Team 4

Sprint 1 Planning

10/03/2016

Intelligent Search

Sidhant Chitkara, Alexander Meyer, Apoorva Parmar,
Yash Pujara, James Shao, Michael Vieck

SPRINT OVERVIEW

The goal for this sprint is to create a working early prototype. We have selected a few [3] search criteria that we hope to implement into the lexical search. We will build our class structure off of these functionalities in order to gauge the difficulty in implementing future searches. We also aim to have user accounts and preferences done by the end of this split. User preferences and favorite information will carry over into most searches and will be used intelligently by the data processing without prompting from the user.

There are a few challenges for this sprint. The first is creating a working lexical search that does not get confused between different search criterias. The lexical search is a main feature of the project and will become more complicated as functionality is added. This sprint will challenge us to create a working prototype and base for future sprints. Another challenge will be having the Data Processing class intelligently use preference information to tailor search result data for the user. This functionality should occur in most [Purdue Dining] searches and will ideally create a streamlined and simplified experience for the user.

Scrums will be held during the original class times - Tuesday and Thursday starting at 3:00pm. The meetings will ideally be 10-15 minutes and led by James.

CURRENT SPRINT DETAILS

*Note: Identical tasks are listed again if they are relevant to multiple user stories.

Task Team	Team Members	Total Hours
Front End	Apoorva, Michael	20
Tokenizer	Yash, Sidhant	20
Services	Alex	9
Data Processing	James, Alex	12
Database	James	7.5
*Note: Tasks are assigned to teams and members can choose what task they are working on. **Note: Team Members are free to choose their Task Team and can choose to change teams.		58.5

Team 4

Sprint 1 Planning

10/03/2016

1. **When users search for specific dining court menus on a specific day then return the dining court menu information and item cards.**

Team	Hours	Description
Front End	2	Have a functioning search bar and pass to Tokenizer
Front End	3	Display results on item cards
Tokenizer	1	List all possible search permutations for this specific search
Tokenizer	9	Pick out keywords needed to build query and write C++ logic
Services	1	Create function that checks the date and determines what queries / API calls are needed. Pass the data to Data Processing.
Database	1	Create tables for items and dining courts.
Database	2.5	Create script that updates menu table every day
Data Processing	2	Create functions that can convert the returned API / database data to JSON format and pass to the front end.

Acceptance Criteria: Successful search of *"What is at [DINING COURT] on [DATE]"* or *"[DINING COURT] Menu for [DATE]"*. The [Date] should be accepted as Month/Date or Month Date or Today/Tomorrow.

2. **Users would like to search for a specific item in a specific dining court (for today) then return Y/N.**

Team	Hours	Description
Tokenizer	1	List all possible search permutations for this specific search
Tokenizer	4	Pick out keywords needed to build query and write C++ logic
Data Processing	2	Create functions that can convert the returned API / database data to JSON format and pass to the front end.
Database	1	Create tables for items and dining courts.
Database	2.5	Create script that updates menu table every day
Services	1	Create function that checks the date and determines what

		queries / API calls are needed. Pass the data to Data Processing.

Acceptance Criteria: Successful search of *"Does [DINING COURT] have [ITEM NAME]"* or *"Is [ITEM NAME] at [DINING COURT]?"*. The query will return whether this item is being served at the dining court. This query will only work for TODAY this sprint.

3. **Users would like to search for a specific item across all dining courts (for today) then return what dining courts have that item**

Team	Hours	Description
Tokenizer	1	List all possible search permutations for this specific search
Tokenizer	4	Pick out keywords needed to build query and write C++ logic
Data Processing	2	Create functions that can convert the returned API / database data to JSON format and pass to the front end.
Database	1	Create tables for items and dining courts.
Database	2.5	Create script that updates menu table every day
Services	1	Create function that checks the date and determines what queries / API calls are needed. Pass the data to Data Processing.

Acceptance Criteria: Successful search of *"Where can I find [ITEM NAME]"* or *"Is [ITEM NAME] being served?"*. This query will return a list of Dining courts that the food is being served at today. This query will only work for TODAY this sprint.

4. **Users should be able to create an account then log in with the credentials used.**

Team	Hours	Description
Front End	4.5	Have a functioning log-in / account creation page and pass the information to Services
Services	1	Create function that checks if user exists and then creates database query to create user and return success / fail to front end
Database	1	Create table for users

Acceptance Criteria: Does the login page work. Can users create an account and then log in. Is the user's information, such as saved preferences saved.

5. **Users would like to be able to log in then go to the landing page.**

Team	Hours	Description
Front-End	.5	Take user from login page to landing page on the click of a button.
Services	1	Create a function that checks if user exists and correct credentials are given for the process
Data Processing	2	Creates a function that calls to the database to retrieve the user’s saved preferences

Acceptance Criteria: On User login, the login page redirects to the landing page

6. **Users would like to be able to log out then go to the log-in page**

Team	Hours	Description
Front-End	3	Add a logout button to the right of the search bar or in the nav menu and route user from the current page.

Acceptance Criteria: On User logout, the current page redirects to the login page

7. **Users would like to be able to log in and out with their Facebook and then go to the respective pages (landing and log-in).**

Team	Hours	Description
Front-End	.5	Add facebook button and libraries.
Front-End	1	Setup Facebook O-Authentication and route user to landing page.
Services	1	Insert User into user table as Facebook user

Acceptance Criteria: Is Facebook login/logout integrated into native login and logout? Test by having users create an account using facebook.

8. **Users would like to be able to change user settings such as their password, name, and dietary restrictions and then have the results update immediately**

Team	Hours	Description
Front-End	4	Create a settings page that the user can change preferences in.
Database	1	Create a table for user preferences
Services	2	Create a function that allows the users to change different saved preferences.

Acceptance Criteria: Does the preferences page save changes? Change user information and reload - is the new information displayed?

9. **Users would like to be able to click on item cards that then flip over to display more detailed nutrition and dietary information.**

Team	Hours	Description
Front-End	1	Setup view text and ui updating callbacks.
Front-End	.5	Add smooth animations to the cards to add user interaction depth.
Database	1	Create table for items and item information

Acceptance Criteria: On search results, can item cards be clicked. Does the card flip over and display more detailed item information?

10. **Users would like to be able to set favorited items by “pinning” the item or dining court card and then it would display under favorites**

Team	Hours	Description
Front-End	.5	Add a favorites button to all the cards.
Front-End	.5	Add filter button that displays only the favorite items.
Services	1	Create a Function that allows user to add certain saved material to database.

Database	1	Create table for user preferences

Acceptance Criteria: Can the User click on the item / dining court and add it to their favorites. Check if the item / dining court shows under the favorites section.

11. **Users would like to be able to set user food preferences (allergies, dietary restrictions) under User Settings and then food item cards that match those preferences will not be displayed.**

Team	Hours	Description
Front-End	1	Hide cards on payload data returned.
Services	1	Make sure to check Database for saved preferences and then limit sql query based on these saved preferences.
Data Processing	1	Create functions that can convert the returned API / database data to JSON format and pass to the front end.
Database	1	Create table for user preferences
Data Processing	3	Create function that takes item / menu information and marks items that match the user's preferences. Marked data is then sent to front end where it can be displayed or hidden

Acceptance Criteria: Does the preferences page persist changes? Search for menu that contains items that should be hidden based on allergies and see if item card is shown or not.

REMAINING BACKLOG

Backlog ID	Functional Requirements
1	As a user, I would like to be able to search for specific foods.
2	As a user, I would like to be able to search for specific cuisines (food styles).
3	As a user, I would like to sort food items based on popularity, meal type and cuisine type.
4	As a user, I would like to be able to track calories consumed.

5	As a user, I would like to be able to find foods around a certain calorie level.
6	As a user, I would like to be able to use location service to find the closest dining court.
7	As a user, I would like to have notifications sent to my phone about my favorite foods being served.
8	As a user, I would like to have a profile image.
9	As a user, I would like to be able to find the nearest bus.
10	As a user, I would like to be able to find the shortest time to my destination.
11	As a user, I would like to find the estimate bus time.
12	As a user, I would like to have the ability to search routes.
13	As a user, I would like to change my password if forgotten.

Non- Functional

Backlog ID	Non-Functional Requirements
1	As a user, I would like to have a fast response time.
2	As a system admin, I would like to be able to scale the project up, if need be.
3	As a user, I would like the accuracy of my searches to be high.
4	As a user, I would like to have access to this on Mobile.
5	As a user, I would like to have access to it on Web.
6	As a developer, I would like to be able to add more APIs in the future.

7	As a developer, I want to be able to handle a large volume of traffic at any given point in time.
8	As a user, I want a fast and secure way to log in.