meetchewthere Medium-Fi Prototype

Team Members: Lindsey Kostas, Senthilnathan V, Clay Jones, Jesse Candido

meetchewthere helps people with dietary restrictions find places to eat, meet people with the same dietary restrictions at events and also create your own event and invite others to join you.

Tools:

The meetchewthere prototype was designed with Sketch and animated using Marvel.

Sketch: Sketch is a professional design application intended to ease the process of designing websites, icons, and interfaces. We elected to use Sketch to design our prototype because one of our team members has past experience with and is very comfortable using the platform.

Marvel: Marvel is a free online platform that allows users to turn sketches, mockups, and designs into web, mobile and apple watch app prototypes. It supports a Sketch plugin which means that we were able to import our screens directly from Sketch into the Marvel platform. Overall, the process of animating the Sketch screens was straightforward; however, we experienced two frustrations. First, Marvel does not have a memory of the current state of the user. As a result, the existence of back buttons and dynamic content in our app results in the need to duplicate many pages or show the user inconsistent results. Second, once a screen is imported into Marvel, there is no easy way to edit that screen which means all edits must be done back in Sketch and the revised screens then must be re-imported. This process became difficult and time-consuming, since only two of our team members have Mac computers and were able to update the designs as necessary.

Operating Instructions:

On-boarding:

Upon opening the app, the user sees the sign-in page. If the user has an account they can sign-in with their email and password or via Facebook. A new user can click the create account option and then create an account in a simple two-step process. First they enter their name, email and password, after which they are requested to select at least one dietary restriction. Users can search for specific restrictions and click on the plus/minus signs to add/subtract selected restrictions. Once at least one restriction is selected the user can complete their registration by clicking on “Done” in the upper-right corner of the screen.

Notes:

1) This app is geographically based and the user's "home" location will be selected based on the zip code in which they create their account. In app event and restaurant recommendations will be based on proximity to this zip code.

2) The list of restrictions is hard-coded and the final product will have a much more extensive list. The search functionality acts as a filter on the existing list so that as the user types, options that do not match the query are removed from the list.

3) Marvel does not support typing functionality, as a result the entering a user credential for sign-in/up and typing a search for dietary restrictions are not simulated by our prototype.

4) For the purposes of demonstrating the functionality of the product, we have assumed our user is dairy free and gluten free, to proceed with sign-up please click the dairy free plus first, then the gluten free plus, and then click "Done".

Task 1

Find a Diet Friendly Meal: Upon sign-in/up the app opens to an events feed homepage. The user's current location is indicated by the tab-bar at the top of the screen in which "Events" is currently highlighted. Click on the "Restaurants" tab to switch to the restaurants list view. This view shows a list of all restaurants that have known options for the user's dietary restriction in their area. Users can search for restaurants by:

1) Clicking on the filter in the top right corner next to the search bar to filter by distance, rating, and price; or

2) Typing a search into the search bar

Users can click on a given restaurant to see detailed information for that restaurant. This information is provided in a new page which lists the restaurant's cuisine, ratings, menu, location, phone number, link to its website, and upcoming meetchewthere events. Ratings are broken down by dietary restriction and the user only sees those ratings relevant to their own restrictions. Users can click on the stars to view the comments of other meetchewthere users with the given restriction. The menu brings the user to a page that lists the known diet friendly options given the user's restrictions at that restaurant. To see the full menu the user must click on the restaurant's website linked at the bottom of the page. The list of meetchewthere events shows upcoming events that will be hosted at that restaurant. Users can click on a given event to see the event details.

Notes:

1) The list of four restaurants is hard coded with enough detail to demonstrate the functionality of the application. Only Umami Burger is clickable for detailed information.

2) The restaurant search filter feature simply switches between sorting the list by distance and by rating on-click to show functionality. In the final product we plan to have a drop down menu with distance, ratings, and price as options.

3) Again Marvel does not support typing functionality, as a result the entering a restaurant search is not supported by our prototype. Instead we simulate this process by allowing the user to click on the search field, as if they would to begin typing, upon which "Burgers" is filled into the search bar and the resulting screen is displayed. The user can click the x in the right side of the search bar to clear their search.

4) We chose not to model the ability of users to write comments about a given restaurant as this was not an essential feature of showing the functionality of finding a place to eat.

5) The clickable features to simulate the functionality of the restaurant finding process are the filter, the search bar, Umami Burger, the star ratings, the menu, and the website link.

6) The user can return to the restaurant search page by pressing the back arrow at the top left of the screen.

Task 2

Share a Diet Friendly Meal with People Like You by Joining an Event: Users can find events in 2 ways:

1) Using the event list view. In this view the user is presented with a list of upcoming events in their area that cater to their dietary restriction. The user can scroll through the list and modify what they see through a series of four filter selections of date, time, distance, and price. This view shows the event name, date and time, restaurant at which it will be held, price, restrictions it caters to, and number of people going. The user can click on a given event to see more information.

2) Using the restaurant list view. This is the same view used to accomplish task 1, but we will focus on the fact that each restaurant listing also displays the number of, if any, upcoming events. Again the user can browse or search for a restaurant as in task one and click on the desired restaurant with upcoming events. The user is brought to the restaurant details page which includes a list of the upcoming events. The user can click on the events to get see more information.

Once the user clicks on an event, they are brought to that event's page. This page reiterates the event name, date and time, location, price and restrictions. However, the user can now see how many spots are available, links to the restaurant's diet friendly menu options and the restaurant details page, the name of the event organizer and an organizer generated description of what the event actually entails. The user can see who is going to the event by clicking on the person count. The user can join the event by clicking on the join button at the bottom of the screen, which subsequently becomes an unjoin button. Upon the joining an event, the user is added to the people going list and they now have access to the event chat room. The button to access the chat room appears to the right of the attendee count and when clicked, the user is taken to a messaging board between all current members. If the user clicks unjoin, they are removed from the attendees list and no longer have access to the chat room.

Notes:

1) In the event view, the list of four events is hard coded with enough detail to demonstrate the functionality of the application. Only Burger Party is clickable for detailed information.

2) The event search filters are not functional in this prototype, but just intended to provide a frame of reference for how an user would go about searching events. The prototype only supports a simulation of the intended functionality whereby if the user clicks on the distance filter, the filter switches between 10 and 5 miles, and the list is updated accordingly. In the final product we plan to have a drop down menus/date and time pickers for the four filters.

3) On the Umami Burger details page, only the Burger Party upcoming event is clickable for detailed information.

4) Again since Marvel does not support typing functionality, active chatting is not supported by our prototype. Instead we have displayed to static messages to illustrate what the chat room would look like to the user.

5) The user can return to the event view/restaurant details page by pressing the back arrow at the top left of the screen.

Task 3

Build your Community and Create a Food Centered Event: Users can create an event by clicking on the "Create Event" option on the bottom Nav Bar that appears on most of the app screens. Once clicked the user is presented with an event details form with required fields to fill out. These fields included event name, date, time, description, number of people, and location. Event name, description, and number of people are typed inputs. Date and time will be selected

using a datepicker. The user can then select the restaurant at which they want to host the event by clicking on the location bar. Upon doing so they are brought to a screen that resembles the restaurant list view with the only difference being the absence of the nav bar. As in task 1, the user can search for and select a restaurant using the same process of filtering and typed searches. Similarly, the user can see detailed information about a restaurant by clicking on it from this list. The user can click “Choose” in the top right corner of the detailed information page to select the restaurant as the event location, after which they are brought back to event details form with their selected restaurant filled in as the location. After all the fields are filled in, the user can click on “Next” in the top right corner of the screen to proceed. The user is then given the option to explicitly invite friends if they so choose. The user can select/unselect friends to invite by clicking on the plus/minus signs to the right of users names. After selecting whoever, if anyone, the user wishes to explicitly notify about their event, the user clicks “Done” and they are brought to the final review of the event that they have created. They then can click “Create Event” if they are satisfied or click the back button in the upper left corner to return and edit their event. After clicking “Create Event” the user is directed to their “My Events” page and their event is now listed under their upcoming and hosted events.

Notes:

1) Again Marvel does not support typing functionality, as a result the entering event information is not supported by our prototype. Instead we simulate this process by allowing the user to click on the event name, date, time, description, or number of spots fields as if they would to begin typing, upon which all typable fields are populated.

2) As a result, the event details are hard-coded to demonstrate functionality and the user will be hosting the event Sandwich Party at 7:50pm on November 15, 2016 at Umami Burger with 15 people.

3) The location choice is fully functional in that the user can click on this field and be directed to the restaurant search page to select the location. However, since the search and filter functionalities were already demonstrated in task one we have chosen to disable these functionalities for this task. Also, as in the rest of the prototype, Umami Burger is the only clickable option.

4) The list of friends to invite is hard coded and limited to three for the purposes of showing the proposed functionality of the app. We plan for the final product to have a list that is generated based on users with whom you attended past events.

5) To show the selection functionality, the user can either click next if they decide to not invite anyone or click the plus/minus next to Senthilnathan. No other users can be selected/unselected.

Non-Task Related Functionality

Navigation Bar: A global app navigation bar is located at the bottom of most screens. This nav bar serves to provide easy and convenient navigation through the app. The nav bar has three options: Discover, Create Event, and My Events. The current selection is highlighted in blue and the user can click to change the selection. Clicking discover brings the user to the event list view. Clicking Create Event brings the user to the empty create event form. Clicking My Events brings the user the a view of the user's upcoming events (described below in more detail).

My Events: This series of three pages is the place where the user can manage their events. It is accessible via the My Events option in the nav bar, which when clicked brings the user to the user's upcoming events page. This page resembles the event list view in that it is searchable through the date, time, distance, and price filters but now in place of the tab bar which listed events and restaurants as options there is a tab bar which lists past, upcoming, and hosted as options. Currently, the upcoming tab is highlighted to indicate the user's current view. The user can toggle between the three views by clicking on the respective tabs. The past view shows events the user has already attended, the upcoming view shows events the user has joined which have yet to happen, and the hosted view shows only user hosted events. Note that past and upcoming events show both user hosted and user joined events and hosted events are denoted by a red figure next to the event name. A user can see the event details page for any given event by clicking on that event. A user can edit the events they own by clicking that event and then clicking the Edit button at the top right of the screen.

Current Limitations

Many of the current limitation were mentioned above, but we have summarized them here for convenience.

1) Marvel does not support typed input so any and all features that involve such input are simulated through static hard-coded features.

2) We chose to fully develop only one restaurant and one event to demonstrate the functionality our our app. As a result, there are many unclickable options but we felt that our one event and one restaurant were sufficient to provide the user with the experience of using the app.

3) The content of our app is not dynamic and scrolling is not enabled. Instead there are hard-coded lists of restrictions, restaurants, restaurant reviews, events, and users as well as chat messages to give the user an idea of what the app entails without building out full dynamic functionality. We believe that users of the prototype can understand this decision and that these features are not essential to relaying the functionality of the app itself.

4) Due to the hard-coded data, the dates of the events are not always relevant.

5) Back functionality is not fully representative of how streamlined the process will work in the real prototype due to Marvel's inability to trace the user's path to a current page.

6) By the nature of how customized our app is to each individual user, for our prototype, the user's choices and event information are all pre-selected. The user can go through the process of finding a restaurant, joining an event, and creating an event, but they do not actually get to make any of their own decisions.