

Assignment 1 — Dairy Manufacturing & Distribution Platform

A. Business Requirements Document (BRD)

1) Executive Summary

A national dairy producer with multiple plants and warehouses needs a unified system to (1) manage inventory across raw materials, WIP, and finished goods and (2) provide the quickest delivery to B2B (distributors, modern trade) and B2C (own app/marketplaces) customers with cold-chain constraints.

2) Goals & KPIs

- G1: Real-time inventory accuracy $\geq 98\%$ across locations
- G2: Stockout rate $< 2\%$ for top 100 SKUs
- G3: OTIF (On-Time, In-Full) $\geq 95\%$
- G4: Average delivery time $\downarrow 25\%$ (baseline to be established)
- G5: Shrinkage/Wastage $\downarrow 30\%$ (cold-chain losses, expiry)

3) In Scope

- Multi-site inventory (plants, DCs, dark stores, vans)
- Batch/lot, expiry and temperature tracking
- Demand capture (orders from D2C app, call center, portals, EDI)
- Allocation & route optimization (multi-drop, time windows, cold-chain)
- Replenishment planning (min/max, reorder point, FEFO)
- Production consumption & back-flush (WIP $\rightarrow$ FG)
- Returns/RTO & reverse logistics
- Role-based dashboards & alerts
- Core integrations (ERP/Accounting, e-commerce, TMS/Maps, IoT sensors)

Out of Scope (Phase-1): Dynamic pricing, loyalty engine, advanced S&OP, internationalization, drone delivery.

4) Stakeholders

- COO (Sponsor), Head of Supply Chain, Plant Managers, Warehouse Managers, Transport Manager, Sales Ops, Customer Care, IT/InfoSec, Finance.

5) Assumptions & Constraints

- Cold-chain compliance (2–8°C for dairy; -18°C for ice-cream)
- FEFO required for all dispatches
- Trucks with GPS; optional temperature IoT gateways
- Peak loads during summer/festivals
- SLA windows differ by channel (Horeca vs D2C)

6) High-Level Business Requirements (selected)

- BR-01 Real-time stock by SKU, batch, expiry, location, and temperature band.
- BR-02 FEFO auto-allocation; manual override with reason codes.
- BR-03 Multi-channel order capture; dedupe and validation (credit, geofence, cut-off times).
- BR-04 Route planning with time windows, capacity, temperature constraints, traffic ETA.
- BR-05 Van stock management & ePOD (OTP/signature/photo), returns logging.
- BR-06 Production issue/receipt posting aligned to BOM & yields.
- BR-07 Replenishment proposals per DC by sales velocity, seasonality, safety stock.
- BR-08 Exception alerts (low stock, temperature breach, ageing/near-expiry).
- BR-09 Audit trails, maker-checker for adjustments.
- BR-10 API layer for ERP (GL postings), e-commerce, TMS, IoT.

7) Success Metrics

- Inventory accuracy $\geq 98\%$, shrinkage $\leq$ target, OTIF $\geq 95\%$, average delivery time $\downarrow 25\%$, near-expiry write-off $\downarrow 30\%$.

8) Risks & Mitigations

- Data quality $\rightarrow$ cleanse + dual-run;
- Change resistance $\rightarrow$ role-based training, super-users;
- IoT reliability $\rightarrow$ buffered logging + alerting;
- Seasonal spikes $\rightarrow$ autoscaling infra + capacity planning.

9) Reporting & Dashboards

- Control tower: OTIF, ATP by region, ageing heatmap, temperature breaches, route KPIs, pick/pack productivity, wastage.

B. Development & Resource Plan

1) Delivery Approach

- Agile (Scrum) — 2-week sprints; CI/CD; feature flags.
- Environments: Dev → QA → UAT → Prod; blue/green releases.

2) Phases & Timeline (indicative 24 weeks)

1. Discovery & Design (4 wks): workshops, AS-IS/TO-BE, NFRs, integration specs.
2. Foundations (2 wks): cloud, IAM, environments, pipelines.
3. Core Inventory + Inbound (6 wks): items/batches/FEFO, receipts, moves, cycle counts.
4. Order & Allocation (4 wks): multi-channel capture, ATP, FEFO allocation.
5. Routing & Delivery (4 wks): optimizer, driver app, ePOD, van stock.
6. Integrations & IoT (2 wks): ERP, e-comm, maps/TMS, temp sensors.
7. Reporting & UAT (2 wks): dashboards, performance, security, UAT.
8. Go-Live & Hypercare (2 wks).

3) Team & Effort (FTE)

- Product/BA: 1 Lead BA, 1 BA
- Engineering: 1 Architect, 3 Backend, 2 Frontend, 1 Mobile, 1 DevOps, 1 Data/BI
- QA: 2 QA (functional + automation)
- Design: 1 UX
- SM/PM: 1
- Change/Training: 1 enablement lead

4) Tech Stack (suggested)

- Backend: Node.js/Java + REST/GraphQL
- DB: PostgreSQL (OLTP), Redis (caching)

- Queue: Kafka/RabbitMQ
- Optimization: OR-Tools/Optaplanner
- Mobile (Drivers): Flutter/React Native
- Maps/ETA: Google Maps/OSM provider
- IoT: MQTT gateway + device twin
- BI: Power BI
- Cloud: AWS/Azure (managed Postgres, container services)

5) Non-Functional Requirements

- Availability 99.9%, p95 API < 300ms, scale to peak +50%, PII encryption, SOC2-aligned logging.

6) RACI (snippet)

- Process design: BA (R), Sponsor (A), Ops (C), Eng (I)
- Architecture: Architect (R), PM (A), SecOps (C), BA (I)
- UAT sign-off: Sponsor (A), Ops Managers (R), BA (C)

7) Training & Change

- Role-based guides, train-the-trainer, sandbox practice, helpdesk SOPs.

8) ROM Budget (illustrative)

- Team (6 months), infra, licenses, contingency 15%. (Can be tailored to your rates.)

1. Executive Summary

The company manages multiple plants and warehouses for ice-cream and milk products. Currently, processes are manual and fragmented, leading to inventory inaccuracy, wastage, and delayed deliveries.

This BRD proposes a centralized software solution for Inventory Management and Quick Delivery Optimization to enhance operational efficiency, reduce wastage, and improve customer satisfaction.

2. AS-IS Process (Current State)

AS-IS Inventory Management

- Each warehouse maintains stock records manually (Excel sheets, local software).

- No real-time visibility of stock across all plants/warehouses.
- Expiry and batch management tracked inconsistently, leading to wastage.
- Replenishment is reactive, based on manual stock checks.

AS-IS Order Management & Delivery

- Orders come from multiple sources (phone, distributors, retailers).
- Warehouse assignment done manually, not always nearest one.
- Delivery routes planned by local staff, often suboptimal.
- Limited or no tracking of delivery vehicles.
- Customers don't get real-time updates on order status.

AS-IS Reporting

- Fragmented reporting using spreadsheets.
- No consolidated dashboards for management.
- Difficult to forecast demand and track seasonal variations.

3. TO-BE Process (Future State)

TO-BE Inventory Management

- Centralized real-time inventory system across all plants/warehouses.
- Automated batch and expiry tracking for perishable products.
- System-generated replenishment triggers when stock falls below threshold.
- Integration with production planning to optimize supply chain.

TO-BE Order Management & Delivery

- Orders consolidated from all channels into a single Order Management System (OMS).
- Orders automatically routed to the nearest warehouse with available stock.
- Delivery routes optimized using GPS/AI-based route planning.
- Real-time tracking of delivery vehicles.
- Customers receive order and delivery status updates via SMS/app.

TO-BE Reporting & Analytics

- Centralized dashboard for stock levels, wastage, order fulfillment.
- Predictive analytics for demand forecasting (seasonal spikes).
- KPIs tracked (On-time delivery %, Inventory turnover, Wastage %).
- Business intelligence for management decision-making.

Area	AS-IS (Current)	TO-BE (Future)
Inventory Visibility	Manual, siloed, inaccurate	Centralized, real-time, automated
Batch/Expiry Mgmt	Inconsistent, manual	Automated, system-controlled
Order Allocation	Manual, non-optimized	Auto-allocation to nearest warehouse
Delivery Planning	Manual routes	AI-based optimized routing
Customer Updates	No real-time tracking	Real-time updates via app/SMS
Reporting	Fragmented spreadsheets	Centralized dashboards, predictive analytics
Forecasting	Absent	Data-driven demand forecasting
Integration	None/limited	Full ERP/3rd-party logistics integration

Water fall format

1. Requirements Phase

1.1 Business Objectives

- Centralize inventory management across all warehouses/plants.
- Enable quickest delivery of products to customers.
- Reduce wastage of perishable items through expiry tracking.
- Provide forecasting and analytics for decision-making.

1.2 Scope

In-Scope:

- Inventory management system.
- Order management and delivery optimization.

- Integration with ERP and logistics APIs.
- Reporting and analytics dashboards.

Out of Scope:

- Plant production line automation.
- External 3rd-party delivery fleet (unless via integration).

1.3 Stakeholders

- CEO / Management
- Supply Chain Head
- Warehouse Managers
- Sales & Marketing Teams
- IT/ERP Teams

2. Design Phase

2.1 System Architecture (High Level)

- Centralized Inventory & OMS database.
- Warehouse Module (stock, expiry, replenishment).
- Order Management Module (multi-channel input, allocation).
- Delivery Optimization Module (GPS integration, route planning).

- Analytics & BI Layer (dashboards, forecasting).

2.2 Process Flow (TO-BE)

Step 1: Customer places order →

Step 2: OMS routes to nearest warehouse →

Step 3: Warehouse picks & packs →

Step 4: Delivery system assigns optimized route →

Step 5: Customer receives real-time status →

Step 6: Delivery completed & feedback captured →

Step 7: Reports generated.

3. Development Phase

- Build modules in iterations:
 1. Inventory Module
 2. Order Management Module
 3. Delivery Optimization Module
 4. Reporting & Analytics

-
- Integrate with ERP/Finance system.
- API integration for logistics (if required).

4. Testing Phase

- Unit Testing: Verify individual modules.
- Integration Testing: Validate ERP, GPS, and API integration.
- User Acceptance Testing (UAT): Warehouse staff, supply chain team validate workflows.
- Performance Testing: Handle 10,000+ orders/day.
- Security Testing: Ensure role-based access and compliance.

5. Deployment Phase

- Deploy in pilot warehouses first.
- Phase-wise rollout to all plants & warehouses.
- Parallel run for 2–3 weeks to ensure data accuracy.

6. Maintenance Phase

- Ongoing system monitoring & bug fixes.
- Periodic performance optimization.
- Training and support for new staff.
- Continuous updates for demand forecasting models.

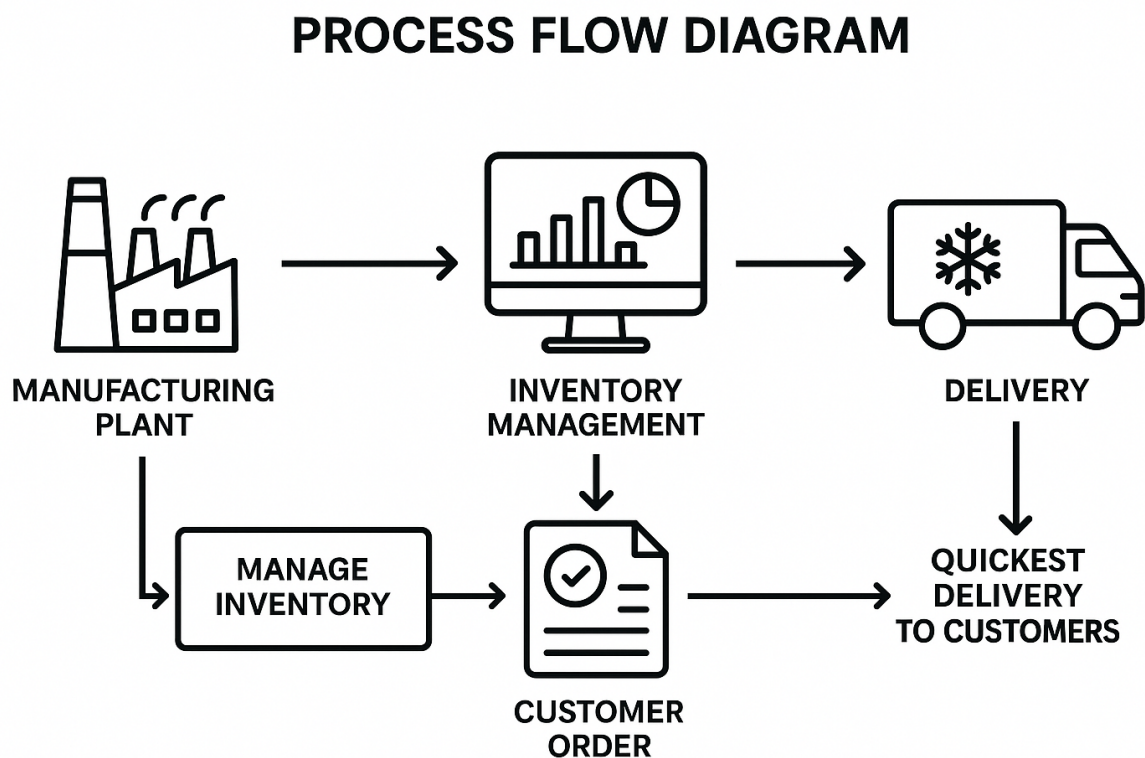
Phase	Duration
Requirements	3 weeks
Design	4 weeks
Development	10 weeks
Testing	4 weeks

Deployment	3 weeks
Maintenance (Ongoing)	Continuou s

8. Success Criteria

- Reduce delivery time by 20–30%.
- Improve stock accuracy to 95%+.
- Reduce wastage by 15–20%.
- Achieve 95%+ on-time deliveries.

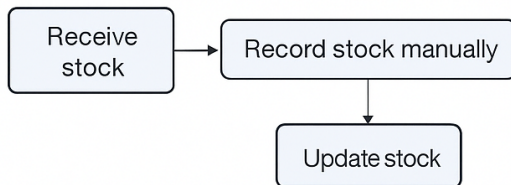
C. Process Flow Diagram (high level)



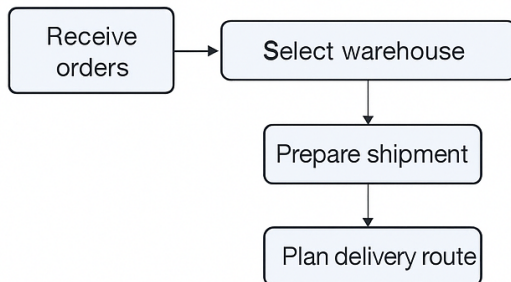
BUSINESS REQUIREMENT DOCUMENT (BRD)

AS-IS Process

AS-IS Inventory Management

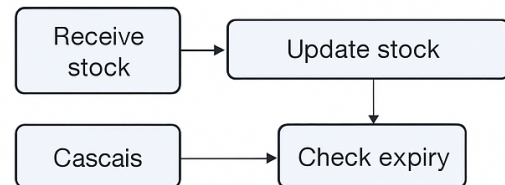


AS-IS Order Management & Delivery

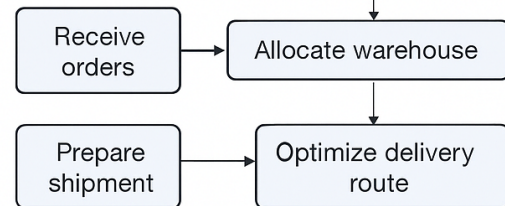


TO-BE Process

TO-BE Inventory Management



TO-BE Order Management & Delivery



Assignment 2

Business Requirement Document (BRD) & Software Requirement Specification (SRS) Ticketing System

Business Requirement Document (BRD)

1. Executive Summary

The company currently handles support and service requests through emails, phone calls, and spreadsheets. This leads to inefficiencies, missed SLAs, lack of accountability, and poor customer experience.

The proposed Ticketing System will provide a centralized, automated platform for logging, tracking, and resolving customer and internal issues. The system will ensure faster response, better SLA compliance, and data-driven decision-making.

2. Business Objectives

- Provide centralized ticket logging and tracking for customers and employees.
- Improve service efficiency and SLA compliance.
- Enable real-time communication between users and support agents.
- Provide dashboards and analytics for management.
- Enhance customer satisfaction through transparency and timely updates.

3. AS-IS Process (Current State)

- Tickets are raised via email/phone, with no standard format.
- No unique ticket ID – difficult to track progress.
- Assignment is manual, causing delays.
- No central repository of past tickets/solutions.
- SLA breaches go unnoticed until escalated manually.
- Reporting is spreadsheet-based, time-consuming, and inaccurate.

4. TO-BE Process (Future State)

- Tickets logged via web portal, mobile app, or email integration.
- Each ticket gets a unique ID.
- Auto-assignment based on category, priority, and workload balancing.
- Customers can track ticket status in real-time.
- SLA monitoring with automatic escalations.
- Knowledge base for common issues to improve resolution.
- Dashboards and analytics for management.

5. Scope

In Scope:

- Ticket lifecycle management (Create → Assign → Track → Resolve → Close).
- SLA definition and automated escalation.
- Notifications via email, SMS, or app.
- Reports and dashboards (SLAs, productivity, ticket trends).
- Multi-role access (Customer, Agent, Manager, Admin).
- Knowledge base integration.

Out of Scope (Future Enhancements):

- Chatbot/AI-driven ticket creation.
- Integration with external CRMs (Salesforce, Zoho, etc.).
- Voice-based ticket creation.

6. Stakeholders

Stakeholder	Role/Interest
Customers/End Users	Log issues and track status
Support Agents	Resolve assigned tickets
Support Manager	Monitor workload, SLA, escalations
IT Administrator	Configure system, manage users
Senior Management	View reports, track KPIs

7. Business Requirements

Functional Requirements:

1. Ticket Management

- Unique ticket ID generation.
- Ticket creation via web, app, email.
- Ticket categorization (Issue type, department, product line).
- Ticket prioritization (High, Medium, Low).
- Status tracking (Open, In-progress, Resolved, Closed).

2. Assignment & Escalation

- Auto-assignment based on category/agent availability.
- Manual reassignment by manager.
- SLA timers (e.g., High priority → response within 2 hrs).
- Escalation alerts on SLA breach.

3. Communication

- Comment threads on tickets.
- Email/SMS/app notifications for updates.
- Attachments supported (images, PDFs, logs).

4. Dashboards & Reporting

- Ticket backlog view.
- SLA compliance reports.
- Agent productivity dashboards.
- Trend analysis by category/time.

5. Knowledge Base

- Predefined FAQs and solutions.
- Link knowledge base articles to tickets.

Non-Functional Requirements:

- Scalability: Handle 10,000+ tickets/day.

- Security: Role-based access, data encryption, audit trail.
- Availability: 99.9% uptime.
- Performance: Response time <2 seconds for queries.
- Usability: Intuitive web & mobile interface.

8. Gap Analysis (AS-IS vs TO-BE)

Area	AS-IS	TO-BE
Ticket Logging	Email/phone	Web/App/Email integration
Tracking	Manual	Real-time with status updates
Assignment	Manual	Auto-assignment with rules
SLA Monitoring	Absent	Automated with escalation
Reporting	Spreadsheet	Dashboards & Analytics
Knowledge Base	None	Centralized FAQs & solutions

9. Risks & Dependencies

- User adoption resistance.
- Integration with existing IT systems may face delays.
- Data migration challenges from spreadsheets.
- Dependence on email/SMS gateways for notifications.

10. Success Metrics

- SLA compliance improved by 30%.
- Average resolution time reduced by 25%.
- 90%+ tickets updated with real-time status.
- Increase in customer satisfaction (CSAT) score by 20%.

Software Requirement Specification (SRS)

1. Introduction

1.1 Purpose

Define the functional and non-functional requirements for a centralized Ticketing System that logs, tracks, and resolves support issues.

1.2 Intended Users

- Customers

- Support Agents
- Support Managers
- Admins

1.3 Assumptions

- Internet access available to all users.
- System hosted on cloud.
- Standard browsers and mobile devices supported.

2. Functional Requirements

2.1 Ticket Lifecycle

- Create: User submits ticket.
- Assign: Auto/manual assignment.
- Track: Status updates visible to customer.
- Resolve: Agent provides solution.
- Close: Ticket closed after user confirmation.

2.2 Roles & Permissions

- Customer: Create/view own tickets, add comments.
- Agent: View assigned tickets, update, resolve.
- Manager: Reassign, monitor SLA, generate reports.
- Admin: Configure categories, SLAs, manage users.

2.3 Notifications

- Ticket created.
- Status updated.
- SLA breach alert.
- Resolution notification.

2.4 Reporting

- Open vs closed tickets.
- SLA compliance metrics.
- Agent workload distribution.
- Category-wise issue trends.

3. Non-Functional Requirements (NFRs)

- Performance: Response <2 sec.
- Scalability: Support multi-location deployment.
- Availability: 99.9% uptime.
- Security: Authentication, RBAC, encrypted data.
- Maintainability: Modular architecture for easy upgrades.

4. System Flow (High Level)

Step 1: User creates ticket →

Step 2: System assigns ticket →

Step 3: Agent updates ticket progress →

Step 4: SLA monitored, escalations triggered if needed →

Step 5: Ticket resolved →

Step 6: Customer confirms closure →

Step 7: Reports generated

5. Dependencies

- Email/SMS gateway.
- Cloud hosting provider.
- Existing user directory (for authentication).

A. Introduction Letter (Business Analyst → Client)

Subject: Kick-off: Business Understanding for Dairy Inventory & Delivery Platform

Dear ,

I'm , the Business Analyst assigned to lead discovery and collaborate with your operations, supply chain, and IT teams. My role is to translate your business goals—accurate inventory and fastest delivery—into clear, testable requirements and an implementation plan.

Over the next few weeks, I will:

- Map AS-IS processes at plants, warehouses, and last-mile operations
- Align TO-BE workflows and success metrics (shrinkage, delivery time)
- Define data, integrations (ERP, e-commerce, TMS, IoT), and compliance requirements
- Facilitate workshops and share prototypes for feedback

Proposed first steps:

1. 90-min kick-off (scope, milestones)
2. Plant/DC walkthrough (virtual or onsite)
3. Data & integration deep-dive (SKU/batch/expiry, orders, routes)

Please share stakeholder contacts and any existing SOPs, reports, and integration docs. Looking forward to partnering with you.

Best regards,

| Business Analyst

<Phone/Email>

B. Brief BRD & SRS —

Ticketing System

(Support/ITSM)

BRD (Brief)

- Objective: Centralize customer/IT support with SLAs, assignments, and analytics.
- Scope: Ticket creation (portal/email/API), triage, auto-assignment, SLA timers, status lifecycle, comments, attachments, knowledge base link, CSAT survey, reports.
- KPIs: First Response Time (FRT), Resolution Time, SLA breach rate, Reopen rate, CSAT/NPS, backlog age.
- Risks: Low adoption → targeted training; vague categories → taxonomy workshop.

SRS (Brief)

Functional Requirements

- FR-01 Create ticket (user portal, email parser, API); capture requester, category, priority, impact, description, attachments.
- FR-02 Assignment engine: skill/queue-based; round-robin fallback.
- FR-03 SLA engine: start/stop/pause on statuses; holidays & business hours calendars.
- FR-04 Status lifecycle: New → In-Progress → Pending-Customer → Resolved → Closed; Reopen permitted.
- FR-05 Collaboration: comments (internal/external), @mentions, file attachments, tags.
- FR-06 Knowledge suggestions on creation and during triage.
- FR-07 Notifications: email/SMS/push on key events.
- FR-08 Audit log for all changes.
- FR-09 Reports & dashboards; export to CSV.
- FR-10 RBAC: Admin, Agent, Supervisor, Requester.

Non-Functional Requirements

- Availability 99.9%, p95 API < 300ms, SSO (SAML/OAuth2), GDPR/PII controls, OWASP Top-10 mitigations, data retention rules.

Interfaces

- Email (IMAP/API), HRIS/CRM for user sync, webhook for CSAT tools, BI for analytics.

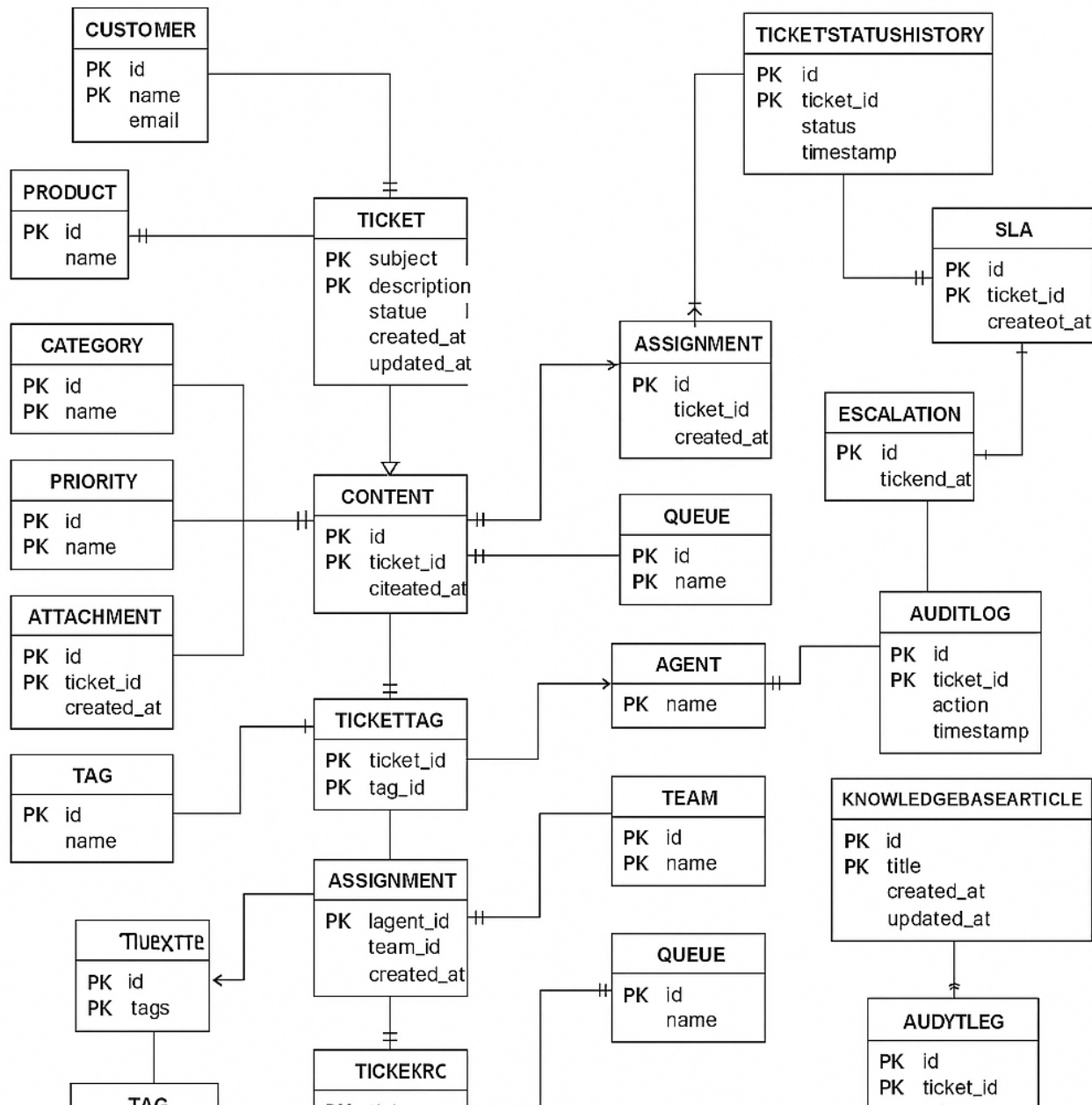
C

An ERD (Entity–Relationship Diagram) is a type of visual model used in database and system design. It shows the entities (things we want to store data about), their attributes, and the relationships between them.

Key Components of an ERD:

1. Entities → Represent real-world objects or concepts.
 - Example: Customer, Ticket, Agent, Product.
 - Usually drawn as rectangles.

2.



3. Attributes → Properties or details of entities.

- Example: Customer → name, email, phone.
- Shown inside or attached to the entity.

4.

5. Relationships → Show how entities are connected.

- Example: A Customer creates many Tickets.
- Represented by lines/diamonds between entities.

6.

7. Cardinality → Defines the number of instances in the relationship.

- 1..1 → One-to-one (e.g., each ticket has one SLA).
- 1..* → One-to-many (e.g., one customer can raise many tickets).
- .. → Many-to-many (e.g., agents can handle many tickets, and tickets can be reassigned to many agents).

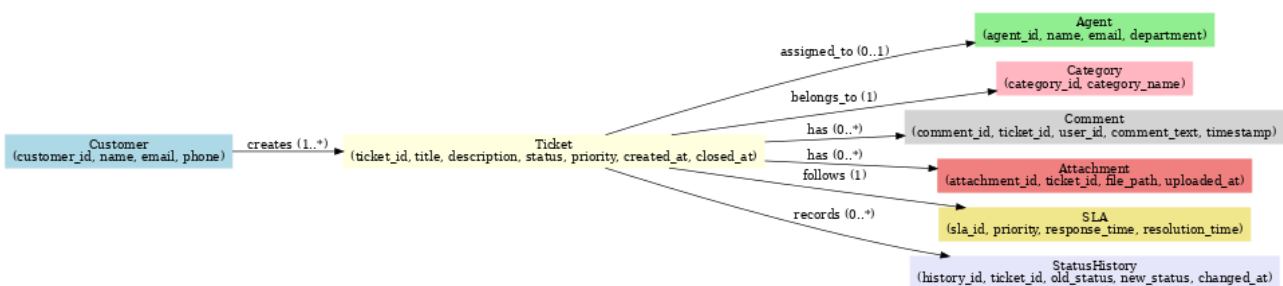
8.

✓ Why ERDs are Important

- Help visualize data structure before building the system.
- Ensure there are no data duplication or missing links.
- Make communication easier between business analysts, developers, and DBAs.
- Serve as a reference document for system development and maintenance.

⚡ Example (Ticketing System ERD – simplified):

- Customer creates → Ticket.
- Ticket is assigned to → Agent.
- Ticket has → Category, Comments, Attachments, SLA.



D. User Stories — Shopping on an E-commerce Site

Epic: Place Order for Ice-Cream & Dairy

- As a customer, I want to discover products, add to cart, pay securely, and receive timely delivery, so that I get fresh products conveniently.

Story 1: Browse & Search

As a customer I want to search/filter by category, price, dietary tags (sugar-free) so that I find the right products.

Acceptance Criteria

- Given I enter a query, when results load, then I see relevant products with images, price, stock & rating.
- Filters by category/price/dietary are combinable and persist on pagination.

Story 2: Product Detail & Availability

As a customer I want to view product details (ingredients, allergens, shelf life) and see delivery slot availability.

AC

- FEFO batch determines available slots by my pincode; out-of-stock shows “Notify me”.

Story 3: Cart & Substitutions

As a customer I want to add items to cart and select substitution preferences.

AC

- Cart recalculates totals, taxes, and cold-chain delivery fee; suggests substitutions for OOS items.

Story 4: Address & Slot Selection

As a customer I want to save addresses and choose a delivery slot.

AC

- Slots respect cut-off times and route capacity; blocked slots cannot be selected.

Story 5: Payment

As a customer I want multiple payment options (UPI, cards, COD).

AC

- Successful payment creates an order and sends confirmation; failed payment keeps cart for 24h.

Story 6: Order Tracking

As a customer I want live order status and driver ETA.

AC

- Tracking page shows packed/out-for-delivery with map ETA; temperature compliance badge on delivery.

Story 7: Returns/Refund

As a customer I want to raise issues (melted, damaged, missing).

AC

- Return request creates a ticket linked to the order; refund or replacement workflow triggers.

Story 8: Admin Inventory

As an ops manager I want to view stock by batch/expiry and push near-expiry promotions.

AC

- Items within X days of expiry appear in “Act Now” list with bulk markdown action.

Got it! To make this comprehensive, I will prepare 40 user stories for an e-commerce shopping system with the following details for each:

- User Story (As a..., I want..., So that...)
- Business Value (BV) – the importance to the business (scale 1–10)
- Cost/Complexity (CP) – effort/complexity (scale 1–10)
- Tasks – steps needed to implement
- Acceptance Criteria (AC) – conditions for story completion

Here's a detailed example batch for 40 user stories:

1. User Registration

- Story: As a new user, I want to register an account so that I can place orders.
- BV: 9
- CP: 3
- Tasks:
 1. Create registration form (name, email, password)
 2. Validate email format
 3. Send verification email
- AC: User can successfully create an account and receives a verification email.

2. User Login

- Story: As a user, I want to log in so that I can access my account.

- BV: 10
- CP: 2
- Tasks:
 1. Implement login page
 2. Validate credentials
 3. Maintain session
-
- AC: Valid credentials allow login; invalid credentials show error.

3. Password Recovery

- Story: As a user, I want to reset my password if forgotten.
- BV: 8
- CP: 3
- Tasks:
 1. Create “Forgot Password” flow
 2. Send reset email
 3. Update password securely
-
- AC: User can reset password via email link.

4. Browse Products

- Story: As a user, I want to browse products by category so I can find items easily.
- BV: 9
- CP: 4
- Tasks:
 1. Display product categories
 2. Show product lists with images, prices
 3. Pagination for large catalogs
-
- AC: User can view products filtered by category.

5. Product Search

- Story: As a user, I want to search products by keywords so that I can quickly find items.
- BV: 10
- CP: 5
- Tasks:

1. Implement search bar
 2. Connect search to product database
 3. Show search results ranked by relevance
- - AC: User sees relevant products based on search input.

6. Product Details

- Story: As a user, I want to view product details so I can know more before buying.
- BV: 9
- CP: 3
- Tasks:
 1. Create product detail page
 2. Show description, price, reviews, availability
-
- AC: Product page displays all relevant information correctly.

7. Add to Cart

- Story: As a user, I want to add items to a cart so I can purchase multiple items together.
- BV: 10
- CP: 4
- Tasks:
 1. Implement “Add to Cart” button
 2. Store cart items in session/database
-
- AC: Item is successfully added to the cart.

8. View Cart

- Story: As a user, I want to view my cart so that I can review my items.
- BV: 10
- CP: 3
- Tasks:
 1. Create cart page
 2. Show list of items, quantity, subtotal
-
- AC: User sees all items in cart with correct pricing.

9. Update Cart

- Story: As a user, I want to update item quantity or remove items so that I can manage my cart.
- BV: 9
- CP: 4
- Tasks:
 1. Add quantity change functionality
 2. Add “Remove Item” button
-
- AC: Cart updates quantities and totals correctly.

10. Apply Discount Code

- Story: As a user, I want to apply a discount code so I can get a price reduction.
- BV: 8
- CP: 5
- Tasks:

1. Add discount code input
 2. Validate code and apply discount
- - AC: Valid code reduces total; invalid code shows error.

11. Checkout Process

- Story: As a user, I want to checkout items so I can complete my purchase.
- BV: 10
- CP: 6
- Tasks:
 1. Create checkout page
 2. Collect shipping info, payment method
 3. Confirm order
-
- AC: User can successfully complete checkout and see order summary.

12. Payment Integration

- Story: As a user, I want to pay using multiple payment methods so that I have convenience.
- BV: 10
- CP: 7
- Tasks:
 1. Integrate payment gateways (UPI, Card, Wallet)
 2. Handle success/failure responses
-
- AC: Payment is processed and confirmed; failures are notified.

13. Order Confirmation

- Story: As a user, I want an order confirmation email so I know my purchase is successful.
- BV: 9
- CP: 3
- Tasks:
 1. Generate order confirmation email
 2. Include order summary
-
- AC: User receives email with correct order details.

14. Track Order

- Story: As a user, I want to track my order so that I know the delivery status.
- BV: 8
- CP: 5
- Tasks:
 1. Implement order status API
 2. Show progress in user account
-
- AC: User can see live order status updates.

15. Product Reviews

- Story: As a user, I want to leave reviews so others can benefit from my experience.
- BV: 7
- CP: 4
- Tasks:
 1. Add review submission form

2. Display reviews on product page

-
- AC: Review is visible on product page after submission.

16. Wishlist

- Story: As a user, I want to add items to wishlist so I can buy them later.
- BV: 8
- CP: 4
- Tasks:
 1. Implement “Add to Wishlist”
 2. Save wishlist items in account
-
- AC: Wishlist displays saved items correctly.

17. Recently Viewed Items

- Story: As a user, I want to see recently viewed items so I can easily find products again.

- BV: 7
- CP: 3
- Tasks:
 1. Track product views
 2. Display list in account/homepage
-
- AC: User sees last 5–10 viewed products.

18. Product Recommendations

- Story: As a user, I want recommended products based on my browsing so I can find items I may like.
- BV: 9
- CP: 6
- Tasks:
 1. Implement recommendation engine
 2. Display recommendations on product/cart page
-
- AC: Recommendations appear based on user behavior.

19. Category Filters

- Story: As a user, I want to filter products by price, brand, and rating so I can narrow options.
- BV: 9
- CP: 5
- Tasks:
 1. Add filter sidebar
 2. Implement filtering logic
-
- AC: Products display correctly based on selected filters.

20. Sort Products

- Story: As a user, I want to sort products by price or rating so I can find the best options.
- BV: 8
- CP: 3
- Tasks:
 1. Add sort dropdown
 2. Apply sorting logic
-

- AC: Products sort correctly by selected criteria.

21. Multiple Addresses

- Story: As a user, I want to save multiple shipping addresses for faster checkout.
- BV: 7
- CP: 4
- Tasks:
 1. Implement address management
 2. Allow selection during checkout
-
- AC: User can save, edit, and select addresses.

22. Guest Checkout

- Story: As a user, I want to checkout without registration so I can quickly buy.
- BV: 8
- CP: 4

- Tasks:
 1. Enable guest checkout flow
 2. Collect email and shipping info
-
- AC: Guest user can place order without creating an account.

23. Inventory Check

- Story: As a user, I want to know product stock availability so I can avoid ordering unavailable items.
- BV: 9
- CP: 3
- Tasks:
 1. Show stock status on product page
-
- AC: Stock levels are accurate; out-of-stock items cannot be added.

24. Return/Refund

- Story: As a user, I want to return items and get a refund if needed.
- BV: 8
- CP: 6
- Tasks:
 1. Implement return request form
 2. Process refunds via payment gateway
-
- AC: User can request return; refund is processed after approval.

25. Email Notifications

- Story: As a user, I want email notifications for order updates so I stay informed.
- BV: 8
- CP: 3
- Tasks:
 1. Trigger emails for status changes
-
- AC: User receives emails for order placement, shipment, and delivery.

26. Mobile-Friendly Design

- Story: As a user, I want the website mobile-friendly so I can shop from any device.
- BV: 9
- CP: 5
- Tasks:
 1. Responsive UI design
 2. Test on multiple devices
-
- AC: Website functions correctly on smartphones and tablets.

27. Social Media Login

- Story: As a user, I want to login via social media so I can save time.
- BV: 7
- CP: 4
- Tasks:
 1. Integrate Google/Facebook login
-
- AC: User can login using social accounts successfully.

28. Multi-Currency

- Story: As a user, I want to see prices in my currency so I can understand costs better.
- BV: 6
- CP: 6
- Tasks:
 1. Implement currency selection
 2. Convert prices dynamically
-
- AC: Prices display correctly based on selected currency.

29. Order History

- Story: As a user, I want to see my past orders so I can reorder easily.
- BV: 8
- CP: 3

- Tasks:
 1. Display past orders in account
-
- AC: User sees a complete order history with details.

30. Live Chat Support

- Story: As a user, I want live chat support so I can resolve queries instantly.
- BV: 7
- CP: 6
- Tasks:
 1. Integrate chat widget
 2. Connect to support team
-
- AC: Users can chat live; responses are received.

31. Product Comparison

- Story: As a user, I want to compare products so I can make informed choices.
- BV: 8
- CP: 5
- Tasks:
 1. Implement “Add to Compare” feature
 2. Show side-by-side comparison
-
- AC: Selected products display comparison accurately.

32. Recently Purchased Recommendations

- Story: As a user, I want recommendations based on my past purchases so I can buy complementary products.
- BV: 7
- CP: 5
- Tasks:
 1. Track purchases
 2. Display recommendations
-
- AC: Recommendations show relevant products.

33. Loyalty Points

- Story: As a user, I want to earn loyalty points for purchases so I can redeem them later.
- BV: 8
- CP: 6
- Tasks:
 1. Track points per order
 2. Allow redemption at checkout
-
- AC: Points are added and redeemed correctly.

34. Gift Cards

- Story: As a user, I want to buy gift cards so I can give them to others.
- BV: 7
- CP: 5
- Tasks:
 1. Implement gift card purchase

2. Email gift card codes

-
- AC: Gift card is delivered correctly to recipient email.

35. Product Zoom

- Story: As a user, I want to zoom into product images so I can see details clearly.
- BV: 6
- CP: 3
- Tasks:

1. Implement image zoom feature

-
- AC: User can zoom images without distortion.

36. Out-of-Stock Notifications

- Story: As a user, I want to get notified when an out-of-stock product is available.
- BV: 7

- CP: 4
- Tasks:
 1. Add “Notify Me” feature
 2. Send email when back in stock
-
- AC: Notification is sent when product is available.

37. Multiple Payment Options

- Story: As a user, I want to use wallet, credit/debit, and net banking so I have flexible payment options.
- BV: 9
- CP: 7
- Tasks:
 1. Integrate multiple payment methods
-
- AC: Payment succeeds with any supported method.

38. Order Cancellation

- Story: As a user, I want to cancel an order within a certain time so I can avoid unwanted purchases.
- BV: 8
- CP: 4
- Tasks:
 1. Implement cancellation option
 2. Process refunds automatically
-
- AC: Order is cancelled and user receives confirmation/refund.

39. Delivery Time Estimates

- Story: As a user, I want to see estimated delivery time so I can plan accordingly.
- BV: 8
- CP: 3
- Tasks:
 1. Calculate delivery based on location
-
- AC: Delivery estimate displays correctly during checkout.

40. Push Notifications

- Story: As a user, I want push notifications for offers and updates so I don't miss deals.
- BV: 7
- CP: 5
- Tasks:
 1. Implement push notification subscription
 2. Send notifications for promotions
-
- AC: User receives push notifications correctly.