University of Toronto Scarborough

Team Environment, Release & Sprint Plan

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 - Team Environment

1.1 What tools will you use for your task board?

In the team agreement, we stated Trello would be our main task board tool. However, we have decided to move to Jira as it comes with free automation tools for managing agile life cycles. Both applications were developed by Atlassian so it will make transitioning from Trello to Jira much smoother.

1.2 What tools will you use for your burn-down chart?

We will use the same tool for the burn-down chart that we are using for the task board, Jira.

1.3 Who will maintain the burn-down chart? How?

The generated burn-down chart in Jira will be reviewed and finalized by Winnie as she is the team's scrum master and has experience using Jira.

1.4 What is every team member's role?

Winnie Lam: Scrum Master
- Facilitate stand-up meeting, group meetings, and sprint planning
- In charge of the backlog and burndown chart

Tony Cao: Team Coordinator
- Prepare final reports and slides for deliverables
- Responsible for meeting minutes

Calvin Cheng: Product Owner
- Communication of information between team and clients

Robert Nichita: Technical Lead
- Responsible for planning, coordination, and specifications development
- Dictates which development techniques should be used and sets the team coding standard

Alac Wong: Architectural Lead
- Responsible for architecture decisions such as framework choice and overall solution design

1.5 What tools, if any, will you use for communication?

Our team will be using Discord, Jira and Microsoft Teams for communications.

1.6 When do you plan to meet in person?

We do not plan to meet in person due to the COVID-19 pandemic along with its physical distancing requirements. In addition, it will be more convenient to meet virtually due to the locations of various team members. We will be having daily stand-up calls at 10 pm on Discord that will last for approximately 15 minutes with notes or recaps being posted on Discord afterwards.

1.7 How will you use your repository on GitHub?

In our GitHub repository, we will be utilizing the master, development and feature branch design. All features will be made in feature branches and on completion will be merged into the development branch. There will be one separate feature branch per task id, with the naming convention of *TASK-ID/branch-name*. When the feature branch is merged and no longer has a use, it will be deleted. Once everything is working in the development branch, it will get pushed to the master branch the day before the deadline.

The rule for what is committed is to mainly utilize the auto-generated .gitignore file by Angular and Django. Whatever is specified in those files, we do not commit. Commit messages should be in the form of *TASK-ID: what was completed*. The only time a team member is allowed to commit directly to the master branch is when uploading snapshots of the task boards along with the burndown chart and deliverable materials (reports and slides). All other commits that fall outside of the above mentioned two commits must be done on the feature branches and not directly on the development or master branch. These commits will then be merged into the development or master branch only by pull requests.

We will be using Docusaurus to keep documentation for both frontend and backend work. The code for the documentation site is kept in the repository.

1.8 Which machines will be used for development by each team member? E.g., a CMS virtual machine, a Linux laptop, a Windows home computer, etc.

Each team member will only be expected to use their regular computer used for school work. Here is a list of team members and their respective computer machines:
Winnie Lam: Windows 10 (Desktop) and Apple Macbook Pro (Laptop)
Tony Cao: Apple Macbook Pro Laptop
Calvin Cheng: Asus VivoBook (Windows 10 Laptop)
Robert Nichita: Lenovo T580 Laptop - Windows 10 Home Edition - Insider Program
Alac Wong: HP Laptop Windows 10

1.9 What is your DoD (definition of done)?

From our team agreement, code is classified as "done" if it passes all the test cases created and approved for that particular snippet of code. In addition, the code is reviewed by at least one other team member and is documented so that other team members can understand the code. Any pull requests are also merged. Lastly, a feature is "done" when it is verified with the business client that it meets their requirements.

2 - Release Plan

We will have weekly sprints that 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. These shorter sprints also make planning easier and encourages the team to break down stories or features into smaller chunks. 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.

3 - Product Backlog

For our interactive product backlog, see the Jira backlog at this link.

Feature: *Login & Signup*

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

- 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.

- 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.

- 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: Medium

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.

- 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

4 - Sprint Plan

4.1 Assignment of Story Points and Priority

The story points for each task are calculated on a scale of 10, with 1 point being a relatively easy task and 10 points being a difficult task. An example of a task with 1 point was uploading a restaurant's contact information (address, email, phone number, etc.) on the webpage. We determine the points by how comfortable the team members are with the topic and how much research will need to be put into it. After determining that, we consider implementation and how long or complex it might be.

A set of high priority features were given by the clients. When selecting tasks for each sprint, we will prioritize those high priority features that relate to one another. This is so that the larger component of what is high in priority will be completed as a whole.

4.2 Sprint Backlog

For our interactive sprint 1 backlog, see the Jira backlog at this link.

Feature: *Backend Set Up*

[Task] Set up Django framework on Docker
- Points: 4
- Acceptance Criteria:
  - Given that the developer wants to run the project locally, they should be able to run the Django server from their computers by following instructions in the README for installation and run.

[Task] Set up Angular framework on Docker
- Points: 4
- Acceptance Criteria:
  - Given that the developer wants to run the project locally, they should be able to run the Angular client-server from their computers by following instructions in the README for installation and run.

[Task] Set up MongoDB
- Points: 4
- Acceptance Criteria:
  - Given that the developer wants to create a database to store data, the database has been initialized in the cloud and the Django server can interact with the database while deployed to AWS.

Feature: *System Design*
[Task] UML diagrams of system functions - 5 points
- Points: 5
- Acceptance Criteria:
  - Given that a developer sees the UML diagrams, they should be able to determine the dependency relationships between various components of the design.

Feature: *Front End - Customer Pages*
[Task] Create UI for Home Page - 6 points
- Points: 6
- Acceptance Criteria:
  - Given that the customer enters the website, they should be able to navigate on the homepage of the site without signing in. The user should be able to see the

option to log in, sign up, search, displays of some dishes, some restaurant owner stories, and a footer.

[User Story] As Janet Jackson (a healthcare worker), I want to pay in debit/credit, so that I can reduce the spread of germs that comes with handling cash.
- Points: 3
- Acceptance Criteria:
  - Given that Janet has pressed the pay button on her order, she should be able to see a window that allows her to input payment information (debit/credit info).

Feature: *Log in and Sign-Up*

[User Story] As a customer, I want to be able to create an account so that I can buy my food online. *(6 points in total)*
- [Task] Create UI for signup
  - Points: 3
  - Acceptance Criteria:
    i. Given that the customer is presented with a signup page, they should be able to enter their personal information such as name, address, email, and password.
    ii. Given that the customer clicks signup when all the required fields are filled, the customer should be redirected to the main page with their new account signed in.
    iii. Given that the customer does not fill in all the required fields before clicking sign up, a warning should appear and request the user to fill in the required fields.

- [Task] Store user data from account sign up
  - Points: 3
  - Acceptance Criteria:
    i. Given that the customer presses the sign up button after filling in the required information, the new user data should be stored into the database for future consumption.

[User Story] As a customer, I want to be able to log in so that I can access my ordering information. *(9 points in total)*
- [Task] Encryption and saving user login info
  - Points: 6
  - Acceptance Criteria:
    i. Given that the customer clicks login when the required fields are filled out, the page should reflect the user's specific account with their information retrieved from the database.

- [Task] Create UI for login
  - Points: 3
  - Acceptance Criteria:
    i. Given that the customer is presented with the login page, they should be able to input their username and link to existing social media accounts.
    ii. Given that the customer clicks on log in when the fields are filled out, they should be redirected to the main page with their account signed in.
    iii. Given that the customer clicks on log in when a field is missing, an error pop up should appear telling the user to enter in the required fields.
    iv. Given that the customer clicks on login when at least one of the fields are incorrect, a warning should appear telling the user to ensure they have entered the correct credentials.

Feature: *Website Deployment*

[User Story] 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
- 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).

4.3 Assignment of Tasks

Below is a rough plan of what each team member during our first sprint:
- Winnie Lam: Set up Angular framework on Docker (2) and create the user interface for the homepage (5)
- Tony Cao: Create the user interface for the signup (6), login (9) and debit/credit card payment page (11)
- Calvin Cheng: Design the UML diagram for the various systems (4) and work on storing user data from when they sign up for an account (7)
- Robert Nichita: Set up Django framework on Docker (1) and work on deploying the website onto a remote server (10)
- Alac Wong: Set up MongoDB (3), work on encryption and save user login information (8)

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

1. Set up Angular framework, set up Django framework, set up MongoDB
2. Design UML, deploying the website
3. UI for the home page
4. UI for the signup and login, store signup and login data into the database, encryption of sensitive information
5. UI for the debit/credit card payment page

Iteration Plan:

Task ID	Cost	Priority	Mon (D1)	Tue (D2)	Wed (D3)	Thur (D4)	Fri (D5)	Sat (D6)	Sun (D7)	Dev
1	4	1	1	2	1					**RN**
2	4	2	1	2	1					**WL**
3	4	3	1	2	1					**AW**
4	5	4	1	2	2					**CC**
10	6	5			1	2	2	1		**RN**
5	6	6			1	2	2	1		**WL**
6	3	7	1	2						**TC**
7	3	8				1	2			**CC**
8	6	9			1	2	2	1		**AW**
9	3	10			1	2				**TC**
11	3	11					1	2		**TC**
Total	47		5	10	9	9	9	5	0	
Work Left	47		47	44	27	24	21	21	0	

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

4.4 Initial Task Board

4.5 Initial Burn-Down Chart
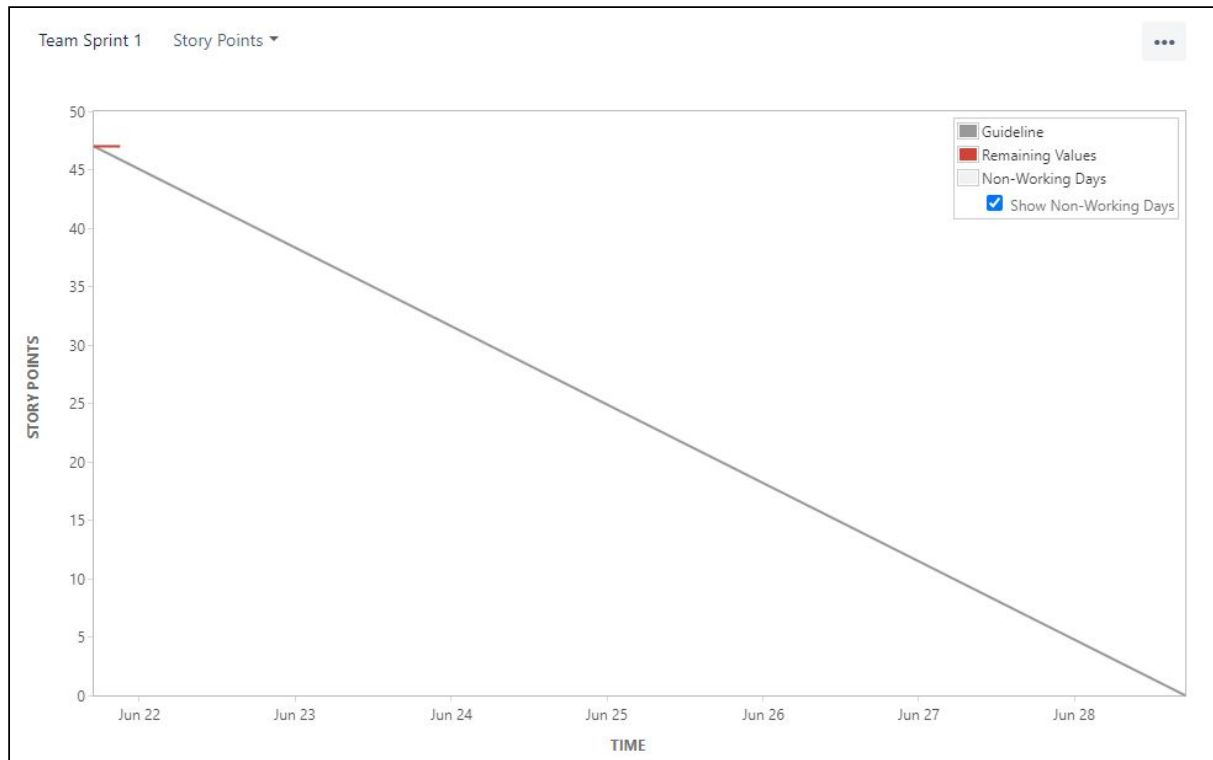
5 - System Components

5.1 UML Diagrams

See attached file in our deliverable 3 repo called UML_Use_Case.pdf for our UML use case diagram for the Scarborough Dining system.

5.2 Component Diagrams

See attached file in our deliverable 3 repo called SCDining_Component_Diagram.pdf for our components diagram for the Scarborough Dining system.

Paragraphs for each component can be found below:

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 third party authentication service auth0 to securely authenticate the user and then 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. This component uses mongodb atlas as its database and uses google cloud storage to store media files such as images and videos where links to corresponding media are found in mongodb.

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.

Media Component: This component is responsible for providing the user with an interface to view and interact with a restaurant's discussion board. This component calls the user table inside the database component to determine permissions for modifying different elements of the page and calls the restaurant component to modify page data per user interaction. This component will interact with both customers and restaurant personnel as they will both be able to view and interact with the discussion board.

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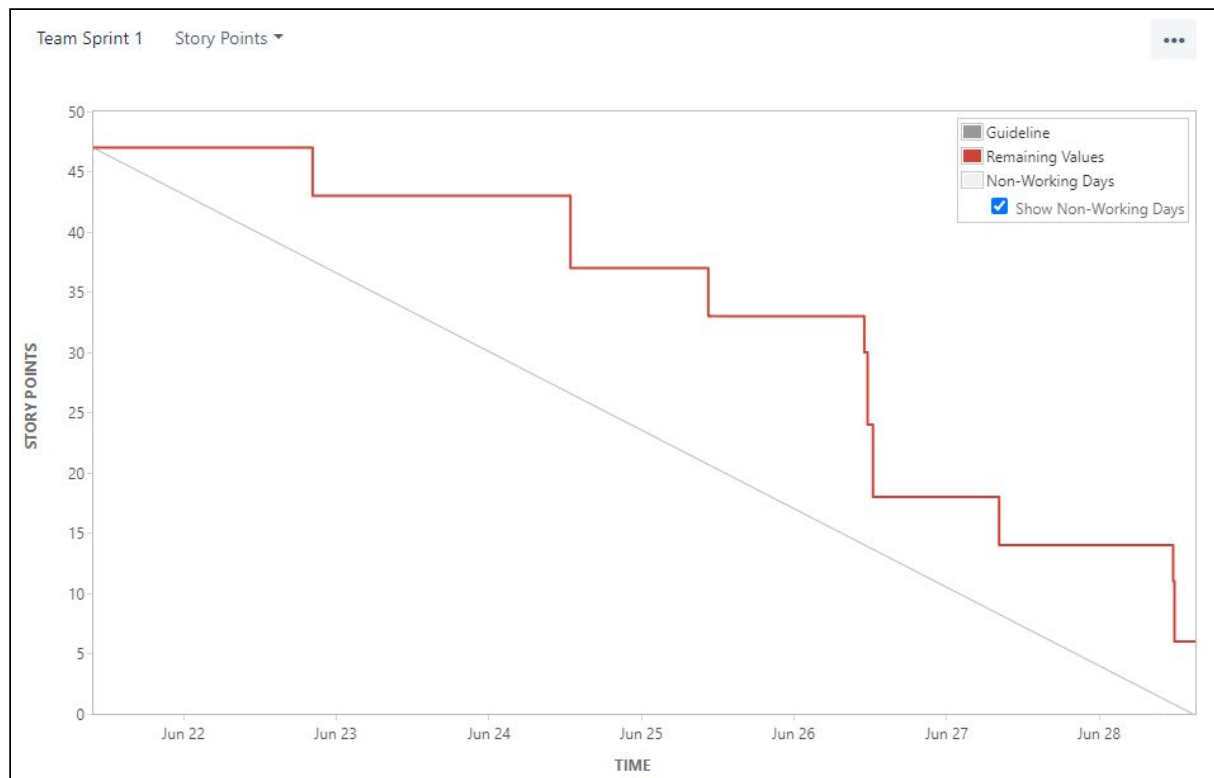
Restaurant Dashboard: This component is responsible for providing the restaurant owner with an interface to accept/decline orders. This component receives data from a web socket from payment components and sends data back to the payment component from its own socket. This component will interact with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

6 - End of Sprint 1 (June 28)

6.1 June 28 Task Board

6.2 June 28 Burn-Down Chart

6.3 Product Backlog Changes

Upon the completion of the sprint, there were a few changes to the product backlog.
The first change was regarding user sign up and log in. Initially, we had planned to use a 3rd party service called Auth0 in the backend to facilitate user sign up and login. However, we moved Auth0 to the front end as we realized it would be simpler to integrate into our project if it was done in the front end. In addition, as a result of using Auth0, we were able to remove the user story of resetting passwords and Auth0 facilitates that independently so there is no extra work on our ends. The second change we made was that we added a new set of user stories to the product backlog. We added a new set of user stories pertaining to how restaurants can accept orders from customers. Particularly the fact a dashboard needs to be built for them to accept or cancel orders, mark orders as complete and view all orders placed through Scarborough Dining.

7 - Retrospection

7.1 How did your project progress from deliverable 2 to deliverable 3?

From deliverable 2 to deliverable 3, we managed to create the first release of the project. This deliverable was focused more on the actual sprint planning and coding of the project. The transition was not problematic after the initial setup of the tech stack was completed.

7.2 Project Velocity

The estimated project velocity for this sprint was to complete 47 story points in one week. The actual project velocity was 41 points, with deploying the site being pushed to the next sprint. Deploying a working website that connects to the database proved more trouble than predicted. The burndown chart mainly stayed above the estimated line, but this can be attributed to the slow start of setting up the tech stack.

7.3 Did you follow your plan(s) exactly?

The plans were not followed exactly. The set up took longer than expected, which blocked most of the other tasks to be started later than expected. The deployment task also proved to be a problem and has been pushed to the next sprint.

Due to the results of research into the potential libraries to use, some tasks had to be moved around. The authentication task was moved from Alac to Winnie since it was easier to implement it in Angular than Django. The authentication software used also came with a premade signup and log in UI, moving Tony's task of creating those UIs to Winnie as well. Using the authentication software overlapped the tasks, which we did not expect to happen.

7.4 What difficulties have you encountered?

There were a few difficulties encountered for this sprint. One of them was onboarding teammates that do not have much experience with the tech stack being used. With the limited time to complete their tasks, they had to figure out how to run and code in Angular or Django. There was also a lot of difficulty with installing and running Docker for some team members. This included Windows 10 updates and Docker not running as expected sometimes in production mode. Networking problems also appeared a lot when the website was hosted, where functionality worked locally but not in production due to network issues.

There were software difficulties in creating UMLs. They were difficult to work with, but there was not much of an option on which software to use.

Another difficulty was retasking items, such as moving authentication from Django to Angular. The move from the backend to frontend and using the authentication software caused 3 tasks to be reassigned and be potentially overlapping.

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

Robert made the decisions as planned by reshuffling the tasks to redistribute work. Otherwise, we did not need the contingency plan.