# **Chapter II: System Development Process**

**1.Analysis**

**1.1 Requirement Analysis**

The requirements were gathered through discussions with the café owner and observation of day-to-day operations. Stakeholders identified include café staff (cashiers, baristas), the café owner (admin), and customers (indirect users).

**Functional Requirements:**

* Product and category listing
* Order placement and modification
* Automated billing and receipt generation
* Daily/weekly/monthly sales reporting
* Admin dashboard for item management and price updates
* Login authentication for staff and admin

**Non-Functional Requirements:**

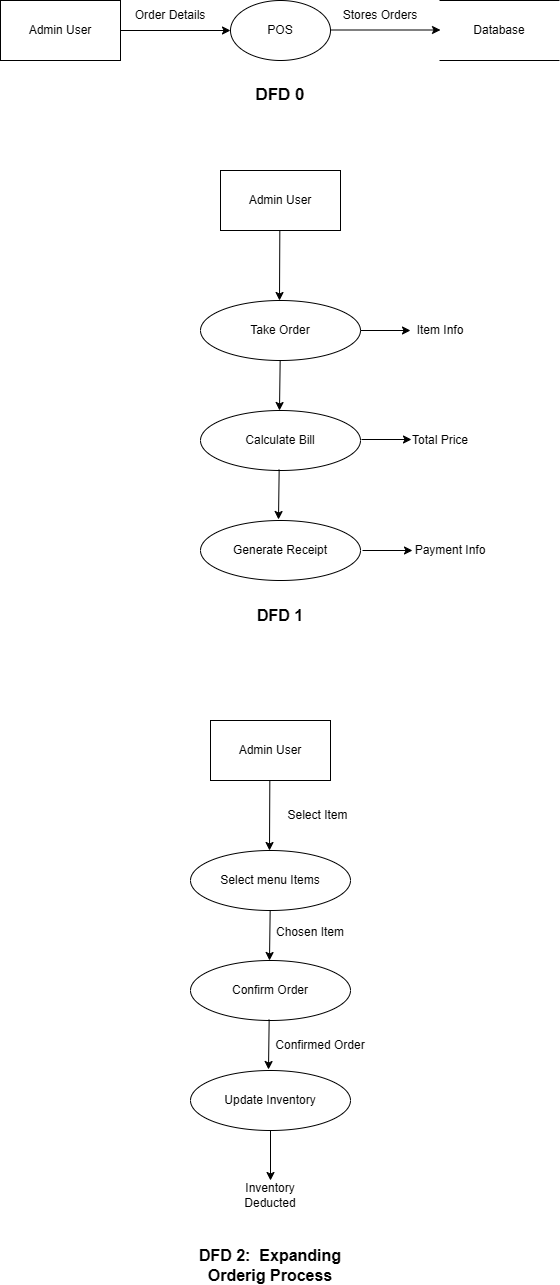
* User-friendly interface with minimal learning curve
* Quick response time for transaction processing
* Secure data handling

**1.2 Feasibility Study**

* **Technical Feasibility:**  
  The system is technically feasible using modern web technologies such as React, Node.js, and MySQL/Supabase. These are accessible and offer rapid development capabilities.
* **Operational Feasibility:**  
  Café staff can quickly learn and adapt to the interface. Simple UI design and guided workflows ensure minimal training.
* **Economic Feasibility:**  
  Since the project will be built using open-source technologies and does not rely on expensive hardware integration, development costs are minimal.

**1.3 Structured Modelling**

**1.3.1 DFD Diagram :**

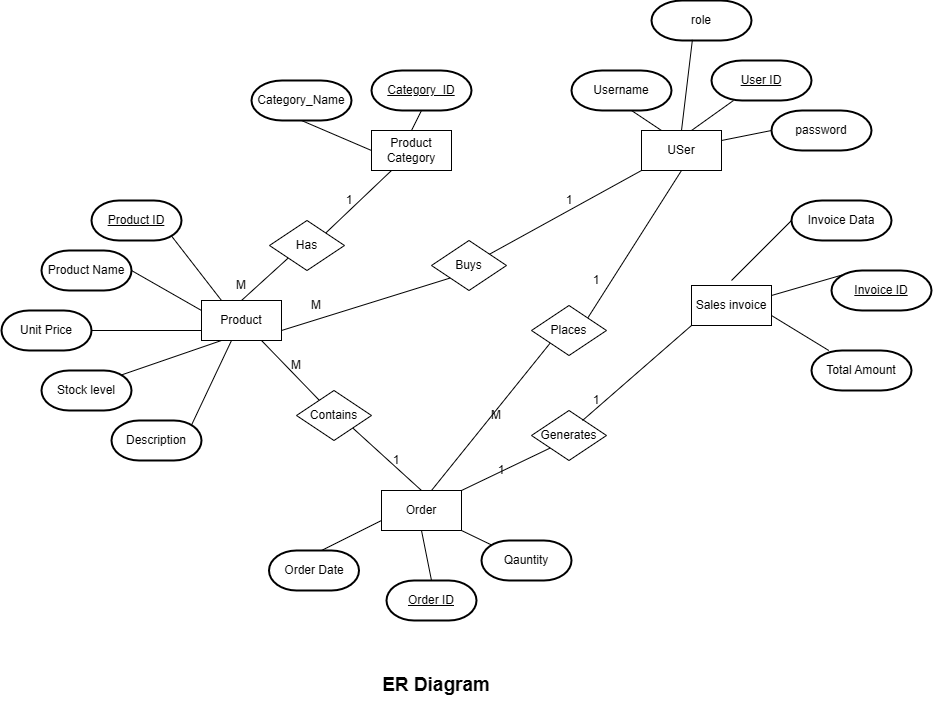


 **Level 0**: Entire POS system as one process.

 **Level 1**: Core processes — Take Order, Calculate Bill, Process Payment, Generate Receipt.

 **Level 2**: Detailed view of “Take Order”.

**1.3.2 ER Diagram :**



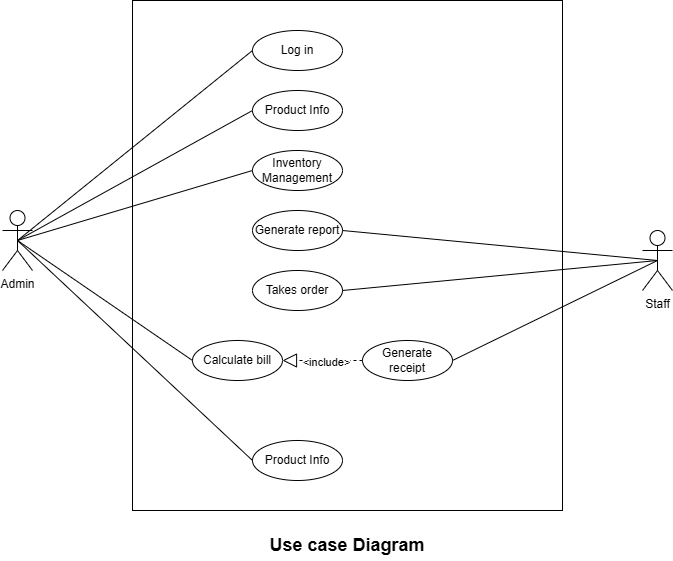
**ER Diagram Key Entities:**

* Users (userId, username, password, role)
* Products (productId, name, category, price, quantity,description)
* Product Category(Category name,Category\_ID)
* Orders (orderId, quantity,order date))
* SalesInvoice (InvoiceData, InvoiceId, totalAmount)

**Relationships:**

* One User can place many Orders
* One User can buys many Products
* One Order contains many Products
* One Order generates one sales Invoice

**1.3.3 Use Case diagram**



**2. Design**

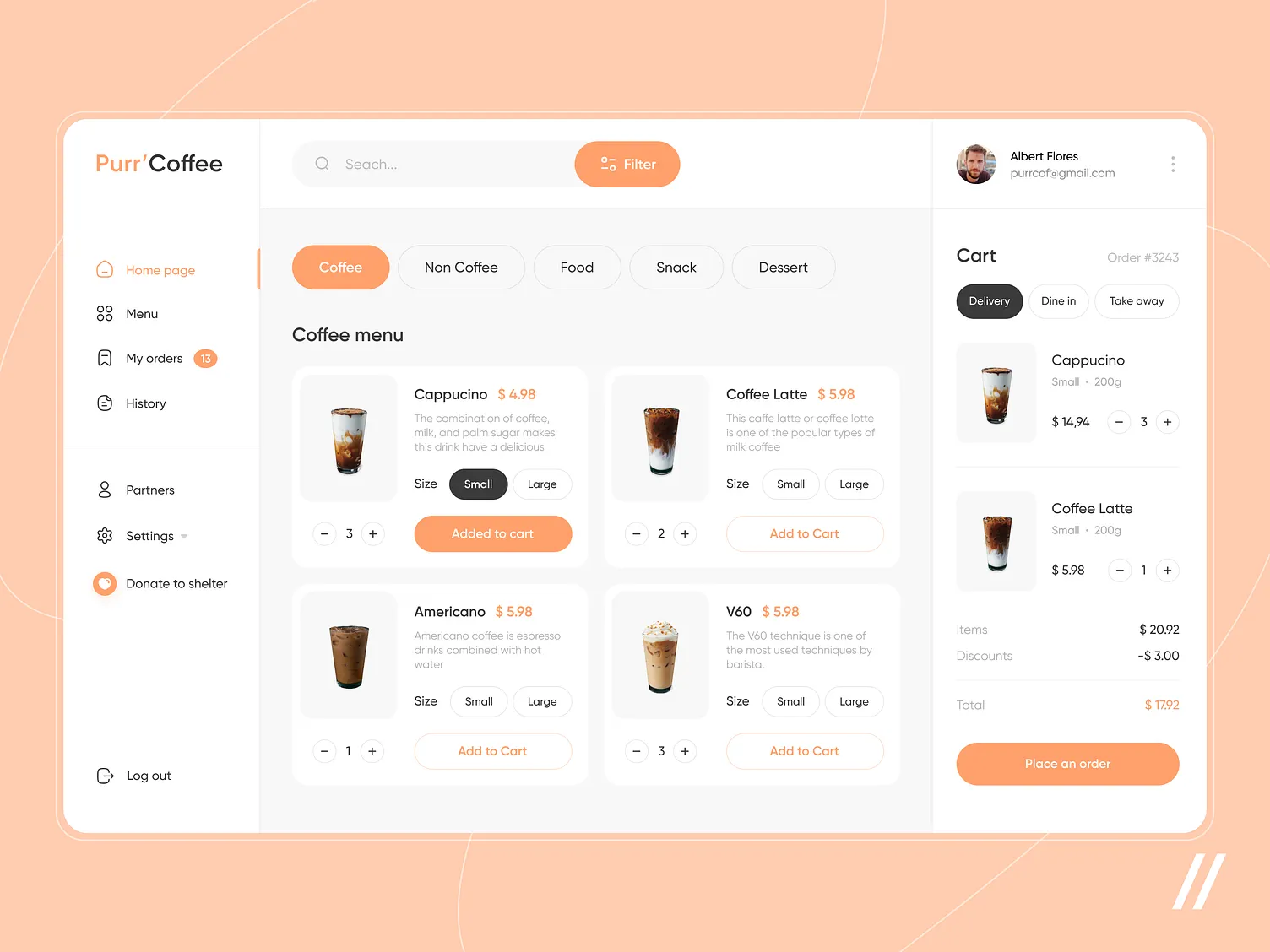
**2.1 User Interface Design**:

* **Login Screen**
* **POS Terminal Interface**: Product list, order summary, “Place Order” button
* **Admin Dashboard**: Product management, sales report view

**Design Principles:**

* Minimalist layout with clear call-to-actions
* Use of color-coded buttons for quick navigation
* Responsive layout for desktop use

**Sample UI:**



**2.2 Database Design / Object-Oriented Design Models**

### 🔹 1. User

| **Field** | **Data Type** | **Constraints** |
| --- | --- | --- |
| UserID | INT | PRIMARY KEY |
| Username | VARCHAR(50) | UNIQUE, NOT NULL |
| Password | VARCHAR(255) | NOT NULL |
| Role | VARCHAR(30) | NOT NULL (e.g. Admin, Cashier) |

### 🔹 2. ProductCategory

| **Field** | **Data Type** | **Constraints** |
| --- | --- | --- |
| CategoryID | INT | PRIMARY KEY |
| CategoryName | VARCHAR(50) | NOT NULL |

### 🔹 3. Product

| **Field** | **Data Type** | **Constraints** |
| --- | --- | --- |
| ProductID | INT | PRIMARY KEY |
| ProductName | VARCHAR(100) | NOT NULL |
| UnitPrice | DECIMAL(8,2) | NOT NULL |
| StockLevel | INT | NOT NULL DEFAULT 0 |
| Description | TEXT | NULLABLE |
| CategoryID | INT | FOREIGN KEY → ProductCategory(CategoryID) |

### 🔹 4. Order

| **Field** | **Data Type** | **Constraints** |
| --- | --- | --- |
| OrderID | INT | PRIMARY KEY |
| OrderDate | DATETIME | NOT NULL |
| UserID | INT | FOREIGN KEY → User(UserID) |

### 🔹 5. OrderItem

| **Field** | **Data Type** | **Constraints** |
| --- | --- | --- |
| OrderID | INT | FOREIGN KEY → Order(OrderID) |
| ProductID | INT | FOREIGN KEY → Product(ProductID) |
| Quantity | INT | NOT NULL |
| **Composite PK**: OrderID, ProductID |  |  |

### 🔹 6. SalesInvoice

| **Field** | **Data Type** | **Constraints** |
| --- | --- | --- |
| InvoiceID | INT | PRIMARY KEY |
| OrderID | INT | FOREIGN KEY → Order(OrderID) |
| InvoiceDate | DATETIME | NOT NULL |
| TotalAmount | DECIMAL(10,2) | NOT NULL |

## **Relationships Overview**

* One User places many Orders
* Each Order contains many Products (via OrderItem)
* Each Order generates one SalesInvoice
* Each Product belongs to one Category
* No customer data is stored — only internal staff manages orders and inventory.

**3. Implementation**

**3.1 Tools and Technologies Used**

* **Frontend:** React.js, Tailwind CSS
* **Backend:** Node.js with Express.js
* **Database:** MySQL or Supabase
* **Development Tools:** VS Code, GitHub
* **Deployment:** Local server, cloud deployment (Vercel)

**3.2 Module Description**

| **Module** | **Description** |
| --- | --- |
| **Authentication** | Allows login for staff with role-based access |
| **Order Management** | Select products, adjust quantities, generate order |
| **Billing System** | Calculates totals, taxes, and generates printable bill |
| **Product Management** | Admin can add/update/delete products |
| **Sales Reporting** | Displays sales data with filters (daily/weekly/monthly) |

**3.3 Testing**

* **Unit Testing:** Conducted on individual functions (e.g., price calculation, login validation)
* **Integration Testing:** Ensures seamless flow from order creation to sales recording
* **User Acceptance Testing (UAT):** To be conducted with café staff to verify ease of use and accuracy