

DELIVERABLE 4



Team_01(GitHub)
Anson Feng
Jiawei Qiao
Vanessa Pierre
Yolanda Su

Table Of Contents

Project Backlog	4
Release Plan	8
Sprint 2: Sprint Plan	8
Sprint 2: Acceptance Criteria	10
Sprint 2: Team Task Assignments	11
Sprint 2: Task Board	13
Sprint Start	13
Sprint End	13
Sprint 2: Burndown Chart	14
Sprint Start	14
Sprint End	14
Sprint 3: Sprint Plan	15
Sprint 3: Acceptance Criteria	16
Sprint 3: Team Task Assignments	16
Sprint 3: Task Board	17
Sprint Start	17
Sprint End	17
Sprint 3: Burndown Chart	18
Sprint Start	18
Sprint End	18
Sprint 4: Sprint Plan	19
Sprint 4: Acceptance Criteria	21
Sprint 4: Team Task Assignments	23
Sprint 4: Task Board	24
Sprint Start	24
Sprint End	25
Sprint 4: Burndown Chart	25
Sprint Start	25
Sprint End	26
High-level Architecture	26
Validation	32
Project Velocity	33

Estimated Velocity	33
Actual Velocity	33
Changes from Deliverable 3	33
Retrospective	34
Release	35

Project Backlog

1 point = 1 working hour

Greyed out rows represent stories that are already done.

Persona 1: David (Restaurant Owner)

Type	Priority	Story	Back-end Points	Front-end Points	Test Automation Points	Total Story Points
Registration						
Story	Critical	As David (restaurant owner), I want to be able to register my account by using my name, email, password and my restaurant name and address.	6.5	4	3.5	14
Coupons						
Story	High	As David (restaurant owner), I want to create a coupon with a name, redemption cost in points, description, and an optional expiration and begin date.	4	3	2.5	9.5
Story	High	As David (restaurant owner), I want to be able to see all my restaurant's current coupons.	1	6	2	9
Story	High	As David (restaurant owner), I want to be able to delete an existing coupon.	1	1	2	4
Story	High	As David (restaurant owner), I want a graphical interface allowing me to see how many redeemed coupons customers have at my restaurant.	1	2	3	6
Story	Low	As David (restaurant owner), I want to be able to use existing coupons as a template for new coupons.	1	3	3	7
Achievements						
Story	High	As David (restaurant owner), I want to be able to add new achievements to my restaurant by inputting name, description, their rewarding experience and points, and their requirement within two types: 1. Spending X dollars in one visit; 2. Purchasing what "Item" X time.	2	2.5	3	7.5

Story	High	As David (restaurant owner), I want a default template for customer achievements, so that I can set up achievements more easily.	1.5	2	2	5.5
Story	High	As David (restaurant owner), I want to be able to see all my restaurant's current achievements.	1	2	2.5	5.5
Story	High	As David (restaurant owner), I want to be able to delete achievements from my restaurant.	1	1	1.5	3.5
Story	Low	As David (restaurant owner), I want to be able to modify achievements' name, description, experience, points, and requirements from my restaurant.	1	2	2.5	5.5
Restaurant Settings						
Epic	Low	As David (restaurant owner), I want an interface for editing basic restaurant settings such as restaurant name, address, etc.	0.5	5	6	11.5
Employees						
Story	Critical	As David (restaurant owner) I want to be able to create an employee account for my employee by inputting their name, email, and password.	3	2	2.5	7.5
Story	High	As David (restaurant owner), I want to be able to see all employee accounts associated with my restaurant	2	3	3	8
Story	High	As David (restaurant owner), I want to be able to delete accounts associated with my restaurant.	2	2	1	5
Customer Level						
Epic	Low	As David (restaurant owner) I want to be able to set the additional reward points earned by customers based on their user level at my restaurant.	4	4	5	13
Menu						
Epic	Low	As David (restaurant owner), I want a menu system allowing me to upload, preview and modify my menu onto the application.	6	6	6	18

Persona 2: Daniel (Employee)

Type	Priority	Story	Back-end Points	Front-end Points	Test Automation Points	Total Story Points
QR Code Scanning						
Story	High	As Daniel (employee) I want to be able to scan customer coupon QR codes to mark a customer's coupon used.	4	1	3	8
Story	High	As Daniel (employee) I want to be able to scan customer coupon QR codes to update customer progress on a given achievement.	1.5	1	3	5.5

Persona 3: Kevin (Customer)

Type	Priority	Story	Back-end Points	Front-end Points	Test Automation Points	Total Story Points
Registration						
Story	Critical	As Kevin (customer), I want to be able to register my account using my name, email and password.	0.5	0.5	2	3
Search						
Story	Critical	As Kevin (customer), I want a search bar to search for restaurants that are available on the app.	3	2.5	3	8.5
Coupons						
Story	High	As Kevin (customer) I want to be able to see available coupons for exchange at a specific restaurant.	1	1.5	2	4.5
Story	High	As Kevin (customer) I want to be able to purchase coupons by redeeming my points.	2	2.5	2	6.5
Story	High	As Kevin (customer) I want to be able to see what coupons I already redeemed.	1	1	2.5	4.5

Achievements						
Story	High	As Kevin (customer) I want to be able to see what achievements are available.	1	1	3.5	5.5
Story	High	As Kevin (customer), I want a progress bar for each achievement that tracks how close I am to completing the achievement.	1	3	2	6
Story	Low	As Kevin (customer) I want to be able to see a confirmation screen with the number of points earned after completing an achievement.	0.5	1	1.5	3
Points						
Story	High	As Kevin (customer), I want a display for each restaurant that tracks how many points I have earned at that restaurant so far.	1	1	2	4
Customer Level						
Story	High	As Kevin (customer), I want to be able to see my current customer level and the experience needed to reach the next level at a given restaurant.	2.5	3	2	7.5
Story	Medium	As Kevin (customer), I want to see user leaderboards for each restaurant on the app so that I can compare my level with other customers and compete to see who has the highest level at a given restaurant.	2	4	3	9
Customer Profile						
Story	Medium	As Kevin (customer), I want a profile page containing my name, email and a list of links to restaurants that I visited before.	2	2.5	4	8.5
Custom Character						
Epic	Low	As Kevin (customer), I want to be able to customize my user character with equipable character items earned from levelling up at restaurants.	10	11	11	32

We rewrote some stories based on deliverable 3 feedback as they were too vague and not testable. The priorities on some stories were also adjusted based on team discussion and client feedback, and all stories were given an additional story point estimate for automating unit tests, called automation points. We determined these automation points, like all other front-end and back-end story pointing done for our product backlog, via team planning poker.

Release Plan

We did not make any changes to our release plan since deliverable 3. Our sprint length remained as 1 week long, ending on Sundays at 6 PM.

Sprint 2: Sprint Plan

Priority	Story	Back-end Points	Front-end Points	Tasks	Back-end Points	Front-end Points
Critical	As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements.	3	1.75	T009: Design search algorithm for restaurants	1	0
				T010: Code search function with respect to restaurant database	1.5	0
				T012: Implement UI using back-end functions	0	1.75
				T013: Automate tests	0.5	0
Critical	-	-	-	T030: Learn Flask and refactorize with Python	4	4
Critical	As David (restaurant owner) I want to be able to create an employee account for my employee.	3	2	T031: Design employee type accounts for the user account system and associate every employee with a restaurant	1	0
				T032: Code employee account creation relative to	1.5	0

				the user database		
				T033: Design the employee account creation UI for the restaurant owner	0	0.5
				T034: Implement the employee account creation UI for the owner with backend	0	1.5
				T035: Automate tests	0.5	0
High	As David (restaurant owner), I want to be able to delete an existing coupon.	1	1	T036: Code database query for deleting coupons	0.8	0
				T037: Design coupon deletion UI in relation to existing coupon list UI	0	0.25
				T038: Implement coupon deletion UI with backend	0	0.75
				T039: Automate tests	0.2	0
High	As David (restaurant owner), I want to be able to see all employee accounts associated with my restaurant	2	3	T040: Code database query for fetching all employee user accounts associated with a given restaurant	1.5	0
				T041: Design UI for listing employee accounts by employee name/email	0	0.5
				T042: Implement employee list UI with backend	0	2.5
				T043: Automate tests	0.5	0
High	As David (restaurant owner), I want to be able to delete accounts associated with my restaurant.	2	2	T044: Code database query for deleting employee accounts	1	0

				T045: Design employee account deletion UI in relation to existing employee list UI	0	1
				T046: Implement employee account deletion UI with backend	0	1
				T047: Automate tests	1	0

Sprint 2: Acceptance Criteria

Story	Acceptance Criteria
As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements.	<p>Scenario: The customer searches for restaurants that match a given name or keyword.</p> <p>Given that the customer wants to find all restaurants related to a chosen keyword, when the customer inputs the keyword, all restaurants whose names have the given name/keyword as a substring appear.</p> <p>Scenario: The user searches for a restaurant name that is not on the app.</p> <p>Given that the customer wants to find an unavailable restaurant, when the customer inputs the restaurant name into the search bar, no restaurants appear.</p>
As David (restaurant owner) I want to be able to create an employee account for my employee.	<p>Scenario: The restaurant owner creates a new employee account.</p> <p>Given that the restaurant owner has entered a valid email, name, and password, when the owner clicks the create account button, a new employee appears in the list of restaurant employees.</p> <p>Scenario: The restaurant owner creates a new employee account using a duplicated email.</p> <p>Given that there is already a user account with the given email, when the owner tries to create a new employee account with the email, an error message appears and the new employee is not in the list of restaurant employees.</p> <p>Scenario: The restaurant owner creates a new employee account with non-matching information in the two password fields.</p>

	Given that the input in the two password fields does not match, when the owner tries to create a new employee account with the email, an error message appears and the new employee is not in the list of restaurant employees.
As David (restaurant owner), I want to be able to delete an existing coupon.	<p>Scenario: The restaurant owner deletes an existing coupon.</p> <p>Given that the restaurant owner has created at least one coupon, when the owner clicks delete on a coupon entry in the list, the coupon is no longer in from the restaurant's coupon list.</p>
As David (restaurant owner), I want to be able to see all employee accounts associated with my restaurant	<p>Scenario: The restaurant owner opens the employee page and has not added employees.</p> <p>Given that the restaurant owner has not added employees, when the owner opens the employee page, there are no employee entries shown.</p> <p>Scenario: The restaurant owner opens the employee page and has added at least one employee.</p> <p>Given that the restaurant owner has added at least one employee, when the owner opens the employee page, the owner sees a list of all and only employees at the owner's restaurant.</p>
As David (restaurant owner), I want to be able to delete accounts associated with my restaurant.	<p>Scenario: The restaurant owner deletes an existing employee.</p> <p>Given that the restaurant owner has added at least one employee, when the owner clicks delete on an employee entry in the list, the employee is no longer in the restaurant's employee list and a user cannot log in using those employee account credentials.</p>

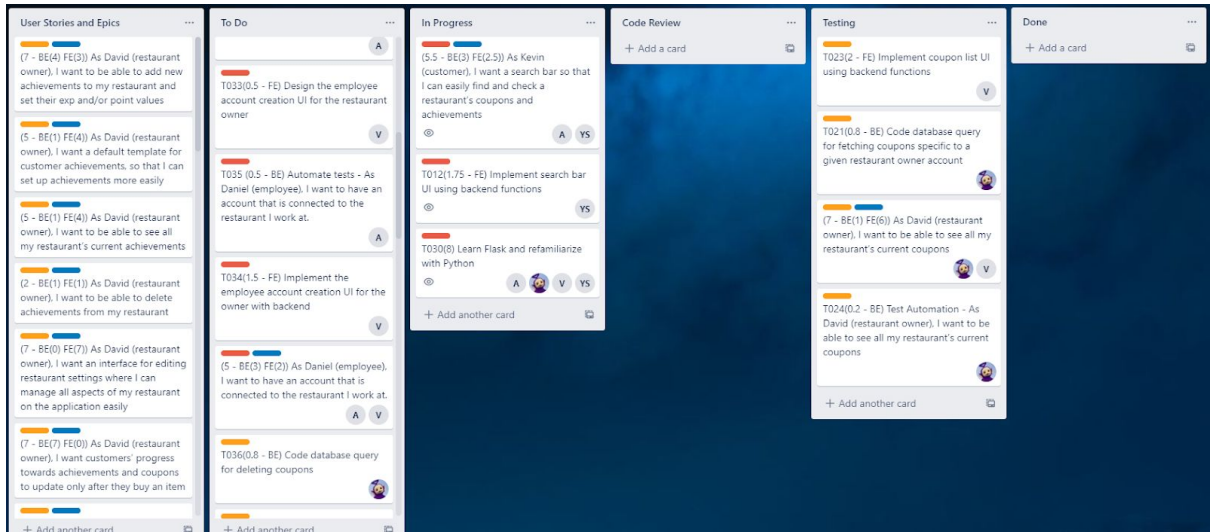
Sprint 2: Team Task Assignments

Task ID	Cost	Priority	Day 0	Day1	Day 2	Day 3	Day 4	Day 5	Day 6	Assignment
T009	1	Critical	0	1	0	0	0	0	0	Anson
T010	1.5	Critical	0	0	0	1.5	0	0	0	Anson
T012	1.75	Critical	0	0	0.5	1.25	0	0	0	Yolanda
T013	0.5	Critical	0	0	0	0.5	0	0	0	Anson
T030.1	2	Critical	0	0	0	0	2	0	0	Yolanda
T030.2	2	Critical	0	0	0	0.5	0.5	0.5	0.5	Anson
T030.3	2	Critical	0	0	0	0	1.5	0.5	0	Jiawei

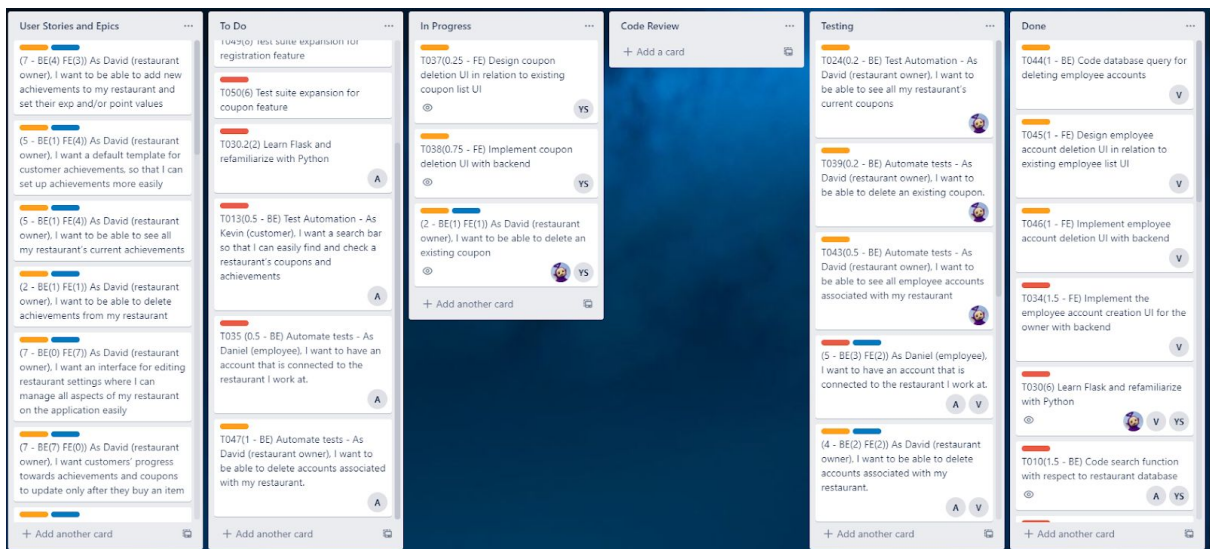
T030.4	2	Critical	0	0	0	0	0	0	2	Vanessa
T031	1	Critical	0	0	0	0	1	0	0	Anson
T032	1.5	Critical	0	0	0	0	0	1.5	0	Anson
T033	0.5	Critical	0	0	0.5	0	0	0	0	Vanessa
T034	1.5	Critical	0	0	0	0	1.5	0	0	Vanessa
T035	0.5	Critical	0	0	0	0	0	0	0.5	Anson
T036	0.8	High	0	0	0	0	0.8	0	0	Jiawei
T037	0.25	High	0	0	0	0	0.25	0	0	Yolanda
T038	0.75	High	0	0	0	0	0.75	0	0	Yolanda
T039	0.2	High	0	0	0	0	0.2	0	0	Jiawei
T040	1.5	High	0	0	0	0	1	0.5	0	Jiawei
T041	0.5	High	0	0	0	0	0	0.5	0	Yolanda
T042	2.5	High	0	0	0	0	0	1	1.5	Yolanda
T043	0.5	High	0	0	0	0	0	0.5	0	Jiawei
T044	1	High	0	0	0	0	1	0	0	Anson
T045	1	High	0	0	0	1	0	0	0	Vanessa
T046	1	High	0	0	0	0	1	0	0	Vanessa
T047	1	High	0	0	0	0	0	1	0	Anson
Total Points Completed:			0	1	1.5	5.75	9.5	7	7	
Total Points Left:			31.75	30.75	29.25	23.5	14	7	0	

Sprint 2: Task Board

Sprint Start

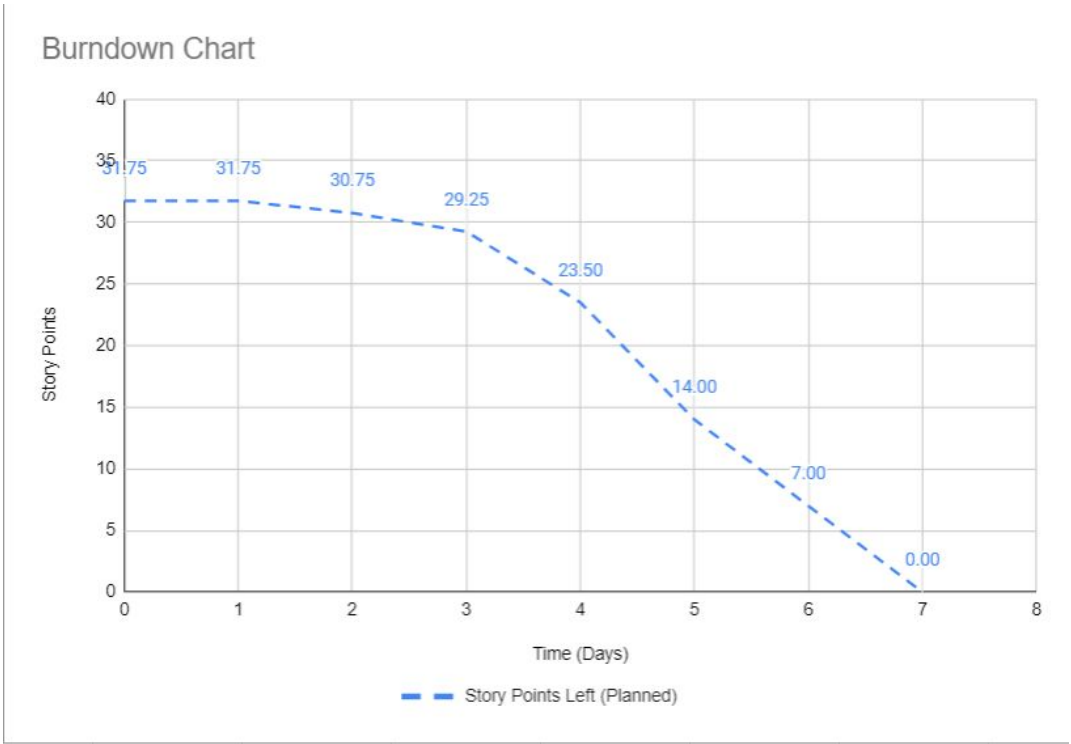


Sprint End

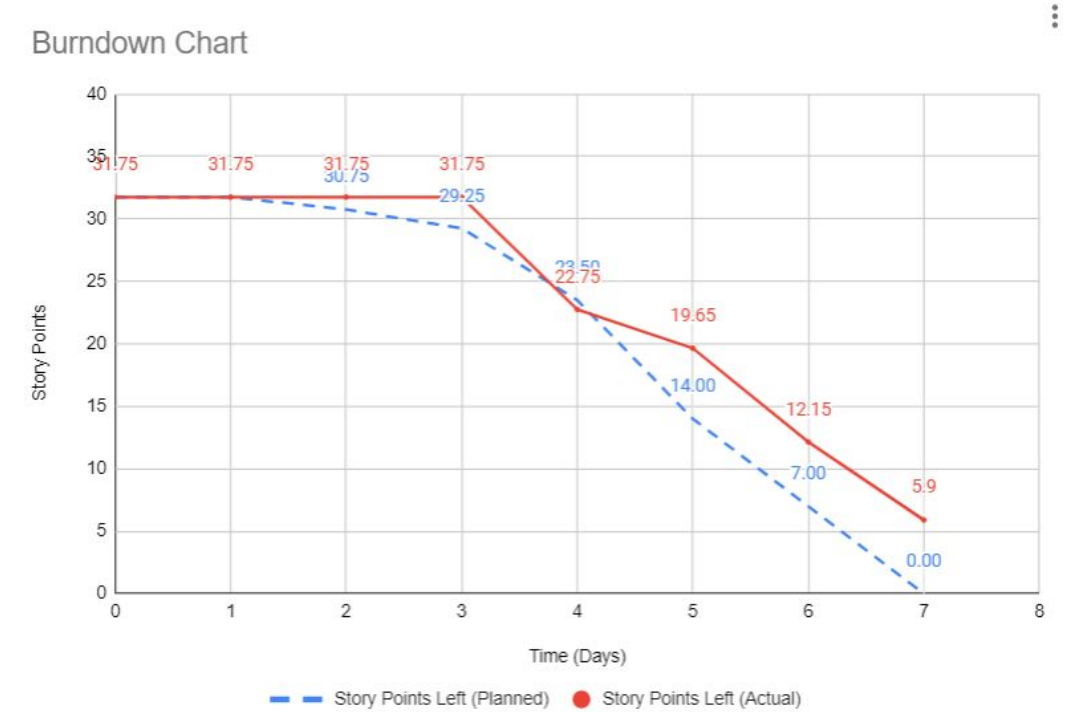


Sprint 2: Burndown Chart

Sprint Start



Sprint End



Sprint 3: Sprint Plan

Priority	Story	Points	Tasks	Points
Critical	-	20	T030: Learn Flask and refamiliarize with Python	2
			T051: Learning Python testing	18
Critical	As David (restaurant owner), I want to be able to register my account.	3.5	T052: Automate tests	3.5
Critical	As Kevin (customer), I want to be able to register my account	2	T053: Automate tests	2
High	As David (restaurant owner), I want to create a coupon with my selected requirements	2.5	T054: Automate tests	2.5
High	As David (restaurant owner), I want to be able to see all my restaurant's current coupons	2	T055: Automate tests	2
High	As David (restaurant owner), I want to be able to delete an existing coupon.	2	T056: Automate tests	2
Critical	As David (restaurant owner) I want to be able to create an employee account for my employee.	2.5	T057: Automate tests	2.5
High	As David (restaurant owner), I want to be able to see all employee accounts associated with my restaurant	3	T058: Automate tests	3
High	As David (restaurant owner), I want to be able to delete accounts associated with my restaurant.	1	T059: Automate tests	1
High	As David (restaurant owner), I want to be able to delete an existing coupon	1	T037: Design coupon deletion UI in relation to existing coupon list UI	0.25
			T038: Implement coupon deletion UI with backend	0.75

Sprint 3: Acceptance Criteria

Story	Acceptance Criteria
As David (restaurant owner), I want to be able to delete an existing coupon.	Scenario: The restaurant owner deletes an existing coupon. Given that the restaurant owner has created at least one coupon, when the owner clicks delete on a coupon entry in the list, the coupon is no longer in the restaurant's coupon list.

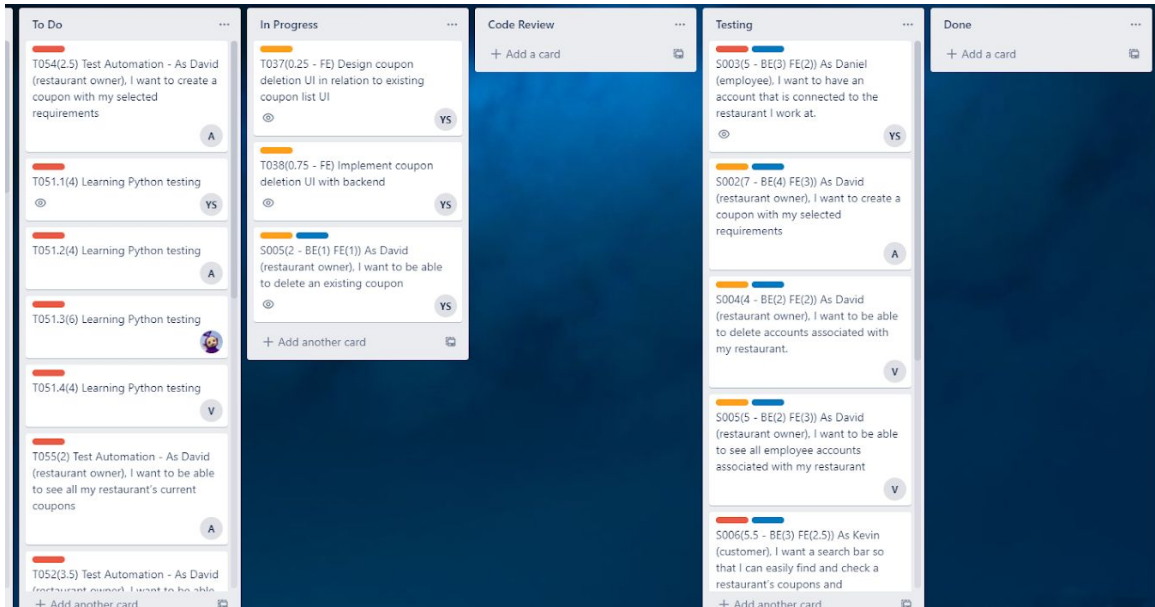
Since we already implemented and acceptance tested all other stories in previous sprints, and there is only a test automation task for those stories in the current sprint, we did not write new acceptance criteria for those stories.

Sprint 3: Team Task Assignments

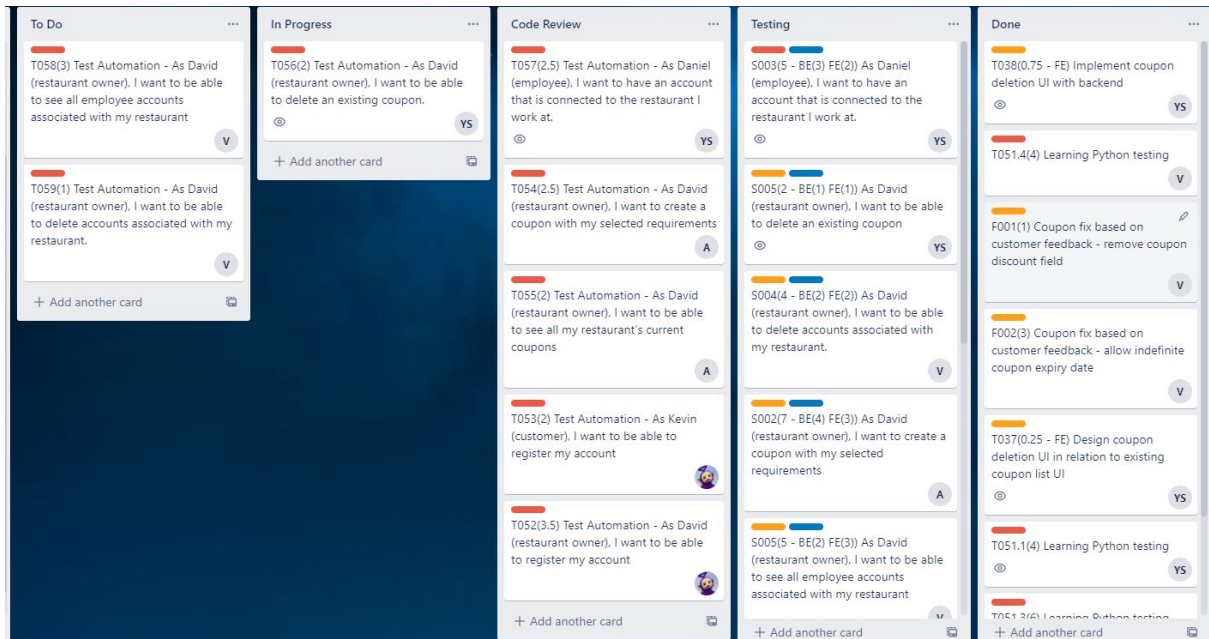
Task ID	Cost	Priority	Day 0	Day1	Day 2	Day 3	Day 4	Day 5	Day 6	Assignment
T037	0.25	High	0	0.25	0	0	0	0	0	Yolanda
T038	0.25	High	0	0.75	0	0	0	0	0	Yolanda
T030	2	High	0	2	0	0	0	0	0	Anson
T051.1	4	Critical	0	0	2	2	0	0	0	Yolanda
T051.2	4	Critical	0	0	2	2	0	0	0	Anson
T051.3	6	Critical	0	0	2	1	1	2	0	Jiawei
T051.4	4	Critical	0	0	0	2	2	0	0	Vanessa
T052	3.5	Critical	0	0	0	2	1.5	0	0	Jiawei
T053	2	Critical	0	0	0	0	1	1	0	Jiawei
T054	2.5	Critical	0	0	0	1	1.5	0	0	Anson
T055	2	Critical	0	0	0	0	1	1	0	Anson
T056	2	Critical	0	0	0	0	1	1	0	Yolanda
T057	2.5	Critical	0	0	0	0	0	0.5	2	Yolanda
T058	3	Critical	0	0	0	0	0	2	1	Vanessa
T059	1	Critical	0	0	0	0	0	0	1	Vanessa
Total Points Completed:			0	3	0	8	10	12	6.5	
Total Points Left:			39.5	36.5	36.5	28.5	18.5	6.5	0	

Sprint 3: Task Board

Sprint Start



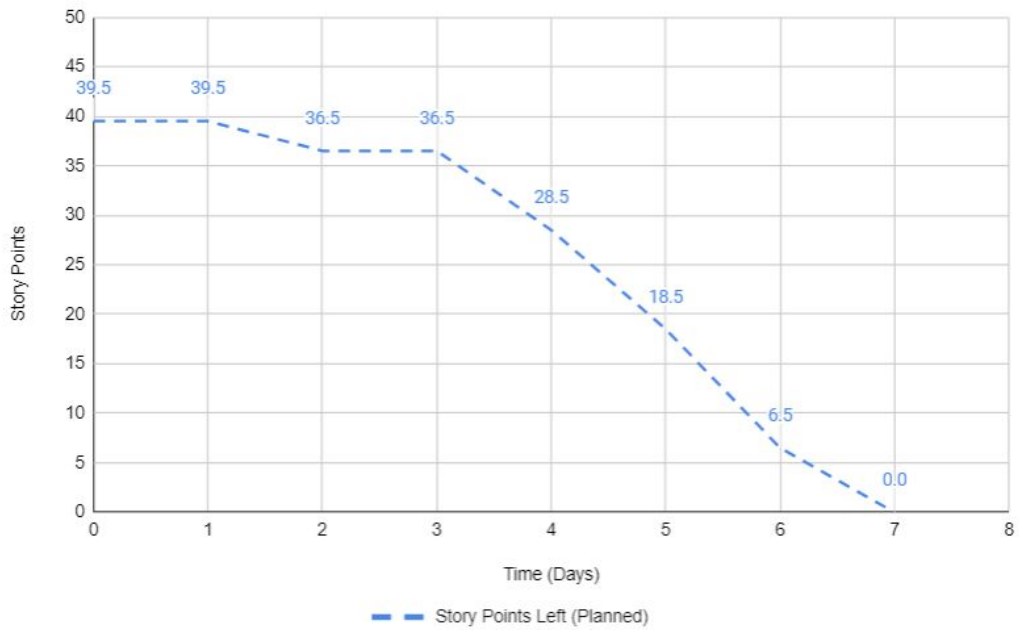
Sprint End



Sprint 3: Burndown Chart

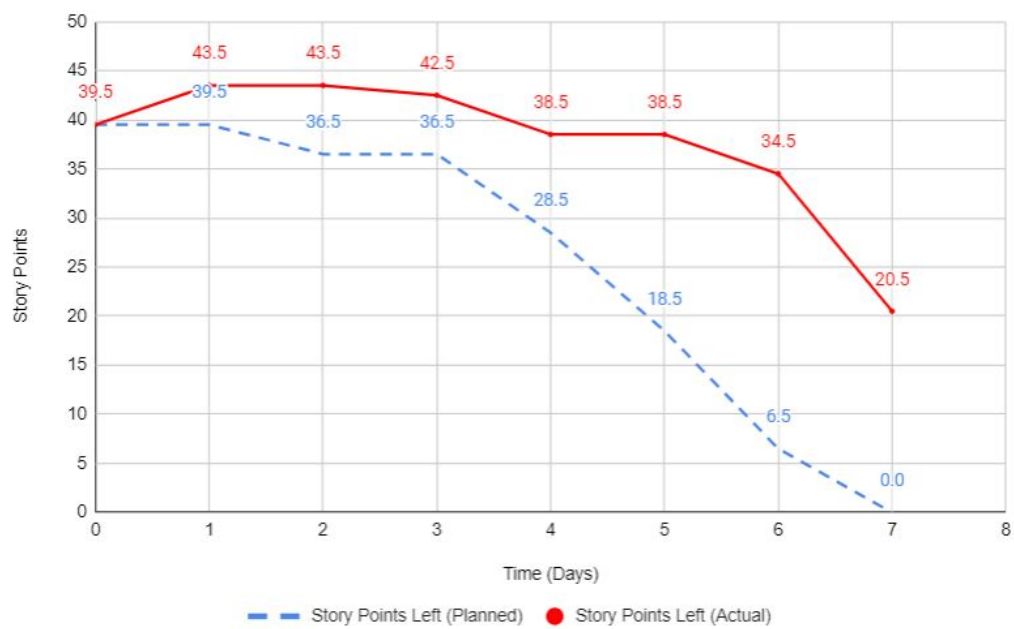
Sprint Start

Burndown Chart



Sprint End

Burndown Chart



Sprint 4: Sprint Plan

Priority	Story	Points	Tasks	Points
Critical	-	-	T061: Refactor app.py	4
			T062: Refactor and complete unit test suite	6
High	As David (restaurant owner), I want to be able to add new achievements to my restaurant and set their experience and/or point values	7.5	T063: Set up achievements table in the database	1
			T064: Code backend for creating and storing new achievements	2
			T065: Design achievement creation UI	0.5
			T066: Implement achievement creation UI	1
			T067: Automate Tests	3
High	As David (restaurant owner), I want a graphical interface allowing me to see how many coupons a customer currently has that they can redeem	6	T068: Set up a table for redeemed coupons linked to the coupon issuing restaurant and redeeming customer	1
			T069: Code owner functions for fetching redeemed coupons related to a given restaurant	2
			T070: Design redeemed coupon list UI for owners	2
			T071: Implement redeemed coupon list UI	1
			T072: Automate tests	3

High	As Kevin (customer), I want to be able to see my current customer level and the experience needed to reach the next level at a given restaurant.	7.5	T073: Set up a table for customer total experience gained at a given restaurant	0.5
			T074: Design an algorithm for converting experience quantity into user level	0.5
			T075: Code functions for fetching experience data and converting to level	2
			T076: Design customer UI for seeing their experience and level at a restaurant	0.5
			T077: Implement experience and level UI	2
			T078: Automate tests	2
High	As Kevin (customer) I want to be able to see available coupons for exchange at a specific restaurant.	4.5	T079: Code function to fetch active and non-expired coupons from a given restaurant	2
			T080: Design available coupon list UI for a customer checking a given restaurant	1.5
			T081: Implement coupon list UI	1
			T082: Automate tests	2

Sprint 4: Acceptance Criteria

Story	Acceptance Criteria
<p>As David (restaurant owner), I want to be able to add new achievements to my restaurant and set their experience and/or point values.</p>	<p>Scenario: The restaurant owner tries to add an achievement with negative experience and/or negative points.</p> <p>Given that the restaurant owner has entered a negative experience value and/or negative points value for their new achievement, when the owner clicks the create button, an error message is shown and no new achievement is shown on the achievement list.</p> <p>Scenario: The restaurant owner tries to add an achievement with non-negative experience and non-negative points.</p> <p>Given that the restaurant owner has entered a non-negative experience value and non-negative points value for their new achievement, when the owner clicks the create button, a new achievement appears on the achievement list with all values matching the given input values.</p> <p>Scenario: The restaurant owner wants to add an achievement of the form "Buy item x times".</p> <p>Given that the restaurant owner wants to add an achievement of the form "Buy item x times", when the owner visits the create achievement page, there is an option to choose an achievement template that takes in an integer x and an item name in the form "Buy item x times".</p> <p>Scenario: The restaurant owner tries to add an achievement of the form "Buy item x times" and sets x as a negative value.</p> <p>Given that the restaurant owner has filled in an achievement template of the form "Buy item x times" with a negative x value, when the owner clicks the create button, an error message is shown and no new achievement is shown on the achievement list.</p> <p>Scenario: The restaurant owner wants to add an achievement of the form "Spend x dollars in a single visit".</p> <p>Given that the restaurant owner wants to add an achievement of the form "Spend x dollars in a single visit", when the owner visits the create achievement page, there is an option to choose an achievement template that takes in an integer x in the form "Spend x dollars in a single visit".</p>
<p>As David (restaurant owner), I want a graphical interface allowing me to see how many coupons a</p>	<p>Scenario: There are no redeemed coupons at the owner's restaurant.</p> <p>Given that there are no redeemed coupons at the owner's restaurant, when the owner opens the redeemed coupon list</p>

<p>customer currently has that they can redeem</p>	<p>page, the owner does not see any coupon entries.</p> <p>Scenario: There is at least one redeemed coupon at the owner's restaurant.</p> <p>Given that at least one coupon has been redeemed at the owner's restaurant, when the owner opens the redeemed coupon list page, all coupons redeemed by customers at the restaurant are shown with coupons marked as either used or unused.</p> <p>Scenario: The same user redeems multiple coupons at the owner's restaurant.</p> <p>Given that the same user redeems multiple coupons at the owner's restaurant, when the owner opens the redeemed coupon list page, all coupons redeemed by that customer at the restaurant are shown with coupons marked as either used or unused.</p> <p>Scenario: Multiple users redeem the same coupon at the owner's restaurant.</p> <p>Given that multiple users redeem the same coupon at the owner's restaurant, when the owner opens the redeemed coupon list page, the owner sees each customer redemption of that coupon as a separate redeemed coupon entry, with coupons marked as either used or unused.</p>
<p>As Kevin (customer), I want to be able to see my current customer level and the experience needed to reach the next level at a given restaurant.</p>	<p>Scenario: The customer is visiting the restaurant's page for the first time.</p> <p>Given that the customer is visiting a restaurant's page for the first time, when the customer opens the page, their level and experience both appear as zero.</p> <p>Scenario: The customer has earned some experience at the given restaurant before.</p> <p>Given that the customer has earned some experience at a given restaurant before, when the customer opens the page, the customer sees their current customer level and the experience needed to reach the next level up at the restaurant.</p>
<p>As Kevin (customer) I want to be able to see available coupons for exchange at a specific restaurant.</p>	<p>Scenario: A given restaurant does not offer coupons.</p> <p>Given that a restaurant does not have coupons available for exchange, when the customer opens the restaurant's coupon list page, there are no coupon entries shown.</p> <p>Scenario: A given restaurant offers at least one active coupon.</p> <p>Given that the restaurant has at least one active coupon available for exchange, when the customer opens the restaurant's coupon list page, the customer sees a list of all</p>

	<p>available coupons at the restaurant.</p> <p>Scenario: The restaurant has expired coupon offers.</p> <p>Given that the restaurant has at least one expired coupon offer, when the customer opens the restaurant's coupon list page, all expired coupons at the restaurant are not visible in the coupon list.</p>
--	--

Sprint 4: Team Task Assignments

Task ID	Cost	Priority	Day 0	Day1	Day 2	Day 3	Day 4	Day 5	Day 6	Assignment
T061	4	Critical	4	0	0	0	0	0	0	Vanessa
T062	6	Critical	0	2	2	2	0	0	0	Jiawei
T063	1	High	0	1	0	0	0	0	0	Anson
T064	2	High	0	1	1	0	0	0	0	Anson
T065	0.5	High	0	0	0.5	0	0	0	0	Anson
T066	1	High	0	0	1	0	0	0	0	Anson
T067	3	High	0	0	0	0	2	1	0	Anson
T068	1	High	0	0	1	0	0	0	0	Vanessa
T069	2	High	0	0	0	2	0	0	0	Vanessa
T070	2	High	0	0	0	0	2	0	0	Vanessa
T071	1	High	0	0	0	0	1	0	0	Vanessa
T072	3	High	0	0	0	0	0	3	0	Vanessa
T073	0.5	High	0	0	0.5	0	0	0	0	Yolanda
T074	0.5	High	0	0	0.5	0	0	0	0	Yolanda
T075	2	High	0	0	0	2	0	0	0	Yolanda
T076	0.5	High	0	0	0.5	0	0	0	0	Yolanda
T077	2	High	0	0	0	0	2	0	0	Yolanda
T078	2	High	0	0	0	0	0	2	0	Yolanda
T079	2	High	0	0	0	0	2	0	0	Vanessa
T080	1.5	High	0	0	0	0	0	1.5	0	Vanessa

T081	1	High	0	0	0	0	0	1	0	Vanessa
T082	2	High	0	0	0	0	0	0	2	Vanessa
Total Points Completed:			4	1	6	10	7	10.5	2	
Total Points Left:			36.5	35.5	29.5	19.5	12.5	2	0	

Sprint 4: Task Board

Sprint Start

The screenshot shows a Jira Task Board for Sprint 4. The board is organized into five columns: To Do, In Progress, Code Review, Testing, and Done. Each column has a header with a plus icon and a minus icon. The 'To Do' column contains six tasks, each with a progress bar and an assignee icon. The 'Testing' column contains five stories, each with a progress bar and an assignee icon. The 'In Progress', 'Code Review', and 'Done' columns are currently empty.

To Do

- T061(4) Refactor app.py (Assignee: V)
- T062(6) Refactor and complete unit test suite (Assignee: A)
- S008(7.5 - BE(2) FE(2.5) A(3)) As David (restaurant owner), I want to be able to add new achievements to my restaurant and set their exp and/or point values (Assignee: A)
- T063(1) Set up achievements table in database (Assignee: A)
- T064(2) Code backend for creating and storing new achievements (Assignee: A)
- T065(0.5) Design achievement creation UI (Assignee: A)

In Progress

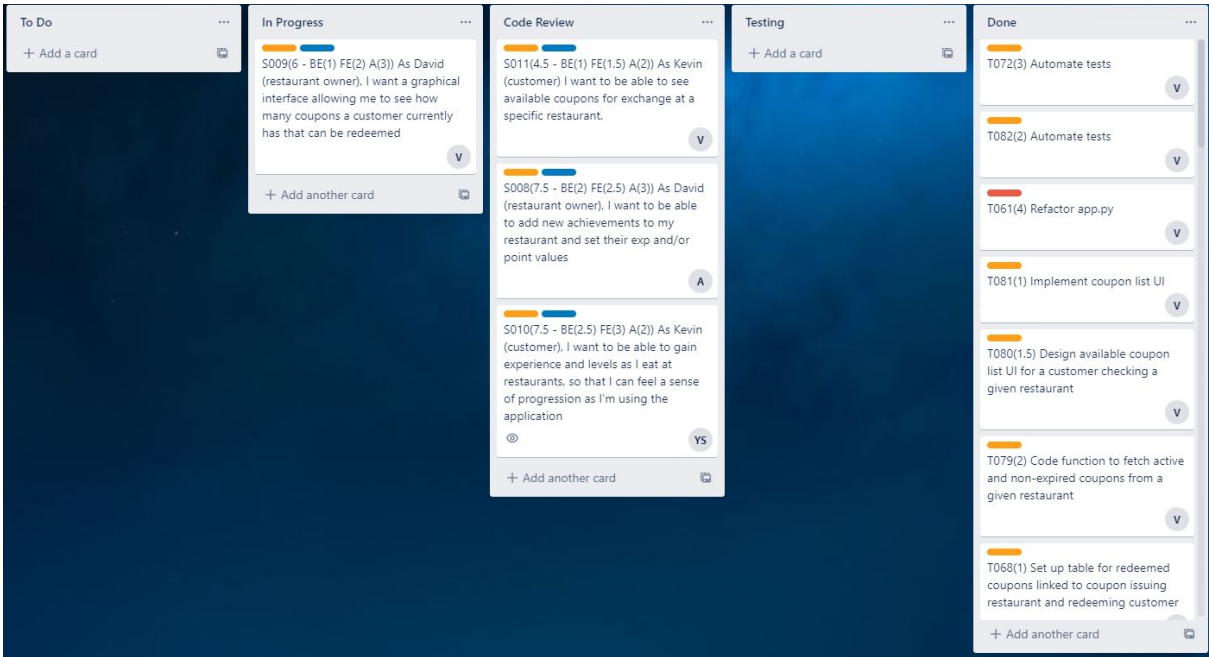
Code Review

Testing

- S003(5 - BE(3) FE(2)) As Daniel (employee), I want to have an account that is connected to the restaurant I work at. (Assignee: V)
- S005(2 - BE(1) FE(1)) As David (restaurant owner), I want to be able to delete an existing coupon (Assignee: V)
- S004(4 - BE(2) FE(2)) As David (restaurant owner), I want to be able to delete accounts associated with my restaurant. (Assignee: V)
- S002(7 - BE(4) FE(3)) As David (restaurant owner), I want to create a coupon with my selected requirements (Assignee: A)
- S005(5 - BE(2) FE(3)) As David (restaurant owner), I want to be able to see all employee accounts associated with my restaurant (Assignee: V)

Done

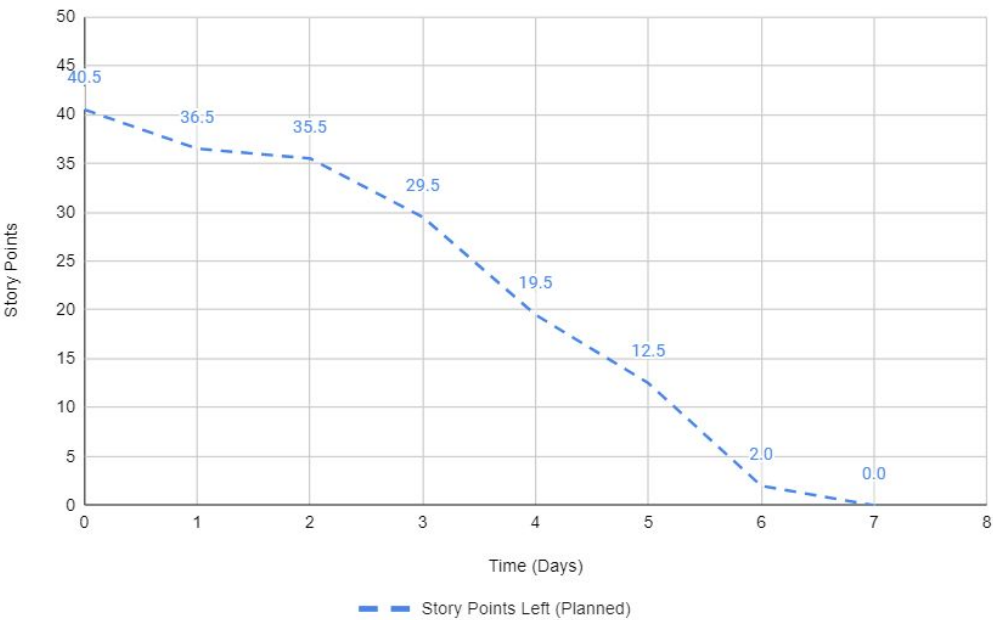
Sprint End



Sprint 4: Burndown Chart

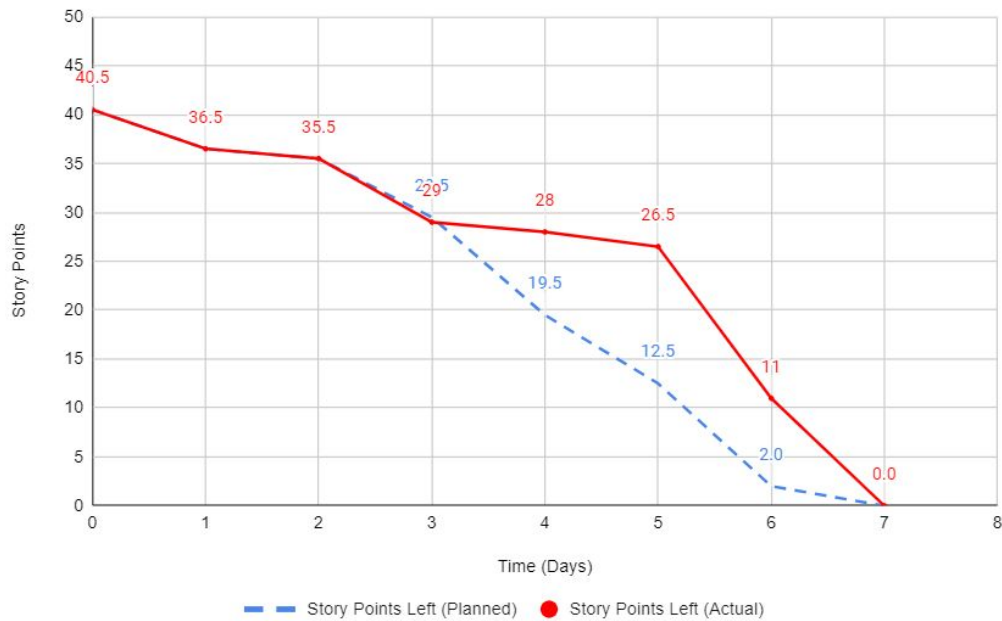
Sprint Start

Burndown Chart

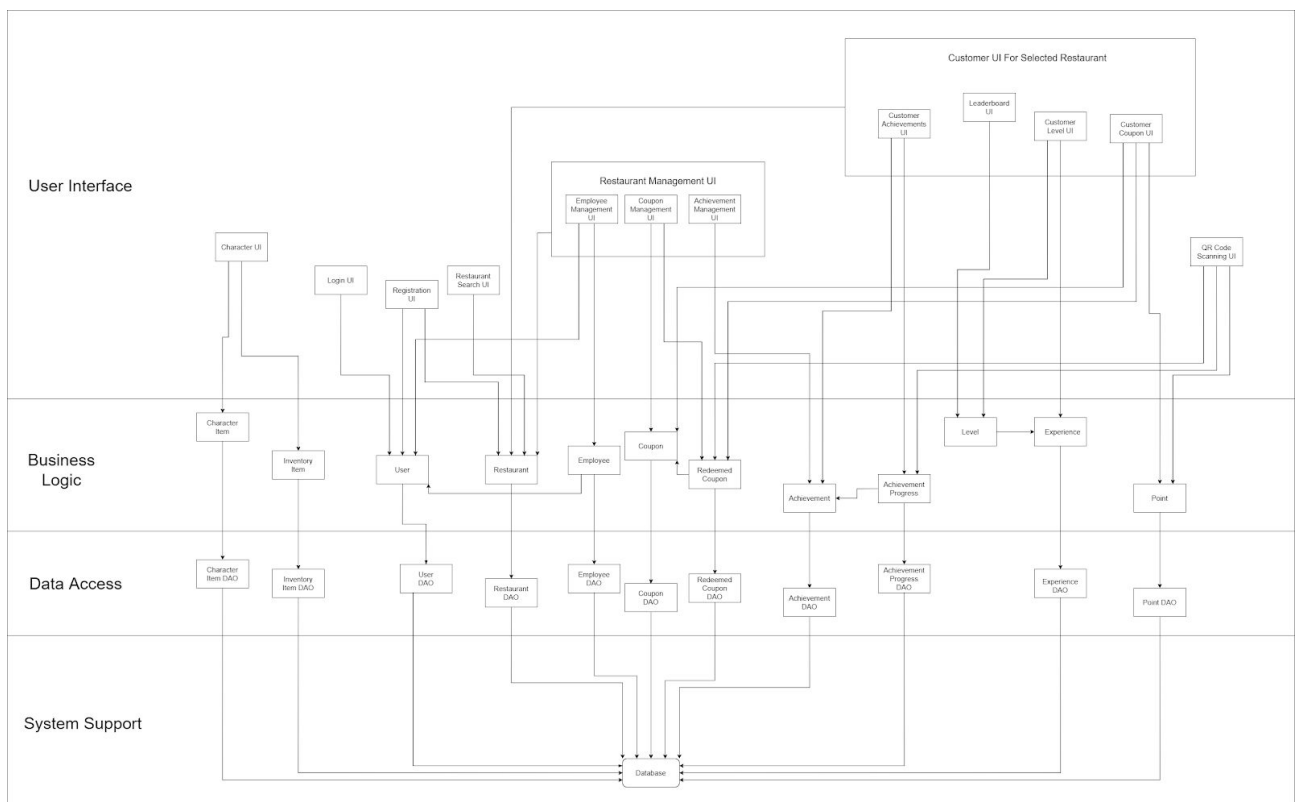


Sprint End

Burndown Chart



High-level Architecture



See the full system architecture diagram in the deliverable 4 appendix.

Component	Description	Dependencies
Login UI	The login UI is responsible for allowing all users to log in to their user account.	The login UI relies on the user component to retrieve user account data.
Registration UI	The registration UI is responsible for allowing restaurant owners and customers to log in to their user account.	<p>The registration UI relies on the user component to insert new user account data.</p> <p>The registration UI relies on the restaurant component to insert new restaurant data.</p>
Restaurant Search UI	The restaurant search UI is responsible for allowing customers to search for restaurants on the app.	The restaurant search UI relies on the restaurant component to retrieve restaurants with names that match the search query.
Restaurant Management UI	The restaurant management UI is responsible for allowing restaurant owners to modify their restaurant on the app.	The restaurant management UI relies on the restaurant component to retrieve and modify restaurant data.
Employee Management UI	The employee management UI is responsible for allowing restaurant owners to create and modify employee accounts at their restaurant.	<p>The employee management UI relies on the restaurant component to retrieve and modify employee data.</p> <p>The employee management UI relies on the user component to insert new employee account data.</p>
Coupon Management UI	The coupon management UI is responsible for allowing restaurant owners to view, create and modify coupons from their restaurant.	<p>The coupon management UI relies on the coupon component to retrieve and modify coupon data.</p> <p>The coupon management UI relies on the redeemed coupon component to retrieve redeemed coupon data.</p>
Achievement Management UI	The achievement management UI is responsible for allowing restaurant owners to view, create and modify achievements from their restaurant.	The achievement management UI relies on the achievement component to retrieve and modify achievement data.
Customer UI For Selected Restaurant	The customer restaurant UI is responsible for allowing customers to see an overview of	The customer restaurant UI relies on the restaurant component to retrieve restaurant data.

	specific restaurants' coupons, achievements, leaderboards, etc.	
Customer Achievements UI	The customer achievements UI is responsible for allowing customers to view available and already obtained achievements offered at a given restaurant.	<p>The customer achievements UI relies on the achievement component to retrieve data on available achievements.</p> <p>The customer achievements UI relies on the achievement progress component to retrieve customers' achievement progress data.</p>
Leaderboard UI	The leaderboard UI is responsible for allowing users to see the highest level customers at a given restaurant.	The leaderboard UI relies on the level component to retrieve level data for all customers at a given restaurant.
Customer Level UI	The customer level UI is responsible for allowing customers to see their level and level-up progress at a given restaurant.	<p>The leaderboard UI relies on the level component to calculate level data for customers.</p> <p>The leaderboard UI relies on the experience component to retrieve customer experience data.</p>
Customer Coupon UI	The customer coupon UI is responsible for allowing customers to view available and already obtained coupons and buy coupons offered at a given restaurant.	<p>The customer coupon UI relies on the coupon component to retrieve available coupon data that the customer may redeem.</p> <p>The customer coupon UI relies on the redeemed coupon component to retrieve coupon data for coupons the customer has already redeemed.</p> <p>The customer coupon UI relies on the points component to update points data after a user has spent points redeeming coupons.</p>

QR Code Scanning UI	The QR code scanning UI is responsible for allowing employees to use coupons and update achievement progress for customers.	<p>The QR code scanning management UI relies on the achievement progress component to update customer achievement progress data.</p> <p>The QR code scanning management UI relies on the points component to update customer points data after marking an achievement complete.</p> <p>The QR code scanning management UI relies on the redeemed coupon component to update redeemed coupon data and mark redeemed coupons as used.</p>
Character UI	The character UI is responsible for allowing customers to view and modify their custom character avatar.	<p>The character UI relies on the character item component to retrieve data for available character items.</p> <p>The character UI relies on the inventory item component to retrieve data for equipable inventory items, and update data to mark items as equipped or unequipped.</p>
Character Item	The character item component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for obtainable and equipable character items.	The character item component relies on the character item DAO to access character item data.
Inventory Item	The inventory item component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for character items a user has added to their inventory.	The inventory item relies on the inventory item DAO to access inventory item data.
User	The user component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) user account data.	The user component relies on the user DAO to access user account data.
Restaurant	The restaurant component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) restaurant data.	The restaurant component relies on the restaurant DAO to access restaurant data.

Employee	The employee component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) restaurant permissions data for employees.	<p>The employee component relies on the employee DAO to access employee data.</p> <p>The employee component relies on the user component to modify employee user account data.</p>
Coupon	The coupon component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) coupon data for coupons offered at restaurants.	The coupon component relies on the coupon DAO to access coupon data.
Redeemed Coupon	The redeemed coupon component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for coupons purchased and/or redeemed by a customer.	<p>The redeemed coupon component relies on the redeemed coupon DAO to access redeemed coupon data.</p> <p>The redeemed coupon component relies on the coupon component to retrieve data for the original coupon offer.</p>
Achievement	The achievement component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for achievements available at restaurants.	The achievement component relies on the achievement DAO to access achievement data.
Achievement Progress	The achievement progress component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for customers' progress on completing achievements.	<p>The achievement progress component relies on the achievement progress DAO to access achievement progress data.</p> <p>The achievement progress component relies on the achievement component to retrieve data for the original achievement that the customer is making progress on.</p>
Point	The point component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for the points earned by a customer at a given restaurant.	The point component relies on the point DAO to access points data.
Experience	The experience component is responsible for manipulating (e.g. inserting, deleting, modifying, etc.) data for the experience earned by a customer at a given restaurant.	The experience component relies on the experience DAO to access experience data.

Level	The level component is responsible for calculating a customer's current level and progress to the next nearest level based on their accumulated experience at a given restaurant.	The level component relies on the experience component to retrieve customer experience data for a given restaurant.
Character Item DAO	The character item DAO is responsible for providing access to character item data in the database.	The character item DAO relies on the database to store character item data.
Inventory Item DAO	The inventory item DAO is responsible for providing access to inventory item data in the database.	The inventory item DAO relies on the database to store inventory item data.
User DAO	The user DAO is responsible for providing access to user data in the database.	The user DAO relies on the database to store user data.
Restaurant DAO	The restaurant DAO is responsible for providing access to restaurant data in the database.	The restaurant DAO relies on the database to store restaurant data.
Employee DAO	The employee DAO is responsible for providing access to employee data in the database.	The employee DAO relies on the database to store employee data.
Coupon DAO	The coupon DAO is responsible for providing access to coupon data in the database.	The coupon DAO relies on the database to store coupon data.
Redeemed Coupon DAO	The redeemed coupon DAO is responsible for providing access to redeemed coupon data in the database.	The redeemed coupon DAO relies on the database to store redeemed coupon data.
Achievement DAO	The achievement DAO is responsible for providing access to achievement data in the database.	The achievement DAO relies on the database to store achievement data.
Achievement Progress DAO	The achievement progress DAO is responsible for providing access to achievement progress data in the database.	The achievement progress DAO relies on the database to store achievement progress data.
Point DAO	The point DAO is responsible for providing access to points data in the database.	The point DAO relies on the database to store points data.

Experience DAO	The experience DAO is responsible for providing access to experience data in the database.	The experience DAO relies on the database to store experience data.
Database	The database is responsible for storing all object data for the application.	The database does not have dependencies.

For the redesign of our system architecture, the main change we made was removing the UI management layer and adding a data access layer so that our design better reflects our actual application structure. We added DAO (Data Access Object) components corresponding to the components in the business logic layer accordingly. We also removed the POS related UI component that was present in our deliverable 3 design due to the removal of the POS stories in our product backlog and added the QR scanning UI representative of our new story for updating achievement progress that replaced the POS stories. Finally, we removed a few unnecessary components (e.g. the separate customer and owner components) and added missing components (e.g. achievement progress and redeemed coupon) in the business layer based on team discussion and our deliverable 3 feedback.

Validation

In this deliverable, our main method of system validation was through emailing the client with our questions, as well as demoing our product through either video meetings or via emailed demo videos. See the deliverable 4 appendix for documented evidence of these activities.

After receiving our client's feedback through these validation methods, we ended up changing the priority of certain stories, such as raising the priority (from medium to high) of the story for a feature displaying coupons redeemed by customers at a given restaurant to the restaurant owner. We also removed the stories related to the POS integration feature as our client confirmed it was not a high priority, and chose to use a different way of implementing a convenient way of updating achievement progress (via scanning with QR code) instead.

We also added fixes to some features based on client feedback. For example, our coupons originally required a start and expiry date, but after hearing from the client we added the option to set coupon duration as indefinite. We also originally had a separate coupon name and coupon discount fields, but based on client feedback about the discount field being redundant we removed the discount field from our coupon model. Conversely, we did not incorporate the feedback initially given in our first demo to integrate customer login with PickEasy's account system as upon further communication

with the client via email, the client changed their mind and decided it was not a requirement.

Project Velocity

Estimated Velocity

Our estimated velocity was 31.75 for sprint 2, 39.5 for sprint 3, and 40.5 for sprint 4.

We estimated our project velocity for every sprint by summing together the number of points each member of the team was able to commit to the sprint.

Actual Velocity

Our actual velocity was 25.85 for sprint 2, 23 for sprint 3, and 40.5 for sprint 4.

Changes from Deliverable 3

While we had our first sprint in the previous deliverable and began developing using the Agile process, we missed several important steps and aspects of the Agile process, which we attempted to rectify for deliverable 4. These changes to our development process mainly involved changes to our meeting schedule, as well as changes to our validation and verification process.

Firstly, we moved our retrospective meeting from 1 PM on Sundays to 6 PM so that the retro could take place after the end of our sprint. We also added sprint review and sprint planning meetings that we omitted in deliverable 1, both taking place consecutively following our retrospective meeting, with the combined meetings spanning from 6 PM to 8 PM.

Next, to improve our verification process, we needed to create a complete unit test suite. To facilitate this, we repointed our user stories, adding automation points meant for allocating time to automating unit tests. We assigned these point values during a team Zoom meeting, using our usual story pointing method of planning poker. Also, as part of improving validation, we needed a more formalized code review process from deliverable 3. We accomplished this by beginning to use pull requests to request and complete code reviews for our feature branches.

To improve our validation process, instead of limiting our client communications to asking questions about product requirements, we added demoing our implemented features to our client for approval as part of our development process. This is to verify that our product always adheres to the client's expectations and wants, especially if those expectations change.

To ensure that all features on the master branch were both verified and validated, we created a Development branch in our Github. We only merge code reviewed feature branches with completed unit test suites (i.e. fully verified branches) into Development. Then, we must demo the Development branch build to the client and receive their approval (i.e. must be fully validated) before we merge Development into master.

Finally, based on deliverable 3 feedback and team discussion, we changed how we estimated velocity for sprints. In deliverable 3 we estimated by dividing the total product backlog story points by the total number of sprints, but we noticed this number was lower than the number of points the team was capable of committing to a sprint. Therefore, as mentioned before, we switched to estimating our velocity for every sprint by summing together the number of points each member of the team was able to commit to the sprint.

Retrospective

During sprint 2, we found that our actual velocity was lower than our estimated velocity as we were unable to complete some of the tasks in the sprint plan. After discussing as a team, we felt our productivity was negatively impacted by our development process. We found that developing features with multiple developers working on the same story, one being the front-end and the other being the back-end developer, was resulting in a lot of duplicated work by multiple programmers on a story. There would also often be merging and other communication issues when the front-end and back-end developers worked on a story at the same time. As a result, we decided to stop working in frontend-backend pairs, instead, we now have one person work on both the front-end and back-end of an individual story.

Our sprint 3 plan, consisting mainly of tasks meant to focus on fixing our lacking unit test suite coverage, was significantly disrupted by the realization that our code was not testable as we did not adhere to our system design. There was a lack of modularization in our code base, resulting in a lot of confusion among team members concerning how to design and write unit tests, what needed testing and what didn't need testing, and how to run the tests. This confusion resulted in us writing a lot of tests that either didn't work, were redundant or tested parts of our system that didn't need testing, such as testing Flask SQL queries. This ultimately resulted in sprint 3 being unproductive, and we underperformed compared to our estimated velocity by a lot. Sprint 3 also had an increase in "story points left" beyond our estimated velocity, as we had to add fix tasks into the sprint a few days in, for fixing previously completed features after receiving client feedback on those features during our demo.

After sprint 3, we realized we needed to spend time in sprint 4 refactoring all our existing code to be testable, and completely redo our unit test suite accordingly. This left us behind on our product backlog. Otherwise, sprint 4 went well, we did not end up deviating from our sprint 4 plan in any significant way.

Overall, we did not end up needing to use our contingency plan during this deliverable.

Release

Our most recent release can be found at <https://pickeasy-beta.herokuapp.com>.