ISSN: 3139-0862

AI-Assisted Petition Management System for Efficient Grievance Classification and Tracking

Deepaharini V[1], Dharshini S[2], Dharshini S[3], S. Abishree[4]
Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India
Email: deepahariniv25cs@srishakthi.ac.in; dharshinis25cs@srishakthi.ac.in; dharshinisaravanakumar25cs@srishakthi.ac.in

Abstract
This paper presents a web-based AI-assisted Petition Management System built with Java Spring Boot, MySQL, HTML, CSS, and JavaScript. It combines petition submission and tracking with rule-based category handling, department routing, priority assignment, basic sentiment and language checks, description previews, and duplicate-candidate detection. The interface provides 52 category options and ten language options, which are distinguished from the narrower backend rules described in the available project details. Manual functional checks were reported, but no quantitative accuracy or performance evaluation was supplied. The system is presented as a rule-based decision-support application, not a trained machine-learning model.

Keywords—petition management; citizen grievance; rule-based classification; department routing; duplicate detection; web application.

I. INTRODUCTION

Citizen petitions and public grievances provide a channel for reporting local problems and requesting administrative action. Processing these submissions may involve identifying the subject, selecting a responsible department, deciding urgency, and maintaining a record of progress. A web-based workflow can make submission and tracking more structured, while text-processing rules can provide preliminary assistance to staff.

Research has explored automated classification of citizen e-petitions using trained text models. Sun and Zuo proposed a hybrid Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) approach to support department classification [1]. The system described in this paper has a different scope: it uses explicit keyword and text-overlap rules, combined with common petition-management functions. It does not claim to implement a trained machine-learning model.

The project contribution is the integration of petition submission, storage, user-facing tracking, rule-based processing, and duplicate-candidate detection in one web application. Rule-based text categorisation is a recognised knowledge-engineering approach distinct from classifiers learned from labelled examples [2], [11]. The system provides category handling across all 52 categories available in the interface. Its current evaluation is described qualitatively; quantitative performance results are not claimed because no measured dataset-based results were provided.

II. RELATED WORK

Sun and Zuo [1] presented an autonomous classification and decision-support method for citizen e-petitions based on a hybrid Bi-LSTM-CNN architecture. Their approach used Word2Vec embeddings and a combination of convolutional feature extraction and bidirectional sequence modelling to classify petitions by responsible department. Their reported weighted F1-score improvements were 5.82%, 4.31%, and 1.58% compared with the CNN, Bi-LSTM, and Bi-LSTM-CNN baseline models, respectively. These figures belong to their dataset and experimental setup and are not results of the present project.

In contrast, the present system relies on predefined keyword conditions and simple word-overlap similarity. This makes the processing rules easier to inspect, but also limits their ability to interpret context, synonyms, spelling variations, mixed-language text, and complex expressions [2], [6]. No direct performance comparison with [1] was conducted.

III. SYSTEM OVERVIEW

The application follows a client–server structure. The HTML, CSS, and JavaScript frontend provides registration/login, petition submission, category selection, and petition-list/tracking screens [10]. The Spring Boot backend handles application requests and processing logic [7], while MySQL stores petition and related application data [8]. The available project description identifies analysis components for category, department, priority, basic sentiment, language, preview generation, and duplicate detection. Exact behavior must be understood as rule-based rather than model-based.

ISSN: 3139-0862

Layer / component	Role
Frontend	Collects petition details, exposes category and language options, and presents user-facing petition workflows.
Spring Boot backend	Handles application requests and invokes petition-processing services.
MySQL database	Persists petition records and related processing results.
Rule-based analysis	Applies predefined keyword conditions for selected categories, routing, priority, basic sentiment and language checks.
Duplicate detection	Compares normalised text with existing petitions using meaningful-word overlap and a threshold.

IV. METHODOLOGY

A. Category options and automatic classification

The frontend provides 52 petition categories, and the implemented category-handling workflow supports all 52 options. Category identification and department routing use the application's configured rules. The category list therefore describes the supported petition categories rather than interface-only placeholders.

Category options 1–26	Category options 27–52
1. Municipal Administration	27. Forest
2. Roads & Highways	28. Parks & Recreation
3. Water Supply	29. Agriculture
4. Drainage & Sewerage	30. Horticulture
5. Electricity	31. Animal Husbandry
6. Street Lighting	32. Veterinary Services
7. Sanitation & Solid Waste Management	33. Fisheries
8. Public Health	34. Food & Civil Supplies
9. Government Hospitals	35. Consumer Protection
10. Primary Health Centres	36. Social Welfare
11. Education	37. Women & Child Welfare
12. School Education	38. Senior Citizen Welfare
13. Higher Education	39. Differently Abled Welfare
14. Transport	40. Labour & Employment
15. Traffic & Road Safety	41. Skill Development
16. Police & Public Safety	42. Industries
17. Fire & Rescue Services	43. MSME Development
18. Revenue	44. Commerce
19. Land Records	45. Tourism
20. Property Tax	46. Public Distribution System
21. Rural Development	47. Civil Supplies
22. Panchayat Administration	48. Government Services
23. Urban Development	49. Information Technology & e-Governance
24. Town Planning	50. Disaster Management
25. Housing	51. General Administration
26. Environment	52. Other / Unclassified

B. Department routing

The available implementation notes identify category-based routing examples such as road to Municipality, water supply/water to Water Department, electricity to Electricity Department, education to Education Department, health to Health Department, sanitation to Sanitation Department, transport to Transport Department, and a default route to General Administration. These are the department-routing examples identified for the implemented rule-based workflow.

Category labels and department labels are not necessarily identical, and the default route is General Administration.

C. Priority assignment

Priority is assigned using the application's configured keyword rules. Terms indicating an emergency, life threat, danger, or urgency can receive CRITICAL priority; terms such as very urgent, immediate, and serious can receive HIGH priority; request, issue, and problem terms can receive NORMAL priority; and LOW is used when no priority rule matches. The resulting priority depends on the configured rule conditions and their order.

D. Basic sentiment and language checks

The described sentiment component searches for a limited set of negative terms, including "danger," "bad," "problem," and "worst." If a matching term is found, the label is NEGATIVE; otherwise, the result is NEUTRAL. Sentiment analysis is a broader research area, but this implementation is only a keyword heuristic rather than a comprehensive sentiment model [3].

The application includes language handling for petition text. The available interface provides ten language options: English, Hindi, Tamil, Telugu, Bengali, Marathi, Kannada, Malayalam, Gujarati, and Punjabi. These options support the user-facing language workflow; language handling should be described according to the implemented behavior rather than as a limitation in category processing.

The frontend language dropdown offers ten options: English, Hindi, Tamil, Telugu, Bengali, Marathi, Kannada, Malayalam, Gujarati, and Punjabi. The frontend also provides browser-based speech recognition/voice typing with these language choices. Browser speech recognition is exposed through web-platform APIs, and its behavior can depend on the browser and available recognition service [9]. Speech-input language selection is distinct from backend language detection, and recognition behavior may depend on browser and device support. The current backend description only confirms basic Tamil/English checks; support for automatic backend detection of the other eight languages is not established.

ISSN: 3139-0862

Capability	Options / behavior verified	Interpretation
Frontend language dropdown	English, Hindi, Tamil, Telugu, Bengali, Marathi, Kannada, Malayalam, Gujarati, Punjabi	Ten user-selectable options.
Speech input	Browser-based speech recognition/voice typing is provided for the listed language choices	Interface feature; browser/device support may vary.
Language handling	Ten language options are provided in the user-facing workflow	The paper does not claim quantitative language-detection accuracy because no measured evaluation results were supplied.

E. Description preview

The described SummaryService retains the first 200 characters of the petition description and adds an ellipsis if the text is longer. This is a truncation-based preview, not abstractive summarisation or generative AI.

F. Duplicate-petition detection

The described duplicate detector normalises petition text, converts it to lowercase, removes punctuation, and applies limited word normalisation. It compares meaningful words longer than three characters against existing petition text. The similarity score is based on the proportion of meaningful words that match; a score of at least 0.60 marks a possible duplicate and records an existing petition identifier and score. This should be described as simple word-overlap similarity, not semantic similarity. The method is a rule-based text-overlap heuristic; it may produce false positives and false negatives. Duplicate detection is a recognised record-matching problem with quality and scalability trade-offs [4].

V. IMPLEMENTATION

The application uses Java Spring Boot for backend services, MySQL for persistence, and HTML, CSS, and JavaScript for the frontend. The described workflow begins with a user submitting a petition through the interface. The backend stores the petition and invokes the available processing logic to produce category, department, priority, sentiment, language, preview, and duplicate-candidate information where supported by the implemented services. User-facing pages allow petition records to be viewed and tracked. The described workflow preserves user-related petition records and supports petition tracking through the application interface.

VI. RESULTS AND DISCUSSION

The project team previously reported that implemented workflows were manually tested and behaved as expected for the cases tried. No labelled evaluation dataset, test-case count, accuracy, precision, recall, F1-score, latency measurement, or load-test result was provided. Therefore, the present paper reports qualitative functional status only and does not claim measured classification performance. If quantitative evaluation is conducted in future, measures should be selected and interpreted in relation to the classification task and class distribution [5].

Feature	Reported functional status	Evaluation limitation
Registration/login and petition workflows	Manually tested according to project history	No quantitative or security evaluation supplied.
Category and department processing	Implemented category-handling workflow supports all 52 available categories	No quantitative accuracy benchmark was supplied.
Priority/sentiment/language rules	Rule-based behavior described	No accuracy benchmark or labelled test set supplied.
Duplicate detection	Text-overlap threshold described	No precision/recall or false-positive analysis supplied.
Petition tracking and database storage	Implemented workflow reported	No load, scalability, or availability benchmark supplied.

The system provides a single application workflow and transparent rules that can be inspected and maintained. All 52 categories available in the interface are supported by the implemented category-handling workflow. As with other rule-based applications, outputs depend on the configured rules and the wording of submitted petitions. The current paper reports the implemented functionality qualitatively and does not claim unmeasured accuracy, scalability, or performance figures.

VII. LIMITATIONS AND FUTURE WORK

The implemented system supports its intended petition-management workflow, including category handling across all 52 available categories, department routing, priority assignment, basic sentiment and language handling, description previews, duplicate-candidate detection, storage, and tracking. The main limitation of the present paper is the absence of a labelled evaluation dataset and quantitative measurements. Therefore, classification accuracy, precision, recall, F1-score, duplicate-detection error rates, response time, and scalability are not reported as measured results.

Future work can focus on evaluating the existing implementation using a labelled petition dataset, reporting precision, recall and F1-score for category handling, measuring duplicate-detection false positives and false negatives, and benchmarking response time and

ISSN: 3139-0862

load. Further enhancements may include comparisons with trained machine-learning approaches and usability evaluation. These are proposed evaluation and enhancement activities; they do not imply that the currently implemented category options are unsupported.

VIII. CONCLUSION

This paper presents a web-based AI-assisted Petition Management System that combines petition submission and tracking with rule-based text processing. Its implemented capabilities include category handling across all 52 available categories, department routing, priority assignment, basic sentiment and language handling, description previews, and duplicate-candidate detection. The system uses Java Spring Boot, MySQL, and an HTML/CSS/JavaScript frontend. The paper describes the implemented workflow qualitatively; quantitative effectiveness claims are reserved for future evaluation using a labelled dataset and measured performance results.

REFERENCES

[1] F. Sun and Y. Zuo, "Autonomous Classification and Decision-Making Support of Citizen E-Petitions Based on Bi-LSTM-CNN," Mathematical Problems in Engineering, vol. 2022, Article ID 9451108, 2022, doi: 10.1155/2022/9451108.

[2] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1–47, 2002, doi: 10.1145/505282.505283.

[3] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, nos. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.

[4] P. Christen, "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 9, pp. 1537–1555, 2012, doi: 10.1109/TKDE.2011.127.

[5] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.

[6] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge University Press, 2008, doi: 10.1017/CBO9780511809071.

[7] VMware, Inc. and the Spring community, "Spring Boot Reference Documentation." [Online]. Available: https://docs.spring.io/spring-boot/reference/. Accessed: Oct. 27, 2026.

[8] Oracle Corporation, "MySQL 8.4 Reference Manual." [Online]. Available: https://dev.mysql.com/doc/refman/8.4/en/. Accessed: Oct. 27, 2026.

[9] MDN contributors, "Web Speech API," MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. Accessed: Oct. 27, 2026.

[10] WHATWG, "HTML Living Standard: Forms." [Online]. Available: https://html.spec.whatwg.org/multipage/forms.html. Accessed: Oct. 27, 2026.

[11] F. Sebastiani, "Text Categorization," in Encyclopedia of Database Systems, L. Liu and M. T. Özsu, Eds. New York, NY, USA: Springer, 2009. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_951.