**Mocks – 4**

**Questions:**

A company is having manufacturing plants and warehouses in various parts of the country. They manufacture ice-cream and milk products. They want to build software to achieve two goals.

• Manage the inventory

• Quickest delivery to the customers

**Assignment 1:**

**1. Please make a BRD which can be presented to the client along with complete development and resource plan.**

**Business Requirement Document (BRD)**

**Project Name:** Inventory Management and Quick Delivery Software for Ice Cream and Milk Product Manufacturing Company.

**Project ID:** INV001

**Document Version: 1.0**

**Date:** 16/01/2025

**1. Executive Summary**

The purpose of this project is to develop a comprehensive software solution to effectively manage inventory and ensure prompt delivery of ice cream and milk products. The software will enhance operational efficiency by integrating real-time inventory tracking and delivery route optimization across manufacturing plants and warehouses, leading to improved customer satisfaction and reduced wastage.

**2. Business Objectives**

* Efficiently manage inventory across multiple locations, including manufacturing plants and warehouses.
* Minimize delays and optimize delivery schedules to meet customer expectations.
* Reduce wastage of perishable goods through improved inventory and expiry tracking.
* Provide stakeholders with real-time visibility and insights into inventory and logistics operations.
* Enhance overall operational efficiency and reduce associated costs.

**3. Scope of the Project**

**3.1 In-Scope**

* **Inventory Management:**
  + Centralized real-time tracking of stock levels.
  + Configurable stock level alerts and notifications for replenishment.
  + Automated reordering system based on predefined thresholds.
  + Batch and expiry tracking for perishable goods.
* **Delivery Optimization:**
  + Dynamic route optimization using real-time traffic data.
  + Real-time tracking of delivery vehicles.
  + Integration with customer order systems for seamless order updates.
  + Analytics on delivery performance and turnaround times.
* **Reporting and Analytics:**
  + Detailed inventory performance and usage reports.
  + Delivery time analysis and optimization recommendations.
  + Predictive analytics for demand forecasting based on historical data.

**3.2 Out-of-Scope**

* Integration with external e-commerce platforms.
* Development of a consumer-facing mobile application (may be considered for future enhancements).

**Approvals:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Name** | **Title** | **Signature** | **Date** |
| Project Sponsor | Ramesh Kumar | Director of Operations | Signed | 21/08/2024 |
| Business Owner | Anjali Sharma | HR Head | Signed | 22/08/2024 |
| Project Manager | Arjun Singh | Senior Project Manager | Signed | 22/08/2024 |
| System Architect | Priya Iyer | Lead System Architect | Signed | 23/08/2024 |
| Development Lead | Rajesh Gupta | Development Team Lead | Signed | 24/08/2024 |
| UX/UI Lead | Sneha Nair | UX/UI Designer | Signed | 24/08/2024 |
| Quality Lead | Vikram Reddy | QA Manager | Signed | 25/08/2024 |
| Content Lead | Meera Menon | Content Strategist | Signed | 25/08/2024 |

**RACI Chart:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Position** | **R** | **A** | **S** | **C** | **I** |
| **Ramesh Kumar** | Project Sponsor |  | ✓ |  |  | ✓ |
| **Anjali Sharma** | Business Owner |  | ✓ |  | ✓ |  |
| **Arjun Singh** | Project Manager | ✓ | ✓ | ✓ |  | ✓ |
| **Priya Iyer** | System Architect | ✓ | ✓ |  |  |  |
| **Rajesh Gupta** | Development Lead | ✓ |  |  | ✓ |  |
| **Sneha Nair** | User Experience Lead |  |  | ✓ | ✓ |  |
| **Vikram Reddy** | Quality Lead |  |  | ✓ | ✓ |  |
| **Meera Menon** | Content Lead |  |  | ✓ | ✓ |  |
| **Rahul Verma** | Client Representative |  |  | ✓ | ✓ |  |

* **R (Responsible):** The person(s) responsible for performing the task.
* **A (Accountable):** The person who is ultimately accountable for the task’s completion.
* **S (Supportive):** People who provide support and resources to complete the task.
* **C (Consulted):** People who are consulted and whose opinions are sought.
* **I (Informed):** People who are kept informed about the task but are not directly involved.

**Business Goals:**

**Efficient Inventory Management:**

* **Objective:** To maintain real-time tracking and management of inventory levels across all manufacturing plants and warehouses.
* **Key Goals:**
  + Optimize stock levels to prevent overstocking or stockouts.
  + Enable automated reordering of products based on predefined thresholds.
  + Ensure accurate and up-to-date inventory data across multiple locations.
  + Streamline the management of perishable items, such as ice-cream and milk products, to minimize wastage.
  + Provide detailed reporting on inventory status, product movement, and warehouse utilization.

**Quickest Delivery to Customers:**

* **Objective:** To reduce delivery time and ensure timely and efficient distribution of products to customers.
* **Key Goals:**
  + Implement a route optimization system to ensure the fastest and most cost-effective delivery routes.
  + Enable real-time tracking and updates for both customers and delivery teams.
  + Ensure that the nearest warehouse or plant is always prioritized for deliveries to customers, reducing delivery time.
  + Improve communication and coordination between warehouses, plants, and delivery teams.
  + Enhance delivery scheduling to meet customer demands, including emergency or expedited deliveries.
  + Offer customers flexible delivery options (e.g., same-day delivery, scheduled delivery).

**Modules:**

**Inventory Management Module**

* + Real-time inventory tracking.
  + Stock level management and alerts for replenishment.
  + Batch tracking for perishable items (ice-cream and milk products).
  + Integration with barcode/QR code systems.
  + Expiry date and shelf-life monitoring.
  + Transfer management between warehouses and plants.
  + Reporting and analytics for inventory performance.

**2. Order Management Module**

* + Order processing and status tracking.
  + Automated order allocation to the nearest warehouse or plant.
  + Order cancellation, modification, and returns management.
  + Customer notifications for order confirmation and delivery updates.
  + Integration with customer-facing systems (e.g., website or mobile app).

**3. Delivery Management Module**

* + Route optimization for quickest deliveries.
  + Delivery scheduling and prioritization.
  + Integration with GPS for real-time delivery tracking.
  + Proof of delivery (e-signatures, images).
  + Delivery fleet management and assignment.

**4. Warehouse Management Module**

* + Warehouse space optimization.
  + Product categorization and storage location tracking.
  + Inbound and outbound logistics management.
  + Quality checks for incoming and outgoing products.

**5. Customer Management Module**

* + Customer database and profile management.
  + Order history and preferences tracking.
  + Feedback and complaints management.
  + Loyalty programs and discounts.

**6. Reporting and Analytics Module**

* + Inventory turnover reports.
  + Delivery time analysis and performance metrics.
  + Sales reports and demand forecasting.
  + Wastage and loss tracking reports.
  + Real-time dashboards for key metrics.

**7. User Management and Role-Based Access Control Module**

* + Role-based access for different user types (e.g., admins, warehouse staff, delivery teams).
  + User activity logs and audit trails.
  + Secure authentication mechanisms.

**8. Integration and API Module**

* + APIs for integration with external systems (e.g., e-commerce platforms, CRM tools).
  + Payment gateway integration (if needed for direct sales).
  + Integration with IoT devices like temperature sensors in warehouses.

**9. Notification and Alerts Module**

* + Stock alerts for replenishment and expiry warnings.
  + Delivery reminders and updates for customers and staff.
  + Notifications for system errors or delays.

**Functional and Non – Functional Requirement:**

**Functional Requirement:**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Description** | **Priority** |
| FR001 | Track real-time inventory across all manufacturing plants and warehouses. | High |
| FR002 | Allow users to add, update, and delete inventory items. | High |
| FR003 | Generate alerts for inventory replenishment based on pre-set thresholds. | High |
| FR004 | Enable batch tracking, including expiry dates for products. | High |
| FR005 | Categorize inventory by type, location, and status. | High |
| FR006 | Provide a dashboard showing inventory levels, trends, and alerts. | High |
| FR007 | Integrate with barcode/QR code scanners for inventory tracking. | Medium |
| FR008 | Enable stock transfers between locations, including approval workflows. | Medium |
| FR009 | Generate reports for stock movement, utilization, and wastage analysis. | Medium |
| FR010 | Integrate with GPS systems for delivery route optimization. | High |
| FR011 | Integrate with IoT sensors for warehouse temperature monitoring. | Medium |
| FR012 | Notify users about low stock or other critical alerts via email/SMS. | High |
| FR013 | Support automated order placement for replenishment when stocks are low. | High |
| FR014 | Allow customers to place orders online. | High |
| FR015 | Enable customers to track their orders in real-time. | High |
| FR016 | Integrate with external payment gateways for customer orders. | Medium |
| FR017 | Support multi-tier user roles with specific access permissions. | High |
| FR018 | Maintain a comprehensive audit log for all inventory transactions. | High |
| FR019 | Enable product bundling for order placement (e.g., ice cream combos). | Medium |
| FR020 | Optimize packing and scheduling for delivery based on order priorities. | High |
| FR021 | Estimate delivery times for customer orders based on location. | High |
| FR022 | Support integration with third-party logistics providers. | Medium |
| FR023 | Generate automated invoices for customer orders. | High |
| FR024 | Calculate and display delivery costs dynamically. | Medium |
| FR025 | Support configurable tax settings for billing and invoicing. | Medium |
| FR026 | Allow customers to schedule deliveries based on their preferences. | Medium |
| FR027 | Support multi-location warehouse search for order fulfillment. | High |
| FR028 | Display order history and statuses for customers. | High |
| FR029 | Provide real-time inventory updates post order placement. | High |
| FR030 | Synchronize financial data with existing ERP systems. | Medium |
| FR031 | Provide predictive analytics for sales and inventory trends. | Medium |
| FR032 | Track and display damaged or expired stock. | Medium |
| FR033 | Allow bulk upload of inventory data via CSV/Excel files. | Low |
| FR034 | Allow custom reporting based on user-defined parameters. | Medium |
| FR035 | Provide role-specific dashboards for operational staff and managers. | Medium |
| FR036 | Prioritize high-demand products in inventory restocking. | Medium |
| FR037 | Support API integrations with e-commerce platforms. | Medium |
| FR038 | Track and display warehouse capacity utilization in real-time. | Medium |
| FR039 | Enable dynamic pricing for bulk orders. | Medium |
| FR040 | Provide API-based integrations with analytics tools (e.g., Power BI). | Medium |

**Non – Functional Requirement:**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Description** | **Priority** |
| NFR001 | Handle 1,000 concurrent users at peak load. | High |
| NFR002 | Provide a response time of under 2 seconds for all critical operations. | High |
| NFR003 | Ensure 99.9% uptime for critical functionalities. | High |
| NFR004 | Support scaling to 2x the current number of warehouses. | High |
| NFR005 | Encrypt all sensitive data in transit and at rest. | High |
| NFR006 | Comply with food safety standards and regulations. | High |
| NFR007 | Provide a multilingual interface supporting English and Hindi. | Medium |
| NFR008 | Ensure seamless integration with legacy systems. | High |
| NFR009 | Provide real-time data updates with a latency of less than 1 second. | Medium |
| NFR010 | Support mobile and desktop platforms. | High |
| NFR011 | Provide role-specific customizations for user interfaces. | High |
| NFR012 | Back up all data every 30 minutes. | High |
| NFR013 | Ensure recovery from data loss within 1 hour. | High |
| NFR014 | Provide RESTful APIs for third-party integrations. | High |
| NFR015 | Support IoT protocols for sensor data exchange. | Medium |
| NFR016 | Use a cloud-based architecture for flexibility and scalability. | High |
| NFR017 | Ensure a seamless user experience with an intuitive UI design. | High |
| NFR018 | Complete software updates within 2 hours of deployment. | Medium |
| NFR019 | Generate reports within 10 seconds of a user request. | Medium |
| NFR020 | Archive historical data for at least 5 years. | Medium |
| NFR021 | Limit downtime during maintenance to 1 hour per month. | High |
| NFR022 | Support multi-factor authentication for all user roles. | High |
| NFR023 | Integrate seamlessly with third-party logistics tools. | Medium |
| NFR024 | Ensure that reports are customizable based on user roles. | Medium |
| NFR025 | Use secure payment gateways with PCI compliance. | Medium |
| NFR026 | Support concurrent deliveries across 100 locations. | High |
| NFR027 | Perform scalability testing every quarter. | Medium |
| NFR028 | Adhere to ISO 27001 for data security. | High |
| NFR029 | Support data exports in multiple formats (e.g., CSV, Excel, PDF). | Medium |
| NFR030 | Minimize power consumption for IoT integrations. | Low |
| NFR031 | Ensure load balancing for high availability. | High |
| NFR032 | Allow real-time order status updates via APIs. | High |
| NFR033 | Ensure database query execution time under 3 seconds. | Medium |
| NFR034 | Support multiple delivery prioritization strategies. | Medium |
| NFR035 | Allow automated deployment of new features with zero downtime. | Medium |
| NFR036 | Maintain a secure sandbox environment for testing new modules. | Medium |
| NFR037 | Provide automated SLA monitoring for logistics integrations. | Medium |
| NFR038 | Adhere to GDPR and local privacy laws. | High |
| NFR039 | Provide high-resolution UI elements for seamless cross-device usage. | Medium |
| NFR040 | Ensure transaction logs are immutable for auditing. | High |

**Assumptions:**

**Inventory Details**

* The company already has a database of all inventory items categorized by product type, location, and current stock levels.
* Existing barcode or QR code labels are available for inventory tracking.

**Manufacturing and Warehousing**

* Manufacturing plants and warehouses are equipped with IoT-enabled devices for temperature and humidity monitoring, where required.
* Stock is periodically updated and synchronized across all warehouses.

**Delivery System**

* The company has an established logistics network with in-house and third-party delivery partners.
* Delivery vehicles are GPS-enabled for real-time location tracking.

**Software Integration**

* The company uses basic ERP or other software systems that can be integrated with the new solution.
* APIs or interfaces for data exchange between the new software and existing systems are available.

**User Roles and Permissions**

* Employees in relevant roles (inventory managers, logistics coordinators, etc.) have basic technical skills for operating the software.
* The system will have multi-level user access with permissions based on roles.

**Customer Interaction**

* Customers can place orders via a website or mobile app linked to the software.
* Most customers have access to smartphones for order tracking and notifications.

**Regulatory Compliance**

* The software will adhere to local regulations for food safety, data protection, and delivery protocols.
* Taxation rules and requirements are well-documented and will be provided for implementation.

**Scalability**

* The software is designed to accommodate a 50% growth in inventory and customer base over the next five years.
* New warehouse or manufacturing locations can be added seamlessly.

**Technology**

* Reliable and high-speed internet is available at all locations to ensure uninterrupted system functionality.
* Cloud-based infrastructure is assumed for flexibility and scalability.

**Training and Support**

* The company will provide training sessions for employees to understand and operate the new system.
* A dedicated support team will be available to address system issues post-deployment.

**Budget Constraints:**

* The project must be completed within the allocated budget.
* Any additional features or upgrades may require separate approvals.

**Timeline Constraints**

* The software development and deployment must be completed within a specified timeframe to align with business operations.
* Testing and training need to be completed before the start of the next peak season.

**Resource Availability**

* Limited availability of key stakeholders for requirement gathering and validation.
* The development team has a predefined capacity and cannot exceed a certain workload.

**Technological Constraints**

* The software must be compatible with existing hardware and infrastructure, such as IoT devices, barcode scanners, and GPS systems.
* Legacy systems may limit integration capabilities or require additional middleware.

**Regulatory Compliance**

* The solution must comply with government regulations for food safety, data privacy, and taxation.
* Constraints may arise from regional variations in regulations across the country.

**Data Accuracy**

* The software's effectiveness depends on the accuracy of existing inventory and logistics data, which may require initial cleanup and validation.

**Geographic Spread**

* The company's operations are spread across various parts of the country, which may lead to challenges in real-time synchronization and network reliability.

**Delivery Constraints**

* Delivery routes and times are subject to external factors such as weather conditions, traffic, and vehicle availability.
* Cold chain logistics for ice cream and milk products require strict temperature control during transportation.

**Training and Adoption**

* The system must be user-friendly as employees may have varying levels of technical proficiency.
* Resistance to adopting new technology may cause delays in implementation.

**Scalability**

* The software must handle current operations while allowing for future expansion, which may be constrained by initial design decisions.

**Security Constraints**

* Data security measures must not overly complicate the user experience while ensuring compliance with industry standards.

**Third-party Dependencies**

* Reliance on third-party delivery services or logistics providers may limit the system's end-to-end control over operations.

**Performance Requirements**

* The system must deliver real-time updates, but processing speed could be constrained by network bandwidth and server capacity.

**Risks:**

**Operational Risks**

1. **Inventory Inaccuracy**
   * Existing discrepancies in inventory data could lead to errors in the system, resulting in overstocking or stockouts.
2. **Supply Chain Disruption**
   * Delays in the supply chain due to natural disasters, strikes, or transportation issues may affect delivery schedules.
3. **Cold Chain Failures**
   * The failure to maintain the required temperature during storage or transportation may result in product spoilage.
4. **Resistance to Change**
   * Employees may be resistant to adopting the new system, causing delays in implementation and inefficiencies in operations.
5. **Inconsistent Internet Connectivity**
   * Remote warehouses and plants may face connectivity issues, affecting real-time data updates.

**Technical Risks**

1. **Integration Challenges**
   * Difficulties in integrating the new software with existing systems, such as IoT devices, barcode scanners, or ERP platforms.
2. **Performance Issues**
   * High transaction volumes may lead to system slowdowns or crashes, especially during peak seasons.
3. **Data Migration Risks**
   * Errors during data migration from legacy systems could result in loss or corruption of critical data.
4. **Cybersecurity Threats**
   * Potential risks of data breaches or hacking, leading to loss of sensitive business information.
5. **Scalability Limitations**
   * The system may fail to handle future growth in operations if not designed for scalability.

**Financial Risks**

1. **Budget Overruns**
   * Unforeseen complexities in software development could lead to exceeding the allocated budget.
2. **Loss of Revenue**
   * Delayed implementation or system downtime may result in missed sales opportunities.

**Compliance Risks**

1. **Regulatory Non-Compliance**
   * Failure to comply with regional food safety, taxation, or data privacy regulations could result in fines or penalties.
2. **Environmental Impact Regulations**
   * Cold chain operations may need to meet specific environmental standards, such as reduced carbon emissions.

**Customer Satisfaction Risks**

1. **Delivery Delays**
   * Failure to ensure timely delivery may lead to dissatisfied customers and damage the company’s reputation.
2. **User Experience Issues**
   * Poorly designed user interfaces may lead to inefficiencies or errors in system usage by employees.

**Project Risks**

1. **Timeline Overruns**
   * Delays in requirement gathering, development, or testing phases may push the project beyond its planned deadline.
2. **Lack of Stakeholder Alignment**
   * Conflicting priorities or unclear requirements from stakeholders could derail the project.
3. **Vendor Dependency**
   * Dependence on external vendors for software components or logistics services may introduce risks if vendors fail to deliver.
4. **Inadequate Training**
   * Employees may not be adequately trained to use the system effectively, reducing its overall benefits.

**Resource Planning:**

**1. Budget Breakdown**

|  |  |  |
| --- | --- | --- |
| **Category** | **Estimated Cost (₹)** | **Percentage of Budget** |
| Software Development | 35,00,000 | 35% |
| Infrastructure (Hardware/Cloud) | 15,00,000 | 15% |
| Testing & Quality Assurance | 10,00,000 | 10% |
| UX/UI Design | 8,00,000 | 8% |
| Project Management | 7,00,000 | 7% |
| Training & Change Management | 5,00,000 | 5% |
| Miscellaneous (Documentation, etc.) | 5,00,000 | 5% |
| Contingency Reserve | 15,00,000 | 15% |
| **Total** | **1,00,00,000** | **100%** |

**2. Team Allocation**

|  |  |
| --- | --- |
| **Role** | **No. of Resources** |
| **Project Manager** | 1 |
| **System Architect** | 1 |
| **Backend Developers** | 3 |
| **Frontend Developers** | 2 |
| **UX/UI Designers** | 2 |
| **Business Analyst** | 2 |
| **QA Engineers** | 2 |
| **DevOps Engineer** | 1 |
| **Trainers (Post-Implementation)** | 2 |
| **Content Specialist** | 1 |
| **Client Support Team** | 2 |
| **Total** | 19 |

**3. Timeline Allocation**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Duration (Months)** | **Key Activities** |
| **1. Requirement Gathering & Analysis** | 2 | Stakeholder meetings, BRD preparation, approvals. |
| **2. System Design** | 2 | Architecture design, wireframes, and prototypes. |
| **3. Development** | 6 | Backend and frontend development, integrations. |
| **4. Testing & QA** | 2 | System testing, UAT, bug fixes, and refinements. |
| **5. Deployment** | 1 | Final deployment, go-live preparations. |
| **6. Post-Implementation Support** | 1 | Training, troubleshooting, and handover. |

**4. Infrastructure and Tools**

|  |  |  |
| --- | --- | --- |
| **Item** | **Description** | **Estimated Cost (₹)** |
| Cloud Services | AWS/Azure for hosting | 10,00,000 |
| Development Tools | IDEs, version control | 3,00,000 |
| Testing Tools | Selenium, JIRA, etc. | 2,00,000 |
| Hardware | Laptops, devices | 3,00,000 |
| Total | - | **18,00,000** |

**5. Contingency Plan**

* Budget Reserve (₹15,00,000): Reserved for unexpected costs like additional resources, infrastructure scaling, or software updates.

**2. Prepare process flow diagram using your imagination.**



**Assignment 2:**

**1.Write an introduction letter to a client introducing yourself as a business analyst in charge of working with the client and his team to start the business understanding process.**

**Answer:**

**Subject:** Introduction as Your Business Analyst

**Dear Rajesh,**

I hope this message finds you well. Myself Rakshana R, and I am thrilled to introduce myself as the Business Analyst assigned to collaborate with you and your team on the development of a comprehensive software solution. This project aims to address your key business objectives: efficient inventory management and ensuring the quickest delivery of your ice-cream and milk products to customers across the country. With a strong focus on delivering tailored and actionable solutions, I am committed to guiding the initial phases of understanding your business requirements and aligning them with the project’s goals.

**About My Role: As your Business Analyst, my primary responsibilities include:**

* **Understanding Your Business Needs:** Collaborating closely with stakeholders to identify and analyze challenges related to inventory management and delivery operations.
* **Requirements Documentation:** Gathering and documenting detailed requirements for the proposed software solution, ensuring alignment with your business objectives.
* **Solution Alignment:** Working with development teams to design a solution that optimizes inventory processes and delivery logistics.
* **Facilitating Communication:** Acting as the central point of contact between your team and our development team to ensure transparency and smooth progress throughout the project.

**Key Benefits of the Project:**

1. **Enhanced Inventory Management:** The solution will provide real-time tracking of raw materials and finished goods, helping reduce wastage and ensure availability.
2. **Optimized Delivery Operations:** By leveraging advanced algorithms, the software will optimize delivery routes to ensure quicker and cost-effective service to customers.
3. **Improved Decision-Making:** With integrated analytics and reporting features, your team will gain valuable insights into operational performance.

**Next Steps: To initiate this exciting journey, I propose scheduling an initial meeting where we can:**

1. Discuss your specific challenges and pain points in detail.
2. Identify key stakeholders and define their roles in the project.
3. Outline a preliminary timeline, deliverables, and success criteria for the software solution.
4. Address any questions or concerns to ensure we are aligned on project expectations.

Please let me know a convenient time for this meeting. I am flexible and can adapt to your schedule to make the discussion as productive as possible.

**Contact Information:** Should you have any questions or require immediate assistance, please do not hesitate to reach out to me directly at rakshana@abccompany.com or +91 90000 00000.

I am genuinely excited about the opportunity to collaborate with you and your team to create a solution that streamlines your operations and enhances customer satisfaction. Thank you for entrusting us with this important project. Together, I am confident we will achieve outstanding results.

**Warm regards,**

**Rakshana R  
Business Analyst  
Rakshana R  
90000 00000**

**2. Prepare a brief BRD and SRS for a project- Horoscope or Ticketing system or online store.**

**BRD Template:**

**Online Store:**

**Project Name:** Online Store Management and Optimization Software

**Project ID:** OS001

**Document Version:** 1.0

**Date:** 23/01/2025

**1. Executive Summary**

The goal of this project is to develop an online store management software to streamline operations, enhance customer experience, and optimize the e-commerce workflow. This software will enable real-time inventory tracking, efficient order processing, and personalized customer interactions, ensuring seamless online shopping experiences and improved operational efficiency.

**2. Business Objectives**

* + Efficiently manage online inventory, including stock levels, pricing, and product availability.
  + Ensure quick and accurate order processing and delivery.
  + Provide customers with a user-friendly interface for browsing, purchasing, and tracking orders.
  + Integrate analytics to track sales, customer preferences, and operational efficiency.
  + Enhance customer satisfaction and retention through personalized experiences and reliable services.

**3. Scope of the Project**

**3.1 In-Scope**

**Inventory Management:**

• Real-time stock tracking with automated alerts for replenishment.

• Product categorization and detailed descriptions for easy navigation.

• Dynamic pricing adjustments based on demand and stock levels.

**Order Management:**

* + Order placement, processing, and status tracking.
  + Integration with payment gateways for secure transactions.
  + Return and refund management.

**Delivery Optimization:**

* + Real-time delivery tracking for customers and admins.
  + Dynamic route optimization for delivery efficiency.
  + Notifications for delivery updates.

**Customer Engagement:**

* + Personalized product recommendations based on browsing history.
  + Customer support integration for queries and complaints.
  + Loyalty programs and special offers.

**Reporting and Analytics:**

* + Sales performance reports.
  + Customer behavior analytics.
  + Inventory turnover and replenishment insights.

**3.2 Out-of-Scope**

* + Offline retail store integration.
  + Integration with third-party marketplaces (e.g., Amazon, eBay).

**Approvals**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Name** | **Title** | **Signature** | **Date** |
| Project Sponsor | Anil Kapoor | E-commerce Director | Signed | 20/01/2025 |
| Business Owner | Priya Menon | Operations Head | Signed | 21/01/2025 |
| Project Manager | Rahul Sharma | Senior Project Manager | Signed | 21/01/2025 |
| System Architect | Sneha Iyer | Lead System Architect | Signed | 22/01/2025 |
| Development Lead | Karan Mehta | Development Team Lead | Signed | 22/01/2025 |
| UX/UI Lead | Megha Nair | UX/UI Designer | Signed | 23/01/2025 |
| Quality Lead | Aditya Rao | QA Manager | Signed | 23/01/2025 |

**Business Goals**

Efficient Online Store Management:

**Objective:** To simplify and optimize the management of the online store’s operations.

**Key Goals:**

* + Ensure real-time inventory updates for accurate stock levels.
  + Automate reordering and replenishment processes.
  + Provide detailed insights into product performance and sales trends.
  + Enable easy management of product listings and categories.

**Enhanced Customer Experience:**

**Objective:** To provide a seamless, user-friendly online shopping platform.

**Key Goals:**

* + Ensure fast and secure payment processing.
  + Provide real-time updates on order status and delivery.
  + Personalize customer recommendations to improve sales.
  + Offer responsive customer support for resolving queries promptly.

**Modules**

**Inventory Management Module**

* + Real-time inventory tracking and updates.
  + Product categorization and search optimization.
  + Alerts for low stock levels.
  + Integration with barcode scanning systems.

**Order Management Module**

* + Order placement and status tracking.
  + Integration with payment gateways.
  + Management of returns, cancellations, and refunds.
  + Notifications for order confirmation and shipping.

**Delivery Management Module**

* + Route optimization for timely deliveries.
  + Real-time delivery tracking for customers.
  + Fleet and delivery assignment management.
  + Proof of delivery (e.g., e-signatures).

**Customer Engagement Module**

* + Personalized product recommendations.
  + Loyalty program management.
  + Customer feedback and complaint resolution.

**Reporting and Analytics Module**

* + Real-time sales dashboards.
  + Inventory and stock performance reports.
  + Customer behavior analytics.
  + Predictive analytics for demand forecasting.

**User Management and Role-Based Access Control Module**

* + Role-specific permissions and access control.
  + Audit trails and user activity logs.
  + Secure login and authentication mechanisms.

**Notification and Alerts Module**

* + Order and delivery status updates.
  + Stock replenishment and expiration alerts.
  + Promotional notifications for customers.

**RACI Matrix:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Position** | **R** | **A** | **S** | **C** | **I** |
| Ramesh Kumar | Project Sponsor |  | ✓ |  | ✓ |  |
| Anjali Sharma | Business Owner | ✓ |  | ✓ |  |  |
| Arjun Singh | Project Manager | ✓ | ✓ | ✓ |  | ✓ |
| Priya Iyer | System Architect | ✓ | ✓ |  |  |  |
| Rajesh Gupta | Development Lead | ✓ |  |  | ✓ |  |
| Sneha Nair | User Experience Lead |  | ✓ |  | ✓ |  |
| Vikram Reddy | Quality Lead |  | ✓ |  | ✓ |  |
| Meera Menon | Content Lead |  | ✓ |  | ✓ |  |
| Rahul Verma | Client Representative |  | ✓ |  | ✓ |  |

**Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Description** | **Priority** |
| FR001 | Allow users to register, log in, and manage their profiles. | High |
| FR002 | Enable users to browse products by categories and search filters. | High |
| FR003 | Allow users to add, update, or remove items from their cart. | High |
| FR004 | Support guest checkout for faster transactions. | High |
| FR005 | Enable secure payment options with multiple gateways. | High |
| FR006 | Provide real-time stock availability updates. | High |
| FR007 | Allow customers to apply promotional codes during checkout. | Medium |
| FR008 | Generate and send order confirmations via email/SMS. | High |
| FR009 | Allow users to track their orders in real-time. | High |
| FR010 | Support product reviews and ratings by customers. | Medium |
| FR011 | Enable wishlist creation for future purchases. | Medium |
| FR012 | Provide automated recommendations based on user behavior. | High |
| FR013 | Allow integration with third-party logistics for delivery. | Medium |
| FR014 | Provide a return/refund management system. | High |
| FR015 | Allow customers to request gift wrapping for their orders. | Low |
| FR016 | Support multilingual product descriptions. | Medium |
| FR017 | Enable live chat support for customer queries. | Medium |
| FR018 | Allow admin users to add, edit, or delete products. | High |
| FR019 | Provide reporting tools for sales analytics and inventory. | High |
| FR020 | Allow product sorting by price, rating, and popularity. | Medium |
| FR021 | Enable location-based product availability checks. | High |
| FR022 | Support recurring order subscriptions for specific products. | Medium |
| FR023 | Allow dynamic pricing based on demand and supply. | Medium |
| FR024 | Provide product comparisons for customers. | Medium |
| FR025 | Allow filtering products based on sustainability (e.g., eco-friendly). | Low |
| FR026 | Integrate with external CRM systems for customer management. | Medium |
| FR027 | Provide order history and reorder functionality. | High |
| FR028 | Support bulk order placements for B2B customers. | Medium |
| FR029 | Allow custom messages for gift deliveries. | Low |
| FR030 | Enable users to save multiple shipping addresses. | Medium |
| FR031 | Provide a product availability alert system. | Medium |
| FR032 | Enable barcode scanning for warehouse management. | Medium |
| FR033 | Support multiple payment methods, including wallets and BNPL. | High |
| FR034 | Allow scheduled deliveries based on customer preferences. | Medium |
| FR035 | Enable loyalty point accumulation and redemption. | Medium |
| FR036 | Provide a secure login option using social media accounts. | Low |
| FR037 | Track delivery statuses and notify customers of updates. | High |
| FR038 | Allow customers to leave suggestions for out-of-stock items. | Low |
| FR039 | Enable product tagging for better search visibility. | Medium |
| FR040 | Provide admin dashboards for monitoring website performance. | High |

**Non-Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Description** | **Priority** |
| NFR001 | Ensure a response time of under 2 seconds for key user actions. | High |
| NFR002 | Provide 99.9% system uptime, excluding scheduled maintenance. | High |
| NFR003 | Support 1,000 concurrent users with no performance degradation. | High |
| NFR004 | Encrypt sensitive data at rest and during transmission. | High |
| NFR005 | Ensure compliance with PCI DSS for payment security. | High |
| NFR006 | Provide a scalable architecture to handle peak traffic. | High |
| NFR007 | Enable seamless cross-platform usage (desktop and mobile). | High |
| NFR008 | Adhere to GDPR and other local privacy laws. | High |
| NFR009 | Back up all system data every 15 minutes. | Medium |
| NFR010 | Restore data within 30 minutes of a failure. | High |
| NFR011 | Ensure a multilingual interface for global users. | Medium |
| NFR012 | Allow updates with zero downtime through CI/CD pipelines. | Medium |
| NFR013 | Provide real-time monitoring of server health and performance. | High |
| NFR014 | Generate sales reports within 10 seconds. | Medium |
| NFR015 | Allow user sessions to expire after 15 minutes of inactivity. | High |
| NFR016 | Ensure integration compatibility with third-party APIs. | High |
| NFR017 | Store logs for at least 6 months for auditing purposes. | Medium |
| NFR018 | Maintain an intuitive and accessible UI/UX design. | High |
| NFR019 | Support role-based access control for secure admin operations. | High |
| NFR020 | Ensure scalability testing every six months. | Medium |
| NFR021 | Provide role-specific performance dashboards. | Medium |
| NFR022 | Archive inactive customer accounts after one year. | Medium |
| NFR023 | Minimize server-side energy consumption for eco-friendliness. | Low |
| NFR024 | Ensure all UI components are WCAG 2.1 compliant. | Medium |
| NFR025 | Allow exports of financial reports in multiple formats. | Medium |
| NFR026 | Support delivery simulations for logistics planning. | Medium |
| NFR027 | Monitor API usage and throttle requests beyond set limits. | High |
| NFR028 | Adhere to ISO 27001 standards for security management. | High |
| NFR029 | Provide disaster recovery support for critical failures. | High |
| NFR030 | Test the platform on all major browsers quarterly. | Medium |
| NFR031 | Allow dynamic scaling for holiday sales peaks. | Medium |
| NFR032 | Minimize page load times to under 3 seconds for all pages. | High |
| NFR033 | Ensure all payments are processed within 5 seconds. | High |
| NFR034 | Automate SLA compliance tracking for third-party integrations. | Medium |
| NFR035 | Use a Content Delivery Network (CDN) to distribute static assets. | Medium |
| NFR036 | Ensure transactions are logged immutably for compliance. | High |
| NFR037 | Provide localization support for currencies and taxes. | Medium |
| NFR038 | Enable automatic failover for high availability. | High |
| NFR039 | Perform load testing before major promotional events. | High |
| NFR040 | Secure admin operations with two-factor authentication. | High |

**Assumptions:**

1. Resource Availability: All key team members (e.g., developers, designers, content creators) are available as required for the project.
2. Technology Stack: The project will use the agreed-upon technology stack and third-party tools without any major changes during implementation.
3. Business Requirements: Business requirements will remain stable throughout the project and will be well defined at the beginning.
4. Stakeholder Engagement: Stakeholders will be actively engaged and provide timely feedback.
5. Budget Allocation: The project will have a consistent budget allocation throughout its lifecycle with no unexpected financial constraints.
6. Testing Environment: A suitable testing environment will be provided for the quality assurance process, and all necessary data will be available for testing.

**Constraints:**

1. Time Limit: The project must be completed within a fixed timeline, which may limit the ability to incorporate additional features or modifications during development.
2. Budget: There is a predefined budget for the project, limiting the flexibility for extra expenses or features.
3. Compliance Regulations: The online store must comply with all local laws and regulations (e.g., data protection, privacy laws, etc.), which might affect design and functionality.
4. Integration with Third-party Services: The availability and reliability of third-party services (payment gateways, shipping services) could constrain the project.
5. Technology Compatibility: The store needs to be compatible with different devices, browsers, and operating systems, which may create challenges in the development process.
6. Skill Limitations: The project might face challenges due to the current skill levels of some team members or the availability of subject matter experts.

**Risks:**

1. Requirement Changes: Any significant changes in the business requirements during the project lifecycle could cause delays and require additional resources.
2. Scope Creep: Uncontrolled expansion of scope or adding features beyond the original requirements could affect deadlines and budget.
3. Technology Failures: Technical issues or failure of any chosen software or tools could cause delays in development and deployment.
4. Integration Issues: Difficulty in integrating the online store with external systems such as payment gateways, inventory systems, or CRM platforms could disrupt the project flow.
5. Team Dependency: Over-dependence on key individuals for specific tasks could cause delays if they are unavailable or leave the project unexpectedly.
6. Quality Assurance Failures: Insufficient testing or overlooking critical bugs could result in poor user experience or issues after the launch, affecting brand reputation.
7. Client Stakeholder Delays: Delays in receiving feedback or approvals from client representatives or other stakeholders may result in project timeline setbacks.
8. Cybersecurity Threats: The online store may be vulnerable to security threats like data breaches or cyberattacks, affecting the project's success and user trust.

**Resource Planning:**

**1. Human Resources:**

* **Project Manager (PM)**: Essential to oversee the entire project, ensure adherence to timelines, and manage the team.
  + **Salary Estimate**: ₹10-15 Lakhs per annum
* **Business Analysts (BA)**: 2 to 3 BAs to gather requirements and document business processes, supporting project execution.
  + **Salary Estimate**: ₹8-12 Lakhs per annum per person
* **System Architect**: To design the overall system architecture and ensure scalability.
  + **Salary Estimate**: ₹12-18 Lakhs per annum
* **Development Team**: Front-end and back-end developers, including 1 or 2 senior developers and 2-3 junior developers.
  + **Salary Estimate**: ₹10-20 Lakhs per annum per senior developer and ₹6-10 Lakhs per junior developer
* **UX/UI Designers**: For creating an intuitive and user-friendly experience, including 1-2 designers.
  + **Salary Estimate**: ₹8-12 Lakhs per annum per designer
* **Quality Assurance (QA)**: 2-3 testers for performing quality checks and ensuring bug-free releases.
  + **Salary Estimate**: ₹5-8 Lakhs per annum per QA specialist
* **Content Team**: Writers, SEO specialists, and content strategists for maintaining the online store's content.
  + **Salary Estimate**: ₹6-8 Lakhs per annum per person
* **Support Staff**: For administrative tasks, customer support, etc.
  + **Salary Estimate**: ₹3-5 Lakhs per annum per person

**2. Technology Costs:**

* **Software Licenses/Tools**: MS Office, project management tools (e.g., Jira), design tools (e.g., Figma, Adobe), and development environments.
  + **Estimated Cost**: ₹10-20 Lakhs for the entire duration
* **Server and Hosting Costs**: Cloud storage, servers for staging/production environments.
  + **Estimated Cost**: ₹10-15 Lakhs for 18 months
* **Security Tools/Firewalls**: To ensure secure transactions and data protection.
  + **Estimated Cost**: ₹5-10 Lakhs

**3. Training and Development:**

* **Training for Staff**: In case new tools or technologies are introduced, some funds will be allocated for training the team.
  + **Estimated Cost**: ₹2-5 Lakhs

**4. Miscellaneous Expenses:**

* **Contingency Fund**: For unforeseen expenses, last-minute changes, or additional resources required during the project.
  + **Estimated Cost**: ₹10-15 Lakhs
* **Travel and Miscellaneous Costs**: For any team meetings, conferences, or customer-related travel.
  + **Estimated Cost**: ₹5-8 Lakhs

**Estimated Breakdown:**

|  |  |
| --- | --- |
| **Resource/Expense** | **Estimated Cost Range (₹)** |
| Project Manager Salary | 10-15 Lakhs |
| Business Analysts (2-3) | 16-36 Lakhs |
| System Architect Salary | 12-18 Lakhs |
| Development Team (5-6) | 50-80 Lakhs |
| UX/UI Designers (1-2) | 8-24 Lakhs |
| Quality Assurance Team (2-3) | 10-24 Lakhs |
| Content Team (2-3) | 12-24 Lakhs |
| Support Staff | 6-10 Lakhs |
| Software/Tools/License | 10-20 Lakhs |
| Server/Hosting | 10-15 Lakhs |
| Security Tools/Firewalls | 5-10 Lakhs |
| Training & Development | 2-5 Lakhs |
| Miscellaneous/Contingency | 15-20 Lakhs |
| **Total Estimated Budget** | **₹2 Crores** |

**Timeline:**

**The 18-month timeline can be divided into phases:**

1. **Initiation and Planning (1-2 months):**
   * Resource allocation, finalizing project scope, business requirements, and detailed project planning.
2. **Design Phase (3-4 months):**
   * System architecture design, wireframing, UX/UI design, and approval from stakeholders.
3. **Development Phase (6-8 months):**
   * Core development of the online store, including front-end and back-end functionality, integration with third-party services, etc.
4. **Testing and Quality Assurance (2-3 months):**
   * Testing for bugs, user experience, functionality, and security.
5. **Deployment and Go-Live (1 month):**
   * Final testing, deployment to production, and user training.
6. **Post-Launch Support and Maintenance (1 month):**
   * Fixing bugs or issues reported by users after launch, and providing ongoing support.

**Software Requirements Specification (SRS) for Online Store**

**1. Introduction**

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for the development of an online store system. The system will allow users to browse, purchase, and manage orders for products online.

**1.1 Purpose**

The purpose of this SRS is to define the features, functionalities, and behavior of the online store application. It will provide a comprehensive guideline for developers and stakeholders to understand the system's requirements.

**1.2 Scope**

The online store will provide a platform for users to purchase products in different categories. It will include user registration, product catalog, shopping cart, order management, and payment gateway integration.

**1.3 Definitions, Acronyms, and Abbreviations**

* Admin: The system administrator responsible for managing the online store, users, and product catalog.
* User/Customer: A person who browses and purchases items from the store.
* Product Catalog: The list of products available for purchase in the online store.
* Shopping Cart: A temporary storage for selected products before proceeding to checkout.
* Payment Gateway: A system used to process transactions securely.

**2. System Overview**

The online store will allow users to view and search for products, add them to the cart, and complete the purchase via a secure payment system. Admins will manage product listings, orders, and customer accounts. The platform will support multiple payment options and provide order tracking.

**3. Functional Requirements**

**3.1 User Management**

* 3.1.1 User Registration: Users should be able to register by providing an email, phone number, and password.
* 3.1.2 User Login/Logout: Registered users can log in to the platform using their credentials or social media accounts.
* 3.1.3 Profile Management: Users can view and update their profile information, such as name, shipping address, and password.
* 3.1.4 Forgot Password: Users can reset their password via email or phone number.

**3.2 Product Catalog Management**

* 3.2.1 Product Search: Users can search for products using categories, filters, and keywords.
* 3.2.2 Product Details: Each product will have a detailed page displaying its description, price, images, and reviews.
* 3.2.3 Product Availability: The system will show real-time stock availability.
* 3.2.4 Add to Cart: Users can add selected products to their shopping cart.

**3.3 Shopping Cart & Checkout**

* 3.3.1 View Cart: Users can view the products in their cart with details such as quantity, price, and total.
* 3.3.2 Modify Cart: Users can change quantities or remove products from the cart.
* 3.3.3 Checkout: Users can proceed to checkout and enter shipping and payment details.
* 3.3.4 Order Confirmation: Once the purchase is successful, an order confirmation will be displayed with the order details.

**3.4 Payment Processing**

* 3.4.1 Payment Gateway Integration: The system will integrate with third-party payment gateways (e.g., PayPal, Stripe) for secure payments.
* 3.4.2 Multiple Payment Options: The system will support credit cards, debit cards, and mobile wallets.

**3.5 Order Management**

* 3.5.1 Order Tracking: Users can track the status of their order (e.g., processing, shipped, delivered).
* 3.5.2 Order History: Users can view their past orders and repeat purchases.

**3.6 Admin Panel**

* 3.6.1 Product Management: Admins can add, update, or remove products from the catalog.
* 3.6.2 Order Management: Admins can view and manage customer orders, including status updates and refunds.
* 3.6.3 Customer Management: Admins can view customer profiles, including order history and contact information.

**4. Non-Functional Requirements**

**4.1 Performance**

* The system should be able to handle at least 1,000 simultaneous users.
* The product pages and checkout process should load within 3 seconds.

**4.2 Security**

* All user data, including payment details, should be encrypted.
* Secure authentication mechanisms (e.g., 2-factor authentication) should be implemented for sensitive operations.

**4.3 Usability**

* The platform should have an intuitive and user-friendly interface.
* It should be responsive and accessible on both desktop and mobile devices.

**4.4 Reliability**

* The system should have 99% uptime and support automatic backups for data recovery.

**4.5 Scalability**

* The system should be able to scale horizontally to handle more users and product listings as the business grows.

**5. Assumptions and Dependencies**

* The platform will require a stable internet connection for both users and admins.
* Payment gateway services will be available and functional.

**6. External Interface Requirements**

**6.1 User Interface**

* The user interface will be designed using HTML, CSS, and JavaScript.
* The UI will be mobile-responsive and accessible via modern browsers.

**6.2 Hardware Interface**

* The platform will run on cloud servers (e.g., AWS, Azure) or on-premises servers with sufficient resources to handle the load.

**7. System Features**

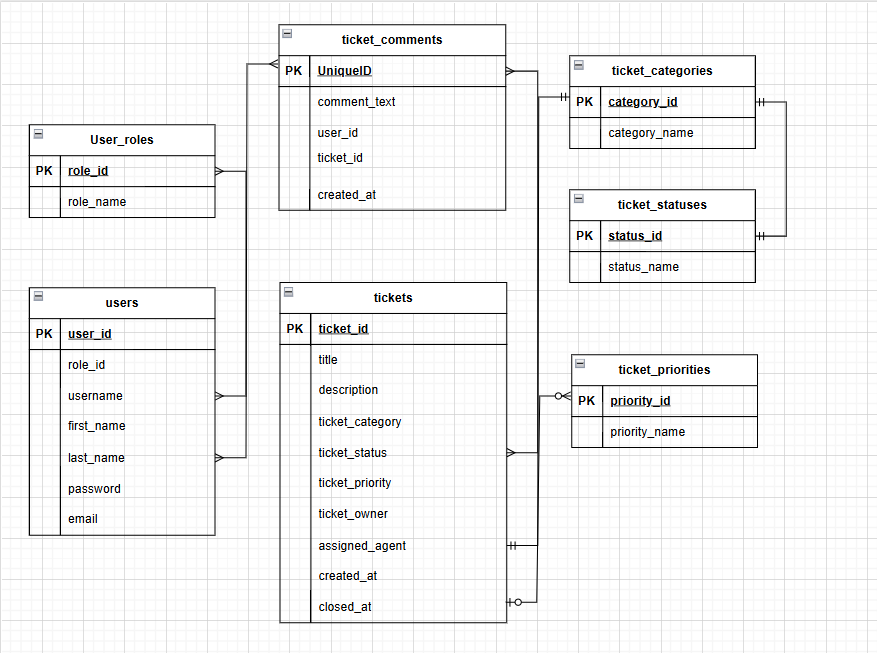
* 7.1 Search Functionality: Users can search for products by name, category, or keywords.
* 7.2 Product Recommendations: The system will recommend similar or popular products based on user behavior.
* 7.3 Reviews and Ratings: Users can leave ratings and reviews for purchased products.
* 7.4 Wishlist: Users can save products to a Wishlist for future purchase.

**8. Glossary**

* E-commerce: The process of buying and selling goods or services over the internet.
* Frontend: The part of the system that users interact with (UI).
* Backend: The server-side logic and database of the system.

**3. Make an ERD of creating a support ticket/Ticketing life cycle.**

**ER Diagram of Support Ticket:**

****

This ER diagram represents a comprehensive database schema for a ticketing system, involving users, roles, tickets, ticket categories, statuses, priorities, and comments. Below is a detailed explanation of the entities and their relationships:

**Entities and Attributes:**

1. **Users Table:**
   * user\_id (PK): A unique identifier for each user.
   * role\_id (FK): A foreign key linking the user to the User\_roles table, indicating the user's role.
   * username: The login username of the user.
   * first\_name: The user's first name.
   * last\_name: The user's last name.
   * password: The user's password for authentication.
   * email: The user's email address.

**Description:** This table stores the information of all users in the system, such as their personal details and their role, which determines their access permissions within the system.

1. **User\_roles Table:**
   * role\_id (PK): A unique identifier for each user role.
   * role\_name: The name of the role (e.g., "Admin", "Agent", "Customer").

**Description:** This table defines the roles assigned to users. It helps in managing access control by specifying what each role can do within the system (e.g., admin can manage tickets, agents can respond to them, etc.).

1. **Tickets Table:**
   * ticket\_id (PK): A unique identifier for each ticket.
   * title: The title or summary of the ticket.
   * description: A detailed description of the issue or request raised in the ticket.
   * ticket\_category (FK): A foreign key referencing ticket\_categories, specifying the category of the ticket (e.g., "Technical Issue", "Billing Query").
   * ticket\_status (FK): A foreign key referencing ticket\_statuses, indicating the current status of the ticket (e.g., "Open", "In Progress", "Closed").
   * ticket\_priority (FK): A foreign key referencing ticket\_priorities, denoting the priority level of the ticket (e.g., "High", "Medium", "Low").
   * ticket\_owner (FK): A foreign key referencing the users table. This field indicates the user (typically an admin or customer) who created the ticket.
   * assigned\_agent: The user (typically an agent) who is assigned to work on the ticket.

**Description:** The tickets table holds all the information related to individual tickets raised within the system. It connects to other tables like categories, statuses, and priorities to track the ticket's journey through the system.

1. **Ticket\_categories Table:**
   * category\_id (PK): A unique identifier for each ticket category.
   * category\_name: The name of the category (e.g., "Technical", "Billing").

**Description:** This table categorizes tickets, helping to classify them into different types (e.g., "Technical Issue", "Service Request"). It ensures tickets are organized according to their nature and allows for better filtering and tracking.

1. **Ticket\_statuses Table:**
   * status\_id (PK): A unique identifier for each ticket status.
   * status\_name: The name of the ticket status (e.g., "Open", "In Progress", "Closed").

**Description:** This table maintains the possible statuses of tickets. It is used to track the current state of each ticket in the system. For example, a ticket may move through statuses such as "Open", "In Progress", and "Resolved".

1. **Ticket\_priorities Table:**
   * priority\_id (PK): A unique identifier for each priority level.
   * priority\_name: The name of the priority level (e.g., "High", "Medium", "Low").

**Description:** This table defines different priority levels for tickets. It helps in determining how urgently a ticket should be addressed by the team.

1. **Ticket\_comments Table:**
   * UniqueID (PK): A unique identifier for each comment.
   * comment\_text: The text content of the comment.
   * user\_id (FK): A foreign key referencing the users table, indicating the user who made the comment.
   * ticket\_id (FK): A foreign key referencing the tickets table, linking the comment to a specific ticket.
   * created\_at: The timestamp when the comment was made.

**Description:** This table stores all comments made on tickets. Each comment is linked to a ticket (via ticket\_id) and a user (via user\_id). Comments allow users to add notes, updates, or feedback on tickets, enabling communication between the ticket owner, assigned agents, and other relevant users.

**Relationships:**

1. **Users to User\_roles (One-to-Many):** Each user is associated with one role (e.g., “Admin”, “Agent”), but each role can be assigned to many users. This is represented by the role\_id foreign key in the users table.
2. **Tickets to Ticket\_categories, Ticket\_statuses, Ticket\_priorities (Many-to-One):**
   * Each ticket belongs to one category, one status, and one priority.
   * The ticket\_category, ticket\_status, and ticket\_priority fields in the tickets table reference their respective foreign keys in the ticket\_categories, ticket\_statuses, and ticket\_priorities tables.
3. **Tickets to Users (One-to-Many):**
   * Each ticket is created by one user (the ticket\_owner) and can be assigned to one or more agents (via the assigned\_agent field).
   * The ticket\_owner field in the tickets table is a foreign key that references the users table, while assigned\_agent is also a foreign key linking to the same users table.
4. **Ticket\_comments to Tickets and Users (Many-to-One):**
   * Each comment is linked to one ticket and one user.
   * The ticket\_id in the ticket\_comments table references the tickets table, while the user\_id references the users table.

**Normalization & Design:**

* **Normalization:** The diagram follows normalization principles to eliminate redundancy and ensure data integrity. Each entity is stored separately with relationships managed via foreign keys.
* **Referential Integrity:** Foreign key constraints ensure that data is consistent across related tables (e.g., a comment must belong to an existing ticket, and a ticket owner must be an existing user).

**Conclusion:**

This database schema is designed to manage a ticketing system effectively. It allows for user management, ticket categorization, status tracking, priority setting, and commenting, making it robust for real-world ticket management scenarios.