Production																						
Create New Production Step																		C			C	
Delete Production Step																		D			D	
Production Planning	RU	RU	RU	RU	RU	C	C			R	R	R						U			U	
User																						
Create New User									CR													
Delete User									D													
Vendor																						
Create New Vendor												C										
Delete Vendor												D										
Edit Vendor Details												U										
Outlet																						
Create New Outlet													C									
Delete Outlet													D									
Quality Control																						
Create QC Report														C	C							
Manage Quality Control														R	R							
Payment Request																						
Manage Payment Request									R							RU						
Submit Payment Request									R							C						
Delivery																						
Delivery																		RU				
Driver Delivery Tracking																		U				
HoO Dashboard																						
HoO Dashboard	R					R			R	R			R				R		R	R		
Purchase Orders																						
Create purchase order (Approved PR)		RU								C	C			RU	RU		R					
Create purchase order (Mon & Thurs)		RU								C	C			RU	RU							
Manage Purchase Order										RU												
View Purchase Order										R	R											
Daily																						
Daily Operation																		RU				U
Daily Production	RU																	C			U	
Bill of Material																						
View BOM	R	R	R	R	R																	
Order																						
View Order History	R												R						R	R		

5.10. State Transition Diagram

1. Production

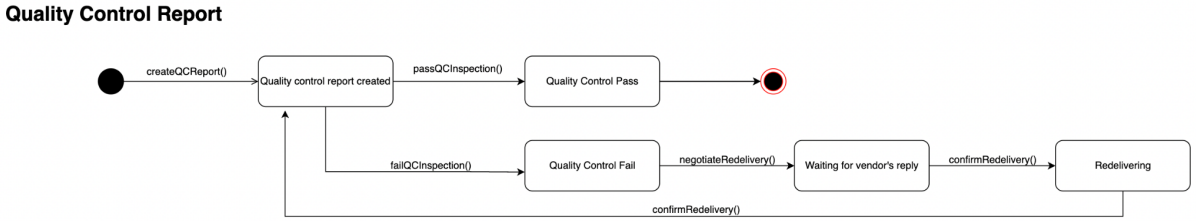
The state transition diagram represents the different states that a production process goes through, starting with "startProduction." The states include "Not started," "On-progress," "Completed," "On-the-way to PIK," "On-the-way to Pakubuwono," "Completed Delivery," "On-going Ice Cream Preparation," "On-going Cookies Preparation," and "Morning Preparation Completed." The transitions between these states depict the progression of the production process, providing a visual representation of the various stages and statuses involved.

2. Payment Request



The state transition diagram for the payment request starts with the creation of a payment request using the "createPaymentRequest()" action. The payment request then transitions to the "created" state. From there, the payment request status can be updated to "pending" or "rejected." If it is set to "pending," it can further transition to "waiting for payment" upon submission of the payment request proof. Finally, when the payment is completed, the status is set to "complete."

3. Quality Control Report

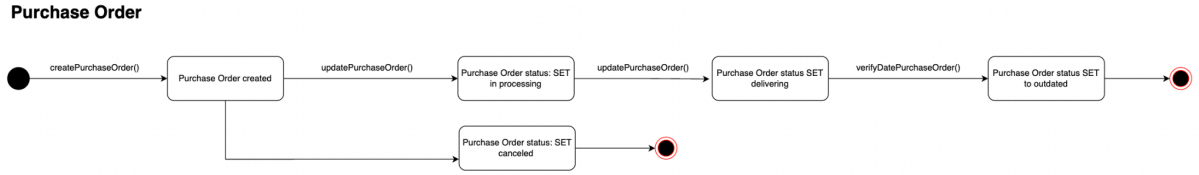


The state transition diagram for the quality control report begins with the action "createQCReport()" which transitions the report to the "created" state. From there, the report can undergo a quality control inspection, leading to two possible outcomes. If the inspection passes, the report transitions to the state "quality control pass." However, if the inspection fails, the report moves to the state "QC Fail" and enters a negotiation process with the vendor, represented by the transitions "negotiate redelivery" and "status set as waiting for vendor's reply." If a redelivery is confirmed, the status changes to "redelivering," and once again, it goes back to the "created" state.

4. Purchase Request

The state transition diagram for the Purchase Request begins with the action "createPurchaseRequest()" which transitions the request to the "created" state. From there, the request can be reviewed and approved, leading to the status being set to "pending" in the "approvePurchaseRequest" transition. The request can then be updated, with the status changing to "urgent" in the "update purchase request" transition. Further updates can be made, resulting in the status being set to "outdated" in the "updatePurchaseRequest" transition.

5. Purchase Order



The state transition diagram for the Purchase Order begins with the action "createPurchaseOrder," which transitions the order to the "created" state. From there, the order can be updated, with the option to set it as "in processing" or "cancelled." If set as "in processing," the order can be further updated to the "delivering" state. At any point, the order can undergo a verification process, resulting in the order being set to "outdated" in the "verifyDatePurchaseOrder" transition. This state transition diagram represents the various states and possible transitions for a Purchase Order.

5.11. SQL Statement

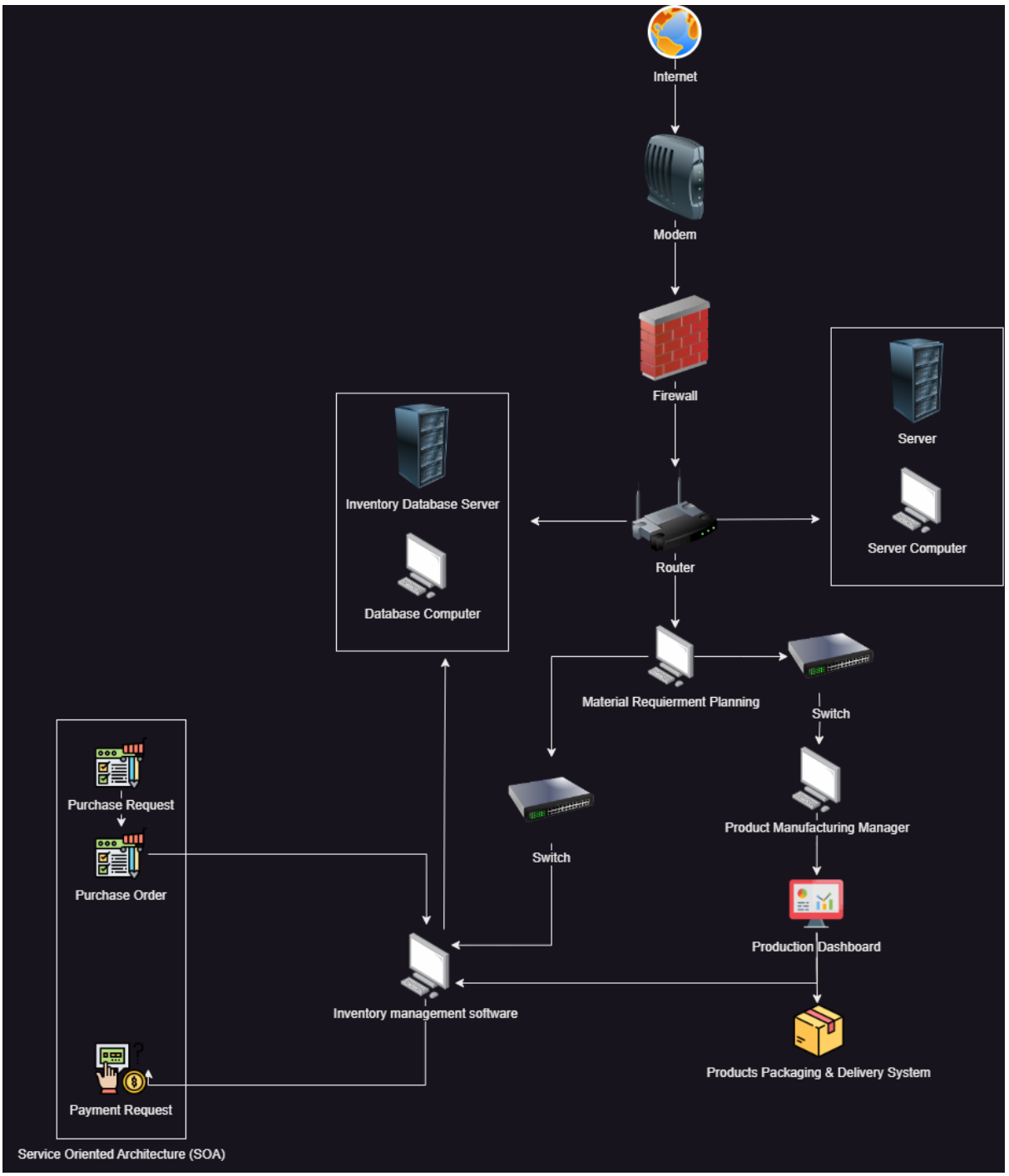
Within the programming scope defined earlier from End-to-End diagram, below are the SQL statements used in the DataAccess class where SELECT and UPDATE are used in the SQL statement to retrieve and modify data from the database.

Method	SQL Statement
getLoginDetails()	SELECT * FROM userdb WHERE Email=? AND Password=?
getProductData(String productName)	SELECT * FROM product WHERE productName='"+productName+"'
updateProductionStep(String firstPS, String secondPS, String thirdPS, String fourthPS, String fifthPS, String sixthPS, String seventhPS, String eighthPS)	UPDATE productionStep " + "SET stepName = CASE " + " WHEN stepID = 1 THEN '" + firstPS + "' " + " WHEN stepID = 2 THEN '" + secondPS + "' " + " WHEN stepID = 3 THEN '" + thirdPS + "' " + " WHEN stepID = 4 THEN '" + fourthPS + "' " + " WHEN stepID = 5 THEN '" + fifthPS + "' " + " WHEN stepID = 6 THEN '" + sixthPS + "' " + " WHEN stepID = 7 THEN '" + seventhPS + "' " +

	" WHEN stepID = 8 THEN '" + eighthPS + "' " + "END, " + "productCategory = CASE " + " WHEN stepName = '" + firstPS + "' THEN productCategory " + " WHEN stepName = '" + secondPS + "' THEN productCategory " + " WHEN stepName = '" + thirdPS + "' THEN productCategory " + " WHEN stepName = '" + fourthPS + "' THEN productCategory " + " WHEN stepName = '" + fifthPS + "' THEN productCategory " + " WHEN stepName = '" + sixthPS + "' THEN productCategory " + " WHEN stepName = '" + seventhPS + "' THEN productCategory " + " WHEN stepName = '" + eighthPS + "' THEN productCategory " + "END, " + "stepType = CASE " + " WHEN stepName = '" + firstPS + "' THEN stepType " + " WHEN stepName = '" + secondPS + "' THEN stepType " + " WHEN stepName = '" + thirdPS + "' THEN stepType " + " WHEN stepName = '" + fourthPS + "' THEN stepType " + " WHEN stepName = '" + fifthPS + "' THEN stepType " + " WHEN stepName = '" + sixthPS + "' THEN stepType " + " WHEN stepName = '" + seventhPS + "' THEN stepType " + " WHEN stepName = '" + eighthPS + "' THEN stepType " + "END
String getPurchaseOrderData()	SELECT * from purchaseorder
String getPurchaseRequestData()	SELECT * from purchaserequest
updateProductMTSQuantities(String productName, String mtsQtyWeekdays, String mtsQtyWeekends)	UPDATE product SET mtsQtyWeekdays = '" + mtsQtyWeekdays + "', mtsQtyWeekends = '" + mtsQtyWeekends + "' WHERE productName = '" + productName + "'

updateDeliveryQuantities(String productName, JTextField mtsWeekdaysPKBW, JTextField mtsWeekdaysPIK, JTextField mtsWeekendPKBW, JTextField mtsWeekendPIK)	UPDATE product SET pkbwDeliveryQtyWky = '" + mtsWeekdaysPKBW.getText() + "', pikDeliveryQtyWky = '" + mtsWeekdaysPIK.getText() + "', pkbwDeliveryQtyWkd = '" + mtsWeekendPKBW.getText() + "', pikDeliveryQtyWkd = '" + mtsWeekendPIK.getText() + "' WHERE productName = '" + productName + "'
updatePIKPreparationDetails(String productName, JTextField MPWdy, JTextField RPWdy, JTextField MPWkd, JTextField RPWkd)	UPDATE product SET pikMorningPrepWky = '" + MPWdy.getText() + "', pikReBakingPointWkky = '" + RPWdy.getText() + "', pikMorningPrepWkd = '" + MPWkd.getText() + "', pikReBakingPointWkd = '" + RPWkd.getText() + "' WHERE productName = '" + productName + "'
updatePKBWPreparationDetails(String productName, JTextField MPWdy, JTextField RPWdy, JTextField MPWkd, JTextField RPWkd)	UPDATE product SET pkbwMorningPrepWky = '" + MPWdy.getText() + "', pkbwReBakingPointWkky = '" + RPWdy.getText() + "', pkbwMorningPrepWkd = '" + MPWkd.getText() + "', pkbwReBakingPointWkd = '" + RPWkd.getText() + "' WHERE productName = '" + productName + "'

5.12. Network Diagram



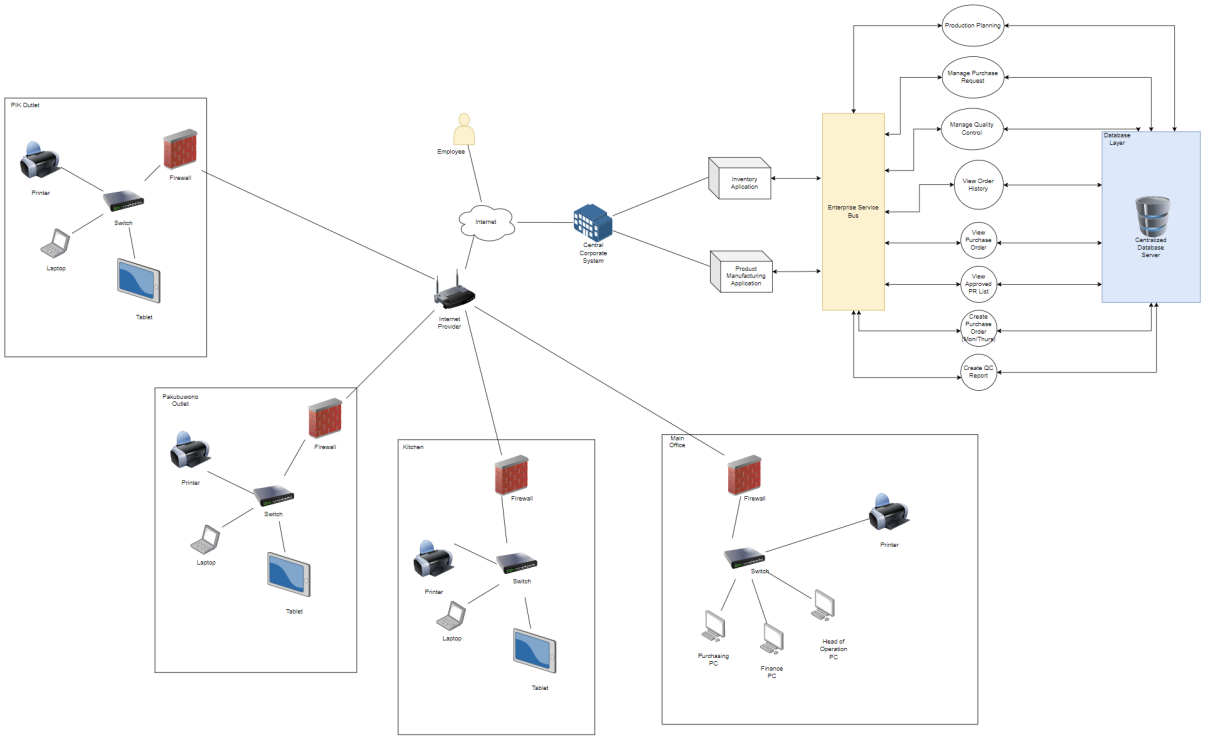
Here's the network diagram we came up with to illustrate the computer network architecture of our solution for Gotham Treats. At the heart of this infrastructure is the Internet, which is connected to a modem to provide a broadband connection. The firewall, which filters incoming and outgoing traffic, is then connected to the modem and plays an

important role in securing the network. The firewall is connected to a router, which acts as a central point for distributing traffic to various locations.

The router divides traffic into three different flows. A server and a server computer are connected by the first flow, creating a computer system specialised in query processing and data storage. The second flow is directed to a database server and a database computer, which host a central database used to manage the company's inventory. Finally, the third flow is connected to material requirements planning (MRP) software, which facilitates management of the resources needed for production.

Two switches connect the MRP software. The Head of Operations, who oversees production, is connected to the first switch. The production dashboard, which provides real-time information on production operations, is followed by the data flow to the packaging and product delivery system. Inventory management software is connected to the second switch to facilitate inventory management and monitoring. This data stream is also connected to the database server and computer, enabling real-time inventory updating and synchronisation.

In conclusion, the architecture depicted in this diagram allows for effective connectivity, efficient resource management, and real-time monitoring of production and inventory management procedures.



The network diagram we produced to show the Gotham Treats solution's computer network architecture can be seen above. The Internet is at the centre of this system. The Internet is next linked to the internet provider, which is then connected to multiple firewalls, which is crucial for protecting the network as it filters both incoming and outgoing data. The router, which the firewall is attached to, serves as the hub for delivering traffic to other sites. The firewalls serve as the door protection for 4 locations being the kitchen, main office, Pakubuwono outlet and PIK outlet. All employees are connected to the internet in this network.

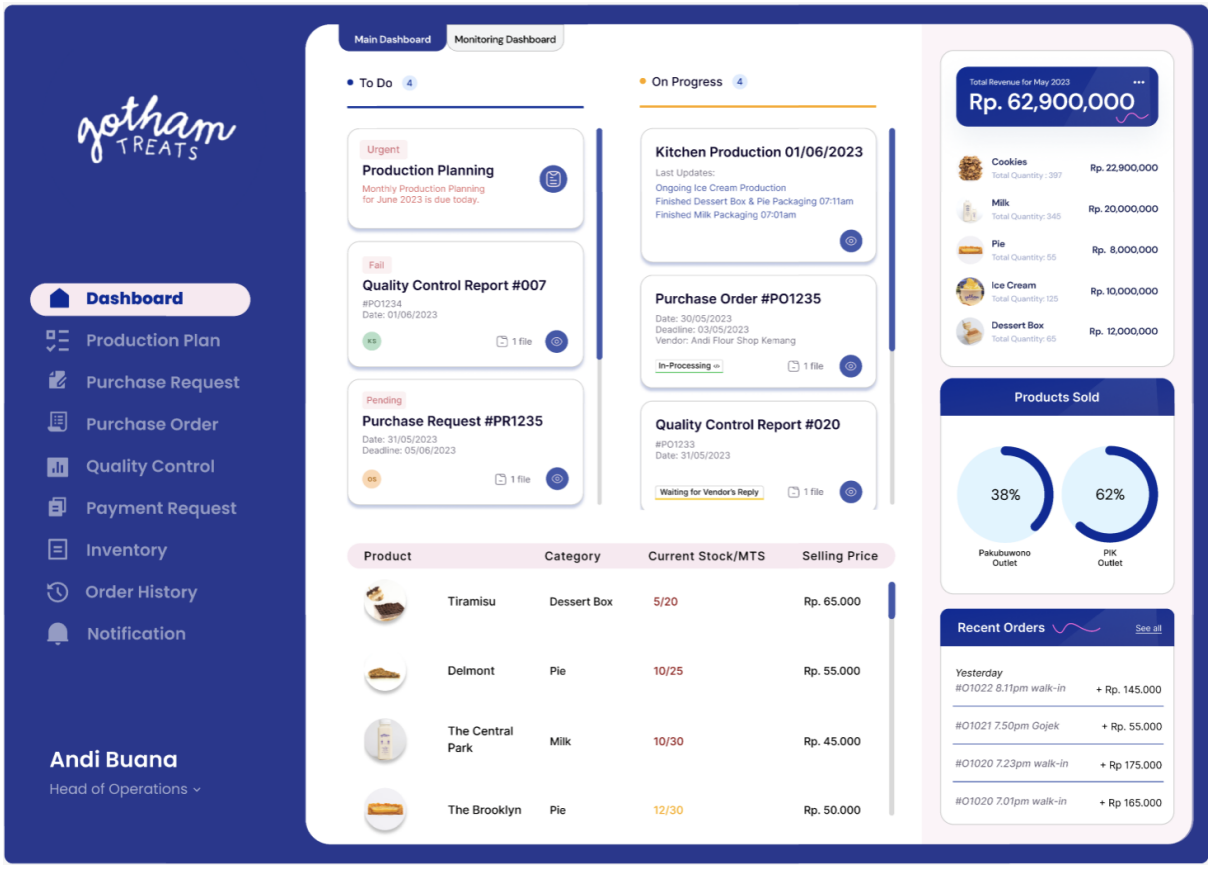
The internet serves the connection for the central corporate system containing two applications which are the inventory and product manufacturing applications. The applications are connected and able to communicate with each other through an enterprise service bus (ESB). The ESB allows for the use cases to be able to perform and modify data that will be saved in the databases at the centralised database server.

5.14. UI / UX

Link to our UI/UX Design:

https://www.figma.com/file/QQJYtozhYhxab7ALjDGg66/Gotham-Treats-UI%2FUX?type=design&node-id=0%3A1&t=cnkEorg5dULDwWcA-1

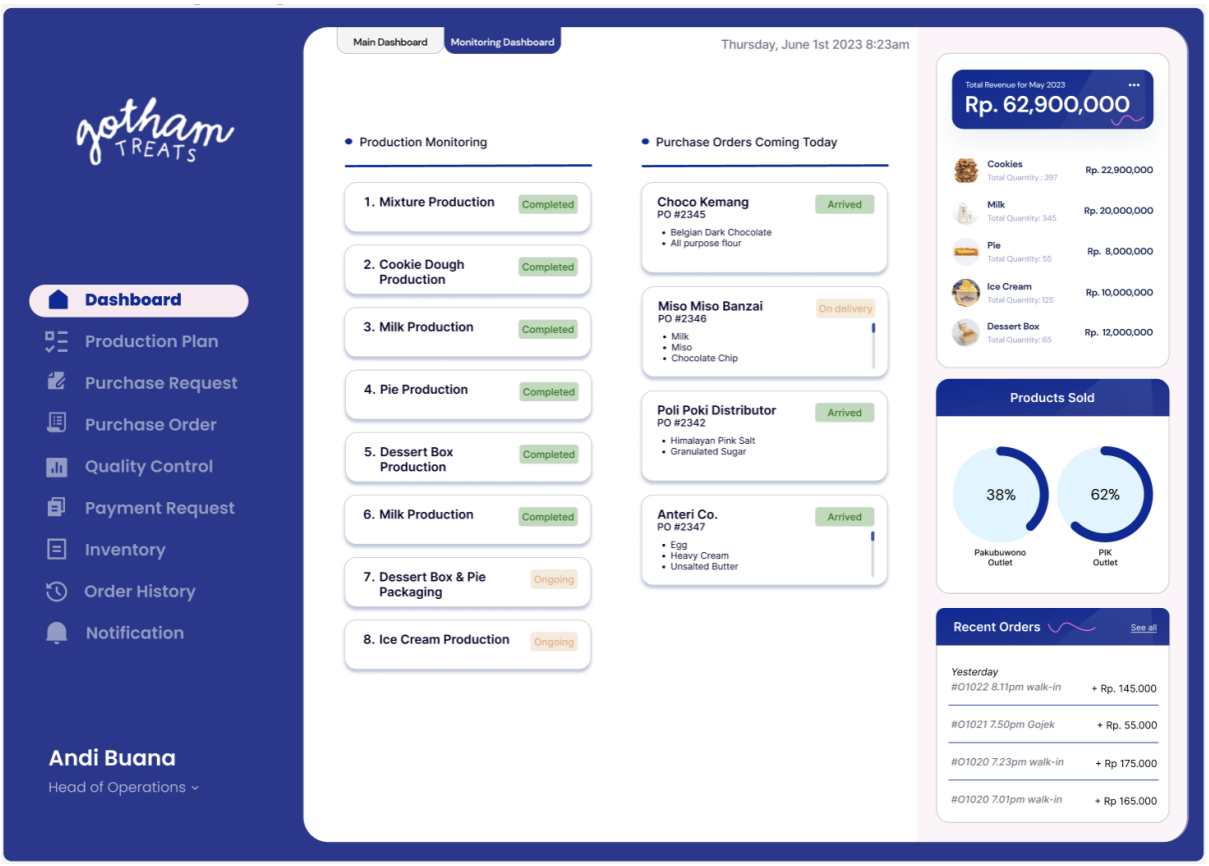
The Head of Operations Main Dashboard



Information Provided:

To do list, On progress processes, Sales recap, Recent orders, and current products stock.

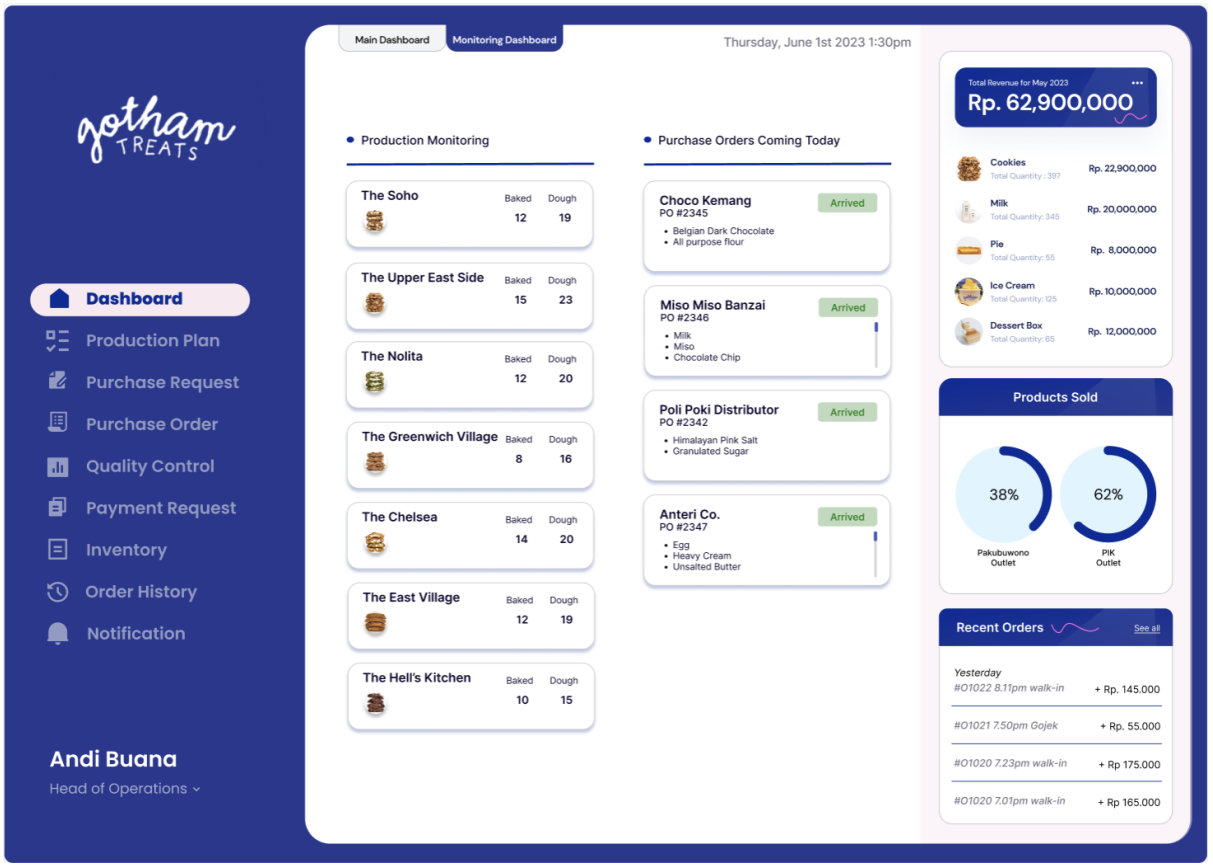
The Head of Operations Monitoring Dashboard (Morning During Production)



Information Provided:

Production progress, purchase orders coming today

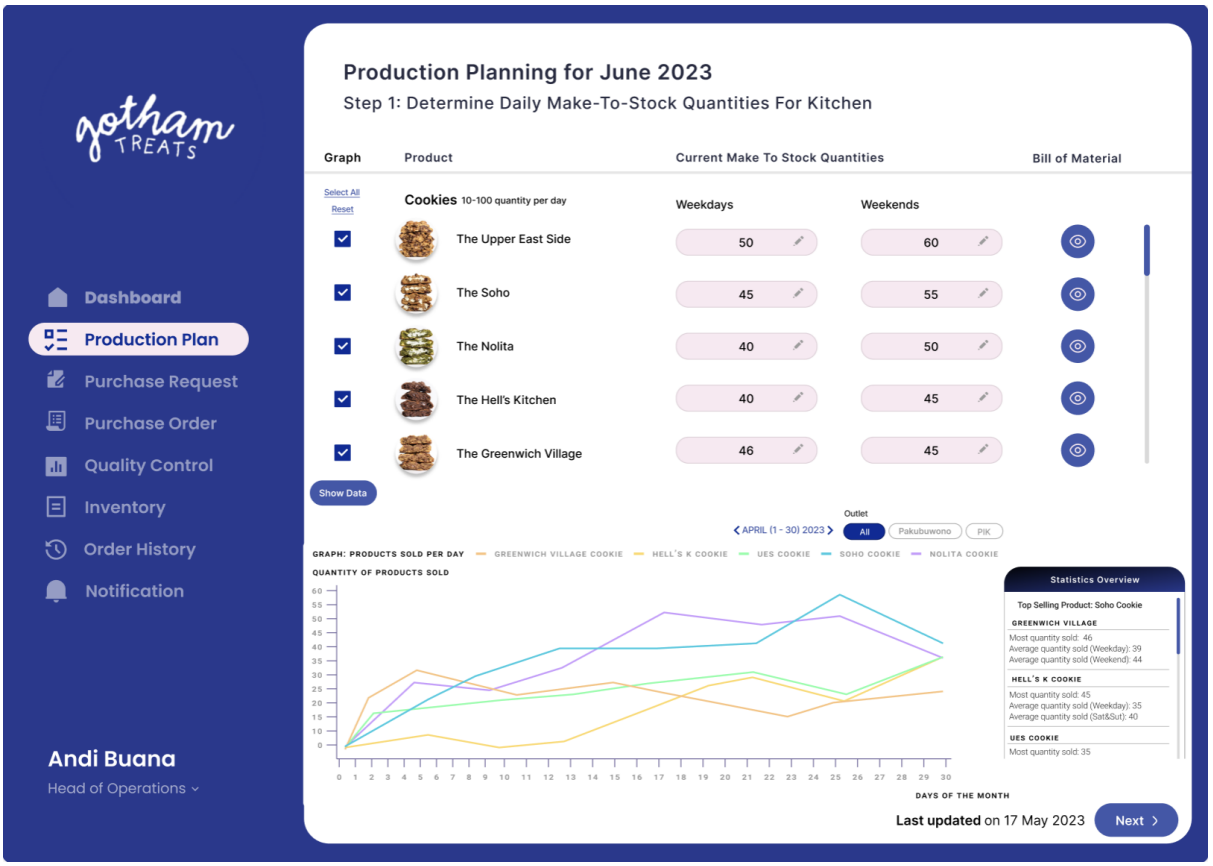
The Head of Operations Monitoring Dashboard (After Production)



Information Provided:

Cookies baking progress, purchase orders coming today

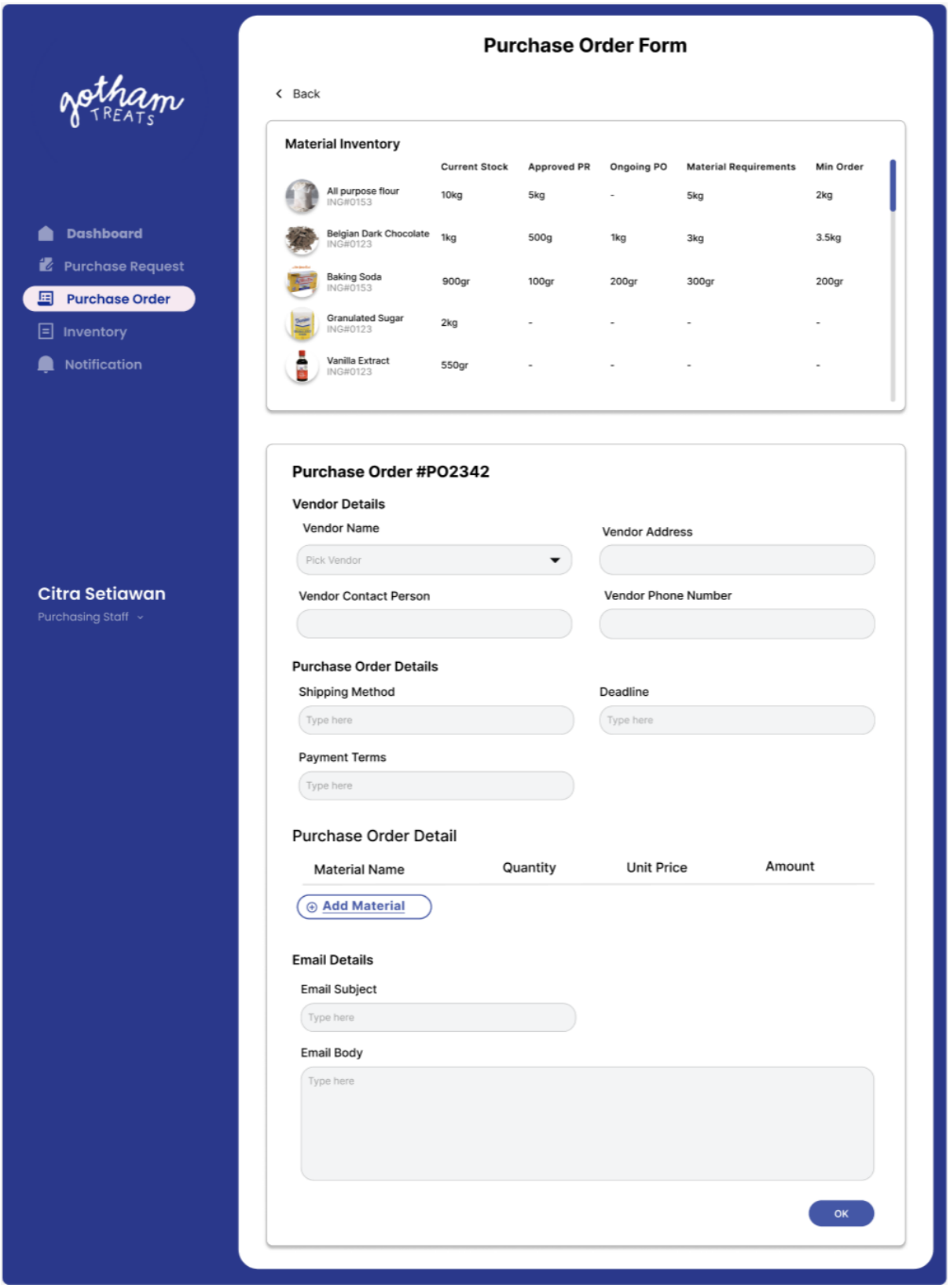
The Production Planning (Step 1)



Description:

The production planning menu for the Head of Operations. The menu displays the currently saved make-to-stock quantities for each product. Additionally, there is statistical data of sales that has been processed into a graph for the HoO to have data to make informed decisions on adjusting the MTS quantities and more for the continuing steps in this production plan.

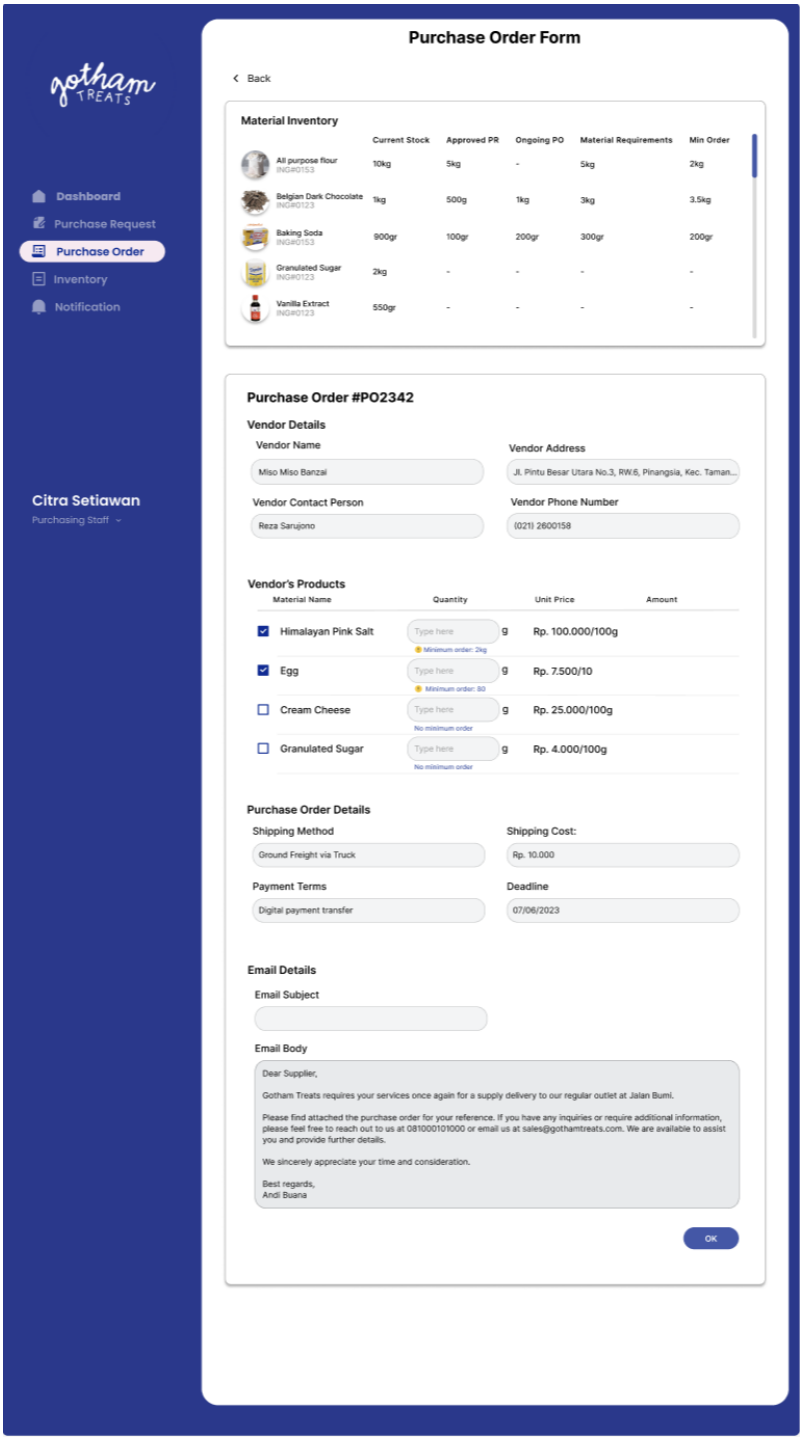
The Purchase Order Form (Blank)



Description:

The purchasing staff fills a purchase order.

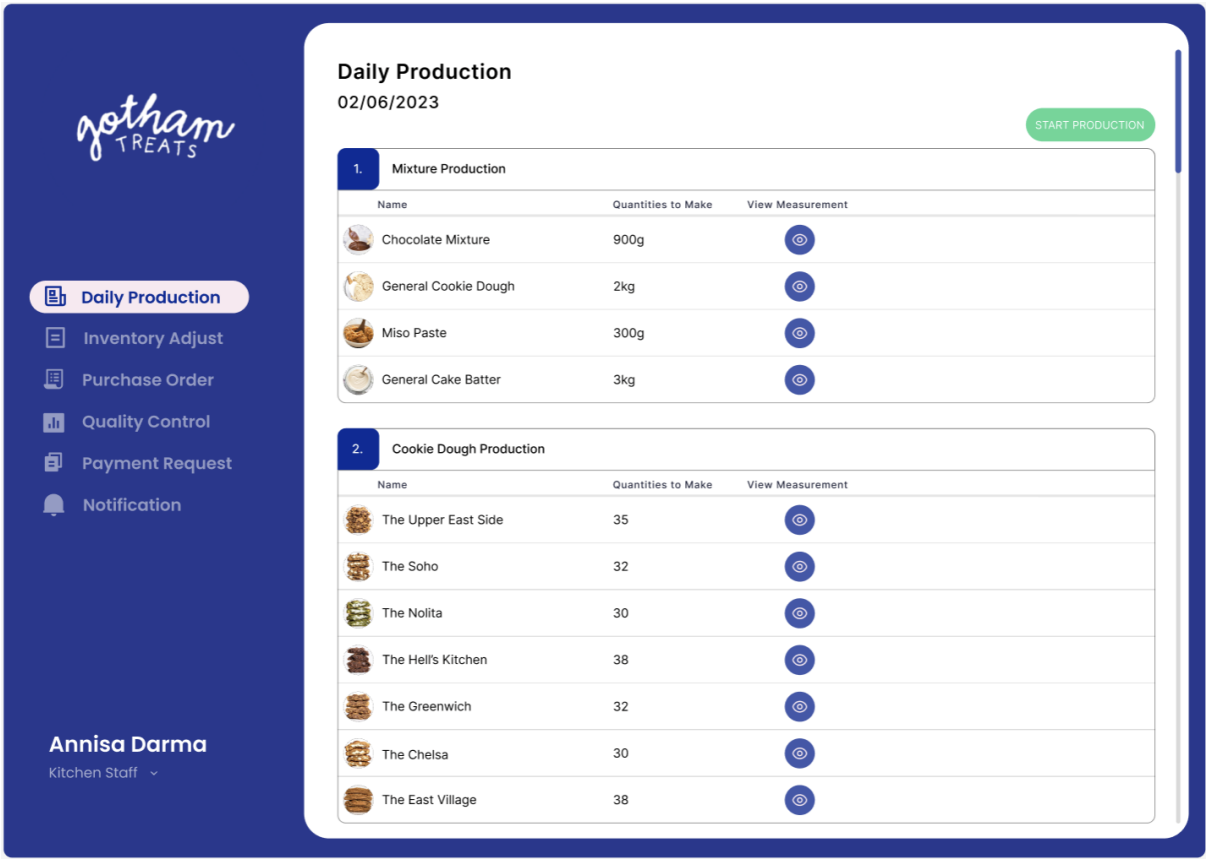
The Purchase Order Form (Filled)



Description:

The purchasing staff has filled most of the purchase order form. The form shows the minimum order of each selected material based on the material requirements.

Daily Production



Information Provided:

Daily production for the kitchen to execute. The data in this page is generated according to the production plan.

Chapter VI: Closing

6.1. Conclusion

To conclude, our final project suggests implementing a digitised inventory management system for Gotham Treats, addressing their challenges and streamlining their operations. Through comprehensive company analysis, we gained insights into their specific requirements, enabling us to design a tailored solution that automated inventory tracking, monitoring, and procurement processes.

By understanding the needs of the actors involved, we developed a system that provided real-time inventory information, automated reordering based on predefined thresholds, and valuable sales insights. The utilisation of UML diagrams ensured a clear understanding of the system's structure and functionalities, guiding the development process and ensuring alignment with Gotham Treats' operational needs.

The implemented solution significantly improved operational efficiency, reduced errors, and enhanced inventory management for Gotham Treats. Manual tracking and inaccurate stock levels became a thing of the past, while sales insights enabled data-driven decision-making. With this digitised inventory management system in place, Gotham Treats is now better positioned to meet customer demands, optimise their offerings, and drive business growth in the competitive dessert market.