# **(2) Stakeholders & Users Identification**

### **Stakeholders**

* Students (end users – complaint lodgers)
* Staff / Faculty
* Departmental handlers
* Dean / Registrar / Authorities
* System Admin
* Software Developers / Testers
* Ventors

**Elicitation Techniques:**

**1. Surveys / Questionnaires (Students):**

Google forms were distributed to students to understand their current methods of raising complaints, satisfaction with the process, and expectations for new features such as attachments, categorization, urgency marking, tracking, notifications, and escalation.

From this, key artefacts identified were usability requirements, a complaint submission module, and features like status tracking, categorization, escalation, and notifications.

**2. Focus Groups (Student Representatives / Committee):**

Discussions were conducted with student representatives to gather collective concerns about the existing complaint system.

Through these discussions desired improvements were identified, highlighting the need for features such as complaint status tracking, a simple and user-friendly interface.

**3. Interviews (Staff & Faculty&management)**

Interviews were conducted with staff, faculty, and department heads to gain a understanding of the complaint handling system. Staff and faculty interviews focused on capturing workflows and the need for a structured routing process, while discussions with management addressed reporting requirements, performance metrics, and escalation procedures.

These interactions help us to identify key artefacts, including the functional requirement for automatically routing complaints to the appropriate departments, categorization of complaints by type, and the escalation hierarchy for unresolved complaints.

**4. Observation**

The current complaint handling process was observed, including manual registers and email communication. This observation revealed inefficiencies and delays in managing complaints.

Key artefacts identified include the need for an analytics and reporting dashboard and the requirement to replace manual tracking with an automated system.

**5.** **Document Analysis**

Existing IT policy documents, complaint records, escalation rules, and data handling guidelines were reviewed to understand the system from a regulatory and administrative perspective.

Key artefacts identified include security requirements such as authentication and encryption, as well as the need for role-based access control.

**6. Prototyping + Data Review (IT/ML Developers)**

Prototyping and data review were conducted with IT and ML developers to refine system design and functionality. UI mockups were developed for the complaint submission form, admin panel, and dashboard, while historical complaint data was analyzed to assess the feasibility of ML-based categorization.

Key artefacts identified include the requirement for ML-based automatic categorization and routing, as well as validation of prototype features such as notifications and analytics.

**(3) Functional Requirements (FRs)**

* **Complaint Log in & Submission** : Surveys & Interviews with students/staff.
* **User Registration & Authentication** : Document analysis of IT security rules.
* **Role-based Access Control** : Interviews,observation and document review.
* **Auto-Categorization & Routing (ML)** : Interviews and data review with IT team.
* **Track Complaint Status** : Focus groups with students.
* **Analytics & Reporting** : Interviews with management + observation.
* **Notification (Email Alerts)** : Prototyping with IT team (SMTP integration).
* **Admin Panel** : Interaction with system administrators.

**Non-Functional Requirements (NFRs)**

* Usability : Surveys & focus groups with students (simple, user-friendly).
* Security : Document analysis of IT policies (authentication, encryption, privacy).
* Scalability : Interaction with the IT team (must handle peak loads).
* Reliability : Interviews with staff (ensure complaints are not lost).
* Performance : Interviews with staff (acceptable response time).
* Maintainability : Prototyping with developers (easy updates & fixes).