# SOFTWARE ENGINEERING Final Report

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE COURSE:

**BSCSS3001: Software Engineering** 

By:

Arya Bhattacharyya (21f2000436) Varun Venkatesh (21f1000743) Chirag Goel (21f2000540)



INDIAN INSTITUTE OF TECHNOLOGY, MADRAS April, 2023

# Report on work done in different milestones

- Milestone 1 :The group identified the different users of the application in terms of the primary, secondary and tertiary users. Post the same, user stories (following SMART guidelines) were written for the different users identified earlier. 25 user stories which had the mandatory features as suggested by the project and some additional features as deemed necessary were compiled and submitted before the deadline.
- <u>Milestone 2</u>: Four storyboards (using Comicgen and Google Slides) were depicted using comic strips:
  - 2 storyboards were made from the perspective of a support agent
  - 1 storyboard was made from the perspective of a student
  - 1 storyboard was made from the perspective of a manager.

Wireframes were designed for the application using the design heuristics that were learnt in the Software Engineering course. Essential usability principles that were incorporated in our wireframe was as follows:

- Effectiveness: To ensure that users can carry out their work efficiently.
- Efficiency: Each user can effectively carry out their task in a minimal number of steps.
- Safety: Confirmation prompts to prevent any unwanted actions
- Learnability: Useful tool-tips on mouse hover
- Memorability: Easy to use, consistent interface, which helps the user to remember the same.

Some design heuristics that were followed during this phase were:

- Consistency: Layout of different pages are similar.
- Clean and Functional Design: Clutter free easy to recognize design which focuses on features and functionality, with pleasant aesthetics.
- Show Status: Students can see if support agents have viewed the ticket raised by them. Post that, on answering the ticket, the ticket is displayed as closed (which can be reopened by the student if required), again showing the status of their ticket.
- o Provide Help: Useful tool-tips on mouse hover.
- Milestone 3: A rough project schedule was drawn upon keeping in mind the deadlines for each milestone and the amount of work that was required for the same. Sprints were planned for the different tasks. Jira was chosen as an appropriate project management tool and a Gantt Chart roadmap was plotted to keep track of the progress. Different components were designed as UML diagrams using the Star UML software to model the software system. Details of scrum meetings conducted up until this stage were presented as "minutes of the meet" in the milestone 3 submission.

- Milestone 4: API endpoints were described in the YAML file (Using OpenAPI 3.0 specification) and the coding for the backend of the application was also commenced during this milestone. The API specification was supposed to be the first version of the backend, which we believed could be subsequently changed based on any additional requirements/dependencies.
- <u>Milestone 5</u>: Extensive testing (100+ tests) was done using Pytest for the code written for the backend. Tests directly relating to user stories were documented in the Milestone 5 report using the format given in the instructions. All tests were passed. Frontend coding for the software was commenced in this milestone.
- Milestone 6: Frontend was completed and fully integrated with the backend.
  Frontend-backend integration was tested manually for all the pages to ensure
  that the software was behaving as expected. The second version of the API
  documentation was completed based on the changes and new introductions to
  the API. Finally, a report was made for the completed project and a video demo
  was recorded for the working application.

# **Implementation Details of the Project**

- Technologies and tools used
  - o <u>Technologies for the backend</u>
    - Flask
    - Flask Restful (For creating API endpoints)
    - Flask SQLAlchemy
    - Pytest (Only for testing)
    - Swagger Editor (Only for API documentation)
    - Postman, Thunderclient (For checking API endpoints manually)
    - Mailgun A Transactional email service provider is used to send relevant emails to the users
  - Technologies for the frontend
    - Vue 3 CLI
    - Javascript
    - Vue Router
    - Vuex Store
    - Bootstrap (for aesthetics and styling)
    - HTML and CSS
    - ESlint to help with debugging the frontend
    - Axios and Fetch API for requests.

## General technologies used

- Github (for versioning, code management, tracking, reviewing, issues, etc)
- Algolia Search is used in the backend and frontend to create a smooth search experience
- Jira (for project management)
- ComicGen (for creating comics for storyboarding)
- Star UML (for creating UML diagrams)

#### Hosting

Currently the application is not hosted publicly.

### • Instructions to run our application

- o On Ubuntu/MAC OS:
  - Git clone the repository.
  - Change the directory to the "backend" directory inside the "Milestone-6-Final-Submission" directory using the command : cd ./Milestone-6-Final-Submission/Code/backend
  - Create a Python virtual environment using the command: python3 -m venv "<<Name of the virtual environment">>>
  - Activate the virtual environment using the command: source <<Name of the virtual environment>>/bin/activate
  - Install the requirements using the command : pip3 install -r requirements.txt
  - Now, on the terminal type the following command to start the flask server:
    - python3 main.py
  - In the same "backend" directory, in a new terminal inside the virtual environment start the redis server by typing: redis-server
  - Similarly, start the celery worker in the "backend" directory inside the virtual environment by typing: celery -A main.celery worker -I info
  - Furthermore start the celery beat in the "backend" directory inside the virtual environment by typing: celery -A main.celery beat --max-interval 1 -I info
  - For the frontend of the application, we would require node js and npm. We recommend to install the latest LTS version of Node js from the following link: <a href="https://nodejs.org/en">https://nodejs.org/en</a>
  - Once Node js is installed and npm is working, open a new terminal and change the directory to the "Code" folder inside the "Milestone-6-Final-Submission" directory which we saw earlier.

- Once inside the directory, change the directory to "frontend" by using the command: cd ./frontend/
- Now, run the following command to install the necessary packages for the frontend: npm install.
- After successful installation of the required packages, serve the frontend using the command: npm run serve
- The frontend server would be available on the URL: localhost:8080
- Please note that for the email functionality and search functionality to work, you would need API keys. In order to secure these API keys, they aren't part of the repository. You would need to request the keys from us

## o On Windows:

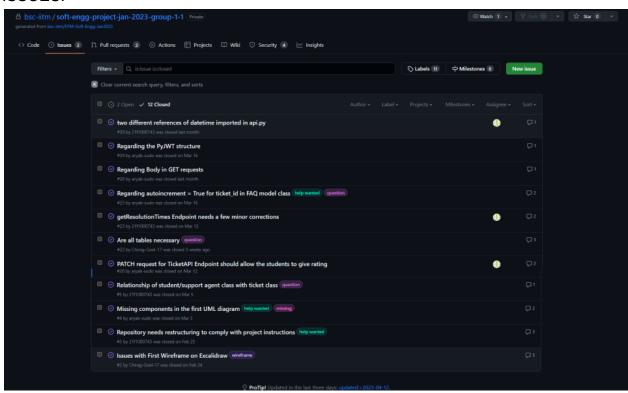
- Git clone the repository.
- Change the directory to the "backend" directory inside the "Milestone-6-Final-Submission" directory using the command : cd .\Milestone-6-Final-Submission\Code\backend
- Create a Python virtual environment using the command: python3 -m venv "<<Name of the virtual environment">>>
- Activate the virtual environment using the command: .\<<Name of the virtual environment>>\Scripts\activate (You might have to enable Activate.ps1 on Windows, for which you can use the command (in a single line of the terminal): Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser {If you are still not able to activate your virtual environment, please visit these links to know more:
  - a. <a href="https://docs.python.org/3/library/venv.html">https://docs.python.org/3/library/venv.html</a>
  - b. <a href="https://stackoverflow.com/questions/18713086/virtualenv-wo">https://stackoverflow.com/questions/18713086/virtualenv-wo</a> nt-activate-on-windows
- Install the requirements using the command : pip install -r requirements.txt
- Now, on the terminal type the following command to start the flask server:
  - python main.py
- In the same "backend" directory, in a new terminal inside the virtual environment start the redis server by typing: redis-server
- Similarly, start the celery worker in the "backend" directory inside the virtual environment by typing: celery -A main.celery worker -I info

- Furthermore start the celery beat in the "backend" directory inside the virtual environment by typing: celery -A main.celery beat --max-interval 1 -I info
- For the frontend of the application, we would require node js and npm. We recommend to install the latest LTS version of Node js from the following link: <a href="https://nodejs.org/en">https://nodejs.org/en</a>
- Once Node js is installed and npm is working, open a new terminal and change the directory to the "Code" folder inside the "Milestone-6-Final-Submission" directory which we saw earlier.
- Once inside the directory, change the directory to "frontend" by using the command: cd .\frontend\
- Now, run the following command to install the necessary packages for the frontend: npm install.
- After successful installation of the required packages, serve the frontend using the command: npm run serve
- The frontend server would be available on the URL: localhost:8080

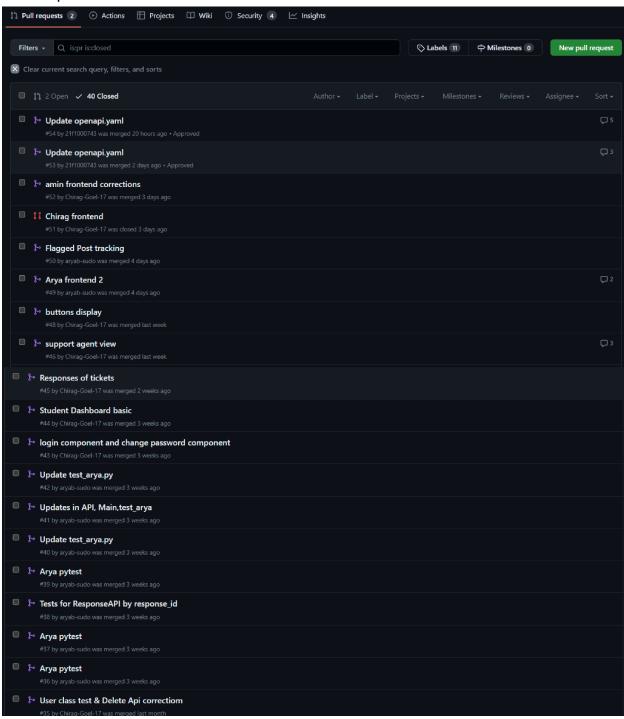
# Code Review, Issue Reporting and Tracking

This was completely done on GitHub.

#### ISSUES:



## Pull Requests:



#### Code Reviews

