**CHAPTER III: SYSTEM ANALYSIS AND DESIGN**

**3.1 Introduction**

The systems analysis is a process which involves collecting factual data, understanding the processes involved, identifying problems, and recommending feasible suggestions for improving the system’s functioning. A new system must be designed based on the user requirements and the detailed analysis of the existing system. This phase is the phase of system designing, which is the most crucial phase in the system development. The logical system design arrived at as a result of systems analysis is converted into physical system design. The analysis of the existing system, the analysis of the proposed system, and the design and development processes will all be covered in this chapter.

**3.2 Analysis of the Current System**

**3.2.1 Problem of the Current System**

The current system at Sauvis Rwanda Technical and Distribution Solution Ltd relies heavily on traditional methods for managing shoe inventory and sales. This manual approach has led to several issues, including:

1. **Inefficiency in Inventory Management**: The manual tracking of stock levels leads to inaccuracies and delays in inventory updates.
2. **Customer Dissatisfaction**: Customers often face issues with product availability and cannot easily browse the store’s inventory.
3. **Operational Delays**: The lack of an integrated system causes delays in processing orders, managing stock, and generating sales reports.
4. **Data Management Challenges**: Manual record-keeping is prone to errors and makes it difficult to retrieve and analyze sales and inventory data efficiently.

The need for a digital solution is evident to streamline operations, improve customer satisfaction, and enhance data management.

**3.3 Analysis of the New System**

**3.3.1 Introduction**

The proposed new system aims to address the issues identified in the current system by implementing an online shoe storing management system. This system will facilitate efficient inventory management, improve customer experience, and provide real-time data analytics for better decision-making.

**3.3.2 System Requirements**

**Functional Requirements**

1. **User Registration and Authentication**: The system should allow users to register, login, and manage their profiles.
2. **Product Catalog Management**: Administrators should be able to add, update, and delete products in the inventory.
3. **Real-Time Inventory Tracking**: The system should update stock levels in real-time as sales are made.
4. **Order Management**: Customers should be able to place orders online, and administrators should manage order processing and fulfillment.
5. **Payment Processing**: Integration with payment gateways to facilitate online transactions.
6. **Reporting and Analytics**: Generate reports on sales, inventory levels, and customer behavior.

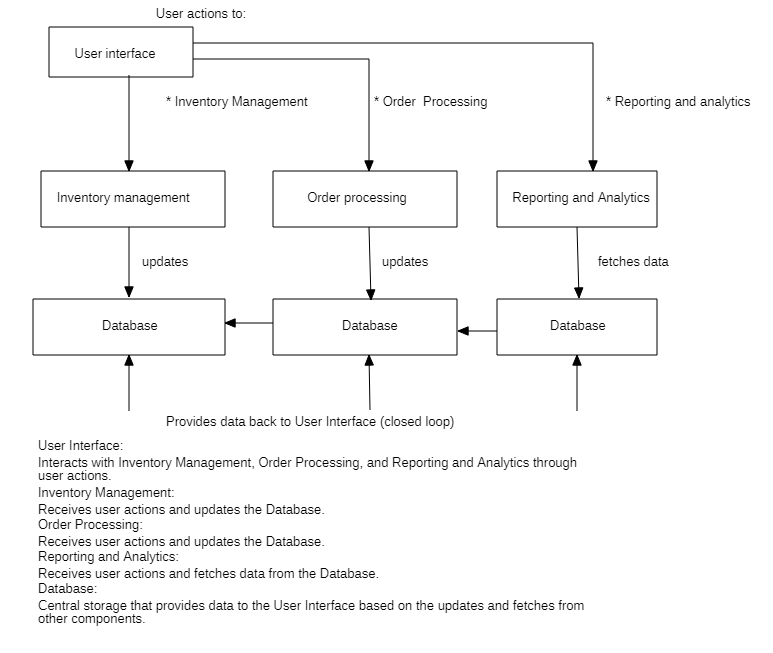
**Non-Functional Requirements**

1. **Scalability**: The system should be able to handle an increasing number of users and transactions.
2. **Security**: Ensure data security and user privacy through encryption and secure access controls.
3. **Usability**: The system should have an intuitive and user-friendly interface.
4. **Performance**: The system should be responsive and provide quick access to information.
5. **Reliability**: Ensure system availability and minimal downtime.

3.3.4.3 System Design Methodology

**3.3.3 Functional Diagram**

The functional diagram will illustrate the relationship between the principal components of the system, including user interfaces, database, inventory management, order processing, and reporting modules.



**3.3.4 Methodological Approach**

**3.3.4.1 Data Collection Techniques**

To gather the necessary information for system development, the following techniques were used:

1. **Interviews**: Conducted with store staff to understand current processes and challenges.
2. **On-site Observation**: Observed the day-to-day operations of the store to identify inefficiencies.
3. **Questionnaires**: Distributed to customers to gather feedback on their shopping experience and preferences.

**3.3.4.2 Software Development Methodology**

The Agile model was selected for the development of the system. Agile methodology is advisable as it engages multiple software developers in just one project to share knowledge and experiences. The iterative nature of Agile allows for continuous improvement and adaptation based on feedback.

Steps followed in the Agile model:

1. **Requirement Gathering and Analysis**: Collected detailed requirements from stakeholders.
2. **Design**: Created initial design prototypes and system architecture.
3. **Development**: Developed the system in iterative cycles, with each iteration building on the previous one.
4. **Testing**: Conducted testing at the end of each iteration to identify and fix issues.
5. **Deployment**: Deployed the system for user testing and feedback.
6. **Review and Maintenance**: Continuously reviewed and improved the system based on user feedback.

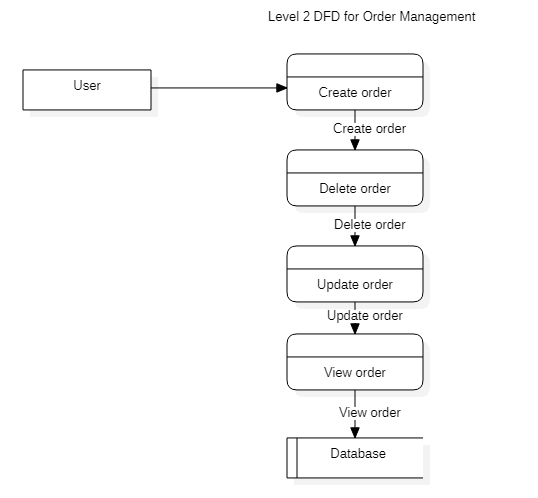
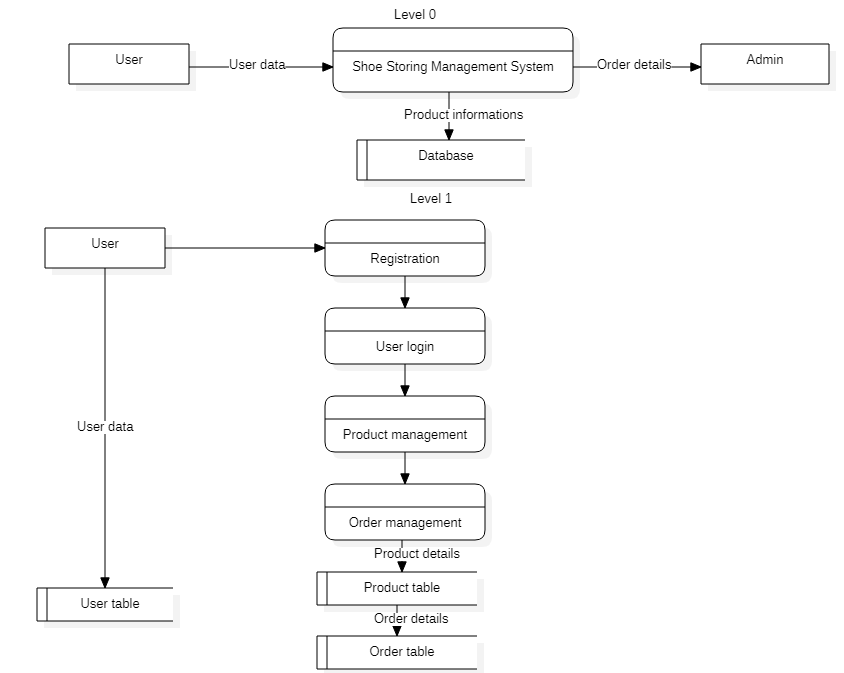
**3.3.4.3 System Design Methodology**

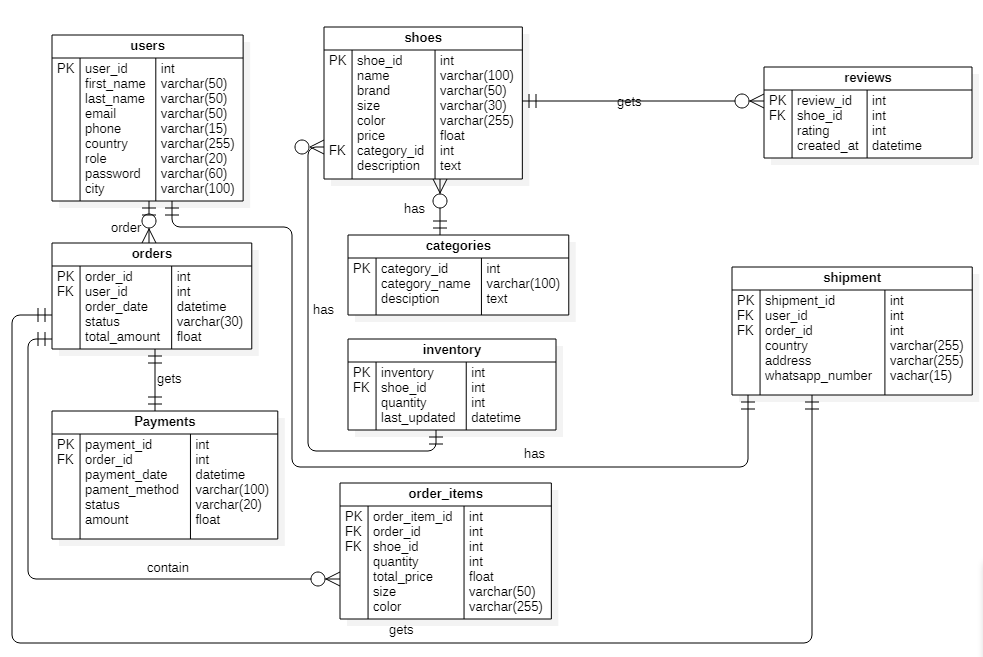
The Structured System Analysis and Design Methodology (SSADM) was chosen for system design. This approach is suitable for creating a system with well-defined and clear processes.

**Tools used in SSADM:**

* **Dataflow Diagram (DFD)**: Represents the flow of data through the system at various levels.

**Level 0,1 and 2: Context DFD**



**Entity Relationship Diagram (ERD)**: Exhibits and explains all the entities found in the proposed Database, their relationships, and attributes  
  
 **ii. Relationships Among Entities**

1. **Users - Orders**: One-to-Many (One user can place many orders)
2. **Orders - Order\_Items**: One-to-Many (One order can contain multiple order items)
3. **Orders - Payments**: One-to-One (Each order can have one payment associated)
4. **Shoes - Categories**: Many-to-One (Many shoes can belong to one category)
5. **Shoes - Reviews**: One-to-Many (One shoe can have multiple reviews)
6. **Shoes - Inventory**: One-to-One (Each shoe has one inventory record)
7. **Users - Shipment**: One-to-Many (One user can have many shipments)
8. **Orders - Shipment**: One-to-One (Each order can have one shipment record)

**Data Dictionary**: Lists all attributes found in every table created in the database. For each field, the name, data type, and specific constraints are provided.

