**1. 🔄 Auto-Assignment Logic for Inquiries (5.1.1)**

| **Inquiry Type** | **Auto-Assigned To** | **Workflow** |
| --- | --- | --- |
| Claims (assigned employer) | Claims Checklist User | **Agent → Attendee (attend and close)** |
| Compliance (assigned employer) | Assigned Compliance Officer | **Agent → Attendee (attend and close)** |
| Claims (unassigned employer) | Claims Focal Person | **Agent → Focal → Attendee (reverse or attend and close)** |
| Compliance (unassigned employer) | Compliance Focal Person | **Agent → Focal → Attendee (reverse or attend and close)** |
| Unregistered Employee/Employer regarding claims/compliance | Claims Focal Person | **Agent → Focal → Attendee (reverse or attend and close)** |
| Outside Claims/Compliance | Focal Person of relevant section/unit | **Agent → Focal → Attendee (reverse or attend and close)** |

**Implementation Idea:**  
Use **rules-based logic** with if-else or a switch-case to determine auto-assignment based on subject and employer status.

**2. 🚨 Inquiry Escalation (5.1.2)**

* If an inquiry remains **unattended within SLA**, it’s auto-escalated to:
  + **Head of Unit** or **Manager**
* Head/Manager can re-assign it to an Attendee, who then handles it **per normal flow**.

**Implementation:**  
Use a **scheduled job** (e.g., Laravel scheduler or cron) to check for **aging tickets** and escalate automatically.

**3. 🛠️ Complaint Handling (5.2.1)**

| **Complaint Type** | **Workflow** |
| --- | --- |
| **Minor – Unit** | Agent → Coordinator (rate) → Head of Unit (assign) → Attendee (attend & recommend) → Head of Unit (close) |
| **Minor – Directorate** | Agent → Coordinator (rate) → Director (assign) → Manager (assign) → Attendee (attend & recommend) → Manager (close) |
| **Major – Unit** | Agent → Coordinator (rate) → Head of Unit (assign & recommend) → Attendee (attend & recommend) → Head of Unit → DG (approve & close) |
| **Major – Directorate** | Agent → Coordinator (rate) → Director (assign) → Manager (assign & recommend) → Attendee → Manager → Director → DG (approve & close) |
| **Complaint to Inquiry** | Coordinator can **change complaint to inquiry**, triggering auto-assignment logic (same as inquiries in 5.1.1) |

**Implementation:**  
Use **workflow states** and **role-based permissions** to allow each user to take only valid actions at their step.

**4. 💡 Suggestion & Compliment Handling (5.3.1)**

| **Flow Type** | **Workflow** |
| --- | --- |
| Unit | Agent → Coordinator → Head of Unit (note and close) |
| Directorate | Agent → Coordinator → Director → Manager (note and close) |

**Implementation:**  
Treat suggestions/compliments as **tickets with special category**, follow simplified workflows.

**5. 📬 Notification Triggers**

| **Event** | **Recipient(s)** | **Method** |
| --- | --- | --- |
| Inquiry assigned | Attendee | Email |
| Inquiry closed | Agent (system), Manager (email), Client | In-system + Email + SMS |
| Complaint assigned | Assigned user | Email |
| Complaint closed | Agent, Coordinator, Director, Manager | System/Email + SMS |
| Complaint escalated (per SLA) | Attendee | Email |
| Compliment/Suggestion registered | DG | Notification |
| Compliment/Suggestion forwarded | Attendee | Email |

**Integration Required**:

* **GMS** → for email
* **mGOV** → for SMS
* **Contact Center**, **MAC**, **e-Office**, **Self-Service Portal** → for inquiry intake and stakeholder search

**6. 🔍 Search Functionalities**

| **User Role** | **Searchable By** |
| --- | --- |
| Agent, Coordinator | Ticket ID, Name, Phone Number, NIDA Number |
| Director, Manager | Ticket ID, Name, Phone Number |

**7. 📊 Report Management**

**i. General Request Report Columns (5.4.i)**

| **Column** | **Column** |
| --- | --- |
| Ticket ID | Complaint Description |
| Name of Employer | Category of Complaint |
| Phone Number | Responsible Directorate/Unit |
| Region | Date of Review and Resolution |
| District | Resolution Details |
| Subject | Date of Feedback |
| Category | Agent Name |
| Date of Registration | Status (Open/Closed) |
| Date of Resolution | Pending Request (Section & Employee) |
| Channel | Request Registered Date |
| Location (Region and District) | Attended By |
| Aging |  |

**ii. Complaint Management Report (5.4.ii)**

* Follows similar structure to General Report but **filtered for complaint tickets only**.

**Implementation:**  
Build **dynamic queries** with filters for date range, category, status, etc., and **export as Excel/PDF**.

**✅ Next Steps:**

Would you like help on any of the following?

* Database schema for ticket types, workflow status, and escalation
* Laravel or Spring Boot logic for auto-assignment and SLA escalation
* Notification service integration (mGOV, GMS)
* Report generation (backend logic or Excel templates)
* Frontend form/UI for workflow handling

Let me know which part you'd like to dive into!

To handle the detailed workflows for **Inquiries, Complaints, Suggestions/Compliments**, along with **Assignment, Escalation, Notifications, and Reporting**, you need several relational tables.

Here’s a proposed **Database Table Structure** to implement all these scenarios efficiently.

**🗂️ 1. tickets table (Main ticket data)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key (Ticket ID) |
| requester\_name | VARCHAR | Name of Employer / Employee |
| phone\_number | VARCHAR | Contact number |
| nida\_number | VARCHAR | National ID number (optional) |
| region | VARCHAR | Region |
| district | VARCHAR | District |
| subject | VARCHAR | Subject of ticket |
| category | ENUM | 'Inquiry', 'Complaint', 'Suggestion', 'Compliment' |
| sub\_category | VARCHAR | Claims, Compliance, etc. |
| channel | VARCHAR | Contact method (Phone, Portal, etc.) |
| description | TEXT | Complaint / Inquiry description |
| complaint\_type | ENUM | 'Minor', 'Major', NULL for inquiries |
| responsible\_unit\_id | FK (units) | Responsible Directorate or Unit |
| assigned\_to\_id | FK (users) | Currently assigned user |
| assigned\_to\_role | ENUM | 'Coordinator', 'Attendee', 'Focal', etc. |
| status | ENUM | 'Open', 'In Progress', 'Returned', 'Closed' |
| request\_registered\_date | TIMESTAMP | Date of registration |
| date\_of\_resolution | TIMESTAMP | When ticket was resolved |
| date\_of\_feedback | TIMESTAMP | Feedback date (if any) |
| date\_of\_review\_resolution | TIMESTAMP | Review date |
| resolution\_details | TEXT | Final resolution summary |
| attended\_by\_id | FK (users) | Who attended the ticket |
| aging\_days | INT | Calculated as DATEDIFF(NOW(), request\_registered\_date) |
| created\_at | TIMESTAMP | Record creation |
| updated\_at | TIMESTAMP | Record update |

**🗂️ 2. ticket\_assignments table (Audit of assignments & returns)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| ticket\_id | FK (tickets) | Associated ticket |
| assigned\_by\_id | FK (users) | User who assigned or returned |
| assigned\_to\_id | FK (users) | User to whom ticket was assigned |
| assigned\_to\_role | ENUM | 'Coordinator', 'Attendee', 'Focal', etc. |
| action | ENUM | 'Assigned', 'Returned', 'Reassigned' |
| reason | TEXT | Reason for return or reassignment (nullable) |
| created\_at | TIMESTAMP | Timestamp of action |

**🗂️ 3. users table (Assignee and requester details)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| name | VARCHAR | User's full name |
| email | VARCHAR | Email address |
| phone\_number | VARCHAR | Phone number |
| role | ENUM | 'Agent', 'Coordinator', 'Attendee', 'Manager', 'DG', etc. |
| unit\_id | FK (units) | Affiliated unit/directorate |
| active\_status | BOOLEAN | Is user active? |
| created\_at | TIMESTAMP | Created |

**🗂️ 4. units table (Directorates or sections)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| name | VARCHAR | Unit or Directorate name |
| section\_type | ENUM | 'Claims', 'Compliance', etc. |
| created\_at | TIMESTAMP | Created |

**🗂️ 5. ticket\_followups table (Agent follow-ups)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| ticket\_id | FK (tickets) | Related ticket |
| added\_by\_id | FK (users) | User who added follow-up |
| note | TEXT | Follow-up content |
| created\_at | TIMESTAMP | Timestamp |

**🗂️ 6. action\_items table (Attendee updates/progress)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| ticket\_id | FK (tickets) | Associated ticket |
| updated\_by\_id | FK (users) | Attendee who added progress |
| update\_note | TEXT | Progress detail |
| created\_at | TIMESTAMP | Timestamp |

**🗂️ 7. notifications table (Optional – for in-system notifications)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| ticket\_id | FK (tickets) | Associated ticket |
| recipient\_id | FK (users) | Who receives the notification |
| message | TEXT | Notification message |
| channel | ENUM | 'Email', 'SMS', 'In-System' |
| status | ENUM | 'Pending', 'Sent', 'Failed' |
| created\_at | TIMESTAMP | Timestamp |

**🗂️ 8. integrations table (Optional – for tracking external system usage)**

| **Column Name** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID / INT | Primary Key |
| ticket\_id | FK (tickets) | Associated ticket |
| system\_name | VARCHAR | e.g., MAC, e-Office, mGOV |
| action\_taken | TEXT | e.g., 'Name search', 'SMS Sent' |
| timestamp | TIMESTAMP | Timestamp |

**📝 Summary Mapping to Requirements**

| **Requirement Section** | **Tables Involved** |
| --- | --- |
| Inquiry Assignment | tickets, users, ticket\_assignments |
| Complaint Workflow | tickets, ticket\_assignments, users |
| Escalation Tracking | tickets, ticket\_assignments |
| Follow-up & Action | ticket\_followups, action\_items |
| Notifications | notifications, external integrations |
| Reporting | tickets, joins with users, units |
| Search | Query tickets + filters on phone, ID, NIDA |
| Integration | integrations |

**🧱 Implementation Tip**

You can use **Eloquent relationships** in Laravel:

* Ticket::hasMany(TicketAssignment)
* User::hasMany(TicketAssignment, 'assigned\_by\_id')
* Ticket::hasMany(ActionItem)
* Ticket::hasMany(FollowUp)