Ticket System Integration with Discourse and Webhooks

A Project Report for the

Software Engineering

(Milestone 1)

Submitted By:

1. Aditya R	**21f1006862@ds.study.iitm.ac.in**
2. Ashutosh Kumar Barnwal	**21f1001709@ds.study.iitm.ac.in**
3. Kanishk Mishra	**21f1006627@ds.study.iitm.ac.in**
4. Nikhil Guru Venkatesh	**21f3000424@ds.study.iitm.ac.in**
5. Shubhankar Jaiswal	**21f1006828@ds.study.iitm.ac.in**
6. Sushobhan Bhargav	**22f1000948@ds.study.iitm.ac.in**
7. Utkarsh Kumar Yadav	**21f1006520@ds.study.iitm.ac.in**

IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

Contents

Problem Statement

Integration Requirements for IITM BS Ticketing System

The IITM BS team has chosen to adopt the **"Online support ticket system for the IITM BS degree program",** for managing queries and concerns within the IITM BS degree program. As part of this integration, the team aims to seamlessly integrate the ticketing system into their existing ecosystem by implementing the following functionalities:

1. **Discourse Integration:**
   Alongside the core functionalities of ticket creation and listing, the ticketing system is required to integrate with Discourse, a platform for community discussion and communication. This integration entails the creation of a Discourse thread for each ticket generated within the ticketing system. Various configurations and rules can be implemented to manage Discourse threads, such as setting initial thread visibility (private/public) and enabling notifications for thread activity (e.g., replies, likes). The integration will leverage the Discourse API documentation to explore available features and determine optimal integration strategies.

2. **Webhooks Integration:**
   To address high-priority tickets promptly, the ticketing system will integrate with webhooks to enable real-time notifications for urgent and high-priority tickets. These notifications will be transmitted to GChat, facilitating swift action and escalation by higher authorities. The integration with Google Chat webhooks will enable seamless communication and collaboration, ensuring efficient handling of critical issues within the IITM BS degree program.

Prototype Solution Implementation:

For this project, the focus will be on developing a prototype solution that demonstrates the integration capabilities of the ticketing system. The prototype will be hosted locally, providing a sandbox environment for testing and validation. Through the prototype, the integration with Discourse and webhooks will be showcased and the ability to handle urgent tickets effectively.

Milestone 1: Identify User Requirements

1.1 Identifying Users of the Application

In the process of requirement gathering and analysis, it is essential to identify the various categories of users who will interact with our system.

Users can be broadly classified into three main categories:

1. **Primary Users:** Primary users are the frequent users of the system who directly engage with its functionalities. They are the primary beneficiaries of the software. For instance, in our project, primary users include:
    - Students of the IITM BS degree program: They will primarily interact with the support ticketing system to raise queries and concerns.
    - Support staff: Responsible for addressing the support tickets raised by students and managing the ticketing system.
    - Admins: Administrators with elevated privileges responsible for managing the ticketing system and overseeing support operations.
2. **Secondary Users:** Secondary users are those who indirectly interact with the system, often through an intermediary. They may not directly utilize the software but rely on it for information or coordination purposes. In our context, secondary users could encompass:
    - Faculty Members: Academic staff who may occasionally interact with the ticketing system to address student concerns related to academic matters.
    - IT Staff: Technical personnel responsible for maintaining and troubleshooting the ticketing system's infrastructure and integrations.
3. **Tertiary Users:** Tertiary users do not directly use the software but are impacted by its introduction or influence its adoption. They may include external stakeholders or entities whose actions or decisions are influenced by the software. For instance, prospective students or regulatory bodies associated with the IITM BS degree program may fall under this category.

- Higher authorities: They may receive notifications for high-priority and urgent tickets through the Webhooks integration, enabling them to handle escalations effectively.
- External Stakeholders: Individuals or entities outside the IITM BS degree program who may interact with the ticketing system for collaboration or information-sharing purposes.

1.2 User Stories for New Features and Integrations

The agile lifecycle commences with Behavior Driven Design (BDD), where we delve into the behavior of the application both before and during development. Continuous refinement of requirements is integral to meet evolving expectations. In the agile approach, requirements are expressed as User Stories, which replace the traditional Software Requirements Specification (SRS) from a planning and documentation standpoint.

User Stories serve as succinct, informal descriptions in plain language, outlining what a user seeks to accomplish within the software product and the value it brings to them. Following the "Role-feature-benefit" pattern, each User Story encapsulates the smallest unit of work achievable within a single sprint:

As a [type of user],

I want [an action],

So that [a benefit / value]

This structured format ensures clarity and alignment between user needs, system functionalities, and the resulting benefits, facilitating efficient development iterations within the agile framework.

User Stories:

As a Student,

Scenario 1

- I want to create a support ticket for a specific concern or query,
- So that I can receive assistance from the support team promptly and should be informed through notification or an email/GChat (webhook).

Scenario 2

- I want to view a list of similar tickets before creating a new one with the help of Discourse integration,
- So that I can avoid duplicating existing support requests and contribute to efficient ticket management.

Scenario 3

- I want to like or +1 an existing support ticket,
- So that popular concerns or queries can be prioritized by the support team.

As a Support Staff,

Scenario 1

- I want to receive notifications when a new support ticket is created through notification or an email/GChat (webhook),
- So that I can promptly address student concerns and assist.

Scenario 2

- I want to mark a ticket as resolved once the concern has been addressed,
- So that students are notified through an email/GChat (webhook) and the ticket is closed appropriately.

As an Admin,

Scenario 1

- I want to categorize resolved tickets and add them to the FAQ section,
- So that future students can access updated FAQs and find answers to common queries efficiently with the integration of Discourse.

Scenario 2

- I want to manage user roles and permissions,
- So that access to system features and data is controlled based on user roles and responsibilities.

As a Faculty Member,

- I want to access the ticketing system to address academic-related concerns raised by students,
- So that I can provide support and guidance to students when needed.

As an IT Staff,

- I want to ensure the reliability and security of the ticketing system's infrastructure,
- So that the system operates smoothly and student data is protected from unauthorized access or breaches.

Acceptance Criteria:

- Integration with Discourse to fetch created tickets and synchronize query data between platforms.
- Implementation of real-time notifications via Google Chat for staff members whenever a new query is created in the ticket system, ensuring prompt attention to user queries.
- Timestamps are displayed on notifications to provide staff members with context regarding query creation times and enable them to prioritize responses accordingly.

- Seamless navigation between Discourse and the ticket system to facilitate easy access to previous query resolutions and prevent redundant efforts by staff members.
- User-friendly interface for staff members to manage notification settings and customize preferences for query alerts, optimizing workflow efficiency and responsiveness to user queries.

These user stories will serve as the foundation for identifying and prioritizing features and integrations to be implemented in the existing ticketing system.
They align with the SMART (Specific, Measurable, Achievable, Relevant, Time-bound) guidelines and address the needs and expectations of various user groups involved in the support process.