System Architecture Design - Week 11-12
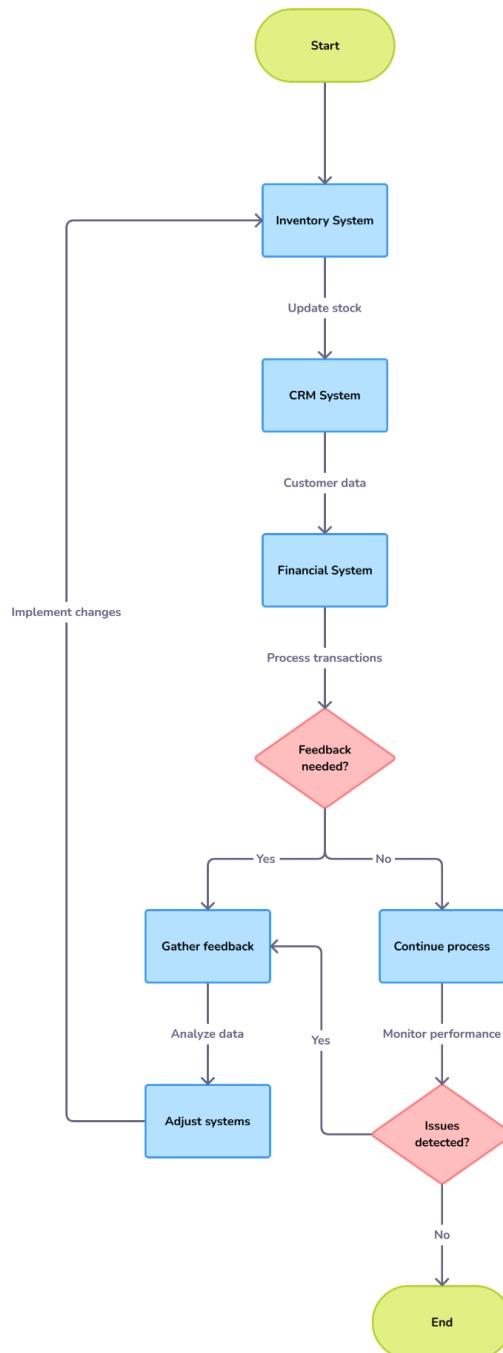
Group Members:

Allister, Christopher, Imam, Ryufath

System Architecture Diagram

System Boundaries

The scope of the system is defined as the following:

1. Inventory Management Boundary: Responsible for managing and tracking stock, supplies, tools, and assets.
2. CRM Boundary: Manages customer interactions, sales data, and customer support information.
3. Financial Boundary: Manages financial data including invoices, payroll, and daily accounting.
4. Integration Platform Boundary: Acts as a central hub for all data transactions, transforming and routing data between systems.

Subsystems and Components

The system will include the following subsystems and components:

1. Inventory System
   a. Inventory Database: Database that records the current stocks and items inventory
   b. Supply management: A system to monitor incoming stocks and items that were ordered.
   c. Order Management: A system to check the stock availability when a customer is ordering and set aside the ordered item from availability.
   d. Supply Chain Log: Keeping track of when new supply enters the inventory.
2. CRM System
   a. Transaction record: Recording transactions for accounting and investigating issues regarding a certain transaction.
   b. Customer database: Record existing customers for personalized advertisement and for the case of emergency such as possible malicious customers.
   c. Customer support: To help respond and receive feedback from customers who are experiencing issues. It may be using a chatbot or manually.
   d. Marketing Automation: Automatically distributes advertisements according to the parameters set by the shop.
3. Finance System
   a. Accounting: Keeping track of current money, income, and expenses.
   b. Payroll: Preparing money to pay the company's employees.
   c. Budgeting: Separating the money for buying assets, materials, resources, etc.
   d. Financial report generator: The system would make a report based on the history of the company's money.

Interfaces, Communication Protocols, and Data Exchange Mechanism

1. Interface

Each subsystem are able to interact with each other through interfaces such as:

- Inventory-CRM Interface: API endpoints that support bidirectional data transfer, enabling inventory updates based on CRM sales status and vice versa.
- CRM-Financial Interface: API endpoints for customer and billing information to sync, reducing duplicate data entry across systems.
- Inventory-Financial Interface: API connections to automatically log transactions when stock is adjusted.

2. Communication Protocols

The protocols that will be utilized are:

- RESTful APIs: Used for synchronous requests because REST APIs allow real-time data sharing between systems such as for updating inventory after a sale.
- Webhooks and Event-Driven Architecture: To support asynchronous data flows and reduce latency, the system can use webhooks to trigger updates in response to events (e.g., inventory updates when a sale is made).
- Message Queueing: For high-volume or high-frequency data flows, message queuing ensures reliable delivery without overwhelming the network.

3. Data Exchange Mechanism

In order to conduct data exchange, the system will require:

- Data Transformation and Normalization: Since data formats may vary (e.g., different naming conventions for customer IDs), data transformation tools in the integration platform will ensure consistency. This includes converting data formats, applying standardized schemas, and error-checking.
- ETL (Extract, Transform, Load) for Historical Data: For initial data migration, an ETL process can transfer and transform existing records into a compatible format before integration.
- Data Sync Frequency and Real-Time Updates: Key data, such as stock levels and customer information, will sync in real-time, while other data, like daily financial summaries, may update at scheduled intervals to reduce system load. The data exchange between CRM and inventory is based on stock availability. During the transaction, the CRM will be making the bills for the system. It will send the bill to the finance system to wait until the customer pays off their order. The finance will then inform the CRM that the payment is done to confirm order completion. It will also inform the inventory to decrease the stock according to the order. So, it is needed to keep data in real time