

## **@Home - Medium-Fi Report**

Adam - Team Manager

Laila - Development

Gavin - Design

Documentation and User Testing shared

### **Problem and Solution Overview**

The garbage disposal is clogged. You've just chipped a bathroom tile. Your apartment is a mess. The leg on your chair snapped. These are just some of the many problems that we face in the home. While we'd all love to take out the time to make sure our homes are up to spec, the pace of life today and perhaps general laziness holds us back. For many of these problems, we need the expertise of a plumber or electrician and the efficiency of a handyman or cleaner. However, finding someone nearby with the skill, time, and willingness to do the job is time intensive enough to postpone the task to take care of more immediate concerns. @Home is our solution to this problem. It's a platform that helps people access home service providers while helping service providers establish a market for their product. A customer can open the app, select a service, add some relevant details, and be matched with service providers near them. A service provider is notified when they are being matched and provide a price for the customer's review. When the customer requests the service, the provider knows where to go and proceeds to the location in order to provide the service. People need a faster, easier, and more reliable way to solve problems at home. People need @Home.

### **Tasks**

The three tasks we decided on were requesting and finding a service provider (difficult), responding to requests by potential customers (medium), and payment and rating the other party (easy). The first task is specific to customers; they will be the only ones searching for service providers. This is difficult as it involves specifying which type of service provider you are looking for, what kind of work you need done, how long you need that work done in, and finally deciding from a list of service providers. Because of all this, this task spans 4 screens and finally ends in a map showing how far the service provider is from the customer's home.

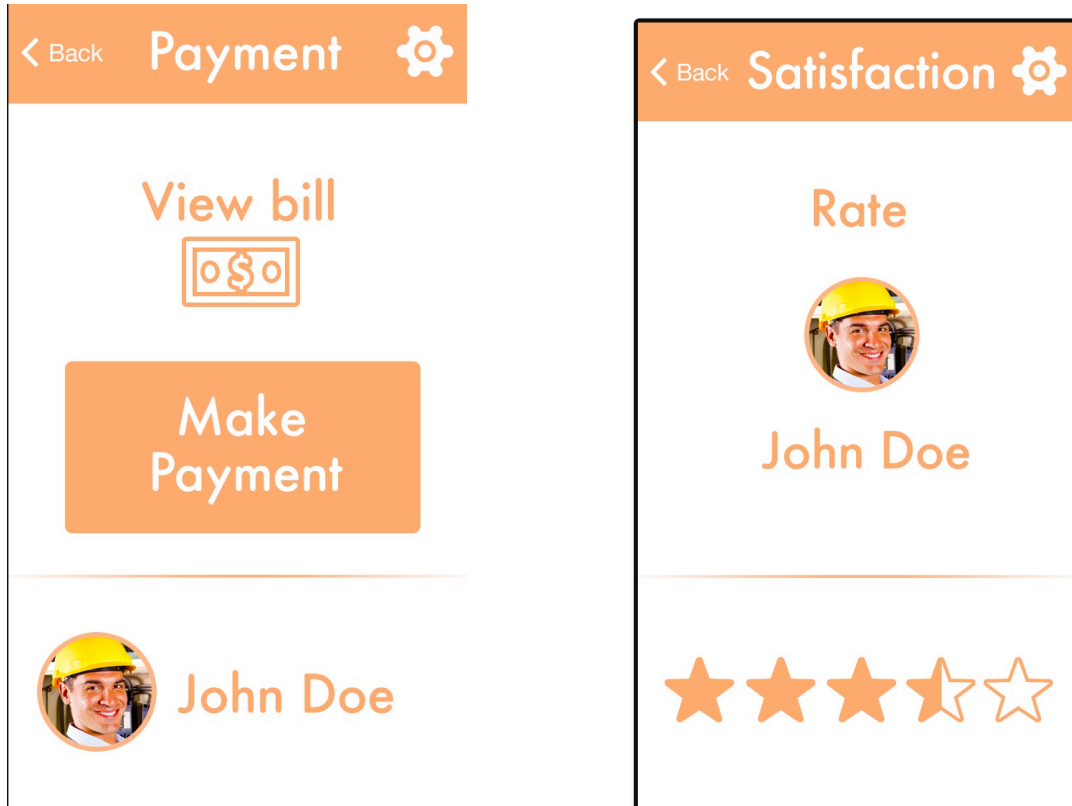
The second task, responding to requests by potential customers, is specific to service providers, and is in a separate prototype altogether. This is medium difficulty as looking forward, the service provider may want to specify characteristics he/she needs in jobs.

Lastly, payment and ratings are common to both parties, as the service provider and customer both have to approve the rating in order for it to be processed. This is because we want to provide valuable feedback to all sides involved; we want customers to know how other customers' experiences have been so they can choose service providers carefully, and vice versa. For this reason, we added the functionality for specifying payment and rating the customer in the service providers' app as well. This is easy as it just takes two taps and not much thought.

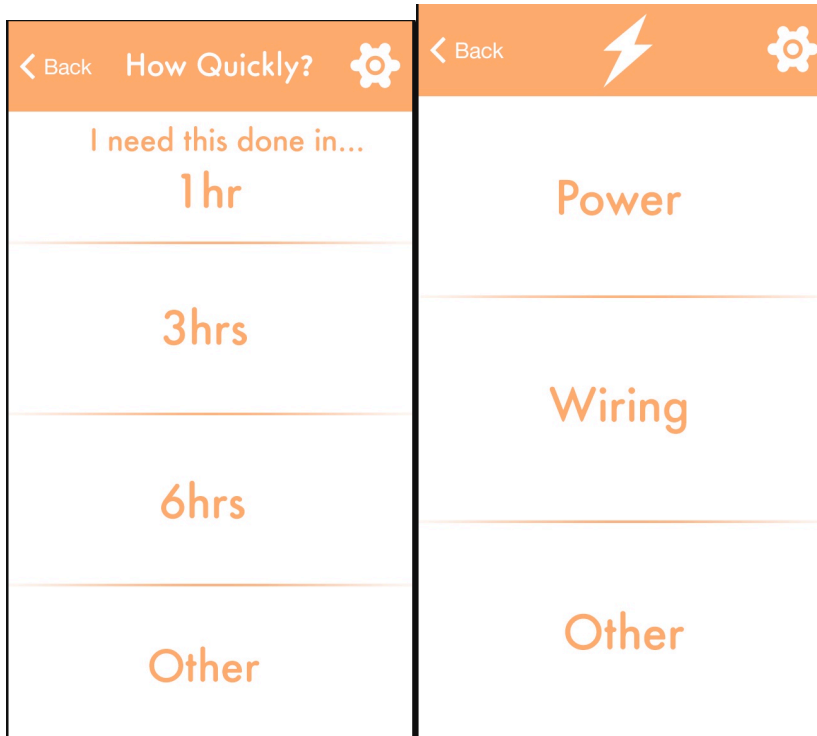
Based on the input we received on our low-fi prototype, we split the payment and rating into two separate steps; this way, the service provider and customer can process the payment while they are still next to each other, but they can still rate each other in private.

### Revised Interface Design

As mentioned above, the payment and rating was split into two separate screens as follows:



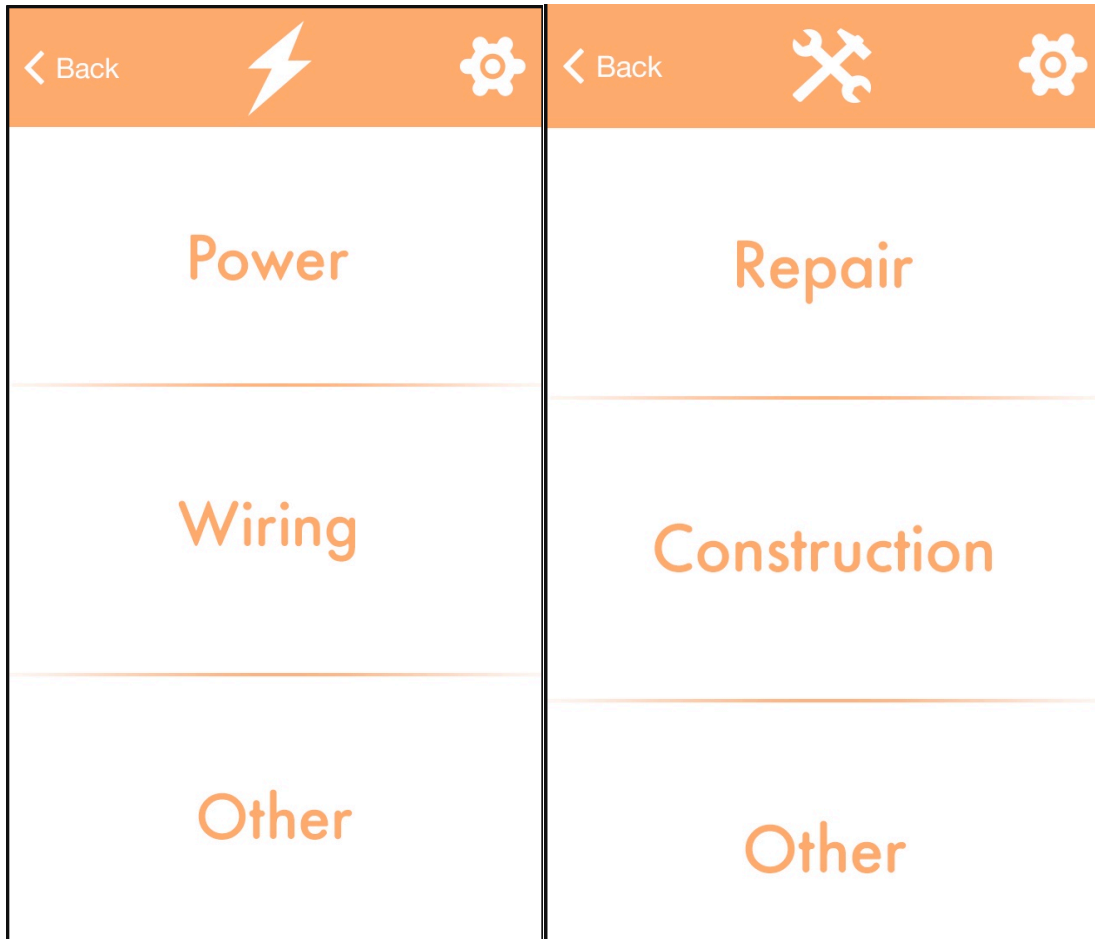
We were also told that the word “custom” was confusing, as people couldn’t understand what was to be customized. In order to make it more clear that we were just giving people the option to specify their problem in their own words, we decided to go with the word “other.” This shows on the screens where the customer chooses the type of service provider he needs, the type of service he/she needs, and how long the customer needs this done in. Thus, it’s a significant change. This shows as follows:



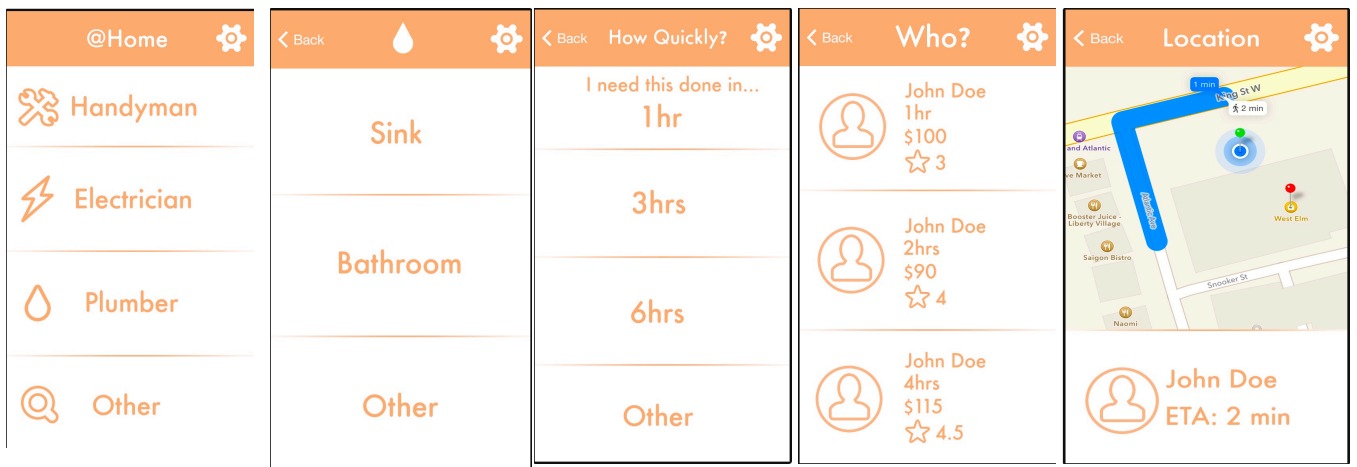
Furthermore, on the page where the customer needs to specify how long they need the task done in, it is now clear that the duration is of how long the customer needs it done in, and not how long the task will take. There was some confusion regarding this at section, so this change in our interface design was very important in order for people to understand the task fully and be able to carry it out properly. The page now looks as follows:



Lastly, we added titles to pages that made it clear what the page was aiming to do as follows:

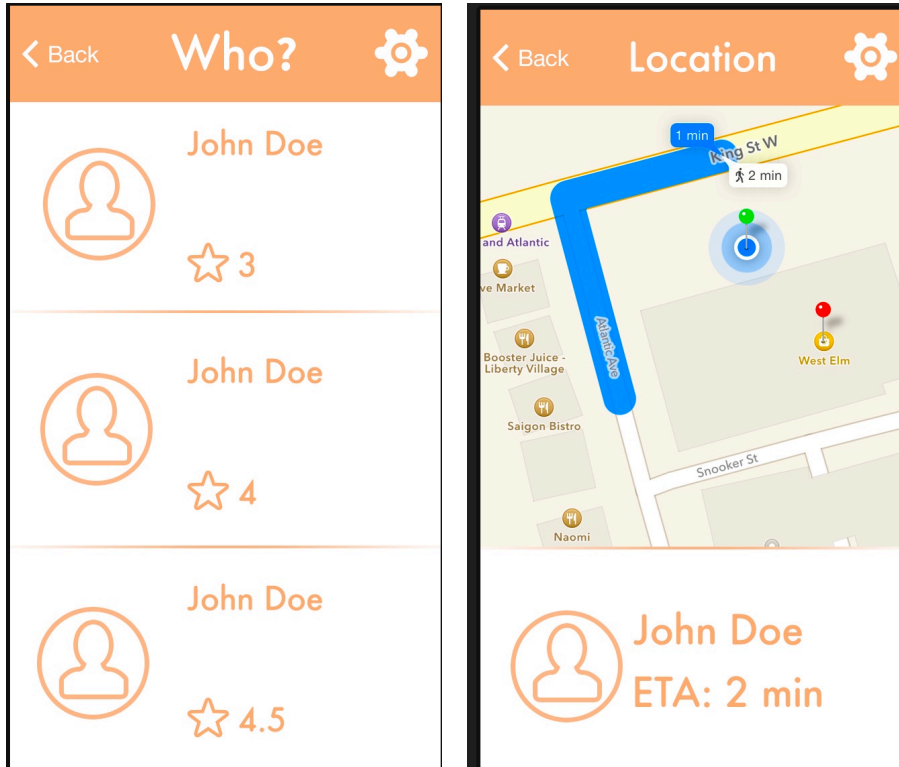


Now, let's imagine I'm a homeowner whose sink pipe burst. I'm inviting people over to my house in a few hours, so I need the pipe fixed within the next hour. I'll call a plumber using @Home with the following steps:



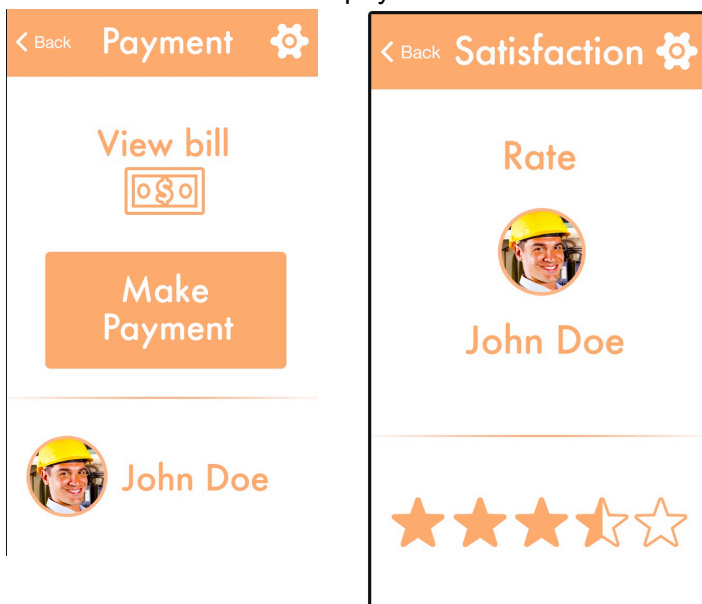
First I'll click plumber, then sink, then 1 hr. I'll be given options for all the plumbers around, and I can choose the one who'll get here the quickest. This demonstrates task 1.

Now, suppose you are a service provider in my vicinity when I send out the request for help. You can accept my request as follows:



Here, suppose I'm the second John Doe with 4 stars. Then you can click on my name, and the request is automatically accepted and you're given my address along with a map of how to get there. This demonstrates task 2.

Finally, for task 3, assume now that you and I are done with the work and only have to process payment now. We can do that as follows, from each of our phones:



## **Prototype Overview**

Link to Prototype (user): <http://marvl.in/4i7ih4>

Link to Prototype (service provider): <http://marvl.in/1hbbi9>

README:

<https://drive.google.com/a/stanford.edu/file/d/0B3nj2jKw2VJYW5NW1VUXMzRmM/view?usp=sharing>

To build this medium fidelity prototype, we used the Marvel web and mobile prototyping application. This platform enabled us to create a usable prototype of @Home with a level of graphical detail that resembles what we hope the eventual product would look and feel like. Marvel allowed us to upload our custom images and graphics and make them serve as the various screens of our application. From the home screen icon to the different tasks, we created the images on Photoshop and Marvel allowed us to link them together to build the prototype. The program let us define certain areas of the screen as buttons, which would then link to the appropriate new screen in our workflow. Marvel also has a swipe transition effect that we used in our prototype between screens. Collectively, these features let us stitch and wireframe parts of our UI together to ultimately build our partly functional prototype.

However, there are limits to the usefulness of such a platform. First, we have not implemented the back end notifications and network management features that are ultimately necessary to bring users and service providers together. To tackle this issue, we have some Wizard of Oz techniques by pre-populating the lists of service providers and users with names. Second, certain pop-up window functionality such as filters for pricing at the request/respond screen and menus for calling a service provider / user on the screen with the map. We plan on implementing this on our next iteration. On the final page of the workflow, we have a map that is simply a standard screen shot from a mobile maps application. There is no person-tracking feature or interactive map in this prototype, which is due to both the lack of an actual @Home community and the limits of Marvel as a development platform.

## **Other Screenshots**