



## Group 4 - Green Beans

Prototype & Testing Protocol

Tiffany Lu, Justin Le, Yohan Lee, Chantalle Matro, Ella Gabrielle Uchtmann

May 1st, 2024

## INTERACTIVE PROTOTYPE VS STATIC WIREFRAMES

We will create detailed static wireframes that are relatively high fidelity in appearance using Figma before we all begin prioritizing coding. This will allow us to work out design decisions and specifications before we jump into the coding. We believe that creating these detailed static wireframes will make the coding process easier. Our primary developer has already begun minimal coding of our prototype to set us up for success while the design team is developing a more finalized figma prototype. If we find while we are coding that a design decision we made is not plausible for the time and resources that we have available for this project, we will iterate on the original design to make a more plausible solution within the Figma prototype. Once we have verified with the team that the new interaction is more plausible we will code that version of the feature. We are deciding not to go all the way to creating an interactive prototype because we feel as though that would take an unnecessary amount of time given the scope of this project. We are instead opting for adding arrows, comments, and additional frames into our Figma document to finalize the flow and interactivity of our designs.

## ACCEPTANCE TESTS & PROCESS

**Stores/Events Search Feature Test:** A user searches for a store or event within the search bar at the top of the page of the Stores or Events Page.

- **Successful Test Criteria:** Once a user searches for a store or event within the coordinating page's search bar, the web application displays the store that matches the user's input from the search bar. Additionally, the web application will display "result not found" if the user input is invalid (meaning their input types are not valid or their input does not match any data within our dataset of stores/events).
- **Failed Test Criteria:** Once a user searches for a store or event within the coordinating page's search bar, the web application does not display stores/event that match the user's input even though it is valid (meaning that their input matches data within our dataset and their input type is valid). Additionally, the web application displays content that is not relevant to the user's input. Moreover, the web application does not display "results not found" if the user's input is invalid. Or the web application displays "results not found" even though their user input within the search bar is valid.

**Stores/Events Filter Feature Test:** Users will select drop down filters within the Stores and/or Events Pages to filter through different criteria to match their preferences.

- **Successful Test Criteria:** Once a user has selected a filter option, the web application will display all the stores/events that match the selected filters. If no

stores/events match the filters selected “results not found” will be displayed on the page. If a user does not select a filter, all stores will render on the page.

- Failed Test Criteria: Once a user has selected a filter option, the web application does not display all the stores/events that match the filter. The web application could also display “results not found” even though the filter is valid. Additionally, the web application does not display “result not found” if the filter input is invalid.

**Quiz Feature Test:** Users take a quiz that asks them questions about their personality and preferences to match them with stores. Once the user finishes the quiz, they will be directed to a list of stores that match their personality and preferences.

- Successful Test Criteria: The user will take the quiz successfully by answering question by question and answering all questions. At the end, their corresponding result will be displayed along with a list of stores that match their results.
- Failed Test Criteria: The user will take the test successfully. Once the test is complete and their results render, unrelated results to their answers display. Additionally a list of stores not directly correlated with their results will render.

## ACCEPTANCE TESTS LIMITATIONS

- Stores/Events Search Feature
  - In-house vs. real-world network conditions may differ
  - User queries may differ from testing vs. actual usage
    - Ex. A real user attempts to search for “vintage market with food trucks,” and since this query is more complex and specific, there may be no results.
- Stores/Events Filter
  - In-house vs. real-world network conditions may differ
  - In-house test may not account for the combinations of filter options possible
    - Ex. A real user attempts to filter for a specific date range that conflicts with their chosen location for events, which was not a part of the in-house testing environment, so this may result in unexpected behaviors.
- Interactive Cards
  - The testing environment might not accurately reflect the varied conditions under which users will interact with the cards like different devices, internet speeds, distractions, etc.
    - Ex. A real user may experience delays or inconsistencies in loading the content of the cards and the responsiveness of the interactions due to their slow network speed.

- **Quiz**

- In-house testing may lack the ability to capture nuanced user feedback during the quiz-taking process, such as hesitation, confusion, or frustration
  - Ex. A real user may encounter unclear questions, but due to the limited ability to capture user feedback, we may not be able to address their concerns promptly, potentially leading users to exit the quiz.
- The team's testing environment might not represent the full spectrum of users who will interact with the quiz, including individuals with diverse backgrounds, knowledge levels, and accessibility needs
- The in-house testing environment may not cover the range of devices and browsers that users will use to access the quiz, potentially leading to compatibility issues that are not identified during testing

## **CONDUCTING USER TESTING**

We will conduct user testing by first recruiting users. Due to the time constraints of this project, these users will most likely be users already within the team's social circles. As we are college students with a target audience of college students, this is a plausible way to recruit users for testing purposes. We will reach out to potential users and ask to set up a time to meet with them.

We will use the website When2Meet to schedule time with potential users. Our goal is to user testing with five users at least once throughout the development of our product. Additionally, we will most likely conduct user testing one-on-one as we have conflicting schedules within our group. This could potentially be a limitation as it would be preferable to have one person from our group lead the testing and have another person write thorough notes so that we can gather meaningful insights and feedback.

Prior to meeting with users we will create a script that includes information about who we are, what we are developing, and tasks to have them perform as we go through the user testing process (see drafted script below). We will create tasks centered around the main functionalities of our web application. Insights from these questions will help us evaluate and iterate our current coded product so that we can make user interaction with our product more smooth and intuitive.

After we each conduct a user test, we will add our key insights into the documents within a shared google drive folder. We can then all have a discussion about how to solve pain points within the design, flow, and interactions of our web application at that point in time.

### **Working User Testing Script**

## Introduction

Hello, thank you for agreeing to user testing with us today. We are an undergraduate student team called Green Beans at the University of Washington. We are working on a project aimed to provide a centralized platform to help students shop more sustainably. Through this user test, we will ask you to perform a series of tasks. We ask you to walk us through your thought process as you work to complete the tasks. We will be writing down notes throughout this process. We strongly encourage you to share your thoughts openly and honestly. There are no right or wrong answers. Your feedback will play a pivotal role in shaping the direction of our project.

## Questions/Tasks:

1. From the Home Page, find stores that have the lowest cost, click “more details” to read more about the store.
  - a. Pass Scenario: The user successfully navigates to the Stores Page, then filters by the lowest cost, and then clicks the “More Details” button to read more about the store.
  - b. Fail Scenario: The user is unsuccessful at any of the following tasks: navigates to the Stores Page, filters by the lowest cost, clicks the “More Details” button to read more about the store.
2. From the Stores Page, find the last event listed on the Events Page and find the date and time that the event occurs.
  - a. Pass Scenario: The user successfully navigates to the Events Page, then scrolls to the bottom of the event cards, then clicks more details to find the date and time the event occurs.
  - b. Fail Scenario: The user is unsuccessful at any of the following tasks: navigates to the Events Page, scrolls to the bottom of the event cards, clicks more details to find the date and time the event occurs.
3. Once you have read about the last event listed, search for the same event by name.
  - a. Pass Scenario: The user successfully exits out of the “more details” page of the event from the previous question, then scrolls to the top of the Events Page, and then searches the event by name.
  - b. Fail Scenario: The user is unsuccessful at any of the following tasks: exits out of the “more details” page of the event from the previous question, scrolls to the top of the Events Page, searches the event by name
4. Take the quiz. Once you take the quiz, find a store listed that matches your results.
  - a. Pass Scenario: The user successfully navigates to the Quiz Page and clicks ‘Start Quiz’, then answers each question, then completes the quiz, then finds a store that matches their results from the provided list within their results.

- b. Fail Scenario: The user is unsuccessful at any of the following tasks:  
navigates to the Quiz Page, clicks 'Start Quiz', answers each question, completes the quiz, finds a store that matches their results from the provided list within their results.

### Conclusion

Thank you for joining our user testing session today! Your insights are invaluable in refining our project aimed at sustainable shopping for students.

## **PRIORITIZING BUG FIXES**

Our team will prioritize bug fixes by examining the severity of each bug as well as considering the priority of our features which directly correspond to the importance of each feature to the functionality and purpose of our website. We will be referencing our feature prioritization list (P0 - P2).

Prioritizing bug fixing based on the importance of each feature is vital. In the case that a high priority feature (e.g. listing events/stores) is compromised by a bug versus when a low priority feature (e.g. About Page) is compromised by a bug, we need to invest more time in taking action for the higher priority feature to ensure the functionality of our product. The team has decided on fixing bugs based on feature priorities because if we do not, making time for smaller feature's bugs will be a major setback in the development of our application in achieving its main functionality as an informational hub for students to discover sustainable stores and events.

However, this does not mean that we ignore bugs for lower priority features (e.g. About Page). We are emphasizing that we first want to prioritize features that could be detrimental to achieving our application's main functionalities. The severity of bugs is also important to consider. If two bugs are occurring and one of them is not as severe, we will need to consider the time that we have to resolve it, the severity of the bug on the features functionality, as well and the prioritization of the feature. We will consider all of these variables to tackle bugs on a case-by-case basis.

## **RE-TESTING AFTER BUG FIXES**

Our re-testing process will involve several steps to verify the effectiveness of the bug fixes and confirm that the solution meets the required quality standards. First, we will conduct regression testing to ensure that the bug fixes have not inadvertently introduced new issues or affected previously working functionalities. This involves re-running existing test cases and verifying that all previously identified bugs have been resolved without causing any regressions.

Additionally, we will prioritize the re-testing of critical and high-priority bugs to validate that they have been adequately addressed and no longer pose a risk to the functionality or usability of the product. This may involve executing targeted test cases specifically designed to validate the fixes for these issues.

Furthermore, we will perform exploratory testing to uncover any potential edge cases or scenarios that were not covered in the initial testing phase. This allows us to validate the robustness of the bug fixes under varied conditions and user interactions.

Throughout the re-testing process, we will document any new issues discovered through github commits and collaborate closely with the development team to address them promptly. Once all re-testing activities have been completed and the product is deemed stable, we will conduct a final round of validation to ensure that it meets the acceptance criteria and is ready for deployment to production.