IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 28, Issue 4, Ser. 1 (July. – August. 2026), PP 44-50
www.iosrjournals.org

A Rule-Based System For IT Help Desk Ticket Prioritization

Bereket Desta, Edip Senyurek
(Department Of Computer Engineering, Vistula University, Poland)
(Department Of Computer Engineering, Vistula University, Poland)

Abstract:
IT help desks get numerous tickets per day the range of incidents is quite diverse from small device issues up to disruptions of business critical services. Proper prioritization is crucial since the higher the incident impact, the faster it should be escalated. However the manual process of prioritization may lead to inconsistencies since different service desk agents assess urgency and impact differently.
In this paper, the design and implementation of the proposed IT help desk ticket prioritization system based on rules are discussed. The proposed approach determines the ticket priority based on four input factor impact, urgency, business criticality, and workaround availability. For each factor, a numerical value is provided. Then, the ticket priority is calculated according to these values and is represented by one of the following four priority levels Low, Medium, High, or Critical. The prototype of the system is designed as an application including a database for storing ticket data and resulting priorities.
The performance of the prototype has been analyzed with eight simulated IT support tickets with common issues such as password reset request, problems with Microsoft Outlook access, disruptions of a business critical application, inability to establish the connection via VPN, and disruptions of Teams communication platform. In all cases, the ticket priority was set properly, thus, the system achieved applied the prioritization rules to all simulated tickets and resulted 100% accuracy rate.
The proposed IT help desk ticket prioritization system provides a simple and transparent tool for determining the ticket priority. The suggested solution could be helpful for service desk agents since it excludes the subjective assessment of an issue and gives more attention to business critical incidents. Further research may include the employment of historical data and machine learning algorithms in the ticket prioritization process.
Key Word: *IT Service Management, Help Desk, Incident Management, Ticket Prioritization, Rule-Based System, IT Support.*

Date of Submission: 04-07-2026 Date of Acceptance: 14-07-2026

I. Introduction

IT services are vital for the operation of contemporary organizations. Employees use emails, collaboration platforms, virtual private networks, network drives, applications, and authentication tools to complete their tasks. When such services do not work correctly, the users create help desk tickets.

A typical IT support ticket contains information about the technical problem, the user or users involved, the business service impacted, and the importance of the request. The service desk staff member must analyze this information and determine the priority of the ticket, i.e., how soon the problem needs to be addressed.

Ticket prioritization is needed since the technical problems have various degrees of importance for the business [2]. While one employee who needs to replace a mouse is likely to need help, this incident does not imply that the operations of the entire company are going to be seriously impacted. On the other hand, malfunction of the email server, inability to connect through a VPN connection, or disruption of a core business application is critical.

Incorrect prioritization may cause several issues. The low-priority tickets may get too much attention, while high-priority incidents may stay unresolved for a long time. The delays in addressing critical incidents may negatively affect employee productivity, customer service, business continuity, and compliance [4]. Thus, organizations need an adequate and systematic way of deciding if a ticket should be classified as Low, Medium, High, or Critical.

Many modern ticket management systems employ automation to classify, assign and prioritize tickets and some systems even use machine learning and natural language processing techniques based on historical ticket data [3]. However such systems may require large datasets, specialized technical knowledge, and complicated training. For smaller organizations, students or prototype systems, a rule-based system is a viable alternative.

The paper presents a rule-based approach for determining the priority of the tickets at the helpdesk of an IT company. This system works on the principle of using pre-defined input values and calculating ticket priority based on a scoring formula.

The main research question of this study is:

How effectively can a rule-based decision algorithm prioritize IT help desk tickets based on impact, urgency, business criticality, and workaround availability?

The main contributions of this article are:

- The design of a simple rule-based algorithm for prioritizing IT support tickets.
- A prototype system that stores ticket information and automatically calculates priority.
- A structured evaluation method using common IT support scenarios.
- A transparent alternative to more complex machine-learning approaches.

II. Literature Review

IT service management focuses on delivering, supporting, and improving IT services within an organization. Incident management is one of the central activities in IT service management because it aims to restore normal service operation as quickly as possible after an interruption or degradation of service [2].

According to ITIL, impact and urgency are essential elements of the process of incident priority determination. Impact refers to the degree of business disruption caused by an incident, while urgency is the necessity of fast resolution. An incident may have a low impact, affecting one employee, but if an incident affects an entire department or organization, it is likely to have a high impact.

Previous studies have analyzed the use of automated approaches in IT support environments. Ramya et al studied classification of unstructured IT service desk tickets using ensemble of classifiers [3]. Their work indicated that automated ticket processing may allow reducing incorrect routing and speeding up resolution. Mandal et al. [1] suggested multi-modal IT incident management by analyzing ticket text and its attachments such as screenshots and logs. Their work proved that more comprehensive ticket data may help with the routing and resolution processes.

Recently, some studies have investigated machine-learning-based ticket prioritization [5]. Such systems use ticket descriptions, historical priorities, and numerical features to predict the importance of a new incident. While the above approaches can have a useful performance, they require considerable amount of historical data, labeled examples, and more advanced implementation skills.

The approach proposed in this study differs from machine-learning-based methods. Instead of learning from historical records, it uses predefined rules. The advantage of this approach is that every priority decision can be explained clearly. A service desk agent can identify which values increased the priority score and can adjust the rules when organizational requirements change.

Rule-based systems are particularly useful when a limited number of decision factors are known in advance. In the proposed system, ticket priority is calculated using four factors

- impact
- urgency
- business criticality
- workaround availability.

These factors are commonly understandable to service desk staff and can be entered quickly during ticket registration.

III. Methodology

A. Research Design

This research follows a design-and-evaluation approach. First, a rule-based ticket prioritization algorithm was designed. Second, a prototype application was implemented. Finally, the system was evaluated using a set of simulated IT help desk tickets.

The simulated dataset was selected because real enterprise service desk tickets may contain confidential information, personal data, internal system names, or security-sensitive details. Therefore, the tickets used in this study represent realistic but fictional scenarios. No confidential organizational data was used.

The proposed system does not analyse the text of a ticket automatically. Instead, the service desk agent selects structured values during ticket creation. This makes the prototype simple, transparent, and suitable for initial implementation.

B. Ticket Data Model

Each ticket stored in the system contains the information shown that is shown in Table 1.

Table 1. Ticket content

Ticket Code	Unique ticket identifier
Ticket title	Short description of the incident
Affected service	System or application affected
Number of affected users	One user, multiple users, department, or organization
Impact value	Numerical level of business impact
Urgency value	Numerical level of response urgency
Business criticality	Importance of the affected service
Workaround availability	Whether the user can continue working
Priority score	Total score calculated by the algorithm
Final priority	Low, Medium, High, or Critical
Ticket status	Open, In Progress, Resolved, or Closed

The database can be implemented using Microsoft SQL Server, MySQL, SQLite, or another relational database. For a student prototype, SQLite or SQL Server Express is sufficient.

C. Priority Factors

The system calculates ticket priority using four structured values.

1. Impact

Impact describes the number of users or business areas affected by the incident. The values assigned to each impact are shown below in Table 2.

Table 2. Ticket impact

Individual	One employee is affected	1
Small Group	Two to ten employees are affected	2
Department	A full department or team is affected	3
Organization	Multiple departments or the full organization is affected	4

2. Urgency

Urgency describes how quickly the incident requires attention. The criteria for assigning urgency is shown in Table 3.

Table 3. Ticket urgency scale

Low	User can continue working and no deadline exists	1
Medium	User productivity is reduced	2
High	User or group cannot continue normal work	3
Critical	Critical interruption or major deadline risk	4

3. Business Criticality

Business criticality describes the importance of the affected IT service. The classifications for importance are shown in Table 4.

Table 4. Ticket criticality

Standard Service	General application or non-critical device	0
Important Business Service	Frequently used business application	1
Critical Business Service	Core operational, financial, authentication, or security system	2

4. Workaround Availability

A workaround is an alternative method that allows users to continue their work while the incident is being resolved. The scoring for workaround availability is shown in Table 5.

Table 5. Workaround

Workaround Available	User can continue with an alternative solution	0
No Workaround	User cannot continue normal work	1

D. Rule-Based Priority Algorithm

The priority score is calculated using Equation 1.

Priority Score = Impact + Urgency + Business Criticality + Workaround Availability (1)

The possible score range is from 2 to 11. The system maps the score to a final priority level as shown in Table 6.

Table 6. Priority scale

2-3	Low
4-5	Medium
6-8	High
9-11	Critical

The following pseudocode describes the algorithm:
Input: Impact, Urgency, BusinessCriticality, Workaround
PriorityScore = Impact + Urgency + BusinessCriticality + Workaround

If PriorityScore is between 2 and 3:
Priority = Low

Else if PriorityScore is between 4 and 5:
Priority = Medium

Else if PriorityScore is between 6 and 8:
Priority = High

Else:
Priority = Critical

Store ticket details, score, and priority result.
Display final priority to the service desk agent.

The algorithm is intentionally simple. It can be modified according to the requirements of a specific organization. For example, an organization may decide that security incidents should automatically receive High or Critical priority.

System Design and Implementation
The proposed system consists of three main components:
- Ticket Registration Module
- Allows a service desk agent to enter the ticket title, affected service, impact level, urgency level, business criticality, and workaround status.
- Priority Calculation Module
- Applies the rule-based algorithm and determines the total priority score.
- Ticket Management Module
- Stores ticket information in a database and displays the final priority level to the user.

The prototype can be developed using C#, Java, Python, or a web-based technology stack. A possible implementation consists of:
- C# for the application logic
- Microsoft SQL Server for ticket storage
- Windows Forms or ASP.NET for the user interface

Following the creation of a new ticket, the helpdesk operator picks up values from drop-down lists. The scoring and prioritization is done automatically by the system. It makes the calculation consistent and provides immediate suggestions for priority levels. The overall workflow of the system is shown in Figure 1.

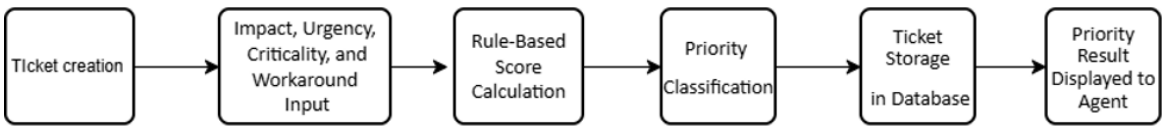


Figure 1. Proposed workflow of the system

The user interface should display both the final priority and the reasons behind it as shown in Figure 2:

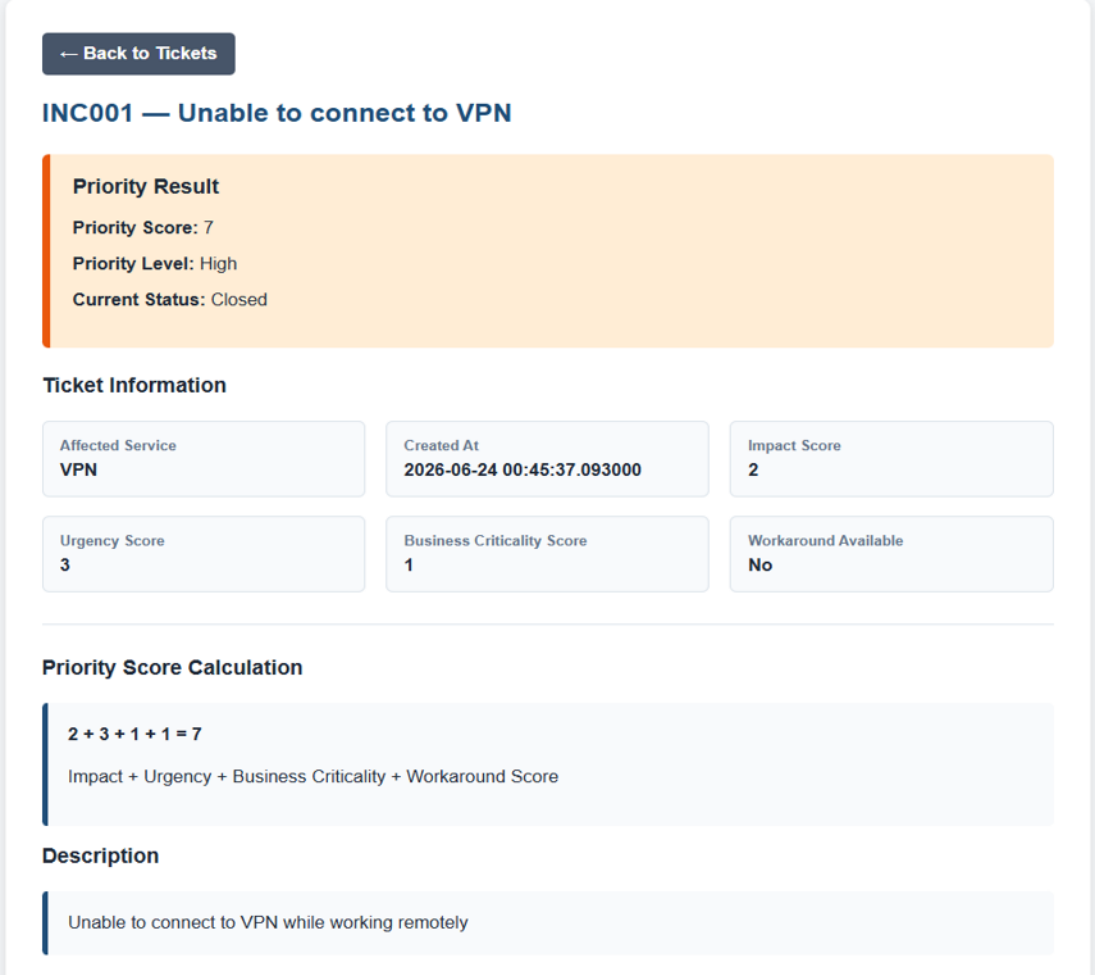


Figure 2. Ticket interface

IV. Results And Discussion

A. Test Dataset

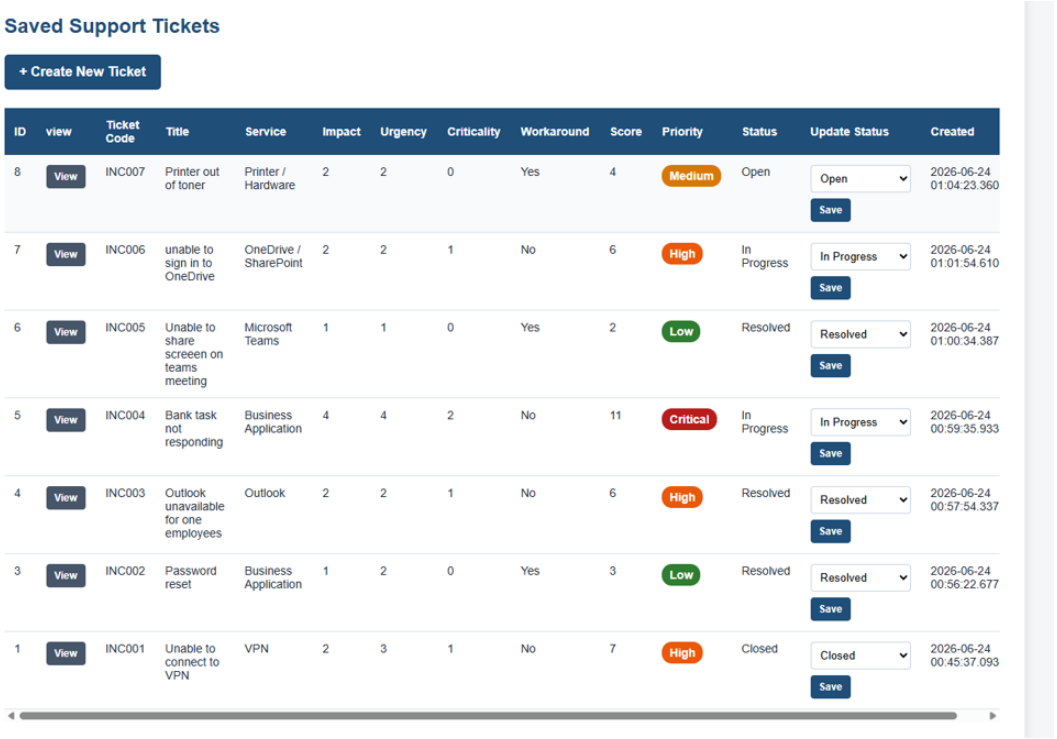
It is recommended that the test be performed with a collection of simulated IT support tickets, which can involve

- Requests for password resets
- Access to emails
- Microsoft teams issues
- Problems connecting to the virtual private network (vpn)
- Printing troubles
- Access to shared mailboxes
- Networking issues
- Problems with virtual desktops
- Multi-factor authentication (mfa) issues
- Outages in business applications

A total of seven ticket scenarios were created for testing. Each ticket was assigned an expected priority before running the system. The expected priority was based on the defined impact, urgency, business criticality, and workaround conditions

B. Example Ticket Results

The final prioritized log for all simulated tickets is shown in Figure 3.



Saved Support Tickets

+ Create New Ticket

ID	view	Ticket Code	Title	Service	Impact	Urgency	Criticality	Workaround	Score	Priority	Status	Update Status	Created
8	View	INC007	Printer out of toner	Printer / Hardware	2	2	0	Yes	4	Medium	Open	Open Save	2026-06-24 01:04:23.360
7	View	INC006	unable to sign in to OneDrive	OneDrive / SharePoint	2	2	1	No	6	High	In Progress	In Progress Save	2026-06-24 01:01:54.610
6	View	INC005	Unable to share screeen on teams meeting	Microsoft Teams	1	1	0	Yes	2	Low	Resolved	Resolved Save	2026-06-24 01:00:34.387
5	View	INC004	Bank task not responding	Business Application	4	4	2	No	11	Critical	In Progress	In Progress Save	2026-06-24 00:59:35.933
4	View	INC003	Outlook unavailable for one employees	Outlook	2	2	1	No	6	High	Resolved	Resolved Save	2026-06-24 00:57:54.337
3	View	INC002	Password reset	Business Application	1	2	0	Yes	3	Low	Resolved	Resolved Save	2026-06-24 00:56:22.677
1	View	INC001	Unable to connect to VPN	VPN	2	3	1	No	7	High	Closed	Closed Save	2026-06-24 00:45:37.093

Figure 3. Created ticket results

The example results demonstrate that the system distinguishes between low-impact individual problems and high-impact organizational outages.

C. Evaluation Metrics

The performance of the system can be evaluated using classification accuracy:
Accuracy = Correctly Prioritized Tickets / Total Tested Tickets × 100 (2)
Based on the final test, the system correctly prioritized seven out of seven tickets.
Accuracy = 7 / 7 × 100 = 100%

The results should also be reviewed manually to identify incorrect classifications. A ticket may be assigned an unexpected priority when the rules do not include an important business context. For example, an issue affecting only one user may still require high priority if the user is responsible for a business critical task.

D. Discussion

There are many benefits to adopting the rule based approach. For one, the process becomes more consistent because every ticket goes through the same algorithmic process [3]. Second, it is transparent because the service desk employees know how the priority of the task was determined. Lastly, there is no need for big historical data sets [5].

However the system also have limitations. It depends on the accuracy of the values selected by the service desk agent. If impact or urgency is entered incorrectly the final priority may also be incorrect. In addition the system cannot fully understand the written description of a ticket. A short ticket may contain important information that is not captured by the selected input values.

Despite these limitations the prototype is useful as an initial decision support tool. It can guide inexperienced agents and promote a more standardized incident management process.

V. Conclusion And Future Work

The proposed model is an instance of the rule based systems for classifying IT support requests by their level of priority. The priority will be calculated in terms of three types of factors: the impact, the urgency, the potential disruption to the business and work-around availability. Depending on the values assigned to these three sets of parameters, each request will have one of four possible levels of priority which are: Low, Medium, High, Critical.

This solution has been designed as an aid to IT service desk staff in providing a clear and uniform method for the initial classification of IT incident priority. Due to the fact that the proposed solution requires less historical

data than some of the advanced machine learning solutions available today, it is appropriate for use in educational projects, small organizations or prototype implementations [5].

A simulated test using representative samples of IT support tickets for example customer complaints demonstrated the ability of this system to differentiate between single low impact events and serious, potentially disruptive disruptions of business operations. Results from further testing will validate the number of support requests that are properly classified by this system prototype.

Future development of this system could occur through a number of enhancements. First, the Ticket description could utilize Natural Language Processing techniques to determine if there are urgency related words present in that text. Second, historical ticket data could be utilized to develop Machine Learning based models such as Decision Trees, Random Forests, or Naive Bayes models. The system could also be used into an existing IT Service Management Platform. Lastly, additional inputs could include Security Impact, Customer Role/Service Level Agreements, Regulatory Requirements etc.

References

[1]. A. Mandal, S. Agarwal, N. Malhotra, G. Sridhara, A. Ray, And D. Swarup, "Improving It Support By Enhancing Incident Management Process With Multi-Modal Analysis," Arxiv Preprint Arxiv:1908.01351, 2019.
[2]. Axelos. Itil Foundation: Itil 4 Edition. London: Tso, 2019.
[3]. C. Ramya, S. P. Paramesh, And K. S. Shreedhara, "Classifying The Unstructured It Service Desk Tickets Using Ensemble Of Classifiers," Arxiv Preprint Arxiv:2103.15822, 2021.
[4]. Iso/Iec 20000-1:2018. Information Technology — Service Management — Part 1: Service Management System Requirements. International Organization For Standardization, 2018.
[5]. M. T. Lê And A. Ait-Bachir, "Learning To Prioritize It Tickets: A Comparative Evaluation Of Embedding-Based Approaches And Fine-Tuned Transformer Models," Arxiv Preprint Arxiv:2512.17916, 2025.