Sprint 2 Retrospective

- Date: 4/17/19
- Time: 11:00 am
- Location: SGM 101
- Participants:
    - Farrell Hohman
    - Rahul Mohan
    - Alexander Reyes
    - Kevin Yeung
    - Weihang Zhang

For Sprint 2, we aimed to complete REQ 1, REQ 3, REQ 4, REQ 5, and REQ 7, as shown in the Product Backlog in the Sprint 2 Report. During the demo, we received feedback on our implementation of these requirements.

REQ 1: Web application must be secure and protect confidentiality of a user's ImHungry data

- **Feedback**
    - We do not need anything related to SQL injection attacks
    - Need to implement HTTPS
    - Login and Password
        - Accessing any of the pages within I'm Hungry must be protected with user login
        - Login page should display before any of the other application's pages
        - User should not be able to access any page except login page before logging in
        - No guest feature
        - Password must be hashed
    - Pre-written user info into the database and proceed to log-in with those credentials (pre-configured users is okay)
    - ~~User should need to register an account if they do not already have one~~
        - ~~User can register from the login page~~
    - All data should persist for each specific user
        - List data, search history, etc
- **Result**
    - Use HTTPS
    - Create Login Page
        - Username
        - Password
            - Hash the password
    - Create pre-configured login information

        - User can use the pre-configured login info to enter the application

REQ 2: Maintain information beyond just a single session

- **Feedback**:
    - Maintain the information for the grocery lists
- **Results**:
    - Once the grocery list functionality works, add this data to the database

REQ 3: Allow for pagination of results returned by the search.

- **Feedback**:
    - Backend tests needed
    - Only 5 page numbers should appear at the bottom
        - If you have more than 5 pages, only display 5
    - “Previous” button before the first page
    - “Next” button after the last page
    - Sliding window of pagination
        - When you click on a page, that page number should be in the middle and the subsequent other 4 pages that are displayed will adjust
        - i.e. 10 pages total, click on Page 4, display should be:
            - prev, 2, 3, 4, 5, 6, next
    - Constraints
        - Floor should be Page 1
        - Ceiling should be the Max Page
        - i.e. When you are on Page 2, it should not be in the middle
    - Indicate which page is clicked on by some sort of highlight to that page number
    - Empty tables should not appear at the end if there are less than 5 results per page
- **Results**:
    - Add a “Previous” and “Next” button
    - Highlight the Page that is clicked
    - Implement the sliding window
    - Create backend tests

REQ 4: View results of prior searches by clicking on a quick access list that shows prior search terms.

- **Feedback**:
    - Each Quick Access card should contain
        - Image Collage
        - Title of the Search
    - Clicking on each card should produce the information for that result (miles, number of search terms, etc.)
    - When a term has been clicked, push the new one to the front of the Quick Access

    - The newest search requested from the user on the Search Page should not appear in Quick Access
    - UI needs to be fixed to match the rest of the UI
- **Results**:
    - Update the Quick Access cards to include the image collage and Search term/title
    - Fix how the Quick Access is updated according to what is considered a recent search
    - Update the UI of the horizontal scroll

REQ 5: User interfaces must look modern and be attractive.

- **Feedback**:
    - “Edit” button on List Management Pages should be the same layout as all of the other buttons
    - Button Menu should be centered
    - “Select List” text on Dropdown Menu should be centered
    - Quick Access UI needs to be updated
    - List Management Page has results going off the page, which breaks the color of the page
- **Results**:
    - Update the “Edit” button
    - Center the buttons and button text
    - Fix the Quick Access UI
    - Fix the bug on the List Management Page

REQ 7: Reorder any of the three predetermined lists.

- **Feedback**:
    - Check and update the order on the backend database
    - Persist the reordering
- **Results**:
    - Update the database to change the order when the user reorders the results on the list so that the order can persist

The tests that we wrote for REQ 6 and REQ 8 did not pass and as a result, we were not able to receive feedback on these requirements. Moving forward with Sprint 3, we will implement all of the new feedback that was received as well as implement REQ 6 and REQ 8 with our own interpretation. We will also have more extensive and comprehensive testing.