Sprint Planning #2

Product Backlog

1. Web application must be secure and protect confidentiality of a user's ImHungry data.
2. Maintain information beyond just a single session.
3. Allow for pagination of results returned by the search.
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.
6. Keep track of a grocery list for selected recipes.
7. Reorder any of the three predetermined lists.
8. Set the radius of the restaurant search.

Sprint Planning

Date: April 3, 2019
Time: 11:00 PM - 12:30 AM
Location: Lobby of USC Village Building #9
Participants: Will Borie, Alex Colello, Connor Buckley, Emily Jin, Julia Wada

Link to Github Commit for the Sprint Backlog:
https://github.com/AlexColello/CS310GroupC/commit/b7cda29be8fd150b3db6a9fec58c7cf4936b87db

Discussion Notes

Decide what parts of the project to refactor.

We decided on four parts of the codebase to refactor in this sprint. First, we will change all of the passing of information from the back-end to the front-end via requests to be via session storage. The rationale for making this change is that there were duplicate pieces of information being sent to and from the front-end, with some information using requests and others using sessions. This decreased the design quality of the project because the current implementation is confusing and has low readability. Second, we will implement a cache for search results to make the results page load faster on repeated search. This will allow our tests to run faster and will thus increase the design quality by optimizing the implementation. Third, we will reduce and compact the amount of code we have in our ListManagementServlet. Our servlet currently tracks which page each result came from, which is unnecessary information, so this will be removed moving forward. Finally, we will consolidate repeated code in the backend into functions to increase readability by decreasing the total source lines of code.

Breakdown features into subtasks and assign tasks to team members.

Julia, Will - frontend
Alex, Connor - backend
Emily - SQL

Determine and Justify features that should be completed for Sprint 2.

Since the Sprint 2 Demo will be our last chance to get feedback and understand what the requirements will be, we have decided to put all of the features on the product backlog that we have not begun implementing into the Sprint 2 backlog. We believe that, given the progress we made in Sprint 1 in setting up the database for implementing future features, we will be able to have enough to demo for each of the features listed in the product backlog by the end of Sprint 2. Bugs and small changes were set as individual tasks while larger features were decomposed into UI and back-end tasks which were assigned to the appropriate team members.

Feature list and task breakdown

1. (1) Web application must be secure and protect confidentiality of a user's ImHungry data
    - 1.1. Login backend - Connor
        - ~~1.1.1. Verify user logins with passwords~~
        - ~~1.1.2. Encrypt user passwords~~
    - ~~1.2. Login frontend - Julia, Will~~
        - ~~1.2.1. Add in relevant buttons~~
        - ~~1.2.2. Create a new page as needed~~
    - ~~1.3. Signup backend - Connor~~
        - ~~1.3.1. Verify user signup~~
        - ~~1.3.2. Encrypt user passwords~~
    - 1.4. Add https functionality
    - 1.5. Remove guest functionality
    - 1.6. Fix login/logout buttons
2. (2) Maintain information beyond just a single session - Emily
    - 2.1. Set-up SQL tables (1, 2, 4)
        - ~~2.1.1. Username & password table~~
        - ~~2.1.2. Search history table~~
        - 2.1.3. Lists table
    - 2.2. Grocery Lists
    - 2.3. Three pre-defined lists
3. (3) Allow for pagination of results returned by the search - Will, Alex
    - ~~3.1. Look into this and find the most efficient way of doing pagination~~
        - 3.1.1. Add the highlighted current page
        - 3.1.2. Add tests for clicking on pagination buttons
        - 3.1.3. Change results per page to 10
        - 3.1.4. Set limit on how many pages are shown in pagination bar (like google)
        - 3.1.5. Change buttons to have “next” and “previous” text

4. (4) View results of prior searches by clicking on a quick access list that shows prior search terms - Connor, Emily
    - ~~4.1. Store search query, rerun search when it's clicked on~~
    - ~~4.2. Populate this quick access list with data from the SQL table of searches~~
    - 4.3. ??? New page
    - 4.4. Add functionality to the bottom of the page
    - 4.5. Create items to have collage and image of itself
5. (5) User interfaces must look modern and be attractive - Julia, Will
    - ~~5.1. Review over Alex's CSS and make simple changes as needed~~
    - ~~5.2. Mock-ups of new pages~~
    - 5.3. Fix the List management UI
    - 5.4. New Grocery List UI??? And back-end
    - 5.5. Fix Cucumber tests to test clicking on things such as quick access List
    - 5.6. Fix resizing of the stars
6. (8) Set the radius of the restaurant search
    - 6.1. Convert from meters to miles
    - 6.2. Add an error message in the case that no restaurants can be found within the specified radius
    - 6.3. A user should not be able to search without a radius
7. (6) Keep track of a grocery list for selected recipes
    - 7.1. Button for each item in recipe ingredients
    - 7.2. Add new Grocery List Page
    - 7.3. Add "remove" button for each item in grocery list
    - 7.4. Add link to grocery list (icon or words)
    - 7.5. Create a new data type for storing and modifying in the database and create servlet (modify database maybe)
8. (7) Reorder any of the three predetermined lists.
    - 8.1. Add buttons (up and down arrow) to reorder lists
    - 8.2. Create backend to handle reordering of list
9. Best Practices (Refactoring)
    - 9.1. Changing/Cleaning Session Storage Usage
    - 9.2. Search Caching
    - 9.3. Clean up the List Management Servlet
    - 9.4. Move repeated or ugly code on backend into functions
        - 9.4.1. Encryption for password
        - 9.4.2. JSON creation in servlets (use GSON instead)
        - 9.4.3. Moving Favorites to top of results and removing Do Not Show results