# System Design Document: E-commerce App

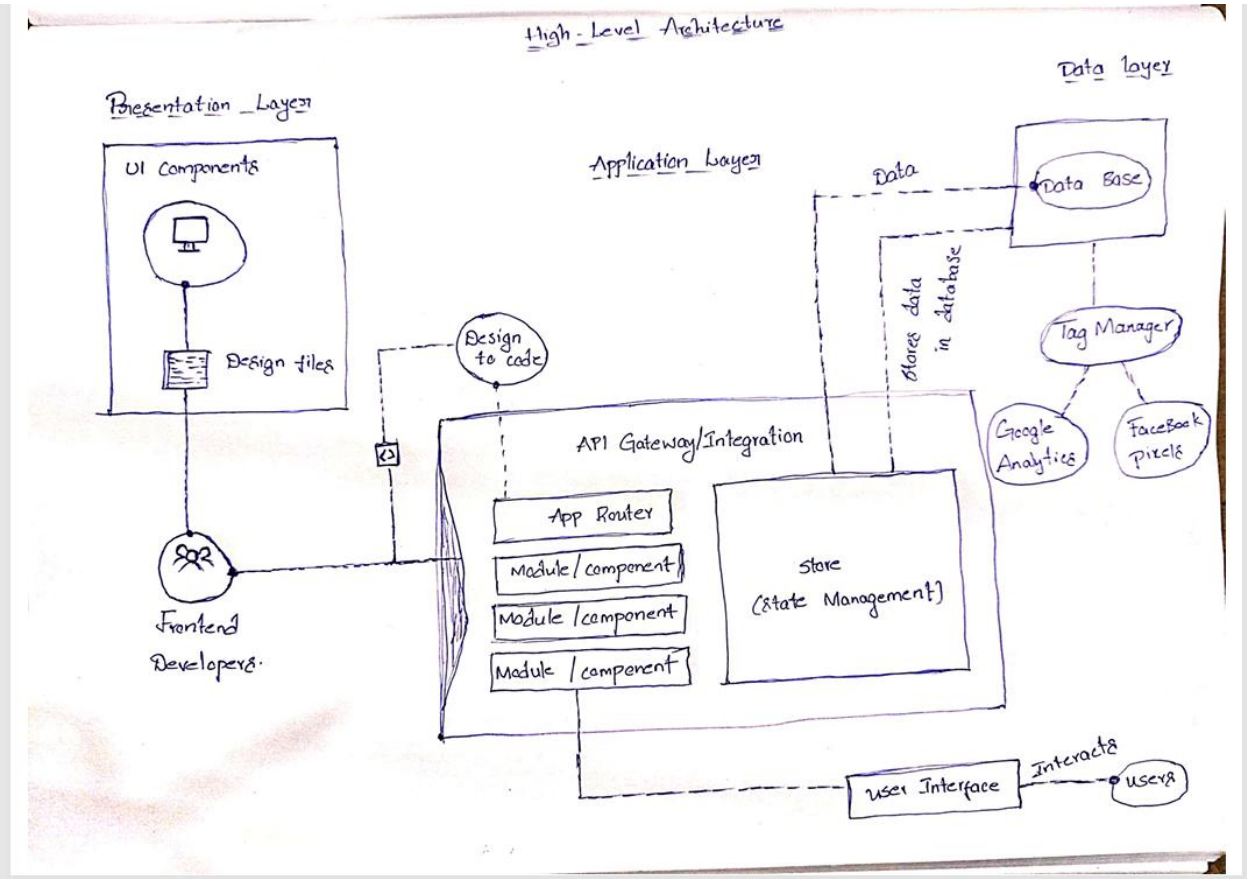
## 1. Introduction

**1.1 Purpose:** To design and develop a robust e-commerce application that enables users to browse, select, and purchase products online.

**1.2 Scope:** This document outlines the high-level system design, including the system architecture, data flow, and key components.

## 2. System Architecture

### 2.1 High-Level Architecture:




layered architecture diagram

- **Presentation Layer:**
  - User interface (web and mobile)
  - Handles user interactions and displays information
- **Business Logic Layer:**

    - Manages business rules and processes
    - Handles product catalog, shopping cart, and order processing
- **Data Access Layer:**
    - Interacts with the database
    - Retrieves and stores product, user, and order information

**2.2 Technology Stack:**

- **Frontend:** React, Angular, or Vue.js
- **Backend:** Node.js, Python (Django or Flask), or Ruby on Rails
- **Database:** MySQL, PostgreSQL, or MongoDB
- **Cloud Platform:** AWS, Azure, or GCP

## 3. Data Flow

## data flow diagram

1. **User Interaction:**
    - User browses product catalog.
    - User adds products to the shopping cart.
    - User proceeds to checkout.
2. **System Processing:**
    - **Product Catalog:** Fetches product information from the database.
    - **Shopping Cart:** Stores selected products and calculates the total price.
    - **Checkout:** Processes payment, generates order, and updates inventory.
3. **Data Storage:**
    - **Product Information:** Stores product details, images, and prices.
    - **User Information:** Stores user profiles and order history.
    - **Order Information:** Stores order details, payment information, and shipping address.

## 4. Key Components and Their Responsibilities

- **User Interface:** Provides a user-friendly interface for browsing products, searching, and checkout.
- **Product Catalog:** Manages product information, including categories, descriptions, and pricing.
- **Shopping Cart:** Allows users to add, remove, and modify products in their cart.
- **Checkout:** Handles payment processing, order confirmation, and shipping information.
- **User Management:** Manages user accounts, profiles, and preferences.
- **Inventory Management:** Tracks product stock levels and updates inventory upon order fulfillment.
- **Order Management:** Processes orders, generates invoices, and handles returns and refunds.

- **Payment Gateway:** Integrates with payment gateways to process payments securely.

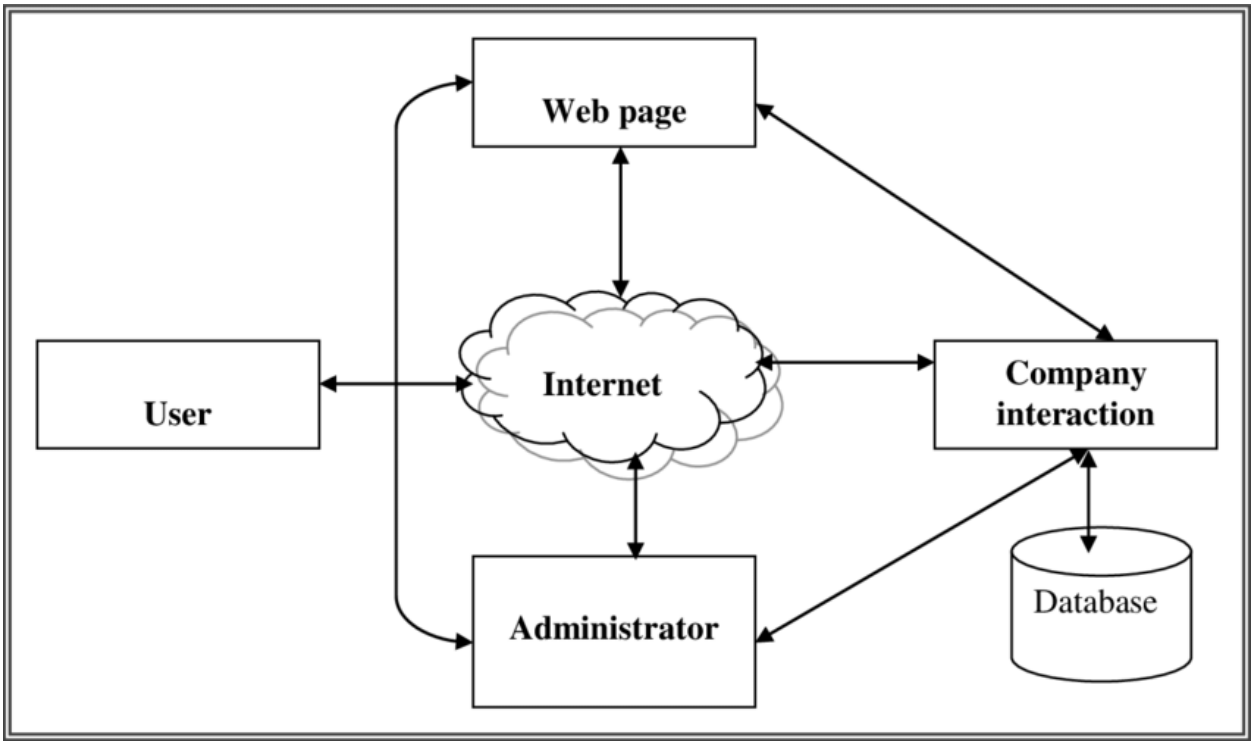
## 5. Security Considerations

- **Data Encryption:** Encrypt sensitive data like credit card information.
- **Secure Authentication:** Implement strong authentication mechanisms.
- **Input Validation:** Validate user input to prevent malicious attacks.
- **Regular Security Audits:** Conduct regular security assessments.

## 6. Future Considerations

- **Mobile App:** Develop a mobile app for iOS and Android.
- **Personalization:** Implement personalized product recommendations.
- **Loyalty Program:** Introduce a loyalty program to reward customers.
- **Social Integration:** Allow users to share products on social media.

By following this system design, we can build a scalable and user-friendly e-commerce application.



Google Network Ecommerce Diagram: