PHASE 4 REQUIREMENT ANALYSIS DOCUMENT

Streamlining Ticket Assignment for Efficient Support Operations

TEAM ID	**NM2025TMID08979**
Team Leader	Yogesh S
Team Member	Selvin Joshva A
Team Member	Naveen P
Team Member	Ravin Akash S

Introduction:

Purpose:

The purpose of this document is to define the **functional and non-functional requirements** for developing an intelligent **ticket assignment system** that automates the process of ticket classification and routing in support operations.
This document will serve as a reference for developers, testers, and stakeholders to ensure a shared understanding of system functionality and design constraints.

Project Overview:

In current support operations, tickets are often manually categorized and assigned, leading to inefficiencies, inconsistent response times, and uneven workload distribution.
The proposed system will leverage **Artificial Intelligence (AI)** and **Natural Language Processing (NLP)** to automatically analyze ticket descriptions, classify their type, and assign them to the most suitable agent based on expertise, workload, and availability.

Scope:

The system aims to:

- Automate ticket categorization and assignment.
- Balance workload among agents.
- Provide real-time dashboards for monitoring and analytics.
- Integrate seamlessly with existing support systems like **Jira**, **Zendesk**, or **Freshdesk**.

The system will be implemented as a **web-based application** with **AI-driven backend logic**, **cloud-based deployment**, and a **user-friendly interface**.

System Overview:

System Goals:

- Reduce manual effort in ticket triage.
- Decrease ticket resolution time.
- Improve efficiency and fairness in workload distribution.
- Offer transparency through live dashboards.

Users of the System:

User Type	Description	Access Rights
Admin	Manages system configurations, agents, and ticket categories	Full access
Support Agent	Resolves assigned tickets and updates status	Limited access
End User / Customer	Raises tickets via a form or email	Ticket submission only

Functional Requirements:

ID	Requirement Description	Priority
FR1	The system shall allow users to **submit new tickets** with description, category (optional), and priority.	High
FR2	The system shall **automatically classify** incoming tickets using NLP based on their content.	High
FR3	The system shall **assign tickets** to the most suitable agent based on skillset, workload, and availability.	High
FR4	The system shall maintain a **database** of agents, tickets, and assignments.	High
FR5	The system shall allow agents to **view and manage** assigned tickets.	High
FR6	The system shall provide an **admin dashboard** showing ticket flow, workload distribution, and performance.	Medium
FR7	The system shall send **notifications or alerts** when a new ticket is assigned.	Medium
FR8	The system shall allow admin users to **add, edit, or deactivate agents**.	Medium
FR9	The system shall allow **manual reassignment** of tickets if required.	Medium
FR10	The system shall record **ticket history and metrics** for analytics.	Low
FR11	The system shall provide **search and filter options** for tickets.	Low

Non-Functional Requirements:

Category	Requirement	Description
Performance	Response Time	System should process and assign a ticket within **5 seconds**.
Scalability	Scalability	System should handle up to **10,000 tickets/day** with minimal latency.
Usability	User-Friendly UI	The interface should be intuitive and simple for non-technical users.
Security	Data Protection	User data and ticket content must be encrypted and protected.
Availability	Uptime	System should be available **99% of the time**.
Portability	Web-Based	Accessible via browsers on desktop and mobile.
Maintainability	Modular Code	Codebase should follow modular architecture for easy updates.
Integration	API Ready	System must support RESTful API for integration with external tools.

System Requirements:

Hardware Requirements:

Component	Specification
Processor	Intel i5 or higher
RAM	8 GB minimum
Storage	10 GB free space
Internet	Stable connection (for cloud access)

Software Requirements:

Component	Specification
Operating System	Windows / Linux / macOS
Programming Language	Python, JavaScript
Frameworks	Flask, React
Database	PostgreSQL / MongoDB
AI Libraries	TensorFlow, Scikit-learn, SpaCy
Cloud Platform	Render / AWS / Netlify

Data Requirements:

- Ticket data will include: Ticket ID, description, category, priority, timestamp, and status.
- Agent data will include: Agent ID, name, expertise, workload, and availability.
- System will log every assignment, update, and resolution.
- Dataset for model training (sample IT support tickets) will be preprocessed and anonymized.

System Interfaces:

Interface	Description
User Interface (UI)	Web-based front end for users, agents, and admins.
Backend API	REST API for communication between frontend and backend.
Database Interface	SQL queries for data storage and retrieval.
Integration Interface	API endpoints for connecting to third-party systems (e.g., Jira).

Constraints and Assumptions:

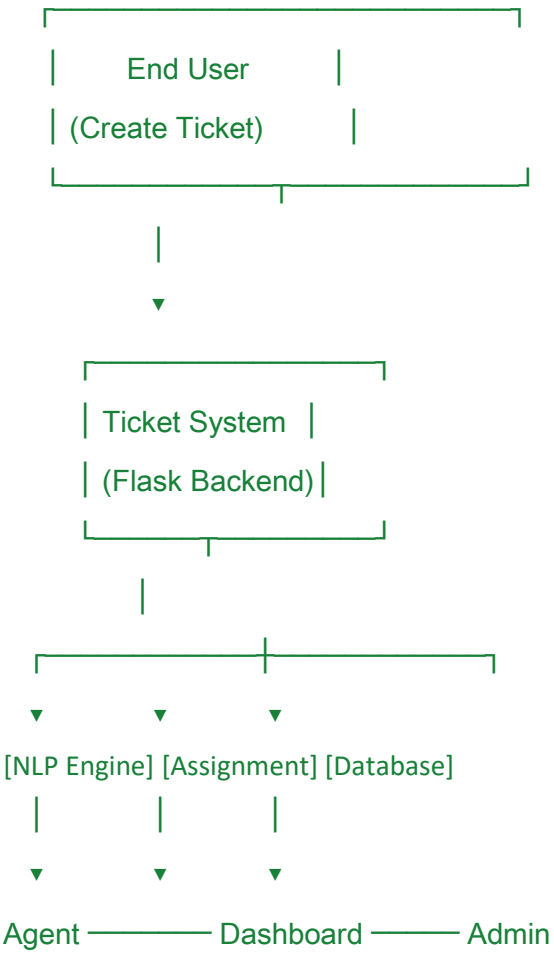
Constraints:

- The system must comply with data privacy standards (e.g., GDPR).
- Limited by available computational power for AI processing.
- Initial deployment restricted to web-based platform.

Assumptions:

- Agents have predefined skillsets stored in the database.
- Input tickets are written in English (first phase).
- Dataset for model training is available and clean.

Use Case Diagram:



Use Case Descriptions:

Use Case ID	Use Case Name	Description	Actors	Outcome
UC1	Submit Ticket	End user submits a new ticket	User	Ticket saved and processed
UC2	Classify Ticket	System analyzes ticket description	System	Category identified
UC3	Assign Ticket	Ticket assigned to best-fit agent	System	Assignment logged
UC4	Manage Tickets	Agents view/update their tickets	Agent	Updated status
UC5	Dashboard Monitoring	Admin views overall system status	Admin	Insights & analytics

Acceptance Criteria:

- Ticket classification accuracy ≥ **85%**.
- Ticket assignment time ≤ **5 seconds**.
- System uptime ≥ **99%**.
- Web dashboard loads within **3 seconds**.
- Secure login and role-based access control.

Future Enhancements:

- Add **multilingual ticket support** using translation APIs.
- Integrate **chatbot interface** for automatic ticket creation.
- Implement **AI learning feedback loop** to improve accuracy.
- Introduce **sentiment analysis** to prioritize critical issues.

Conclusion:

This **Requirement Analysis Document** outlines the necessary functional, non-functional, and technical needs for implementing the **Streamlining Ticket Assignment System**. The requirements defined here will guide the subsequent **system design and implementation** phases to ensure an efficient, intelligent, and scalable support management solution.