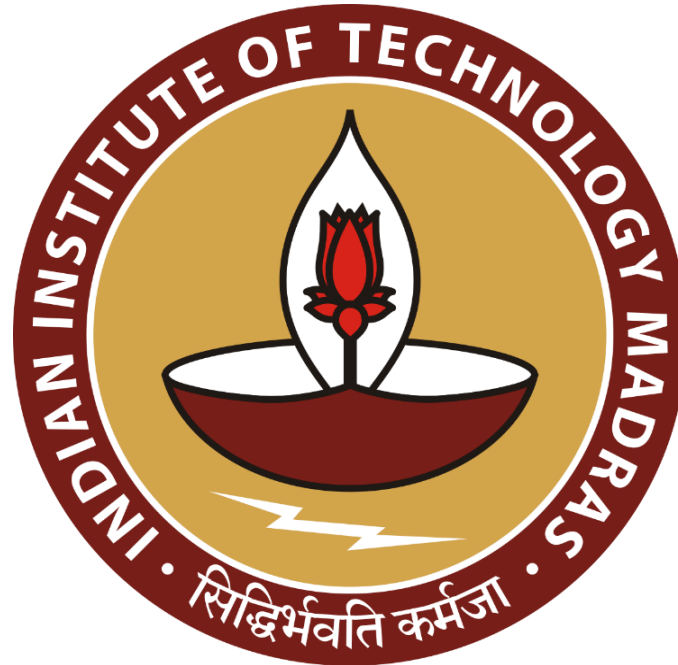


SOFTWARE ENGINEERING

(Course ID: BSCS3001)



Online Ticketing System with Discourse and Gchat Integration

Milestone 3

By

Anupam Kumar Jha (21f1004905)

Kevin Joshua T (22f1001410)

Pranav Wankhedkar (21f1000120)

Sachin Singh (21f1003251)

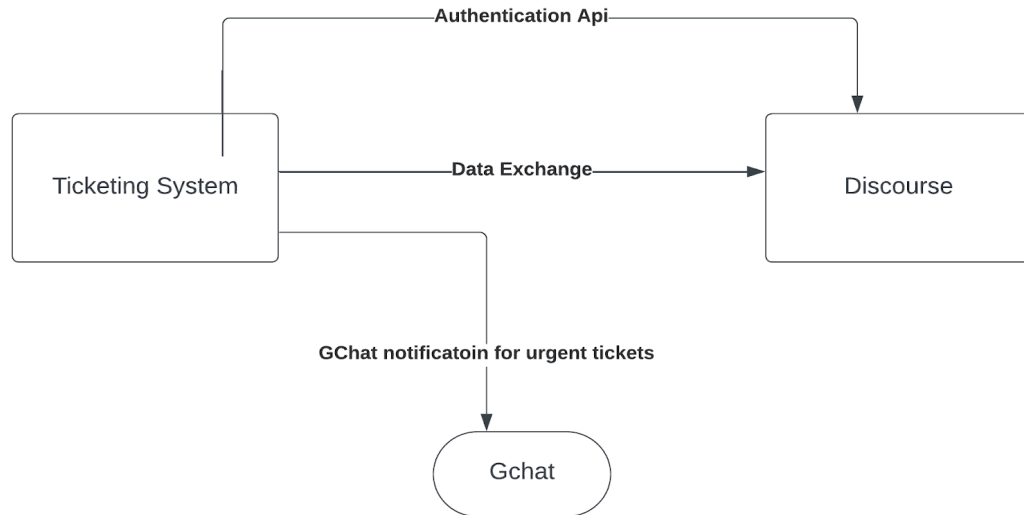
Ujit Kumar (21f3000786)

Utpal Dutta (21f2000524)

Submission Date: 06 Mar. 2024

Design of Components

❖ Overall System Design:



❖ **Authentication Mechanism:** Authentication of users will happen at the ticketing system level. Once the user is authenticated, they can be seamlessly signed in and have access to the discourse portal. Alternatively users can also sign up by registering themselves in the portal.

❖ Student View Enhancements:

- ❑ **Cross-Platform Query Display:** When a ticket is created in the app, it also appears on the discourse platform, ensuring visibility and community engagement. During data exchange process via api Ticketing system shares description, user detail of the ticket owner, severity level. Once the ticket is closed in the ticketing system, the status of the respective thread is also updated in the discourse portal.
- ❑ **Advanced Search Capabilities:** Users can easily search for queries using tags and categories, enhancing the user experience by providing quick access to relevant information.

❖ Support Agent View Enhancements:

- ❑ **Integrated Communication:** Replies made in one platform (ticketing app or discourse) are synchronised across both, streamlining the resolution process.
- ❑ **Priority Notifications:** Urgent or high-priority queries trigger notifications via Google Chat, aiming for rapid responses and reduced resolution times.
- ❑ **Visualisation Tools:** A pie-chart visualisation of pending and resolved queries helps support agents manage their workload more effectively.

❖ **Admin and Manager View Enhancements:**

- ❑ **Security and Oversight:** Admins can monitor for suspicious activities and take action against malicious users, maintaining a safe and secure platform.
- ❑ **Resolution Oversight:** Managers have the capability to investigate unresolved tickets further, ensuring accountability and consistent support quality.
- ❑ **Insight from discourse:** Managers can have course/tag/timeline wise discourse thread counts enabling them to have a complete view of the overall activities, nature and severity of discussions going on in the portal.

❖ **Admin View Refinements:**

- ❑ **Proactive Security Measures:** Admins are equipped with tools to closely monitor the platform for any suspicious activities, ensuring a safe and secure environment for all users. This capability is crucial for maintaining user trust and preventing abuse.
- ❑ **Robust User Management:** The ability to dynamically manage users, including the addition of new users via email IDs and the removal of users based on their username, allows for efficient platform administration. This flexibility is essential for adapting to the evolving needs of the user base.

❖ **Manager View Refinements:**

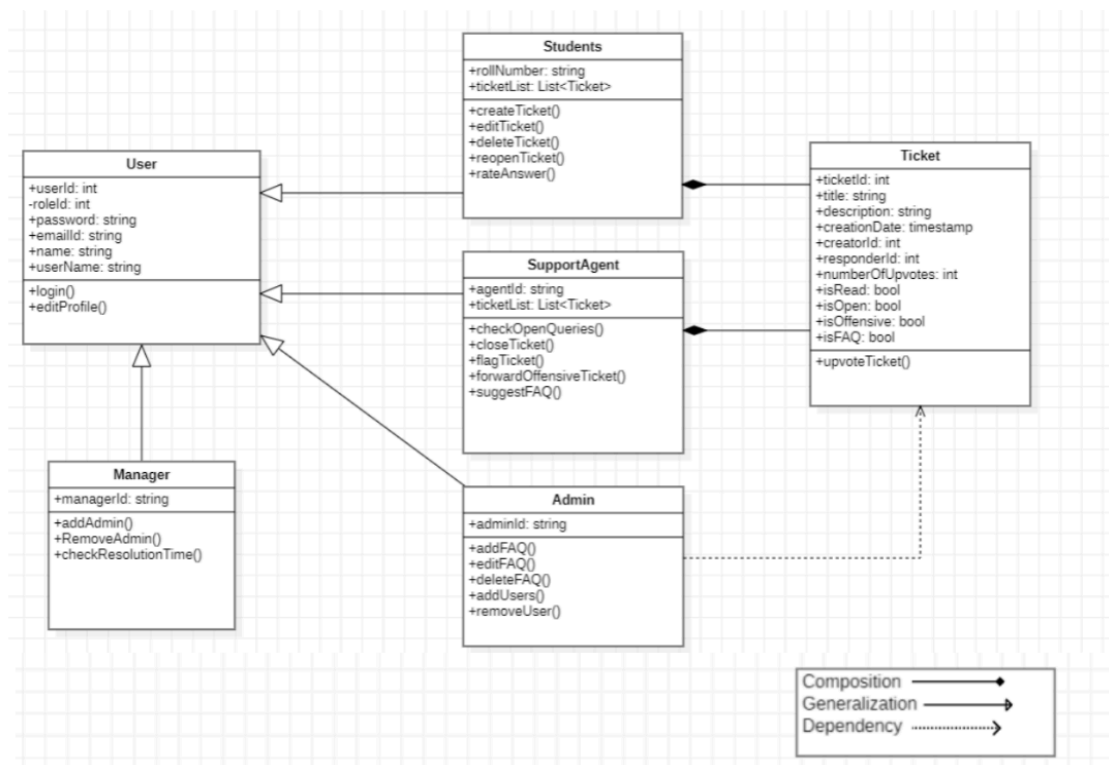
- ❑ **In-depth Ticket Resolution Oversight:** Managers possess enhanced tools to delve into tickets that remain unresolved, providing an additional layer of scrutiny and

ensuring that no query goes unanswered. This function underscores the commitment to high-quality support and customer satisfaction.

- ❑ **Performance Monitoring and Improvement:** Through scheduled cron jobs, managers can identify support agents with resolution times exceeding the desired thresholds, fostering a culture of accountability and continuous improvement. This system helps highlight areas for training and development, ensuring the team's performance aligns with service standards.

Software Design

❖ Class Diagram:



❖ APIs:

<div>Custom Discourse API</div> <div>- Create Post</div> <div>- Update Post</div> <div>- Delete Post</div> <div>- Read Post</div>	<div>Custom Webhook API</div> <div>- Check Post</div> <div>- Message Admin</div> <div>- Log the Request</div>
<div>Student API's</div> <div>- Tag API</div> <div>- Issue Track API</div> <div>- User Profile API</div> <div>- Filter Post API</div>	<div>Support Manager API's</div> <div>- Staff Data API - Analytics</div> <div>- Staff Last Seen API</div> <div>- Filter Post API</div>
	<div>Support Staff API's</div> <div>- Open/Closed Thread</div>

Sprint Schedule

- ❖ **Sprint 1:** Identify different users and write user stories for them.

Deadline: 10/02/24 – 15/02/24

- ❖ **Sprint 2:** Create storyboards for different users and make wireframes for every user keeping the usability design and heuristics in mind.

Deadline: 15/02/24 – 23/02/24

- ❖ **Sprint 3:** Design the software architecture of the app and schedule the implementation of user stories.

Deadline: 29/02/24 – 05/03/24

- ❖ **Sprint 4:** Implementing different user stories and creating API endpoints for them and documenting all the API endpoints in a consolidated YAML file.

Deadline: 06/03/24 – 15/03/24

- ❖ **Sprint 5:** Design test cases and suits to do unit, integration, system and acceptance testing.

Deadline: 16/03/24 – 28/03/24

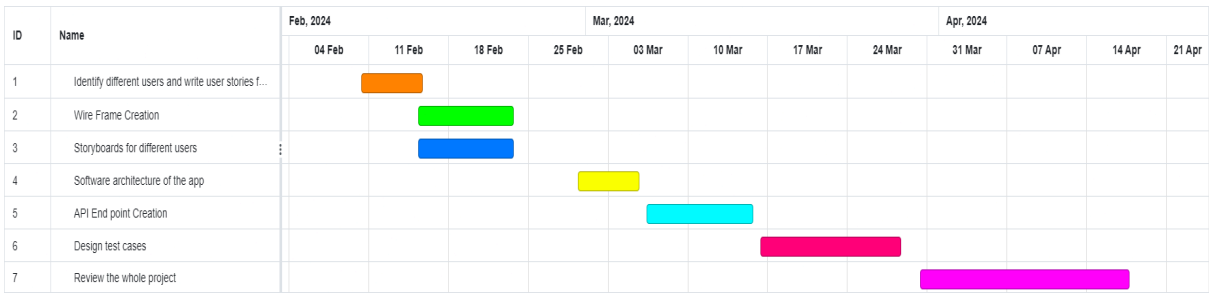
- ❖ **Sprint 6:** Review the whole project, find any bugs and fix them, creating a consolidated report on the whole project and making a presentation video of the app.

Deadline: 30/03/24 – 17/04/24

Minutes of Scrum Meetings

- ❖ **Scrum Meeting 1 (09/02/24):** This was an introductory meeting. Team members gathered and got to know each other and discussed how the team would proceed for the project.
- ❖ **Scrum Meeting 2 (13/02/24):** Team creates a roadmap to complete milestone 1 after discussing the problem statement thoroughly. We also identified different users of the app (Primary, Secondary, Tertiary). Everyone agreed to think and write user stories offline on a consolidated sheet.
- ❖ **Scrum Meeting 3 (15/02/24):** Team creates a roadmap to complete milestone 2. Different tasks have been assigned to different members. Team has discussed the structure of the storyboard and wireframe.
- ❖ **Scrum Meeting 6 (04/03/24):** Team creates a roadmap to complete milestone 3. We identified a tool to manage our project in our case is Trello. Different team members have been assigned for different tasks of milestone 3.

❖ **Gantt Chart:**



❖ **Trello Board:**

