**1. Presentation Layer (React.js)**

The frontend will handle the user interface, user input, and communication with the backend.

**Components:**

* **Login Component:** For user authentication.
* **Chatbot Component:** For interacting with the chatbot.
* **Ticket List Component:** For displaying active and resolved tickets.
* **Agent Dashboard Component:** For managing tickets, viewing performance metrics, and handling user queries.

**Responsibilities:**

* Handle user interactions.
* Display data fetched from the backend.
* Send user inputs to the backend via API calls.

**2. Application Layer (Flask Services)**

The backend services will manage the application logic, process user requests, and communicate with the domain layer.

**Components:**

* **API Controllers:** Handle HTTP requests and route them to appropriate services.
* **Services:** Contain the application-specific business logic.

**Responsibilities:**

* Authenticate users.
* Route requests to the appropriate business logic.
* Return responses to the frontend.

**3. Domain Layer (Business Logic)**

This layer contains the core business logic and rules. It is independent of frameworks and external resources.

**Components:**

* **Entities:** Represent the core business objects (e.g., User, Ticket).
* **Repositories:** Define interfaces for data access, which are implemented in the infrastructure layer.
* **Use Cases:** Represent the business actions and operations.

**Responsibilities:**

* Define business rules and policies.
* Ensure business logic is isolated from other concerns.

**4. Infrastructure Layer (Data Storage and External Services)**

This layer interacts with external systems and frameworks, such as databases and third-party services.

**Components:**

* **Database Implementations:** Concrete implementations of repository interfaces.
* **External Services:** Integration with external APIs and services.

**Responsibilities:**

* Handle data persistence.
* Integrate with external systems.

**Flow of Data**

1. **User Interaction:**
   * User interacts with the frontend (React.js).
   * Frontend sends an API request to the backend (Flask).
2. **Request Handling:**
   * Flask controllers receive the request and pass it to the appropriate service in the application layer.
3. **Business Logic Execution:**
   * The service in the application layer interacts with the domain layer to execute business logic.
   * Use cases in the domain layer manage the core operations and may call repository interfaces to fetch or persist data.
4. **Data Persistence and Retrieval:**
   * Repositories in the domain layer use implementations from the infrastructure layer to interact with databases.
   * NoSQL database for chatbot interactions.
   * SQL database for user-agent interactions.
5. **Response to Frontend:**
   * The application layer service compiles the response and sends it back to the frontend via the Flask controller.
6. **UI Update:**
   * The frontend updates the UI based on the response from the backend.