University of Toronto Scarborough

Deliverable 4

CSCC01H3 - Summer 2020

Team Members of Team 8:

Alac Wong

Calvin Cheng

Jin Rong (Tony) Cao

Robert Nichita

Winnie Lam

Team TA:

Sally Moon

Table Of Contents

1 - Release Plan

We will continue with our model of having weekly sprints. These weekly sprints begin on Sunday at 5 pm EST and end the following Sunday at 4:59 pm EST for a sprint length of one week. The reason we choose one week sprints is that people, in general, have been known to work better under deadlines, hence the shorter sprints. This also encourages the team to break down stories or features into smaller chunks which makes planning easier and more accurate. With a sprint length of one week, impediments and slowdowns are highlighted more quickly since the team is expected to get feature(s) done weekly. This will allow for greater visibility and understanding of the team's progress within a sprint.

2 - Product Backlog

2.1 New User Stories Added since Deliverable 3

After consulting with our client regarding the system we have developed, new user stories were added to the product backlog based on the feedback we received. Namely, the new features revolved around the ability for restaurant owners to edit their pages, allowing users to browse restaurants by geographical location and implementing verification through text messages.

As Al Capone (system admin), I want to be able to edit the home page content on the site.

- Points: 7
- Priority: Medium

As Robert Downey (a restaurant manager), I want to be able to add items to my menu on set up.

- Points: 5
- Priority: High

As Robert Downey (a restaurant manager), I want to be able to set up my restaurant page details.

- Points: 5
- Priority: High

As a customer, I want to be able to view nearby restaurants on a map.

- Points: 8
- Priority: Medium

As a customer, I want to be able to view my location on a map.

- Points: 8
- Priority: High

As a customer, I want to be able to use SMS verification for my account.

- Points: 5
- Priority: Low

2.2 Existing Product Backlog from Deliverable 3

Feature: *Login & Signup*

As a restaurant owner, I want to log in so that I can update my restaurant information. (COMPLETED - SPRINT 2)

- Points: 7
- Priority: High

[Deleted] As Al Capone (system admin), I want to allow customers and restaurant owners to reset their password.

- This user story was deleted due to the fact we used Auth0 to handle login and sign up of users
- Auth0 also handles the process of resetting a password and there's no action required on our ends for the process of login, user sign up and resetting the password

Feature: *Connecting with Customers*

As Hames Jarden (a restaurant owner), I want to upload videos from files so that I can update my page with videos that I do not want on other platforms.

- Points: 5
- Priority: High

As Hames Jarden (a restaurant owner), I want to embed videos from YouTube so that I can easily link videos on different platforms without uploading twice.

- Points: 4
- Priority: High

As Hames Jarden (a restaurant owner), I want to delete comments from timeline posts so I can remove the comments I do not want.

- Points: 3
- Priority: Medium

As Hames Jarden (a restaurant owner), I want to delete timeline posts so I can get rid of irrelevant content.

- Points: 3
- Priority: Medium

As Hames Jarden (a restaurant owner), I want to list and link my social media platforms so that I can connect with customers on different platforms. (COMPLETED - SPRINT 4)

- Points: 1
- Priority: Low

Feature: *Restaurant Profiles (i.e. Address, Menu Items, Reviews, etc...)*

As Robert Downey (a restaurant manager), I want to change items on the menu so that I can adapt to my seasonal menu.

- Points: 5
- Priority: High

As Steve Hobbs (a small business owner), I would like to have access to restaurants' contact information so that I can call them to modify my orders. (COMPLETED - SPRINT 2)

- Points: 1
- Priority: High

As Bob Lee (a cuisine exploring user), I want to be able to see pictures and reviews of dishes served at a restaurant so I can decide if it suits my siblings' appetites.
(COMPLETED - SPRINT 2)

- Points: 6
- Priority: High

As Rachel Lin (a food ordering user), I want the restaurants to indicate what kind of food they serve so that I can select which different type of food to try.

- Points: 3
- Priority: High

As Robert Downey (a restaurant manager), I want to change restaurant information and bio so that I can keep customers informed.

- Points: 2
- Priority: High

As Robert Downey (a restaurant manager), I want to have the option to mark a dish as out of stock/sold out on the web page so that I would not have to cancel on a customer.

- Points: 2
- Priority: Medium

As Rachel Lin (a food ordering user), I want to see user ratings of the restaurant so that I can decide to choose to eat there or not. (COMPLETED - SPRINT 2)

- Points: 2

- Priority: Medium

As Robert Downey (a restaurant manager), I want to change the colours/themes of my web page so that it suits my restaurant aesthetics.

- Points: 2
- Priority: Low

As Karen D'Souza (a restrictive food ordering user), I want to have symbols next to restaurant dishes to denote meals that contain common allergies so I am certain that I am not consuming something dangerous.

- Points: 3
- Priority: Low

Feature: *Customer Placing Orders*

As Steve Hobbs (a small business owner), I want it to be simple to place orders for multiple dishes or quantities of a dish so that I can order for an entire group of people.

- Points: 2
- Priority: High

As Steve Hobbs (a small business owner), I would like to be able to cancel my orders so that I do not pay for incorrect orders.

- Points: 3
- Priority: High

As Bob Lee (a cuisine exploring user), I would like to be able to order from UberEats/DoorDash so that I do not need to leave my house for food.

- Points: 2
- Priority: High

As Steve Hobbs (a small business owner), I would like to see the full transaction history on my business' account for tax reasons.

- Points: 4
- Priority: Medium

As Alice Wong (a food ordering user), I want the option to tip so that I can reward good service.

- Points: 2
- Priority: Low

Feature: *Restaurant Accepting Orders (Restaurant Dashboard)*

As a restaurant manager, I want to be able to view all orders placed.

- Points: 9
- Priority: High

As a restaurant manager, I want to be able to accept orders.

- Points: 7
- Priority: High

As a restaurant manager, I want to be able to cancel orders.

- Points: 7
- Priority: High

As a restaurant manager, I want to be able to mark orders as completed.

- Points: 4
- Priority: High

Feature: *Search Engines/Restaurant Filtering*

As Bob Lee (a cuisine exploring user), I want to view restaurants that serve a specific cuisine of a dish so that I can explore different cuisines.

- Points: 5
- Priority: High

As Rachel Lin (a food ordering user), I want to see a list of recommended restaurants and/or dishes so I can order from restaurants that others like.

- Points: 6
- Priority: High

As Alice Wong (a food ordering user), I want to search for food within a certain price range so that I can stay within my budget.

- Points: 3
- Priority: High

As Alice Wong (a food order user), I want to search for new cuisines, so that I can try new types of food.

- Points: 7
- Priority: High

As Alice Wong (a food order user), I want to search by proximity, so that the food comes quickly to my apartment.

- Points: 9
- Priority: Medium

Feature: *Specials and Discounts*

As Robert Downey (a restaurant manager), I would like to be able to push special deals further up the search engine, so that they are more visible to customers.

- Points: 2
- Priority: Medium

As Robert Downey (a restaurant manager), I would like to indicate which of the dishes have specials so that it attracts the consumers' attention.

- Points: 2
- Priority: Medium

As Alice Wong (a food ordering user), I want to be able to see discounted foods so that I can stay within my budget for expenses.

- Points: 3
- Priority: Low

As Karen D'Souza (a restrictive food ordering user), I want to view restaurants that cater to a specific dietary restriction such as Vegetarian, Halal or Organic foods so that I can suit my family's dietary needs.

- Points: 3
- Priority: Low

Feature: *Restaurant Recipes*

As Robert Downey (a restaurant manager), I want to upload my family recipes, so that I can pass on the generations of good food to others.

- Points: 4
- Priority: Low

As Bob Lee (a cuisine exploring user), I want to be able to view the recipe of a dish so that I can attempt to recreate it at home.

- Points: 1
- Priority: Low

Feature: *Restaurant Reviews and Favourite Restaurants*

As Alice Wong (a food ordering user), I want to save/favourite my favourite dishes so that I can continue to order it and support the business.

- Points: 4
- Priority: Low

As Janet Jackson (a healthcare worker), I want to rate and leave reviews for restaurants, so that I can voice my opinion on the food I eat.

- Points: 3
- Priority: Low

Feature: *Platform Analytics*

As Al Capone (system admin), I would like to be able to view statistics and analytics of user data so that I can gain insights into customer usage trends and modify our services accordingly. -

- Points: 7
- Priority: Low

3 - Sprint Plan

3.1 Sprint 2: June 28th to July 5th

3.1.1 Sprint Backlog

Feature: *Restaurant Pages*

[FRONT] Create UI for restaurant page

- Points: 4
- Priority: Medium
- Acceptance Criteria:
    - Given that the user wants to view details of a specific restaurant, when they press the button to view the restaurant, they should be presented with their contact information, social media links, link to discussion, menu, link to media, and reviews.

[TASK] Create UI for listing all restaurants

- Points: 6
- Priority: High
- Acceptance Criteria:
    - Given that a user wants to see all restaurants on Scarborough Dining, when they click “see all listings” then they should be brought to a page listing all restaurants currently on Scarborough Dining.

As Bob Lee (a cuisine exploring user), I want to be able to see pictures and reviews of dishes served at a restaurant so I can decide if it suits my siblings’ appetites.

- Points: 6

- Priority: High
- Acceptance Criteria:
    - Given that Bob wants to view pictures and reviews of dishes, when he presses to view details of a dish, he should see the details of the dish (includes name, potential allergies, pictures, and reviews).

As Steve Hobbs (a small business owner), I would like to have access to restaurants' contact information so that I can call them to modify my orders.

- Points: 1
- Priority: High
- Acceptance Criteria:
    - Given that Steve wants to see a restaurant's contact information, when he presses the button to see the restaurant page, he should be able to see restaurant contact information.

As Rachel Lin (a food ordering user), I want to see user ratings of the restaurant so that I can decide to choose to eat there or not.

- Points: 2
- Priority: Medium
- Acceptance Criteria:
    - Given that Rachel wants to view average ratings of all restaurants when she clicks the view all restaurants page then Rachel should be presented with a list of restaurants and their corresponding average rating.
    - Given that Rachel wants to view all ratings of a particular restaurant when she clicks the ratings page of a restaurant profile then she should be presented with a list of user ratings (descriptions - if available and a score out of 5).

[TASK] Social Media Links

- Points: 1
- Priority: Medium
- Acceptance Criteria:
    - Given that the customer goes onto the restaurant page when they press a button to view the restaurant, they should be able to click and access links to social media platforms of the restaurant.

Feature: *Webpage Login and Signup*

As a restaurant owner, I want to log in so that I can update my restaurant information.

- Points: 7
- Priority: High
- Acceptance Criteria:
    - Given that the restaurant owner signs up for an account, they should be able to upgrade their account to RO with a button.

    - Given that the RO clicks upgrade, they should have access to pages only ROs can see.
    - Given that the RO clicks upgrade, the account role in the database should be updated.

[FRONT] Create a button to upgrade the default customer account to restaurant owner.

- Points: 4
- Priority: High
- Acceptance Criteria:
    - Given that the restaurant owner signs up for an account, when they click to upgrade their account to RO status then their account should be set to "Restaurant Owner" tier and they should be redirected to RO related UIs.
    - Given that the restaurant owner signs up for an account, when they click to upgrade their account to RO status then they should have access to pages only ROs can see.

[BACK] Create endpoint to update role for accounts

- Points: 3
- Priority: High
- Acceptance Criteria:
    - Given that the RO clicks the upgrade button, when they have been verified then the account role in the database should be updated from basic user to restaurant owner.

Feature: *Web Page Database*

[BACK] Populate database with schema (MOVED TO SPRINT 3)

- The task was not completed and moved to the next sprint due to team member availability
- Points: 6
- Priority: High
- Acceptance Criteria: None as not a user story

Feature: *Webpage Search Engine*

[BACK] Generate cuisine dictionary

- Points: 4
- Priority: High
- Acceptance Criteria: None as not a user story

[BACK] Generate auto tags for dishes (MOVED TO SPRINT 3)

- The task was not completed and moved to the next sprint due to team member availability

- Points: 5
- Priority: High
- Acceptance Criteria: None as not a user story

[BACK] Generate manual tags for dishes
- Points: 5
- Priority: High
- Acceptance Criteria:
    - Given that the RO provides a tag when a food item is selected on their page then the tag will be attached to the food item

Feature: *Webpage Maintenance*

As Al Capone (system admin), I would like to be able to deploy the website to a remote server so that it can be accessed by potential food ordering customers.
- Points: 6
- Priority: High
- Acceptance Criteria:
    - Given that Al Capone is presented with the website link, when he enters it into the web he should be able to see the homepage of the website (indicating it can be accessed).

3.1.2 Assignment of Tasks

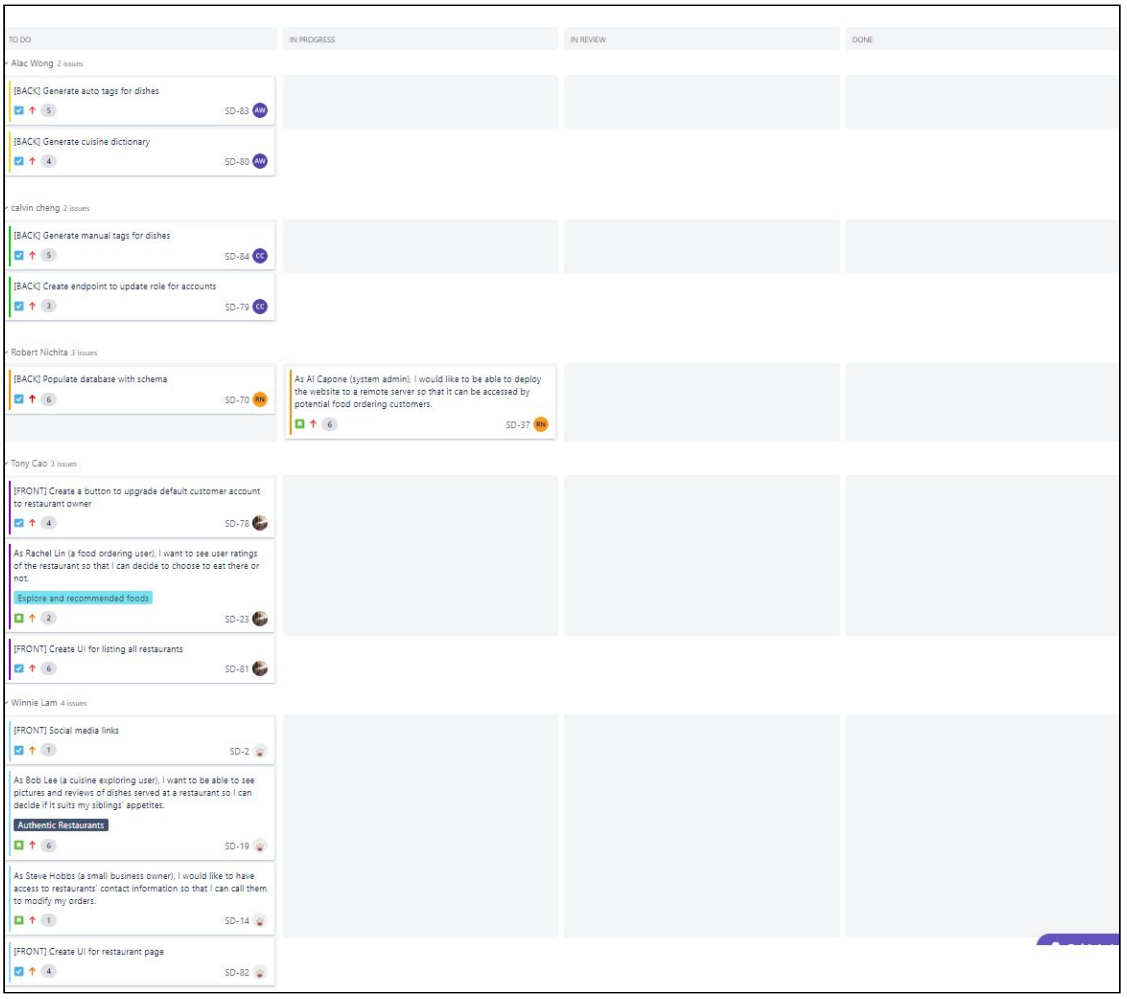
Below is a rough plan of what each team member during the second sprint:
- Winnie Lam: Created user interfaces for the various restaurant profiles (basic restaurant information and menu)
- Tony Cao: Created user interfaces for browsing all restaurants on the Scarborough Dining webpage (restaurant name, review and contact information), create button to upgrade basic users to upgrade button
- Calvin Cheng: Created manual tags for dishes to be used for the search engine in the feature, create endpoints to upgrade users from a basic role to a restaurant personnel role
- Robert Nichita: Worked on webpage deployment to ensure the webpage is deployed properly on a remote server
- Alac Wong: Generated a cuisine dictionary (keywords to cuisine type)

The order/priority of the tasks for the sprint will be as such (keeping in mind most tasks will happen concurrently):
1. Web Page deployment
2. Designing restaurant profile pages
3. Allowing users to upgrade from a basic user to restaurant personnel

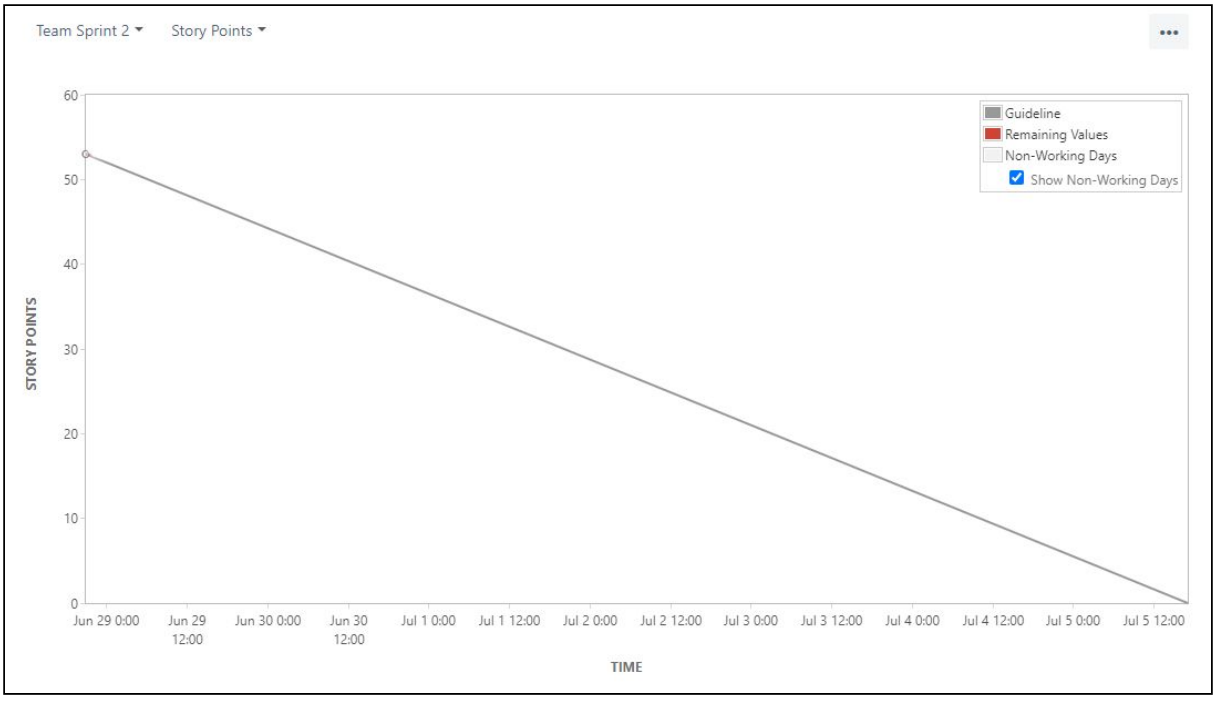
4. Creating user interfaces to let users browse all restaurants on Scarborough Dining
5. Creating tags that can be used as the foundation of the search engine

Legend
AW = Alac Wong
CC = Calvin Cheng
RN = Robert Nichita
TC = Tony Cao
WL = Winnie Lam

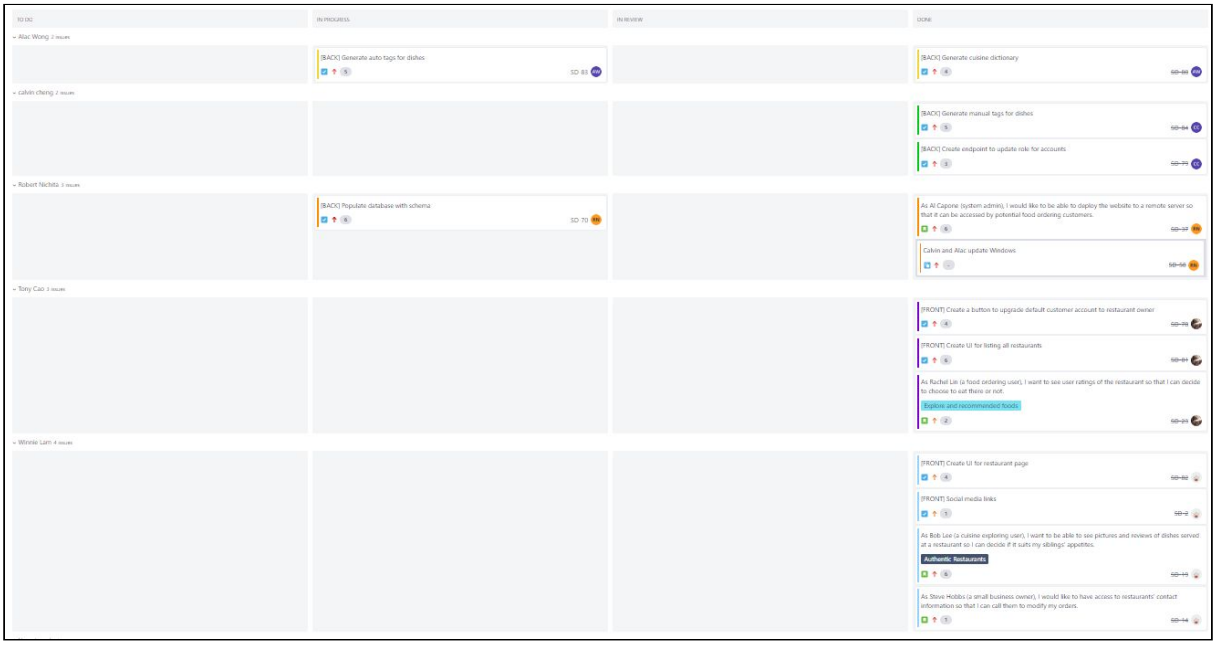
3.1.3 Initial Task Board



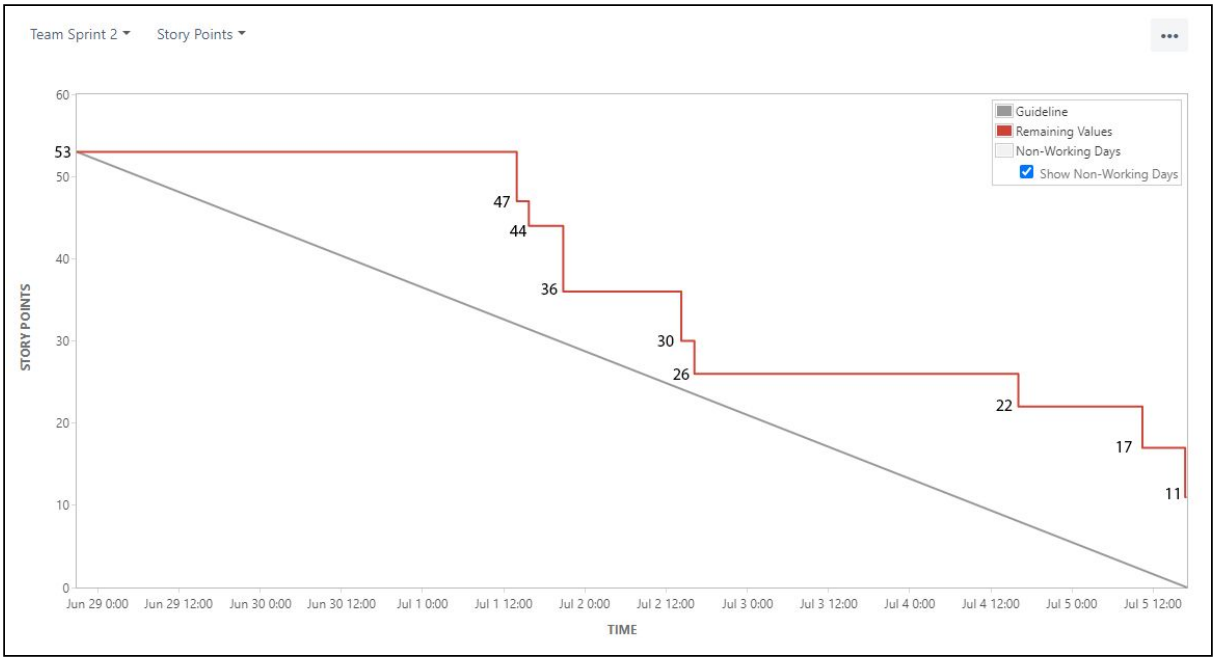
3.1.4 Initial Burn-Down Chart



3.1.5 Final Task Board



3.1.6 Final Burn-Down Chart



3.2 Sprint 3: July 5th to 12th

3.2.1 Sprint Backlog

Feature: *Restaurant Pages (Information)*

[FRONT] Populate the dishes listing tab with database information

- Points: 5
- Priority: High
- Acceptance Criteria:
    - Given the user switches to the dishes sub-tab, when they are on the browse tab, then the user should be able to see all the dishes from all restaurants

[FRONT] Design RO dashboard UI

- Points: 5
- Priority: High
- Acceptance Criteria:
    - Given that a food order user wants to view all orders on Scarborough Dining, when they login then they should be redirected to a page containing new and current orders and have an option to view all orders.

Feature: *Database Information*

[BACK] Create endpoint to get details of a restaurant

- Points: 2
- Priority: High

- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

[BACK] Create endpoint to get a list of all restaurants

- Points: 2
- Priority: High
- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

[BACK] Create endpoint to get a list of dishes

- Points: 2
- Priority: High
- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

[BACK] Populate database with schema

- Points: 6
- Priority: High
- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

[FRONT] Fix restaurant listing UI to populate with database information

- Points: 3
- Priority: High
- Acceptance Criteria:
    - Given that the customer wants to view a list of all restaurants, when they click the browse tab, they should be able to see a list of all restaurants that are in the database.
    - Given that the customer wants to view a specific restaurant, when they click the “view restaurant” on the restaurant card, it should direct them to the correct restaurant page using its id as a query.

[FRONT] Fix individual restaurant page UI to populate with database information

- Points: 3
- Priority: High
- Acceptance Criteria:
    - Given that the customer wants to view a specific restaurant, when clicking on the “view restaurant“ card in the listing, it should redirect the customer to the correct page using its id in the query.

[FRONT] Fix restaurant page's menu to populate with database information

- Points: 3

- Priority: High
- Acceptance Criteria:
    - Given that the customer wants to see what the restaurant has to offer, when they navigate onto the restaurant's page, they should be able to see what is offered on the menu and its categories.

Feature: *Restaurant Tagging (Search Engine)*

[BACK] Generate auto tags for dishes

- Points: 5
- Priority: High
- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

Feature: *Login and Signup*

[BACK] Create endpoint to get logged in user data

- Points: 3
- Priority: High
- Acceptance Criteria:
    - N/A as was task in the back end that did not involve user

[BACK] Modify sign up query to default to BU

- Points: 1
- Priority: High
- Acceptance Criteria:
    - Given that a request is made to sign up, the role of the created user profile should default to a basic user role.
    - Given that a user signs up through the auth0 signup UI, a request should be sent with the required information, and without the user role field

[FRONT] Fix login role assignment using user data

- Points: 2
- Priority: High
- Acceptance Criteria:
    - Given that a user wants to create an account when they sign up for an account then they should be given a default role of basic user provided.

[FRONT] Fix upgrade button to appear depending on the role

- Points: 2
- Priority: High
- Acceptance Criteria:

    - Given that the account logged in is a basic user, when they want to upgrade their account to a restaurant account as they belong to a restaurant then they should be presented with an option to upgrade their account.

3.2.2 Assignment of Tasks

Below is a rough plan of what each team member during the third sprint:

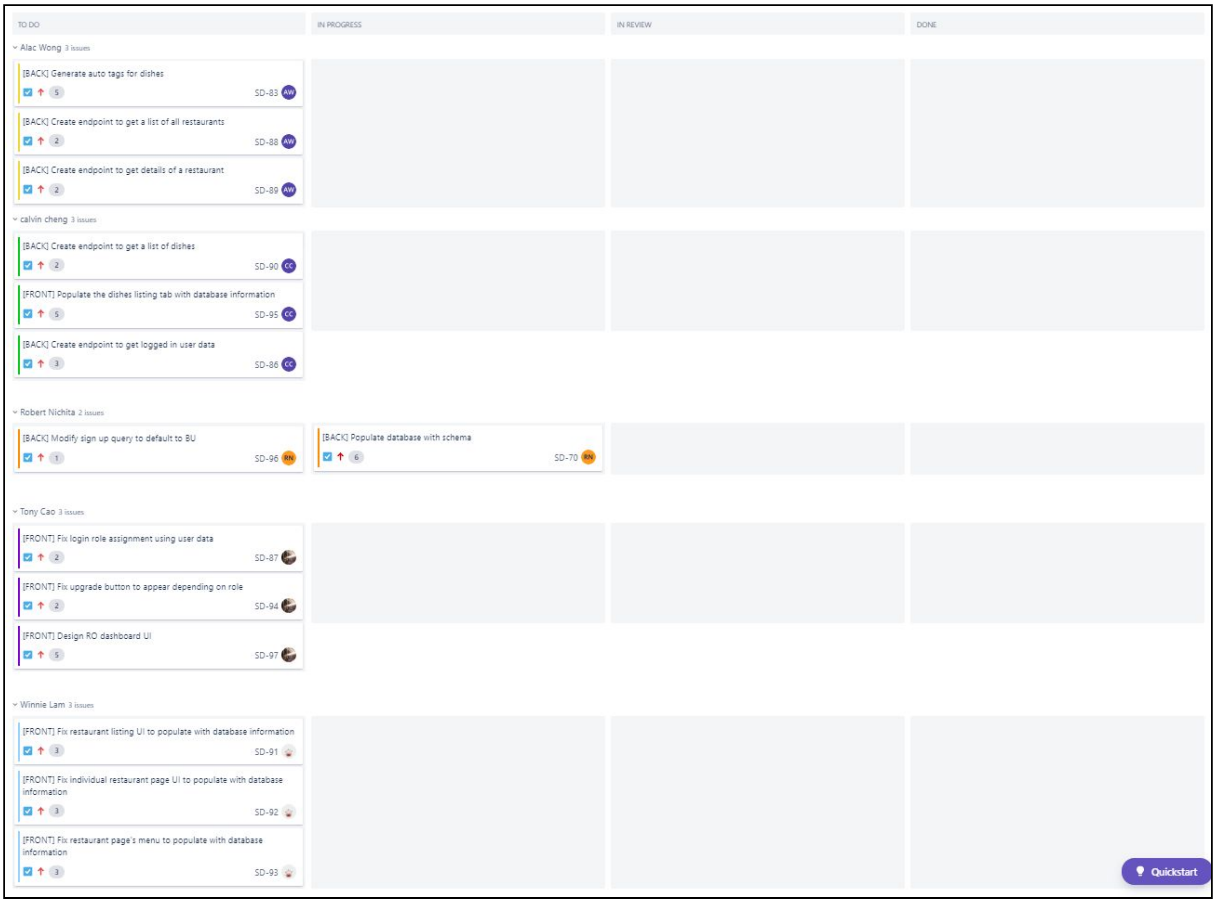
- Winnie Lam: Populate restaurants, dishes and restaurant profile pages with data in the database instead of running from local JSON files
- Tony Cao: Fix upgrade button to ensure that it only appears if the logged-in user is a basic user and create restaurant dashboard UI for viewing new, in-progress and all orders
- Calvin Cheng: Create endpoints to retrieve user information (mainly the role) from the logged-in user and to retrieve dishes stored in the database, create the page to view all dishes on Scarborough Dining
- Robert Nichita: Work on modifying signup query to default to basic user and populating the database with information relative to the database schema
- Alac Wong: Generate endpoints to get the list of all restaurants stored in the database along with corresponding restaurants details stored in the database and generate automatic tags for dishes

The order/priority of the tasks for the sprint will be as such (keeping in mind most tasks will happen concurrently):

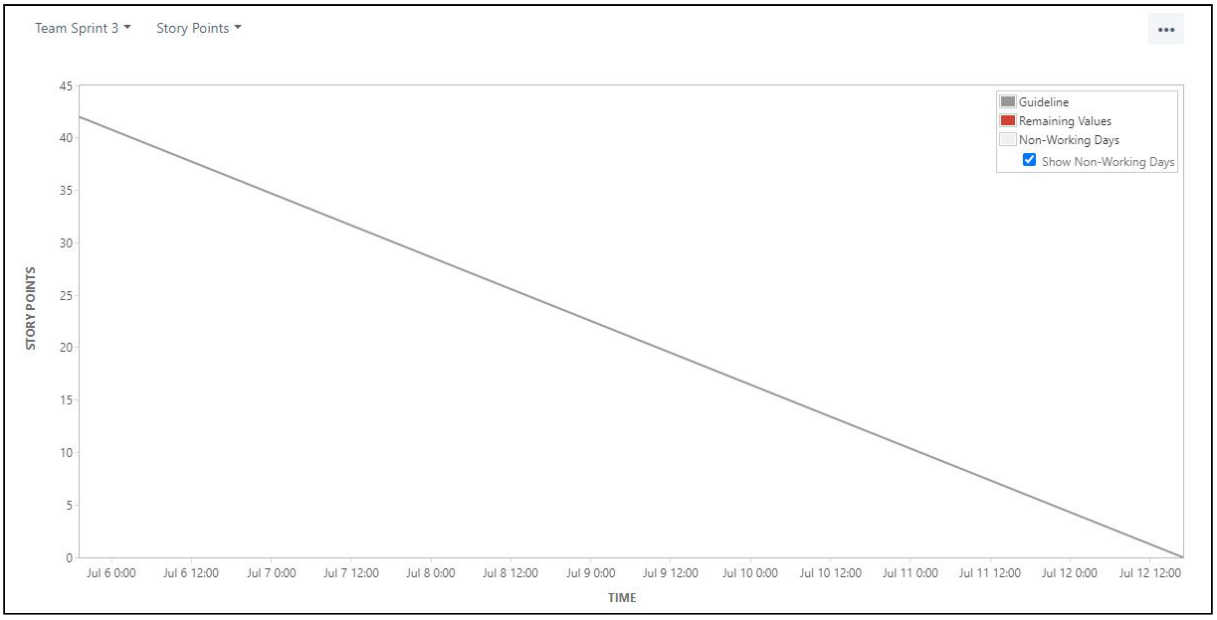
1. Setting up the database with mock data and loading it onto the webpage
2. Upgrading users to a restaurant personnel properly (no duplicate upgrading)
3. Generating tags for dishes

Legend
AW = Alac Wong
CC = Calvin Cheng
RN = Robert Nichita
TC = Tony Cao
WL = Winnie Lam

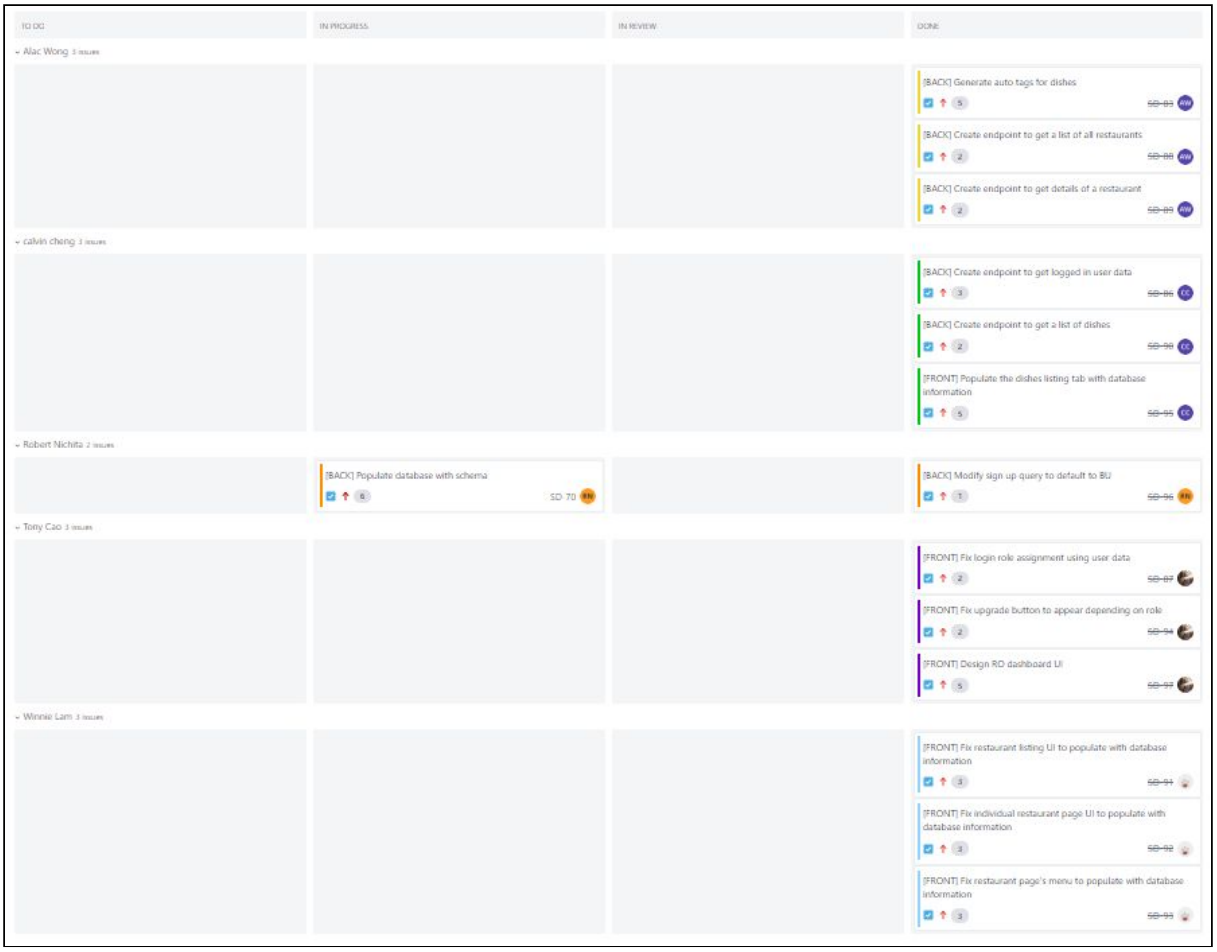
3.2.3 Initial Task Board



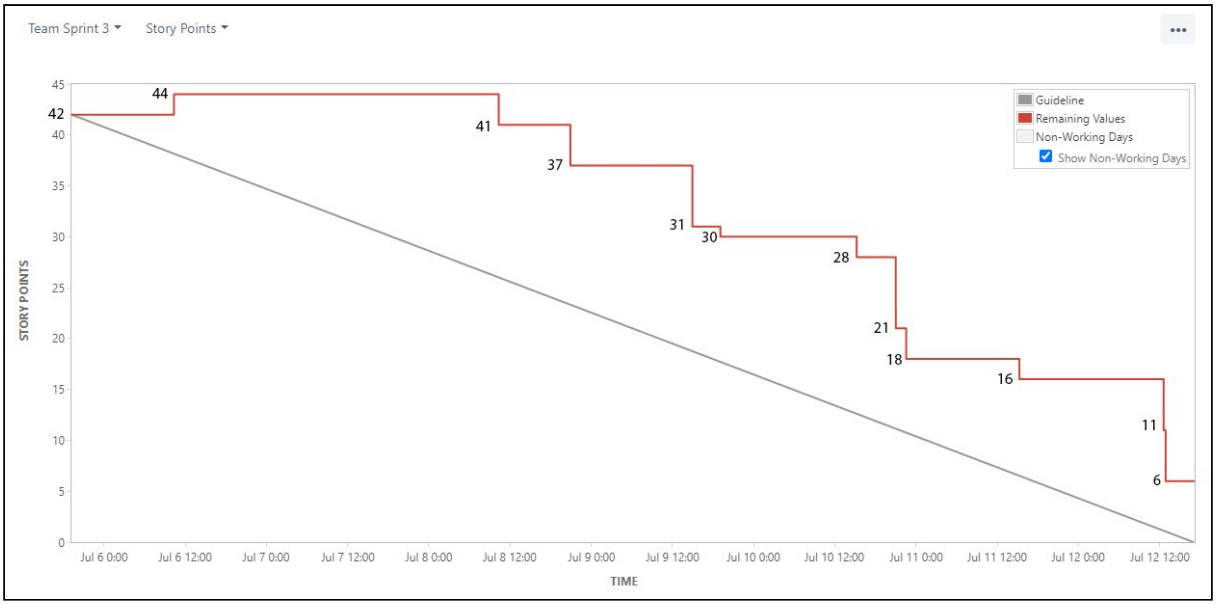
3.2.4 Initial Burn-Down Chart



3.2.5 Final Task Board



3.2.6 Final Burn-Down Chart



3.3 Sprint 4: July 12th to 19th

3.3.1 Sprint Backlog

Feature: *Login and Signup*

[BACK] Create endpoint to check if email exists in the database

- Points: 1
- Priority: Medium
- Acceptance Criteria:
    - Given a user email, when it exists, then True will be returned. Given a user email, when it does not exist, then False will be returned.

[FRONT] Create auth guard for RO

- Points: 2
- Priority: Medium
- Acceptance Criteria:
    - Given that the user is a basic user, when logging into the site, they should not be able to access or see the dashboard page.
    - Given that the user is a restaurant owner, when logging into the site, they should be able to access and see the dashboard page.

[FRONT] Fix error on upgrade account using email check

- Points: 2
- Priority: Medium
- Acceptance Criteria:
    - N/A as this did not affect the user it was just fixing a bug

[BACK] Google Cloud storage implementation

- Points: 9
- Priority: Medium
- Acceptance Criteria:
    - Given that the request includes a file and a valid user id, when this endpoint is called, upload file to google cloud storage and update the appropriate user's picture with the link acquired from google cloud storage.

Feature: *System Testing and Validation*

[BACK] Reconfigure endpoints

- Points: 2
- Priority: Medium
- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

[FRONT] Fix the endpoints based on the new names

- Points: 1
- Priority: Medium
- Acceptance Criteria:
    - N/A as this was required for the front end based off back end changed

[BACK] Create restaurant object on upgrade account

- Points: 3
- Priority: Medium
- Acceptance Criteria:
    - Given the user provides a valid email, role, and restaurant fields, when the role is RO then a new Restaurant will be created and connected to the user.

Feature: *Restaurant Profile*

[BACK] Create endpoint to edit restaurant object using restaurant id

- Points: 2
- Priority: Medium
- Acceptance Criteria:
    - Given a restaurant id and the parameters to change when the restaurant exists then the DB will be updated with the new fields for the restaurant.

[BACK] Populate database with schema

- Points: 10
- Priority: High
- Acceptance Criteria:
    - N/A as was a task in the back end that did not involve user

As Hames Jarden (a restaurant owner), I want to list and link my social media platforms so that I can connect with customers on different platforms.

- Points: 1
- Priority: Low
- Acceptance Criteria:
    - Given that a restaurant owner wants to list their social media platforms, when they fill in their social media handles (Instagram, Twitter, etc..) then they should be able to view and be directed to these links on their restaurant profile pages.

[FRONT] Create UI for restaurant initial set up forms

- Points: 5
- Priority: Medium
- Acceptance Criteria:

    - Given that a user wants to upgrade the account to a restaurant account, when they click confirm for “upgrade account” then they should be displayed with a page that is a fillable form of restaurant details (name, address, phone number, etc...) to set up their account.
    - Given that a restaurant personnel finishes filling in the form, when they click “save & next” then they should be brought to a page letting them fill in the information of the restaurant (i.e. about us page).

[FRONT] Create UI for adding menu items

- Points: 5
- Priority: Medium
- Acceptance Criteria:
    - Given that the restaurant owner wants to create their restaurant page, when they upgrade their account from BU to RO, they should be able to add items to their menu., including dish name, category, images, price, cuisine, dish information, and allergies.
    - Given that the restaurant owner adds a dish to their menu, when they add the dish, they should be able to delete it from their initial menu set up.
    - Given that the restaurant owner adds a dish to their menu, when they add the dish, they should be able to edit it from their initial menu set up.
    - Given that the restaurant owner is not ready to add dishes yet, when they reach the initial menu set up, they should be able to skip over that portion and have their page generated without a menu for now.

Sprint Administrative

[TASK] Deliverable 4 Report

- Points: 4
- Priority: High
- Acceptance Criteria:
    - N/A - this task was set up to acknowledge how large the deliverable 4 report is

3.3.2 Assignment of Tasks

Below is a rough plan of what each team member during our fourth sprint:

- Winnie Lam: Reroute web pages to ensure that users only access the needed pages based on their profile and work on creating pages to let restaurant owners upload menu items
- Tony Cao: Work on creating forms to ensure that when a user upgrades their account they are required to fill out a set of forms to create a record of the restaurant in the database
- Calvin Cheng: Work on creating endpoint to ensure we can store restaurant information and dishes when restaurant personnel upgrade their account

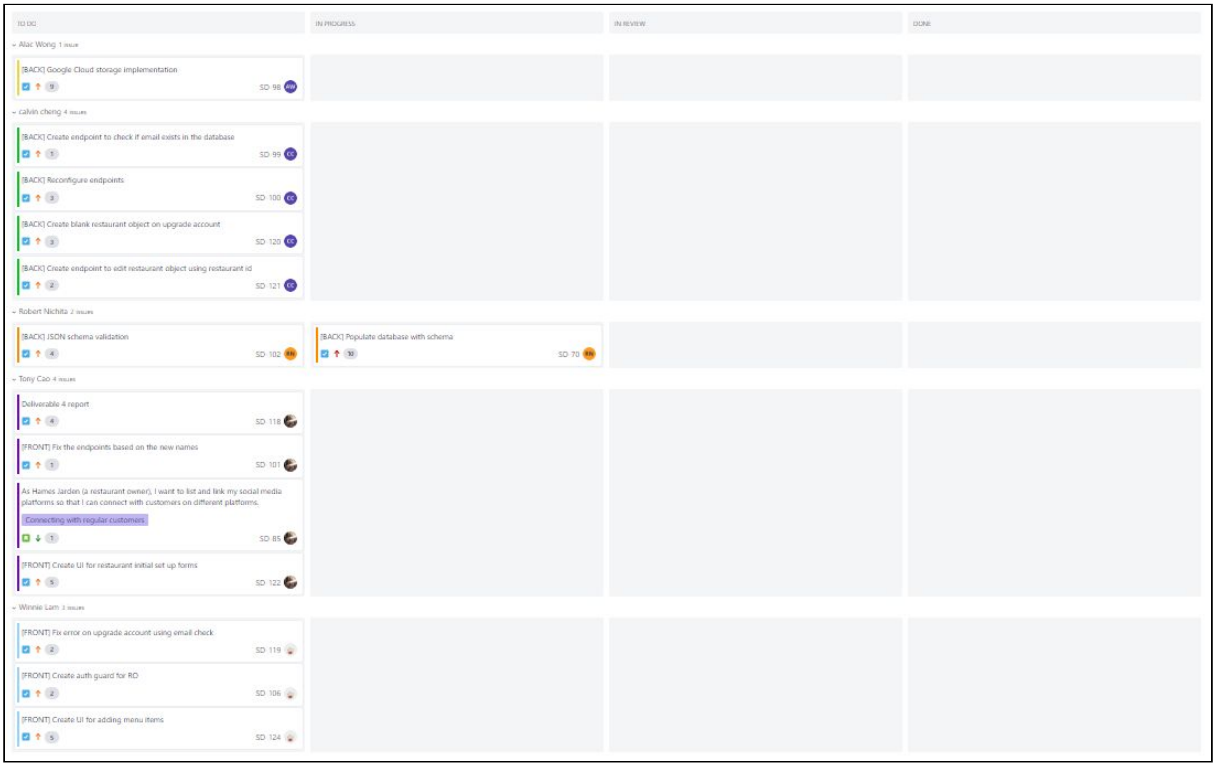
- Robert Nichita: Continue populating the database and validating JSON schema for requests
- Alac Wong: Explore uploading images option using Google Cloud storage

The order/priority of the tasks for the sprint will be as such (keeping in mind most tasks will happen concurrently):

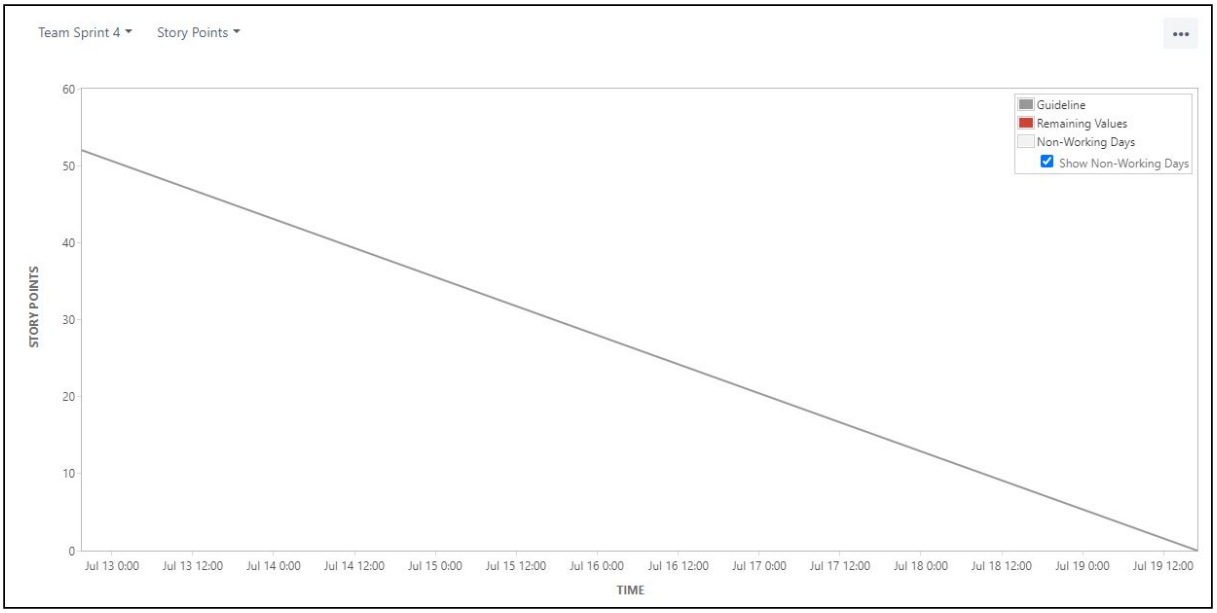
1. Letting restaurant personnel input restaurant information when they sign up
2. Letting restaurant personnel add menu items to their restaurant
3. Populating the database with fake information for testing purposes

Legend
AW = Alac Wong
CC = Calvin Cheng
RN = Robert Nichita
TC = Tony Cao
WL = Winnie Lam

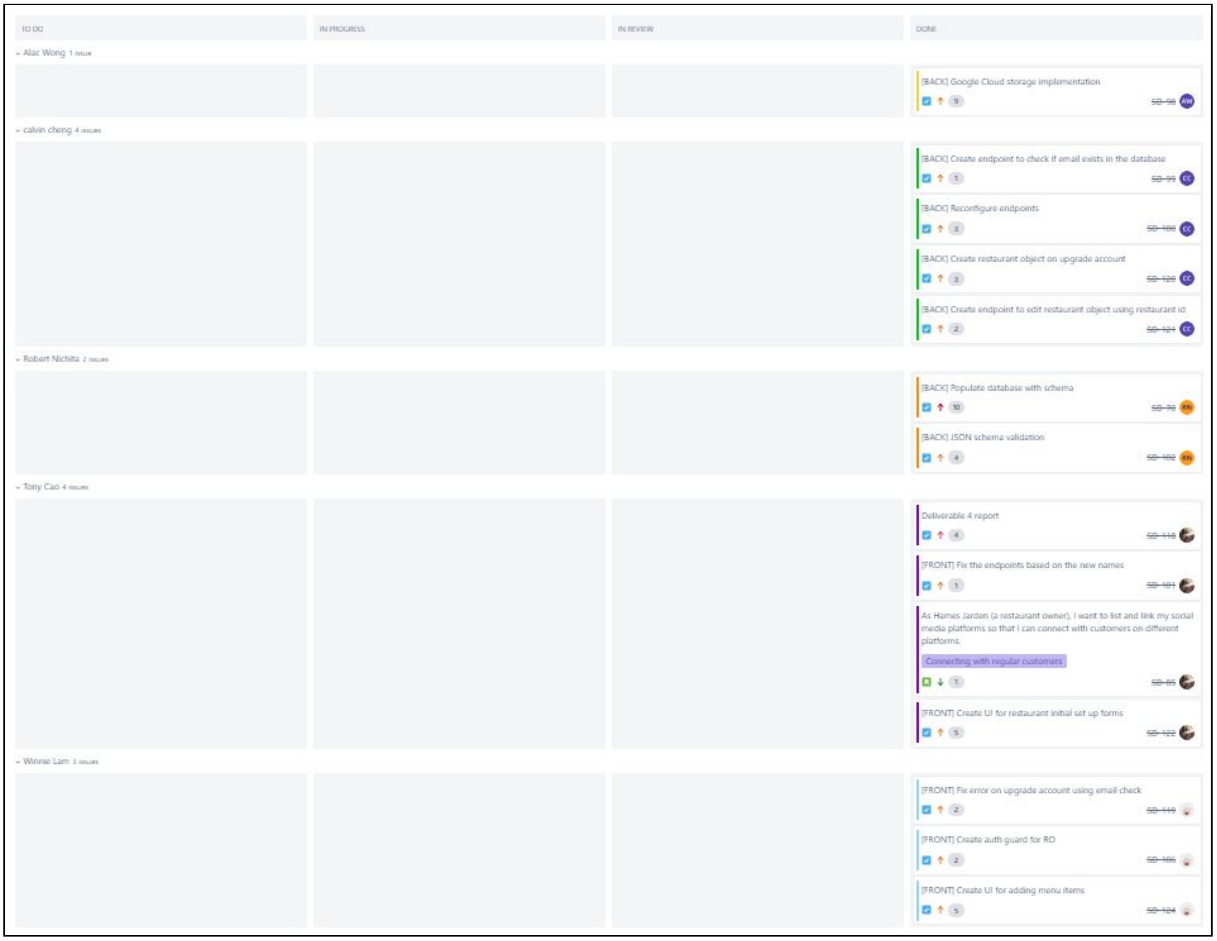
3.3.3 Initial Task Board



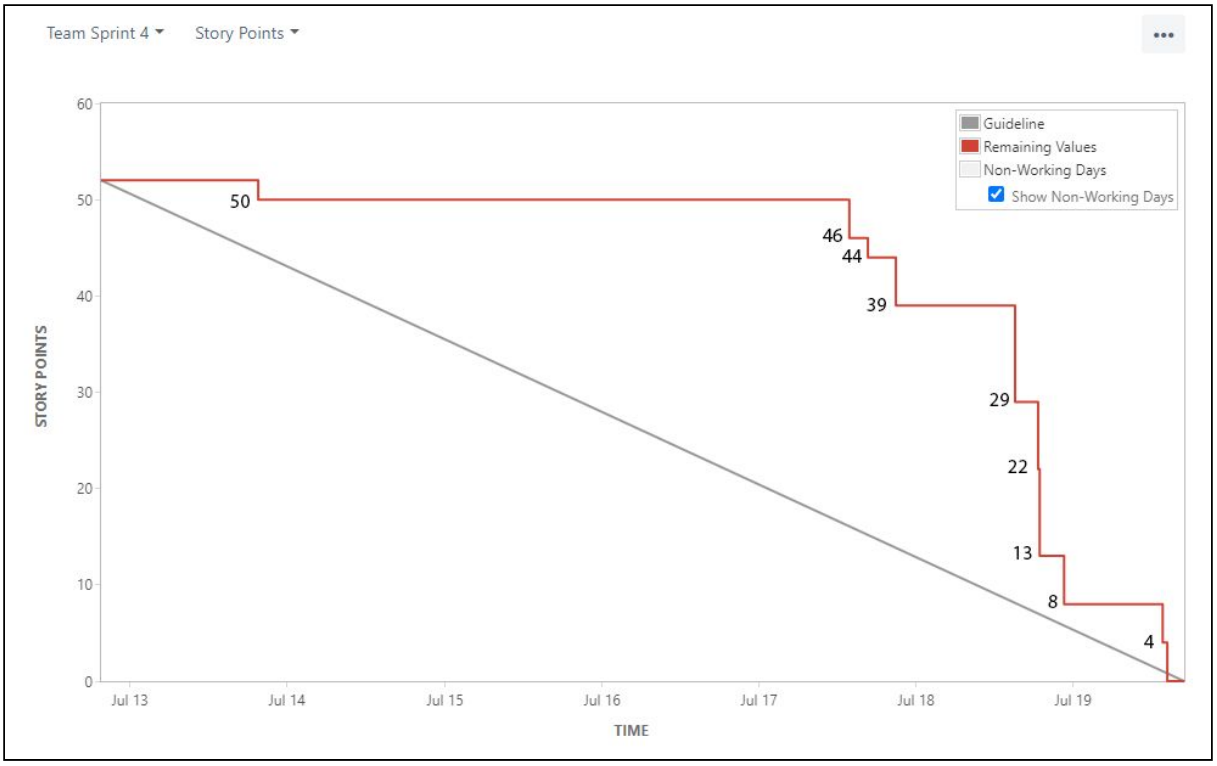
3.3.4 Initial Burn-Down Chart



3.3.5 Final Task Board



3.3.6 Final Burn-Down Chart



4 - System Components

4.1 Change in Components

In terms of change to our components, there is now a new geolocation component to reflect the request to add geolocation to the webpage to allow users to see their current location and nearby restaurants. In addition, a component that was changed was the media component. The media component has been changed to reflect the required functionality for restaurant personnel to upload images for their webpage.

4.2 Updated UML Diagrams

See attached file in our deliverable 4 folder called UML_Use_Case_Updated.pdf for our UML use case diagram for the Scarborough Dining system.

Updates to the UML diagram: A new actor representing the SCDining System administrators who may be using the server. This shows that the admins can dump information from the database, seed new randomly generated information into the database for testing reasons, and edit the static content served on the homepage.

4.3 Updated Component Diagrams

See attached file in our deliverable 4 folder called SCDining_Component_Diagram_Updated.pdf for our components diagram for the Scarborough Dining system.

Paragraphs for changes to the component design can be found below:

Geolocation component: There has been a request to add a Geolocation service to the application to allow users to see restaurants nearby on the map. We have not completed researching or designing interactions with this component. It remains a skeleton disconnected from the component diagram for this reason.

Media Component: The media component has been updated to reflect its planned structure within the system. When restaurant owners edit their menu or bio section, they will be able to upload images using a chunked upload interface onto the server. The server will then queue these uploads to the cloud media storage using its API to store them off of the server, then deleting the uploaded images from the server to save space. The images will then reside on the cloud where they can be accessed and downloaded from the frontend via a link generated by the API at the time of upload.

Below are the paragraphs for the remainder of the components once again:

Authentication component: This component is responsible for securely authenticating the user and then retrieving/storing user data to all other components except the database component. This component calls the Database component to retrieve data of a logged-in user or store data of a new user. This component will interact with both customers and restaurant personnel as they will be using the same login and sign up page but given access to only pages that their specific profile can view.

Database Component: This component is responsible for storing user data, restaurant data and maintaining a collection between restaurant pages and subsequent tags. All other components except the restaurant dashboard call this component to either store, retrieve, or update data. This component will interact with both customers and restaurant personnel as they will be viewing all the items on each page based on the database population.

Page editing component: This component is responsible for allowing a restaurant owner to successfully update their website. After updating their page, this component is then responsible for updating tags associated with the modified website. This component calls the Database component's restaurant table to update website data, and then update corresponding tags for updated websites. This component will only interact with restaurant owners as they

will be the ones who will be able to make changes such as menu changes, limited-time specials or profile changes such as phone number, address or operating hours.

Homepage Component: This component is responsible for providing the user with an attractive interface that features different restaurants and dishes to the user. This component calls the database component to get different restaurants and search for specials from the restaurant table. This component will interact with both customers and restaurant personnel since it will be the first page they land on when going onto the site.

Page Component: This component is responsible for providing the user with an interface to view a restaurant's page as well as providing interactions such as favoriting and reviewing that page. This component interacts with the database component by loading the page data as well as updating its user reviews from the restaurant table and updating a user's favorites from the user table. This component will interact with mainly customers only as this page will be providing useful information that a customer may want to know before placing orders such as what other users thought of the restaurants or which restaurants they have enjoyed in the past.

Filter Component: This component is responsible for providing the user with an interface for searching for different restaurants or dishes. This component calls the tag collection table from the database component to find relevant restaurants to display. This component will interact with customers only as this provides customers a way to narrow down a long list of restaurants to restaurants that either cater to their taste, budget or is within proximity.

Payment Component: This component is responsible for providing the user with an interface to pay for orders, track transactions and send data to the restaurant dashboard. This component calls the database component's transaction table to record transactions and sends payment order requests to the restaurant dashboard. This component will interact with both customers, restaurant personnel and financial institutions. When customers pay for an order, a transaction will be sent to the financial institution to process payment. Then upon successful payments, the financial institution will send the information back to the restaurant who can then accept the order and process it.

Restaurant Dashboard: This component is responsible for providing the restaurant owner with an interface to accept/decline orders. This component receives data from a web socket from payment components and sends data back to the payment component from its socket. This component will interact with the restaurant personnel since it allows restaurants to interact with orders placed by customers.

5 - Verification

5.1 Unit Tests

Documented test cases can be found in server/{app}/test**.py where app is the corresponding app to test case.

Specific apps, test suites, or even individual test cases can be run using the format: "python manage.py test App.Test_Suite.Test_Case" depending on how deep you want to go.

5.1.1 Testing Table Legend

Column	Column Description
Test Case Name	Name of Test Case
App	Django app test case is testing
Test Suite	Test Suite for test case
Evaluation Criteria	Criteria function must pass to pass the test case
(Possible Risk) Description	A description of possible risks associated with test case failure
Magnitude	Measurement of how dangerous possible risks associated with test case failure
Probability	Measurement of how likely possible risks may occur associated with test case failure
Priority	Priority of importance for a function to pass test case, priority is influenced by probability, magnitude and the function's priority (found in backlog) it is testing

5.1.2 Master Testing Table

Our nicely formatted master testing table in a Markdown format can be found in our repository in the following section: team_08-project/sd-site/docs/backend/backend.md.

However, below is an export of that table (in the format of screenshots due to the number of columns in the table:

Test Case Name	App	Test Suite	Evaluation Criteria	(Possible Risks) Description	Magnitude	Probability	Priority
test_signup	user	SDUserTestCases	Checks to see if The User has been inserted into the database	No new Users can be made	High	High	High
test_signup_invalid_role	user	SDUserTestCases	Checks to see the proper Validation Error is thrown	Anything can be used as a role	High	Low	Medium
test_reassign_RO_to_BU	user	SDUserTestCases	Checks if the role change to BU is reflected in the database	Users won't be able to be 'demoted'	Low	Low	Low
test_reassign_BU_to_RO	user	SDUserTestCases	Checks if the role change to RO and new restaurant id is reflected in the database	New ROs won't be able to be created	High	Low	Medium
test_data	user	SDUserTestCases	Checks the user data returned is the matching the queried user	Display incorrect data for users	High	High	High
test_exists_true	user	SDUserTestCases	Checks if an existing user exists	Users will be unable to upgrade	High	High	High
test_exists_false	user	SDUserTestCases	Checks if a non-existing user exists	Unable to sign up users	High	High	High
test_add_randomizer	utils	UtilityTestCases	Adds a new ["key": randomization_function()] pair into the seeder dictionary	Incorrectly add new randomization functions to seeders	Low	Low	Low
test_gen_rand_dict	utils	UtilityTestCases	Generates random data based on given seeding dictionary	Inability to randomly generate data for seeding	Low	Low	Low
test_clean_dict	utils	UtilityTestCases	Cleans invalid randomization functions (Non JSON-Encodable outputs) from a given seeding dictionary	Potentially broken seeding scripts with functions that produce non JSON-encodable outputs	Low	Low	Low

Trello

test_clear_tags	restaurant	RestaurantTestCases	Tag ids are correctly purged from a Food document	Tags remain in database which may result in referencing non-existent tag documents	Medium	Medium	Medium
test_clear_foods	restaurant	RestaurantTestCases	Food ids are correctly purged from Tag document	Foods remain tagged which may result in search engine displaying non-existent/incorrect documents	Medium	Medium	Medium
test_food_ids	restaurant	RestaurantTestCases	Tag ids are correctly updated from tagging an existing Tag document	Food will not be associated with subsequent tag which may cause incorrect search engine results	Medium	Low	Medium
test_tag_ids	restaurant	RestaurantTestCases	Tag ids are correctly updated from tagging an existing Tag document	Restaurant owners will be unable to tag their dishes resulting in search engines skipping their dishes	Medium	High	Medium
test_tag_creation	restaurant	RestaurantTestCases	Tag document is correctly generated upon tagging with a "new" tag word	New Tag documents will not be generated, resulting in limited search engine results	Medium	Low	Medium
test_foods_already_tagged	restaurant	RestaurantTestCases	Food ids are not duplicated upon tagging an already tagged (Food, Tag) couple	Duplicate food ids take up extra space in the database and slow down querying	Low	Low	Low
test_tags_already_tagged	restaurant	RestaurantTestCases	Tag ids are not duplicated upon tagging an already tagged (Food, Tag) couple	Duplicate tag ids take up extra space in the database and slow down querying	Low	Low	Low
test_auto	restaurant	RestaurantTestCases	Correct Tag document correctly automatically generated based on Food's description	Search engine results become slowly reliant on user input and cannot provide robust results to the user	Medium	High	Medium
test_get_all_foods	restaurant	RestaurantTestCases	All food documents within the database are correctly retrieved	Frontend will be unable feature dishes on the homepage	Medium	High	Medium

test_find_restaurant	restaurant	RestaurantTestCases	Correct restaurant document is retrieved given primary key 'id'	Frontend will be unable to documents associated with that specific restaurant such as dishes and users	High	High	High
test_find_all_restaurant	restaurant	RestaurantTestCases	All restaurant documents are retrieved from database	Frontend will is unable to display restaurant data	High	High	High
test_insert_restaurant	restaurant	RestaurantTestCases	Given restaurant data, restaurant document is inserted into database representing said data	New restaurants cannot be added to the database	High	High	High
test_edit_restaurant	restaurant	RestaurantTestCases	Given new restaurant data, restaurant document is updated to represent new data	Restaurant data becomes static and cannot be changed by restaurant owner	Medium	Medium	Medium

6 - Validation

6.1 Meeting Minutes

All members were present when meeting with the client. The main topics that were discussed during these client meetings were the existing design/theme of the page, the login and signup of users, current format for displaying restaurants and designs for future features that will be implemented. We first demoed our deployed product in a breakout room to give them a sense of how the deployed web page looked. Then we moved on to showing Figma designs to ensure that the user interfaces that we are going to be implementing in the feature are what they expected. Lastly, we showed them our current product backlog and our priority of user stories to ensure that we were working on the most wanted features and functionality first before working on extra features that would be nice to have.

6.2 Demos Produced

During our client demos, we went through a walkthrough of the currently deployed web page to show them a sense of how the live webpage currently looks like. We walked through the login and signup procedure and the different aspects of the homepage. Afterwards, we showed our Figma designs to the clients regarding the design of new user interfaces that are to be implemented in our future sprints. Figma designs are attached in a file called “Team Aqua - Scarborough Dining (Customer).pdf” and “Team Aqua - Scarborough Dining (Restaurant Owner).pdf”.

6.3 Questions Prepared for Clients

Below is a list of questions that we prepared for clients to ask about regarding our overall design and plans:

- What details should we put on each dish card? Ex: do we want price, review, description
- Do we want the webpage focused more on food purchasing or displaying Scarborough restaurant owners?
- How is our current theme? Are the colours, fonts and pictures well suited for the Scarborough dining platform?
- How does our current website design reflect your expectations? Do we need to rebrand the webpage differently?
- How do we incorporate more of the ability to showcase Scarborough restaurant owners?
- What would you say is a higher priority/focus, displaying Scarborough restaurants and their history (owners similar to 150neighbours.ca) or displaying the actual dishes and various cuisines available in Scarborough?
- Is our current priority of tasks in our product backlog good? Do any to be prioritized higher?
- Will we be provided with any restaurant information for our database?

The questions we prepared mainly helped us ensure that we were building a web page that was the client's vision and that it has all the added functionality they were expecting.

6.4 Feedback From Clients

The clients provided lots of insightful feedback on how to enhance our current design, the feedback we got was:

- Prioritize and add functionality for restaurant owners to edit pages, add their restaurant information and dishes
- Ensure the webpage is optimized so that it can be viewed on laptops, tablets and phones
- Prioritize the ability to look up restaurants by geographical locations (i.e. closest restaurant to current GPS location)
- Always show the current address of the user in the navigation bar, applicable to only the food orderers
- Ensure that the database data can be exported as an excel flat file or similar formats

Overall, the feedback from clients helped us plan our sprint backlog betters as they wanted to prioritize restaurant owners being able to edit pages, look up restaurants on a map and showcasing more about owners than about ordering food.

7 - Retrospection

7.1 How did your project progress from deliverable 3 to deliverable 4?

Compared to the sprints that lead up to deliverable 3, we found that we were able to get more coding done in both the front and back end as the frameworks were fully set up. In addition, the more sprints that went our team got a clearer picture of who is working on what issues (i.e. frontend vs. backend) and what tasks each member can take on due to increased skills in that. For instance, Tony was able to do more frontend related works as he was getting a better grasp of angular compared to initial sprints. Overall, we also found that we were beginning to work more efficiently as a team as we were able to reduce sprint planning meetings from about 3 hours in total to roughly 1 to 1.5 hours.

7.2 Project Velocity

Sprint 2 started with 53 story points but ended with 42 points completed. There 11 points worth of incomplete tasks came from the task of populating the database and performing the auto-tagging of dishes.

Sprint 3 started with 42 points (due to midterm season) and ended with 38 points completed. After sprint 3 started, a task was re-pointed from 3 to 5 bringing the total sprint point to 44. There were 6 points worth of incomplete tasks that arose from the population of the database as it was discovered the task was longer than expected.

Sprint 4 started with 52 story points and ended with 52 points complete. This was the first sprint that our team managed to finish successfully with all tasks scheduled in the sprint completed. These may be a result of having more experience with pointing tasks and each member knowing what tasks they are capable of completing within the sprint.

7.3 Did you follow your plan(s) exactly?

We did not follow our sprint plans exactly which was expected as we began entering the midterm season. However, we did not worry too much as we know that agile allows for change so we were sure that everything would work out properly in the end. For instance, we had tasks that we were unable to complete in one sprint (mainly due to team member's availability) so they had to be pushed to future sprints. One thing that we managed to avoid was reassigning tasks, for each sprint backlog team members were given tasks that were completed by them. This helped reduce the number of times we had to reassign task xyz from member abc to member def and so on for the original tasks that member def was working on. We did manage to have one sprint that almost followed our plan exactly as we managed to finish all tasks in the task board for that sprint and burndown to zero story points.

7.4 What difficulties have you encountered?

As expected there were a few difficulties that were encountered in the sprints that took place between deliverable 3 and deliverable 4. Our first difficulty was that in some situations we overestimated the number of story points per sprint that team members could handle. A reason for this was because many midterms were occurring between deliverable 3 and deliverable 4. This resulted in pushing some tasks from one sprint to the next sprint as there was simply not enough time available to get them done. Another difficulty that we faced was when pointing tasks. Occasionally there would be tasks that were underestimated as the team may not have been familiar with the technology needed to implement that or understand the complexity of the task. This created “burnups” in our burndown chart as a task would have 2-3 points more than it was originally estimated to have.

7.5 Was your contingency plan useful at that point or did you have to come up with a new solution?

Luckily, during the sprints that occurred between deliverables 3 and 4, there were not too many major issues for team members. As a team, we acknowledged the busy schedule that midterms would cause so we began assigning fewer tasks to ensure that team members will be able to finish that task by the end of the corresponding sprint.