**Product Requirements**

|  |  |
| --- | --- |
| **Team** | ☺ |

# Brief problem statement

Indecision is the bane of mankind’s beautiful minds. People do not like to make decisions. Research has shown that when people are given multiple choices of equal validity it is easier for them to not decide at all. Everyone has been in a situation where they want to get food with friends or a significant other and nobody can decide on a place to eat. Or maybe you’ve been in a new town using google or yelp to find a place to eat. Those services are so impersonal. You can’t see the food except for a few users submitted pictures, that are usually a hit or miss, and you become overwhelmed with options.

People are also very good at making impulse decision and committing to it. How often have you seen some new technology or food that wowed you and for the next few days it feels like you need that in your life? This is what our app does, it shows users meals near them. One at a time. This will help people decide on a place to eat that looks good to them. Users will be able to swipe yes or no on meals that are near them. Once they find the meal that hooks them they will be given the details of the restaurant and can then go enjoy their meal.

# System requirements

Our system requires an android device with android 4.2 or later, GPS, and Internet access.

# Users profile

This system is intended for everyone over the age of 16 (legal driving age). Eating is a universal activity and our system helps solve a problem that everyone has.

# Feature requirements (user stories)

List the Priority as 1 (High Priority - Critical) to 3 (Low Priority – Would be nice if we have time)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User Story Name** | **Description** | **Priority** |
| 1 | Date Night | User opens the app, changes the price setting, and presses the button to start searching. | 2 |
| 2 | New to Town | A user is unfamiliar with his surroundings so he enables GPS and opens our app. Next, he selects to start searching and the nearby food shows up. | 1 |
| 3 | Looks good | A user that is searching for new food on our app has found a food he likes. After he swipes to keep the food, it displays the food details, saves it too his likes, and resets the app by emptying the queue | 1 |
| 4 | Again! | A previously satisfied user wants to find the same meal he/she had previously. After opening the app, the user taps on the like icon and then selects the food they previously liked. The app then shows the details of that food | 1 |
| 5 | Shots Shots Shots | Some friends want to go to a place with drinks. After launching the app they chose the settings for “only show if serves alcohol”. Then the user selects to start searching | 3 |
| 6 | Walking Distance | A user wants a restraint close to them. After launching the app they change the distance setting to the desired amount and click to start searching | 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Start Searching | | |
| Created By: | Joshua Tews | Last Updated By: | Joshua Tews |
| Date Created: | 9/21/2016 | Date Last Updated: | 9/21/2016 |

|  |  |
| --- | --- |
| Actors: | Hungry Person (HP) |
| Description: | HP selects to start searching for new foods |
| Preconditions: | 1. User has GPS turned on 2. opened the app |
| Postconditions: | 1. The queue of food items begins appearing |
| Normal Flow: | 1. User selects to start searching 2. The System uses the GPS and Yelp Api to get nearby restaurants 3. The System requests from a list from our database using the restaurant list 4. The System draws the first item in the list and saves the rest of the queue locally |
| Alternative Flows: | If the HP does not have GPS on, the program asks the user to turn gps on.  Or the user could use the settings page to add filtering options. (ie. Price, distance, alchohol) |
| Exceptions: | If the yelp api is down, then the program returns and error. |
| Includes: |  |
| Priority: | Very High |
| Frequency of Use: | Very Often |
| Business Rules: |  |
| Special Requirements: | Possible need for special algorithm for choosing items from the database |
| Assumptions: |  |
| Notes and Issues: | Make sure the user has GPS turned on |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | |  |
| Use Case Name: | Limit restaurant search radius | |  |
| Created By: | Edgar Sanchez | Last Updated By: | Edgar Sanchez |
| Date Created: | 9/23/2016 | Date Last Updated: | 9/23/2016 |

|  |  |  |
| --- | --- | --- |
| Actors: | Hungry Person (HP) | |
| Description: | Allows HP to limit the number of nearby restaurants that are used to display food options. | |
| Preconditions: | 1. | User has GPS turned on |
|  | 2. | Opened the app |
|  | 3. | Swiped in from the left |
| Postconditions: | 1. | Taps on the apply button |
| Normal Flow: | 1. | HP launches the app |
|  | 2. | HP swipes in from the left |
|  | 3. | HP adjusts the search radius slider |
|  | 4. | HP taps apply button to return to main screen |
|  | 5. | HP resumes searching for food |
| Alternative Flows: | If HP does not set a radius, the app will set a system default based to display at most a certain number of options. | |
| Exceptions: | If GPS signal is lost, app will ask for address. | |
| Includes: |  | |
| Priority: | Medium | |
| Frequency of Use: | Often | |
| Business Rules: |  | |
| Special Requirements: | GPS coordinates or address location will be required to request restaurants from Yelp. | |
| Assumptions: |  | |
| Notes and Issues: | Make sure the user has GPS turned on | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | CO-OP selection | | |
| Created By: | Brashad Hasley | Last Updated By: | Brashad Hasley |
| Date Created: | 09/22/2016 | Date Last Updated: | 09/22/2016 |

|  |  |
| --- | --- |
| Actors: | Couple (CP) |
| Description: | Want to find a decent Meal to be shared by two people. |
| Preconditions: | 1. User has GPS on. 2. User Starts app 3. User sets distance 4. User Selects to dine in. |
| Postconditions: | 1. System has list of restaurants selected by user picks. 2. System has sorted list by distance. |
| Normal Flow: | 1. System takes in user search guidelines 2. System uses Yelp api for restaurant locations. 3. System sets sorted queue 4. System takes user choice and presents the items location(s) by distance |
| Alternative Flows: | If user selects No go back to step 3 with new list.  If Gps is off Use user last known location then step 1 |
| Exceptions: | If Data base cannot be accessed Return Error. |
| Includes: | Date night, New in town, Again |
| Priority: | Very High |
| Frequency of Use: | Often |
| Business Rules: | One person one item rule. No shared plates. |
| Special Requirements: | Portion sizes. |
| Assumptions: | That we would be able to identify viable food items large enough for two people who don’t want a full single plate portion. |
| Notes and Issues: | May want to pick restaurants that have lots of appetizers allowing for couples to select smaller cheaper mills made of appetizers instead of a complete meal. |

# Prototypes:

Figure 2 Secondary menus, Settings(left) likes(right)

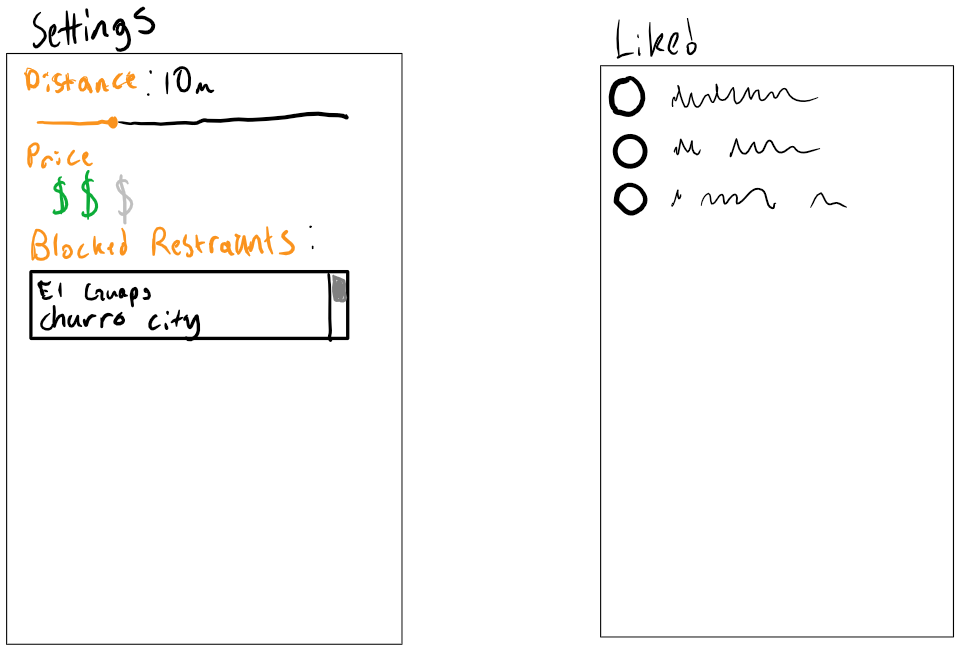
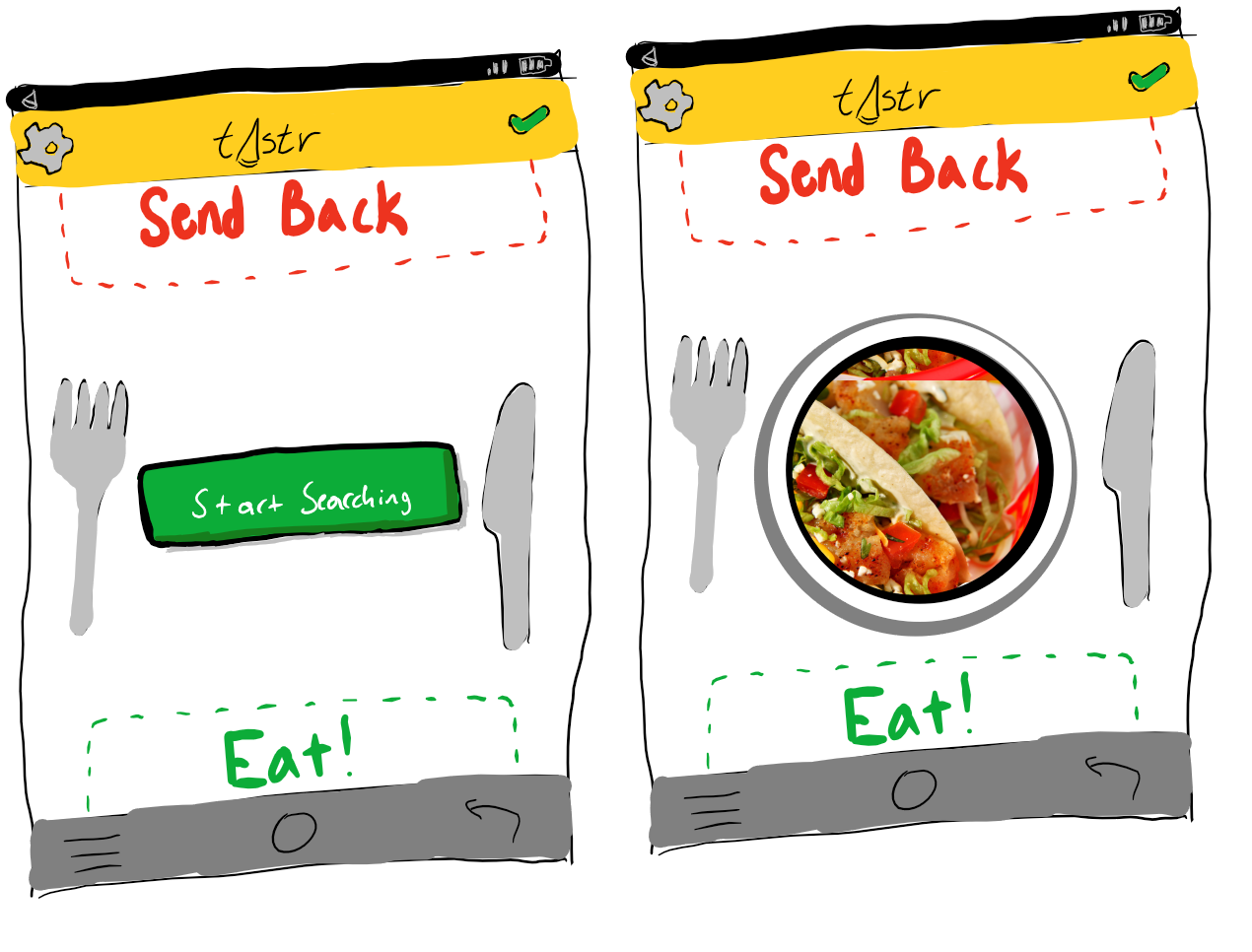


Figure 2 Main Screen for swiping. Initial page(left) searching page(right)

# Non-Functional Requirements

We need to reliably and accurately show restraint in the correct vicinity of the user/device. The UI should feel intuitive and reminiscent the tinder UI with our own take on it. The app design should enforce the idea of committing to an impulse choice of food. If we keep showing other options indecision will set in.