\n - Newline

Team 19

Deliverable 4

Notice

- Due to some issues with django backend while integrating and the loss of a key team member in that area, we are in the process of switching to node js.

Overview of Tasks

- Our goal is to complete around 5 hours/points per week. Hence 10 points per sprint
- Our tasks can be found on Kanban board
    - Stories contains our user stories, the parents of our tasks
    - Task are the tickets to that make up the feature defined by our user story

Overview of Testing

Frontend

- Will be similar to acceptance test

Backend

- Tests written for account management, organizations management, users management and upload submissions

Accepance Test

- The first page that the user comes across is the login page. The login page asks to enter the email address and the password of the user and upon clicking the login button, it checks if the email address and password is valid. If it is, then it lets the user in the app (to the upload page) upon clicking the submit button otherwise it makes the input text field red and does not redirect anywhere. In the app, there is a dashboard on the left which contains links to all the different pages. There are different links for TEQ and other service providing organizations. For TEQ, the dashboard contains a links to reports page, profile page, add account page and add organization page whereas other organizations do not have link to add organization and add account page. In the upload page, there is a list of files that the organization has uploaded in the past and there is upload button. Upon clicking the upload button, a pop up comes up which allows the user to drag a file into it or by clicking on it, it opens up a directory in the computer to select file. After selecting a file, the user can click the upload button to upload the file. This file will be updated on the list of files. Moreover, the user can also add new templates by inputting the rows of the template and clicking add template. In the reports page, there is a list that contains all the reports created by that organization. For each report, there is also a delete button which can be used to delete the report. Moreover, there are 6 different types of charts in the reports page. When the user clicks on a chart, it opens a pop up where the user can select the query and column number and using the data the user selected, the graph is generated. The user can do the same for all 6 charts. In the add account page, the organization can create a user account where they have to enter the organization id, first name, last name, admin (True if they want to create an admin account, False if they don't want to), password and confirm password. After clicking the submit button, the user account is created for that organization. In the add organization page, the TEQ can add other organizations to the database by inputting the organization id and the name and clicking the submit button. In the profile page, for the TEQ organization, there is a list of all the organizations in the database and beside it, the user can see all the details of each organization as it selects each organization from the list. For other organizations, there is a list of users in the organization and on clicking each users, the detailed information of each user is displayed.

Code Review

Code Review Strategy

Our guideline to do code review is to check if the code meets the following requirements.

- Has good readability.
- Meets the client needs.
- Has been written so that changes can be easily adaptable.
- Is implemented to be efficient.
- Can be reusable.
- Can be easily testable.

Code Review Summary

Frederic Pun

Front-end

Reviewing the code, more specifically the login page. There are some issues with user experience. It lacks the ability to provide feedback to the user for when a login fails (even on an extremely basic level of an alert). However, the functionality of the code is clean and optimized. The code is kept clean (meaning without any dead code), and uses caching when possible to reduce the number of unnecessary work when the UI is being built.

The biggest issue however, which is dominant in most of the code, is the lack of comments. This would greatly help the onboarding of our team members that would allow for better efficiency

Back-end

When reviewing the backend code (Django version), the one good thing is the extremely clean handler for account operations. It is laid out cleanly, documented cleanly, and is extremely clear what operations and checks are happening as the request runs through the function. Something that can be improved would be the introduction of utility methods to reduce the amount of repeated code.

The biggest issue however is the lack of standard and efficiency. The whole entire project lacks file structure to organize handlers. The biggest offender however is within the uploading section, which has too much dead code/files, and has camelcase naming (when Python standard is to use kebab case)

I would leave my review of the new backend code to the other team members due to my possible bias from being currently the main/sole contributor of it currently.

Chaoyue Xi

Back-end

For the old backend code(Django version), we have finished the data upload part. We succeeded to upload test data to our postgreSQL database on Leop's computer. However, we failed to do the same thing on another computer because we cannot migrate database. And we cannot figure out the reason. We also met many other trouble because of the lack of the understanding of Django. Finally, we realized that we have to give up Django.

Our new backend code is in Nodejs. Nodejs is much easier to learn because it is closer to natural language and one of our teammate Fredric is familiar with Nodejs. I think the design for our new backend code is pretty good. There are three parts: util contains helper functions, controllers use help functions to realize main ability and routes call controllers. For now, the biggest advantage of this design is that it is convenient for me to write unit testing. For example, if I want to test login, I just need to call '/login' in auth.js in routes and I don't need to care about others.

Qingtian Wang

Back-End

As the django-based backend, due to the slow progression and lack of skills in most of the backend team, the quality of code is not high enough to be evaluated as efficient, or clear. Even though some comments were made for reading, and a document was created for front-end to understand endpoints in backend, the django based backend will still need a lot of improvements if development were to continue.

Fortunately, as one of our decisive member for engine selection left from the team, we started using Node.Js as our new backend engine. One benefit of Node.js is its similarity to natural language, thus reduce the amount of comments needed for programming and unit testing. We also got a better document for frontend - backend communication details. Lastly but not least, there are many helper function created by Frederic for unit testing that saved a lot of code and work. One of the improvements that can be made for node.js backend is make more local variables for texts that will either be response or be keyword for database operation, as there seems to be too many duplicate texts.

Vishwaa Patel

Front-end

In different front-end pages such as Profile, add account and add organization, the same components like Modal (a pop up window), a list which can contains any fragment, a form (set of input fields in order) are used which is good in terms of code reusability. Moreover, there are separate classes that fetch information from the database and pass it on to the component. Therefore, the view and the model are seprarated. However, the code still requires some cleaning of the methods and the values cached to improve readability and efficiency.

Code Review Debreifing Meeting Video

https://drive.google.com/open?id=1sFSPtYLRu9XzUKXdt-hFtJUSZ00spxhU

Sprint 3

Duration: October 28 - November 4
https://docs.google.com/spreadsheets/d/1jQ8F4uXXfon3L4rK_rBWbQHDeP-phsp9FXXHslXHDFU/edit?usp=sharing

Backlog

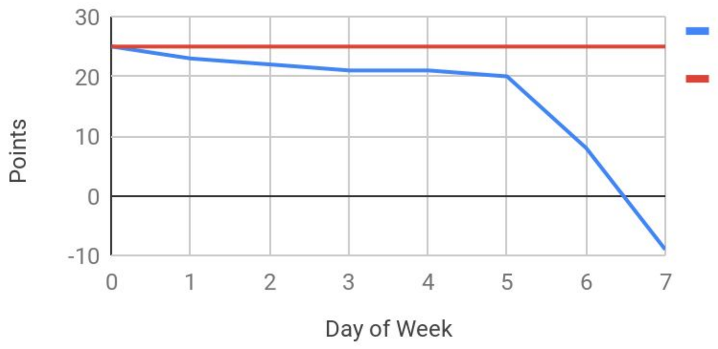
https://dev.azure.com/cscc01f18/Deliverables/_sprints/taskboard/Deliverables%20Team/Deliverables/Iteration%202

Burndown Chart

User Stories	Task	Dependency	Story Points	1	2	3	4	5	6	7
S-1	1-F-2		2	V:2						
S-2	2-F-1		1		V:1					
S-3	3-F-1		1			V:1				
S-6	6-B-1		8					J:4	J:4	
S-7	7-F-1		3						F:3	
S-7	7-B-1		3							C:3
S-8	8-F-1		1						F:1	
S-8	8-B-1		2							C:2
S-9	9-F-1		1							F:1
S-9	9-B-1		5							Q:5
S-10	10-F-1		3							F:3
S-10	10-B-1		3							
S-11	11-F-1		1					V:1		
S-11	11-B-1		3							
S-11	11-F-2		3						V:2	V:1
S-11	11-B-2		1							
S-12	12-F-1		3							
S-12	12-B-1		2							
S-13	13-F-1		5							
S-13	13-B-1		3							

Burndown Chart

Data\Day	0	1	2	3	4	5	6	7
planned	25	23	22	21	21	20	8	-9
actual	25	25	25	25	25	25	25	25



Sprint Report

- Made the transition from django to nodejs for the backend development.
- Completed the integration of login.
- The backend for organization add/remove/modify, user add/delete/modify and upload and submission add/delete/modify in progress.

Sprint 4

Duration: November 5- 11

https://docs.google.com/spreadsheets/d/1jQ8F4uXXfon3L4rK_rBWbQHDeP-phsp9FXXHslXHDFU/edit?usp=sharing

Backlog

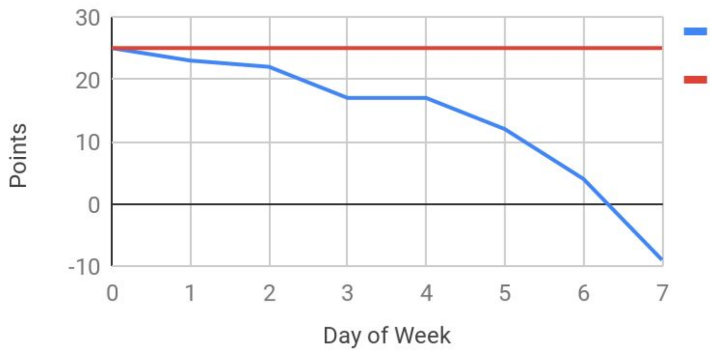
https://dev.azure.com/cscc01f18/Deliverables/_sprints/taskboard/Deliverables%20Team/Deliverables/Iteration%203

Burndown Chart

User Stories	Task	Dependency	Story Points	1	2	3	4	5	6	7
S-1	1-F-2		2	V:2						
S-2	2-F-1		1		V:1					
S-3	3-F-1		1			V:1				
S-6	6-B-1		8					J:4	J:4	
S-7	7-F-1		3			F:3				
S-7	7-B-1		3							C:3
S-8	8-F-1		1					F:1		
S-8	8-B-1		2							C:2
S-9	9-F-1		1					F:1		
S-9	9-B-1		5							Q:5
S-10	10-F-1		3							F:3
S-10	10-B-1		3							
S-11	11-F-1		1			V:1				
S-11	11-B-1		3							
S-11	11-F-2		3					V:3		
S-11	11-B-2		1							
S-12	12-F-1		3							
S-12	12-B-1		2							
S-13	13-F-1		5							
S-13	13-B-1		3							

Burndown Chart

Data\Day	0	1	2	3	4	5	6	7
planned	25	23	22	17	17	12	4	-9
actual	25	25	25	25	25	25	25	25



Sprint Report

- The endpoints for the organization add/remove/modify, user add/modify/delete and submission upload/modify/delete are ready.
- Testing has been performed on these endpoints and for authentication of user (login/logout).
- The integration for the front-end and the back-end is under progress. Setting up the required adaptors on both ends.
- Even though not a lot of tickets moved, after the integration between frontend and backend, a lot of tickets would move to resolved