desktop

application Development

**ProposaLl**

**PREPARED FOR:**

**Mr. Dipendra Silwal**

Freelancer | Web Developer |

Kathmandu, Nepal

(Project Supervisor, BCA)

OCEM

June 17, 2024

Prepard By

**CRF CREW**



**Gaindakot-2, Nawalparasi**

**Nepal**

**Proposal for Canteen Management**

**System (CMS)**

**Table of Contents**

[1.Introduction 1](#_Toc169535989)

[1.1 Overview 1](#_Toc169535990)

[1.2 Purpose of the Proposal 1](#_Toc169535991)

[2. Aim and Scope 1](#_Toc169535992)

[2.1 Aim 1](#_Toc169535993)

[2.2 Scope 2](#_Toc169535994)

[3. Current Problem and Proposed Solution 3](#_Toc169535995)

[3.1 Current Problem 3](#_Toc169535996)

[3.2 Proposed Solution 3](#_Toc169535997)

[4. High-Level Requirements 4](#_Toc169535998)

[4.1 Functional Requirements: 4](#_Toc169535999)

[4.2 Non-Functional Requirements: 5](#_Toc169536000)

[5. Requirement Prioritization Table 6](#_Toc169536001)

[6. Design 7](#_Toc169536002)

[6.1 ER Diagram 7](#_Toc169536003)

[6.2 Use Case Diagram 9](#_Toc169536004)

[6.3 Data Flow Diagram (DFD) 11](#_Toc169536005)

[6.4 UML Class Diagram 16](#_Toc169536006)

[7. Detailed Budget 18](#_Toc169536007)

[8. Detailed Timeline 19](#_Toc169536008)

[9. User Interface 20](#_Toc169536009)

[10. Conclusion 21](#_Toc169536010)

[11. References 22](#_Toc169536011)

# 1.Introduction

**1.1 Overview**

The Canteen Management System (CMS) is a comprehensive desktop application designed to streamline and enhance the operational efficiency of canteen services. Developed using VB.NET and Microsoft SQL Server, the CMS aims to automate and centralize various aspects of canteen management, including item offerings, customer interactions, order processing, inventory control, and billing procedures. This proposal outlines the rationale, objectives, and anticipated benefits of implementing the CMS within a canteen environment.

**1.2 Purpose of the Proposal**

The purpose of this proposal is to present a detailed plan for the development and implementation of the CMS. It highlights the system's features, the technological stack to be used, the benefits it offers, and the steps required to achieve a successful deployment. By providing a clear roadmap, this proposal aims to secure the necessary approvals and resources to commence the project, ensuring that all stakeholders are aligned with the system's goals and capabilities.

# 2. Aim and Scope

## 2.1 Aim

The primary aim of the Canteen Management System (CMS) project is to develop a comprehensive, efficient, and user-friendly desktop application that streamlines and automates the various operations of a canteen. By leveraging modern technologies such as VB.NET and Microsoft SQL Server, the CMS seeks to enhance the accuracy, efficiency, and overall management of canteen activities. This project aims to address the current challenges faced by canteens, reduce manual labor, minimize errors, and ultimately improve the service experience for customers, including students, staff, and other patrons. The CMS aims to achieve the following specific objectives:

1. **Operational Efficiency**: Automate routine tasks to reduce processing times and improve accuracy.
2. **User Experience**: Provide a seamless and intuitive interface that simplifies the management of canteen operations for staff with varying levels of technical expertise.
3. **Data Integrity**: Ensure secure and accurate management of all data related to items, customers, orders, and financial transactions.
4. **Inventory Management**: Implement effective inventory control mechanisms to maintain optimal stock levels and reduce waste.
5. **Reporting and Analytics**: Offer comprehensive reporting tools that enable informed decision-making and strategic planning.

## 2.2 Scope

The scope of the Canteen Management System (CMS) project is to develop a comprehensive desktop application using VB.NET and Microsoft SQL Server, aimed at improving the efficiency and management of canteen operations. The project encompasses various functionalities such as user, item, customer, order, billing, reporting, security, performance, usability, reliability, and database management. Key features include user account management, item and inventory control, customer profile management, order facilitation and customization, billing and receipt generation, robust security measures, responsive user interface design, reliable system performance, and efficient database schema design. The project will follow a structured development approach from requirements gathering to maintenance to ensure successful delivery of a fully functional and user-friendly system tailored to canteen operations.

**Functionalities**:

* User management: Account creation, roles, permissions.
* Item management: Inventory control, item categorization, pricing.
* Customer management: Profiles, order history.
* Order management: Facilitation, customization options.
* Billing: Automated generation of bills and receipts.
* Reporting: Comprehensive reports on sales, inventory, etc.
* Security: Role-based access control, data encryption.
* Performance: Optimized for speed and responsiveness.
* Usability: Intuitive user interface design.
* Reliability: Stable operation under normal conditions.
* Database management: Efficient schema design, data integrity.

# 3. Current Problem and Proposed Solution

## ****3.1 Current Problem****

Traditional methods of managing canteen operations are fraught with inefficiencies and limitations, posing significant challenges to both canteen staff and customers. Manual processes for inventory management, order processing, and billing often result in errors, delays, and inconsistencies. Moreover, the lack of centralized systems hampers data accuracy and inhibits the ability to derive actionable insights for decision-making. These challenges not only impede operational efficiency but also diminish the overall customer experience, leading to dissatisfaction and lost opportunities for revenue generation.

## ****3.2 Proposed Solution****

To address these challenges and usher in a new era of efficiency and excellence in canteen management, we propose the development and implementation of a comprehensive Canteen Management System (CMS). This innovative solution, developed using VB.NET and Microsoft SQL Server, aims to revolutionize the way canteen operations are conducted by offering a holistic platform that integrates key functionalities into a seamless and user-friendly interface.

The proposed CMS will empower canteen staff with intuitive tools and automation capabilities to streamline processes and enhance productivity. From user management and item tracking to order processing and reporting, the system will cover all aspects of canteen operations, ensuring accuracy, reliability, and customer satisfaction.

Key features of the proposed CMS include:

1. **Efficient Inventory Management:** The system will enable real-time tracking of inventory levels, automatic low-stock alerts, and seamless integration with ordering processes to prevent stockouts and optimize inventory replenishment.
2. **Streamlined Order Processing:** Customers will benefit from a user-friendly interface for placing orders, customizing preferences, and receiving prompt confirmation, thereby enhancing their overall experience and satisfaction.
3. **Automated Billing:** The CMS will automate billing processes, offering diverse payment options and generating receipts for transactions, ensuring accuracy and transparency in financial transactions.
4. **Comprehensive Reporting:** Administrators will have access to comprehensive reports on sales, inventory levels, and other key metrics, empowering data-driven decision-making and strategic planning.

By embracing innovation and leveraging the power of technology, the proposed CMS promises to revolutionize canteen management, driving efficiency, accuracy, and customer satisfaction to new heights. We believe that this transformative solution will not only address the current challenges faced by canteens but also pave the way for future growth and success in the ever-evolving food service industry.

# 4. High-Level Requirements

## 4.1 Functional Requirements:

1. **User Management:**
   * Ability to create, edit, and delete user accounts for canteen staff.
   * Option to assign different access levels to user accounts (e.g., cashier, inventory manager).
2. **Item Management:**
   * Capability to add, edit, and delete items offered in the canteen, including name, description, price, and category.
   * Functionality to maintain stock levels for each item and set up low-stock alerts.
3. **Customer Management:**
   * Optional feature to create and manage customer profiles.
4. **Order Management:**
   * Capability for customers to place orders, either in person
   * Option to customize orders (e.g., adding special instructions).
   * Display of order details including item names, quantities, and total price.
   * Ability to modify orders before confirmation.
5. **Billing:**
   * Automated calculation of order totals.
   * Provision of various payment options (e.g., cash, card).
   * Generation of receipts for orders.
6. **Reporting:**
   * Generation of reports on sales, inventory levels, and other relevant data.
   * Ability to filter and export reports in different formats (e.g., PDF, CSV).

## 4.2 Non-Functional Requirements:

1. **Security:**
   * Implementation of robust user authentication and authorization mechanisms.
   * Secure storage of sensitive data such as customer information and financial data.
2. **Performance:**
   * Responsive system handling user requests efficiently.
   * Minimal downtime to ensure continuous operation.
3. **Usability:**
   * Intuitive user interface that is easy to navigate for users with varying technical skills.
   * Accessibility features to accommodate users with disabilities.
4. **Reliability:**
   * Reliable system operation to minimize disruptions to canteen operations.
   * Data integrity measures to prevent loss or corruption of critical information.
5. **Maintainability:**
   * Well-documented codebase to facilitate ease of maintenance and future enhancements.
   * Modular design to enable scalability and flexibility in accommodating changes or updates.

By adhering to these high-level requirements, the proposed Canteen Management System (CMS) will strive to meet the functional needs of canteen operations while also addressing critical non-functional aspects such as security, performance, usability, reliability, and maintainability. These requirements form the foundation for the development of a robust and user-centric solution that aims to enhance efficiency, accuracy, and customer satisfaction in canteen management

# 5. Requirement Prioritization Table

Requirement prioritization is a crucial aspect of project management, particularly in software development, where resources and time are limited. It involves the systematic process of ranking requirements based on their importance and impact on the project's goals and objectives.

|  |  |  |
| --- | --- | --- |
| Priority | Requirement | Description |
| High | User Management | Essential for security and access control. |
| High | Item Management | Crucial for managing inventory and ensuring stock availability. |
| High | Order Management | Vital for processing customer orders efficiently and accurately. |
| High | Billing | Necessary for handling financial transactions and generating receipts. |
| High | Security | Critical for protecting sensitive data and ensuring compliance with privacy regulations. |
| Medium | Reporting | Important for tracking performance metrics and making data-driven decisions. |
| Medium | Performance | Significant for providing a responsive and efficient user experience. |
| Medium | Usability | Important for ensuring ease of use and accessibility for users with varying technical skills. |
| Medium | Reliability | Important for minimizing downtime and ensuring uninterrupted canteen operations. |
| Low | Maintainability | Important for facilitating ease of maintenance and future enhancements. |
| Low | Customer Management | Optional feature for managing customer profiles and pre-registration. |

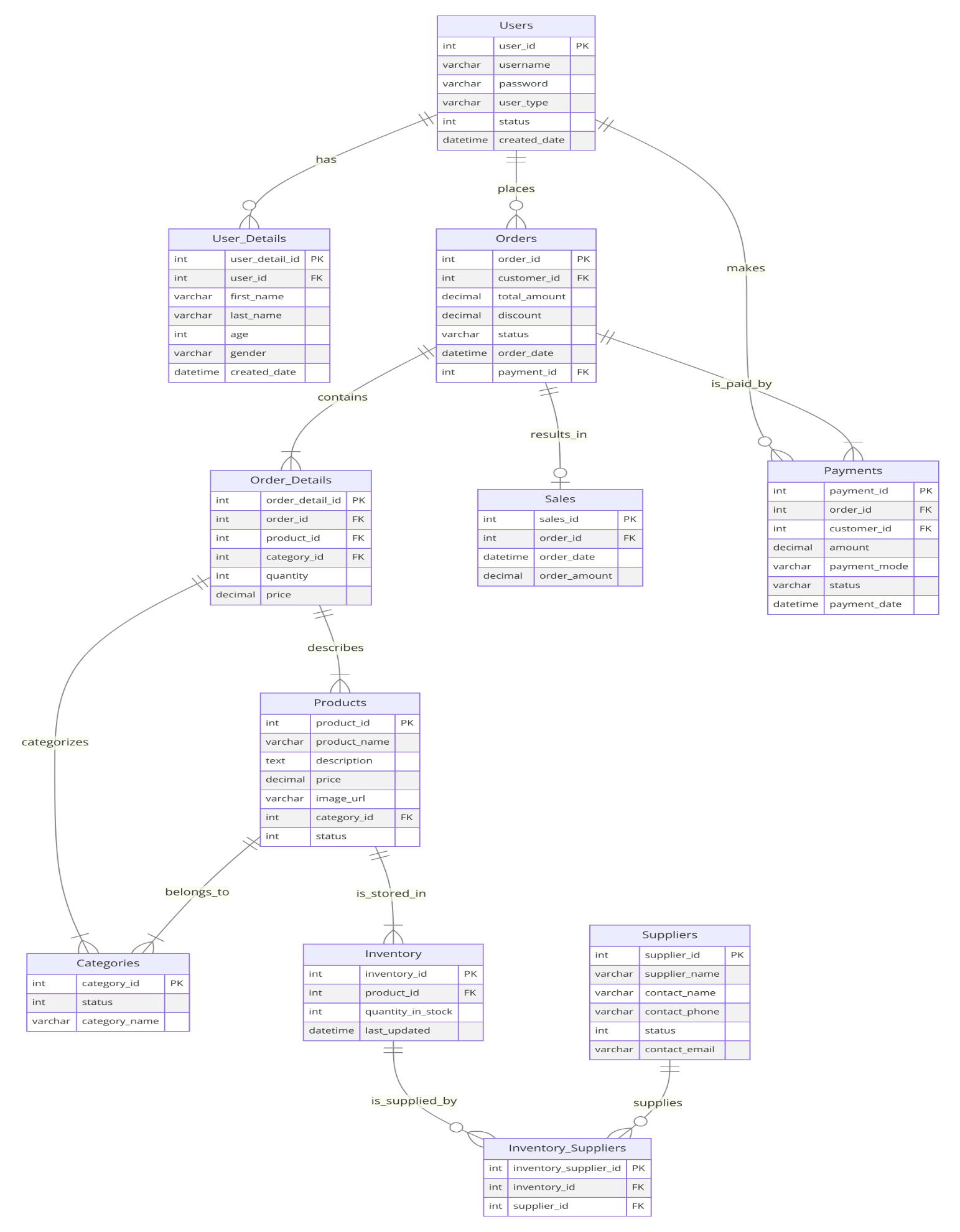
# 6. Design

## 6.1 ER Diagram

An Entity-Relationship Diagram (ERD) is a visual tool used to represent the structure of a database by outlining its entities, attributes, and the relationships between them. Entities, depicted as rectangles, represent objects or concepts, while attributes, shown as ovals, describe these entities. Relationships, illustrated by diamonds or lines, define how entities interact, with cardinality indicating the numerical relationship between them. ERDs serve as blueprints for database design, clarifying system requirements, facilitating database creation, improving communication among stakeholders, and identifying data redundancies. In proposals, ERDs effectively illustrate data organization, enhancing the understanding and alignment of system architecture with project goals.

**Components of an ER Diagram**

1. **Entities**:
   * Represent objects or concepts with distinct existence.
   * Depicted as rectangles.
   * Example: *Student*, *Course*.
2. **Attributes**:
   * Describe properties or characteristics of entities.
   * Depicted as ovals.
   * Example: *StudentID*, *Name*.
3. **Relationships**:
   * Illustrate associations between entities.
   * Depicted as diamonds or lines connecting entities.
   * Example: *Enrollment* (between *Student* and *Course*).
4. **Cardinality**:
   * Indicates the numerical relationship between entities.
   * Example: One-to-One (1:1), One-to-Many (1), Many-to-Many (M).

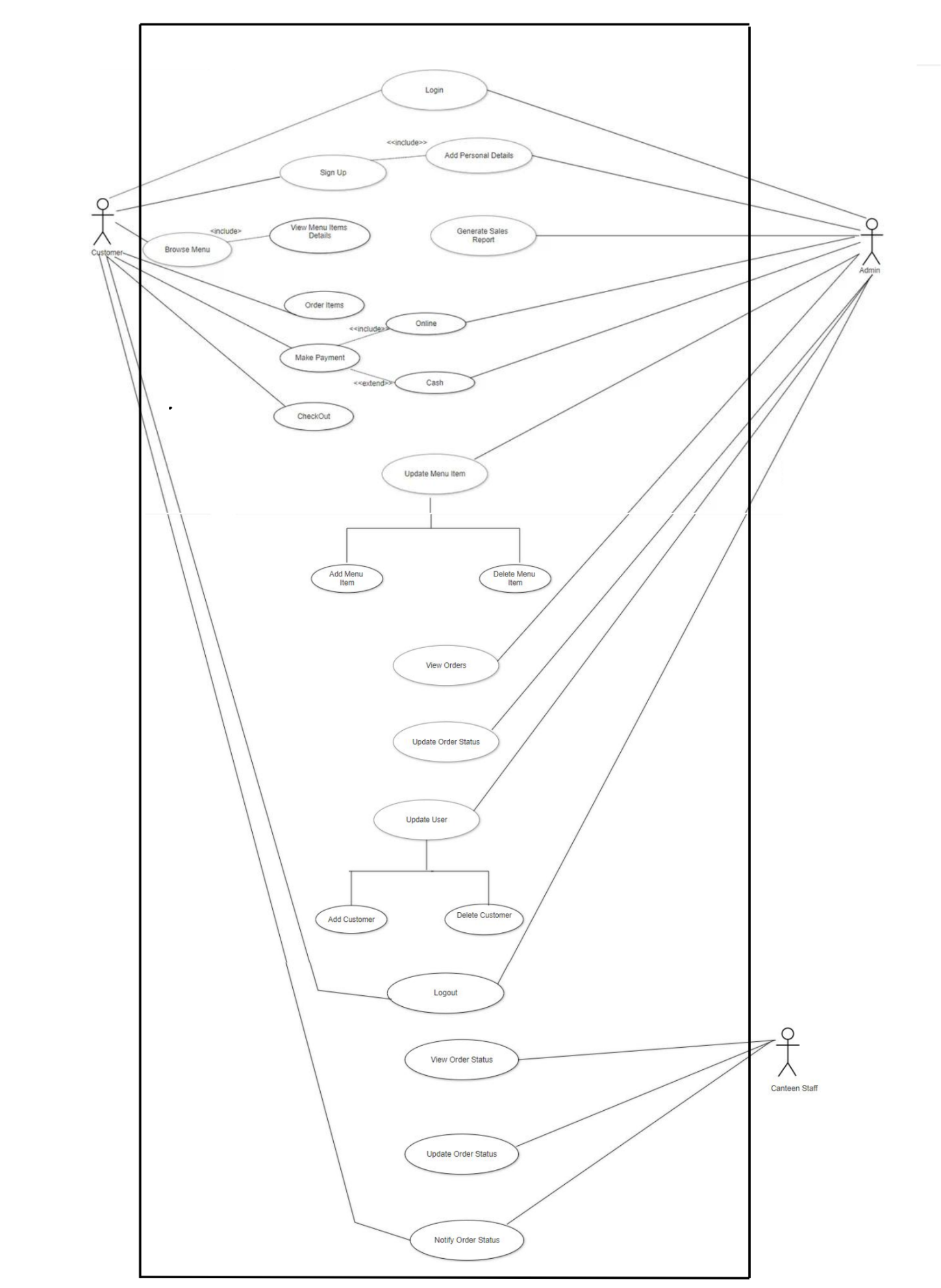


## 6.2 Use Case Diagram

A Use Case Diagram is a visual representation that illustrates how users interact with a system and its functionalities. It depicts the various use cases, actors (users or external systems), and their relationships within the system. Use cases represent specific interactions between users (actors) and the system to achieve particular goals. Actors are individuals, external systems, or other entities that interact with the system, depicted as stick figures or blocks outside the system boundary. Relationships between actors and use cases show which actors are involved in each use case, clarifying system behavior and user roles. Use Case Diagrams are invaluable tools in system analysis and design, aiding in requirements gathering, communication among stakeholders, and validation of system functionality.

**Components of a Use Case Diagram:**

1. **Actors**:
   * Represent users, external systems, or other entities interacting with the system.
   * Shown outside the system boundary as stick figures or blocks.
   * Example: *Customer*, *Administrator*.
2. **Use Cases**:
   * Represent specific functionalities or actions the system performs to achieve a goal for an actor.
   * Shown inside the system boundary as ovals.
   * Example: *Place Order*, *Generate Report*.
3. **Relationships**:
   * Connect actors with use cases to indicate which actors are involved in each use case.
   * Arrowed lines show the direction of interaction.
   * Example: *Customer* interacts with *Place Order*.
4. **System Boundary**:
   * Represents the scope and boundary of the system under consideration.
   * Use cases and actors are placed inside this boundary.
5. **Include and Extend Relationships** (optional):
   * **Include**: Indicates that one use case includes the functionality of another.
   * **Extend**: Shows optional or exceptional behavior that extends a base use case.



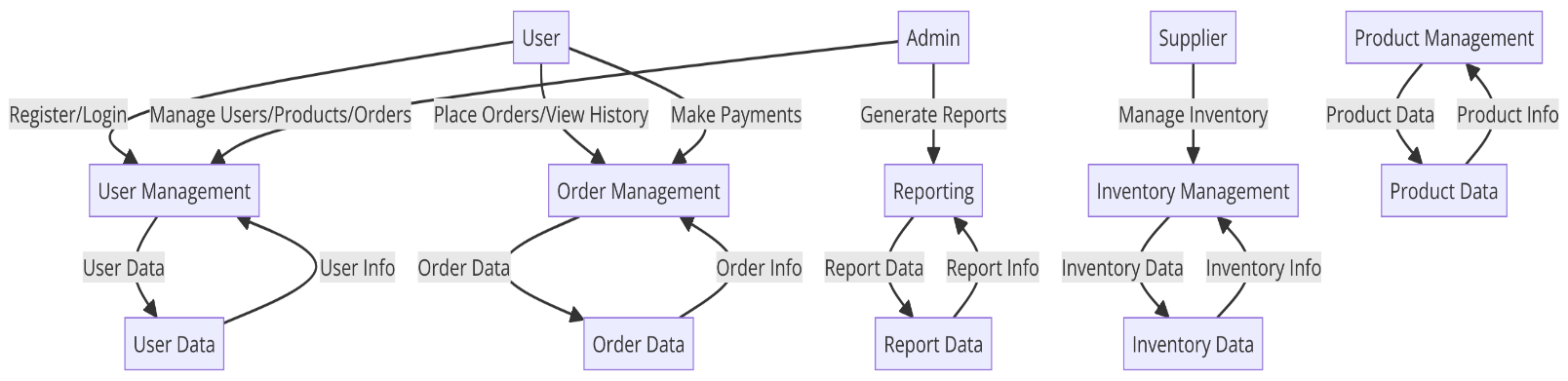
## 6.3 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a visual representation of how data flows through a system and how it is processed. It illustrates the movement of data between processes, data stores, and external entities. Processes represent functions or transformations that occur within the system, manipulating data inputs to produce outputs. Data stores depict where data is persisted within the system. External entities are sources or destinations of data outside the system boundary. Arrows indicate the direction of data flow, showing how data moves through the system and undergoes transformation. DFDs are instrumental in system analysis and design, facilitating a clear understanding of data flow and enabling stakeholders to identify inefficiencies, redundancies, and opportunities for improvement within the system architecture.

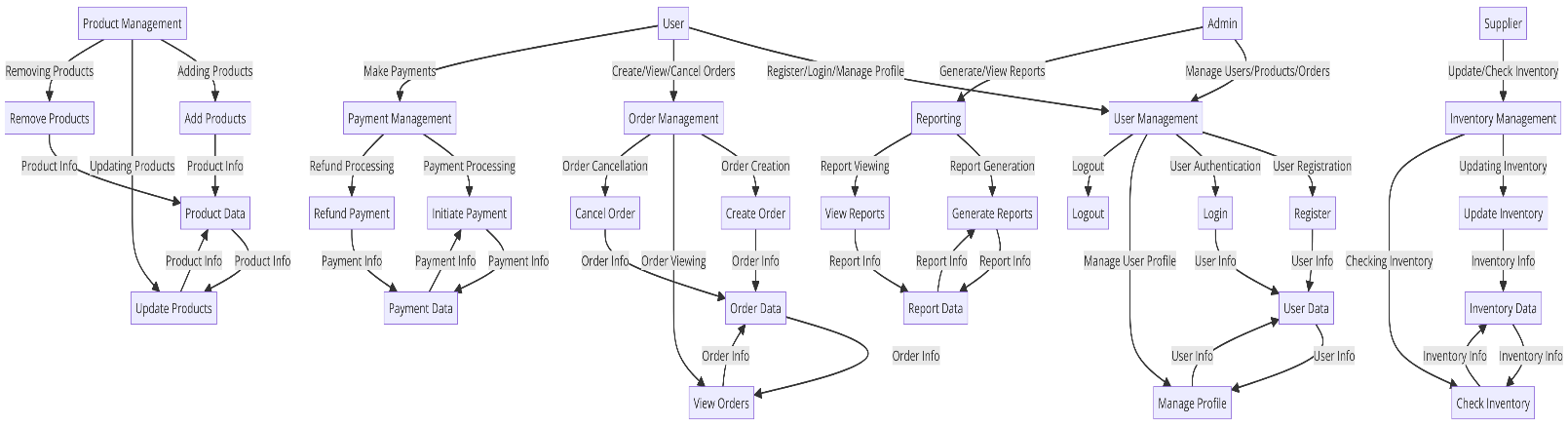
**Components of a Data Flow Diagram:**

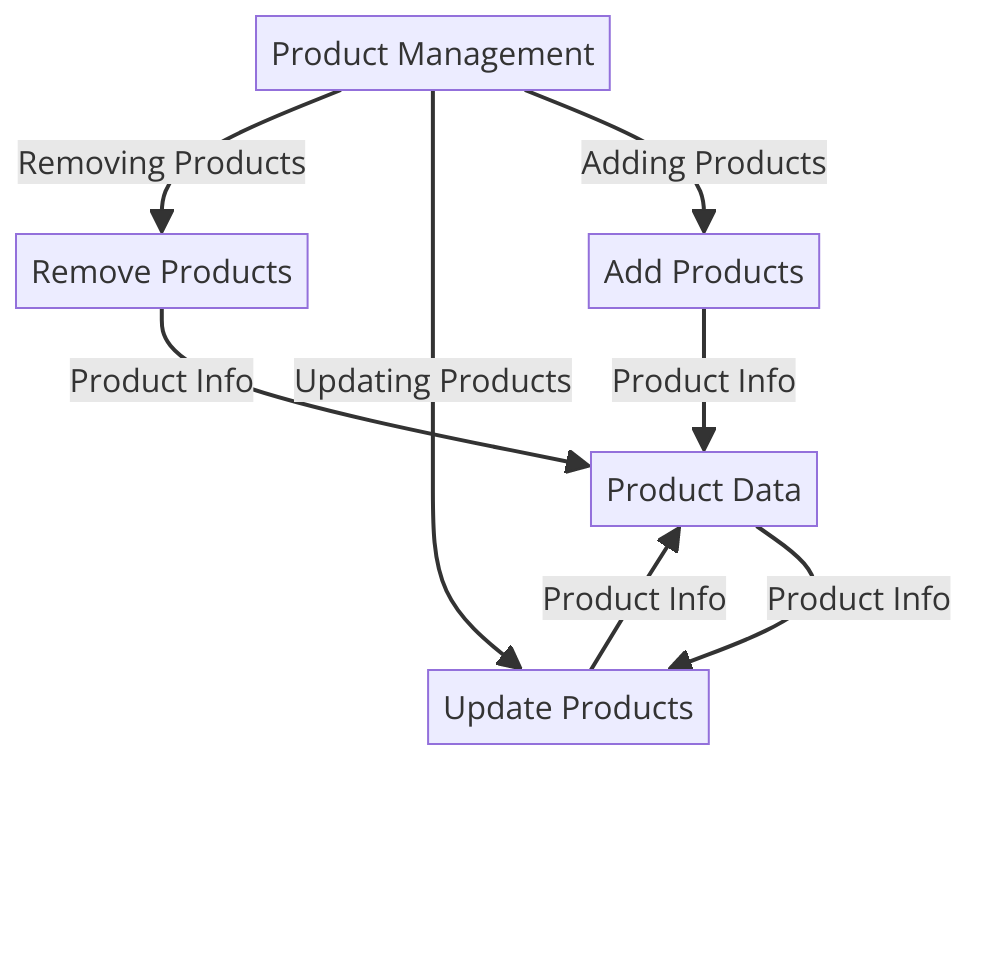
1. **Processes**:
   * Represent activities or transformations that manipulate data inputs to produce outputs.
   * Shown as circles or ovals with meaningful labels.
   * Example: *Process Order*, *Calculate Payroll*.
2. **Data Flows**:
   * Represent the movement of data between processes, data stores, and external entities.
   * Arrows indicate the direction of data flow.
   * Example: Data flowing from *Customer* (external entity) to *Place Order* (process).
3. **Data Stores**:
   * Represent where data is stored or persisted within the system.
   * Shown as rectangles.
   * Example: *Customer Database*, *Inventory Database*.
4. **External Entities**:
   * Represent sources or destinations of data outside the system boundary.
   * Shown as squares or rectangles.
   * Example: *Customer*, *Supplier*.

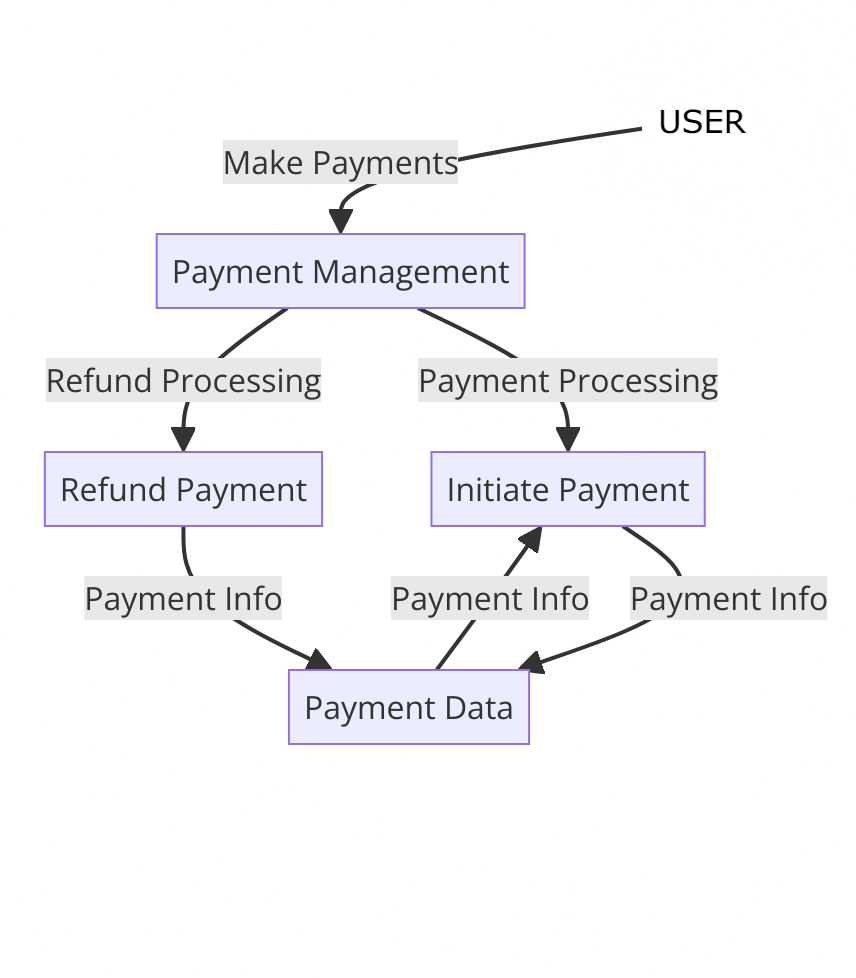
**Level 0 DFD**

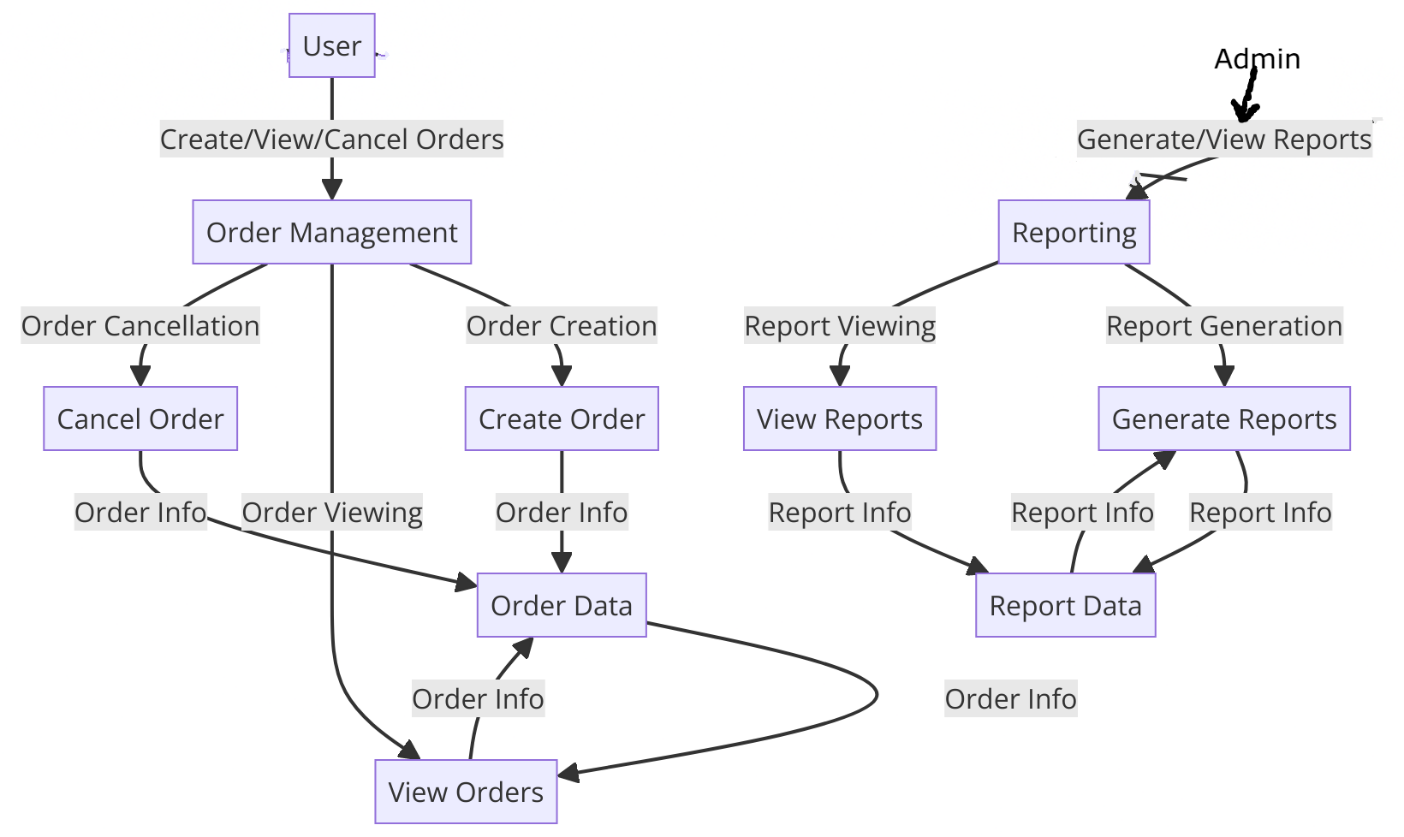


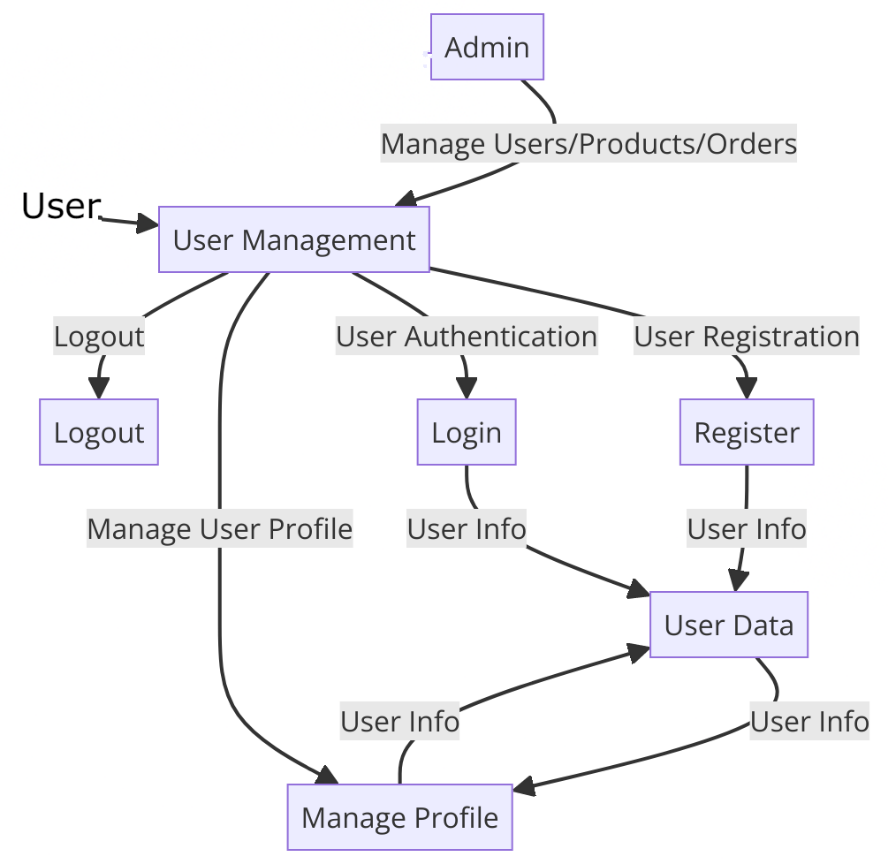
**Level 1 DFD**

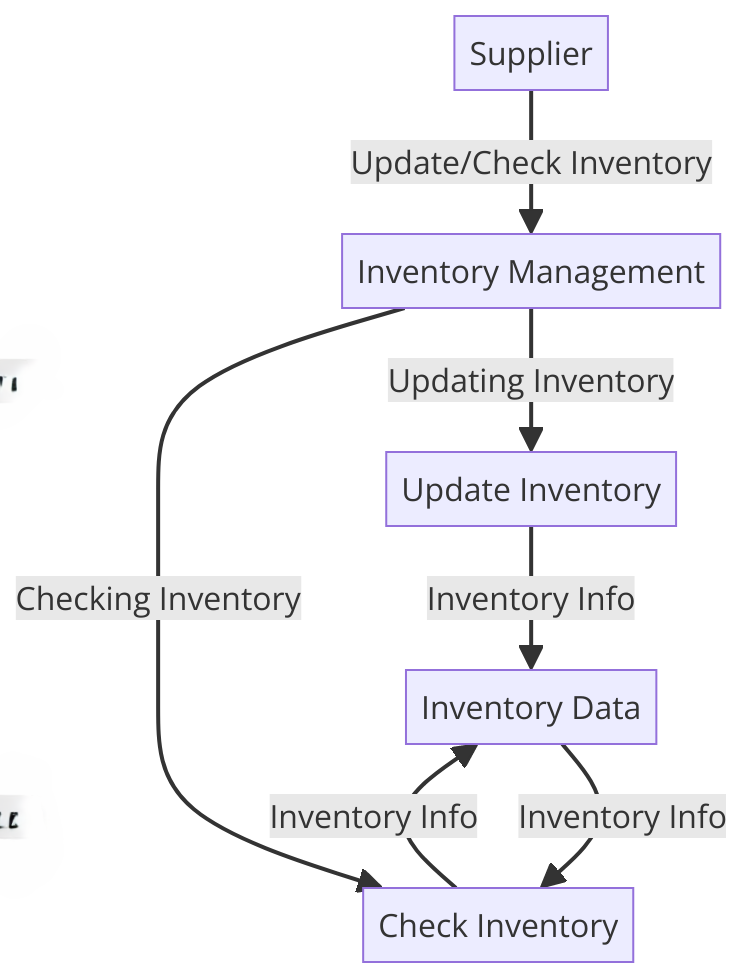










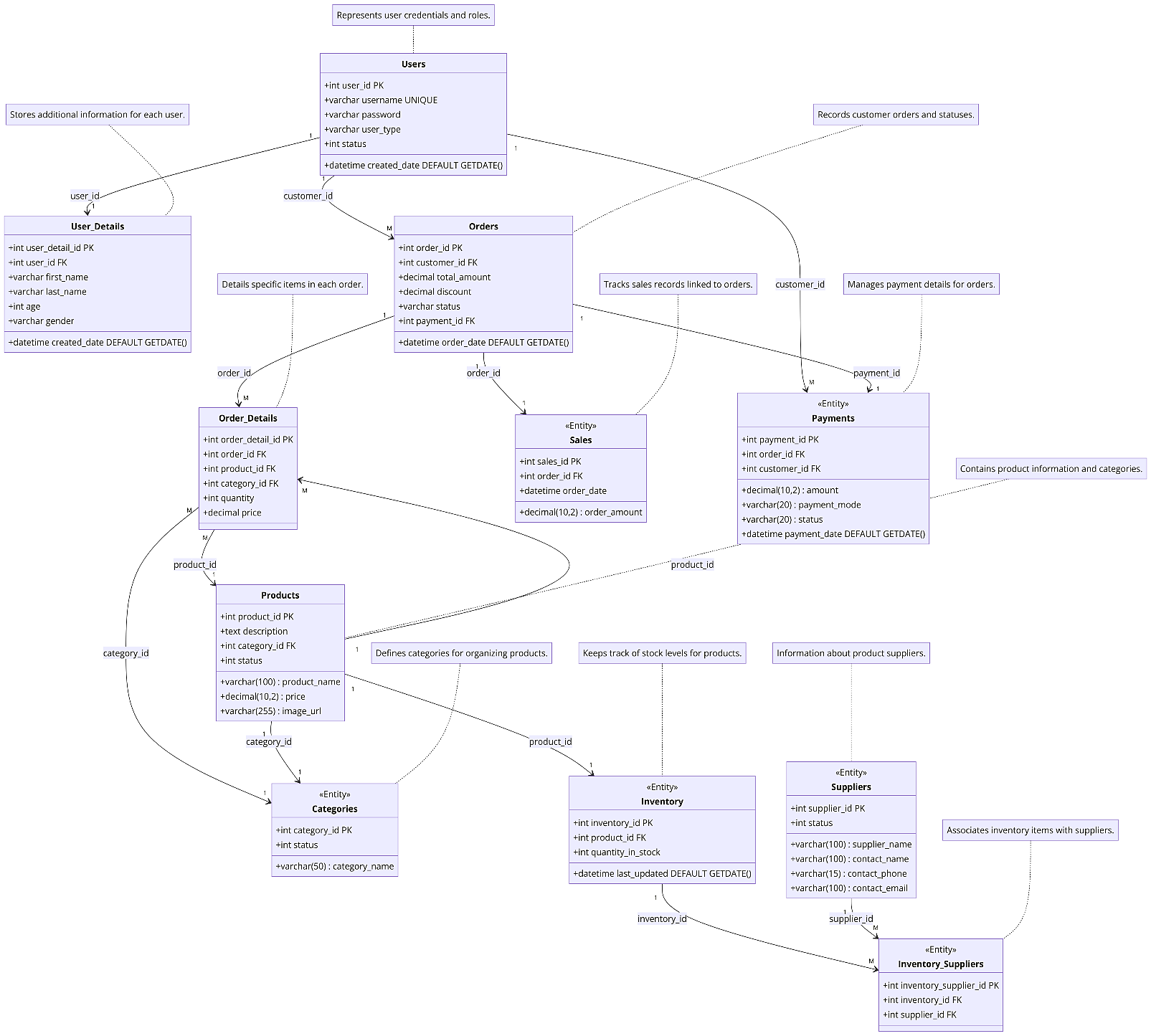


## 6.4 UML Class Diagram

A UML Class Diagram is a type of static structure diagram in the Unified Modeling Language (UML) that represents the structure and relationships of classes and interfaces within a system. It provides a blueprint of the system's classes, their attributes, methods, and the relationships among objects. Classes represent blueprints for objects, encapsulating data (attributes) and behaviors (methods) related to a specific entity or concept. Relationships between classes illustrate how objects collaborate and interact in the system. UML Class Diagrams are instrumental in system design and analysis, aiding in understanding class structure, identifying class responsibilities, and facilitating communication among stakeholders.

**Components of a UML Class Diagram:**

1. **Classes**:
   * Represent templates for objects that share common structure, behavior, and relationships.
   * Shown as rectangles with three compartments: class name, attributes, and methods.
   * Example: *Customer*, *Order*, *Product*.
2. **Attributes**:
   * Represent data or properties of a class.
   * Shown in the second compartment of the class rectangle.
   * Example: *CustomerID*, *Name*, *Address*.
3. **Methods**:
   * Represent behaviors or operations that a class can perform.
   * Shown in the third compartment of the class rectangle.
   * Example: *placeOrder()*, *calculateTotal()*.
4. **Relationships**:
   * Types of relationships include:
     + **Association**: Represents a relationship where objects of one class are connected to objects of another class.
     + **Aggregation**: Represents a "whole-part" relationship where one class (whole) contains objects of another class (part), but parts can exist independently.
     + **Composition**: Represents a stronger form of aggregation where the parts cannot exist without the whole.
     + **Inheritance**: Represents an "is-a" relationship where one class (subclass or derived class) inherits attributes and methods from another class (superclass or base class).
     + **Dependency**: Represents a relationship where one class relies on another class.
5. **Multiplicities**:
   * Indicate how many instances of one class are related to instances of another class in associations.
   * Shown near association lines (e.g., 1, \*, 0..1).



# 7. Detailed Budget

A budget is a financial plan that outlines expected income and expenses over a specific period, typically aligned with the duration of a project or fiscal year. It serves as a crucial tool for financial management and decision-making, ensuring resources are allocated efficiently to achieve organizational goals. Budgets typically include both revenue (income) and expenditures (costs), providing a comprehensive overview of financial activities

A well-structured budget not only helps in financial planning and management but also serves as a tool for tracking progress, evaluating performance, and making informed decisions to achieve financial goals effectively.

|  |  |
| --- | --- |
| Development Phase | Estimated Cost (NRS) |
| Analysis and Planning |  |
| Project Manager (1 month) | 35000 |
| Miscellaneous Expenses | 2500 |
| Contingency (10% of Total) | 3850 |
| Total for Analysis and Planning | **41,350** |
| Development |  |
| Software Developers (3 months) | 90,000 |
| Database Administrators (3 months) | 50,000 |
| Miscellaneous Expenses | 2,500 |
| Contingency (10% of Total) | 14,250 |
| Total for Development | **1,56,750** |
| Testing and Quality Assurance |  |
| Quality Assurance/Testers (2 months) | 40,000 |
| Miscellaneous Expenses | 2,500 |
| Contingency (10% of Total) | 4,250 |
| Total for Testing and QA | **46,750** |
| Deployment and Maintenance |  |
| Miscellaneous Expenses | 25,000 |
| Contingency (10% of Total) | 2500 |
| Total for Deployment and Maintenance | **27,500** |
| Grand Total Estimated Budget | **2,31,000** |

# 8. Detailed Timeline

A timeline is a visual representation of the sequence of events or activities within a project, outlining key milestones and deadlines. It serves as a roadmap for project management, helping to organize tasks, allocate resources, and track progress over time. A well-defined timeline ensures that project objectives are met within specified timeframes, facilitating efficient project execution and coordination among team members and stakeholders.

|  |  |
| --- | --- |
| Phase | Duration |
| Requirement Analysis | 2 weeks |
| Design | 3 weeks |
| Development | 8 weeks |
| Testing | 4 weeks |
| Deployment | 2 weeks |
| Documentation | 2 weeks |
| Total | **21 weeks** |

# 9. User Interface

1. **Customer Interface:**
   * Browse menu and place orders
   * View order history and status
   * Update profile information
2. **Admin Interface:**
   * Manage user accounts and permissions
   * Add, edit, or remove menu items
   * View and process orders
   * Generate reports on sales and inventory
   * Configure system settings
3. **Registration Interface:**
   * New users can register for an account by providing required information such as name, email address, and password.
   * Features:
     1. Input fields for user registration information.
     2. Validation checks for username format and password strength.
     3. Button to submit registration details.
4. **Login Interface:**
   * Registered users can log in to their accounts to access personalized features and functionalities.
   * Features:
     1. Input fields for email address and password.
     2. Option for users to reset forgotten passwords.
     3. Secure authentication process.
     4. Button to submit login credentials.

# 10. Conclusion

In conclusion, the Canteen Management System (CMS) offers a comprehensive solution for streamlining and enhancing operations within canteens. With distinct interfaces tailored for customers, administrators, and staff, the system aims to optimize efficiency, improve accuracy, and elevate the overall dining experience. The customer interface empowers users to effortlessly browse menu offerings, place orders, and provide valuable feedback. By facilitating seamless transactions and personalized interactions, it fosters customer satisfaction and loyalty.

Administrators benefit from a robust interface equipped with powerful tools for managing user accounts, menu items, orders, and system settings. With access to insightful reports on sales and inventory, administrators can make informed decisions to drive business growth and success. Staff members utilize a dedicated interface designed to streamline order processing, inventory management, and communication. By centralizing key tasks and providing real-time updates, the staff interface enhances operational efficiency and enables superior customer service.

Furthermore, the inclusion of registration, login, and sign-up interfaces ensures a secure and user-friendly experience for accessing the system. These functionalities empower users to create and manage their accounts with ease, fostering a sense of ownership and engagement. In essence, the Canteen Management System represents a holistic approach to modernizing canteen operations, embracing technology to deliver convenience, reliability, and innovation. By leveraging these interfaces, canteens can optimize their processes, delight customers, and thrive in today's dynamic food service industry.

# 11. References

1. Reddy, GS 2023, “Canteen Management System,” *IJERT*, https://doi.org/10.17577/IJERTV12IS040301.
2. *Project report on Canteen Management System | Academic Projects*, https://www.freeprojectz.com/project-report/6811.
3. *Edit diagram*, https://diagrams.helpful.dev/