Project Design Phase - Solution Architecture

Date	June 2025
Team ID	LTVIP2025TMID57241
Project Name	Service Desk for Customer Complaint Resolution
Maximum Marks	4 Marks

Solution Architecture Overview:

The **Service Desk** system uses a **MERN stack-based client-server architecture** to provide a seamless complaint management experience. It bridges user interactions with backend logic and real-time data flow, ensuring modularity, scalability, and clarity in communication.

Architecture Layers

1. Frontend (Client Layer)

- **Technology**: React.js with Tailwind CSS
- **Responsibilities**:
  - Provides role-based interfaces for **Users**, **Agents**, and **Admins**
  - Enables complaint registration, tracking, and real-time messaging
  - Makes HTTP requests via **Axios** to backend APIs
  - Supports responsive design with light/dark mode toggling
  - Integrates a clean **chat UI** for user-agent interaction

2. Backend (Application Layer)

- **Technology**: Node.js with Express.js
- **Responsibilities**:
  - Hosts RESTful APIs for login, complaint management, chat, and user roles
  - Manages **JWT-based authentication** and role authorization
  - Handles complaint assignment and routing logic
  - Maintains secure interaction between frontend and database

3. Database (Storage Layer)

- **Technology**: MongoDB Atlas (Cloud-hosted NoSQL)
- **Responsibilities**:
  - Stores user details, complaint records, chat history, roles, and statuses
  - Uses a **document-based schema (Mongoose)** for flexibility and scalability

4. Optional Integrations

- **Socket.io**: Enables real-time messaging between users and agents
- **Email/SMS Gateway** (Future Scope): For sending complaint status notifications

Data Flow Overview

1. **User Authentication**
    - User registers/logs in via the frontend
    - Credentials are verified in the backend and **JWT tokens** are issued
    - User session data is securely stored in browser (cookies/localStorage)
2. **Complaint Submission**
    - User submits a complaint through their dashboard
    - Complaint is stored in MongoDB and shown on the admin panel
3. **Complaint Assignment**
    - Admin views unassigned complaints
    - Assigns complaints to available agents manually (admin-controlled routing)
4. **Live Chat Communication**
    - Real-time chat between user and agent using **Socket.io**
    - Messages are stored with timestamps for transparency
5. **Status Updates and Tracking**
    - Agent updates complaint status (e.g., Open, In Progress, Resolved)
    - Updates are reflected live on the user's dashboard
    - Future integration: Status change notifications via email/SMS

Architecture Diagram:

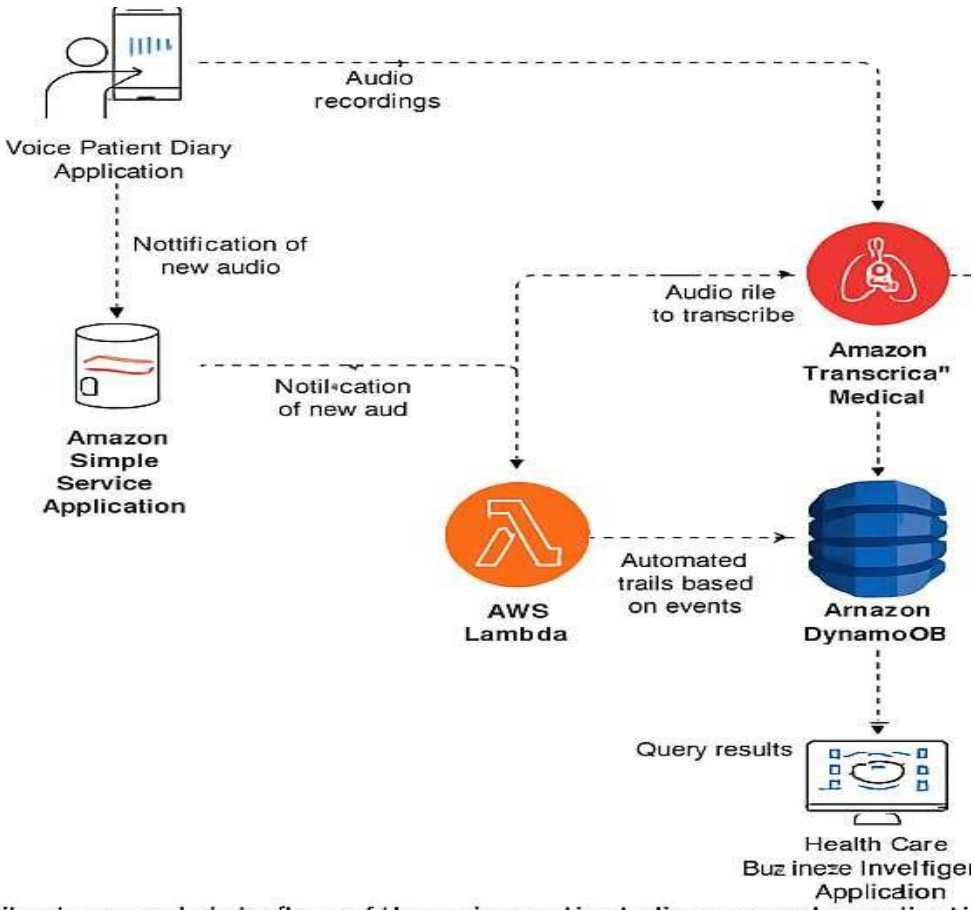

Figure 1: Architecture and data flow of the voice patient-diary sample applicati