

* **Client Devices**: Represent the user and driver applications, connected to the system through the API Gateway.
* **API Gateway**: Routes requests from clients to appropriate backend services.
* **Ride Management Service**: Manages ride bookings and statuses.
* **Payment Service**: Handles payments and transactions.
* **User Management Service**: Manages user profiles and authentication.
* **Geolocation & Matching Service**: Handles real-time location tracking and driver-rider matching.
* **Notification Service**: Sends real-time notifications to users and drivers.
* **Driver Management Service**: Manages driver profiles, availability, and ratings.

Online shopping system

--------------------------------

| Presentation |

| (Web Interface, Mobile App) |

+-------------------------------

|

v

+---------------------------------+

| Application Layer |

| (Controllers: ProductController)|

| (Services: ShoppingCartService) |

+---------------------------------+

|

v

+---------------------------------+

| Business Logic Layer |

| (Business Rules: Discount Rules)|

| (Domain Models: Product, Order) |

+---------------------------------+

|

v

+---------------------------------+

| Data Access Layer |

| (Repositories: ProductRepo) |

| (ORM: Hibernate) |

+---------------------------------+

|

v

+---------------------------------+

| Database Layer |

| (MySQL: Products, Orders, Users)|

+---------------------------------+

### **Detailed Flow:**

1. **User Interaction:**
   * The user opens the web interface or mobile app, searching for a product.
2. **Request Handling:**
   * The search request is sent to the ProductController in the **Application Layer**.
3. **Business Processing:**
   * ProductController invokes the ShoppingCartService in the **Business Logic Layer** to apply any business rules, such as checking inventory or calculating discounts.
4. **Data Retrieval:**
   * The ShoppingCartService uses the ProductRepository in the **Data Access Layer** to query the database for the requested product information.
5. **Data Response:**
   * The ProductRepository retrieves the data from the **Database Layer** (MySQL in this case) and returns it through the layers back to the user interface.
6. **Presentation:**
   * The product information is presented to the user on the web or mobile interface.

This layered architecture ensures that each part of the system is modular, making it easier to maintain, scale, and test.