

Comments over Code (COC)



CSCC01

Team 18

Table of Contents

Tasks	3
Sprint Report	4
Sprint Plan	4
Sprint Execution	4
Burndown Chart	4
Client Meeting Questions	4
Code Review	5
Strategy	5
Summary	5
Nathan Chau	5
Jikai Long	6
Tanaan Karunakaran	6
Chenkai Lin	6
Debriefing Meeting	6
Acceptance Test	6
User Story 1	6
User Story 2	7
User Story 3	7

Tasks

U1	User is able to login with username and password in the login page	
Task	Description	Story
T1	Create backend database to store username password	5
T2	Create UI for login page(2 input box and a login button)	3
T3	Connecting UI with backend	2
U2	User is able to upload excel file to database	
Task	Description	Story
T4	Create File upload buttonPoints	2
T5	Reading and extracting data from excel file into a json file	5
T6	Pushing data to database	2
U3	Format uploaded data	
Task	Description	Story
T7	Format the raw json data uploaded by user to be more structures and organized	5
T8	Handle inconsistencies of formating by creating an object that has all the default values that each attribute should be and then comparing the object with the data in json.	3
T9	Handle duplicates by checking if an id of the row exists. If so update the row	3
U4	Display data on the screen	
T10	Display data on the screen	5
U5	Export data into an data sheet	
T11	Exporting all data	5

Sprint Report

Sprint Plan

User Stories	Tasks	Dependency	Story Points	11/03	11/04	11/05	11/06	11/07	11/08	11/09
U3	T7		5	J:2	J:2	J:1				
U3	T8		3							
U3	T8		3					N:1	N:1	N:1
U4	T9		5	C:3	C:2					
U5	T10		5	T:2	T:2	T:1				

Sprint Execution

User Stories	Tasks	Dependency	Story Points	11/03	11/04	11/05	11/06	11/07	11/08	11/09
U3	T7		5						J:3	
U3	T8		3							J:3
U3	T9		3						N:1	N:1
U4	T10		5							C:3
U5	T11		5						T:1	

Burndown Chart

Days	11/03	11/04	11/05	11/06	11/07	11/08	11/09
Provisial (in story points)	18	18	13	3	3	3	0
Actual (in story points)	18	18	18	18	18	10	7

Client Meeting Questions

1. Q: What type of operations would the company/users want to do? (example, edit information in a file)

A: There are 3 types of users: Agencies and Participating Organizations, TEQ company, and UTSC Project staff.

Each type of user will have different types of uses with the app.

Agency and Participating Organizations:

a) These users can only unload excel/csv files. This is the only action they can do. As such, they will not be able to view the data once they upload a file or view anything else.

TEQ Company: These users can do several actions.

- a) Users can upload excel/csv files.
- b) Can run queries of the collected data (example, how many people live in Scarborough)
- c) Create new accounts for new users (organizations and UTSC project staff). As such, giving restrictions to accounts.
- d) Adding or modifying existing excel templates.
- e) Be able to generate reports based on information in the excel files

UTSC Project Staff: These users can do several actions

- a) View the collected data.
- b) Edit the code of the app so they can edit the code in the future if necessary.
- c) See data about the project (very vague)
- d) Perform analysis based on the information in system (very vague)

2. Q: What would be displayed when searching, like searching?

A: The rows of the matched data. For example, if I want all people that live in the city Scarborough, I enter Scarborough in the search field. This will give me all the rows that have people living in Scarborough.

Code Review

Strategy

Our strategy of reviewing is to look into the code of all other members and try to identify any improvements or flaws within the code.

Things to look for: - efficiency of code

- clarity and organized structure of code
- comments if necessary
- valid variable names

Summary

Nathan Chau

Back-End code:

- naming conventions could be more clear i.e camel notations
- well commented which brings clarity to the code

- code can be more consistent in spacing and alignments

Jikai Long

Front-End code: The front-end looks very clean since there are a lot of helper functions it makes the structure easier to understand what is going on, but it does need comment on most functions. All variable names are clearly indicated and the interface looks nice.

Login: The login code, still, need more comment for me to understand how variable gets picked by the input field. But overall, everything including database is declared in a simple way, pretty good quality.

Tanaan Karunakaran

Back-End code: Code is well commented and is easy to read. The way it was coded makes sense. Most code in the back end is understandable and easy to understand.

Front-End (Login): Some things are not commented. Some variable names are poorly chosen. It takes time to understand what the code is actually doing. Even though the login looks nice, it will be slightly annoying to add-on to the Ui.

Chenkai Lin

Front-end code: clear and easy to understand

datauploader: naming maybe confuse developers, functions and variables should be more organizational.

Hugh Ding

Back-end code: comments occasionally slightly misleading but otherwise well done; well organized; some inconsistency in naming conventions

Front-end code: structured well but lacks comments

Debriefing Meeting

<https://www.youtube.com/watch?v=6kaiVVK8XF0>

Acceptance Test

User Story 1

User can login by typing their username and password in the given textbox and click Continue to login into the web applications..

User Story 2

User can upload a csv file by clicking on upload and choosing the csv file with a set template.

User Story 3

Data is automatically formatted and warns user if there are any empty sheets or duplicate data.