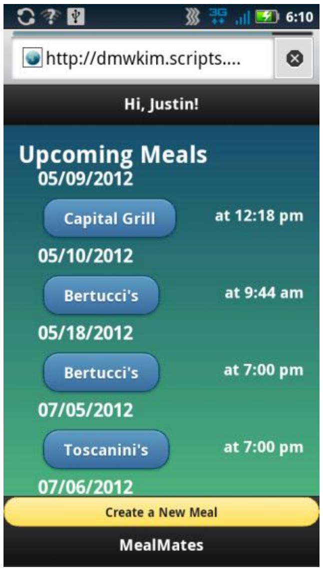
Brian Bell, David Kim, Justin Merritt

6.813 GR6

Design

Our final design, just like the Paper and Computer Prototypes, starts with a homepage, as shown below:



http://dmwkim.scripts....

Hi, Justin!

Upcoming Meals

05/09/2012

Capital Grill at 12:18 pm

05/10/2012

Bertucci's at 9:44 am

05/18/2012

Bertucci's at 7:00 pm

07/05/2012

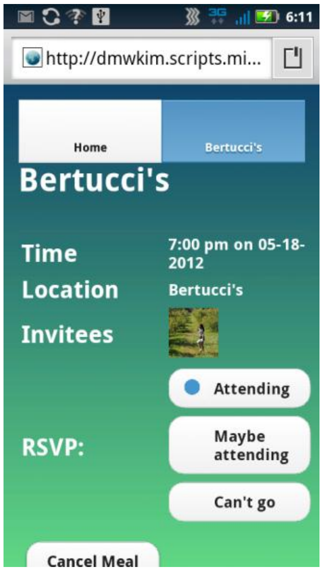
Toscanini's at 7:00 pm

07/06/2012

Create a New Meal

MealMates

The homepage gives you an overview of upcoming meals, and has a button to let you create a new meal. If you click on one of the blue restaurant buttons, you are brought to a summary page:



http://dmwkim.scripts.mi...

Home | Bertucci's

Bertucci's

Time: 7:00 pm on 05-18-2012

Location: Bertucci's

Invitees

RSVP:

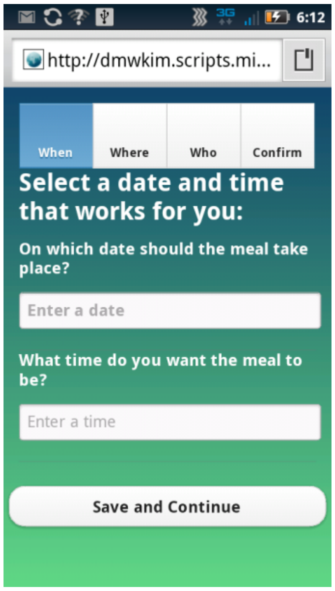
● Attending

Maybe attending

Can't go

Cancel Meal

The summary page lets you view a snapshot of the meal information, modify your RSVP status, and if you are the creator of the meal, allows you to cancel the meal. Going back to the homepage, if you click on "Create a New Meal" you are brought to the following page:



Select a date and time that works for you:

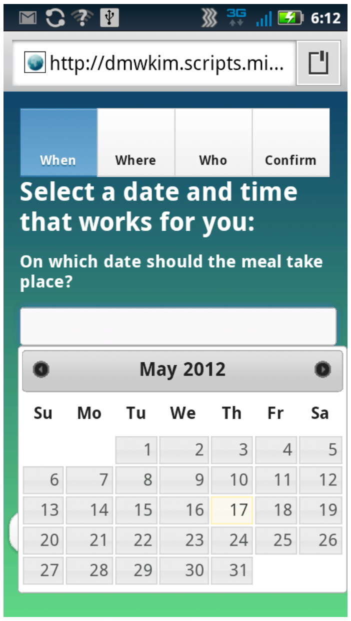
On which date should the meal take place?

Enter a date

What time do you want the meal to be?

Enter a time

The "When" tab allows you to specify a time and a place for the meal. Clicking on the input for "Enter a date" pops up an interactive calendar that lets you choose the date with a quick tap, and clicking on the input for "Enter a time" pops up a sliding time interface that allows you to set the time. Both features are shown in the images below:



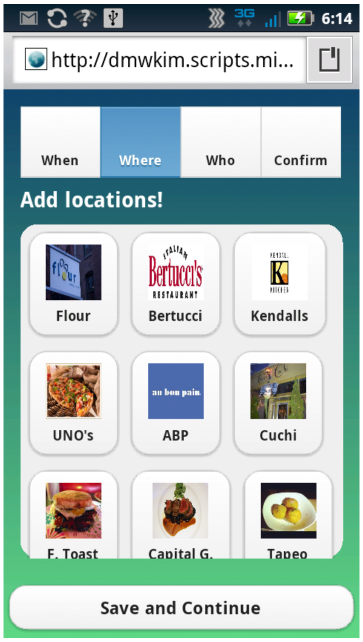
Select a date and time that works for you:

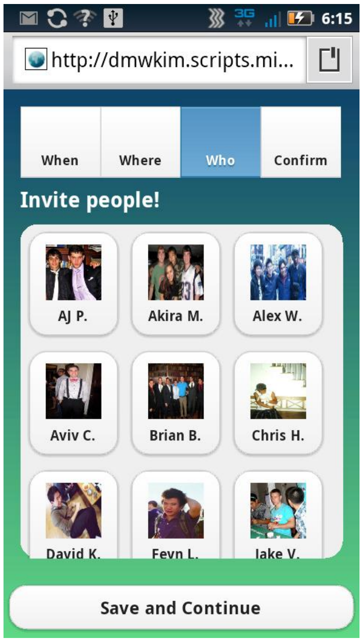
On which date should the meal take place?

May 2012

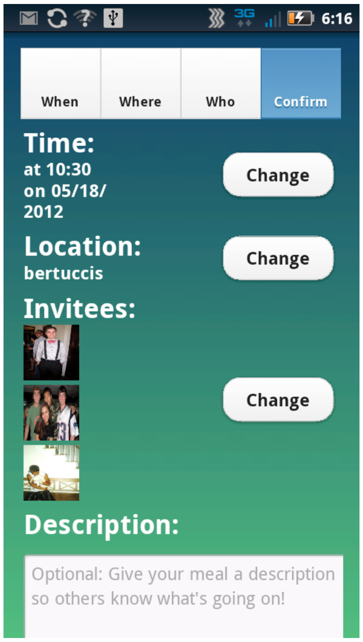
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Once the time and date have been input, you move on to the where and when pages, which allow you to pick a location and invite friends to the meal. These two interfaces are similar, except that you are only allowed to choose one restaurant, whereas you are allowed to invite an arbitrary number of friends. These interfaces are shown below:





When you are satisfied with all of your choices, you can move to the Confirm screen to review your choices and add an optional description:



Time:
at 10:30
on 05/18/
2012

Location:
bertuccis

Invitees:

Description:

Optional: Give your meal a description so others know what's going on!

You can then confirm your meal, and it will show up on the homepage.

Changes Made to the Design

Removal of the "What" Page: The largest change we made to the design between the Computer Prototype and the Implementation phases was the removal of the "What" tab. Many of our users said that it didn't make sense for a meal with a few friends to have a title, and that the forced inclusion of a title broke our metaphor for a meal. In response, we removed the "What" tab and moved the Description input over to the Confirm page. We feel that this makes more sense, as it will be easier to add a Description to a meal after you have already decided when, where, and who you will be inviting. This also had consequences in our Homepage, as we were no longer able to use the title as an identifier. We decided to just use the name of the restaurant as an identifier instead.

Change the Time input to a slider: We also moderately changed the When page between the Computer Prototype and the implementation. We decided to implement a slider to pick the time, which we felt was more tactile.

Removal of Drag and Drop Functionality: In the "Where" and the "Who" pages, we removed the drag and drop functionality and replaced it with simply tapping on whichever locations or people you would like. The reason we made this change was the limited screen space on a mobile phone -- it took too much screen real-estate to have a sizeable drop zone, so users were making many errors.

Removal of the ability to rank restaurant locations: While the primary purpose of our application was to allow some feedback when choosing the time and location of the meal, we felt that users did not understand how ranking their choices was ultimately going to affect where they ended up going to eat, so for now we just have the creator of the meal specify a location.

Brian Bell, David Kim, Justin Merritt

Implementation

Quick Overview of Application Features

- Front: standard HTML/CSS/jQuery mobile
- Mid: php
- Back sql database

As we developed a web application, the front-end was implemented in the HTML/CSS/Javascript stack. In particular, we used the jQuery mobile framework to greatly simplify implementation. The trade-off here was that the jQuery mobile framework was very rigid and hard to customize, yet allowed for rapid prototyping and iterative development.

The back-end was implemented using PHP, accessing a MySQL database to store all information. The front-end pulled data from the database through a GET call to a PHP back-end, which simply pulled all stored rows from a “meals” database table. Similarly, a modification to or deletion of the meal simply made changes to the database table row corresponding to that meal, such that the home screen of our application would display those changes.

An unfortunate consequence of developing a mobile application on a web platform is the speed and responsiveness of the application. It’s clear that a native application, as it requires much less network latency overhead in loading the interface (at the very least), would feel more responsive. We are also limited by the technology of the browser at hand, whereas in a native application, this limitation does not exist.

Another consequence of developing on the web platform is that the keyboard cannot be turned off when selecting an input box. This was of particular annoyance when the user selected the input box that popped up a jQuery UI calendar “datepicker” object, which was then obscured by the half-screen touch keyboard, which could not be disabled.

Brian Bell, David Kim, Justin Merritt

User Testing

We conducted our user testing through both a survey and in-person user testing, where we observed users' actions and hesitations during use of our interface. The survey was composed of questions about the general impressions the user had of the interface.

1. What is the most frustrating part of this application?
2. What is the most gratifying part of this application?
3. What tasks were efficient and speedy?
4. Did you feel like you wasted time on any part of the application?
5. What part of the application confused you the most?
6. If you had to narrate this interface to your grandmother over the phone, do you think it would be easy or difficult? Where might you get stuck?
7. What single feature do you wish was in this application?

For the in person user tests, we initially let the user perform the same tasks we presented them with in paper prototyping:

Task 1: Create a New Meal
Task 2: Modify your RSVP Status for a Meal
Task 3: Cancel a Meal

We gathered several conclusions from our user testing:

- **Efficiency - Date sliders - Minor** - Users reported trouble getting exactly the right time they were going for using the date sliders. We improved this by including some optional radio buttons to snap the interface to 15 minute intervals, reducing the impact of this problem
- **Learnability - Virtual keyboard - Minor** - Users reported a bit a frustration with the fact that the virtual keyboard presented itself at every opportunity. We cannot disable this; mere selection of any form in the browser brings this up. One option is to not use a form at all, but instead have the date and time dialogues pop up at the press of the button.
- **Safety - Meal Refresh - Minor** - Users were confused after they initially submitted their final meal confirmation because the meal would not load in the Homepage until a page refresh. The users felt that the meal they just made wasn't actually being counted for anything. A solution that would provide more feedback is to just update the homepage or force a refresh upon going back to the homepage

Users were representative of our population overall. All of them were generally young, well versed in typical app affordances, interested in going out and meeting up with their friends.

Brian Bell, David Kim, Justin Merritt

Reflection

All in all, creating this mobile web app was a huge learning experience for the three of us. None of us had any mobile development experience of any kind, and we decided to take the plunge and learn something new as opposed to staying in our comfort zone developing conventional web applications. The variable screen sizes, different browser standards, the challenges of navigating with a touch screen and maintaining external consistency with other mobile web applications were all new challenges we embraced.

We chose to use a tool called Codiqa for our initial computer prototype, and in retrospect that was wasted time on our part. We invested a lot of effort into producing a high fidelity computer prototype, much of which would have been better spent working towards the front end of the actual application for use as the computer prototype. Further, Codiqa is itself still a prototype in the experimental stages, which caused us a lot of grief as we tried to execute our vision. Eventually we just scrapped most of it and moved forward.

At one point, we considered implementing a negotiation dialogue between invitees to choose the meal’s location. The back-end would take each invitees restaurant choices and then optimize everyone’s preferences to reach a mutually beneficial decision that would serve as the final location. As we moved forward with the application, we decided to scrap it since it wasn’t core to the design of our user experience, though it’s certainly a backend feature we’ll consider implementing in the future as an interesting project.

Limiting our scope on some of the non-UI features allowed us to spend more time getting feedback and iteratively developing our core UI. We moved from swiping-based navigation, to arrows in the corners, to the four tab system we have now based on iteratively speaking with users. We found out that the “cool factor” of swiping was less important to them compared with the familiar arrow affordance, and that the feedback of having all four tabs let them know how long the meal creation process was going to be, and allowed easy jumping between pages.

We evaluated the results of our observations based on how much they impaired the efficiency of our app, above the other two usability qualities. We figured that users who got over the learnability hump would appreciate and re-use the app based on how easily they could create and manage their events, and as long as the safety was sufficient such that the app wasn’t frustrating to use or seemed buggy. In the end, as development rolled on, we focused more on learnability and safety which had been emphasized yet clearly secondary early on the design process. We feel that the final application takes all three usability dimensions into account.