

Sprint Planning #3

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.

Sprint Planning

Date: April 17, 2019

Time: 10:30 PM - 11:30 PM

Location: USC Village Building #9

Participants: Will Borie, Alex Colello, Connor Buckley, Emily Jin, Julia Wada

Link to selfie:

<https://github.com/AlexColello/CS310GroupC/blob/master/310-Documentation/Sprint3/sprint3planning.JPG>

Link to Github Commit for the Sprint Backlog:

Discussion Notes

Decide what parts of the project to refactor.

Refactoring #1: User functions in the database need to be fixed because all other database functions behave differently. Inserting an item should return the index of the last item that was inserted (alternatively, the ID number for each inserted row), and searching for a user should return their ID number instead of the ResultSet. However, in the case of the Users table, this is the only function that returns a ResultSet when searching for a user, and returns void when inserting into the database. In order to maintain consistency and expected behavior, these two functions will be updated to conform to the rest of the functions.

Refactoring #2: Our List Management functions currently have a lot of repeated code between restaurants and recipes. For this refactoring, we will reduce our total lines of code by eliminating repeated code while still maintaining functionality. We plan to create templated functions that can handle restaurants and recipes in one concise function to drastically reduce the lines of code, which ultimately should be able to cut in half and increase readability.

Refactoring #3: User data should not be stored in the Javascript session storage, so for this Sprint we will move all our variables to the Java session instead. Additionally, all user data will be cleared when the user logs out. The reason for this is because Javascript session storage can be modified from the console; in order to increase the security of our application, sensitive data will be moved into Java sessions instead.

Refactoring #4: Our current Cucumber tests are really long and convoluted, so for this Sprint we will increase the readability and understandability of our test cases. Currently the feature files are very difficult to read because the majority of the keywords that are used in the scenarios are Then and And, which do not correctly describe what actions are being performed in the test. Increasing the readability will make the tests easier to debug, more useful, and will improve the general quality of the code.

Breakdown features into subtasks and assign tasks to team members.

Julia, Will - frontend

Alex, Connor - backend

Emily - SQL

Subtasks are shown and assigned on the Trello Sprint backlog.

Determine and justify features that should be completed for Sprint 3.

Other than the tasks that were created for refactoring, the features that need to be completed are the ones left that we are still missing points for. Reordering the list and radius functionality are no longer on our sprint backlog as we met all the requirements for the feature. Most of the tasks that need to be completed include those related to the search history and grocery list features. We elicited many details such as having the search history be a horizontal list and that each item needs to have checkmark in the grocery list. One of the other major tasks to undertake is to create the https functionality to finish the security feature. Other changes to the UI and pagination are more minor.

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**
 - 2.2.1. **Make sure that the grocery list checks persist**
 - 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~~
 - 4.6. **Change implementation so recent searches don't show current search**
 - 4.7. **Make horizontal list that is scrollable**
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~~
 - 5.7. **Make sure that when there is nothing in a list the headers do not appear**
 - 5.8. **Chang printable version to get rid of button container outline**
 - 5.9. **Display check if ingredient is checked**
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**
 - 7.3.1. **(Fix this button to be removed when item in grocery list is removed)**

- ~~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)~~

7.6. Duplicate items in the grocery list should be merged together

7.7. Items on the grocery list page should be able to be “checked” and “unchecked”

7.7.1. The “checked” attribute should persist

- ~~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~~

9.5. Sprint 3 Refactoring

- 9.5.1. Users table - needs to NOT return a Result Set (and other database cleaning)**
- 9.5.2. Cucumber tests**
- 9.5.3. Clean up JavaScript session storage -> Move to Java Session, also clear on logout**
- 9.5.4. Refactor List Management, move repeated code to functions**