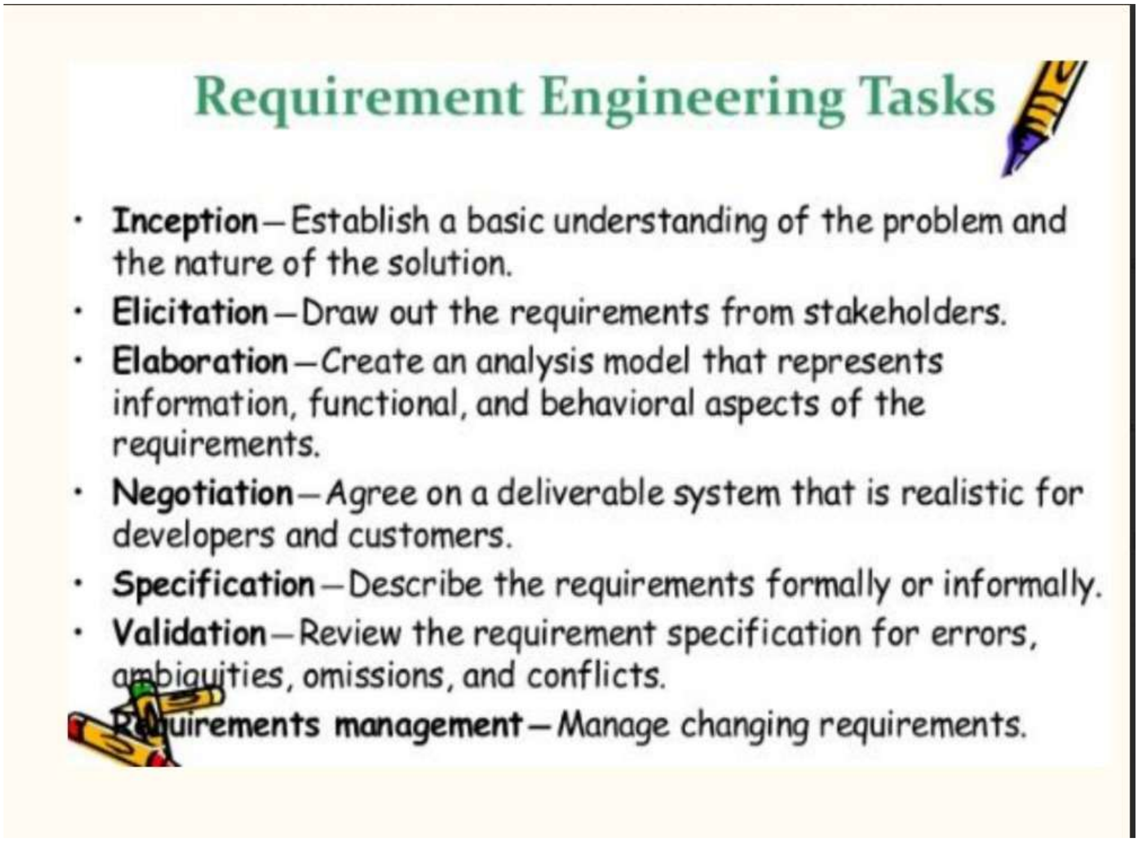
QB MODULE 3

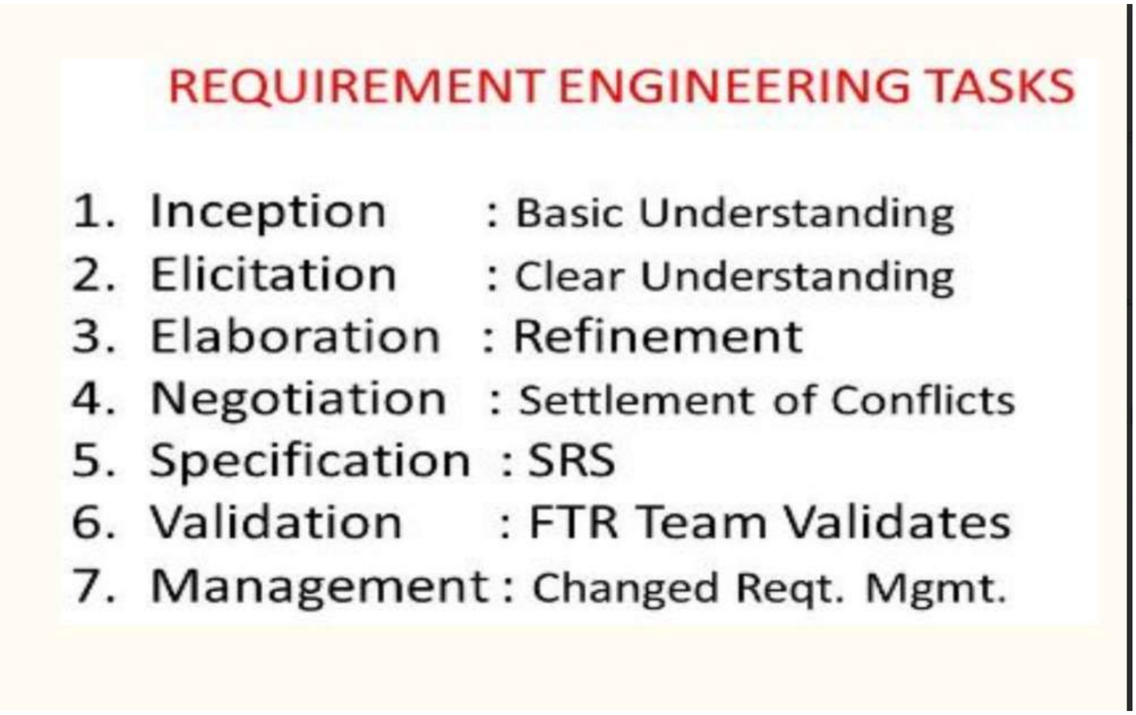
Q1 Requirement Engineering is the process of defining, documenting and maintaining the requirements. It is a process of gathering and defining service provided by the system. Requirements Engineering

Process consists of the following main activities.



Requirement Engineering Tasks

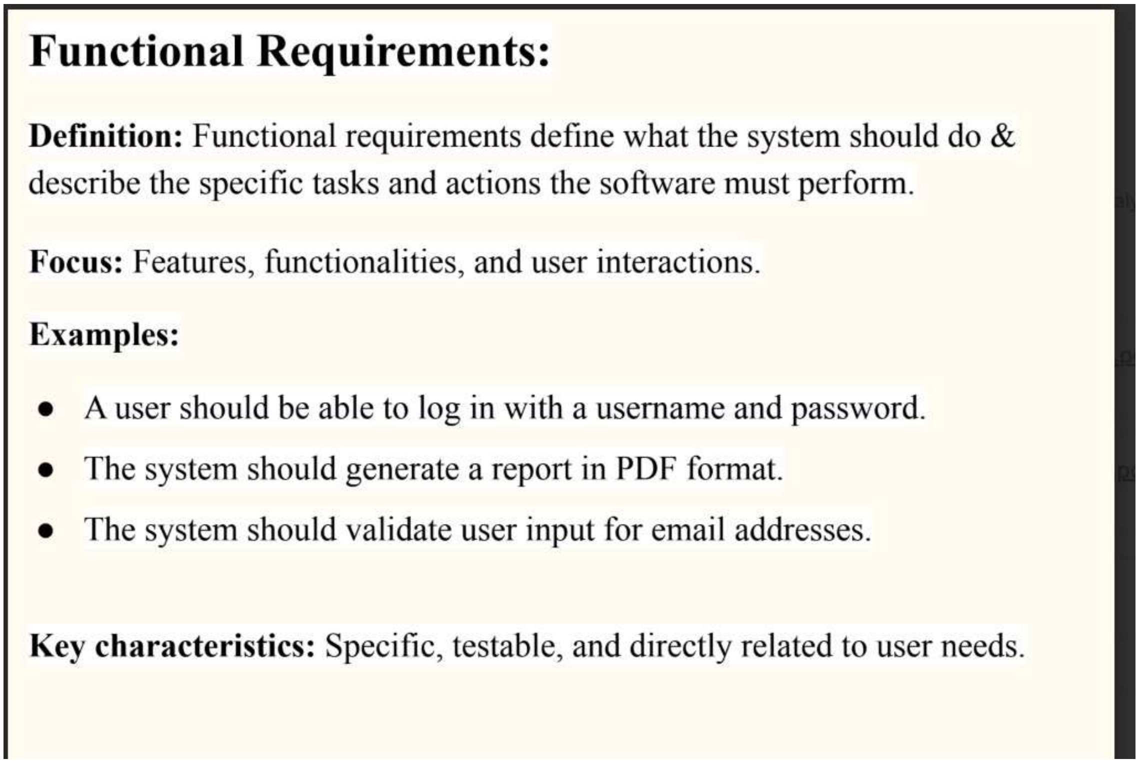
- **Inception**—Establish a basic understanding of the problem and the nature of the solution.
- **Elicitation**—Draw out the requirements from stakeholders.
- **Elaboration**—Create an analysis model that represents information, functional, and behavioral aspects of the requirements.
- **Negotiation**—Agree on a deliverable system that is realistic for developers and customers.
- **Specification**—Describe the requirements formally or informally.
- **Validation**—Review the requirement specification for errors, ambiguities, omissions, and conflicts.
- **Requirements management**—Manage changing requirements.



REQUIREMENT ENGINEERING TASKS

1. Inception : Basic Understanding
2. Elicitation : Clear Understanding
3. Elaboration : Refinement
4. Negotiation : Settlement of Conflicts
5. Specification : SRS
6. Validation : FTR Team Validates
7. Management : Changed Reqt. Mgmt.

Q2>Explain functional and non-functional requirement



Functional Requirements:

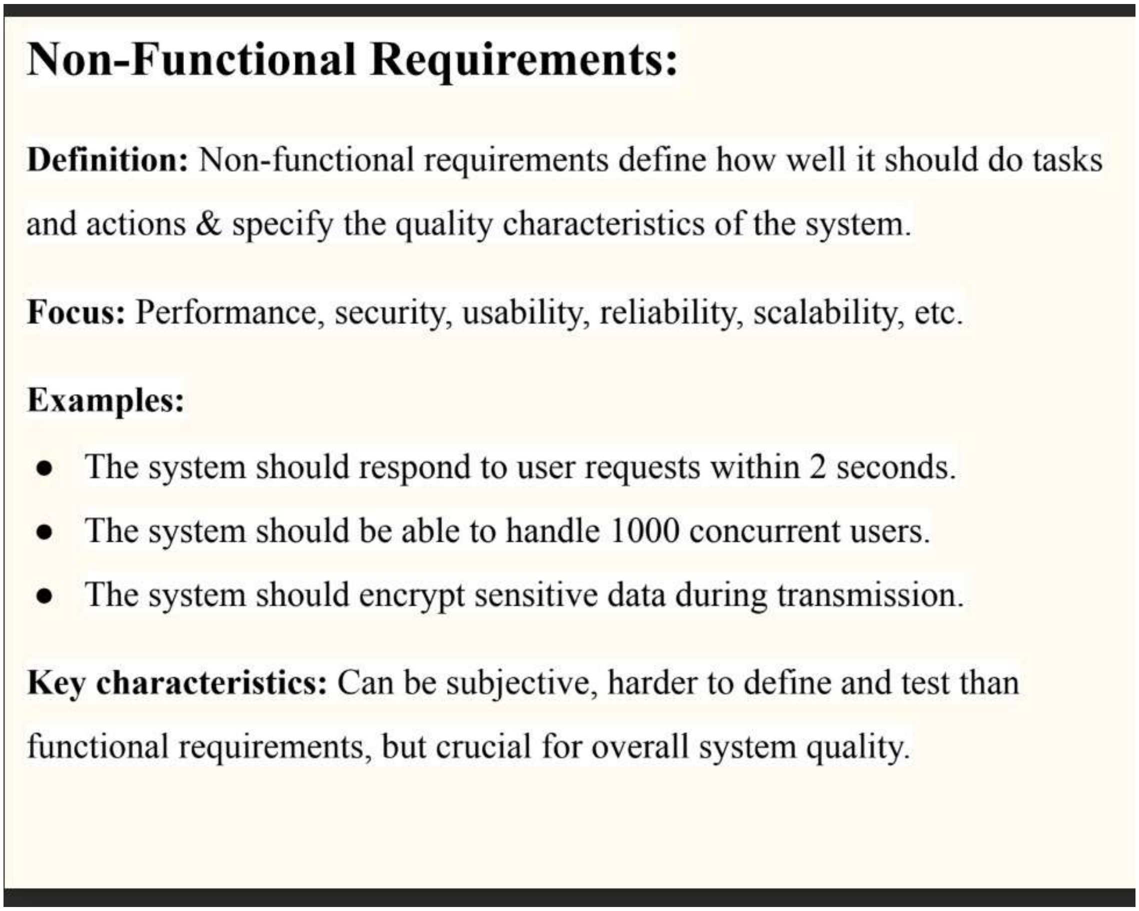
Definition: Functional requirements define what the system should do & describe the specific tasks and actions the software must perform.

Focus: Features, functionalities, and user interactions.

Examples:

- A user should be able to log in with a username and password.
- The system should generate a report in PDF format.
- The system should validate user input for email addresses.

Key characteristics: Specific, testable, and directly related to user needs.



Non-Functional Requirements:

Definition: Non-functional requirements define how well it should do tasks and actions & specify the quality characteristics of the system.

Focus: Performance, security, usability, reliability, scalability, etc.

Examples:

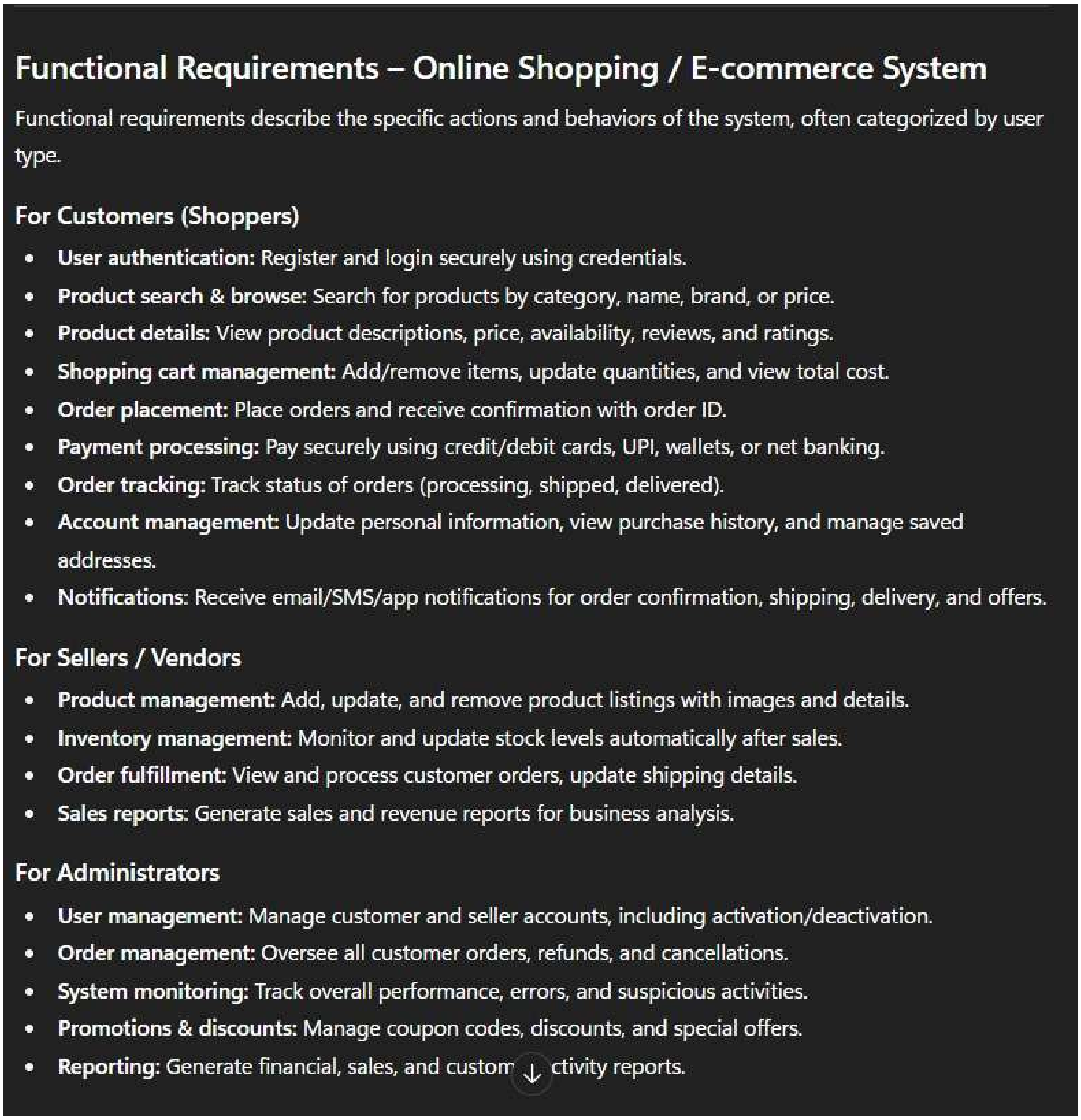
- The system should respond to user requests within 2 seconds.
- The system should be able to handle 1000 concurrent users.
- The system should encrypt sensitive data during transmission.

Key characteristics: Can be subjective, harder to define and test than functional requirements, but crucial for overall system quality.

Q3 Similar to Q2

Q4 Analyze the functional and non-functional

requirement Library management system



Functional Requirements – Online Shopping / E-commerce System

Functional requirements describe the specific actions and behaviors of the system, often categorized by user type.

For Customers (Shoppers)

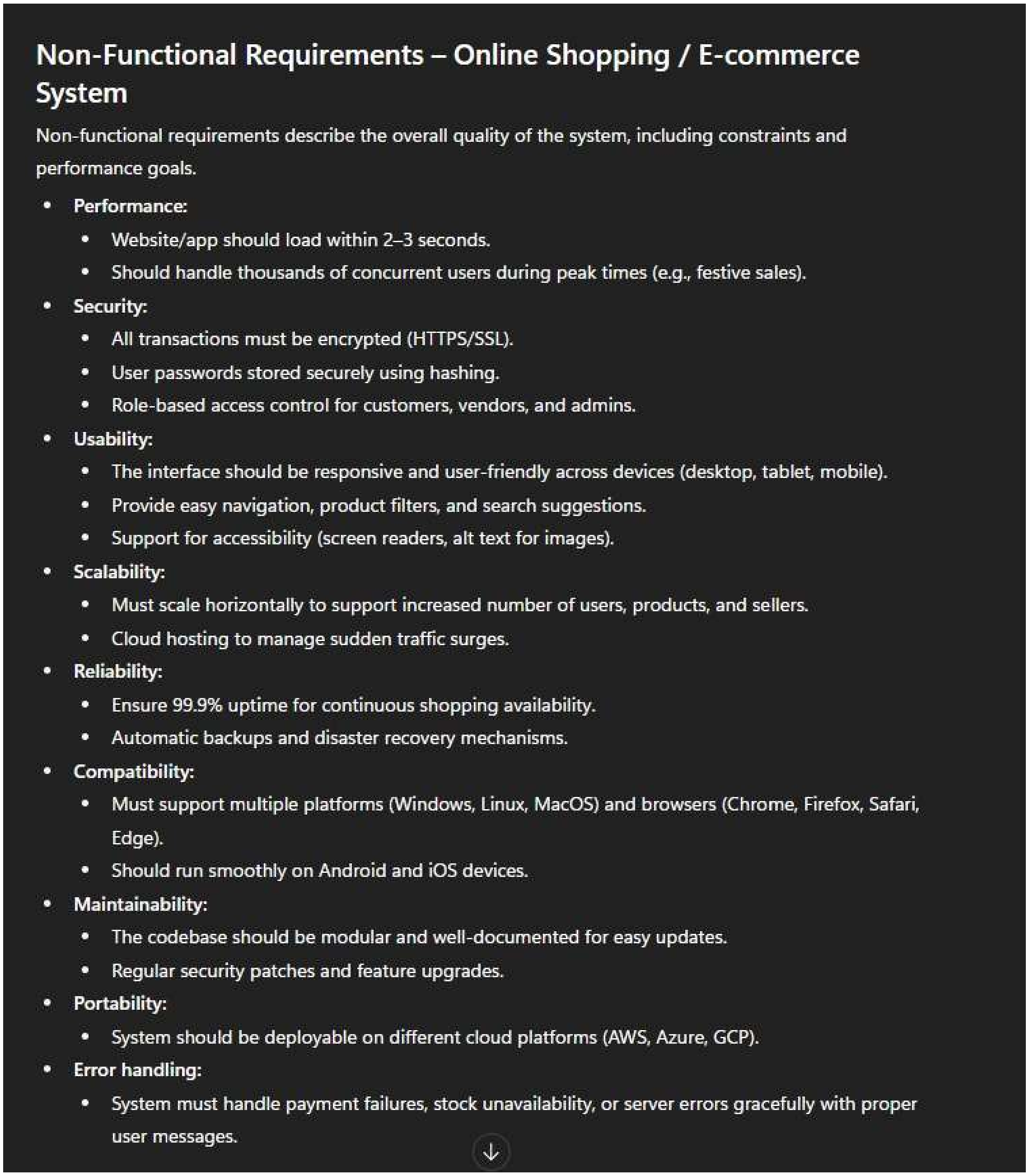
- **User authentication:** Register and login securely using credentials.
- **Product search & browse:** Search for products by category, name, brand, or price.
- **Product details:** View product descriptions, price, availability, reviews, and ratings.
- **Shopping cart management:** Add/remove items, update quantities, and view total cost.
- **Order placement:** Place orders and receive confirmation with order ID.
- **Payment processing:** Pay securely using credit/debit cards, UPI, wallets, or net banking.
- **Order tracking:** Track status of orders (processing, shipped, delivered).
- **Account management:** Update personal information, view purchase history, and manage saved addresses.
- **Notifications:** Receive email/SMS/app notifications for order confirmation, shipping, delivery, and offers.

For Sellers / Vendors

- **Product management:** Add, update, and remove product listings with images and details.
- **Inventory management:** Monitor and update stock levels automatically after sales.
- **Order fulfillment:** View and process customer orders, update shipping details.
- **Sales reports:** Generate sales and revenue reports for business analysis.

For Administrators

- **User management:** Manage customer and seller accounts, including activation/deactivation.
- **Order management:** Oversee all customer orders, refunds, and cancellations.
- **System monitoring:** Track overall performance, errors, and suspicious activities.
- **Promotions & discounts:** Manage coupon codes, discounts, and special offers.
- **Reporting:** Generate financial, sales, and custom ctivity reports.

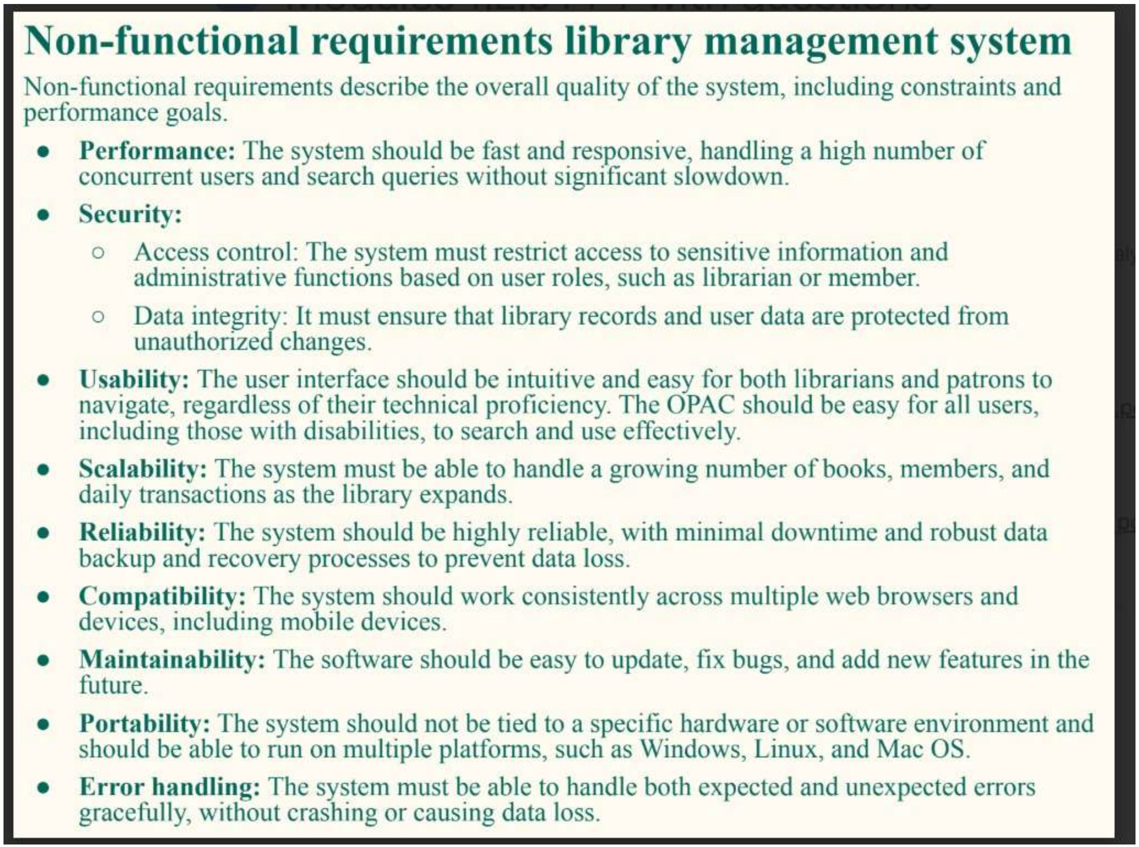


Non-Functional Requirements – Online Shopping / E-commerce System

Non-functional requirements describe the overall quality of the system, including constraints and performance goals.

- **Performance:**
  - Website/app should load within 2–3 seconds.
  - Should handle thousands of concurrent users during peak times (e.g., festive sales).
- **Security:**
  - All transactions must be encrypted (HTTPS/SSL).
  - User passwords stored securely using hashing.
  - Role-based access control for customers, vendors, and admins.
- **Usability:**
  - The interface should be responsive and user-friendly across devices (desktop, tablet, mobile).
  - Provide easy navigation, product filters, and search suggestions.
  - Support for accessibility (screen readers, alt text for images).
- **Scalability:**
  - Must scale horizontally to support increased number of users, products, and sellers.
  - Cloud hosting to manage sudden traffic surges.
- **Reliability:**
  - Ensure 99.9% uptime for continuous shopping availability.
  - Automatic backups and disaster recovery mechanisms.
- **Compatibility:**
  - Must support multiple platforms (Windows, Linux, MacOS) and browsers (Chrome, Firefox, Safari, Edge).
  - Should run smoothly on Android and iOS devices.
- **Maintainability:**
  - The codebase should be modular and well-documented for easy updates.
  - Regular security patches and feature upgrades.
- **Portability:**
  - System should be deployable on different cloud platforms (AWS, Azure, GCP).
- **Error handling:**
  - System must handle payment failures, stock unavailability, or server errors gracefully with proper user messages.

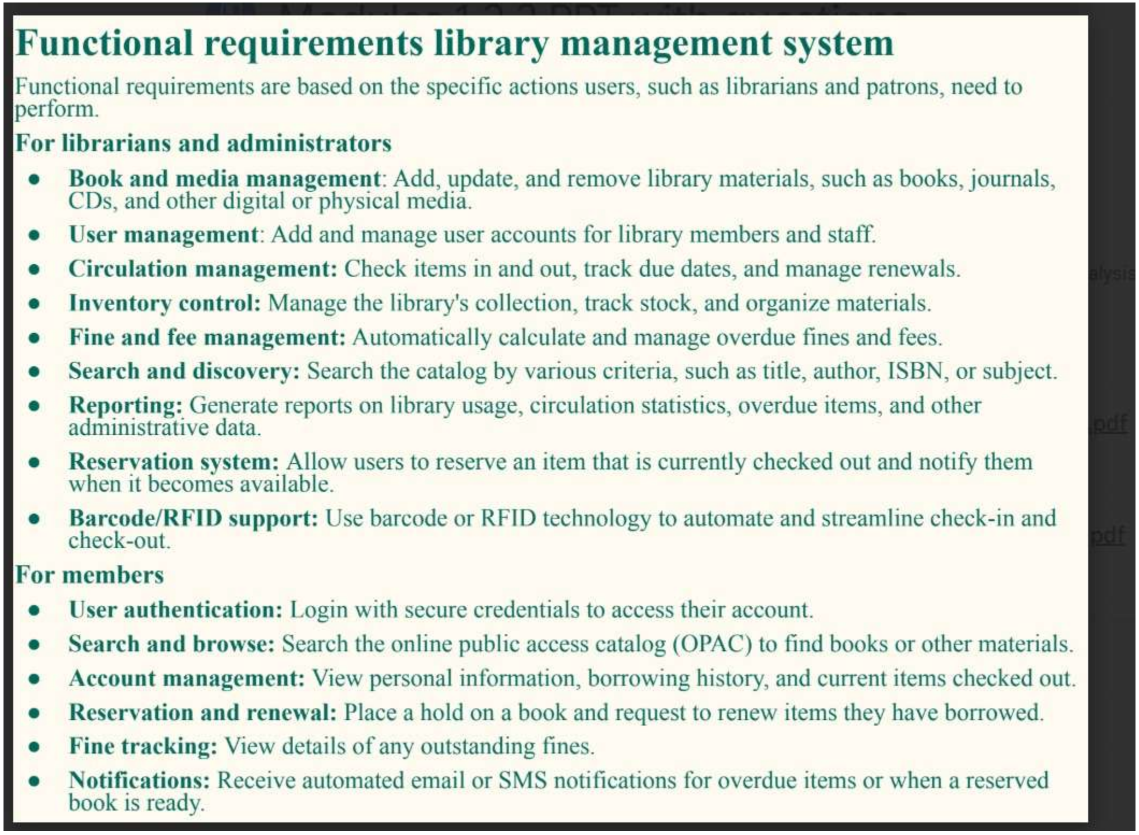
Q5 Analyze the functional and non-functional requirement Library management system



Non-functional requirements library management system

Non-functional requirements describe the overall quality of the system, including constraints and performance goals.

- **Performance:** The system should be fast and responsive, handling a high number of concurrent users and search queries without significant slowdown.
- **Security:**
  - Access control: The system must restrict access to sensitive information and administrative functions based on user roles, such as librarian or member.
  - Data integrity: It must ensure that library records and user data are protected from unauthorized changes.
- **Usability:** The user interface should be intuitive and easy for both librarians and patrons to navigate, regardless of their technical proficiency. The OPAC should be easy for all users, including those with disabilities, to search and use effectively.
- **Scalability:** The system must be able to handle a growing number of books, members, and daily transactions as the library expands.
- **Reliability:** The system should be highly reliable, with minimal downtime and robust data backup and recovery processes to prevent data loss.
- **Compatibility:** The system should work consistently across multiple web browsers and devices, including mobile devices.
- **Maintainability:** The software should be easy to update, fix bugs, and add new features in the future.
- **Portability:** The system should not be tied to a specific hardware or software environment and should be able to run on multiple platforms, such as Windows, Linux, and Mac OS.
- **Error handling:** The system must be able to handle both expected and unexpected errors gracefully, without crashing or causing data loss.



Functional requirements library management system

Functional requirements are based on the specific actions users, such as librarians and patrons, need to perform.

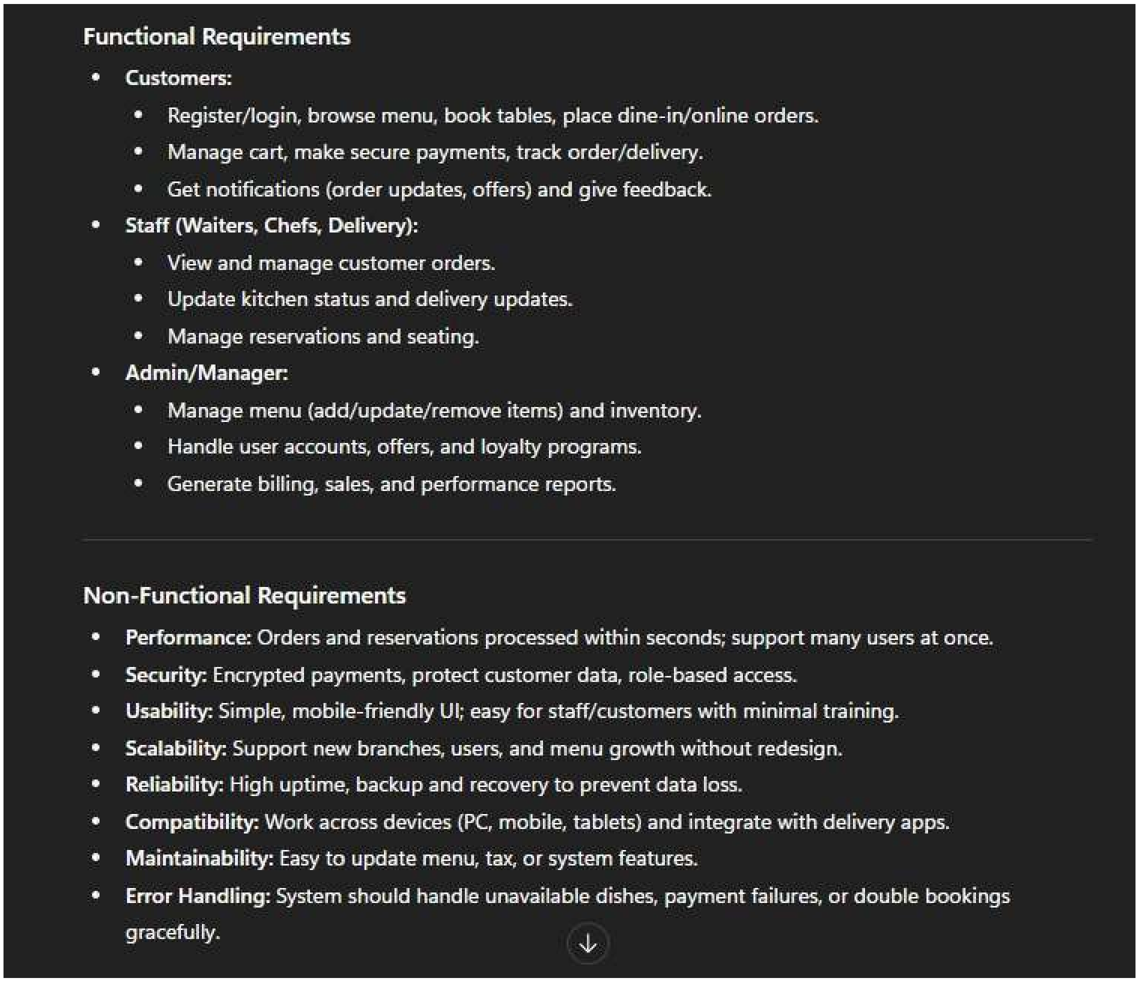
For librarians and administrators

- **Book and media management**: Add, update, and remove library materials, such as books, journals, CDs, and other digital or physical media.
- **User management**: Add and manage user accounts for library members and staff.
- **Circulation management:** Check items in and out, track due dates, and manage renewals.
- **Inventory control:** Manage the library's collection, track stock, and organize materials.
- **Fine and fee management:** Automatically calculate and manage overdue fines and fees.
- **Search and discovery:** Search the catalog by various criteria, such as title, author, ISBN, or subject.
- **Reporting:** Generate reports on library usage, circulation statistics, overdue items, and other administrative data.
- **Reservation system:** Allow users to reserve an item that is currently checked out and notify them when it becomes available.
- **Barcode/RFID support:** Use barcode or RFID technology to automate and streamline check-in and check-out.

For members

- **User authentication:** Login with secure credentials to access their account.
- **Search and browse:** Search the online public access catalog (OPAC) to find books or other materials.
- **Account management:** View personal information, borrowing history, and current items checked out.
- **Reservation and renewal:** Place a hold on a book and request to renew items they have borrowed.
- **Fine tracking:** View details of any outstanding fines.
- **Notifications:** Receive automated email or SMS notifications for overdue items or when a reserved book is ready.

Q6 Analyze the functional and non-functional requirement Restaurant management system



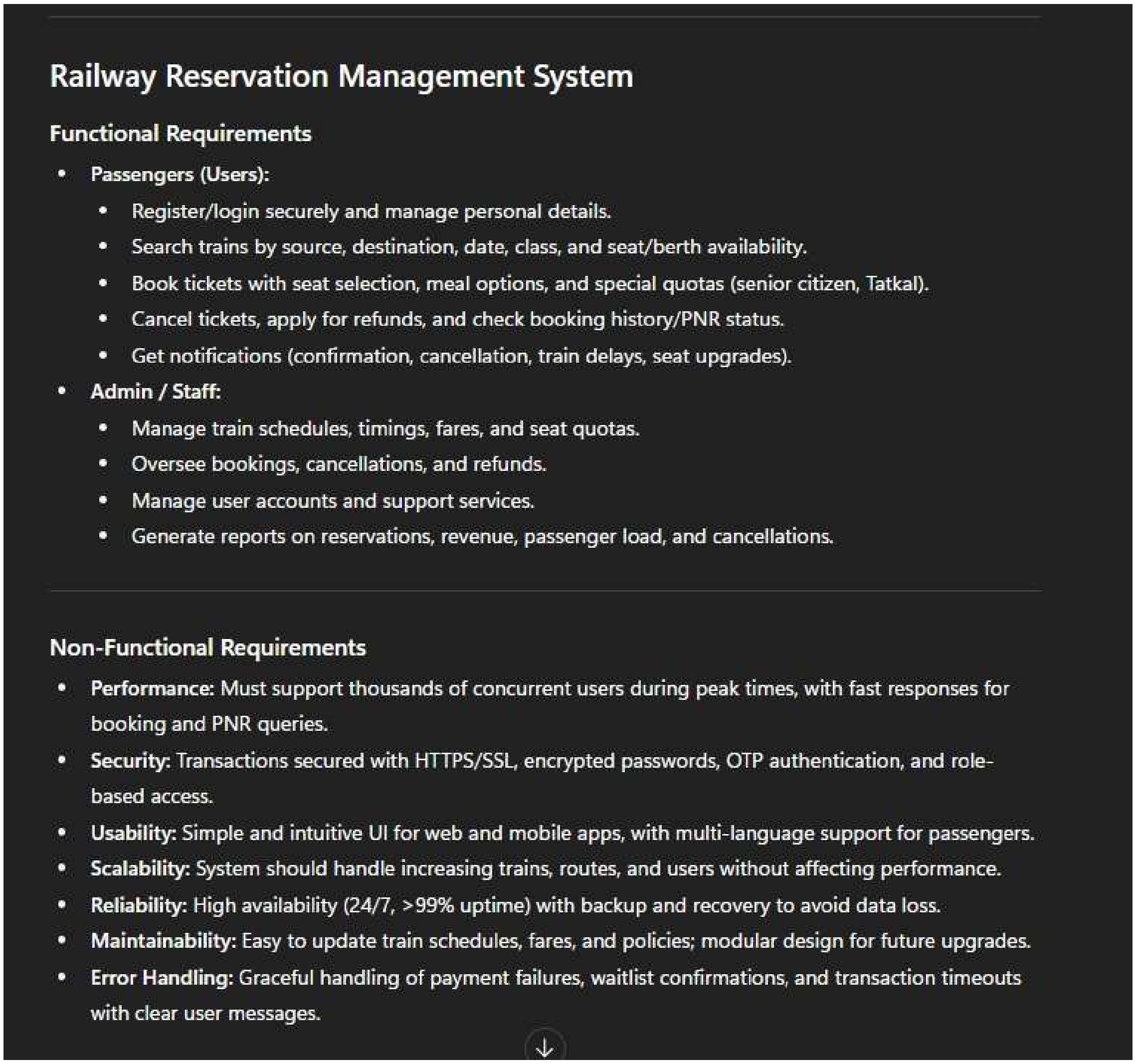
Functional Requirements

- **Customers:**
  - Register/login, browse menu, book tables, place dine-in/online orders.
  - Manage cart, make secure payments, track order/delivery.
  - Get notifications (order updates, offers) and give feedback.
- **Staff (Waiters, Chefs, Delivery):**
  - View and manage customer orders.
  - Update kitchen status and delivery updates.
  - Manage reservations and seating.
- **Admin/Manager:**
  - Manage menu (add/update/remove items) and inventory.
  - Handle user accounts, offers, and loyalty programs.
  - Generate billing, sales, and performance reports.

Non-Functional Requirements

- **Performance:** Orders and reservations processed within seconds; support many users at once.
- **Security:** Encrypted payments, protect customer data, role-based access.
- **Usability:** Simple, mobile-friendly UI; easy for staff/customers with minimal training.
- **Scalability:** Support new branches, users, and menu growth without redesign.
- **Reliability:** High uptime, backup and recovery to prevent data loss.
- **Compatibility:** Work across devices (PC, mobile, tablets) and integrate with delivery apps.
- **Maintainability:** Easy to update menu, tax, or system features.
- **Error Handling:** System should handle unavailable dishes, payment failures, or double bookings gracefully.

Q7 Analyze the functional and non-functional requirement Railway reservation management system



Railway Reservation Management System

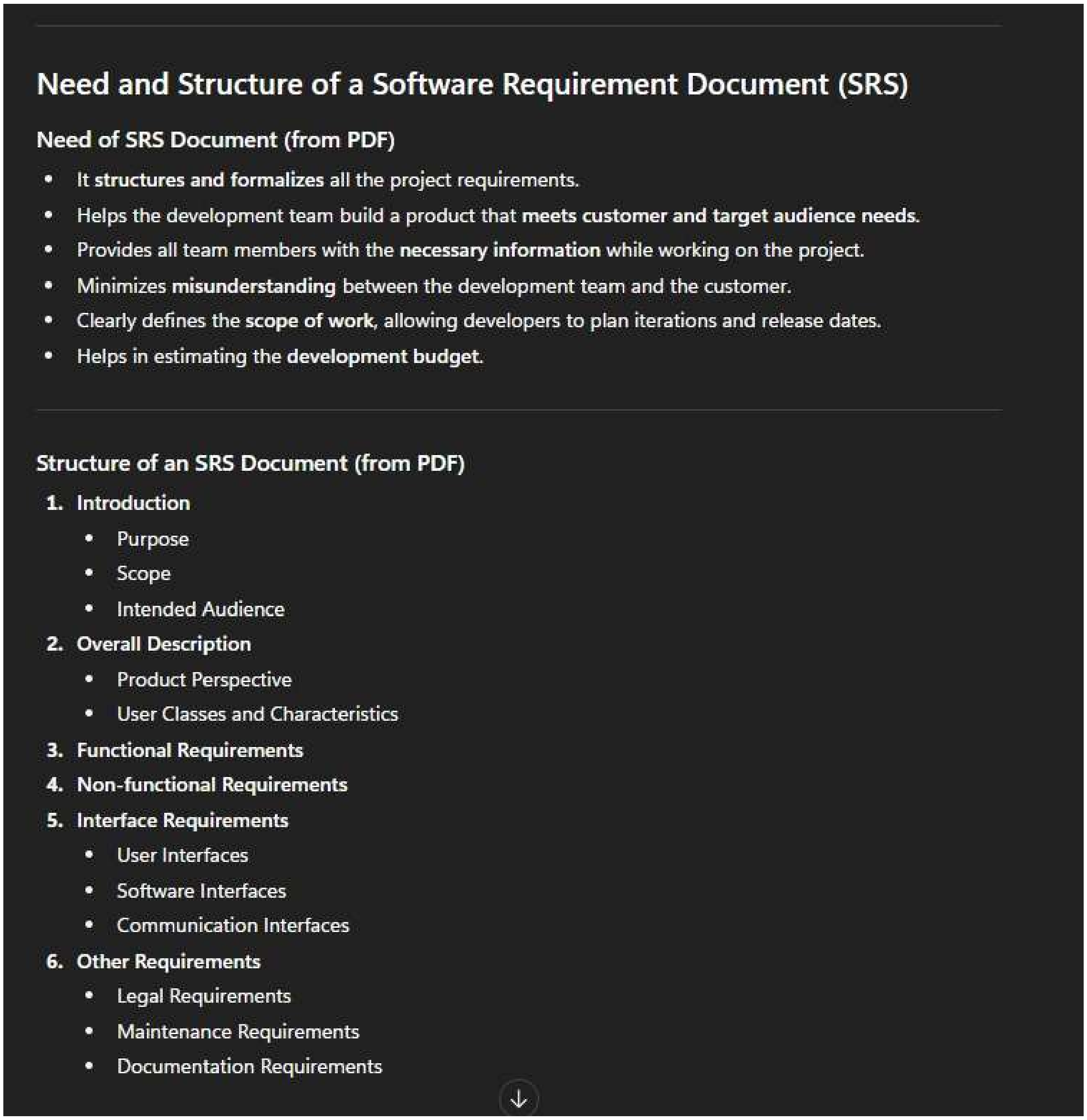
Functional Requirements

- **Passengers (Users):**
  - Register/login securely and manage personal details.
  - Search trains by source, destination, date, class, and seat/berth availability.
  - Book tickets with seat selection, meal options, and special quotas (senior citizen, Tatkal).
  - Cancel tickets, apply for refunds, and check booking history/PNR status.
  - Get notifications (confirmation, cancellation, train delays, seat upgrades).
- **Admin / Staff:**
  - Manage train schedules, timings, fares, and seat quotas.
  - Oversee bookings, cancellations, and refunds.
  - Manage user accounts and support services.
  - Generate reports on reservations, revenue, passenger load, and cancellations.

Non-Functional Requirements

- **Performance:** Must support thousands of concurrent users during peak times, with fast responses for booking and PNR queries.
- **Security:** Transactions secured with HTTPS/SSL, encrypted passwords, OTP authentication, and role-based access.
- **Usability:** Simple and intuitive UI for web and mobile apps, with multi-language support for passengers.
- **Scalability:** System should handle increasing trains, routes, and users without affecting performance.
- **Reliability:** High availability (24/7, >99% uptime) with backup and recovery to avoid data loss.
- **Maintainability:** Easy to update train schedules, fares, and policies; modular design for future upgrades.
- **Error Handling:** Graceful handling of payment failures, waitlist confirmations, and transaction timeouts with clear user messages.

Q8 Explain in detail about the need and structure of a software requirement document.



Need and Structure of a Software Requirement Document (SRS)

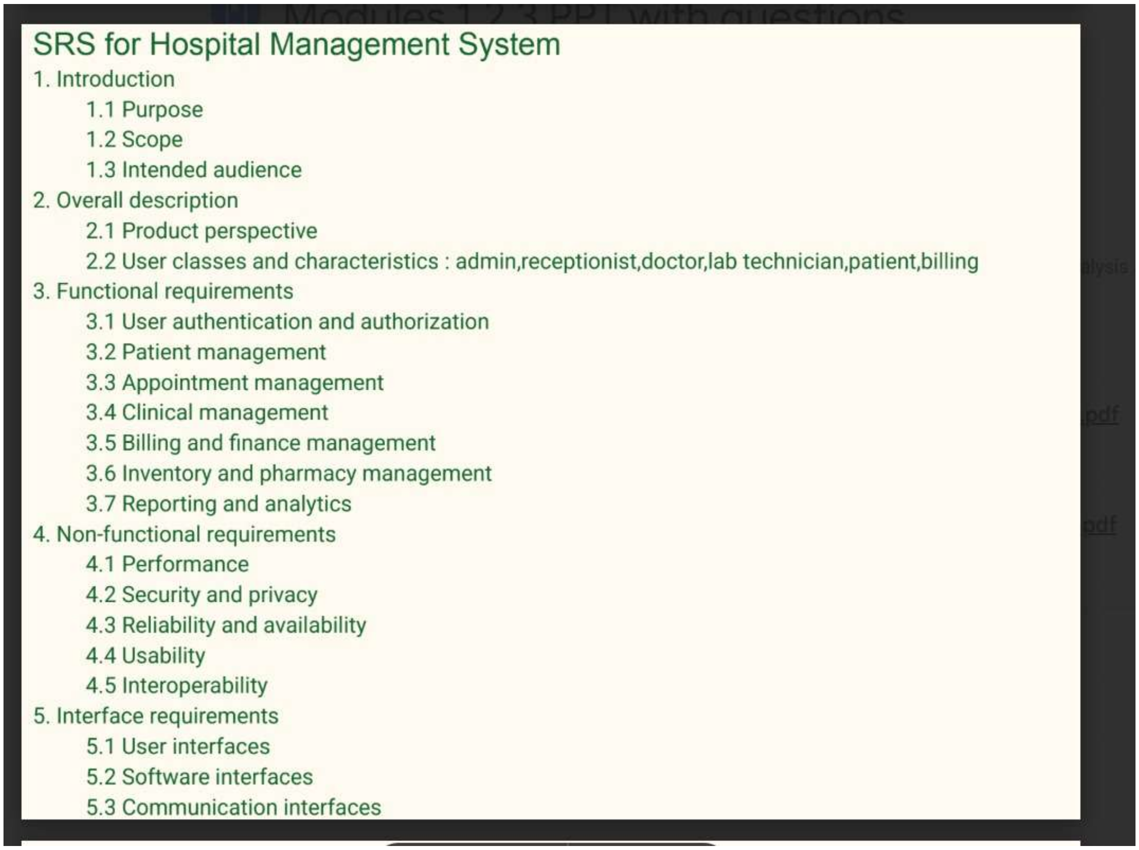
Need of SRS Document (from PDF)

- It **structures and formalizes** all the project requirements.
- Helps the development team build a product that **meets customer and target audience needs.**
- Provides all team members with the **necessary information** while working on the project.
- Minimizes **misunderstanding** between the development team and the customer.
- Clearly defines the **scope of work**, allowing developers to plan iterations and release dates.
- Helps in estimating the **development budget**.

Structure of an SRS Document (from PDF)

1. **Introduction**
   - Purpose
   - Scope
   - Intended Audience
2. **Overall Description**
   - Product Perspective
   - User Classes and Characteristics
3. **Functional Requirements**
4. **Non-functional Requirements**
5. **Interface Requirements**
   - User Interfaces
   - Software Interfaces
   - Communication Interfaces
6. **Other Requirements**
   - Legal Requirements
   - Maintenance Requirements
   - Documentation Requirements

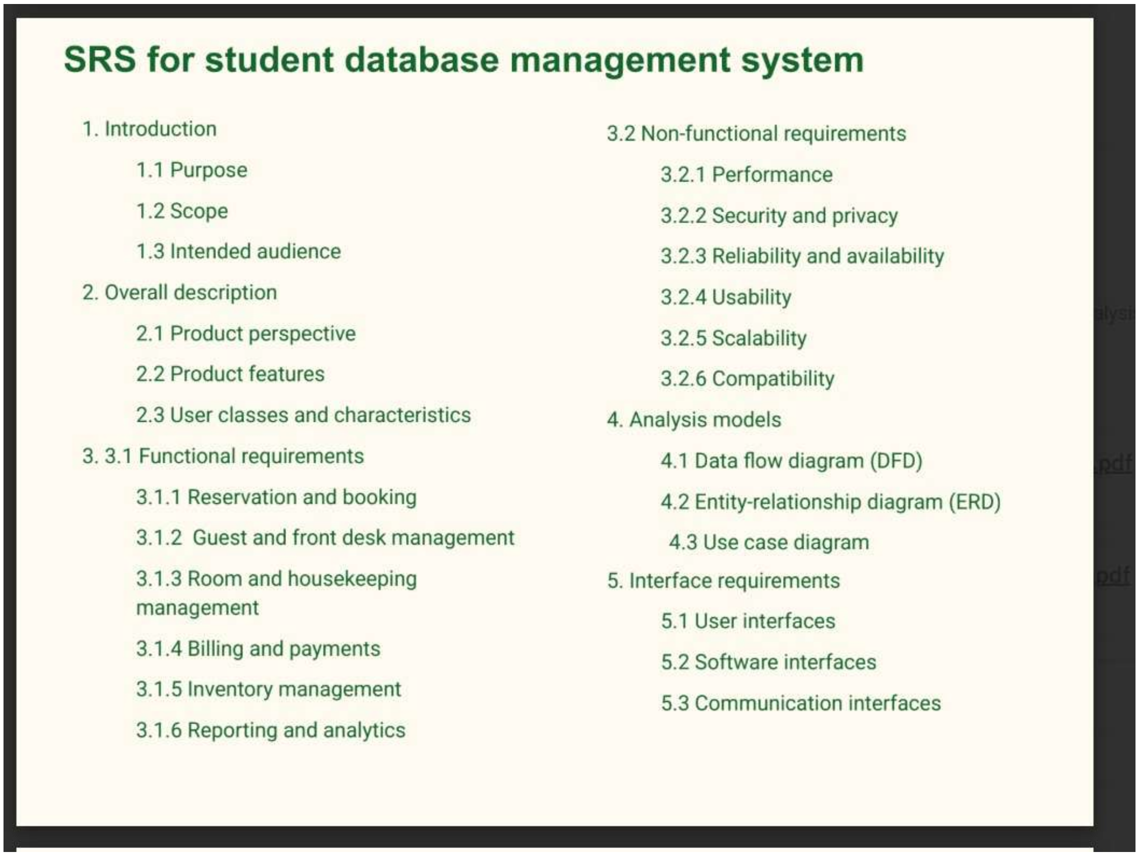
Q9



SRS for Hospital Management System

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Intended audience
2. Overall description
    - 2.1 Product perspective
    - 2.2 User classes and characteristics : admin,receptionist,doctor,lab technician,patient,billing
3. Functional requirements
    - 3.1 User authentication and authorization
    - 3.2 Patient management
    - 3.3 Appointment management
    - 3.4 Clinical management
    - 3.5 Billing and finance management
    - 3.6 Inventory and pharmacy management
    - 3.7 Reporting and analytics
4. Non-functional requirements
    - 4.1 Performance
    - 4.2 Security and privacy
    - 4.3 Reliability and availability
    - 4.4 Usability
    - 4.5 Interoperability
5. Interface requirements
    - 5.1 User interfaces
    - 5.2 Software interfaces
    - 5.3 Communication interfaces

Q10



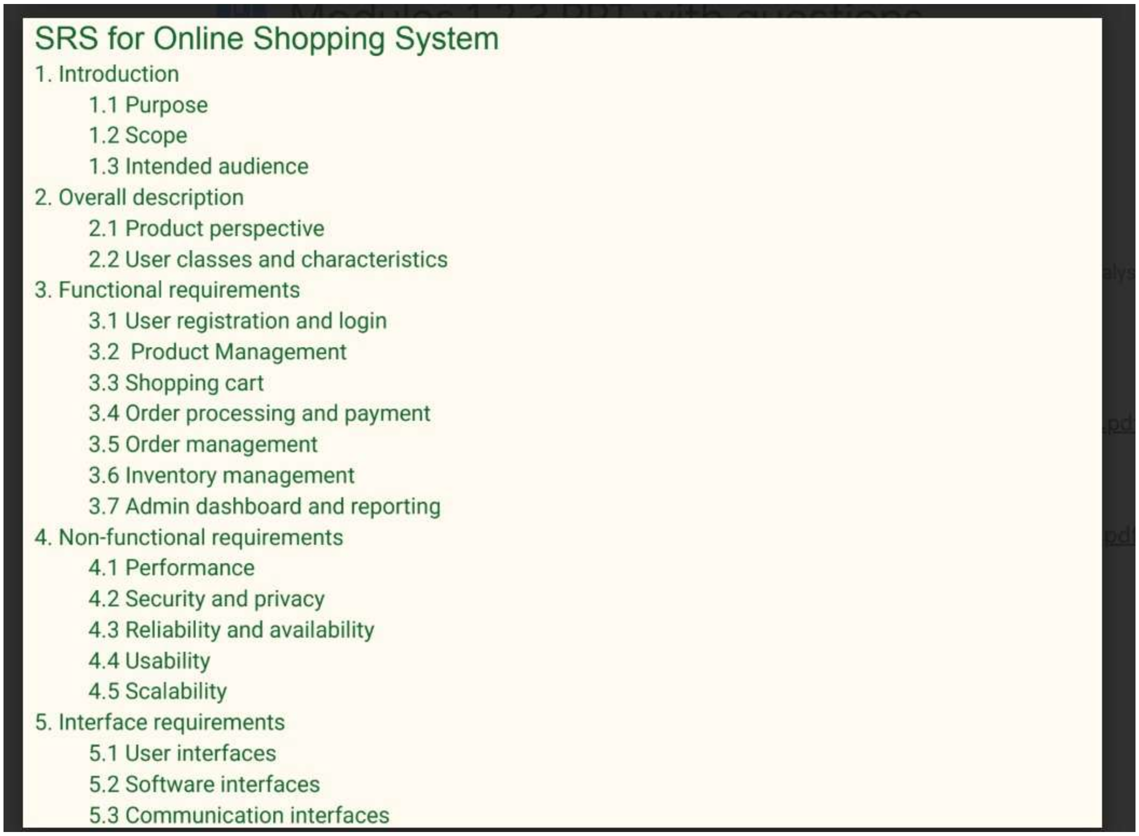
SRS for student database management system

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Intended audience
2. Overall description
    - 2.1 Product perspective
    - 2.2 Product features
    - 2.3 User classes and characteristics
3. 3.1 Functional requirements
    - 3.1.1 Reservation and booking
    - 3.1.2 Guest and front desk management
    - 3.1.3 Room and housekeeping management
    - 3.1.4 Billing and payments
    - 3.1.5 Inventory management
    - 3.1.6 Reporting and analytics

3.2 Non-functional requirements
- 3.2.1 Performance
- 3.2.2 Security and privacy
- 3.2.3 Reliability and availability
- 3.2.4 Usability
- 3.2.5 Scalability
- 3.2.6 Compatibility

4. Analysis models
    - 4.1 Data flow diagram (DFD)
    - 4.2 Entity-relationship diagram (ERD)
    - 4.3 Use case diagram
5. Interface requirements
    - 5.1 User interfaces
    - 5.2 Software interfaces
    - 5.3 Communication interfaces

Q11



SRS for Online Shopping System

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Intended audience
2. Overall description
    - 2.1 Product perspective
    - 2.2 User classes and characteristics
3. Functional requirements
    - 3.1 User registration and login
    - 3.2 Product Management
    - 3.3 Shopping cart
    - 3.4 Order processing and payment
    - 3.5 Order management
    - 3.6 Inventory management
    - 3.7 Admin dashboard and reporting
4. Non-functional requirements
    - 4.1 Performance
    - 4.2 Security and privacy
    - 4.3 Reliability and availability
    - 4.4 Usability
    - 4.5 Scalability
5. Interface requirements
    - 5.1 User interfaces
    - 5.2 Software interfaces
    - 5.3 Communication interfaces