

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	01 NOVEMBER 2025
Team ID	NM2025TMID05565
Project Name	Streamlining Ticket Assignment for Efficient Support Operations
Maximum Marks	4 Marks

Technical Architecture:

The “**Streamlining Ticket Assignment for Efficient Support Operations**” project is designed to automate and optimize the process of assigning tickets to the most suitable support agents based on priority, workload, and skill level.

The architecture integrates **frontend interfaces**, **backend logic**, **database components**, and **notification systems**, all hosted on a **cloud-based environment**.

It ensures smooth communication between users, the support system, and administrators while maintaining security, scalability, and efficiency.

Architecture Overview:

Layers included in the architecture:

- User Interface Layer:** Enables users and agents to interact through a dashboard or portal.
- Application Logic Layer:** Implements business rules for automatic ticket assignment and status updates.
- Database Layer:** Stores all ticket, user, and assignment data.
- Notification Layer:** Sends automated alerts and updates to users and agents.
- Integration Layer:** Connects with external systems like CRM or email services.
- Infrastructure Layer:** Hosted on a scalable cloud service ensuring high availability.

Example:

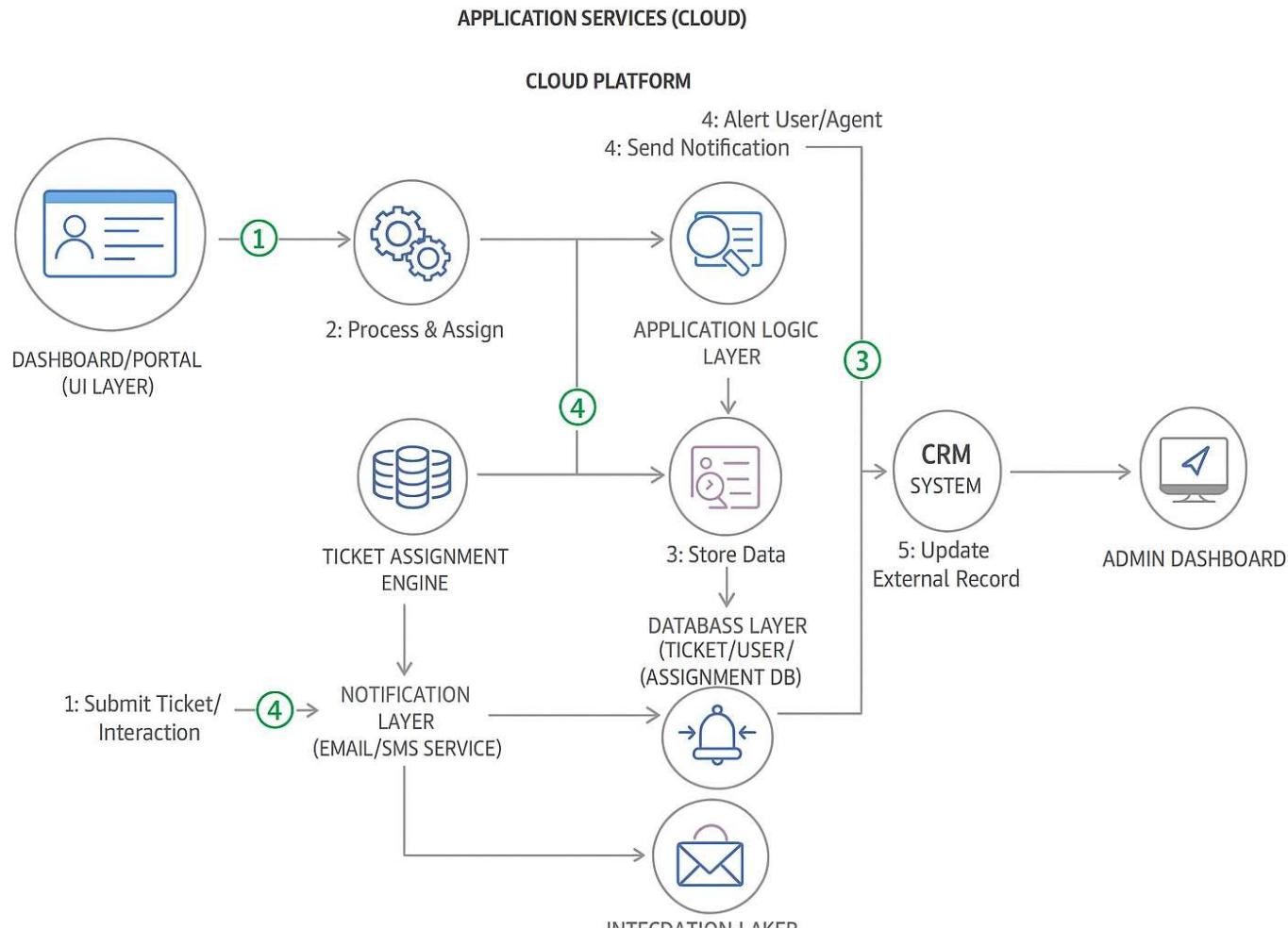


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Users and support agents interact through a web-based dashboard for ticket creation and management.	Angular, HTML5, CSS3, Bootstrap
2.	Application Logic – 1	Handles ticket creation, updates, and assignment rules based on agent workload and ticket priority.	Node.js / Express.js
3.	Application Logic – 2	Implements automatic ticket distribution using defined algorithms.	Workflow Engine / Business Logic Scripts
4.	Application Logic – 3	Sends notifications and updates to users, agents, and admins in real time.	Email & Push Notification Service (Firebase / Nodemailer)
5.	Database	Stores all ticket, user, and performance data securely.	MongoDB / MySQL
6.	Cloud Database	Managed cloud database for high availability and automatic scaling.	MongoDB Atlas / AWS RDS
7.	File Storage	Stores logs, attachments, and related files.	AWS S3 / Google Cloud Storage
8.	External API – 1	Integrates with third-party CRM or email services for better workflow.	REST APIs / Webhooks
9.	External API – 2	(Optional) Integration with Chatbot or AI-based customer query assistant.	OpenAI API / Dialogflow
10.	Machine Learning Model	(Optional) Used for predicting ticket category and priority in future enhancements.	Python (scikit-learn / TensorFlow)
11.	Infrastructure (Server / Cloud)	Application hosted on a cloud-based SaaS platform ensuring uptime and scalability.	AWS / Azure Cloud (SaaS Deployment)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Uses open-source frontend and backend frameworks for flexibility and customization.	Angular, Node.js, Express.js
2.	Security Implementations	Authentication, authorization, and secure data handling for all users.	JWT, HTTPS, Role-Based Access Control
3.	Scalable Architecture	Designed for scaling both horizontally and vertically based on ticket volume.	Cloud Auto Scaling (AWS / Azure)
4.	Availability	Ensures 99.9% uptime using distributed cloud instances.	Load Balancers, Replication
5.	Performance	Fast response times and minimal latency through optimized queries and caching.	Redis Cache / Indexed DB
6.	Maintainability	Modular codebase allowing easy updates and new feature integration.	Microservices Architecture
7.	Reliability	Fault-tolerant system with automatic recovery in case of failures.	Cloud Backup / Redundancy
8.	Interoperability	Can connect seamlessly with email systems, chat tools, and CRM platforms.	REST APIs / Webhooks

