

CSCC01 Deliverable 5

Henry Huang, Justin Tran, Richard Gao, Riyasat Talukder, Sayon Sivakumaran

Aug 10, 2020

Contents

1 Product Backlog	4
1.1 Changes to Product Backlog	7
2 Release Plan	9
3 Sprint 5 Plan	10
3.1 Sprint 5 Backlog	10
3.2 Initial Taskboard and Burndown Chart	14
3.3 Intermediate Taskboard and Burndown Chart	15
3.4 Final Taskboard and Burndown Chart	16
4 Sprint 6 Plan	17
4.1 Sprint 6 Backlog	17
4.2 Initial Taskboard and Burndown Chart	22
4.3 Intermediate Taskboard and Burndown Chart	23
4.4 Final Taskboard and Burndown Chart	25
5 Software Architecture	26
5.1 Explanation of Components & Dependencies	27
6 Verification	31
6.1 Unit Testing Strategy	31
7 Validation	32
7.1 Sprint 5 Meetings	32
7.1.1 Friday, July 24 [30 minutes]:	32
7.2 Sprint 6 Meetings	36
7.2.1 Friday, July 31 [25 minutes]:	36
8 Retrospection	38

8.1	Project Overview	38
8.2	Project Velocities	38

1 Product Backlog

Customer: Ordering Process

- [Done] As Tom (a customer), I want to register for an account on the website with just my first name, last name, email, phone number, and password.

Priority: 1, Estimated Cost: 6

- [Done] As Tom (a customer), I want to be able to create an account on the website using Google sign-in.

Priority: 1, Estimated Cost: 6

- [Done] As Tom (a customer), I want to login to my account using either a registered email and password or a registered Google account.

Priority: 1, Estimated Cost: 6

- [Done] As Tom (a customer), I want to logout of my account on the website so that my information does not get compromised.

Priority: 1, Estimated Cost: 2

- [Done] As Tom (a customer), I want to view a list of restaurants which displays their name and logo.

Priority: 1, Estimated Cost: 4

- [Done] As Tom (a customer), I want to select a restaurant from a list of restaurants to view specific information regarding it, such as its contact information, location, menu, and prices, in the form of text and pictures.

Priority: 1, Estimated Cost: 5

- As Tom (a customer), I want to be able to select items from a restaurant's menu, specify their quantity, and have them placed in a virtual shopping cart for purchase.

Priority: 3, Estimated Cost: 6

- As Tom (a customer), I want to be able to remove specific items from my virtual shopping cart.

Priority: 3, Estimated Cost: 2

- As Tom (a customer), I want to place an order by checking out the items in my shopping cart and having them be sent to the restaurant.

Priority: 3, Estimated Cost: 6

- As Tom (a customer), I want to receive a virtual receipt of my order by email, which lists the items ordered and their individual prices, the total price, and any discounts applied.

Priority: 6, Estimated Cost: 10

Customer: Website Navigation and Searching

- As Tom (a customer), I want to have a search engine to search for restaurants that are nearby, by providing an address and obtaining a list of restaurants that are deemed nearby, so that I do not have to commute too far for takeout orders.

Priority: 2, Estimated Cost: 12

- As Tom (a customer), I want to have a search engine to search for food options by providing a type of cuisine and obtaining a list of restaurants which sell that type of cuisine, so that I can conveniently select food based on my current tastes.

Priority: 2, Estimated Cost: 6

- As Tom (a customer), I want to have a search engine to search for food by price by providing a specific price range and obtaining a list of restaurants whose average food prices fall within that range, so that I can easily find food based on my current budget.

Priority: 3, Estimated Cost: 6

- As Tom (a customer), I want to select certain restaurants as my favourites and have them put into a personal list which I can view.

Priority: 6, Estimated Cost: 6

- As Tom (a customer), I want the website to visually adjust so as to be viewable on a mobile phone.

Priority: 7, Estimated Cost: 8

Restaurant Owner: Uploading Information

- [Done] As Jane (a restaurant owner), I want to register for an account on the website with just my first name, last name, email, phone number, password, restaurant name, restaurant phone number, restaurant address, and the type of cuisine my restaurant sells.

Priority: 1, Estimated Cost: 10

- [Done] As Jane (a restaurant owner), I want to upload basic information about my restaurant such as a name, an address, a telephone number, and the type of food we serve to a restaurant profile for customers to see.

Priority: 1, Estimated Cost: 6

- [Done] As Jane (a restaurant owner), I want to upload text, pictures, and videos to my restaurant profile that showcase my restaurant's food and menu.

Priority: 1, Estimated Cost: 6

- As Jane (a restaurant owner), I want to include categories in the menu, such as "Starters", "Appetizers", and "Mains", that I can place menu items under.

Priority: 6, Estimated Cost: 4

Restaurant Owner: Community Engagement

- [Done] As Jane (a restaurant owner), I want to share stories regarding my restaurant by uploading text, pictures, and videos to a section on my restaurant profile about my culture, recipe origins, and cooking tips for customers to see.

Priority: 3, Estimated Cost: 8

- As Jane (a restaurant owner), I want to post announcements regarding my restaurant by uploading text to an “Announcements” section on my restaurant profile for customers to see.

Priority: 3, Estimated Cost: 8

Restaurant Owner: Customer Service

- As Jane (a restaurant owner), I want to view data that I have collected on customer orders, such as total number of customers ordering from my restaurant and most frequently ordered menu items.

Priority: 7, Estimated Cost: 10

General: Community Engagement

- As a registered user of the website, I want to make posts in which I can upload text and pictures to a “Community Board” section on the website for others to see.

Priority: 6, Estimated Cost: 12

1.1 Changes to Product Backlog

- **Modified User Story:** “As Tom (a customer), I want to be able to remove specific items from my virtual shopping cart *and cancel my order before payment by removing all items from the cart*” → “As Tom (a customer), I want to be able to remove specific items from my virtual shopping cart.”
 - Removed the requirement about specifically cancelling an order since the ability to remove items from the cart already addresses the requirement.
- **Modified User Story:** “As Tom (a customer), I want to place an order by checking out the items in my shopping cart, *providing my payment details, whether I am ordering takeout or delivery, and, if I am ordering delivery, my delivery address and a short delivery note for the courier*” → “As Tom (a customer), I want to place an order by checking out the items in my shopping cart and *having them be sent to the restaurant*.”
 - It was mentioned in a client meeting that handling payment and delivery was not required. Thus, parts of the user story dealing with payment and delivery were removed. Instead, it is just required to have the ability to send the shopping cart contents to the restaurant.
- **Deleted User Story:** As Tom (a customer), I want the online transactions related to my order to be handled securely such that my payment details and personal information are kept safe.
 - As mentioned above, handling payment is not a required feature to have. Therefore, this user story has become obsolete.
- **Deleted User Story:** As Tom (a customer), I want to be given the option to have my payment details and delivery address saved to the account that I am logged into after providing these details in the first delivery order that I place, so that this information can be automatically used for future orders on the account.
 - As mentioned above, handling payment and delivery is not a required feature to have. Therefore, this user story has become obsolete.
- **Deleted User Story:** As Tom (a customer), I want to give ratings to restaurants that I have ordered from, on a scale of 1 to 5.
 - It was mentioned in a client meeting that ratings were not required to have.
- **Deleted User Story:** As Tom (a customer), I want to sort restaurants by customer ratings, so that I can easily determine which restaurants provide the most widely recommended services.
 - Since ratings are not required, this requirement also becomes obsolete.
- **Deleted User Story:** As Tom (a customer), I want to keep track of my delivery, so that I can plan my schedule accordingly.
 - It was mentioned in a client meeting that this was not a required feature to have.

- **New User Story:** As Tom (a customer), I want the website to visually adjust so as to be viewable on a mobile phone.

Priority: 7, Estimated Cost: 8

- It was mentioned in a client meeting that this was required. However, it was regarded a low priority requirement, to be completed after the other requirements of the website are implemented.

- **New User Story:** As Jane (a restaurant owner), I want to include categories in the menu, such as “Starters”, “Appetizers”, and “Mains”, that I can place menu items under.

Priority: 6, Estimated Cost: 4

- It was mentioned in a client meeting that this was required. However, it was regarded as a low priority requirement.

- **Deleted User Story:** As Jane (a restaurant owner), I want to occasionally offer promotions of my own design, such as flat-rate or percentage discounts, the details of which I can upload to my restaurant profile for customers to see.

- It was mentioned in a client meeting that this was not a required feature to have. It was stated that the main purpose of the website was to provide a platform to showcase restaurants and allow restaurant owners to share stories about their restaurant. In particular, e-commerce was not considered a significant part of the website.

- **New User Story:** As Jane (a restaurant owner), I want to post announcements regarding my restaurant by uploading text to an “Announcements” section on my restaurant profile for customers to see.

Priority: 3, Estimated Cost: 8

- It was mentioned in a client meeting that this was required, and should have a fairly high priority.

- **Deleted User Story:** As Jane (a restaurant owner), I want to collect data on the customers that order from my restaurant, such as their customer ID, their location, what they ordered, when they placed the order, and any discount codes they used.

- This requirement did not represent a requirement of the restaurant owner because the restaurant owner themselves do not collect any data. As well, this requirement is addressed across several other user stories, since any data that is required is collected in order to implement the user stories.

- **Modified User Story:** “As Jane (a restaurant owner), I want to view data that I have collected on customer orders, such as total number of customers ordering from my restaurant, most frequently ordered menu items, *and total number of customers who ordered using a discount promotion*” → “As Jane (a restaurant owner), I want to view data that I have collected on customer orders, such as total number of customers ordering from my restaurant and most frequently ordered menu items.” Also changed priority of this user story from 5 → 7.

- Since promotions were removed as a requirement, we removed the requirement to display data regarding how many customers ordered using discount promotions. As well, since it was mentioned by the clients that e-commerce was not a significant aspect of the website, we lowered the priority of the user story from 5 to 7.

2 Release Plan

The length of our sprints will be one week long. Since we do not have a large amount of time to develop our software, we decided to go with a fairly short sprint length. This is so that we can more accurately measure our progress by making sure that we are up to date each week rather than, for example, every two weeks.

One week sprints also allow us to easily correspond each sprint with a priority in our user stories. Since our priorities range from 1 to 6 and we have roughly 6 weeks to complete our project, we can assign priority 1 user stories to sprint 1, priority 2 user stories to sprint 2, and so on.

However, instead of having our sprints start each Tuesday and end the following Monday, we chose to start our sprint 5 a little earlier, on Friday, July 24th, and had it end on Monday, August 3rd. This was to give us more time to make progress on the tasks that were carried over from sprint 4. Sprint 6 returned back to the original sprint schedule, starting on Tuesday, August 4th, and ending on Monday, August 10th.

3 Sprint 5 Plan

3.1 Sprint 5 Backlog

Note: Items in the Sprint Backlog are organized according to the following:

- **User Story ID:** User Story
 - **Task ID (User Story ID) [Cost of Task]:** Task
 - Acceptance Criteria
-

Customer: Ordering Process

- **U1:** As Tom (a customer), I want to login to my account using either a registered email and password or a registered Google account.

Priority: 1, Estimated Cost: 6

- **T1 (U1) [Cost 1]:** Frontend: UI to login to a Google account.
 - Given Tom has registered for an account, when he clicks the “Login” button in the navigation bar of the website, then he should see a form which asks for his email and password.
 - Given Tom has registered for a Google account, when he clicks the “Google Login” button in the navigation bar of the website, then he should see a form which asks for his Google email.
 - Given Tom has filled out all fields of the “Login” or “Google Login” form, when he clicks the “Submit” button, then he should be logged into his account.
 - Given Tom has not filled out all fields of the “Login” or “Google Login” form, when he clicks the “Submit” button, then he will be notified that not all required fields were filled in.
- **T2 (U1) [Cost 5]:** Backend: Handling login session.
 - Given Tom has registered for a Google account, when he logs in to his Google account, he will be given a login session that marks him as logged in so that other components of the website may use their user information attached to the account.

- **U2:** As Tom (a customer), I want to logout of my account on the website so that my information does not get compromised.

Priority: 1, Estimated Cost: 2

- **T3 (U2) [Cost 1]:** Frontend: UI to logout of Google account.
 - Given Tom is logged into his account, when he clicks the “Logout” button, then he should be logged out of his account.

- **T4 (U2) [Cost 1]:** Backend: Handling login session.
 - Given Tom is in a login session, when he clicks the “Logout” button, then his login session should end.

- **U3:** As Tom (a customer), I want to be able to select items from a restaurant’s menu, specify their quantity, and have them placed in a virtual shopping cart for purchase.

Priority: 3, Estimated Cost: 6

- **T5 (U3) [Cost 4]:** Frontend: UI to click on items from a restaurant’s menu, specify their quantity, and see the virtual shopping cart get updated.
 - Given Tom wants to add an item to his shopping cart, when he clicks on an item from a restaurant’s menu, then he should see a dropdown list which allows him to select a quantity for the item.
 - Given Tom has clicked on an item from a restaurant’s menu, when he clicks the “Add to Order” button, then the item should appear in his shopping cart.
 - Given Tom has placed items into his shopping cart, when he clicks the shopping cart icon, then he should see a list of the items he ordered including their quantities and prices, and the total price of the order.
 - Given Tom has items from a particular restaurant already in his shopping cart, when he attempts to add an item from a different restaurant to his shopping cart, then he will see a pop-up telling him that he can only order from one restaurant at a time.
- **T6 (U3) [Cost 2]:** Backend: Ability to store shopping cart data in database.
 - Given Tom has clicked on an item from a restaurant’s menu, when he clicks the “Add to Shopping Cart” button, then the item should be stored to the shopping cart field of his user document in the database.
- **U4:** As Tom (a customer), I want to be able to remove specific items from my virtual shopping cart.

Priority: 3, Estimated Cost: 2

- **T7 (U4) [Cost 1]:** Frontend: UI to remove items from the shopping cart.
 - Given Tom wants to remove an item from his shopping cart, when he clicks the shopping cart button, then he should see a delete icon next to each item in his shopping cart which allows him to remove the item from his shopping cart.
- **T8 (U4) [Cost 1]:** Backend: Ability to remove shopping cart data in database.
 - Given Tom has an item in his shopping cart, when he clicks the delete icon next to the item, then the item should be removed from the shopping cart field of his user document in the database.
- **U5:** As Tom (a customer), I want to place an order by checking out the items in my shopping cart and having them be sent to the restaurant.

Priority: 3, Estimated Cost: 6

- **T9 (U5) [Cost 2]:** Frontend: UI to check out the shopping cart.
 - Given Tom wants to check out his shopping cart, when he clicks the shopping cart button, then he should see a “Place Order” button that places his order.

- Given Tom has not added any items to his shopping cart, when he clicks the shopping button, then he will not see a “Place Order” button.
- **T10 (U5) [Cost 4]:** Backend: Have details of the order sent to the restaurant which was ordered from.
 - Given Tom has items in his shopping cart, when he clicks the “Place Order” button to place his order, then details of his order such as items ordered, quantity of each item, price of each item, and total price are sent to the restaurant’s database.

Customer: Website Navigation and Searching

- **U6:** As Tom (a customer), I want to have a search engine to search for restaurants that are nearby, by providing an address and obtaining a list of restaurants that are deemed nearby, so that I do not have to commute too far for takeout orders.

Priority: 2, Estimated Cost: 12

- **T11 (U6) [Cost 3]:** Frontend: UI to provide an address and see a list of nearby restaurants.
 - Given Tom wants to search for restaurants that are nearby, when he clicks the search bar in the header of the website, then he should be able to type an address into the search bar.
 - Given Tom has entered an address in the search bar, when he clicks the “Search” button, then he should see a list of restaurants that are near the address.
 - Given Tom has entered an address in the search bar that is not nearby any registered restaurants on the website, when he clicks the “Search” button, then he should be notified that there are no results.
- **T12 (U6) [Cost 9]:** Backend: Take customer’s address information and generate a list of restaurants which are deemed nearby.
 - Given Tom has entered an address in the search bar, when he clicks the “Search” button, then the address should be queried in the database to find any nearby restaurants.
 - Given there are restaurants nearby the address given by Tom, when the database is queried for results, then a list of nearby restaurants should be obtained.
 - Given there are no restaurants nearby the address given by Tom, when the database is queried for results, then an empty list should be obtained.
- **U7:** As Tom (a customer), I want to have a search engine to search for food options by providing a type of cuisine and obtaining a list of restaurants which sell that type of cuisine, so that I can conveniently select food based on my current tastes.

Priority: 2, Estimated Cost: 6

- **T13 (U7) [Cost 2]:** Frontend: UI to provide a type of cuisine and see a list of restaurants which sell that type of cuisine.
 - Given Tom wants to search for restaurants by cuisine type, when he clicks the search bar, then he should be able to type a cuisine type into the search bar.

- Given Tom has entered a valid cuisine type in the search bar, when he clicks the “Search” button, then he should see a list of restaurants which offer that type of cuisine.
- Given Tom has entered a cuisine type that does not match any cuisine type of any registered restaurant in the search bar, when he clicks the “Search” button, then he should be notified that there are no results.
- **T14 (U7) [Cost 4]:** Backend: Take the given type of cuisine and generate a list of restaurants which sell that type of cuisine by querying the database.
 - Given Tom has entered a cuisine type in the search bar, when he clicks the “Search” button, then the address should be queried in the database to find any restaurants which contain cuisine of that type.
 - Given there are restaurants which offer the cuisine type given by Tom, when the database is queried for results, then a list of restaurants which offer that cuisine type should be obtained.
 - Given there are no restaurants which offer the cuisine type given by Tom, when the database is queried for results, then an empty list should be obtained.

Restaurant Owner: Community Engagement

- **U8:** As Jane (a restaurant owner), I want to post announcements regarding my restaurant by uploading text to an “Announcements” section on my restaurant profile for customers to see.

Priority: 3, Estimated Cost: 8

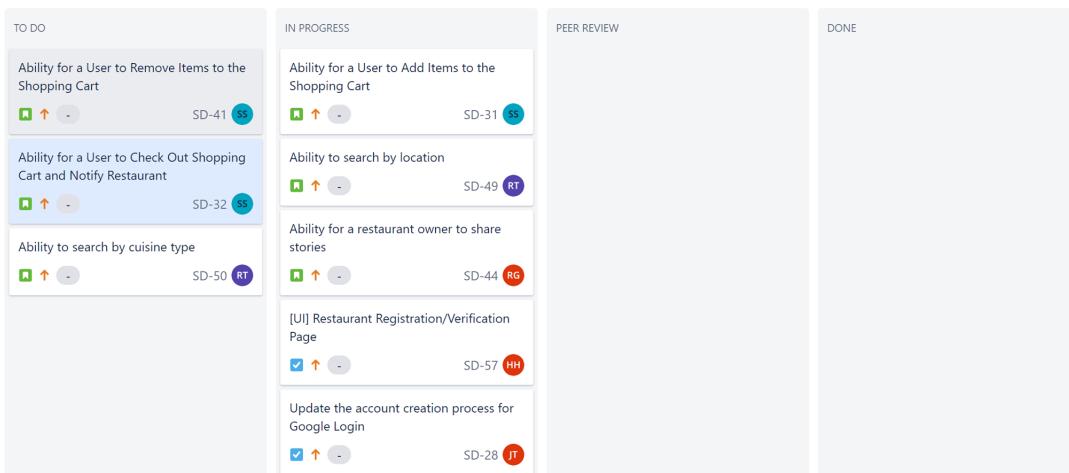
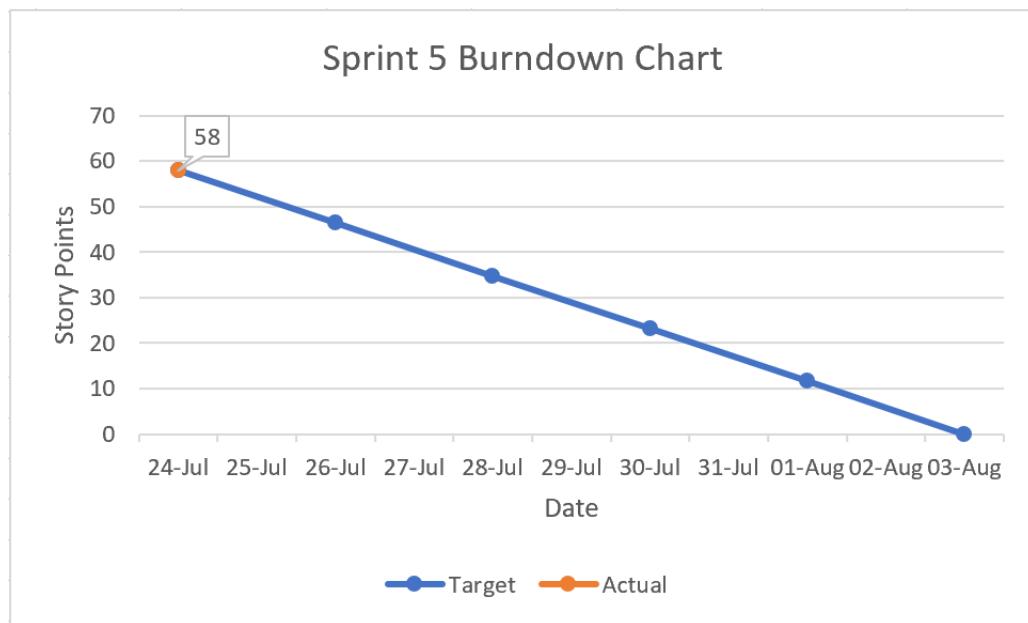
- **T15 (U8) [Cost 4]:** Frontend: Announcements upload user interface and have announcements show up in restaurant profile.
 - Given Jane wants to upload an announcement, when she clicks on the “Manage Announcements” button, then she will see a form asking for a title and body of her announcement.
 - Given Jane fills out the announcements form, when she clicks on the “Post announcement” button, then the announcement will appear under the “Announcements” section in her restaurant profile.
 - Given Jane has posted an announcement, when she clicks on the delete icon next to the announcement, then the announcement will disappear from the list of announcements.
- **T16 (U8) [Cost 4]:** Backend: Have uploaded announcements stored to the database and retrieve the announcements from the database.
 - Given Jane fills out the announcements form, when she clicks on the “Post announcement” button, then the announcement will be added to her restaurant’s document in the database.
 - Given Jane has posted an announcement, when she clicks on the delete icon next to the announcement, then the announcement will be deleted from her restaurant’s document in the database.
 - Given Jane has posted an announcement, when she navigates to the her restaurant profile on the website, the announcement will be pulled from the database and placed under the “Announcements” section.

Miscellaneous Tasks

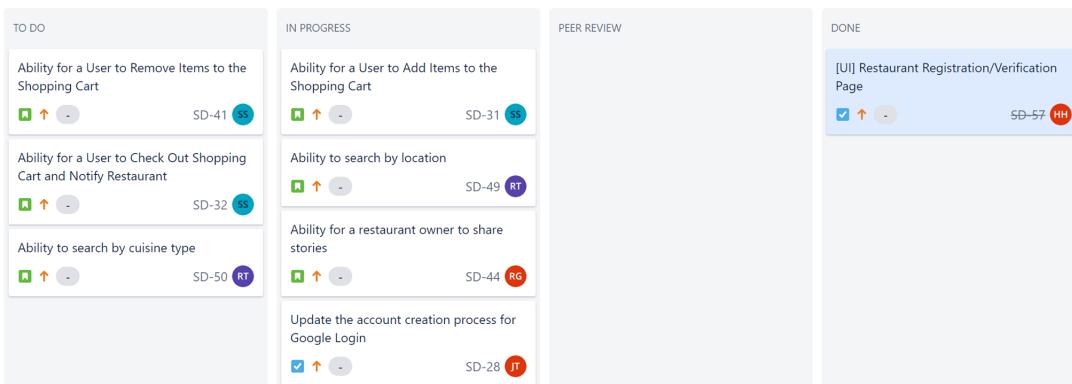
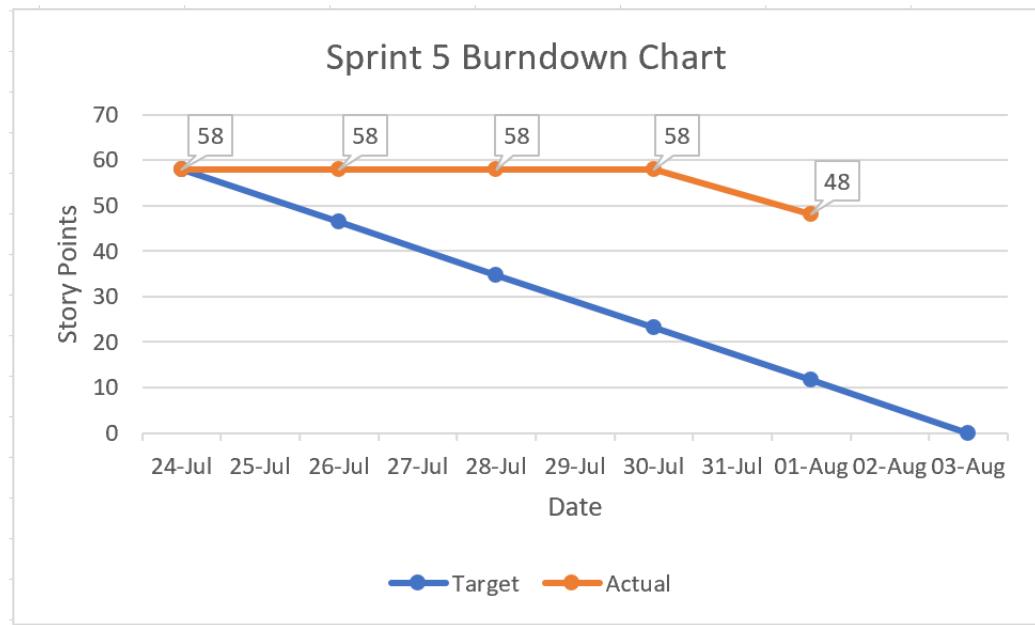
- **T17 [Cost 10]:** Setup the ability for an administrator to accept or decline restaurant registration requests to the website.

Priority: 3, Estimated Cost: 10

3.2 Initial Taskboard and Burndown Chart

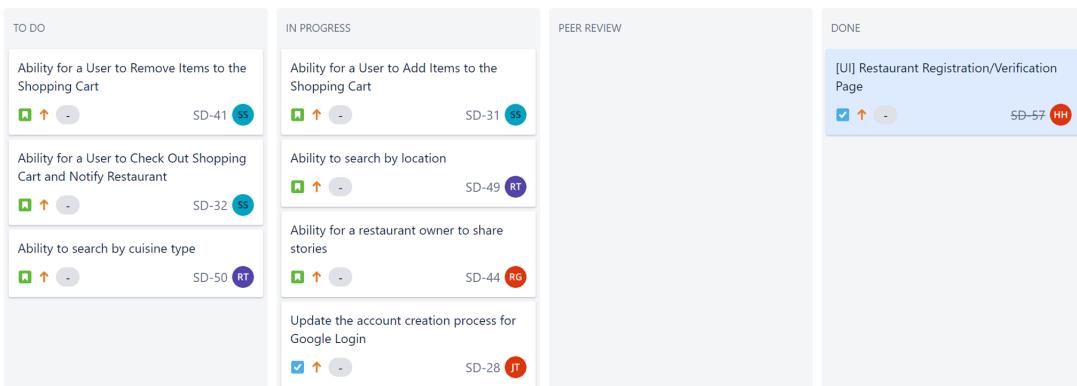
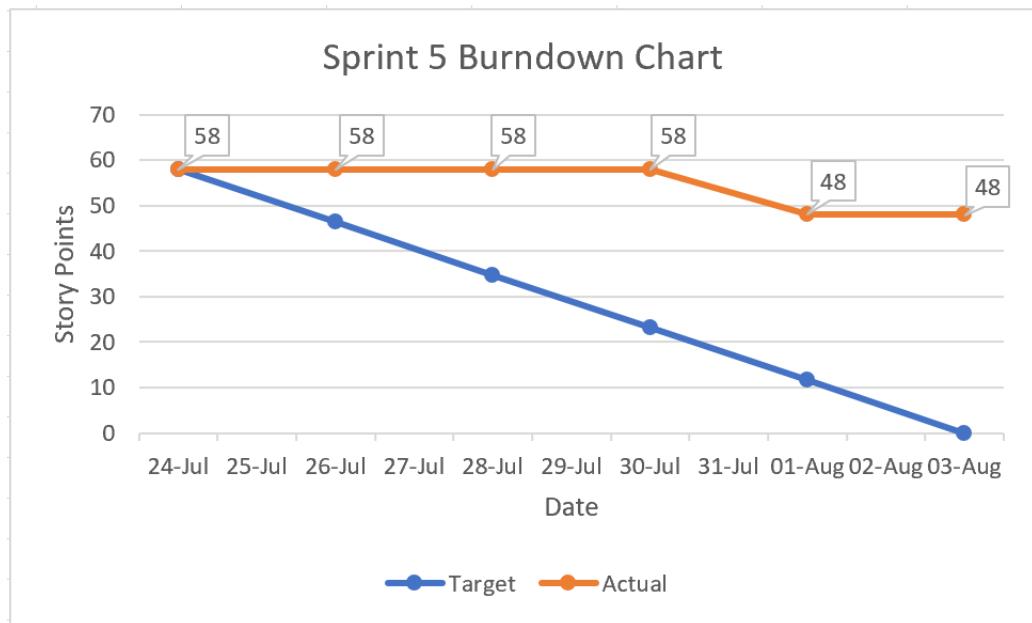


3.3 Intermediate Taskboard and Burndown Chart



These snapshots highlight the completion of the restaurant registration and verification page. This page allows an administrator (i.e., our clients) to accept or decline restaurants that have filled out a registration on the website. If accepted, the restaurant's information will be added to the database and will appear on the website.

3.4 Final Taskboard and Burndown Chart



4 Sprint 6 Plan

4.1 Sprint 6 Backlog

Note: Items in the Sprint Backlog are organized according to the following:

- **User Story ID:** User Story
 - **Task ID (User Story ID) [Cost of Task]:** Task
 - Acceptance Criteria
-

Customer: Ordering Process

- **U1:** As Tom (a customer), I want to login to my account using either a registered email and password or a registered Google account.

Priority: 1, Estimated Cost: 6

- **T1 (U1) [Cost 1]:** Frontend: UI to login to a Google account.
 - Given Tom has registered for an account, when he clicks the “Login” button in the navigation bar of the website, then he should see a form which asks for his email and password.
 - Given Tom has registered for a Google account, when he clicks the “Google Login” button in the navigation bar of the website, then he should see a form which asks for his Google email.
 - Given Tom has filled out all fields of the “Login” or “Google Login” form, when he clicks the “Submit” button, then he should be logged into his account.
 - Given Tom has not filled out all fields of the “Login” or “Google Login” form, when he clicks the “Submit” button, then he will be notified that not all required fields were filled in.
- **T2 (U1) [Cost 5]:** Backend: Handling login session.
 - Given Tom has registered for a Google account, when he logs in to his Google account, he will be given a login session that marks him as logged in so that other components of the website may use their user information attached to the account.

- **U2:** As Tom (a customer), I want to logout of my account on the website so that my information does not get compromised.

Priority: 1, Estimated Cost: 2

- **T3 (U2) [Cost 1]:** Frontend: UI to logout of Google account.
 - Given Tom is logged into his account, when he clicks the “Logout” button, then he should be logged out of his account.

- **T4 (U2) [Cost 1]:** Backend: Handling login session.
 - Given Tom is in a login session, when he clicks the “Logout” button, then his login session should end.

- **U3:** As Tom (a customer), I want to be able to select items from a restaurant’s menu, specify their quantity, and have them placed in a virtual shopping cart for purchase.

Priority: 3, Estimated Cost: 6

- **T5 (U3) [Cost 4]:** Frontend: UI to click on items from a restaurant’s menu, specify their quantity, and see the virtual shopping cart get updated.
 - Given Tom wants to add an item to his shopping cart, when he clicks on an item from a restaurant’s menu, then he should see a dropdown list which allows him to select a quantity for the item.
 - Given Tom has clicked on an item from a restaurant’s menu, when he clicks the “Add to Order” button, then the item should appear in his shopping cart.
 - Given Tom has placed items into his shopping cart, when he clicks the shopping cart icon, then he should see a list of the items he ordered including their quantities and prices, and the total price of the order.
 - Given Tom has items from a particular restaurant already in his shopping cart, when he attempts to add an item from a different restaurant to his shopping cart, then he will see a pop-up telling him that he can only order from one restaurant at a time.
- **T6 (U3) [Cost 2]:** Backend: Ability to store shopping cart data in database.
 - Given Tom has clicked on an item from a restaurant’s menu, when he clicks the “Add to Shopping Cart” button, then the item should be stored to the shopping cart field of his user document in the database.

- **U4:** As Tom (a customer), I want to be able to remove specific items from my virtual shopping cart.

Priority: 3, Estimated Cost: 2

- **T7 (U4) [Cost 1]:** Frontend: UI to remove items from the shopping cart.
 - Given Tom wants to remove an item from his shopping cart, when he clicks the shopping cart button, then he should see a delete icon next to each item in his shopping cart which allows him to remove the item from his shopping cart.
- **T8 (U4) [Cost 1]:** Backend: Ability to remove shopping cart data in database.
 - Given Tom has an item in his shopping cart, when he clicks the delete icon next to the item, then the item should be removed from the shopping cart field of his user document in the database.
- **U5:** As Tom (a customer), I want to place an order by checking out the items in my shopping cart and having them be sent to the restaurant.

Priority: 3, Estimated Cost: 6

- **T9 (U5) [Cost 2]:** Frontend: UI to check out the shopping cart.
 - Given Tom wants to check out his shopping cart, when he clicks the shopping cart button, then he should see a “Place Order” button that places his order.

- Given Tom has not added any items to his shopping cart, when he clicks the shopping button, then he will not see a “Place Order” button.
- **T10 (U5) [Cost 4]:** Backend: Have details of the order sent to the restaurant which was ordered from.
 - Given Tom has items in his shopping cart, when he clicks the “Place Order” button to place his order, then details of his order such as items ordered, quantity of each item, price of each item, and total price are sent to the restaurant’s database.

Customer: Website Navigation and Searching

- **U6:** As Tom (a customer), I want to have a search engine to search for restaurants that are nearby, by providing an address and obtaining a list of restaurants that are deemed nearby, so that I do not have to commute too far for takeout orders.

Priority: 2, Estimated Cost: 12

- **T11 (U6) [Cost 3]:** Frontend: UI to provide an address and see a list of nearby restaurants.
 - Given Tom wants to search for restaurants that are nearby, when he clicks the search bar in the header of the website, then he should be able to type an address into the search bar.
 - Given Tom has entered an address in the search bar, when he clicks the “Search” button, then he should see a list of restaurants that are near the address.
 - Given Tom has entered an address in the search bar that is not nearby any registered restaurants on the website, when he clicks the “Search” button, then he should be notified that there are no results.
- **T12 (U6) [Cost 9]:** Backend: Take customer’s address information and generate a list of restaurants which are deemed nearby.
 - Given Tom has entered an address in the search bar, when he clicks the “Search” button, then the address should be queried in the database to find any nearby restaurants.
 - Given there are restaurants nearby the address given by Tom, when the database is queried for results, then a list of nearby restaurants should be obtained.
 - Given there are no restaurants nearby the address given by Tom, when the database is queried for results, then an empty list should be obtained.

- **U7:** As Tom (a customer), I want to have a search engine to search for food options by providing a type of cuisine and obtaining a list of restaurants which sell that type of cuisine, so that I can conveniently select food based on my current tastes.

Priority: 2, Estimated Cost: 6

- **T13 (U7) [Cost 2]:** Frontend: UI to provide a type of cuisine and see a list of restaurants which sell that type of cuisine.
 - Given Tom wants to search for restaurants by cuisine type, when he clicks the search bar, then he should be able to type a cuisine type into the search bar.

- Given Tom has entered a valid cuisine type in the search bar, when he clicks the “Search” button, then he should see a list of restaurants which offer that type of cuisine.
 - Given Tom has entered a cuisine type that does not match any cuisine type of any registered restaurant in the search bar, when he clicks the “Search” button, then he should be notified that there are no results.
- **T14 (U7) [Cost 4]:** Backend: Take the given type of cuisine and generate a list of restaurants which sell that type of cuisine by querying the database.
 - Given Tom has entered a cuisine type in the search bar, when he clicks the “Search” button, then the address should be queried in the database to find any restaurants which contain cuisine of that type.
 - Given there are restaurants which offer the cuisine type given by Tom, when the database is queried for results, then a list of restaurants which offer that cuisine type should be obtained.
 - Given there are no restaurants which offer the cuisine type given by Tom, when the database is queried for results, then an empty list should be obtained.
- **(U8):** As Tom (a customer), I want to select certain restaurants as my favourites and have them put into a personal list which I can view.
Priority: 6, Estimated Cost: 6
 - **T15 (U8) [Cost 5]:** Frontend: UI to select restaurants as favourites and ability to see a list of favourited restaurants.
 - Given Tom wants to favourite a restaurant, when he navigates to the restaurant list, then he should see a star icon on top of the restaurant’s card that he wants to favourite.
 - Given Tom wants to see a list of his favourite restaurants, when he clicks the “Favourites” button, then he should see a list of restaurants that he previously favourited.
 - Given Tom clicks the star icon on the restaurant’s card, when he clicks on the “Favourites” button, then he should see the restaurant’s name and logo appear in a list of his favourite restaurants.
 - **T16 (U8) [Cost 1]:** Backend: Have a customer’s favourited restaurant’s stored in the database.
 - Given Tom wants to add a restaurant from the restaurant list to his favourites, when he clicks the star icon on top of the restaurant’s card, then the restaurant’s ID should be stored in a “favourites” field in his user document.

Restaurant Owner: Community Engagement

- **U9:** As Jane (a restaurant owner), I want to post announcements regarding my restaurant by uploading text to an “Announcements” section on my restaurant profile for customers to see.

Priority: 3, Estimated Cost: 8

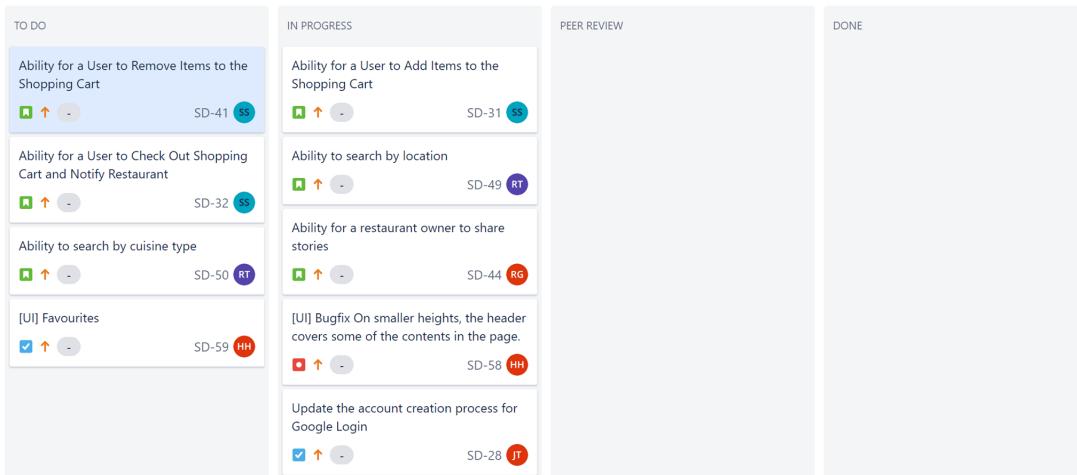
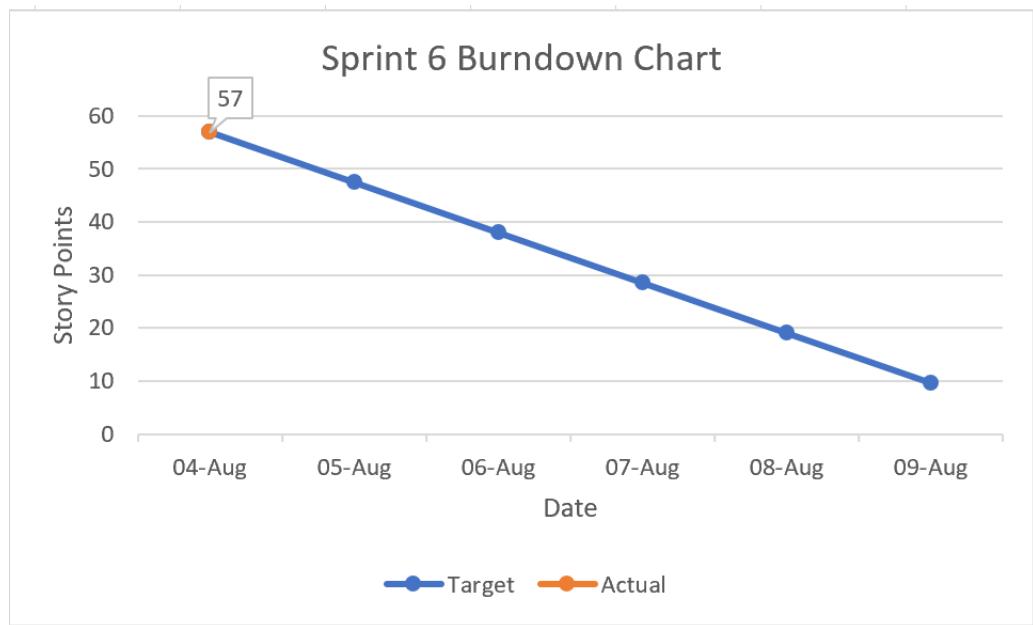
- **T17 (U9) [Cost 4]:** Frontend: Announcements upload user interface and have announcements show up in restaurant profile.
 - Given Jane wants to upload an announcement, when she clicks on the “Manage Announcements” button, then she will see a form asking for a title and body of her announcement.
 - Given Jane fills out the announcements form, when she clicks on the “Post announcement” button, then the announcement will appear under the “Announcements” section in her restaurant profile.
 - Given Jane has posted an announcement, when she clicks on the delete icon next to the announcement, then the announcement will disappear from the list of announcements.
- **T18 (U9) [Cost 4]:** Backend: Have uploaded announcements stored to the database and retrieve the announcements from the database.
 - Given Jane fills out the announcements form, when she clicks on the “Post announcement” button, then the announcement will be added to her restaurant’s document in the database.
 - Given Jane has posted an announcement, when she clicks on the delete icon next to the announcement, then the announcement will be deleted from her restaurant’s document in the database.
 - Given Jane has posted an announcement, when she navigates to the her restaurant profile on the website, the announcement will be pulled from the database and placed under the “Announcements” section.

Miscellaneous Tasks

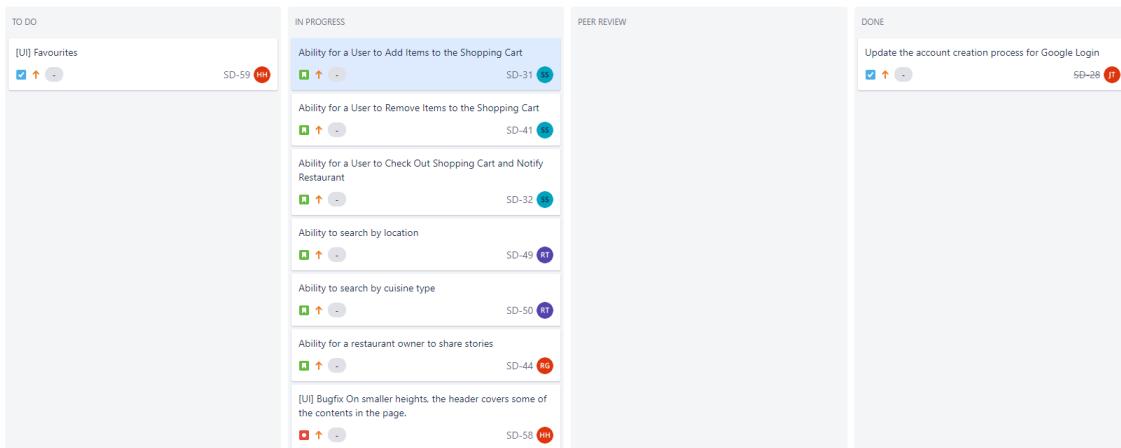
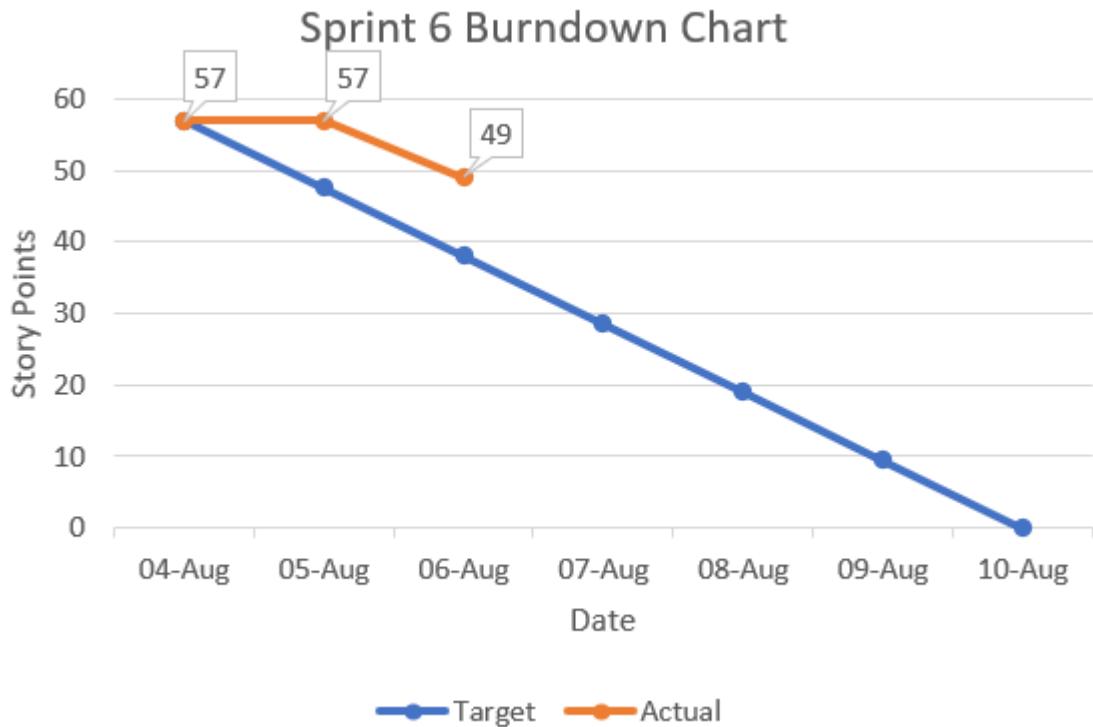
- **T19 [Cost 3]:** Adjust UI of header so that it does not cover contents of the page on smaller screen sizes.

Priority: 6, Estimated Cost: 3

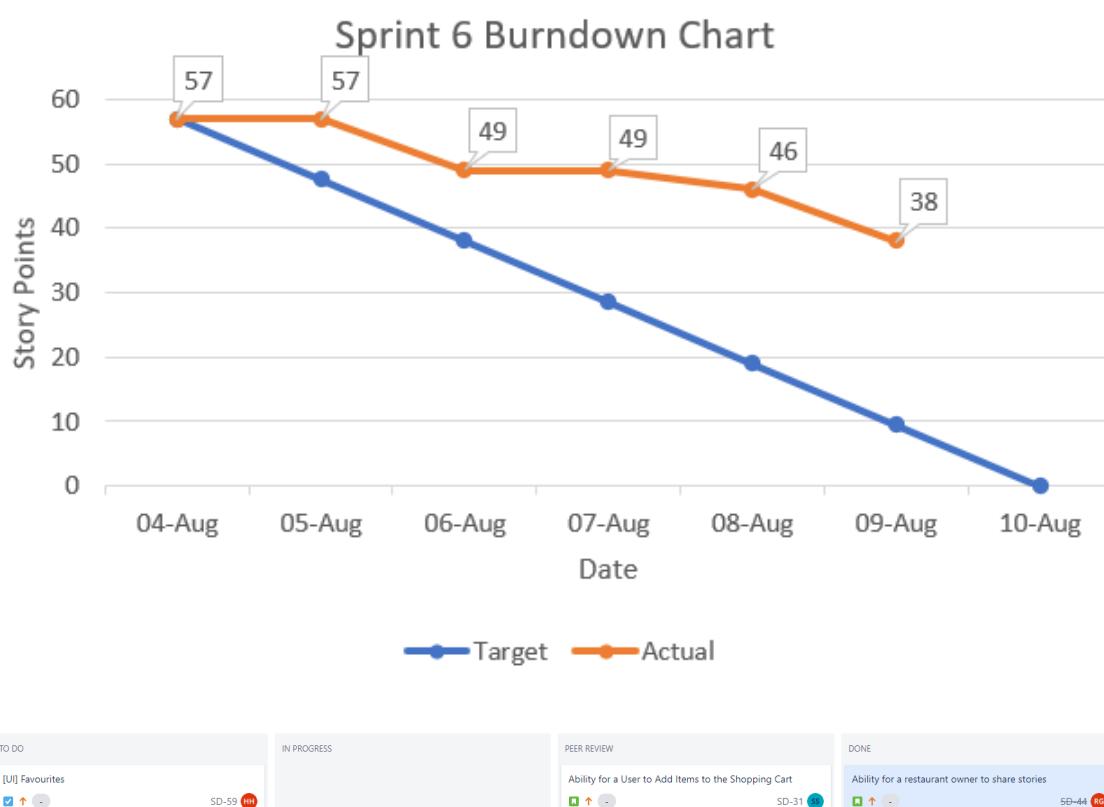
4.2 Initial Taskboard and Burndown Chart



4.3 Intermediate Taskboard and Burndown Chart



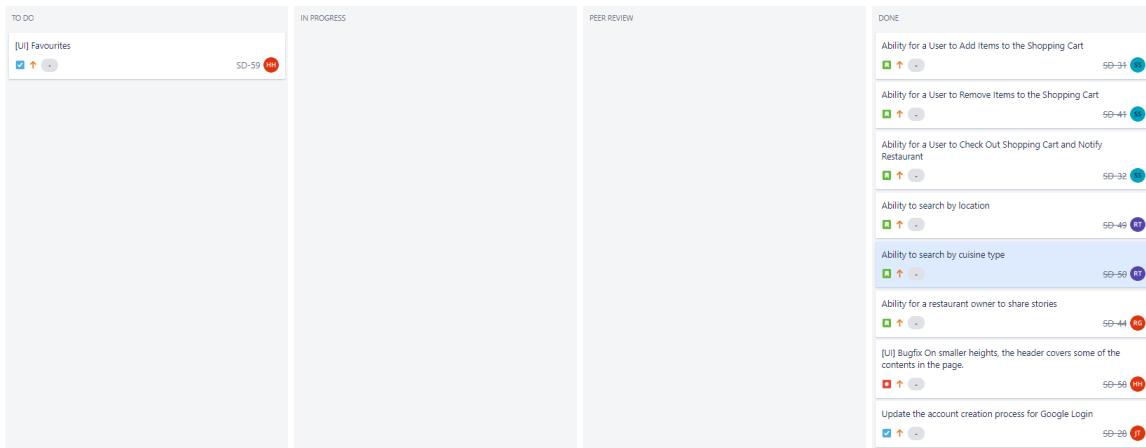
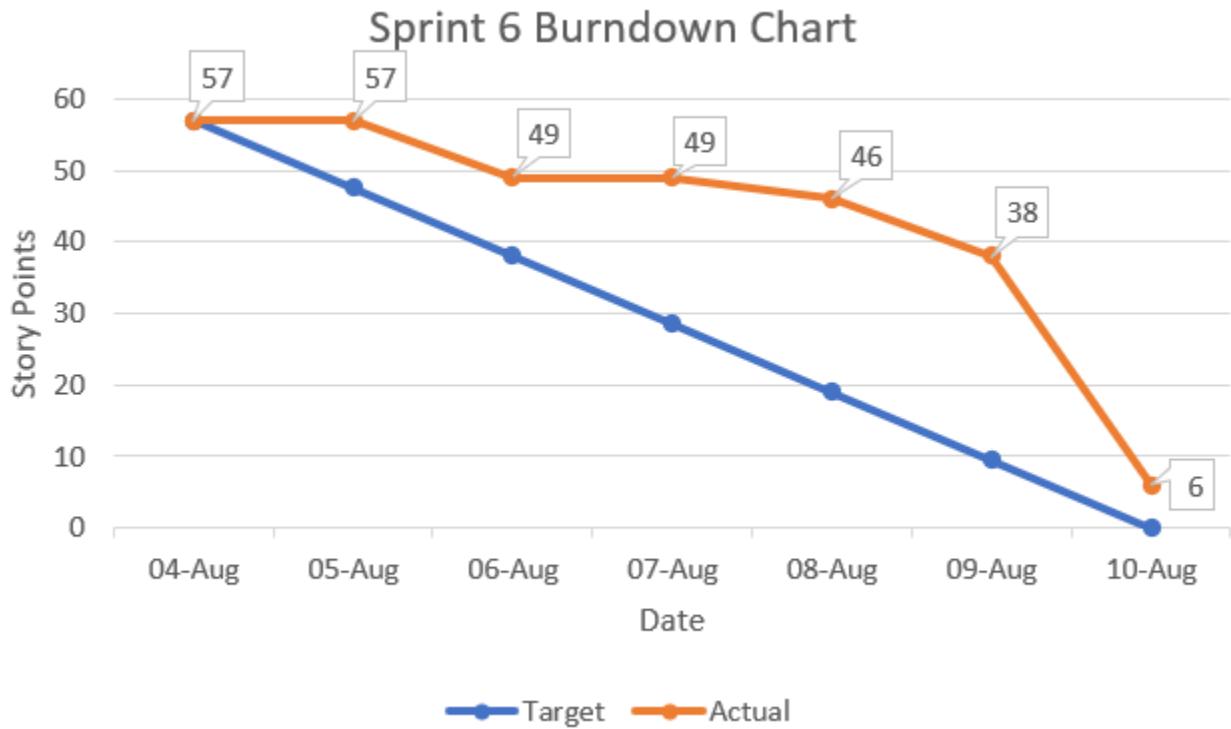
We completed the Google login and logout feature, which helped us to implement several other components which relied on restaurant and customer authentication.



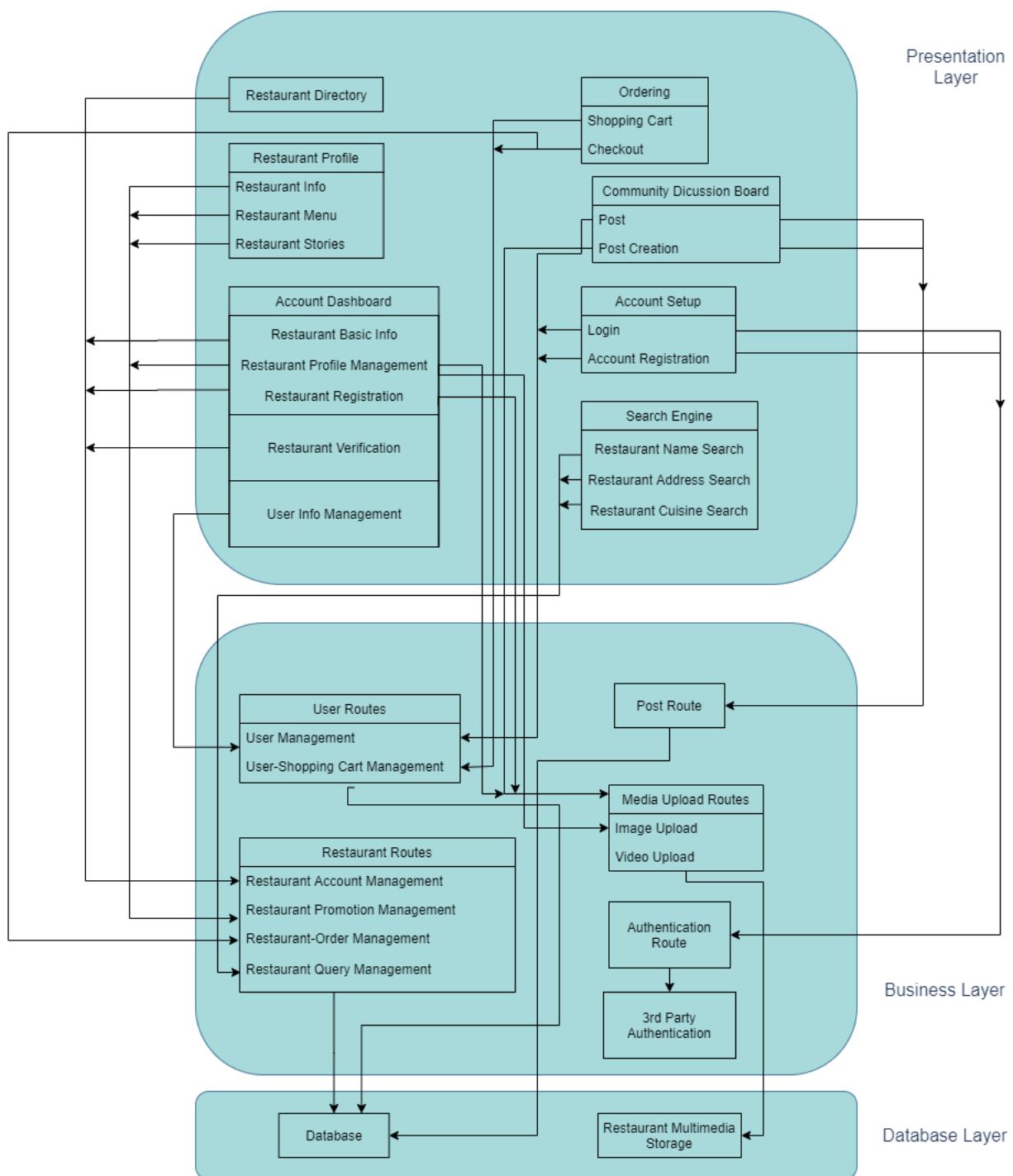
TO DO	IN PROGRESS	PEER REVIEW	DONE
[UI] Favourites ✓ ↑ - SD-59 HI		Ability for a User to Add Items to the Shopping Cart SD-31 OK Ability for a User to Remove Items to the Shopping Cart SD-41 OK Ability for a User to Check Out Shopping Cart and Notify Restaurant SD-32 OK Ability to search by location SD-49 RT Ability to search by cuisine type SD-50 RT	Ability for a restaurant owner to share stories SD-44 RG [UI] Bugfix On smaller heights, the header covers some of the contents in the page. SD-58 HI Update the account creation process for Google Login SD-28 RT

We completed the announcements requirement and put all of the other user stories (besides the favourites feature) in Peer Review.

4.4 Final Taskboard and Burndown Chart



5 Software Architecture



5.1 Explanation of Components & Dependencies

Note: Items in this section are organized according to the following:

System Component:

- *Subcomponent:* Description of subcomponent.
 - Dependency of subcomponent.

Not all system components may have subcomponents and not all subcomponents may have dependencies.

Account Dashboard:

- *Restaurant Basic Info:* Allows restaurant owners to edit their account information.
 - Uses Restaurant Account Management to save and fetch information from the database regarding the restaurant's data.
- *Restaurant Profile Management:* Allows restaurant owner to edit profile information.
 - Uses Restaurant Promotion Management to save and fetch restaurant info, menu and announcement information regarding the restaurant.
- *Restaurant Registration:* Allows users to register a restaurant.
 - Uses Restaurant Account Management to save restaurant information into the database.
- *Restaurant Verification:* Allows admin users to verify a restaurant to be displayed for the application.
 - Uses Restaurant Account Management to verify a restaurant.
- *User Info Management:* Allows users to edit their account information.
 - Uses User Management to save and fetch user information from the database regarding the user's data.

Restaurant Profile:

- *Restaurant Information:* Displays a Restaurant's profile information.
 - Uses Restaurant Promotion Management to retrieve the Restaurant's profile information.
- *Restaurant Menu:* Displays a Restaurant's menu information.
 - Uses Restaurant Promotion management to retrieve the Restaurant's menu information.

- *Restaurant Announcements*: Displays a Restaurant's announcements.
 - Uses Restaurant Promotion Management to retrieve the Restaurant's announcements.

Restaurant Directory: Allows customers to view a list of restaurants registered on the website according to their preference.

- Uses Restaurant Account Management to retrieve information from a restaurant.

Community Discussion Board

- *Post*: Displays information from a post.
 - Uses Post Route to retrieve information post information.
- *Post Creation*: Allows restaurant owners to create a post.
 - Uses Post Route to save a post.
 - Uses Restaurant Account Management to authorize if the user is an restaurant owner.
 - Uses Media Upload Route to allow a restaurant owner to upload an image.

Account Setup

- *Login*: Allows a returning user to log into their account.
 - Uses Authentication Route to authenticate log in credentials.
 - Uses User Management to fetch user information from the database regarding the user's data.
- *Account Registration*: Allows a new user to create an account.
 - Uses Authentication Route to authenticate a new use.
 - Uses User Management to fetch new user's information from the database and change the user's information on first time account set-up.

Search Engine

- *Restaurant Name Search*: Allows customers to filter the list of restaurants by name.
 - Uses Restaurant Query Management to retrieve restaurants that matches the restaurants name.
- *Restaurant Address Search*: Allows customers to filter the list of restaurants by address.
 - Uses Restaurant Query Management to retrieve restaurants that matches the address.
- *Restaurant Cuisine Search*: Allows customers to filter the list of restaurants by cuisines type.

- Uses Restaurant Query Management to retrieve restaurants that sell that type of cuisine.

Ordering:

- *Shopping Cart*: Allows customers to add to, remove from, and see a shopping cart containing a list of menu items they would like to order.
 - Uses Restaurant Account Management to retrieve information about menu items from a restaurant.
- *Checkout*: Allows customers to checkout their shopping cart by notifying the restaurant that they ordered from.
 - Uses Restaurant Account Management to retrieve information from a restaurant.

Restaurant Routes: Handles the routing of HTTP requests to handle restaurant data.

- *Restaurant Account Management*:
 - Handles requests involving a restaurant's account information.
- *Restaurant Promotion Management*:
 - Handles requests involving a restaurant's promotional information.
- *Restaurant-Order Management*:
 - Handles requests involving a restaurant's order information.
- *Restaurant Query Management*:
 - Handles requests involving a restaurant's account information.

User Route: Handles the routing of HTTP requests to handle user data.

- Uses User Management to manage a user's data in the database.
- Uses User-Shopping Cart Management to manage a user's shopping cart data.

Post Routes: Handles the routing of HTTP requests to handle post data.

Media Upload Routes: Handles the routing of HTTP requests to upload media.

- *Image Upload*: Handle requests involving image upload.
- *Video Upload*: Handle requests involving video upload.

Authentication Route: Allows for requests to authenticate users and receive information about authenticated users.

- Uses 3rd Party Authentication to communicate with Google's API.

Restaurant Multimedia Storage: Storage used to store images and videos uploaded from a user.

Database: Database used to store information regarding users, restaurants and posts.

6 Verification

6.1 Unit Testing Strategy

Our approach towards unit testing primarily focused on testing the functions in the front end that involved requesting data from the back-end routes. This was done by creating mock instances of the components, and then mocking the responses that would be received in the back-end. This was because many of our back-end routes were straight forward, and did not involve significant transformations of either the requests or the responses, and as a result, we focused on ensuring that the test cases correctly called front end state changes and that the correct back end functions are called at the appropriate times.

```
// Tests the constructor initializes state variables and makes call to get data
describe('constructor', () => {
    it('should set state variables correctly and call _getRestaurantList', () => {
        expect(instance.state.name).toEqual(mockRestaurant.name);
        expect(instance.state.id).toEqual(mockRestaurant._id);
        expect(instance.state.address).toEqual(mockRestaurant.address);
        expect(instance.state.description).toEqual(mockRestaurant.longDescription);
        expect(instance.state.phoneNumber).toEqual(mockRestaurant.phoneNumber);
        expect(instance.state.activeTab).toEqual("info");
        expect(getRestaurantListSpy).toHaveBeenCalledWith(mockRestaurant._id);
    });
});

// Tests _setTab changes to correct active tab
describe('_setTab', () => {
    it('should set activeTab on button click', () => {
        event = {
            target: {name: "menu"}
        }
        expect(instance.state.activeTab).toEqual("info");
        page.find('.nav-link').at(0).simulate('click', event);

        expect(setStateSpy).toHaveBeenCalledWith({
            activeTab: event.target.name
        });
        expect(instance.state.activeTab).toEqual("menu");
    });
});
```

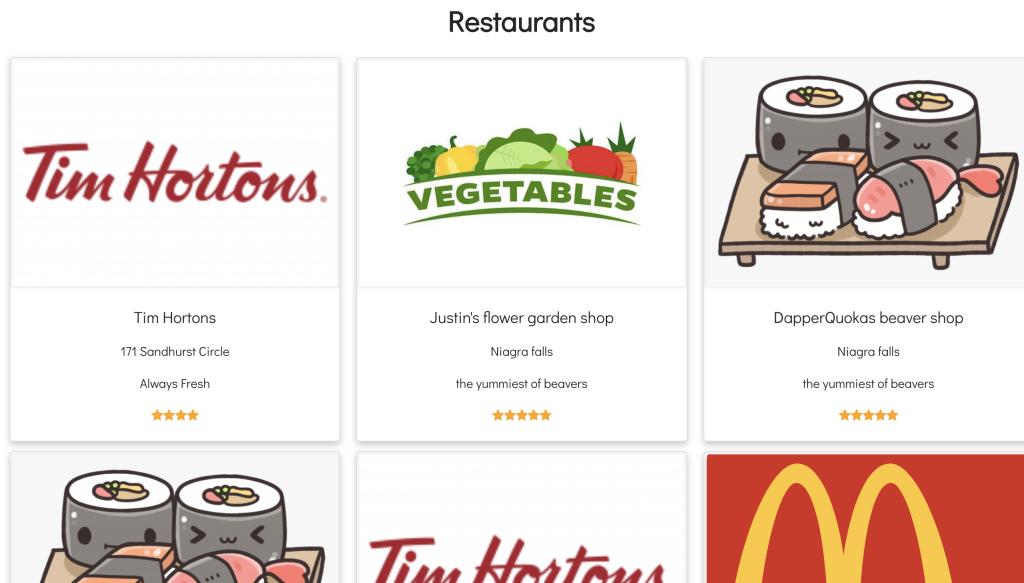
7 Validation

7.1 Sprint 5 Meetings

7.1.1 Friday, July 24 [30 minutes]:

We showed our clients:

- Google registration
 - The UI and layout of the website
 - Ability to upload restaurant information
 - Ability to register an account via Google
 - Ability to see a list of restaurants, although not fully connected to the backend
 - Ability to click on a restaurant in the list and their restaurant profile which includes a picture and description of the restaurant
-



Tim Hortons

Always Fresh
171 Sandhurst Circle | 123-456-7890

Tim Hortons.

Menu | Info | Announcements

Some of the questions we had were:

- Are there any elements of the UI that should be changed?
- How should the layout of the restaurant profile look like?
- How should we handle video uploads to the website?

Specific feedback about our release:

- The layout of the restaurant profile page was clarified. The picture of the restaurant owner should be on the left and cover approximately 25% of the page. On the right of the picture, there should be a paragraph that describes the restaurant owner. Finally, there should be a video below the picture and the text.
- The header of the restaurant profile should be made to look nicer and the “Info” section should be the first page that shows up on the restaurant profile.
- Owner will upload videos via YouTube and we should embed it on the restaurant page.
- The header should be accessible to the user at all times. It should be changed to a floating header.
- Modifications to the restaurant object needs to be made. This includes removing the address field, as well as the ratings field. Instead, add the cuisine types for each restaurant.
- On the restaurant list, add more information on each restaurant card such as cuisine types and establishment date.
- Ratings of restaurants are unnecessary. As well, search by ratings is unnecessary.

Changes made based on the feedback:

henryhhuang commented 14 days ago • edited

- Edited the restaurant profile tabs
- tabs clear the info page if switching to menu or announcements

Chipotle Mexican Grill

The screenshot shows the updated restaurant profile page for Chipotle Mexican Grill. The header now features the restaurant's name prominently. Below the header is a group photo of several Chipotle employees in black uniforms. The main content area contains a large amount of placeholder text (Lorem ipsum) and a small image of a dish.

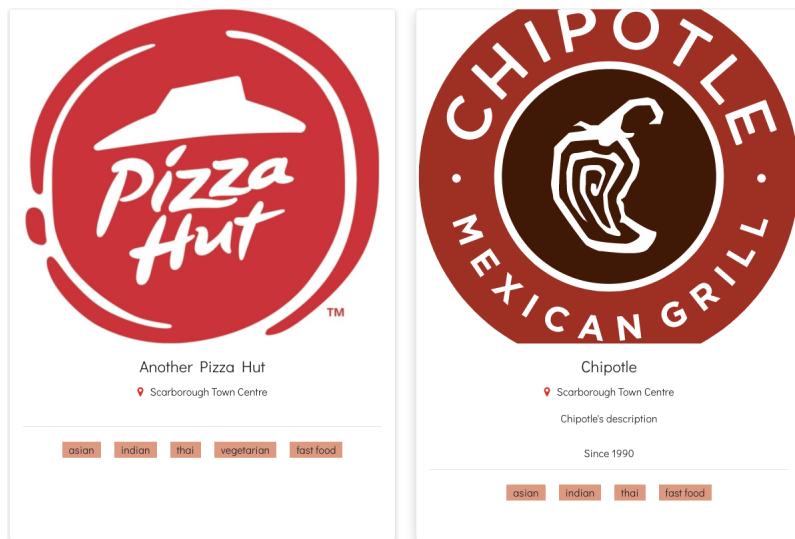
• Also fixed some padding issues on smaller devices

The restaurant profile page has now been updated to showcase the restaurant owner and description of the restaurant owner. As well, the header of the restaurant profile was made to look nicer and starts with the “Info” section, as specified by the client. These changes follow the layout specified from the meetings.

Home Sign-Up for an Account Register Restaurant Manage Restaurant Information Log In

The screenshot shows the updated restaurant profile page. The header now includes the restaurant's name and a small image. Below the header is a group photo of employees. At the bottom of the page is a YouTube video player showing a video titled "Carson - Good Food, Good Person".

The restaurant profile page also has an embedded Youtube video at the bottom. We also changed the header so that it is accessible to the user at all times, as well as modified the fields in the restaurant model according to the feedback.



We added more information to each restaurant card in the restaurant list such as address, description, and establishment date.

7.2 Sprint 6 Meetings

7.2.1 Friday, July 31 [25 minutes]:

We showed our clients:

- The UI and layout of the website
- Ability to upload restaurant information
- Ability to see a list of restaurants
- Ability to see the information and ratings of restaurants in the restaurant list
- Ability to click on a restaurant in the list and their restaurant profile which includes a picture, description of the restaurant, and introductory video, in the layout desired by the clients
- Ability to manage requested restaurants
- Only restaurants verified through the “Manage Requested Restaurants” page will show up in the restaurant list

Restaurant Name	Restaurant Address	City	Postal Code	Owner	Email	Phone Number		
UTSC's Indian Restaurant	1265 Military Trail	Toronto	M1C1A4	Sayon	sayon.sivakumaran@mail.utoronto.ca	1234567890		
ChickNdelish	3 Heale avenue	Scarborough	M1N3X6	Sayon	sayon.sivakumaran@mail.utoronto.ca	1234567890		
Hakka legend	3 Heale avenue	Scarborough	M1N3X6	Sayon	sayon.sivakumaran@mail.utoronto.ca	1234567890		
ChickNdelish	3 Heale avenue	Scarborough	M1N3X6	Sayon	sayon.sivakumaran@mail.utoronto.ca	1234567890		
Tim Hortons	Scarborough Town Centre	Scarborough	ABCDEF	Timmy	timhortons1@email.com	1234567890		
Subway	Scarborough Town Centre	Scarborough	ABCDEF	SubwayOwner2	subway1@test.com	1234567890		
Timmy Test	Scarborough Town Centre	Scarborough	ABCDEF	TimmyTest	timmytest@test.com	1234567890		<small>Handled</small>

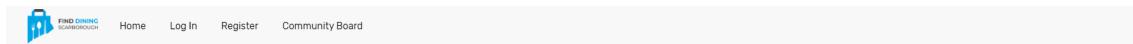
Some of the questions we had were:

- Where should the search bar be located on the website?
- Are the ordering and delivery features mandatory to have?
- Are there any elements of the UI that should be changed?
- Is it necessary to have partial searching or is exact searching okay?

Specific feedback about our release:

- Search bar should be located at the top right of the header.
- Restaurant owners will not upload information. The Scarborough Dining team will upload information when the owner requests it.
- The ordering process should not deal with payment. We only need to provide an interface for it.
- Delivery features such as specifying a delivery order and delivery tracking are not required to be implemented.
- Integrate the given official logo into the website's header.
- It would be nice to have partial searching, but exact searching would be enough.

Changes made based on the feedback:



We added the official logo to the website's header, as requested by the client.



Placed the search bar to the far right of the header.

A screenshot of the search results page titled "Search Results". The search term "fast food" is entered in the search bar. Three results are displayed: 1) "Pizza Hut" located at "Piazza Hut Street" with a "test description" and categories "fast food", "vegan", and "vegetarian". 2) "Another Pizza Hut" located at "Scarborough Town Centre" with categories "asian", "indian", "thai", "vegetarian", and "fast food". 3) "Chipotle" located at "Scarborough Town Centre" with a "Since 1990" note, a "Chipotle's description" link, and categories "asian", "indian", "thai", and "fast food".

localhost:3000/#/search_results/fast food

Added exact searching to the search functionality.

As well, we removed several user stories regarding payment and delivery.

8 Retrospection

8.1 Project Overview

How did your project progress through the term?

Throughout the duration of the project from deliverable 1 to deliverable 5 the project taken on many forms during its development. Development started off slow in deliverable 1, where most of the time was put into researching and learning new technologies. In deliverable 1 we implemented hollow front-end components that could be interacted with but did not communicate with the back-end. Most of deliverable 2 was spent connecting the front-end components created in deliverable 1 to the back-end, so that we could have fully functional, albeit basic components. By the end of deliverable 3, all basic features that would constitute a complete product were implemented but most features were very bare bones and not fleshed out. For deliverable 4 the basic features were fleshed out and improved upon, while the team began development on the more complicated features that would complete a functional product. At the end of deliverable 5, we have implemented all the features that the client had requested, and have a product that can be demonstrated. In summary, the project has improved greatly over the course of the term, moving from planning and designing, to the implementation of our first release, all the way to a MVP.

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

During the sprint there were no problems that warranted the need to use the contingency plan. All members were present everyday of the sprint, the team ensured this through conducting daily stand-up meetings. It was evident that all members were active and diligent in the development process of the project.

8.2 Project Velocities

Sprint 5 - Estimated Project Velocity: 58

- Our estimated project velocity was based on our burn down chart from the previous sprint. We estimated to complete 78 points in our last sprint. However, we were only able to complete 30 points. This indicated that our estimation was too high and we should adjust it in the next sprint. We decided to go with 58 points for sprint 5. This is significantly lower than our previous estimation and believe it is manageable as a lot of tasks with notable progress were carried over.

Sprint 5 - Actual Project Velocity: 10

- Our actual velocity was lower than usual. Many of us were busy with assignments, midterms, and interviews, and so our ability to work consistently on the project was hindered. The work that we did in this sprint progressed us to the point of near completion for most of the tasks. However, we did not complete any to the point where we could send them into peer review

and mark them as completed. However, through the progress we made this sprint, we were able to complete a large amount of our work in the next sprint.

Sprint 6 - Estimated Project Velocity: 57

- We decided to go with an estimated velocity similar to the previous sprint. This is primarily due to most of the tasks from the previous sprint were carried over into this sprint. As this was our final sprint, we will be focused on completing the tasks before the deadline. We believe we will be able to finish most of these tasks in this sprint and approach our estimated project velocity.

Sprint 6 - Actual Project Velocity: 51

- Our actual velocity was higher than usual. We completed a lot more tasks than we usually do. This is mainly due to this being our last sprint. We were required to complete the requirements specified by our client. This resulted in us spending a significant more time working on the website compared to the previous sprints.

Did you follow your plan(s) exactly? What difficulties have you encountered?

We did not follow our plan exactly for Sprint 5. As mentioned earlier, we had several team members working on other courses. As a result, we did not com

What is the most important thing you learned from working on this project?

We have learned many things during this project, some important things that we learned were how to manage git repositories effectively, manage our time, and the importance of planning out a software project before actually coding. During the course of this project we learned that we should consistently push and add to our git repository. Due to a lack of preparation and planning during the initial stages of the project, we were consistently behind schedule and had an unclear idea of what the project's system design should look like. We had begun coding before considering the software architecture and as a result, everyone in the team had a slightly different idea of what the software system was supposed to look like and what the end goal of the project was.

Final Product: <https://scarborough-dining.herokuapp.com/#/>