COS20031

Computing Technology Design Project

Request for Proposal (RFP) for Design and Development of a Direct-to-Consumer ECommerce Platform

Instructor: Dr. Hai Sam Nang

Project Team Member:

- Nguyen Thuan Khang (104171078)
- Doan Nhat Long (104167705)
- Le Ba Tung (104175915)
- Lau Ngoc Quyen (104198996)

Table of Contents

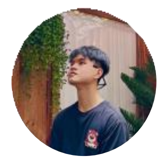
Team Home

Welcome to QLTK

"Teamwork makes the dream work." – John C. Maxwell

✨ About QLTK ✨
Our team is focused on launching a direct-to-consumer e-commerce platform for Trung Nguyen Coffee Corp. Key initiatives include streamlining the customer journey, offering personalized shopping experiences, and optimizing logistics. We aim to increase market reach, strengthen brand loyalty, and track metrics such as customer retention, sales growth, and user engagement.

Team metrics
Join Atlassian Updated Sep 9, 2024 ON TRACK
Team meeting Updated Sep 16, 2024 ON TRACK
Project Milestones Achieved Updated Sep 19, 2024 ON TRACK
Finalize Project Proposal Updated Sep 28, 2024 ON TRACK

About us	
 **Lau Ngoc Quyen** *Project Manager, Scrum Master*	 **Nguyen Thuan Khang** *Quality Manager*
 **Le Ba Tung** *Developer*	 **Doan Nhat Long** *Resource Manager*

Latest updates	Where to find us	Feature resources

Complete the ERD diagram and reorganize the project proposal structure. Schedule a final team meeting for the latest update.	**Facebook**: Nguyễn Thuận Khang (Team Representative) **Contact Mail:** khangntsws00434@fpt.edu.vn **Contact Number:** 0354089422	*https://github.com/quynezz* *https://www.facebook.com/profile.php?id=100022160617179* *https://www.instagram.com/_andrew204_/*

Our calendar

Latest updates	Creating a Project Proposal	Accomplished Project Milestones
1. Review team members' onboarding progress. *(6/9/2024)* 2. Schedule introductions and any meeting sessions needed for the week. *(8/9/2024)* 3. Hold a weekly team meeting to discuss current priorities, challenges, and goals. *(6/9/2024 - 29/9/2024)* 4. Share progress updates on the e-commerce platform launch. *(8/9/2024 - 29/9/2024)* 5. Address any roadblocks and align on the week's focus areas. *(8/9/2024)*	1. Create Team Home Page. *(9/9/2024)* 2. Create Project Plan Page. *(10/9/2024 - 11/9/2024)* 3. Create Roles and Responsibilities Page. *(12/9/2024)* 4. Create Risk Assessment and Persona Pages. *(13/9/2024) - (16/9/2024)* 5. Create Empathy Map Page. *(17/9/2024)* 6. The rest includes (Product requirements, Initial ER diagram, Appendix 1: Team member profiles, Appendix 2: Team meeting). *(18/9/2024 - 26/9/2024)*	1. Evaluate the completion status of tasks from the previous week. *(27/9/2024)* 2. Revise project milestones and confirm that deadlines are being met. *(27/9/2024)* 3. Reallocate tasks as necessary to achieve project objectives. *(28/9/2024)*

Project Plan

💡 Project Overview

Driver	Nguyen Thuan Khang
Approver	Dr. Hang Sam Nang
Contributors	Nguyen Thuan Khang Doan Nhat Long Lau Ngoc Quyen Le Ba Tung
Informed	Doan Nhat Long - Lau Ngoc Quyen - Le Ba Tung
Objective	The project aims to develop a Direct-to-Consumer (DTC) e-commerce platform for **Trung Nguyen Coffee Corp**. This platform will enable the company to expand its market presence, boost customer engagement, and strengthen brand loyalty by offering a seamless, personalized online shopping experience.
Due date	October 1, 2024
Key outcomes	1. **Improved Customer Engagement**: Directly interact with customers, offering personalized shopping experiences. 2. **Operational Efficiency**: Streamline ordering, fulfillment, and customer service processes. 3. **Sales Growth**: Increase sales by bypassing intermediaries and improving market reach. 4. **Customer Satisfaction (CSAT) Scores**: Enhance customer satisfaction through optimized user experiences.
Status	DONE

🤔 Problem Statement

- Trung Nguyen Coffee Corp faces limitations in its current retail strategy, relying heavily on intermediaries for distribution. This leads to a lack of direct engagement with consumers, limited market insights, and restricted brand loyalty opportunities. The absence of an e- commerce platform hampers its ability to scale operations and provide a personalized customer experience, which is increasingly important in today's digital economy.

🎯 Scope

Must have	• Fully functional e-commerce platform • Responsive design optimized for mobile devices • Multilingual and multicurrency support for international markets • Secure payment gateways and order tracking • Inventory management integration • Personalized product recommendations • Compliance with data privacy laws (GDPR, PCI DSS)
Nice to have	• Integration with loyalty programs and subscription services • Advanced analytics and reporting tools • Social media integration for seamless product sharing
Not in scope	• Physical store management • Offline marketing or sales channels • In-depth third-party marketplace integrations (e.g., Amazon, eBay)

🗓️ Timeline


2024
02-Sep
09-Sep
16-Sep
23-Sep
30-Sep
07-Oct
14-Oct
21-Oct
28-Oct
First Week
TeamHom...
ProjectPl...
RoleAndResponsibiliti...
Second Week
RiskAssessmen...
Perso...
Thrid W...
EmpathyMapP...
ProjectRequirement...
Fourth ...
ERDDiagram
Appendix1and2

🚩Milestones and deadlines

Milestone	Owner	Deadline	Status
Join Confluence and Jira	Everyone	Sep 6, 2024	**Complete**
Create Team Member Profiles	Everyone	Sep 9, 2024	**Complete**
Create Team Home Page	Ngoc Quyen	Sep 9, 2024	**Complete**
Create Project Plan Page	Nhat Long, Thuan Khang	Sep 10-11, 2024	**Complete**
Create Role and Responsibilities Page	Nhat Long, Ngoc Quyen	Sep 12, 2024	**Complete**
Create Risk Assesment Page	Ngoc Quyen, Ba Tung	Sep 14, 2024	**Complete**
Create Persona	Ba Tung, Thuan Khang, Ngoc Quyen	Sep 15-16, 2024	**Complete**
Create Product Requirement	Ngoc Quyen, Thuan Khang	Sep 17-18, 2024	**Complete**
Create Initial ERD Diagram	Ba Tung, Ngoc Quyen	Sep 20-22, 2024	**Complete**
Appendix 1 And Appendix 2	Nhat Long, Thuan Khang	Sep 23-28, 2024	**Complete**

🤝 Commitment Of Out Project Plan

<div>Together, We'll Make It Work</div>

1. Team Collaboration:

All team members commit to full collaboration and communication throughout the project. Each individual will actively participate in team meetings, provide timely updates, and ensure their assigned tasks are completed by the set deadlines.

2. Quality Assurance:

Our team is committed to delivering a high-quality e-commerce platform, adhering to industry best practices in development, security, and user experience design. Rigorous testing will be conducted during each phase to ensure the platform meets all requirements.

3. Customer Focus:

We are committed to creating a platform that delivers excellent value to Trung Nguyen Coffee Corp's customers. By focusing on usability, personalization, and a smooth shopping experience, the platform will engage consumers and foster long-term loyalty.

4. Timely Delivery:

Every effort will be made to meet or exceed the project timeline. The team will adhere to the project schedule, making adjustments as necessary to ensure milestones are reached on time, without compromising on quality.

5. Transparency and Accountability:

We commit to maintaining full transparency throughout the project lifecycle. Regular reports will be provided on the status of deliverables, and any risks or delays will be communicated promptly to the project stakeholders.

6. Post-launch Support:

After the successful launch of the platform, our team will provide ongoing support, including addressing any issues, updating features, and training internal teams to manage the system effectively.

Persona
Persona 1: Office Manager

Persona name	**Nguyen Van A**
Persona role	Office Manager
Job description	Responsible for overseeing office operations, managing team performance, and ensuring resources like coffee supplies are always available for employees.

🏢 Company

Company name	**L&P Tech Solutions**
Company size	10.000+ employees
Industry	IT/Software

👥 Demographic information

Age	35
Gender	Male
Income	$150,000+/year
Education level	Bachelor’s in Business Administration
Residential environment	Urban apartment in Ho Chi Minh City

🖋 Personal quote

"I need things to be efficient and reliable, just like my coffee."

Biography

A has been working as an office manager for 10 years. He enjoys being organized and ensuring his team has the best environment to work in. A coffee enthusiast, A drinks several cups a day and ensures the office is always stocked with good-quality coffee.

Professional goals	Motivators
• Streamline office supply management. Maximize employee satisfaction. • Balance quality and cost when purchasing supplies.	• Convenience of purchasing coffee online. Fast and reliable delivery to the office. • Bulk purchase discounts or subscription options.
Challenges	**Sources of information**
• Balancing cost and quality when ordering coffee in bulk. • Dealing with slow or unreliable deliveries.	• Office supply websites Customer reviews • Social media coffee groups

Persona 2: Marketing Specialist

🏢 Company

Persona name	**Le Thi Minh Trang**
Persona role	Marketing Specialist
Job description	Creates and executes marketing strategies, focuses on brand awareness, and loves coffee as a personal hobby.
Company name	XYZ Travel Agency
Company size	50+ employees
Industry	Tourism & Hospitality

👥 Demographic information

Age	28
Gender	Female
Income	$40,000 - $60,000/year
Education level	Bachelor's in Marketing
Residential environment	Downtown apartment in Hanoi

🖋 Personal quote

"I love discovering new coffee flavors from around the world."

📜 Biography

Minh Trang is passionate about traveling and coffee culture. She enjoys trying different coffee varieties and often shares her experiences on social media. She values sustainability and prefers brands that offer transparency about their sourcing and production.

<table>
<tr><th>Professional goals</th><th>Motivators</th></tr>
<tr><td>Boost her company’s brand visibility.
Create engaging content for coffee and lifestyle lovers.</td><td>Unique coffee products with a story.
Detailed product descriptions about origin and flavor.
Easy-to-navigate, visually appealing e-commerce platforms.</td></tr>
<tr><th>Challenges</th><th>Sources of information</th></tr>
<tr><td>Finding rare coffee blends.
Navigating between multiple websites for different coffee brands.</td><td>Instagram influencers Coffee review blogs
Online forums about coffee brewing</td></tr>
</table>

Persona 3: Entrepreneur

🏢 Company

Persona name	**Nguyen Hung Quan**
Persona role	Entrepreneur
Job description	Owns a chain of high-end restaurants and frequently hosts business meetings, where premium coffee is always served.
Company name	Quan's Bistro Group
Company size	1000+ employees
Industry	Hospitality

👥 Demographic information

Age	45
Gender	Male
Income	$50,000+/year
Education level	Bachelor’s in Business Administration
Residential environment	Urban apartment in Ho Chi Minh City

🖋 Personal quote

"Only the best coffee for my clients and partners."

📜 Biography

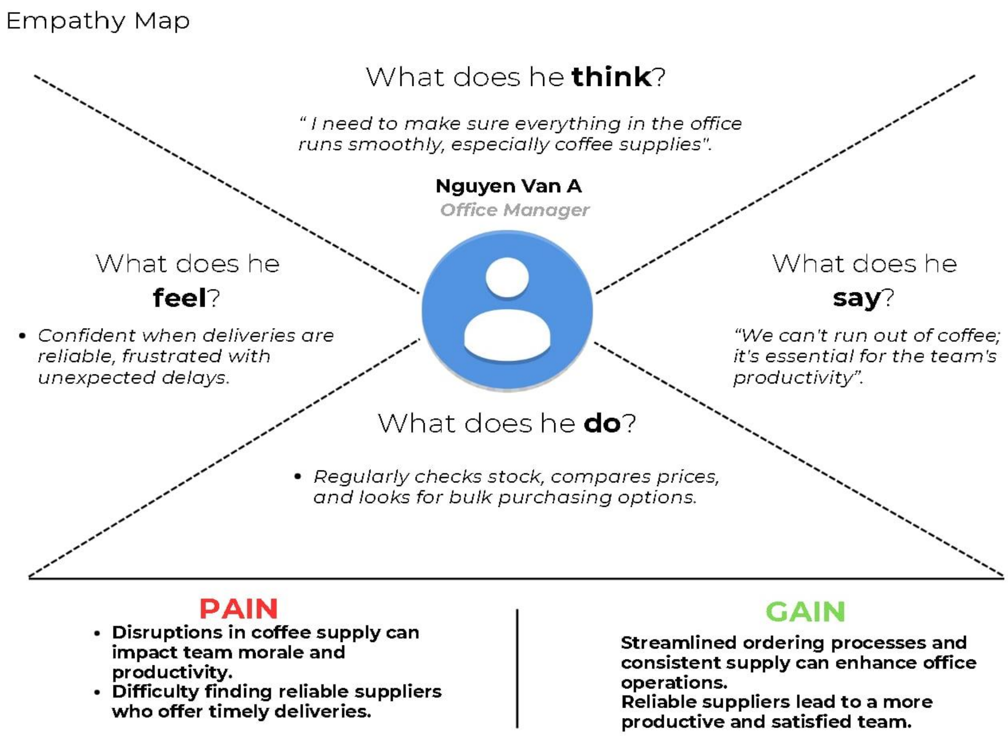
Quan is a successful restaurateur who understands the importance of quality, especially when it comes to the dining experience. He frequently purchases premium coffee not only for his restaurants but also for personal use during business meetings with high-profile clients.

Professional goals	Motivators
• Continue growing her freelance business. • Develop strong relationships with local clients.	• Affordability without sacrificing quality. • Convenient delivery to her home or favorite cafes. • Subscription services for her favorite blends.

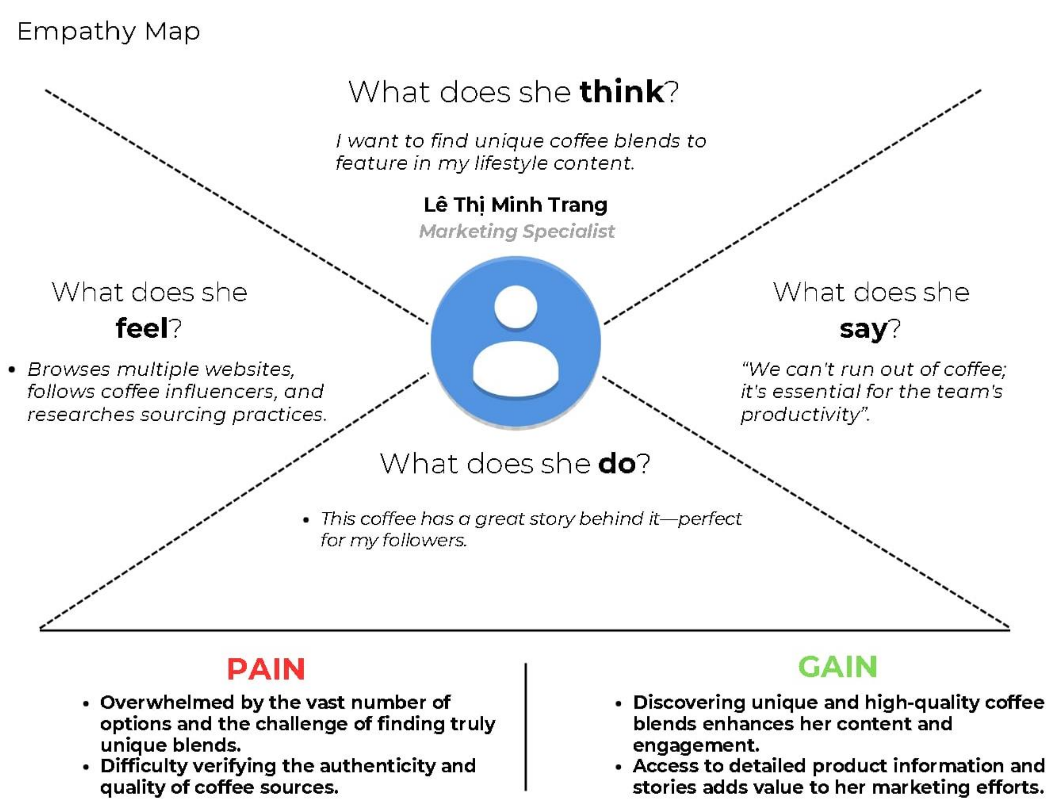
Challenges	Sources of information
• Keeping her budget in check while enjoying premium coffee. • Finding time to research different coffee options.	• Instagram and Pinterest • Design and lifestyle blogs • Coffee subscription boxes

Empathy map

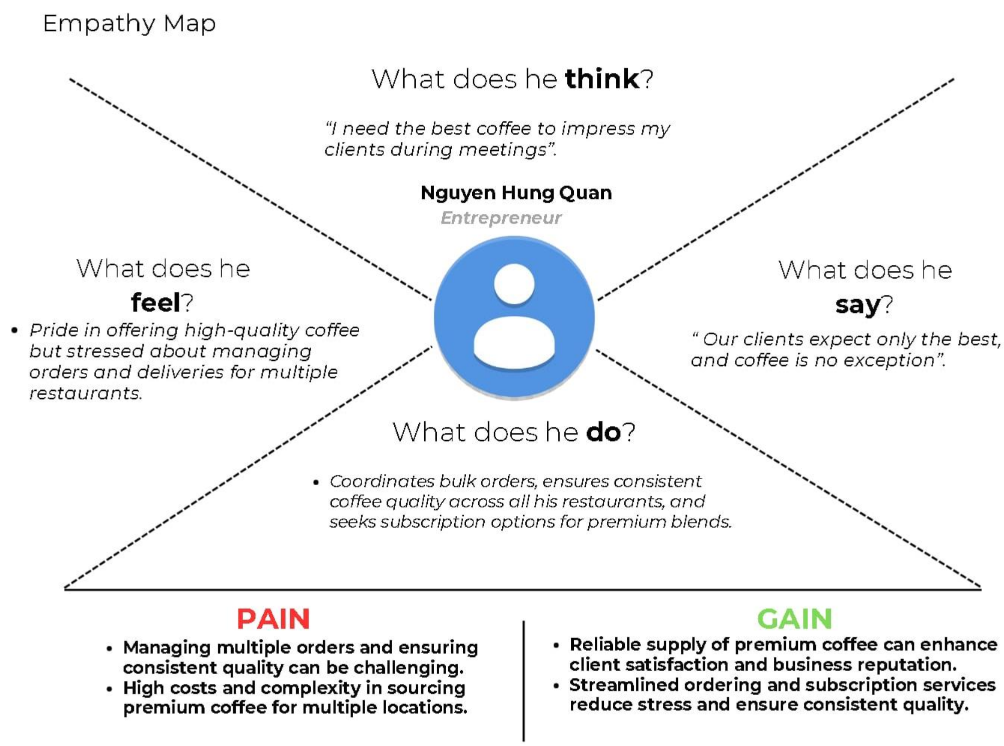
1. Office Manager



2. Marketing Specialist



3. Entrepreneur



Role and Responsibilities

📘 Role and Responsibilities

Roles	Responsibilities (what others think)	Responsibilities (what I think)	Key Performance Indicators (KPIs)	Assigned person
Resource Manager	• **Utilization Optimization:** Ensure optimal use of resources and balance workloads. • **Collaboration:** Work with project managers and stakeholders on resource needs. • **Change Management:** Ensure smooth adjustments to resource allocation without disrupting project progress. • **Monitor Utilization Rates:** Track how effectively resources are being used across projects. • **Tracking Resource Metrics:** Develop and maintain dashboards or reports to track key resource metrics, such as utilization, availability, and project progress.	• **Monitor Utilization Rates:** Track how effectively resources are being used across projects to ensure optimal performance. • **Tracking Resource Metrics:** Develop and maintain dashboards or reports to track key resource metrics, such as utilization, availability, and project progress.	• Resource Utilization Rate > 85% • Timely allocation of resources for 100% of project phases • Reporting accuracy on resource metrics	**Doan Nhat Long**

Developer	• **Bug Fixes and Patches**: Address reported issues and apply patches to fix bugs or security vulnerabilities. • **Testing and Debugging**: Perform unit testing, identify bugs, and resolve issues to ensure the software functions correctly and is free of defects.	• **Database Management**: Design and implement databases, write queries, and ensure proper data management and retrieval. • **Collaboration Tools**: Use tools like **JIRA**, **Trello**, or **Asana** to track project progress, bugs, and feature requests.	• Bug Fixing Rate within < 48 hours • 100% adherence to coding standards • Completion of assigned tasks within sprint deadlines	**Le Ba Tung**
Project Manager, Scrum Master	• **Define Project Scope**: Identify project goals, deliverables, and requirements by working with stakeholders and team members. • **Assemble and Lead the Team:** Select team members, assign roles, and guide the team toward project goals. • **Track Progress:** Use tools like **Cofluence**, or **JIRA** to monitor task completion, manage timelines, and ensure deadlines are met.	• **Develop Project Plan:** Create a detailed project plan outlining timelines, milestones, deliverables, and resources required. • **Motivate and Support Team:** Keep team members motivated and provide the necessary support to ensure successful project execution. • **Monitor Performance:** Track individual and team performance, providing feedback, and resolving any issues to keep the project on track.	• Milestone Achievement Rate > 90% • Budget variance < 5% • Stakeholder satisfaction score > 80% • Sprint Velocity Improvement Rate > 5% per sprint • 100% on-time completion of Scrum events • Team satisfaction score > 80%	**Lau Ngoc Quyen**

Quality Manager	• **Ensure Quality Standards**: Oversee the project's adherence to quality standards by conducting regular reviews and evaluations. • **Oversee QA Processes**: Supervise the quality assurance process, including audits, inspections, and testing to verify product or service quality.	• **Inspection and Testing**: Oversee regular inspections and testing of products, processes, and services to detect issues. • **Code Review and Testing**: Ensure that development teams conduct code reviews, unit tests, integration tests, and user acceptance testing (UAT) to identify bugs and performance issues early in the development cycle.	• Defect detection rate > 95% • 100% code coverage for unit tests • Compliance with quality standards in 100% of deliverables	**Nguyen Thuan Khang**

Risk Assessment

📋 Background

Trung Nguyen Coffee Corp, a prominent coffee brand in Vietnam, is embarking on a significant digital transformation through the launch of a direct-to-consumer (DTC) e-commerce platform. This strategic initiative is designed to eliminate conventional retail middlemen, enabling the company to establish direct connections with consumers on a domestic and international scale. Through offering a tailored online shopping experience, the company aims to enhance consumer engagement, expand its market reach, and cultivate greater brand loyalty. The project signifies a deliberate effort to leverage evolving consumer preferences towards online shopping, all the while optimizing operations and driving profitability.

Risk management

RISK RATING			
LOW	**MEDIUM**	**HIGH**	**EXTREME**
• Acceple • Ok to proceed	• As low as reasonably practicable • Take mitigation efforts	• Generally unacceptable • Seek support	• Intolerable • Place event on hold

Likelihood vs. Severity				
LIKELIHOOD	**SEVERITY**			
	ACCEPTABLE *Little to no effect on event*	**TOLERABLE** *Effects are felt, but not critical to outcome*	**UNDESIRABLE** *Serious impact to the course of action and outcome*	**INTOLERABLE** *Could result in disaster*
IMPROBABLE *Risk is unlikely to occur*		Resource shortages		
POSSIBLE *Risk will likely occur*	Miscommunication among team members	Market Acceptance	Quality issues in developed features	

PROBABLE *Risk will occur*			Delayed delivery of platform components	Integration issues with existing systems

Identified Risks

Risk	Likelihood	Severity	Risk Rating	Mitigation Actions	Responsibility Party
Resource shortages	POSSIBLE	TOLERABLE	**MEDIUM**	Maintain a buffer of resources and cross-train team members	Resource Manager
Delayed delivery of platform components	PROBABLE	UNDESIRABLE	**HIGH**	Develop a contingency plan with alternate suppliers.	Project Manager
Quality issues in developed features	POSSIBLE	UNDESIRABLE	**MEDIUM**	Implement thorough QA processes and code reviews.	Quality Manager
Miscommuni-cation among team members	POSSIBLE	TOLERABLE	**MEDIUM**	Regular status meetings and clear communication channels.	Scrum Master
Integration issues with existing systems	PROBABLE	UNDESIRABLE	**HIGH**	Conduct integration tests early and ensure alignment with current systems.	Developer

✅ Action Items

1. Resource Shortages: *@Doan Nhat Long (Swinburne HCM)*

- Maintain a buffer of resources
- Crosstrain team members.

2. Delayed Delivery of Platform Components: *@Lau Ngoc Quyen (Swinburne DN)*

- Develop a contingency plan with alternate suppliers

3. Quality Issues in Developed Features: *@Nguyen Thuan Khang (Swinburne HCM)*

- Implement thorough QA processes
- Conduct regular code reviews.

4. Miscommunication Among Team Members: *@Lau Ngoc Quyen (Swinburne DN)*

- Hold regular status meetings
- Establish clear communication channels

5. Integration Issues with Existing Systems: *@Le Ba Tung (Swinburne HCM)*

- Conduct integration tests early
- Ensure alignment with current systems.

Product Requirement

Target release	*Sep 29, 2024*
Epic	Build Direct-to-Consumer E-Commerce Platform
Document status	**DONE**
Document owner	*@Lau Ngoc Quyen* (Swinburne DN)
Designer	*@Lau Ngoc Quyen (Swinburne DN)* *@Nguyen Thuan Khang* (Swinburne HCM) *@Doan Nhat Long* (Swinburne HCM)
Tech lead	*@Le Ba Tung* (Swinburne HCM)
Technical writers	*@Lau Ngoc Quyen* (Swinburne DN) *@Nguyen Thuan Khang* (Swinburne HCM)
QA	*@Nguyen Thuan Khang* (Swinburne HCM) *@Lau Ngoc Quyen* (Swinburne DN)

🎯 Objective

Develop a user-friendly, scalable, and secure e-commerce platform for **Trung Nguyen Coffee Corp** to engage directly with consumers, enhance user experience, and expand market reach.

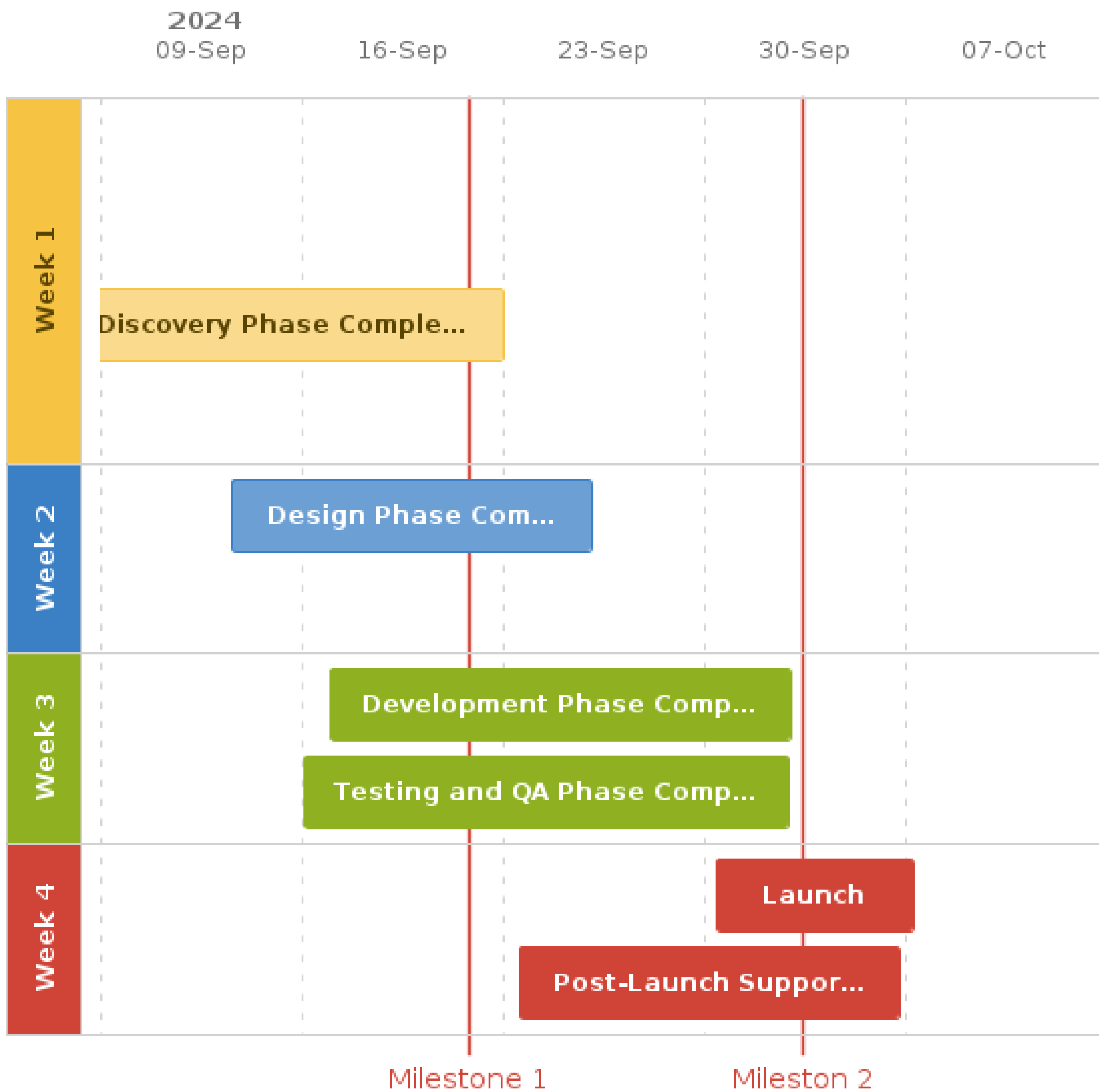
📊 Success metrics

Goal	Metric
Increase User Engagement	Increase direct consumer interactions by 30% within the first 6 months post-launch.
Enhance User Satisfaction	Achieve a Customer Satisfaction Score (CSAT) score of 4.5 out of 5 or higher.
Expand Market Reach	Enter 5 new international markets within the first year.
Improve Operational Efficiency	Reduce order processing time by 25% within the first 6 months.
Optimize Data Insights	Ensure 80% of marketing campaigns and product decisions are informed by data insights.
Ensure E-Commerce Functionality	Achieve a conversion rate of 2.5% or higher.
Achieve Security and Compliance	Zero security incidents per year.

🤔 Assumptions

- The platform will support multilingual and multicurrency features from launch.
- Integration with existing CRM and ERP systems will be feasible.
- All necessary third-party tools and integrations are accessible and compatible.

Milestones



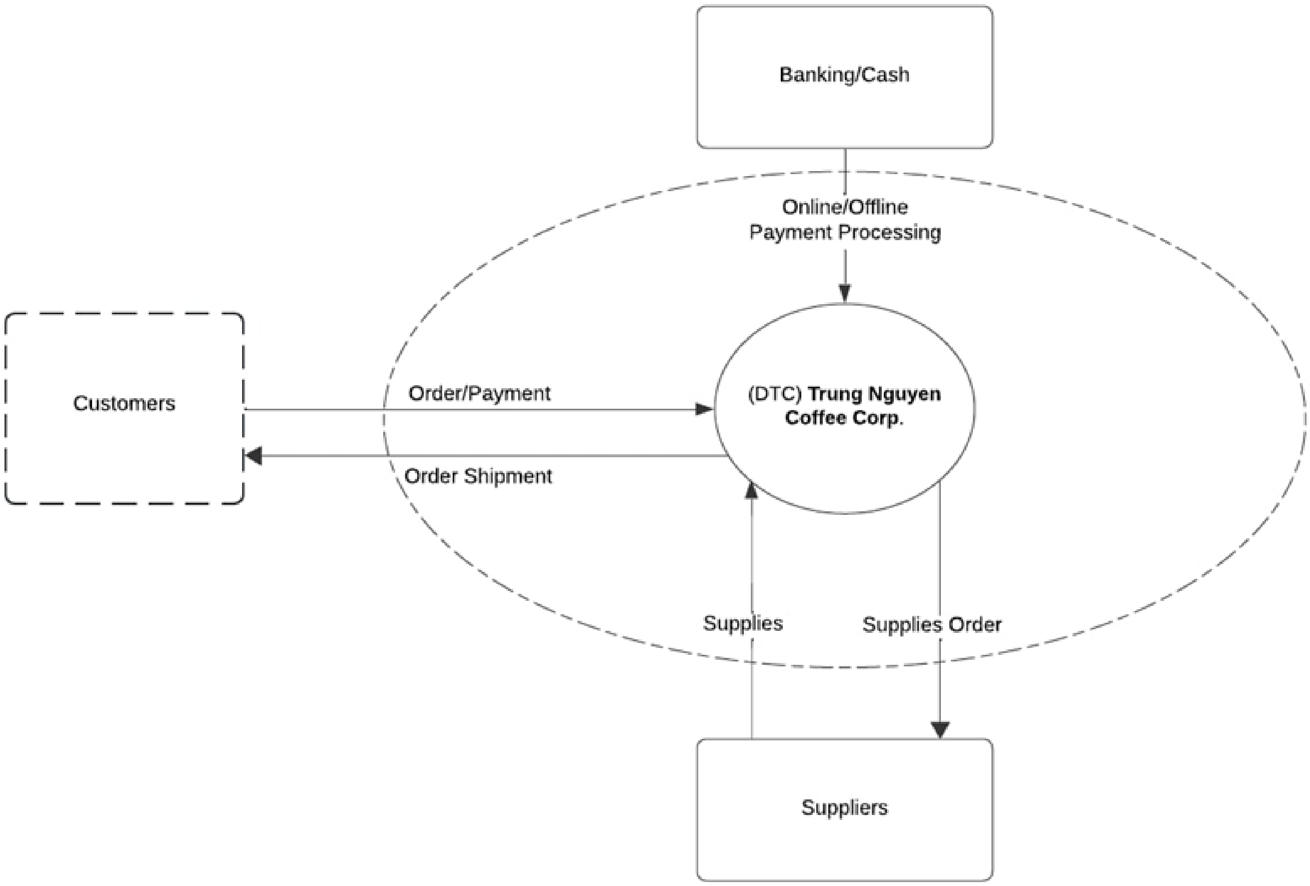
2024
09-Sep
16-Sep
23-Sep
30-Sep
07-Oct
Week 1
Discovery Phase Comple...
Week 2
Design Phase Com...
Week 3
Development Phase Comp...
Testing and QA Phase Comp...
Week 4
Launch
Post-Launch Suppor...
Milestone 1
Mileston 2

🗒️Requirements

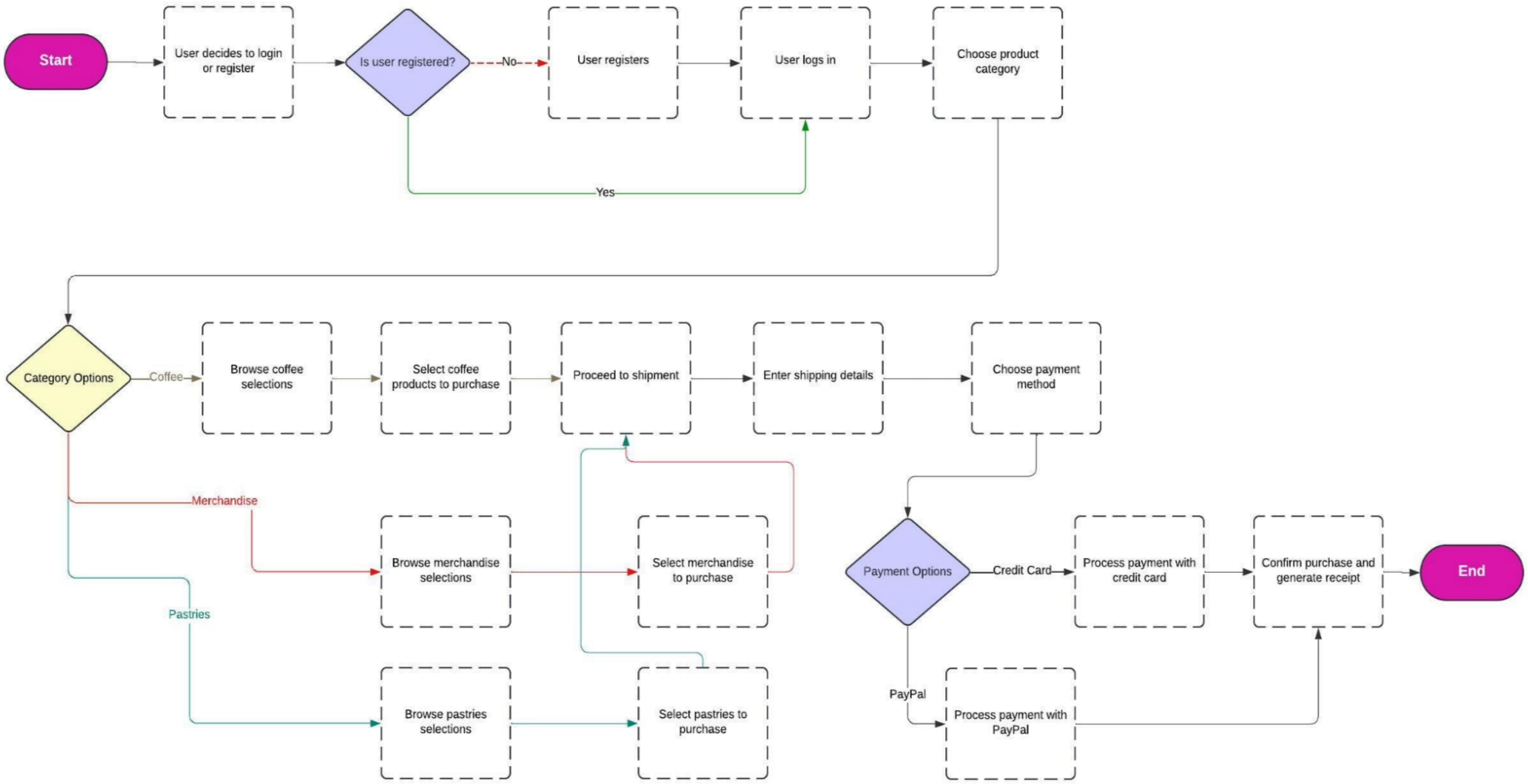
Requirement	User Story	Importance	Notes
User Interface (UI)	As a user, I want a visually appealing interface so that I enjoy using the platform.	HIGH	Must align with brand identity.
User Experience (UX)	As a user, I want easy navigation and simple checkout so that I can quickly complete my purchase.	HIGH	Focus on smooth customer journey.
Mobile Optimization	As a mobile user, I want a responsive design so that I can shop comfortably on my phone	HIGH	Ensure compatibility across devices.
Multilingual and Multicurrency Support	As an international customer, I want to view content in my language and currency so that I can easily make purchases.	MEDIUM	Include major languages and currencies.
E-commerce Functionality	As a shopper, I want detailed product listings and secure payment options so that I can make informed purchases confidently.	HIGH	Essential for core shopping experience.
Inventory Management	As a store manager, I want real- time stock updates so that I can manage inventory efficiently.	MEDIUM	Critical for maintaining stock levels
Logistics and Shipping Integration	As a customer, I want real-time shipping rate calculations and trackable shipping options	MEDIUM	Enhances shipping transparency.

Data Security	As a user, I want my data to be secure so that I can trust the platform with my personal information.	**HIGH**	Adhere to data privacy regulations.
PCI Compliance	As a marketing team member, I want to manage content easily so that I can keep the platform up to date.	**MEDIUM**	Facilitate easy content updates
CMS Integration	As a marketing team member, I want to manage content easily so that I can keep the platform up to date.	**MEDIUM**	Facilitate easy content updates.
Data Analytics	As a business owner, I want to track customer behavior and sales performance so that I can make informed decisions.	**HIGH**	Integrate with analytics tools.
Email Marketing Integration	As a marketer, I want to automate email campaigns so that I can engage with customers effectively.	**MEDIUM**	Support for major email platforms.
SEO Optimization	As a marketer, I want the platform optimized for search engines so that we can attract organic traffic.	**MEDIUM**	Follow SEO best practices.

🎨 User interaction and design



Data-flow diagram level 0



Data-flow diagram level 1

Instructions Based on the Flow Diagram:

1. **Start the Process:**
   - Begin by deciding whether to **login** or **register**.

2. **User Registration/Login:**
   - If the user is **registered**, they can log in directly.
   - If the user is **not registered**, they need to complete the registration process and then log in.

3. **Choose Product Category:**
   - After logging in, the user selects a **product category** from the available options: **Coffee**, **Merchandise**, or **Pastries**.
4. **Browsing and Selection:**
   - **Coffee:** Browse the coffee selections and choose products to purchase.
   - **Merchandise:** Browse available merchandise and select items to purchase.
   - **Pastries:** Browse pastry options and choose items for purchase.

5. **Proceed to Shipment:**
    - After selecting the products, proceed to the **shipment** stage.
    - Enter **shipping details** such as address and delivery preferences.

6. **Choose Payment Method:**
    - Select a payment method from the available options: **Credit Card** or **PayPal**.

7. **Process Payment:**
    - **Credit Card:** Process the payment with your credit card.
    - **PayPal:** Process the payment through PayPal.

8. **Confirm Purchase:**
    - After successful payment, confirm the purchase and generate a receipt.

9. **End the Process:**
    - The process is complete, and the transaction is finalized.

Open Questions

"Question"	**Answer**	**Date Answered**
What rewards will customers value most? Consider exclusive content or event access?	Include points for purchases, redeemable for discounts or products, and tiered rewards. Personalize based on purchase history.	Sep 26, 2024
Are there specific regulations for the new markets we're entering?	Check for compliance with CCPA, PIPA, and other local regulations based on target markets.	Sep 27, 2024
Any preferred payment processors or gateways?	Support credit/debit cards, digital wallets, bank transfers, and local options.	Feb 22, 2024
Which languages and currencies should we prioritize?	Support multiple languages and currencies with real- time conversion and localization.	Sep 27, 2024

💸Budget Breakdown

Category	Details	Estimated Cost
Design and Development Costs		
UI/UX Design	Wireframes, mockups, user flows	$15,000
Platform Development	Coding, feature implementation, integration	$50,000
Licensing Fees		
Software Licenses	Third-party tools or platforms	$5,000
API Integrations	Integration with external services	$3,000
Third-party Tools and Integrations		
Analytics Tools	Tools like Google Analytics	$2,000
Marketing Tools	Email marketing platforms, SEO tools	$1,500
Post-launch Maintenance and Support		
Technical Support (6 months)	Ongoing support for troubleshooting and fixes	$10,000
Updates	Platform updates and enhancements	$4,000
Training and Documentation		
Training Sessions	Training Trung Nguyen Coffee Corp's team	$3,000
Documentation	User guides and manuals	$1,500
Total Estimated Budget		$95,000

⚠️Out of Scope

Physical Store Integration:

- **Description**: Integration with Trung Nguyen Coffee Corp's physical store systems or in-store point-of-sale (POS) systems.
- **Reason**: The focus is on developing a digital e-commerce platform, not on integrating with physical store operations.

Advanced AI Features:

- **Description**: Implementation of advanced AI technologies such as fully autonomous customer service chatbots or advanced predictive analytics.
- **Reason**: The project will include standard AI tools but not advanced or cutting-edge AI features.

Custom Hardware Development

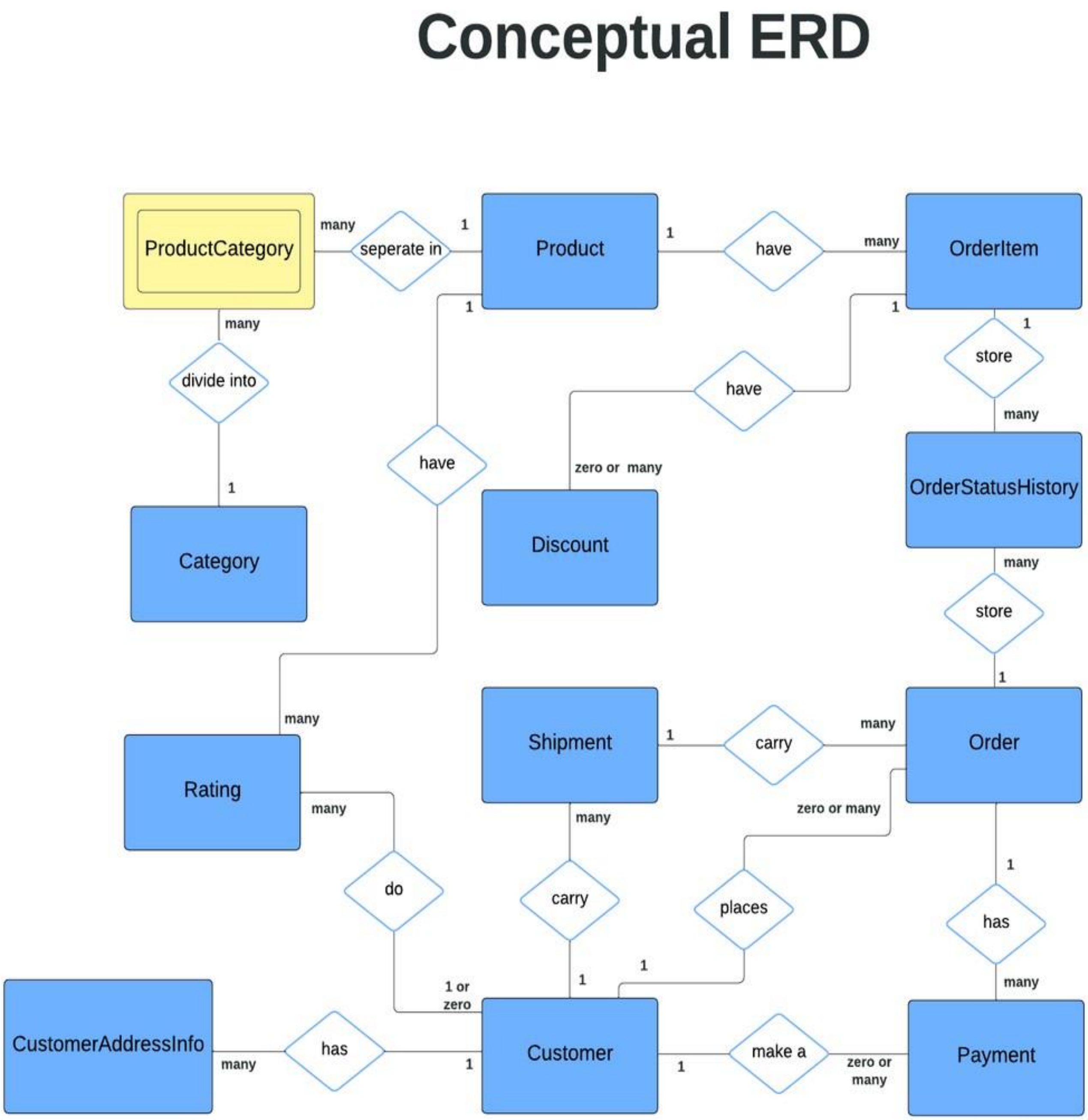
- **Description**: Development or integration of custom hardware solutions, such as digital kiosks or in-store tablets.
- **Reason**: The project scope is limited to software and digital services, excluding physical hardware.

Third-Party Platform Modifications:

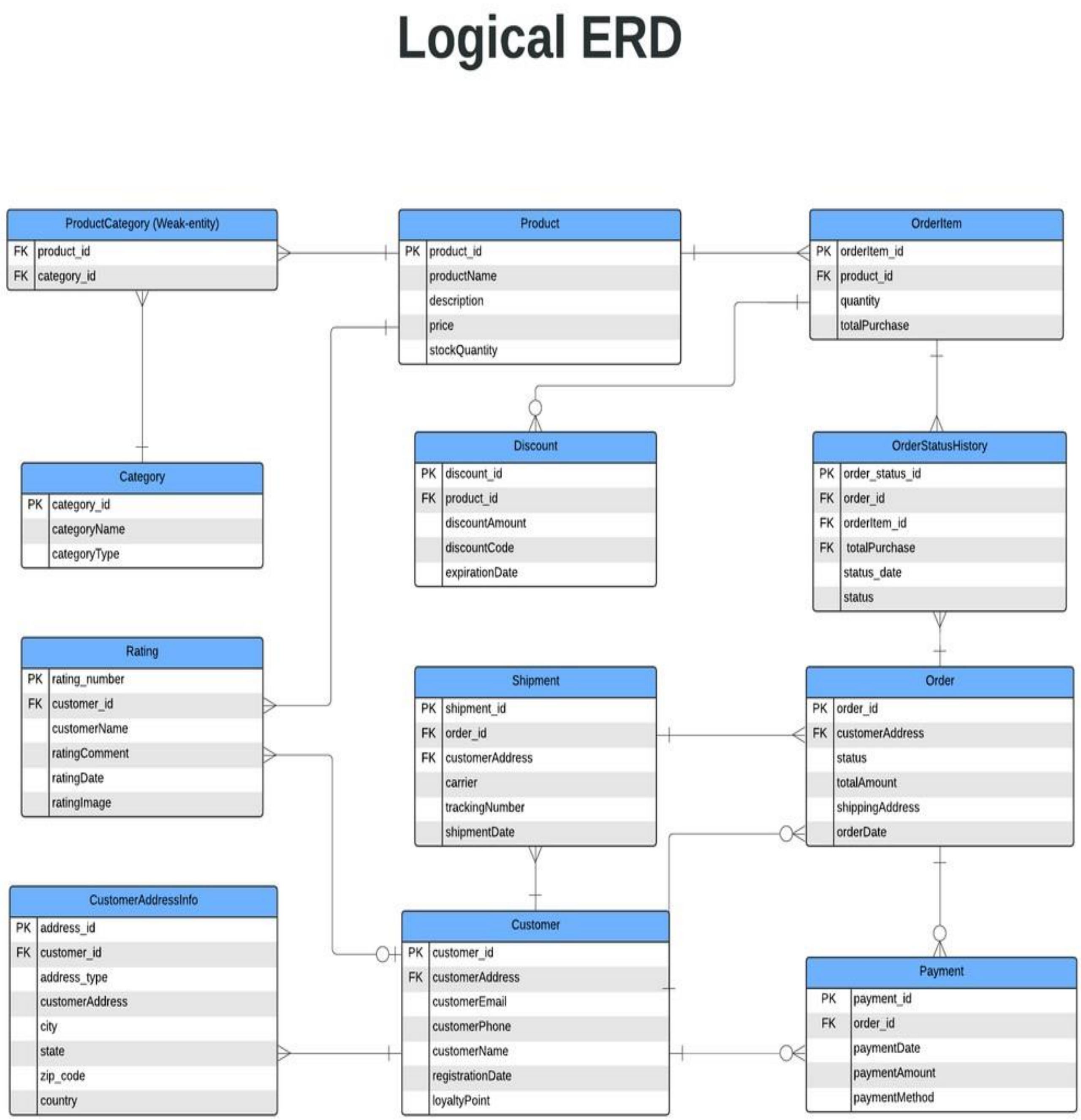
- **Description**: Custom development or modification of third-party e-commerce platforms (e.g., Shopify, Magento) for specific features.
- **Reason**: The platform will be developed as a new, custom solution rather than modifying existing third-party systems

Initial ERD Diagram

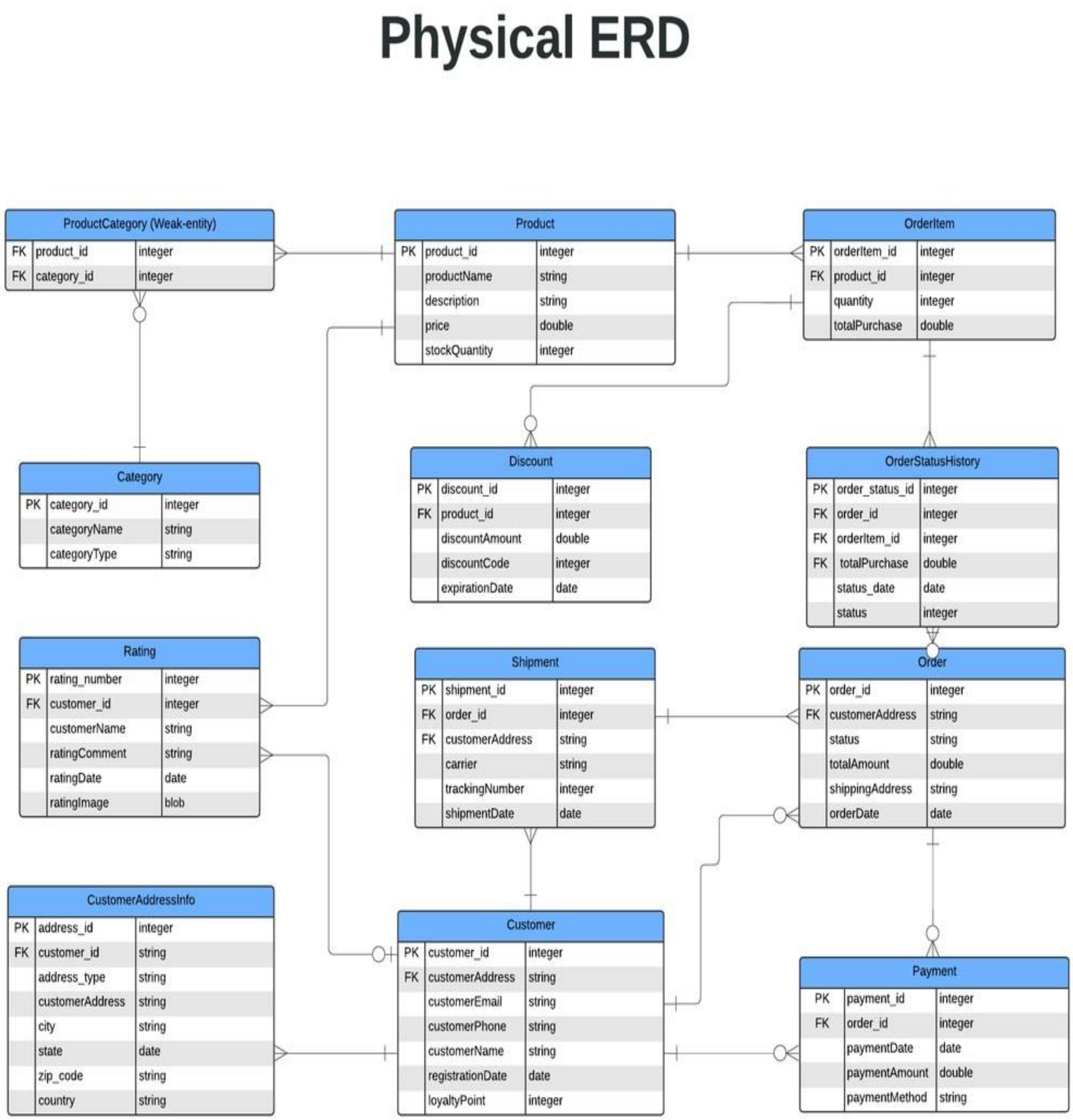
ERD level 0 (Conceptual Diagram)



ERD level 1 (Logical Diagram)



ERD level 2 (Physical Diagram)



ERD explaination:

1. ProductCategory (Weak Entity)

- **Attributes:**
  - product_id (FK): Foreign key from **Product.**
  - category_id (FK): Foreign key from **Category.**

Relationship:

- **Product:** A weak entity that links to **Product** through a **1-to-many** relationship.
- **Category: Many Products** can belong to **one Category** through this entity, forming a **many-to-many** relationship between products and categories.

2. Category

- **Attributes:**
  - category_id: Primary key for identifying categories.
  - categoryName: Name of the category.
  - categoryType: Describes the type of category (e.g., Electronics, Clothing).

Relationship:

- **ProductCategory: 1 Category** can contain **many Products** through the weak entity, forming a **many-to-many** relationship.
- **Product:** Linked to products through the weak entity ProductCategory.

3. Product

- **Attributes:**
  - product_id: Primary key for the product.
  - productName: Name of the product.
  - description: Product description.
  - price: Price of the product.
  - stockQuantity: Quantity available in stock.

Relationship:

- **ProductCategory: 1 Product** can belong to **many categories.**
- **OrderItem: 1 Product** can be part of **many OrderItems** (1-to-many).
- **Discount: 1 Product** can have **many Discounts** (1-to-many).
- **Rating: 1 Product** can receive **many Ratings** (1-to-many).
- **Shipment: Many Products** can be part of a **shipment** via orders.

4. Discount

- **Attributes:**
  - discount_id: Primary key for identifying discounts.
  - discountCode: The discount code.
  - discountAmount: The amount of discount.
  - product_id (FK): Foreign key from **Product**.
  - expirationDate: The date the discount expires.

Relationship:

- **Product: 1 Product** can have **many Discounts**, but each discount is applied to **only 1 Product** (many-to-1).

5. OrderItem

- **Attributes:**
  - orderItem_id: Primary key for the order item.
  - order_id (FK): Foreign key from **Order**.
  - product_id (FK): Foreign key from **Product**.

- quantity: Quantity of the product ordered.
- totalPurchase: Total amount for the particular order item.

Relationship:

- **Order: 1 Order** can contain **many OrderItems** (1-to-many).
- **Product: 1 Product** can appear in **many OrderItems** (1-to-many).
- **OrderStatusHistory:** Each order item can have **many statuses** (1-to-many).

6. OrderStatusHistory

- **Attributes:**
  - order_status_id: Primary key for tracking order statuses.
  - order_id (FK): Foreign key from **Order**.
  - orderItem_id (FK): Foreign key from **OrderItem**.
  - status_date: Date when the status was updated.
  - status: Current status of the order (e.g., "shipped", "pending").

Relationship:

- **OrderItem: 1 OrderItem** can have **many statuses** (1-to-many).
- **Order: 1 Order** may have **multiple statuses** tracked by OrderStatusHistory (1-to-many).

7. Order

- **Attributes:**
  - order_id: Primary key for the order.
  - customerAddress: The address associated with the order.
  - status: Current status of the order.
  - totalAmount: Total amount of the order.
  - shippingAddress: The address for delivery.
  - orderDate: Date the order was placed.

Relationship:

- **OrderItem: 1 Order** contains **many OrderItems** (1-to-many).
- **OrderStatusHistory: 1 Order** can have **many statuses in its history** (1-to-many).
- **Customer: 1 Customer** can place **many Orders** (1-to-many).
- **CustomerAddressInfo: 1 CustomerAddress** can be linked to **many Orders** (1-to-many).
- **Payment: 1 Order** can have **many Payments** (1-to-many).
- **Shipment: 1 Order** can have **many Shipments** (1-to-many).

8. Rating

- **Attributes:**
  - rating_number: Primary key for the rating.
  - customer_id (FK): Foreign key from **Customer**.
  - customerName: Name of the customer giving the rating.
  - ratingComment: Comments provided by the customer.
  - ratingDate: Date of the rating.
  - ratingImage: Blob (binary large object) to store an image associated with the rating.

Relationship:

- **Product: 1 Product** can have **many Ratings** (1-to-many).
- **Customer: 1 Customer** can give **many Ratings** (1-to-many).

9. Shipment

- **Attributes:**
  - shipment_id: Primary key for the shipment.
  - order_id (FK): Foreign key from **Order**.
  - customerAddress: Address where the shipment is sent.
  - carrier: The shipping service used.
  - trackingNumber: Tracking number for the shipment.

- shipmentDate: Date when the shipment was dispatched.

Relationship:

- **Order: 1 Order** can have **many Shipments** (1-to-many).
- **Customer: 1 Customer** can receive **many Shipments** (1-to-many).

10. Customer

- **Attributes:**
  - customer_id: Primary key for the customer.
  - customerAddress: Address associated with the customer.
  - customerEmail: Email address of the customer.
  - customerPhone: Phone number of the customer.
  - customerName: Name of the customer.
  - registrationDate: Date the customer registered.
  - loyaltyPoint: Points the customer has earned from purchases.

Relationship:

- **Order: 1 Customer** can place **many Orders** (1-to-many).
- **CustomerAddressInfo: 1 Customer** can have **many addresses** (1-to-many).
- **Shipment: 1 Customer** can be linked to **many Shipments** (1-to-many).
- **Rating: 1 Customer** can provide **many Ratings** (1-to-many).

11. CustomerAddressInfo

- **Attributes:**
  - address_id: Primary key for the customer address.
  - customer_id (FK): Foreign key from **Customer**.
  - address_type: The type of address (e.g., home, work).
  - customerAddress: The actual address.
  - city: City of the address.

  - state: State of the address.
  - zip_code: Zip code of the address.
  - country: Country where the address is located.

Relationship:

- **Customer: 1 Customer** can have **many addresses** (1-to-many).
- **Order: 1 CustomerAddress** can be used for **many Orders** (1-to-many).

12. Payment

- **Attributes:**
  - payment_id: Primary key for the payment.
  - order_id (FK): Foreign key from **Order**.
  - paymentDate: Date when the payment was made.
  - paymentAmount: Amount of the payment.
  - paymentMethod: Method used for payment (e.g., credit card, PayPal).

Relationship:

- **Order: 1 Order** can have **many Payments** (1-to-many).

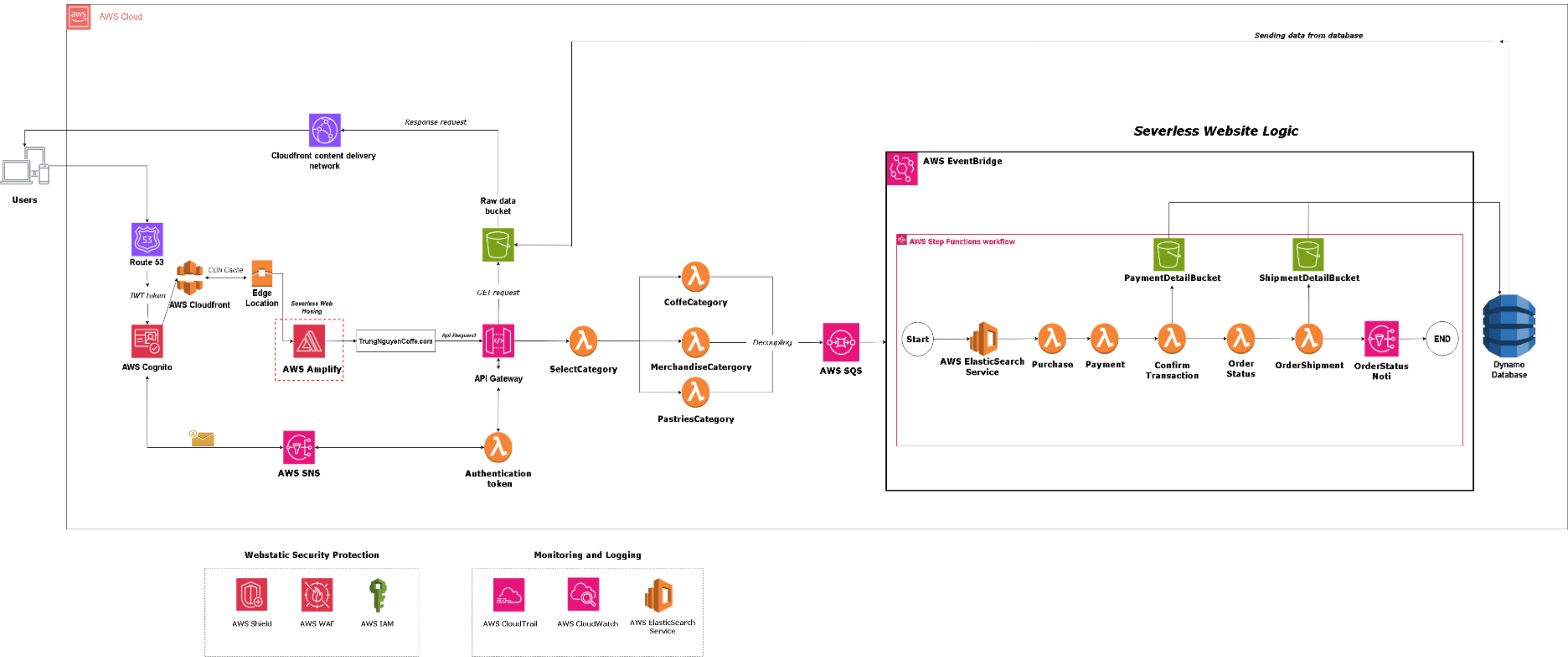
Web Hosting Tool

Diagram owner	@Lau Ngoc Quyen (Swinburne DN)
Team	🌟 @Lau Ngoc Quyen (Swinburne DN) 🌟 @Doan Nhat Long (Swinburne HCM) 🌟 @Nguyen Thuan Khang (Swinburne HCM) 🌟 @Le Ba Tung (Swinburne HCM)
Status	**DONE**
Last date updated	Sep 28, 2024
On this page	

🏆 Objective:

This document outlines the development tools and technologies to be used in building the Direct-to-Consumer (D2C) e-commerce platform for **Trung Nguyen Coffee Corp**.

📐 Architecture:



Architecture flow

Main Sections:

1. External Devices and Data Sources:

- Devices: These are the user devices interacting with the system, possibly desktops, mobile phones, or tablets.
- Data Inputs: Devices send requests that are routed through AWS Route 53 for DNS resolution. The JWT token is used for authentication, ensuring secure interactions.
- **AWS CloudFront**: Requests go through AWS CloudFront, which serves cached content via its Edge Locations, ensuring low-latency delivery across the globe.

2. Authentication and Security:

- **AWS Cognito**: Manages user authentication, generating JWT tokens for validating user identity and access permissions.

Security Protection:

- **AWS Shield**: Protects against Distributed Denial of Service (DDoS) attacks.
- **AWS WAF (Web Application Firewall)**: Provides filtering of malicious traffic to further secure the platform.
- **AWS IAM**: Manages access control for the system's resources.

3. Data Ingestion and Preprocessing:

- **S3 Bucket**: Stores raw data, images, or product-related assets requested from the serverless platform hosted on AWS Amplify.
- **API Gateway**: Acts as a front-end service to direct incoming HTTP requests to appropriate backend services. It triggers the appropriate backend **Lambda** functions.
- **SNS (Simple Notification Service)**: Facilitates communication between services and queues messages for processing.

4. Core Business Logic and Processing:

- **AWS Lambda**: Serverless compute functions are triggered to handle specific tasks:
- **Category Functions**: Handles requests for browsing product categories (e.g., Coffee, Merchandise, Pastries) via dedicated **Lambda** functions.
- **Step Functions**: Orchestrates the workflow of various tasks. For example:
- PurchaseStep: Processes purchases.
- PaymentInfo: Manages payment details and passes data to other steps.
- PaymentDetailHistory: Stores payment histories for future reference.
- **AWS SNS**: Sends relevant notifications between various **Lambda** functions or triggers additional steps.

5. Storage and Database:

- **DynamoDB**: Stores processed transactional data, including payment details, order history, and customer information. It is a NoSQL database, providing scalable and low-latency access.

6. Real-Time Data Processing:

- **AWS ElasticSearch Service**: Utilized for searching and analyzing large amounts of data related to user interactions or transactions.

7. Monitoring and Logging:

- **AWS CloudWatch**: Monitors system performance, collects logs from **Lambda** functions, and provides real-time operational insights.
- **AWS CloudTrail**: Tracks API calls for auditing and compliance purposes.

8. Customer Notifications:

- **Amazon SNS**: Sends email or SMS notifications to customers about order status updates, promotions, or transactional information. For example:
- Notifications after the successful purchase.
- Payment verification updates.
- This architecture leverages AWS's serverless and highly available services to create a secure and efficient data pipeline, offering scalable e-commerce functionality to **Trung Nguyen Coffee Corp**. The system is optimized for real-time data processing, security, and monitoring, ensuring seamless user experiences while maintaining data integrity.

Summary:

This architecture demonstrates a serverless and scalable data pipeline that integrates **real-time data ingestion**, **processing**, and **validation** using **AWS services** like **API Gateway**, **S3**, **Step Functions**, and **Lambda**. The system is designed for handling transactional workflows, emphasizing security and monitoring through services like **AWS WAF** and **CloudWatch**.

🚀Deployment strategy

🖼 Frontend Development

1. Framework: React.js

- **Purpose:** Efficient and scalable user interface development.
- **Reason for selection:** React.js is widely used for creating fast and interactive user interfaces. It offers flexibility, reusability of components, and is backed

by a large community for support.

- **Alternatives considered:** Angular, Vue.js.
- **Libraries/Extensions:**
  - React Router: For navigation and routing within the platform.
  - Styled Components: For enhanced styling management.

2. Mobile Optimization Framework: React Native

- **Purpose:** Mobile optimization for Android and iOS devices.
- **Reason for selection:** Allows the creation of native-like apps using the same codebase as the web platform.
- **Alternatives considered:** Flutter, Ionic.

🔐 Backend Development

1. Language: Node.js with Express

- **Purpose:** Build a scalable and fast backend API for handling requests, processing data, and interacting with the database.
- **Reason for selection:** Node.js is well-suited for real-time applications and offers excellent performance for handling numerous requests.
- **Alternatives considered:** Django (Python), Ruby on Rails.

2. Database: MongoDB

- **Purpose:** NoSQL database for handling dynamic product data, customer information, and orders.
- **Reason for selection:** MongoDB offers flexibility with unstructured data, allowing for quick changes and adaptability to new features.
- **Alternatives considered:** MySQL, PostgreSQL.

🏗️ Infrastructure and Deployment

1. Cloud Service Provider: AWS (Amazon Web Services)

- **Purpose:** Hosting, storage, and cloud computing.

- **Reason for selection:** AWS offers scalable infrastructure, robust security, and a variety of services that fit e-commerce needs.
- **Alternatives considered:** Microsoft Azure, Google Cloud Platform.

2. Containerization: Docker

- **Purpose:** Simplifies the deployment process by using containers to ensure consistency across development, staging, and production environments.
- **Reason for selection:** Docker allows efficient use of resources and easier collaboration between developers.
- **Alternatives considered:** Vagrant.

3. CI/CD Pipeline: Jenkins

- **Purpose:** Automate the integration and deployment process.
- **Reason for selection:** Jenkins is highly customizable and integrates with various tools for testing, building, and deploying code.
- **Alternatives considered:** CircleCI, TravisCI.

🏬 Version Control

1. Tool: GitHub

- **Purpose:** Source code management and version control.
- **Reason for selection:** GitHub offers excellent collaboration features, including pull requests, issue tracking, and version history.
- **Alternatives considered:** GitLab, Bitbucket.

🛡️ Security and Compliance

1. Security: OAuth 2.0

- **Purpose:** Implement secure user authentication and authorization.

- **Reason for selection:** OAuth 2.0 ensures secure handling of tokens and user sessions.
- **Alternatives considered:** JWT (JSON Web Token).

2. Compliance: PCI DSS

- **Purpose:** Ensure compliance with Payment Card Industry Data Security Standards for handling online transactions securely.
- **Reason for selection:** PCI DSS is mandatory for handling credit card information securely.

💻 Testing and QA

1. **Tool: Selenium**
    - **Purpose: Automated testing for frontend and backend functionalities.**
    - **Reason for selection: Selenium supports multiple browsers and is highly customizable for end-to-end testing.**
    - **Alternatives considered: Cypress, TestCafe.**
2. **Tool: Mocha**
    - **Purpose:** Unit testing for backend functionality.
    - **Reason for selection:** Mocha provides flexible asynchronous testing, ideal for Node.js environments.
    - **Alternatives considered:** Jest.

📈 Analytics and Reporting

1. **Tool: Google Analytics**
    - **Purpose:** Track customer interactions, sales data, and user behavior on the platform.
    - **Reason for selection:** Google Analytics provides in-depth insights and detailed reports essential for data-driven marketing.
    - **Alternatives considered:** Mixpanel, Hotjar.

🤝 Collaboration and Project Management

1. **Tool: Jira**
   - **Purpose:** Manage tasks, track progress, and organize sprints using Agile methodologies.
   - **Reason for selection:** Jira offers robust features for task management, sprint planning, and tracking team velocity.
   - **Alternatives considered:** Trello, monday.com | Your go-to work platform

2. **Tool: Confluence**
   - **Purpose:** Documentation and knowledge-sharing among the team.
   - **Reason for selection:** Confluence integrates seamlessly with Jira and allows team collaboration on all project documentation.
   - **Alternatives considered:** Notion.

Appendix 1 – My user manual

Ngoc Quyen Profile:	
Environments I like to work in	• I prefer a quiet, focused work environment with minimal distractions. A private coffee shop for study or a small team workspace would be ideal.
Preferred working hours	• I try to maintain a healthy work-life balance, so I avoid working excessively long hours or working on weekends unless it's absolutely necessary.
Communication preferences	• I prefer clear, concise, and direct communication, whether it's via email, video calls, or in-person meetings. Effective and straightforward communication helps me stay on track.
Preferred ways to receive feedback	• I'm open to feedback delivered in a respectful and constructive manner, as it helps me to continually grow and improve. I value feedback that is solution-oriented and focused on development.
Things I need	• Reliable and up-to-date technology to perform my work effectively. • Supportive and collaborative team members who are willing to share knowledge and provide assistance when needed.
How I learn best	• I'm a visual and auditory learner, so I benefit from a combination of written materials, demonstrations, and hands-on practice. Seeing and hearing concepts in action makes them easier to understand.
Things I struggle with	• I sometimes find it challenging to manage my time when I have too

	many tasks at once. Additionally, staying focused on less interesting but necessary tasks can be difficult at times.
Things I love	• Going to the gym to stay healthy and energized. • Coding and continuously improving my programming skills. • Watching movies to relax and enjoy creative storytelling. • Spending quality time with my loved ones, as it keeps me grounded and fulfilled. • Learning new things and expanding my knowledge. • Collaborating with talented and passionate colleagues to achieve shared goals. • The sense of accomplishment I feel when I complete a project to the best of my abilities.
If I were an animated gif/meme/animal/song, I would be…	• I would be a motivational workout GIF—something that pumps you up and encourages you to push forward.
My favorite saying	• "Dreams are just dreams until they become a reality. Don't worry about how grand your dreams are and how small you are. Because everyone starts from somewhere."
Other things I want you to know about me	• I enjoy spending time at the gym, watching movies, and hanging out with my loved ones. Coding is a big part of my life, and I always strive to improve myself in every aspect.
How I unwind after work	• After work, I love to hit the gym for a good workout. It helps me clear my mind and relieve stress. I also enjoy watching movies and spending time with loved ones for relaxation.

How I stay organized	• I use task management tools and a digital calendar to prioritize my work. Breaking down larger tasks into smaller, more manageable ones helps me stay on top of everything. I also make sure to block out time for personal activities and self-care.
What motivates me	• I'm motivated by challenges and the opportunity to learn new things. I enjoy working towards goals and achieving milestones, whether in personal fitness, coding projects, or professional development. Support from loved ones and a positive work environment also keep me going.

Thuan Khang Profile:	
Environments I like to work in	• I prefer a quiet, focused work environment with minimal distractions. A private coffe for study or small team workspace would be ideal.
Preferred working hours	• I try to maintain a healthy work-life balance, so I avoid working excessively long hours or on weekends unless it's absolutely necessary.
Communication preferences	• I prefer clear, concise, and direct communication, whether it's via email, video calls, or in-person meetings.
Preferred ways to receive feedback	• I'm open to feedback delivered in a respectful and constructive manner, as it helps me to continually grow and improve.
Things I need	• Reliable and up-to-date technology to perform my work effectively. • Supportive and collaborative team members who are willing to share knowledge and provide assistance when needed.
How I learn best	• I'm a visual and auditory learner, so I benefit from a combination of written materials, demonstrations, and hands-on practice.
Things I struggle with	• Managing multiple tasks simultaneously under tight deadlines can sometimes be overwhelming. • I find it challenging to delegate tasks because I prefer ensuring things are done correctly.
Things I love	• Continuously learning and expanding my knowledge and skills. • Collaborating with talented and passionate colleagues to achieve shared goals.

	• The sense of accomplishment and pride I feel when I complete a project or task to the best of my abilities.
If I were an animated gif/meme/animal/song, I would be...	• I would be a **wolf** because I value teamwork, loyalty, and the ability to lead when necessary while also working independently when needed.
My favorite saying	• "Dreams are just dreams until they become a reality. Don't worry about how grand your dreams are and how small you are. Because everyone starts from somewhere."
Other things I want you to know about me	• I value continuous improvement and seek opportunities to learn new skills in both my personal and professional life. • I appreciate humor and light-hearted moments in the workplace to keep the environment enjoyable and engaging.
My Ideal Team Dynamics	• I enjoy working in a team where there is a shared sense of responsibility and respect for each other's strengths. I appreciate open communication, mutual trust, and the willingness to help each other grow.
Challenges I'm Looking to Overcome	• I want to improve my ability to prioritize tasks more efficiently and manage time better to reduce stress during tight deadlines.
My Professional Goals	• I aim to deepen my technical expertise, especially in areas like software development and project management, while also taking on leadership roles to contribute more to team success.

Ba Tung Profile:	
Environments I like to work in	• I thrive in collaborative environments where I can bounce ideas off my colleagues. However, I also appreciate moments of quiet focus, especially when tackling complex tasks. A flexible workspace with both open areas for teamwork and quiet spots for concentration would be perfect.
Preferred working hours	• I prefer a balanced work schedule, with early mornings being my most productive time. I like to wrap up by the late afternoon, leaving time for personal activities. If needed, I'm open to working evenings, but I avoid working on weekends whenever possible.
Communication preferences	• I appreciate open and honest communication, whether it's through quick check-ins, Slack messages, or face-to-face discussions. I prefer to keep things transparent and clear to avoid misunderstandings and enjoy collaborative problem-solving sessions.
Preferred ways to receive feedback	• I value feedback that is direct and constructive, especially when it includes actionable steps I can take to improve. I like to have regular feedback discussions to ensure I'm on the right track and to continuously learn.

Things I need	• A reliable team with whom I can share ideas and get quick feedback. • Access to learning resources to further develop my skills and stay updated in my field. • Time to focus on projects that require deep thought without interruptions.
How I learn best	• I learn best by doing—hands-on experience is my go-to method. I also enjoy reading and researching topics in detail, especially when I'm starting something new. Practical applications and real-world examples help solidify my understanding.
Things I struggle with	• Sometimes I can be too focused on the details and miss the bigger picture. I also tend to overcommit myself, so I'm working on saying "no" more often to avoid burnout.
Things I love	• Being part of a motivated and supportive team. • Exploring new hobbies and learning new skills in my free time. • Solving complex problems through logic and creativity. • Taking time to relax and recharge, whether through travel, reading, or spending time outdoors. • Celebrating achievements, both big and small, with the people around me.
If I were an animated gif/meme/animal/song, I would be...	• I would be a "teamwork makes the dream work" meme—highlighting the importance of collaboration to reach goals efficiently.

My favorite saying	• "Success is not the key to happiness. Happiness is the key to success. If you love what you are doing, you will be successful."
Other things I want you to know about me	• I believe in continuous self-improvement. Whether it's in my work or personal life, I always seek out new opportunities to grow. I'm passionate about learning, and I love sharing that passion with others.
How I unwind after work	• I like to unwind by spending time with friends or family, engaging in a hobby like reading or gaming, or simply going for a walk outside. I also enjoy cooking and trying out new recipes as a way to relax and disconnect from work.
How I stay organized	• I use a mix of digital tools like task management apps and traditional methods like a written planner. I prioritize tasks using the Eisenhower matrix, which helps me focus on what's important and urgent. Regular reviews of my progress keep me on track.
What motivates me	• I'm motivated by the opportunity to solve challenging problems and the sense of achievement that comes from completing difficult tasks. I also draw motivation from working with inspiring colleagues and contributing to a larger purpose or goal.

Nhat Long Profile:	
Environments I like to work in	• I enjoy dynamic and fast-paced environments where things are constantly evolving. I thrive in a bustling, energetic workspace where I can easily collaborate with others. Being around people who are passionate and driven energizes me. I also love brainstorming in open spaces or casual settings like coffee shops.
Preferred working hours	• I'm not much of a morning person, so I prefer to start my day later, around 10 AM. I don't mind working into the evening if it means I can be more productive when my energy peaks. I'm flexible with my schedule and willing to adjust to meet team needs.
Communication preferences	• I love face-to-face conversations and quick video calls. I find it more engaging to talk through ideas in real-time, rather than waiting on emails or messages. That being said, I appreciate regular check-ins and updates via chat to keep things flowing smoothly.
Preferred ways to receive feedback	• I'm open to direct and candid feedback. The more straightforward, the better. I prefer to know exactly what I can improve on and don't shy away from constructive criticism. In fact, I welcome it as long as it's delivered respectfully. Quick feedback loops work best for me so I can adjust and move forward quickly.

Things I need	• Opportunities to bounce ideas off of teammates regularly. • Flexibility in how I manage my tasks and time, especially when I'm in a creative mode. • A fun, engaging team that loves to brainstorm and think outside the box. • Challenges that push me to think differently and come up with unique solutions.
How I learn best	• I'm an experiential learner, meaning I learn best by diving in and figuring things out through trial and error. I thrive in environments where I can experiment, try new things, and not be afraid to fail. I also enjoy learning through collaboration—sharing knowledge and skills with others.
Things I struggle with	• Sometimes I can get overly excited about new ideas and have a hard time sticking to one project or focus. I also tend to procrastinate on tasks that don't interest me as much, so I'm constantly working on finding ways to stay motivated with the less exciting parts of the job.
Things I love	• Engaging in creative problem-solving and thinking outside the box. • Participating in brainstorming sessions that lead to innovative ideas. • Working on projects that make a real impact and seeing the results firsthand. • Surrounding myself with energetic and enthusiastic people who are just as excited about the work as I am. • Spontaneous discussions or hangouts with friends and colleagues, especially when it leads to new ideas or laughter.

If I were an animated gif/meme/animal/song, I would be…	• I would be a "firework" GIF—full of energy, unpredictability, and excitement, constantly exploding with new ideas.
My favorite saying	• "Do what you love, and you'll never work a day in your life."
Other things I want you to know about me	• I'm always up for a challenge. Even when something feels impossible, I'll try my best to figure it out. I'm driven by creativity and love exploring new approaches. I also value having fun and believe that bringing joy to work can help boost productivity.
How I handle stress	• When I'm feeling stressed, I prefer to take a step back and breathe. I usually go for a short walk, meditate, or listen to calming music to clear my mind. Physical activity helps me release tension as well. I also like to break tasks down into smaller, more manageable steps to reduce the feeling of being overwhelmed.
What inspires me	• I'm deeply inspired by people who persevere through challenges and find success, whether in business, art, or life. I love reading about innovative leaders and thinkers, especially those who've managed to turn obstacles into opportunities. Seeing people overcome adversity motivates me to push myself further.
How I balance personal and professional life	• I value maintaining a strong separation between work and my personal life. I prioritize making time for family, friends, and hobbies, and I aim to keep my weekends free to recharge. Work is important, but I believe that having a fulfilling personal life allows me to bring my best self to the job.

Appendix 2 – Team Meeting notes

Meeting notes – Sep 9, 2024

📅 Date
Sep 9, 2024

👥 Participants
- @Lau Ngoc Quyen (Swinburne DN)
- @Doan Nhat Long (Swinburne HCM)
- @Nguyen Thuan Khang (Swinburne HCM)
- @Le Ba Tung (Swinburne HCM)

📕 Goals
1. **Initiate contact with Confluence and Jira platforms:**
   - Get team members acquainted with the tools and establish basic project structures.

2. **Topic Discussion:**
   - Discuss potential project topics, finalize the selection, and assign research responsibilities.

3. **Establish Core Project Pages:**
   - Create three essential pages on Confluence: Team Home, Team Member, and Project Plan pages.

4. **Coordinate Follow-up Meeting:**
   - Set a follow-up meeting to review progress and adjust next steps.

🗣️Discussion topics

Time	Item	Presenter	Notes
1 hour	Familiarize with Confluence and Jira platforms	Lau Ngoc Quyen	• Each team member will create individual pages to familiarize themselves with Confluence and explore Jira for task management. Temporary pages will be used.
30 minutes	Topic discussion	Lau Ngoc Quyen	• Selecting a suitable project topic based on team interest and skillsets. Conduct initial research to gather relevant data on the selected topic.
1 hour and 45 minutes	Create three separate pages.	Team Member	• Set up three main pages on Confluence: Team Home, Team Member, and Project Plan. These will serve as the foundation for the project's documentation.

🔖Detailed Discussion Overview:

1. Familiarize with Confluence and Jira Platforms:

- **Lau Ngoc Quyen** led the team through the basics of using **Confluence** for documentation and **Jira** for managing tasks. Team members will:
    - Create individual pages to practice Confluence functionality.
    - Explore **Jira** to understand how tasks will be tracked, assigned, and completed.
    - Discuss any challenges or questions related to using the tools.
- Each team member is expected to spend time experimenting with these tools before the next meeting to ensure a smooth workflow moving forward.

2. Topic Discussion:

- The team discussed multiple potential topics, aligning with project goals, available resources, and the interests of each member.

- After deliberation, the project topic was narrowed down to a few options, focusing on areas that leverage both technical and research skills. A final decision will be made after further research into these areas.

- **Nguyen Thuan Khang** led the topic discussion, asking team members for their inputs and preferences. He also assigned research tasks to ensure that each topic is thoroughly investigated before finalizing the selection.

3. Create Core Project Pages:

- The team agreed to set up three primary pages on Confluence:
    a. **Team Home Page:** A central hub that will contain key information, objectives, and an overview of the project.

    b. **Team Member Page:** A space where each team member can provide a brief introduction, their roles, and their contributions to the project.

    c. **Project Plan Page:** A detailed page outlining the scope, timeline, and deliverables of the project.
- **Lau Ngoc Quyen** guided the team through the structure of these pages, ensuring each member understood the format and content expectations.

✅ Action items

☒ **Create Individual Confluence Pages**

- Each member to create a personal Confluence page to familiarize themselves with the platform and explore functionality.

☒ **Set Up Core Project Pages**

- Team Home, Team Member, and Project Plan pages should be completed, with initial content prepared for review in the next meeting.

☒ **Research Project Topic**

- Conduct thorough research on the proposed topics and compile data to present during the next meeting for final topic selection.

☒ **Facilitate Jira Setup**

- **Lau Ngoc Quyen** will ensure all team members are set up on Jira and understand how to use the platform for task assignments and progress tracking.

⤴ Decisions

↰Project Tools:

- Confluence will be used for documentation and collaboration, while Jira will handle project management and task assignments.

↰Page Setup:

- Team will establish the Team Home, Team Member, and Project Plan pages in the next session.

↰Next Meeting:

- A follow-up team meeting will be coordinated to review progress on the project pages and topic research.

📅 Meeting Adjourned

- **Next Meeting Date:** Sep 16, 2024

Meeting notes – Sep 16, 2024

📅 Date
Sep 16, 2024

👥 Participants
- @Lau Ngoc Quyen (Swinburne DN)
- @Doan Nhat Long (Swinburne HCM)
- @Nguyen Thuan Khang (Swinburne HCM)
- @Le Ba Tung (Swinburne HCM)

🥅 Goals
1. **Finalize and Review Risk Assessment and Persona Pages:**
    - The team aims to complete the Risk Assessment and Persona pages, ensuring they are well-structured, comprehensive, and reflect accurate details of the project.

2. **Start the Creation of the Empathy Map Page:**
    - Lay the groundwork for the Empathy Map, focusing on gathering insights about user behaviors, emotions, and pain points, which will be crucial for project development.

3. **Discuss Upcoming Steps for Product Requirements:**
    - Begin brainstorming and planning for the Product Requirements document, setting clear milestones and determining the scope of what needs to be covered in the next sprint.

🗣Discussion topics

Time	Item	Presenter	Notes
45 minutes	Review and Finalize Risk Assessment and Persona Pages	Lau Ngoc Quyen	• Team members to review the current content on the Risk Assessment and Persona pages. Ensure all identified risks have documented mitigation strategies, and Persona data is aligned with target demographics.
30 minutes	Begin Creation of the Empathy Map Page	Doan Nhat Long	• Discuss the format and approach for building the Empathy Map. Outline key areas of focus such as user goals, needs, frustrations, and emotional journeys. Delegate page creation tasks.
50 minutes	Plan for Upcoming Steps, Including Initial Product Requirements	Le Ba Tung, Nguyen Thuan Khang	• Outline the primary components of the Product Requirements, including functional and non-functional requirements. Set deadlines and allocate responsibilities for drafting the document.

📕**Detailed Discussion Overview:**

1. Review and Finalize Risk Assessment and Persona Pages:

- **Risk Assessment:**
    - Each risk listed in the document was revisited. Mitigation strategies were re-evaluated to ensure they are realistic and actionable. The team also discussed adding new risks that may have emerged during the

course of the project, such as stakeholder alignment issues or potential technology changes.
    - Specific attention was paid to how risks are ranked (in terms of likelihood and impact). The assessment will need to be completed with clear ownership of each risk item to ensure accountability.
- **Persona Pages:**
    - The team confirmed that the Persona pages reflect accurate and relevant data regarding the target users of the platform. Each persona was evaluated for completeness, including user goals, pain points, demographic information, and their relationship to the platform's intended features.
    - It was noted that the personas should serve as a living document, continually updated as more user data is gathered.

2. Begin Creation of the Empathy Map Page:

- The team discussed the structure of the Empathy Map, agreeing that it should focus on the following categories:
    - **User Goals:** What users are trying to achieve.
    - **User Emotions:** How users feel at different stages of interacting with the product.
    - **User Pain Points:** What frustrates users, and where challenges arise in their journey.
    - **User Motivations:** What drives users to take action.
    - **Opportunities for Product Improvement:** How the product can address user needs and frustrations.
- **Lau Ngoc Quyen** suggested utilizing user interviews and feedback gathered from early research to populate the Empathy Map, with a plan to review this in the next sprint.

3. Planning for Product Requirements:

- The team discussed the next major milestone, which is the creation of the **Product Requirements Document (PRD).**
    - **Functional Requirements** should focus on the core features of the e-commerce platform, such as user authentication, product catalog, payment processing, and shipment tracking.

    - **Non-functional Requirements** should cover system performance, security, scalability, and usability aspects.

    - **Nguyen Thuan Khang** proposed dividing the PRD creation tasks between team members, with a detailed outline to be ready by the next sprint planning session.

- **Deadlines:** Initial draft due on 25th September 2024, with the final review planned for 30th September 2024.

✅Action items

☒**Review and finalize Risk Assessment** DONE

- Ensure all risks are documented with mitigation plans and responsible owners. Add new risks if necessary.

☒**Persona Pages Completion** DONE

- Refine personas based on the latest insights and feedback from users

☒**Create Empathy Map Page** DONE

- Team to start working on user behavior insights and map emotional journeys across key product touchpoints.

☒**Outline for Product Requirements** DONE

- Draft the core elements of the PRD, focusing on both functional and non-functional requirements.

⤴️Decisions

↰ **Risk Assessment:**

- Confluence will be used for documentation and collaboration, while Jira will handle project management and task assignments.

↰ **Persona Pages:**

- Team will establish the Team Home, Team Member, and Project Plan pages in the next session.

↰ **Product Requirements:**

- The initial draft of the Product Requirements Document will be created by 25th September 2024, with a final review by 30th September. This will ensure that the project stays on schedule and meets the functional needs of the platform.

↰ **Empathy Map Creation:**

- The Empathy Map will focus on gaining a deeper understanding of user pain points and motivations. This will guide future iterations of the platform's design and feature prioritization.

📅 Meeting Adjourned

- **Next Meeting Date:** Sep 19, 2024

Meeting notes – Sep 19, 2024

📅 Date
Sep 19, 2024

👥Participants

- @Lau Ngoc Quyen (Swinburne DN)
- @Doan Nhat Long (Swinburne HCM)
- @Nguyen Thuan Khang (Swinburne HCM)
- @Le Ba Tung (Swinburne HCM)

📕Goals

1. **Finalize Initial ER Diagram:**
   - Review and finalize the initial Entity Relationship (ER) diagram for the project.

2. **Team Member Profiles:**
   - Complete and review **Appendix 1:** Team Member Profiles on Confluence.

3. **Team Meeting Notes Compilation:**
   - Create and review Appendix 2: Team Meeting notes, summarizing key discussions and decisions.

4. **Set Timeline for Next Steps:**
   - Plan the timeline and tasks for creating additional documentation and refining the project structure.

🗣Discussion topics

Time	Item	Presenter	Notes
45 minutes	Review Initial ER Diagram	Lau Ngoc Quyen	• The team will review the initial ER diagram, discussing necessary changes and refinements. Key entities and relationships will be finalized
30 minutes	Team Member Profiles	Doan Nhat Long	• Review and finalize the content for Appendix 1, ensuring each member's profile is updated with roles, skills, and contributions to the project.
50 minutes	Team Meeting Notes (Appendix 2)	Team Members	• Compile key meeting notes, summarizing all relevant decisions made to date. Finalize Appendix 2 for use in documentation.
15 minutes	Next Steps Planning and Timeline	Lau Ngoc Quyen	• Set deadlines for upcoming documentation tasks, including revisions to the ER diagram, member profile updates, and creation of the empathy map page.

🔖Detailed Discussion Overview:

1. Review Initial ER Diagram:

- The team will review the draft of the **Entity Relationship (ER) Diagram** to ensure all entities, attributes, and relationships align with the project's goals and structure.

- **Lau Ngoc Quyen** will guide the discussion to finalize any changes, ensuring the ER diagram accurately reflects the database structure needed for the project.

- The finalized ER diagram will be used as a reference for future documentation and project development.

2. Team Member Profiles (Appendix 1):

- Each member's profile will be reviewed to ensure the inclusion of key information such as roles, responsibilities, skills, and contributions.

- The profiles will be compiled into **Appendix 1**, which will be attached to the main project documentation as part of the project's resource planning.

- **Nguyen Thuan Khang** will coordinate the updates to ensure all profiles are consistent and detailed.

3. Compile Team Meeting Notes (Appendix 2):

- The team will summarize key points and decisions from all previous meetings and compile them into **Appendix 2.**

- **Team Members** will collaborate to ensure all important discussions are reflected in the meeting notes, especially those related to tool selection, project timelines, and team roles.

- Appendix 2 will serve as a reference for future discussions and task assignments.

4. Next Steps and Timeline Planning:

- After reviewing the ER diagram and meeting notes, the team will discuss the timeline for upcoming tasks:
  - Further **revisions to the ER diagram** based on feedback.
  - **Finalization of Team Member Profiles** for Appendix 1.
  - **Creation of the Empathy Map** and other key project pages (e.g., Product Requirements Page).

- **Lau Ngoc Quyen** will lead the discussion on assigning tasks and setting deadlines for the next meeting.

✅Action items

☒**Finalize Initial ER Diagram** **DONE**

- Ensure that all required changes to the ER diagram are made and reviewed by the team.

☒**Complete Appendix 1: Team Profiles** **DONE**

- Finalize team member profiles, ensuring all relevant data (roles, skills, responsibilities) are included in Appendix 1.

☒**Compile Appendix 2: Meeting Notes** **DONE**

- Summarize meeting notes from all prior discussions and finalize Appendix 2 for the project documentation.

☒**Create Empathy Map Page** **DONE**

- Begin Developing the Empathy Map page as part of the project's documentation.

⤴️Decisions

↰ **ER Diagram Refinement:**

- The team will make any necessary revisions to the ER diagram before the 26th of September 2024, ensuring it accurately reflects the project structure.

↰ **Finalization of Team Profiles (Appendix 1):**

- All team member profiles will be finalized and added to Appendix 1, providing an overview of the team's skills and roles.

↰ **Completion of Meeting Notes (Appendix 2):**

- Appendix 2 will be finalized, capturing all previous meeting discussions, decisions, and task assignments for future reference.

↰ **Empathy Map and Future Pages:**

- The team will start working on the Empathy Map and additional project documentation such as the Product Requirements page.

🗓️ Meeting Adjourned

- **Next Meeting Date:** Sep 28, 2024

Meeting notes – Sep 28, 2024

📅 Date
Sep 28, 2024

👥Participants
- @Lau Ngoc Quyen (Swinburne DN)
- @Doan Nhat Long (Swinburne HCM)
- @Nguyen Thuan Khang (Swinburne HCM)
- @Le Ba Tung (Swinburne HCM)

📕Goals
1. **Finalize Project Proposal:**
   - Review and finalize all sections of the project proposal.

2. **Confirm Submission Details:**
   - Finalize the method and timeline for submitting the project proposal.

3. **Prepare for Future Phases:**
   - Discuss next steps post-submission, including the feedback review and project implementation.

🗣Discussion topics

Time	Item	Presenter	Notes
45 minutes	Final Review of Project Proposal	Team Member	• Conduct a thorough review of the entire project proposal, focusing on key sections such as the introduction, requirements, and solution approach.
30 minutes	Submission Details and Deadlines	Team Member	• Confirm the proposal submission process, including deadlines, required formats, and recipient details.
15 minutes	Future Steps and Planning	Team Member	• Discuss the next steps after submission, including potential adjustments based on feedback and preparing for project execution.

🔖Detailed Discussion Overview:

1. Final Review of Project Proposal:

- **Lau Ngoc Quyen** will guide the review of the final draft of the project proposal, ensuring that all necessary sections are complete and clearly written.

- The team will focus on verifying the accuracy of:
  - **Problem Statement:** Clear identification of the project's goals.
  - **Requirements Analysis:** Detailing user stories, technical requirements, and other essential elements.
  - **Solution Approach:** Outlining the proposed solution, its architecture, and implementation strategy.
- Any last-minute changes will be discussed and incorporated during this session.

2. Submission Details and Deadlines:
- **Nguyen Thuan Khang** will lead a discussion on the logistics of submitting the proposal:

- **Submission Format:** Confirm the format (PDF, Word document, etc.) required for submission.

- **Submission Deadline:** Reconfirm the exact deadline for submitting the proposal and the responsible party for submission.

- **Submission Process:** Detail how and to whom the project proposal will be submitted.

3. Future Steps and Planning:

- After finalizing the proposal submission, the team will discuss post-submission steps:
    - **Feedback Review Process:** Preparing to incorporate feedback from reviewers or stakeholders.

    - **Project Execution Planning:** Discuss the high-level plan for starting the implementation phase once approval is obtained.

✅Action items

☒Finalize Project Proposal DONE

- Complete any last-minute revisions to the project proposal and ensure it's ready for submission necessary.

☒Confirm Submission Details DONE

- Ensure all details regarding submission format, deadline, and the recipient are confirmed.

☒Prepare for Post-Submission Review DONE

- Prepare for the feedback review phase and plan adjustments or changes based on received feedback

⤴Decisions

↰ Final Project Proposal Approved:

- Confluence will be used for documentation and collaboration, while Jira will handle project management and task assignments.

↰ Proposal Submission Method Confirmed:

- The proposal will be submitted in the required format (e.g., PDF) by **Nguyen Thuan Khang** through the appropriate channel before the deadline.

↰ Next Steps Post-Submission:

- After submitting the proposal, the team will prepare for the feedback review process and begin planning the project implementation phase.

Team Health Check

Team name	**QLTK - COS20031**
Sponsor	Swinburne University of Technology
Health monitor cadence	**Monthly check-in:** This method focusing on workload balance, collaboration, and any immediate challenges. The goal is to identify small issues and improve team efficiency. **Quarterly review:** This is a more thorough examination of your team's dynamics, project alignment, and individual improvement. It addresses and highlights strategies to improve teamwork, productivity, and professional growth. **Annual review:** This is an opportunity to discuss the team's yearly achievements, challenges, and alignment with the corporate aims. A discussion workshop and in-depth feedback sessions are part of the process to design a roadmap for the following year, emphasizing long-term development and improvements.

🧑‍🤝‍🧑 Team health assessment

With your team, read the definition of each attribute of healthy, high-performing teams out loud. On the count of three have each person rate how they feel the team is doing compared to each definition (thumbs-up/green, thumbs-sideways/yellow, thumbs-down/red). Record the results of each attribute rating in the table. Highlight each cell using this color code: **HEALTHY** = "We're strong here", **BIT SICK** = "We're ok… but a little shaky", **SICK** = "We're not healthy".

🚩 Full-time owner

There is **one lead who is accountable** for the result of this project. This needs to be someone whose time is at least 80% dedicated to it, and who can champion the mission inside and outside of the team.

Sep 9, 2024	**Thuan Khang**, the project's full-time proprietor, is currently unable to dedicate 80% of his time due to personal concerns. We are actively working to allocate tasks and resources to compensate for his reduced availability. While we are concerned about this situation, we believe there are opportunities for improvement as we move forward.
Sep 15-16, 2024	**Thuan Khang** has not yet completed his personal commitments, which limits his ability to fully focus on the project. While he cannot dedicate 80% of his time to this endeavor, I have confidence in his past management abilities and the team's automation skills. We believe that, despite these challenges, our project has strong potential for success.
Sep 23, 2024	He has willingly extended his working hours to ensure he can commit at least 80% of his time to this initiative, promoting its growth and success. **Thuan Khang's** passion and focus on the project have contributed to a strong and productive working environment.
Sep 30, 2024	**Thuan Khang** has made this project his top priority and has developed a work schedule that enables him to dedicate at least 80% of his time to it. The project's progress has been greatly accelerated thanks to **Thuan Khang's** excellent time management skills and clear vision.
Oct 9, 2024	**Thuan Khang** has taken significant steps to ensure effective time and resource management. Each time he dedicates more than 80% of his time to the project, noticeable improvements are made.

Oct 16, 2024	During this time, **Thuan Khang** ran into several project management issues. We were a little concerned at this point because of a few issues that were impeding the project's progress.

⚖️ Balanced team

Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.

Sep 9, 2024	In the early stages, we assigned tasks to each team member appropriately. However, we were caught off guard when unforeseen issues arose, leading to several complications that negatively impacted our project.
Sep 15-16, 2024	We already understand how to organize and assign responsibilities efficiently. However, as we continue to acclimate to our new roles, it takes time for us to adapt and identify who should take charge when unusual tasks arise. This can extend the time required to complete tasks and potentially lead to complications.
Sep 23, 2024	We are accustomed to taking breaks from our duties, which is why we have policies and plans in place to address any unusual situations that may arise. By doing this, we have been able to ensure consistent project progress and productivity.
Sep 30, 2024	We have combined our knowledge and skills to enable everyone to complete the project efficiently and as soon as it is practical for them to do so.
Oct 9, 2024	We have established clear guidelines regarding responsibilities and have implemented plans and processes to address any unusual situations that may arise. We found this approach to be highly beneficial while executing our project.
Oct 16, 2024	Team members have a solid understanding of their assigned tasks, and we have divided responsibilities in a clear and systematic manner. The project progresses on schedule due to the team's consensus and mutual willingness to support one another. Our readiness to respond and find quick solutions makes it easier for us to manage unforeseen challenges.

💞 Shared understanding

The team has a **common understanding of why they're here**, the problem/need, are convinced about the idea, confident they have what they need, and trust each other.

Sep 9, 2024	We have established clear guidelines regarding responsibilities and have implemented plans and processes to address any unusual situations that may arise. We found this approach to be highly beneficial while executing our project.
Sep 15-16, 2024	To promote a spirit of change, it is important to enhance team members' trust and support for one another. Mutual trust and understanding are essential for building high collective effectiveness.
Sep 23, 2024	The team leader creates project diagrams to communicate objectives and tasks, allowing everyone to see how the components fit together.
Sep 30, 2024	Invite team members to propose recommendations for project fixes or adjust strategies as needed. The team feels more confident in responding to changes when everyone is informed about the objectives.
Oct 9, 2024	Hold regular meetings to ensure the team stays focused on the decisions and goals of the project.
Oct 16, 2024	Invite team members to propose recommendations for project fixes or adjust strategies as needed. The team feels more confident in responding to changes when everyone is informed about the objectives.

💍 Value and metrics

It's clear **what success means** from a business and user's perspective, and there is a unique value proposition in place for the target users and to the business. Success is defined, with a goal, and how it will be measured.

Sep 9, 2024	To ensure project success, establish specific targets. The development and construction of goals should be guided by user and business requirements.
Sep 15-16, 2024	Identify precise metrics to measure success, which may encompass revenue, conversion rates, user counts, or other indicators pertinent to the project.
Sep 23, 2024	We create and cultivate distinctive values for both businesses and users. This project has made significant achievements in the B2B2C sector, drawing attention and enticing users.
Sep 30, 2024	We compile feedback and reviews from businesses and clients to expand and enhance the project. This approach includes leveraging social networks, reviews, and direct customer surveys. Additionally, associations contribute opinion pieces or engage in discussions to share their perspectives.
Oct 9, 2024	We effectively and transparently communicate the values of our goods and services to our clientele, utilizing social media and marketing campaigns to achieve this.
Oct 16, 2024	We conduct assessments to monitor the project's progress and make necessary adjustments, ensuring that everything stays on track and meets objectives.

🥽 Proof of concept

Some sort of demonstration has been created and tested, that demonstrates why this problem needs to be solved, and demonstrates its value.

Sep 9, 2024	To save time and money, a proof of concept must be produced early in the development process.
Sep 15-16, 2024	A cross-functional team must develop the proof of concept to ensure it satisfies the requirements of all parties involved.
Sep 23, 2024	To ensure the proof of concept functions well and meets all requirements, our team needs to complete a thorough testing procedure.
Sep 30, 2024	Stakeholders must review the proof of concept to provide their input.
Oct 9, 2024	A procedure must be established for regularly reviewing the proof of concept to accommodate any modifications to the solution.
Oct 16, 2024	"The proof of concept is used to create a detailed development plan.

📄 One-pager

The **project is summarized** in a one-pager and shared with anyone so that they understand the purpose of the project, and its value.

Sep 9, 2024	Our group must establish a consistent procedure for updating the one-pager, such as revising it before every stakeholder meeting and after any significant project modification.
Sep 15-16, 2024	We must be mindful of one-pager presentations by employing clear, concise language, using illustrative graphics and charts, and dividing content with headings and subheadings.
Sep 23, 2024	Sharing a one-pager with the appropriate audience is essential.

Sep 30, 2024	To effectively convey information about the project, one-pagers must be used. Consequently, our team should employ one-pagers to spread knowledge about the project at meetings, conferences, and other gatherings.
Oct 9, 2024	One-pagers need to be reviewed periodically to determine whether updates are necessary to reflect shifts in the team's perspective on the project.
Oct 16, 2024	"One-pagers must be used to monitor our progress toward the project's objectives.

🧘 Managed dependencies

Clear understanding of complexity, infrastructure involved, risks, resources, effort, and timeline. Clear understanding of **who we depend on, and who depends on us.**

Sep 9, 2024	For managing dependencies, our team needs a well-defined procedure tailored to the project's size and type. This process must clearly identify the stages to be followed, the parties involved, and each party's responsibilities.
Sep 15-16, 2024	Technology and tools are essential for facilitating dependency management. Therefore, to support dependency management, the team must acquire and apply the necessary tools and technologies. These tools will enable our team to monitor the status of interdependent tasks, receive updates, and identify potential issues.
Sep 23, 2024	Risks associated with dependencies must be recognized and assessed. Steps must be taken to mitigate these risks, such as diversifying sources of supply, creating backup plans, and enhancing adaptability.
Sep 30, 2024	The team must develop a strategy for testing and assessing dependency management. This strategy should specify evaluation criteria and frequency. With this support, the team will ensure effective dependency management.

Oct 9, 2024	Handling dependencies is crucial for teams. Team members need training and development in dependency management. Therefore, the group must create a curriculum covering fundamentals, tools and techniques, and best practices.
Oct 16, 2024	Dependency management is an essential component of project management. It must be integrated into the project management process. Identifying project dependencies and creating a suitable management strategy should be incorporated into the team's project management procedure.

🏇 Velocity

The team is making **incremental progress** by shipping concrete iterations to stakeholders (and, even better, to production), learning along the way, and **implementing lessons learned**, resulting in greater success.

Sep 9, 2024	Although the team is growing rapidly, there is definitely room for development. To ensure steady and sustainable growth, the team requires a comprehensive and extended development plan.
Sep 15-16, 2024	For assessing development speed, our team needs precise measurements. This will allow us to track and enhance progress more effectively. Specific metrics, such as the number of features released and the time it takes to complete projects, must be chosen to assess development speed.
Sep 23, 2024	Close collaboration between team members is essential for accelerated progress. Therefore, the team needs to engage in activities such as regular meetings, knowledgesharing sessions, and training sessions to improve coordination.
Sep 30, 2024	We must seek assistance from other sources, including communities, professionals, and consulting firms. This will facilitate the team's rapid development.
Oct 9, 2024	The team must routinely assess elements such as member knowledge and expertise, the technology and tools used, and the working environment. Strategies to enhance these areas should then be devised.

Oct 16, 2024	The development of the team and the quality of the product or service must be balanced. The team must have a realistic plan to ensure both speed and quality in the development of the product or service.

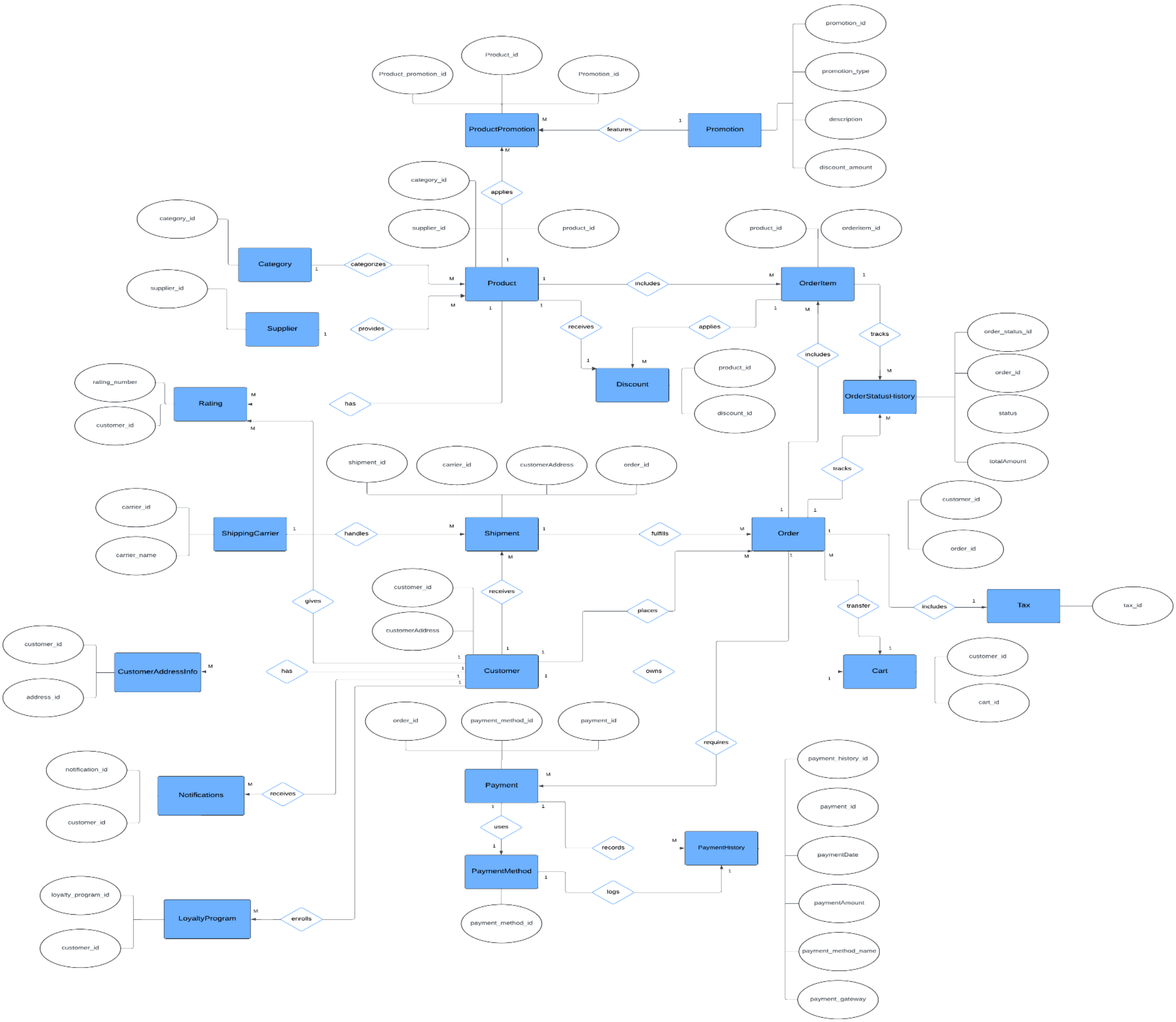
🎯 Focus areas

Ask your team to collectively come up with one attribute you want to focus on. Then, call out ways to move the **SICK** or **BIT SICK** toward **HEALTHY** . Make sure they are actionable, specific, and measurable.

Date	Focus areas and action items
Sep 10,2024	☒ Establish limitations and evaluate the current scalability of the system.
Sep 10,2024	☒ Assign manageable responsibilities to team members so they can collaborate and solve issues when the full-time owner is unavailable.
Sep 19,2024	☒ Create backup plans in advance to swiftly and efficiently address unexpected problems
Sep 25,2024	☒ Focus on implementation and prioritize activities to avoid atypical tasks from impacting the project.
Oct 10,2024	☒ Conduct vulnerability analyses and security tests on the system.
Oct 18,2024	☒ Use performance monitoring tools for routine system performance analysis.

Initial ER Diagram

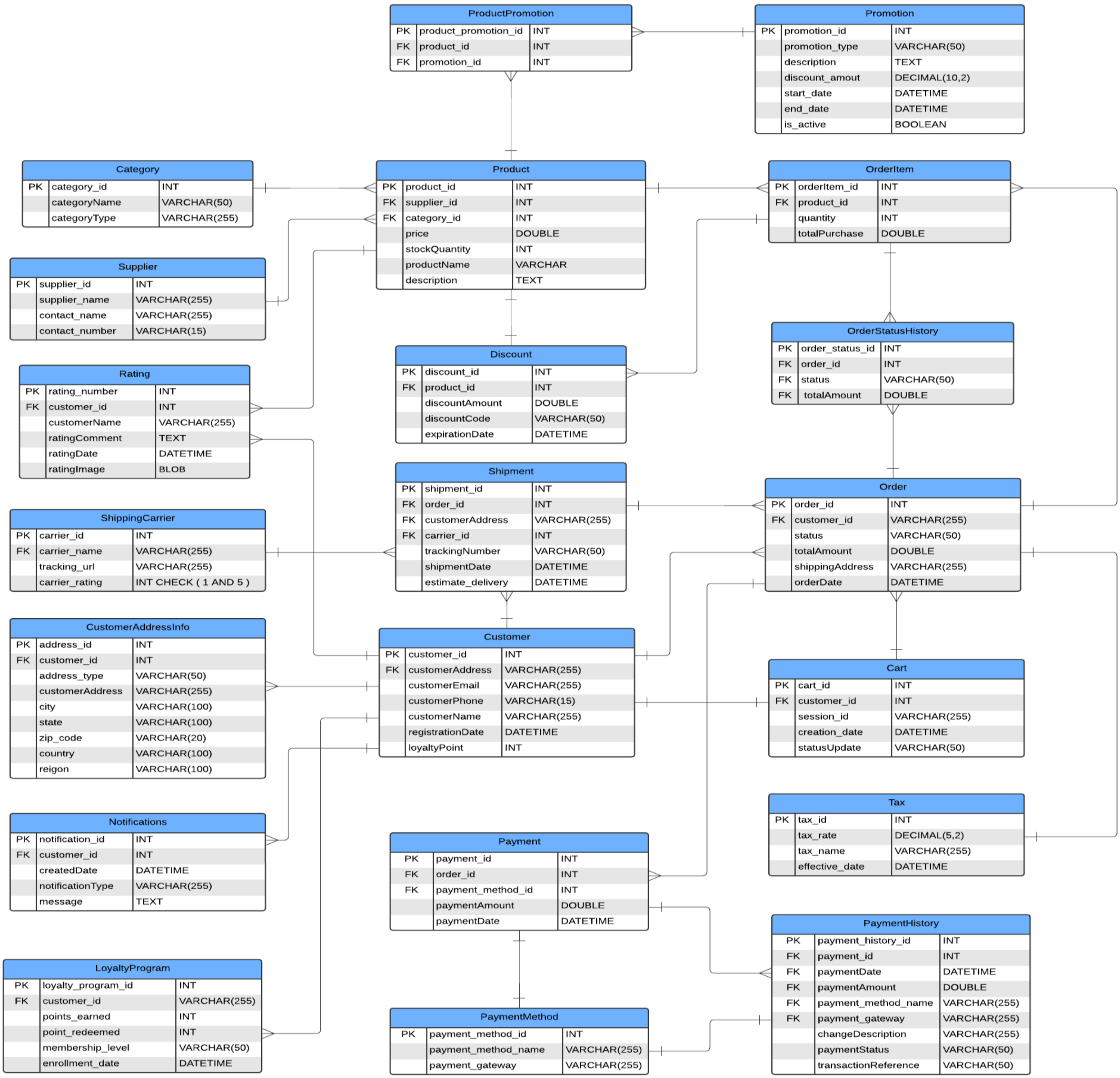
Conceptual Relationship Diagram



Conceptual Relationship Diagram

Entity Relationship Diagram

Note: Foreign Key (FK), Primary Key (PK)



ERD Explaination and Relationship

ERD Explaination :

1. Supplier

- **Purpose**: Represents suppliers who provide products to the business.
- **Attributes**:
    - **supplier_id**: Unique identifier for each supplier (PK).
    - **supplier_name**: Name of the supplier.
    - **contact_name**: Primary contact person at the supplier.
    - **contact_number**: Contact number for the supplier.

2. Category

- **Purpose**: Categorizes products to make it easier for customers to browse.
- **Attributes**:
    - **category_id**: Unique identifier for each category (PK).
    - **category_name**: Name of the category (e.g., electronics, clothing).
    - **category_type**: Additional type information about the category.

3. Product

- **Purpose**: Represents products that can be sold.
- **Attributes**:
    - **product_id**: Unique identifier for each product (PK).
    - **supplier_id**: Identifier for the supplier of the product (FK).
    - **category_id**: Identifier for the category the product belongs to (FK).
    - **price**: Selling price of the product.
    - **stock_quantity**: Number of items available in stock.
    - **product_name**: Name of the product.
    - **description**: Description of the product features.

4. ShippingCarrier

- **Purpose**: Represents shipping carriers used for order deliveries.
- **Attributes**:

- carrier_id: Unique identifier for each carrier (PK).
    - **carrier_name**: Name of the shipping carrier.
    - **tracking_url**: URL for tracking shipments.
    - **carrier_rating**: Rating of the carrier (1-5).

5. Shipment

- **Purpose**: Records details of shipments for orders.
- **Attributes**:
    - **shipment_id**: Unique identifier for each shipment (PK).
    - **order_id**: Identifier for the order being shipped (FK).
    - **customer_address**: Address where the order is shipped.
    - **carrier_id**: Identifier for the shipping carrier (FK).
    - **tracking_number**: Unique number for tracking the shipment.
    - **shipment_date**: Date the shipment was made.
    - **estimate_delivery**: Estimated delivery date.

6. OrderStatusHistory

- **Purpose**: Tracks the status changes of an order.
- **Attributes**:
    - **order_status_id**: Unique identifier for each order status entry (PK).
    - **order_id**: Identifier for the associated order (FK).
    - **status**: Current status of the order (e.g., processing, shipped).
    - **total_amount**: Total amount for the order at that status.

7. Customer

- **Purpose**: Represents customers who place orders.
- **Attributes**:
    - **customer_id**: Unique identifier for each customer (PK).
    - **customer_address**: Address of the customer.
    - **customer_email**: Email address of the customer.
    - **customer_phone**: Phone number of the customer.
    - **customer_name**: Name of the customer.

- registration_date: Date when the customer registered.
- loyalty_points: Points accumulated by the customer in the loyalty program.

8. Order

- **Purpose**: Represents customer orders.
- **Attributes**:
  - **order_id**: Unique identifier for each order (PK).
  - **customer_id**: Identifier for the customer who placed the order (FK).
  - **status**: Current status of the order (e.g., completed, canceled).
  - **total_amount**: Total amount of the order.
  - **shipping_address**: Address where the order will be shipped.
  - **order_date**: Date the order was placed.

9. OrderItem

- **Purpose**: Represents individual items in an order.
- **Attributes**:
  - **order_item_id**: Unique identifier for each order item (PK).
  - **order_id**: Identifier for the associated order (FK).
  - **product_id**: Identifier for the product being ordered (FK).
  - **quantity**: Quantity of the product ordered.
  - **total_purchase**: Total cost for this item (quantity x price).

10. Discount

- **Purpose**: Represents discounts applicable to products.
- **Attributes**:
  - **discount_id**: Unique identifier for each discount (PK).
  - **product_id**: Identifier for the product to which the discount applies (FK).
  - **discount_amount**: Amount or percentage of the discount.
  - **discount_code**: Code required to apply the discount.
  - **expiration_date**: Date when the discount expires.

11. CustomerAddressInfo

- **Purpose**: Stores multiple addresses for customers.

- **Attributes**:
    - **address_id**: Unique identifier for each address entry (PK).
    - **customer_id**: Identifier for the customer (FK).
    - **address_type**: Type of address (e.g., billing, shipping).
    - **customer_address**: Street address.
    - **city**: City of the address.
    - **state**: State of the address.
    - **zip_code**: Postal code.
    - **country**: Country of the address.
    - **region**: Region information.

12. Rating

- **Purpose**: Captures customer ratings and feedback on products.
- **Attributes**:
    - **rating_id**: Unique identifier for each rating (PK).
    - **customer_id**: Identifier for the customer who made the rating (FK).
    - **rating_comment**: Customer's comments about the product.
    - **rating_date**: Date when the rating was submitted.
    - **rating_image**: Optional image uploaded with the rating.

13. Tax

- **Purpose**: Represents tax rates applicable to purchases.
- **Attributes**:
    - **tax_id**: Unique identifier for each tax entry (PK).
    - **tax_rate**: Tax percentage.
    - **tax_name**: Name of the tax (e.g., VAT, sales tax).
    - **effective_date**: Date when the tax rate becomes effective.

14. PaymentMethod

- **Purpose**: Stores payment methods available to customers.
- **Attributes**:
    - **payment_method_id**: Unique identifier for each payment method (PK).

- **payment_method_name**: Name of the payment method (e.g., credit card, PayPal).
- **payment_gateway**: Gateway used for processing the payment.

15. Cart

- **Purpose**: Represents the shopping cart of a customer.
- **Attributes**:
  - **cart_id**: Unique identifier for each cart (PK).
  - **customer_id**: Identifier for the customer (FK).
  - **session_id**: Identifier for the customer's session.
  - **creation_date**: Date when the cart was created.
  - **status_update**: Current status of the cart (e.g., active, abandoned).

16. Payment

- **Purpose**: Records payments made for orders.
- **Attributes**:
  - **payment_id**: Unique identifier for each payment (PK).
  - **order_id**: Identifier for the associated order (FK).
  - **payment_method_id**: Identifier for the payment method used (FK).
  - **payment_amount**: Amount paid.
  - **payment_date**: Date when the payment was made.

17. PaymentHistory

- **Purpose**: Tracks changes and history of payments.
- **Attributes**:
  - **payment_history_id**: Unique identifier for each payment history entry (PK).
  - **payment_id**: Identifier for the associated payment (FK).
  - **payment_date**: Date of the payment.
  - **payment_amount**: Amount of the payment.
  - **payment_method_name**: Name of the payment method used.
  - **payment_gateway**: Gateway used for the transaction.
  - **change_description**: Description of any changes to the payment.
  - **payment_status**: Current status of the payment (e.g., completed, pending).

- **transaction_reference**: Reference number for the transaction.

18. Notifications

- **Purpose**: Sends notifications to customers about their orders or promotions.
- **Attributes**:
  - **notification_id**: Unique identifier for each notification (PK).
  - **customer_id**: Identifier for the customer receiving the notification (FK).
  - **created_date**: Date when the notification was created.
  - **notification_type**: Type of notification (e.g., order update, promotion).
  - **message**: Message content of the notification.

19. LoyaltyProgram

- **Purpose**: Tracks customer participation in loyalty programs.
- **Attributes**:
  - **loyalty_program_id**: Unique identifier for each loyalty program entry (PK).
  - **customer_id**: Identifier for the customer in the loyalty program (FK).
  - **points_earned**: Total points earned by the customer.
  - **points_redeemed**: Total points redeemed by the customer.
  - **membership_level**: Level of membership in the loyalty program (e.g., silver, gold).
  - **enrollment_date**: Date when the customer enrolled in the program.

20. Promotion

- **Purpose**: Represents special promotions available to customers.
- **Attributes**:
  - **promotion_id**: Unique identifier for each promotion (PK).
  - **promotion_type**: Type of promotion (e.g., seasonal sale, clearance).
  - **description**: Description of the promotion.
  - **discount_amount**: Amount of discount offered.
  - **start_date**: Date when the promotion starts.
  - **end_date**: Date when the promotion ends.
  - **is_active**: Indicates whether the promotion is currently active.

21. ProductPromotion

- **Purpose**: Links products to specific promotions.
- **Attributes**:
  - **product_promotion_id**: Unique identifier for each product promotion (PK).
  - **product_id**: Identifier for the associated product (FK).
  - **promotion_id**: Identifier for the associated promotion (FK).

Relationship Between ERD entities:

1. Supplier and Product:

- **Relationship**: One-to-Many.
- **Explanation**: Each supplier can provide multiple products, but each product is supplied by one specific supplier.
- **Foreign Key**: supplier_id in the Product table references supplier_id in the Supplier table.

Example: A coffee supplier (from Đắk Lắk, Lâm Đồng) might offer various types of coffee beans, and all those products will have the same supplier_id.

---

2. Category and Product:

- **Relationship**: One-to-Many.
- **Explanation**: A category can include multiple products (e.g., "Coffee Beans"), but each product belongs to one category.
- **Foreign Key**: category_id in the Product table references category_id in the Category table.

Example: The category "Coffee Machines" will include various products like espresso machines, grinders, etc., but each machine belongs to that single category.

---

3. Customer and Order:

- **Relationship**: One-to-Many.
- **Explanation**: A customer can place multiple orders over time, but each order belongs to only one customer.
- **Foreign Key**: customer_id in the Order table references customer_id in the Customer table.

Example: A customer named Long can have multiple orders across different dates, but each order belongs to Long only.

---

4. Order and OrderItem:

- **Relationship**: One-to-Many.
- **Explanation**: An order can have multiple items (products), but each item belongs to a single order.
- **Foreign Key**: order_id in the OrderItem table references order_id in the Order table.

Example: An order placed by a customer may include 3 products (like coffee beans, a grinder, and filters), which would each have an entry in OrderItem linked to the same order_id.

---

5. Product and OrderItem:

- **Relationship**: One-to-Many.
- **Explanation**: A product can appear in multiple order items (since many customers can order the same product), but each order item contains only one product.
- **Foreign Key**: product_id in the OrderItem table references product_id in the Product table.

Example: The product "Arabica, Robusta Coffee Beans" can appear in many different orders placed by various customers.

---

6. Shipment and Order:

- **Relationship**: One-to-Many.
- **Explanation**: Each order can have only one shipment associated with it, and each shipment refers to one specific order.
- **Foreign Key**: order_id in the Shipment table references order_id in the Order table.

Example: Once an order is processed, it is shipped out as a single shipment (E.g:Shopee, Lazada).

---

7. ShippingCarrier and Shipment:

- **Relationship**: One-to-Many.
- **Explanation**: A shipping carrier can handle multiple shipments, but each shipment is handled by a single carrier.
- **Foreign Key**: carrier_id in the Shipment table references carrier_id in the ShippingCarrier table.

Example: A shipment might be delivered via Lazada, Shopee, or XanhSM, and the carrier_id would identify which one.

---

8. Customer and CustomerAddressInfo:

- **Relationship**: One-to-Many.
- **Explanation**: A customer can have multiple addresses (e.g., home, work), but each address belongs to one customer.
- **Foreign Key**: customer_id in the CustomerAddressInfo table references customer_id in the Customer table.

Example: Khang can have multiple shipping addresses like his home and office, but each address is tied only to Khang.

---

9. Customer and Rating:

- **Relationship**: One-to-Many.
- **Explanation**: A customer can leave multiple ratings (for different products), but each rating belongs to one customer.
- **Foreign Key**: customer_id in the Rating table references customer_id in the Customer table.

Example: Tung might leave a rating for "Espresso Machine" and another for "Coffee Beans," but both ratings belong to Tung.

---

10. Product and Discount:

- **Relationship**: One-to-One.
- **Explanation**: A product can have multiple discounts applied over time, but each discount applies to one specific product.
- **Foreign Key**: product_id in the Discount table references product_id in the Product table.

Example: A product might have a 10% discount one week, followed by a 5%-off discount the next week.

---

11. Product and ProductPromotion:

- **Relationship**: One-to-Many.
- **Explanation**: A product can be part of multiple promotions, and a promotion can apply to multiple products. The ProductPromotion table resolves this many-to-many relationship.
- **Foreign Keys**:
  - product_id in ProductPromotion references product_id in Product.
  - promotion_id in ProductPromotion references promotion_id in Promotion.

Example: A promotion for "Free Shipping" (E.g: Grab, Shopee) could apply to several products, while a single product could be part of a "Buy One, Get One" and a "10% off" promotion simultaneously.

---

12. Customer and Cart:

- **Relationship**: One-to-One.
- **Explanation**: A customer can have multiple shopping sessions (carts), but each cart belongs to one customer.
- **Foreign Key**: customer_id in the Cart table references customer_id in the Customer table.

Example: A customer might browse and create a cart today and then start a new shopping session next week, creating a new cart.

---

13. Order and Payment:

- **Relationship**: One-to-Many.
- **Explanation**: An order can have multiple payments (e.g., partial payments, advance payment), but each payment is associated with a specific order.
- **Foreign Key**: order_id in the Payment table references order_id in the Order table.

Example: An order might be paid for in two installments, or one full payment.

---

14. Payment and PaymentHistory:

- **Relationship**: One-to-Many.
- **Explanation**: A payment can have multiple records of changes or statuses, but each change or status is tied to one payment.
- **Foreign Key**: payment_id in the PaymentHistory table references payment_id in the Payment table.

Example: A payment (E.g: Zalopay, Momo) might first be marked as "Pending" and later updated to "Completed" in the PaymentHistory table.

---

15. Customer and Notifications:

- **Relationship**: One-to-Many.
- **Explanation**: A customer can receive multiple notifications (e.g., for order updates, promotional messages), but each notification is tied to one customer.
- **Foreign Key**: customer_id in the Notifications table references customer_id in the Customer table.

Example: A customer might receive a notification when their order is shipped and another when a promotion is live.

---

16. Customer and LoyaltyProgram:

- **Relationship**: One-to-Many.
- **Explanation**: A customer has only one loyalty program membership, and each loyalty program membership belongs to one customer.
- **Foreign Key**: customer_id in the LoyaltyProgram table references customer_id in the Customer table.

Example: A customer can earn points from purchases and then redeem those points based on their loyalty program level (e.g., "Vip1" or "Vip2").

---

17. Promotion and ProductPromotion:

- **Relationship**: One-to-Many.
- **Explanation**: A promotion can apply to many products, and a product can be part of many promotions. This is managed by the ProductPromotion table.
- **Foreign Keys**:
  - promotion_id in ProductPromotion references promotion_id in Promotion.
  - product_id in ProductPromotion references product_id in Product.

Example: A "Black Friday, Mega Live" promotion could apply to various products, and a single product could be eligible for multiple promotions.

---

18. Order and Tax:

- **Relationship:** One-to-One.
- **Explanation:** Each order can have exactly one associated invoice, and each invoice corresponds to exactly one order. This relationship ensures that every order generates a unique invoice.
- **Foreign Key:** order_id in the Tax table references order_id in the Order table.

Example: Each order on Trung Nguyen Coffee Corp's e-commerce platform will have a unique associated tax record, ensuring accurate tax calculation and compliance with regional regulations.

---

19. Product and Rating:

- **Relationship:** One-to-Many.
- **Explanation:** Each product can have multiple ratings given by different customers, while each rating corresponds to exactly one product. This allows for a product to receive feedback from many users, providing a comprehensive view of its quality and performance.

- **Foreign Key:** product_id in Rating references product_id in Product

Example: Each “coffee sữa, coffee trứng” can receive multiple ratings from different customers, allowing for comprehensive feedback on quality and performance through user reviews.

---

20. PaymentMethod and PaymentHistory:

- **Relationship:** One-to-One.
- **Explanation:** Each payment method can have one corresponding entry in the payment history, which details the status and specifics of a transaction made using that payment method. This ensures that every payment method is associated with a unique historical record.
- **Foreign Key:** payment_method_id in the PaymentHistory table references payment_method_id in the PaymentMethod table.

Example: Each payment method is linked to a unique payment history entry, providing specific transaction details and ensuring accurate record-keeping for transactions.

---

21. Cart and Order:

- **Relationship:** One-to-Many.
- **Explanation:** Each cart can be converted into a single order upon checkout, and each order originates from one specific cart.
- **Foreign Key:** cart_id in the Order table references cart_id in the Cart table.

Example: Each cart can generate multiple orders over time, while every order is derived from a single cart, ensuring clear tracking of purchases.

---

22. Payment and PaymentMethod:

- **Relationship:** One-to-One.
- **Explanation:** A payment can be associated with a specific payment method, but each payment method can be used for multiple payments.
- **Foreign Key:** payment_method_id in the Payment table references payment_method_id in the PaymentMethod table.

Example: Each payment is linked to a specific payment method, while that payment method can facilitate multiple payments (E.g: cash, bank), allowing for streamlined transaction processing.

---

23. OrderItem and Discount:

- **Relationship:** One-to-Many.

- **Explanation:** Each order item can have multiple discounts applied to it, while each discount can be associated with many order items. This allows flexibility in managing how discounts are applied to specific items within an order.
- **Foreign Key:** order_item_id in the Discount table references order_item_id in the OrderItem table.

Example: Each order item can receive various discounts, and each discount can be applied to multiple order items, enabling dynamic pricing strategies and promotions.

---

24. OrderItem and OrderStatusHistory:

- **Relationship:** One-to-Many.
- **Explanation:** Each order item can have multiple status history entries associated with it, tracking its status changes throughout the order process (e.g., processing, shipped, delivered, canceled). Conversely, each status entry pertains to exactly one order item.
- **Foreign Key:** order_item_id in the OrderStatusHistory table references order_item_id in the OrderItem table.

Example: Each order item can have numerous status history entries, detailing its journey through various stages, while each status entry corresponds to a single order item.

---

25. Order and OrderStatusHistory:

- **Relationship:** One-to-Many.
- **Explanation:** Each order can have multiple status history entries associated with it, tracking changes to the order's status throughout its lifecycle (e.g., placed, processing, shipped, delivered, canceled). Conversely, each status entry pertains to exactly one order.
- **Foreign Key:** order_id in the OrderStatusHistory table references order_id in the OrderItem table.

Example: Each order can have multiple status history entries that document its lifecycle, with each entry related to a specific order, ensuring transparent tracking of order progress.

---

26. Customer and Shipment:

- **Relationship:** One-to-Many.
- **Explanation:** Each customer can have multiple shipments associated with their orders, while each shipment corresponds to exactly one customer. This relationship allows for the tracking of all shipments that a customer has received over time.
- **Foreign Key:** customer_id in the Shipment table references customer_id in the Customer table.

Example: Each customer can have multiple shipments (E.g: Bee, XanhSM) linked to their orders, with each shipment associated with a single customer, enabling effective tracking of all received orders.

SQL Queries:

```
CREATE TABLE Supplier (
supplier_id INT AUTO_INCREMENT,
supplier_name VARCHAR(255) NOT NULL,
contact_name VARCHAR(255) NOT NULL,
contact_number VARCHAR(15) NOT NULL,
PRIMARY KEY (supplier_id)
);

CREATE TABLE Category (
category_id INT AUTO_INCREMENT,
category_name VARCHAR(50) NOT NULL,
category_type VARCHAR(255) NOT NULL,
PRIMARY KEY (category_id)
);

CREATE TABLE Product (
product_id INT AUTO_INCREMENT,
supplier_id INT NOT NULL,
category_id INT NOT NULL,
price DECIMAL(10, 2) NOT NULL,
stock_quantity INT NOT NULL,
product_name VARCHAR(255) NOT NULL,
description TEXT NOT NULL,
PRIMARY KEY (product_id),
FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id),
FOREIGN KEY (category_id) REFERENCES Category(category_id)
);

CREATE TABLE ShippingCarrier (
carrier_id INT AUTO_INCREMENT,
carrier_name VARCHAR(255) NOT NULL,
tracking_url VARCHAR(255) NOT NULL,
carrier_rating INT CHECK (carrier_rating BETWEEN 1 AND 5) NOT NULL,
PRIMARY KEY (carrier_id)
);

CREATE TABLE Shipment (
shipment_id INT AUTO_INCREMENT,
order_id INT NOT NULL,
customer_address VARCHAR(255) NOT NULL,
carrier_id INT NOT NULL,
tracking_number VARCHAR(50) NOT NULL,
shipment_date DATETIME NOT NULL,
estimate_delivery DATETIME NOT NULL,
PRIMARY KEY (shipment_id),
```

```
FOREIGN KEY (carrier_id) REFERENCES ShippingCarrier(carrier_id)
);

CREATE TABLE OrderStatusHistory (
order_status_id INT AUTO_INCREMENT,
order_id INT NOT NULL,
status VARCHAR(50) NOT NULL,
total_amount DECIMAL(10, 2) NOT NULL,
PRIMARY KEY (order_status_id)
);

CREATE TABLE Customer (
customer_id INT AUTO_INCREMENT,
customer_address VARCHAR(255) NOT NULL,
customer_email VARCHAR(255) NOT NULL,
customer_phone VARCHAR(15) NOT NULL,
customer_name VARCHAR(255) NOT NULL,
registration_date DATETIME NOT NULL,
loyalty_points INT NOT NULL DEFAULT 0,
PRIMARY KEY (customer_id)
);

CREATE TABLE Order (
order_id INT AUTO_INCREMENT,
customer_id INT NOT NULL,
status VARCHAR(50) NOT NULL,
total_amount DECIMAL(10, 2) NOT NULL,
shipping_address VARCHAR(255) NOT NULL,
order_date DATETIME NOT NULL,
PRIMARY KEY (order_id),
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE OrderItem (
order_item_id INT AUTO_INCREMENT,
order_id INT NOT NULL,
product_id INT NOT NULL,
quantity INT NOT NULL,
total_purchase DECIMAL(10, 2) NOT NULL,
PRIMARY KEY (order_item_id),
FOREIGN KEY (order_id) REFERENCES Order(order_id),
FOREIGN KEY (product_id) REFERENCES Product(product_id)
);

CREATE TABLE Discount (
discount_id INT AUTO_INCREMENT,
product_id INT NOT NULL,
discount_amount DECIMAL(10, 2) NOT NULL,
```

```
discount_code VARCHAR(50) NOT NULL,
expiration_date DATETIME NOT NULL,
PRIMARY KEY (discount_id),
FOREIGN KEY (product_id) REFERENCES Product(product_id)
);

CREATE TABLE CustomerAddressInfo (
address_id INT AUTO_INCREMENT,
customer_id INT NOT NULL,
address_type VARCHAR(50) NOT NULL,
customer_address VARCHAR(255) NOT NULL,
city VARCHAR(100) NOT NULL,
state VARCHAR(100) NOT NULL,
zip_code VARCHAR(20) NOT NULL,
country VARCHAR(100) NOT NULL,
region VARCHAR(100) NOT NULL,
PRIMARY KEY (address_id),
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Rating (
rating_id INT AUTO_INCREMENT,
customer_id INT NOT NULL,
rating_comment TEXT NOT NULL,
rating_date DATETIME NOT NULL,
rating_image BLOB,
PRIMARY KEY (rating_id),
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Tax (
tax_id INT AUTO_INCREMENT,
tax_rate DECIMAL(5, 2) NOT NULL,
tax_name VARCHAR(255) NOT NULL,
effective_date DATETIME NOT NULL,
PRIMARY KEY (tax_id)
);

CREATE TABLE PaymentMethod (
payment_method_id INT AUTO_INCREMENT,
payment_method_name VARCHAR(255) NOT NULL,
payment_gateway VARCHAR(255) NOT NULL,
PRIMARY KEY (payment_method_id)
);

CREATE TABLE Cart (
cart_id INT AUTO_INCREMENT,
customer_id INT NOT NULL,
```

```
session_id VARCHAR(255) NOT NULL,
creation_date DATETIME NOT NULL,
status_update VARCHAR(50) NOT NULL,
PRIMARY KEY (cart_id),
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Payment (
payment_id INT AUTO_INCREMENT,
order_id INT NOT NULL,
payment_method_id INT NOT NULL,
payment_amount DECIMAL(10, 2) NOT NULL,
payment_date DATETIME NOT NULL,
PRIMARY KEY (payment_id),
FOREIGN KEY (order_id) REFERENCES Order(order_id),
FOREIGN KEY (payment_method_id) REFERENCES PaymentMethod(payment_method_id)
);

CREATE TABLE PaymentHistory (
payment_history_id INT AUTO_INCREMENT,
payment_id INT NOT NULL,
payment_date DATETIME NOT NULL,
payment_amount DECIMAL(10, 2) NOT NULL,
payment_method_name VARCHAR(255) NOT NULL,
payment_gateway VARCHAR(255) NOT NULL,
change_description VARCHAR(255),
payment_status VARCHAR(50) NOT NULL,
transaction_reference VARCHAR(50) NOT NULL,
PRIMARY KEY (payment_history_id),
FOREIGN KEY (payment_id) REFERENCES Payment(payment_id)
);

CREATE TABLE Notifications (
notification_id INT AUTO_INCREMENT,
customer_id INT NOT NULL,
created_date DATETIME NOT NULL,
notification_type VARCHAR(255) NOT NULL,
message TEXT NOT NULL,
PRIMARY KEY (notification_id),
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE LoyaltyProgram (
loyalty_program_id INT AUTO_INCREMENT,
customer_id INT NOT NULL,
points_earned INT NOT NULL DEFAULT 0,
points_redeemed INT NOT NULL DEFAULT 0,
membership_level VARCHAR(50) NOT NULL,
```

```
enrollment_date DATETIME NOT NULL,
PRIMARY KEY (loyalty_program_id),
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Promotion (
promotion_id INT AUTO_INCREMENT,
promotion_type VARCHAR(50) NOT NULL,
description TEXT NOT NULL,
discount_amount DECIMAL(10, 2) NOT NULL,
start_date DATETIME NOT NULL,
end_date DATETIME NOT NULL,
is_active BOOLEAN NOT NULL DEFAULT TRUE,
PRIMARY KEY (promotion_id)
);

CREATE TABLE ProductPromotion (
product_promotion_id INT AUTO_INCREMENT,
product_id INT NOT NULL,
promotion_id INT NOT NULL,
PRIMARY KEY (product_promotion_id),
FOREIGN KEY (product_id) REFERENCES Product(product_id),
FOREIGN KEY (promotion_id) REFERENCES Promotion(promotion_id)
);
```

Our team has integrated Jira with GitHub to streamline code storage and enhance collaboration. We can n ow track commits directly from Jira issues, making the workflow smoother and more efficient.

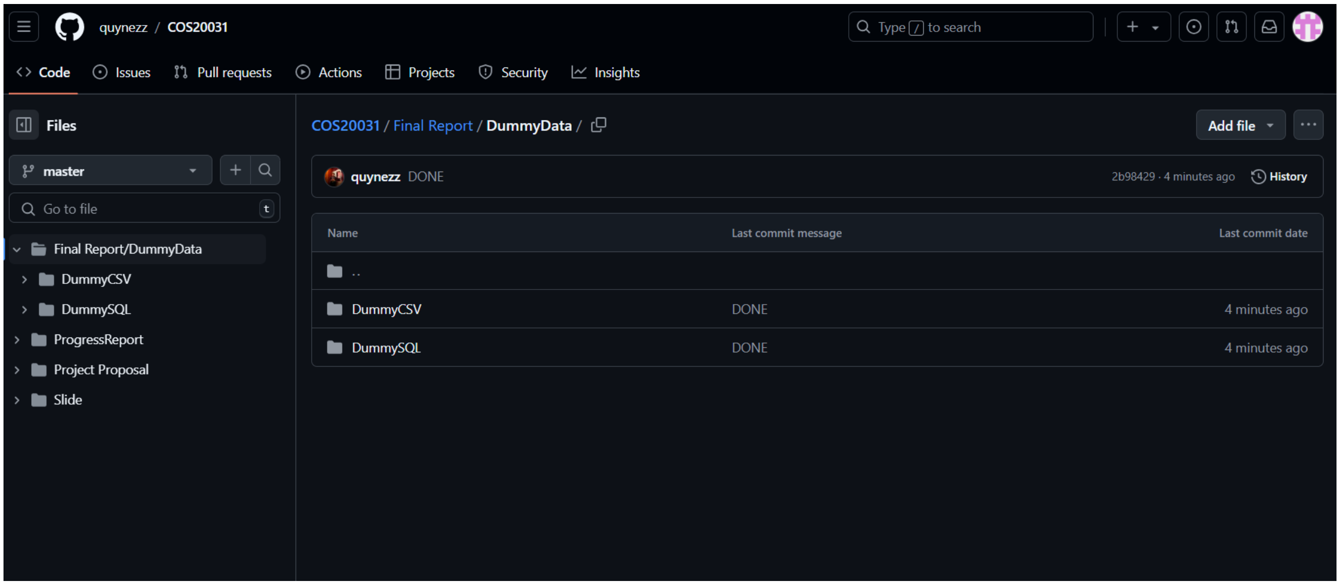
Link Github: https://github.com/quynezz/COS20031

🌹 Next steps

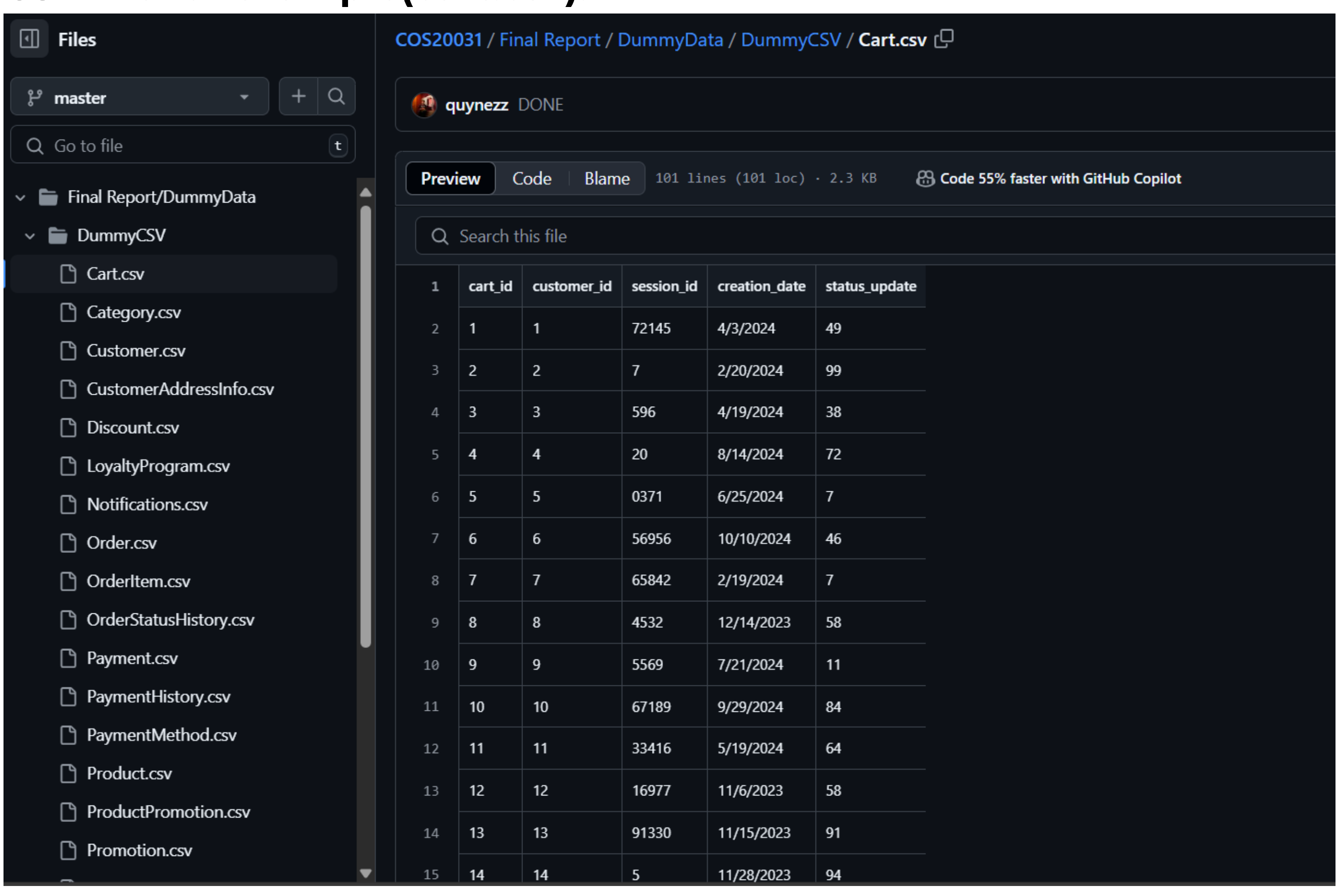
1. Important steps in the process of constructing an information system include creating databases with MySQL, creating ER diagrams, database normalization, and installing and configuring databases.
2. The data requirements of the system are defined using ER diagrams. Entities, properties, and relationships between entities in a system can all be described using ER diagrams.
3. The ER diagram serves as the basis for the database diagram. The database's structure is shown in the database diagram, which includes primary keys, foreign keys, and tables.
4. The practice of structuring data in a database to reduce redundancy and enhance integrity is known as database normalization. It involves five stages, from level 1 to level 5.
5. The final step in developing a database is installing MySQL and building the database. MySQL, a free relational database management system (RDBMS), is widely used in web and mobile applications.

Evidence on data creation (Dummy data for testing enviroment)

Storing testing database:



CSV file for example(cart.csv):



Sql impletation example (customer.sql):

First ten table:

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (1, '740 Ruskin Hill', 'dpfaffel0@google.cn', '900-791-4824', 'Danielle Pfaffel', '1/13/2024', 35);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (2, '97 Steensland Point', 'bosmar1@whitehouse.gov', '957-415-8461', 'Beryle Osmar', '8/16/2024', 86);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (3, '0 Messerschmidt Parkway', 'asterman2@instagram.com', '569-679-4092', 'Alfonso Sterman', '12/1/2023', 8);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (4, '55 Luster Alley', 'odonett3@g.co', '877-584-1261', 'Ortensia Donett', '1/12/2024', 52);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (5, '816 8th Road', 'aburbridge4@gnu.org', '185-769-8997', 'Ajay Burbridge', '6/27/2024', 98);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (6, '29711 Rutledge Lane', 'lwilloughway5@typepad.com', '141-979-5384', 'Leshia Willoughway', '1/30/2024', 78);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (7, '4 Riverside Hill', 'hvales6@yelp.com', '950-449-4698', 'Herculie Vales', '5/16/2024', 51);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (8, '864 Cody Center', 'hakenhead7@webs.com', '560-692-2807', 'Hubie Akenhead', '4/4/2024', 22);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (9, '9106 Banding Parkway', 'gcordes8@bluehost.com', '579-892-0727', 'Giordano Cordes', '8/16/2024', 56);

insert into Cart (customer_id, customer_address, customer_email, customer_phone, customer_name, registration_date, loyalty_points) values (10, '303 Little Fleur Pass', 'rdownton9@fastcompany.com', '680-380-5089', 'Reynard Downton', '2/20/2024', 68);

....

Use Cases and SQL Implementation

Usecase1: Customer Registration and Profile Management

Scenario: New online customer sign up and add theirs information in general and theirs address information in particular

```sql
INSERT INTO
 Customer ( customer_id, customer_name, customer_email, customer_phone, registration_date, loyalty_points )
VALUES ( 1001, 'Ngoc Quyen' , 'NQ123@email.com', '+1234567890' , CURRENT_TIMESTAMP, 0 );

 INSERT INTO
 CustomerAddressInfo ( address_id, customer_id, address_type, customer_address, city,
state, zip_code, country, region ) VALUES ( 1, 1001, 'shipping' , 'A35BachDang' ,
'Ho Chi Minh City' , 'MA', '02108', 'VietNam', 'Southern'  );
```

Explaination :

- A new customer named Ngoc Quyen is signing up for an account to receive fully access to the Trung Nguyen's Coffe D2C platform
- Ngoc Quyen have Customerid is 1001, following with others related information (Specific address, zip code ... etc.)
- Timestamps the registration
- Initializes loyalty program with 0 points

Usecase2: Product Browsing and Manage Shopping Cart

Scenario: Online customers add products to cart

```sql
INSERT INTO
 Cart ( cart_id, customer_id, session_id, creation_date, status_update )
VALUES ( 101, 1001, 'SESSION123' , CURRENT_TIMESTAMP, 'active' );

SELECT
 p.product_id, p.product_name, p.price, p.stock_quantity, c.category_name
FROM Product p JOIN Category c ON p.category_id = c.category_id WHERE
 p.stock_quantity > 0 ORDER BY c.category_name, p.product_name;
```

Explaination:

- A new shopping cart is created for CustomerID 1001
- assigns 101 for CartID
- Records the current shopping session
- Mark cart as ‘active’
- Records timestamp for cart creation
- Show all the only in-stock product

Usecase3: Ordering Process

Scenario: Customer places an order

BEGIN TRANSACTION;

-- Create order

```sql
INSERT INTO Order ( order_id, customer_id, status
, total_amount, shipping_address, order_date ) VALUES ( 5001, 1001, 'processing' ,
150.00 VND , '123 Main Street, Boston, MA 02108' , CURRENT_TIMESTAMP);
```

-- Add order items

```sql
INSERT INTO
 OrderItem ( order_item_id, order_id, product_id, quantity, total_purchase )
VALUES (1, 5001, 101, 2, 100.000 VND), (2, 5001, 102, 1, 50.000 VND);
```

-- Update product stock

```sql
UPDATE Product SET stock_quantity = stock_quantity - 2 WHERE product_id = 101;
UPDATE Product SET stock_quantity = stock_quantity - 1 WHERE product_id = 102;
```

-- Record order status

```sql
INSERT INTO OrderStatusHistory ( order_status_id, order_id, status , total_amount ) VALUES ( 1, 5001, 'processing' , 150.00 VND ); COMMIT;

COMMIT;
```

Explaination :

- Transaction command start new transaction to ensure all the operation inside it performed as a single unit of work. If any line fail, it rolls back.
- Places an order comes with 'Processing ' status cost 150.00 VND in total, following with the information of ordered customer
- Including all the orderitem detail (id, each product’s cost, ... etc)
- Updating all remaining product stock
- Record the Order detail in OrderStatusHistory consist of all the detail about the Order’s transaction.

Usecase4: Payment Processing

Scenario: Online payment for orders

```sql
INSERT INTO
 Payment ( payment_id, order_id, payment_method_id, payment_amount, payment_date )
VALUES ( 901, 5001, 1,150.00 VND , CURRENT_TIMESTAMP );

INSERT INTO
 PaymentHistory ( payment_history_id, payment_id, payment_date, payment_amount, pa
yment_method_name, payment_gateway, payment_status, transaction_reference )
VALUES ( 1, 901, CURRENT_TIMESTAMP, 150.00, VND,  'QR Scan', 'VnPay', 'completed'
, 'TXN123456' );
```

Explaination :

- Processing a Product payment and records it in PaymentHistory
- Each of the payment's detail included (processing time, total cost, payment method... etc)

Usecase5: Shipping and Order Tracking

Scenario: Order is shipped and tracking information is up-to-date

```sql
INSERT INTO
 Shipment ( shipment_id, order_id, customer_address, carrier_id, tracking_number,
shipment_date, estimate_delivery )
VALUES ( 701, 5001, ‘A35 Bach Dang’, MA 02108', 1, 'TRACK123456
', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP + INTERVAL '3 days' );

UPDATE OrderStatusHistory SET status = 'shipped' WHERE order_id = 5001;
```

Explaination :

- Tracking shipping product directly
- Each of the product has it own’s shipping detail (shipment id, order id, online customer address, tracking number to look up,... etc)
- After done processing “Shipping” to “Shipped”, it record the Product Status in OrderStatusHistory with order id.

Usecase6: Customer Reviews and Ratings

Scenario: Customer leaves a product review

```sql
INSERT INTO Rating ( rating_id, customer_id, rating_comment, rating_date ) VALUES
 ( 401, 1001, 'Great coffee! Fast shipping and excellent quality'
, CURRENT_TIMESTAMP );
```

Explaination :

- Customer will express their feelings and based on the levels, they will leaves an rating comes with a Compliments or Criticsm based on the product enjoyment levels.

Usecase7: Promotion and Discount Application

Scenario: Apply promotion and discount to specific products

```sql
INSERT INTO Promotion ( promotion_id, promotion_type, description
, discount_amount, start_dat , end_date, is_active ) VALUES ( 301, 'seasonal' ,
'Summer Coffee Sale' , 15.000 VND, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP +
 INTERVAL '30 days', true );

INSERT INTO ProductPromotion ( product_promotion_id, product_id, promotion_id )
VALUES (1, 101, 301), (2, 102, 301);

SELECT
 p.promotion_type, p.description, p.discount_amount, pr.product_name, pr.price,
(pr.price
- p.discount_amount) as discounted_price

 FROM Promotion p

JOIN ProductPromotion pp ON p.promotion_id = pp.promotion_id

JOIN Product pr ON pp.product_id = pr.product_id

WHERE p.is_active = true

     AND CURRENT_TIMESTAMP BETWEEN p.start_date AND p.end_date;
```

Applying promotiong for coffe product based on each of the seasonal (Summer as an example), following with total discount amount ,starting date, ending date

Indexes

Product Table

- supplier_id: Foreign key, useful for joining with Supplier.
- **Index on** supplier_id:
    - **SQL Statement**:

```sql
CREATE INDEX idx_product_supplier ON Product (supplier_id);
--code9
```

    - **Reason to create**: This index is beneficial when querying products by their suppliers. In an e-commerce platform, users often filter products based on the supplier, so this index can speed up those queries (Famous suppliers with reputation)
    - **Reason to drop:** If it's determined that the supplier_id index is not utilized frequently (e.g., suppliers are rarely queried or searched by products), consider dropping it:

```sql
DROP INDEX idx_product_supplier ON Product;
```

- **Index on** category_id:
    - **SQL Statement**:

```sql
CREATE INDEX idx_product_category ON Product (category_id);
```

    - **Reason to create**: Products are often displayed by category. An index on this column improves performance when querying for products belonging to specific categories.
    - product_name: Frequently searched attribute.

- **Index on** product_name:
    - **SQL Statement**:

```sql
CREATE INDEX idx_product_name ON Product (product_name);
```

    - **Reason to create**: If customers frequently search for products by name, this index will enhance the performance of those search queries.
    - price: If products are often filtered or sorted by price.

- **Index on** price (optional):
    - **SQL Statement**:

```sql
CREATE INDEX idx_product_price ON Product (price);
```

    - **Reason to create**: If there are queries that filter or sort products by price, having an index on this column can improve performance, particularly in situations where products are displayed in price ranges.

Order Table

- **Potential Indexes**:
  - order_id (PK): Primary key, already indexed.
  - customer_id: Foreign key, essential for joining with Customer.
  - **Index on** customer_id:
  - **SQL Statement**:

```sql
CREATE INDEX idx_order_customer ON Order (customer_id);
```

- **Reason**: This index improves performance when querying orders by customer, which is a common operation, especially in e-commerce applications.
  - status: Commonly used for filtering orders.
  - **Index on** status
  - **SQL Statement**:

```sql
CREATE INDEX idx_order_status ON Order (status);
```

- **Reason to create**: If orders are frequently filtered by their status (e.g., "shipped," "pending," "canceled"), this index can speed up those queries.
- **SQL Statement**:

```sql
DROP INDEX idx_order_status ON Order;
```

- **Reason to Drop**: For example, if after some analysis, it's found that the status column is rarely queried, it might make consideration on dropping the index:

- order_date: If frequently queried to filter orders by date.
- **Index on** order_date:
- **SQL Statement**:

```
CREATE INDEX idx_order_date ON Order (order_date);
```

- **Reason**: Many queries might need to filter or sort orders by their date. An index on this column can enhance the efficiency of such operations.

Major-Specific Work

CyberSecurity

I'm **Lau Ngoc Quyen**, a dedicate student with a strong passion for cybersecurity and working with database. My specific contribution to group project is preventing cyber threats and preventing threats to sensitive data.

To secure a Direct-to-Consumer (D2C) coffee platform against various web attacks, it's essential to integrate security measures throughout our future platform's architecture and development processes. Here's a breakdown of best practices to mitigate each of the attacks that our group member have discussed:
First, let define some of the most dangerous web-attack that hackers could use and prevention follow-up:

SQL injection
Using Parameterized Queries: Always use prepared statements or ORM (Obeject-Relational Mapping) libraries that enforece parameterized queries to prevent SQL injection
Input Validation: Sanitize and validate all user inputs, especially for the fields that interact the project's database
Minimal Privilege Access: Configure and assign a specific database accouints with the minimum permissions necessary

Command Injection
Use Whitelisted Commands: Only allow specific, sanitized commands if using shell commands.
Avoid Shell Calls: Avoid direct shell access unless absolutely necessary, and consider using library functions.
Sanitize Input: Ensure user inputs are strictly validated and sanitized

Man-in-the-Middle (MitM)
Use TLS/SSL: Encrypt all traffic between the server and clients using HTTPS.ds.
Secure Configuration: Configure SSL/TLS securely with strong cipher suites.
Public Wi-Fi Warnings: If possible, provide and educate users with security tips about using your site securely, especially on public networks.

Denial of Service (DoS) and Distributed Denial of Service (DDoS)
Rate Limiting: Limit the number of requests allowed per IP in a given time period.
Use a WAF and CDN: A Web Application Firewall (WAF) can block malicious traffic, while Content Delivery Networks (CDNs) help handle sudden traffic spikes.
Application-Level Caching: Use caching for common, unchanging pages to reduce server load.

For Users:

1. We will encourage users to create a strong and long enough passwords, in order to avoiding reusing password across sites.
2. Enable Two-Factor Authentication (2FA): Offer 2FA options, such as SMS re-sending codes or authenticator apps like : Authenticator, ...
3. Avoid Public Wi-fi for Sensitive Actions: Educate users to avoid using public netword to access accounts. Encourage using a VPN or other secure network when entering personal information (bank account mostly)

Software Development

I’m **Nguyen Thuan Khang**, along with me is **Le Ba Tung.** We are both enthusiatic students with a keen in front-end development. With our former knowledge and experiences of various front-end frameworks, Our collaborative spirit and technical expertise will play a crucial role in creating an engaging and user-friendly interface that showcases the brand's offerings and enhances the customer experience.

Framworks and Tools that we mainly use:

1. **React:** which is a Javscript library for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components and manage the application state efficiently.
2. **Figma:** A collaborative design tool used for creating user interfaces and prototypes. Figma allows teams to design, prototype, and iterate on UI/UX designs in real-time, enhancing collaboration.
3. **Postman: Function:** An API testing tool that simplifies the process of developing and testing APIs. It provides an intuitive interface for sending requests, inspecting responses, and automating tests.

Additional Tools

1. **CSS Preprocessors (Tailwind CSS):** Extend CSS with features like variables, nested rules, and mixins, making stylesheets more maintainable and organized.
2. **Version Control (e.g., Git):** A system for tracking changes in source code during software development, enabling collaboration among multiple developers.

AI and Data Analytics

I'm **Doan Nhat Long**, and I'm excited to share my contributions to our Direct-to-Consumer coffee platform project, focusing on AI and data analytics.

Personalized Recommendation Engines:

- **Tool**: **Amazon Personalize**
- **Function**: Uses machine learning algorithms to analyze customer preferences and behavior, offering personalized coffee recommendations based on past purchases and browsing history.

Chatbots for Customer Support:

- **Tool**: **Dialogflow** or **ChatGPT**
- **Function**: Provides instant responses to customer inquiries, assisting with order tracking, product information, and FAQs, improving customer satisfaction and engagement.

Predictive Analytics:

- **Tool**: **Google Analytics with AI features** or **Tableau**
- **Function**: Analyzes historical sales data and customer behavior to predict future trends, helping to optimize inventory and marketing strategies.

Dynamic Pricing Models:

- **Tool**: **Zaius** or **PricingAI**
- **Function**: Uses AI algorithms to adjust prices based on demand, competitor pricing, and inventory levels, ensuring competitive pricing strategies.

Retrospective

📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the 4Ls Retrospective Play.

Team	Team QLTK
Team members	@Lau Ngoc Quyen (Swinburne DN) @Doan Nhat Long (Swinburne HCM) @Le Ba Tung (Swinburne HCM) @Nguyen Thuan Khang (Swinburne HCM)
Date	Nov 4 2024

💭 4Ls retrospective

Project's Member	**Milestones**	**Loved**	**Longed for**	**Loathed**	**Learned**
Nguyen Thuan Khang	04/09/2024 - 15/11/2024	I enjoy group collaboration because it generates diverse ideas, makes projects more dynamic, and motivates us to keep improving.	I hope that in a teamwork, every perspective will be acknowledged and assessed objectively to determine how it can contribute to improve the project.	It can be challenging to understand, synthesize, and apply the vast amount of knowledge we acquire.	I learned how to use Confluence effectively to manange the project and Jira to keep track with the timeline and missions
Lau Ngoc Quyen	04/09/2024 - 15/11/2024	I enjoyed researching the problem we've been facing.	I would like to have more materials suitable for our project.	It's very difficult to understand all the materials and apply them to match our topic.	I learned how to manage the backend of the project.

Le Ba Tung	04/09/2024 - 15/11/2024	I love to support my teammates and team leader in solving problems that arise on the project.	I hope to learn more about the project so that I can gain more experience for my future work.	disagreement during discussion	I learned how to design and organize the project to ensure tasks are divided efficiently and logically.
Doan Nhat Long	04/09/2024 - 15/11/2024	I love working with databases and analyzing what we can work with.	I would love to improve our data analysis skills.	learning how to manage a variety of different data.	Organizing and understanding the given data, and using it efficiently.

⚡ Action plan

Action	**Owner**	**Due date**	**Action items**
Confluence Pages and Team Management	@Le Ba Tung (Swinburne HCM) @Nguyen Thuan Khang (Swinburne HCM)	Nov 4, 2024	All the confluence pages, make the arrangement to preficent team-work.
Use cases and SQL statements, transactions, Performance (indexes)	@Lau Ngoc Quyen (Swinburne DN)	Nov 7, 2024	SQL Implementation both creation and testing for future web-platform
Document on data creation and null values	@Doan Nhat Long (Swinburne HCM)	Nov 10, 2024	Document and generating dummy data for testing enviroments