**Assignment 1:**

**Business Requirements Document (BRD)**

Project Title: Inventory Management and Quick Delivery System for Dairy Products

Date: [11/JAN/2025]

Prepared By: [Rutuja Madnoorkar]

**Introduction:** The dairy industry is a vital component of the global food supply chain, providing essential nutrition to millions of people. Efficient inventory management and timely delivery of dairy products are critical due to their perishable nature. This project aims to address these challenges by designing an advanced Inventory Management and Quick Delivery System specifically tailored for the dairy industry.

**Document Purpose:** This document outlines the business requirements for the proposed system. It serves as a comprehensive guide for stakeholders, project team members, and decision-makers, ensuring a shared understanding of the project’s goals, scope, and deliverable. By defining these requirements, we aim to facilitate a seamless implementation of the system and achieve the desired business outcomes.

**Intended Audience:**

Internal Stakeholders: Project managers, business analysts, IT team members, and operational staff.

External Stakeholders: Dairy suppliers, distributors, and logistics partners.

Decision-Makers: Senior management, financial officers, and product development leads.

**Project Background:** Dairy products are highly perishable, requiring robust systems to manage inventory and ensure timely deliveries. Traditional systems often fail to account for the specific needs of dairy products, such as temperature control and expiration tracking. This project was initiated to address these gaps by leveraging modern technology and data analytics to create a system that enhances operational efficiency and customer satisfaction.

**Purpose of the Business Requirements The purpose of documenting business requirements is to:**

Provide a clear understanding of the system’s scope and functionalities.

Align stakeholder expectations with project deliverable.

Serve as a reference for system design, development, and implementation.

Ensure compliance with regulatory standards for dairy product handling.

**Business Goals/Objectives to be Achieved:**

Implement a real-time inventory management system to reduce waste and optimize stock levels.

Develop a quick delivery mechanism to ensure on-time delivery of dairy products.

Integrate temperature and quality monitoring to maintain product integrity.

Enhance customer satisfaction by providing accurate delivery updates and improved service reliability.

**Benefits/Rationale:**

Operational Efficiency: Streamlined inventory processes reduce manual errors and save time.

Cost Savings: Minimized product wastage and optimized delivery routes lower operational costs.

Customer Trust: Consistent quality and timely delivery foster customer loyalty.

Regulatory Compliance: Automated tracking ensures adherence to industry standards.

**Stakeholders:**

Primary Stakeholders: Dairy farmers, distributors, and end consumers.

Secondary Stakeholders: IT service providers, regulatory authorities, and logistics partners.

**Dependencies on Existing Systems:**

Integration with current inventory and ERP systems.

Compatibility with existing logistics and supply chain software.

Data exchange with quality control and temperature monitoring tools.

**References:**

Industry guidelines for dairy product handling and transportation.

Current inventory management software documentation.

Case studies on successful delivery systems in the dairy industry.

**Assumptions:**

All stakeholders will actively participate in the project phases.

Necessary funding and resources will be made available.

The existing IT infrastructure can support the new system with minimal upgrades.

End users will receive adequate training for system adoption.

**Requirements Scope:**

The Inventory Management and Quick Delivery System aims to optimize the management, tracking, and delivery of perishable dairy products. This system ensures inventory accuracy, reduces waste due to spoilage, and enhances customer satisfaction through quick and efficient delivery.

**In Scope:**

1. Inventory Management:

Real-time tracking of stock levels for all dairy products.

Automated alerts for low stock levels and nearing expiry dates.

Batch tracking for product recalls or quality control.

Temperature monitoring for storage compliance.

1. Order Management:

Integration with customer-facing platforms for real-time order placement.

Handling of bulk and individual orders.

Order prioritization based on delivery time frames and product perish ability.

1. Quick Delivery System:

Route optimization for faster delivery.

Real-time delivery tracking for customers.

Integration with third-party logistics providers for extended delivery networks.

1. Reporting and Analytics:

Dashboards for sales, inventory trends, and delivery performance.

Analytics for demand forecasting and supply chain optimization.

1. System Integration:

Integration with existing Enterprise Resource Planning (ERP) systems.

Compatibility with Point of Sale (POS) systems for seamless updates.

API connections for third-party applications.

**Out of Scope:**

1. Non-Dairy Products:

Managing inventory or delivery for non-dairy items.

1. Delivery Vehicle Management:

Maintenance and servicing schedules of delivery vehicles.

1. Customer Relationship Management (CRM):

Advanced CRM functionalities, such as loyalty programs or targeted marketing campaigns.

1. In-House Logistics Infrastructure:

Development of physical delivery infrastructure (e.g., warehouses or vehicles).

1. Regulatory Compliance Framework:

Legal advisory services for food safety compliance or certifications.

1. International Shipping:

Delivery system functionalities for cross-border operations.

**Development Plan:**

 Phase 1: Requirement Gathering (2 weeks)

Phase 2: System Design (3 weeks)

Phase 3: Development (12 weeks)

Phase 4: Testing (4 weeks)

Phase 5: Deployment and Training (2 weeks)

**Resource Plan**:

Team Structure:

Project Manager: 1

Business Analyst: 1

UI/UX Designers: 2

Back-end Developers: 3

Frontend Developers: 2

QA Engineers: 2

DevOps Engineer: 1

Technical Support Staff: 2

**Key Tools and Technologies:**

Front-end: React.js or Angular.

Backend: Node.js, Python (Django/Flask).

Database: PostgreSQL.

Cloud Platform: Azure.

Integration Tools: API s for GPS and mapping services.

Testing Tools: JIRA.

**Risks and Mitigation:**

Risk: Delays in requirement finalization.

Mitigation: Early stakeholder alignment and iterative reviews.

Risk: Integration challenges with existing systems.

Mitigation: Detailed API documentation and sandbox testing.

Risk: Data security concerns.

Mitigation: Regular security audits and compliance adherence.

**Success Criteria:**

Reduction in inventory wastage by 20% within the first six months.

Improved delivery time by 30% within the first year.

Positive user feedback with a satisfaction score of 85% or higher.

Accurate real-time reporting with minimal errors.

**Functional Requirements:**

1. Inventory Management:

Real-time stock updates.

Alerts for low stock levels.

Categorization of dairy products by type, size, and expiration date.

Batch tracking for quality control.

1. Order Processing:

Quick order entry and processing.

Real-time order tracking for customers.

Automated invoice generation.

1. Delivery Management:

Route optimization for deliveries.

Real-time delivery status updates.

Delivery confirmation through customer signature.

1. Customer Management:

Customer account creation and management.

Maintenance of customer order history.

Integration with customer feedback mechanisms.

1. Reporting and Analytics:

Generation of sales, stock, and delivery reports.

Insights on customer preferences and trends.

Expiry date-based stock reports.

1. System Integration:

Integration with payment gateways.

Integration with third-party logistics systems.

API support for external tools.

**Actor Profiles Specification:**

1. Inventory Manager:

Manages stock levels and tracks product batches.

Monitors expiry dates and restocking requirements.

1. Delivery Personnel:

Handles the delivery of orders.

Updates the system with delivery statuses.

1. Customer:

Places and tracks orders.

Provides feedback on products and services.

1. System Administrator:

Manages user roles and permissions.

Ensures the system's smooth functioning.

1. Sales Representative:

Assists customers with order placement.

Provides support for inquiries and issues.

**Essential Use Case Diagram:**



**Essential Use Case Specifications:**

1. Use Case: Manage Inventory

Actor: Inventory Manager

Precondition: Inventory Manager is logged in.

Steps:

Access inventory dashboard.

View current stock levels and batch details.

Update stock levels as required.

Set alerts for low stock levels.

Post condition: Inventory data is updated in the system.

1. Use Case: Process Order

Actor: Sales Representative / Customer

Precondition: Customer account exists.

Steps:

Select dairy products and add to cart.

Confirm order details.

Make payment via the integrated gateway.

Receive order confirmation.

Post condition: Order is placed and visible in the system.

1. Use Case: Manage Delivery

Actor: Delivery Personnel

Precondition: Delivery task is assigned.

Steps:

View assigned deliveries.

Update delivery status en route.

Collect customer signature upon delivery.

Confirm completion in the system.

Post condition: Delivery status is updated.

**Function Definition Report:**

1. Inventory Management

Real-time tracking and alerts for stock updates.

Expiry-based reporting and batch tracking.

1. Order Processing:

Efficient entry, tracking, and confirmation of orders.

Automated invoicing and payment handling.

1. Delivery Management:

Streamlined routing and live delivery updates.

Verification and confirmation mechanisms.

1. Customer Management:

Tools for account and feedback management.

Historical data analysis for improved service.

1. Reporting and Analytics:

Visualizations and insights to aid decision-making.

Expiry and trend-based reporting.

#### Business Rules:

|  |  |  |  |
| --- | --- | --- | --- |
| **Rule ID** | **Description** | **Applies To** | **Action/Outcome** |
| BR-01 | Stock levels below threshold trigger alerts. | Inventory Management | Notification sent to Inventory Mgr |
| BR-02 | Orders are confirmed only after payment. | Order Processing | Order status updated to "Confirmed" |
| BR-03 | Deliveries require customer signature. | Delivery Management | Delivery status set to "Complete" |
| BR-04 | Customer feedback updates satisfaction metrics. | Customer Management | Metrics updated for reporting |
| BR-05 | Reports auto-generated every month. | Reporting and Analytics | Reports emailed to admin |

**Data Requirements:**

Identifying the right data requirements is essential to ensure efficient operations and customer satisfaction.Below is a breakdown of potential data requirements:

1. Product Data:

Product ID: Unique identifier for each product.

Product Name: Name of the dairy product (e.g., milk, yogurt, cheese).

Category: Type of dairy product (e.g., fresh milk, fermented, cream-based).

Batch Number: Identification for production batches.

Expiry Date: Critical for ensuring freshness and avoiding spoilage.

Stock Quantity: Current stock levels for each product.

Storage Requirements: Temperature and handling guidelines.

Supplier Details: Information about the supplier or producer.

2. Inventory Data:

Warehouse ID: Unique identifier for each storage location.

Warehouse Location: Address and facility details.

Storage Capacity: Maximum inventory capacity of each warehouse.

Current Stock Levels: Real-time data on stock availability.

Restock Alerts: Threshold levels for triggering restocking actions.

Stock Movement Logs: Records of stock addition, removal, or transfer.

Wastage Data: Details on spoiled or expired items.

3. Order Management Data:

Order ID: Unique identifier for customer orders.

Customer Details: Name, contact information, and delivery address.

Order Date: When the order was placed.

Delivery Date: Scheduled delivery time.

Order Status: Pending, in process, dispatched, delivered, or canceled.

Order Items: List of products and quantities in each order.

Payment Details: Payment method, status, and transaction details.

Special Instructions: Notes for delivery (e.g., contact less delivery).

4. Delivery Data:

Delivery ID: Unique identifier for delivery transactions.

Vehicle Details: Vehicle type, capacity, and registration number.

Delivery Personnel: Names and contact details of drivers.

Route Optimization Data: Preferred and alternate delivery routes.

Delivery Time Estimates: Expected time for order completion.

Delivery Confirmation: Customer signature or digital acknowledgment.

Real-Time Tracking: GPS tracking data for live monitoring.

5. Supplier and Procurement Data:

Supplier ID: Unique identifier for suppliers.

Supplier Name: Contact and organization details.

Product Catalog: List of products offered by each supplier.

Purchase Orders: Records of orders placed to suppliers.

Lead Times: Estimated time from ordering to delivery.

Payment Terms: Billing cycles and conditions.

6. Customer Data:

Customer ID: Unique identifier for each customer.

Customer Preferences: History of purchases and preferred products.

Feedback and Ratings: Data from reviews to improve services.

Loyalty Program Details: Points or discounts earned by customers.

Communication Logs: Emails, messages, and complaints.

**Data Conversion:**

Implementing a data conversion process for an Inventory Management and Quick Delivery System for Dairy Products involves several steps to ensure seamless integration between legacy systems (if any) and the new system. The process is particularly critical due to the perishable nature of dairy products and the need for accurate inventory tracking and timely delivery.

Key Aspects of Data Conversion for Dairy Products

1. Data Categories to Convert:

Inventory Data:

Product details (name, SKU, category, weight/volume, shelf life, etc.).

Stock levels (current quantities, storage locations, expiration dates).

Supplier information (name, contact, product list).

Delivery Data:

Delivery routes, schedules, and areas.

Customer details (name, address, contact, order history).

Delivery partner details (drivers, vehicles, etc.).

Sales and Order Data:

Historical sales records.

Open orders and backorders.

Pricing, discounts, and offers.

Operational Data:

Warehouse details (zones, temperature requirements, etc.).

Performance metrics (order fulfillment rates, delivery times).

2. Source and Target Systems:

Source System: Legacy inventory systems, Excel sheets, or manual records.

Target System: New inventory management and delivery system with integrated features like real-time stock updates and route optimization.

3. Data Mapping

Map each field in the source data to its corresponding field in the new system. For example:

Source: Product\_ID → Target: SKU\_Code

Source: Quantity\_on\_Hand → Target: Stock\_Level

Source: Expiry\_Date → Target: Expiration\_Date

4. Data Conversion Process

1. Data Extraction:

Extract data from legacy systems, databases, or spreadsheets.

1. Data Cleaning:

Remove duplicate records.

Fix inconsistencies (e.g., uniform date formats, correct units like liters/kilograms).

Validate product expiration dates.

1. Data Transformation:

Convert formats (e.g., Excel to database-friendly formats like CSV).

Standardize units of measure.

Encode temperature-sensitive storage requirements (e.g., cold storage flags).

1. Data Loading:

Load transformed data into the new system.

1. Testing and Validation:

Reconcile inventory counts and delivery schedules.

Run sample orders to test the system's functionality.

5. Conversion Considerations

* Real-Time Updates: Dairy products often have short shelf lives. Ensure inventory data is updated in real-time to avoid waste.
* Integration with IoT Sensors:

Connect warehouse temperature sensors to the system for cold chain monitoring.

* Batch Management:

Track products by batch numbers for efficient recalls if needed.

* Customer Data Privacy:

Encrypt sensitive data to comply with regulations (e.g., GDPR).

**Data Retention and Archiving:**

Data Retention and Archiving involve storing data for a specified period and preserving historical information for compliance, business analysis, or reference. These practices are essential for regulatory compliance, efficient resource management, and ensuring operational continuity.

Key Concepts:

1. Data Retention: The practice of keeping data for a defined period based on legal, regulatory, or business needs. After this period, data is often deleted or archived.
2. Data Archiving: The process of moving inactive or infrequently used data to a separate storage location for long-term preservation. Archived data can be accessed when needed but is not actively used in day-to-day operations.

**FOI/Privacy Implications:**

Freedom of Information (FOI) and Privacy Implications are critical considerations when managing data retention, archiving, and accessibility. These concepts are particularly relevant in public sector organizations, private enterprises handling sensitive personal data, and industries bound by strict regulatory frameworks.

Key Concepts:

1. Freedom of Information (FOI):

FOI laws grant individuals or organizations the right to access certain types of information held by public bodies.

Examples: The Freedom of Information Act 2000 (UK) and the Freedom of Information Act 1966 (USA).

Implications:

Organizations must retain data that may be subject to FOI requests.

Ensuring timely retrieval and response to requests is crucial.

1. Privacy Implications:

Privacy laws regulate how organizations collect, store, use, and share personal data.

Examples: General Data Protection Regulation (GDPR) (EU), California Consumer Privacy Act (CCPA) (USA).

Implications:

Organizations must balance transparency with the need to protect sensitive data.

Retention and archiving policies must comply with privacy obligations.

**Privacy Considerations:**

1. Data Minimization
2. Consent Management
3. Access Control
4. Breach Management

**Data Definition Report: Overview**

A Data Definition Report provides a structured overview of the Data Architecture and the definitions of data used within a system.

Sections of a Data Definition Report

1. Introduction:

Purpose of the report.

Scope of the data architecture being defined.

Stakeholders (e.g., IT teams, business analysts, data stewards).

2. Data Architecture Overview:

High-level description of the system’s data model.

Explanation of the architectural approach (e.g., relational, NoSQL, data lake, etc.).

Diagram illustrating the key entities and relationships.

3. Data Entities and Attributes:

Detailed table or description of each data entity, including:

Entity Name: Unique name of the data entity.

Description: Brief explanation of the entity’s purpose.

Attributes: List of attributes/fields, including:

Attribute Name.

Data Type (e.g., string, integer, date).

Format (e.g., YYYY-MM-DD for dates).

Length/Size (e.g., maximum character length for strings).

Constraints (e.g., mandatory, unique).

Primary Key: The unique identifier for the entity.

Foreign Keys: Attributes linking the entity to others.

4. Data Relationships:

Description of how entities are related, including:

One-to-One: E.g., a user profile and login credentials.

One-to-Many: E.g., a customer and their orders.

Many-to-Many: E.g., students and courses in a university system.

Entity-Relationship (ER) Diagram to visually represent relationships.

5. Business Rules:

List of rules that govern data usage, such as:

Validations (e.g., email must have an "@" symbol).

Data consistency requirements (e.g., inventory stock cannot be negative).

Retention policies (e.g., retain orders for 7 years).

**Domain Class Definition Report:**

This section provides a narrative description of the Domain Class Model in the context of a Use Case-driven approach. The Domain Class Definition Report outlines the core classes, their attributes, relationships, and roles within the system’s architecture, using narrative text to clarify how they interact to fulfill the specified use cases.

Domain Class Model: Narrative Definitions

1. Customer:

Description:

The Customer class represents individuals or organizations purchasing products or services. This class is central to all customer-facing use cases, such as placing orders, viewing account details, and tracking deliveries.

Attributes:

Customer\_ID (Integer): Unique identifier for the customer.

Name (String): Full name of the customer.

Email (String): Contact email address.

Phone\_Number (String): Contact phone number.

Address (String): Physical address for delivery.

Relationships:

One-to-Many with Order: A customer can place multiple orders.

One-to-Many with Feedback: A customer can provide feedback on multiple products.

2. Product:

Description:

The Product class represents items available for purchase, including their characteristics and stock levels.

Attributes:

Product\_ID (String): Unique identifier for the product (e.g., SKU).

Name (String): Name of the product.

Category (String): Classification of the product (e.g., Dairy, Beverages).

Price (Decimal): Cost of the product.

Stock\_Quantity (Integer): Number of items available in inventory.

Relationships:

One-to-Many with Order\_Item: A product can be associated with multiple order items.

3. Order:

Description:

The Order class represents a purchase transaction initiated by a customer.

Attributes:

Order\_ID (Integer): Unique identifier for the order.

Order\_Date (DateTime): Date and time when the order was placed.

Status (String): Current status of the order (e.g., Pending, Delivered).

Relationships:

Many-to-One with Customer: Each order is linked to one customer.

One-to-Many with Order\_Item: An order can include multiple order items.

4. Order\_Item:

Description:

The Order\_Item class represents the individual items included in an order.

Attributes:

Order\_Item\_ID (Integer): Unique identifier for the order item.

Quantity (Integer): Number of units of the product in the order.

Subtotal (Decimal): Calculated subtotal for the line item (Quantity × Product Price).

Relationships:

Many-to-One with Order: Each order item is linked to one order.

Many-to-One with Product: Each order item is linked to one product.

5. Delivery:

Description:

The Delivery class tracks the process of delivering orders to customers.

Attributes:

Delivery\_ID (Integer): Unique identifier for the delivery.

Delivery\_Date (Date-time): Date and time when the order is delivered.

Status (String): Status of the delivery (e.g., Scheduled, In Transit, Completed).

Relationships:

One-to-One with Order: Each delivery is linked to one order.

**Entity Definition Report:**

The Entity Definition Report provides a comprehensive description of the data entities within a system.

Sections of an Entity Definition Report

1. Introduction:

Purpose of the report.

Scope: Details which data entities are being defined, and the systems or processes they represent.

Audience: Intended readers, such as data architects, developers, business users, or auditors.

2. Data Entity Overview:

Entity Name: The unique identifier of the entity.

Description: A brief explanation of what the entity represents and its purpose within the system.

Scope: Clarifies if the entity applies to internal, external, or operational data flows.

3. Entity Attributes:

Attribute Name: The name of the attribute/column.

Data Type: Defines the type of data (e.g., String, Integer, DateTime, Boolean).

Length: Character length or maximum value if applicable (e.g., for strings).

Format: Any format restrictions (e.g., date format YYYY-MM-DD, phone number format XXX-XXX-XXXX).

Primary Key: The unique identifier for the entity.

Foreign Keys: Attributes linking to other entities (in relationships).

Constraints:

Mandatory: Whether the attribute must have a value.

Unique: Whether the value of the attribute must be unique across all records.

Default Value: Any default value applied if none is provided.

4. Entity Relationships:

Relationship Type: One-to-One, One-to-Many, or Many-to-Many.

Target Entity: The entity this relationship connects to.

Description: An explanation of the relationship and how it supports business processes.

Cardinality: Number of records this entity can have in relation to another (e.g., one Customer can place many Orders).

5. Business Rules:

Detailed business rules that govern the usage of data within the entity:

Validation Rules: E.g., data must not be blank, email must be in valid format.

Data Consistency Rules: E.g., product stock cannot go below zero.

Retention Policies: Data must be kept for 7 years for compliance.

6. Entity Security:

Description of data sensitivity and access control policies:

Data Classification: Internal, Confidential, Public.

Access Levels: Who can read, write, update, or delete this data.

Encryption: Details on encryption algorithms, if applicable.

7. Data Sources:

Source System: Where the data originates (e.g., transactional systems, external APIs, IoT devices).

Source Description: Explanation of how the data is gathered or entered into the system.

8. Entity Usage:

Business Use Cases: How this entity supports specific processes or workflows.

System Use: Description of how the entity is accessed and utilized in system processes (e.g., during order processing, customer management, etc.).

**Non-Functional requirements:**

Non-functional requirements define the system's operational qualities and constraints, focusing on how the system performs rather than what it does. These requirements are critical for ensuring usability, scalability, reliability, and maintainability.

Categories of Non-Functional Requirements

1. Performance Requirements:

2. Scalability

3. Reliability

4. Availability

5. Security

6. Maintainability

7. Usability

8. Interoperability

9. Compliance

10. Data Retention and Archiving

11. Localization

12. Audit and Monitoring:

**Security Requirements:**

1. Authentication and Authorization:

Authentication:

All users must authenticate using a secure mechanism, such as:

Multi-Factor Authentication (MFA) for all user accounts, combining a password and an OTP sent via SMS/email or generated through an authenticator app.

Passwords must adhere to a strong password policy:

Minimum of 12 characters, including uppercase, lowercase, numbers, and special characters.

Passwords must expire every 90 days.

Previous 5 passwords cannot be reused.

Use Single Sign-On (SSO) where applicable to streamline authentication processes securely.

Authorization:

Role-based Access Control (RBAC) must ensure users can only access resources and actions within their defined roles.

Privileged access must be limited to administrators and audited regularly.

2. Data Protection:

Data Encryption:

All sensitive data at rest must be encrypted using AES-256.

Data in transit must be encrypted using TLS 1.3 or higher.

Personal Data Masking:

Personal Identifiable Information (PII) must be masked in logs, reports, and user interfaces unless explicitly authorized.

Data Integrity:

All data modifications must be logged to ensure traceability.

Mechanisms to detect and prevent tampering must be implemented, such as check sums and digital signatures.

3. Access Control:

Principle of Least Privilege:

Users and systems must only have access to the minimum resources required to perform their functions.

Session Management:

User sessions must automatically time out after 15 minutes of inactivity.

Concurrent sessions must be limited to prevent account abuse.

Administrative Access:

All administrative actions must require additional authentication, such as re-entering credentials or MFA.

4. Audit and Monitoring:

Logging:

Maintain audit logs for all critical actions, including logins, data access, modifications, and deletions.

Logs must include details such as timestamps, user IDs, IP addresses, and the nature of the action.

Monitoring:

Implement real-time monitoring tools to detect and respond to anomalies, such as unauthorized access attempts or unusual activity patterns.

Alerts must be sent to administrators for high-severity events within 5 minutes.

Log Retention:

Logs must be retained for a minimum of 1 year for operational purposes and 7 years for compliance, depending on legal requirements.

5. Security Testing:

Conduct regular penetration testing to identify vulnerabilities.

Perform static and dynamic code analysis to identify security flaws during development.

Implement vulnerability scanning tools to ensure systems remain compliant with current security standards.

6. Threat Management:

Incident Response:

Establish an incident response plan, including roles, responsibilities, and communication protocols during security incidents.

Incidents must be acknowledged within 15 minutes and resolved or escalated within 24 hours.

Malware Protection:

Deploy endpoint protection tools to detect and mitigate malware threats.

Regularly update antivirus definitions and block-lists.

**Authentication Requirements:**

Authentication is a critical component of any system’s security architecture. It ensures that only authorized users or systems can access resources, thereby safeguarding sensitive data and functionality. Below are the detailed authentication requirements:

1. Authentication Mechanisms
2. Biometric Authentication
3. Single Sign-On
4. Role-Based Authentication
5. Device-Based Authentication
6. Session Management

**Authorization and Access Controls:**

Authorization and access controls define who has permission to access what resources and how those permissions are enforced. These controls are essential to prevent unauthorized actions and ensure system integrity, confidentiality, and accountability.

1. Principles of Authorization:

1.1. Principle of Least Privilege (POLP)

1.2. Role-Based Access Control (RBAC)

1.3. Attribute-Based Access Control (ABAC)

1.4. Separation of Duties (SoD)

1.5. Zero Trust Security

**Availability Requirements:**

1. Availability Goals
2. High-Availability Architecture
3. Disaster Recovery
4. Scalability Requirements
5. Monitoring and Alerts
6. Maintenance Windows
7. Environmental Considerations
8. Availability Testing

**Usability Requirements:**

Usability requirements define the system's ease of use and accessibility for users. These requirements ensure the system meets user expectations, supports efficient task completion, and provides an intuitive experience for diverse user groups.

1. User-Centered Design Principles
2. Response Time and Feedback
3. Error Prevention and Recovery
4. Personalization and Customization
5. Multi-Device Usability
6. Training and Support
7. Usability Testing and Metrics
8. System Feedback and Notifications

**System Help Requirements:**

The system help requirements define the assistance mechanisms available to users to ensure they can efficiently navigate, understand, and use the system. These requirements focus on providing timely and relevant support to users of all skill levels.

1. Integrated Help Features
2. User Documentation
3. Support Channels
4. Search Functionality
5. Self-Service Tools

**Performance Requirements:**

Performance requirements specify the system's expected behavior under various conditions, ensuring it delivers a responsive, reliable, and efficient user experience. These requirements focus on response times, scalability, resource utilization, and overall system efficiency.

1. Response Time Requirements
2. Throughput Requirements
3. Scalability
4. Availability and Uptime
5. Resource Utilization

**Scalability Requirements:**

Scalability requirements define the system's ability to handle increasing workloads, user demands, and data volumes without performance degradation. They ensure the system remains reliable and efficient as the business grows.

1. General Scalability Objectives
2. User Load Scalability
3. Data Volume Scalability
4. Geographic Scalability
5. Application Scalability

**Interface Requirements:**

The interface requirements specify how the system interacts with users and other systems, ensuring effective communication, ease of use, and integration. This section covers both User Interfaces (UI) and System Interfaces (SI) requirements.

1. User Interface Requirements
2. System Interface Requirements
3. Interface Testing Requirements
4. Compliance and Standards
5. Future-Proofing

AS-IS & TO-BE:

#### **AS-IS Inventory Management**:

1. **Manual Data Entry**
2. **Limited Visibility**
3. **Inefficient Stock Management**
4. **High Lead Times**
5. **No Integration**

#### **AS-IS Quick Delivery System**:

1. **Manual Dispatching**
2. **Lack of Tracking**
3. **Delayed Order Fulfillment**
4. **Ineffective Communication**

#### **TO-BE Inventory Management**:

1. **Automated Data Entry**
2. **Real-Time Visibility**
3. **Automated Stock Alerts**
4. **Integration with Forecasting**
5. **Mobile Access**

#### **TO-BE Quick Delivery System**:

1. **Route Optimization**
2. **Real-Time Tracking**
3. **Automated Dispatching**
4. **Customer Communication**
5. **Integration with Inventory**
6. **Prepare process flow diagram using your imagination.**

Process flow diagram is a visual representation of a process or system that uses shapes & lines to show the steps involved. PFD,s are used in many fields to help people understand, plan, & improve processes.

Process Flow Description for Inventory Management and Delivery System:

1.Customer Order Placement

Input: Customer places an order via the web/mobile app or through customer support.

Output: Order details are captured and forwarded to the Order Processing System.

2.Order Processing System

Action: Validates the order (e.g., availability of items, payment confirmation).

Decision: If inventory is insufficient, notify the customer; else, proceed to inventory allocation.

3.Inventory Management System

Input: The system retrieves real-time stock levels from the database.\n

Action: Updates inventory records to reserve stock for the order.

Output: Sends updated stock data to the ERP system and initiates a delivery request.

4.Delivery Optimization Module

Input: Receives delivery request along with customer details and location.

 Action:

Calculate the best route using GPS/mapping services.

Prioritize the order based on proximity and urgency.

Assign delivery personnel or vehicles.\n

Output: Updates the delivery schedule and notifies the customer.

5.ERP System Integration

Synchronizes inventory data, order status, and delivery updates for enterprise-level reporting.

6.Customer Notifications

Real-time updates about order status (e.g., packed, shipped, out for delivery, delivered).

7.Feedback Loop

Captures delivery performance metrics (e.g., time taken, delays).

Logs customer feedback for service improvement.



**Assignment 2:**

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

Mrs. Rutuja Madnoorkar

Business Analyst

Herald Logic

7559315328

Date: 11/JAN/2024

Sushant Thakur

Industrial Innovators

91 spring road Godrej & Boyce,

Gate no. 2, plant no. 6

Lal Bahadur Shastri Marg

Mumbai, Maharashtra 400079

Subject: Introduction as Business Analyst

Dear Sushant,

I hope this message finds you well. My name is Rutuja Madnoorkar, and I am the Business Analyst assigned to work closely with you and your team on to build software to manage the inventory & quickest delivery to customer. It is both a pleasure and a privilege to collaborate with you in starting the business understanding process for this exciting initiative.

My primary role will be to act as the liaison between your team and ours, ensuring that your vision and requirements are effectively captured and translated into actionable solutions. I will be focusing on understanding your business needs, identifying potential challenges, and recommending strategies to optimize outcomes.

As we embark on this journey together, I aim to:

-Develop a thorough understanding of your business goals and objectives.

-Facilitate discussions to gather requirements and insights from your team.

-Provide detailed documentation and analysis to guide the project.

-Maintain clear and open communication throughout the process.

I look forward to our initial discussions and the opportunity to learn more about your organization’s operations, priorities, and expectations. Please feel free to reach out to me at rutuja.madnoorkar18@gmail.com or 7559315328 should you have any questions or require additional information prior to our first meeting.

Thank you for entrusting us with this project. I am confident that our collaboration will yield successful outcomes, and I am excited to begin working with you and your team.

Warm regards,

Mrs. Rutuja Madnoorkar

Business Analyst

Herald Logic

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

BRD Document: A Business requirement document is a formal document that outlines a projects goals, scope, & requirements. It is used by project managers & business analysts to manage business requirements in an organized way.

SRS Document: A software requirements specification document is a detailed description of a software products requirements, expectations, & design. It’s a blueprint for the software being developed. & is used by software developers throughout the product development process.

**Business Requirement Document (BRD)**

**Introduction:**

**Project Title: Online Ticketing System**

**Description:The Online Ticketing System is a web-based platform designed to simplify the process of booking tickets for events such as movies, concerts, and sports. Users can browse available events, select seats, and make payments seamlessly. The system ensures real-time updates on ticket availability and secure payment processing.**

**Project Background:**

**Increased Efficiency: Reducing manual workload through automated workflows.**

**Enhanced User Experience: Providing a seamless way for customers or internal teams to raise and track issues.**

**Improved Accountability: Ensuring all tasks are tracked with clear ownership and deadlines.**

**Scalability: Supporting future growth with customizable and adaptable features.**

**Business Requirements:**

**Key Purposes of Business Requirements in a Ticketing System:**

1. **Define Objectives: Clarify the business goals that the ticketing system should support, such as improving customer service response times, tracking internal requests, or automating task assignments.**
2. **Guide System Development: Provide a clear direction for the development team, helping them understand what features and functionalities are essential to meet business needs.**
3. **Ensure Alignment: Ensure that the ticketing system aligns with the organization's operational workflows, processes, and strategic priorities.**
4. **Facilitate Stakeholder Collaboration: Serve as a common ground for collaboration among various stakeholders, including business teams, IT, and end-users, by outlining shared expectations and desired outcomes.**
5. **Improve Communication: Establish a detailed framework for communicating business needs to the technical team, making sure both sides have a mutual understanding of requirements and constraints.**
6. **Enhance Decision-Making: Provide a basis for making informed decisions regarding system design, feature prioritization, and resource allocation.**
7. **Establish Performance Metrics: Define the key performance indicators (KPIs) that will be used to measure the success of the ticketing system in meeting business needs.**

**Benefits and Rationale:**

1. **Improved Efficiency:**

**Automates repetitive tasks like ticket creation, prioritization, and assignment, reducing manual effort and speeding up service delivery.**

**Streamlines internal workflows, enabling teams to manage tasks more effectively and focus on core responsibilities.**

1. **Enhanced Collaboration:**

**Facilitates seamless communication and collaboration among team members, departments, and stakeholders by providing a centralized platform for tracking issues, requests, and updates.**

**Ensures everyone has visibility into task statuses, progress, and outcomes.**

1. **Faster Issue Resolution:**

**Prioritization features help to quickly address critical issues, ensuring prompt responses and faster resolution times.**

**Enables better monitoring of ticket progress, ensuring bottlenecks are identified and addressed swiftly.**

1. **Increased Accountability:**

**Clearly assigns responsibility to team members for specific tasks, improving accountability and ensuring that no request falls through the cracks.**

**Tracks the life-cycle of each ticket, providing transparency into who is working on what and at what stage.**

1. **Scalability and Flexibility:**

**Easily scales to accommodate growing workloads and expanding teams, making it suitable for organizations of various sizes and industries.**

**Provides customizable workflows, fields, and integrations that can adapt to changing business needs.**

**Stakeholders:**

**Business Teams**

**IT and System Administrators**

**Product Owners**

**End Users**

**Executive Leadership**

**Quality Assurance (QA) Teams**

**Legal and Compliance Teams**

**Goals and Objectives:**

**Goals:**

* **To create a user-friendly platform for ticket booking.**
* **To provide real-time updates on ticket availability and pricing.**
* **To ensure secure and efficient payment processing.**

**Objectives:**

* **Develop a responsive web interface accessible via desktop and mobile devices.**
* **Integrate payment gateways for multiple payment options.**
* **Enable users to manage bookings and view their purchase history.**

**Dependencies on existing systems:**

**IT Infrastructure Dependencies**

**Software and Applications**

**Data Dependencies**

**Process Dependencies**

**Security and Compliance Dependencies**

Assumptions:

System Usage and Adoption

Data Availability and Accessibility

IT and Technical Resources

Project Management and Implementation

System Functionality

Requirements Scope:

In-Scope Requirements:

Ticket Management

Integration with Existing Systems

Reporting and Analytics

User Roles and Permissions

Notifications and Alerts

Out-of-Scope Requirements:

Development of New Core Systems

Major Overhaul of Existing IT Infrastructure

Customer Relationship Management (CRM) Customization

External System Customization

Long-term Data Archiving Solutions

 Functional Requirements:

-User registration and login functionality.

-Event browsing by category, date, and location.

-Real-time seat availability updates.

-Secure payment gateway integration.

-Booking management features.

-Admin dashboard for managing events and generating reports.

**Actor Profiles Specification:**

End Users

Customer Service Agents

**IT Support Technicians**

System Administrators

Product Owners

**Use case Diagram:**

****

**Use case specification:**

**Use Case 1: Browse Events**

**Actors: User**

**Description: The user browses available events by category, date, or location.**

**Precondition: The user has access to the system.**

**Post condition: The user views a list of events and their details.**

**Main Flow:**

**User logs into the system.**

**User selects the browse events option.**

**The system displays available events.**

**Use Case 2: Select Seats**

**Actors: User**

**Description: The user selects preferred seats for an event.**

**Precondition: The user has chosen an event.**

**Post condition: The selected seats are reserved temporarily.**

**Main Flow:**

**User selects an event.**

**System displays seating availability.**

**User selects desired seats.**

Use Case 3: Make Payment

Actors: User, Payment Gateway

Description: The user completes payment for tickets.

Precondition: The user has selected seats.

Post condition: The booking is confirmed and payment is processed.

Main Flow:

User proceeds to checkout.

System redirects to the payment gateway.

User completes payment.

System confirms booking.

Use Case 4: Manage Booking

Actors: User

Description: The user can view, modify, or cancel bookings.

Precondition: The user has a confirmed booking.

Post condition: The changes are reflected in the system.

Main Flow:

User navigates to the booking management section.

User views booking details.

User modifies or cancels the booking.

Use Case 5: View Reports

Actors: Admin

Description: The admin generates reports on ticket sales and system performance.

Precondition: The admin has logged into the system.

Post-condition: Reports are generated successfully.

Main Flow:

Admin logs in.

Admin navigates to the reports section.

Admin selects the report type and time-frame.

System generates the report.

Business Rules:

1. Ticket Creation Rules:

Rule: All new tickets must include a valid issue type (e.g., Incident, Request, Change).

2. Priority Assignment:

Rule: The priority of a ticket must be assigned based on the impact and urgency of the issue.

3. Escalation Process:

Rule: Tickets should be escalated if they are unresolved after a predefined Service Level Agreement (SLA).

4. Assignment of Tickets:

Rule: Tickets are automatically assigned to the team or individual responsible based on the ticket type.

5. Resolution Timelines:

Rule: Tickets must be resolved within the defined SLA, which varies by ticket priority.

6. Data Validation:

Rule: Ticket descriptions must be limited to 500 characters and should not contain any special characters.

Data Requirements:

Data Requirements in a Ticketing System specify the types of information that must be captured, stored, and processed to ensure the effective functioning of the system. These data requirements support essential functionalities such as ticket creation, tracking, reporting, and analytics. Properly defined data requirements ensure accuracy, consistency, and usability across the system.

1. Basic Ticket Information:

Requirement: Every ticket must capture basic information such as ticket ID, issue type, and description.

Example Fields:

Ticket ID (unique identifier)

Issue type (e.g., Incident, Request, Change)

Description (detailed explanation of the issue)

Priority (High, Medium, Low)

2. User Data:

Requirement: Capture details about users submitting tickets or assigned to tickets.

Example Fields:

User ID (unique identifier)

Full name

Email address

Department or team affiliation

3. Date and Time Information:

Requirement: Capture relevant timestamps for tracking ticket lifecycles.

Example Fields:

Ticket creation date and time

Last update date and time

Ticket resolution date and time

SLA breach dates (if applicable)

4. Status and Workflow Data:

Requirement: Maintain data on ticket status and workflow stages.

Example Fields:

Status (Open, In Progress, Resolved, Closed)

Workflow stage (assigned, pending, under review)

Reason for status change (e.g., awaiting user feedback)

5. Attachment Data:

Requirement: Store file attachments linked to tickets securely.

Example Fields:

File name

File type (e.g., PDF, image)

File size

Uploaded by (user ID)

6. Priority and Urgency Data:

Requirement: Capture ticket priority and urgency to help prioritize tasks.

Example Fields:

Priority (High, Medium, Low)

Urgency (Critical, Normal, Low)

7. Categorization and Tags:

Requirement: Support categorization and tagging of tickets for better organization and searchability.

Example Fields:

Ticket category (Technical, HR, Customer Support)

Tags (e.g., Bug, Performance Issue)

Non-Functional Requirements:

-Scalability

-Security

-Performance

-Availability

-Usability

Security Requirements:

Security Requirements in a Ticketing System ensure that sensitive information is protected, system access is controlled, and data integrity is maintained.

1. Authentication:

Requirement: The ticketing system must implement secure user authentication to ensure only authorized users access the system.

2. Authorization:

Requirement: Access to system functions and data should be based on users’ roles and permissions.

3. Data Encryption:

Requirement: Data in transit and at rest must be encrypted to prevent unauthorized access.

4. Access Control:

Requirement: Ensure strict access control measures to prevent unauthorized users from accessing sensitive information or performing unauthorized actions.

5. Session Management:

Requirement: The system must implement secure session management to prevent session hijacking and ensure sessions are timely expired.

6. Audit Trails:

Requirement: The system must maintain detailed audit trails to record user actions and system events.

7. Data Minimization:

Requirement: The system should collect only the data necessary to perform its functions and avoid unnecessary storage of sensitive information.

8. Secure Communication:

Requirement: Ensure secure communication channels for data exchange, preventing interception during transmission.

9. Vulnerability Management:

Requirement: The system must have mechanisms to identify, address, and mitigate vulnerabilities in a timely manner.

Availability Requirements:

Availability Requirements in a ticketing system define the expected uptime, accessibility, and performance levels the system must maintain to ensure continuous operations.

1. Uptime:

Requirement: The ticketing system must have a target uptime percentage, ensuring it remains operational during specified periods.

2. High Availability (HA):

Requirement: The ticketing system should have measures in place to ensure continuous availability even in the event of hardware or network failures.

3. Disaster Recovery (DR):

Requirement: The system must have a disaster recovery plan that ensures data recovery and system availability in case of outages.

4. Maintenance Window:

Requirement: The system should have a defined maintenance window during which updates and patches can be applied without impacting end users.

5. Response Time:

Requirement: The ticketing system must meet specific response time requirements for both user actions and system-generated responses.

6. Fault Tolerance:

Requirement: The system should be capable of continuing to operate during faults without significant performance degradation.

7. Load Balancing:

Requirement: The ticketing system must implement load balancing to distribute user requests evenly across servers, ensuring optimal performance.

8. Data Redundancy:

Requirement: The system must implement data redundancy to ensure continuous access to critical information.

Usability Requirements:

Usability Requirements in a ticketing system define the features, design, and functionality that ensure ease of use, accessibility, and user satisfaction.

1. Intuitive User Interface (UI):

Requirement: The ticketing system must have an intuitive and user-friendly interface that allows users to easily navigate through features and functions.

2. Role-Specific Dashboards:

Requirement: Provide role-specific dashboards that offer relevant information and functionalities based on user roles (e.g., administrators, support agents, customers).

3. Efficient Ticket Creation:

Requirement: The system must allow users to efficiently create new tickets by minimizing the number of fields and steps required.

4. Search and Filtering Functionality:

Requirement: The system must provide robust search and filtering functionalities to help users quickly find relevant tickets.

5. Help and Guidance:

Requirement: The ticketing system must provide accessible help features, such as inline tool tips, user guides, and FAQs, to assist users in completing tasks.

6. Multi-Device Accessibility:

Requirement: The ticketing system must be accessible on different devices, such as desktops, tablets, and mobile phones, ensuring usability across platforms.

7. Real-Time Notifications:

Requirement: The system must provide real-time notifications to keep users informed about ticket updates, approvals, or escalations.

Performance Requirements:

Performance Requirements in a ticketing system define the expected operational efficiency and response times under various conditions.

1. Response Time

Requirement: The system must meet specific response time targets for user actions, such as ticket creation, updates, and searches.

2. System Scalability

Requirement: The ticketing system must be scalable to handle increasing numbers of users and tickets without degradation in performance.

3. Database Performance

Requirement: The database used in the ticketing system must perform efficiently to handle high query loads and frequent data updates.

4. Ticket Processing Speed

Requirement: The system must efficiently process tickets, including updates and status changes, within acceptable time-frames.

5. Search and Filtering Performance

Requirement: The system should deliver quick and responsive search results when filtering tickets.

6. Concurrency Handling

Requirement: The ticketing system must be able to handle high levels of concurrent users accessing and interacting with tickets.

7.Response for System Notifications

Requirement: Automated system notifications (such as ticket updates or SLA breaches) must be delivered quickly.

Scalability Requirements:

scalability Requirements in a ticketing system ensure that the system can effectively grow in capacity and performance to meet increased demand over time.

1. Horizontal Scalability

Requirement: The system must be designed to allow the addition of more servers, nodes, or instances to distribute load evenly and maintain performance.

2. Vertical Scalability

Requirement: The system must allow for increasing the capacity of individual servers (e.g., upgrading server hardware) to handle more processing power or storage.

3. Scalability Under High Load

Requirement: The system must be able to handle peak loads, including ticket submissions, searches, and updates, without performance degradation.

4. Concurrent User Handling

Requirement: The system must be capable of handling a growing number of concurrent users without performance degradation.

5. Ticket Volume Growth

Requirement: The system must be able to handle an increasing number of tickets submitted and resolved over time without performance declines.

Interface Requirements:

Interface Requirements in a ticketing system define the specifications for how the system interacts with users, other systems, devices, and external applications.

1. User Interface (UI)

Requirement: The ticketing system must provide a user-friendly interface that is intuitive, visually consistent, and easy to navigate.

2. Application Programming Interface (API)

Requirement: The system must expose APIs that allow external applications and systems to interact with it, such as retrieving ticket data or submitting new tickets.

3. Integration with External Systems

Requirement: The ticketing system must integrate with other internal and external systems, such as CRM tools, help-desk solutions, or payment gateways.

4. Data Import and Export Interfaces

Requirement: The system must support bulk data import and export functionalities to allow seamless migration of data or ticket details.

5. System-to-System Interfaces

Requirement: The system must provide reliable interfaces for communication between different ticketing system components (e.g., between the front-end, back-end, and database).

AS-IS & TO-BE:

#### **AS-IS System:**

1. **Manual Ticket Creation**
2. **Limited Tracking**
3. **No Integration**
4. **Slow Resolution Times**
5. **Lack of Automation**
6. **Poor Communication**

#### **TO-BE System**:

1. **Automated Ticket Creation**
2. **Real-Time Tracking**
3. **Integration with Business Tools**
4. **Faster Resolution Times**
5. **Robust Automation**
6. **Effective Collaboration**
7. **Self-Service Options**
8. **Multi-Channel Support**

**Software Requirements Specification (SRS) for Ticketing System**

Introduction:

Purpose:

The purpose of this document is to define the functional and non-functional requirements for an online ticketing system. The system will facilitate users in purchasing tickets for events, concerts, movies, or transportation. It ensures a user-friendly experience with secure payment and efficient management of ticket sales.The system enables efficient collaboration between support teams, prioritizes tasks based on severity and impact, and provides a clear audit trail of customer interactions. It enhances customer satisfaction by ensuring timely and organized responses to service requests.

Scope:

The ticketing system will allow users to browse events, view details, and purchase tickets through a secure platform. The system will also support administrators in managing events, tickets, and users.

Overview:

The Ticketing System functions as an essential tool for organizations to manage service requests and incidents systematically. It encompasses the following key features:

1. Ticket Creation:

Users (internal staff or external customers) can create service tickets by submitting requests or reporting issues through various channels such as email, web forms, or mobile apps.

1. Ticket Management:

The system allows tickets to be categorized based on types (e.g., technical support, service requests, system issues) and assigned to appropriate teams or individuals.

1. Prioritization:

Tickets are prioritized according to their urgency and impact using predefined rules or custom configurations. This ensures that critical issues receive immediate attention.

1. Collaboration:

Multiple team members can collaborate on the same ticket, exchanging comments, attaching files, and sharing relevant information to facilitate issue resolution.

1. Automation:

The system automates repetitive tasks such as assigning tickets, sending notifications, and escalating issues to higher priority when certain conditions are met.

1. Tracking and Reporting:

Tickets provide comprehensive tracking, enabling teams to monitor progress, identify bottlenecks, and generate reports on service performance and issue resolution timelines.

Goals and Objectives:

Goals:

-Simplify the ticket-purchasing process.

-Provide a centralized platform for event management.

-Ensure a secure and reliable transaction system.

Objectives:

-Enable users to search and filter events.

-Provide a smooth checkout process with multiple payment options.

-Offer real-time availability updates for tickets.

-Ensure the scalability of the platform to handle high traffic.

Acronyms and definitions:

| **Acronym** | **Term** | **Description** |
| --- | --- | --- |
| **SRS** | Software Requirements Specification | A detailed document that describes the system requirements, user needs, functional requirements, and constraints for a ticketing system. |
| **SLA** | Service Level Agreement | A contract between a service provider and a customer that defines the level of service expected, including response times, resolution times, and other performance metrics. |
| **ITSM** | IT Service Management | A set of policies, processes, and activities used by organizations to design, deliver, manage, and improve IT services, including service request management through ticketing systems. |
| **CR** | Change Request | A formal request for modifying system configurations, workflows, or processes in the ticketing system to improve performance or introduce new features. |
| **API** | Application Programming Interface | A set of rules and protocols that allow different software applications, like the ticketing system, to communicate and interact with each other. |
| **KPIs** | Key Performance Indicators | Metrics used to measure the performance of the ticketing system, such as resolution time, customer satisfaction, or ticket backlog. |

Operational Requirements:

Operational requirements outline the essential conditions and capabilities that a ticketing system must meet to ensure smooth day-to-day functioning and effective service delivery. These requirements focus on system performance, usability, reliability, scalability, and security.

System Availability: The ticketing system must be available 24/7 to support continuous service operations, with minimal downtime.

Performance: The system should handle a high volume of ticket transactions without significant delays or crashes.

Scalability:The ticketing system must be scalable to support growing numbers of users, tickets, and data.

Data Security:The system must ensure secure storage, encryption, and access control to protect sensitive information.

Backups:Regular backups of ticket data should be conducted to prevent data loss in case of system failure.

Software Requirements:

| **Requirement** | **Description** |
| --- | --- |
| **Operating System** | The system must be compatible with common operating systems such as Windows, Linux, or cloud-based platforms. |
| **Database** | The system must integrate with a database, like SQL or NoSQL, for storing and managing ticket information. |
| **Integration APIs** | The ticketing system must include APIs to integrate with other tools (e.g., CRM, email systems, chat applications). |
| **Middleware** | The system may require middleware for managing interactions between various software components. |
| **Third-Party Tools** | The ticketing system should support integration with commonly used third-party tools and extensions (e.g., Slack, Microsoft Teams). |
| **Cloud Compatibility** | The ticketing system must be compatible with cloud-based environments for scalability and ease of deployment. |
| **Version Control** | The system must support version control to manage software updates and changes. |

### ****Hardware Requirements:****

| **Requirement** | **Description** |
| --- | --- |
| **Server Requirements** | The ticketing system must run on dedicated servers or cloud infrastructure with appropriate specifications. |
| **Database Requirements** | The system requires a robust database to store ticket information securely and efficiently. |
| **Network Requirements** | The system must have a stable and secure network connection to ensure consistent access. |
| **Storage Requirements** | Adequate storage capacity to handle ticket data, attachments, and system logs over time. |

Design and Implementation Constraints:

Design Constraints:

Design constraints focus on the architectural, functional, and usability limitations imposed during system design.

1. Scalability Limitation: The ticketing system must remain efficient even when handling a high volume of concurrent users and tickets. Scalability must not compromise performance or reliability.
2. Integration Restriction: The ticketing system must ensure seamless integration with existing IT infrastructure and tools (e.g., CRM, ERP), limiting third-party dependencies that could cause system instability.
3. Usability Standards: The design must comply with accessibility standards, ensuring usability for all users, including those with disabilities, using widely accepted design patterns and UI best practices.
4. Compliance Requirements: The system’s design must account for legal and regulatory requirements, such as data privacy (GDPR, HIPAA) and industry best practices, which impact data handling, storage, and security.
5. User Role Management: The system must provide a role-based access model, restricting the visibility and control of tickets based on user roles and permissions, ensuring data security.

Implementation Constraints:

Implementation constraints address technical limitations, resource availability, and system performance considerations during the coding, testing, and deployment phases.

1. Resource Availability: The system must be implemented within the constraints of available hardware, such as server capacity, network bandwidth, and storage limitations. There is limited scope for hardware upgrades in certain environments.
2. Technology Stack: The choice of development frameworks, languages, and databases must align with existing organizational technology stacks and resources to ensure compatibility and ease of future maintenance.
3. Development Time-frame: The development of the ticketing system must adhere to strict timelines, with minimal delays in each phase of development, from coding to testing and deployment.
4. Legacy System Integration: If integrating with legacy systems, the implementation must ensure backward compatibility, minimizing disruption to existing workflows and data structures.
5. High Availability Constraints: The system must maintain high availability even under peak load conditions, with limited tolerance for downtime to ensure continuous service delivery.
6. Security Implementation: Implementing robust security measures, such as encryption and access controls, must align with the resources available and the constraints on data sensitivity.

System Overview:

Key Features of the Ticketing System:

1. Ticket Management:

Centralized repository for tracking and resolving service requests and incidents.

Ability to create, update, and resolve tickets with assigned priorities and statuses.

1. Automation:

Automates repetitive tasks such as ticket assignment, notification alerts, and escalations based on pre-defined rules.

1. Collaboration:

Enables team collaboration through internal comments, file sharing, and discussions linked to specific tickets.

1. Prioritization:

Provides the ability to categorize tickets based on their urgency and business impact, ensuring critical issues are addressed first.

1. Service Level Agreement (SLA):

Tracks adherence to SLAs, ensuring that service commitments (response and resolution times) are met.

Multi-Channel Support:

### **System Architecture**:

* **Frontend**:

Web-based and mobile interfaces for end-users (customers or internal staff) to interact with the ticketing system.

* **Backend**:

Core ticket management, automation, database storage, and processing components that handle ticket creation, assignment, resolution, and reporting.

* **Database**:

Centralized data storage where all ticket details, user interactions, and historical data are securely stored and managed.

* **APIs**:

Facilitates integration with external systems such as CRM, ITSM tools, and third-party communication platforms.

### **System Benefits**:

* **Improved Service Delivery**: Streamlines the resolution process, ensuring faster response and resolution times, which directly impacts customer satisfaction.
* **Enhanced Collaboration**: Facilitates seamless communication between teams, reducing manual handovers and ensuring better issue tracking.
* **Efficient Ticket Management**: Reduces ticket backlog by prioritizing and automating workflows, improving overall team productivity.
* **Scalability**: Built to accommodate increasing user loads and growing service requests, making it suitable for both small teams and large enterprises.

### **System Use Cases**:

* **IT Support**: Tracks and resolves technical support requests efficiently, ensuring system stability and uptime.
* **Customer Support**: Manages customer service interactions, including issue tracking, complaint resolution, and service request fulfillment.
* **Operational Requests**: Handles internal requests such as equipment maintenance, building services, or access permissions within the organization.

Benefits of the Proposed System:

1.Improved Service Delivery

2. Increased Operational Efficiency

3. Enhanced Collaboration

4. Better Customer Experience

5. Enhanced Security

6. Efficient Reporting and Analytics

7. Flexible Integration

8. Scalability

9. Reduced Downtime

10. Cost Efficiency

1. Use case Diagram:

A use case is a description of a system’s behavior as it responds to a request from one of its users or another system. It defines a sequence of actions that the system performs to achieve a particular goal or provide a specific service to the user. Use cases are often used in systems analysis & design to capture the functional requirements of a system.



4. Use Case Specifications:

4.1 Browse Events:

Actor: User

Description: User searches and filters events based on location, date, or category.

Preconditions: User has access to the platform.

Post conditions: List of relevant events is displayed.

4.2 View Event Details:

Actor: User

Description: User views detailed information about a selected event.

Preconditions: Event exists in the database.

Postconditions: Event details are displayed.

4.3 Purchase Tickets:

Actor: User

Description: User selects tickets, proceeds to checkout, and completes payment.

Preconditions: Tickets are available, and payment gateway is functional.

Post conditions: Payment is processed, and ticket confirmation is sent.

4.4 Manage Events:

Actor: Admin

Description: Admin adds, updates, or removes events.

Preconditions: Admin is logged in.

Post conditions: Event database is updated.

4.5 Generate Reports:

Actor: Admin

Description: Admin generates sales and user activity reports.

Preconditions: Data exists for the selected period.

Post conditions: Reports are generated and available for download.

5. Functional Requirements:

-User Registration: Users can sign up and manage their profiles.

-Event Browsing and Filtering: Users can search for events by category, location, or date.

-Ticket Purchase: Secure transaction process with multiple payment options.

-Real-Time Availability: Tickets are updated in real-time.

-Admin Management: Admins can add, edit, or delete events.

6. Non-Functional Requirements:

-Scalability

-Security

-Performance

-Reliability

-Usability

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

Entity relationship diagram is a visual representation of how entities relate to each other in a database or information system. An ERD is a flowchart like diagram that uses symbols like rectangles,ovals,& diamonds to show how entities are connected.

How its used:

-Design databases

-Maintain databases

-Troubleshoot issues

-Train new team members



**4. User story of shopping from e-commerce.**

User stories: User story is a brief description of software feature or functionality written from the perspective of the end user of a system. In software development & product management. A user story is an informal, natural language description of features of a software system. They facilitate sense making & communication; & may help software teams document their understanding of the system & its context.

|  |  |  |
| --- | --- | --- |
| User story No. 1 | Task: 2 | Priority-HIGHEST |
| As a user, I want to create an account so that I can save my preferences. |
| BV:08 | CP:05 |
| Acceptance Criteria:User can sign up, verify email, and log in successfully. |

|  |  |  |
| --- | --- | --- |
| User story No. 2 | Task: 3 | Priority-HIGHEST |
| As a user, I want to log in securely using my credentials. |
| BV:10 | CP:04 |
| Acceptance Criteria:User can log in with valid credentials and see an error for invalid attempts. |

|  |  |  |
| --- | --- | --- |
| User story No. 3 | Task: 2 | Priority-HIGHEST |
| As a user, I want to browse products by category. |
| BV:09 | CP:06 |
| Acceptance Criteria:Categories are displayed, and products load correctly when a category is selected. |

|  |  |  |
| --- | --- | --- |
| User story No. 4 | Task: 2 | Priority-HIGHEST |
| As a user, I want to search for products using keywords. |
| BV:10 | CP:07 |
| Acceptance Criteria:Search results are relevant to the entered keywords. |
| User story No. 5 | Task: 2 | Priority-HIGHEST |
| As a user, I want to view product details, including images and reviews. |
| BV:09 | CP:06 |
| Acceptance Criteria:Product page shows image carousel, descriptions, and customer reviews. |

|  |  |  |
| --- | --- | --- |
| User story No. 6 | Task: 2 | Priority-HIGHEST |
| As a user, I want to add products to my cart for later purchase. |
| BV:10 | CP:07 |
| Acceptance Criteria:Products added to the cart appear in the cart page and persist during the session. |

|  |  |  |
| --- | --- | --- |
| User story No. 7 | Task: 1 | Priority-HIGHEST |
| As a user, I want to update quantities of products in my cart. |
| BV:08 | CP:05 |
| Acceptance Criteria:Quantity updates reflect in the cart total immediately. |

|  |  |  |
| --- | --- | --- |
| User story No. 8 | Task: 1 | Priority-HIGHEST |
| As a user, I want to remove products from my cart. |
| BV:09 | CP:04 |
| Acceptance Criteria:Products removed are no longer displayed in the cart. |

|  |  |  |
| --- | --- | --- |
| User story No. 9 | Task: 1 | Priority-HIGHEST |
| As a user, I want to see the total cost of my cart, including tax. |
| BV:08 | CP:06 |
| Acceptance Criteria:Cart displays subtotal, tax, and total accurately. |

|  |  |  |
| --- | --- | --- |
| User story No. 10 | Task: 3 | Priority-HIGHEST |
| As a user, I want to save items to a wish-list for future reference. |
| BV:07 | CP:05 |
| Acceptance Criteria:Wish-list functionality allows adding, removing, and viewing saved items. |

|  |  |  |
| --- | --- | --- |
| User story No. 11 | Task: 2 | Priority-HIGHEST |
| As a user, I want to apply discount codes during checkout. |
| BV:08 | CP:06 |
| Acceptance Criteria:Discount code field accepts valid codes and updates totals correctly. |

|  |  |  |
| --- | --- | --- |
| User story No. 12 | Task: 1 | Priority-HIGHEST |
| As a user, I want to select a shipping method for my purchase. |
| BV:08 | CP:05 |
| Acceptance Criteria:Shipping options are displayed with costs and estimated delivery times. |

|  |  |  |
| --- | --- | --- |
| User story No. 13 | Task: 1 | Priority-HIGHEST |
| As a user, I want to securely enter my payment details during checkout. |
| BV:10 | CP:08 |
| Acceptance Criteria:Payment gateway processes payments securely and shows confirmation. |

|  |  |  |
| --- | --- | --- |
| User story No. 14 | Task: 1 | Priority-HIGHEST |
| As a user, I want to track my order status post-purchase. |
| BV:09 | CP:07 |
| Acceptance Criteria:Order tracking page shows current status and expected delivery date. |

|  |  |  |
| --- | --- | --- |
| User story No. 15 | Task: 1 | Priority-HIGHEST |
| As a user, I want to view my order history. |
| BV:08 | CP:05 |
| Acceptance Criteria:User dashboard displays previous orders with details and status. |

|  |  |  |
| --- | --- | --- |
| User story No. 16 | Task: 1 | Priority-HIGHEST |
| As a user, I want to view my order history |
| BV:09 | CP:04 |
| Acceptance Criteria:Order confirmation email is sent immediately after purchase. |

|  |  |  |
| --- | --- | --- |
| User story No. 17 | Task: 1 | Priority-HIGHEST |
| As a user, I want to filter products by price range. |
| BV:07 | CP:06 |
| Acceptance Criteria:Price filter shows relevant products within the selected range. |

|  |  |  |
| --- | --- | --- |
| User story No. 18 | Task: 1 | Priority-HIGHEST |
| As a user, I want to filter products by customer ratings. |
| BV:08 | CP:05 |
| Acceptance Criteria:Products displayed match the selected rating filter. |

|  |  |  |
| --- | --- | --- |
| User story No. 19 | Task: 1 | Priority-LOWEST |
| As a user, I want to sort products by relevance, price, and popularity. |
| BV:08 | CP:06 |
| Acceptance Criteria:Sorting options display products correctly based on selected criteria. |

|  |  |  |
| --- | --- | --- |
| User story No. 20 | Task: 1 | Priority-LOWEST |
| As a user, I want to receive recommendations based on my browsing history. |
| BV:09 | CP:07 |
| Acceptance Criteria:Recommendations on the homepage are relevant to recent activities. |

|  |  |  |
| --- | --- | --- |
| User story No. 21 | Task: 1 | Priority-LOWEST |
| As a user, I want to compare products side-by-side. |
| BV:08 | CP:06 |
| Acceptance Criteria:Comparison tool displays selected products with specifications and features. |

|  |  |  |
| --- | --- | --- |
| User story No. 22 | Task: 1 | Priority-LOWEST |
| As a user, I want to read customer reviews before purchasing a product. |
| BV:10 | CP:04 |
| Acceptance Criteria:Review section displays user feedback with ratings and comments. |

|  |  |  |
| --- | --- | --- |
| User story No. 23 | Task: 1 | Priority-LOWEST |
| As a user, I want to see a "recently viewed" section for quick navigation. |
| BV:07 | CP:05 |
| Acceptance Criteria:Recently viewed items appear on the homepage or sidebar. |

|  |  |  |
| --- | --- | --- |
| User story No. 24 | Task: 1 | Priority-LOWEST |
| As a user, I want to pay using multiple payment methods. |
| BV:09 | CP:07 |
| Acceptance Criteria:Payment gateway supports credit/debit cards, PayPal, and digital wallets. |

|  |  |  |
| --- | --- | --- |
| User story No. 25 | Task: 1 | Priority-LOWEST |
| As a user, I want to track my delivery on a map in real time. |
| BV:08 | CP:09 |
| Acceptance Criteria:Real-time map tracking shows driver location and estimated delivery time. |

|  |  |  |
| --- | --- | --- |
| User story No. 26 | Task: 1 | Priority-LOWEST |
| As a user, I want to return or exchange products if needed. |
| BV:08 | CP:07 |
| Acceptance Criteria:Return/exchange process is accessible from the order details page. |

|  |  |  |
| --- | --- | --- |
| User story No. 27 | Task: 1 | Priority-LOWEST |
| As a user, I want to contact customer support for any issues. |
| BV:09 | CP:06 |
| Acceptance Criteria:Customer support page provides chat, email, and phone options. |

|  |  |  |
| --- | --- | --- |
| User story No. 28 | Task: 1 | Priority-LOWEST |
| As a user, I want to receive notifications for sales and special offers. |
| BV:08 | CP:05 |
| Acceptance Criteria:Notifications are sent via email or app push based on user preferences. |

|  |  |  |
| --- | --- | --- |
| User story No. 29 | Task: 1 | Priority-HIGHEST |
| As a user, I want to pre-order upcoming products. |
| BV:07 | CP:06 |
| Acceptance Criteria:Pre-order option is available for eligible products with estimated release dates. |

|  |  |  |
| --- | --- | --- |
| User story No. 30 | Task: 1 | Priority-HIGHEST |
| As a user, I want to view product availability at nearby stores. |
| BV:08 | CP:07 |
| Acceptance Criteria:Store locator shows inventory for selected products in nearby locations. |

|  |  |  |
| --- | --- | --- |
| User story No. 31 | Task: 1 | Priority-HIGHEST |
| As a user, I want to have my address autofilled using location services. |
| BV:08 | CP:06 |
| Acceptance Criteria:Location autofill suggests accurate address options during checkout. |

|  |  |  |
| --- | --- | --- |
| User story No. 32 | Task: 1 | Priority-HIGHEST |
| As a user, I want to upload images for product reviews. |
| BV:07 | CP:06 |
| Acceptance Criteria:Review form accepts image uploads and displays them in the review section. |

|  |  |  |
| --- | --- | --- |
| User story No. 33 | Task: 1 | Priority-HIGHEST |
| As a user, I want to share products with friends via social media. |
| BV:07 | CP:05 |
| Acceptance Criteria:Share button generates sharable links for social platforms. |

|  |  |  |
| --- | --- | --- |
| User story No. 34 | Task: 1 | Priority-HIGHEST |
| As a user, I want to view deals of the day prominently. |
| BV:09 | CP:05 |
| Acceptance Criteria:Homepage highlights daily deals with countdown timers and discounts. |

|  |  |  |
| --- | --- | --- |
| User story No. 35 | Task: 1 | Priority-HIGHEST |
| As a user, I want to pay using installment plans for expensive products. |
| BV:08 | CP:08 |
| Acceptance Criteria:Eligible products display installment options during checkout. |

|  |  |  |
| --- | --- | --- |
| User story No. 36 | Task: 1 | Priority-HIGHEST |
| As a user, I want to gift wrap items during checkout. |
| BV:06 | CP:04 |
| Acceptance Criteria:Gift wrap option is available with additional charges and custom messages. |

|  |  |  |
| --- | --- | --- |
| User story No. 37 | Task: 1 | Priority-HIGHEST |
| As a user, I want to receive loyalty points for purchases. |
| BV:09 | CP:07 |
| Acceptance Criteria:Points are credited to the user account immediately after the purchase. |

|  |  |  |
| --- | --- | --- |
| User story No. 38 | Task: 1 | Priority-HIGHEST |
| As a user, I want to use voice search to find products. |
| BV:07 | CP:08 |
| Acceptance Criteria:Voice search accurately recognizes commands and displays relevant products. |

|  |  |  |
| --- | --- | --- |
| User story No. 39 | Task: 1 | Priority-HIGHEST |
| As a user, I want to schedule my delivery at a convenient time. |
| BV:08 | CP:07 |
| Acceptance Criteria:Delivery scheduler allows selecting specific dates and time slots during checkout. |

|  |  |  |
| --- | --- | --- |
| User story No. 40 | Task: 1 | Priority-LOWEST |
| As a user, I want to see trending products based on user interests. |
| BV:08 | CP:06 |
| Acceptance Criteria:Homepage displays trending products based on user preferences and popular items. |