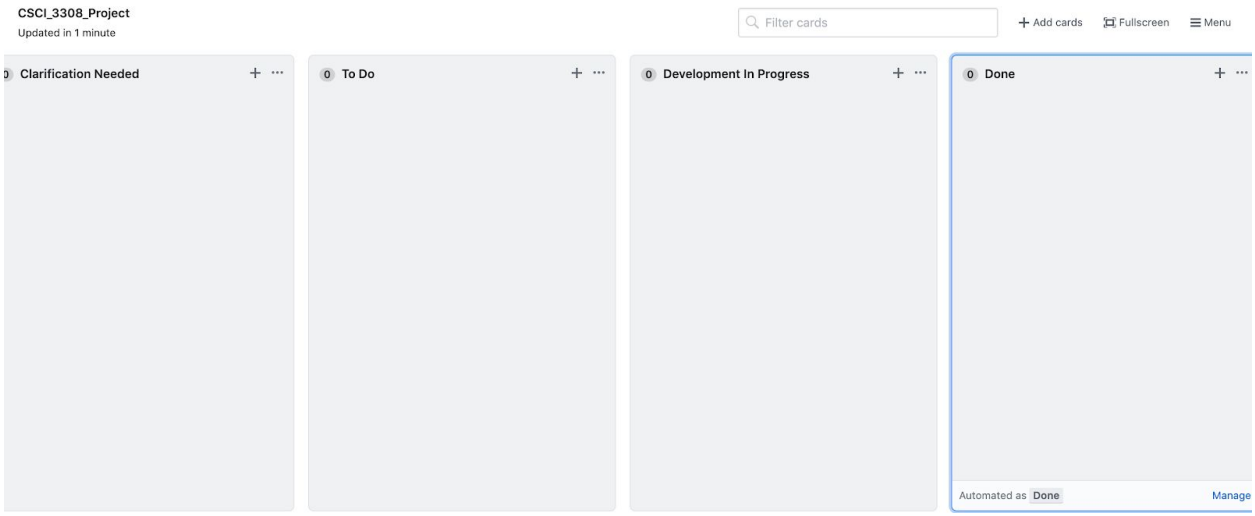
CSCI 3308 Project Milestone 2
Team Number: 205-3
Team Name: NULL
Team Members: Erik Pohle, Matthew Wu, Abdulaziz Alabdulrazzaq, Anand Zupa, Bradley Spangler
Application Name: Hunger: NULL

Project Plan

Project Management Tool:

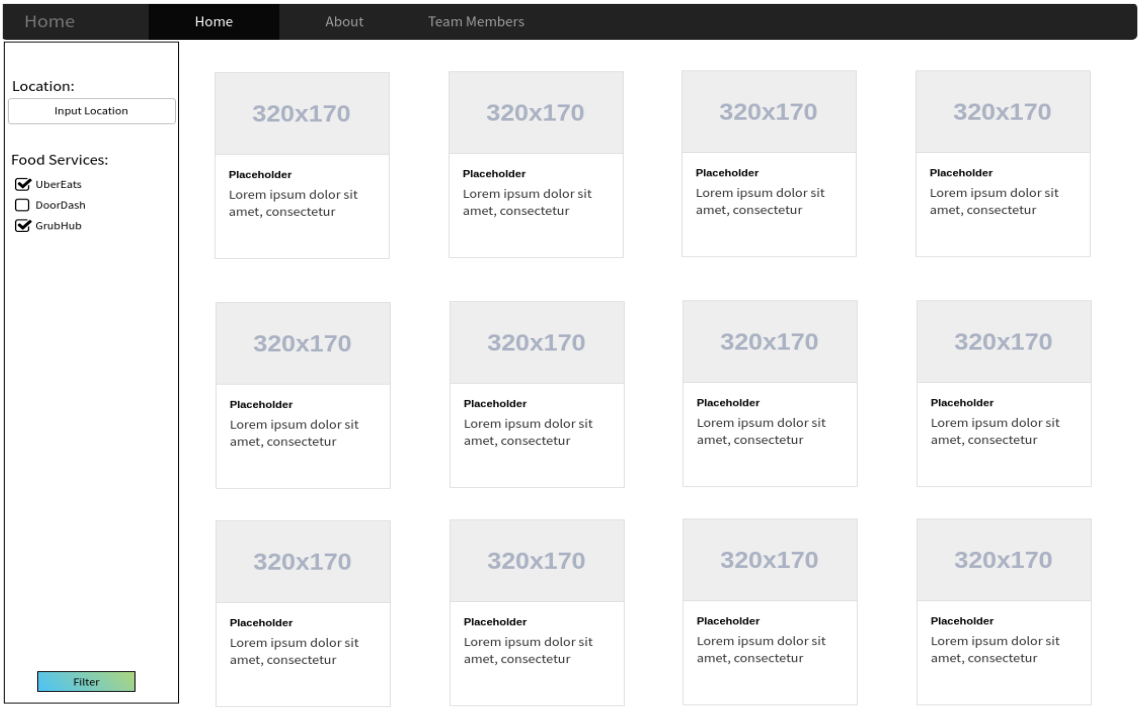
- GitHub Boards/Kanban Board :



- Four Groups: Clarification Needed, To Do, Development In Progress, Done
    - Clarification Needed: When a feature has run into a roadblock or we are waiting for some more information regarding a feature, then the feature card gets moved to Clarification Needed.
    - To Do: When a feature has just been created and no development is being done on it, then the feature card waits in To Do.
    - Development In Progress: When a feature has been picked up by a developer and is being worked on, the feature is moved to Development In Progress.
    - When a feature is complete, then the feature card is moved to Done.

10/20/19 - 10/27/19 (Assignee: Matthew Wu):

- Framework of app should be complete by End Of Week. HTML/CSS
- At least be able to have one major food delivery service (DoorDash, GrubHub, UberEats, etc) properly displayed.



- Sprint 1:
    - Goal: Basic rough outline of the HTML/CSS
        - Requirements:
            - Navigation Bar at top
            - Three different pages (Home, About, Team Members)
            - Left-Hand Filters Grid (Apply all filters)
            - Main content will be the stores along with what they serve (If stores overlap between Food Services, map them into the same content box)

10/27/2019 - 11/10/2019 (Assignee: Erik Pohle):

- Have the website be properly populated with options from GrubHub/DoorDash/UberEats, etc.
    - Sub Requisites:
        - Web Scraper if no API is found
            - Selenium is a good option

            - Once data is gathered, need a database
    - Sprint 2:
        - Goal: Be able to gather data from the Food Services websites
            - Requirements:
                - Find an API to gather data
                    - If no API, create/find a web scraper
                - Be able to gather data using API/web scraper

10/27/2019 - 11/10/2019 (Assignee: Abdulaziz):

- Database is created
- Database can properly store data
- Database can properly distribute data
- Sprint 3:
    - Goal: Be able to store data from the Food Services websites in a database
        - Requirements:
            - Have a database that can be accessed by the website
            - Database can properly store the data found from the API/web scraper

11/10/2019 - 11/20/2019 (Assignee: Anand Zupa):

- Create Account Feature
- Create Cart Feature
- Sprint 4:
    - Goal: Be able to have users, remember users, and view carts for users
        - Requirements:
            - Users can create a new account
                - User Create Account Page
            - Users can login to their account
                - User Login Page
            - Users can view their carts
                - User Cart Page

11/20/2019 - 11/30/2019 (Assignee: Erik Pohle):

- View Order History
- Add Items to Cart
- Checkout/Execute Order
- Sprint 5:
    - Goal: Users can view order history, add items to cart, and checkout
        - Requirements:

- Users are able to view order history
    - Order history page
- Users can add items to cart
    - Prompt “Success” when an item is added
- Users can checkout
    - Redirect to the website that they selected after viewing all options.
        - Have the redirect include their current cart

12/1/2019:

- Final project due Monday, December 2nd

Project Features

1. Access to all available food delivery services
    - Hunger is able to integrate with major food delivery services. Users can browse restaurants’ menus and make purchases from each delivery service from one app.
2. Create account
    - Users can create accounts to store their information for future purchases.
3. View order history
    - Users can browse past purchases.
4. Add items to cart
    - User can create orders in our app
5. Checkout/execute order
    - Users can execute orders from food delivery services though Hunger
6. Apply filters to search options
    - Users can view and filter food options from the available delivery services in a search function.

Requirement 1

Feature: Access to all available food delivery services

- Hunger is able to integrate with major food delivery services. Users can browse restaurants’ menus and make purchases from each delivery service from one app.

Functional Requirements:

- Users can see the different restaurants and which apps deliver to which restaurants
- Users can login to their account
- Users can view their order history
- Users can apply filters and customize what options they see

Non-functional Requirements:

- Either integrate with the API of different delivery services or web scrape and populate a database
- The web scraper/api call operates within 0.5 seconds - 2 seconds depending on user latency
- The redirect to the food service when checking out should include the items the user ordered, their location, and name
- The web scraper needs to be general, meaning that one webscraper should be able to scrape the elements from all the different food service sites

As a user, I want to be able to see all my options for food delivery in one place, so that I don't have to switch between apps and websites to find the restaurants I want.

Acceptance criteria: The user story is complete when I can see the various restaurants and services that deliver.

Requirement 2

Feature: Create account

- Users can create accounts to store their information for future purchases.

Functional Requirements:

- Users can see their information such as the name they gave, the address/location, etc.
- Users can edit this information and keep it updated

Non-functional Requirements:

- Password protected users
- Password recovery via email or phone number
- Ways to hold the information given by the user (user object?)

As a user, I want to be able to save my information so that I don't have to reinput it every time I want to use this app.

Acceptance criteria: The user story is complete when I am able to save my information and can access it easily.

Requirement 3

Feature: View order history

- Users can browse past purchases
- Users can delete past purchases from their history
- Users can re-order a previously ordered meal

Functional Requirements:

- A list of most recent orders
- Ranking of restaurants that have the most orders per user
- Option to delete a past purchase
- Option to order a past purchase again

Non-functional Requirements:

- Tracking redirects off website per user
- Sorting by most ordered and most recent orders
- Way to delete entries

As a user I want to be able to see my previous orders so I can easily see which restaurants I like and want to order from again.

Acceptance criteria: The user story is complete when orders are stored and easily accessible to the user.

Requirement 4

Feature: Add items to cart

- User can create orders in our app

Functional Requirements:

- Having a cart
- Showing restaurant menus
- Storing location of user

Non-functional Requirements:

- Way to store and show what items are in the cart
- Cart stored in such a way that it can be transferred to selected delivery service

As a user I want to be able to choose what foods I'd like from the restaurant I've chosen so I can put my order through.

Acceptance criteria: The user story is complete when I am able to add menu items from a selected restaurant to my cart.

Requirement 5

Feature: Checkout/execute order

- Users can execute orders from food delivery services though Hunger

Functional Requirements:

- When ready to checkout, the user is redirected to selected delivery service

Non-functional Requirements:

- Orders are executed from Hunger and connect to the chosen delivery service
-

As a user, I would like to place food orders because it is the primary reason I use Hunger.

Acceptance criteria: The user story is complete when I have successfully placed my order and the food is on the way.

Requirement 6

Feature: Apply filters to search options

- Users can view and filter food options from the available delivery services in a search function.

Functional Requirements:

- Select food delivery service
- Select price range

Non-functional Requirements:

- Sql queries, eg: where service = 'doordash'
- Possible join tables
- If a search filter is applied, hide or show appropriate results

As a user, I would like to specify search preferences so I can find what I am looking for more efficiently.

Acceptance criteria: The user story is complete when I apply query preferences to help narrow down my search.