

SOFTWARE ENGINEERING

Final Report

By:

Pankaj Bora (21f1000210)

Arvind Sankar (21f1002061)

Anubhav Lohani (21f1003415)

The following report was prepared and submitted to meet the requirements of the course Software Engineering (BSCCS3001) offered as part of the BS Online degree program by Indian Institute of Technology, Madras.

Introduction

The following is a report containing details on how milestones were met by applying the principles, practices and methodologies of software development discussed in the course.

Detailed report on work done from Milestone 1 through Milestone 5.

- **Milestone 1** essentially involved identifying types of users and generating user stories of identified users. The link to the report can be found [here](#).
- In **Milestone 2**, storyboards were created along with low-fidelity wireframe prototypes to create a roadmap for implementing the applications to address the identified user stories. The storyboard and the wireframes prototypes can be found [here](#).

-
- For **Milestone 3**, a project management schedule was created to deconstruct the application into several design components, and establish a timeline for completion of the projection. The report can be found [here](#).
 - For **Milestone 4**, API endpoints were planned using swagger. The YML file can be accessed [here](#).
 - For **Milestone 5**, several test cases were created using pytest with an aim to proceed with a test driven development for the development of the application. The report for the test cases can be accessed [here](#).

Each of the milestone reports has been attached in Appendix A.

Implementation details of your project

Technologies and tools used.

- Flask
- Vue

Where is your application hosted(if hosted)?

- *NA/run locally*

And instructions to run your application.

- *Make sure you are in: ./Milestone-6-Final-Submission/code/backend. Run the following commands:*
 - *pip install -r requirements.txt*
 - *python main.py*
- *Make sure you are in: ./Milestone-6-Final-Submission/code/frontend. Run the following commands(IMPORTANT: need node v16.18.0):*
 - *npm install*
 - *npm run dev*

Code Review: Issue Reporting and Tracking

The code review process was done using Github and the issue reporting and tracking was done using Github Issues. Code was reviewed manually to find and fix bugs, improve code quality, and ensure that the code is maintainable and easy to understand. It was mostly done during periodic meetings and after each batch of commits corresponding to a component. While several issues were raised, tracked and resolved during the development process, the following screenshot shows an example how one issue was tracked and resolved:

Potential security issue in positive routes #3

 Open arvindsankariitm opened this issue 36 minutes ago · 1 comment

 arvindsankariitm commented 36 minutes ago  Tip ...

The mark_duplicate route (and a few others) in our Flask application allows any user to mark any ticket as a duplicate when only the user who created the ticket should be able to mark it as such. This poses a security vulnerability as it could potentially allow users to mark tickets created by other users as solved.

We need to update this route to ensure that only the user who created the ticket can mark it as a duplicate. We should also consider implementing additional security measures such as input validation to prevent any malicious input from users.

  arvindsankariitm self-assigned this 36 minutes ago

 coderxpb commented 14 minutes ago  Tip ...

yes, please implement that. i've also changed some of the routes to no longer accept user_id from the api requests but use the authentication to get user_id of the current user. this particular issue should be resolved by checking if current_user.id == ticket.user_id

Note: Due to the small size of the team, many issues were raised and resolved using direct messaging applications and video conferencing tools to improve speed of development. At the same time, deliberate efforts were made to learn and use issue tracking tools to prepare ourselves for settings where direct messaging may not be practical.

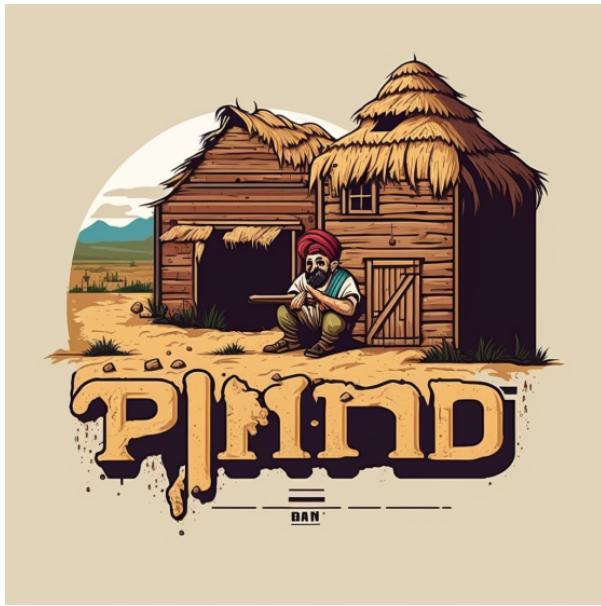
Conclusion

This report has provided a detailed overview of the milestone reports and the implementation details of the project. Each of the milestones were met while keeping in mind the best software development principles and practices. Furthermore, emphasis was given to applying and correlating various concepts and strategies involved in software development as discussed in the course.

The project began with an abstract ideation phase followed by the identification of users and user stories. Low-fidelity wireframes and storyboards were created to create a roadmap for the development of the application. A project management schedule was also created to break down the application into several design components and to establish a timeline for completion of the project. Further, API endpoints were planned and several test cases were created to ensure the quality of the application through a test driven development process.

These phases were followed by the actualizing of the application wherein the backend was developed using Flask and the frontend was developed using Vue.

The project also involved the code review process, i.e. essentially issue reporting and tracking process during the development of the application. The application went through several iterations in order to ensure that all the user stories were implemented and the application was of high quality. The project was successfully completed with the help of the principles, practices and methodologies of software development discussed in the course.



SOFTWARE ENGINEERING
MILESTONE 1

Introduction

The Milestone 1 essentially involved identifying types of users and generating user stories of identified users. The subsequent pages elaborate on the same.

Users

Primary Users:

- Service provider/IITM BS admin and support team
- Service users/enrolled students
- Future students

Secondary Users:

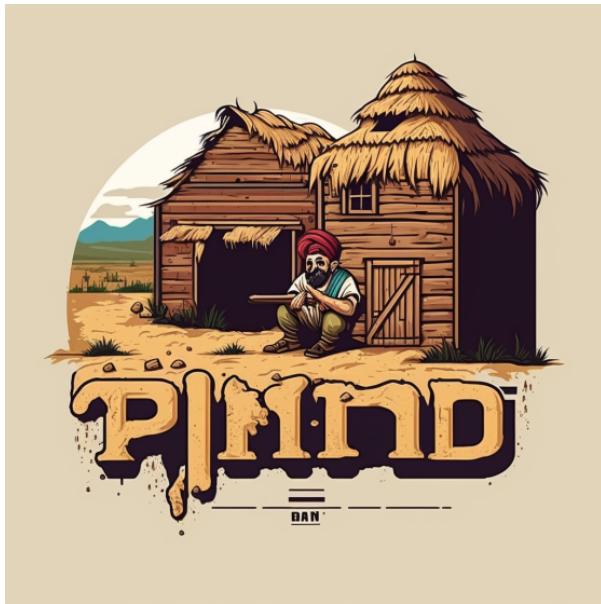
- IITM BS Faculty and TAs
- Prof. Andrew, program coordinators

Tertiary users:

- Directors
 - The Senate & Governing Board
-

User Stories

1. As a support staff, I want to see the most liked queries, so that those could be resolved first.
2. As a student, I want to add a like or +1 to the queries raised by other students, so that I and other students can quantify how urgent the complaint is.
3. As a future student, I want to see a list of FAQs, so that I could resolve my queries without the need to email support for every small detail.
4. As a support staff, I want to track unique concerns and queries raised by students, so that said concerns can be prioritized.
5. As a support staff, I want to be able to resolve queries by linking already resolved queries that are very similar, so that I can reduce duplication of answers and effort.
6. As a student, I want to escalate my query after 7 days have passed without any response, so that my query would be answered on time.
7. As a student, I want to filter the FAQs by categories such as Admissions, Payment etc., so that I only need to browse through what's relevant to me.
8. As the support team, I want a notification to be sent to students whenever a ticket is resolved, so that all students are informed of the resolution of the issue.
9. As a student, I want to receive a notification whenever a ticket is resolved, so that I am informed that my issue is resolved.
10. As a student, I want to be able to search all queries, so that I can find whether my query was raised /resolved previously.



SOFTWARE ENGINEERING
MILESTONE 2

The following milestone involved the creation of a storyboard and a wireframe prototype to form the application.

The storyboard was created to capture the user journey and provide an overview of the journey/experience of the users. The narrative formed therein provided insights on the pain points of the users and potential solutions that may address the same. The wireframe prototype was used to validate the user journey and to provide a visual representation of the tentative user experience. Both the storyboard and the wireframe were created based on the user stories identified in milestone 1.

Storyboard

At student end

It has been two weeks and this is the
third time I've reminded.
My issue still hasn't been resolved.



At another student's end

This doesn't solve my
issue!

Dear student,
Abcd
.....

This is not the
issue

Reply



Hmm, can't seem to find if I'm eligible or not
on the website. Let me email them.



At yet another student end

Portal seems to be down.
I should email support.



Portal seems to be down. I should email support!



Portal seems to be down. I should email support!!



At IITM HQ

I'm receiving 20-30 complaints daily about unresponsive support team. The programs rating has gone down to 3 stars and people are more hesitant to enroll.



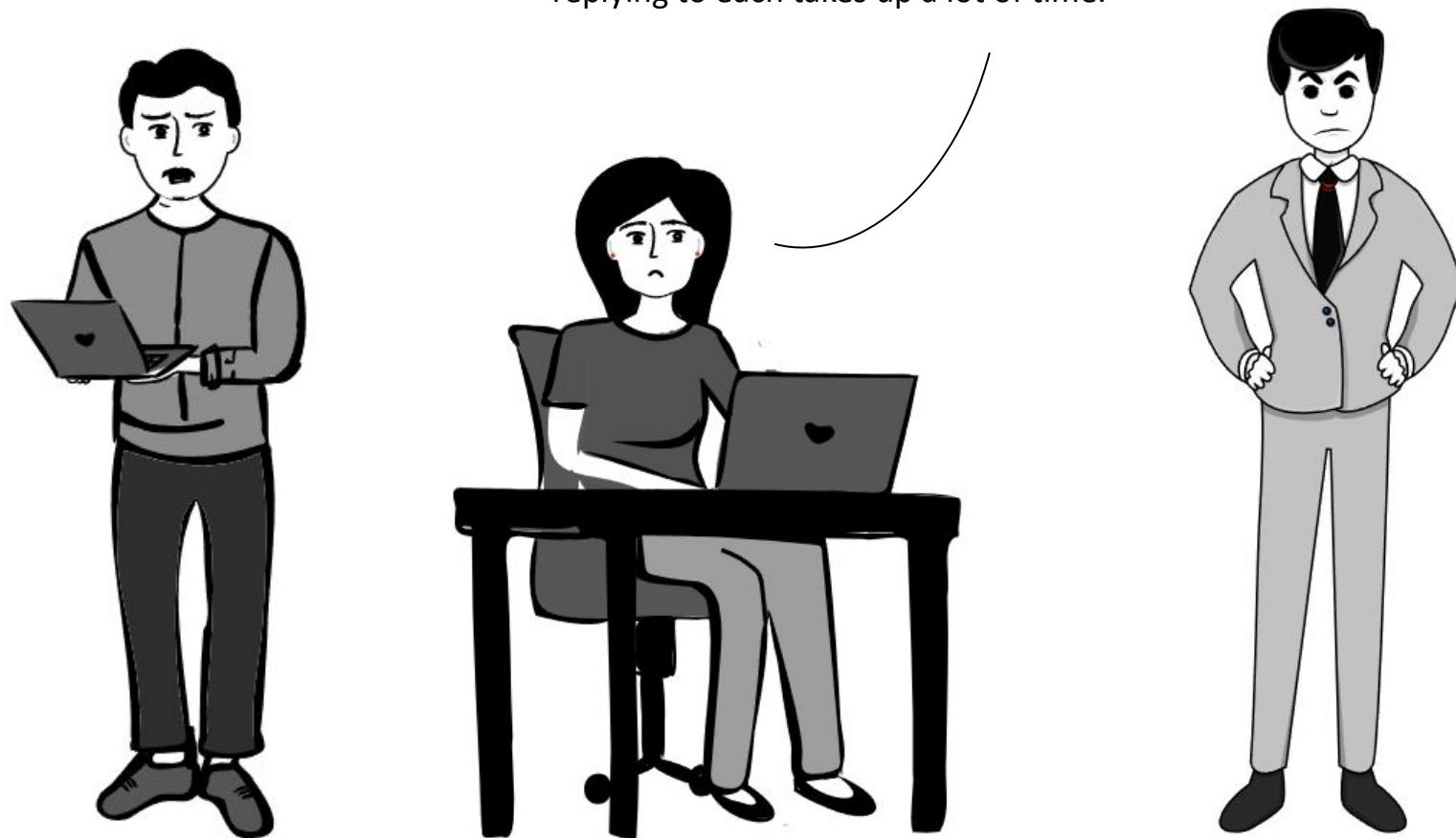
At IITM HQ

Sir, you're getting 20-30 complaints
while we're getting 200-300 emails
daily.



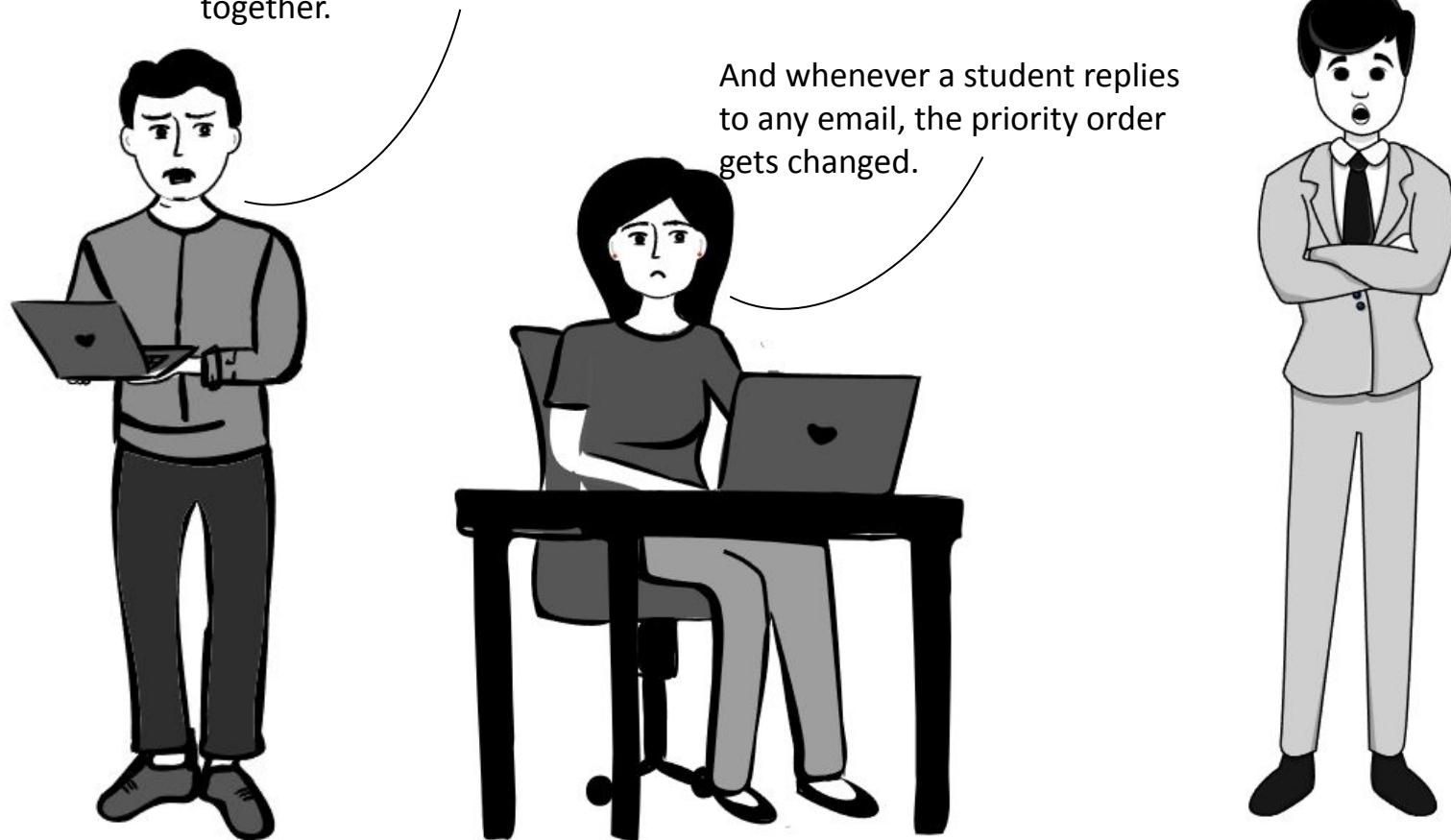
At IITM HQ

A lot of students also raise same issue and replying to each takes up a lot of time.



At IITM HQ

And since there's no proper way to organize or categorize queries, we can't solve or raise them together.



And whenever a student replies to any email, the priority order gets changed.

At IITM HQ

Very well. Given the current state, the board had already decided to give PIND a try. Let's see if it is helpful for us.



At IITM HQ

Let me look at all the admission related queries first.



Raised to: Course team

XYZ
XYZ
XYZ

Likes: ###

View Details Resolve

osam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui

A screenshot of a digital interface for managing admission-related queries. At the top right are dropdown menus for 'Filter' (set to 'Payment') and 'Most Liked'. Below them is a list of three items, each with the identifier 'XYZ'. To the right of the list is a placeholder 'Likes: ###' with a cursor hovering over it. Below the list is a large text block containing a Latin passage. At the bottom are two buttons: 'View Details' (blue) and 'Resolve' (green). The bottom half of the screen is a large, light gray area with a faint 'X' watermark.

The most liked query seems to be about the recent eligibility criteria changes.



Ticket Number: #####
Raised to: Course team

User Name

Title: Issue on Xyz issye

Issue: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.
Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.
Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.
Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Relied/pending Likes: ###

[Reply](#) [Mark as Duplicate](#)

At IITM HQ

responds



At coffeeville

PIND



At coffeeville



Let's go! Looks like I'm eligible.



At IITM HQ

Hmm, this query has already been solved Yesterday.

Home Posts FAQs Profile

Logout

Ticket Number: #####

Raised to: Course team

User Name

Title: Issue on Xyz issye

Issue: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Relied/pending

Likes: ###

Reply

Mark as Duplicate



Admin Name

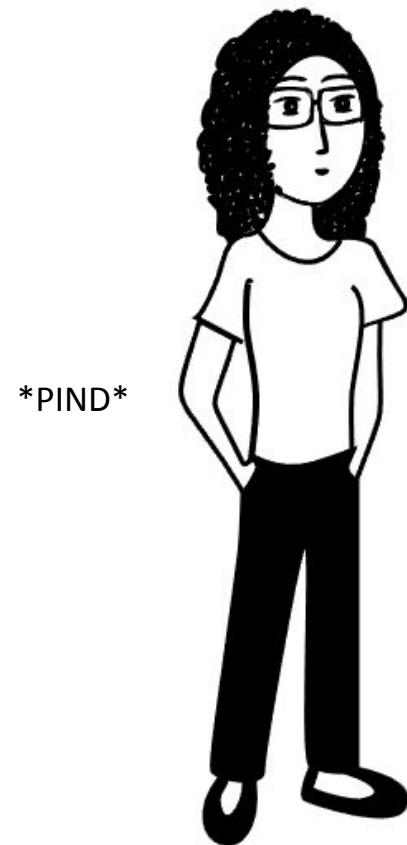


At IITM HQ

Let me find the resolved ticket and link
it as solution.



*le student





Oh ok, this makes
sense.

Ticket Number: #####

User details

Title: Issue on Xyz issye

Issue: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Replied/pending

Like [number]



Admin details

Solution: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Mark as Solved

Reply



I also faced an issue with payment, let me search whether others have raise similar query.

Title: *Payment issue* 

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut o Nemo enim ipsam voluptatem quia voluptas sit aspernatur a...
Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut o Nemo enim ipsam voluptatem quia voluptas sit aspernatur a...
Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut o Nemo enim ipsam voluptatem quia voluptas sit aspernatur a...

[Show More](#)

Tags: [Admin support](#) [Payment](#) [Course](#) [Misc](#)



Oh ok, this is the same doubt that I had. The solution makes sense. I'll upvote it.

Ticket Number: #####

User details

Title: Issue on Xyz issye

Issue: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Relied/pending

Like [number]





No, I don't think this is correct, let me ask for more clarification.

Like [number]

Admin details

Solution: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Mark as Solved

Reply

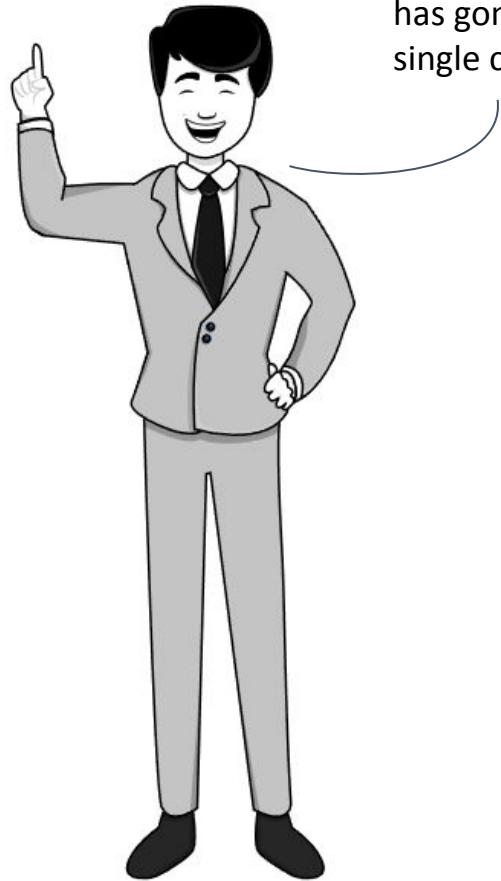


At IITM HQ

click clack *click clack*



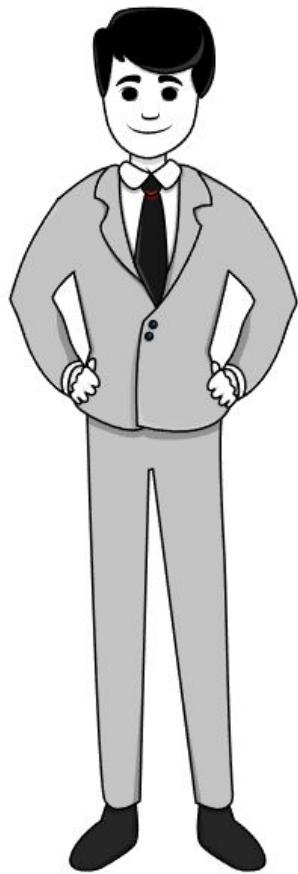
At IITM HQ



Good job, you two. The rating of the program has gone up to 4.5 stars and I haven't received a single complaint in a week.



At IITM HQ

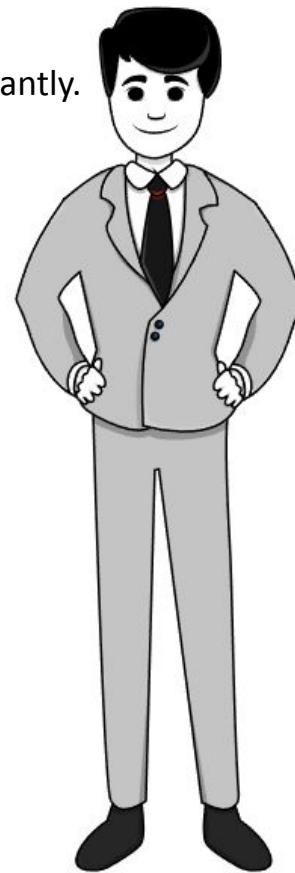


Haha, no problem sir, actually our work has gotten a lot simpler and easier with PIND.



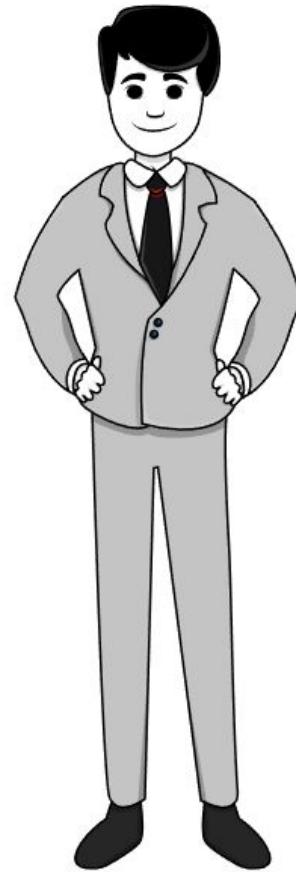
At IITM HQ

Yes, duplicate queries and commonly Asked questions have reduced significantly.

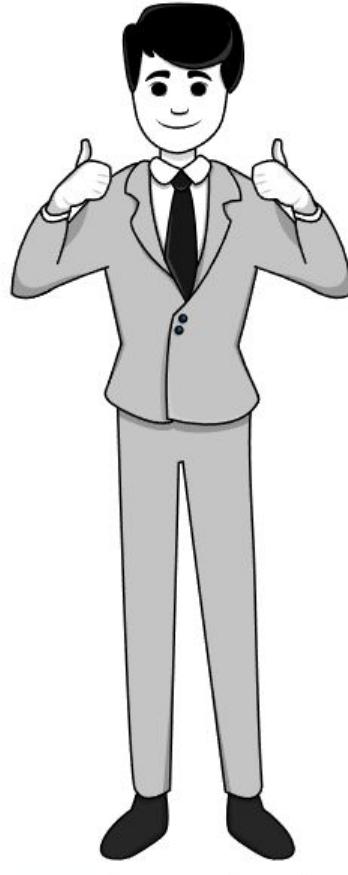


At IITM HQ

Even when we do get duplicate queries it only takes a few seconds to resolve them now.



At IITM HQ



Not only that but overall organisation and communication is much faster now.



Wireframe Prototype

Register to Pindtopia

Sign up for a free account

First name

Last name

Student

Admin

Email address

Create password

Register

Client Side

Posts

Home Posts FAQs Profile Logout

Posts

FilterRaise ticket

Ticket Number: #####Raised to: Course teamUser ID: #####Likes: ###

Title: Issue on Xyz issye

Details: Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Relied/pending View Details Like [number]

Ticket Number: #####Raised to: Course teamUser ID: #####

Title: Issue on Xyz issye

Details: Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Relied/pending View Details Like [number]

Show more

Ticket Number: #####

User details

Title: Issue on Xyz issye

Issue: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Replied/pending

[Like \[number\]](#)



Admin details

Solution: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

[Mark as Solved](#)

[Reply](#)

FAQs

Question: Issue on Xyz issye

Details: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Answer: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Question: Issue on Xyz issye

Details: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Answer: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Question: Issue on Xyz issye

Details: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Answer: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

[Show more](#)

Ticket ID: #####

Title: Enter title here

Details: Enter details here

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Tags:

[Admin support](#)

[Payment](#)

[Course](#)

[Misc](#)

[Submit ticket](#)

Ticket ID: #####

Title: Payment issue

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut o Nemo enim ipsam voluptatem quia voluptas sit aspernatur a...

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut o Nemo enim ipsam voluptatem quia voluptas sit aspernatur a...

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut o Nemo enim ipsam voluptatem quia voluptas sit aspernatur a...

Show More

Tags:

Admin support

Payment

Course

Misc

Submit ticket

Support Side

Open Tickets

[Filter](#)[Most Liked](#)

Ticket Number: #####

Raised to: Course team

XYZ

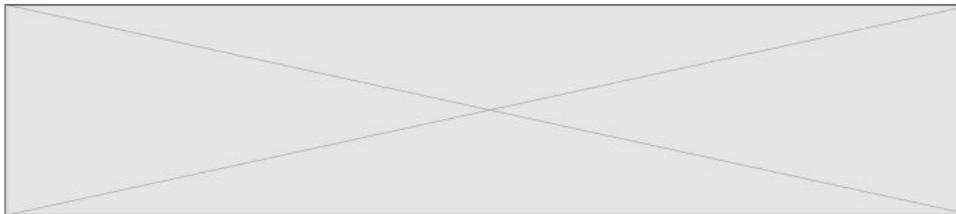
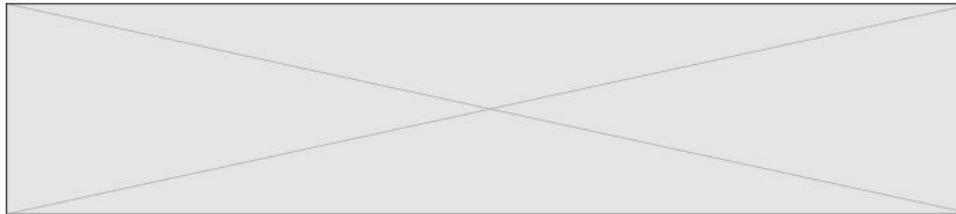
XYZ

XYZ

Likes: ###

Title: Issue on Xyz issue

Details: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

[View Details](#)[Resolve](#)[Show more](#)

Ticket Number: #####

Raised to: Course team



User Name

Title: Issue on Xyz issye

Issue: Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Replied/pending

Likes: ###

[Reply](#)[Mark as Duplicate](#)

Admin Name

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

[Reply](#)

Raised to: Course team

Ticket Number: #####



User Name

Title: Issue on Xyz issye

Issue: Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Status: Addressed/Replied/pending

Likes: ###

[Reply](#)[Mark as Duplicate](#)

Search for original post

Admin Name

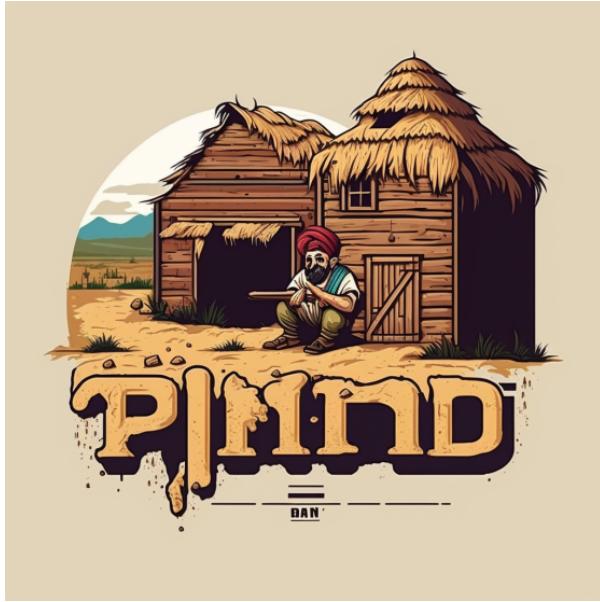
Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

Nemo enim ipsum voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem.

[Reply](#)

Link to [Eigma](#)



SOFTWARE ENGINEERING
MILESTONE 3

Introduction

The following is a report comprising PIND's efforts in identifying software design requirements and laying down a road map with a tentative timeline for the completion of each of the identified requirements. The following sections detail the efforts taken for the completion of Milestones 3 and beyond.

Component Identification

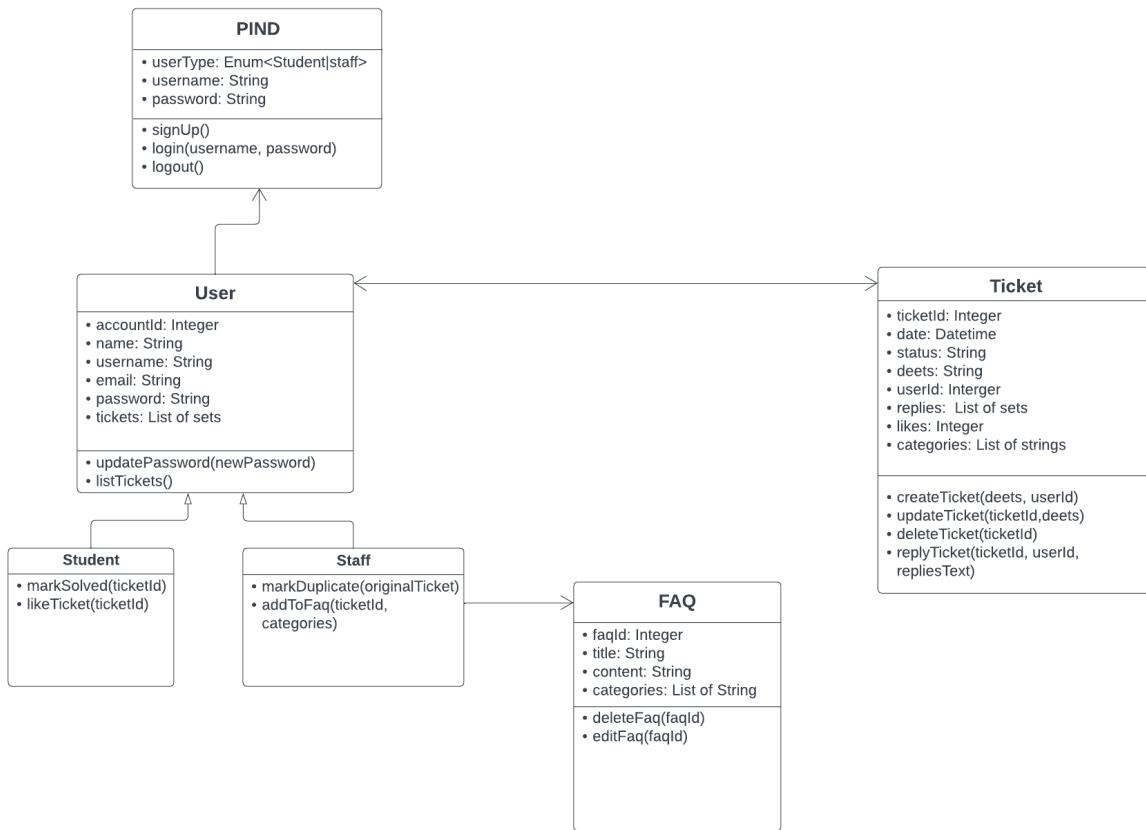
A meeting was held to discuss and identify the core components of the project based on software design principles. Essentially, 3 components were found to be the most critical for the development of the ticket management application, viz. Ticket management component, user management component, and Frequently Asked Questions (FAQs) component.

User management component would allow the ticket management system/application to create and manage user accounts. The user management component may only allow certain operations to be performed only by authorized users. Ticket management component is the core of the system. In addition to CRUD operations on tickets it allows

users to do additional operations on tickets such as indicate urgency of an issue, converting tickets to FAQ, filtering and sorting them. The FAQs component provides an interface for displaying frequently asked questions as categorized by the support staff.

Class Diagrams

Further, the team discussed and prepared a tentative class diagram, specifically outlining the attributes and methods associated with each class, as well as their relationships with other classes. The class diagram was prepared with the intention of modularizing the development process for the application proposed to be built. It was agreed that the identified components and classes would be subject to iterative modifications for providing flexibility during the development process.



Project scheduling

Notion was chosen as a tool for project scheduling. The identified components and the classes were further deconstructed into specific, measurable, achievable, relevant, and time-bound tasks and assigned to each member of the team. A tentative timeline was also agreed upon for sprints and iterations, and timings of our bi-weekly scrum meetings.

A screenshot of the assigned tasks and their agreed-upon timelines may be found in the Milestone 3 folder.

Aa Task name	Status	People	Dates
⌚ Endpoint: FAQ CRUD	Planning	(A) Anubhav Lohani	March 10, 2023
⌚ Endpoint: Ticket CRUD	Planning	(P) Pankaj Bora	March 10, 2023
⌚ Endpoint: Account + Auth	Planning	(A) ARVIND SANKAR	March 10, 2023
⌚ Research python test suites	Planning	(P) Pankaj Bora	March 15, 2023
⌚ Research testing best practices	Planning	(A) ARVIND SANKAR	March 16, 2023
⌚ Create testing POCs	OPEN	(A) Anubhav Lohani	March 17, 2023
⌚ Endpoint: Ticket Misc	Planning	(A) Anubhav Lohani	March 17, 2023
⌚ Endpoint: User Actions	Planning	(A) ARVIND SANKAR	March 17, 2023
⌚ Write User CRUD tests	Planning	(P) Pankaj Bora	March 20, 2023
⌚ Write Ticket Misc tests	Planning	(A) Anubhav Lohani	March 23, 2023
⌚ Write FAQ tests	Planning	(P) Pankaj Bora	March 24, 2023
⌚ Write Ticket CRUD tests	Planning	(A) ARVIND SANKAR	March 24, 2023

The screenshot shows a Notion Gantt chart for the month of March 2023. The tasks are organized into three main columns: To-do, In Progress, and Planning. The tasks listed are:

- To-do: Endpoint: FAQ CRUD (Anubhav Lohani), Endpoint: Ticket CRUD (Pankaj Bora), Endpoint: Account + Auth (Arvind Sankar)
- In Progress: Create testing POCs (Anubhav Lohani), Research python test suites (Pankaj Bora)
- Planning: Research testing best practices (Anubhav Lohani), Write User CRUD tests (Pankaj Bora), Write Ticket Misc tests (Anubhav Lohani), Write FAQ tests (Pankaj Bora), Write Ticket CRUD tests (Arvind Sankar), Endpoint: Ticket Misc (Anubhav Lohani), Endpoint: User Actions (Anubhav Lohani), Write User CRUD tests (Pankaj Bora), Write Ticket Misc tests (Anubhav Lohani), Write FAQ tests (Pankaj Bora), Write Ticket CRUD tests (Arvind Sankar)

Meeting Notes

PIND INC (Pending)

MEETING NO. 04

26 February 2023 / 9:30 PM

ATTENDEES

Arvind Sankar, Pankaj Bora, Anubhav Lohani

AGENDA

1. Review support side wireframe progress
2. Review client side wireframe progress
3. Review storyboard progress
4. Finalize storyboard and wireframe

ACTION ITEMS

1. Add UI elements to the storyboard.
2. Proofread storyboard dialog.
3. Convert figma wireframes to pdf.
4. Create and upload week 2 milestone reports.

PIND INC (Pending)

MEETING NO. 06

04 March 2023 / 9:30 PM

ATTENDEES

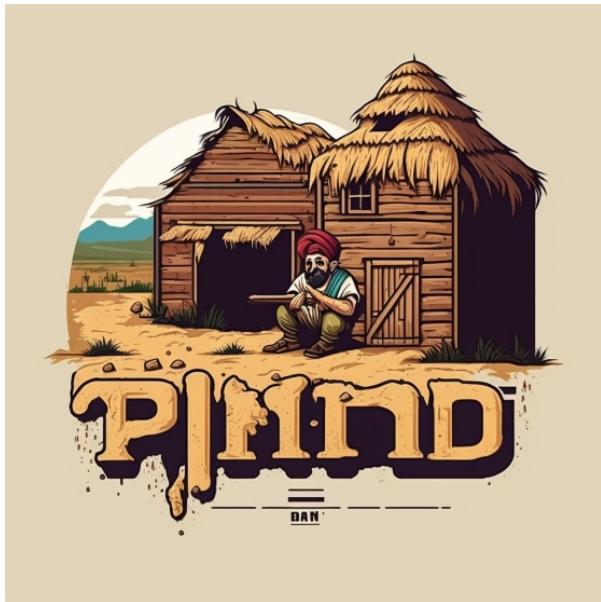
Arvind Sankar, Pankaj Bora, Anubhav Lohani

AGENDA

1. Finalize on a project management tool
2. Identify components and finalize class diagrams
3. Assign and schedule tasks for building the application

ACTION ITEMS

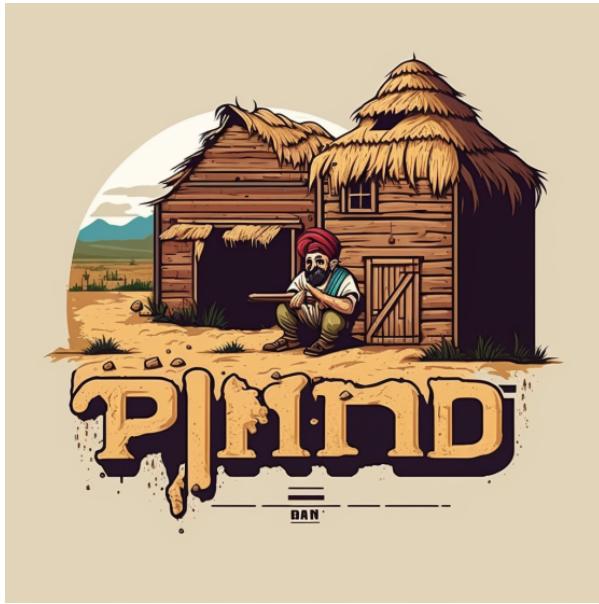
1. Identify components of the application
2. Modularize current class diagram
3. Choose a project management tool
4. Assign tasks and prepare timelines for building the application



SOFTWARE ENGINEERING
MITONE 4

Introduction

The YML file indicating the API endpoints planned for the application can be found [here](#).



SOFTWARE ENGINEERING
MILESTONE 5

Introduction

A test-driven development approach was used to transform the components and APIs identified for the ticket management application into actual code. The following section describes subset of the test cases formulated for implementing and testing functionality of the application. The code for the application and the unit tests are available in this [link](#).

Endpoints setup

The application currently includes the following endpoints and functions:

Home Endpoint

Route: '/' with HTTP method GET

Function name: home()

Description: This endpoint returns a string "You have found this API. Badhai ho" as a response. It is accessed with a GET request to the root URL of the application.

User register Endpoint

Route: '/register' with HTTP method POST

Function name: register()

Description: This endpoint allows users to register by sending a JSON payload in the request body. The payload should contain name, username, email, password, and admin fields. The function hashes the password and creates a new User object in the database with the provided data. It returns a JSON response with a success key set to True upon successful registration.

User login Endpoint

Route: '/login' with HTTP method POST

Function name: login()

Description: This endpoint allows users to log in by sending HTTP basic authentication credentials in the request header. The function calls an authenticate_user() function to authenticate the user and returns the authentication data as a JSON response.

Ticket creation Endpoint

Route: '/ticket' with HTTP method POST

Function name: create_ticket(current_user)

Description: This endpoint allows authenticated users with a token to create a new ticket by sending a JSON payload in the request body. The payload should contain title, content, date, and user_id fields. The function creates a new Ticket object in the database with the provided data and returns a JSON response with a success key set to True upon successful ticket creation. This endpoint also has a token_required decorator which requires the user to be authenticated with a valid token.

FAQ Endpoint

Route: '/faq' with HTTP method GET

Function name: all_faqs()

Description: This endpoint returns all FAQs from the database as a JSON response. It retrieves all Faq objects from the database, serializes them, and returns them in a JSON response with a faqs key containing the serialized FAQs. It is accessed with a GET request to the /faq URL of the application.

Fixture Setup

The following fixtures were used to create a test environment and initialize a fixture indicative of a temporary database to perform, among other things, perform CRUD operations on tickets, ensuring the authorized user access the required endpoints, etc.

```
@pytest.fixture()

def app():
    app = Flask(__name__)
    app.config.update(
        SECRET_KEY=SECRET_KEY,
        SQLALCHEMY_DATABASE_URI=TEST_SQLALCHEMY_DATABASE_URI
    )
    app.register_blueprint(appc)
    app.app_context().push()
    db.init_app(app)
    Ticket.query.delete()
    db.session.commit()
    return app

@pytest.fixture()
def client(app):
    return app.test_client()
```

Test Cases

Test Case 1:

Page being tested: '/ticket'

Inputs: POST request without authentication headers, with JSON data including "date", "title", and "content"

Expected Output: The response should typically return a HTTP Status Code: 401 (Unauthorized) with a message stating "Unauthorized request". Further, since a request to create ticket is made by user without the authentication header/token, the database will not be updated, thereby returning 0 as Number of tickets in the database.

Actual Output:

HTTP Status Code: 401 (Unauthorized)

Response Data: "Unauthorized request" in the response data

Number of tickets in the database: 0

Result: Success

```
(se) D:\Projects\IITM\SE\proj\source>pytest --verbose -k test_create_ticket_admin tests.py
===== test session starts =====
platform win32 -- Python 3.10.5, pytest-7.2.2, pluggy-0.13.1 -- D:\Projects\IITM\SE\proj\source\se\Scripts\python.exe
cachedir: .pytest_cache
rootdir: D:\Projects\IITM\SE\proj\source
collected 5 items / 4 deselected / 1 selected

tests.py::test_create_ticket_admin PASSED [100%]

===== warnings summary =====
database.py:5
  D:\Projects\IITM\SE\proj\source\database.py:5: MovedIn20Warning: The ``declarative_base()`` function is now available as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: https://sqlalchemy.me/e/b8d9)
    Base = declarative_base()

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1 passed, 4 deselected, 1 warning in 0.62s =====
```

Test Case 2:

Page being tested: '/ticket'

Inputs: POST request with authentication headers for an admin user, with JSON data including "date", "title", and "content"

Expected Output: The response should typically return a HTTP Status Code: 403 (Forbidden) with a message stating "Forbidden". Further, since a request to create ticket is made by an unauthorized user, the database will not be updated, thereby returning 0 as Number of tickets in the database.

Actual Output:

HTTP Status Code: 403 (Forbidden)

Response Data: "Forbidden" in the response data

Number of tickets in the database: 0

Result: Success

```
(se) D:\Projects\IITM\SE\proj\source>pytest --verbose -k test_create_ticket_admin tests.py
=====
test session starts =====
platform win32 -- Python 3.10.5, pytest-7.2.2, pluggy-0.13.1 -- D:\Projects\IITM\SE\proj\source\se\Scripts\python.exe
cachedir: .pytest_cache
rootdir: D:\Projects\IITM\SE\proj\source
collected 5 items / 4 deselected / 1 selected

tests.py::test_create_ticket_admin PASSED [100%]

=====
warnings summary =====
database.py:5
D:\Projects\IITM\SE\proj\source\database.py:5: MovedIn20Warning: The ``declarative_base()`` function is now available as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: https://sqlalche.me/e/b8d9)
    Base = declarative_base()

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1 passed, 4 deselected, 1 warning in 0.69s =====
```

Test Case 3:

Page being tested: '/ticket'

Inputs: POST request with authentication headers for a student user, with JSON data including "date", "title", and "content"

Expected Output: The response returns an HTTP Status Code: 200 (OK). Further, since a request to create ticket is made by an authorized user, the database will be updated, thereby returning 1 as Number of tickets in the database. Further, details of the ticket would be returned as follows:

Ticket details in the database:

date: "123123123"

title: "test test1"

content: "test_create_ticket_without_auth"

status: "Open"

likes: 0

Actual Output:

HTTP Status Code: 200 (OK)

Number of tickets in the database: 1

Ticket details in the database:

date: "123123123"

title: "test test1"

content: "test_create_ticket_without_auth"

status: "Open"

likes: 0

Result: Success

```
(se) D:\Projects\IITM\SE\proj\source>pytest --verbose -k test_create_ticket_student_with_complet
ejson tests.py
=====
platform win32 -- Python 3.10.5, pytest-7.2.2, pluggy-0.13.1 -- D:\Projects\IITM\SE\proj\source\
se\Scripts\python.exe
cachedir: .pytest_cache
rootdir: D:\Projects\IITM\SE\proj\source
collected 5 items / 4 deselected / 1 selected

tests.py::test_create_ticket_student_with_completejson PASSED [100%]

=====
warnings summary =====
database.py:5
D:\Projects\IITM\SE\proj\source\database.py:5: MovedIn20Warning: The ``declarative_base()`` fu
nction is now available as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Backgroun
d on SQLAlchemy 2.0 at: https://sqlalche.me/e/b8d9)
    Base = declarative_base()

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1 passed, 4 deselected, 1 warning in 0.61s =====
```

Test Case 4:

Page being tested: '/ticket'

Inputs: POST request with authentication headers for a student user, with JSON data including "date", "title", and "content", and extra data including "likes" and "status"

Expected Output: The response returns an HTTP Status Code: 200 (OK). Further, since a request to create ticket is made by an authorized user, the database will be updated, thereby returning 1 as Number of tickets in the database. Further, details of the ticket would be returned as follows:

Ticket details in the database:

date: "123123123"

title: "test test1"

content: "test_create_ticket_without_auth"

status: "Open"

likes: 0 (likes field should not be updated as per the input)

Actual Output:

HTTP Status Code: 200 (OK)

Number of tickets in the database: 1

Ticket details in the database:

date: "123123123"

title: "test test1"

content: "test_create_ticket_without_auth"

status: "Open"

likes: 0 (likes field should not be updated as per the input)

Result: Success

```
(se) D:\Projects\IITM\SE\proj\source>pytest --verbose -k test_create_ticket_student_extrajson tests.py
===== test session starts =====
platform win32 -- Python 3.10.5, pytest-7.2.2, pluggy-0.13.1 -- D:\Projects\IITM\SE\proj\source\se\Scripts\python.exe
cachedir: .pytest_cache
rootdir: D:\Projects\IITM\SE\proj\source
collected 5 items / 4 deselected / 1 selected

tests.py::test_create_ticket_student_extrajson PASSED [100%]

===== warnings summary =====
database.py:5
  D:\Projects\IITM\SE\proj\source\database.py:5: MovedIn20Warning: The ``declarative_base()`` function is now available as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: https://sqlalche.me/e/b8d9)
    Base = declarative_base()

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1 passed, 4 deselected, 1 warning in 0.68s =====
```

Test Case 5:

Page being tested: '/ticket'

Inputs: POST request with authentication headers for a student user, with incomplete JSON data missing required fields

Expected Output: The response should typically return a HTTP Status Code: 400 with a message stating "Bad Request". Further, since a request to create ticket is without the required information, the database will not be updated, thereby returning 0 as Number of tickets in the database.

Actual Output:

Response status code should be 400 (Bad Request)

Response data should contain the message "Bad Request"

Number of tickets in the database should be 0

Result: Success

```
(se) D:\Projects\IITM\SE\proj\source>pytest --verbose -k test_create_ticket_student_incompletejson tests.py
===== test session starts =====
platform win32 -- Python 3.10.5, pytest-7.2.2, pluggy-0.13.1 -- D:\Projects\IITM\SE\proj\source\se\Scripts\python.exe
cachedir: .pytest_cache
rootdir: D:\Projects\IITM\SE\proj\source
collected 5 items / 4 deselected / 1 selected

tests.py::test_create_ticket_student_incompletejson PASSED [100%]

===== warnings summary =====
database.py:5
  D:\Projects\IITM\SE\proj\source\database.py:5: MovedIn20Warning: The ``declarative_base()`` function is now available as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: https://sqlalche.me/e/b8d9)
    Base = declarative_base()

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1 passed, 4 deselected, 1 warning in 0.63s =====
```