

<b>Date</b>	31/10/2025
<b>Team ID</b>	NM2025TMID02533
<b>Project name</b>	Streamlining Ticket Assignment for Efficient Support Operations
<b>Maximum mark</b>	4 marks

## PROJECT DESIGN PHASE 2

### Data Flow Diagram (DFD) – Streamlining Ticket Assignment for Efficient Support Operations

#### Level 0 – Context Diagram

##### Overview:

This high-level diagram shows how data moves between external entities and the main ticket assignment system.

##### External Entities:

- **Customer/User:** Submits new support tickets.
- **Support Agent:** Receives assigned tickets and updates status.
- **Admin/Manager:** Monitors performance, workload, and system reports.

##### Main Process:

- **Streamlined Ticket Assignment System** – The central process that handles ticket creation, classification, routing, and monitoring.

##### Data Flows:

- From **Customer** → **System**: Ticket details (issue type, priority, description).
- From **System** → **Support Agent**: Assigned ticket notifications and workload updates.
- From **Support Agent** → **System**: Ticket progress, resolution updates, and status changes.
- From **System** → **Admin**: Analytics reports and performance summaries.

#### Level 1 – Detailed Data Flow Diagram

##### Process 1: Ticket Creation

- Input: Ticket data (submitted by user).
- Output: Ticket record stored in the database.
- Data Store: *Ticket Database*.

## **Process 2: Ticket Classification**

- Input: Ticket record from Ticket Database.
- Function: Analyze content using keywords, urgency, and category.
- Output: Ticket category and priority tag.
- Data Store: *Ticket Classification Rules*.

## **Process 3: Ticket Assignment**

- Input: Classified ticket and Agent Database.
- Function: Match ticket with suitable agent based on skills, workload, and availability.
- Output: Assigned ticket sent to agent.
- Data Store: *Agent Profile Database*.

## **Process 4: Ticket Tracking & Update**

- Input: Agent updates ticket status (in-progress, resolved).
- Output: Updated records and progress tracking.
- Data Store: *Ticket Tracking Log*.

## **Process 5: Reporting & Analytics**

- Input: Ticket and agent performance data.
- Function: Generate insights for workload, resolution time, and performance efficiency.
- Output: Dashboards and reports
- for Admin.
- Data Store: *Analytics Database*.

---

## **Data Stores**

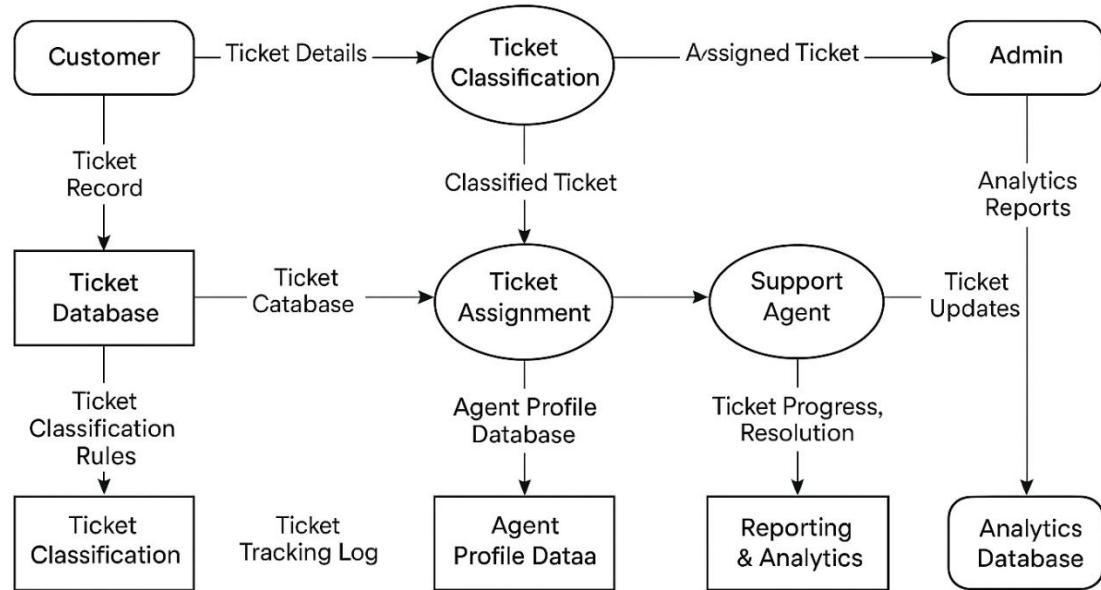
1. **Ticket Database:** Stores all created tickets.
2. **Agent Profile Database:** Stores agent details, skills, and workload.
3. **Ticket Classification Rules:** Contains predefined keywords, rules, and AI models for classification.
4. **Ticket Tracking Log:** Tracks ticket progress and updates.
5. **Analytics Database:** Stores performance and efficiency data for reporting.

## **Key Data Flows**

- **Customer → System:** Ticket details.
- **System → Agent:** Assigned ticket and alerts.
- **Agent → System:** Ticket updates and resolution info.
- **System → Admin:** Performance analytics and reports.

## EXAMPLE

**Data Flow Diagram**  
Streamlining Ticket Assignment for Efficient Support Operations



## User Stories

### 1. Support Agent

- As a support agent, I want tickets to be automatically assigned based on my skills and workload, so that I can focus on resolving issues efficiently without spending time on manual ticket selection.

### 2. Support Manager

- As a support manager, I want to monitor ticket distribution and agent performance in real time, so that I can ensure fair workload balance and improve overall team productivity.

### 3. End User (Customer)

- As a customer, I want my support tickets to be assigned to the right agent quickly, so that my issue gets resolved faster and communication is more effective.

#### 4. System Administrator

- As a system administrator, I want to configure and maintain the ticket assignment rules and algorithms, so that the system remains flexible, accurate, and aligned with evolving business needs.

#### 5. Quality Assurance Analyst

- As a QA analyst, I want to analyze ticket resolution trends and feedback, so that I can identify recurring issues and recommend improvements for system reliability.

