a.

General Description of the Original Java Program :

The Java program is an e-commerce management system that simulates the interaction between buyers and sellers within a simple shopping environment.

sellers can register and list products for sale, while buyers can register browse products by category, add items to shopping carts and place orders.

the core of the program is an ECommerceSystem manager class that handles the main business logic, including storing users and managing carts and orders.

A menu-driven console interface allows users to interact with the system, with each

input delegating to appropriate methods within the manager class.

Built using object-oriented principles, the system includes core classes such as Buyer, Seller, Product, Cart, and Order, each encapsulating its own data and behavior, while ECommerceSystem serves as the central coordinator between them.

b.

The database is required to store persistent information about the core entities of the e-commerce system. It ensures that key data is saved between sessions, enabling consistent access to user and product information.

Specifically, the database is used to store the following types of data:

**Buyers** – including their usernames and order history

**Sellers** – including their usernames and the products they offer

**Products** – including name, price, quantity, and category

**Shopping Carts** – holding the list of products selected by buyers before purchase

**Orders** – recording completed purchases with product details, shipping address, and

total price.

in future expansions, the database may also store additional entities such as delivery information, product reviews, or payment records.

requirement for database:

* Persistency – allows us to save data throw time, even that application is closed, and save information that be used in future.
* Multi-user Access – enables multiple users to connect simultaneously and retrieve information, which is essential for scalability
* Querying & Filtering – allows efficient access to specific data, such as products by category or orders by date.
* Data Integrity & Constraints – ability to supervise on data integrity, throw constraints and keys.
* Scalability – supports growth of the system by allowing the addition of new buyers, sellers, and products.
* Integration with External Systems – enables different components such as APIs, mobile apps, to access and update the same shared data.

c.

users of the system :

The system supports multiple users , each with different roles and responsibilities.

* Buyers – regular users who can register, browse products by category, add items to their shopping cart and place orders.
* Sellers - responsible for listing products for sale, each seller can add, update or remove products from their catalog.
* Administrators – system-level users who may have higher privileges such as approving sellers, managing users.

d.

Entities :

* Buyer ( **BuyerID**, username, password, address, orderHistorySize)
* Seller **( SellerID**, username, password, productCount)
* Administrator(**AdministratorID**, username, password )
* Product ( **ProductID**, SellerID, name, price, catagory, spacialpackging, packagingcost)
* Order ( **OrderID**, BuyerID, shippingAddress, totalprice, orderdate)
* Cart (**CartID**, BuyerID)
* Category ( **CatagoryID**, catagory)
* Cart\_products (**CartID, ProductID**)
* Order\_Products ( **OrderID, ProductID**)

Relationships between entities :

* Buyer has Orders (1:M)
* Buyer has Cart (1:1)
* Seller sell Products (1:M)
* Product has Category (1:1)
* Order has Products (M:N)
* Cart has Product (M:N)