Team Byte Me

TEAM 07

Deliverable 3

Members:
Yiyang Zhou
Yifei Yin
Ruixin Zhuang
Xinyi Chen
Min Qi Zhang

June 20, 2020

Table of Contents

Environment Setup

Q: What tools will you use for your task board?

A: We are using ZenHub Board as our task board since it can nicely integrate with GitHub. However, we want to note that there will be some tasks with an “Epic” tag under a blue background in our task board screenshots. We are aware that there should be no epics in our sprint backlog. We are just using this feature in ZenHub Board to group tasks into one user story and have story points (estimates) calculated automatically. Actual “Epics” have “[Epic]” in the title, and tasks have “[#user story]” in the title. If there are no prefixes, then it is a user story.

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

A: We will use ZenHub for our burn-down chart since ZenHub can automatically generate the burn-down chart based on the tasks and user stories on the task board.

Q: Who will maintain the burn-down chart? How?

A: Min Qi Zhang is responsible for maintaining and submitting the burn-down chart every 48 hours to the GitHub repository. By the end of each day (before 11:59 pm), she will receive the progress from all the team members, then she will update the tasks on the burn-down chart.

Q: What is every team member’s role?

A:

- Yiyang Zhou is responsible for submitting the deliverables to the GitHub repository.
- Yifei Yin is the scrum master of our team. His primary responsibilities are solving the technical problems for team members, hosting stand-up meetings to get updates of the progress and ensuring team members are following scrum processes.
- Ruixin Zhuang is responsible for contacting the TA to ask any questions regarding the project.
- Xinyi Chen is responsible for taking the brief notes, taking the meeting attendance and keeping track of time for meetings.
- Min Qi Zhang is responsible for maintaining and submitting the burn-down chart every 48 hours to the GitHub repository.

Q: What tools, if any, will you use for communication?

A: We will use WeChat and Discord for communication. We have a WeChat group discussing any aspect of our project, such as consulting the team members’ ideas, posting the announcements, and sharing the information about the client meetings. We mainly use Discord for weekly meetings. Using Discord, we can conduct voice calls and screen sharing to improve our meetings’ productivity and efficiency.

Q: When do you plan to meet in person?

A: Due to the current circumstances, we are not able to meet in person. Instead, we will hold most of our meetings in Discord.
Besides the weekly meeting with the TA at 3:00 pm on Thursday, we have our weekly meeting every Saturday at 4:00 pm. At the end of each sprint, we also do a retrospective meeting right after the weekly meeting, during which we will review our project progress and make plans for the next sprint. To keep everyone on the same page, we have daily stand-up for 10 minutes at 1:00 pm on Discord.
In addition to the regular meetings, we may also arrange emergency meetings if necessary.

Q: How will you use your repository on GitHub?

A: We are using GitHub to store and submit assignments as directed by the course. That includes project deliverables (stored in a folder named Deliverables), burn-down charts and task board snapshots (stored in a folder named Burn-Down Chart & Task Board), project source code (stored in folders named *backend* and *frontend-**).

We will have one main branch called “master.” Members should create a feature branch with appropriate naming (see team agreement) from the master branch. They should then do feature development and commit their changes to the branch. They need to create a pull request on GitHub and may request another member to review and give feedback about their changes along the way. Finally, their last changes must be reviewed by another member. The author and reviewer(s) are both responsible for this merge; they need to make sure this merge commit does not unintentionally break other prominent features.

Git should automatically ignore Auto-generated files (see each .gitignore for the list of files), team members should not commit these files to the project. A single commit should contain a meaningful set of changes to the project files. Each commit on master or pull requests will be automatically style-checked and built, team members should confirm that these tasks pass before requesting reviews.

The name of each feature branch follows the format of *[<Task Number #>-]<User Story #>-<short description>* where task number is optional, short description should be no more than 20 characters long, and words are all smaller cases and separated using hyphen.

Commit messages should follow the following rules:

- Capitalize the first letter of the summary
- Do not end the summary with a period
- Use present tense
- Summarize should be less than 50 characters
- No spelling or grammar errors
- See example commit messages in .github/contributing.md

Q: Which machines will be used for development by each team member?
A:

Name	MacBook	Linux Laptop	Windows Laptop	Windows Home Computer
Yiyang Zhou	√		√	
Yifei Yin	√			
Ruixin Zhuang		√		√
Xinyi Chen	√			
Min Qi Zhang			√	√

Q: What is your DoD (definition of done)?
A:

1. All tasks of this user story are completed.
2. Changes have fulfilled the acceptance criteria.
3. Functionality verified by the client.
4. Commits do not break other functionalities.
5. Implementations have been reviewed by at least one other team member.
6. Codes have been style-checked.
7. Commits are merged back into the master branch and are ready to be released.

Additional notes:

1. Style-checks must be done using ESLint configurations given in each project.
2. We have configured GitHub Workflows to build and style-check commits automatically. Team members need to verify the basic functionalities of the system as well.

Product Backlog

There are four levels of priority in our project:

- 0 - emergency priority
- 1 - high priority: Assign to the basic setups required for the entire project and the requirements that are proposed by clients.
- 2 - medium priority: Assign to user stories that are relatively important.
- 3 - low priority: Assign to user stories that are not mandatory to the project, including filter, sort, search restaurant and analytics.

We usually do not assign priority 0 at the planning stage.

Each story point represents 2 hours of work; for example, a user story with 2 story points requires 4 hours to implement. For each user story, each team member proposes an estimate. We keep proposing until all team members agree on the same story point. When story points are different, the member who chooses a story point different from the majority needs to explain the reason behind it. After we have decided on an estimated story point for a user story, we further divide the points and assign them to the tasks using the same method described above.

User Stories

We have written Terminology and Explanations for all the wording we use. We put it in the last page of our report.

Customers

Register, Login and Logout

#52 As Jerry (a customer), I want to register a new account with my email and password. (Estimated Cost: 3; Priority: 1)
#54 As Jerry (a customer), I want to be able to login to the system with my email and password. (Estimated Cost: 2; Priority: 1)
#53 As Jerry (a customer), I want to be able to logout of the application. (Estimated Cost: 2; Priority: 1)

View Restaurants and Promotions

#3 As Jerry (a customer), I want to see a list of all the restaurants, so I can find the ones that attract me. (Estimated Cost: 5; Priority: 1)

#4 As Jerry (a customer), I want to search restaurants by their names, so that I can find restaurants that I know. (Estimated Cost: 3; Priority: 3)

#5 As Jerry (a customer), I want to check restaurants near my location, given a specific range. (Estimated Cost: 4; Priority: 3)

#6 As Jerry (a customer), I want to see a list of available promotions, so that I can participate and get benefits. (Estimated Cost: 3; Priority: 2)

#7 As Jerry (a customer), I want to sort promotions by the expiration dates, so that I will not miss any coupons. (Estimated Cost: 3; Priority: 2)

#8 As Jerry (a customer), I want to see all the current promotions of a specific restaurant. (Estimated Cost: 3; Priority: 1)

Participate in Promotions

#10 (Epic) As Jerry (a customer), I want to earn points and coupons from accomplishing achievements. (Estimated Cost: 4; Priority: 1)

View Coupons and Achievements

#13 As Jerry (a customer), I want to browse all coupons available for me. (Estimated Cost: 2; Priority: 1)

#14 As Jerry (a customer), I want to sort my coupons according to their expiration dates. (Estimated Cost: 3; Priority: 3)

#16 As Jerry (a customer), I want to filter my coupons according to the restaurant types. (Estimated Cost: 3; Priority: 3)

#18 As Jerry (a customer), I want to see obtained achievements for each restaurant. (Estimated Cost: 3; Priority: 3)

#19 As Jerry (a customer), I want to see my current available points at a specific restaurant and items that I can redeem with my points. (Estimated Cost: 5; Priority: 3)

Redeem

#21 As Jerry (a customer), I want to use coupons at check-out, so that I can get discounts from restaurants. (Estimated Cost: 4; Priority: 1)

#22 As Jerry (a customer), I want to redeem coupons and items with my points. (Estimated Cost: 3; Priority: 3)

Restaurant Owners

Register, Login and Logout

#55 As a restaurant owner, I want to register a new account with an email, restaurant name and password. (Estimated Cost: 3; Priority: 1)
#56 As a restaurant owner, I want to be able to logout of the system. (Estimated Cost: 2; Priority: 1)

Validate Coupons and Promotions

#29 As a restaurant owner, I want to check if a customer has satisfied our promotion requirements within 10 seconds. (Estimated Cost: 3; Priority: 1)

#28 As a restaurant owner, I want to validate customers' coupons within 10 seconds. (Estimated Cost: 3; Priority: 1)

View Promotions

#27 As a restaurant owner, I want to filter promotions according to their status (upcoming, active or expired). (Estimated Cost: 3; Priority: 3)

Rewards and Points

#20 As Bob (a restaurant owner), I want to be able to customize rewards for an achievement: give customers 100 points (more details specified below) or give them coupons (such as 20% off on any order, 5 dollars off on ordering this item) which they can use at our restaurant (Estimated Cost: 6; Priority: 1)

#17 As Bob (a restaurant owner), I want to set what customers can redeem with their points. For example, they can redeem a free ice cream with 300 points or get a 50% off coupon with 200 points. (Estimated Cost: 6; Priority: 3)

#15 (Epic) As Bob (a restaurant owner), I want to start a promotion collaboratively with other restaurant owners, and customers can get interchangeable coupons across both of the restaurants. (Estimated Cost: 6; Priority: 3)

Analytics

#12 As Bob (a restaurant owner), I want to see the demographics (age, gender and food preference) of my customers. (Estimated Cost: 5; Priority: 3)

#11 As Bob (a restaurant owner), I want to see how many people are participating in each of the promotions, so that I know which ones are more attractive. (Estimated Cost: 2; Priority: 3)

#9 As Bob (a restaurant owner), I want to see how many people have obtained each of the coupons, so that I know which ones are more attractive. (Estimated Cost: 4; Priority: 3)

Release Plan

Length of Sprints

We choose the length of our sprints to be one week. Usually, the duration for a sprint is between two weeks to four weeks. However, since we only have two months left for this semester, we have to shorten the sprint duration to seven days.

If we have three-week-long or four-week-long sprints, the stakeholders will only participate in two demos before the entire project is finished. We can not receive enough feedback to mitigate the risks. If we have two-week-long sprints, we will not be able to present our first release before the next interview, which is on July 2. That is the reason why we choose our length of sprints to be seven days.

Sprint Plan

Sprint Backlog

#31 As Alice/Bob (restaurant owners), I want to be able to login to the system with my email and password. (Total Estimated Cost: 8; Priority: 1)

Tasks

- **#30** Set up the back-end server & database and have one working API endpoint (Estimated Cost: 3)
- **#32** Create the UI for the login section (Estimated Cost: 3)
- **#33** Validate the password with the data stored in the back-end (Estimated Cost: 2)

Acceptance Criteria

- Given that Alice and Bob (as restaurant owners) want to use our product, when they open the login page, then there should be two input fields on the page for email and password.
- Given that Alice and Bob (as restaurant owners) have filled in the email and password correctly, when they click on the “Login” button, then they should be directed to the home page.
- Given that Alice and Bob (as restaurant owners) have filled in the email and password incorrectly, when they click on the “Login” button, then an error message shows up.

#39 As Alice/Bob (restaurant owners), I want to be able to see a list of promotions that I have posted. (Total Estimated Cost: 2; Priority: 1)

Acceptance Criteria

- Given that Alice and Bob (as restaurant owners) have logged into the system and there are promotions created for this restaurant, when they are on the promotions page, then they should see a list of all promotions.
- Given that Alice and Bob (as restaurant owners) have logged into the system and there are no promotions created for this restaurant, when they are on the promotions page, then they should see a text saying there are no promotions.

#26 As Alice/Bob (restaurant owners), I want to be able to create promotions with customizable titles, images, descriptions and other relevant information, so that promotion looks attractive, and rules are clear and transparent. (Total Estimated Cost: 6; Priority: 1)

Tasks

- **#34** Create the UI of “create promotion” page (Able to add title, description and rewards and to upload images) (Estimated Cost: 4)
- **#35** Save promotions (including the subtasks it contains) in the database (Estimated Cost: 2)

Acceptance Criteria

- Given that Alice and Bob (as restaurant owners) are on the home page, when they click on the “Create a Promotion” button, then they should be directed to the “create promotion” page.
- Given that Alice and Bob (as restaurant owners) are on the “create promotion” page, when they click on “Add a Subtask” button, then a popup should appear for them to enter information for subtasks.
- Given that Alice and Bob (as restaurant owners) are on the “create promotion” page, when they click on the “Upload Images” button, then a popup should appear for them to pick and upload images for this promotion.
- Given that Alice and Bob (as restaurant owners) are on the “create promotion” page, when they click on the “Remove Images” button, then the relevant image should be removed.
- Given that Alice and Bob (as restaurant owners) are on the “create promotion” page and are done with filling in all the fields and creating all the subtasks, when they click on the “Post a Promotion” button, then a message shows up informing them that data has been sent to the database.
- Given that Alice and Bob (as restaurant owners) are on the “create promotion” page with problematic input, when they click on the “Post a Promotion” button, then a proper error message shows up informing them what field(s) have problematic input.
- Given that Alice and Bob (as restaurant owners) are on the “create promotion” page, when they click on the “Discard a Promotion” button, then the promotion is discarded.

#23 As Bob (a restaurant owner), I want to customize rules for a promotion. For example, order one item from each of three categories within ten days. (Total Estimated Cost: 5; Priority: 1)

Tasks

- **#36** Add functionality of creating/deleting subtasks (Estimated Cost: 3)
- **#40** Add functionality of setting starting and expiration dates for promotions (Estimated Cost: 2)

Acceptance Criteria

- Given that Bob (as restaurant owner) wants to add a subtask to the promotion, when he clicks on the “Add a Subtask” button, then a popup should show up with a text box to enter the description.
- Given that Bob (as restaurant owner) has customized a subtask, when he clicks on the “Save” button, then this customized subtask is shown on the promotion template.
- Given that Bob (as restaurant owner) wants to set starting and expiration dates, when he clicks on the starting date or expiration date, then a date selection box should show up.
- Given that Bob (as restaurant owner) wants to set a limit on how long customers need to complete the tasks once started, when he clicks on “Validate for …”, then a date duration selection box should show up.
- Given that Bob (as restaurant owner) has set updates for promotions, when the real world time changes, the status of promotions should be changed to “Expired/Activated” in the database accordingly.

#24 As Alice (a restaurant owner), I want to set rules for promotion by picking from a list of presets. For example, visit the restaurant ten times within a month. (Total Estimated Cost: 5; Priority: 1)

Tasks

- **#37** Popup for a preset customization form for each template (Estimated Cost: 2)
- **#38** Add some presets templates for tasks, including (Estimated Cost: 3)
    - N-th visit
    - Order X specific drinks
    - Visit N times within X days
    - Spend X dollars in an order
    - Buy X get Y free

Acceptance Criteria

- Given that Alice (as restaurant owner) wants to select a preset, when she clicks on the “Preset Templates” button, then a list of presets are displayed with a brief description of the template.
- Given that Alice (as restaurant owner) wants to use a preset for her task, when she chooses a preset template, then a preset customization form pops out for easy template setup.
- Given that Alice (as restaurant owner) completed the preset customization form, when she clicks “Finish,” then subtasks are loaded accordingly in the promotion creation page.

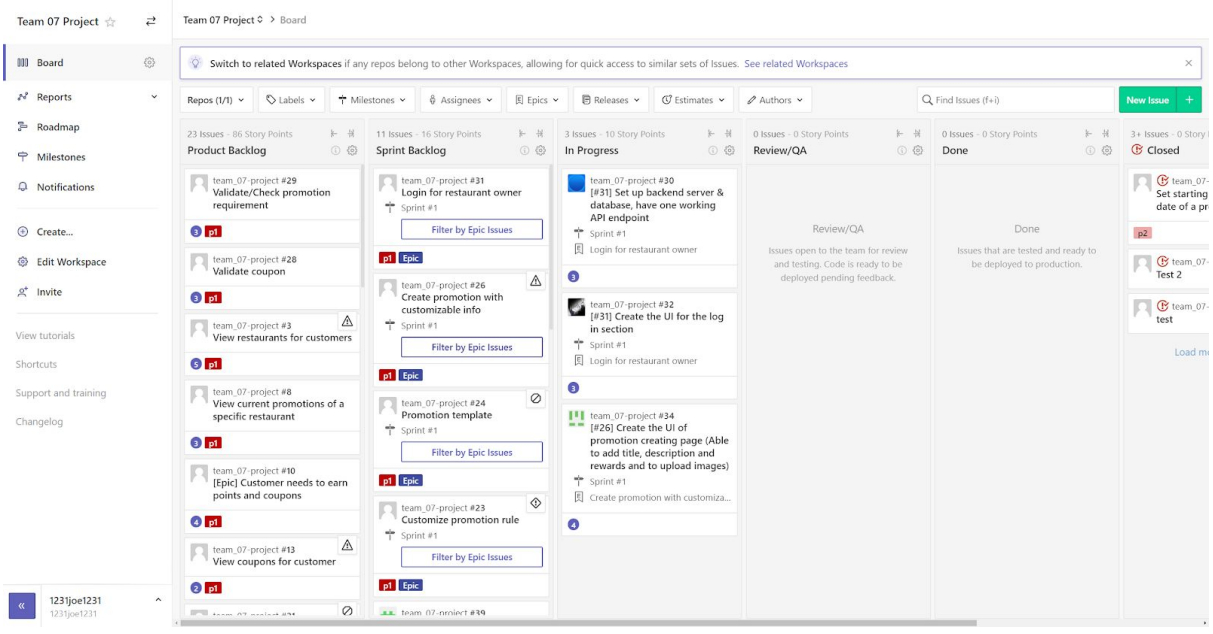
Division of Labour

Task Number	Assignee	Cost	Expected Start On (June)	Expect End On (June)
#36	Yiyang Zhou	3	24	27
#40	Yiyang Zhou	2	25	27
#30	Yifei Yin	3	22	22
#37	Yifei Yin	2	27	28
#32	Ruixin Zhuang	3	25	26
#33	Ruixin Zhuang	2	27	28
#34	Xinyi Chen	4	25	28
#35	Xinyi Chen	2	27	28
#38	Min Qi Zhang	3	25	27
#39	Min Qi Zhang	2	27	28

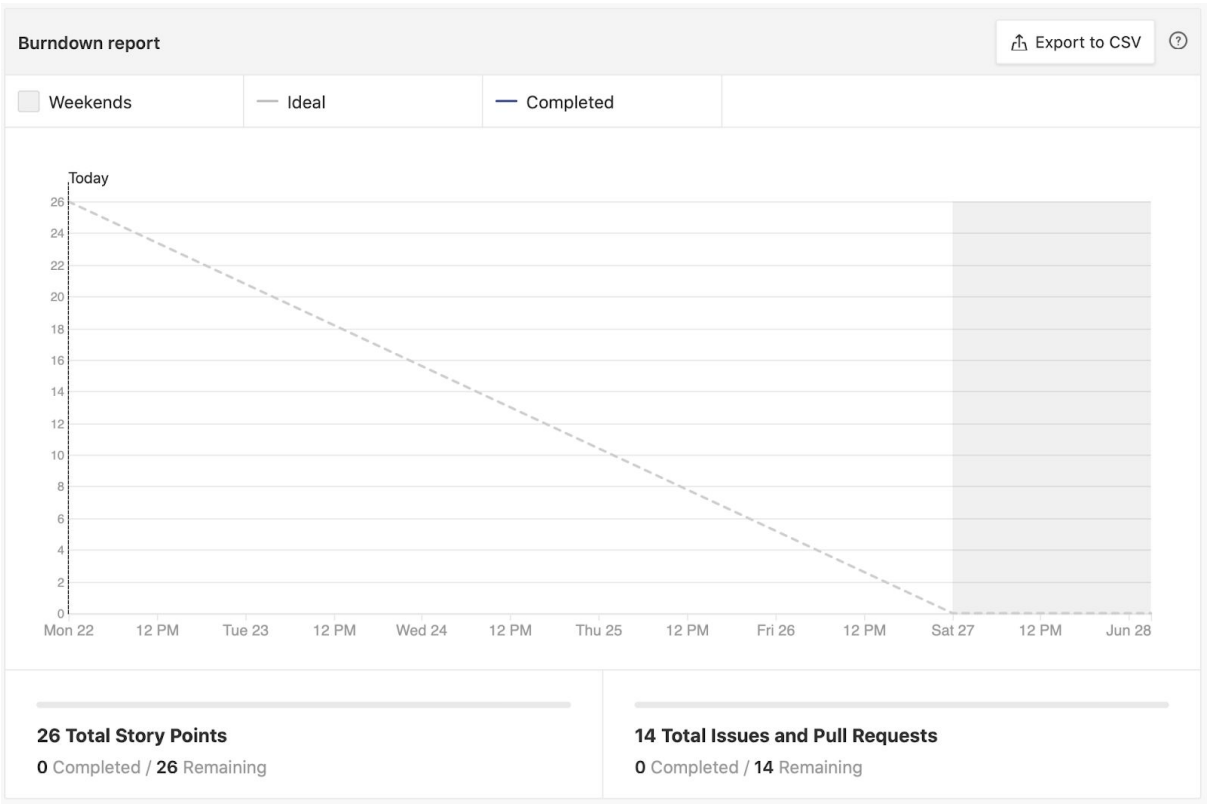
Snapshots

Please check out this invitation link to see our board.
https://app.zenhub.com/workspaces/team-07-project-5eee6d65ba3415001991daf1/board

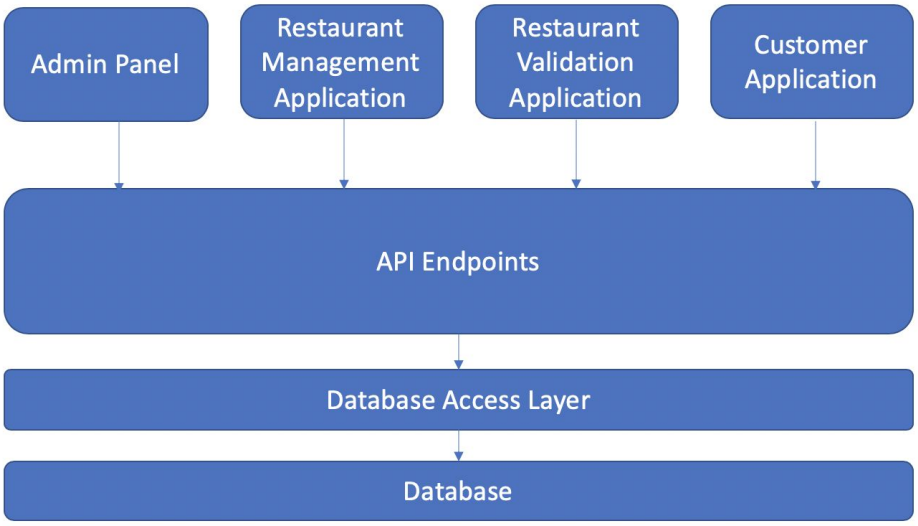
Initial Setup of the Task Board



Initial Setup of the Burn-down Chart



System Components



Layer 1 - Front-End Applications

- The admin panel is used by developers to set up, inspect and manage back-end data. It is automatically generated by the back-end framework we are using.
- The restaurant management application is used by restaurant owners/managers to manage achievements, promotions, coupons, etc. for their restaurant.
- The restaurant validation application is used by restaurant staff to validate customer's coupons.
- The customer application is used by customers. They can use it to explore restaurants, participate in promotions, etc.

All these components use the back-end API endpoints to fetch and update user data. They depend on the back-end, but they do not depend on each other.

Layer 2 - API Endpoints

This is the public application programming interface for the system. All applications must access data through this interface. It uses the database access layer to access and manipulate the database.

Layer 3 - Database Access Layer

This layer provides a more friendly and safe interface of accessing and manipulating data of the database. It is used directly inside API endpoints handlers. It needs to be connected to a compatible database.

Layer 4 - Database

This layer stores all the raw user data of the system, including details about restaurants, customers, achievement, etc. This layer is required for all other components.

Retrospection

From deliverable 2 to deliverable 3, we have set up both the back-end with Strapi and front-end with React. During this time, we focused on the essential features for restaurant owners to customize their promotions. For the product backlog, we have added the user stories to the task board and assigned the priority to each of them. For the sprint backlog, we have selected some high-priority user stories from the product backlog and provided the acceptance criteria; we have also broken down some technically complex user stories into smaller tasks.

We have scheduled 26 story points for sprint #1 and expected to complete these story points at a constant velocity. However, we only completed 16 story points in sprint #1. At the beginning of the sprint, story points decreased slowly; as we got closer to the end of the first sprint, the story points decreased faster.

We only had one sprint. We have tried our best to follow our plan, including daily stand-up meetings, maintaining burn-down charts, and completing coding tasks within designated time frames. Unfortunately, we have planned too much work in this sprint. Thus, we did not finish everything in the sprint backlog. These are some main challenges we have met in this sprint:

1. We had to spend time setting up and getting ourselves accustomed to using Strapi and React because many of us were using these tools for the first time.
2. Sometimes we had different approaches in our front-end and back-end design, which can cause problems when merging the codes.

To overcome the challenges, we have arranged extra meetings for team members to help each other with the tools and answered questions. We also had additional meetings for code review and discussion to make sure everyone is on the same page and has no conflicts in design.

However, we did not get a chance to use our contingency plan since all situations mentioned in the contingency plan have not happened to our team.

Plan for the next sprint

We will plan a more appropriate workload in the next sprint since we have planned too much in the first sprint. This also caused our burn-down chart to seem not productive.
We will spend more time choosing user stories and assigning tasks to ensure less dependency among tasks. Some team members cannot do anything in the first half of the sprint because their tasks depend on other tasks.
We will spend more time considering the acceptance criteria. In the first sprint, we found that some acceptance criteria differ from what clients actually needed as we frequently join the client meeting and send emails to confirm their ideas while implementing our system. Implementing unnecessary functionalities and unexpected functionalities caused us plenty of time. This could be avoided at the beginning of the sprint. In the next sprint, we will plan in more detail and confirm our plan with the client more early.

Changes to Deliverable 3

The following are the changes we made in deliverable 3 according to Jaya's feedback:

1. We have changed the chart of division of labor, so that each task has a more specific time period to complete.
2. We have updated our plans for daily stand-ups and retrospective meetings.
3. We have provided more details regarding the usage of GitHub: rules for branch name, commit, pull request and auto-generated files.
4. We have additional notes under the DoD section to clarify how we construct the style-check.
5. We have added one more requirement for DoD.
    a. "Functionality verified by the client."
6. We have explained how we calculated the story points and priorities for each user story/task.

Changes to Our Product Backlog

1. We have converted user story #25 to a task. It is a task of the user story #23, because starting and expiring dates are part of the rules.
    a. "As Alice/Bob (restaurant owners), I want to be able to add starting dates and expiring dates, so that I can schedule promotions in advance and set a deadline for promotions." (#25)
    b. "As Bob (a restaurant owner), I want to customize rules for a promotion. For example, order one item from each of three categories within ten days." (#23)
2. We have added user story #39 to our product backlog.
    a. "As a restaurant owner, I want to be able to see a list of promotions that I have posted." (#39)
3. We have changed the priority of task #20 to level 1 since we figured out that rewards are part of the achievements. As required by the clients, "achievements should be fully customizable/configured by the restaurant owners".
    a. As Bob (a restaurant owner), I want to be able to customize rewards for an achievement: give customers 100 points (more details specified below) or give them coupons (such as 20% off on any order, 5 dollars off on ordering this item) which they can use at our restaurant.
4. We have added 5 more user stories (#52, #53, #54, #55 and #56) for registration, login and logout. We asked our client about whether we should include those features, the answer we received is yes.
    a. "As Jerry (a customer), I want to register a new account with my email and password." (#52)
    b. "As Jerry (a customer), I want to be able to logout of the application." (#53)
    c. "As Jerry (a customer), I want to be able to login to the system with my email and password." (#54)
    d. "As a restaurant owner, I want to register a new account with an email, restaurant name and password." (#55)
    e. "As a restaurant owner, I want to be able to logout of the system." (#56)

Terminology and Explanations

Restaurant owner: This includes both manager and staff. Since the required functionality overlaps (restaurant manager has one more feature than staff, i.e., creating a promotion), we combine these two types of users into one persona. Our user stories include all the functionality and features wanted by both managers and staff.

Promotion: Contain a list of subtasks created by the restaurant owner. The process of participating in a promotion, completing tasks and validating promotions is the achievement path that our TA has mentioned in our weekly meeting. After this process, the customer will gain the achievement.

Reward: Since achievement is an abstract and general concept, we use the word “reward” to indicate the benefit of going through the achievement path and acquiring the achievement. In our system, we have two kinds of reward, points and coupons.

As we have explained our system during the weekly meeting with our TA, the process of our system goes like the following:

1. First, the restaurant owner creates a promotion. Once a promotion is created, customers should be able to view it.
2. Second, customers can choose to participate in a particular promotion by clicking the button "Participate".
3. Then, after he/she completes all the tasks required by this particular promotion, he/she can press the button to let the restaurant owners verify. Our system will automatically send a message to the corresponding restaurant.
4. The restaurant staff needs to verify it manually and press the button "Verified" if the customer passes all the subtasks for some achievement. After that, customers will receive this achievement and corresponding reward.
5. Finally, he/she can redeem rewards with coupons or points.

This whole process involves quite a lot of technical and implementation details. Therefore, we use words such as "validate", "verify", "use" to simplify the above process, and they make more sense from the user's perspective. However, we will definitely include all the details when we break those user stories into tasks and write the acceptance criteria.