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

**Business Requirements Document (BRD)**

**Project Name:** Ticketing System  
**Client:** XYZ   
**Date:** 20.02.25  
**Prepared By:** P Singa Ram

**1. Introduction**

The **Ticketing System** is designed to manage and track customer or internal support requests efficiently. It enables users to create, assign, and resolve tickets while ensuring proper communication between stakeholders. The system aims to improve response times, enhance issue resolution, and provide insights into operational performance.

**2. Goal**

The goal of this project is to develop a **centralized ticketing system** that allows users to report issues, monitor their progress, and ensure timely resolution by support teams.

**3. Objectives**

* **Streamline Ticket Creation & Management**: Provide a structured workflow for users to submit, update, and track tickets.
* **Enhance Communication**: Enable notifications and collaboration between users and support agents.
* **Automate Ticket Assignment**: Assign tickets based on predefined rules and agent availability.
* **Improve Resolution Times**: Reduce delays through prioritization and escalation mechanisms.
* **Provide Analytics & Reporting**: Generate reports on ticket trends, resolution times, and performance metrics.

**4. Use Case Diagram**

A diagram of a ticketing system

AI-generated content may be incorrect.

**5. Use Case Specification**

**Use Case: Create Support Ticket**

**Use Case Specification: Create Support Ticket**

#### **1. Use Case Name**

Create Support Ticket

#### **2. Use Case ID**

UC-101

#### **3. Actors**

Primary Actor: Customer

Secondary Actor: Support Agent (for notifications)

#### **4. Description**

This use case describes how a customer creates a support ticket to report an issue or request assistance. The system records the ticket and notifies the support team.

#### **5. Preconditions**

The customer must be logged into the system.

The customer must have a valid account.

#### **6. Postconditions**

A new support ticket is created and stored in the system.

The support team is notified of the new ticket.

The customer receives a confirmation of the ticket creation.

#### **7. Basic Flow (Main Success Scenario)**

Customer navigates to the "Support" section of the system.

Customer clicks on "Create New Ticket."

System displays a form for ticket details.

Customer enters the following details:

Subject

Description of the issue

Priority (Low, Medium, High)

Category (e.g., Billing, Technical, General)

Customer submits the form.

System validates the input data.

System creates a new ticket with a unique ticket ID.

System notifies the support team via email or dashboard notification.

System displays a confirmation message to the customer.

System logs the ticket in the database.

#### **8. Alternative Flows**

**Alternative Flow 1: Customer cancels ticket creation**

Customer clicks "Cancel" before submitting the form.

System returns the customer to the Support homepage.

**Alternative Flow 2: Customer attaches a file**

Customer uploads a file (e.g., screenshot, document) to provide additional context.

System validates the file size and type.

System attaches the file to the ticket.

#### **9. Exception Flows**

**Exception 1: Invalid or missing date**

System detects missing or invalid data (e.g., empty subject or description).

System displays an error message and prompts the customer to correct the input.

**Exception 2: File upload fails**

System rejects the file if it exceeds the size limit or is an unsupported type.

System displays an error message and prompts the customer to upload a valid file.

#### **10. Special Requirements**

The system must support file attachments up to 10MB in size.

Notifications to the support team must be sent within 1 minute of ticket creation.

The ticket ID must be unique and follow the format: TICKET-XXXXXX (where XXXXXX is a 6-digit number).

#### **11. Assumptions**

The customer has access to the internet and a valid account.

The support team is available to handle incoming tickets.

#### **12. Notes**

The system should allow customers to save a draft of the ticket if they need to complete it later.

#### **13. Attachments**

Wireframe of the "Create New Ticket" form.

Example of a ticket confirmation email.

**Use Case Specification: Update Ticket Status**

#### **1. Use Case Name**

Update Ticket Status

#### **2. Use Case ID**

UC-102

#### **3. Actors**

Primary Actor: Support Agent

Secondary Actor: System (for automated status updates)

#### **4. Description**

This use case describes how a support agent or the system updates the status of a ticket (e.g., Open, In Progress, Resolved). The status change is logged, and relevant stakeholders are notified.

#### **5. Preconditions**

The ticket must exist in the system.

The support agent must be logged in and have the necessary permissions to update the ticket status.

#### **6. Postconditions**

The ticket status is updated in the system.

The status change is logged in the ticket history.

Relevant stakeholders (e.g., customer, support team) are notified of the status change.

#### **7. Basic Flow (Main Success Scenario)**

Support Agent navigates to the ticket details page.

System displays the current status of the ticket.

Support Agent selects a new status from the available options (e.g., Open, In Progress, Resolved).

Support Agent provides an optional comment explaining the status change.

Support Agent clicks "Update Status."

System validates the new status.

System updates the ticket status in the database.

System logs the status change in the ticket history.

System notifies the customer and other relevant stakeholders (e.g., support team) of the status change.

System displays a confirmation message to the Support Agent.

#### **8. Alternative Flows**

**Alternative Flow 1: System updates status automatically.**

System detects a condition that requires an automatic status update (e.g., ticket resolved after a customer confirms resolution).

System updates the status without manual intervention.

System logs the change and notifies stakeholders.

**Alternative Flow 2: Support Agent cancels the update**

Support Agent clicks "Cancel" before submitting the update.

System returns the Support Agent to the ticket details page without making any changes.

#### **9. Exception Flow**

**Exception 1: Invalid status selected**

System detects an invalid status (e.g., a status that does not exist or is not allowed for the current workflow).

System displays an error message and prompts the Support Agent to select a valid status.

**Exception 2: Permission denied**

Support Agent does not have the necessary permissions to update the ticket status.

System displays an error message and denies access to the status update feature.

#### **Special Requirements**

#### The system must log all status changes with a timestamp and the name of the Support Agent (or "System" for automated updates).

Notifications must be sent within 1 minute of the status change.

The system must enforce workflow rules (e.g., a ticket cannot move from "Resolved" back to "Open" without approval).

#### **11. Assumptions**

The Support Agent has access to the ticket details page.

The system has a predefined list of valid statuses (e.g., Open, In Progress, Resolved, Closed).

#### **12. Notes**

The system should allow administrators to configure workflow rules for status transitions.

Notifications should include a link to the ticket for easy access.

#### **13. Attachments**

Wireframe of the ticket details page with the status update feature.

Example of a status change notification email.

**Use Case Specification: Close Ticket**

#### **1. Use Case Name**

**Close Ticket**

#### **2. Use Case ID**

UC-102

#### **3. Actors**

* **Primary Actor**: Support Agent
* **Secondary Actor**: Customer (for notifications)

#### **4. Description**

This use case describes how a support agent closes a resolved support ticket. The system updates the ticket status to "Closed" and notifies the customer.

#### **5. Preconditions**

* The ticket must have a status of "Resolved."
* The support agent must be logged into the system.

#### **6. Postconditions**

* The ticket status is updated to "Closed."
* The customer receives a confirmation of ticket closure.
* The ticket history is archived for future reference.

#### **7. Basic Flow (Main Success Scenario)**

1. Support Agent logs into the system.
2. Support Agent navigates to the **"Ticket Management"** section.
3. Support Agent selects a **resolved ticket** from the list.
4. Support Agent clicks on the **"Close Ticket"** button.
5. System prompts for confirmation.
6. Support Agent confirms the closure.
7. System updates the ticket status to **"Closed"**.
8. System logs the closure activity.
9. System sends a **notification email/SMS** to the customer about ticket closure.
10. The use case ends successfully.

#### **8. Alternative Flows**

**Alternative Flow 1: Ticket Cannot Be Closed**

* If the ticket is not in the **"Resolved"** state, the system displays an error message:  
  "Ticket must be resolved before closing."
* The agent must resolve the issue before attempting closure.

#### **9. Exception Flows**

**Exception 1: System Error Occurs**

* If the system encounters an error while closing the ticket, it displays:  
  "System error. Please try again later."
* The agent can retry later or contact IT support

U**se Case Specification: Generate Report**

#### **1. Use Case Name**

**Generate Report**

#### **2. Use Case ID**

**UC-103**

#### **3. Actors**

* **Primary Actor:** Administrator
* **Secondary Actor:** Support Agent (for viewing reports)

#### **4. Description**

This use case describes how an administrator generates system reports to analyze support ticket trends, response times, and resolution efficiency. The system processes selected criteria and generates a report in the desired format.

#### **5. Preconditions**

* The administrator must be logged into the system.
* The system must have sufficient historical ticket data.

#### **6. Postconditions**

* A report is generated based on the selected criteria.
* The administrator can view, download, or export the report.

#### **7. Basic Flow (Main Success Scenario)**

1. The administrator logs into the system.
2. The administrator navigates to the **"Reports"** section.
3. The administrator selects the **report type** (e.g., Ticket Summary, Resolution Time, Agent Performance).
4. The administrator selects **filter criteria** (e.g., date range, department, priority level).
5. The administrator selects the desired **report format** (e.g., PDF, Excel, CSV).
6. The administrator clicks **"Generate Report"**.
7. The system fetches relevant data and generates the report.
8. The system displays a **preview of the report**.
9. The administrator downloads or exports the report as needed.

#### **8. Alternative Flows**

**Alternative Flow 1: No Data Found**

* If no matching records are found, the system displays a message:  
  "No data available for the selected criteria."
* The administrator can modify the filter options and retry.

**Alternative Flow 2: Report Scheduled for Later**

* If the report is too large, the administrator can schedule it for background processing.
* The system notifies the administrator when the report is ready.

#### **9. Exception Flows**

**Exception 1: System Error Occurs**

* If the system fails to generate the report, it displays an error message:  
  "An error occurred while processing the report. Please try again later."
* The system logs the error for IT review.

**Exception 2: Unauthorized Access**

* If a non-admin user tries to generate a report, the system denies access and logs the attempt.

#### **10. Special Requirements**

* The system should allow **scheduled reports** to be automatically generated at predefined intervals.
* Reports must be stored securely and should be **accessible only to authorized personnel**.
* Large reports should be processed asynchronously to avoid system slowdown.

#### **11. Assumptions**

* The system has **adequate historical data** for reporting.
* The administrator has permission to generate and view reports.

#### **12. Notes**

* A dashboard widget can be added for quick access to frequently generated reports.
* Reports should be visually structured with charts and graphs for better insights.

#### **10. Special Requirements**

* The system must log **who closed the ticket** and **when**.
* Closed tickets must be stored for **audit and reporting**.
* Customers should be able to reopen a closed ticket within a defined period (e.g., 7 days).

#### **11. Assumptions**

* The support team follows the resolution workflow before closing tickets.
* Customers can provide feedback upon ticket closure.

#### **12. Notes**

* A ticket closure report should be available for supervisors.
* The system may include an **auto-close feature** if a resolved ticket is inactive for a specified time (e.g., 72 hours).

#### **1. Use Case Name**

**Search for Tickets**

#### **2. Use Case ID**

**UC-104**

#### **3. Actors**

* **Primary Actor:** Support Agent
* **Secondary Actor:** Administrator

#### **4. Description**

This use case describes how a support agent searches for tickets in the system based on various filters such as ticket ID, customer name, issue type, priority, and status. The system retrieves and displays matching results for review and further actions.

#### **5. Preconditions**

* The support agent must be logged into the system.
* The system must have existing tickets stored in the database.

#### **6. Postconditions**

* A list of tickets matching the search criteria is displayed.
* The support agent can select and open a ticket for further action.

#### **7. Basic Flow (Main Success Scenario)**

1. The support agent logs into the system.
2. The agent navigates to the **"Search Tickets"** section.
3. The agent enters search criteria (e.g., ticket ID, customer name, issue type, priority, status).
4. The system processes the query and retrieves matching tickets.
5. The system displays the search results in a sortable table.
6. The agent clicks on a ticket to view details.
7. The system opens the selected ticket for review.

#### **8. Alternative Flows**

**Alternative Flow 1: No Matching Results**

* If no tickets match the search criteria, the system displays:  
  "No tickets found. Please refine your search."
* The agent can modify search criteria and retry.

**Alternative Flow 2: Search by Date Range**

* Instead of a specific ticket ID or customer name, the agent selects a **date range**.
* The system retrieves all tickets created within that range.

#### **9. Exception Flows**

**Exception 1: System Timeout or Database Error**

* If the system fails to retrieve ticket data, it displays:  
  "An error occurred while searching for tickets. Please try again later."
* The system logs the error for IT support.

#### **10. Special Requirements**

* The system should support **auto-suggestions** while entering search parameters.
* Search results should be **paginated** to handle large datasets.
* Agents should be able to **export search results** to Excel or CSV.

#### **11. Assumptions**

* The support team frequently searches for tickets using **different filters**.
* The database maintains an **efficient indexing system** for faster queries.

#### **12. Notes**

* The system should allow **advanced search options** (e.g., search within ticket descriptions).
* **Sorting and filtering** should be available in the results table.

1. **Business Requirements**

**6.1 Functional Requirements**

**Ticket Creation & Management**

* Users can create new tickets with details such as issue type, priority, and description.
* Support agents can update ticket statuses (e.g., Open, In Progress, Resolved, Closed).
* Tickets can be assigned to specific agents or teams based on predefined rules.
* Users can view and track the status and history of their submitted tickets.
* Support agents can add internal notes, resolutions, and attachments to tickets.

**Notification & Communication**

* Automated email/SMS notifications for ticket creation, status updates, and resolution.
* In-app notifications for real-time updates on ticket progress.
* Collaboration tools for communication between users and support agents within the ticket.

**Ticket Assignment & Routing**

* Automated ticket assignment based on agent availability, skills, and workload.
* Escalation mechanisms for high-priority or overdue tickets.
* Manual reassignment of tickets by supervisors or managers.

**Reporting & Analytics**

* Dashboard displaying key metrics such as ticket volume, resolution time, and agent performance.
* Customizable reports on ticket trends, user satisfaction, and operational efficiency.
* Export reports in various formats (e.g., Excel, PDF).

**User Roles & Access Control**

* Role-based access control for users, support agents, supervisors, and administrators.
* Permissions to view, edit, or delete tickets based on roles.
* Secure login with multi-factor authentication.

**6.2 Non-Functional Requirements**

**Performance & Scalability**

* The system should support concurrent access by at least 1000 users without performance degradation.
* Scalability to handle increased ticket volume during peak periods.

**Security**

* Secure data storage and transmission with encryption (e.g., SSL/TLS).
* Compliance with data privacy regulations (e.g., GDPR).
* Regular security audits and vulnerability assessments.

**Usability & Accessibility**

* User-friendly interface with intuitive navigation.
* Accessible to users with disabilities, compliant with WCAG 2.1 standards.
* Responsive design for compatibility with desktop, tablet, and mobile devices.

**Reliability & Availability**

* System availability of 99.9% to ensure continuous access.
* Automatic failover and disaster recovery mechanisms.
* Regular data backups to prevent data loss.

**Integration & Compatibility**

* Integration with third-party systems (e.g., CRM, email, SMS gateways).
* Compatibility with major browsers (e.g., Chrome, Firefox, Safari, Edge).
* API support for custom integrations and data synchronization.

**Maintainability & Support**

* Modular architecture for easy maintenance and updates.
* Comprehensive documentation for developers and end-users.
* Dedicated support team for issue resolution and enhancements.

# Make an ERD of creating a support ticket.

A diagram of a program

AI-generated content may be incorrect.