Foodie Connector

Sprint 1 Planning Document

Team 25: Zequan Wu, Shuqi Ma, Shiwen Xu, Xiaonan Shen

Sprint Overview

Overview

During the first sprint, we are planning to create initial frontend and backend applications, set up developing environments, and configure the automatic testing and deployment. Besides this, we are going to implement the functions related to authentication, such as register, login, and password resetting.

Scrum Master

Xiaonan Shen

Meeting Schedule

Every Tuesday and Thursday 10:00 am - 11:00 am

Risks/Challenges

Since it is our first sprint, most team members need plenty of time getting familiar with the languages, frameworks, and tools we are using. Also, it will be difficult to keep a high code quality when we are not very familiar with the languages and frameworks. So it is important to discuss and make decisions on code styles and API interfaces before implementing anything, which also takes some time. We will be using lint and automatic testing to prevent bad codes.

Current Sprint Detail

User Story #1

As a user, I would like to register for an account using my email address.

Sub-task	Estimated Time	Responsible Person
Create ‘users’ table on the database	1h	Xiaonan Shen
Implement an API for registering	1h	Xiaonan Shen
Create a user interface for registering on the web client	4h	Shiwen Xu
Unit test - correct and incorrect input	2h	Shiwen Xu

Acceptance Criteria

- Given that the API for registering is implemented, when the API is called, the email and hashed password will be inserted into the database if the email address does not already exist. Then the result of registering will be returned.
- Given that the user interface for registering is implemented, when the user types in the email and password to register, a user-friendly message of whether the register succeeds will be shown. The user needs to enter the password twice, and password match will be verified in the frontend.
- Given that the register for an account is functional, a ‘users’ table should exist in the database, and new user information can be inserted in the table.

User Story #2

As a user, I would like to log in.

Sub-task	Estimated Time	Responsible Person
Implement an API for login	1h	Xiaonan Shen
Create a user interface for login on the web client	4h	Shiwen Xu
Unit test - correct and incorrect passwords	2h	Shiwen Xu

Acceptance Criteria

- Given that the API for login is implemented, when the API is called, the email will be used to query the user information from the database and the password will be hashed to verify.
- Given that the user has been successfully logged in, the user information will be stored in the session.
- Given that the user interface for login is implemented, when the user types in the email and password to log in, a user-friendly message of whether the login succeeds will be shown.

User Story #3

As a user, I would like to reset my password if I forget it.

Sub-task	Estimated Time	Responsible Person
Setup SMTP client on the backend so that we can send emails to users	2h	Xiaonan Shen
Implement an API for sending verification codes	1h	Xiaonan Shen
Implement an API for resetting password	1h	Xiaonan Shen
Create a user interface for resetting password on the web client	5h	Shuqi Ma
Unit test - correct and incorrect verification code (via email)	3h	Shiwen Xu

Acceptance Criteria

- Given that the API for sending verification code is implemented, when the API is called, an email containing a verification code will be sent to the given email address.
- Given that the verification code received and enter the code in the specific input area, the user could enter the new password and confirm the resetting.
- Given that the API for resetting password is implemented, when the API is called, the new password will be hashed and stored in the database if the verification code matches our record.

User Story #4

As a user, I would like to save my address so that I don't have to type them in every time.

Sub-task	Estimated Time	Responsible Person
Create ‘addresses’ and ‘users_addresses’ tables in the database	1h	Xiaonan Shen
Implement an API that returns all the addresses of the current user	1h	Xiaonan Shen
Implement an API for adding addresses	1h	Xiaonan Shen
Implement an API for editing addresses	1h	Xiaonan Shen
Implement an API for deleting addresses	1h	Xiaonan Shen
Setup Google Maps JavaScript API on the web client	4h	Shuqi Ma
Create a user interface for showing address list	2h	Shuqi Ma
Create a user interface for adding addresses	2h	Shuqi Ma
Create a user interface for editing addresses	2h	Shuqi Ma

Acceptance Criteria

- Given that the APIs for adding is implemented, when the APIs are called, the address will be added to the database.
- Given that the APIs for editing is implemented, when the APIs are called, the address willbe updated in the database.
- Given that the API for deleting is implemented, when the API is called, the corresponding record will be removed from the database.
- Given that the Google Maps JavaScript API is successfully set up and the user interfaces are implemented, when the user is typing in the address, autocomplete will be shown.

User Story #44

As a user, I would like to save my payment information.

Sub-task	Estimated Time	Responsible Person
Create ‘cards’ table in the database	1h	Xiaonan Shen
Setup PHP library for Stripe API	1h	Xiaonan Shen
Implement an API that returns all the payment methods of the current user	1h	Xiaonan Shen
Implement an API for adding payment methods	1h	Xiaonan Shen
Implement an API for editing payment methods	1h	Xiaonan Shen
Implement an API for deleting payment methods	1h	Xiaonan Shen
Setup Stripe.js on the web client	4h	Zequan Wu
Create a user interface for showing payment methods	4h	Zequan Wu
Create a user interface for adding payment methods	2h	Zequan Wu
Create a user interface for editing payment methods	2h	Zequan Wu

Acceptance Criteria

- Given that the APIs for adding is implemented, when the APIs are called, the address will be added to the database.
- Given that the APIs for editing is implemented, when the APIs are called, the address will be updated in the database.
- Given that the API for deleting is implemented, when the API is called, the corresponding address will be removed from the database.
- Given that the user interface for showing payment methods, users can view all saved payment methods on web client.
- Given that the user interface for adding payment methods, users can add payment methods on web client.
- Given that the user interface for editing payment methods, users can edit payment methods on web client.

User Story #5

As a user, I would like to change my password.

Sub-task	Estimated Time	Responsible Person
Implement an API for changing password using the old password	1h	Xiaonan Shen
Create a user interface for changing password	5h	Shuqi Ma
Unit test - correct and incorrect old password	2h	Shuqi Ma

Acceptance Criteria

- Given that the API for changing the password using old password, when the server received the request, the old password will be hashed and verified. If old password match, the new password will be hashed and stored into the database, otherwise, return an error code.
- Given that the changing password is functional, the new password will be stored in ‘users’ table in database.
- Given that the user interface for changing password is implemented, when users enter the old password, new password and submit, the server will update their passwords.

User Story #6

As a user, I would like to change my email address.

Sub-task	Estimated Time	Responsible Person
Implement an API for changing the email address	1h	Xiaonan Shen
Create a user interface for changing the email address	4h	Shuqi Ma
Unit test - correct or incorrect verification code	2h	Shiwen Xu

Acceptance Criteria

- Given that the API for changing email address is implemented, when the API is called, the user’s email address will be updated in the database.
- Given that the API for sending verification code is implemented, when the API is called, an email containing a verification code will be sent to the new email address.
- Given that the user interface for changing email address is implemented, when users enter the verification code, the server will update their emails with the new one.
- Given that the changing email address is functional, the new email will be stored in the database if the verification code matches our record.

User Story #7

As a user, I would like to have a general address that based on my current geolocation so that I don’t have to type in my address before I decided.

Sub-task	Estimated Time	Responsible Person
Create a functionality that requests current location from the browser	5h	Shiwen Xu
Create API call that requests a human-readable address from Google Maps API	5h	Shiwen Xu

Acceptance Criteria

- Given that the functionality for requesting current location is implemented, when the user first opens the restaurant list page, he/she will be prompted to allow the website to use the device’s location.
- Given that the API call is created, when the browser returns the location of the device and make the API call, the user will see an address that is near his/her current location.
- Given that the API call is created, after the API call, the frontend will receive a place ID that can be used to request for nearby restaurant list.

User Story #8

As a user, I would like to see the available restaurants, together with distance, estimated delivery time, order minimum, and delivery fee.

Sub-task	Estimated Time	Responsible Person
Add ‘restaurants’ table in the database	1h	Xiaonan Shen
Implement an API for list all nearby restaurants available at that time	1h	Xiaonan Shen
Create a user interface for view the available restaurants’ information	5h	Zequan Wu
Unit test - different local time for available restaurants	2h	Shiwen Xu

Acceptance Criteria

- Given that the API for listing all available restaurants at that time, when the server received the request, the client’s local time and address will be used to query the available restaurants from the database.
- Given that the user interface for filtering the restaurant list is implemented, after the address is generated or typed in by the user, a list of all available restaurants nearby will be shown.
- Given that the user interface is implemented, after the restaurant list is shown, the user will be able to see distance, estimated delivery time, order minimum, and delivery fee for each restaurant.

User Story #9

As a user, I would like to search for specific restaurants by name.

Sub-task	Estimated Time	Responsible Person
Add search bar in user story #8 user interface	2h	Zequan Wu
Unit test - existing and non-existing restaurant names	2h	Zequan Wu

Acceptance Criteria

- Given that the search bar is implemented, when a user enters the name of the restaurant, the user will see real-time search results of the restaurant.
- Given that the search bar is implemented, when a user enters the name of the restaurant but there is no match, a message will be shown.
- Given that the search bar is implemented, when a user is typing in the search bar, the search history will be shown below.

User Story #10

As a user, I would like to filter the restaurant list by category, distance, and estimated delivery time.

Sub-task	Estimated Time	Responsible Person
Implement the API for filtering restaurants by given constraints	1h	Xiaonan Shen
Create a user interface for view the filtered restaurant list returned by the server	4h	Zequan Wu

Acceptance Criteria

- Given that the API for filtering the restaurant list is implemented, when the server received the request, the category, distance and estimated delivery time will be used to query the restaurants from the database.
- Given that the user interface for filtering the restaurant list is implemented, when the user opens the restaurant list page, all available filter factors will be shown.
- Given that the user interface for filtering the restaurant list is implemented, when a user selects restaurant category, distance and estimated delivery time, the user will see a list of restaurants match the selections.

User Story #13

As a user, I would like to sort the restaurant list by distance, estimated delivery time, and rating.

Sub-task	Estimated Time	Responsible Person
Implement the API for sorting restaurants by given constraints	1h	Xiaonan Shen

Create a user interface for sorting restaurants	2h	Zequan Wu

Acceptance Criteria

- Given that the API for sorting restaurants is implemented, when the API is called, it will return a restaurant list that is ordered by the given column.
- Given that the user interface for sorting restaurants is implemented, when the user open the restaurant list page, all available sorting options will be shown.
- Given that the user interface for sorting restaurants is implemented, when the user selects criteria to sort, the user will see the restaurant sorted in that order.

User Story #35

As an administrator, I would like to login to a dashboard.

Sub-task	Estimated Time	Responsible Person
Create ‘admin_users’ table in the database	1h	Xiaonan Shen
Create authentication guard for admin users in the Laravel backend	1h	Xiaonan Shen
Setup Voyager, a Laravel Admin Package	1h	Xiaonan Shen

Acceptance Criteria

- Given that the Voyager is successfully set up, administrators will be able to create admin users using php artisan commands.
- Given that the Voyager is successfully set up, an admin user will be able to login.
- Given that the administrator is successfully logged in, he/she will be redirected to the dashboard.

User Story #36

As an administrator, I would like to manage restaurants.

Sub-task	Estimated Time	Responsible Person
Add new BREAD (Browser, Read, Edit, Add, Delete) model for restaurants in the Voyager package	1h	Xiaonan Shen

Acceptance Criteria

- Given that the BREAD model is successfully added, administrators will be able to see the list of all restaurants.
- Given that the BREAD model is successfully added, when add a new restaurant, the information will be stored in the database.
- Given that the BREAD model is successfully added, when edit a restaurant, the information will be updated in the database.
- Given that the BREAD model is successfully added, when remove a restaurant, the corresponding record will be deleted from the database.

Tools and Code Reviewing

Sub-task	Estimated Time	Responsible Person
Set up local develop environment	4h (1h / person)	Zequan Wu, Shuqi Ma, Shiwen Xu, Xiaonan Shen
Review and format each others code	8h (2h / person)	Zequan Wu, Shuqi Ma, Shiwen Xu, Xiaonan Shen

Backlog

1. ~~As a user, I would like to register for an account using my email address.~~
2. ~~As a user, I would like to log in.~~
3. ~~As a user, I would like to reset my password if I forget it.~~
4. ~~As a user, I would like to save my address so that I don't have to type them in every time.~~
5. ~~As a user, I would like to change my password.~~
6. ~~As a user, I would like to change my email address.~~
7. ~~As a user, I would like to have a general address based on my current geolocation so that I don’t have to type in my address before I decide.~~
8. ~~As a user, I would like to see the available restaurants, together with distance, estimated delivery time, order minimum, and delivery fee.~~
9. ~~As a user, I would like to search for specific restaurants by name.~~
10. ~~As a user, I would like to filter the restaurant list by category, distance, and estimated delivery time.~~
13. ~~As a user, I would like to sort the restaurant list by distance, estimated delivery time, and rating.~~
16. As a user, I would like to see the menus for each available restaurants.
17. As a user, I would like to see sub options for each item on the menu.
18. As a user, I would like to create group orders.
19. As an order creator, I would like to generate a link of the order.
20. As an order creator, I would like to share the link to others using text messages, emails, or other third party applications.
21. As an order creator, I would like to generate a QR code of the order so that it will be more convenient for face-to-face sharing.
22. As an order creator, I would like to set the order to private so that only people who know the link or the QR code can see and join it.
23. As an order creator, I would like to set the order public so that it can be seen and joined by all nearby users.
24. As an order creator, I would like to set a time limitation in which other users can join the order.
25. As an order creator, I would like to be the only user who confirms the order.
26. As an order creator, I would like other users set themselves as ready before I could confirm the order so that we can make sure everyone has selected the items they want.
27. As a user, I would like to see all nearby public orders that I can join.
28. As a user, I would like to split the total cost so that I only pay the price and delivery fee proportional to the food chosen by me.
29. As a user, I would like to add friends.
30. As a user, I would like to invite friends from the friend list to join the order.

31. As a user, I would like to track my order status.
32. As a user, I would like to track my driver’s real-time location.
33. As a user, I would like to rate the restaurants from which I’ve ordered.
34. As a user, I would like to see my order history so I could choose the same restaurant again easily.
~~35. As an administrator, I would like to login to a dashboard.~~
~~36. As an administrator, I would like to add restaurants.~~
~~37. As an administrator, I would like to update restaurants.~~
~~38. As an administrator, I would like to remove restaurants.~~
39. As a administrator, I would like to manage restaurants' menus.
40. As a user, I would like to receive notifications when my order status updated.
41. As a restaurant employee, I would like to see the detail of all active orders.
42. As a restaurant employee, I would like to update the status of orders.
~~44. As a user, I would like to save my payment information.~~