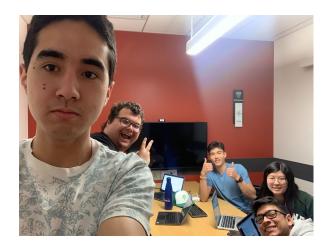# Sprint Planning

Date: 3/22/19
Time: 1PM
Location: JFF 325
Participants: All
GitHub Link:
https://github.com/njpietrow/ImHungry/commit/968c03bcd0462463b40fb763928aa327d1042293

Sprint Backlog

1. Web application must be secure and protect confidentiality of a user's ImHungry data
2. Maintain information beyond just a single session
4. View results of prior searches by clicking on a quick access list that shows prior search terms.
5. User interfaces must look modern and be attractive
8. Set the radius of the restaurant search

Group G chose the afore listed items because they are either related to one another, simple, or will require ongoing changes through sprints 2 and 3. Items 1, 2, and 4 on the product backlog require a database to store user accounts and information beyond a single session. Item 5 will need to change continuously as we implement new features and receive feedback from product owners. Lastly, setting the radius of the restaurant search will be relatively simple because the code base from project 1 we will be using made a call to a google API that had search radius as a parameter. We wanted to begin working on more than half of the total product backlog items to get an early start and have adequate time to make necessary changes in Sprints 2 and 3.

Task Breakdown

Application security:

-Backend: Kevin, Chris
-Frontend: Aahad, Tong
Data persistence:
-Backend: Kevin, Chris
-Frontend: Aahad, Tong
Quick access:
-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 stuff:
    - 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