LONDON METROPOLITAN UNIVERSITY

informatics college • pokhara

CS5002NP & Software Engineering

Level 5

Coursework - 1

Semester

Assignment Due Date: 9th January, 2025

Assignment Submission Date: 9th January, 2025

Submitted To: Sandeep Gurung

Student Name	London met ID
Jonsen Gaire	23048808
Kaushal Thapa	23048815
Prabesh Marasini	23048854
Beeju BK	23048775
Manoj Neupane	23048832

*I confirm that I understand my coursework needs to be submitted online via **My Second Teacher** under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

Table of Contents

Table of Figures

Table of Tables

1. Introduction:

The Global Tech Corporation is trying to upgrade their warehouse operations by implementing a next-generation Inventory management System (IMS) in Nepal. In the previous project, due to significant challenges, it hampered the operational efficiency and overall success of the project. Due to the lack of clearly defined scope and proper documentation the previous project didn't go as planned and suffered failure. Not defining the project clearly and documentation created confusion regarding system features, integration points, and end goals. It resulted in poor resource allocation, causing delays and budget overruns with no regular meetings or progress reports to ensure the objectives of the project.

Due to their past failure and experiences, The Global Tech Corporation is now trying to implement a new IMS that emphasizes smooth project management. The new IMS will incorporate analysis and design ensuring clear project goals. The new IMS system will also focus on improving communication among team members and it will try to create more meetings among team members for better understanding of the project. The Global Tech Corporation aims to create a more reliable and efficient system which is operational and enhances user experience and satisfaction.

This report will detail the Software Requirement Specification which includes Functional Requirements and Non-Functional requirements. Not only SRS, but it will also detail the Data Flow Diagram which shows the flows between the processes and data. The Functional requirements define the feature and design a system should offer to the end user. The Non-Functional requirements define the security, flexibility, maintainability, portability, performance etc. requirements that a system should have.

2. Business Case

2.1 Executive Summary

Global Tech Corporation is looking to implement a modern Inventory Management System (IMS) primarily focusing on enhancing warehouse operations in Nepal. This project continues the previous unsuccessful attempt that highlighted significant issues like critical deficiencies, including unclear project scope, unclear documentation, poor team management, poor resource allocation, and unstructured communication plan among teams. The lack of structured project management led to budget overruns, operational delays, and ultimately, a failed implementation that left the warehouse struggling with inefficiencies and customer dissatisfaction.

2.2 Goals

This project aims to successfully develop a modern Inventory Management System (IMS) by addressing past deficiencies that led to a failed attempt and solving them. This project plans to focus on proper budget planning with proper risk management strategy along with employee training. The IMS will focus on enhancing warehouse operations, improving customer satisfaction, also its sustainability and scalability for future growth.

2.3 Objectives

- Create a proper documentation of all requirements to avoid uncertainty.
- Creating more specific features and functionality for customers.
- Defining end goals more clearly and concisely.
- Preparing proper communication plan among team members to avoid conflict and uncertainty.
- Identify potential risks to mitigate system failure and technical errors.
- To avoid past mistakes by estimating proper resource allocation and budget plan.
- A time-to-time survey should be held to identify problems and improve accordingly.

- Reducing manual efforts and implementing automation to operate the project more efficiently.
- Improving and maintaining customer satisfaction by frequent interaction.
- Minimizing and distributing workload during peak hours.

2.4 Scopes

- Implementing user registration and login system for users and admin panel.
- Purchase functionality for the items, products and for history of purchases.
- Purchase reports to track new inventory acquisitions, detailing quantities, costs, and suppliers for customers.
- Sales report to facilitate decision making for admins of total sales, quantity to identify profit or loss made during different period of time.

2.5 Problem Statements

- There was no clear scope without any documentation outlining any of the specific features of the system.
- The resources were not allocated properly which led to budget overrun and delays in other critical warehouse operations.
- The project was not managed properly.
- There seemed to be a lack of communication between team members with no regular meetings or progress reports.
- No review or evaluation were conducted to ensure whether the project objectives were met or not.

2.6 Proposed Solutions

- The scope needs to be clearly mentioned followed with proper documentation outlines for the specific features of the system.
- The resources should be allocated properly to every stakeholder.
- Right candidate for the project management should be chosen and given the responsibility for the best project management.
- The meetings should be conducted on allocated times with proper subjects, agendas and their discussions.
- After every milestone is completed, an evaluation or review should be conducted to ensure the best possible outcome of the objectives.

2.7 **Estimated Costs:**

Table 1: Table of Estimated costs

Description	Estimated Costs	Comments
Development Costs	NPR 5,00,000	Costs for software design, development, and coding.
Testing Costs	NPR 2,00,000	Testing to ensure system functionality and security purposes
Deployment Costs	NPR 50,000	Migration of code, database and backend to the deploy platform(i.e. AWS)
Maintenance Costs	NPR 4,50,000	It covers regular updates, bug fixes, and technical support.
Documentation Costs	NPR 70,000	Creations of detailed project with explanation and requirements to cover the project scopes.
Total Costs	NPR 12,70,000	Total costs for overall project

2.8 **Risks Mitigating Plans:**

Table 2: Table of Risk Mitigating Plans

Description	Mitigation Plans
Development	Regular Inspection and resource allocation in design and development of software.
Testing	Ensuring early identification and resolution of issues with different risk management methods and experienced, skilled testers.
Deployment	Make sure everything is deployed properly and check if there is server unable due to improper configurations.
Maintenance	Regular updates, bug fixes, and technical support.
Documentation	Create a dedicated team to ensure proper alignment of project documentation.

2.9 TimeLine

Deadline: 2024-11-26 to 2025-12-01

2.10 Assumptions

1. **User Assumptions**

- Any user can register to become an admin or a customer.
- A user must be registered so that they can make purchases.
- Our system will have only an admin user.
- The details about the admin will be stored in a different table.

2. **Payment Assumptions**

- Both admin and customers can make payments and they have two different processes for making the payments, i.e. purchase payment for admin and product payment for customers.
- External entity payment provider will be integrated in the system for the convenience to the users while making payment.

3. **Report Preparation Assumptions**

- System generates report on the basis of user: sales report for admin or product report for customers.
- The system provides functionality of filtering and searching by date.

4. **Product Management Assumptions**

- The Admin will add and update products into the system.

5. **Purchase Order Assumptions**

- The Admin will make the purchases.
- The Admin will have privilege to place or cancel purchase order.

6. **Sales Management Assumptions**

- The system will automatically check for availability of stock before purchase.

2.11 Benefits

- Reduction in manual works which results in operational efficiency.

- With its enhanced functional operations and smooth performance, it will

increase the profit by 18% within 3 months from the implementation.

- Allocating resources efficiently optimizes productivity of workforce and reduces costs.
- Reduction in inefficiencies, operational delays increases customer satisfaction.
- Correct tracking of inventory reduces the risks of overstock and stock out.
- Creation of centralized and well managed databases leads to improvisation of decision making and marketing strategies.

2.12 **Recommendation**

- Set Clear Milestones which is attainable and also helps to track the progress of ongoing project.
- Conduct meetings with stakeholders after each milestones which helps to mitigate the small issues from turning into bigger problems.
- Establish a support team which ensures to keep the system running smoothly and addresses any issues quickly.
- By having a clear financial plan, it will make sure IMS is a cost effective solution and adds beneficial returns to the business than previous system.

3. SRS (Software Requirement Specification):

A Software Requirements Specification (SRS) is a document that explains or describes about the functionality of the expected software. It explains what the software will do to fulfill the requirements of the stakeholders (Krüger, 2023).

This Software Requirements Specification (SRS) document outlines the requirements for the Inventory Management System (IMS) developed by Global Tech Corporation. The purpose of this document is to provide a comprehensive description of the system's functionalities, constraints, and performance expectations. The IMS aims to streamline warehouse operations, enhance inventory tracking, and improve decision-making processes for both administrators and customers.

Requirements that reduce the development time and cost

1. Meeting regularly with customers and stakeholders to gather and organize all the requirements in clear way.
2. Creating model or prototype in an effective way to explain expectations early in a project.
3. Using automation and advance tools to generate code faster, to test the software faster.
4. Right candidate for the project management should be chosen and given the responsibility for the best project management.
5. The meetings should be conducted on allocated times with proper subjects, agendas and their discussions.
6. After every milestone is completed, an evaluation or review should be conducted to ensure the best possible outcome of the objectives.
7. Providing training to developers for secure coding to avoid weakness of code.

3.1 Functional Requirements:

1. Access users/admins to the system

- **User Registration:**
  - Users (Admin and buyers) should be able to register with their details (i.e**.** name, phone, Gmail) in the system.

- User entered password needs to encrypted using strong encryption (i.e. hashing, md5) at rest and transit.
- Users entered data must be checked in database before registering as a new user/admin.
- There should be a facility for verifying the mail through OTP to ensure data accuracy**.**
- Different database anomalies must be removed, and database must be well designed to prevent data duplication while registering.

- **User Login:**
  - Registered users must be logged in using their details (i.e. Email. password) to access their specific roles within the system.
  - Features must be separated as per role (admin/customer) of the logged in user.
  - 2FA (Two-factor authentication) is needed to ensure security of the user's account.
  - "Forgot password" feature should be added in case users might want to register new password if they have forgotten previous password.
  - Users must be able to keep and track their login history.

2. Purchase Order:

- **Add Purchase:**
  - Users must be able to record new purchases from suppliers by choosing product details, quantities, costs, and supplier information.
  - The data entered must be validated correctly to ensure correct data is entered.
  - **"**Search and filter functionality **"**of the product for the convenience of the user.

- **View Purchase:**
  - User must be able to see past purchases made by the user from a particular supplier with different details like (date, quantities, and costs).

3. **Generate Report:**

- **Purchase Report:**
  - The system must generate reports detailing new inventory acquisitions, including quantities, costs, and suppliers for customers**.**
  - "Search and filter by date functionality" of the purchase history for the convenience of the user.
- **Sales Report**
  - Admin must be able to generate sales reports that provide information on total sales, quantities sold, profits, and losses over specific time periods.

4. **Sales Report:**

- Users must be able to view all sales orders, along with their status (completed or pending).
- Track the delivery method of each orders with delivery timelines, contact information, address and dispatching details.

5. **Product Management:**

- **Add Product:**
  - Admins must be able to input new products into the inventory system with details such as product name, description, price, and stock quantity.
- **View Product:**

- Users must be able to access current stock levels, sales history, and supplier data for products.

6. Payment:

- Customer must have options to compare prices between similar items which helps in decision making for users.
- Customer must be able to securely make payment of the items with integrated payment in the system.
- Customer payment data must be securely stored.

3.2 Non-Functional Requirements

Non-functional requirements are the minimum requirements for quality software. It defines the quality attribute of a software system. They are essential to ensure the usability and effectiveness of the entire system. (Charles, 2024).They judge any software on the basis of security, usability, portability and responsiveness. (Moore, 2024).There are various types of non-functional requirements such as: scalability, availability, reliability, data integrity, and so on. (Moore, 2024).

3.2.1 Design and Implementation Constraints

Design and implementation constraints are the restrictions, limitations or dependencies that have impact on the software solution. (Global, 2024)

- **Usability Constraints**

  The User Interface should fulfill the basic requirements for usability by any human. Let's take Web Content Accessibility Guidelines (WCAG) as an example, it gives us a guideline how should the web contents be developed. (HubSpot, 2021)Similarly, our software should be able to be completed suitable to use by any mere user.

- **Security**

  The credentials of the users are to be recorded, give permissions, records of the goods and products are also needs to be recorded, and the most importantly, the payment credentials of the users will be passed.

Hence, the software shall have end-to-end data-encryption with standard encryption protocols such as: AES, Blowfish, and so on.

- **Budget Constraints**

  The software should be developed and deployed under the budget allocated by the corporation.

- **Time Constraints**

  The software should be developed within the specified time, followed by tracking every progress to the milestone.

- **Reliability Constraints**

  The software must have as minimum error as possible. It should handle and recover the error within very short period of time.

- **Performance Constraints**

  Under normal operating conditions, the software should respond to the user under 2 seconds. Also, it should be able to handle minimum 1000 users using simultaneously without any performance degradation.

3.2.2 External Interface Required:

- **User Interfaces:**

  The system's online interface should be user-friendly and responsive, allowing easy access to features like forums, community activities, plant sales, and certification courses.

- **Hardware Interfaces:**

  The system should be compatible with common devices, including desktops, laptops, tablets, and smartphones.

- **Communication Interface:**

  The server and user devices should communicate using secure communication channel to ensure security. It should use HTTPs protocol.

- **Software Interfaces:**

  The system should integrate with secure online payment gateways to handle transactions.

3.2.3 Other Non-Functional requirements:

There are other non-functional requirements such as: data scalability, access control, quality coding, network processing time, system management, system design architecture, maintenance, delivery, development standards, and de-ployment and timing requirements.

4. Group Tasks

4.1 Environmental model specification

4.1.1 Data Flow Diagram (DFD):

A Data Flow Diagram (DFD) is a visual tool that uses specific symbols to show how data moves within a business. It helps illustrate the operations of a business by focusing on data flow. DFDs are often used in system analysis and design to help understand and track how data moves and how processes work within a system (Robinsom, 2024).

- Components of DFD are as follows:
  - Data Flow
  - Data Stores
  - Processes
  - External Entities

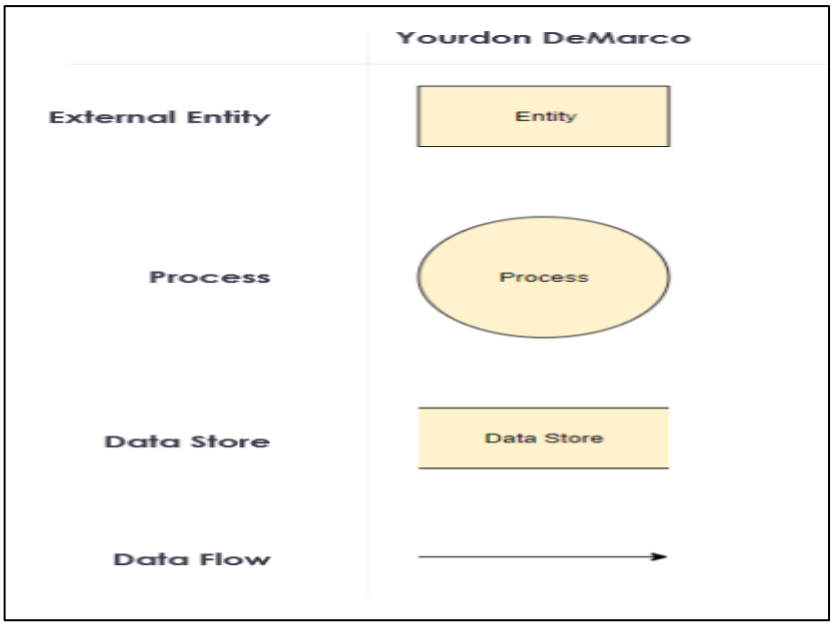


Figure 1: Components of DFD

4.1.2 Context Level Diagram / Data Flow Diagram (DFD) Level 0:

A Level 0 Data Flow Diagram (DFD), also known as a Context Level Diagram, visually represents the system or process being analyzed. It provides an overview of the system's boundaries, external entities, and the main interactions between them. As the most abstract level in a DFD hierarchy, a context level diagram helps to achieve a clearer and more complete understanding of the system (Lindemulder, 2024).

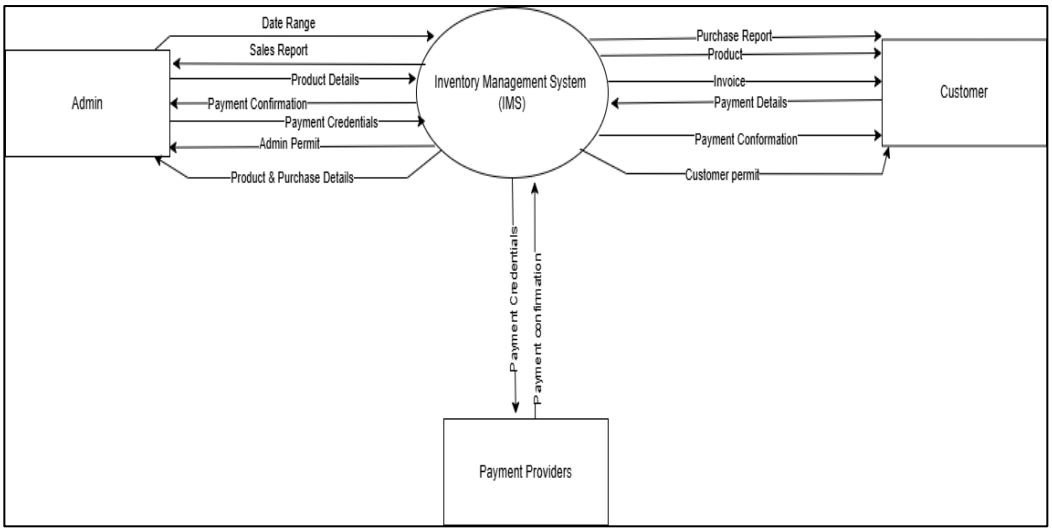


Figure 2: DFD level 0 of inventory management system

4.1.3 Data Flow Diagram (DFD) Level 1:

DFD Level 1 gives a clearer and more detailed look at the system than the higher-level diagrams. It takes the main processes, data stores, data flows, and external entities from the context diagram (Level 0) and expands on them. This level breaks down the main functions of the system into smaller parts, showing how data moves and is processed. The main focus of Level 1 DFD is on how these smaller processes connect with each other and the data flows between them (Lindemulder, 2024).

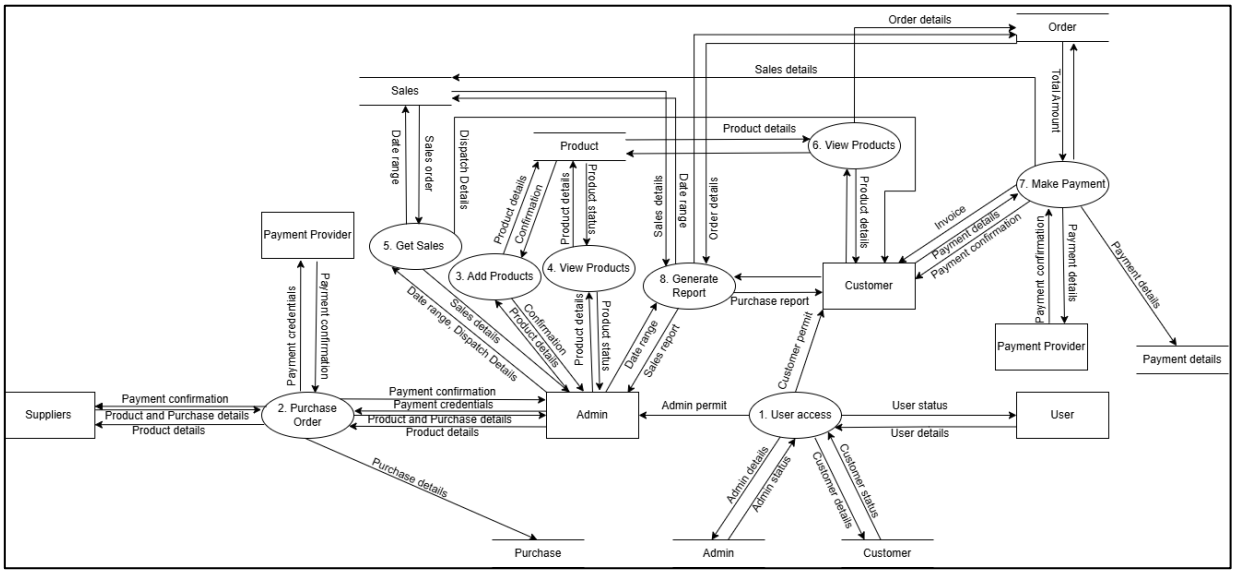


Figure 3: DFD of Inventory Management System (level 1)

4.1.4 Data Flow Diagram (DFD) Level 2:

A Level 2 Data Flow Diagram (DFD) provides a more detailed examination of a specific process or sub process within a system, building on the insights gained from a Level 1 DFD. It enhances understanding of how data flows and interacts among various components of the process. This diagram breaks down the processes identified in the Level 1 DFD into intricate data flows, data storage, and sub processes, allowing for a deeper exploration of the system's internal operations (Lindemulder, 2024).

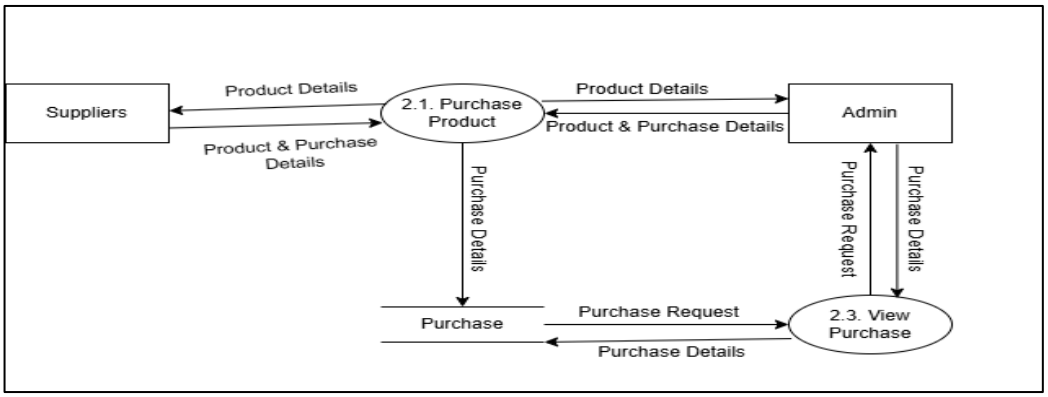


Figure 4 DFD-2 of purchase order

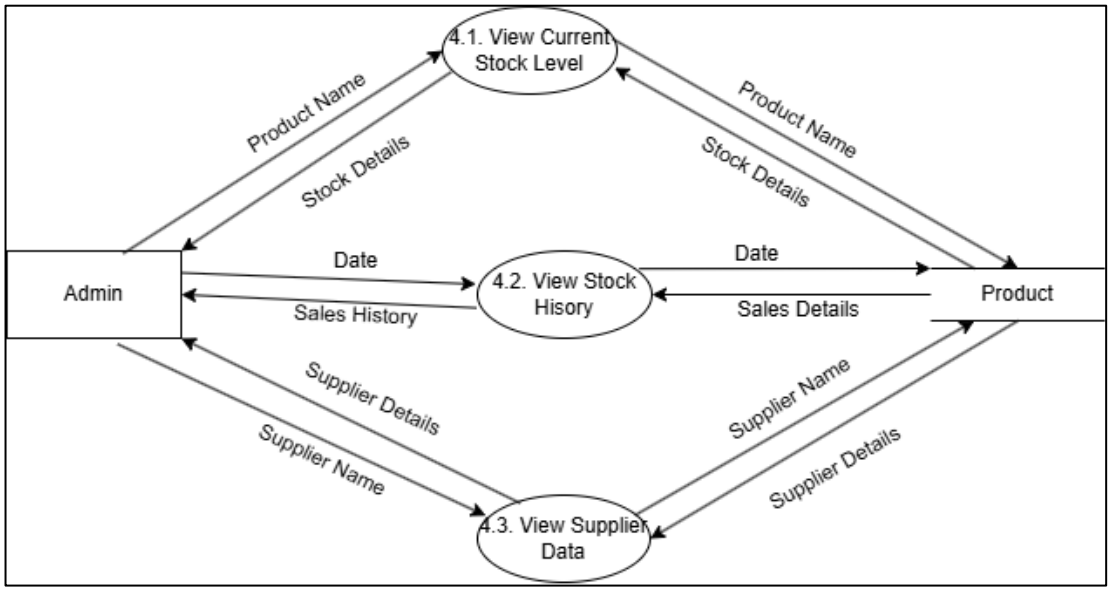


Figure 5: DFD-2 of Real-time stock Update

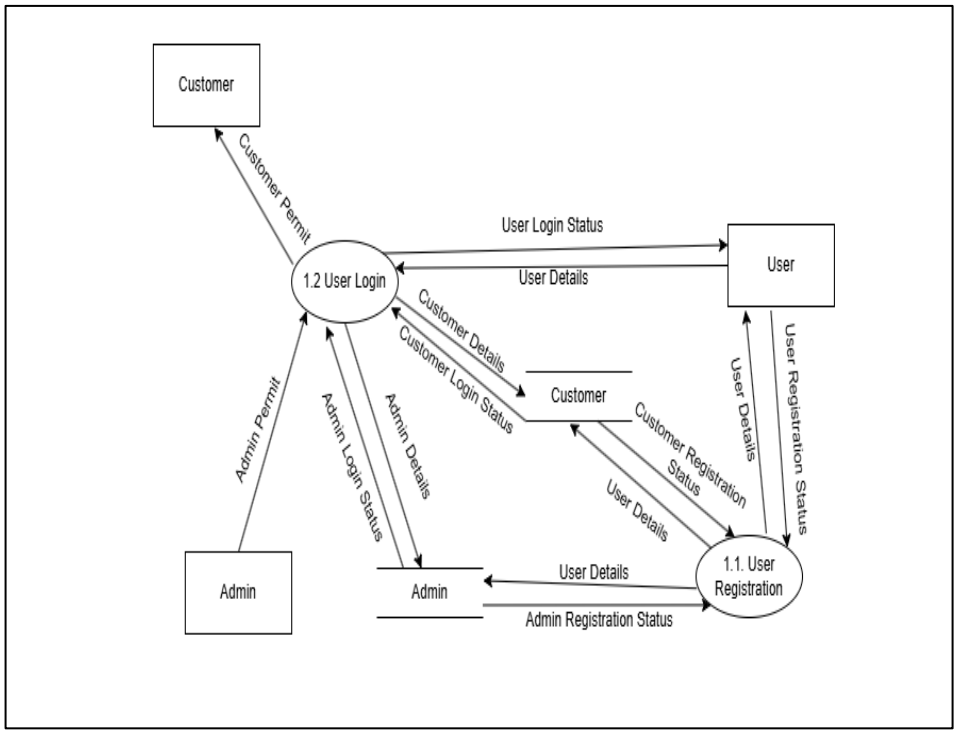


Figure 6: DFD-2 of user access

4.2 Internal Model Specification for the System:

4.2.1 Entity Representation Diagram:

An Entity-Relationship Diagram (ERD) visually represents the relationships and data within a system or database. ERDs show how data is logically organized and how different entities or tables in a database are connected. They illustrate the relationships between the various entities in the database. ER models can represent real-world objects, such as a person, a car, or a company, and the connections between them. In summary, an ER Diagram serves as the structural layout of the database (Belcic, 2024).

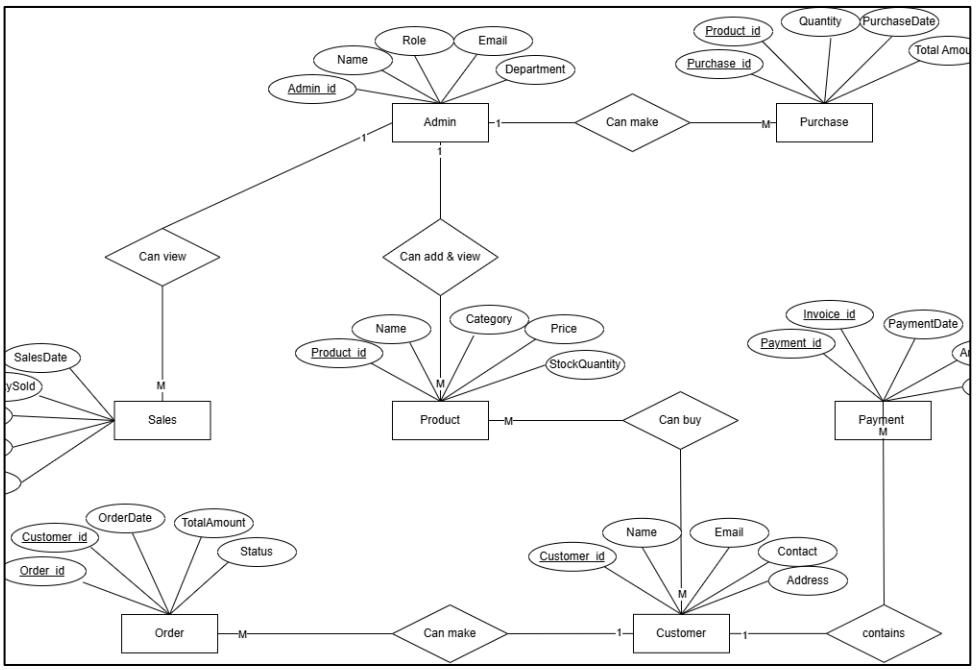


Figure 7: ERD representation of Inventory Management System

4.2.2 Data Dictionary

A data dictionary is essentially a well-organized collection of information that describes the various elements within a database. It serves as a reference guide for users, detailing the names, definitions, and attributes of the data elements involved in a database, information system, or research project. This means it provides insights into what each piece of data represents, how it can be used, and its significance in the context of a project (Austin, 2024).

In more practical terms, you can think of a data dictionary as a "cheat sheet" that

lists crucial details about the data elements, including their special properties and rules. For instance, it specifies attributes such as primary keys, foreign keys, and constraints like NOT NULL and UNIQUE. Overall, it acts as a valuable resource that helps users understand the structure and relationships within the database, making it easier to manage and utilize the data effectively (Austin, 2024).

Here are the data dictionary that we have made for each entity in this coursework:-

Table 3: Data Dictionary of Admin

Column	Constraints	Data Type	Size	Description
admin_ID	Primary key, Unique	Int	20	Unique identifier for admin.
admin_Name	Not null	Varchar	20	Full name for each admin
admin_role	Not Null	Varchar	20	Role of the admin in the system.
admin_phone	Not Null	Int	10	Phone of the admin
admin_address	Not Null	Varchar	8	Address of the admin.
admin_Email	Not Null	Varchar	30	Email address of the admin.

Entity Name: Payment Details

Description: It represents the payment details of customers.

Table 4: Data Dictionary of Payment Details

Column	Constraints	Data Type	Size	Description
payment_ID	Primary key, Unique	Int	5	Unique identifier for a payment table
payment_date	Not null	Date	10	It represent payment date of customer.

payment_method	Not Null	Varchar	20	Payment method used by customer.
customer_ID	Foreign key	Int	5	Phone of the admin
payment_amount	Not Null	Int	20	Address of the admin.
product_ID	Foreign key	Int	30	Id of the product
order_id	Foreign key	Int	5	Id of the order of user

Entity Name: Purchase Payment

Description: It represents the payment details of admin purchase made to buy products.

Table 5: Data Dictionary of Purchase Payment

Column	Constraints	Data Type	Size	Description
Purchas_payment_ID	Primary key, Unique	Int	5	Unique identifier for a purchase payment ID.
Purchase_payment_date	Not null	Date	10	It represent payment date of Admin.
Purchase_payment_method	Not Null	Varchar	20	Payment method used by admin.
Purchase_admin_ID	Foreign Key	Int	5	Id of the admin.
Purchase_payment_amount	Not Null	Int	20	Purchase payment amount.
Purchase_product_ID	Foreign key	Int	30	Id of the product.
Purchase_order_id	Foreign key	Int	5	Id of the order of user.

Entity Name: Product

Description: The product entity represents the goods or services offered by business.

Table 6: Data Dictionary of Product

Attributes	Datatype	Size	Constraints	Description
product_id	int	10	Pk, Not Null	ID of the product, for the unique identification of each product
Product_name	Varchar	50	Not Null	Name of the product
Category	Varchar	50	Not Null	ID of the category where the product belongs
price	int	10	Not Null	Price of the product
StockQuantity	Int	10	Not Null	Number of units of the product currently in stock

Entity Name: Customer

Description: The customer entity represents individual or organization entities that purchase goods or services from a business.

Table 7: Data Dictionary of Customer

Attributes	Datatype	Size	Constraints	Description
customer_id	int	10	Pk, Not Null	Unique identifier assigned to each customer in the system.
customer_name	Varchar	50	Not Null	Full name of the customer, typically including both first and last names.

Email	Varchar	50	Not Null	Email address of the customer. Must be unique to ensure each customer.
Contact	int	10	Not Null	Contact number of the customer,
Address	Varchar	50	Not Null	Complete address of the customer, including street address, city, state/province, zip code, and country.

Entity Name: Order

Description: The order entity represents the information of customer's purchase. It includes all the essential information about each order.

Table 8: Data Dictionary of Order

Attributes	Data type	Size	Constraints	Description
Order_id	int	10	Pk, Not Null	Unique identifier for each order placed.
customer_id	int	10	FK, Not Null	Foreign key referencing the customer table indicating the customer who placed the order.
Order_date	date	8	Not Null	Date and time when the order was placed.
Total_amount	int	10	Not Null	Total amount of the order, including taxes and shipping costs.
Status	Varchar	20	Not Null	Current status of the order (e.g. Pending, processing, shipped, delivered)

Entity Name: Purchase

Description: The purchase entity will represent and records the transaction. It will include details of purchase made by users.

Table 9: Data Dictionary of Purchase

Columns	Datatype	Size	Constraints	Description
Purchase_id	int	10	Pk,Not Null	It is a unique integer for each purchase to distinguish each and every purchase made.
Product_id	int	10	Fk, Not Null	It is Product_id from Product table referring to the user.
Quantity	int	10	Not Null	It refers to the number of items purchased.
PurchaseDate	Date	8	Not Null	It will record the date of the item purchased.
TotalCost	int	10	Not Null	It will represent the total cost of the purchased items.

Entity Name: Sales

Description: The sales entity will keep the track of the items sold. Each sales record will include volume of product sold and revenue generated.

Table 10: Data Dictionary of Sales

Columns	Datatype	Size	Constraints	Description
Sales_id	int	10	Pk, Not Null	It is a unique integer for each sales transaction to distinguish each and every sales made.

Order_id	int	10	Fk, Not Null	It is Order_id from Order table referring to a specific customer Order.
Product_id	int	10	Fk, Not Null	It is Product_id from Product table referring to the product to be sold.
QuantitySold	int	8	Not Null	It will refer to the number of items sold to the user.
SalesDate	Date	8	Not Null	It will record the date of the sales made.

4.2.3 Major Data Definition of Entities

Admin = admin_ID + admin_Name + admin_role + admin_phone + admin_address.

Payment Details = {payment_ID + payment_date + payment_method + customer_ID + payment_amount + product_ID + order_id}.

Product = {product_id + Product_name + Category + price + StockQuantity}.

Customer = {customer_id + customer_name + Email + Contact + Address}.

Order = {Order_id + customer_id + Order_date + Total_amount + Total + Status}.

Purchase = {Purchase_id + Product_id + Quantity + PurchaseDate + TotalCost}.

Sales = {Sales_id + Order_id + Product_id + QuantitySold + SalesDate}.

4.2.4 Major Data Flows:

It includes definitions and explanations about how data flows, how it's stored, and the various entities involved, particularly those derived from the data sub-model.

The notations used in Major Data Flow are summarized below:

= > is composed of (Definition)

+ > and (sequence of data)

{ } > iteration/repetition of data

[ ] > selection (either or) of data

* * > comment

( ) > optional data

Table 11: Major data-flow of Processes

Payment	Payment details + payment confirmation + invoice + order amount
Generate Report	Sales report + purchase report + date range
Real Time Stock Update	Product details + stock update confirmation
Purchase Order	Product details + Purchased Product Details
Dispatch Order	Sales order + sales order details + dispatch update + order confirmation + dispatch information

4.2.5 Process specifications for elementary processes:

A process specification is created for primitive processes and for higher-level data flow diagram processes (process specifications). A process specification is a documentation method crucial for explaining the logic and formulas involved in generating output data (Choudhary, 2024).

Table 12: Process specification of Product Management

Name	Product Management
Process Number	4
Input	Stock / Sales / Supplier Data
Output	Updated stock
Process Details	This process is responsible for managing all aspects related to products, including stocks, sales and supplier information. It appears to be a complete system for tracking and analyzing product data.

Table 13: Process specification of Make Payment

Name	Make Payment
Process Number	7
Input	order amount, payment details
Output	invoice
Process Details	• User provides their payments details. • All the details from user are checked properly to make errorless.

	• Payment details are securely stored and enrolled in the system. • Payment are done through external payment providers which is integrated in the system. • User receives invoice after payment done

Table 14: Process specification of Report Preparation

Name	Report Preparation
Process Number	8
Input	Date
Output	Generated Report
Process Details	• If the user is admin, date is passed and sales details are retrieved from Sales datastore, else if the user is customer, purchase details are retrieved from Order datastore. • The details passed are validated, calculated and enrolled for report generation. • Report is displayed.

Table 15: Process specs of purchase order

Name	Purchase order
Process Number	2
Input	Product Details
Output	Purchased Product Details
Process Details	User asks the details of product. They get the product details and choose whether they want the product or not. If they want the product they can add purchase and if they do not want the product they can just cancel order. After purchasing admin can view the purchased product.

Table 16: Process specs of dispatch order

Name	Dispatch Order
Process Number	5
Input	User request
Output	Order Confirmation, Dispatch Confirmation, Delivery preferences

Process Details	1. User requests for dispatch order details. 2. Customer provides delivery 3. Admin manages dispatched order details and manages customer information and provides Customer the dispatch confirmation and details.

4.3 Design Specification

4.3.1 Structure chart for the whole system.

Structure charts are used in top-down modular design and structured programming, it helps to break down the system into manageable modules. They help visualize the system's complexity, making it easier for developers to understand and manage. They also assist in identifying potential issues or bottlenecks in the system's design (Lindsey, 1977).

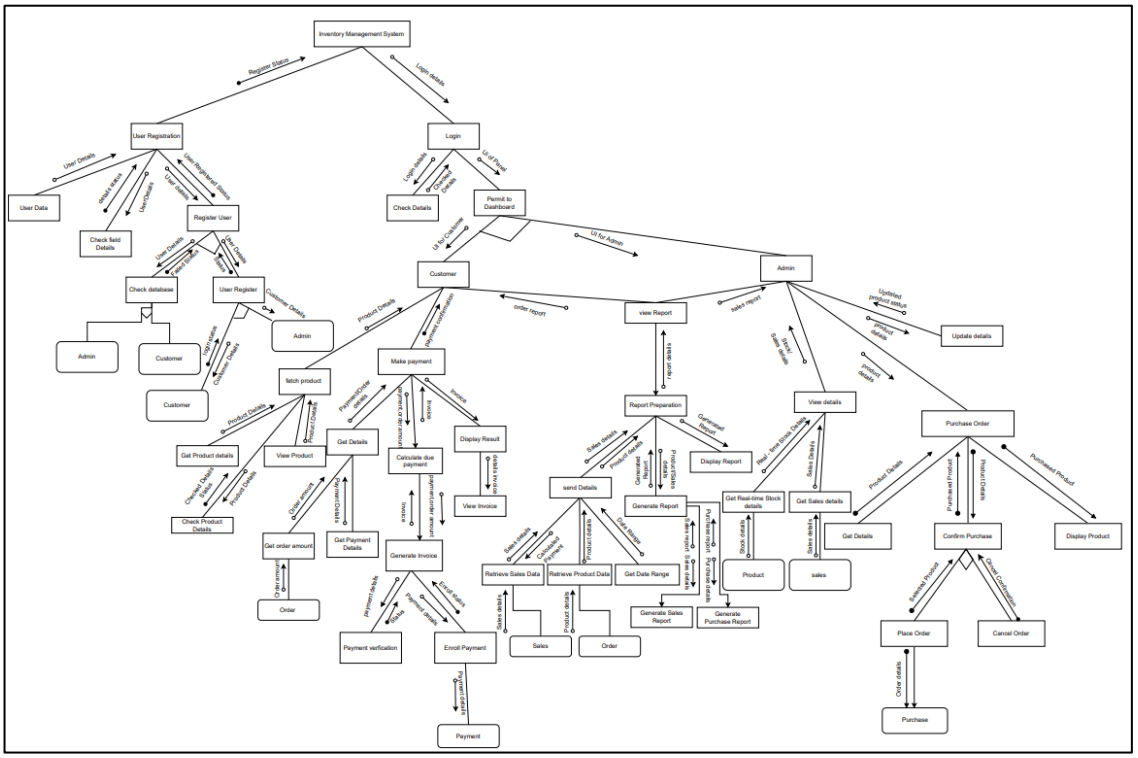


Figure 8: Structure chart of whole system

The structure chart is a chart that represents the hierarchical structure of module. It breaks down the entire system to its fundamental modules, and describes the functions and sub-functions to a greater detail of each module of the system. Furthermore, it also partitions the system into different black boxes which acts as an abstraction, i.e. the functionality of the system is known to the users whereas inner details are unknown. Every input is given to the black boxes, and appropriate outputs are generated and given to the user (Anon, 2024).

The system involves user registration, after which the user gains the permit for the customer or the admin. If the user is admin, it has privileges of managing the product details which includes purchasing products from the suppliers, add the product to the system, view the stock details, update the product details, and also can get the sales details along with the report if the admin wants. Whereas, if the user is customer, it has access to view the products from our database, compare the products and choose the best one, order the products, pay, get the invoice, and also get the purchase report for every report they desire to generate. The system also stores the data of the customer and their history for better user experience.

4.4 Progress Logs

4.4.1 Assumptions

1) User Assumptions:
   - Any user can register to become an admin or a customer.
   - A user must be registered so that they can make purchases.
   - Our system will have only an admin user.
   - The details about the admin will be stored in a different table.
2) Payment Assumptions:
   - Both admin and customers can make payments and they have two different processes for making the payments, i.e. purchase payment for admin and product payment for customers.
   - External entity payment provider will be integrated in the system for the convenience to the users while making payment.
3) Report Preparation Assumptions:
   - System generates report on the basis of user: sales report for admin or product report for customers.
   - The system provides functionality of filtering and searching by date.
4) Product Management Assumptions:
   - The Admin will add and update products into the system.
5) Purchase Order Assumptions:
   - The Admin will make the purchases.
   - The Admin will have privilege to place or cancel purchase order.
6) Sales Management Assumptions:
   - The system will automatically check for availability of stock before purchases.

4.4.2 Group Responsibilities

Table 17: Table of Group member Responsibilities

Group Members	Responsibilities of each member
Jonsen Gaire	**Business case** i. Problem/solution ii. Benefits iii. Goals iv. Assumption **SRS** Non-Functional **Detailed Specs of group task** Group DFD level 1 **Internal Model specs** i. ERD ii. Design Specification iii. Structure chart **Detailed specification of individual tasks** Report Preparation **Summary**
Manoj Neupane	**Business case** i. Introduction ii. Benefits iii. Development/Estimated Time iv. Goals v. Assumption **SRS** i. Introduction ii. Functional **Detailed Specs of group task** Group DFD level 1 **Internal Model specs** i. ERD ii. Data Dictionary iii. Process Specification **Design Specification**

	Structure chart **Detailed specification of individual tasks** Payment
Kaushal Thapa	**Business case** i. Introduction ii. Benefits iii. Objective **SRS** Introduction **Detailed Specs of group task** i. Group DFD level 0 ii. Group DFD level 1 **Internal Model specs** i. ERD ii. Data Dictionary iii. Process Specification **Progress logs** **Detailed specification of individual tasks** Purchase order
Prabesh Marasani	**Business Case** Benefits **SRS** Requirements that reduce time and cost **Detailed Specs of group task** i. Group DFD level 1 ii. Group DFD level 2 **Internal Model specs** i. ERD ii. Data Dictionary iii. Process Specification **Design Specification** Structure chart **Detailed specification of individual tasks** Real-Time Stock Update
Beeju Bk	**Detailed specification of individual tasks** Dispatch Order

4.4.3 Assignment Diary

After the releasement of the coursework, our group members met regularly to discuss and delegate the coursework to each member of our group.

The following table is about the meeting and description of activities done by our group:

Table 18: Table of Meeting - 1

Meeting Log-1	
Date	2024/11/22
Location	Library
Discussion Summary	In this discussion we went through our coursework scenario and began splitting questions among group members.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	1:30 PM to 2:49 PM

Table 19: Table of Meeting - 2

Meeting Log-2	
Date	2024/11/24
Location	Library
Discussion Summary	In this discussion we collected some information on Business case and divided the objectives and goals for each member and started doing it.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	2:35 PM to 6:15 PM

Table 20: Table of Meeting - 3

Meeting Log-3	
Date	2024/11/25
Location	Library
Discussion Summary	In this discussion we went through our first milestone coursework and revised it before our submission time.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	11:15 AM to 12:30 PM

Table 21: Table of Meeting - 4

Meeting Log-4	
Date	2024/12/6
Location	Begnas
Discussion Summary	In this discussion we divided the topics among the group members.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	3:00 PM to 3:30 PM

Table 22: Table of Meeting - 5

Meeting Log-5	
Date	2024/12/11
Location	Study Hall

Discussion Summary	In this discussion we went through the work done by members. We discussed about SRS and Overall introduction
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	11:30 AM to 3:25 PM

Table 23: Table of Meeting - 6

Meeting Log-6	
Date	2024/12/14
Location	Library
Discussion Summary	In this discussion we collected some data on Data Flow Diagram (DFD) and ER diagram of both individual and group.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	1:00 PM to 5:00 PM

Table 24: Table of Meeting - 7

Meeting Log-7	
Date	2024/12/15
Location	Study hall
Discussion Summary	In this discussion we started combining the diagrams of DFD and made Er diagram. Made corrections for errors. After combining the diagrams we went through our second milestone work.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.

Time	12:45 PM to 6:30 PM

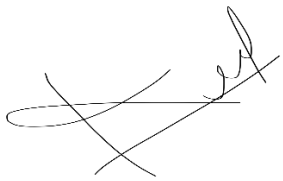
Table 25: Table of Meeting - 8

Meeting Log-8	
Date	2024/12/22
Location	Study Hall
Discussion Summary	In this discussion we made Data Dictionaries for each entities from our Er diagram. We also made our individual structure chart and Progress Specifications
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonsen Gaire, Beeju Bk.
Time	3:10 PM to 5:45 PM

Table 26: Table of Meeting - 9

Meeting Log-9	
Date	2024/12/23
Location	Study Hall
Discussion Summary	In this discussion we completed our individual and group work for milestone 3 and checked it thoroughly.
Group Members	Manoj Neupane, Kaushal Thapa, Prabesh Marasini, Jonseen Gaire, Beeju Bk.
Time	2:25 PM to 6:15 PM

Table 27: Table of Signatures of Group members

Group Members	Signature
Beeju Bk	
Jonsen Gaire	
Kaushal Thapa	
Prabesh Marasini	
Manoj Neupane	

5. Individual Tasks

5.1 Individual task of Payment

Module Name: Payment

Name: Manoj Neupane

London Met ID: 23048832

5.1.1 Introduction to Payment

Introduction:

The payment part of the coursework is an individual task. It is the process where a user gets amount details for payment, provide valid payment credentials, and get the invoice.

At first, the system provides information to the user about the total amount they need for payment. The user provides the payment credentials to the system, which is field validated by the system, sends information to the payment providers, stores the payment details to the payment data store, and generates an invoice for the user.

After that, the system asks user for payment credentials and also captures chosen order details i.e. order amount for payment process. And then, payment details are enrolled in the system, and after that payment details and total cost are forwarded to the payment providers and then, they check whether the payment method is valid or not, if it's valid, they will allow the user to buy the products. Also, the invoice will be sent to the customer.

5.1.2 Environmental Model Specification

Context Level Diagram

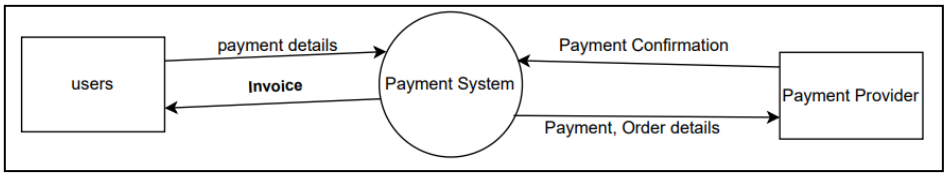


Figure 9: DFD 0 of Payment

5.1.3 Internal Model Specification for the Payment

Data Flow Diagram Level 1

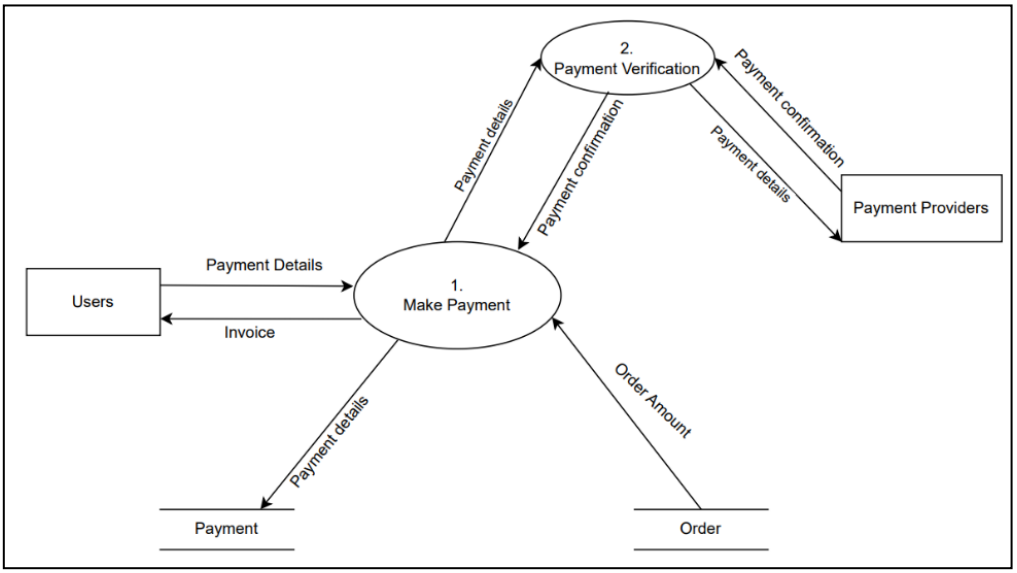


Figure 10: DFD 1 of Payment

Data Flow Diagram Level 2

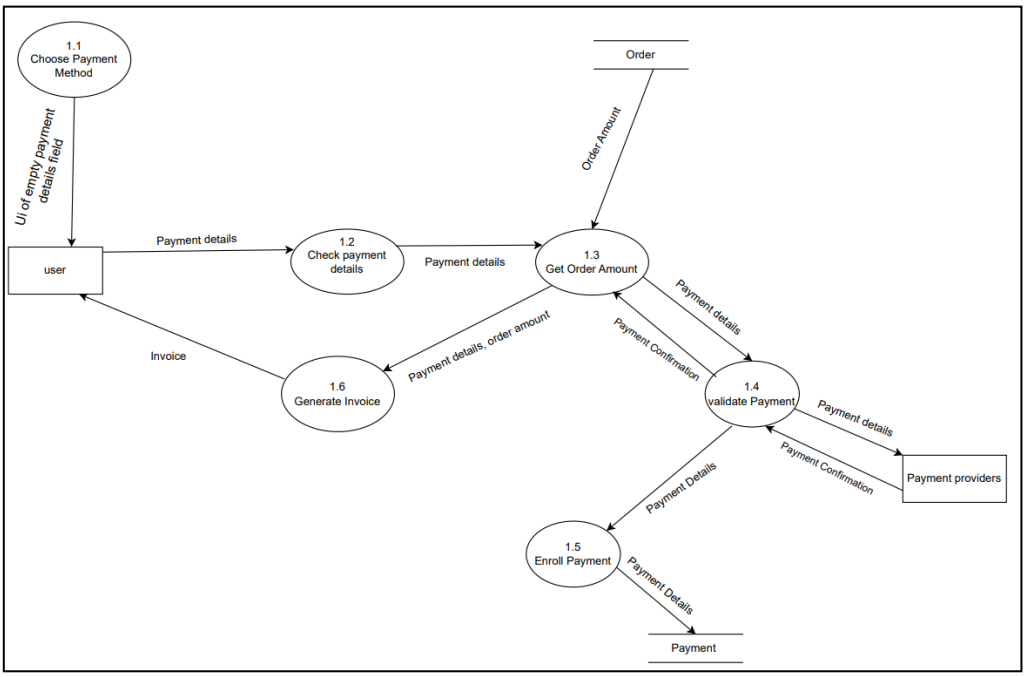


Figure 11: DFD 2 of Payment

5.1.4 Design Specification

Structure Chart of the Payment

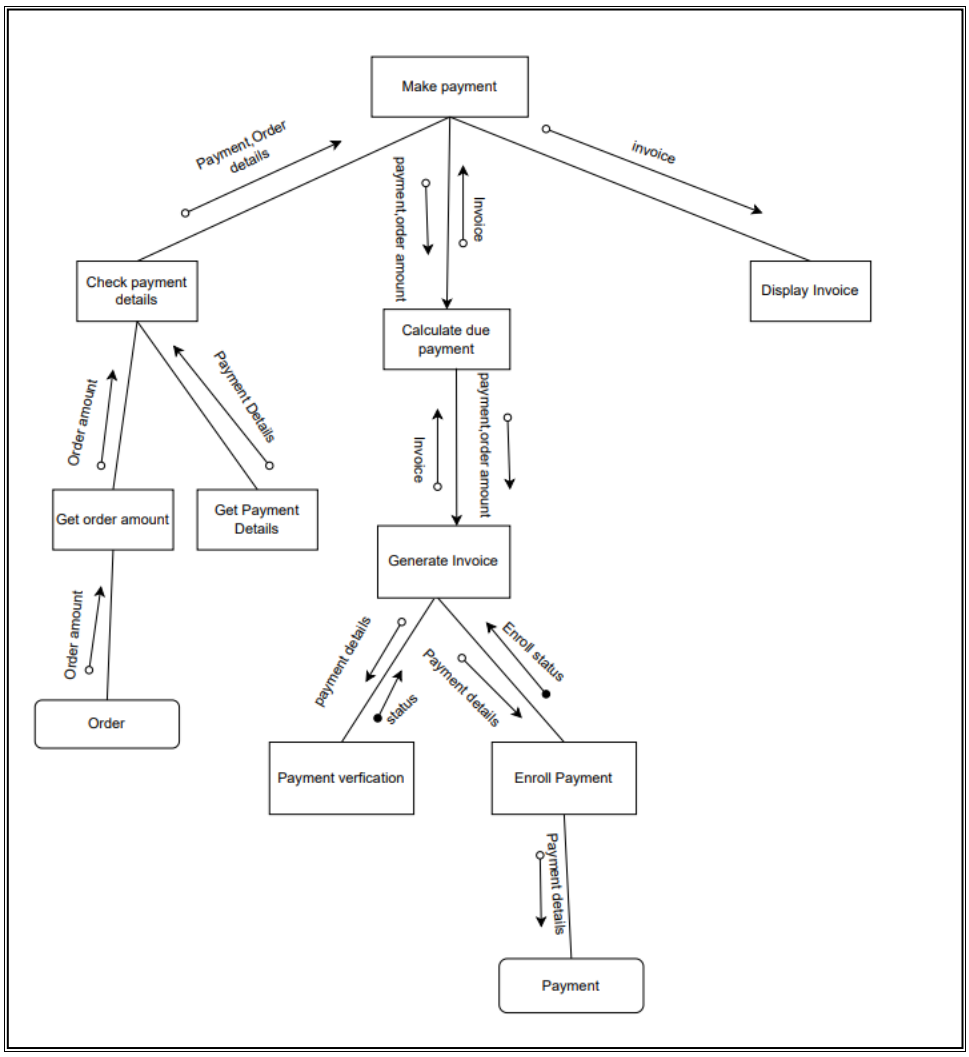


Figure 12: Structure chart of Payment

This structure from above represents the overview of how the “payment” process is working, it shows the data flow between different processes which makes it easily understandable The Structure Chart provides a clear and organized view of the payment process by breaking it down into a hierarchy of interconnected modules. At the top, we have the "Make Payment" module, which acts as the controller of this entire operation. Below it, several key modules work together: "Get Details" gathers all the necessary information, "Process Payment" takes care of the actual transaction, and "Display Result" shares the outcome with the user. The arrows connecting these modules illustrate how data flows between them. This visual representation not only

makes it easier to understand how everything works together. Additionally, it helps developer's spot potential issues early on.

Module Specification Module Name: Payment

Purpose: The main purpose of this module "Make payment" is it allows users to easily pay fees they enrolled in and order they placed in the system.

Pseudocode:

The pseudocode of "Payment" is given below:

START

GET order details

INPUT payment details (order amount, payment details)

VALIDATE payment and order details

CONFIRM payment

AUTHORIZE payment

GENERATE invoice

SEND invoice details to users

DISPLAY invoice

END

Input parameters: order amount, payment details
Output parameters: payment status, receipt
Global Variable: customer_id, admin_id, order_ID
Local variable: payment_ID
Calls: validate payment, authorize payment
Conclusion:

In our group, I was assigned to this module as an individual task. In this Payment section, it ensures payment is securely process whenever a payment is successfully processed, it helps business maintain the accurate. Additionally, the ability to generate detailed payment reports gives businesses valuable insights into sales trends, customer preferences, and financial performance.

This process helps to align the organization goals by integrating secure and efficient payment system in the inventory management system.

5.2 Individual task of Purchase Order

Module Name: Purchase Order

Name: Kaushal Thapa

London Met ID: 23048815

5.2.1 Introduction to Purchase Order

Introduction:

This part of the coursework is an individual task, and I was assigned to make the Purchase Order function. Purchase order function allows user (admin) to buy products and view the purchased products. The add purchase function will give privilege to cancel and place order according to user wish.

Overall, the system will let admin to get product details from suppliers to add up in the stock of the warehouse inventory. After getting the details of the product the system will provide two main functions Add purchase and View Purchase. The Add purchase will have two more inner functions which allows user to place order and cancel order. If the user wants to buy the product selected will be stored in a database and lead the user to Payment section. And the View Purchase will allow users to view the past purchases history which are stored in a purchase database.

5.2.2 Environmental Model Specification

Context Level Diagram

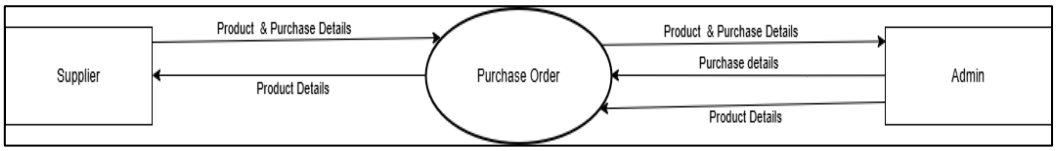


Figure 13: DFD 0 for Purchase order

5.2.3 Internal Model Specification for the Purchase Order

Data Flow Diagram Level 1

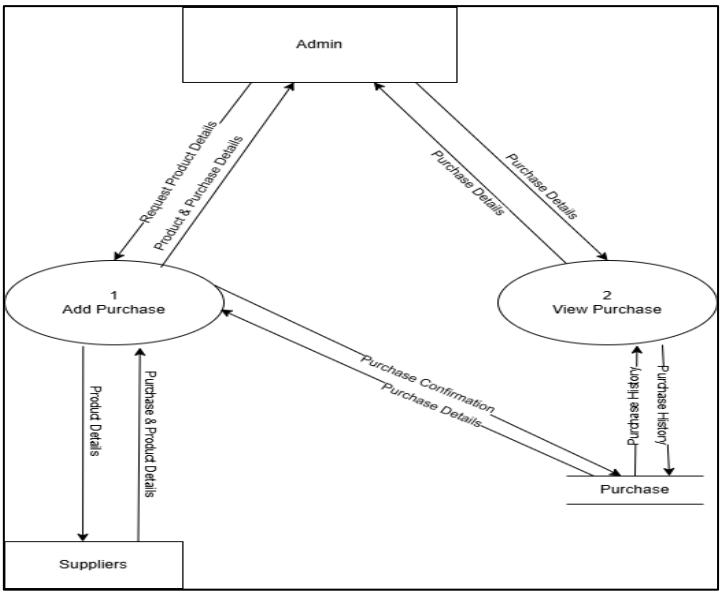


Figure 14: DFD 1 of Purchase Order

Data Flow Diagram Level 2

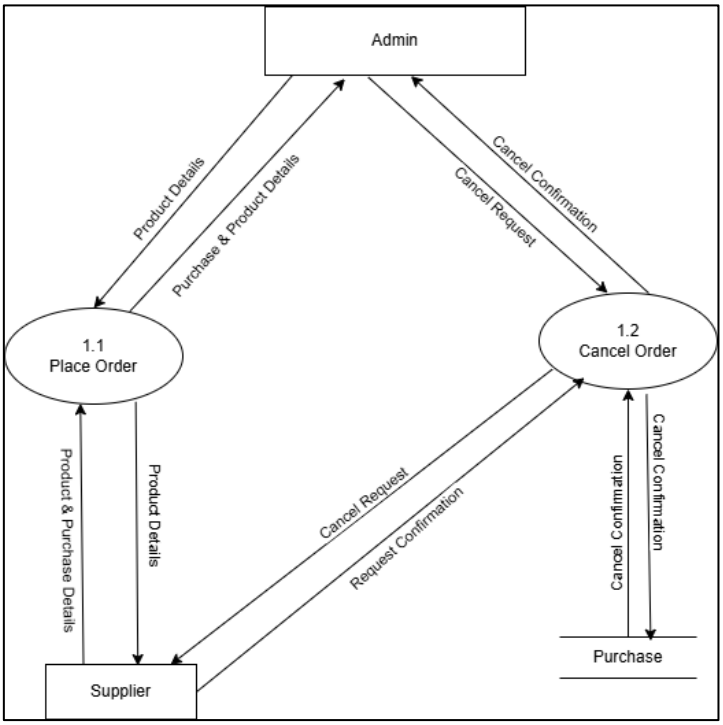


Figure 15: DFD 2 of Purchase Order

5.2.1 Design Specification

Structure Chart of Purchase Order

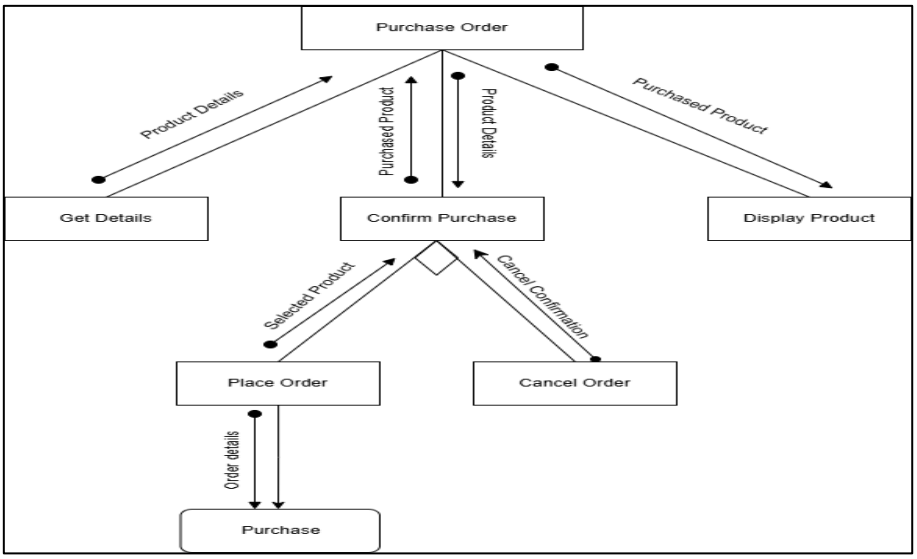


Figure 16: Structure Chart of Purchase Order

Here, the Structure Chart shows the systematic flow of processes within "Purchase Order". The user (admin) sends request for Product Details. The system provides details of product and sends a confirmation for purchase. The confirmation authorizes if the admin wants to buy or not. If the admin wants to buy admin can send details and place order and if the admin doesn't want to buy admin can cancel order and cancelation confirmation is sent. The order placed is then stored in a purchase datastore. After the completion of purchase the purchased product is then displayed to admin.

Module Specification Module Name: Purchase Order

Purpose: To let the admin buy products from supplier.

The process will request product detail and give the details back to the admin. After admin gets details of products, he/she can buy products from the system as well as view past history of purchases.

Input parameters: Product Details

Output parameters: Purchased Product Details

Global Variable: admin_id

Local variable: N/A

Pseudocode:

The pseudocode of “Purchase Order” is given below:

START

Send Product Information

Get Product Details

Place Product Order

Get Purchase Details

Send Purchase Details

Display Purchased Product Details

END

Conclusion:

Throughout this group project it has taught me a lot about teamwork and coordination among team members. In this process of project I learned about the data flow diagrams and structure chart. In the project process I learnt how to design Data Flow Diagram (DFD), wrote Pseudocode, and made Structure chart for the better understanding of the project system.

As a beginner, I had constant problems with Data Flow Diagrams and Structure as they were new to me. We went through frequent discussions with the group members and module teachers helped me overcome the problem. After many trials and errors, I managed to complete all my assigned tasks. This project provided me with depth-knowledge about Data Flow Diagrams and Structure charts.

5.3 Individual task of Report Preparation

Module Name: Report Preparation

Name: Jonsen Gaire

London Met ID: 23048808

5.3.1 Introduction to Report Preparation

Introduction:

This part of the coursework is an individual task and I was assigned to make the Report Preparation function. Report preparation is the process of getting the sales or report details for admin and customer respectively for preparing report on the already designed format or design, fill up the fields required, and provide user their required report, i.e. sales report for admin, and purchase report for customers.

Understanding the flow of the process, first, the system validates the user whether the user is an Admin or a Customer, and authorizes respectively. It takes date range as input from admin, get the sales details from the sales database, prepare a report filling the fields needed, and provide sales report to the admin, whereas customer clicks Generate Report for report preparation, and the system prepares the purchase report from the order table and gives to the customers.

The process prepares the sales report for admin using details from sales datastore, and purchase report for customer using details from order datastore.

5.3.2 Environmental Model Specification

Context Level Diagram

Figure 17: DFD 0 of Report Preparation

5.3.3 Internal Model Specification for the Report Preparation

Data Flow Diagram Level 1

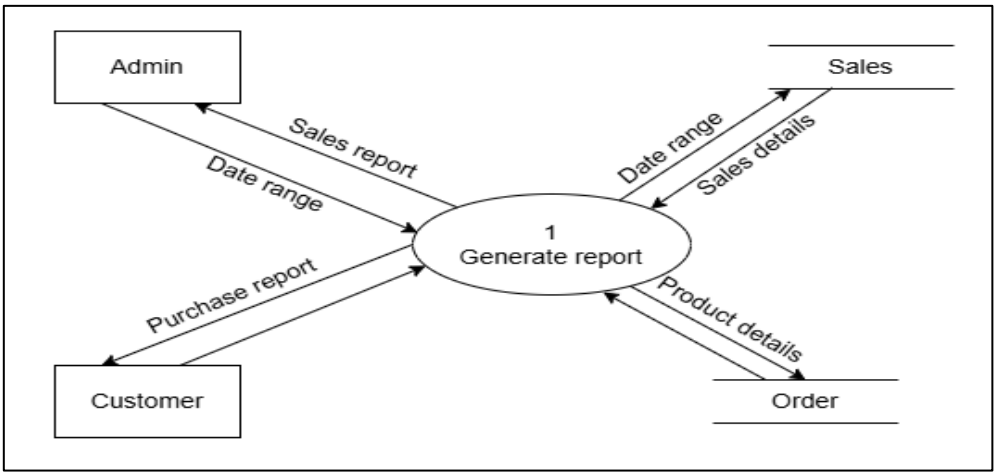


Figure 18: DFD 1 of Report Preparation

Data Flow Diagram Level 2

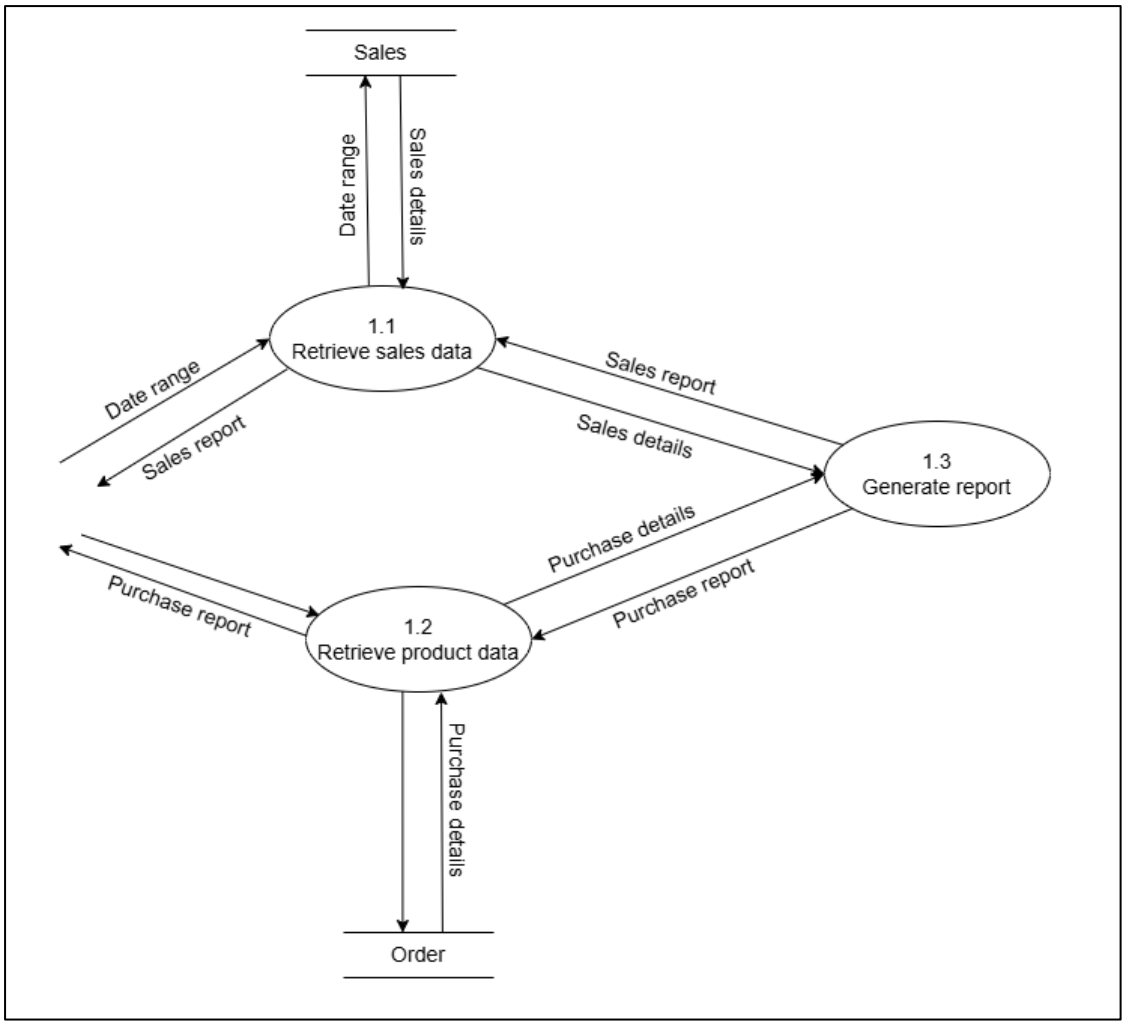


Figure 19: DFD 2 of Report Preparation

5.3.4 Design Specification

Structure Chart of the Report Preparation

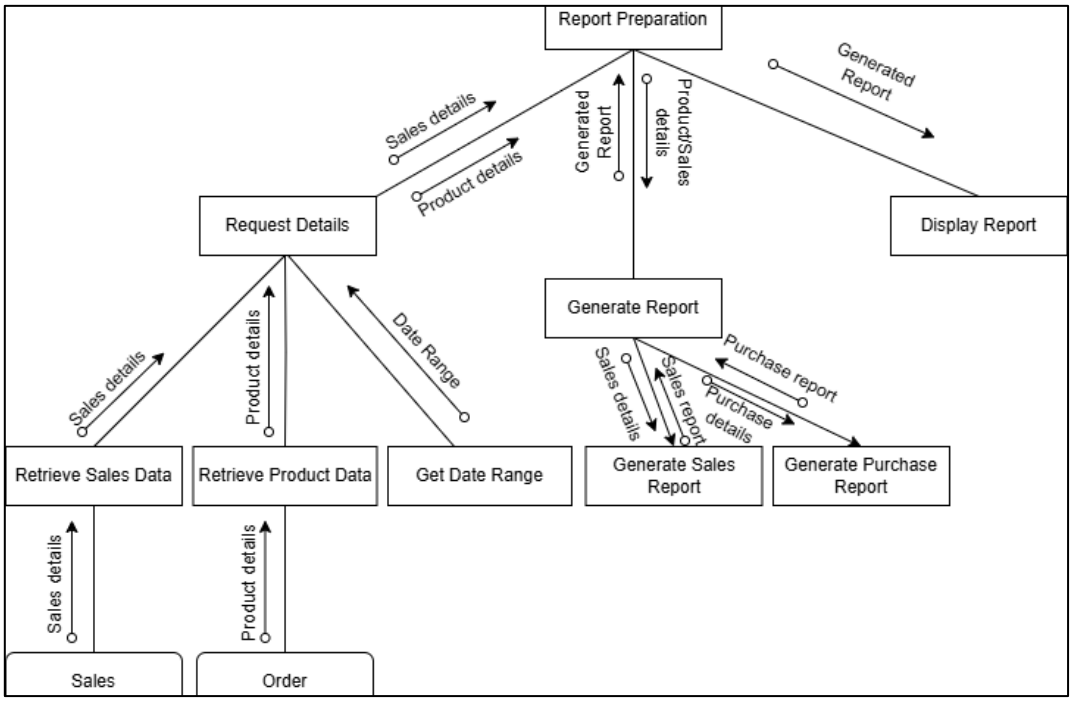


Figure 20: Structure Chart of Report Preparation

The Structure Chart illustrates the systematic flow of processes within the “Report Preparation”. The user sends requests for report preparation. The system validates the user, and authorizes if it’s Admin or Customer. If the credentials authenticate Admin, then every process till the generation of report will be done as Admin’s side. The Sales request will be passed to the system, then the request details will be validated with the Sales data store, and if it matches, it will retrieve the sales details and send it to generate report for report preparation. Henceforth, the report will be prepared and displayed to the Admin. But, if the credentials authenticate Customer, then every process till the generation of report will be done as Customer’s side. The Product request will be passed to the system, then the request details will be validated with the Order data store, and if it matches, it will retrieve the product details and send it to generate report for report preparation. Furthermore, the report will be prepared and displayed to the Customer.

Module Specification Module Name: Report Preparation

Purpose: The main purpose of this module is to get date from the admin, or button click from the customer, and prepare sales reports for admin and purchase report for customers.

Pseudocode

The pseudocode of “Report Preparation” is given below:

START

IF Credentials are from Admin **THEN**

SEND the Date to Generate Report

RETRIEVE the Sales Details

GENERATE Sales Report for Admin

DISPLAY the Generated Sales Report to the Admin

ELSE IF Credentials are from Customer **THEN**

RETRIEVE the Product Details

GENERATE Purchase Report for Customer

DISPLAY the Generated Purchase Report to Customer

END IF

END

Input parameters: date, sales details, purchase details

Output parameters: Sales report, Purchase report

Global variable: admin_id, customer_id

Local variable: N/A

Calls: None

Called by: Generate Report

Conclusion:

In our group project, I was assigned with the “Report Preparation” module as an individual assignment. This module was linked to other parts of the project, and included Data Flow Diagrams (DFD), Pseudocode, and a Structure chart. To start, I designed context level, level 1, and level 2 Data Flow Diagrams (DFD) to outline the whole report creation process. Then, I developed the structure chart for a better understanding of the system.

As a beginner, I struggled initially, but with constant and frequent discussions with my group members and the module teacher helped me overcome those challenges. After countable trials and errors, I managed to complete all my assigned tasks. This project provided me with in-depth knowledge about Data Flow Diagrams and Structure charts

5.4 Individual task of Real Time Stock Update

Module Name: Real Time Stock Update

Name: Prabesh Marasini

London Met ID: 23048854

5.4.1 Introduction to Real Time Stock Update

Introduction:

The “Real-time stock update” system in global tech corporation is designed to make the admin experience easy and efficient by providing accurate and up-to date stock information. It helps the admin to interact with the product database for retrieving and monitoring the stock details in real time. The major process view stock will helps the admin to access detailed information on the quantity of the product and providing accurate and updated data for stock management.

The system enables smooth interface between the admin and the product database. The admin will get product details and update quantity details, while the system confirms that all product stock data is remains updated and available. This system will provide real-time functionality, improving inventory accuracy, operational efficiency and overall administrative effectiveness by providing up to date stock information.

5.4.2 Environmental Model Specification

Context Level Diagram

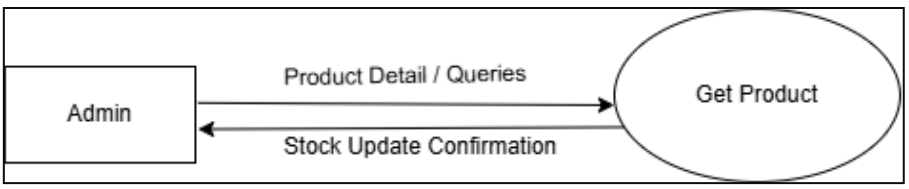


Figure 21: DFD 0 of Real-time Stock Updates

5.4.3 Internal Model Specification for the Real-Time Stock Update

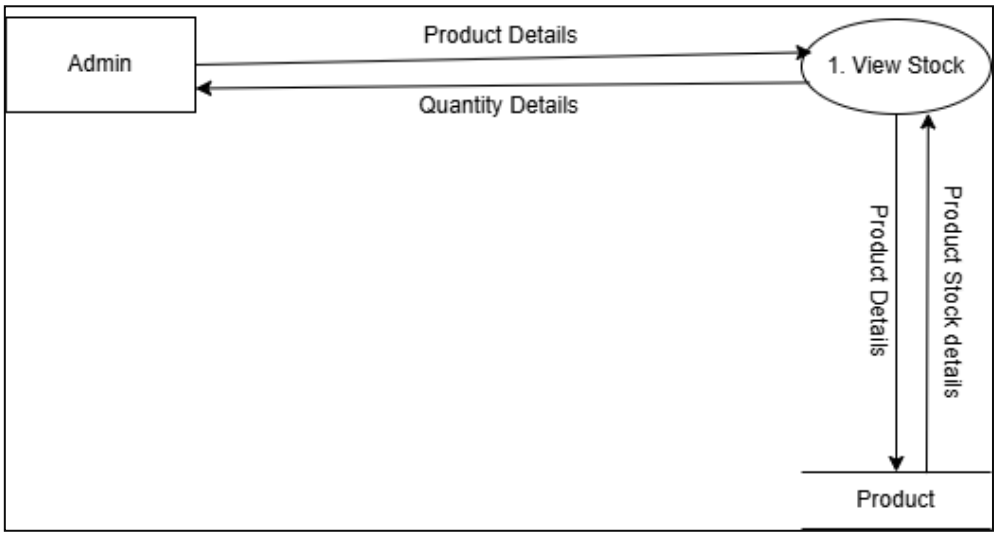


Figure 22: DFD 1 of Real-time Stock Update

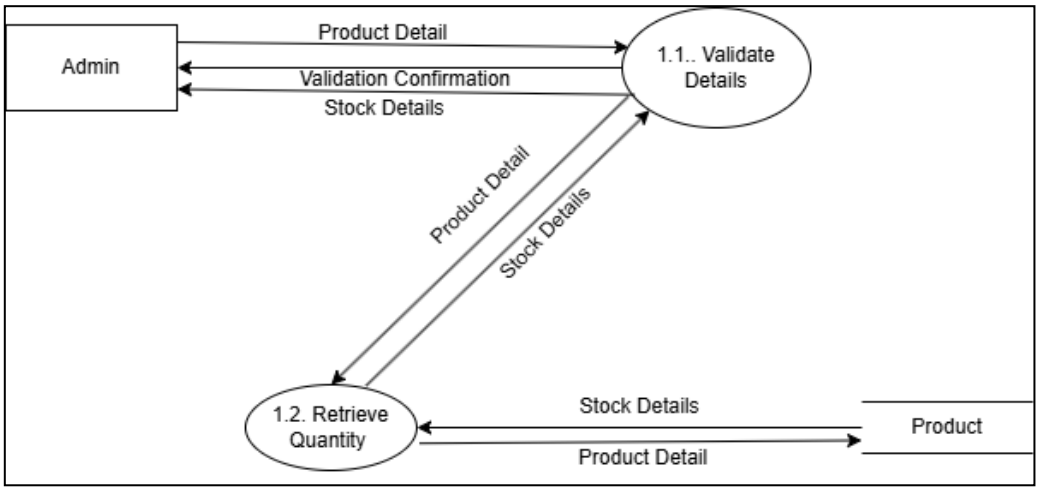


Figure 23: DFD 2 of Real-time Stock Update

5.4.4 Design Specification

Structure Chart of the Real-Time Stock Update

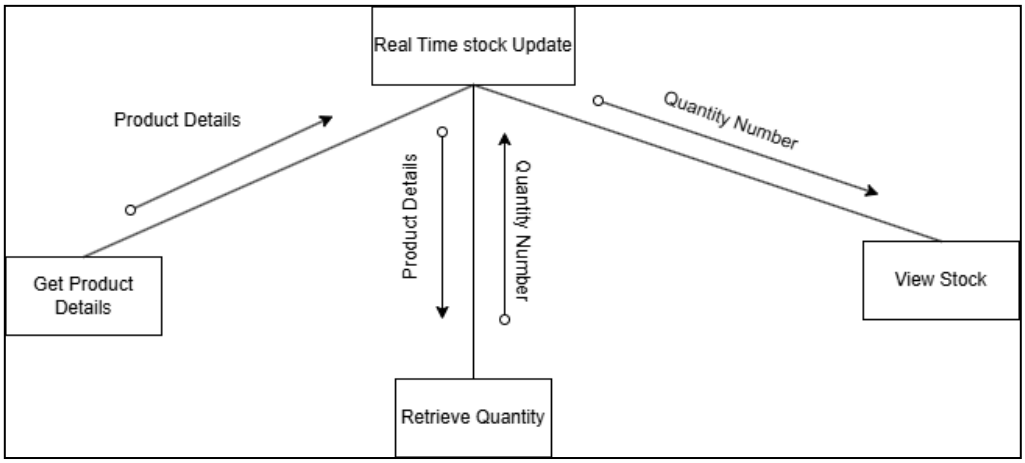


Figure 24: Structure Chart of Real Time Stock

A structure chart is a top-down modular design tool, constructed of squares demonstrating the different modules in the system, and lines that connect them. The structural chart explains the systematic flow of processes within Real Time stock update. In this structure chart, there are three types of data flow input, process and output. In input there is get product details which retrieves product information and in process part there is retrieve quantity which takes the product details from input part retrieves quantity number of the product. At the end there is View stock functionality in output part which displays updated stock information in real-time. The system enhances operational accuracy, ensures process continuity and improves administrative efficiency by managing the flow of data.

Module Specification Module Name: Real Time Stock Management

Purpose: The main purpose of Real time stock update system is to preventing stock outs and overstocking by continuously updating stock data and helping the admin to interact with the product database for retrieving and monitoring the stock details in real time.

Input parameters: Product ID, Quantity Number, product details
Output parameters: Updated stock information
Global Variable: admin_id
Local variable: product_ID

Pseudocode:

The pseudocode of “Real Time Stock Management” is given below:

START

GET product details

RETRIEVE current stock quantity

INPUT quantity number

UPDATE stock quantity in real-time

SAVE updated stock details

DISPLAY updated stock quantity

END

Conclusion:

I was given the responsibility of this module and after doing a lot of research I was able to complete my individual part of the coursework. It was a very nice experience working in a group because we were able to divide big tasks into smaller pieces and complete them effectively along with our individual responsibilities.

The main objective was not to design a system but to work with a team and meet due dates. The individual tasks focused on Real time stock updates and the tasks comprised the context diagram, level 0, 1 and 2 fragments of the DFDs, and a structure chart. In my individual task the real time stock update makes the system management easier and faster by providing admin accurate and timely stock information. This task helped me to understand how to maintain continuous input and output flows while breaking down complex processes into practicable parts.

5.5 Individual task of Dispatch Order

Module Name: Dispatch Order

Name: Beeju BK

London Met ID: 23048775

5.5.1 Introduction to Dispatch Order

Introduction:

The dispatch order process is all about making sure customers get their products accurately and on time. It sends off with the admin receiving the order information and checking its validity. Once the details are confirmed and product availability is verified, the process moves on to preparing the dispatch. The order is then prepped for shipping, and the system updates the ERP (Enterprise Resource Planning) with all the dispatch details. Finally, an update is sent to the logistics team to manage the physical delivery, ensuring the order reaches the customer as promised. This seamless process guarantees a smooth transition from order placement to delivery, involving crucial checks and clear communication between various departments.

5.5.2 Environmental Model Specification

Context Level Diagram

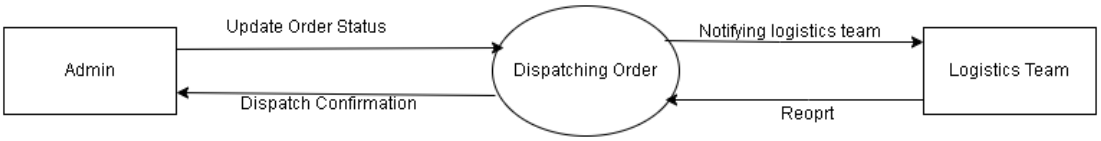


Figure 25: DFD - 0 of Dispatch Order

5.5.3 Internal Model Specification for the Dispatch Order

Data Flow Diagram Level 1

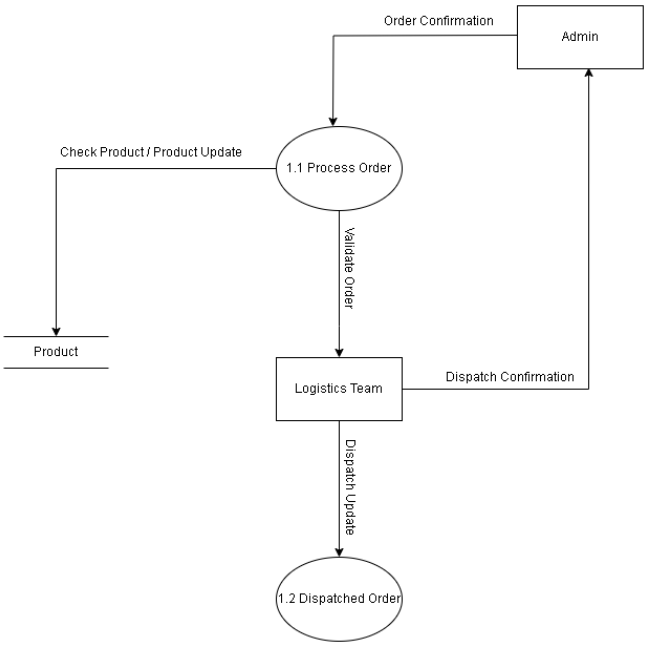


Figure 26: DFD - 1 of Dispatch Order

Data Flow Diagram Level 2

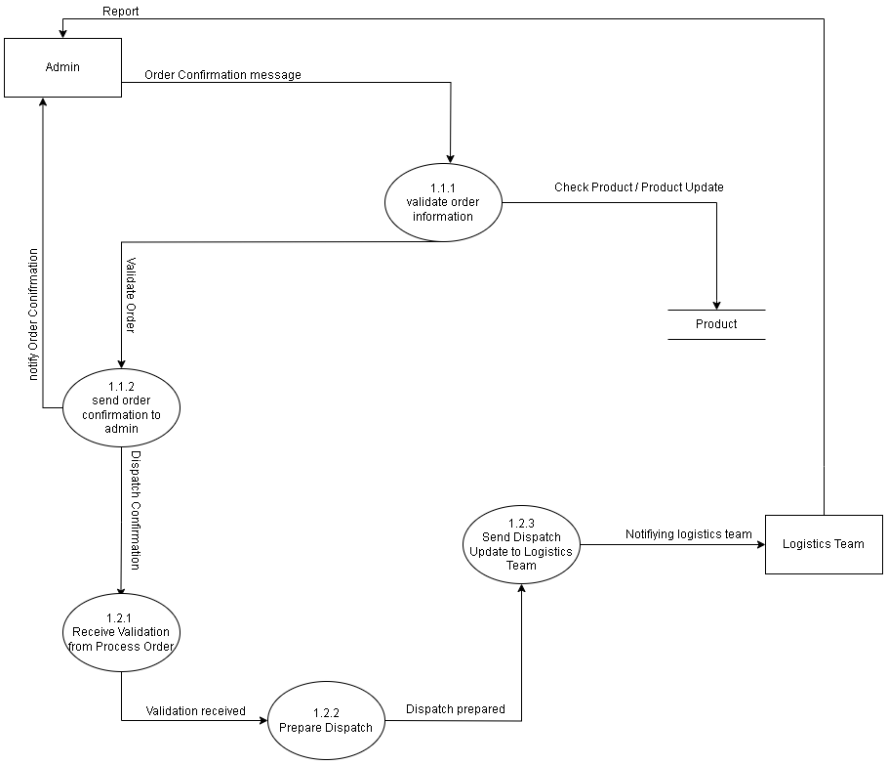


Figure 27: DFD 2 of Dispatch Order

5.5.4 Design Specification

Structure Chart of the Dispatch Order

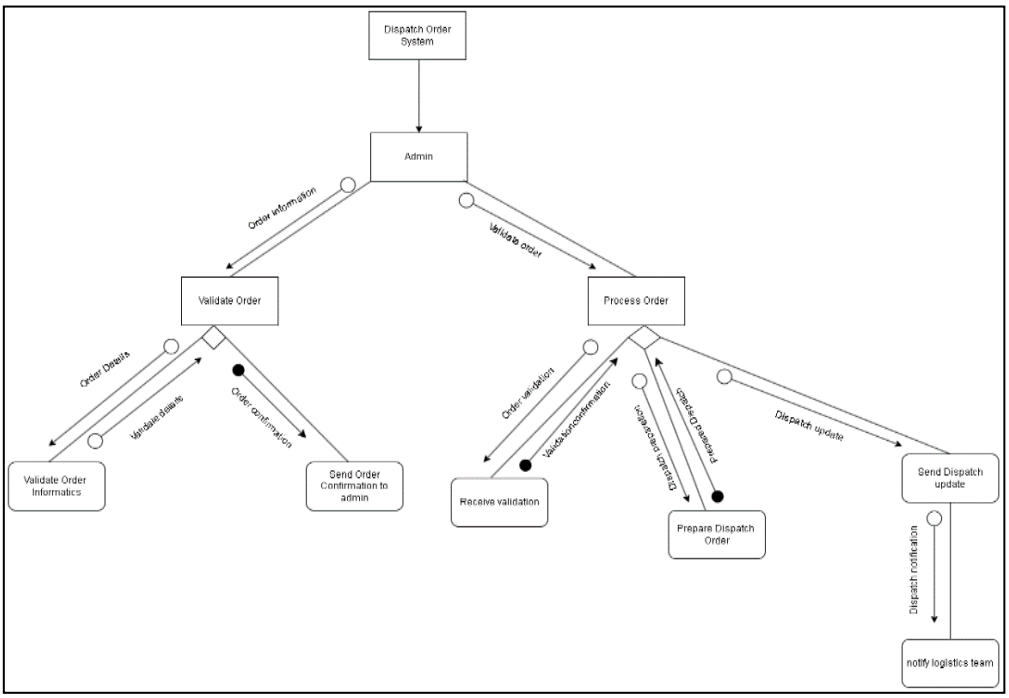


Figure 28: Structure Chart of Dispatch Order

The dispatch order process is all about making sure customers get their products accurately and on time. It send off with the admin receiving the order information and checking its validity. Once the details are confirmed and product availability is verified, the process moves on to preparing the dispatch. The order is then prepped for shipping, and the system updates the ERP (Enterprise Resource Planning) with all the dispatch details. Finally, an update is sent to the logistics team to manage the physical delivery, ensuring the order reaches the customer as promised. This seamless process guarantees a smooth transition from order placement to delivery, involving crucial checks and clear communication between various departments.

Module Specification Module Name: Dispatch Order

Conclusion:

Essentially, this Dispatch Order System structure chart provides a step-wise pictorial representation of the entire process of order dispatch in an organization. It details the process involved in receiving order details, their validation, dispatched redirection, ERP system update, and alerting the logistics team. The systematic procedure followed here makes sure that all essential checking and communications are made, ensuring right and timely delivery of products to customers.

6. Summary

The coursework involved application of software engineering principles in developing a system for Global Tech Company using a real-world example or scenario for the best development and its optimum implementation managing their data and records more effectively and efficiently. The software covers various aspects like user registration, customer login or admin login, orders, payment, sales management, product management, and purchase which are functions of admin, along with storing information about the customers from their personal credentials to their order details and history.

The development process was carried out in different phases, starting with the data flow diagrams for a complete understanding of the system open to anyone for understanding, progressing to entity-relational diagram, and then Structure charts for developing the software in the best way, and as efficiently as possible. We used Yourdon's structure model including its components for Structure chart, Entity-Relational diagrams, and data dictionaries for datastore to understand and implement the project more effectively. It was challenging to deal with the diagrams and the charts as they presented challenges, as clarity and precision were very crucial for a better software.

Working in groups posed additional challenges, which required much more coordination to ensure everyone followed the workflow. Effective Communication within the team was very essential, involving discussions, frequent meetings, research and study through various resources available in the internet.

Even though numerous challenges arose, our each and every member of the team worked together and successfully completed the project. The knowledge and every insight gained from software engineering helped us adapt and work in a systematic approach which led to skill improvement, better teamwork, and the ability to tackle the real-time projects. The flexibility, convenience, reliability, and management of the project were improved and enhanced with the help of Software Engineering.

Overall, in summary, the whole experience was a lot beneficial and rewarding. The coursework not only addressed specific needs of Global Tech Company,

but also provided us valuable experience, insights and skills for software engineering including all the key processes and steps essential to the system development which can be applied to our future projects.

7. References

Anon, 2024. *Structure Charts – Software Engineering.* [Online] Available at: https://www.geeksforgeeks.org/software-engineering-structure-charts/
[Accessed 8 January 2025].

Austin, C., 2024. *splunk.com.* [Online] Available at: https://www.splunk.com/en_us/blog/learn/data-dictionary.html [Accessed 23 12 2024].

Belcic, I., 2024. *ibm.com.* [Online] Available at: https://www.ibm.com/think/topics/entity-relationship-diagram [Accessed 15 12 2024].

Charles, M., 2024. *Functional vs Non Functional Requirements.* [Online] Available at: https://www.guru99.com/functional-vs-non-functional-requirements.html
[Accessed 15 12 2024].

Choudhary, A., 2024. *pharmaguideline.com.* [Online] Available at: https://www.pharmaguideline.com/2022/01/process-specifications-input-output-design.html
[Accessed 23 12 2024].

Global, Q., 2024. *Writing Assumptions and Constraints -SRS.* [Online] Available at: https://qat.com/writing-assumptions-constraints-srs/ [Accessed 15 12 2024].

HubSpot, 2021. *Web Accessibility:.* [Online] Available at: https://blog.hubspot.com/website/web-accessibility [Accessed 15 12 2024].

Krüger, G., 2023. *perforce.com.* [Online] Available at: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document
[Accessed 07 01 2024].

Lindemulder, G., 2024. *ibm.com.* [Online] Available at: https://www.ibm.com/think/topics/data-flow-diagram [Accessed 15 12 2024].

Lindsey, C. H., 1977. Structure Diagram. *Structure charts a structured alternative to flowcharts,* 12(11), pp. 36-49.

Moore, C., 2024. *What is Non-Functional Requirement in Software Engineering?.* [Online] Available at: https://www.guru99.com/non-functional-requirement-type-example.html
[Accessed 15 12 2024].

Robinsom, S., 2024. *techtarget.com.* [Online] Available at: https://www.techtarget.com/searchdatamanagement/definition/data-flow-diagram-DFD
[Accessed 15 12 2024].