Interactive Prototype

David Bai – Design
Aaron Miller – Testing
Joseph Muhm – Documentation, Group Leader

Online Version: http://bit.ly/8Smkow

Contents

Problem Overview 2

Tasks 2

Easy 2
Medium 2
Hard 2

Interface Changes 2

Edit/Save Posts 2
Location Field 3
Text Input 3
Tabs 3
Minor Additional Revisions 4

Interactive Prototype 5

Overview 5
Task Scenarios 5
Implementation 7
Special Techniques 7
Missing Functionality 8

Problem Overview

Maintaining a healthy diet is a difficult task because it is often hard to find healthy choices when planning meals under limiting time constraints. Many people want to eat healthy but cannot find real time suggestions or role models to follow. As a result, it is common for people to settle for whatever is most convenient and pledge to eat healthier next time. We see Fitter as a solution to this problem because it provides meal suggestions in real time and the ability to follow the dietary habits of role models. Users are able to see what others are eating and share recipes for others to try. Fitter provides users with many ideas and recipes providing them with options for healthy eating.

Tasks

Easy

The easiest task is to use Fitter to find a recipe to use a specific ingredient in. This task demonstrates the search functionality to find posts by keyword. Users are able to search based on ingredient to find posts that contain that ingredient (they are also able to search for people's names to find all of that person's posts). A post-filtering feature allows users to narrow-down their search results to help them find exactly what they want.

Medium

A medium-difficulty task is to use Fitter to input and share a recipe. This task demonstrates the ability for users to post data about what they eat. Users are encouraged to add photos and recipes to their posts. These posts are shown in the feed and in search results. Users subscribe to others based on these posts, which encourages users to frequently post interesting information including photos.

Hard

The hardest task is to use Fitter to find a suggestion for what to eat right now. This task demonstrates the user's ability to browse through and filter the feed of posts. The user is able to filter the feed based on whose posts are shown and whether or not to only show posts with recipes. On top of the feed is a set of tabs, which filter whose posts are shown in the feed (only those who you subscribe to, all or Fitter, only your posts, and the posts you have marked as favorite).

Interface Changes

After collecting feedback from user tests with our low fidelity prototype, this interactive prototype includes fixes to the important problems our users pointed out and several additional features they requested.

Edit/Save Posts

A relatively simple, yet forgotten functionality that the user wanted was the option to save posts that they liked for easy access at a later point in time as well as an option to edit their own posts. We decided that you could only edit a post if no one has added feedback to it (added a comment, "liked", or "disliked"). This way, the comments and ratings would stay consistent with the information that was originally attached with the post. To implement the saving ability, you could hit the "star" button on the post and that would make it so you could access all your favorite posts from the Favorite tab on the feed screen.

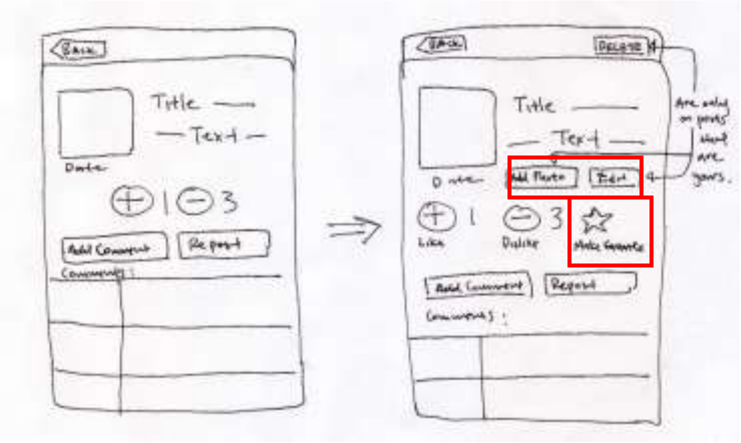

Figure 1. Old vs. revised post details screen.

Location Field

Users had a hard time dealing with the location input field when adding a post. They were either confused with what to add or how it was relevant. To solve this we are going to populate this field with where they are currently based on their GPS location. Rather than just having an address of where they are, the GPS would have the ability to know if the user were at a specific restaurant, store, their own home, etc. If they want to, the user can manually change this field to whatever they want when adding a new post.

Text Input

When inputting text while adding ingredients and steps for recipes, instead of having to add each ingredient/step individually, user can now add everything at once. This makes inputting data more fluid and less of a hassle. Also the ingredients section is pre-populated based on what food item a user enters in the "What" field of the post entry screen. So for example if they entered in "Spaghetti and Meatballs" in the name field, the ingredients section would automatically include pasta, meatballs, marinara sauce, and similar ingredients. If this function gives the user an ingredient they don't need they can delete it easily by editing the section's contents.

Tabs

On the main page we added tabs above the posts (see Figure 2). Users can now filter the feed's posts to show those from people they subscribe to, their own posts, favorite posts, or a list of posts by anyone on Fitter which shows the most popular posts or posts by the most popular users. Now that users have the ability to see the popular posts from everyone on Fitter, they can follow and view posts from popular people such as celebrities and athletes. This also gives new users somewhere to start from since they can easily find people to subscribe to right away. Finally, users can look at the posts they saved with the "favorite" tab and have a quick way to view their own posts with the "Mine" tab.

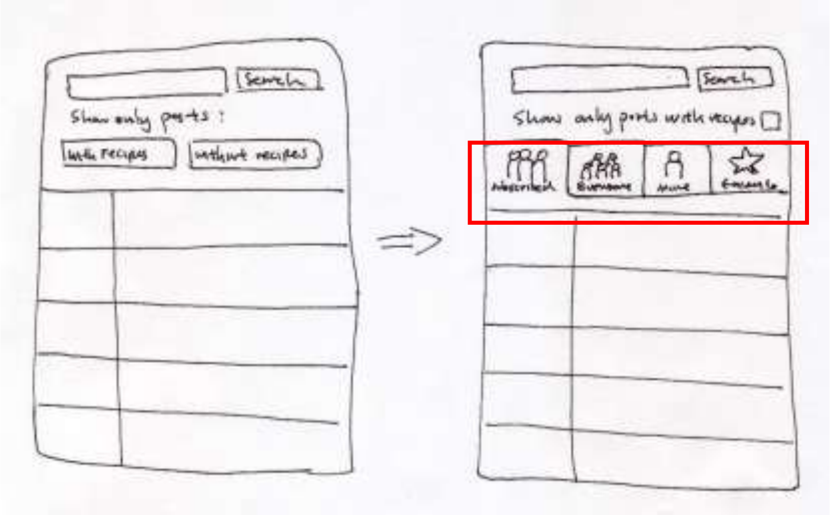

Figure 2. Old vs. revised feed screen.

Minor Additional Revisions

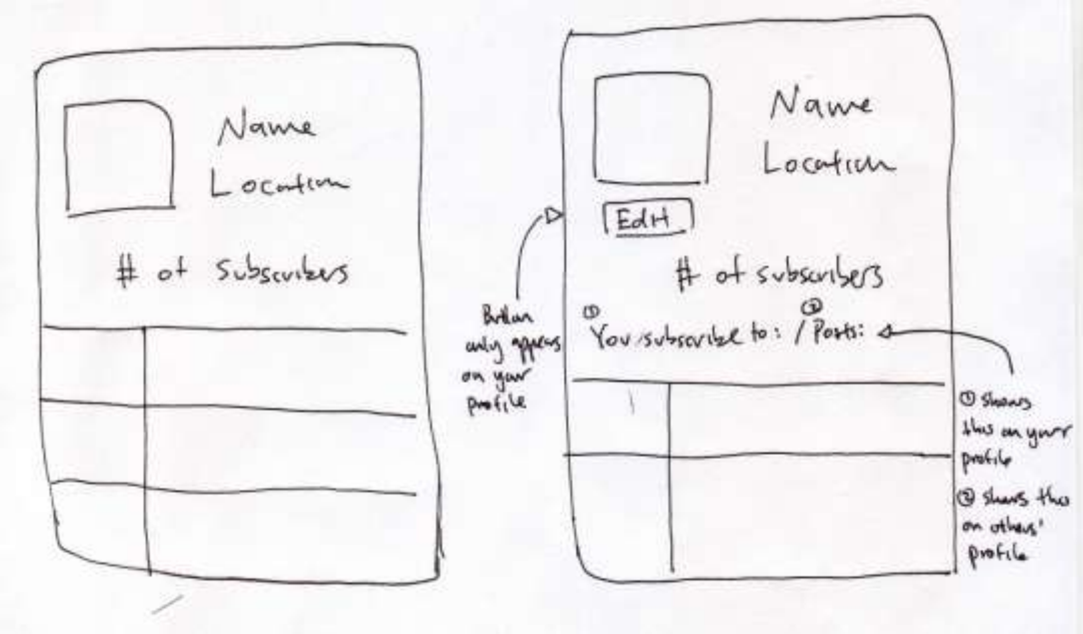
Name
Location
# of Subscribers
Name
Location
Edit
# of subscribers
Button only appears on your profile
① You subscribe to: / ② Posts:
① Shows this on your profile
② Shows this on others' profile

Figure 3. Old vs. finalized profile screen(s).

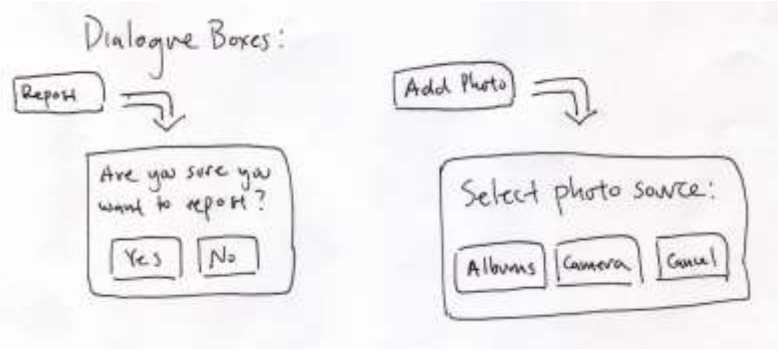

Figure 4. ‘Repost’ and ‘Add Photo’ confirmation dialogue boxes.

Interactive Prototype

Overview

We implemented the interactive prototype as a collection of HTML files, each representing a core screen of the application. Each screen is laid-out with its own cascading style sheet. JavaScript is used to boost the interactivity of the screens in order to support such functionality as searching and feed post filtering. The prototype is best viewed using the latest version of Mozilla's Firefox Web browser on a Windows or Mac system with a screen resolution of at least 1024 x 768.

Task Scenarios

1. To find a recipe to use steak in, the user initially enters “steak” into the text box to the left of the 'Search' button. The user then clicks the 'Search' button and is taken to a similar-looking screen which displays recent posts that contain the word “steak”. The user then checks the box next to the phrase “Show only posts containing recipes” to hide all non-recipe-info-containing posts. Now, like in the first task scenario, the user clicks on each post to view more details and feedback and then make his decision of which recipe to use the steak in.

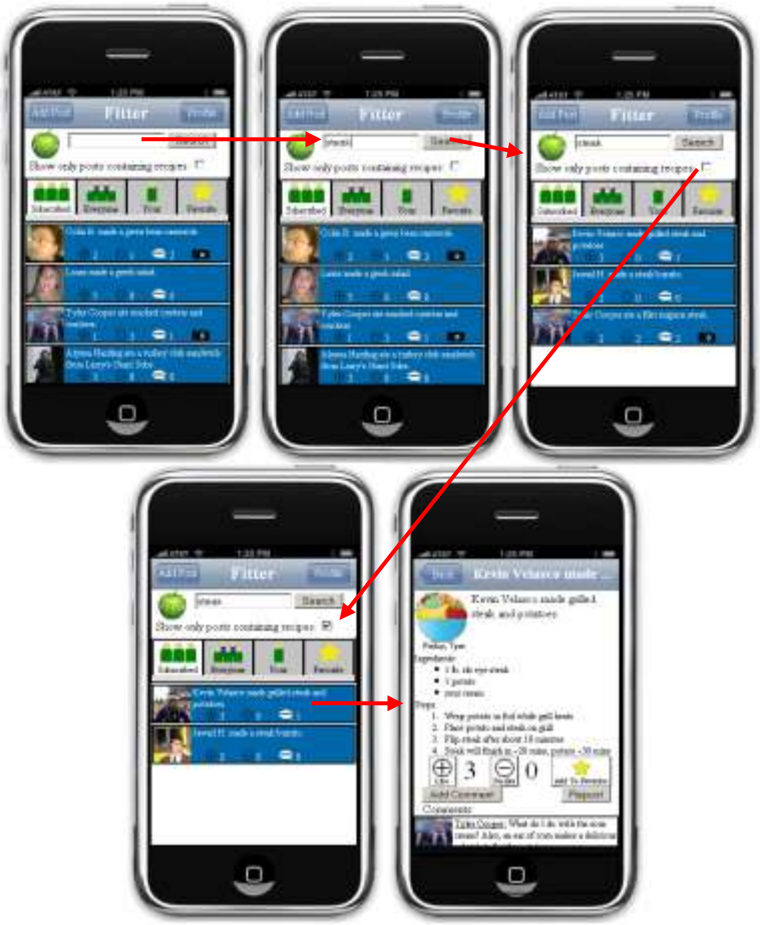
Filter
Search
steak
Show only posts containing recipes
Kevin Velasco made grilled steak and potatoes
Add Comment

Figure 5. Storyboard for completion of easy task’s scenario.

2. To input and share a spaghetti and sausage recipe, the user starts by clicking the 'Add Post' button in the title bar of a feed-containing screen and is taken to a screen which he can enter the recipe's details. On this screen, the user enters the name of recipe ("Spaghetti & Sausage") into the text box underneath the "What did you eat?" text. The user then presses the 'Add Recipe' button to input ingredients and steps. The user then presses the 'Add' button underneath the "Ingredients" text and verifies that the automatically populated ingredients in the editor window's text field are correct. He then presses the 'Ok' button on the top right corner of the editor window to make the ingredients appear. The user then presses the 'Add' button underneath the "Steps" text and verifies that the automatically populated preparation steps in the editor window's text field are correct. He then presses the 'Ok' button on the top right corner of the editor window to make the preparation steps appear. The user presses the 'Add Photo' button and chooses 'Ok' on the dialog box which pops-up to attach a photo of the dish to the post. Finally, he clicks the 'Post' button on the right side of the title bar to complete the addition of the post. The user is then taken to a modified version of the main feed screen which displays his post at the top of the feed. Optionally, he can click his post to view its details and even use the 'Edit' button on the post's details screen to return to a populated version of the screen he used to add the new post.

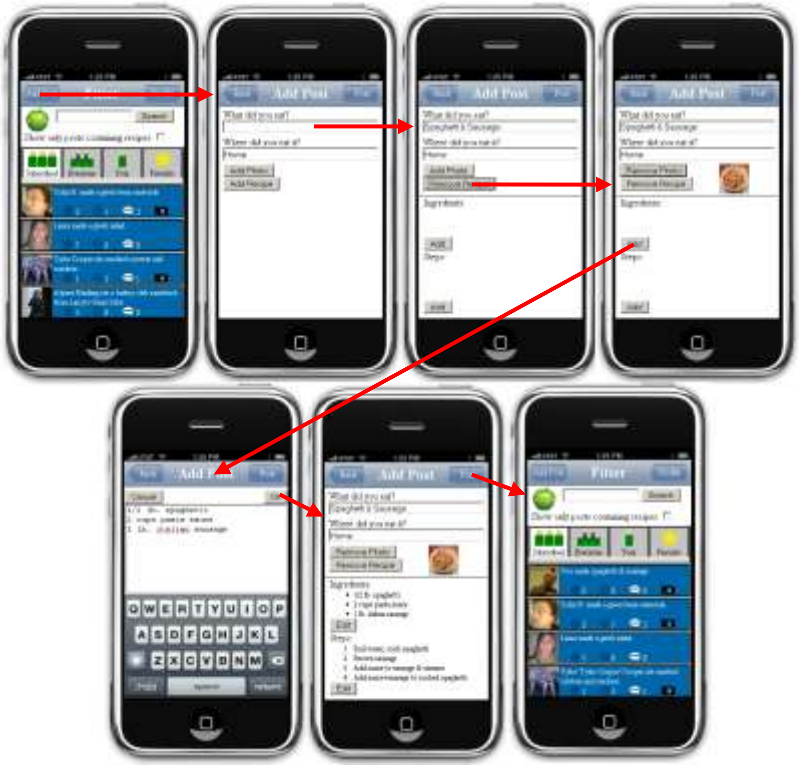

Figure 6. Storyboard for completion of medium task's scenario.

3. To find a suggestion for what to eat right now, the user clicks on posts on the main feed screen he is initially presented with to view more details about the posts. Once the user has viewed the posts, he can use the feedback (comments, number of likes and number of dislikes) along with his tastes and cravings to make a decision about what to eat.

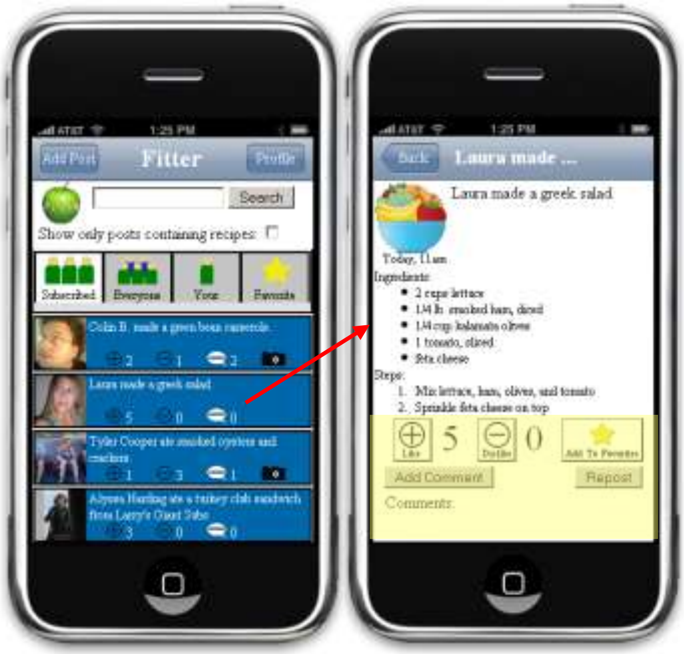

Figure 7. Storyboard for completion of hard task’s scenario.

Implementation

We used the free trial version of Adobe's Dreamweaver CS4 to rapidly create skeletal versions of each screen. Dreamweaver was particularly useful here as it allowed us to visually layout the screens without writing a single line of code. We then used Dreamweaver's split-screen view of code and visual design to tune the screens and add content. Once each screen's content was generated, we used Dreamweaver's code view to add JavaScript code to give each screen more interactive functionality.

While Dreamweaver was initially useful to us, we moved away from it while fine-tuning the appearance of the prototype's screens, using text editors instead. However, we did make use of Dreamweaver when we needed to make significant changes to a screen's layout (rearranging content, adding captions to images, etc.). Dreamweaver's visual design view of the screens was disappointing, as it didn't support the JavaScript code we added and also had noticeable style-rendering differences with the Web browser we were developing the prototype for. Still, Dreamweaver was immensely useful for quickly getting our prototype off the ground and laying out its screens.

Special Techniques

Since we only intended for the prototype's search functionality to be utilized in the completion of the second task scenario, we only implemented a search results page for “steak”. So any time a user searches a feed-containing screen, s/he is taken to the search results page for “steak”.

We also bootstrapped the ingredient or preparation step information for the third task scenario's recipe when the user pushes the 'Add' button in the Ingredients or Steps section on the post entry screen (after pushing the 'Add Recipe' button to display these sections).

Missing Functionality

Of all the functionality we did not implement, the lack of the ability to create your own or modify existing content is the most noticeable. Users of the prototype cannot add comments or give positive or negative feedback on posts, nor can they edit their profile information. Users do not need to be able to do any of these things to complete the scenarios for the tasks. When adding a new post, changes made to the “What did you eat?” text field, “Where did you eat it?” text field, and recipe ingredients and steps have no effect, since all the prototype-generated content in these locations needs to remain as is in order for users to complete the third task scenario. Also, the user can only attach a photo of spaghetti and sausage when pushing the 'Add Photo' while creating a new post, since the only time the user needs to create a new post is to complete the third task scenario (In the fully-implemented application, the confirmation dialog box for adding a photo to a post would show the live camera image from the iPhone and then the image would be captured when the user presses 'Ok').

The prototype is missing profile screens for each of the different users who are associated with posts or comments. We chose not to implement these, as they provide unnecessary information which does not contribute to the completion of the task scenarios. Consequentially, when a user is viewing a feed-containing page and clicks on the user's image for a post he is taken to the post details screen for that post, not that user's profile as he would be in the fully-implemented application.