CS5002NI Software Engineering

20% Group Coursework

AY 2024-2025

Credit: 30

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1	Arun Nagarkoti (C10)	np01cp4a230252	23050194
2	Ananta Gurung(C10)	np01cp4a230411	23049318
3	Pujan Poudyal (C10)	np01cp4a230419	23050282
4	Sunil Phuyal (C10)	np01cp4a230272	23050275
5	Yograj Rijal(C10)	np01cp4a230273	23050308

Assignment Due Date: 9 January 2025

Assignment Submission Date: 9 January 2025

I confirm that I understand my coursework needs to be submitted online via Google Classroom 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

This assessment accounts for twenty percent of our module and required group work. The main goal is to improve teamwork, documentation and apply Software Engineering principle especially Yourdon method. It is also about building project management skills and learning how to work better as a team.

The task is fully based on fixing problems faced by Global Tech Corporation to set up an inventory management system (IMS) for their warehouse in Nepal. Their earlier project failed because of unclear goals, poor planning, week documentation and poor communication between staff. These problems caused delays, budgets overruns and unhappy customers. To avoid the same mistakes, this project will focus more on better project management, system design and analysis.

Each team member will have to do tasks like processing orders, managing access, tracking sales, product details, reports and so on. We will also need to create detailed documentation, including Business Cases, System Requirement Specifications (SRS), Entity Relationship Diagrams (ERD), Data Flow Diagrams (DFD). Along the way we will track progress, document assumptions and fix any issues that come up.

The overall goal is to build a scalable and reliable inventory management system that meets the needs of the warehouse. Also, this assessment gives us experience in working as a team and applying software engineering practices and learning how to analyze systems, document processes and so on. At the end of the assessments, we should have better project management skills and be more prepared for real world software development challenges.

2. Business Case

Business case is a way to explain why starting a project or taking on a task makes sense. It lays out the reasons for solving a problem or grabbing an opportunity. It is written in a clear and organized format, but it can also be shared as a presentation or even a verbal pitch (“What is a business case?,” n.d.). A business case helps to make the best use of resources by showing why a project is worth the investment. It ensures the time, money and effort go toward initiatives that offer the most value and highest return on investment (Atlassian, n.d.).

Business case for Global Tech Corporation who introduce a new and advance Inventory Management System (IMS) are given below:

Project Name	Inventory Management System
Project Manager	Nabin Gurung
History (Current Situation)	• The scope was unclear and documentation was also lacking. • Resources were not used efficiently which caused the project to go over the budget. • Risks like system downtime were disregarded and proper reviews were done only after launching the system.
Goals:	Develop a better inventory system to make warehouse operations smoother, improve inventory tracking and reduce costs.
Scope:	• User accounts with different access levels (admin, customers, and suppliers). • Real-time stock updates and purchase order tracking and detailed reports for sales and inventory.
Plan:	• Start with clear deliverables and stick to them. • Use resources wisely with milestones to track progress • Review the project regularly to make sure it is on track.

Benefits	Reduce inventory holding costs by 25% within 1 month by optimizing stock levels. Lower labor costs by 20% within 3 months by automating manual tasks. Boost order accuracy by 97% within 5 months by implementing a reliable tracking system and errors in inventory management. Provide real-time order and inventory tracking within 8 months by giving immediate updates after deployments Prepare for unexpected challenges from the 7th month through proper testing and planning. Speed up operations by reducing task time by 40 % within 11 months through improved system integration.
Financial Analysis(**Costs:**	Initial Investments: NPR 2,000,000 Development Costs: NPR 500,000 Training and Implementation: NPR 300,000 Ongoing Maintenance (Annual): NPR 100,000
ROI Analysis: **)**	Operational Cost Saving: NPR 400,000 per year Improved Inventory management: NPR 300,000 per year Reduce Labor Costs: NPR 200,000 per year We will get our money back in 18 months 80% return over 3 years Total benefits:1200000 in 3 years projections
Timeline	Project Start: January 2025 in iterative models, Duration: 13 months Key Milestones: Phase 1: System design and development (2-3) months Requirement gathering, System design, Vendor selection, System development Phase 2: Testing and implementation (5-7) months Integration planning, Initial planning Phase 3: Full deployment and training (8-11) months Users' trainings, Data migrations, Go-live preparations Phase 4: Post-implementation and review: (11-13) months

	System monitoring and user support
Human Resources: **Technical resources:**	Project Manager (1), Business Analysts (3), System Developer (4), Training Specialists (3), Quality Assurance Team (2). Computer Hardware, software licenses, Development tools, testing set ups and training rooms.
Risks	System integration challenges. Poor communication between staff. Schedule delays. Extra Expenses. Data accuracy issues.
Success Measure	The system runs almost perfectly (99.9% uptime). Orders are nearly always correct (99 accurate). Stock records are almost accurate (99%). Tasks are 40% faster. Staying on budget. Meeting saving goals.
Future steps	Plan everything carefully. Build the perfect team. Select the suppliers. Secure project funding approval. Start discussions with all stakeholders.

Table 1: Business Case

3. SRS

3.1 Functional Requirement:

1. **User Management and Access Control:**

- The users (Admin and Customer) must register and login with unique credentials.
- The system will validate the credentials while logging in.
- The customer search products and makes purchases, while the admin manages the inventory.

2. **Product Management**

- Admin adds new product data in inventory through add product process.
- Users (Admin, Customer) can view product details like price, supplier details from view product function.
- Customer can also compare the price of similar product while viewing the product.

3. **Purchase Order Management**

- The admin will send purchase requests for the products required to the suppliers by email using the add purchase function.
- Admin can also see the past purchase history which is stored in order database.
- Customer can place order for the product they have selected if the product is available in inventory.
- Suppliers send inventory to warehouse and admin is notified through emails.

4. **Sales Management:**

- The admin can view the details of the sale such as address and dispatching details.
- After viewing the sale order, admin will dispatch order if order status is still pending.

5. **Payment Process:**

- The customer (buyer) can pay for the products via different payment methods like Esewa, PayPal, Khalti etc. The customer can also compare prices of similar products.

6. **Report Generation:**

- The purchase report includes information on purchase details such as quantity, delivery details and date, while the sales report includes total sales and revenue (profit and loss).
- The generated report will be sent to admin and customers according to their sales and purchases report.

7. **Update Stock:**

- When a customer orders or supplier delivers inventory, admin is notified and admin manually updates the stock level to provide real time stock level.

3.1 Non- Functional Requirement:

3.1.1 Design and Implementation Constraints

- The project must follow structure software engineering methodology (such as the Yourdon approach).
- The system should be compatible with Windows and Linux environments.
- IMS will be using relational database management system such PostgreSQL.
- In order to avoid costly updates or tools, development has to stick to defined resource and budget constraints.
- The system needs to work smoothly in existing hardware and software environments.
- The system will be developed using JavaScript, React for the frontend and java (spring boot) for backend.
- The system must integrate existing third-party solutions for payment platforms like PayPal, Esewa, Khalti.

3.1.2 External Interface Required:

User Interfaces:

- Admin will have a dashboard to monitor and manage products, view sale report, process orders and update stocks in inventory.
- Customer/Buyer will have features of viewing product details and compare, secure checkout process.

Hardware Interfaces:

- Required servers at least 16 GB RAM and 1TB storage.
- Support client devise such as pc and smartphone
- Work with standard printers

Software Interfaces:

- Integrates with database management systems, payment gateways, email, services, reporting tools.

Communication Interfaces:

- HTTP protocol for secure communication.
- Implement endpoints for third-party integration.
- Supports email notification.
- Backup communication channels.

3.1.4 Other non-functional Requirements:

- **Performance**: The system should give response to user activities in less than 2 seconds.
- **Scalability**: The system should be developed to handle increasing user and inventory size without dropping performance.
- **Security**: Data encryption should be implemented; secure user authentication should be maintained and payment information should be protected in the system.

- **Accessibility:** In order to ensure simplicity for people with disabilities, the system must follow the guidelines for accessibility.

4. Detailed specification of Group Task

4.1 Environment model specification

4.1.1 Data Flow Diagram (Context Level)

DFD Level 0 is also called a Context Diagram. It contains an overview of the whole system or process. Its main objective is to provide a clear idea of the system as a single high-level activity together with connection to external entities. Anyone who looks at it should understand easily like developers, business analysts, stakeholder (lucid chart, 2024).

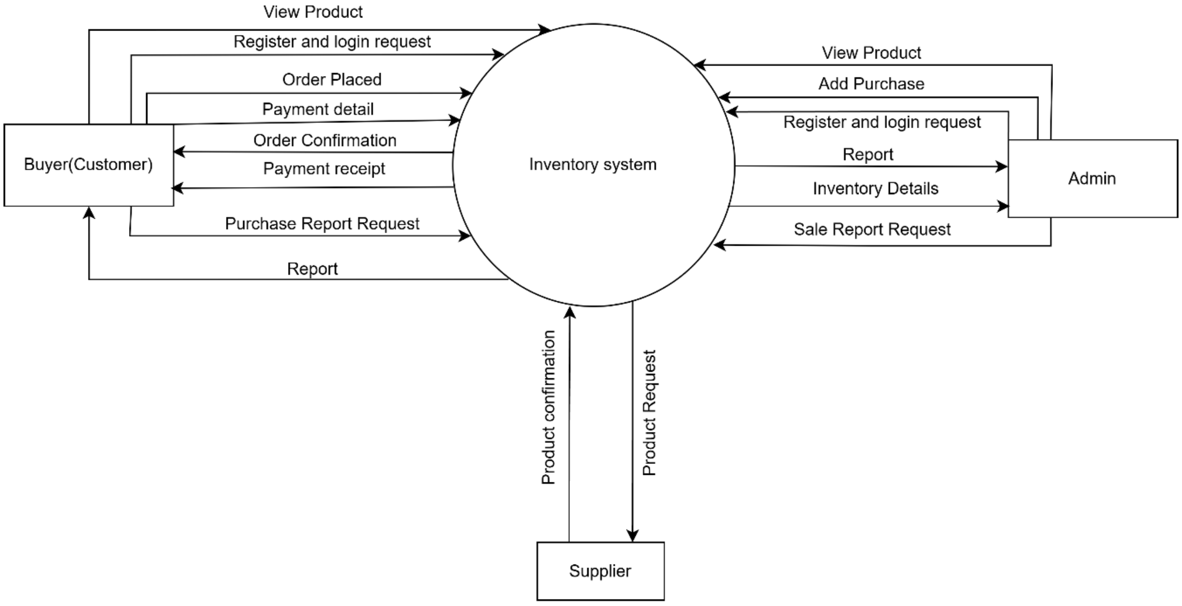

Figure 1: Context Level DFD

4.1.2 Data Flow Diagram (Level-1)

In Level 1 DFD, context diagram or level 0 DFD is breakdown multiple processes. The main purpose of this level is to breakdown the high-level process of context level into sub processes (javatpoint, 2024).

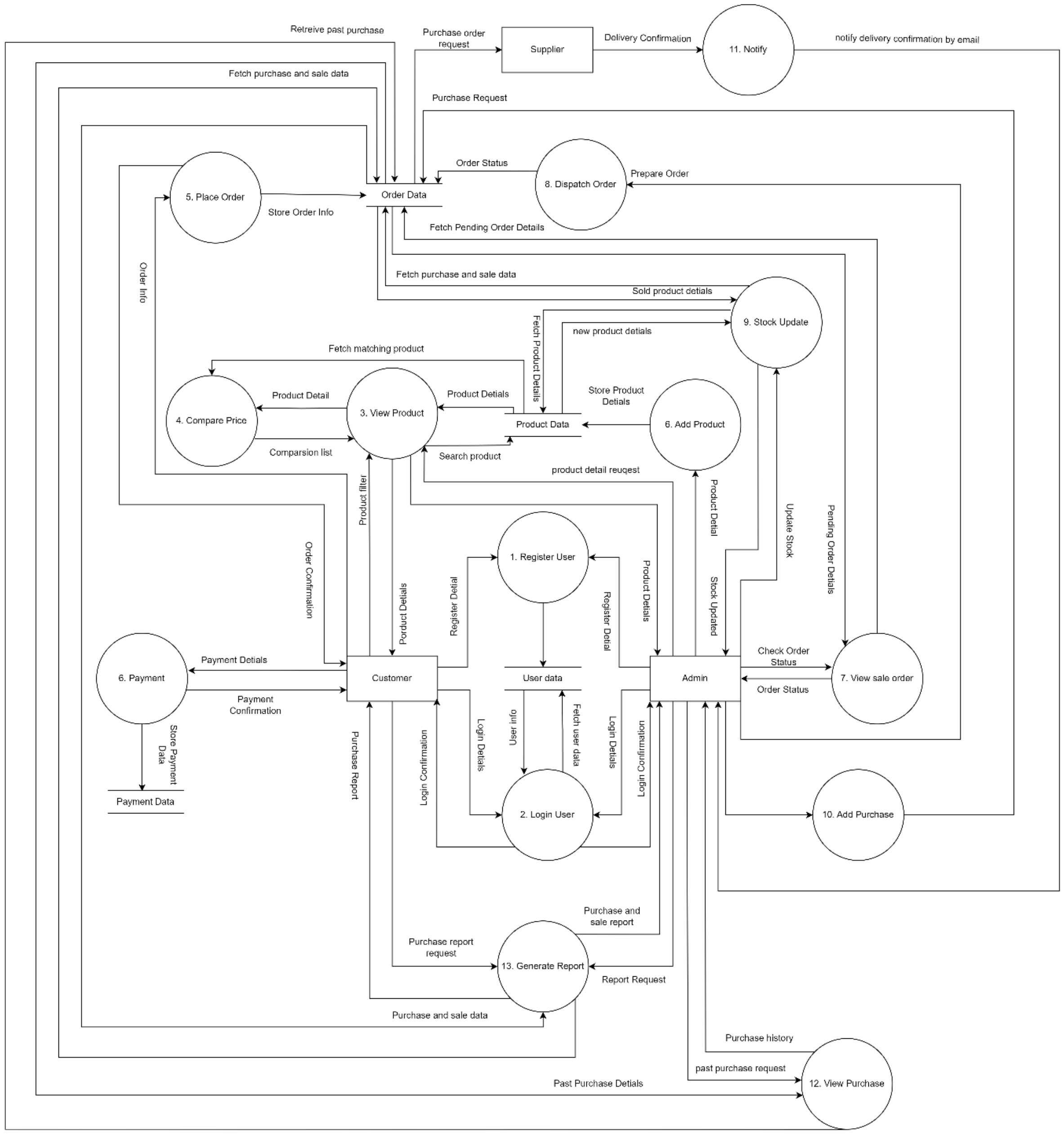

Figure 2: Level 1 DFD

4.1.3 Data Flow Diagram (Level-2)

Level 2 breaks down 1-level DFD sub process into more detailed view of system.

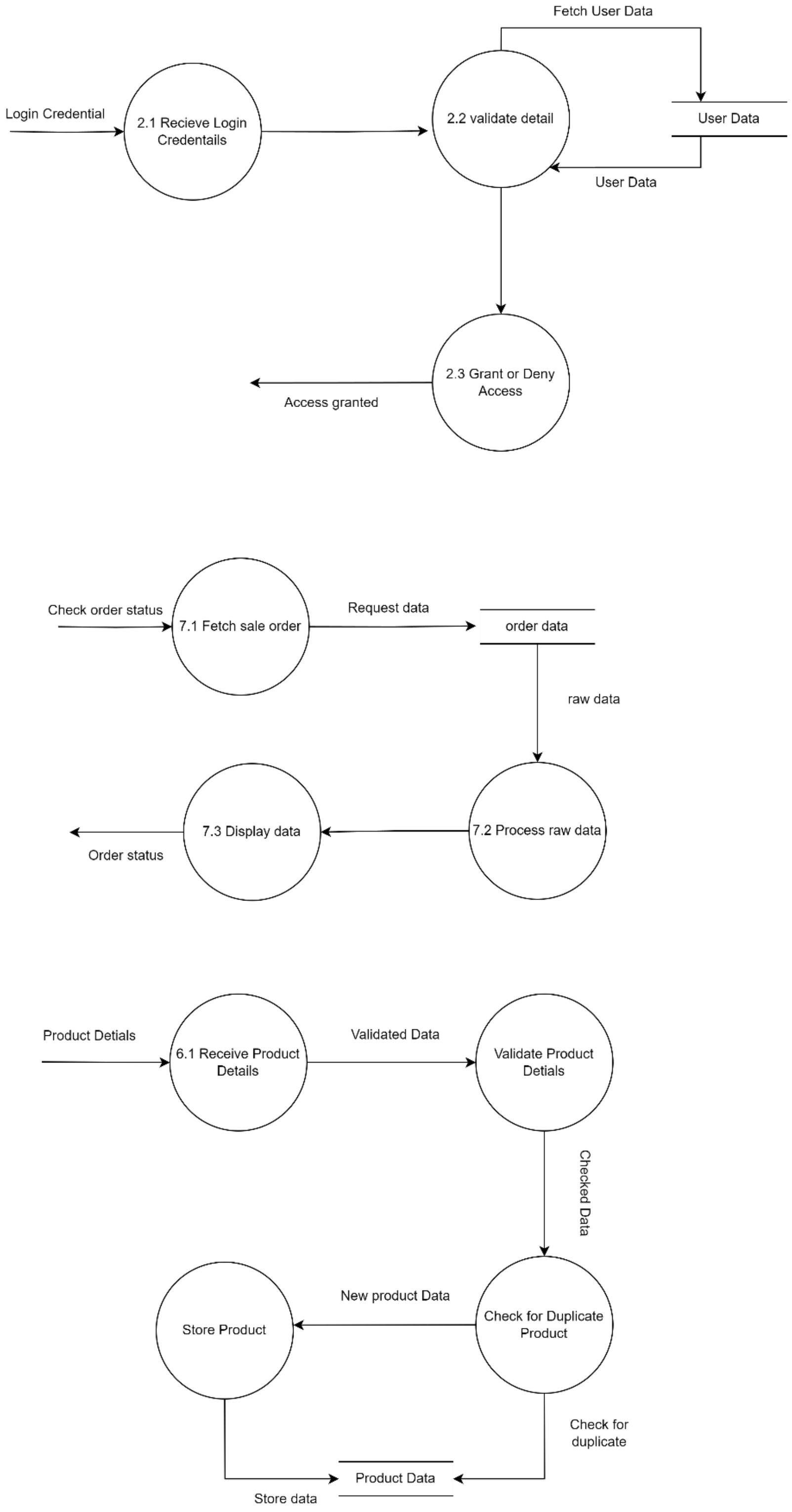

Figure 3: Level 2 DFD

4.2 Internal model specification for the system

4.2.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram, also known as ERD. It is a type of structural diagram for use in database design. It shows the relationship between entities like people, things or concepts in a database. Entities are also present with attributes belonging to it (smartdraw, 2024).

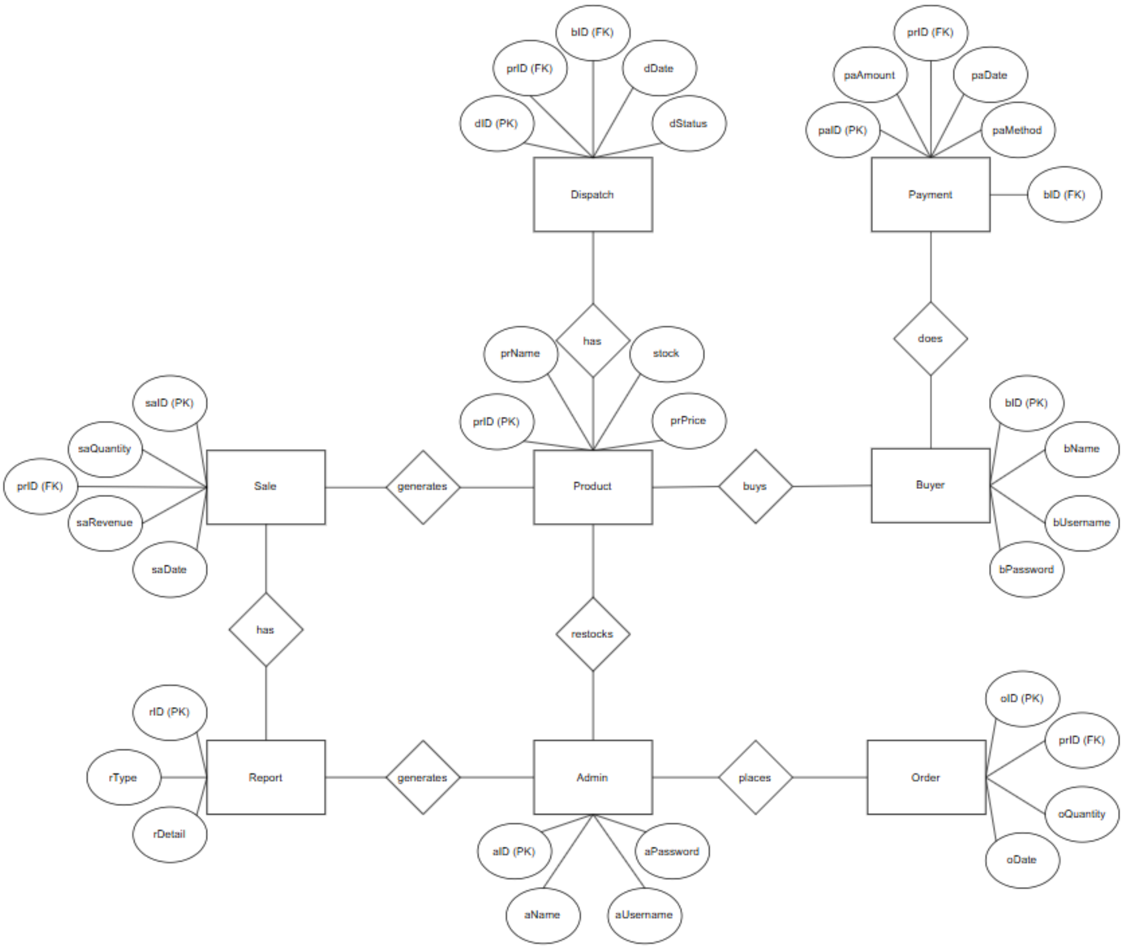

Figure 4: ERD

4.2.2 Data Dictionary

Register user (command)

Login user (command)

Place order (command)

Generate report (command)

Payment (command)

Stock update (command)

Dispatch order (command)

Notify admin (command)

Admin = aID, nName, aUsername, aPassword

Buyer = bID, bName, bUsername, bPassword

Product = prID, prName, Stock, prPrice

Payment = paID, paAmount, paDate, paMethod

Order = oID, oDate, oQuantity

Sale = saID, saQuantity, saDate, saRevenue

Report = rId, rDetail, rType

Dispatch = dId, dDate, dStatus

User data = Admin and buyer detail

Product data = Product Id, price, stock

Order data = Purchase, Sale

Payment data = Buyer’s payment detail

4.2.1 Process specification for elementary processes

A process specification is a way to document, evaluate, and explain the logic and formulas used to generate output data from process input data. Its goal is to define engineering and regulatory requirements and processes. Clear and detailed process standards are necessary to produce consistent, high-quality data (Rose, 2012).

Process Number 2.1:

Name: Receive Login Credentials

Description: This process receives login credentials such username and password from users (Admin, customer).

Input Data flow: Username, password

Output Data flow: Access granted

Process Logic:

- Login details are received from the user.
- Now the details are passed to next sub process i.e. 2.2

Process Number 2.2:

Name: Validate Detail

Description: This process checks details received from the user.

Input Data flow: Login Detail

Output Data flow: Validate

Process Logic:

- Login detail is compared with data from database
- Now the details are passed to next sub process i.e. 2.3

Process Number 2.3:

Name: Grant or Deny access

Description: This process provides access or deny it.

Input Data flow: Validate date

Output Data flow: Access granted

Process Logic:

- If detail is validating user will get access according to the role access
- Now user is logged in system.

Process Number 7.1

Name: Fetch sale order

Description: This process request sale order data from order data

Input Data flow: check order status

Output Data flow: raw data

Process Logic:

- Retrieve sale order data form order database
- And send raw data to next sub process i.e. 7.2

Process Number 7.2:

Name: Process raw data

Description: This process refines raw data into proper readable data

Input Data flow: raw data

Output Data flow: processed data

Process Logic:

- Raw data like Json type format are refine into more readable simple format
- Now after getting simple format data next sub process start i.e. 7.3

4.3 Design specification

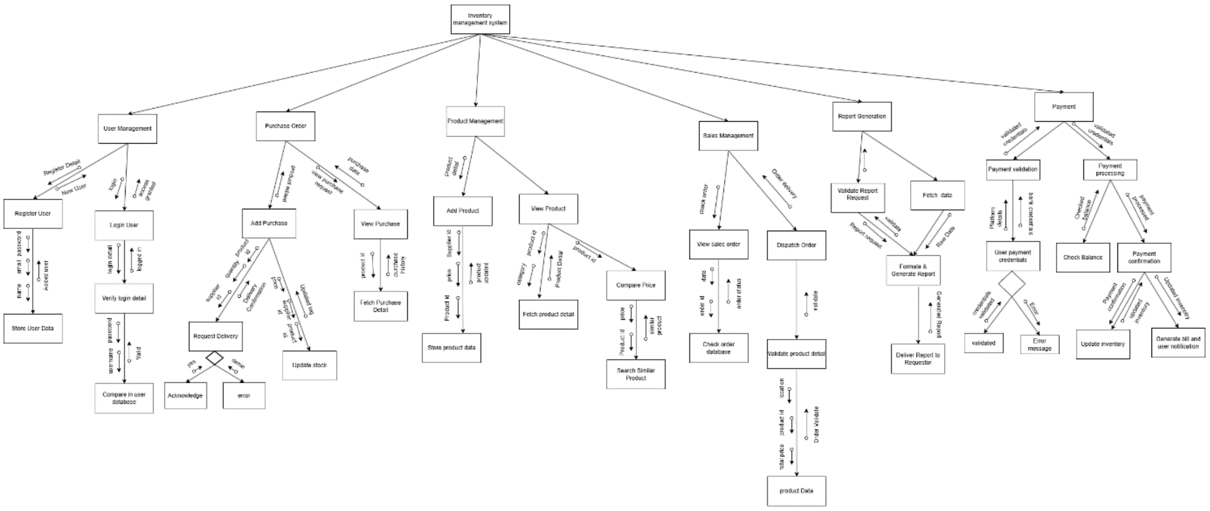

Figure 5: Structured Chart

A structure chart is a visual representation diagram which is typically created using data flow diagram (DFD) as a starting point. Compared to data flow diagram, the function and procedures are broken down into lowest manageable form, the system is analyzed more clearly and deeply (Goel, 2024).

The chart shows how an Inventory Management System is organized and works. It is broken into different parts, each handling specific tasks. First, there is User Management which takes care of registering new users and logging them in. When a user signs up, their details, like name, email, and password, are saved in a database. For login, the system checks the credentials against what is stored and either grants access or shows an error. Next is Purchase Order, where users can add or view purchases. When adding a purchase, the system validates the details, updates the stock, and requests delivery. Users can also check their purchase history or specific purchase details by entering the product ID. Then there is a Product Management which focuses on adding and viewing products. It stores things like product name, price, and stock, and helps users compare prices or find similar items The Sales Management section handles sales orders. Users can view orders, validate product details, and dispatch them, which updates the order status. Report Generation is all about creating reports. It starts by checking if the request is valid, then fetches the required data, formats it, and delivers the final report. Finally, the Payment section manages payments. It validates user payment details, checks account balances, processes the payment, and updates inventory. Afterward, it generates a bill and notifies the user. The flow of information is shown with arrows, and decision points, like “success” or “error,” are represented with diamonds. Overall, the chart gives a clear picture of how the system’s different parts work together.

4.4 Progress Logs

4.4.1 Assumptions

The following assumptions have been done for the design and development of the Inventory Management System:

- **User Roles and Access:** There will be two main user roles in the system: Admins and Customer. Admin will have the ability to manage inventory, process sales and generate reports while Customers will be able to browse products, make purchases and view reports related to their activities.
- **System Availability:** It is assumed that the system is expected to be available 24 hours with only minimal downtime for update and maintenance. This will ensure that the user can access the system anytime they need.
- **Data Management**: All important data, including inventory and sales records will be stored in one central database. Regular backups will be performed to safeguard against data loss. To keep everything accurate input validation will be used to reduce errors and maintain data quality.
- **Report Generation:** Reports will be generated in real time using predesigned templates. Users will be able to view these reports right on their screens or download them as PDF for easy references.
- **Payment Integrations:** The system will support trusted third-party payment services like Esewa, Khalti and PayPal. A secure API will be utilized to process payments safely and efficiently.
- **Security Measures**: The user data will be protected through encryption, secure origin and role-based access control to protect sensitive information
- **Technical Requirements:** The system is expected to work with servers that have 16 GB RAM and 1 TB storage and support PC and mobile devices.
- **Project Timeline and Budget**: The project follows a 13-month iterative development plan with fixed budgets assuming no major changes in resource allocation.

- **Training and support:** User will be trained to make sure they can use the system without problems and support teams will handle any issues after the system is launched.
- **Testing and Risk Management:** The system will go through testing for performance, security and features before final deployment.
- **Communication:** Regular team meetings and progress logs wil ensure clear communication and tracking of deliverables.

4.4.3 Omissions/inconsistencies

While analyzing the project requirements and specifications, we noticed a few gaps and unclear areas that need attention. Fixing these issues early will help make sure the Inventory Management System (IMS) works smoothly and meets its goals. The following omissions and inconsistencies were observed:

- **User Roles and Permissions:** The roles of admin and customer are mentioned but it is unclear if additional roles like suppliers will need separate access permissions to manage inventory data.
- **Report Customization:** The requirements specify purchase and sales reports, but it is not clear if these reports can be customized based on specific times, categories, or filters.
- **Payment Methods:** The system plans to integrate payment gateways like Esewa, Khalti and PayPal but there is no mention of whether offline payments like cash or bank transfers will be supported.
- **Data Backup and Recovery:** No details are provided about how data backup and recovery mechanism will handle potential system failure or data loss.
- **Testing Requirements:** The focus is only on functional testing but other testing types such as security testing, performance testing and loas testing are not mentioned.
- **Risk Management Plan:** Although risks are briefly listed, there is no detailed risk management strategy to handle delays, budget overruns or failures.
- **Feedback Mechanism:** There is no information about whether users can submit feedback or report issues after deployment to improve the system.

4.4.3 Group member responsibilities, groups meeting

Group responsibilities

Members Name	**Assigned tasks**
Ananta Gurung	Erd, Data dictionary, module specification creation and structured chart for individual task of dispatch order.
Arun Nagarkoti	Group task level 1, structure chat of group task, Preparation of SRS, individual task of Payment Integration, DFD
Pujan Poudyal	Module specification, DFD, structured chart of individual task, group task level 2, omission.
Sunil Phuyal	Business case preparation, module specification creation for “Generate Report”, assumption, delivering final report
Yograj Rijal	process specification, Summary, Module specification, DFD, structured chart of individual task.

Table 2: Task Breakdown

Group Meetings

Date	**Location**	**Task**	**Outcome**	**Task for Next meeting**
15 November 2024	College Library	Task allocations and milestone planning	Task assigned and timeline finalized.	Discussion of Business Case
2 December 2024	Online (Discord)	Business Case review and edits	Business Case completed and finalized	SRS and analysis

7 December 2024	College (Kumari-Hall)	SRS and analysis artifacts preparation	Initial draft of DFD and ERD creation.	Making DFD and ERD
14 December 2024	Online (Discord)	Finalized DFD and Erd	DFD and ERD were completed.	Individual task discussion
17 December 2024	College (Canteen)	Individual task updates and SRS review	Confirmed draft of SRS and ready for review.	Reviewing the work done till now
20 December 2024	Online (Discord)	Progress log finalization.	Finalized diagrams and layouts	Documentation and submission
23 December 2024	Online(Discord)	All Documentation things and prepare for milestone submission.	Complete Documentation and ready for review and submission	Remaking of Structured chart
28 December 2024	Cafe	Structured chart of group and Individual	Again, made changes for structure chart after reviewing	Final Recheck of documentation
06 January 2025	Kumari hall	Final Document and recheck	Checking and making changing if required in document	Submission of report
09 January 2025	Library	Submission of final report	Analyzing overall document and submission	null

Table 3: Meeting log

5. Detailed specification of INDIVIDUAL task

5.1 Purchase Order (Pujan Poudyal)

5.1.1 Environment model specification

Context Level Diagram

The diagram shows how the inventory system connects with both the admin and the supplier. The admin shares Purchase order details with the inventory system and gets back updated inventory information. Meanwhile, the supplier provides Supplier details to the system. In short, the inventory system acts as a central hub, managing and updating inventory based on input from both sides.

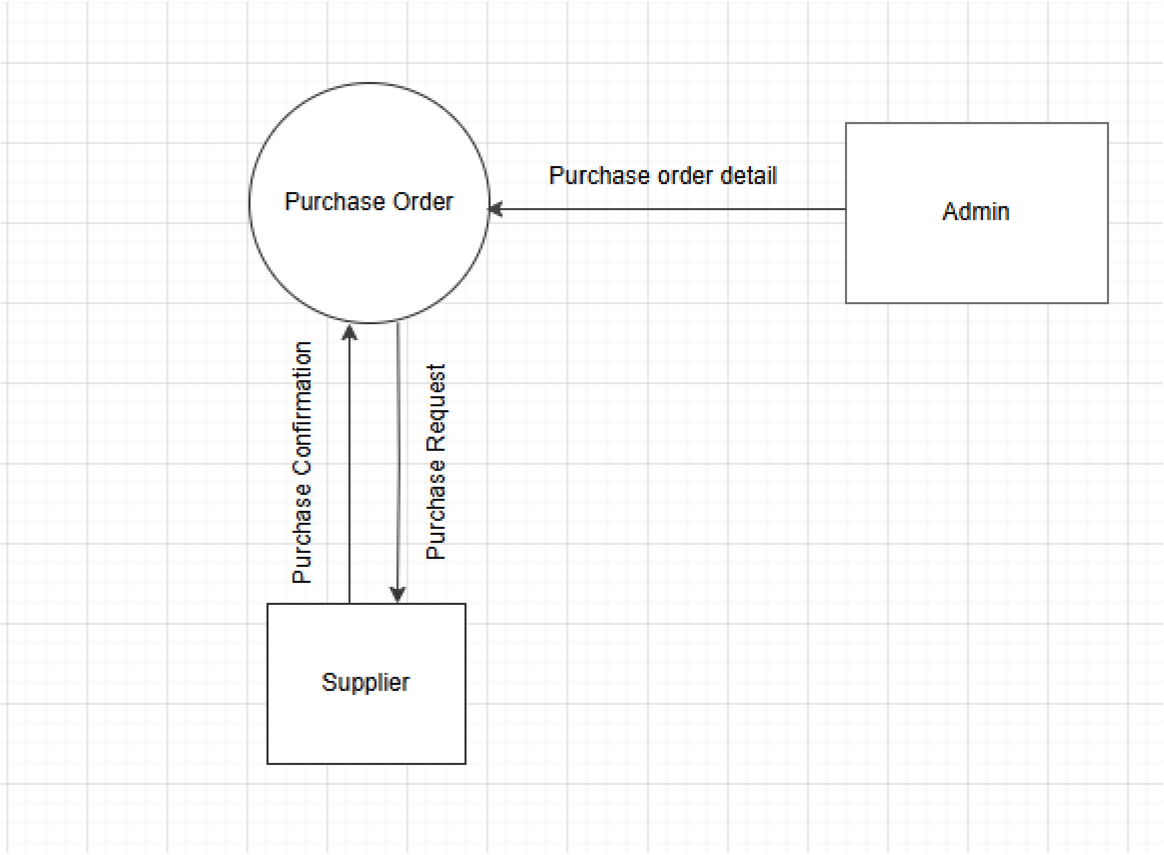

Figure 6: Purchase Order context lvl

5.1.2 Internal Model specification

Level-1 DFD

This Level 1 DFD shows how the purchase process works step by step. The admin sends purchase order details to the purchase order process. An order request is sent to the supplier. Once the supplier confirms the order, the inventory gets updated. The admin can also check the inventory status by sending a view request and receives confirmation in return. This system keeps everything connected and ensures clear communication between the admin, supplier and inventory system.

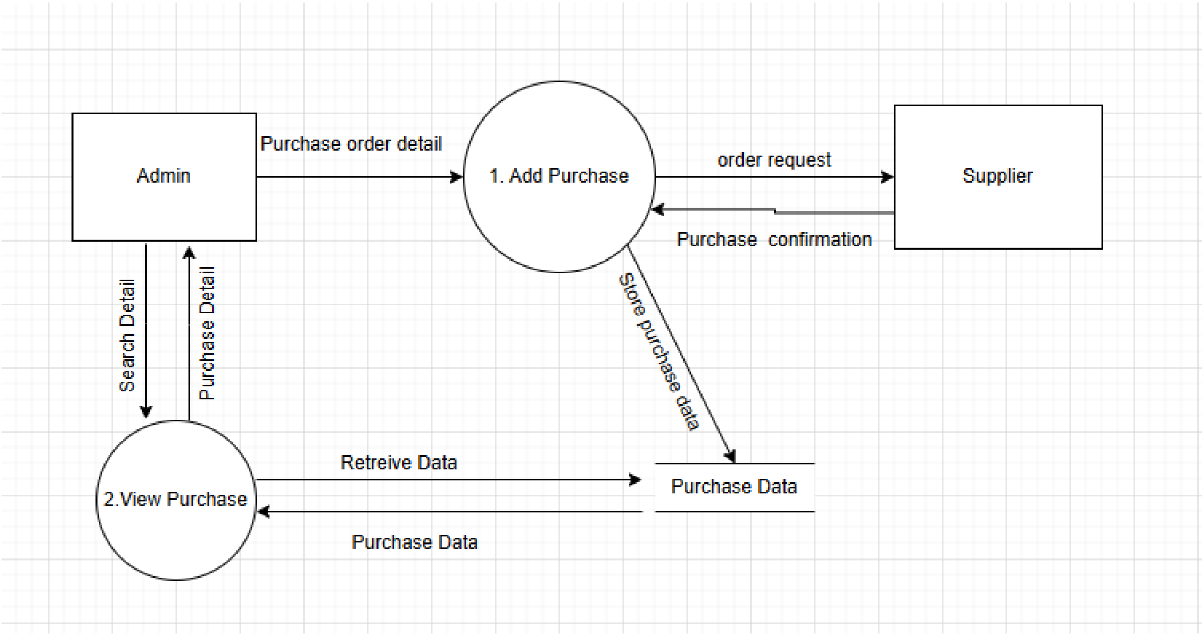

Figure 7: Purchase Order Level 1

Level-2 DFD

This diagram shows how the purchase order process works. The admin sends a purchase order to the system, which forwards it to the supplier. The supplier confirms the order, and the system updates the inventory with the new stock. The admin can view the order status and confirmation through the inventory system. The process ensures smooth communication and keeps inventory accurate.

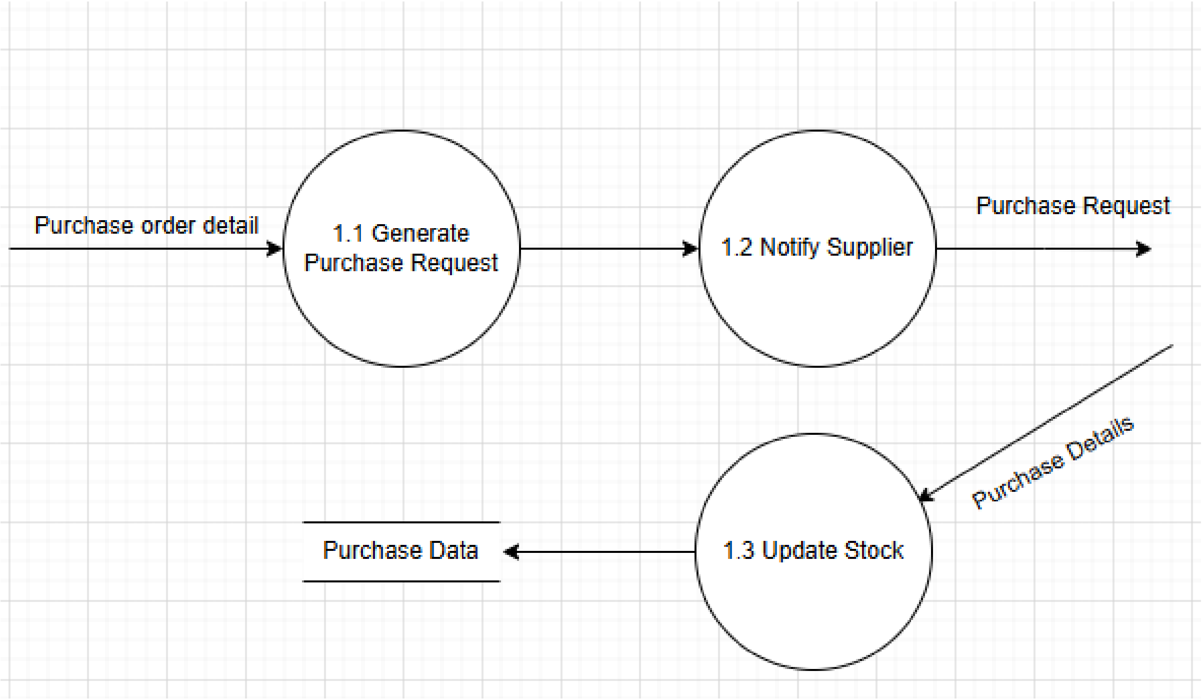

Figure 8: Purchase Order Level 2

5.1.3 Design Specification

Structure Chart

This diagram shows how a Purchase Order System works, focusing on adding and viewing purchases. It explains the steps and connections between different parts of the process. First, the system sends Purchase Order Data to the Add Purchase section, which creates a purchase request. Once the request is made, it gets confirmed, and the system notifies the supplier and updates the stock levels.

The View Purchase section handles requests for details about existing orders. It asks the system for information, retrieves the data, and then displays the purchase details. The diagram highlights how the system manages purchases, keeps suppliers informed, and ensures inventory is updated properly.

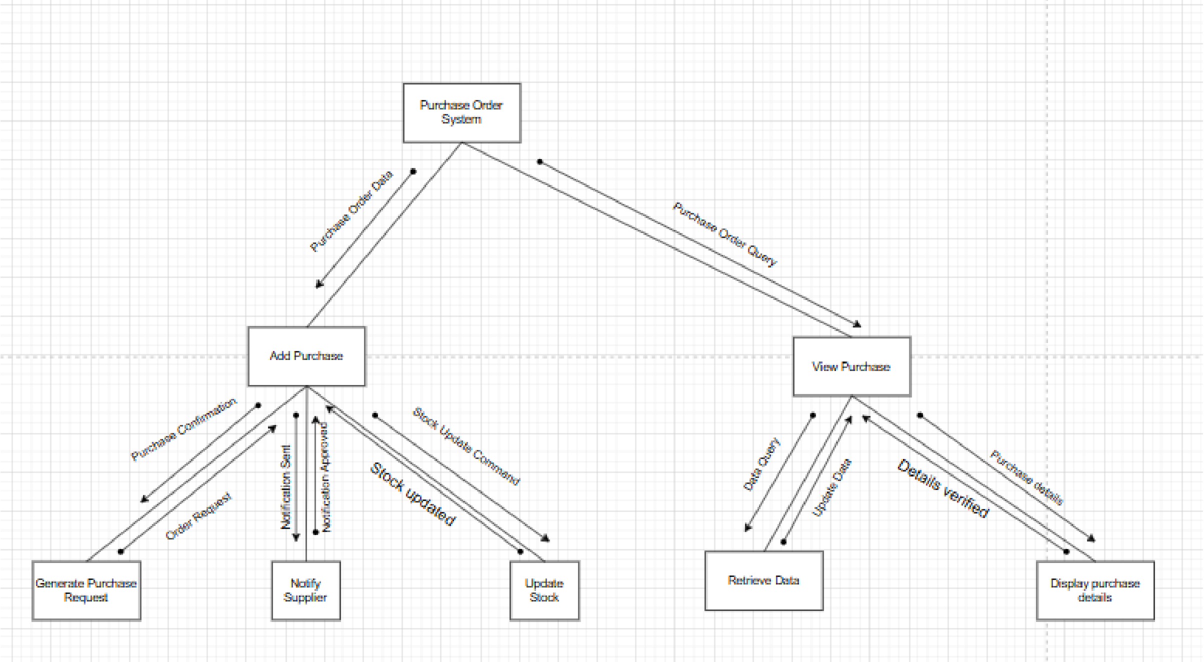

Figure 9: Structure Chart of Purchase Order

Module Specification

Name	Purchase Order System
Purpose	The purchase order system is designed to simplify and organize the buying process. It helps create purchase requests, informs suppliers, and updates stock levels, ensuring all purchase details are recorded accurately for easy tracking and management.

Pseudocode	Start
	INPUT PurchaseOrderData
	DO AddPurchaseModule
	    INPUT OrderRequest
	    Display “Purchase Confirmation Sent’
	ENDO
	DO NotifySupplier
	    DISPLAY “Notification Sent to Supplier”
	ENDO
	DO UpdateStock
	  DISPLAY “Stock Updated”
	ENDO
	ENDO
	INPUT PurchaseOrderQuery FROM PurchaseOrderSystem
	DO ViewPurchaseModule
	  DO RetrieveData
	      INPUT DataQuery
	      DISPLAY “Data Retrieved”
	ENDO

	DO DisplayPurchaseDetails
	    DISPLAY “Purchase Details Displayed”
	ENDO
	ENDO

Input Parameter	Purchase Order Data, Order Request
Output Parameter	Purchase Confirmation, Stock Updated, Purchase Details Displayed
Global Variable	none
Local Variable	OrderRequest, DataQuery
Calls	External and System
Called by	Admin

Table 4: Purchase order system table

5.2 Report preparation (Sunil Phuyal)

5.2.1 Environment model specification

Context Level Diagram

This diagram shows how the report preparation connects with both the buyer and admin. The buyer can request a purchase report and once the system processes it purchase report confirmation is sent back. The admin has access to both purchases and sale reports. They can request either a report or the system responds with a purchase report confirmation or sales report confirmation depending on what they asked for.

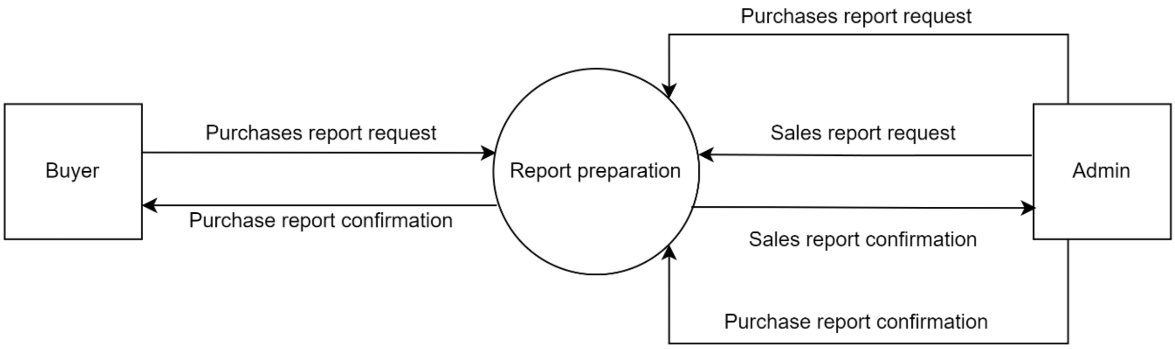

Figure 10: Report Generation context level

5.2.2 Internal Model specification

Level 1 DFD

This Level 1 DFD shows how the system handles reports for both the Buyer and the Admin. It focuses on two main users, the Customer and Admin, who can request and receive purchase reports. The process starts when a purchase report request is submitted for validation. The system checks the request validity and logs details in the Audit log for tracking the activity. Once the request is validated, it moves to fetch relevant data which pulls information from the purchase data store. Then, the data goes through the process and summarizes the data step. This processed data is passed to Format and Generation Report which creates a readable report. Finally, the delivery report to the requestor process sends the finished report to the Customer and admin. The audit log tracks all requests, and the purchase data store the data is accurate.

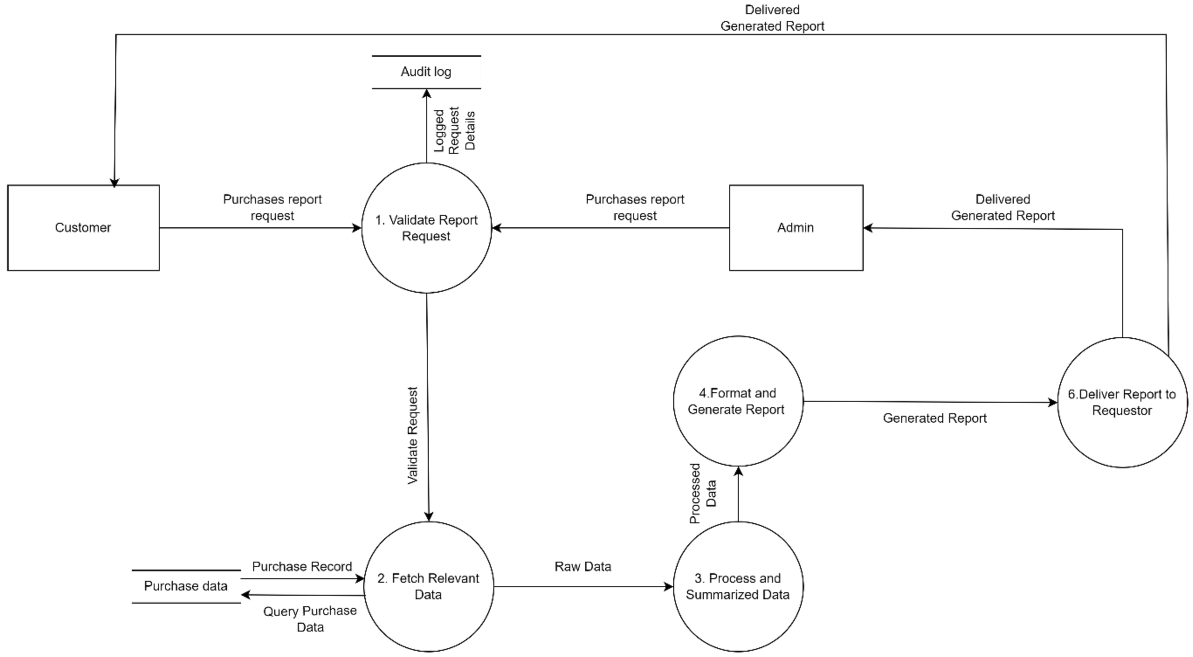

Figure 11: Report Generation level 1

Level-2 DFD

The level 2 DFD shows how the system handles report generation in detail. It starts with the customer logging in and requesting a report. The system verifies the request which logs it for tracking and fetches the necessary sales and purchase data. This data is then filtrated and summarized separately before it is combined into a single report.

The system formats the report using a predefined template and saves it in the database. Once done the customer is notified and they can download the report through the provided link. The entire process ensures smooth data handling, proper tracking and easy access to reports.

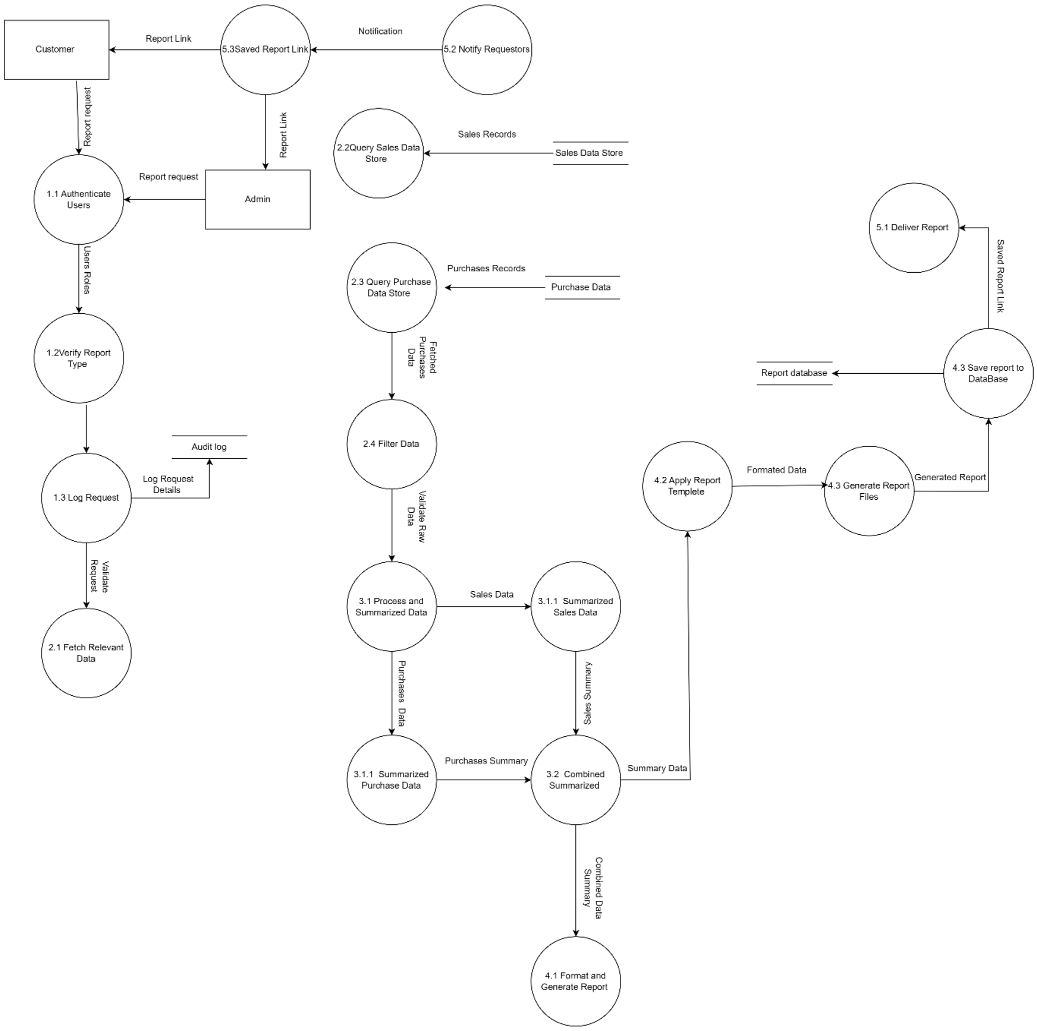

Figure 12: Report Generation level 2

5.2.3 Design Specification

Structure Chart

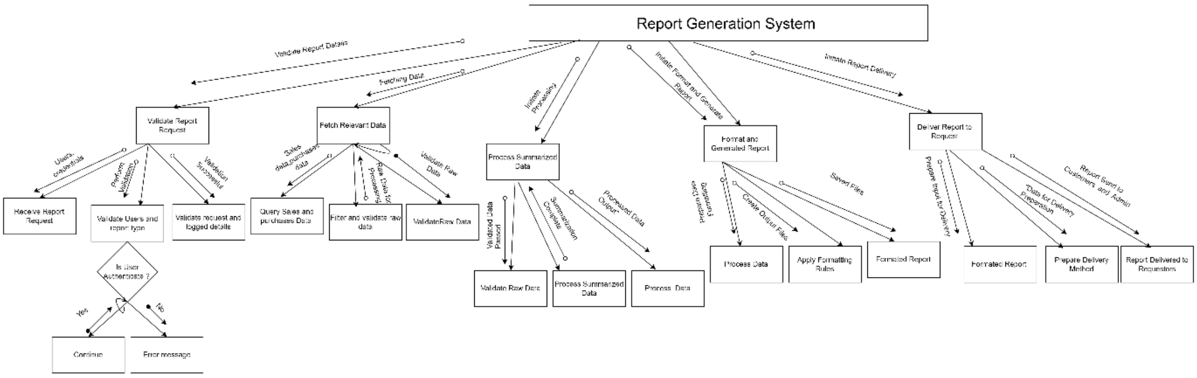

Figure 13: Report Generation structure chart

The structure chart explains how a Report Generation System works by breaking it into simple steps. It starts when someone requests a report. The system first verifies whether the user is authorized and checks if the requested report type is valid. If everything passes these checks, the system logs the request for tracking purposes. However, if the user isn't authenticated, the process breaks down, and they receive an error notification.

Once the user's access is confirmed, the system retrieves the required data. This data is then cleaned and verified to ensure it is accurate and reliable. Afterward, the system processes the data to summarize. Depending on the type of report, additional rules or calculations are applied to refine the information.

Next comes the formatting stage, once formatted the final report is generated in the requested format—whether that's a PDF, spreadsheet, or another option. Finally, the completed report is delivered to the user through their preferred method, like email ..

Module Specification

Name	Report Generation System
Purpose	The module handles report requests by validating user details, fetching, and processing relevant data,

	generating formatted reports and delivering them to the intended recipients.

Pseudocode	**BEGIN** ReportGenerationProcedure
	    INPUT reportRequest WITH parameters
	    CALL ValidateReportRequest
	    CALL FetchRelevantData
	    CALL ProcessAndSummarizeData
	    CALL FormatAndGenerateReport
	    CALL DeliverReportToRequestor
	END ReportGenerationProcedure
	
	PROCEDURE ValidateReportRequest
	    INPUT reportRequest
	    CALL ValidateUser
	    CALL ValidateReportType
	    IF validationStatus is FALSE THEN
	        PRINT " Invalid report request."
	        EXIT ReportGenerationProcedure
	    ELSE
	        LOG"Requestvalidatedsuccessfully."
	    END IF
	END PROCEDURE
	PROCEDURE ValidateUser
	    CALL DataValidationLibrary.Authenticate

	IF userID is NOT valid THEN
	    SET validationStatus = FALSE
	ELSE
	    SET validationStatus = TRUE
	END IF
	END PROCEDURE
	PROCEDURE FetchRelevantData
	    INPUT reportRequest
	    PurchaseDataStore and SalesDataStore
	    CALL ValidateAndFilterData
	END PROCEDURE

Input Parameter	reportRequest,validatedRawData, processedData, formattedReport
Output Parameter	validatedRawData, processedData,formattedReport
Global Variable	validationStatus, reportRequest
Local Variable	purchaseData, salesData, summarizedData, deliveryMethod
Calls	validateReportRequest,ValidateUser, ValidateReportType,FetchRelevantDataValidateAndFilterData, ProcessAndSummarizeData,SummarizeData,FormatAndGenerateReport DeliverReportToRequestor
Called by	Inventory management system.

Table 5: Report Generation System Table

5.3 Real-Times Stock Updates (Yograj Rijal)

5.3.1 Environment model specification

Context Level Diagram

This diagram shows how a real-time stock update system works. Admin sends update stock request to system, then system will ask for product delivery to supplier. Supplier will send confirmation system will update accordingly and notify admin.

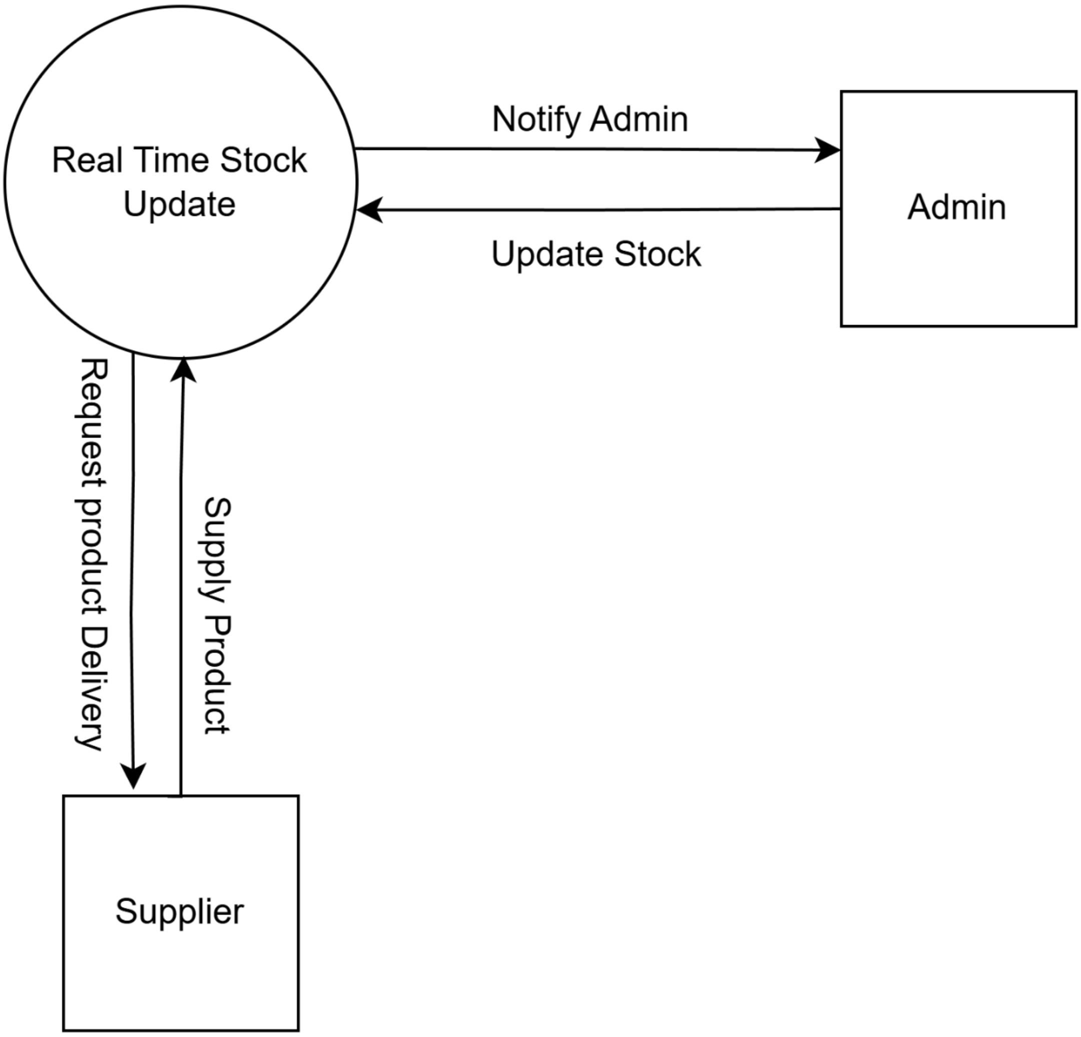

Figure 14: Real time stock update context level

5.3.2 Internal Model specification

Level-1 DFD

This is a more detailed version of context level. Admin will provide input details which contain update stock. First input detail (like stock details) is verified then request is sent to supplier. After supplier acknowledge and send order stock is updated, and admin is notified.

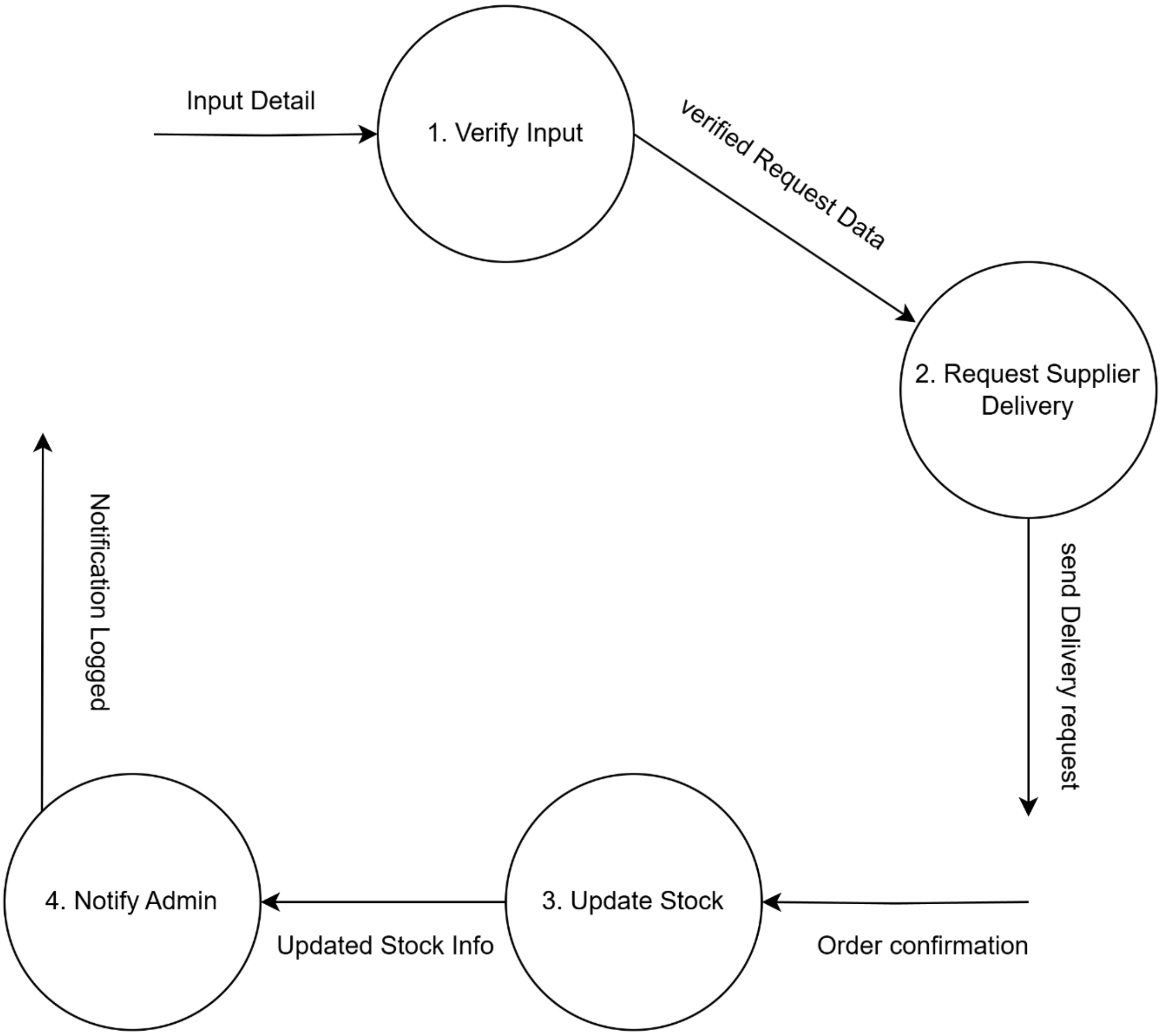

Figure 15: Real time stock update level 1

Level-2 DFD

In level 2, update stock data process has been breakdown even more. Verified Delivery is received from verified supplier delivery process then, current stock level is imported. After we get current stock data, the new stock level is being calculated, and the new stock level is sent to update inventory log. Lastly, it will need to be shown it reflect update level will be shown.

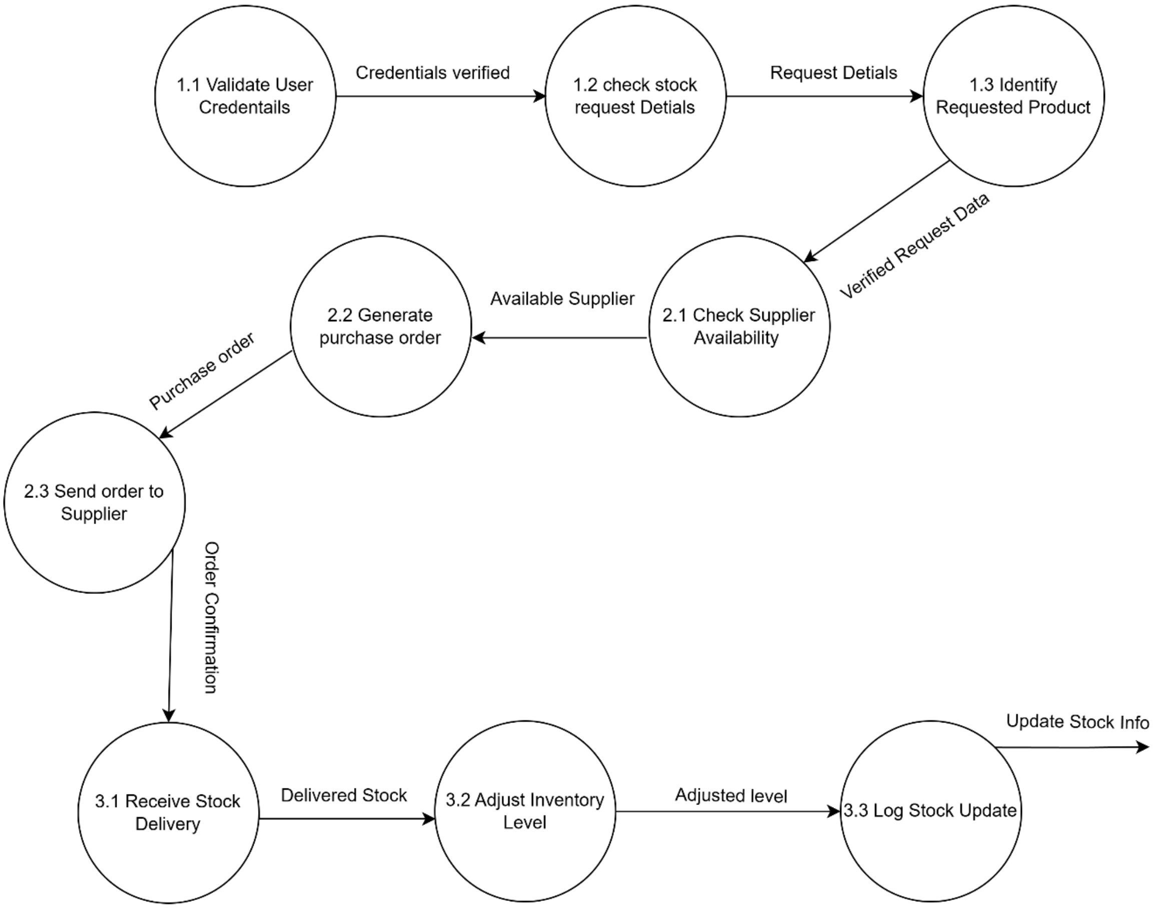
1.1 Validate User Credentails
Credentials verified
1.2 check stock request Detials
Request Detials
1.3 Identify Requested Product
Verified Request Data
2.1 Check Supplier Availability
Available Supplier
2.2 Generate purchase order
Purchase order
2.3 Send order to Supplier
Order Confirmation
3.1 Receive Stock Delivery
Delivered Stock
3.2 Adjust Inventory Level
Adjusted level
3.3 Log Stock Update
Update Stock Info

Figure 16: Real time stock update level 2

5.3.3 Design Specification

Structure Chart

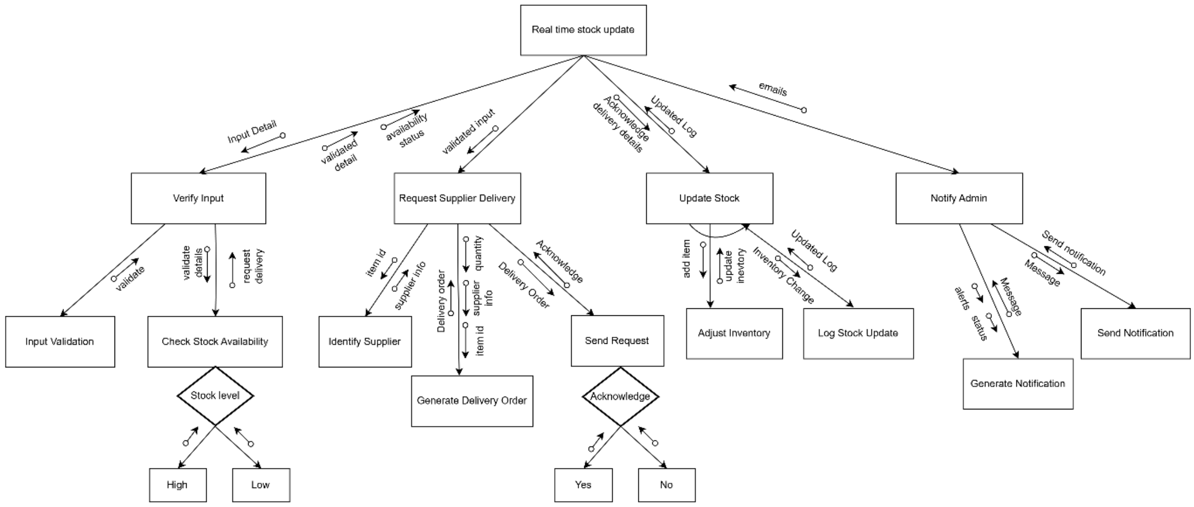

Figure 17: Real time stock update structure chart

Module Specification

Name	**Real Time Stock Updates**
Purpose	The module intends to provide accurate stock level of inventory, also notify admin if stock is low.
Pseudocode	DO INPUT product_id INPUT quantity IF verification is valid THEN DO REQUEST supplier_delivery IF supplier_response is successful THEN UPDATE stock_database

	IF stock level < threshold THEN
	NOTIFY admin(“Stock level of product_id is low”)
	END IF
	ELSE
	DISPLAY “Supplier delivery failed”
	END IF
	END DO
	ELSE
	DISPLAY “Invalid input data for stock request”
	END IF
	END DO

Input Parameter	product_id, quantity
Output Parameter	Updated_stock_level, admin_notification, supplier_response
Global Variable	stock_database
Local Variable	product_id, quantity, supplier_response, threshold
Calls	None
Called by	Admin

Table 6: Real Time Stock Updates Table

5.4 Dispatch Order (Ananta Gurung)

5.4.2 Environment model specification

Context Level Diagram

The diagram illustrates the basic information flow in an order dispatch system. The buyer and their product order information are sent to inventory system and the details are verified, then, the admin receives dispatch details from the system.

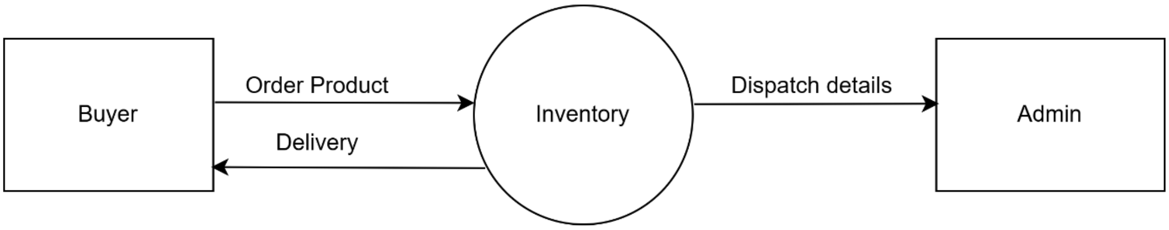

Figure 18: Dispatch Order context level

5.4.2 Internal Model specification

Level-1 DFD

Here, we expanded the single process into its core components, detailing the flow of data between users and the system as the orders are verified and dispatched to the customer.

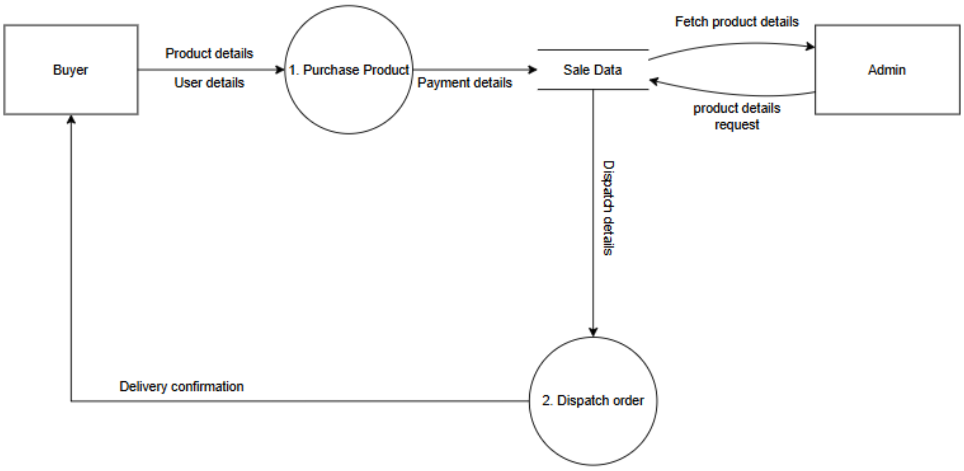

Figure 19: Dispatch order level 1

Level-2 DFD

The diagram shows the order through a delivery system. It starts with the buyer ordering products. The admin then verifies the order and payment. Then, a delivery method is chosen considering the delivery route. Once that is decided, the system dispatches order and tracks the delivery status.

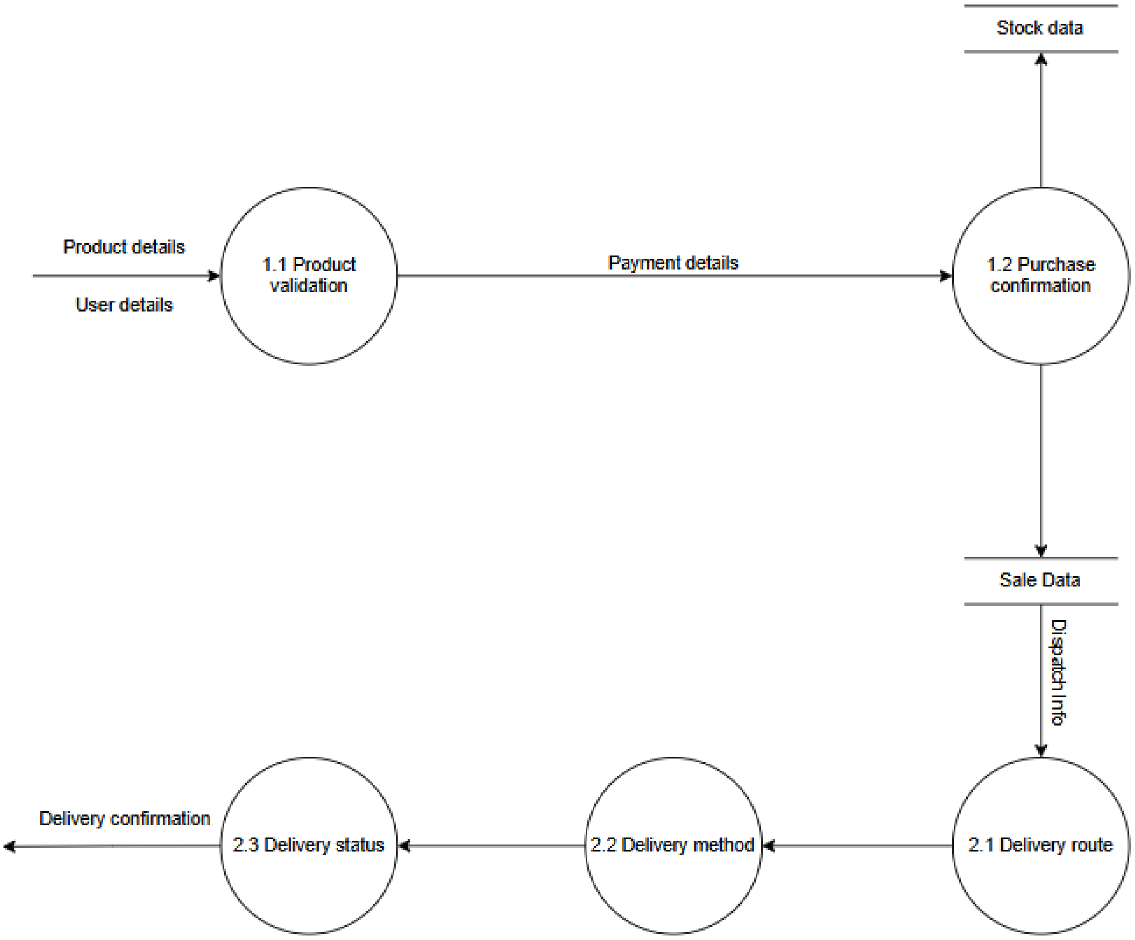

Figure 20: Dispatch order level 2

5.5.3 Design Specification

Structure Chart

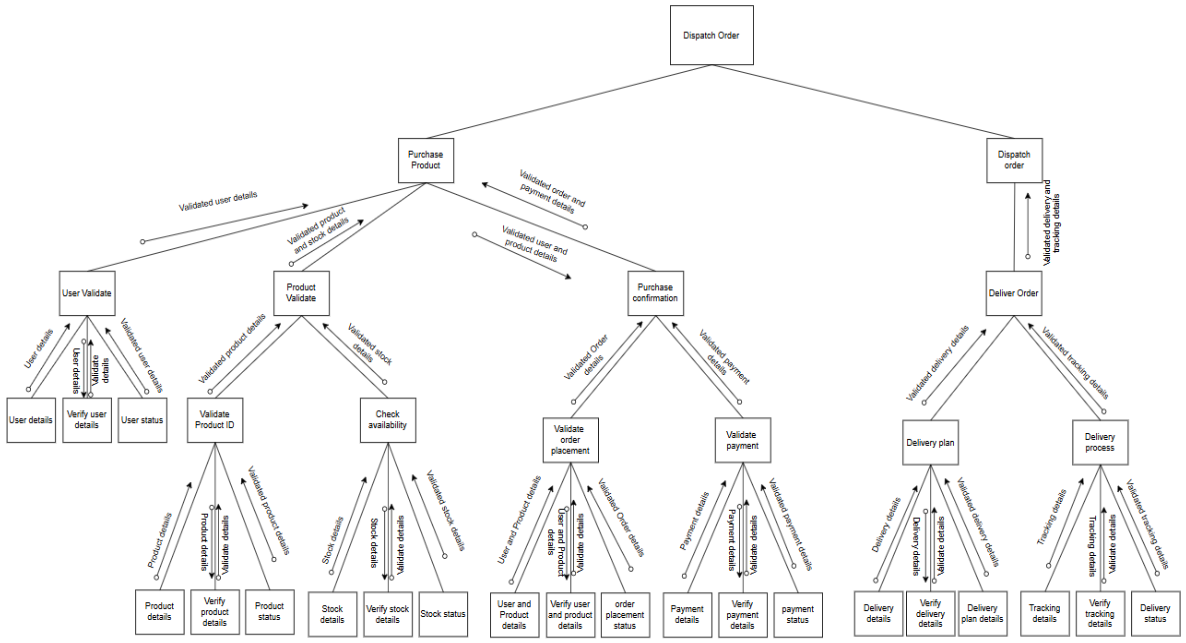

Figure 21: Dispatch order structure chart

Module Specification

Name	**Dispatch order**
Purpose	To deliver an order to buyer’s address after verifying the order and payment details.
Pseudocode	DO INPUT buyer_id INPUT product_id INPUT order_id INPUT payment_id INPUT Address DO IF buyer_id is validated DO IF product_id is validated DO IF order_id is validated DO IF payment_id is validated DO GENERATE dispatch_id CONFIRM dispatch to buyer (Address) DISPLAY “Order has been dispatched. Tracking number: 12ABC” END DO ELSE

	DO DISPLAY “Invalid payment ID” END DO DISPLAY “Invalid order ID” END DO DISPLAY “Invalid product ID” END DO DISPLAY “Invalid buyer ID” END DO END DO
Input parameters	buyer_id, product_id, order_id, payment_id, address
Output parameters	Delivery status
Global variable	none
Local variable	buyer_id, product_id, order_id, payment_id, address, dispatch_id
Call	Delivery status
Called by	Admin

Table 7: Dispatch Order Table

5.5 Payment (Arun Nagarkoti)

Context Level Diagram (Level 0 DFD):

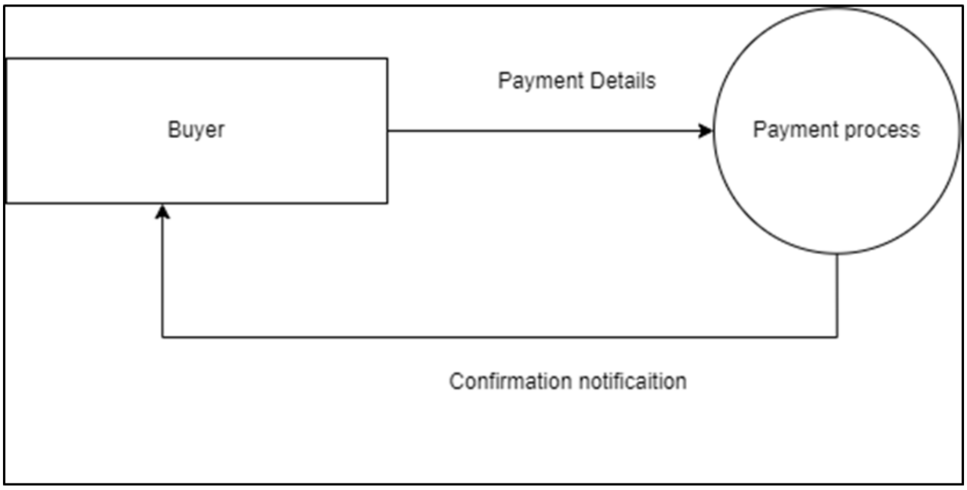

Figure 22: Payment context level

Level 0 DFD Explanation:

The above diagram is a context level data flow diagram i.e. level 0 DFD. The diagram explains how the data flow of “Payment” module works. It shows only the surface/context level information. Here, the buyer/customer provides their payment details ahead, the system then processes the given credentials and runs various verification, and validation checks to process the payment. Following the successful validation of details, the system then responds with confirmation notification to the buyer.

5.5.3 Internal Model Specification

Level 1 Data Flow Diagram:

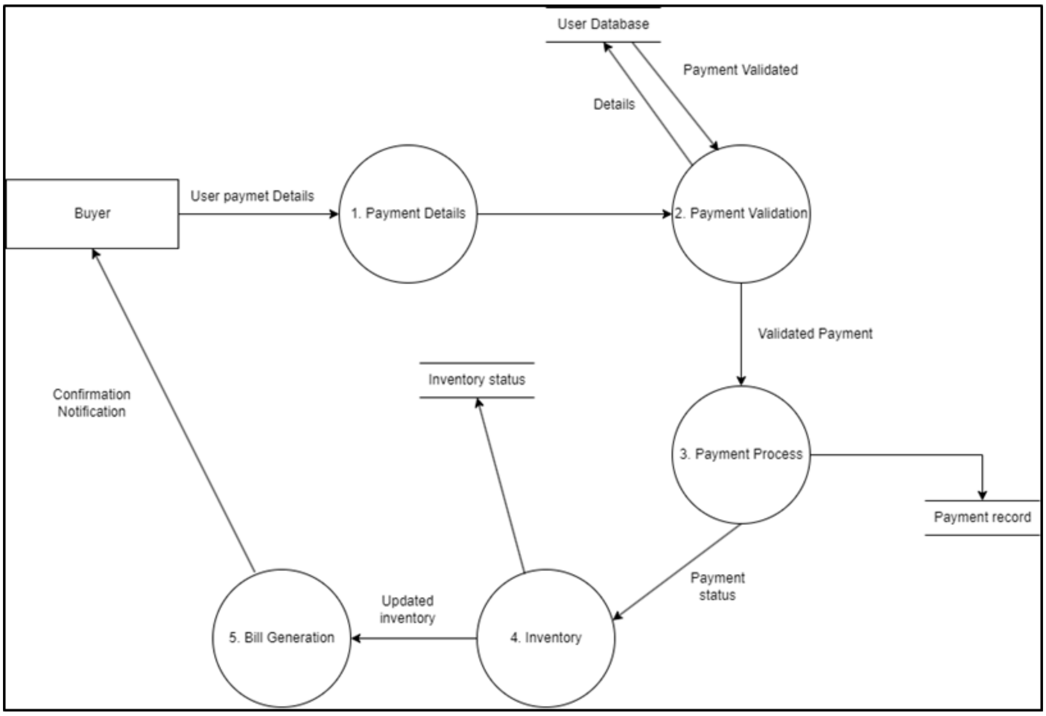

Figure 23: Payment level 1

Level 1 DFD Explanation:

The above diagram is a level 1 data flow diagram of the "Payment" module. In this level, A high-level overview of how user payment details are handled. All the validation, payment processing and the links to inventory as well as to billings as shown in a high-level, a deeper understanding of "Payment" module is provided here.

The first procedure starts with the buyer providing their payment details. All the payment details are then forwarded to "Payment Validation" process for verification of details against the User Database where all the information about all the users is stored. If the given details are validated, the payment moves to the "Payment Process". Here, the "Payment record" database is updated. As the "Inventory" process is based on payment. The inventory database is updated as well. Finally, after all the process, the "Generate Bill" process generate sends a confirmation notification to the buyer.

Level 2 Data Flow Diagram:

1. (Payment details) process:

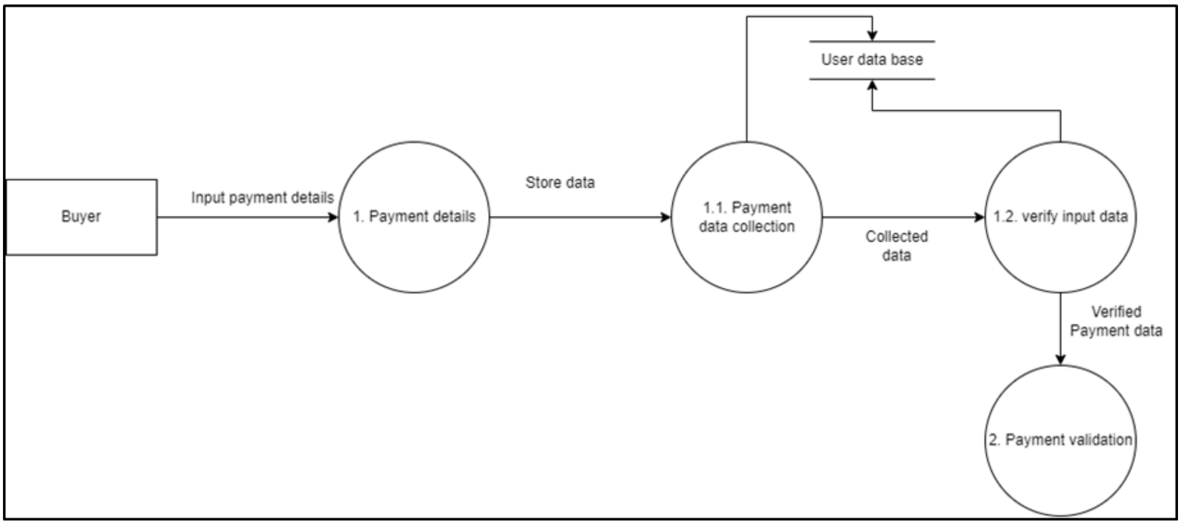

Figure 24: Payment details level 2

2. (Payment validation and Payment process) process:

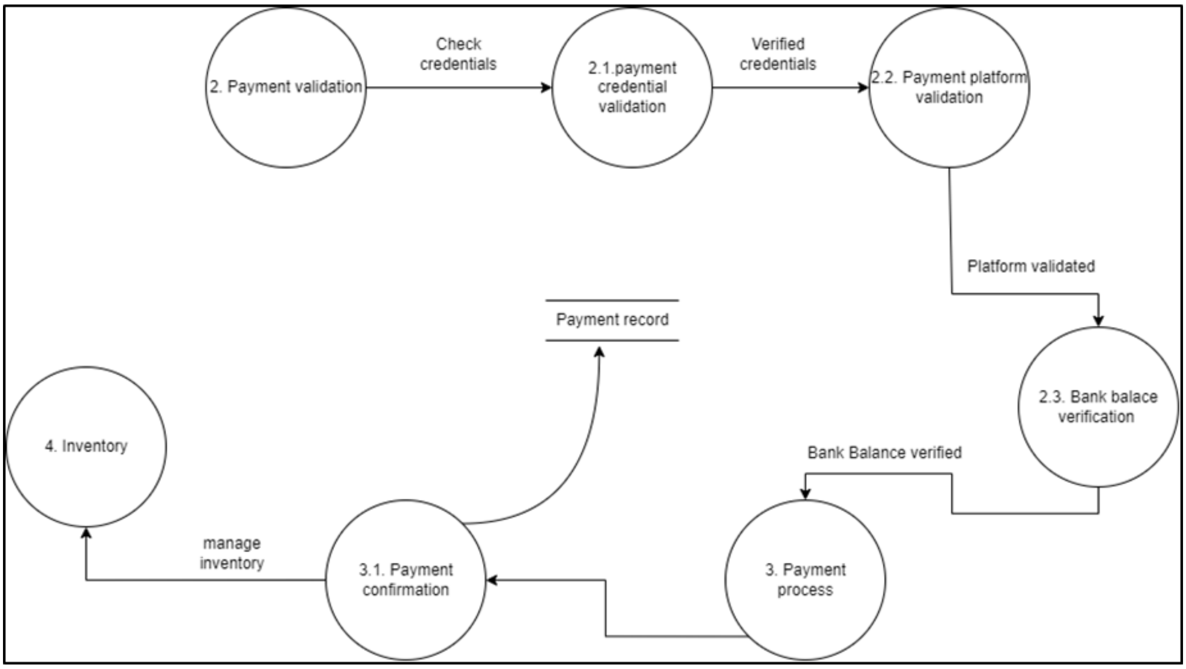

Figure 25: Payment Validation level 2

3. (Inventory and Generate Bill) process:

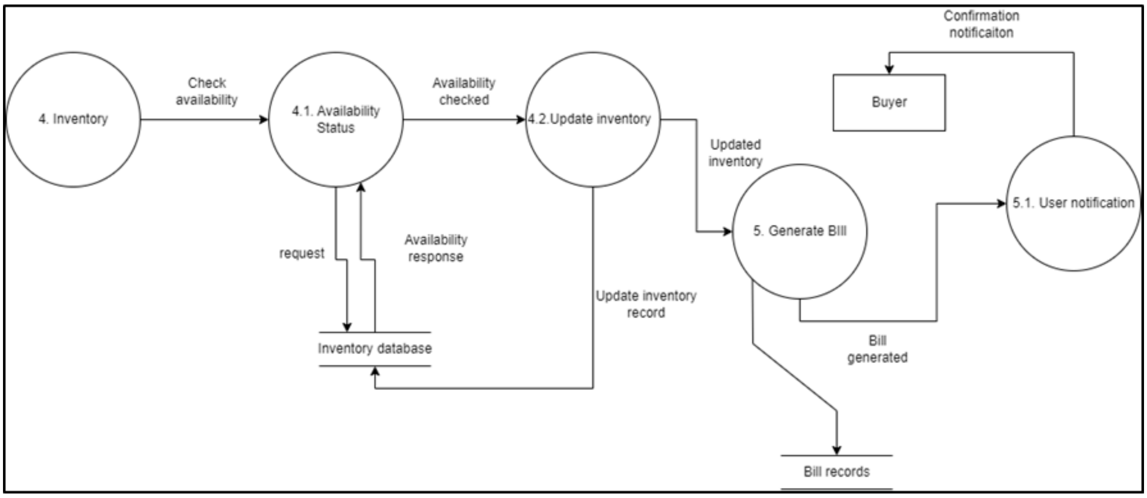

Figure 26: Inventory and generate bill lvl 2

Level 2 DFD Explanation:

The above given three diagrams are the level 2 data flow diagram of “Payment” module. It represents much more deep and detailed processes involved in payment transaction management. Each of the diagram contains different details about different Process. The explanation for each of the diagrams isn given below.

1. Payment details:

In this diagram the initial stages of the payment process are focused on. All the payment credentials provided by the buyer are collected and verified against the “User database”. After the verification of all the details, it is then passed to the “Payment Validation” process for further validations.

2. Payment validation and Payment process:

In this diagram, the inner processes of “Payment Validation” process is represented. All the validations such as payment credentials validation, payment platform validation and bank balance verification are done. After all the validations, the payment process begins which leads to payment confirmation and updates the payment record database. Following all the validations, it is then led to “Inventory” process.

3. Inventory and Generate Bill:

This diagram is about inventory update and bill generation. After the payment confirmation, inventory availability is checked then the records are updated as well. The “Generate Bill” process then generates bill and user notification which is sent to the user/buyer as the confirmation notification.

5.5.3 Design Specification

Structure Chart for “Payment” module:

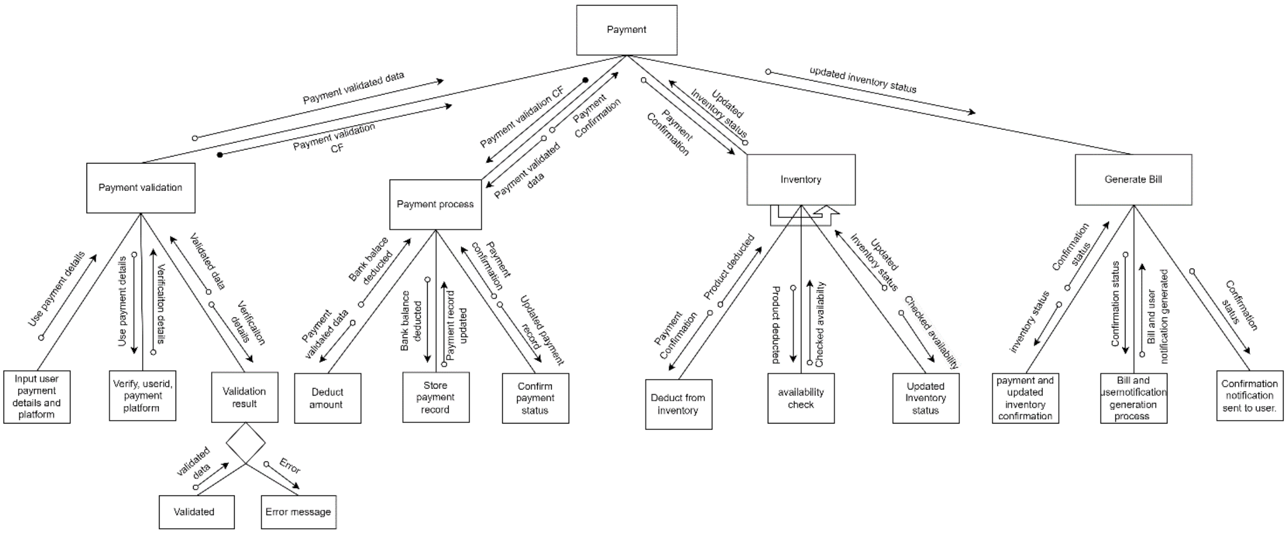

Figure 27: Payment structure chart

Structure Chart Explanation:

The above given diagram is the structured chart of the payment module. This structure chart helps to breakdown the workflow of Payment module into its lowest management level. It clearly shows how the data is flowing from one submodule to another starting from payment validation where all the user details and payment platforms are verified. If not verified, then the loop will iterate again. Once the given credentials are verified then the control flow will go to payment processing where all the payments are recorded and confirmed. After payment confirmation, the availability checking is done, the inventory is updated according to how the products are deducted. Finally, the bill generation process is done. This process generates a bill and sends a confirmation notification which includes payment details. This structure chart gives a clear visual representation of how the workflow is done in the Payment module.

Module specification

Name	**Payment**
Purpose	This module is designed to handle the payment process of buyers. All the process of payment done here starts from validating given details credentials, platform and updating inventory as well as generating bill and sending confirmation notification to the buyer which includes payment details.

Pseudocode	**DO**
	  // Step 1: Input credentials
	    INPUT buyerId
	    INPUT buyerName
	    INPUT paymentPlatform
	 // Step 2: Validate the credentials
	    IF buyerId, buyerName and paymentPlatform are verified in the system.
	  // Step 3: Payment process
	     THEN
	   RECORD paymentStatus in Payment record database.
	 // Step 4: Update inventory
	  SET inventoryUpdate = FALSE
	  WHILE inventoryUpdate = FALSE
	  DO

```
 IF availabilityCheck()
     THEN
         deductFromInventory()
         SET inventoryUpdate= TRUE
   ELSE
         DISPLAY ("Inventory unavailable")
   END IF
END WHILE

 //STEP 5: Update inventory
     UPDATE inventoryStatus

 //Step 6: Generate bill and user notification
            GENERATE bill
            SEND confirmationNotification to buyer
        ELSE
               Display ("Invalid details! Please enter correct details")
           END IF
END DO
```

Input parameters	buyerId, buyerName, paymentDetails and paymentPlatform
Output parameters	paymentStatus, inventoryStatus, confirmationNotification and generatedBill
Global variables	InventoryDatabase, paymentRecordDatabase
Local variable	buyerId, buyerName, paymentPlatform inventoryUpdate and availabilityCheck
Call	deductFromInventory, availabilityCheck, inventoryStatus

Called by	Payment

Table 8: Payment Table

Summary

The aim of this course was to aid and provide a strong and reliable Inventory Management System (IMS) and documented analysis of system requirements for the “Global Tech Corporation”. This IMS was designed and developed after a careful and deep analysis of resource requirements, risks, proper definition of goals and learning from previous mistakes.

This project was able to meet its deadline with the joint effort of all the team members who have come together for this project. The project was divided into smaller portions and every team member was allocated a particular group task as well as an individual task. Documenting business case, risks, omissions and inconsistencies, solutions, functional and non-functional requirements as well as designing different data flow diagrams (DFD) and structure chart for both group and individuals we had done for proper system design. Many different assumptions were made to properly address those conditions. All the individual tasks such as Purchase order, Report preparation, Real time stock, dispatch order and Payment were completed on time with their proper data flow diagram (level 0,1 and 2) DFD, structure chart and their module specification as well. These diagrams were made for proper visual representation of working mechanism for each module and the whole system. All the group tasks were completed and were properly documented on meeting logs as well. The project was divided into 3 milestones, each milestone was met properly with effort of all the team members. During this completion of this project, we were faced with a lot of troubles and walls. Our professors and tutors helped review our work after each milestone and gave us tips which had helped us in the completion of this project.

In conclusion, the completion of this project was only possible due to the joint efforts of our team members and our module teachers. We solved every trouble we faced with our unwavering determination. The designing of Inventory management system concludes here.

Bibliography

Goel, N., 2024. *Library: Structured Chart.* [Online]
Available at: https://www.naukri.com/code360/library/structure-charts-and-hipo-in-system-analysis
[Accessed 9 January 2025].

javatpoint, 2024. *software-engineering-data-flow-diagram.* [Online]
Available at: https://www.javatpoint.com/software-engineering-data-flow-diagrams
[Accessed 15 December 2024].

lucid chart, 2024. *Pages: data-flow-diagram.* [Online]
Available at: https://www.lucidchart.com/pages/data-flow-diagram#:~:text=A%20data%20flow%20diagram%20(DFD,the%20routes%20between%20each%20destination.
[Accessed 15 December 2024].

Rose, M., 2012. *Dictionary/ Business Software/ Project Management.* [Online]
Available at: https://www.techopedia.com/definition/28636/process-specification
[Accessed 21 December 2024].

smartdraw, 2024. *Entity-relationship-diagram.* [Online]
Available at: https://www.smartdraw.com/entity-relationship-diagram/?srsltid=AfmBOopB0SuD-WSdXS_pZQAd0c9Pg0tKC90jmAJhmpZFbtgc-6qS2vD3
[Accessed 16 December 2024].