

Sprint Planning

Date: 4/17/19

Time: 11:30 Am

Location: 310 Lecture

Participants: Tong, Kevin, Chris, Aahad, Nick



Sprint Backlog

--PRODUCT BACKLOG--

1. [20] Web application must be secure and protect confidentiality of a user's ImHungry data - **WE NEED TO USE SSL/HTTPS**
3. [20] Allow for pagination of results returned by the search.
4. [30] View results of prior searches by clicking on a quick access list that shows prior search terms.

6. [25] Keep track of a grocery list for selected recipes - **BUILD FRONT-END**
7. [10] Reorder any of the three predetermined lists.
8. [10] Set the radius of the restaurant search - **CHANGE TO MILES**

--- SPRINT BACKLOG ---

1. [20] Web application must be secure and protect confidentiality of a user's ImHungry data - **WE NEED TO USE SSL/HTTPS**
3. [20] Allow for pagination of results returned by the search.
4. [30] View results of prior searches by clicking on a quick access list that shows prior search terms. - **BUILD FRONT-END**
6. [25] Keep track of a grocery list for selected recipes - **BUILD FRONT-END, DISALLOW DUPLICATES**
7. [10] Reorder any of the three predetermined lists.
8. [10] Set the radius of the restaurant search - **CHANGE TO MILES, RETURN "SORRY" MESSAGE WHEN NO RESTAURANTS FOUND**

Group G chose the afore listed items because this is the final sprint and all required features should be completed by the end of the sprint. After receiving feedback from stakeholders in previous sprints, we have a better idea of what is required and the amount of points that fulfilling each requirement will give our team. With this new info we can prioritize the completion of certain requirements accordingly such as pagination and saving grocery lists.

Task Breakdown

Application security:

- Backend: Kevin, Chris
- Frontend: Aahad, Tong

Data persistence for users:

- Backend: Kevin, Chris
- Frontend: Aahad, Tong

Pagination:

- Backend: Kevin, Chris
- Frontend: Aahad, Tong

Quick access, Prior searches:

- Backend: Kevin, Chris
- Frontend: Nick, Aahad

Modern, attractive:

- Frontend: Tong, Aahad

Restaurant radius:

-Backend: Nick, Tong

-Frontend: Aahad, Tong

Application Security / Data Persistence / Quick Access

- Generate test cases
 - SQL Password requirements:
 - Password incorrect
 - Password correct
 - New account
 - Username already exists
 - Data persistence
 - Results store over session
 - Lists store over session
 - Quick access stores over session
 - Quick access
 - When clicked, perform search again
- SQL implementation
 - Entity-relationship schema development
 - Username
 - Password
 - Predefined lists
 - Previous searches
 - Integration with backend code
 - Satisfy all test cases
- Backend refactoring
 - Manage List singleton becomes SQL read
 - Searches add search terms to SQL backlog

Restaurant radius

- Generate test cases
 - Limit returned restaurants to inside radius
 - Default radius search returns restaurants
 - Invalid radius does not work
- Implementation
 - Read through other available code base
 - Evaluate if using Google API is the best option
 - Add functionality to front end
 - Pass parameter to back end
 - Add parameter to radius search API call
 - Satisfy all test cases

Modern & Attractive

- Generate test cases
 - Login modal appears when Login button clicked on
 - User gets redirected to logged in search page when they successfully login

- User gets error notification if they input an incorrect username or password when successfully logged in
 - Implementation
 - Make system look modern and attractive
 - Use bootstrap
 - Decide on an overall theme for the application
 - Redesign Recipe and Restaurant Pages
- Satisfy all test cases