RouteMyRun

Hi-Fidelity Prototype

Toby Escarez - (Team Manager)
Steven Myhre (Testing Manager)
Lisa Fiedler - (Documentation Manager)
Stephanie Smallman - (Design Manager)

This document is online at:

http://www.cs.washington.edu/education/courses/cse440/CurrentQtr/projects/routemyrun/documents/RouteMyRun-HighFiPrototype.pdf

Our interactive prototype is located at:

http://www.cs.washington.edu/education/courses/cse440/CurrentQtr/projects/routemyrun/prototype/

Problem and Solution Overview

For most joggers the who, what, why and when of jogging come easy, but many people don't have a very good handle on where they will be running. Currently, people have to resort to mapping out a run using Google maps or by simply exploring an area on foot. Many times it is impossible to know which routes are scenic, safe, or challenging without simply running the route. But doing so can end up being a waste of time or even dangerous. Also, it is difficult for avid runners to find routes with similar characteristics to their favorite routes or to find slightly more challenging routes as they improve. If a jogger wants to go for a run in a new place, the quality of their run may be hindered due to the fact that they are unfamiliar with the area. Our application aims to solve these problems by providing an interface that aids in deciding where to run and finds the best route for you in any particular area. Similar to currently available applications, it will provide useful information such as pace or distance and will store statistics for all of a person's past runs. In addition, our application will include a map interface, which will allow users to plan routes more easily and track their real-time location during a run. Routes can be generated based on input such as desired distance, terrain, or points of interest. Clients can also access a list-view of past routes and view their statistics on any chosen route. They can then decide to either run the route again or race their best time on a particular route.

Tasks

Easy: *Go on a quick run and record your stats*

You want to go on a short run through the U-District. You are an avid runner and very familiar with the area, so you already know the route you would like to run. You usually keep a running log and wish to record the pace and the distance of this run. Use RouteMyRun to track your run and record the corresponding statistics.

Medium: *Plan a route, insert waypoints*

Today you wish to go for a long run through both Madison Park and the University of Washington campus. You would like your route to start at your house and loop through both points of interest, while taking into account your preference for only moderate elevation change. You also want to avoid as many major roads as possible and would like to run roughly six miles. Plan a new route, taking into account all the desired constraints.

Difficult: *Look up past runs; know past pace and run the same route faster.*

You wish to improve your overall time on a particular route. First, you need to review your current best pace on the desired route in order to know what time you are trying to beat. Then, during your run, you would like to receive real-time feedback that informs you of how you are performing relative to your past best. Use RouteMyRun to race yourself on the specified route.

Interface Revision Sketches and Rationale

After conducting the Usability Testing, it became apparent that several interface modifications were necessary to improve the overall usability of our prototype.

The first flaw in our interface that was revealed through the user testing was the ambiguity in how to set a waypoint on the map. Our initial version gave no clear indication of how to do so, assuming that clients would intuitively know that tapping the screen would produce the desired result. To alleviate this confusion, we chose to add a small caption at the top of the map that states, "Tap the screen to add waypoint" (Figure 1). This small addition would go unnoticed to experienced users, but would provide much-needed clarification to new and infrequent users.

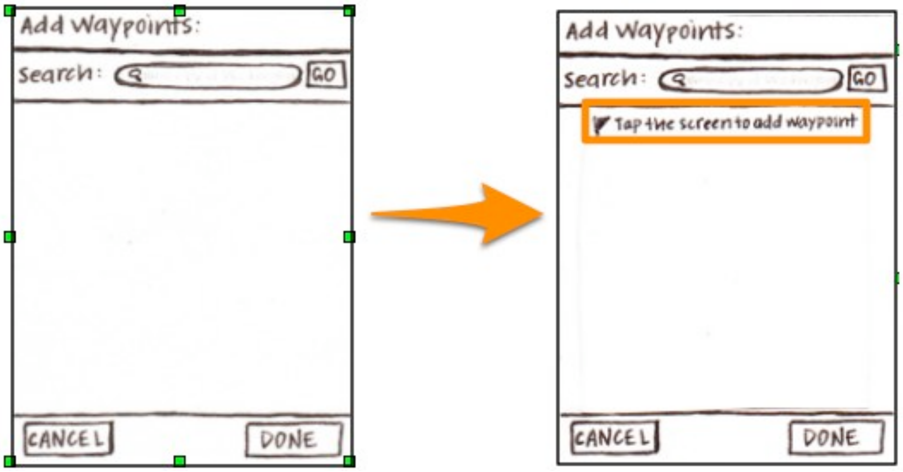


Figure 1

In addition, no visual feedback was given informing the client when a waypoint had been successfully set. In the current version, we decided that a note will appear at the bottom of the Route Planning screen telling the user that "n waypoints have been set," where n is the number of waypoints the user has set (Figure 2).

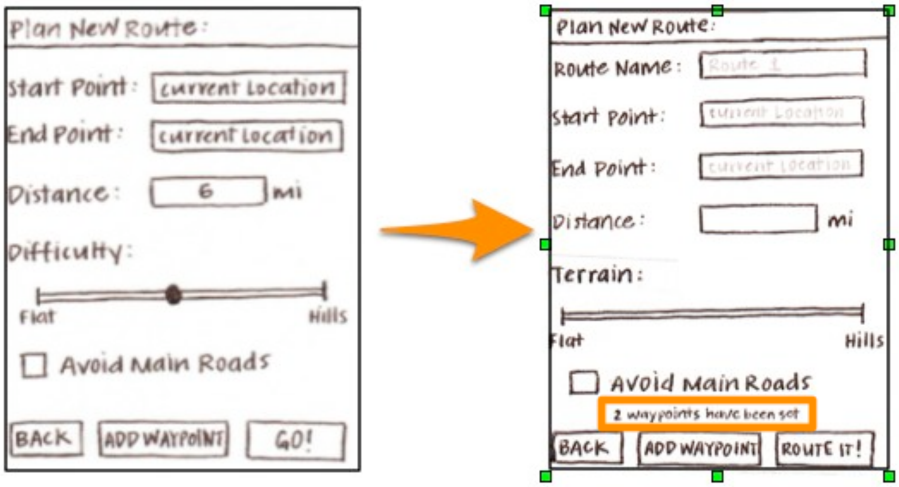


Figure 2

A final change concerns a few of our button labels, which we felt could be worded differently to improve usability. First, on the Suggested Route screen, we have changed the Edit label to Back

(Figure 3), in order to increase internal consistency with the other screens of the interface. Second, on the Route Planning screen, we have modified the Difficulty label to instead say Terrain (Figure 4), since this measure is based solely on the elevation change in the proposed route.

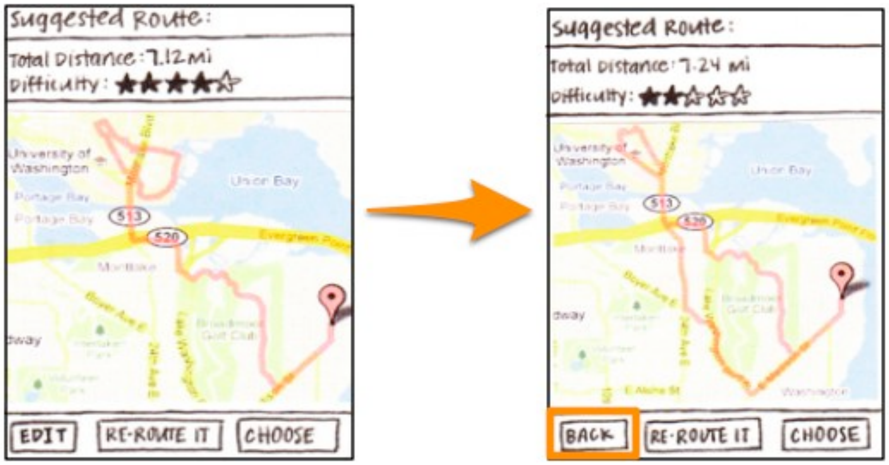


Figure 3

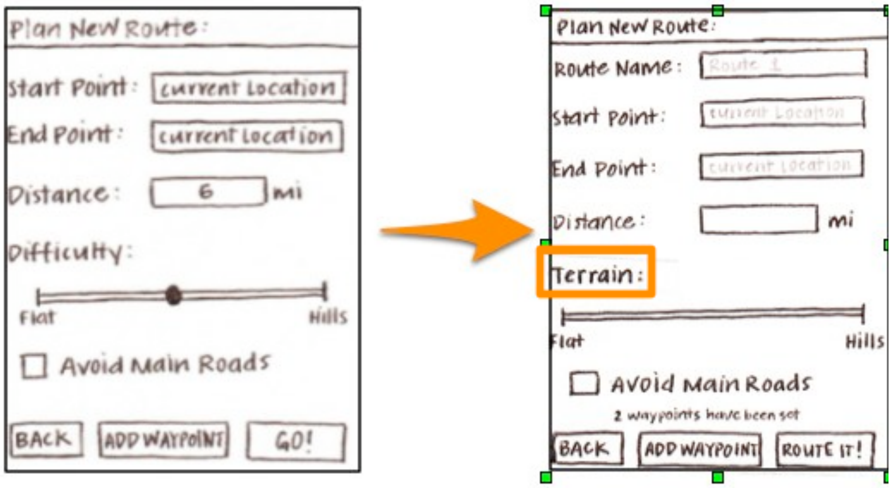


Figure 4

Prototype Overview

Converting our low-fidelity prototype to a high-fidelity prototype required us to make many specific design choices, and we were forced to choose which features to concentrate on and which to leave out. We were sure to implement all of the functionality relevant to the scenarios we chose, although we had to make most of the features follow a fixed path. We used an HTML framework for designing the initial prototype and manually added CSS and Javascript to enhance it further. Using images and Javascript, we were able to create fixed paths that rely on users clicking on the correct locations to successfully execute a given scenario. Instructions are given on the side of the interactive prototype to aid the user in carrying out each task correctly. Since we were designing using iPhone specific elements, some of the visual features and button placement from our low-fidelity prototype had to be modified in order to fit with the iPhone's schema.

Our interactive prototype supports three major flows of execution, as seen on the home screen of our application. The first is for the client who does not want to lose time planning a new route or analyzing statistics, and would simply like to start tracking as quickly as possible. Such a client would select "Quick Run" from the main menu (see Scenario A). The second flow pertains to a client who wishes to go for a run in an unfamiliar area or run a new route. This client would choose "Plan a Route" from the home screen, and would subsequently enter all the desired constraints, approve or revise the suggested route, and either begin tracking or save the generated route for later use (see Scenario B). The last flow would be used by a client who wishes to analyze his previous run statistics, and would perhaps like to run an existing route faster. To accomplish this, he would select "View Routes" from the main menu, and would then choose the route he wishes to analyze. From there, he can choose to view his statistics graphically, set a desired pace, race his personal best on the given route, or simply run the selected route again.

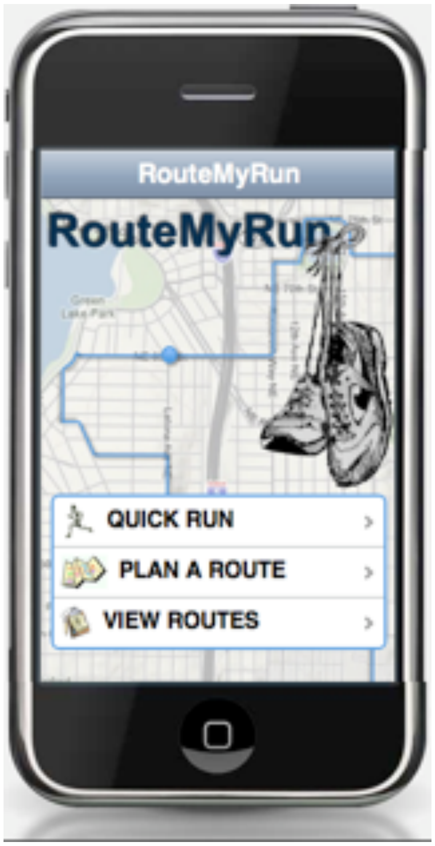


Figure 5

Scenarios

A. Easy: *Go on a quick run and record your statistics.*

Michael is a French exchange student living near the UW campus. He wants to go on a short run through the U-District. He is an avid runner and knows the route he would like to run. He likes to see his improvement over time, so he wishes to record the pace and the distance of this run. He presses the "Quick Start" button on the home screen (see Figure 5). A map of his current location is displayed and he presses the "Begin tracking" button to begin his run (see Figure 6).

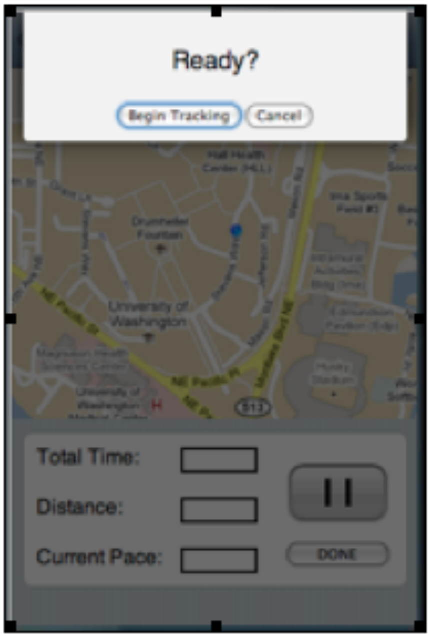
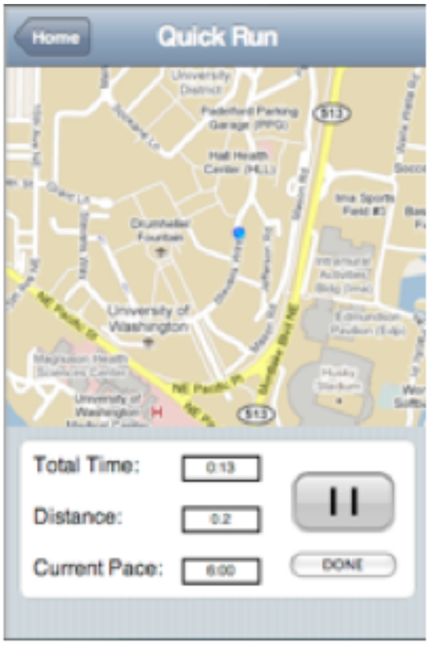

Ready?

Figure 6



Total Time: 0:13

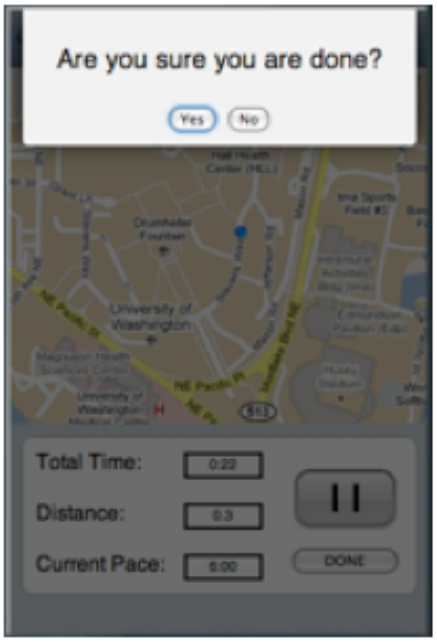
Distance: 0.2

Current Pace: 6:00

Figure 7

While he is on his run (see Figure 7), he comes to a stoplight. He does not want to include this delay in his overall time so he presses the "Pause" button. The light turns green and Michael presses the "Resume" button to continue his tracking.

When he gets back home, he presses the "Done" button and confirms that he is in fact done with this run (see Figure 8). A screen overlay of his run statistics appears to inform him how he did.


Are you sure you are done?

Total Time: 0:22

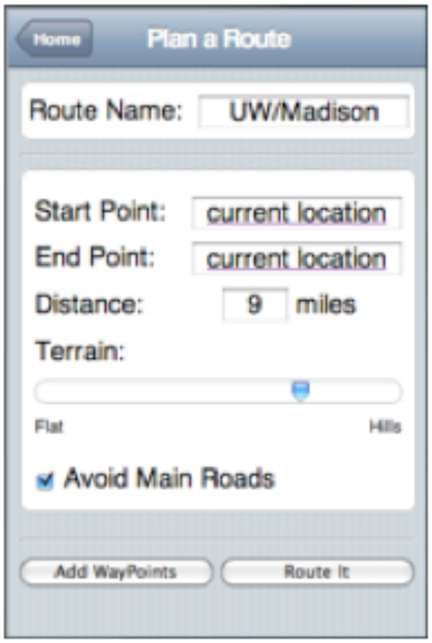
Distance: 0.3

Current Pace: 6:00

Figure 8

B. Medium: *Plan a route, insert waypoints*

Bertha is a stay-at-home mom living in Seattle. She does not have a lot of time today and has hired a babysitter so she can go on a run. She wants to run a new route so she presses the "Plan New Route" icon, which takes her to the corresponding screen (see Figure 9). Here, she verifies that her start and end points are set to her current location and checks the "Avoid Main Roads" box. She decides that she would like to run about seven miles with a moderate amount of hills, so she enters the appropriate mileage into the distance box and slides the "Terrain" slider to the center.



Route Name: UW/Madison

Start Point: current location

End Point: current location

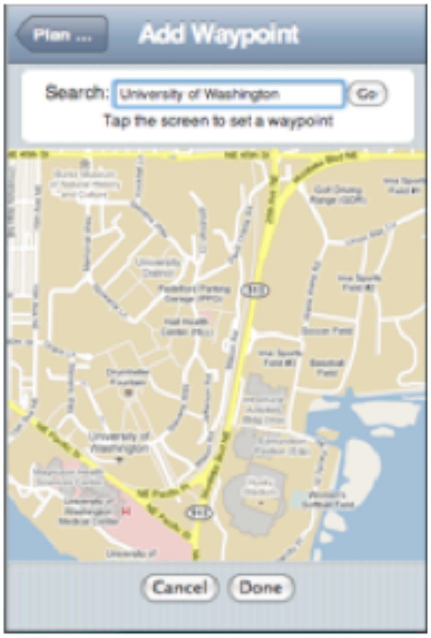
Distance: 9 miles

Terrain:

☑ Avoid Main Roads

Figure 9

She then presses the "Add Waypoints" button in the lower-right corner of the screen (see Figure 10). She slides the map over to Madison Park and taps the screen to set the first waypoint. Next, she taps on the search bar and enters "UW campus" using the pop-up keypad. The map moves over the UW campus and Bertha taps the screen to set another waypoint.

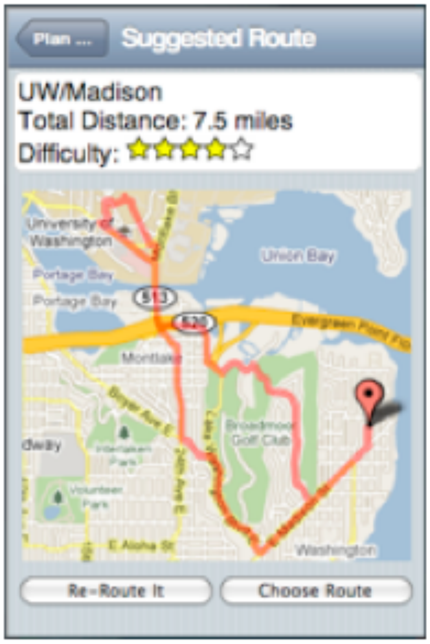


Search: University of Washington

Tap the screen to set a waypoint

Figure 10

After tapping the "Done" button on this screen, the "Plan New Route" screen reappears, now indicating that two waypoints have been set. She then taps on "Route It" and moments later a sample route and its statistics are displayed (see Figure 11). She decides there are too many hills in the proposed route, so she taps "Back" and moves the terrain slider further to the left. Once again, she hits "Route It!"



UW/Madison

Total Distance: 7.5 miles

Difficulty: ★★★★☆

Figure 11

She is now satisfied with this route so she taps "Choose Route" followed by "Run Route Now" (see Figure 12) to begin recording her run.

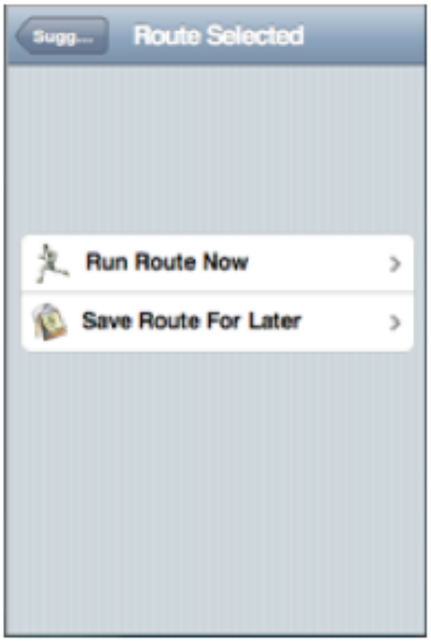
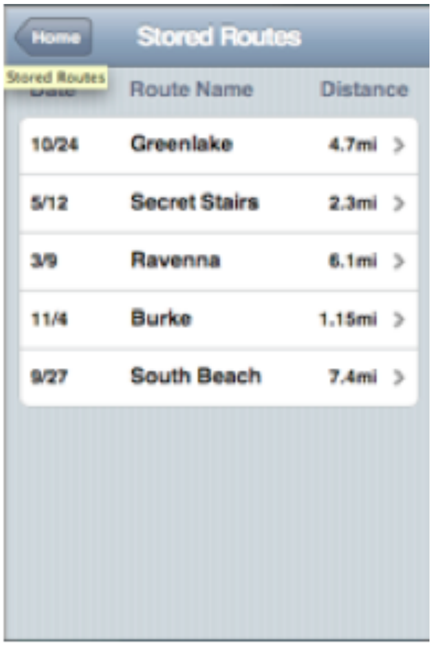


Figure 12

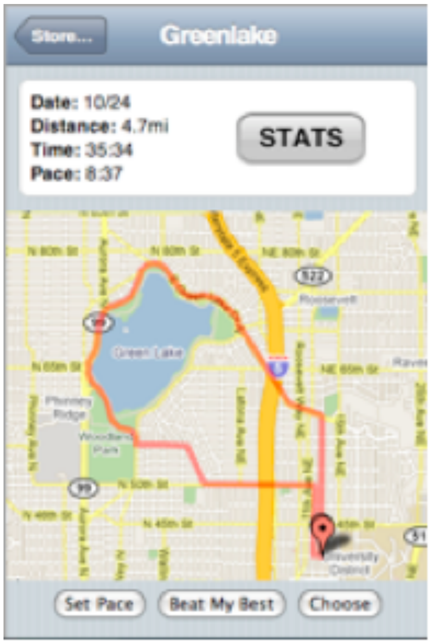
C. Difficult: *Choose a route and race your old PR on that route.*

Davie wants to run his favorite route at a faster pace. He selects the "View Routes" icon, which takes him to a list of all his stored routes (see Figure 13). He scrolls down to one of his favorite runs and taps on it to select it.



Date	Route Name	Distance
10/24	Greenlake	4.7mi
5/12	Secret Stairs	2.3mi
3/9	Ravenna	6.1mi
11/4	Burke	1.15mi
9/27	South Beach	7.4mi

Figure 13

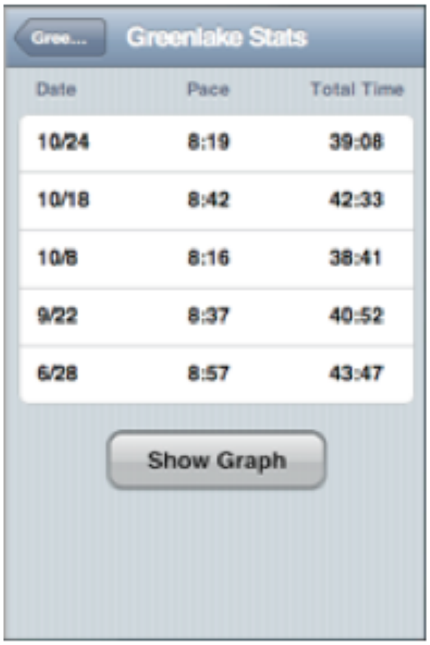


Date: 10/24
Distance: 4.7mi
Time: 35:34
Pace: 8:37

Figure 14

This brings him to a screen that displays a map view of the selected route and some associated statistics (see Figure 14).

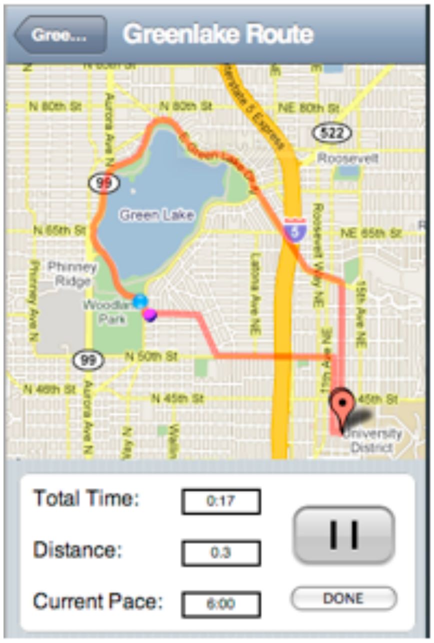
He taps on the "Stats" button so he can view the statistics for all the runs he has done on this route (see Figure 15). He notices that there has not been much improvement in his overall time on this route. He feels confident that he can run this route in a faster overall time today. He taps on the "Back" button, which takes him to this route's main page, and taps on the "Beat My Best" button. A map is displayed with a "Begin Tracking" button overlaid in the middle (see Figure 6).



Date	Pace	Total Time
10/24	8:19	39:08
10/18	8:42	42:33
10/6	8:16	38:41
9/22	8:37	40:52
6/28	8:57	43:47

Figure 15

Once he is ready, he presses this button and begins his run. On the displayed map, there are now two location indicators: one for Davie's current location and the other for his location on his best run of this route (see Figure 16). Throughout the run, Davie is able to quickly see if he is running ahead of his "ghost" runner. At the end of the run, Davie presses the "Done" button and confirms that he is actually done with this run. The screen then displays how much faster or slower he ran the route, and a congratulatory message if appropriate.



Total Time: 0:17

Distance: 0.3

Current Pace: 6:00

Figure 16

Tools Used

Our prototype was constructed using a combination of WebApp.net (found at the following URL: http://webapp-net.com), HTML, JavaScript, and CSS. The icon images on the front page were found using Google Images. We used subversion to maintain code consistency, so that multiple group members could work on development at once.

The chosen tools were helpful in several ways. Specifically, WebApp.net provided a framework in which basic navigation features were already implemented. This easily allowed us to make RouteMyRun look like a legitimate iPhone application. Additionally, WebApp.net is simple to use, so the associated learning curve was relatively short. Using HTML was also helpful, because several members of our group already had experience using this tool.

Although the tools were very effective overall, there were several drawbacks. First, WebApp.net includes very little documentation, so we had to spend time analyzing the code in order to understand how it works. Second, we had to make compromises; since there are already many predefined styles included in WebApp.net, it was difficult to implement certain desired features in a different way. For example, we wanted all the "Back" buttons to look identical in order to increase internal consistency. However, the framework required that the text on the "Back" button's label instead say the title of the previous page. Also, since the framework is all found in one really long HTML file, conflicts among group members were common and had to be resolved. Another difficulty was that the WebApp.net framework only works well in Safari, so we had to make sure that people viewing it knew that, and were forced to use Safari. Finally, with regard to the task scripts shown next to the prototype, it was difficult to make them dynamically change as the client steps through the tasks. This was due to the limitations of the framework we used.

Omitted Features

Various features were omitted in the current version of our prototype. First, the client cannot actually use any of the maps in the traditional iPhone sense; no map scrolling is allowed. Additionally, on the "Plan a Route" screen, it is not possible to set a start or end point. Although it does direct the client to a map view upon tapping the "Start Point" or "End Point" text-box, the client is not actually capable of setting this point to a different location. Without map scrolling enabled, adjusting the start and end location would be impossible.

Another feature that was excluded from our interface was the "Run Stored" feature. At any time throughout the interface execution when the user receives this message, it serves more to end a particular task than anything else. Stored routes are not actually transferred to the "View Routes" database, because no such database exists.

We felt that the features described above would have been too difficult or time-consuming to implement within the context of this project. Moreover, since these features do not appear in our scenarios, we chose to focus on more relevant aspects of the design.

Wizard of Oz Techniques

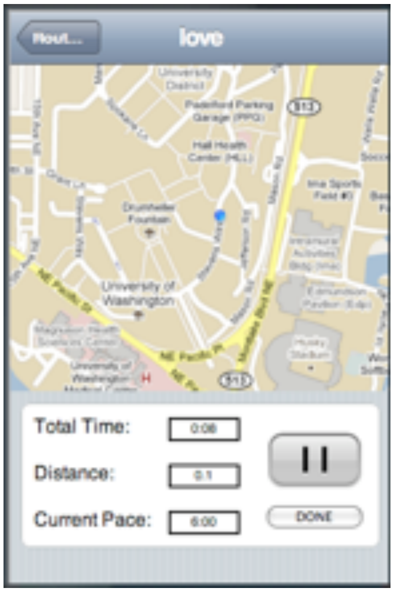
One Wizard of Oz technique that we utilized in our interface concerned the "Add Waypoints" feature. If the client is following our scenario, they will first set a waypoint in Madison Park, then search for the UW Campus in the search bar and set a second waypoint there. However, if the client tries to stray from this pattern (by setting a waypoint in a neighboring area, for example), the same actions that are described above will occur. This part of the interface path is fixed.

Also, to show the user "moving" on the map, we used a timer function in JavaScript that updates the dots on the map to predefined locations along the route. These simulated updates (including location, time, distance and pace) are not accurate, to scale or realistic, but still gives the user the feeling that they are moving via the Wizard of Oz technique.

In the above two examples, our “man-behind-the-curtain” is the functionality provided by the JavaScript code.

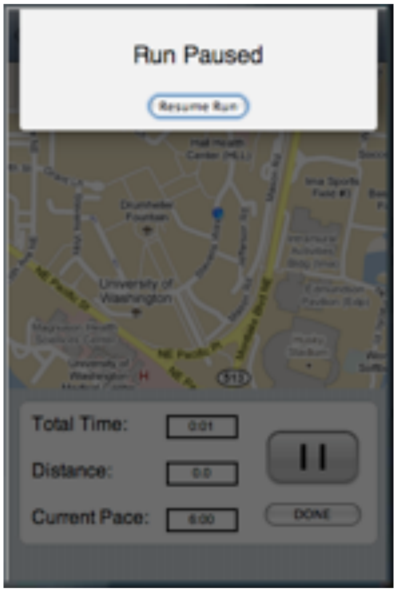
Additionally, since we did not actually implement a database, the list of stored runs is just an imaginary sample. The maps have no relevance to the client's actual current location, nor does the "Search" function allow anything other than searching for one specific predetermined location.

Appendix: Screenshots



Total Time: 0:08
Distance: 0.1
Current Pace: 6:00

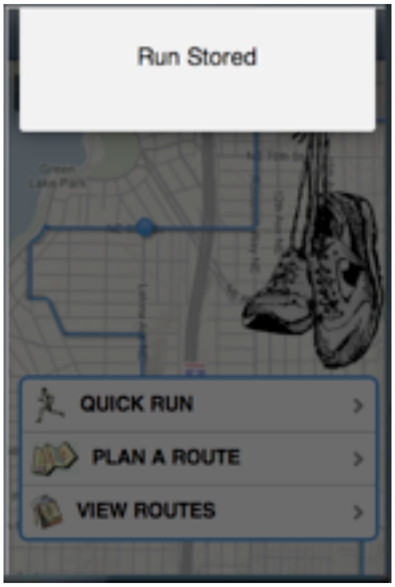
A uniquely named route



Run Paused

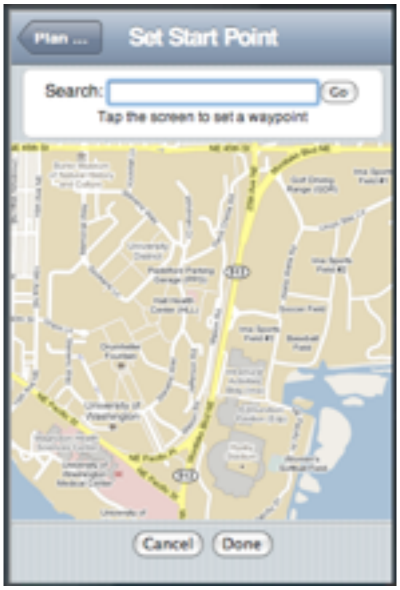
Total Time: 0:01
Distance: 0.0
Current Pace: 6:00

A paused run



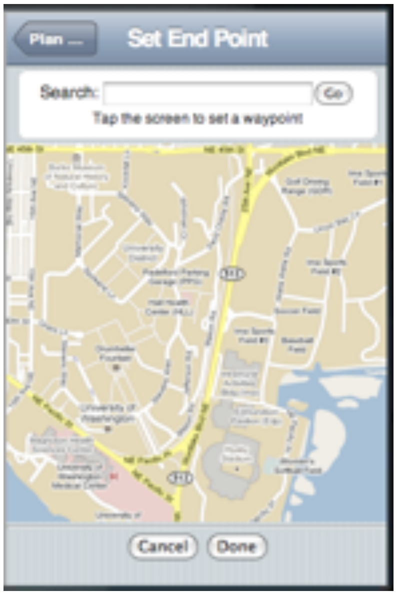
Run Stored

Run is stored after completion



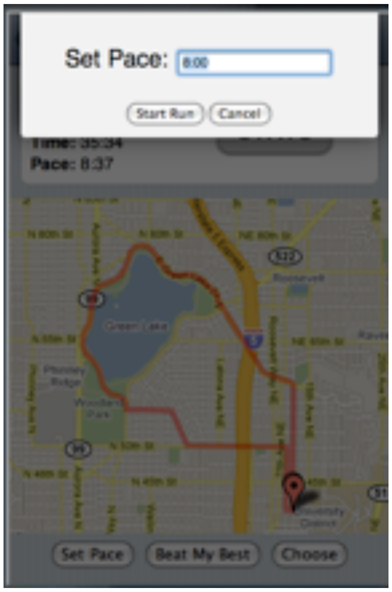
Search:
Tap the screen to set a waypoint

Changing the Starting Position



Search:
Tap the screen to set a waypoint

Changing the Ending Position



Set Pace: 8:00
Time: 35:34
Pace: 8:37

Running against a set pace