

Multi-Location IT Ticketing Systems

Mr. Bhushan V. Mali

Assistant Professor
Department of Computer Science and
Engineering
Sandip University
Nashik, Maharashtra, India 422213

Asmit Mahajan

School of Computer Sciences and
Engineering
Department of Computer Science and
Engineering
Sandip University
Nashik, Maharashtra, India 422213

Abhinav Kalse

School of Computer Sciences and
Engineering
Department of Computer Science and
Engineering
Sandip University
Nashik, Maharashtra, India 422213

Aryan Patil

School of Computer Sciences and
Engineering
Department of Computer Science and
Engineering
Sandip University
Nashik, Maharashtra, India 422213

Omkar Jadhav

School of Computer Sciences and
Engineering
Department of Computer Science and
Engineering
Sandip University
Nashik, Maharashtra, India 422213

Abstract — Organizations that have offices in many places need an IT support system that can record, assign, follow up on and finish tickets from each location. A central help desk can give a workflow, but it must also think about where the user is, which support team is at that site, how busy the engineers are, what the service-level rules say and who the user is. This paper looks at work on IT service management (ITSM), how tickets are classified, how they are sent to the people, how to work across many locations and how to process tickets with smart methods. The paper then links this review to our project, a Multi-Location IT Ticketing System that we plan to build as a web application using PHP and MySQL. The revised paper clearly splits the findings from studies, the design we propose for the project and ideas for future AI work. It does not use the 90-day test or numbers that were in the earlier version. We also explain a score that uses the site, the engineer's skill, whether the engineer is free, how many tickets the engineer has and how urgent the service level is. This gives the project a start while leaving natural-language processing, machine-learning routing and predictive AIOps for later.

Keywords — *Multi-Location IT Ticketing System; IT Service Management; IT Support; Ticket Classification; Location-Aware Routing; Role-Based Access Control; SLA; Natural Language Processing.*

I. INTRODUCTION

IT support is used to manage incidents, service requests, technical problems and the talk between users and support staff. In a company one support queue can be enough. When a company has many offices, the system must also record where the request came from, which team handles that place, which engineers can help and when a ticket has to be sent up the chain. IT service management frameworks give a company a way to handle incidents and service requests. A framework alone does not give the code that runs a multi-location ticket system. The program still has to keep track of people, their roles, where they are, tickets, who gets a ticket, notes, changes in status, how important a ticket is and the service rules. ITIL 4 splits work into parts like Service Desk, Incident Management, Service Request Management and Monitoring and Event Management.

This is helpful when we design how a ticket system will work [1].

Most research on automated help systems focuses on classifying tickets. Early work used TF-IDF and old machine-learning tools, while newer work has tried networks and transformer models. Revina and colleagues found that simple algorithms can still do well if the ticket description and chosen features are good [2]. Fuchs and teammates reviewed machine-learning ways to automate tickets and said that classifying tickets and handling incidents automatically are key areas [3]. Other studies have looked at moving tickets with graph-based ideas and at classifying enterprise tickets with transformer models [4], [5].

Our project is about building a practical Multi-Location IT Ticketing System, not about inventing an AI model. The planned system will have three roles: Administrator, IT Engineer and User. The Administrator will manage people, places and settings. The IT Engineer will work on tickets that are given to him, add notes, change status and record how the problem was fixed. The User will create tickets, can see and follow them. The system also saves where the ticket was made and the date so that it can support branches and later give useful data.

With this focus the paper has three main goals: (i) look at ways that fit multi-location ticketing; (ii) find the gap between smart ticket processing and a small practical role-based ticket system; and (iii) suggest a clear design and routing rule that can first be built with normal web and database tools and later grow with AI.

II. SURVEY METHODOLOGY

This paper is written as a survey that also includes a synthesis focused on a project. The review covers work that relates to IT service management and ticket lifecycle, automatic ticket classification, ticket routing and reassignment, dispatch that's aware of location and workload, role-based access, service management across multiple locations, monitoring of

service level agreements and IT support that uses artificial intelligence.

The literature search used combinations of terms such as “IT service management”, “IT support ticket”, “ticket classification”, “support ticket automation”, “ticket routing”, “ticket reassignment”, “location-aware routing”, “machine learning ticket classification”, “NLP helpdesk” and “multi-location IT support”. We preferred sources from IEEE, ACM, ScienceDirect/Elsevier, Springer, ACL Anthology, MDPI and established conference repositories. We focused on literature from 2020 to 2026 for AI and ticket-automation work but kept older sources when they described methods or frameworks that still matter.

We included studies that were peer-reviewed journal or conference papers, that related directly to IT support, service-desk tickets, ITSM or ticket automation and that gave methodological detail to understand the approach. We excluded studies that did not relate to IT or service tickets, that were generic text-classification papers when a more direct support-ticket source existed or that could not be linked to the goals of this project. The selected work was then grouped by function rather than only by publication year.

The reviewed work is grouped into four areas: ticket intelligence, ticket routing and reassignment, enterprise workflow and access control, and project implementation requirements. This structure keeps research separate from the architecture we propose for our project and from the AI features that we plan for future implementation.

TABLE I. SUMMARY OF THE SURVEYED TECHNICAL CATEGORIES

Category	Representative approach	Relevance to MLTS project	Main limitation identified
Ticket classification	TF-IDF, linguistic features, classical ML, BERT/transformers	Can suggest category/priority from ticket text	Needs labelled domain data and evaluation
Routing / reassignment	Rule-based routing, graph models, skill/workload routing	Supports assignment to suitable IT Engineer	Location and real-time availability must be integrated
Workflow / ITSM	ITIL practices, service desk workflows	Defines incident/request lifecycle	Framework guidance does not implement application logic
Access / multi-location	RBAC, tenant/location scopes	Supports Admin, Engineer, User separation	Fine-grained cross-location policy can become complex
Analytics / AI	Predictive classification, AIOps, LLM assistance	Future automation and proactive support	Requires operational telemetry and validation

III. LITERATURE SYNTHESIS

A. ITSM and Ticket Lifecycle

ITSM systems provide a process for receiving, classifying, assigning, monitoring and closing support requests. In our project the main ticket states are New, Assigned, In Progress, Resolved and Closed. Comments and timestamps give an audit trail. Where needed, incident-type problems can be separated from predefined service requests. ITIL 4 is used as a process

reference, while the actual application must implement the required fields, permissions, screens and status transitions [1].

B. Automated Ticket Classification

Ticket classification is one of the common areas where automation is used in support systems. Revina et al. compared ticket representations and classifiers for IT ticket complexity and showed that relatively simple methods can still be useful when the features are chosen well [2]. Al-Hawari and Barham presented a helpdesk system that links ticket classification with user and administrator workflows, showing how an ML component can be combined with a conventional helpdesk application [6]. Transformer models offer another option for ticket classification. BERT introduced contextual language representations that can be fine-tuned for classification tasks [7]. Recent ITSM work compared logistic regression with multilingual BERT and XLM-Roberta for enterprise support requests. These studies also show why class imbalance and realistic evaluation are important [5]. In our project these models are considered extensions and are not presented as already-trained components.

C. Ticket Routing and Reassignment

Correctly classifying a ticket does not automatically mean that the ticket will reach the right engineer. Assignment can depend on the ticket category, required skill, current workload, engineer location, availability, priority and SLA limits. Schad et al. studied helpdesk ticket reassignments with a graph-based approach and showed that assignment history contains useful information about ticket complexity and organizational workflow [4]. This makes assignment history a useful data source for improvements.

For the version of a project-sized multi-location system, a clear rule or scoring method is easier to inspect than an untested ML model. Therefore, the routing method described in Section V uses visible factors and does not claim machine-learning performance that has not been tested.

D. AI and Intelligent Support

Recent work goes beyond classification and looks at hierarchical labels, contextual language models, graph-based workflow analysis and mixes of rules and machine learning. Zangari et al. reviewed ticket automation research and found classification to be a main automation task, including contextual and hierarchical methods [8]. A 2026 enterprise study also compared transformer-based classifiers and showed that the choice depends on the data, class balance and deployment constraints [5]. For a student project these results suggest starting with a ticket-management system and adding AI only after enough data and an evaluation method are ready.

IV. COMPARATIVE ANALYSIS OF EXISTING APPROACHES

The approaches found in the literature cover parts of the ticketing problem. A centralized ITSM workflow offers a process, but it may not manage branch-level assignment in detail. Text classifiers can help with categorization. They need suitable ticket data and labels. Reassignment models can learn from workflow data, while transformer models give stronger language understanding but require more data and computing power. A multi-location project therefore needs a workflow that already holds location, role and assignment information before more advanced AI is added.

TABLE II. COMPARISON OF EXISTING APPROACHES

Approach	Primary decision	Data needed	Project use
Manual / rule-based	Who handles the ticket?	Ticket fields, roles, location, rules	Initial implementation
TF-IDF + classical ML	What category is the ticket?	Historical labelled ticket text	Future baseline AI
Transformer NLP	What does the ticket mean?	Larger labelled text corpus	Future advanced AI
Graph-based model	Which assignment path is difficult?	Assignment/reassignment history	Future analytics
Location-aware score	Which eligible engineer should receive it?	Site, skill, workload, SLA, availability	Proposed project logic

V. PROPOSED PROJECT-ALIGNED MLTS FRAMEWORK

The proposed framework follows the scope of the Multi-Location IT Ticketing System project. It is a web application that uses PHP for the application layer and MySQL for data storage. The current project is not described as a microservices or hybrid-cloud system because those terms would go beyond the implementation. They can be considered later if the system expands to a larger deployment.

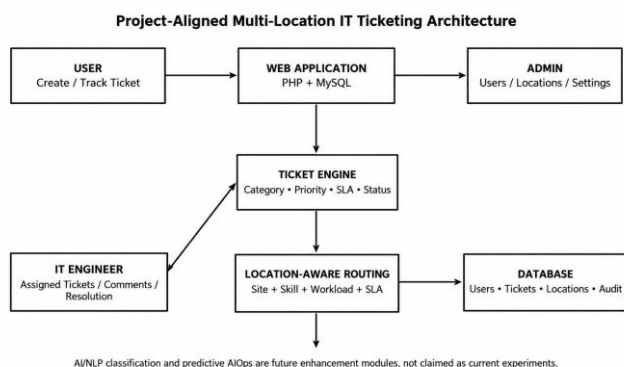


Fig. 1. Project-aligned architecture of the proposed Multi-Location IT Ticketing System.

A. User and Role Layer

The system has three roles. The Administrator has access to manage users, locations and system settings. The IT Engineer works with assigned tickets and can update status, add comments and record a resolution. The User creates tickets and views and tracks tickets belonging to that user. Role-Based Access Control (RBAC) is applied at the application level so that each role can access the functions assigned to it.

B. Ticket and Location Data Model

A ticket should store a ticket ID, requester, location or site title, description, category, priority, status, assigned engineer, creation date, update date, SLA target when applicable and resolution details. Location is stored explicitly instead of being inferred only from the user account. This makes it possible to operate branches in the same system and later generate location-wise reports.

C. Ticket Lifecycle

The workflow starts when a User creates a ticket. The system records the creation date and sets the initial status. The Administrator or routing logic then assigns the ticket to an IT Engineer. The engineer moves the ticket to In Progress, adds comments when needed and records the resolution. After confirmation the ticket can be Closed. Timestamps for these changes can later be used to calculate response time, resolution time, ticket ageing and SLA compliance.

D. Location-Aware Routing Formulation

To provide an analytical part without claiming an untested ML routing algorithm, we propose a simple weighted routing score. For an unassigned ticket t and an eligible engineer e , the score is:

$$R(t,e) = w_1S + w_2L + w_3A + w_4W + w_5U \quad (1)$$

Here S stands for skill/category compatibility, L stands for location compatibility, A stands for engineer availability, W stands for workload balance and U stands for SLA urgency. The weights w_1 through w_5 are adjustable and together add up to 1. A higher score indicates that the engineer is considered a better fit for the assignment. The weights should be chosen or tuned with project data; they should not be set to produce a desired outcome.

Each factor can be scaled between 0 and 1. Skill compatibility is 1 if the engineer is qualified for the ticket category, otherwise it is 0. Location compatibility gives preference to an engineer who works at the branch. Availability shows whether the engineer is currently free. Workload is based on the number of tickets; a lower active-ticket count receives a higher score. SLA urgency rises as the remaining response or resolution time shrinks. Because all factors are visible, an administrator can see why an assignment was chosen. In the future this rule can be compared with a model that is learned from data.

E. AI / NLP Extension

The AI portion is left as future work. A first experiment might use TF-IDF with Logistic Regression, Naive Bayes or SVM to predict ticket categories because these models are light and easy to understand. Later work could compare them with multilingual transformer models. That experiment would require a documented dataset, a train / validation / test split, class distribution, preprocessing steps, hyperparameters and metrics such as precision, recall, macro-F1 and a confusion matrix. This paper does not report any AI accuracy because the project does not yet have a validated training dataset or finished experiment.

F. Calendar and Operational Tracking

The project records the date when each ticket is created. A calendar view can help administrators and engineers look at ticket creation history. If more timestamps are collected later, the same data can be used to calculate first-response time (FRT), mean time to resolution (MTTR), ticket ageing, SLA compliance and workload by location. These numbers must come from project records, not from simulated data.

VI. ANALYTICAL MAPPING OF LITERATURE TO THE PROPOSED PROJECT

Table III separates the insights from studies from the design we propose for our project and from work that still needs

validation. This separation solves the issue of mixing literature observations, proposed architecture and experimental results.

TABLE III. MAPPING OF LITERATURE INSIGHTS TO THE PROPOSED PROJECT

Problem	Literature insight	Project response	Validation status
Ticket categorization	ML/NLP can automate classification [2], [5], [6], [8]	Keep category field; prepare future labelled dataset	Not experimentally evaluated
Wrong assignment	Reassignment history can reveal routing complexity [4]	Use assignment history and explicit routing score	Analytical proposal
Multiple branches	Location is operationally important for distributed support	Store site/location with every ticket and engineer	Implementation requirement
Workload imbalance	Automated support can reduce manual distribution effort [3], [8]	Include active workload in routing score	To be evaluated
SLA management	ITSM requires consistent service handling [1]	Store priority, target time and timestamps	To be evaluated on project data
AI triage	Transformers can model contextual ticket text [5], [7]	Add NLP as future module after data collection	Future work

VII. EVALUATION PLAN FOR THE PROJECT

The earlier manuscript included numerical results from a 90-day simulation, but there was not enough detail to repeat the experiment. Those results have been removed. Once the system is built, the project can be evaluated with real tickets or well-documented synthetic tickets. The evaluation should list the number of tickets, locations, engineers, ticket categories, engineer skills, working schedules, routing rules and SLA definitions. If a simulation is used, the number of independent runs should also be reported.

For the ticketing system, useful metrics include ticket creation success rate, assignment time, average number of open tickets per engineer, ticket ageing, SLA compliance and resolution time. The proposed routing score can be compared with a baseline such as round-robin or single-queue assignment. If an AI classifier is added later, evaluate it with accuracy, macro-F1, precision, recall, per-class results and a confusion matrix. For repeated simulation runs, report the standard deviation and confidence intervals.

A clear comparison can use three configurations: Baseline A – fixed-queue assignment; Baseline B – category-only rule-based assignment; and the proposed configuration – category, location, skill, availability, workload and SLA scoring. Any AI classifier must be evaluated separately. This approach helps identify which part of the system causes a change.

VIII. RESEARCH GAPS AND DISCUSSION

The literature shows work on automatic ticket classification, but ticket classification is only one part of multi-location support. A ticket may be classified correctly, yet the ticket may still be assigned to an unavailable or overloaded engineer or to another branch when local support is available. The proposed

project therefore stores location, role, workload and SLA information as part of the data.

Another gap is the difference between research prototypes and small academic implementations. Some advanced methods need historical datasets, detailed labels, significant computing resources or enterprise telemetry. A PHP and MySQL implementation can first provide the foundation and collect structured ticket histories. Those records can later be used to create a labelled dataset for AI experiments.

Explainability is important. A transparent routing score lets an administrator see the reasons for an assignment. These reasons include matching skills, same-site availability, lower workload or higher SLA urgency. A future learned model can then be compared against this baseline. This keeps the focus on improving the ticket workflow, not just on optimizing a classification metric.

IX. FUTURE ENHANCEMENTS

Future work can be carried out in stages. First, historical ticket data must be cleaned, with labels for category, priority and resolution group. Second, a lightweight NLP classifier can be compared with classical baselines. Third, the routing score can be compared with graph-based methods based on assignment history. Finally, monitoring data and AI-assisted knowledge retrieval can be added for support and suggested resolutions.

Generative AI could also be added for tasks such as summarizing ticket histories, suggesting troubleshooting steps, drafting engineer comments and helping users with common requests. Any automated action must first be supported by validation, access control, logging and human approval. Multilingual support is another extension, but it should be evaluated using the actual language distribution of the target deployment.

X. CONCLUSION

This survey links ITSM workflow with the location and operational information required in a multi-location ticketing system. Previous studies show results from ticket classification, reassignment prediction and transformer-based NLP, but these methods depend on suitable data and proper evaluation. Our project therefore starts with a PHP-MySQL core that contains role-based access, ticket lifecycle management, location information, engineer assignment, comments, status tracking and ticket-date records. The proposed routing score provides an analytical link between the current system and future AI-based routing. The unsupported 90-day simulation and numerical improvement claim from the earlier version have been removed. When the project is experimentally evaluated, results should be reported with defined data, baselines, configurations, repeated runs and statistical measures. Keeping the survey findings, project design and future experiments separate makes the paper easier to understand and reproduce.

ACKNOWLEDGMENT

The authors sincerely thank the Department of Computer Science and Engineering, Sandip University, Nashik for its support during the preparation of this survey and project work. The authors also acknowledge the assistance of faculty members and library resources used to identify literature.

REFERENCES

- [1] PeopleCert, "ITIL 4 Practices," PeopleCert, 2024–2026. Available from the official ITIL practice resources.
- [2] A. Revina, K. Búza, and V. G. Meister, "IT Ticket Classification: The Simpler, the Better," *IEEE Access*, vol. 8, pp. 193380–193395, 2020, doi: 10.1109/ACCESS.2020.3032840.
- [3] S. Fuchs, C. Drieschner, and H. Wittges, "Improving Support Ticket Systems Using Machine Learning: A Literature Review," *Proc. Hawaii International Conference on System Sciences (HICSS-55)*, 2022, doi: 10.24251/HICSS.2022.238.
- [4] J. Schad, R. Sambasivan, and C. Woodward, "Predicting help desk ticket reassignments with graph convolutional networks," *Machine Learning with Applications*, vol. 7, art. 100237, 2022, doi: 10.1016/j.mlwa.2021.100237.
- [5] A. Razma and R. Jurkus, "AI-Based Classification of IT Support Requests in Enterprise Service Management Systems," *Systems*, vol. 14, no. 2, art. 223, 2026, doi: 10.3390/systems14020223.
- [6] F. Al-Hawari and H. Barham, "A machine learning-based help desk system for IT service management," *Journal of King Saud University – Computer and Information Sciences*, vol. 33, no. 6, pp. 702–718, 2021, doi: 10.1016/j.jksuci.2019.04.001.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proc. NAACL-HLT*, pp. 4171–4186, 2019, doi: 10.18653/v1/N19-1423.
- [8] A. Zangari, M. Marcuzzo, M. Schiavi Nato, A. Gasparotto, and A. Albarelli, "Ticket automation: An insight into current research with applications to multi-level classification scenarios," *Expert Systems with Applications*, vol. 225, art. 119984, 2023, doi: 10.1016/j.eswa.2023.119984.
- [9] L. S. Benitez Pereira et al., "Machine Learning for Classification of IT Support Tickets," *Proc. 2023 International Conference on Cyber Management and Engineering (CyMaEn)*, 2023, doi: 10.1109/CyMaEn57228.2023.10051041.
- [10] S. Seanbudh and T. Siriborvornratanakul, "Using Machine Learning to Improve Ticket Classification for IT Service Management System," *Science and Engineering Connect*, 2022.
- [11] N. Crosley and I. Wasito, "Improving IT Support Efficiency Using AI-Driven Ticket Random Forest Classification Technique," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 4, pp. 2283–2293, 2023, doi: 10.33395/sinkron.v8i4.12925.