

1. Introduction

1.1 Purpose

The purpose of this architecture document is to provide an overview of the system's architecture for the Online Retail Database Management System. It outlines the key architectural components and decisions that guide the design, deployment, and management of the system.

1.2 Scope

This document covers the architectural design and structure of the Online Retail Database Management System. It defines various views such as logical, process, implementation, and deployment views to illustrate how the system components interact and function together.

1.3 Definitions, Acronyms, and Abbreviations

- DBMS: Database Management System
- UI: User Interface
- API: Application Programming Interface
- ERP: Enterprise Resource Planning
- SKU: Stock Keeping Unit
- SSL/TLS: Secure Socket Layer/Transport Layer Security

1.4 References

- Online Retail SRS Document (Revision 1.0)
- IEEE Standard for Software Architecture (IEEE Std 1471)
- Database Design Best Practices
- MySQL Official Documentation
- Python Documentation

2. Architectural Representation

The architecture used is a three-tier type of architecture comprising the following layers:

Presentation Layer (User Interface):

- This is the top layer, representing how users interact with the system. It includes customers, administrators, and customer service representatives accessing the platform through web browsers or mobile devices.

Application Layer (Business Logic):

- The middle layer handles the business logic. This is where the application server processes user requests, interacts with the database, and manages inventory, customer orders, product details, etc. It includes services like product management, order management, and payment processing.

Data Layer (Database):

- The bottom layer stores the data and ensures secure, efficient data retrieval. This is where all inventory, customer information, orders, and transactional data are maintained. The database interacts with external services such as payment gateways.

The architecture described is based on the "4+1 View Model" of software architecture, which organizes the system using different perspectives to address various stakeholder concerns.

3. Architectural Goals and Constraints

- Security: All customer data is encrypted during transmission and storage.
- Scalability: The system is designed to handle up to 10,000 simultaneous users.
- Reliability: The system must maintain 99.9% uptime.
- Integration: Interfaces with external payment gateways and shipping providers.
- Compliance: Ensure compliance with GDPR and other relevant data protection laws.
- Limited Resources: A small team (2 people) and a short timeline (3 months) limit the complexity of the system.
- Performance: Must handle 1000 users simultaneously with response times under 5 seconds.

4. Use-Case View

4.1 Architecturally Significant Use Cases

- Product Management: Add, update, delete, and search for products.
- Order Processing: Process customer orders, update inventory, and handle payment transactions.
- Customer Profile Management: Maintain and update customer information.

- Report Generation: Generate sales and inventory reports.
- Error Handling: Notify users and log system errors.

5. Logical View

5.1 Architecture Overview – Package and Subsystem Layering

This section defines the system's logical architecture using a layered approach:

- Presentation Layer: Handles user interfaces for administrators, retail managers, customer service representatives, and customers.
- Business Logic Layer: Contains business rules for product management, order processing, customer management, and reporting.
- Data Access Layer: Manages database interactions, including queries for inventory, customer profiles, and order data.
- Integration Layer: Facilitates interaction with external APIs, such as payment gateways and shipping providers.

6. Process View

6.1 Processes

Key processes include product management, order processing, and customer management, each involving multiple layers of the system.

6.2 Process to Design Elements

This section maps major system processes to the design elements:

- Order Processing: Maps to the order management module, inventory database, and payment gateway API.
- Product Management: Maps to the product database and business logic for updates and searches.

6.3 Process Model to Design

The system adopts a client-server architecture with RESTful API interfaces for external systems such as payment gateways.

6.4 Model Dependencies

- The system depends on secure communication between the database server and external services like payment and shipping providers.
- Front-end depends on MySQL for data retrieval and updates.
- MySQL queries manage database operations and CRUD tasks.

6.5 Processes to the Implementation

- Order Processing Implementation: RESTful API integrated with payment gateways.
- Customer Management Implementation: SQL-based queries to maintain customer profiles.

7. Deployment View

7.1 External Desktop PC

Access for administrators and managers to view and update system information.

7.2 Desktop PC

Used by customer service representatives to manage customer inquiries.

7.3 Registration Server

Handles all backend processes for user registration and authentication.

7.4 Course Catalog

Displays available products for customers to browse and order.

7.5 Billing System

Integrated with external payment gateways to process payments and update order statuses.

8. Performance

The system should support users and enable real-time transaction processing with minimal latency to ensure responsiveness

9. Quality

The system ensures quality through adherence to data protection regulations, user-friendly interfaces, and robust error-handling mechanisms.