MILESTONE – 3 REPORT

By

Team – 28

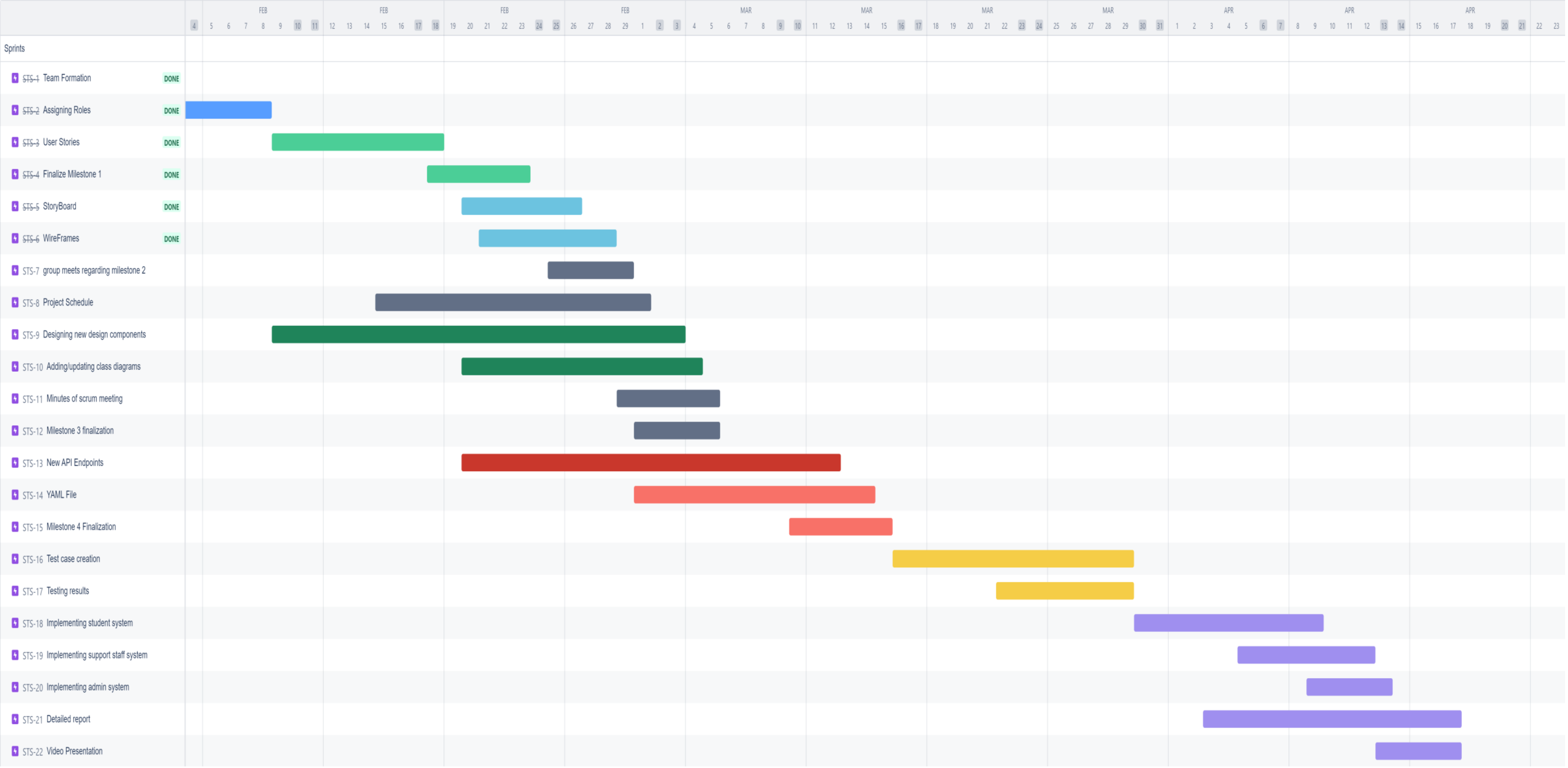
Contents

- ❑ Project Scheduling Tool
- ❑ Gantt Chart
- ❑ Sprint Schedule
- ❑ Components
- ❑ Class Diagram
- ❑ Scrum Meeting Minutes

Project Scheduling Tool

Jira Software

Gantt Chart for Project Schedule



Sprints		
~~STS-1~~	Team Formation	DONE
~~STS-2~~	Assigning Roles	DONE
~~STS-3~~	User Stories	DONE
~~STS-4~~	Finalize Milestone 1	DONE
~~STS-5~~	StoryBoard	DONE
~~STS-6~~	WireFrames	DONE
STS-7	group meets regarding milestone 2	
STS-8	Project Schedule	
STS-9	Designing new design components	
STS-10	Adding/updating class diagrams	
STS-11	Minutes of scrum meeting	
STS-12	Milestone 3 finalization	
STS-13	New API Endpoints	
STS-14	YAML File	
STS-15	Milestone 4 Finalization	
STS-16	Test case creation	
STS-17	Testing results	
STS-18	Implementing student system	
STS-19	Implementing support staff system	
STS-20	Implementing admin system	
STS-21	Detailed report	
STS-22	Video Presentation	

Sprint Schedule

- **Sprint 1: Project Setup and Planning**
  - Date: [1st Feb] - [8th Feb]
  - Tasks: Set up project environment, define project scope, identify team roles and responsibilities, outline sprint goals.
- **Sprint 2: User Story Creation and Prioritization**
  - Date: [9th Feb] - [18th Feb]
  - Tasks: Create user stories based on project requirements, prioritize user stories based on importance and impact.
- **Sprint 3: Wireframing and Design**
  - Date: [20th Feb] - [28th Feb]
  - Tasks: Develop wireframes for user interface, design application layout and navigation.
- **Sprint 4: Backend Development**
  - Date: [29th Feb] - [12th March]
  - Tasks: Develop backend functionality, implement database schema, create API endpoints.
- **Sprint 5: Frontend Development**
  - Date: [1st March] - [14th March]
  - Tasks: Implement frontend components and views, integrate frontend with backend functionality.
- **Sprint 6: Testing and Bug Fixing**
  - Date: [15th March] - [28th March]
  - Tasks: Conduct unit testing, identify and fix bugs and issues, ensure application functionality and stability.
- **Sprint 7: Documentation and Finalization**
  - Date: [1st April] - [17th April]
  - Tasks: Prepare project documentation, finalize reports and presentations, review project for completeness.

Components

- ❑ User Interface
- ❑ Support Ticket Lifecycle
- ❑ Discourse Integration
- ❑ +1 voting System
- ❑ Feedback System
- ❑ Escalation Mechanism

User Interface

- This component encompasses the graphical interface that students, support staff, administrators, and developers interact with to access and manage support tickets.
- It includes features such as ticket creation, viewing ticket status, browsing FAQs, and receiving notifications.
- The UI allows users, including students, support staff, and administrators, to create/resolve a ticket by entering relevant details such as the nature of the issue, its priority level, and any accompanying information.
- Since UI implements role-based access control mechanisms it ensures that users only have access to features and functionalities relevant to their roles.
- The UI is designed to provide an intuitive and user-friendly experience, with clear navigation menus, interactive elements, and responsive design. The goal is to make it easy for users to navigate the system, locate relevant information, and perform tasks efficiently.

Support Ticket Lifecycle

- This component is responsible for handling the creation, tracking, and resolution of support tickets.
- It includes functionality for students to create and manage their tickets, support staff to prioritize and address tickets, and administrators to categorize and archive resolved tickets.
- Once created, tickets are assigned a unique identifier and stored in the system's database. Students can track the status of their tickets in real-time, including updates on whether the ticket is open, in progress, or resolved.
- Support staff members have access to a dashboard or queue where they can view all incoming tickets.
- Admin prioritizes tickets based on factors such as urgency, complexity, and impact on users.
- Once the ticket has been addressed either the student or support staff can mark it “resolved”.

Integrated Discourse

- This component integrates the ticketing system with Discourse, a platform for community discussion and support.
- It allows for the creation of Discourse threads linked to each support ticket, enabling students to follow discussions and solutions related to their queries.
- This components also allows support staff and administrators to moderate Discourse threads created from support tickets by providing options to lock, close, or archive threads based on ticket status or resolution.
- By letting implement features such as likes, comments, and thread subscriptions user engagement is encouraged.
- By tracking metrics related to Discourse thread activity, including views, likes, comments, and resolution rates it is easier to generate reports to analyze user engagement and identify trends in support ticket discussions.

Voting Mechanism

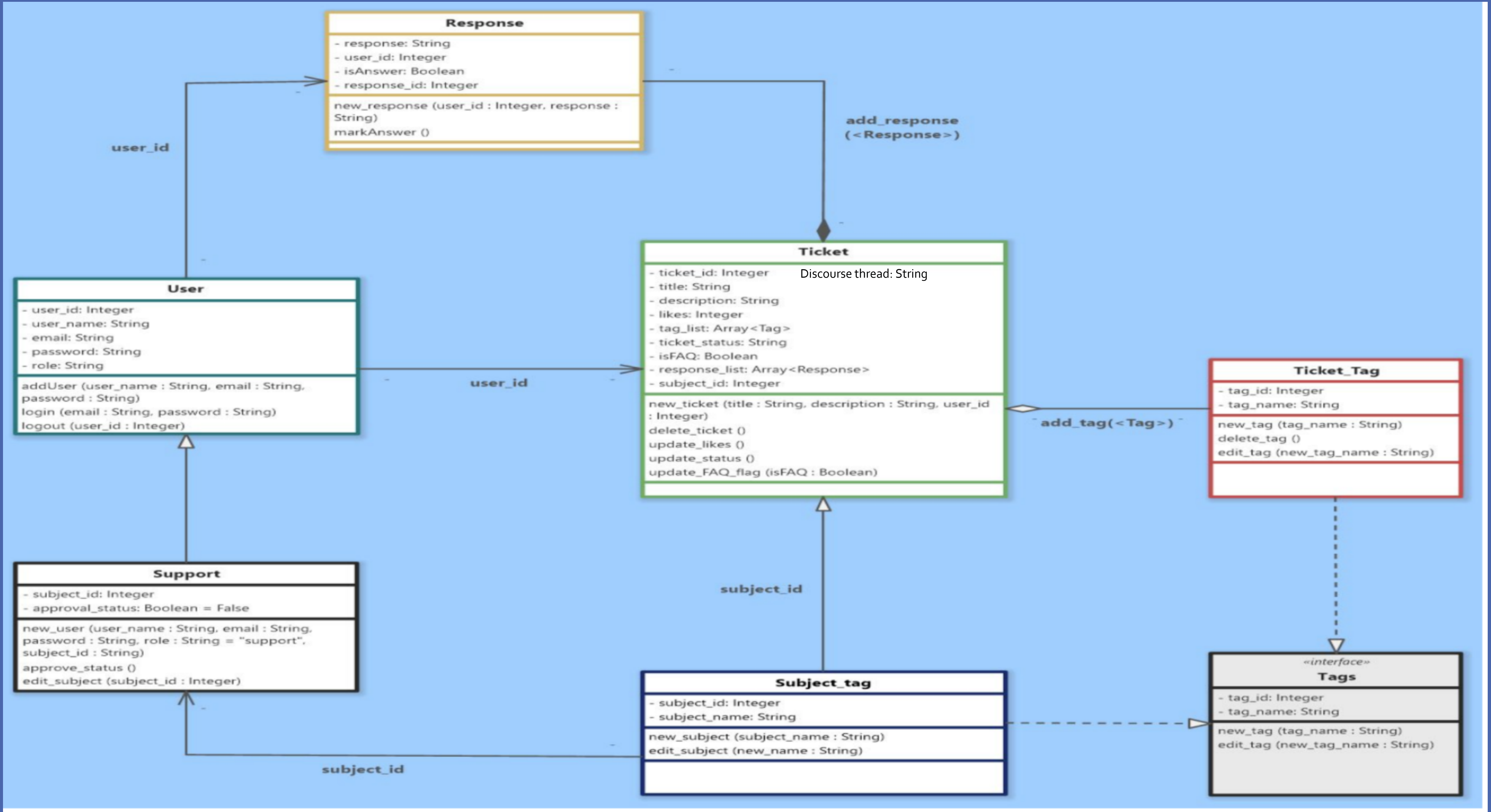
- The +1 voting system enables students to indicate their agreement or interest in existing support tickets that match their concerns.
- It helps avoid duplicate tickets and allows common issues to be prioritized based on popularity.
- Once votes starts to come in thresholds are set for the number of votes required to escalate a ticket's priority.
- Administrators will define rules for how votes influence ticket prioritization and resolution.

Feedback System

- This additional component allows students to provide feedback on their support ticketing experience.
- It includes functionalities for rating the quality of support received, suggesting improvements, and offering general comments or suggestions.
- The feedback mechanism helps in continuously improving the system and enhancing user satisfaction.
- Once the student reviews a ticket response the admin receives it and redirect the feedback to the respective support staff

Escalation Mechanism

- The webhook Integration Component is designed to address high-priority tickets based on predefined criteria such as the nature of the issue, the impact on users, and the urgency of resolution.
- Once an issue is escalated or a ticket has enough votes to consider it a high priority than the admin is alerted and he/she activates webhook for the respective staff.
- Webhook trigger alerts not only for newly identified high-priority tickets but also for tickets that have been escalated due to unresolved issues or escalating customer dissatisfaction.



Response

- response: String
- user_id: Integer
- isAnswer: Boolean
- response_id: Integer

new_response (user_id : Integer, response : String)
markAnswer ()

user_id

add_response (<Response>)

Ticket

- ticket_id: Integer
- title: String
- description: String
- likes: Integer
- tag_list: Array<Tag>
- ticket_status: String
- isFAQ: Boolean
- response_list: Array<Response>
- subject_id: Integer

Discourse thread: String

new_ticket (title : String, description : String, user_id : Integer)
delete_ticket ()
update_likes ()
update_status ()
update_FAQ_flag (isFAQ : Boolean)

User

- user_id: Integer
- user_name: String
- email: String
- password: String
- role: String

addUser (user_name : String, email : String, password : String)
login (email : String, password : String)
logout (user_id : Integer)

user_id

Ticket_Tag

- tag_id: Integer
- tag_name: String

new_tag (tag_name : String)
delete_tag ()
edit_tag (new_tag_name : String)

add_tag(<Tag>)

subject_id

Support

- subject_id: Integer
- approval_status: Boolean = False

new_user (user_name : String, email : String, password : String, role : String = "support", subject_id : String)
approve_status ()
edit_subject (subject_id : Integer)

Subject_tag

- subject_id: Integer
- subject_name: String

new_subject (subject_name : String)
edit_subject (new_name : String)

«interface»
Tags

- tag_id: Integer
- tag_name: String

new_tag (tag_name : String)
edit_tag (new_tag_name : String)

subject_id

Scrum Meeting Minutes

Click Here