

DELIVERABLE 3

PART 1



Team_01(GitHub)

Anson Feng

Jiawei Qiao

Vanessa Pierre

Yolanda Su

Table Of Contents

Project Setup	2
Project Backlog	5
Release Plan	8
Sprint Plan	8
Acceptance Criteria	10
Team Task Assignments	14
Task Board	16
Burndown Chart	17

Project Setup

- **What tools will we use for our task board?**

Trello

- **What tools will we use for our burn-down chart?**

Google Sheets

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

The scrum master (Yolanda) will maintain the chart. The scrum master will update the burndown chart data table at the end of every working day with the most current number of story points left.

- **What is every team member's role?**

Yolanda: Scrum master, Front-end, Holding the meeting, Managing the task board/burndown chart, Recording meeting minutes

Vanessa: Front-end, Editing the final deliverable documents, Recording meeting minutes

Anson: Back-end, Communicating with client/TA, Recording meeting minutes

Jiawei: Back-end, Creating meeting room, Updating repo README file, Recording meeting minutes

Adjustments for the front-end and back-end developer roles will be made every sprint based on the stories worked on in that sprint and the decision of the team's pairs.

The team will be split into pairs of 2, each pair will have 1 person mainly responsible for the front-end and 1 person mainly responsible for the back-end, and the pair will decide among themselves who fills which role. Pairs will work on the same stories (e.g. one person in the pair works on the front end of a story, the other works on the back end of the same story). Pairs are also responsible for reviewing each other's code.

If there are fewer back-end points on a team than front-end points, the back-end developer will help with frontend points after finishing backend work, and vice versa if there are less frontend than backend points. If a card is still incomplete by the end of the sprint, the original assignees to those cards will first complete the leftover work before moving on working with their new partner for the next sprint.

Team member pairing cycle (cycle repeating every 3 sprints):

Yolanda + Anson, Vanessa + Jiawei

Yolanda + Jiawei, Anson + Vanessa

Yolanda + Vanessa, Anson + Jiawei

- **What tools, if any, will we use for communication?**

We will use Slack for general online discussion and Zoom for team meetings.

- **When do we plan to meet in person?**

Daily stand up meetings will take place on zoom, these meetings will be held from 5:30 - 5:45 pm. During these meetings, members will discuss their accomplished daily tasks or issues that are holding them back from competing for their daily tasks.

Weekly team meetings will take place on Wednesdays from 6:00 - 7:00 pm. This meeting will act as our midsprint team check-in meeting. During this weekly meeting members will go more in-depth with how the progression of the application is proceeding so that we may revise our sprint plan and estimates as needed. Additional meeting times will be added when necessary based on team discussion.

TA meetings will take place on Thursday from 6:00 - 7:00 pm.

- **How will we use our repository on GitHub?**

Features will be developed on individual branches, and branches will be pushed to git after functionality is complete. Bug fix commits on a feature will be pushed to that feature's branch if the branch has not yet been merged into master, and will be pushed into a new bug fix branch if the original feature branch is already merged.

Branches will be merged to master after they have been fully tested and approved.

We will have rules for committing:

1. Only commit the source code of the file in order to have regression tests, and never commit unnecessary files
2. A single commit should only include one completed feature (function) so that it is easy to test and revert if any problems occur
3. A commit message should have the first character in uppercase, using past tense, and be less than or equal to 50 characters. It should describe what changes are being committed briefly, and should not describe specific lines of code.

For naming the branches, we will use task ID to name the branch as it is easier to read than task name.

- **Which machines will be used for development by each team member?**

Yolanda: Windows home computer

Vanessa: Macbook

Anson: Windows home computer

Jiawei: Macbook/Windows home computer

• **What is our DoD (definition of done)?**

In order for a story to be considered “done”, all subtasks of that story must be complete. For a task to be considered “done”, completed code must adhere to the Google style guide. All code must also be peer reviewed by at least 1 other team member. The code must be able to be built successfully. The code must also pass all automated tests relevant to the task’s acceptance criteria, as well as pass all previously completed automated tests from related features.

Project Backlog

1 point = 1 working hour

Type	Priority	Story	Backend Points	Frontend Points	Total Story Points
Story	Critical	As David (restaurant owner), I want to be able to register my account as a restaurant owner	6.5	4	11.5
Story	Critical	As Kevin (customer), I want to be able to register my account as a customer	0	0	0
Story	Critical	As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements	3	2.5	5.5
Story	High	As David (restaurant owner), I want to create a coupon with my selected requirements	4	3	7
Story	High	As David (restaurant owner), I want to be able to see all my restaurant's current coupons	1	6	7
Story	High	As David (restaurant owner), I want to be able to delete an existing coupon	1	1	2
Story	High	As David (restaurant owner), I want to be able to add new achievements to my restaurant and set their exp and/or point values	4	3	7
Story	High	As David (restaurant owner), I want a default template for customer achievements, so that I can set up achievements more easily	1	4	5

Story	High	As David (restaurant owner), I want to be able to see all my restaurant's current achievements	1	4	5
Story	High	As David (restaurant owner), I want to be able to delete achievements from my restaurant	1	1	2
Story	High	As David (restaurant owner), I want an interface for editing restaurant settings where I can manage all aspects of my restaurant on the application easily	0	7	7
Story	High	As David (restaurant owner), I want customers' progress towards achievements and coupons to update only after they buy an item.	7	0	7
Story	High	As Kevin (customer), I want to be able to link my account to Shopify or Square so that after purchases my achievements and points will automatically updated	7	2	9
Story	High	As Kevin (customer) I want to be able to redeem coupons at the related restaurant	4	2	6
Story	High	As David (restaurant owner) I want to be able to scan customer coupon QR codes and mark the coupon used	3	7	10
Story	High	As Kevin (customer) I want to be able to see what achievements are available	2	2	4
Story	High	As Kevin (customer), I want to be able to gain experience and levels as I eat at restaurants, so that I can feel a sense of progression as I'm using the application	3	0	3
Story	High	As Kevin (customer) I want to be able to see available coupons and purchase coupons by redeeming my points	2.5	4	6.5
Story	High	As Kevin (customer), I want an interface for each restaurant that tracks all my achievement progress, levels, and points	2	7	9
Story	Medium	As Kevin (customer), I want a profile page so that I can easily see which restaurants I've visited before	3	2	5

Story	Medium	As Kevin (customer) I want to be able to see how many points I gain after completing an achievement	0.5	1	1.5
Story	Medium	As Kevin (customer), I want to see user leaderboards on the app so that I can compare my level with others and compete to see who has the highest level at a restaurant	2	4	6
Story	Medium	As David (restaurant owner), I want a graphical interface allowing me to see how many coupons a customer currently have that can be redeemed	1	4	5
Story	Medium	As David (restaurant owner), I want the option to give customers exp or points based on how much they spent at my restaurant, so that customers will feel rewarded when they dine at my restaurant	3	1	4
Story	Low	As David (restaurant owner), I want to be able to use existing coupons as a template for new coupons	1	2	3
Story	Low	As David (restaurant owner), I want to be able to modify achievements from my restaurant	1	1	2
Epic	Low	As David (restaurant owner), I want a menu system allowing me to upload, preview and modify my menu onto the application	6	9	15
Epic	Low	As David (restaurant owner) I want to be able to configure the benefits customers can get from their user level at my restaurant	3	5	8
Epic	Low	As Kevin (customer), I want to have a customizable character that can represent my progress, so that I will be more engaged in the app and have more fun	9	10	19 (+Assets: ∞)

Release Plan

Our team has decided on weekly sprints, with sprints ending on Sundays at 6pm. Team pairs will pull ~15 points worth of stories from the top of the product backlog every sprint. We chose to have weekly sprints so that we can receive feedback on our feature development and direction more frequently, in terms of both internal feedback between team members and external feedback from the TA and our client. The shorter sprint length also allows us to check our velocity more frequently via our end of sprint meetings, so we have a better idea of how on track we are at any point in time. The frequent sprint summary meetings also allow us to uncover any possible issues in developing planned features earlier on, so that we can review and adjust our product backlog in our learning process as early as needed.

Sprint Plan

Priority	Story	Backend Points	Frontend Points	Tasks	Backend Points	Frontend Points
Critical	As David (restaurant owner), I want to be able to register my account as a restaurant owner	6.5	4	Design the user database with restaurant owner and customer type accounts	1	0
				Design the restaurant database	1	0
				Code the user database with user account creation functionality	2	0
				Code the restaurant database and link restaurant creation to the creation of a restaurant owner user account	2	0
				Design registration and login UI	0	0.5
				Implement UI using database	0	3
				Automate tests	0.5	0.5

Critical	As Kevin (customer), I want to be able to register my account as a customer	0	0	see tasks above		
Critical	As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements	3	2.5	Design search algorithm for restaurants	1	0
				Code search function with respect to restaurant database	1.5	0
				Design search bar UI	0	0.25
				Implement UI using backend functions	0	1.75
				Automate tests	0.5	0.5
High	As David (restaurant owner), I want to create a coupon with my selected requirements	4	3	Design coupon database for restaurant-specific coupons	2	0
				Code the user database with user coupon creation functionality	1.5	0
				Design coupon creation UI	0	0.5
				Implement UI using restaurant-coupon database	0	2
				Automate tests	0.5	0.5
High	As David (restaurant owner), I want to be able to see all my restaurant's current coupons	1	6	Code database query for fetching coupons specific to a given restaurant owner account	0.8	0
				Design UI for displaying list of coupons	0	1
				Implement UI using	0	4

				backend functions		
				Automate tests	0.2	1

Acceptance Criteria

Task	Acceptance Criteria
Design the user database with restaurant owner and customer type accounts	<p>The database is able to store multiple user account entries</p> <p>Each user account is marked as either a restaurant owner account or a customer account</p> <p>Each user account has a unique account ID</p> <p>Each user account has a unique login name/email and a password</p> <p>Each restaurant owner account is associated with a restaurant</p>
Design the restaurant database	<p>The database is able to store multiple restaurants</p> <p>Each restaurant has a unique restaurant ID</p> <p>Each restaurant has a name</p> <p>Each restaurant is associated with a restaurant owner account</p>
Code the user database with user account creation functionality	<p>The implemented database fulfills all the acceptance criteria for the user database design task</p> <p>New accounts can be created</p> <p>Newly created accounts are successfully added to the database</p> <p>New accounts can only be added to the database if login name, password, and account type information is provided</p> <p>New accounts can only be added to the database if the login name is unique</p> <p>New restaurant owner accounts can only be created if a restaurant name is provided</p> <p>Account ID can be retrieved if and <u>only</u> if both login and password information is provided</p>

Code the restaurant database and link restaurant creation to the creation of a restaurant owner user account	<p>The implemented database fulfills all the acceptance criteria for the restaurant database design task</p> <p>New restaurants cannot be created without first creating a new restaurant owner account</p> <p>A new restaurant will be automatically created and successfully added to the database when a restaurant owner account is created</p>
Design registration and login UI	<p>Registration UI prompts user to select either customer or restaurant owner account type</p> <p>Login and registration UIs has a field for login name/email</p> <p>Login and registration UIs has a field for password</p> <p>Registration UI has a restaurant name field for restaurant owner account registration</p> <p>Registration UI has a registration button</p> <p>Registration UI has a confirmation screen after the new account has been successfully registered</p> <p>Login UI has a login button</p>
Implement registration and login UI using database	<p>The implemented UI fulfills all the acceptance criteria for the associated UI design task</p> <p>Selecting customer account type redirects to a UI with only login and password fields</p> <p>Selecting restaurant owner type redirects to a UI with login, password, and restaurant name fields</p> <p>The registration button cannot be pressed if any field is empty</p> <p>The registration button cannot be pressed if the given login name already exists in the database</p> <p>When the registration button is pressed, a new user account is successfully added to the database</p> <p>The login button cannot be pressed if any field is empty</p> <p>When the login button is pressed, the associated account ID is retrieved and cached</p>

Automate tests - As David (restaurant owner), I want to be able to register my account as a restaurant owner	All acceptance criteria for the tasks of this story must be covered in the tests
Design search algorithm for restaurants	<p>The algorithm returns all names from a list that include the given query text as a substring</p> <p>Only the names that include the query text are returned</p>
Code search function with respect to restaurant database	<p>The function returns the IDs of all restaurants in the database whose names include the given query text as a substring</p> <p>Only the restaurants whose names include the query text are returned</p>
Design search bar UI	<p>Search bar UI has a field that takes a search term</p> <p>Search bar UI has a list display to list various restaurant names</p>
Implement UI using backend functions	<p>The implemented UI fulfills all the acceptance criteria for the associated UI design task</p> <p>UI displays the list of restaurants found by the search function using the search term as the query text</p> <p>UI lists restaurants by name</p> <p>List display updates every time a key is pressed while the search term field is active</p>
Automate tests - As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements	All acceptance criteria for the tasks of this story must be covered in the tests
Design coupon database for restaurant-specific coupons	<p>The database is able to store multiple coupons</p> <p>Each coupon has a unique coupon ID</p> <p>Each coupon has a description</p> <p>Each coupon has a price (cost in points)</p> <p>Each coupon is associated with a restaurant</p>

Code the user database with user coupon creation functionality	<p>The implemented database fulfills all the acceptance criteria for the coupon database design task</p> <p>New coupons can be created</p> <p>Newly created coupons are successfully added to the database</p> <p>New coupons cannot be created without an associated restaurant ID</p>
Design coupon creation UI	<p>Coupon UI has a field for price (in point)</p> <p>Coupon UI has a field for coupon description</p> <p>Coupon UI has a create and a cancel button</p>
Implement UI using restaurant-coupon database	<p>The implemented UI fulfills all the acceptance criteria for the associated UI design task</p> <p>Coupon is successfully created and added to the database when the create button is pressed</p> <p>Coupon restaurant ID is the same as the restaurant ID associated with the current logged in account when the create button is pressed</p>
Automate tests - As David (restaurant owner), I want to create a coupon with my selected requirements	<p>All acceptance criteria for the tasks of this story must be covered in the tests</p>
Code database query for fetching coupons specific to a given restaurant owner account	<p>The function returns the IDs of all coupons in the database associated with the given restaurant ID</p> <p>Only the coupons associated with the given restaurant ID are returned</p>
Design UI for displaying list of coupons	<p>UI displays multiple coupons at once</p> <p>UI displays coupon ID, name, and description for each coupon</p>
Implement UI using backend functions	<p>The implemented UI fulfills all the acceptance criteria for the associated UI design task</p> <p>UI displays the full list of coupons found by the coupon fetching function using the restaurant ID associated with the current logged in account</p>

Automate tests - As David (restaurant owner), I want to be able to see all my restaurant's current coupons	All acceptance criteria for the tasks of this story must be covered in the tests
--	--

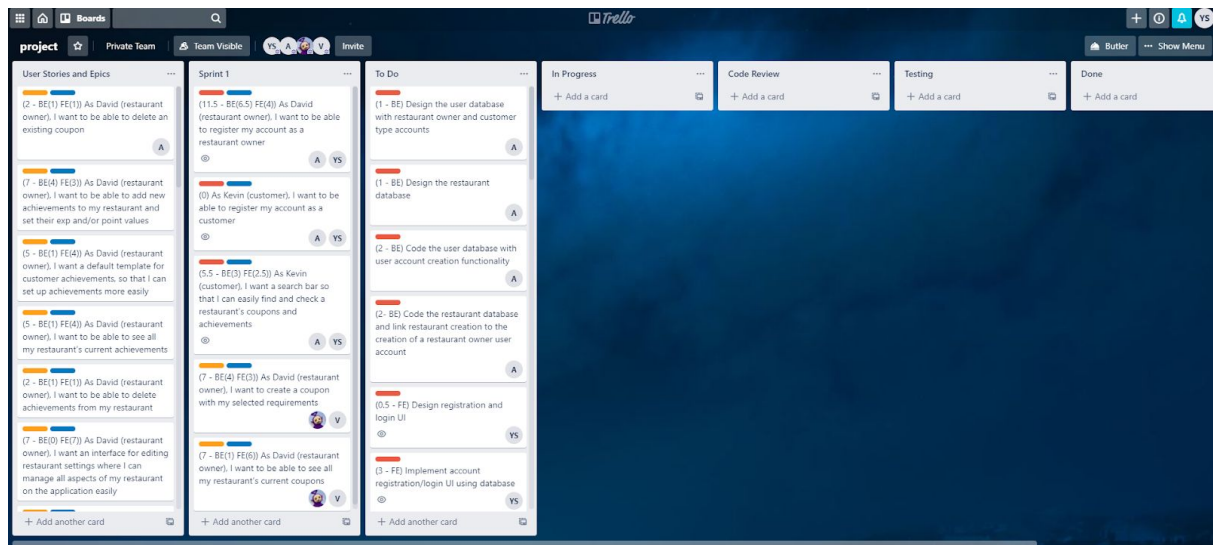
Team Task Assignments

Task	Cost	Priority	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Assignment
Design the user database with restaurant owner and customer type accounts	1	Critical	0	0	1	0	0	0	Anson
Design the restaurant database	1	Critical	0	0	1	0	0	0	Anson
Code the user database with user account creation functionality	2	Critical	0	0	0	2	0	0	Anson
Code the restaurant database and link restaurant creation to the creation of a restaurant owner user account	2	Critical	0	0	0	0	2	0	Anson
Design registration and login UI	0.5	Critical	0	0.5	0	0	0	0	Yolanda
Implement UI using database	3	Critical	0	0	1.5	1.5	0	0	Yolanda
Automate tests - BE - As David (restaurant owner), I want to be able to register my account as a restaurant owner	0.5	Critical	0	0	0	0	0	0.5	Anson
Automate tests - FE - As David (restaurant owner), I want to be able to register my account as a restaurant owner	0.5	Critical	0	0	0	0	0.5	0	Yolanda
Design search algorithm for restaurants	1	Critical	0	0	1	0	0	0	Anson
Code search function with respect to restaurant database	1.5	Critical	0	0	1	0.5	0	0	Anson

Design search bar UI	0.25	Critical	0	0.25	0	0	0	0	Yolanda
Implement UI using backend functions	1.75	Critical	0	0	0	0	1	0.75	Yolanda
Automate tests - BE- As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements	0.5	Critical	0	0	0	0	0.5	0	Anson
Automate tests - FE- As Kevin (customer), I want a search bar so that I can easily find and check a restaurant's coupons and achievements	0.5	Critical	0	0	0	0	0	0.5	Yolanda
Design coupon database for restaurant-specific coupons	2	High	0	0	0	1	1	0	Jiawei
Code the user database with user coupon creation functionality	1.5	High	0	0	0	1	0.5	0	Jiawei
Design coupon creation UI	0.5	High	0	0	0.5	0	0	0	Vanessa
Implement UI using restaurant-coupon database	2	High	0	0	0	2	0	0	Vanessa
Automate tests - BE - As David (restaurant owner), I want to create a coupon with my selected requirements	0.5	High	0	0	0	0	0	0.5	Jiawei
Automate tests - FE - As David (restaurant owner), I want to create a coupon with my selected requirements	0.5	High	0	0	0	0	0	0.5	Vanessa
Code database query for fetching coupons specific to a given restaurant owner account	0.8	High	0	0	0	0	0.5	0.3	Jiawei
Design UI for displaying list of coupons	1	High	0	0	1	0	0	0	Vanessa
Implement UI using	4	High	0	0	0	1	3	0	Vanessa

backend functions									
Automate tests - BE - As David (restaurant owner), I want to be able to see all my restaurant's current coupons	0.2	High	0	0	0	0	0	0.2	Jiawei
Automate tests - FE - As David (restaurant owner), I want to be able to see all my restaurant's current coupons	1	High	0	0	0	0	0	1	Vanessa

Task Board



Burndown Chart

