Software Engineering Project: Milestone 3

Project report submitted to
Indian Institute of Technology, Madras
In partial fulfilment of the requirements for the course

BSCSS3001: Software Engineering

by

Tushar Supe (21F1003637)
Vaidehi Agarwal (21F1003880)

Online BSc in Programming and Data Science
Indian Institute of Technology
Madras 600 036 (India)
2023

PROBLEM STATEMENT

Title: Online support ticket system for the IITM BSc degree program

Description:

The support team at the IITM BSc degree program often get overwhelmed with emails from students regarding queries and concerns. Your task is to create an online support ticketing system for the IITM BSc degree program. Students can create a support ticket for a particular concern or query. Before they create a ticket, the system should also show a list of similar tickets, and allow users to like or +1 an already existing support ticket, so that duplicates are not created. This way popular concerns or queries can be prioritised by the support team.

After the support team addresses the concern, they can mark the ticket as resolved, and an appropriate notification should be sent to concerned users. Another important feature of the ticketing system is dynamic FAQ updation. Many student concerns can be FAQs which will be useful for future students. If appropriate, the support query and response should be added to the FAQ section by support admins, and appropriately categorised, so that an updated FAQ will be readily available to students. The platform should allow users to enrol as students, support staff and admins. Apart from these standard requirements, you can also think of other features which can add value to users.

INDEX

MILESTONE 3:
SCHEDULING AND DESIGN

1.1 Project Schedule

1.1.1 Task Distribution

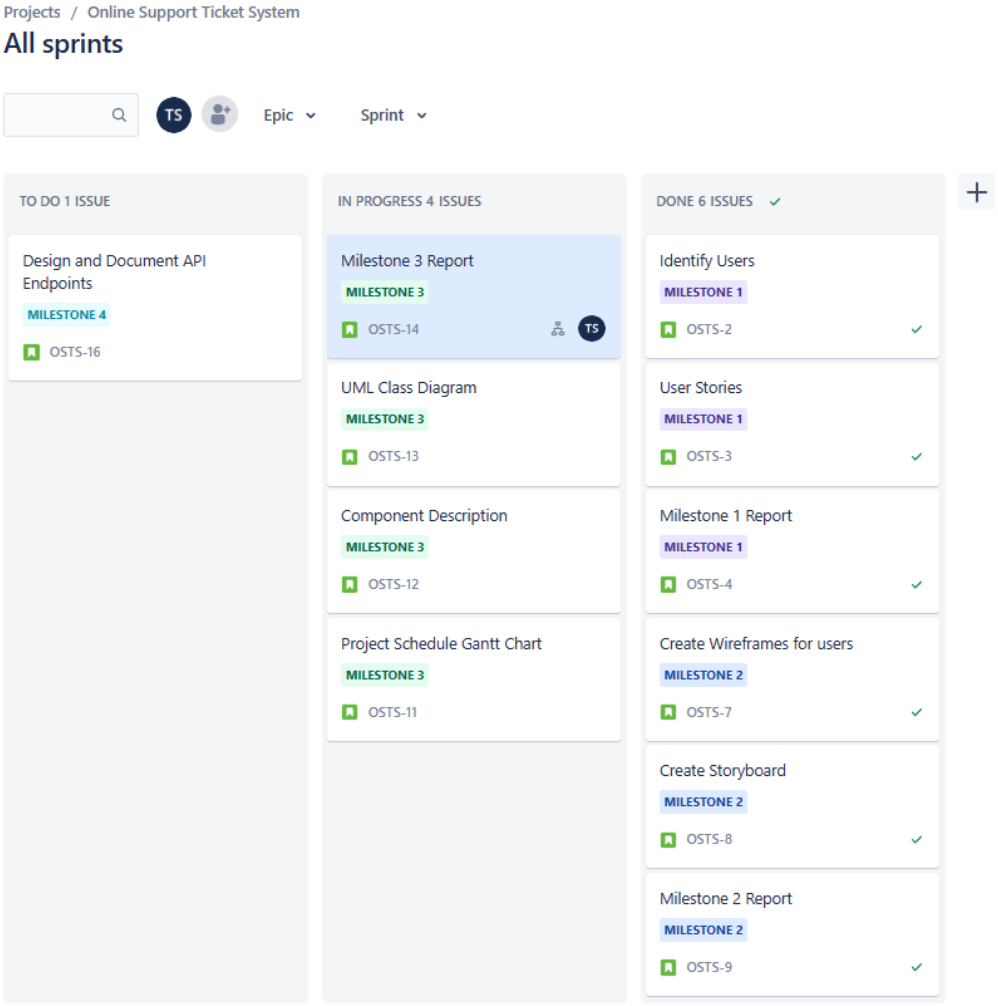
Milestone	Sub-Tasks	Sprint	Assigned To
1 – User Requirements	Identify Users	1	Tushar
	User Stories	1	Vaidehi
	Report	2	Tushar
2 – User Interfaces	Wireframe	3	Tushar
	Storyboard	4	Vaidehi
	Report	4	Tushar
3 – Scheduling	Project Schedule	5	Tushar
	Component Design	5	Vaidehi
	Class Diagram	6	Tushar
	Report	6	Tushar, Vaidehi
4 – API	Design API	7	Tushar, Vaidehi
	Code Review	8	Tushar, Vaidehi
	YAML Document	9	Tushar, Vaidehi
5 – Testing	Test Cases Design	10	Tushar
	Unit Testing	11	Vaidehi
	Report	11	Tushar, Vaidehi
6 – Submission	Frontend Design	12	Tushar, Vaidehi
	Demo	13	Tushar, Vaidehi
	Report	13	Tushar, Vaidehi
	Presentation	14	Tushar, Vaidehi

Each milestone is divided into sub tasks with SMART guidelines and assigned to each of the team member evenly. The sprints are schedules such that the dependency of each components is satisfied. The Gantt-Chart for the project schedule is shown as below. For the high resolution image of the chart, please click here

[1.1] Gantt Chart of Schedule

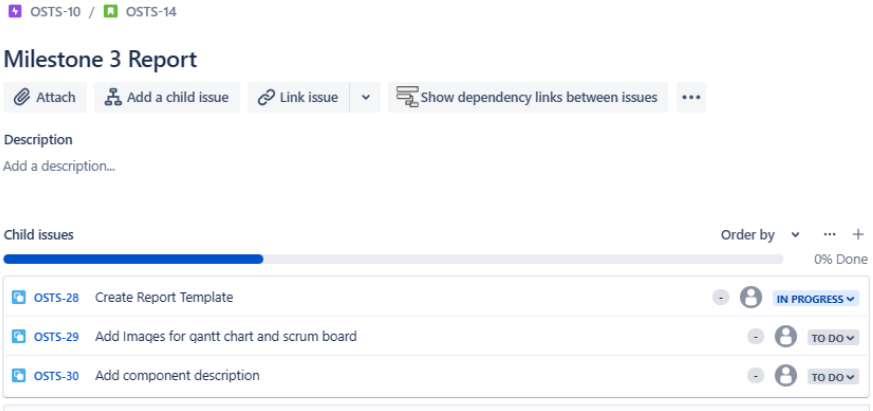
The scrum board consists of ‘To Do’, ‘In Progress’ and ‘Done’ for the active sprint. The scrum board for the milestones till now (completed and in progress) is shown below.

[1.2] Scrum Board (Till Milestone 3)



Each sub-task is divided into even smaller tasks when they are added to ‘In Progress’ tab. A sample is shown below.

[1.3] Dividing Sub-Tasks



1.2 Scrum Meetings

The sprint schedule is shown in Gantt-Chart. ‘Jira’ is used for scheduling the project. The scrum meetings are scheduled at the start of every sub-task at 9 PM to 10 PM (preferably on Saturday or Sunday). The details of few scrum meeting are summarised below.

Scrum Meeting for User Identification

Agenda	Discussion
• Discuss project overview • Understand problem statement • Identify users • Distribute user stories	During the meeting, we discussed the project problem statement. We listed down exact requirements for the project as per the problem statement. Then we discussed the potential users and categorised them into different types. Then we divided the tasks. Tushar was tasked with generating report and the user stories for the admin. Vaidehi was tasked with user stories for student and support staff.

Scrum Meeting for User Interfaces

Agenda	Discussion
• Discuss what wireframes and storyboard means. • Create basic layout as a template for all wireframes • Discuss different options in wireframe. • Distribute tasks	During the meeting, we discussed the wireframe structure. Tushar was tasked with wireframes generation using ‘miro’ and Vaidehi was tasked with storyboarding. We discussed different options that should be placed in wireframes and how it will be connected to each other.

Scrum Meeting for Project Schedule

Agenda	Discussion
• Discuss major milestones • Divide milestones into sub tasks with SMART guidelines. • Decide feasibility and deadlines. • Discuss major components for the project. • Discuss what is class diagram. • Distribute tasks	During the meeting, we discussed major milestones and divided into sub tasks. The deadlines were set after assigning each task to a person and a sprint. The class diagram and Gantt chart was given to Tushar. Vaidehi decide to work on components description.

1.3 Components of the Project

The project is divided into 6 major components:

1. Student view - includes API and Frontend
2. Support view - includes API and Frontend
3. Admin view - includes API and Frontend
4. Ticket CRUD API
5. FAQ CRUD API
6. Authorization – includes user validation during login and register.

1.3.1 Components Description

The components and the short description of sub components is summarised as below.

Auth Components

- Login page (html template + Vue setup + bootstrap styling) (User can login using through email id and password)
- Register page (html template + Vue setup + bootstrap styling) (User need to fill in first and last name + unique email id + password + profile_photo (optional))
- Frontend and Backend data validation for login/register (including validation of punctuations or any other symbol that can breach the security)
- Frontend store JWT and delete when expired (securing software with JWT)
- Backend Create JWT and verify for each request (Creating and verifying new and old requests)
- AuthAPI to handle login/register/logout (authenticated APIs to handle login + register + logout)
- Validate new users registrations (sending information to the admin to accept the new user)
- Methods to send notifications (google chat webhook, email) (Once admin accept or decline the new user, sending notification about their status)

Ticket Components

- CRUD operations with Ticket API (Authenticated APIs for CRUD on Tickets)
- Multiple tickets request with API (Authenticated APIs for GET requests)
- Set up Cache for ticket request (Caching for data retrieval efficiency)
- Create Ticket page (html template + Vue setup + bootstrap styling) (User need to add title + description + priority + tags + add attachment which is optional)
- My tickets page (html template + Vue setup + bootstrap styling) (User can view all tickets or can use filters)

Student Components

- Home page (html template + Vue setup + bootstrap styling) (User can view unsolved tickets and their activity on the software)
- Update profile page (html template + Vue setup + bootstrap styling) (User can change their details except email)
- Search, sort, filter tickets for frontend (User can search, sort, filter to see specific tickets)
- Student API backend (Fetching Data and CRUD operations are implemented using authenticated APIs)

Support Staff Components

- Home page (html template + Vue setup + bootstrap styling) (User can see all the unresolved tickets and their activity + can sort and filter)
- My resolved tickets page (html template + Vue setup + bootstrap styling) (User can see all the tickets that they have solved)
- Resolve Ticket page (html template + Vue setup + bootstrap styling) (User need to add solution + attachment, if required and solve the ticket)
- Update profile and change password page (User can change their details except email)
- Backend API for support staff (Fetching Data and CRUD operations are implemented using authenticated APIs)

Admin Components

- Home page (User can see details of all the students, tickets, support staff and admins)
- Validate users page (User can accept or decline the request of new student or support staff)
- View most upvoted tickets (User can see most upvoted tickets and create FAQ)
- Update profile and change password page (User can change their details except email)
- AdminAPI for backend (Fetching Data and CRUD operations are implemented using authenticated APIs)

FAQ Components

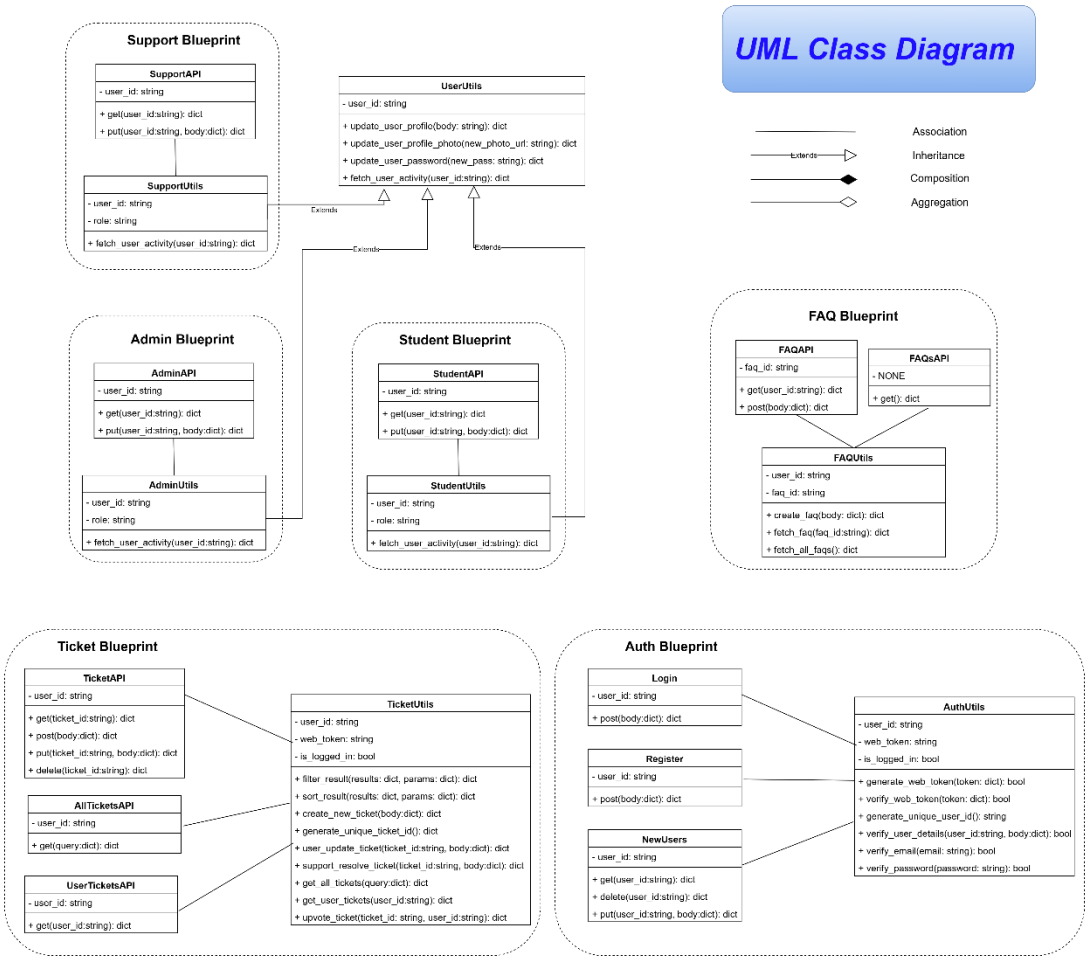
- View all FAQ page (html template + Vue setup + bootstrap styling) (See all the FAQs till now)
- Create FAQ page (html template + Vue setup + bootstrap styling) (Create FAQs for the most upvoted tickets)
- FAQ API for backend (Fetching Data and CRUD operations are implemented using authenticated APIs)

1.4 Class Diagram

Based on the above 6 major components, the UML Class diagram is created in diagrams.io tool. The components are grouped together as a blueprint. For example – *StudentAPI* (contains API end point methods like get, post, put, delete), *StudentUtils* (contains all supporting functions for student API) is grouped together as '*Student Blueprint*'.

The class diagram is shown below. For the high resolution image of the chart, please click here.

[1.4] Class Diagram



REFERENCES

[1] Software Engineering Project: Problem Statement