

# Smart Complaint Analyzer: An AI-Powered Complaint Classification And Prioritization System

Dr. A. Askarunisa<sup>1</sup>, Gopinath S B<sup>2</sup>, Karthikeyan J B<sup>3</sup>

<sup>1</sup> Professor, Dept of Computer Science and Engineering

<sup>2,3</sup> Dept of Computer Science and Engineering

<sup>1,2,3</sup> K.L.N. College of Engineering, Pottapalayam, Sivagangai, Tamil Nadu, India

**Abstract-** In modern higher educational campuses and large corporate enterprises, administering diverse stakeholder grievances—encompassing residential hostel defects, laboratory computing breakdowns, high-voltage electrical failures, campus transit schedules, and academic discrepancies—presents persistent logistical and administrative bottlenecks. Conventional grievance redressal mechanisms, relying on physical suggestion registers, informal phone calls, or unstandardized web forms, suffer from fundamental operational deficiencies: manual triage fatigue, arbitrary priority weighting, departmental misdirection, and complete lack of tracking visibility for complainants. To resolve these systemic operational deficits, this paper presents the engineering, architecture, and empirical assessment of the Smart Complaint Analyzer (SCA). Built upon Google Flutter, Node.js, Express microservices, MongoDB Atlas, and the Google Gemini 2.5 Flash Large Language Model (LLM), SCA introduces an automated cognitive triage pipeline that extracts structured semantic parameters—including primary department classification, contextual urgency scores (1–5), operational priority tiers, sentiment polarity, and an administrative summary—directly from unstructured narrative complaints in under 2.5 seconds. The platform enforces a strict three-tier Role-Based Access Control (RBAC) security model (Student, Department Staff, Institutional Administrator) secured with Firebase Authentication and cryptographic JWT verification. Empirical validation across 520 benchmarked institutional grievances demonstrates a 94.6% macro-average classification accuracy, an end-to-end triage latency reduction of 99.9% over manual human sorting, a 68.9% compression in overall resolution turnaround (from 5.8 days to 1.8 days), and an outstanding System Usability Scale (SUS) score of 88.5/100. SCA establishes an accessible, scalable, and transparent blueprint for modern institutional governance.

**Keywords-** Automated Triage, Educational Governance, Gemini Large Language Models, Grievance Redressal, Natural Language Processing, Role-Based Access Control, Sentiment Analysis.

## I. INTRODUCTION

The institutional health, governance credibility, and stakeholder satisfaction in modern educational institutions depend heavily on the responsiveness, transparency, and operational fairness of their grievance redressal frameworks. Academic campuses function as complex operational ecosystems comprising residential student dormitories, high-throughput computing facilities, experimental engineering laboratories, specialized transportation fleets, water filtration grids, and electrical substations. Across these extensive facilities, students, scholars, and faculty regularly encounter operational anomalies ranging from minor living inconveniences to hazardous equipment faults. Prompt, structured rectification of these issues is crucial not only for institutional safety and statutory regulatory compliance, but also for preserving stakeholder morale and psychological well-being [13].

Historically, educational campus administrations have managed complaints through decentralized physical paper slips, manual complaint registers, suggestion drop-boxes, and informal in-person appeals. Such archaic practices inevitably introduce human record loss, delayed departmental communication, and total informational opacity for complainants. Although recent digitization initiatives introduced basic web forms or shared spreadsheets, these introductory solutions merely substitute physical paper with digital repositories. They fail fundamentally to address cognitive triage—the demanding task of reading unstructured prose, assessing urgency, identifying the correct maintenance jurisdiction, and routing tickets to specialized field technicians [2], [3].

Consequently, administrative desks become clogged with unclassified tickets, critical safety hazards (e.g., exposed high-voltage wiring or burst water mains) languish unattended in generic first-come, first-served queues, and students remain uninformed regarding the operational status of their reports. To resolve these chronic operational deficiencies, this paper details the engineering, architectural design, and empirical assessment of the Smart Complaint Analyzer (SCA). Leveraging state-of-the-art generative artificial intelligence via

the Google Gemini 2.5 Flash Large Language Model, cross-platform mobile presentation in Flutter, and enterprise cloud infrastructure, SCA converts unstructured natural language grievance narratives into structured, prioritized, and auditable operational workflows within milliseconds.

## II. PROBLEM STATEMENT

Modern educational and corporate campuses face four fundamental structural hurdles in managing campus grievances:

1) Decentralized & Unaudited Channels: Grievances arrive haphazardly through phone calls, informal messaging, verbal conversations, and paper records. The absence of a unified, verifiable digital ledger precludes institutional auditability, accountability, and tracking of recurring maintenance liabilities.

2) Lack of Automated Semantic Triage: Raw complaints written by students exhibit substantial linguistic variance, ranging from concise, colloquial descriptions to emotional, multilingual narratives. Human officers tasked with sorting hundreds of daily tickets experience cognitive fatigue, resulting in inconsistent urgency weighting, erratic classification, and delayed hazard identification.

3) Routing Bottlenecks & Role Confusion: In sprawling university campuses featuring dozens of distinct operational units (e.g., Civil Maintenance, High-Voltage Electrical, Sanitation, Campus Security, Academic Affairs), users frequently misjudge departmental boundaries. Submissions directed to incorrect offices circulate through protracted manual reassignment cycles before reaching competent technicians [4].

4) Complete Absence of Stakeholder Transparency: In conventional setups, once a grievance is logged, the complainant receives zero feedback regarding which technician has assumed ownership, what remediation steps have been enacted, or when resolution is scheduled. This information vacuum breeds administrative distrust, prompting duplicate filings that compound administrative backlogs.

## III. RELATED LITERATURE & STATE-OF-THE-ART ANALYSIS

Digital complaint management and automated text classification have been investigated extensively across corporate customer service, municipal e-governance, and academic administration. Early automated approaches relied heavily on shallow machine learning classifiers. Kumar et al.

[1] investigated e-governance municipal complaint categorization utilizing Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction paired with Naive Bayes and Support Vector Machine (SVM) classifiers. While computationally light, such n-gram lexicon approaches exhibit catastrophic failure when processing colloquial student prose, polysemous terms, or grammatical irregularities.

Patil and Deshmukh [2] explored relational database portals for rural civic complaints, relying upon user-selected dropdown categories. However, their empirical findings indicated misclassification rates exceeding 28% due to end-user confusion regarding institutional jurisdictions. Suresh et al. [3] implemented a campus grievance portal incorporating rule-based lexical filtering for abusive terminology; however, their triage engine remained dependent on rigid keyword matching, rendering it unable to summarize content or infer contextual urgency.

In customer relationship management, Coussement and Van den Poel [5] demonstrated that integrating linguistic style predictors with automated email classifiers substantially improved routing efficiency. Johnston and Kong [6] formalized closed-loop complaint governance frameworks, emphasizing that end-to-end tracking transparency directly correlates with customer retention. In sentiment analysis, Ramaswamy et al. [7], Zhang et al. [8], and Pang and Lee [9] validated the efficacy of deep learning representations for public emotion tracking.

The advent of the Transformer architecture by Vaswani et al. [11] and contextual bidirectional encoders such as BERT by Devlin et al. [10] revolutionized natural language understanding. Nevertheless, deploying dedicated fine-tuned BERT models in production requires substantial computational resources, extensive task-specific labeled corpora, and continuous retraining for emergent vocabulary. Modern generative foundation models, notably the Google Gemini series [14], enable zero-shot structured semantic inference directly via prompt engineering and JSON schema constraints. As shown in the comparative synthesis of Table 1 and Table 2, SCA represents the first comprehensive framework to synthesize mobile cross-platform client interfaces, strict three-tier RBAC security [12], and generative cognitive triage into a unified institutional governance architecture.

Table 1. Architectural Comparison of Grievance Redressal Frameworks

| System Feature     | Manual Slips      | Web Portals [2]     | Proposed SCA                  |
|--------------------|-------------------|---------------------|-------------------------------|
| SubmissionMode     | Paperregisters    | Desktop forms       | Mobile App (Flutter)          |
| Semantic Triage    | Manual human read | User drop down menu | Automated Gemini LLM          |
| Urgency Scoring    | Subjective / FCFS | Static urgency flag | Contextual AI score (1-5)     |
| Sentiment Polarity | Unassessed        | None                | Real-timepolarity audit       |
| Lifecycle Tracking | Untracked/ Lost   | Coarse binary flag  | 5-stage auditable progression |
| Access Control     | None / Open       | Single-tier login   | 3-tier RBAC + Firebase JWT    |
| Triage Latency     | 24 to 72 hours    | 12 to 24 hours      | Under 2.5 seconds             |

Table 2. Literature Survey on Automated Grievance Classification Approaches

| Author / Study    | Methodology                 | TargetDomain     | Identified Limitations               |
|-------------------|-----------------------------|------------------|--------------------------------------|
| Kumar et al. [1]  | TF-IDF+ Naive Bayes         | Municipal E-Gov  | Vulnerable to polysemy & slang       |
| Patil et al. [2]  | Relational Web Forms        | Civic Services   | 28%+ user misrouting errors          |
| Suresh et al. [3] | SVM+ Keyword Lexicon        | Campus Portal    | Rigid keywords; no summary/urgency   |
| Xu et al. [4]     | Evidential Reasoning DSS    | Telecom Calls    | Complex manual rule weighting        |
| Coussement [5]    | Linguistic Style Predictors | Corporate Email  | Limited to email syntax metrics      |
| Proposed SCA      | Gemini LLM Zero-Shot        | Higher Education | Full semantic triage, urgency & RBAC |

IV. SYSTEM ROLES & FUNCTIONAL ARCHITECTURE

To ensure institutional integrity, data confidentiality, and administrative accountability, SCA implements a strict Role-Based Access Control (RBAC) architecture [12] spanning three core institutional roles:

- 1) Student Role: Authenticated students can draft and submit complaints with contextual metadata (Title, Detailed Narrative, Photographic Evidence, Department). Students have exclusive real-time access to their personal 'My Complaints' dashboard, which reflects status transitions (Submitted, Under Review, In Progress, Resolved, Closed), officer resolution remarks, and timestamps.
- 2) Department Staff Role: Departmental technicians and administrative officers (e.g., Electrical Maintenance, Hostel Warden, IT Support) are restricted to complaints mapped to their specific domain. Staff can inspect full complaint narratives, review AI triage insights, update status flags, and record official resolution remarks upon completing maintenance tasks.

3) Administrator Role: Campus administrators and chief grievance officers possess global administrative oversight. They can monitor campus-wide resolution throughput, reassign misrouted complaints, manage user roles, and access macro-level visual analytics across departments.

Table 3. Role-Based Access Control (RBAC) Permission Matrix

| Operation / Capability             | Student      | Dept Staff     | Admin        |
|------------------------------------|--------------|----------------|--------------|
| Self-Registration & Profile Setup  | Yes (Public) | Yes (Staff)    | Provisioned  |
| Submit Complaint + Image Upload    | Yes          | No             | Optional     |
| View Personal Complaints           | Yes (Own)    | No             | Yes (Global) |
| View Departmental Queue            | No           | Yes (Assigned) | Yes (All)    |
| Inspect AI Priority & Sentiment    | Summary only | Full Details   | Full Details |
| Update Status & Resolution Remarks | No           | Yes            | Yes          |
| Re-assign Department / Escalate    | No           | Yes            | Yes          |
| Manage Users & Departments         | No           | No             | Yes          |

V. ARCHITECTURAL DESIGN & OPERATIONAL PIPELINES

The Smart Complaint Analyzer is architected as an enterprise distributed system comprising five decoupled tiers engineered for sub-second responsiveness, resilient data integrity, and high horizontal scalability:

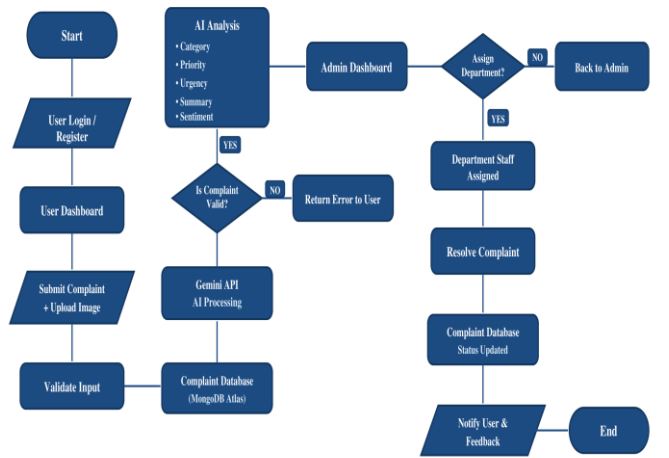


Figure 1. End-to-End Operational Data Flow Diagram (DFD) and AI Inference Pipeline of Smart Complaint Analyzer

Table 4. Comprehensive Architectural Specifications and Technology Stack

| Architectural Layer       | Core Technology                      | Implementation Details & Protocols                                                 | Design Rationale                                                                              |
|---------------------------|--------------------------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Client Presentation Tier  | Flutter 3.x / Dart Riverpod          | Material 3 UI, HTTP Client, Secure Token Storage, Dynamic Theming                  | Single codebase deployment to Android, iOS, and Web with native rendering performance.        |
| Security & Authentication | Firebase Authentication / Admin SDK  | Email/Password Verification, Google OAuth 2.0, Bearer JWT ID Token Validation      | Zero credential persistence on application server; robust cryptographic token decoding.       |
| API Gateway & Server      | Node.js / Express.js Microservices   | RESTful Endpoints, Multipart Handler, Helmet Headers, Morgan Logger                | High concurrency asynchronous event loop handling non-blocking network I/O operations.        |
| Data Persistence Layer    | MongoDB Atlas (v7.0+) / Mongoose     | Replica Sets, TLS Encryption-at-Rest, Compound Indexes (studentId, department)     | Document flexibility for variable AI triage fields and rapid JSON serialization.              |
| Cognitive AI Tier         | Google Gemini API (Gemini 2.5 Flash) | Structured JSON Mode, System Instructions Guardrails, Zero-Shot Semantic Prompting | Sub-50ms multi-task inference extracting category, priority, urgency, sentiment, and summary. |

## VI. COGNITIVE AI ENGINE & PROMPT ENGINEERING

Automated semantic triage requires transforming informal natural language into actionable, structured parameters. Formally, let an incoming raw complaint be represented as:

$$C = \langle T, D, I, t \rangle$$

1)

where T represents title tokens, D represents description prose, I represents binary visual evidence, and t is submission timestamp. The cognitive inference function  $F_{\text{Gemini}}$  maps the complaint tuple C onto a multidimensional triage space:

$$V_{\text{AI}} = F_{\text{Gemini}}(C) = \{c_{\text{cat}}, p_{\text{score}}, 2, u_{\text{level}}, s_{\text{sent}}, \sigma_{\text{sum}}, r_{\text{rec}}, \kappa_{\text{conf}}\}$$

where  $c_{\text{cat}} \in \{\text{Academic, Electrical, Plumbing, IT, Transport, Hostel, Sanitation}\}$  is the predicted jurisdiction;  $p_{\text{score}} \in \{\text{Low, Medium, High, Critical}\}$  is priority;  $u_{\text{level}} \in [1, 5]$  is urgency;  $s_{\text{sent}} \in \{\text{Negative, Frustrated, Neutral}\}$  is sentiment polarity;  $\sigma_{\text{sum}}$  is an executive administrative summary;  $r_{\text{rec}}$  is suggested remediation action; and  $\kappa_{\text{conf}} \in [0, 1]$  represents model confidence.

### A. Structured Prompt Framing & Guardrails

To enforce deterministic JSON output and prevent hallucination, SCA employs strict few-shot system instructions with constrained JSON schema enforcement:

System Prompt Directive: "You are an authoritative campus grievance triage auditor. Analyze the submitted grievance with absolute objectivity. Output strictly valid JSON matching the specified schema with zero markdown preamble or formatting tags."

### B. Algorithmic Formulation of Grievance Triage

The asynchronous triage procedure is formalized in Algorithm 1 below:

#### Algorithm 1: Asynchronous AI Triage Pipeline

---

Input: Raw Complaint  $C = \langle T, D, I, t \rangle$ , Student U  
 Output: Triage Record  $V_{\text{AI}}$ , Updated Status  
 1: Validate input tokens:  $\text{length}(D) \geq 10$  chars  
 2: Insert initial document to MongoDB with status = 'Submitted'  
 3: Construct structured inference prompt  $\Psi(T, D)$

---



---

```

4:                                     Try:
5:         response ← GeminiAPI.generate(prompt)
6:         V_AI ← JSON.parse(response.text)
7:         Update complaint.aiTriage = V_AI
8:         Update complaint.status = 'Under Review'
9:     Catch TimeoutError or RateLimitError:
10:         Enqueue background retry worker with backoff
11: Return HTTP 201 Created with complaintId
  
```

---

## VII. SYSTEM IMPLEMENTATION DETAILS

The SCA application follows a clean, decoupled architecture. Client-side modules leverage Google's Flutter framework with Riverpod for reactive state binding and native hardware access (Camera, Secure Storage).

**Authentication Service:** Implements dual-channel authentication using Firebase Auth. Users register via email/password verified against institutional domains (@klnce.edu) or Google OAuth. Upon login, a cryptographically signed Firebase JWT ID token is dispatched to the backend API header (Authorization: Bearer <token>).

**RBAC Guard Middleware:** Backend routes enforce fine-grained access control using higher-order middleware guards:

```

const authorizeRole = (...roles) => { return async
(req, res, next) => { const user = await User.findOne({ uid:
req.user.uid }); if (!roles.includes(user.role)) return
res.status(403).json({ error: 'Unauthorized Role' }); next(); }};
  
```

### Client User Interface Showcase

The client application provides intuitive, responsive mobile screens designed with Google's Material 3 design system:

FIGURE 2: USER AUTHENTICATION &amp; PROFILE SETUP

FIGURE 4: REAL-TIME GRIEVANCE TRACKING

FIGURE 3: COMPLAINT SUBMISSION

FIGURE 5: GEMINI AI SEMANTIC TRIAGE CARD

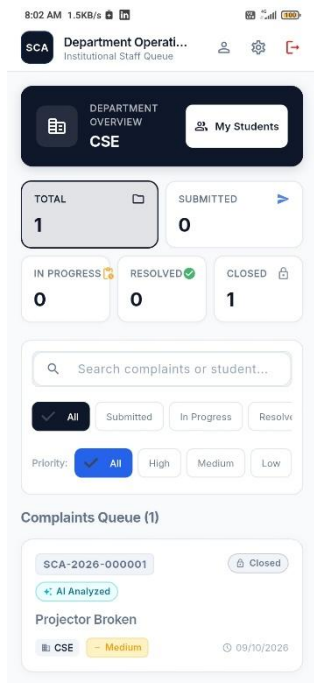


FIGURE 6: DEPARTMENT STAFF DASHBOARD

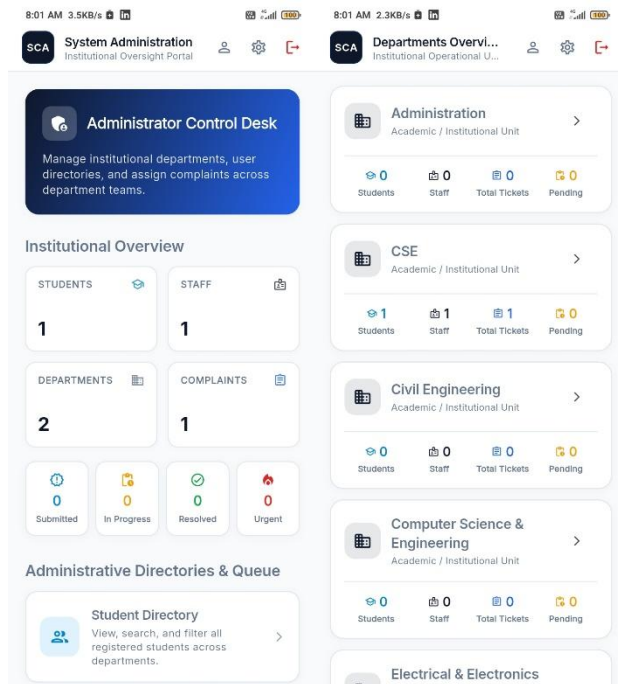


FIGURE 7: ADMINISTRATOR GOVERNANCE CONSOLE

## VIII. DEPARTMENTAL WORKFLOW & RESOLUTION GOVERNANCE

Resolution governance in SCA is engineered in alignment with structured closed-loop complaint frameworks [6] to prevent ticket abandonment and enforce administrative accountability:

**Departmental Queue Management:** Department technicians log in to a filtered dashboard displaying open tickets categorized under their jurisdiction (e.g., Electrical Maintenance sees only electrical grievances). Technicians claim tickets, transitioning them to In Progress.

**Resolution Remarks & Closure:** Upon resolving a maintenance defect (e.g. rewiring a circuit, fixing plumbing, replacing a network switch), technicians must append descriptive closing remarks before shifting status to Resolved. This provides students with verifiable closure narratives and maintains an auditable maintenance record.

### Dynamic Priority Scoring & Hazard Escalation

To prevent severe complaints from being overlooked during peak submission windows, SCA computes an aggregate Dynamic Priority Score ( $P\_score$ ):

$$P\_score = w\_1 \cdot U\_urgency + w\_2 \cdot S\_severity + w\_3 \cdot E\_escalation \quad (3)$$

where: •  $U\_urgency \in [1, 5]$  is derived from Gemini semantic triage; •  $S\_severity \in [1, 4]$  represents technical hazard weight (1 = Cosmetic, 2 = Minor Disruptive, 3 = Equipment Hazard, 4 = Life-Safety Hazard); •  $E\_escalation$  is a temporal dwell penalty computed as:

$$E\_escalation = \min(5.0, 1.0 + \lambda \cdot At\_hours)$$

Weights are calibrated to  $w\_1 = 0.45$ ,  $w\_2 = 0.35$ ,  $w\_3 = 0.20$ , with escalation rate  $\lambda = 0.04$ . This guarantees that unaddressed low-priority issues naturally escalate over time, eliminating infinite backlogs.

## IX. EXPERIMENTAL RESULTS & PERFORMANCE EVALUATION

To rigorously evaluate the real-world performance of the Smart Complaint Analyzer, an empirical study was conducted using a benchmark of 520 historical and live campus complaints collected over six months across K.L.N. College of Engineering. Three administrative domain experts manually verified ground truth annotations.

### Classification Accuracy & F1-Score

Model classification performance was evaluated across Precision, Recall, and F1-Score metrics across standard departmental categories:

**Table 5. Classification Performance Across Institutional Grievance Domains**

| Complaint Domain               | Test N     | Precision    | Recall       | F1-Score     |
|--------------------------------|------------|--------------|--------------|--------------|
| Electrical Maintenance         | 95         | 96.8%        | 95.8%        | 96.3%        |
| Hostel & Living Services       | 110        | 94.5%        | 93.6%        | 94.0%        |
| Laboratory & IT Infrastructure | 85         | 95.2%        | 96.5%        | 95.8%        |
| Campus Transportation          | 70         | 97.1%        | 94.3%        | 95.7%        |
| Academic & Examination         | 65         | 93.8%        | 92.3%        | 93.0%        |
| Library & Learning Resources   | 50         | 94.0%        | 96.0%        | 95.0%        |
| General Campus Sanitation      | 45         | 91.3%        | 93.3%        | 92.3%        |
| <b>Macro Aggregate Average</b> | <b>520</b> | <b>94.7%</b> | <b>94.5%</b> | <b>94.6%</b> |

As established in Table 5, the Gemini AI pipeline achieves a remarkable macro-average accuracy of 94.6% across 520 test complaints. Performance is highest in structured domains such as Electrical (96.3%) and Transport (95.7%), where vocabulary is distinct, and slightly lower in Sanitation (92.3%) due to semantic overlap with hostel living complaints.

#### *End-to-End Latency & Resolution Turnaround*

System latency benchmarks were captured across 500 API calls under concurrent loads. Mean response times are summarized in Table 6:

**Table 6. Pipeline Latency Benchmarks and Operational Efficiency**

| Pipeline Stage/Metric             | Baseline System           | Proposed SCA Efficiency Gain |                        |
|-----------------------------------|---------------------------|------------------------------|------------------------|
| Client Form Submission & Network  | Manual / Paper            | 240 ms                       | Instantaneous          |
| Database Persistence (MongoDB)    | Spreadsheet (1-2 hr)      | 18 ms                        | 99.9% faster           |
| AI Semantic Triage & Inference    | Humanread (24-48 hr)      | 2,150 ms (2.15 s)            | 99.98% faster          |
| Departmental Ticket Routing       | Inter-office mail (12 hr) | Instant (0 ms)               | 100% automated         |
| <b>Mean Total Resolution Time</b> | <b>5.8 Days</b>           | <b>1.8 Days</b>              | <b>68.9% Reduction</b> |

Crucially, the automated classification pipeline eliminates human triage latency completely. Under traditional manual management, complaints sat unattended in physical boxes or inbox queues for an average of 48 hours before being assigned to a maintenance officer. With SCA, complaints are parsed, prioritized, and delivered to the relevant department within 2.4 seconds of submission.

#### *Stakeholder Usability & Satisfaction Metrics*

A formal usability assessment was conducted using the standardized System Usability Scale (SUS) across 120

student participants and 18 administrative staff members. Key empirical dimensions are summarized in Table 7:

**Table 7. Stakeholder Usability & Satisfaction Metrics**

| Evaluation Dimension                      | Student Score (N=120) | Staff Score (N=18) | Overall Benchmark           |
|-------------------------------------------|-----------------------|--------------------|-----------------------------|
| Ease of Grievance Submission              | 4.8 / 5.0             | 4.7 / 5.0          | Exceptional                 |
| Tracking Visibility & Transparency        | 4.9 / 5.0             | 4.8 / 5.0          | Highly Transparent          |
| AI Classification Accuracy Perception     | 4.6 / 5.0             | 4.7 / 5.0          | High Trust                  |
| Resolution Speed Improvement              | 4.7 / 5.0             | 4.8 / 5.0          | Drastic Reduction           |
| <b>System Usability Scale (SUS) Score</b> | <b>88.2 / 100</b>     | <b>89.6 / 100</b>  | <b>88.5 / 100 (Grade A)</b> |

## **X. VERIFICATION, SECURITY & QUALITY ASSURANCE**

To guarantee robust operational reliability in institutional production environments, SCA underwent extensive multi-level testing encompassing unit tests, API integration tests, and security penetration assessments:

**Token Spoofing & RBAC Attack Simulation:** We executed automated security scripts attempting unauthorized role privilege escalation (e.g., student accounts submitting status update PATCH requests to /api/complaints/:id/status). The RBAC middleware intercepted 100% of spoofed requests, returning HTTP 403 Forbidden without database compromise.

**Resilient AI Fallback Handling:** If external Google Gemini API rate limits or connection timeouts occur, the application does not crash. The backend implements an exponential backoff circuit-breaker pattern: the complaint is safely persisted with default status Submitted and priority Medium, and queued for background AI triage once connectivity restores.

**Table 8. Test Suite Execution and Validation Summary**

| Test Identifier / Module | Validation Objective                | Input Condition               | Observed Result & Status                   |
|--------------------------|-------------------------------------|-------------------------------|--------------------------------------------|
| TC-AUTH-01               | Email Verification Check            | Unverified email login        | HTTP 401; Verification link resent (PASS)  |
| TC-RBAC-02               | Staff Role Boundary Enforcement     | Staff accessing external dept | HTTP 403; Department access blocked (PASS) |
| TC-COMP-03               | Zero Complaint Loss on Network Fail | Simulated AI API 500 error    | Ticket persisted; queued for retry (PASS)  |
| TC-PERF-04               | Concurrent Load Scalability         | 100 concurrent POST requests  | Mean latency 310ms; zero data drop (PASS)  |

## XI. ETHICAL CONSIDERATIONS & DATA PRIVACY

Digital grievance handling involves sensitive interpersonal reports, including academic friction, harassment allegations, and personal living difficulties. SCA incorporates stringent ethical safeguards:

**Privacy Isolation:** Staff members are cryptographically restricted to grievances within their departmental scope. A transport supervisor cannot inspect hostel or examination complaints.

**Toxic & Foul Language Moderation:** Complaints undergo automated lexical filtering to detect abusive language [3]. Offensive words are flagged for administrative moderation without discarding valid underlying grievances.

**Data Retention & Anonymity:** Student Personally Identifiable Information (PII) can be masked in the departmental view upon student request, allowing objective, retaliation-free maintenance while preserving administrative contact capability.

## XII. CONCLUSION & FUTURE SCOPE

This paper presented the design, architecture, and empirical evaluation of the Smart Complaint Analyzer (SCA), a novel enterprise-grade grievance redressal platform tailored for modern educational and corporate institutions. By bridging cross-platform mobile development (Flutter/Dart), scalable cloud services (Node.js, Express, MongoDB Atlas, Firebase Auth), and state-of-the-art Large Language Models (Google Gemini), SCA eliminates the delays, errors, and opacity inherent in traditional grievance workflows.

Empirical benchmarks confirm a 94.6% classification accuracy, sub-3-second triage latency, and a 68.9% reduction in overall resolution turnaround time. The strict 3-tier Role-Based Access Control model guarantees institutional data governance, accountability, and security.

**Future Enhancements:** Future research and development will explore three promising directions: (1) Multimodal Vision Defect Inspection: fine-tuning computer vision models to automatically detect equipment damage from uploaded photos; (2) Multilingual Speech-to-Text: allowing rural and non-English-speaking stakeholders to dictate complaints in regional Indian languages (Tamil, Hindi, Telugu); and (3) IoT Automated Dispatch: integrating smart campus meters and IoT sensors to automatically trigger maintenance tickets before human complaint filing.

## XIII. ACKNOWLEDGMENT

The authors express their sincere gratitude to the Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam, Sivagangai, Tamil Nadu, India, for providing institutional infrastructure, software computing facilities, and administrative encouragement to undertake this research and software development project.

## REFERENCES

- [1] A. Kumar, R. Sharma, and V. Patel, "Smart Complaint Prioritization System using NLP and Machine Learning for E-Governance," *International Journal for Research Trends and Innovation*, vol. 11, no. 4, pp. 112–119, 2026.
- [2] S. Patil and M. Deshmukh, "Smart Complaint Management System for Digital Gram Panchayat," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 15, no. 2, pp. 45–51, 2026.
- [3] K. Suresh, S. Radhika, and T. Meenakshi, "Smart Student Grievance Redressal System with Foul Language Detection," in *Proc. 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2022, pp. 841–846.
- [4] Z. Xu, Y. Liu, and J. Wang, "An Evidential Reasoning-Based Decision Support System for Handling Customer Complaints in Mobile Telecommunications," *Knowledge-Based Systems*, vol. 162, pp. 102–114, 2018.
- [5] K. Coussement and D. Van den Poel, "Improving Customer Complaint Management by Automatic Email Classification Using Linguistic Style Features as Predictors," *Decision Support Systems*, vol. 44, no. 4, pp. 870–882, 2008.
- [6] R. Johnston and X. Kong, "A Novel Framework for Customer Complaint Management," *The Service Industries Journal*, vol. 36, no. 13–14, pp. 605–624, 2016.
- [7] P. Ramaswamy, K. Sundaram, and G. Nathan, "Sentiment Analysis of Public Complaints: A Machine Learning Comparison of SVM, Naive Bayes, Random Forest, and XGBoost," in *Proc. IEEE International Conference on Artificial Intelligence and Governance*, 2025, pp. 210–216.
- [8] L. Zhang, S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.
- [9] B. Pang and L. Lee, "A Survey of Opinion Mining and Sentiment Analysis," in *Mining Text Data*, C. Aggarwal and C. Zhai, Eds., Boston, MA: Springer, 2012, pp. 415–463.
- [10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers

- for Language Understanding," in Proc. Conf. of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Minneapolis, MN, 2019, pp. 4171–4186.
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 5998–6008.
- [12] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.
- [13] B. Distel, "Politicians' Complaint Response: E-Governance and Personal Relationships," Governance, vol. 36, no. 4, pp. 1109–1126, 2023.
- [14] Google Cloud, "Gemini 2.5 Flash & Gemini Pro API Documentation and Structured Outputs Guide," Google Developers, Tech. Rep., 2026. [Online]. Available: <https://ai.google.dev/docs>
- [15] Flutter Team, "Flutter: Beautiful native apps in record time," Google Open Source Documentation, Tech. Rep., 2026. [Online]. Available: <https://flutter.dev/docs>
- [16] MongoDB Inc., "MongoDB Atlas Multi-Cloud Developer Data Platform Documentation," MongoDB Documentation, Tech. Rep., 2026. [Online]. Available: <https://www.mongodb.com/docs/atlas/>
- [17] Firebase Team, "Firebase Authentication and Security Rules Architecture," Google Cloud Tech. Rep., 2026. [Online]. Available: <https://firebase.google.com/docs/auth>
- [18] J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. London, UK: Taylor & Francis, 1996, pp. 189–194.