Sprint Planning #1

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: March 22, 2019
Time: 5:00 - 7:00 PM
Location: Lobby of USC Fertitta Hall
Participants: Will Borie, Alex Colello, Connor Buckley, Emily Jin, Julia Wada

Link to Github documentation:
https://github.com/AlexColello/CS310GroupC/tree/master/Documentation

Discussion Notes

Details for each of the required tasks, with 1 being easiest to implement and 5 being the most difficult.

1. [2] Web application must be secure and protect confidentiality of a user's ImHungry data
    - Have a login/signup function so each user can login and save data
2. [2] Maintain information beyond just a single session
    - Create a SQL table to add data to each user
3. [4] Allow for pagination of results returned by the search.
    - Bootstrap could maybe do this??? Need to get more data
    - How much more data should we get?
4. [2] View results of prior searches by clicking on a quick access list that shows prior search terms.
    - Search history in SQL table
    - ****User searches should not be saved unless they are logged in?*
5. [2] User interfaces must look modern and be attractive
    - Figure this out as we go? Probably just bootstrap stuff
    - ****What does this mean? Limits (esp. about collage & design)*
6. [2] Keep track of a grocery list for selected recipes

- Need new button, easy web scraping stuff

7. [3] Reorder any of the three predetermined lists.
    - React? Connecting to backend, verify on SQL
8. [1] Set the radius of the restaurant search
    - Really easy, add an extra request to the Yelp API

Decide base repository, and start a new repository.
For this project, we will be using Alex's Github repository with Julia's CSS.

Set up working environments for local devices.
Code individually on local device. If needed, continue using the VM to demo, and deploy through Maven and Tomcat.

Github procedures for branching, merging, etc.
Each person works on their unique branch. While pair programming, push it and link the commit to the branch. Merge whenever we have a stable feature, and discuss at the next scrum.

Breakdown features into subtasks and assign tasks to team members.
Julia, Will - frontend
Alex, Connor - backend
Emily - SQL

Determine features that should be completed for Sprint 1.

In order to most effectively incorporate all additional features, we first chose to implement the features that were the foundation for other features. For example, the MySQL tables were a high priority due to the fact that the rest of the application would be dependent on this for data persistence. As such, Sprint 1 focuses heavily upon the creation of an accurate and stable database in order to satisfy requirements #1, 2, and 4. Additionally, we also chose to work on tasks that were ambiguous or difficult to implement. For example, pagination was the task we deemed to be the most difficult because of the different possible interpretations of what could be expected, as well as the multiple different ways of implementing said feature. This is also the most complex feature since it will require frontend and backend collaboration. We also included the UI in this sprint because we needed enough time to review over our designs with the core product owners. An updated UI is also necessary in order to access the new features, such as displaying quick access links for the search history and the login/signup features.

Once we determined the features to be accomplished for sprint 1, we then decomposed the features into tasks based on each member's preferences. Alex, Emily, and Connor had worked on the backend of the application for Project 1, while Will and Julia had more experience with the frontend. As such, we broke down the tasks so that each person was working on tasks they felt most comfortable with. Although this task list is not a complete breakdown of all the necessary tasks needed for each feature, we plan on adding in new tasks as needed.

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 and salt user passwords
    1.2. Login frontend - Julia, Will
        1.2.1. Add in relevant buttons
        1.2.2. Create a new page as needed
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
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
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
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
6. (8) Set the radius of the restaurant search - Alex