Software Requirements Specification

*“Elegant : E-Commerce Platform for Jewelry Products”*

Version 1.0

Prepared by Project Team – 16

|  |  |
| --- | --- |
| **Name** | **PRN** |
| Pratik Dilip Shahane | 240840520053 |
| Anchal Gupta | 240840520008 |
| Aadhar Dhote | 240840320001 |
| Neha Shashikant Patil | 240840520039 |
| Vaishali Vijay Padwal | 240840320123 |

Centre for Development of Advanced Computing : Raintree Marg, Near Bharati Vidyapeeth, Opp. Kharghar Railway Station, Sector 7, CBD Belapur, Navi Mumbai - 400 614 - Maharashtra (India) Phone: +91-22-27565303 Fax: +91-22-2756-0004.

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 2

1.4 References 3

2. Overall Description 3

2.1 Product Perspective 3

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 6

2.6 User Documentation 6

2.7 Assumptions and Dependencies 7

3. External Interface Requirements 7

3.1 User Interfaces 7

3.2 Hardware Interfaces 9

*4.* External Interface Requirements 11

*4.1* User Interfaces 11

*4.2* Hardware Interfaces 12

*4.3* Software Interfaces 13

*4.4* Communications Interfaces 13

5. Other Nonfunctional Requirements 14

*5.1* Performance Requirements 14

*5.2* Security Requirements 15

*5.3* Usability Requirements 15

*5.4* Maintainability & Support Requirements 16

*5.5* Business Rules 16

6. Other Requirements 17

*6.1* Regulatory & Legal Compliance 17

*6.2* Licensing & Open-Source Dependencies 18

*6.3* Environmental & Operational Constraints 18

*6.4* Future Enhancements 19

Appendix A: Glossary 19

Appendix B: Analysis Models 20

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## Purpose

**Elegant :** **E-Commerce Platform for Jewelry Products** is being developed to provide an **interactive and user-friendly online shopping experience** for customers while offering **robust management tools** for administrators. This document outlines the **functional and nonfunctional requirements** necessary to design, develop, and implement the system successfully.

The **primary goals** of this platform are:

* To provide a seamless online **shopping experience** for customers interested in **luxury jewelry**.
* To offer **advanced product search and filtering capabilities**, ensuring customers can easily find desired products.
* To enable secure **payment processing** through integrated gateways like **Razorpay and Stripe**.
* To allow **order tracking** with real-time status updates.
* To provide administrators with an intuitive **dashboard** for managing products, orders, and customers.

This document serves as a **point of reference** for:

* **Developers**: To understand the functional scope and implement system features.
* **Testers**: To create test cases and validate functionalities against specifications.
* **Project Managers**: To ensure the project is delivered on time and meets stakeholder expectations.
* **Business Owners/Stakeholders**: To assess the project's value and monitor system capabilities.

By defining **clear and structured requirements**, this document ensures **consistency, reliability, and scalability** in the system’s development lifecycle.

.

## Intended Audience and Reading Suggestions

This document is intended for multiple **stakeholders** involved in the design, development, and deployment of the e-commerce platform.

**1.2.1 Stakeholder Roles**

The table below outlines the intended audience and their reasons for referring to this document:

| **Stakeholder** | **Purpose of Reading** |
| --- | --- |
| **Developers** | Implement system features based on detailed requirements. |
| **Testers** | Design test cases and validate functionalities against specifications. |
| **Project Managers** | Track project progress and align with business goals. |
| **UI/UX Designers** | Ensure the user interface meets user expectations. |
| **System Architects** | Design the system architecture based on constraints and requirements. |
| **Business Owners** | Evaluate the system's effectiveness in meeting business objectives. |

**1.2.2 Reading Suggestions**

Depending on the reader’s role, different sections of this document will be more relevant:

* **For a High-Level Overview**: Refer to **Section 1 (Introduction)** and **Section 2 (Overall Description)**.
* **For Feature Development**: Refer to **Section 3 (System Features)** and **Section 4 (External Interface Requirements)**.
* **For Integration and API Details**: Refer to **Section 4 (External Interface Requirements)**.
* **For Performance, Security, and Compliance**: Refer to **Section 5 (Nonfunctional Requirements)**.
* **For System Constraints and Business Rules**: Refer to **Section 6 (Other Requirements)**.

This structure ensures that readers can **quickly navigate** to the most relevant sections based on their role.

## Product Scope

The **E-Commerce Platform for Jewelry Products** is designed to **bridge the gap between traditional and digital jewelry retail** by providing a **feature-rich, scalable, and secure** online shopping experience.

The system will include two primary user roles:

1. **Customers**: End-users who browse jewelry, add items to a cart, and complete purchases.
2. **Administrators**: Business managers responsible for managing products, inventory, and customer orders.

**1.3.1 Core Functionalities**

The system will provide the following key functionalities:

✅ **Customer Functionalities**

* **Product Catalog**: Customers can browse, filter, and search for jewelry.
* **Shopping Cart**: Items can be added, updated, or removed.
* **Checkout & Payments**: Secure transactions via Razorpay and Stripe.
* **Order Tracking**: Real-time updates on order status (Pending, Shipped, Delivered).
* **Reviews & Ratings**: Customers can rate and review purchased products.

✅ **Admin Functionalities**

* **Product Management**: Admins can add, update, and delete products.
* **Order Management**: Orders can be viewed, updated, and canceled.
* **Customer Management**: Admins can view and manage customer accounts.
* **Sales Reports**: Generate revenue and performance analytics.

**1.3.2 System Benefits**

* **For Customers**:
  + A **seamless** and **secure** shopping experience.
  + Multiple payment options with real-time order tracking.
  + Personalized recommendations based on purchase history.
* **For Administrators**:
  + A **centralized dashboard** for product and order management.
  + Data-driven **insights into customer behavior** and sales trends.
  + **Role-based access control** for security and operational efficiency.

**1.3.3 Limitations**

* The system **requires internet connectivity** for all features.
* Payment processing **depends on third-party APIs (Razorpay, Stripe)**.
* Users **must create an account** for order tracking and purchase history.

## References

The following references were consulted in preparing this document:

| **Reference** | **Description** |
| --- | --- |
| IEEE Standard 830-1998 | Guidelines for writing Software Requirements Specifications (SRS). |
| ReactJS Documentation | Official documentation for frontend development (<https://reactjs.org/docs/>). |
| Spring Boot Documentation | Official documentation for backend development (<https://spring.io/projects/spring-boot>). |
| Razorpay API Documentation | Secure payment gateway API documentation (<https://razorpay.com/docs/api/>). |
| Stripe API Documentation | API reference for handling payments (<https://stripe.com/docs/api>). |

# Overall Description

## Product Perspective

The **E-Commerce Platform for Jewelry Products** is a **standalone web-based system** designed to bridge the gap between traditional jewelry retail and modern e-commerce. The system is developed as a **full-stack web application**, integrating both frontend and backend technologies to provide a seamless experience for **customers** and **administrators**.

### Relationship with Existing Systems

This system is **not an extension** of any existing software; it is an independent, full-fledged e-commerce solution.

It integrates with **third-party payment gateways (Razorpay, Stripe)** for secure payment processing.

The backend API can be extended to support **mobile applications** in the future.

### Key Technologies

| **Component** | **Technology** |
| --- | --- |
| **Frontend** | ReactJS, Material-UI |
| **Backend** | Spring Boot, Hibernate |
| **Database** | MySQL |
| **Security** | JWT Authentication, BCrypt Encryption |
| **Hosting** | AWS Cloud |
| **Payment Integration** | Razorpay, Stripe |

## Product Functions

The system consists of two primary modules:

1. Customer Module – Features designed for customers to browse, purchase, and track orders.
2. Admin Module – Features for administrators to manage products, orders, and customers.

### Customer Module Features

✅ Product Catalog

* Customers can browse, filter, and search for jewelry.
* Each product has detailed descriptions, multiple images, and real-time stock status.

✅ Shopping Cart

* Customers can add products to their cart, update quantities, or remove items.
* Persistent cart for logged-in users.

✅ Checkout & Payment

* Secure payment via Razorpay and Stripe.
* Dynamic calculation of total cost, including applicable discounts.

✅ Order Tracking

* Customers can track real-time order status (Pending → Confirmed → Shipped → Delivered).
* Automatic notifications via email/SMS.

✅ Reviews & Ratings

* Customers can submit star-based ratings (1-5) and written reviews.
* Admins can moderate and remove inappropriate reviews.

### Admin Module Features

✅ Product Management

* Add, update, and delete products from the catalog.
* Set discounts, pricing, stock levels, and featured items.

✅ Order Management

* Track customer orders and update status.
* View order history, process returns, and handle cancellations.

✅ Customer Management

* View and manage customer accounts.
* Disable fraudulent accounts and reset passwords if needed.

✅ Sales Reports & Analytics

* Generate reports on total sales, best-selling products, and customer behavior.
* View monthly sales performance with graphical insights.

## User Classes and Characteristics

The system is designed for two **primary user roles**: **Customers** and **Administrators**.

| **User Class** | **Description** | **Skills Required** |
| --- | --- | --- |
| **Customers** | End-users who browse, purchase jewelry, and track orders. | Basic internet usage skills. |
| **Administrators** | Manage products, orders, and customer details. | Familiarity with e-commerce management. |

**Additional User Considerations:**

* **First-time users** should be able to navigate easily through an **intuitive UI**.
* **Admins should have role-based access control** (e.g., product manager vs. order manager).
* **System logs** should track administrator actions to ensure security.

## Operating Environment

The platform is designed to run as a **web application**, accessible from **desktop, tablet, and mobile devices**.

### Hardware Requirements

| **Component** | **Minimum Requirement** | **Recommended Requirement** |
| --- | --- | --- |
| **Processor** | Intel Core i3 or equivalent | Intel Core i5 or higher |
| **RAM** | 4GB | 8GB+ |
| **Storage** | 50GB HDD | 100GB SSD |
| **Internet Speed** | 5 Mbps | 10 Mbps+ |

### Software Requirements

| **Software Component** | **Requirement** |
| --- | --- |
| **Operating System** | Windows 10+, macOS, Linux |
| **Web Browser** | Chrome (latest), Firefox, Safari |
| **Database Server** | MySQL 8.0+ |
| **Backend Server** | Java 17+, Spring Boot |
| **Frontend Environment** | ReactJS, Node.js |

### Network Requirements

* Requires **24/7 internet connectivity** for real-time updates.
* Uses **HTTPS encryption** to ensure secure data transmission.

## Design and Implementation Constraints

The following constraints apply to the development and deployment of the system:

1. **Security Compliance**
   * All customer transactions must be **secured using HTTPS** and **JWT authentication**.
   * User passwords must be **encrypted using BCrypt hashing**.
2. **Scalability**
   * The system should support up to **10,000 concurrent users**.
   * The backend should be able to **handle 1,000 API requests per second**.
3. **Mobile Responsiveness**
   * The UI must be **fully responsive** across desktop, tablet, and mobile devices.
4. **Third-Party API Dependencies**
   * Payment processing is **dependent on Razorpay and Stripe APIs**.
   * The system requires **email and SMS notification services** for order updates.

## User Documentation

The system will provide **help documentation and tutorials** for customers and administrators.

✅ **For Customers:**

* A **step-by-step guide** on **placing orders, tracking shipments, and making payments**.
* A **FAQs section** addressing common queries about **returns, refunds, and warranties**.

✅ **For Administrators:**

* A **dashboard tutorial** explaining **product, order, and customer management**.
* **Security guidelines** on **handling customer data and sensitive transactions**.

**Documentation will be available in:**

* **PDF format** (downloadable from the admin panel).
* **An interactive help center** on the website.

## Assumptions and Dependencies

### Assumptions

Customers have a stable internet connection to use the platform.

Payment gateways (Razorpay, Stripe) are fully operational at all times.

Users are expected to log in to manage their cart and track orders.

### Dependencies

* Third-Party Services:
  + Payment gateways (Razorpay, Stripe) for transactions.
  + AWS Cloud for hosting the backend.
  + Email & SMS API for notifications.
* Regulatory Compliance:
  + Must comply with GDPR and PCI-DSS standards for data protection.

# External Interface Requirements

## User Interfaces

The **customer module** provides all functionalities required for **browsing products, adding them to a cart, placing orders, and tracking deliveries**.

**3.1.1 Product Browsing & Search**

**Description:**  
Customers can browse jewelry products, view detailed descriptions, and search using filters.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | High |
| **Actors** | Customers |
| **Input** | Customer search query, category selection, sorting criteria |
| **Output** | Display of matching products with images, prices, and availability |

* **Functional Requirements:**
* The system shall allow customers to **search for products** using **keywords** (e.g., "gold necklace").
* The system shall allow filtering by:
  + **Category** (e.g., rings, necklaces, bracelets).
  + **Price Range** (e.g., ₹500 - ₹5000).
  + **Discount** (e.g., 10%, 20%).
  + **Availability** (In Stock, Out of Stock).
* The system shall display **real-time product availability**.
* The system shall allow sorting by:
  + **Price (Low to High, High to Low)**.
  + **Newest Arrivals**.
  + **Customer Ratings**.
* The system shall display **product recommendations** based on browsing history.

**3.1.2 Shopping Cart Management**

**Description:**  
Customers can add, update, or remove products from their shopping cart before proceeding to checkout.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | High |
| **Actors** | Customers |
| **Input** | Product selection, quantity update |
| **Output** | Cart updates with total cost calculation |

* **Functional Requirements:**
* The system shall allow users to **add items** to the cart.
* The system shall allow users to **update item quantity** or **remove items** from the cart.
* The system shall display **real-time total price calculation** including applied discounts.
* The system shall persist the cart **for logged-in users** across multiple sessions.
* The system shall notify the user if a product in the cart **goes out of stock** before checkout.

**3.1.3 Checkout & Payment Processing**

**Description:**  
Customers can place an order and complete secure payments.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | High |
| **Actors** | Customers |
| **Input** | Customer address, payment details |
| **Output** | Order confirmation and payment processing |

* **Functional Requirements:**
* The system shall allow users to **select a delivery address**.
* The system shall integrate **secure payment gateways (Razorpay, Stripe)**.
* The system shall **encrypt** sensitive payment data for security.
* The system shall support **discount coupons and promo codes**.
* The system shall generate and email **order confirmation receipts** upon successful payment.

**3.1.4 Order Tracking & History**

**Description:**  
Customers can track their orders in real time.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | High |
| **Actors** | Customers |
| **Input** | Order ID |
| **Output** | Real-time order tracking |

* **Functional Requirements:**
* The system shall allow customers to **view their order history**.
* The system shall display **real-time tracking updates** (**Pending → Confirmed → Shipped → Delivered**).
* The system shall notify customers via **email/SMS** when order status changes.
* The system shall allow customers to **request order cancellations** before shipping.

**3.1.5 Product Reviews & Ratings**

**Description:**  
Customers can submit reviews and ratings for purchased products.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | Medium |
| **Actors** | Customers |
| **Input** | Rating (1-5 stars), text review |
| **Output** | Display of average product ratings |

* **Functional Requirements:**
* The system shall allow users to **submit a star-based rating (1-5 stars)**.
* The system shall allow users to **write and submit text reviews**.
* The system shall display **average product ratings** based on customer feedback.
* The system shall allow admins to **moderate and remove inappropriate reviews**.

## Hardware Interfaces

The **admin module** provides all functionalities required for managing **products, orders, and customer interactions**.

**3.2.1 Product Management**

**Description:**  
Admins can add, update, and delete products.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | High |
| **Actors** | Admin |
| **Input** | Product details (name, price, category, image) |
| **Output** | Product is added/updated/deleted |

* **Functional Requirements:**
* The system shall allow admins to **add new products** to the catalog.
* The system shall allow admins to **update product details** (price, image, stock level).
* The system shall allow admins to **delete products** from the inventory.
* The system shall display **low-stock alerts** to prevent stockouts.

**3.2.2 Order Management**

**Description:**  
Admins can track and update customer orders.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | High |
| **Actors** | Admin |
| **Input** | Order ID, status update |
| **Output** | Order is processed |

* **Functional Requirements:**
* The system shall allow admins to **view all orders**.
* The system shall allow admins to **update order status** (**Confirmed, Shipped, Delivered**).
* The system shall allow admins to process **returns and refunds**.
* The system shall allow admins to **cancel orders** if payment is not completed.

**3.2.3 Customer Management**

**Description:**  
Admins can view and manage customer accounts.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | Medium |
| **Actors** | Admin |
| **Input** | Customer account data |
| **Output** | Customer details are updated |

* **Functional Requirements:**
* The system shall allow admins to **search and filter customer accounts**.
* The system shall allow admins to **reset customer passwords**.
* The system shall allow admins to **disable fraudulent accounts**.

**3.2.4 Reports & Analytics**

**Description:**  
Admins can generate sales reports and monitor performance.

| **Feature** | **Details** |
| --- | --- |
| **Priority** | Medium |
| **Actors** | Admin |
| **Input** | Date range, sales data |
| **Output** | Graphical sales reports |

* **Functional Requirements:**
* The system shall generate **daily, weekly, and monthly sales reports**.
* The system shall display **best-selling products** and revenue trends.
* The system shall provide **customer purchase behavior analytics**.

# External Interface Requirements

This section outlines how users interact with the system, the necessary hardware/software requirements, and how different system components communicate.

## User Interfaces

The system has **two primary user interfaces**:

* **Customer Interface** – The front-end interface for customers to browse and purchase jewelry.
* **Admin Interface** – The back-end interface for administrators to manage products and orders.

### Customer Interface

**Technology:** ReactJS with Material-UI

**Platform:** Web-based (Desktop & Mobile responsive)

**Navigation:**

Homepage → Product Listing → Product Details → Cart → Checkout → Order Confirmation

Account Dashboard → Order History → Wishlist

**Key UI Elements:**

**Search Bar** – Allows keyword-based product search.

**Filters & Sorting Options** – Enables filtering by category, price, discount, and stock.

**Shopping Cart** – Displays items added, pricing, and total cost.

**Secure Checkout Form** – Captures user details and payment preferences.

**4.1.2 Admin Interface**

**Technology:** ReactJS with Material-UI

**Platform:** Web-based (Restricted to Admins)

**Navigation:**

Admin Dashboard → Product Management → Order Management → Customer Management

**Key UI Elements:**

**Product Management Panel** – Add, update, delete products.

**Order Status Panel** – View and update order progress.

**Sales Analytics Dashboard** – Graphical representation of revenue, top products, and customer behavior.

**4.1.3 User Experience Considerations**

**Mobile-Friendly:** Optimized for both **desktop and mobile screens**.

**Dark Mode Support:** User-preferred UI settings for better accessibility.

**Interactive Feedback:** Visual cues for button clicks, loading spinners, and error messages.

## Hardware Interfaces

The system must run on devices capable of handling **web-based applications**.

| **Component** | **Minimum Requirement** | **Recommended Requirement** |
| --- | --- | --- |
| **Processor** | Intel Core i3 | Intel Core i5+ |
| **RAM** | 4GB | 8GB+ |
| **Storage** | 50GB HDD | 100GB SSD |
| **Internet Speed** | 5 Mbps | 10 Mbps+ |

* The customer interface is optimized for **smartphones, tablets, and desktops**.
* The admin dashboard is best used on **desktop/laptop devices**.

## Software Interfaces

The platform interacts with **various third-party services and APIs**.

| **Component** | **Technology/Dependency** |
| --- | --- |
| **Frontend Framework** | ReactJS, Material-UI |
| **Backend Framework** | Spring Boot (Java) |
| **Database** | MySQL |
| **Authentication** | JWT-based login system |
| **Payment Gateways** | Razorpay, Stripe |
| **Hosting** | AWS Cloud |
| **Email & Notifications** | SMTP-based Email API |
| **Logging & Monitoring** | AWS CloudWatch, ELK Stack |

* The **frontend** communicates with the **backend** via RESTful APIs.
* Payment processing is handled by **third-party payment providers (Razorpay, Stripe)**.
* **Email notifications** are sent via SMTP services.

## Communications Interfaces

The system components communicate using secure **web protocols and APIs**.

| **Interface Type** | **Protocol/Technology Used** |
| --- | --- |
| **Frontend-Backend Communication** | RESTful APIs (HTTPS) |
| **Database Communication** | JDBC (MySQL) |
| **User Authentication** | JWT Token over HTTPS |
| **Payment Processing** | Razorpay API, Stripe API |
| **Email Notifications** | SMTP Protocol |
| **Admin Access Management** | Role-Based Authentication |

### Security Considerations

* All data exchanges between the **frontend and backend must be encrypted** using **HTTPS**.
* **JWT tokens** are used for **user authentication** to prevent unauthorized access.
* **Payment transactions** comply with **PCI-DSS security standards.**

# Other Nonfunctional Requirements

Nonfunctional requirements define system characteristics that impact performance, security, and overall user experience. These requirements ensure the platform meets industry standards, regulatory compliance, and business expectations.

## Performance Requirements

The system must ensure fast response times, efficient resource usage, and scalability.

### Response Time

Product search and filtering should return results within 2 seconds.

Page load time should be under 3 seconds for 90% of users.

Order placement should be processed within 5 seconds.

Admin panel should load dashboards within 2 seconds.

### Scalability

The system should handle up to 10,000 concurrent users without significant performance degradation.

Backend API should support 1,000 requests per second during peak traffic.

The system should dynamically scale resources based on load (AWS Auto-Scaling).

### Availability

The platform should have 99.9% uptime, ensuring minimal downtime for users.

Scheduled maintenance should occur during off-peak hours with prior notification.

## Security Requirements

Security is critical to ensure customer data protection, secure transactions, and prevention of unauthorized access.

### Authentication & Authorization

The system shall use JWT authentication for secure login sessions.

User passwords shall be stored using bcrypt hashing with salted encryption.

Role-based access control (RBAC) shall restrict unauthorized users from accessing admin features.

### Payment Security

The system must comply with PCI-DSS (Payment Card Industry Data Security Standard).

All transactions must be processed through secure HTTPS connections.

Payment information should never be stored in plaintext; instead, tokens must be used.

### Data Protection & Compliance

The system must comply with GDPR (General Data Protection Regulation) to protect user data.

Users should have the option to delete their accounts and personal data upon request.

Two-factor authentication (2FA) should be implemented for admin logins.

## Usability Requirements

Usability focuses on ensuring the platform is user-friendly, accessible, and intuitive.

### User Experience (UX)

The platform must have a clean and modern UI built using Material-UI.

Interactive feedback mechanisms (e.g., loading indicators, error messages, success popups) must be implemented.

Navigation should be intuitive, allowing users to find products and complete purchases easily.

### Accessibility

The platform must follow WCAG 2.1 (Web Content Accessibility Guidelines) to support users with disabilities.

Keyboard navigation support should be available for users who cannot use a mouse.

Color contrast and text-to-speech support should be considered for visually impaired users.

### Mobile Responsiveness

The system should be fully responsive on mobile, tablet, and desktop.

The UI should automatically adjust based on screen resolution.

Touch-friendly interactions should be available for mobile users.

## Maintainability & Support Requirements

Maintainability ensures the system remains easy to update, debug, and extend.

### Code Modularity

The backend must follow microservices architecture to ensure easy updates and maintenance.

The frontend should use component-based architecture (ReactJS) to facilitate code reuse.

### Logging & Monitoring

The system must log all critical events (e.g., login attempts, failed payments, API errors).

Logs should be stored for a minimum of 6 months.

AWS CloudWatch or ELK Stack (Elasticsearch, Logstash, Kibana) should be used for monitoring system performance.

### Backup & Disaster Recovery

The database should have automated backups stored in a secure cloud environment.

In case of a critical failure, the system should restore data within 30 minutes.

## Business Rules

Business rules define policy-based constraints that must be followed.

| Rule | Description |
| --- | --- |
| Return Policy | Customers can request a return within 7 days of delivery. |
| Discount Application | Discounts cannot be stacked (only one discount code per order). |
| Product Availability | Out-of-stock products cannot be added to the cart. |
| Fraud Detection | Orders with unusual behavior (e.g., excessive returns) must be flagged for manual review. |

# Other Requirements

## Regulatory & Legal Compliance

The system must comply with **global and regional e-commerce regulations**, including **data protection, financial transactions, and consumer rights**.

### Data Protection & Privacy Compliance

The platform must adhere to **global data protection laws** to ensure customer privacy.

| **Regulation** | **Description** |
| --- | --- |
| **GDPR (General Data Protection Regulation – EU)** | Users have the right to **delete their personal data** upon request. |
| **CCPA (California Consumer Privacy Act – US)** | Customers can request **what personal data is collected** and **opt-out** of data sharing. |
| **IT Act (India)** | User data must be **securely stored** with **no unauthorized access**. |

### Payment Compliance (PCI-DSS)

The system must **comply with PCI-DSS** (Payment Card Industry Data Security Standard) to **ensure safe transactions**.

**No credit/debit card details should be stored** in the platform.

All transactions must be **processed through Razorpay & Stripe**.

**Fraud prevention mechanisms** (e.g., CAPTCHA, OTP verification) must be in place.

### Consumer Protection Compliance

The system must **clearly display refund policies** and **return procedures**.

Customers must be **notified of successful purchases** via **email & SMS**.

Product descriptions should be **accurate and transparent**.

## Licensing & Open-Source Dependencies

The platform uses **various open-source technologies**. The licensing terms of each must be followed:

| **Technology** | **License Type** | **Compliance Requirements** |
| --- | --- | --- |
| **ReactJS** | MIT License | Must include the original license in distributed code. |
| **Spring Boot** | Apache License 2.0 | Modifications must retain copyright notices. |
| **MySQL** | GPL License | Must comply with GPL-based distribution restrictions. |
| **Material-UI** | MIT License | Free to use with proper attribution. |

**All third-party software licenses** must be **documented in the admin panel**.

**Attribution for open-source libraries** must be **included in the source code repository**.

## Environmental & Operational Constraints

### Server & Hosting Constraints

The system should be **deployed on AWS Cloud** with **auto-scaling enabled**.

The database should be **backed up daily** to prevent data loss.

Downtime for **maintenance should be scheduled during non-peak hours**.

### Performance Constraints

The system should **handle 1,000 API requests per second** without failure.

The frontend should be **optimized** to load pages in **less than 3 seconds**.

### Compatibility Constraints

The platform should work **seamlessly** on the following **browsers**:

Google Chrome (latest)

Mozilla Firefox (latest)

Safari (latest)

Microsoft Edge (latest)

The system must be **mobile-responsive**, adapting to **different screen sizes**.

### Localization & Language Support

The platform should **support multiple languages** for international customers.

All **prices must be displayed in local currency**, based on the user's location.

## Future Enhancements

While **not part of the initial release**, the following features may be implemented in future updates:

* **AI-Powered Jewelry Recommendation System**

Uses **machine learning** to suggest **personalized jewelry items** based on browsing history.

* **Augmented Reality (AR) Try-On**

Customers can **virtually try on jewelry** using their smartphone camera.

* **Loyalty & Rewards Program**

Customers earn **points for purchases** and redeem them for discounts.

* **Integration with Social Media**

Customers can **share products** on Instagram, Facebook, and WhatsApp.

* **Voice Search & Chatbot Support**
  + Allows customers to **search for products using voice commands**.
  + AI-powered chatbot for **real-time customer support**.

Appendix A: Glossary

This glossary defines key terms and abbreviations used throughout the SRS document.

| **Term** | **Definition** |
| --- | --- |
| **API** | Application Programming Interface - allows frontend & backend communication. |
| **JWT** | JSON Web Token - used for secure authentication. |
| **CRUD** | Create, Read, Update, Delete - standard operations for managing database records. |
| **PCI-DSS** | Payment Card Industry Data Security Standard - security rules for online transactions. |
| **GDPR** | General Data Protection Regulation - governs user data privacy and protection. |
| **2FA** | Two-Factor Authentication - enhances security with OTP-based login verification. |
| **AR (Augmented Reality)** | A feature that allows virtual try-ons for jewelry products. |
| **Microservices Architecture** | A system design approach where the backend is divided into independent services. |

Appendix B: Analysis Models

**7. Analysis Models & Diagrams**

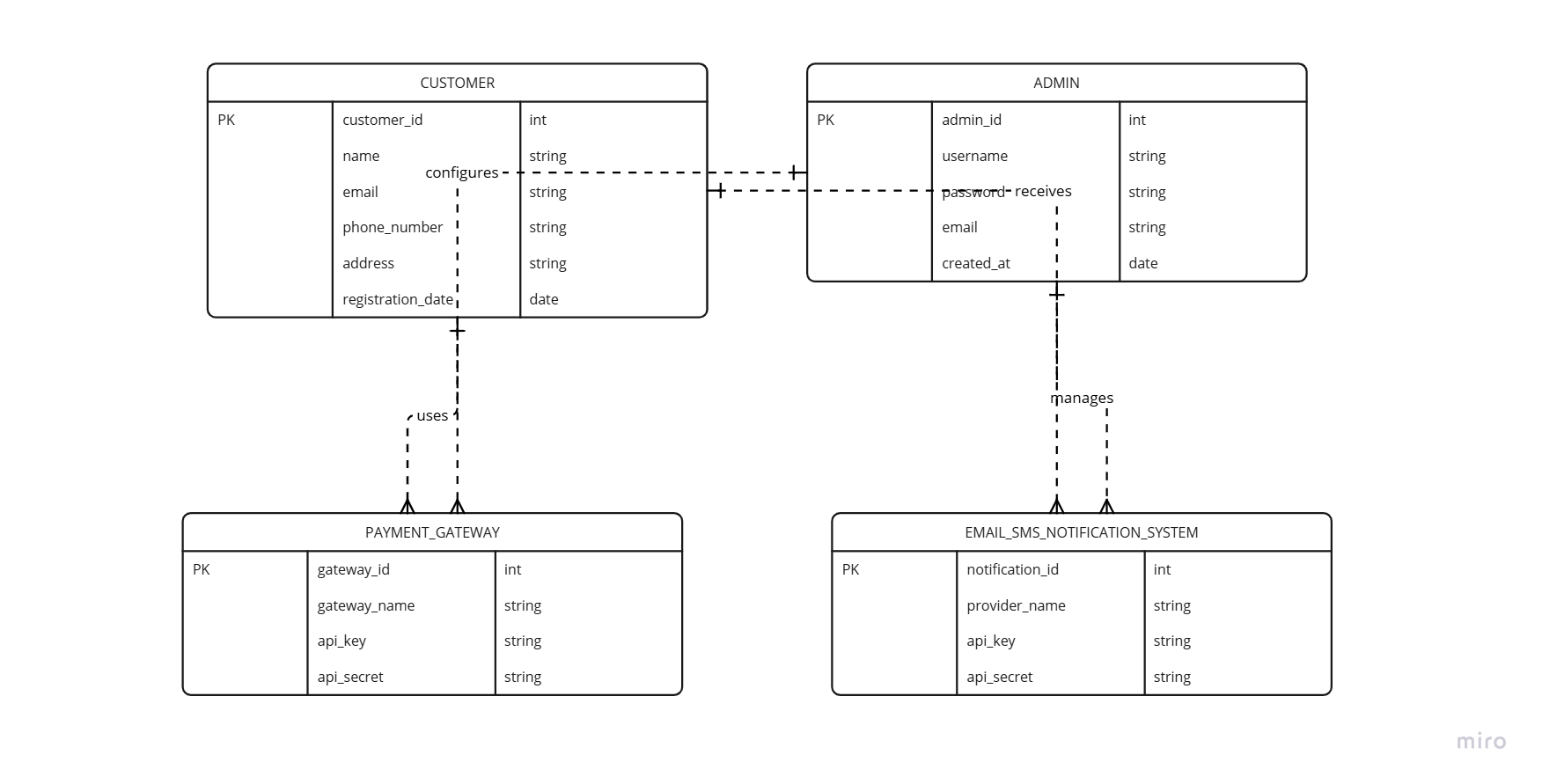
This section provides **visual representations** of the system architecture.

**7.2.1 Use Case Diagram**

A **Use Case Diagram** represents the **main interactions** between users and the system.

**Actors:**

* + - **Customer** (Primary)
    - **Admin** (Primary)
    - **Payment Gateway (Stripe/Razorpay)** (External)
    - **Email/SMS Notification System** (External)

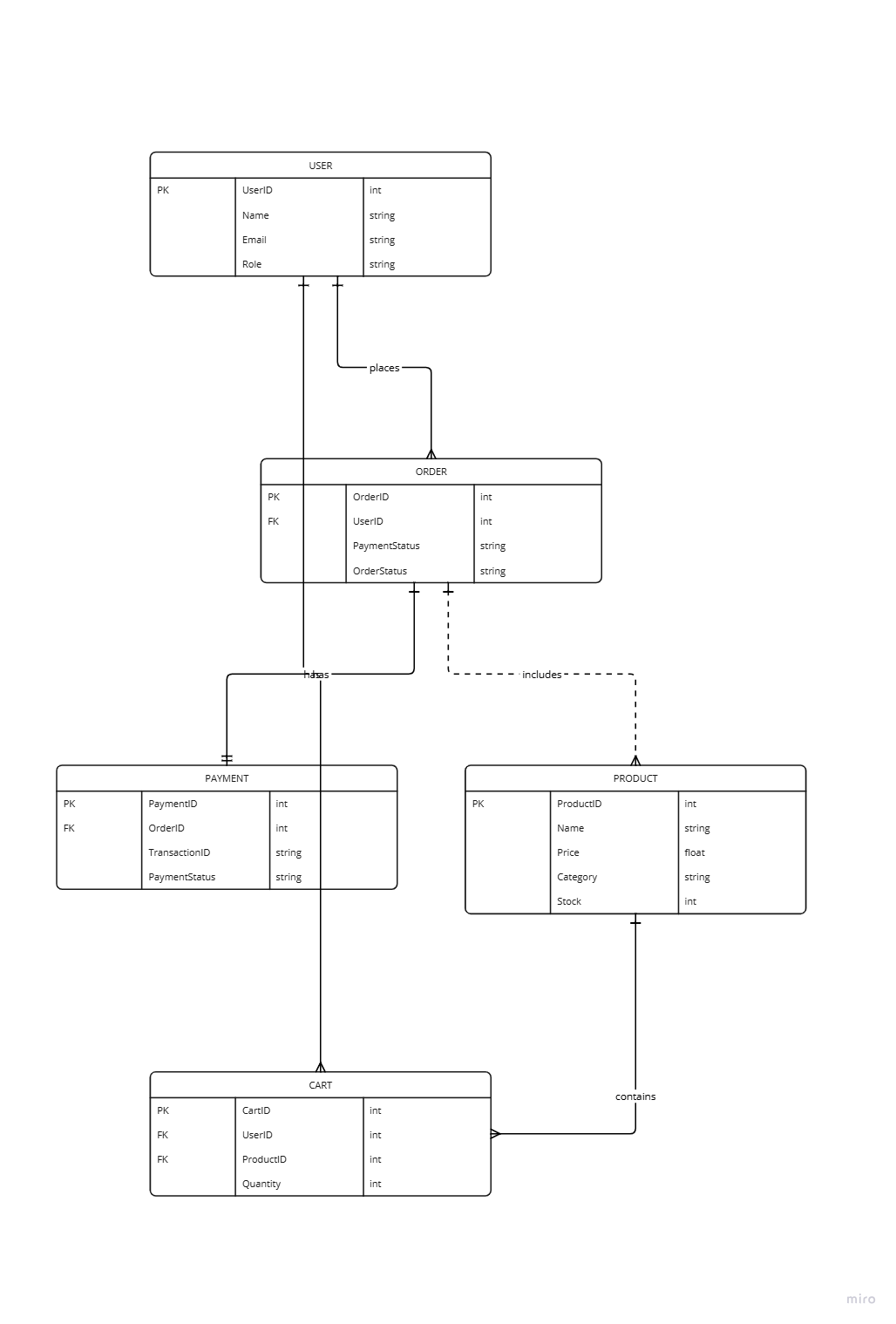


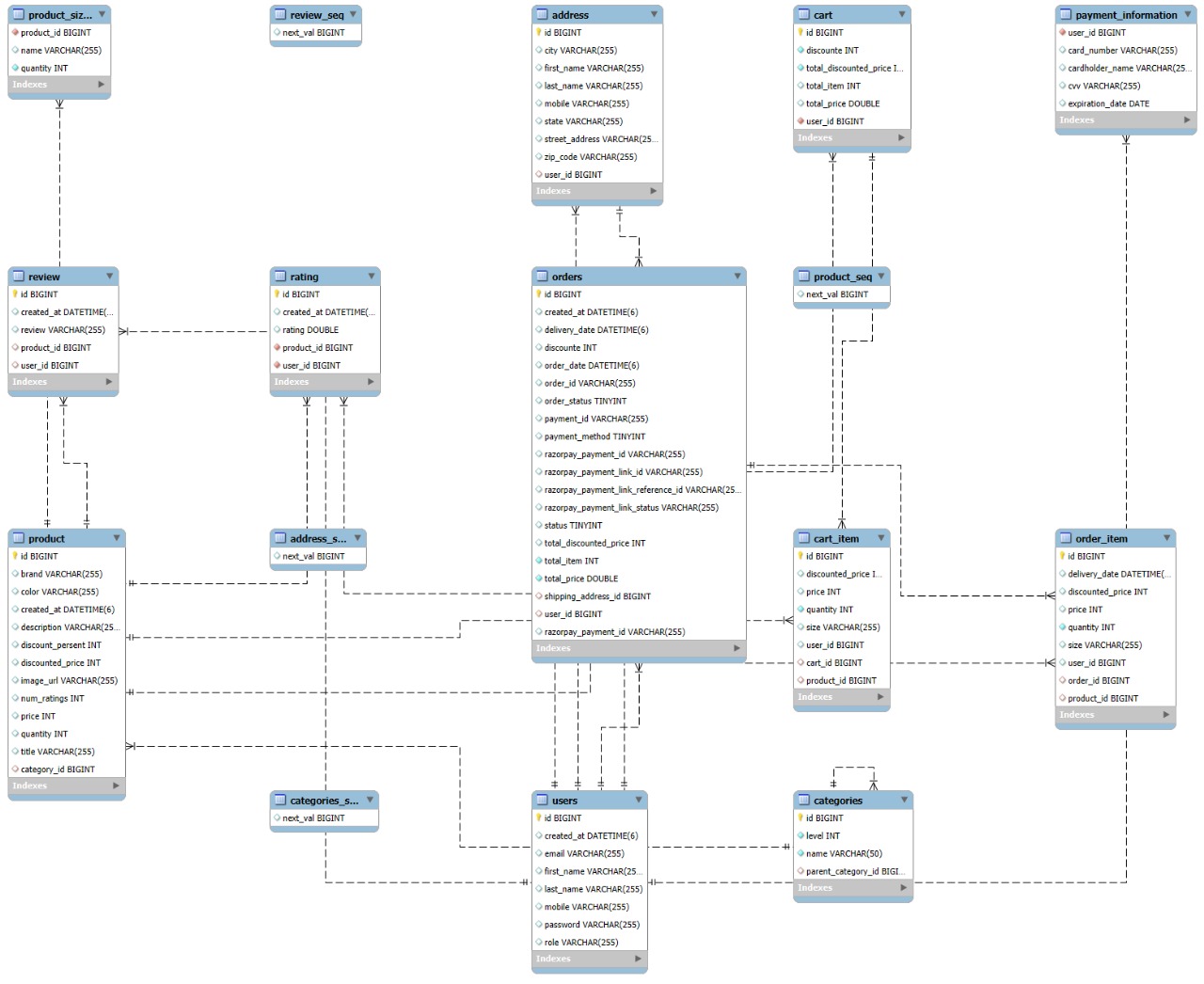
***7.2.2* Entity-Relationship Diagram (ERD)**

An **ER Diagram** shows **how different database entities relate to each other**.

**Key Entities:**

* + **User** (UserID, Name, Email, Role)
  + **Product** (ProductID, Name, Price, Category, Stock)
  + **Order** (OrderID, UserID, PaymentStatus, OrderStatus)
  + **Cart** (CartID, UserID, ProductID, Quantity)
  + **Payment** (PaymentID, OrderID, TransactionID, PaymentStatus)

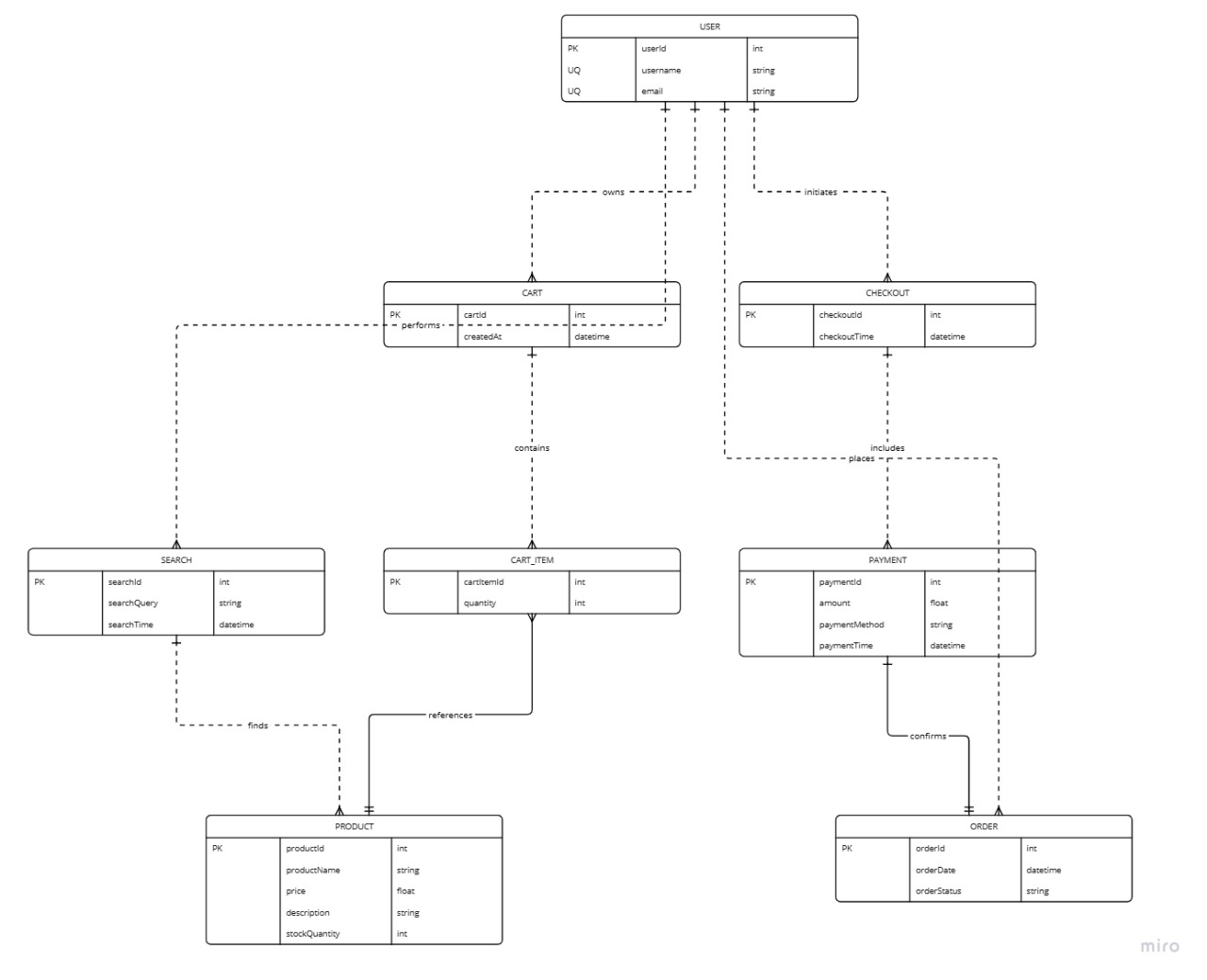




**7.2.3 Data Flow Diagram (DFD - Level 1)**

A **DFD** shows how **data moves through the system**.

📌 **Flow Example:**



**7.3 Assumptions and Dependencies**

This section lists assumptions made during system design and key dependencies.

**7.3.1 Assumptions**

* + - Customers have access to **stable internet connections**.
    - Payment gateways (**Razorpay, Stripe**) are **always operational**.
    - Users **trust the platform** and will not engage in fraudulent activities.
    - The system will handle **peak traffic efficiently**.

**7.3.2 Dependencies**

**Third-Party APIs**:

* + 1. **Payment API**: Stripe & Razorpay.
    2. **Email API**: SMTP or AWS SES.
    3. **SMS Notifications**: Twilio or a similar service.

**Hosting Services**:

* + 1. Backend hosted on **AWS EC2**.

Database hosted on **AWS RDS (MySQL)**.