Sprint Planning

Date: 4/3/19
Time: 10:00 Am
Location: 310 Lecture
Participants: Tong, Kevin, Chris, Aahad
GitHub Link:

Sprint Backlog

--PRODUCT BACKLOG--

1. [20] Web application must be secure and protect confidentiality of a user's ImHungry data
2. [15] Maintain information beyond just a single session
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.

5. [20] User interfaces must look modern and be attractive -- **USER INTERFACE MUST BE CONSISTENT WITH SNAZZY FONT**
6. [25] Keep track of a grocery list for selected recipes
7. [10] Reorder any of the three predetermined lists.
8. [10] Set the radius of the restaurant search

--- SPRINT BACKLOG ---

1. [20] Web application must be secure and protect confidentiality of a user's ImHungry data **[WORKING BLACKBOX TESTS]**
2. [15] Maintain information beyond just a single session **[WORKING BLACKBOX TESTS]**
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. **[WORKING BLACKBOX TESTS]**
5. [20] User interfaces must look modern and be attractive -- **USER INTERFACE MUST BE CONSISTENT WITH SNAZZY FONT**
6. [25] Keep track of a grocery list for selected recipes
8. [10] Set the radius of the restaurant search**[WORKING BLACKBOX TESTS]**

Group G chose the afore listed items because they either need to be improved upon after sprint 1, or have similar underlying requirements that can completed in tandem with one another. We also did not want to add to many items to our product backlog after not receiving feedback after the first Sprint due to some issues that we had with Capybara. Starting on some of the more difficult requirements was a also a priority for us, such as the grocery list, pagination, and quick access for past searches in the I'm Hungry System.

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