CSCC01 | Deliverable 5

Team 10: Joseph Augustine, Husni Fareed, Jesse Francispillai, Maduvan Kasi, Ahmad Shah

Product Backlog:

Updated as of August 10th, 2020.

Static Table:
_: Restaurant Search Engine _: Restaurant Profile Page _: Online Ordering
_: Simple Social Network _: Analytics Dashboard _: Mobile Website

*New user stories and user stories with accelerated priorities have been highlighted in **bold**.

Priority	User Story	Estimated Cost (in hours)
1	As Steve (Restaurant Owner), I want a page to feature my restaurant's address and contact information, so that my customers can get in touch with me directly.	6
2	As Steve (Restaurant Owner), I want to showcase personal stories and anecdotes about my dishes, so that customers can understand the history behind them.	5
3	As Javier (Customer), I want to search for restaurants based on rating, so that I know which places would draw in the most views to my blog.	9
4	As Steve (Restaurant Owner), I want to upload simple photos of my dishes and restaurant, so that I can promote exactly how I did in traditional media.	4
5	As Fiona (Customer), I want to see videos of how restaurants make their food, so that I can feel safe ordering it for me and my children.	4
6	As Bonny (Customer), I want to search for restaurants by price, so that I can stay within my tight budget.	5
7	As Bonny (Customer), I want to browse a restaurant's menu, so I can decide what to order beforehand.	9
8	As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog.	10
9	As Diane (Restaurant Owner), I want to see how often various menu items are ordered, so that I can decide where to expand and where to cut back.	10

10	As Javier (Customer), I want to share my thoughts on restaurants, so that I can grow my brand and public influence.	12
11	**As Steve (Restaurant Owner), I want a quick and simple way to log in to my account, so that I don’t waste time working on my restaurant.**	**18**
12	**As Fiona (Customer), I want to add and remove items from a shopping cart, so that I have an easy way to be mindful about what I’m ordering.**	**18**
13	**As Diane (Restaurant Owner), I want to see how often customers visit my restaurant page, so that I can see if my restaurant is growing in popularity.**	**22**
14	**As Diane (Restaurant Owner), I want to add and remove items from my menu, so that I can always be selling the most-efficient cuisine.**	**13**
15	**As Javier (Customer), I want to explore restaurants on the app without me having to search for them, so that I can be introduced to new restaurants in the area.**	**9**
16	As Fiona (Customer), I want to search for restaurants based on their cuisine, so that I can make sure the food I order meets my dietary needs.	5
17	As Bonny (Customer), I want to order food for pickup, so that I can have it ready when I arrive and save time.	8
18	As Fiona (Customer), I want to order food for delivery, so that I can order on my commute and save time.	4
19	As Steve (Restaurant Owner), I want a simple ticket system to see online orders, so that my kitchen doesn’t waste valuable time learning the new system.	4
20	As Diane (Restaurant Owner), I want to see how often my restaurant appears in various searches, so that I can see what sets me apart from my competitors	6
21	As Steve (Restaurant Owner), I want a daily record of what was ordered, so that I can easily cross-reference with my physical inventory	4
22	As Diane (Restaurant Owner), I want to see a distribution of	5

	when (time/date) my orders come in, so that I can choose effective promotional periods	
23	As Fiona (Customer), I want to see the nutrition labels of any food I order, so that I can ensure it pertains to my dietary needs.	8
24	As Diane (Restaurant Owner), I want to show where my ingredients are sourced from, so that I can differentiate myself from the chain franchises.	6
25	As Diane (Restaurant Owner), I want to see the demographics of the people that order from my restaurant, so that I can hone in on my target audience	6
26	As Diane (Restaurant Owner), I want my customers to be able to share media that links to my restaurant page, so that they can direct their friends there.	4
27	As Javier (Customer), I want to interact with other people's posts, by liking and sharing them, so that I can connect with my friends.	5
28	As Diane (Restaurant Owner), I want to see how customers are interacting with posts about my restaurant, so that I can directly see what they generally like and dislike	4
29	As Diane (Restaurant Owner), I want to make posts that target people who frequent my restaurant, so that I strengthen the 'local community' feel.	7
30	As Javier (Customer), I want to see how often my friends have eaten at a restaurant, so that I can judge how popular and 'trendy' that restaurant is.	8
31	As Bonny (Customer), I want to search for restaurants by location, so that I can make sure I have the means to actually get there.	9
32	As Bonny (Customer), I want a way to see how long it would take me to reach a restaurant via public transit, so that I can ensure I can make it there and back within my schedule.	6
33	As Diane (Restaurant Owner), I want online orders to be sorted by expected pick-up time, so that my kitchen can be efficient with their cooking.	5
34	As Javier (Customer), I want a mobile website with a navigation bar always present, so that I can easily access	4

	every feature with a couple clicks.	
35	As Javier (Customer), I want a mobile website where I can access all content with only vertical scrolling, so that I don’t miss content off to the side or waste time zooming.	4

Release Plan:

Identical to Release Plan presented in Deliverable 4. Sprints are still 1 week long. We felt that no changes were needed. The issues we faced in the retrospection of the first sprint of this deliverable were addressed as we moved onto the next sprint (week). Sprint 1 ran from July 21 to July 27. Sprint 2 ran from July 28 to August 3rd. The past week (August 4 to August 10) was not used as a sprint and no new tasks were worked on. This week was used to fix all bugs and stabilize the deployment, including finalizing all features.

Additionally, when selecting user stories for sprints, we also decided to retrieve a couple of user stories from the ‘Done’ column and re-insert them into the Sprint Backlog. This decision was based on feedback we had received in our Deliverable 4, where the TA had mentioned that features we claimed to have completed were still not functioning properly. In particular, we moved User Story 8 into Sprint Backlog 1 (carried over to Sprint Backlog 2), and we moved User Story 4 into Sprint Backlog 2.

Sprint Plan:

Deliverable 5, Sprint 1

User Stories: 8, 10, 11, 12, 13

Deliverable 5, Sprint 1 Backlog:

Story 8: As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog

Tasks:

- T1: Create UI to allow user to edit information fields (Sprint 2)
- T2: Validate customer registration fields (Sprint 1)
- T3: Update customer information in the backend (Sprint 2)

Acceptance Criteria:

- Given Javier wants to create a profile page for his restaurant, when Javier clicks on the “Customer Signup” button, a web form with relevant fields should appear for him to fill out.

- Given Javier has completed filling out the form to create a customer, when Javier clicks on the “Complete” button, he should be directed to a profile page for his restaurant, with the relevant info published.
- Given Javier wants to edit his restaurant’s profile page, when Javier clicks on the “Edit Profile Page” button, the fields on his page should become editable text boxes.
- Given Javier has finished editing his restaurant’s profile page, when Javier clicks on the “Complete” button, the textboxes should become immutable once again.
- Given Javier is entering information into fields for his restaurant’s profile page, when Javier tries to submit an inappropriate entry (e.g. alphabet characters in “phone number” field), he should be presented with a pop-up detailing the error(s).

Story 10: As Javier (Customer), I want to share my thoughts on restaurants, so that I can grow my brand and public influence.

Tasks:

- T1: Create schema for reviews/thoughts in db
- T2: Create backend POST method for posting reviews/comments
- T3: Create the Front End for the review form
- T4: Create the front end for the review display page
- T5: Link front-end UI to backend methods
- T6: Test end-to-end functionality to ensure fulfillment of the user story.

Acceptance Criteria:

- Given that Javier wants to leave a review for a restaurant, he can click a button to have a review form pop-up
- Given that Javier wants to leave a review for a restaurant, he can select a numerical rating between 1-5, and enter his thoughts in text form.
- Given that Javier wants to leave a review for a restaurant, he can click on a button to save the review and have it show up on the restaurant’s profile page.

Story 11: As Steve (Restaurant Owner), I want a quick and simple way to log in to my account, so that I don’t waste time working on my restaurant.

Tasks:

- T1: Add appropriate packages to the project to support authentication (Sprint 1)
- T2: Replace Restaurant and Customer creation with registration (Sprint 1)
- T3: Have authenticate function to validate the user session (Sprint 2)
- T4: Implement authorization in the web app to restrict each respective user’s access (Sprint 2)
- T5: Perform Sanity Test, create both restaurant owner and customer (Sprint 2)

Acceptance Criteria:

- Given that Steve wants to log in, he is able to navigate from any page on the web app, to the login and registration form.
- Given that Steve doesn't have an account, he can register for an account.
- Given that Steve does have an account, he can login to his account.
- Once logged in, Steve has access restricted to his restaurant's account only, allowing him to edit any attributes about his restaurant.
- No other users, customers or other restaurant owners, have access to edit Steve's Restaurant on the web app.

Story 12: As Fiona (Customer), I want to add and remove items from a shopping cart, so that I have an easy way to be mindful about what I'm ordering.

Tasks:

- T1: Create Cart Schema
- T2: Store cart in client session
- T3: Create page to view cart items and change quantities
- T4: Link restaurant menus to user's cart
- T5: Create warnings to indicate when cart is moved to a different restaurant
- T6: Sanity/Unit test all aspects of cart system

Acceptance Criteria:

- Given Fiona is on a restaurant's menu, when she clicks "Add Item To Cart", the item she selected should be added to her cart with a visual acknowledgement.
- Given Fiona is viewing her cart, when she clicks "Add" next to an item in her cart, the quantity of that item should go up by 1, and the new quantity should be displayed on the page.
- Given Fiona is viewing her cart, when she clicks "Remove" next to an item with quantity >= 2 in her cart, the quantity of that item should go down by 1, and the new quantity should be displayed on the page.
- Given Fiona is viewing her cart, when she clicks "Remove" next to an item with quantity = 1 in her cart, the item should be removed from her cart and the page.
- Given Fiona has a cart from one restaurant, when she clicks "Add Item To Cart" on a different restaurant's menu, a popup should appear warning her that her cart will be replaced.

Story 13: As Diane (Restaurant Owner), I want to see how often customers visit my restaurant page, so that I can see if my restaurant is growing in popularity.

Tasks:

- T1: Update Restaurant Schema to hold data collection for customer visits to the page.
- T2: Ensure that data is dated so that comparisons can be made on a month-to-month basis.

- T3: Create UI for the Analytics Page
- T4: Link the backend to the frontend so that the Analytics Page Loads dynamically.
- T5: Increase Page visits every time the page is visited (NOT COMPLETED)
- T6: Perform Unit Testing to ensure that the page can load changing data.

Acceptance Criteria:

- Given that Diane wants to see how many customers are visiting her restaurant's page, she must be able to view the dashboard from the restaurant's profile page.
- Given that Diane wants to see how many customers are visiting her restaurant's page, she can view the growth in the popularity of her restaurant by comparing the amount of visits between different time frames.

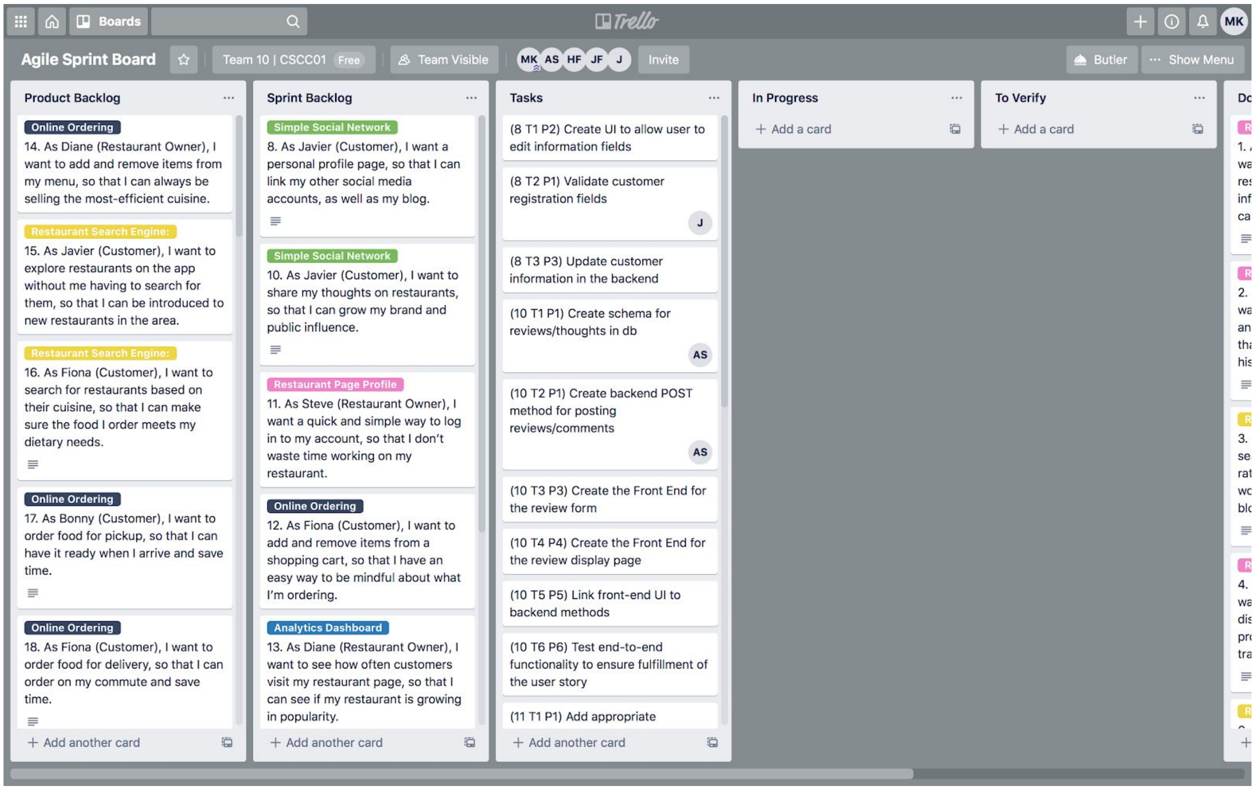
Deliverable 5, Sprint 1: July 21, 2020 - July 27, 2020:

Story Number corresponds to priority in the actual sprint backlog

Story	Task	Cost	Priority	Day 1	Day 2	Day 3	Day 4	Day 5	Assigned
8	T2	2	3	1	0	0	0	**1**	Joseph
10	T1	3	1	1	**2**	0	0	0	Ahmad
	T2	3	2	0	1	1	**1**	0	Ahmad
11	T1	3	1	0	1	1	**1**	0	Husni
	T1	3	2	0	0	0	2	**1**	Husni
12	T1	1	1	**1**	0	0	0	0	Maduvan
	T2	4	1	3	**1**	0	0	0	Maduvan
	T3	2	2	0	**2**	0	0	0	Maduvan
	T4	3	3	0	0	**3**	0	0	Maduvan
	T5	2	4	0	0	0	**2**	0	Maduvan
	T6	6	5	0	1	1	2	**2**	Jesse
13	T1	4	1	1	1	**2**	0	0	Husni
	T2	2	2	0	0	**2**	0	0	Husni
	T3	2	3	0	1	0	**1**	0	Husni
	T4	4	4	0	0	0	2	**2**	Husni
	T5	6	5	0	0	1	1	0	Husni

	T6	1	6	0	0	0	0	**1**	Jesse
Total		51		7	10	11	12	7	
Work Left		51		50	41	32	22	6	

Deliverable 5, Sprint 1, Snapshots at Start of Sprint



Agile Sprint Board | Team 10 | CSCC01 Free | Team Visible

Product Backlog

- Online Ordering
  14. As Diane (Restaurant Owner), I want to add and remove items from my menu, so that I can always be selling the most-efficient cuisine.
- Restaurant Search Engine:
  15. As Javier (Customer), I want to explore restaurants on the app without me having to search for them, so that I can be introduced to new restaurants in the area.
- Restaurant Search Engine:
  16. As Fiona (Customer), I want to search for restaurants based on their cuisine, so that I can make sure the food I order meets my dietary needs.
- Online Ordering
  17. As Bonny (Customer), I want to order food for pickup, so that I can have it ready when I arrive and save time.
- Online Ordering
  18. As Fiona (Customer), I want to order food for delivery, so that I can order on my commute and save time.

+ Add another card

Sprint Backlog

- Simple Social Network
  8. As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog.
- Simple Social Network
  10. As Javier (Customer), I want to share my thoughts on restaurants, so that I can grow my brand and public influence.
- Restaurant Page Profile
  11. As Steve (Restaurant Owner), I want a quick and simple way to log in to my account, so that I don't waste time working on my restaurant.
- Online Ordering
  12. As Fiona (Customer), I want to add and remove items from a shopping cart, so that I have an easy way to be mindful about what I'm ordering.
- Analytics Dashboard
  13. As Diane (Restaurant Owner), I want to see how often customers visit my restaurant page, so that I can see if my restaurant is growing in popularity.

+ Add another card

Tasks

- (8 T1 P2) Create UI to allow user to edit information fields
- (8 T2 P1) Validate customer registration fields — J
- (8 T3 P3) Update customer information in the backend
- (10 T1 P1) Create schema for reviews/thoughts in db — AS
- (10 T2 P1) Create backend POST method for posting reviews/comments — AS
- (10 T3 P3) Create the Front End for the review form
- (10 T4 P4) Create the Front End for the review display page
- (10 T5 P5) Link front-end UI to backend methods
- (10 T6 P6) Test end-to-end functionality to ensure fulfillment of the user story
- (11 T1 P1) Add appropriate

+ Add another card

In Progress

+ Add a card

To Verify

+ Add a card

Dc

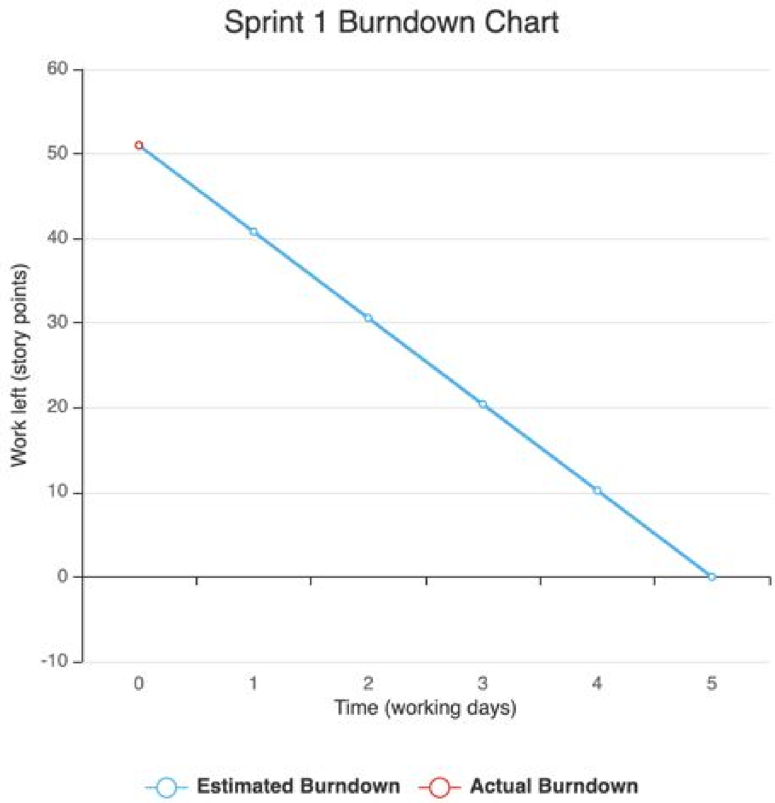
R
1. /
wa
res
inf
ca

R
2.
wa
an
tha
his

R
3.
se
rat
wc
blc

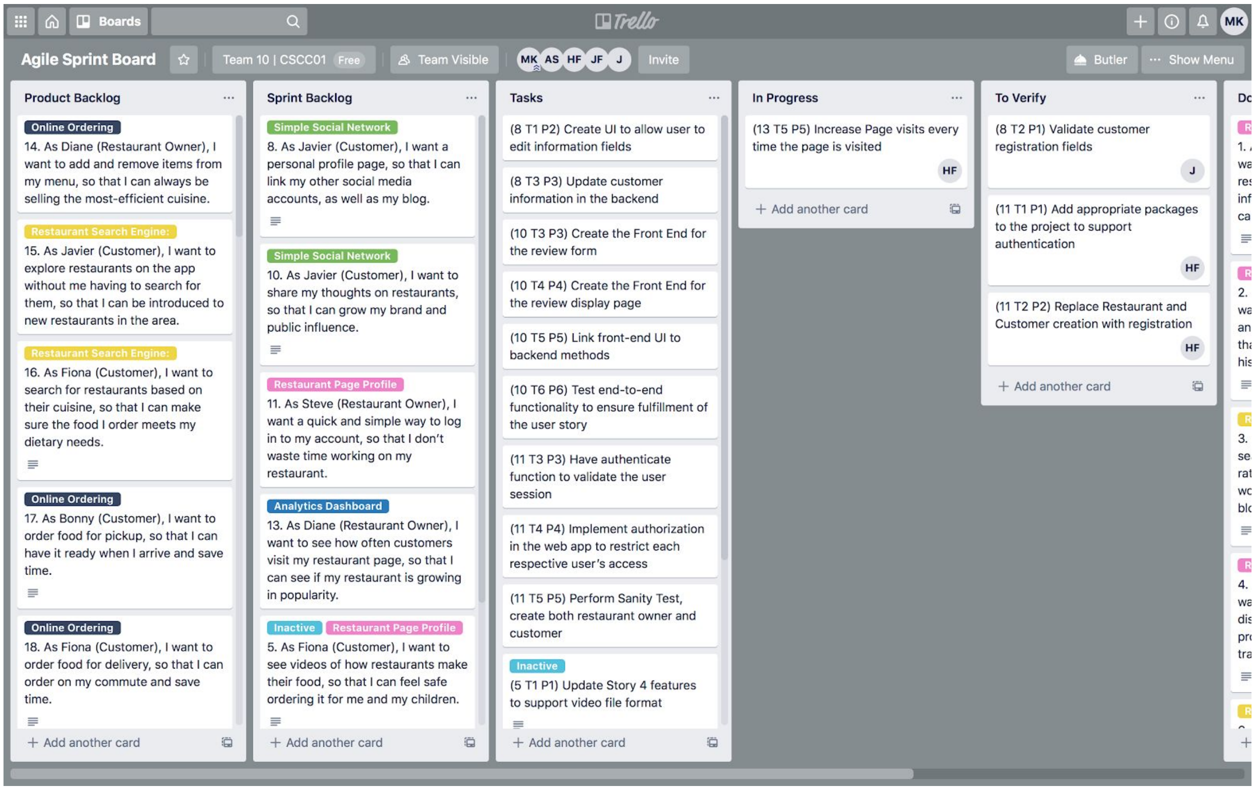
R
4.
wa
dis
pr
tra

R



Actual Burndown (x,y) Coordinates: (0, 51)

Deliverable 5, Sprint 1, Snapshots at End of Sprint



Agile Sprint Board

Team 10 | CSCC01 Free

Team Visible

MK AS HF JF J

Invite

Product Backlog

Online Ordering
14. As Diane (Restaurant Owner), I want to add and remove items from my menu, so that I can always be selling the most-efficient cuisine.

Restaurant Search Engine:
15. As Javier (Customer), I want to explore restaurants on the app without me having to search for them, so that I can be introduced to new restaurants in the area.

Restaurant Search Engine:
16. As Fiona (Customer), I want to search for restaurants based on their cuisine, so that I can make sure the food I order meets my dietary needs.

Online Ordering
17. As Bonny (Customer), I want to order food for pickup, so that I can have it ready when I arrive and save time.

Online Ordering
18. As Fiona (Customer), I want to order food for delivery, so that I can order on my commute and save time.

+ Add another card

Sprint Backlog

Simple Social Network
8. As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog.

Simple Social Network
10. As Javier (Customer), I want to share my thoughts on restaurants, so that I can grow my brand and public influence.

Restaurant Page Profile
11. As Steve (Restaurant Owner), I want a quick and simple way to log in to my account, so that I don't waste time working on my restaurant.

Analytics Dashboard
13. As Diane (Restaurant Owner), I want to see how often customers visit my restaurant page, so that I can see if my restaurant is growing in popularity.

Inactive Restaurant Page Profile
5. As Fiona (Customer), I want to see videos of how restaurants make their food, so that I can feel safe ordering it for me and my children.

+ Add another card

Tasks

(8 T1 P2) Create UI to allow user to edit information fields

(8 T3 P3) Update customer information in the backend

(10 T3 P3) Create the Front End for the review form

(10 T4 P4) Create the Front End for the review display page

(10 T5 P5) Link front-end UI to backend methods

(10 T6 P6) Test end-to-end functionality to ensure fulfillment of the user story

(11 T3 P3) Have authenticate function to validate the user session

(11 T4 P4) Implement authorization in the web app to restrict each respective user's access

(11 T5 P5) Perform Sanity Test, create both restaurant owner and customer

Inactive
(5 T1 P1) Update Story 4 features to support video file format

+ Add another card

In Progress

(13 T5 P5) Increase Page visits every time the page is visited
HF

+ Add another card

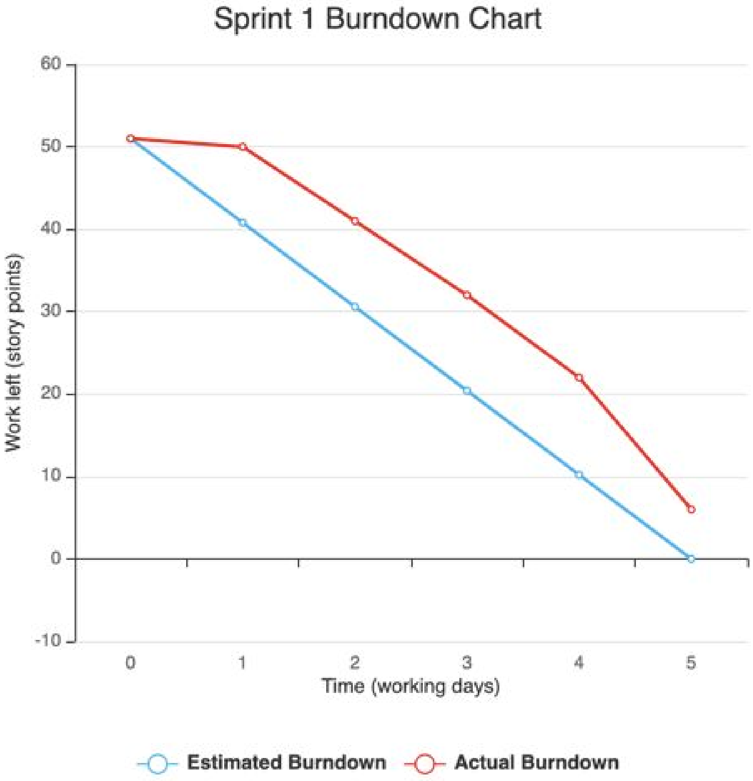
To Verify

(8 T2 P1) Validate customer registration fields
J

(11 T1 P1) Add appropriate packages to the project to support authentication
HF

(11 T2 P2) Replace Restaurant and Customer creation with registration
HF

+ Add another card



Actual Burndown (x,y) Coordinates: (0, 51), (1, 50), (2, 41), (3, 32), (4, 22), (5, 6)

Deliverable 5, Sprint 2

User Stories: 4, 8, 10, 11, 14, 15 [Details for stories 8, 10, and 11 are found in Sprint 1]

Deliverable 5, Sprint 2 Backlog:

Story 4: As Steve (Restaurant Owner), I want to upload simple photos of my dishes and restaurant, so that I can promote exactly how I did in traditional media.

Tasks:

- T1: Store images directly in DB
- T2: Add front-end forms for uploading images to profile page, stories, and menu items
- T3: Create preview for image forms before upload to server

Acceptance Criteria:

- Given Steve wants to upload a photo, and clicks the "Upload Photo" button, he should be directed to a form where he can submit a photo, along with its title and description.
- Given Steve has not completed filling out all of the required fields, and clicks the "Upload" button, he should be prompted that he must first complete all the fields.
- Given Steve has completed filling out all of the required fields, and clicks the "Upload" button, he should be redirected to his restaurant's profile page, with the new photo present.

Story 14: As Diane (Restaurant Owner), I want to add and remove items from my menu, so that I can always be selling the most-efficient cuisine.

Tasks:

- T1: Create dynamic front-end system for editing menu
- T2: Create form for uploading new menu items
- T3: Create backend methods for adding and removing menu items from DB
- T4: Sanitize inputs and sanity test backend methods

Acceptance Criteria:

- Given Diane is viewing her menu, when she clicks "New Item", she should be directed to a form to upload a new menu item.
- Given Diane has completed the form to upload her menu item, when she clicks "Submit", she should be directed back to her menu, with the new item present.
- Given Diane is viewing her menu, when she clicks "Delete", the she selected should be removed from her restaurant's menu and the page.

Story 15: As Javier (Customer), I want to explore restaurants on the app without me having to search for them, so that I can be introduced to new restaurants in the area.

Tasks:

- T1: Create the Explore page and a mock-up UI in the web app.
- T2: Create Backend request to collect data to send to the Explore page.
- T3: Link Backend to the Explore page so that it loads the data dynamically.
- T4: Perform Unit Testing by creating new restaurant with varying prices and ratings, and seeing that it shows up in the explore page.

Acceptance Criteria:

- Given that Javier wants to find a new restaurant, he can navigate to the explore page on the web app.
- Given that Javier is on the explore page, he has the option to view the many restaurants based off different params such as ratings and price.
- Given that Javier is interested in learning more about a restaurant on the explore page, he can directly navigate to the restaurant's profile page from the explore page.

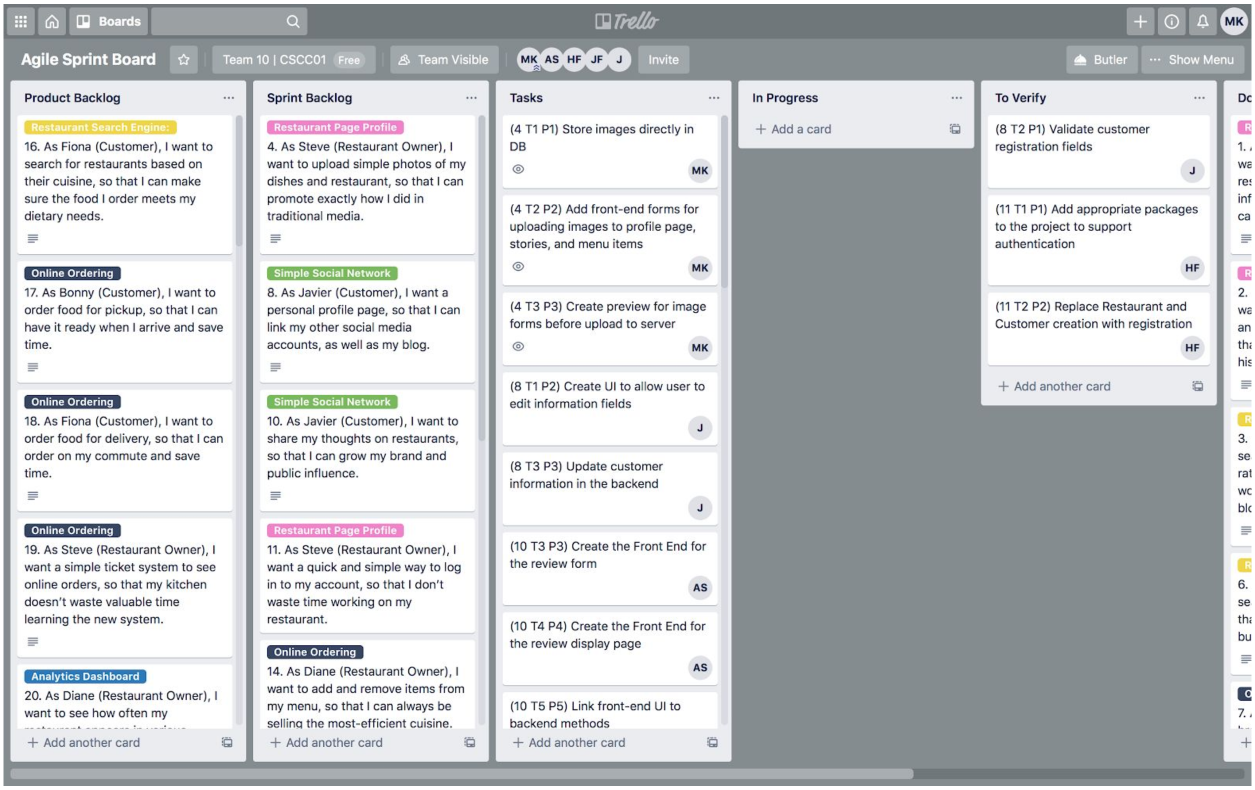
Deliverable 5, Sprint 2: July 28, 2020 - Aug 3, 2020:

Story Number corresponds to priority in the actual sprint backlog

Story	Task	Cost	Priority	Day 1	Day 2	Day 3	Day 4	Day 5	Assigned
4	T1	3	1	**3**	0	0	0	0	Maduvan
	T2	1	2	0	**1**	0	0	0	Maduvan
	T3	1	3	0	**1**	0	0	0	Maduvan
8	T1	2	3	0	1	**1**	0	0	Joseph
	T3	4	4	0	0	0	2	**2**	Joseph
10	T3	5	0	0	0	2	**3**	0	Ahmad
	T4	6	0	0	0	0	2	**4**	Ahmad
	T5	3	0	0	0	0	0	**3**	Ahmad
	T6	2	0	0	0	0	0	**2**	Jesse
11	T3	6	3	0	0	**6**	0	0	Maduvan
	T4	4	4	0	0	1	**3**	0	Maduvan
	T5	2	5	0	0	0	1	**1**	Jesse

14	T1	4	1	0	0	0	3	**1**	Maduvan
	T2	2	1	0	**2**	0	0	0	Maduvan
	T3	4	2	0	0	0	0	**4**	Maduvan
	T4	3	3	0	0	0	0	**3**	Jesse
15	T1	2	1	1	**1**	0	0	0	Husni
	T2	2	2	0	1	**1**	0	0	Husni
	T3	3	3	0	0	0	2	**1**	Husni
	T4	2	4	0	0	0	0	**2**	Jesse
Total		61		4	7	11	16	23	
Work Left		61		58	52	42	33	0	

Deliverable 5, Sprint 2, Snapshots at Start of Sprint



Product Backlog

- Restaurant Search Engine: 16. As Fiona (Customer), I want to search for restaurants based on their cuisine, so that I can make sure the food I order meets my dietary needs.
- Online Ordering 17. As Bonny (Customer), I want to order food for pickup, so that I can have it ready when I arrive and save time.
- Online Ordering 18. As Fiona (Customer), I want to order food for delivery, so that I can order on my commute and save time.
- Online Ordering 19. As Steve (Restaurant Owner), I want a simple ticket system to see online orders, so that my kitchen doesn't waste valuable time learning the new system.
- Analytics Dashboard 20. As Diane (Restaurant Owner), I want to see how often my

+ Add another card

Sprint Backlog

- Restaurant Page Profile 4. As Steve (Restaurant Owner), I want to upload simple photos of my dishes and restaurant, so that I can promote exactly how I did in traditional media.
- Simple Social Network 8. As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog.
- Simple Social Network 10. As Javier (Customer), I want to share my thoughts on restaurants, so that I can grow my brand and public influence.
- Restaurant Page Profile 11. As Steve (Restaurant Owner), I want a quick and simple way to log in to my account, so that I don't waste time working on my restaurant.
- Online Ordering 14. As Diane (Restaurant Owner), I want to add and remove items from my menu, so that I can always be selling the most-efficient cuisine.

+ Add another card

Tasks

- (4 T1 P1) Store images directly in DB — MK
- (4 T2 P2) Add front-end forms for uploading images to profile page, stories, and menu items — MK
- (4 T3 P3) Create preview for image forms before upload to server — MK
- (8 T1 P2) Create UI to allow user to edit information fields — J
- (8 T3 P3) Update customer information in the backend — J
- (10 T3 P3) Create the Front End for the review form — AS
- (10 T4 P4) Create the Front End for the review display page — AS
- (10 T5 P5) Link front-end UI to backend methods

+ Add another card

In Progress

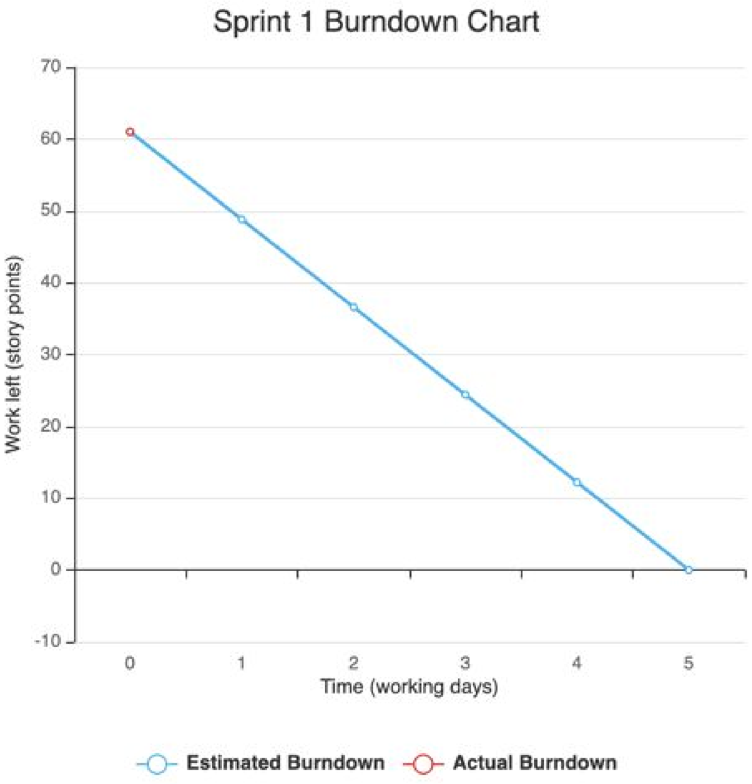
+ Add a card

To Verify

- (8 T2 P1) Validate customer registration fields — J
- (11 T1 P1) Add appropriate packages to the project to support authentication — HF
- (11 T2 P2) Replace Restaurant and Customer creation with registration — HF

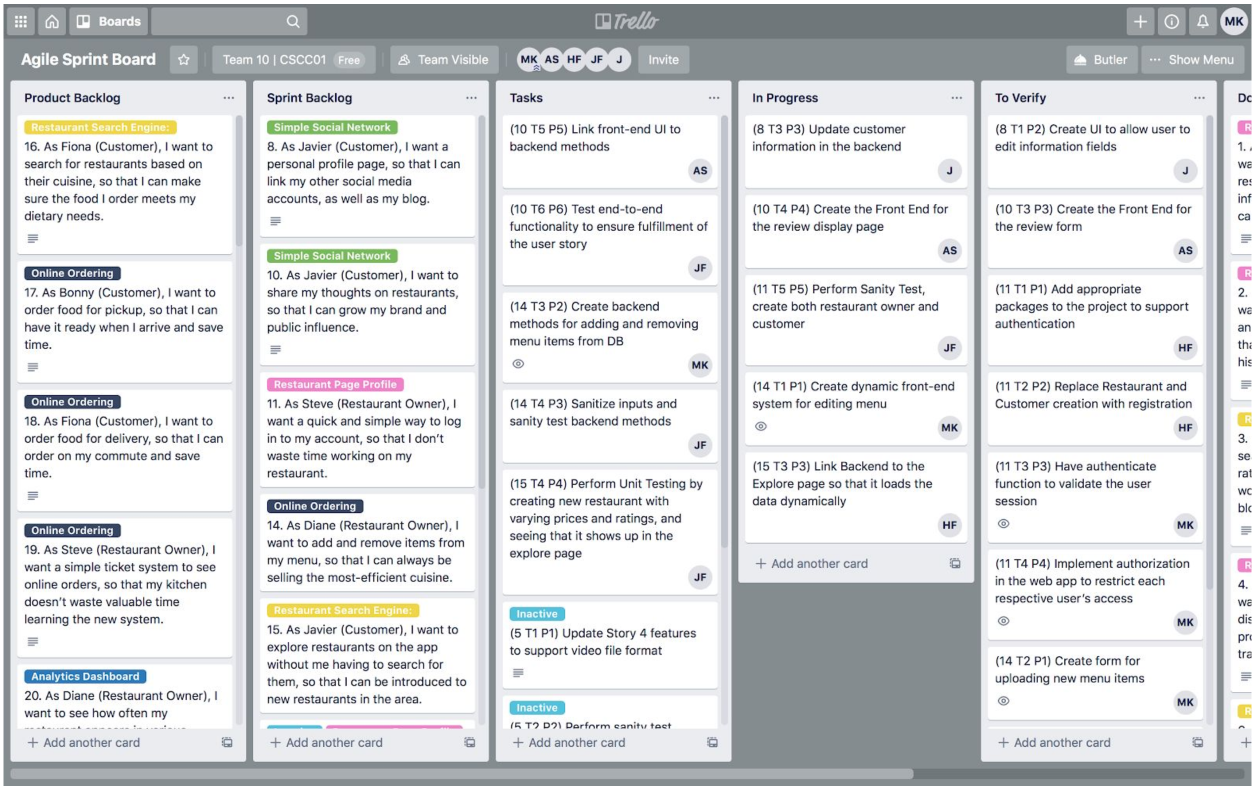
+ Add another card

Dc



Actual Burndown (x,y) Coordinates: (0, 61)

Deliverable 5, Sprint 2, Snapshots on Day 4 of Sprint



Product Backlog

- Restaurant Search Engine: 16. As Fiona (Customer), I want to search for restaurants based on their cuisine, so that I can make sure the food I order meets my dietary needs.
- Online Ordering 17. As Bonny (Customer), I want to order food for pickup, so that I can have it ready when I arrive and save time.
- Online Ordering 18. As Fiona (Customer), I want to order food for delivery, so that I can order on my commute and save time.
- Online Ordering 19. As Steve (Restaurant Owner), I want a simple ticket system to see online orders, so that my kitchen doesn't waste valuable time learning the new system.
- Analytics Dashboard 20. As Diane (Restaurant Owner), I want to see how often my

+ Add another card

Sprint Backlog

- Simple Social Network 8. As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog.
- Simple Social Network 10. As Javier (Customer), I want to share my thoughts on restaurants, so that I can grow my brand and public influence.
- Restaurant Page Profile 11. As Steve (Restaurant Owner), I want a quick and simple way to log in to my account, so that I don't waste time working on my restaurant.
- Online Ordering 14. As Diane (Restaurant Owner), I want to add and remove items from my menu, so that I can always be selling the most-efficient cuisine.
- Restaurant Search Engine: 15. As Javier (Customer), I want to explore restaurants on the app without me having to search for them, so that I can be introduced to new restaurants in the area.

+ Add another card

Tasks

- (10 T5 P5) Link front-end UI to backend methods AS
- (10 T6 P6) Test end-to-end functionality to ensure fulfillment of the user story JF
- (14 T3 P2) Create backend methods for adding and removing menu items from DB MK
- (14 T4 P3) Sanitize inputs and sanity test backend methods JF
- (15 T4 P4) Perform Unit Testing by creating new restaurant with varying prices and ratings, and seeing that it shows up in the explore page JF
- Inactive (5 T1 P1) Update Story 4 features to support video file format
- Inactive (5 T2 P2) Perform sanity test

+ Add another card

In Progress

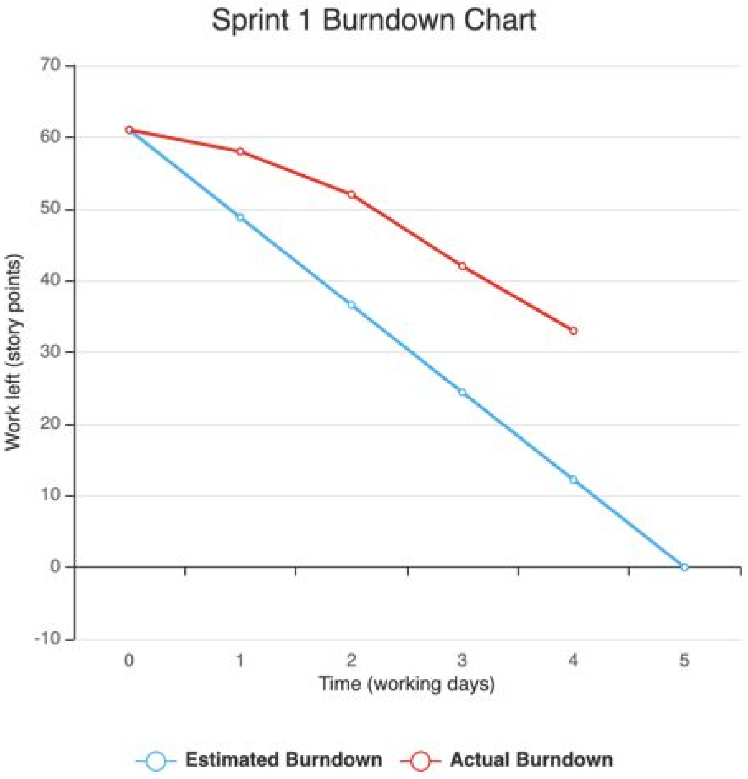
- (8 T3 P3) Update customer information in the backend J
- (10 T4 P4) Create the Front End for the review display page AS
- (11 T5 P5) Perform Sanity Test, create both restaurant owner and customer JF
- (14 T1 P1) Create dynamic front-end system for editing menu MK
- (15 T3 P3) Link Backend to the Explore page so that it loads the data dynamically HF

+ Add another card

To Verify

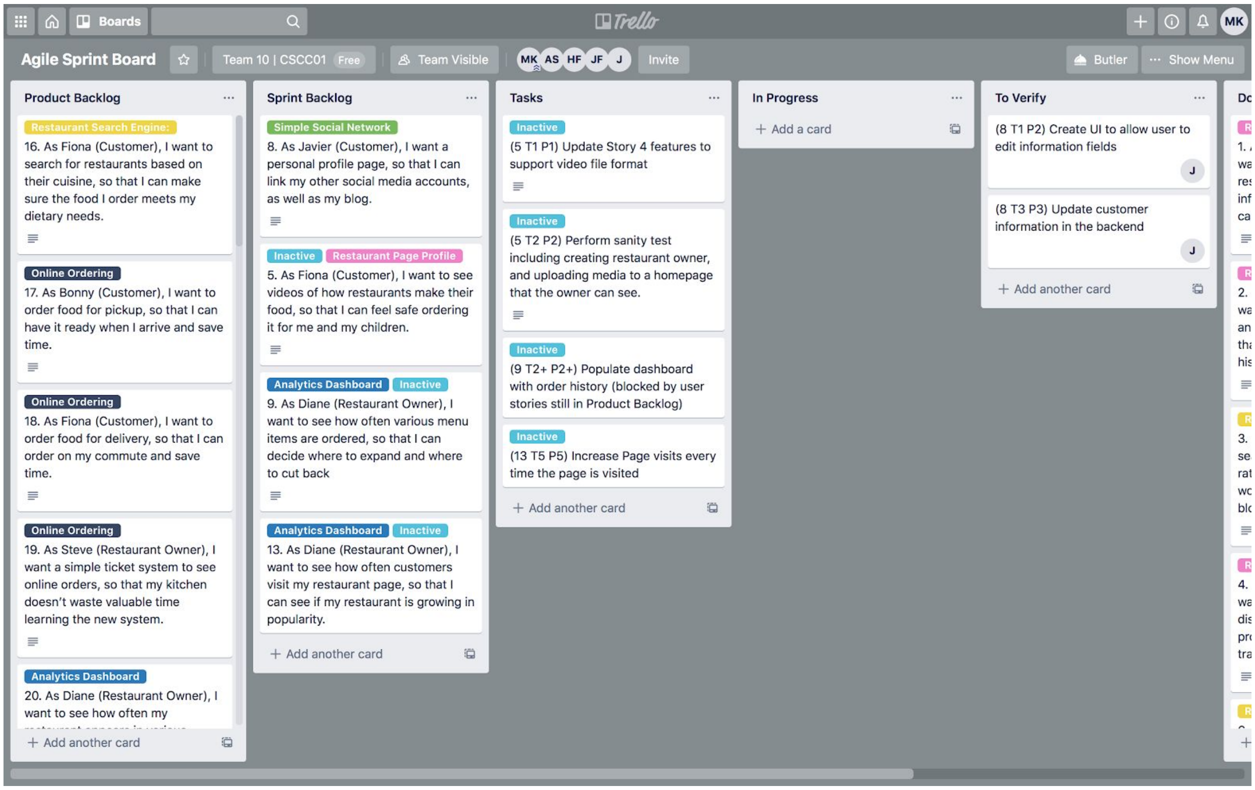
- (8 T1 P2) Create UI to allow user to edit information fields J
- (10 T3 P3) Create the Front End for the review form AS
- (11 T1 P1) Add appropriate packages to the project to support authentication HF
- (11 T2 P2) Replace Restaurant and Customer creation with registration HF
- (11 T3 P3) Have authenticate function to validate the user session MK
- (11 T4 P4) Implement authorization in the web app to restrict each respective user's access MK
- (14 T2 P1) Create form for uploading new menu items MK

+ Add another card



Actual Burndown (x,y) Coordinates: (0, 61), (1, 58), (2, 52), (3, 42), (4, 33)

Deliverable 5, Sprint 2, Snapshots at End of Sprint



Product Backlog

- Restaurant Search Engine: 16. As Fiona (Customer), I want to search for restaurants based on their cuisine, so that I can make sure the food I order meets my dietary needs.
- Online Ordering 17. As Bonny (Customer), I want to order food for pickup, so that I can have it ready when I arrive and save time.
- Online Ordering 18. As Fiona (Customer), I want to order food for delivery, so that I can order on my commute and save time.
- Online Ordering 19. As Steve (Restaurant Owner), I want a simple ticket system to see online orders, so that my kitchen doesn't waste valuable time learning the new system.
- Analytics Dashboard 20. As Diane (Restaurant Owner), I want to see how often my
- + Add another card

Sprint Backlog

- Simple Social Network 8. As Javier (Customer), I want a personal profile page, so that I can link my other social media accounts, as well as my blog.
- Inactive Restaurant Page Profile 5. As Fiona (Customer), I want to see videos of how restaurants make their food, so that I can feel safe ordering it for me and my children.
- Analytics Dashboard Inactive 9. As Diane (Restaurant Owner), I want to see how often various menu items are ordered, so that I can decide where to expand and where to cut back
- Analytics Dashboard Inactive 13. As Diane (Restaurant Owner), I want to see how often customers visit my restaurant page, so that I can see if my restaurant is growing in popularity.
- + Add another card

Tasks

- Inactive (5 T1 P1) Update Story 4 features to support video file format
- Inactive (5 T2 P2) Perform sanity test including creating restaurant owner, and uploading media to a homepage that the owner can see.
- Inactive (9 T2+ P2+) Populate dashboard with order history (blocked by user stories still in Product Backlog)
- Inactive (13 T5 P5) Increase Page visits every time the page is visited
- + Add another card

In Progress

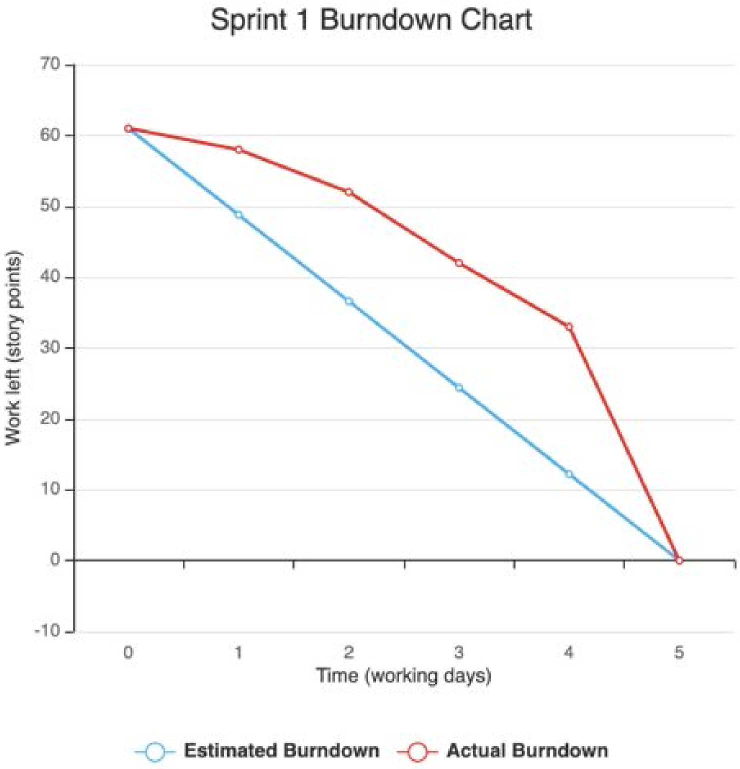
- + Add a card

To Verify

- (8 T1 P2) Create UI to allow user to edit information fields J
- (8 T3 P3) Update customer information in the backend J
- + Add another card

Do

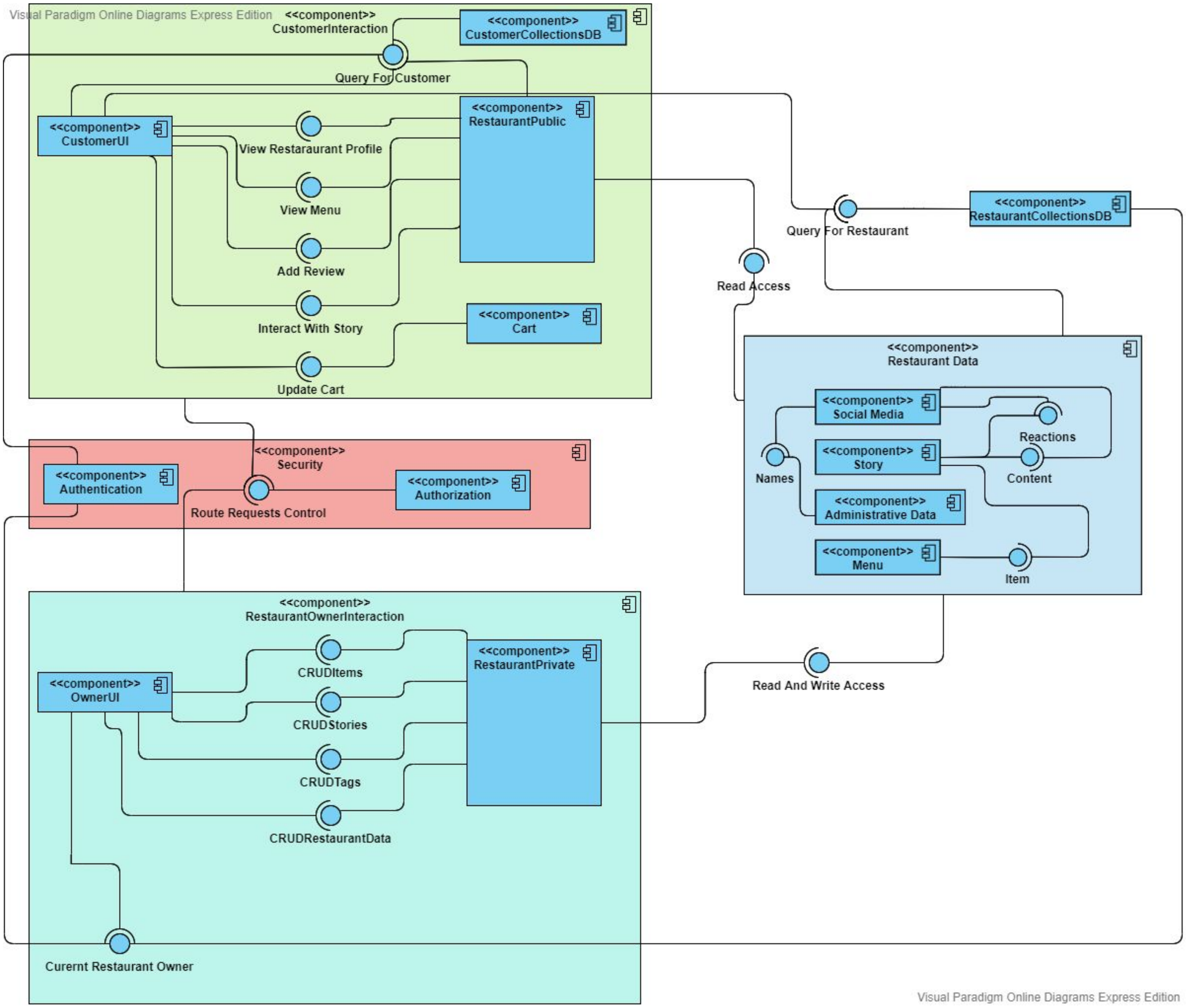
- R 1. wa res inf ca
- R 2. wa an tha his
- R 3. se rat wo blo
- R 4. wa dis pro tra
- R



Actual Burndown (x,y) Coordinates: (0, 61), (1, 58), (2, 52), (3, 42), (4, 33), (5, 0)

Component Diagram & Report

Component Diagram:



The component diagram report is structured such that indented components represent subcomponents of the components diagram above

Report:

CustomerInteraction <<component>>:

- Concerns itself with all the logistics of how a customer user interactions with the web application including both the front end and interactive components. This component is dependant on the “Route Requests Control” interface because based on the respective type of user using the web app, the server must be able to prevent certain requests from being made.
- **CustomerCollectionsDB <<component>>:**
    - The collection of the registered users in the Database.
    - **Query For Customer <<Interface>>:**
        - Provides functionality for the caller to retrieve a customer object from the collection in the database.
- **CustomerUI <<component>>:**
    - The front end that the customer interacts with, it depends on a lot of the interfaces as the UI depends on the access to interactions that the customer should have.
- **RestaurantPublic <<component>>:**
    - Represents the overall restaurant object that the user is able to interact with. It is dependent on the access given from the Restaurant Data object which delegates what data should be sent to the UI as well as querying customers because of user based data stored in the restaurant object.
    - **View Restaurant Profile <<Interface>>:**
        - Let’s the customer view the Restaurant Profile Page
    - **View Menu <<Interface>>:**
        - Let’s the customer view a Restaurant’s menu
    - **Add Review <<Interface>>:**
        - Let’s the customer add a review to the specified restaurant
    - **Interact With Story <<Interface>>:**
        - Let’s the user view the story that the stories that the restaurant interacts with.
- **Cart <<component>>:**
    - The object that holds the user’s selection for to-be order.
    - **Update Cart <<Interface>>:**
        - Let’s the user add and remove items to the cart.

Security <<component>>:

- Responsible for the overall protection of restaurant owners and customer users on the web app.

- **Authorization <<component>>:**
    - Used to ensure that the current user has and doesn’t have access to different parts of the website.
    - **Route Requests Control <<interface>>:**
        - Ensures that the access to resources are protected on the server side.
- **Authentication <<component>>:**
    - Used to start the user session in the web app so that either the restaurant owner or customer can use their respective access.
    - **Query For Customer <<Interface>>:**
        - Provides functionality for the caller to retrieve a customer object from the collection in the database.

RestaurantOwnerInteraction <<component>>:

- Concerns itself with all the logistics of how a restaurant owner user interactions with the web application including both the front end and interactive components.
- This component is dependant on the “Route Requests Control” interface because based on the respective type of user using the web app, the server must be able to prevent certain requests from being made.
- **OwnerUI <<component>>:**
    - The front end that the restaurant owner interacts with, it depends on a lot of the interfaces as the UI depends on the access that the customer should have.
- **RestaurantPrivate <<component>>:**
    - Represents the overall restaurant object that the restaurant owner is able to interact with. It is solely dependent on the access given from the Restaurant Data object which delegates what data should be sent to the UI.
    - **CRUDItems <<interface>>:**
        - Let’s the restaurant owner manipulate items in the menu.
    - **CRUDStories <<interface>>:**
        - Let’s the restaurant owner manipulate stories from the restaurant in the web app.
    - **CRUDTags <<interface>>:**
        - Let’s the restaurant owner manipulate tags from the restaurant in the web app.
    - **CRUDRestaurantData <<interface>>:**
        - Let’s the restaurant owner manipulate profile-based data from the restaurant in the web app.

Restaurant Data <<component>>:

- The component that contains the restaurant object and well as the sub components and the interactions between them to show the inner parts of the restaurant object.
- **Social Media <<component>>:**

  - The marketing perspective of the restaurant which depends on story content, reactions, and names from the administrative data component to create the presentable component in the Web App.
- **Story <<component>>:**
  - The component which contains the data related to each story in the restaurant object in the web app.
  - **Content <<interface>>:**
    - Provides the story content to the caller.
  - **Reactions <<interface>>:**
    - Provides the stories interactions with respect to likes and comments.
- **Administrative Data <<component>>:**
  - The component that contains the core data relevant to the the restaurant including names, email addresses, phone numbers, etc.
  - **Names <<interface>>:**
    - Provides the Restaurant and Owner’s name to the caller.
- **Menu <<component>>:**
  - The component that contains the menu items of the restaurant
  - **Items <<interface>>:**
    - Provides the items of the menu to the caller.
- **Read And Write Access <<interface>>:**
  - Gives read and write access to the respective Restaurant component by passing specific data with will eventually be provided in the front end.
- **Read Access <<interface>>:**
  - Gives read only access to the respective Restaurant component by passing specific data with will eventually be provided in the front end.

RestaurantCollectionsDB <<component>>:

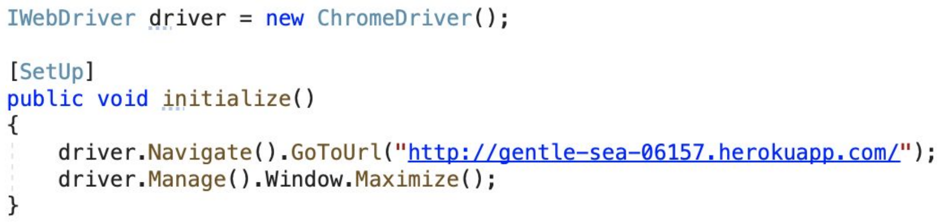
- The collection of the registered restaurants / restaurant owners in the Database.
- **Query For Restaurant <<interface>>:**
  - Provides functionality for the caller to retrieve a restaurant object from the collection in the database.

Verification and Validation: Unit Testing

Each major component/feature that has been implemented in the final deployment of Scarborough Dining is covered by fully automated, front-end unit testing. The entire unit testing strategy is built using the **Selenium Automation Framework** written in **C# with Microsoft Visual Studio**. Selenium is a software tool that allows the automation of functional unit testing for web applications. The tests are documented and grouped appropriately by major features (RestaurantSignUp, CustomerSignUp, RestaurantOrderNow, AddRestaurantStory, etc.) using **NUnit** which is a unit testing framework used alongside Selenium in Visual Studio.

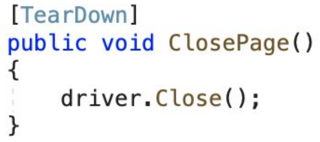
When a new feature has been developed, automation scripts are written which can then be run over and over again with the click of a button. Once an automation test script is written, it runs fully automated. There is no redundancy in the run unless a part of the feature is broken in which case, it will be reported as a failure by NUnit. Each test takes about average 30 seconds of automation once started.

Each test begins by running a SetUp method which opens the Chrome browser, navigates to the URL where Scarbrough Dining is deployed, and maximizes the browser window.



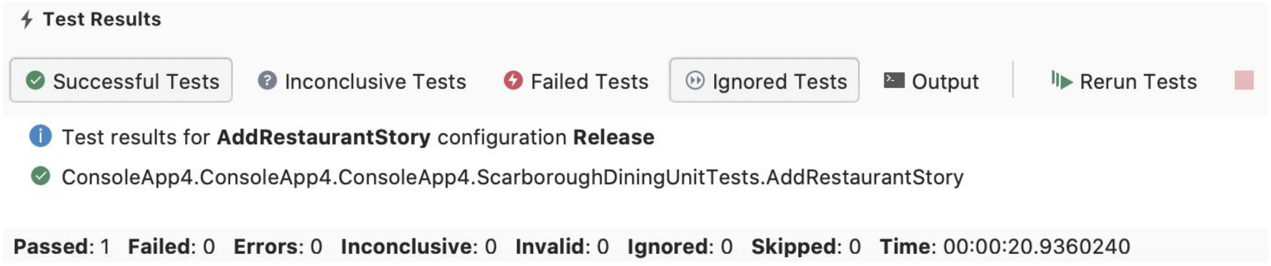
```
IWebDriver driver = new ChromeDriver();

[SetUp]
public void initialize()
{
    driver.Navigate().GoToUrl("http://gentle-sea-06157.herokuapp.com/");
    driver.Manage().Window.Maximize();
}
```

Each test ends by running a Teardown method which simply closes the browser once the automation test is completed.



```
[TearDown]
public void ClosePage()
{
    driver.Close();
}
```

This unit testing framework can also be used as quick and easy sanity testing for quality assurance. Any developer can simply run all unit tests anytime with the click on one button and the tests will all run automatically and return an NUnit test report showing passed and failed tests.



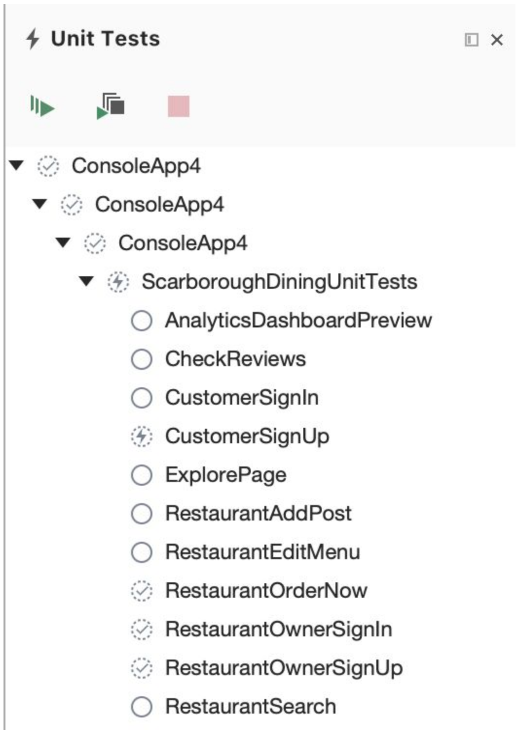
Test results for **AddRestaurantStory** configuration **Release**

ConsoleApp4.ConsoleApp4.ConsoleApp4.ScarboroughDiningUnitTests.AddRestaurantStory

Passed: 1 **Failed**: 0 **Errors**: 0 **Inconclusive**: 0 **Invalid**: 0 **Ignored**: 0 **Skipped**: 0 **Time**: 00:00:20.9360240

The code for all unit tests written thus far can be found in the CSCC01/team_10-project repository insider folder “UnitTests”. Below is the list of unit tests which are grouped appropriately by major feature.

- RestaurantOwnerSignUp
- RestaurantOwnerSignIn
- CustomerSignUp
- CustomerSignIn
- RestaurantEditMenu
- RestaurantAddPost
- RestaurantOrderNow
- RestaurantSearch
- ExplorePage
- CheckReviews
- AnalyticsDashboardPreview



Unit Tests

```
▼ ConsoleApp4
  ▼ ConsoleApp4
    ▼ ConsoleApp4
      ▼ ScarboroughDiningUnitTests
          ○ AnalyticsDashboardPreview
          ○ CheckReviews
          ○ CustomerSignIn
            CustomerSignUp
          ○ ExplorePage
          ○ RestaurantAddPost
          ○ RestaurantEditMenu
            RestaurantOrderNow
            RestaurantOwnerSignIn
            RestaurantOwnerSignUp
          ○ RestaurantSearch
```

Prioritized Features:

FROM HIGHEST TO LOWEST:

1. Profile page
2. Search Engine
3. Authentication
4. Menu
5. Reviews
6. Ordering System
7. Analytics Dashboard
8. Social Network
9. Mobile Interface

Feature capabilities:

Currently the features have been implemented include a restaurant owner and customer profile management, a basic social network, an ordering system,as well as a basic analytical dashboard.

Profile Page: They both have unique capabilities based on login where a restaurant owner can edit their menu and a customer can order from those restaurants.When creating an account whether it be owner or customer the user is not allowed to input wrong values into their registration.This portion has been validated and tested completely disallowing any wrong phone numbers/ emails in particular. However on the profile page there is an edit profile button which redirects the user to the registration form. This part has not been completed yet and currently is not working. The UI was not heavily prioritized including resizing.

Search Engine: The UI of this feature is connected to the permanent navigation bar on the top of every page and allows for users to search for restaurants using various parameters and be navigated to a page containing the returned restaurants from the query. First, the user can search by relevencence, which returns back the closest matching restaurants with their name containing or matching the search text. Second, the user can search by either increasing or decreasing price. Third, the user can search by rating which returns restaurants from the query in a decreasing rating order. The search engine is able to also compare the search string to restaurant tags, for example if a user entered "chicken" restaurants with the tag "chicken" would also be included in the return query, along with actual restaurants whose names contain 'chicken'. The search engine does not account for fuzzy-spelling checks so if the user enters a misspelled search text, the search engine will return no restaurants from the query.

Authentication: In terms of the user authentication both owners and customers are able to login to their account. They both have unique capabilities based on login where a restaurant owner can edit their menu and a customer can order from those restaurants.They are also able to freely signout from their session and login at any point.In terms of the user authentication both owners and customers are able to login to their account. This however was not fully tested and edge cases not checked.

Social network:This features priority was lowered due to the other features being more crucial to the main application.A really basic social network sample was created which is just links to users social media platforms which are allowed to be linked in their profile.

Reviews: The review system is fully functional, and allows users to post reviews by selecting a rating from 1-5, along with typing in a comment. Upon submission of the review, the page is updated to show the new review. If a user is not logged in, they can still view other people's reviews, however they cannot post one until they log in. The Review display also supports pagination.

Analytics dashboard: We decided to lower the priority of this as well because it was believed to be an extension of a viable product. What we were able to create was a basic UI which represents what the board would look like, as well as load data into this UI dynamically from the database - the restaurant schema was updated to hold basic analytical data. None of the information reflects any change made by users however contains portions of their information. What we did also add on was an explorer home page which acts similar to an analytical dashboard. It sorts the restaurants into generic categories and allows the customer to choose one of the following to visit.

Ordering System: The ordering system takes the user to a page with all the menu items currently on sale. Once the user decides what they want to order, they can click on the item to add it to the cart. Once added, the user's cart (located in the navbar) is updated with the item and the user can continue to select additional items to order. Once the user is satisfied, he can click on his cart and he will be taken to a checkout page where they can increase the quantity of the ordered item or remove it from their cart. The actual ordering is not implemented yet.

Menu: Similar to reviews, shows a list of all the currently offered menu items, along with their price, name and description. This form does not allow incorrect input and requires all fields. The restaurant owner of that menu would be able to both create new items as well delete existing items. Customers and other restaurant owners are not given this capability. These features are given based on the users authentication. This menu also includes a pagination which will start paginating items on the menu after every 10th item. The entire menu system has been fully validated and tested. Resizing is not fully implemented and was not prioritized.

Mobile Website: The creation of a mobile friendly website was not prioritized therefore no research was put into requirements needed to run on mobile.

Retrospection:

Brief overview of your entire project saga:
At the beginning of our project, we reviewed extensively the requirements put forth by Scarborough Dining. As we finished the last deliverable of the term today, we were able to accomplish almost all aspects of the minimum viable product and complete more than half of the stories in the Product Backlog. Because we used Agile Development, we were able to quickly find and fix problems in our development sprints which allowed the next sprint to account for these. All in all, the minimum viable product that was produced and deployed at the end of this project showcases the development from each team member and the product follows most requirements from Scarborough Dining.

Estimated project velocity?:
We created our sprint plan using 30-minute intervals as being equal to one story point and each sprint is 5 working days (One week). This deliverable had two sprints. In sprint 1 of this deliverable, we estimated 5 user stories with a total estimated points of 51. In sprint 2 of this deliverable, we estimated 6 user stories with 3 of these stories being leftovers from the first sprint. The total estimated points was 61. The amount of story points we planned for each sprint as we went through each deliverable 1 through 5 increased drastically. The first and second deliverables had average 15 story points planned while the very last sprint in deliverable 5 had 61 points planned showing a dramatic increase over the course of this timeline.

Actual project velocity? Did it change?:
At the end of Sprint 1, we finished 45 story points with 6 story points left over. At the end of Sprint 1, we finished all 61 story points that were estimated. As we neared the end of sprint 2, we completed 23 story points on day 5 to finish the sprint with 0 story points left over. We had to put in extra effort as a team to make sure we kept up with the estimated story points to finalize our project for deliverable 5.

Did we follow our plan exactly or did we have to replan?:
We had two sprints during this deliverable and during the first sprint of this deliverable, our team lost a few working days as we were all sidetracked by other course work. Due to this, we were also unable to attend a few team meetings that were planned in the sprint. We did not need to replan because most of the work was assigned independently so we simply caught up at the next team meeting.

What difficulties have we encountered?
One big difficulty we faced was time constraints from fixing many bugs regarding functionality and UI on older features while still making sure new features were included in our sprint plans so that we could keep up with Scarborough Dining’s requirements. Another issue we faced was when a story was dependent on multiple people, it was difficult to coordinate because many

team members had their own busy schedules. This was fixed by trying to have more frequent team meetings to improve on team communication.

Was our contingency plan useful at these points?:
Our contingency plan was not used directly during these issues, however we were able to troubleshoot many of these issues in the retrospection of the first sprint so that we can improve the second sprint of this deliverable.

Most important things we learned from working on this project?:
This project taught each team member the entire Agile Development SDLC and gave each of us a very hands-on experience as we worked with a real client and a real pitch. We were able to learn the importance of communication as a team during our sprint meetings and the importance of documenting/planning each sprint using tools like the Taskboard. We gained new insight into component diagrams as we learned about the coupling/cohesion of many components in a real end-to-end project.

How does the work done for deliverable 5 compare to the work on all previous deliverables, both in terms of progress and end result?:
Deliverable 5 had a much bigger emphasis on validation of our work such as unit testing. We were able to get a framework set up for automated unit testing whereas it was not completed in other deliverables. More time was also spent on fixing bugs and UI issues to make the final product more presentable. As a team, we were also able to use Agile much more smoothly as we learned all the basics in Deliverables 1 through 4 which made deliverable 5 Agile Development much more efficient.