**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| **DATE:** | **26-06-2025** |
| **Team ID :** | **LTVIP2025TMID55625** |
| **Project Name :** | **ResolveFlow: Online Complaint Registration and Management System** |

## 1. Functional Requirements (What the System Must Do)

Functional requirements define the specific actions or services the system must perform to meet the business needs. They are directly tied to the user roles and system components.

### 1.1. User Management & Authentication

* **FR1.1.1:** The system shall allow new users (Customers) to register for an account securely.
* **FR1.1.2:** The system shall allow registered users (Customers, Agents, Administrators) to log in securely using their credentials.
* **FR1.1.3:** The system shall provide a mechanism for password recovery/reset.
* **FR1.1.4:** The system shall support role-based access control (Customer, Agent, Administrator) to restrict access to specific functionalities and data.

### 1.2. Complaint Submission

* **FR1.2.1:** The system shall allow authenticated Customers to submit new complaints.
* **FR1.2.2:** The complaint submission form shall capture essential details (e.g., description, type, contact information).
* **FR1.2.3:** The system shall allow users to attach relevant files (e.g., images, documents) to their complaints.

### 1.3. Complaint Management & Tracking

* **FR1.3.1:** The system shall display a personalized dashboard for Customers to view their submitted complaints and their current statuses.
* **FR1.3.2:** The system shall provide real-time updates on complaint status changes to the Customer's dashboard.
* **FR1.3.3:** The system shall allow Agents to view, update the status of, and manage assigned complaints.
* **FR1.3.4:** The system shall allow Administrators to view all complaints, their statuses, and assigned agents.
* **FR1.3.5:** The system shall allow Administrators to assign and reassign complaints to Agents.

### 1.4. Communication & Notifications

* **FR1.4.1:** The system shall provide an integrated in-app messaging/chat feature for direct communication between Customers and their assigned Agents.
* **FR1.4.2:** The system shall store and display the full chat history associated with each complaint.
* **FR1.4.3:** The system shall send automated email and/or SMS notifications to Customers for key complaint events (e.g., submission confirmation, status change, agent assignment, new message).

### 1.5. Data Management

* **FR1.5.1:** The system shall reliably store and retrieve all user profiles, complaint details, chat messages, and attachment metadata in the database.

## 2. Non-Functional Requirements (How Well the System Performs)

Non-functional requirements specify quality attributes of the system, defining how well the system must perform its functions.

### 2.1. Performance

* **NFR2.1.1 (Response Time):** The system shall respond to user requests (e.g., page loads, form submissions, data retrieval) within 2 seconds under normal load.
* **NFR2.1.2 (Real-time Latency):** In-app chat messages and complaint status updates shall have a latency of less than 500 milliseconds.
* **NFR2.1.3 (Throughput):** The system shall support at least 100 concurrent users without significant performance degradation.

### 2.2. Scalability

* **NFR2.2.1 (Horizontal Scaling - Backend):** The backend shall be designed to scale horizontally by adding additional server instances to handle increased user load.
* **NFR2.2.2 (Database Scaling):** The database (MongoDB) shall support sharding to handle growing data volumes and user concurrency effectively.
* **NFR2.2.3 (Real-time Scaling):** The real-time communication infrastructure (Socket.io) shall scale to support a large number of concurrent chat sessions.

### 2.3. Security

* **NFR2.3.1 (Authentication):** User passwords shall be securely hashed and stored. The system shall support secure user authentication with mechanisms to prevent brute-force attacks.
* **NFR2.3.2 (Authorization):** Access to data and functionalities shall be strictly controlled based on user roles (RBAC).
* **NFR2.3.3 (Data Encryption - In Transit):** All data transmitted between the frontend, backend, and external services shall be encrypted using HTTPS/SSL/TLS.
* **NFR2.3.4 (Data Encryption - At Rest):** Sensitive user and complaint data stored in the database shall be encrypted where applicable.
* **NFR2.3.5 (Input Validation):** The backend shall rigorously validate all user inputs to prevent common web vulnerabilities (e.g., XSS, injection attacks).
* **NFR2.3.6 (Confidentiality):** The system shall ensure the confidentiality of user data and complaint details, accessible only by authorized individuals.
* **NFR2.3.7 (Auditability):** The system shall maintain logs of critical actions (e.g., complaint status changes, admin assignments) for auditing purposes.

### 2.4. Reliability & Availability

* **NFR2.4.1 (Uptime):** The system shall aim for an uptime of 99.9% annually.
* **NFR2.4.2 (Data Persistence):** All submitted data shall be reliably stored and retrievable, minimizing data loss.

### 2.5. Usability (User Experience)

* **NFR2.5.1 (Intuitive Interface):** The user interface shall be intuitive and easy to navigate for all user roles (Customer, Agent, Administrator).
* **NFR2.5.2 (Responsiveness):** The system's interface shall be fully responsive and optimized for display across various devices (desktop, tablet, mobile).
* **NFR2.5.3 (Accessibility):** The system shall adhere to common web accessibility standards to ensure usability for a broad range of users.

### 2.6. Maintainability

* **NFR2.6.1 (Code Quality):** The codebase shall follow established coding standards and best practices, ensuring readability and ease of maintenance.
* **NFR2.6.2 (Modularity):** The system shall be composed of modular components to allow for independent development, testing, and deployment.
* **NFR2.6.3 (Observability):** The system shall provide adequate logging and monitoring capabilities to identify and diagnose issues.

### 2.7. Deployability

* **NFR2.7.1 (Automated Deployment):** The system shall support automated deployments via Git integration (e.g., on Render).
* **NFR2.7.2 (Environment Management):** The system shall support environment-specific configurations (e.g., development, staging, production).