mWork

Re-imagining the Future of Mobile Work for the Masses

Medium-Fi Prototyping Assignment Write Up

Lea Coligado | Design + Development

Andrea Sy | Managment + Design

Allen Yu | Documentation + User Testing

John Yang-Sammataro | Development + Management

December 7, 2014 | CS147 HCI

Problem and Solution Overview

Problem

Poverty and underemployment are two of the biggest global problems in our day and age. One of the starkest examples is what we call the “micro-task gap”: On one side, companies and individuals are willing to pay to complete millions of small tasks - such as determining the content of a picture - that still can only be performed well with human intelligence. On the other side, over 3 billion people live on less than $2.50 a day.[1] These people could make multiples of their current income by completing micro-tasks. However, existing solutions such as Amazon Mechanical Turk and Samasource only allow workers with full computers to bridge this gap and pass over the increasing number of global smartphone users in all levels of society.[2]

Solution

mWork is “micro-tasks for the masses.” Its goal is to allow anyone to work from anywhere by connecting ***clients*** with micro-tasks that require human intelligence to mobile device carrying ***workers*** who complete this simple work. This allows clients to crowdsource important functions and allows workers to earn disposable income in their spare time.

The platform’s initial target worker demographic is underemployed workers in the developed and developing world experienced in mobile applications. Eventually, the application could expand to any potential worker anywhere. Our solution also focuses on the worker micro-task experiences which are severely lacking on mobile devices since existing client interfaces currently meet most task requester needs.

Task and Final Interface Scenarios

We used three main tasks to guide the creation of our medium fidelity prototype broken into simple, moderate, and complex tasks that represent key touch points in the worker task completion cycle:

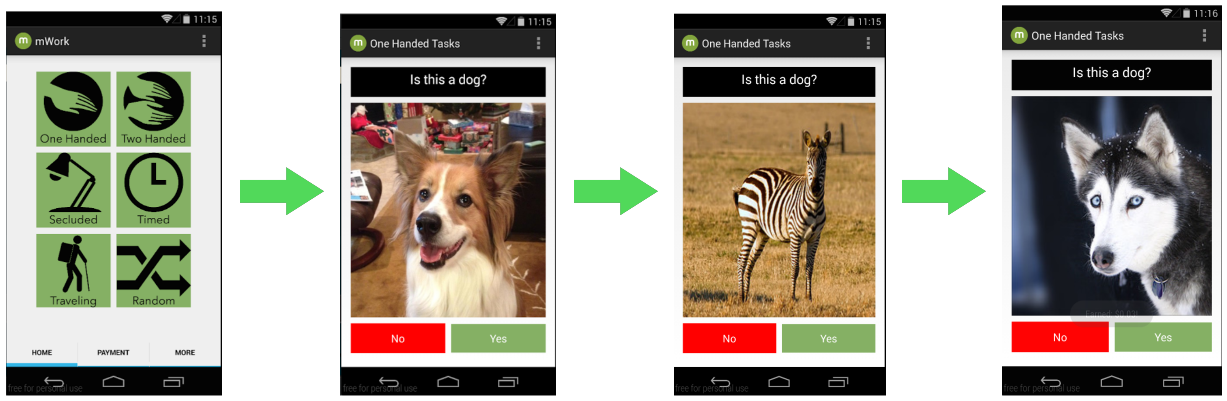
1) **Categorizing content** (*simple*) - This represents a simple task executed by the worker on their mobile interface. A worker is given the task of determining the category for a specific piece of content to answer discrete question. For instance, a user may be displayed a picture and asked to categorize it as a dog or cat. This task was selected as it simulates what most micro-tasks would be like. This is what the user would be doing for the majority of the time when using the app and we wanted to focus on optimizing the user experience in the flow process. Our goal is to make the tasks not seem like monotonous “work” but rather something the user wouldn’t mind or even enjoy doing in their spare time.

[1] Source: https://www.dosomething.org/facts/11-facts-about-global-poverty
[2] Source: http://readwrite.com/2013/05/13/mobile-is-taking-over-the-world

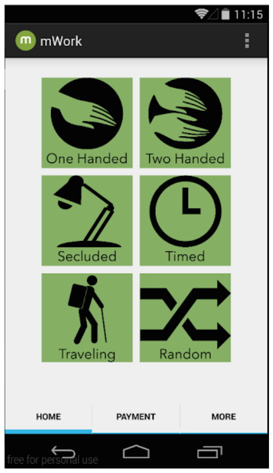
2) **Matching client tasks to workers** (*moderate*) - This tasks represents connecting a worker to a task on their mobile interface based on their current context and desired objectives. There are a variety of micro-tasks that might be requested. A worker needs to navigate the types of tasks available to something appropriate for them to complete. The purpose of this activity is to expand the type of tasks the user can do and determine how the new options affect user experience. Would the user understand what they mean? Would it become too complex? What types of tasks do users prefer? These are all questions we delved into when considering what types of task to offer.

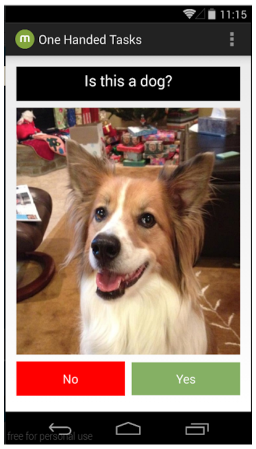
3) **Payments** (*complex*) - This involves a worker using their mobile interface to cash out earnings as a reward for a given task so that they can be compensated for their work. This is an essential part because it is the main incentive component. Our goal is to have a payment process that is simple and straightforward so there would be no hassle when users try to receive their compensation. This function may make or break user experiences and thus extra thought was put into the design of this process.

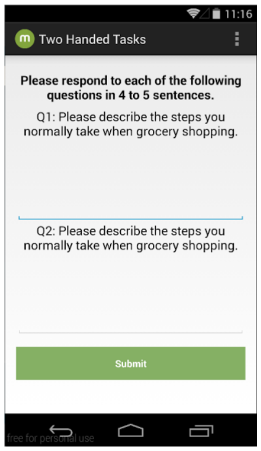
Categorizing Content



Matching Client Tasks to Workers



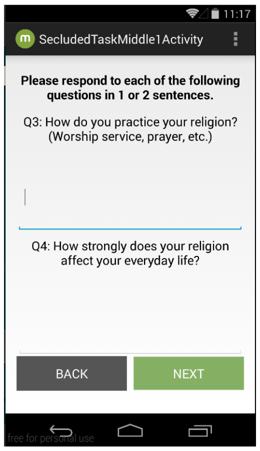




Please respond to each of the following questions in 4 to 5 sentences.

Q1: Please describe the steps you normally take when grocery shopping.

Q2: Please describe the steps you normally take when grocery shopping.



Please respond to each of the following questions in 1 or 2 sentences.

Q3: How do you practice your religion? (Worship service, prayer, etc.)

Q4: How strongly does your religion affect your everyday life?

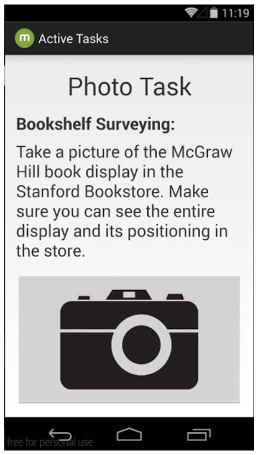
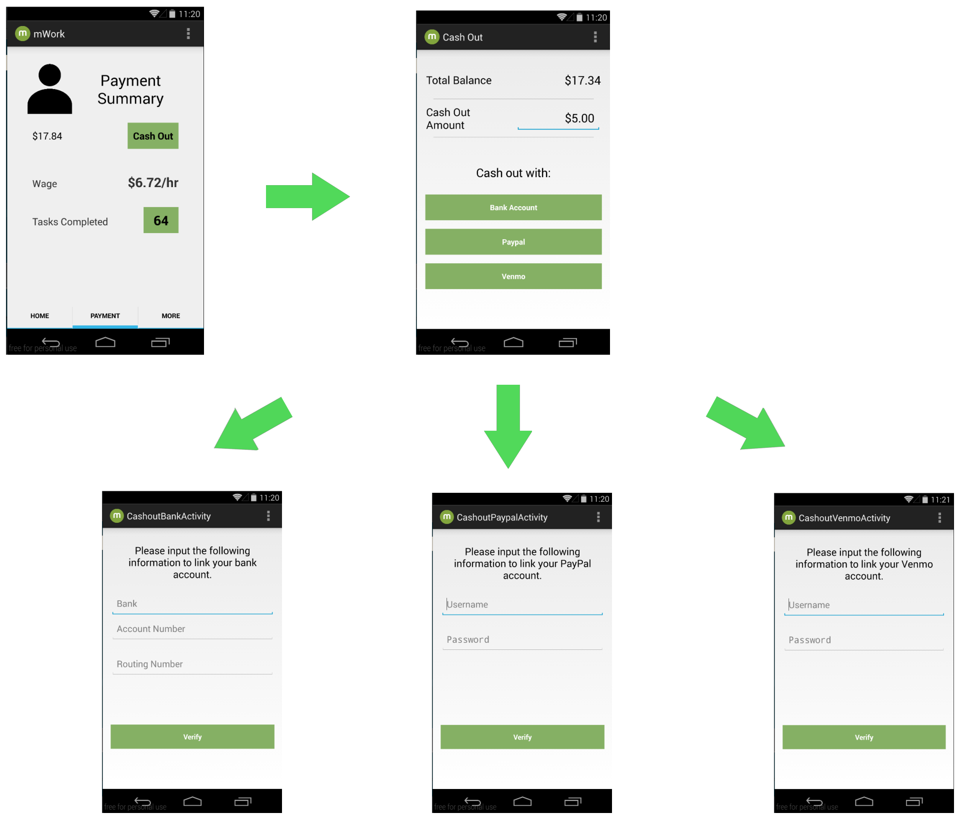


Photo Task

Bookshelf Surveying:

Take a picture of the McGraw Hill book display in the Stanford Bookstore. Make sure you can see the entire display and its positioning in the store.

Payments


mWork

Payment Summary

$17.84 Cash Out

Wage $6.72/hr

Tasks Completed 64

HOME PAYMENT MORE

Cash Out

Total Balance $17.34

Cash Out Amount $5.00

Cash out with:

Bank Account

Paypal

Venmo

CashoutBankActivity

Please input the following information to link your bank account.

Bank

Account Number

Routing Number

Verify

CashoutPaypalActivity

Please input the following information to link your PayPal account.

Username

Password

Verify

CashoutVenmoActivity

Please input the following information to link your Venmo account.

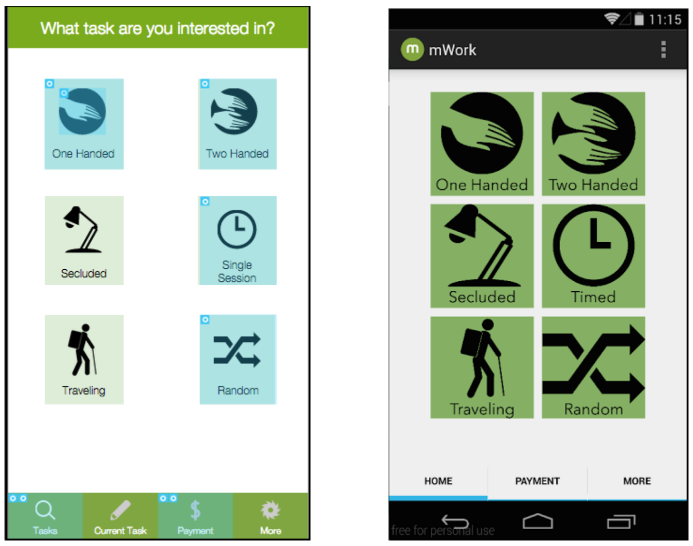
Username

Password

Verify

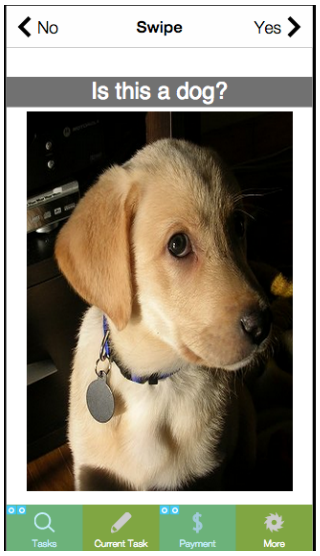
Major Usability Problems Addressed

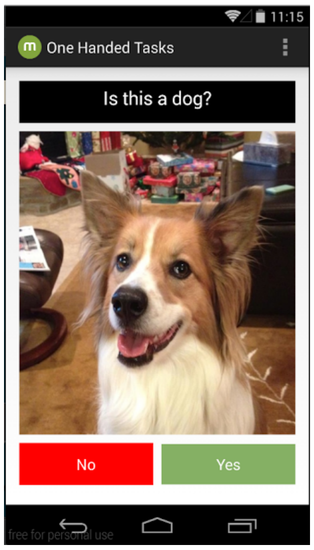
[H2-8 Minimalist Design] [Severity 3]: The icon visibility of the bottom menu buttons is not very good. The gray on green makes everything very hard to read and quickly understand, and it is extremely hard to make out what the icons actually are.



To address the lack of visibility of the bottom menu icons and the ambiguity surrounding their meaning, we changed the color scheme to black text on a white backdrop to increase their visibility. We also scrapped the icons altogether since the buttons' functions (Home, Payment, and More) are evident in themselves.

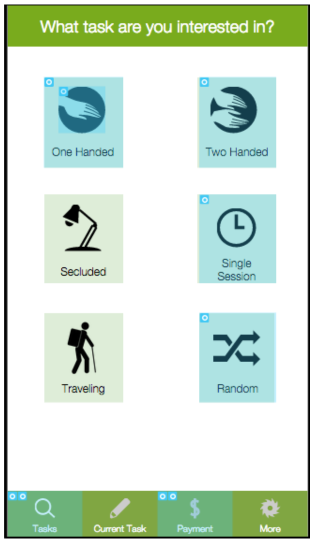
[H2-7 Efficiency of Use] [Severity 2] In the single-handed tasks section, the "Yes" and "No" buttons are unreachable/hard to reach with one hand while holding the phone normally. They are also a long distance away from where a user would normally have their fingers, taking a longer time to get to the button.

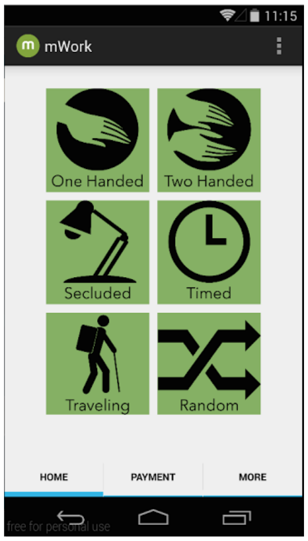




We moved the "Yes/No" selection buttons down the screen to an area where the user can easily access them with his/her thumb. In this way it is a more fluid movement for users to provide answers to one-handed tasks more quickly.

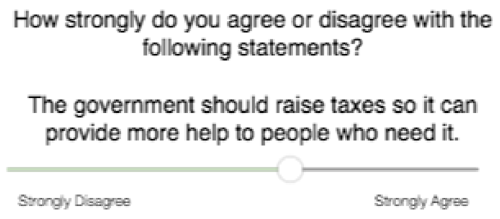
[H2-2 Match Sys & World] [Severity 3] The "Current Task" menu button is very ambiguous in terms of what it does when selected. Are we able to pause a task and return to it where we left off? Or will it provide the user with a randomly selected task?





We opted to scrap this feature entirely within the bottom menu to simplify the options available to the user and avert confusion. Also it is difficult to implement this feature with its dependence on 1) detecting the existence of a current task or multiple tasks and 2) saving the user's progress within one or even multiple tasks.

[H2-2 Match Sys & World] [Severity 4] In the "Single Session" task, both sides of the sliders say "Strongly Agree." This is extremely confusing to the user and does not convey what the question is trying to ask.



How strongly do you agree or disagree with the following statements?

The government should raise taxes so it can provide more help to people who need it.

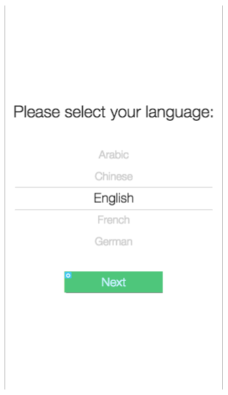
We changed the ends of the sliders to be "Strongly Disagree" and "Strongly Agree" to make their opposing meanings clear.

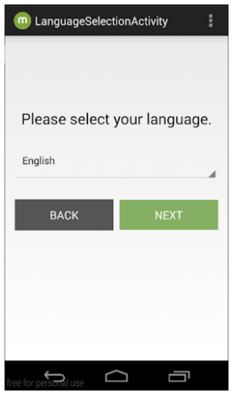
[H2-2 Match Sys & World] [Severity 3] The "New Task" button after finishing a task does not do what the user would expect. Instead of taking the user back to the task selection (main) page in all cases, there are two different behaviors. The first is after doing the Single Handed task, where it provides another picture and then at the end goes back to the login screen. The second is after the Single Session task where it

does what the user would expect.

After a user completes a task s/he is presented with the amount of reward money s/he has earned and may redirect to the task menu to pick a new task by pressing the standardized back button in the Action Bar. This flow is standardized for all types of tasks so that the user may go to a new task in a consistent way.

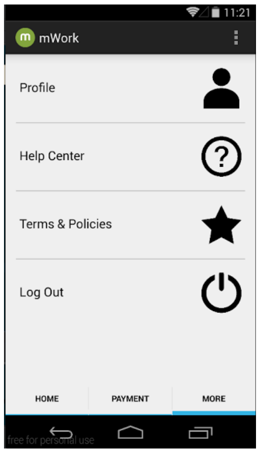
[H2-3 User Control] [Severity 3] Throughout the whole 'log in' or 'sign up' process each with couple of UIs, there is currently no option to return back to the previous page.





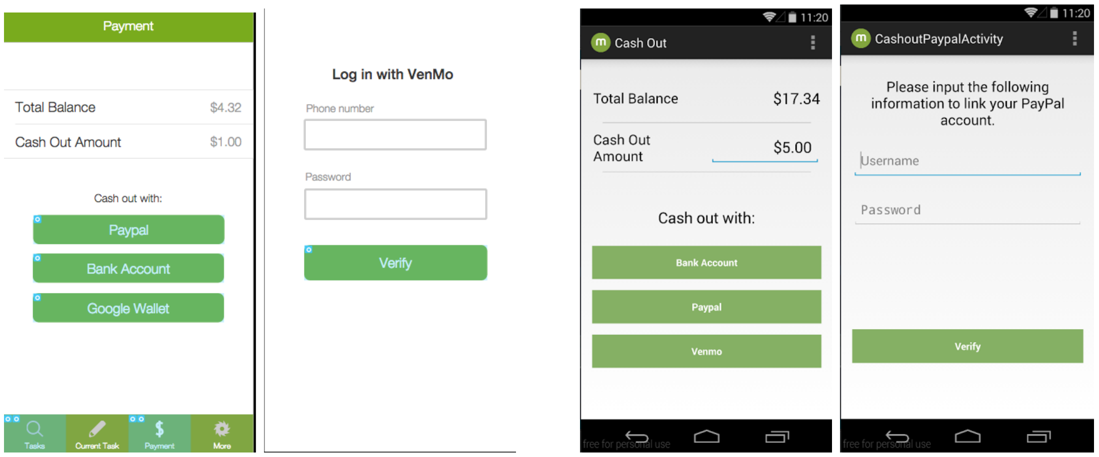
We added a back button at every step of the sign-up process so that users can easily edit any incorrect input they've made.

[H2-3. User Control] [Severity 4] There was no option for users to sign out throughout the whole interface. A user might want to sign out and let the other user sign in and the current interface does not support this option.



We added functionality to the "More" tab of the bottom menu, adding the option for the user to log out of his/her account. In this way, the user can sign out to save data, allow another user to use mWork, etc.

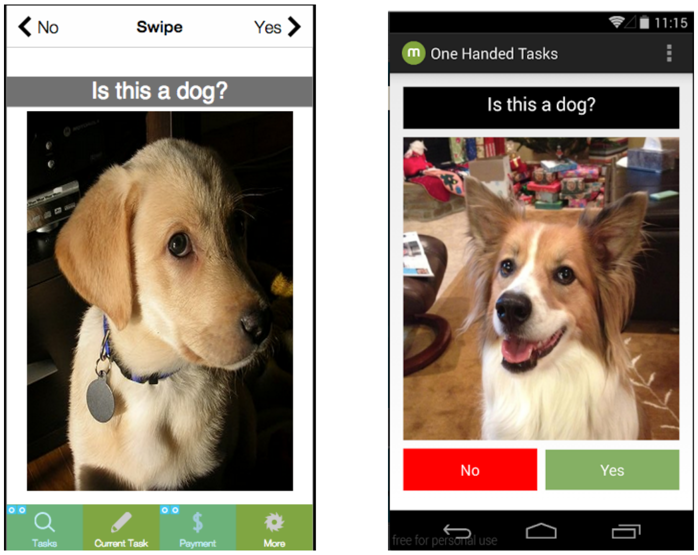
[H2-4. Consistency and Standards] [Severity 4] If a user chooses the 'Paypal' option to cash out, he is directed to the page that asks for his Venmo account. The inconsistency of the wording will confuse the user.



Please input the following information to link your PayPal account.

We fixed this bug so that when the user picks a payment option it redirects to the correct corresponding verification page.

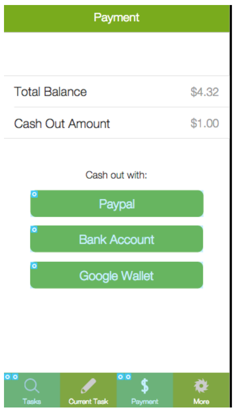
[H2-3. User Control] [Severity 3] Once a user clicks on the 'One Handed' task, there is no option for him to go back or save in the middle of the task. A user might want to save in the middle and come back to the task later on to finish.

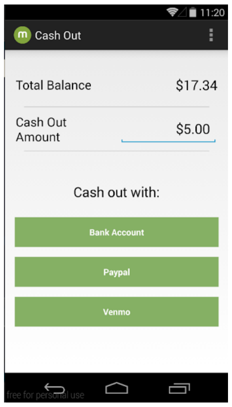


We added a back button so that users in the Action bar so that users can quickly quit the task at hand and go back to the main task menu.

[H2-2. Match Between System and Real World] [Severity 3] Completing the one-handed tasks takes you back to the home screen.
Once a user completes a task s/he is left with a message conveying the reward reaped from it and may go back to the task menu by pressing the back button, described in the number above. In this way, the user can choose another task to tackle from the available range of categories.

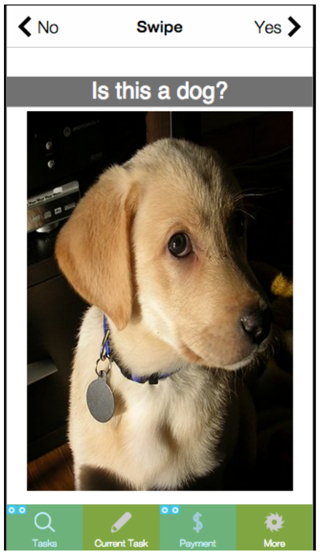
[H2-7 Efficiency of Use] [Severity 3] Though it is intuitive what the difference between “total balance” and “cash out amount” is, it is unclear how to change this value, as tapping it does nothing.

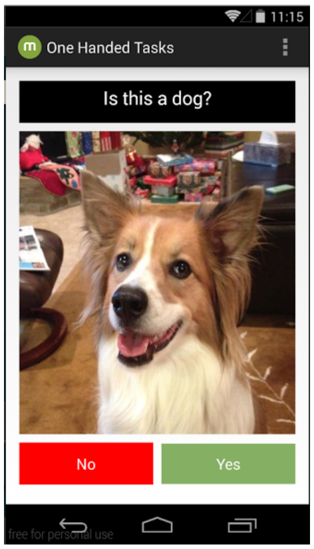




We strive to make it clear that the Cash Out Amount file is editable by underlining the form option in blue and we implement its functionality such that it can be edited. This way the user can actually choose how much s/he wants to cash out, rather than the total balance.

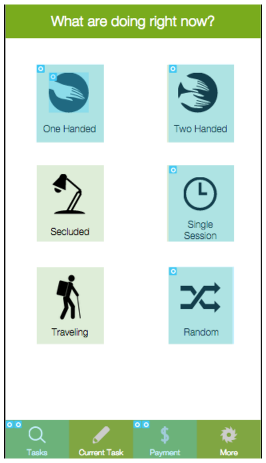
[H2-4. Consistency and Standards] [Severity 3] For the view for one-handed tasks, it is unclear if you are supposed to swipe right and left or tap the "yes" and "no" buttons to answer yes or no.

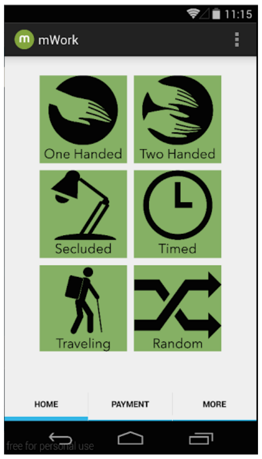




We opted to use buttons for selecting "Yes" or "No," as opposed to the swiping, Tinder-like method. This method is more intuitive to users who are not familiar with swiping to select.

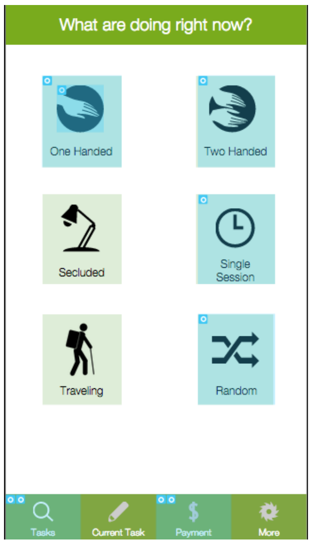
[H2-4. Consistency and standards] [Severity 4] On the main tasks page, the heading says "What are doing right now?". It is unclear what that means on the main screen. This could be extrapolated to situations, where the user could to guess that the system is asking "What are you doing right now" or "What would you like to be doing right now?" or "What's your situation right now?" However, on the whole, users are left with an unclear header message. This is not consistent with keeping this clear to the users.

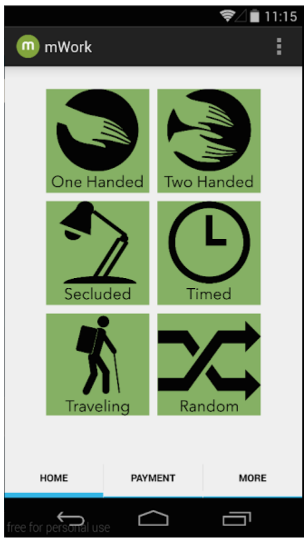




We first decided to change the opening phrase to "What task are you interested in?" then decided to scrap this dialog altogether because the meaning of the task menu is already so straightforward: to choose some type of task. This omission takes the focus off the dialog and places it more on the six options available.

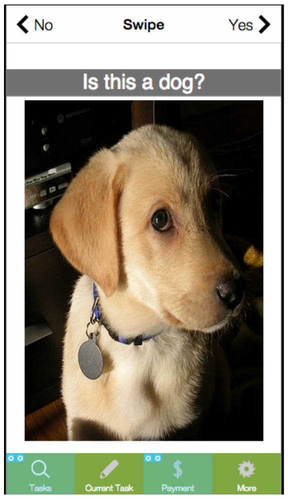
[H2-2. Match between system and the real world, H2-10. Help and documentation] [Severity 4] When on the Tasks page, users could be confused by what the "One Handed", "Two Handed", "Secluded", etc. types of tasks mean. The only one that seemed to universally make sense was the "Random" task type, which meant that selection of the Random task type ensured one of the other five task types was chosen at random. Both the icons and the phrases seem to not generally match each other in a way that makes sense to users.

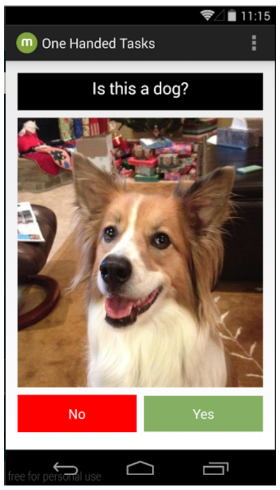


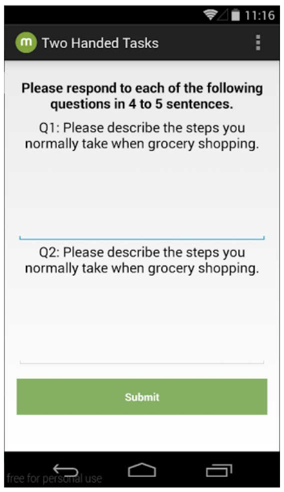


While we agree the meaning of the icons could be ambiguous, we ended up not addressing this violation because we felt these category names most clearly represented the types of tasks they cover. Given more time we would like to add pop-up instructions for first-time users over these icons to provide more information.

[H2-4. Consistency and standards] [Severity 3] Once clicking into the two handed task page, it becomes apparent that users have the option to go "back" to the main page. However, for some reason this option is not available on the one-handed task page, making the interface inconsistent between the types of tasks.



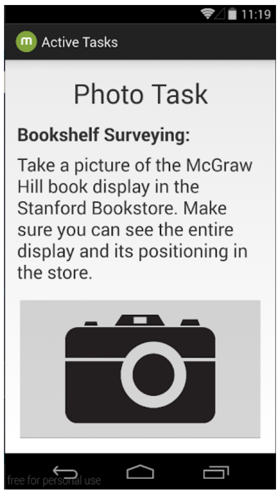




Please respond to each of the following questions in 4 to 5 sentences.

Q1: Please describe the steps you normally take when grocery shopping.

Q2: Please describe the steps you normally take when grocery shopping.



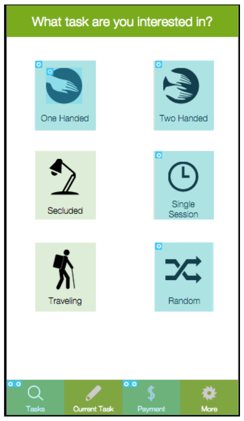
Bookshelf Surveying:

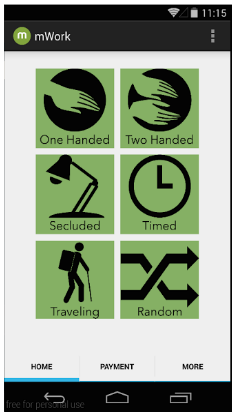
Take a picture of the McGraw Hill book display in the Stanford Bookstore. Make sure you can see the entire display and its positioning in the store.

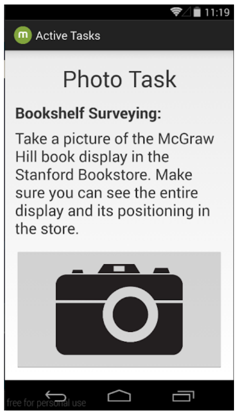
For every task, we provide a button in Android's standardized Action Bar at the top that redirects to the task menu. In this way navigation among all tasks is consistent.

Extra Changes

We implement the "traveling" task category with an example of an on-the-go activity: photo taking. As found in many mechanicalTurk tasks, this activity entails taking a picture of some product or location to assist the paying client. Our implementation provides the user a detailed prompt of what the client wants as well as the option to submit or re-take the photo.

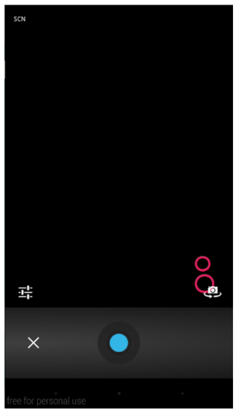


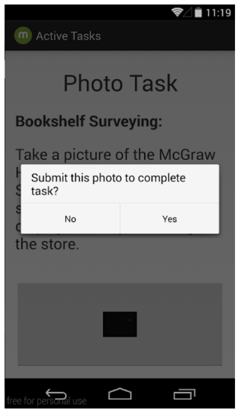




Bookshelf Surveying:

Take a picture of the McGraw Hill book display in the Stanford Bookstore. Make sure you can see the entire display and its positioning in the store.





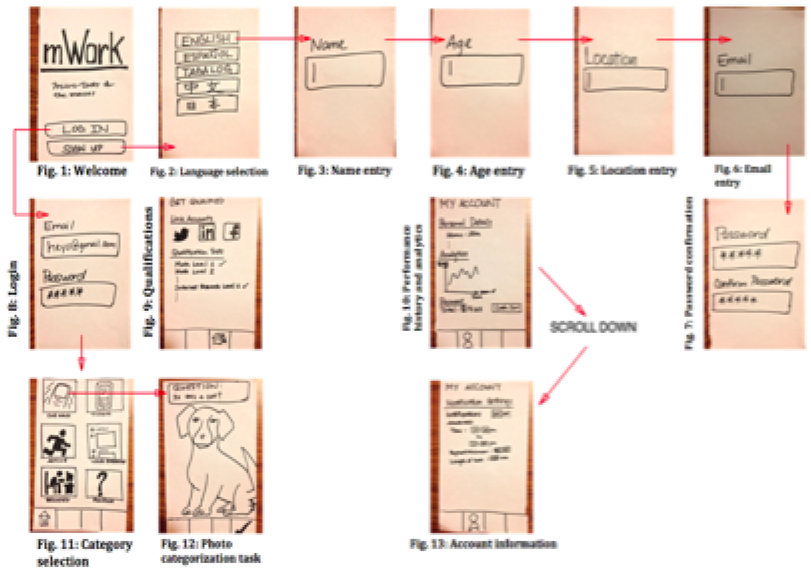
Submit this photo to complete task?

As evident in all the screenshots above, we changed the color scheme to consist of a more muted green and black, as opposed to our original 50 shades of green. In our final design we also adhered to many of the standard Android patterning conventions, including having a consistent Action Bar at the top. This way our entire app looked more professional and consistent with the Android platform.

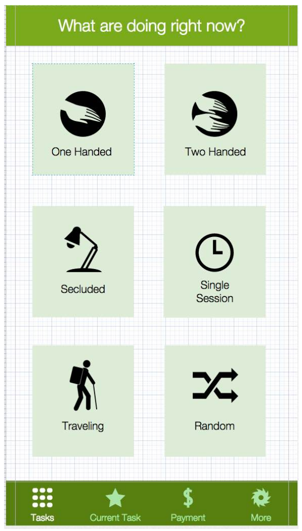
Design Evolution

We began our design process through sketching with pen and paper (as seen below) before moving to our first prototyping tool - POP. POP allowed us to test our preliminary user flows and layouts. After initial user testing, the major design flaw we found was that the different icons we used for task selection was not intuitive. We decided to iterate on this by choosing simpler icons and adding text to the icons.

QUESTION:
Is this a cat?
YES
NO



The next major design iteration was for our medium fi prototype using proto.io. In this iteration, we wanted to focus on the visual and aesthetics of the prototype. This meant using more visuals and less text in all our screens and conducting several changes in the color patches that we used. The major design flaw we found using the heuristic evaluation was that the menu bar at the bottom confused our users. They did not understand the flow between the Task page and the Current Task page.

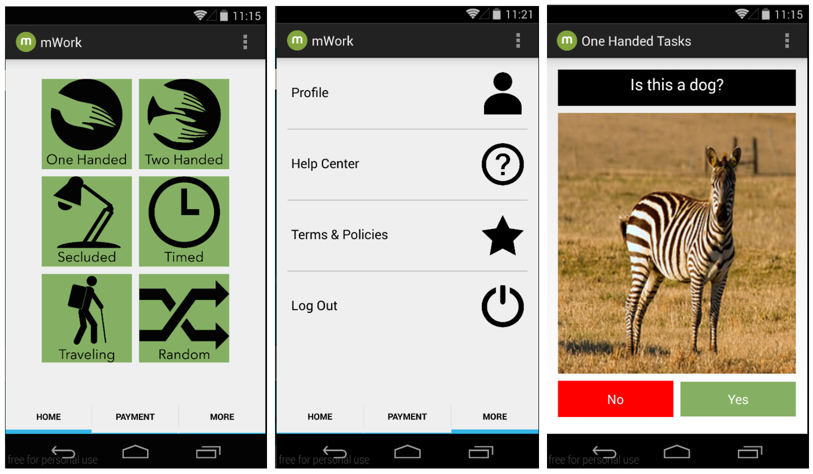


For our final Android implementation, we found that the Android platform has very different different design guidelines compared with iOS. Most of our prototyping tools leaned towards iOS or web development since there is not a very built out Android design ecosystem. This made the final design and implementation on our Android app a bit harder.

After going over the Android design guidelines, we realized we would need to make a number of changes to be consistent with the platform. More specifically, we had to incorporate the Android's back and home buttons in our app as well as use the widgets that were provided and required by the system.

One widget enjoyed using was the Android Toast notifications, which were the popups that notified the user about events. We got a lot of great feedback from our users who enjoyed seeing how much they earned after each task.

In retrospect, we would have explored the android design tools and code earlier in the process to get an idea of how the system worked before designing the medium fi and "higher fi" prototypes. This would have allowed us to incorporate the most up-to-date Android practices and the transition to the high-fi prototype would have been much smoother.



Some good things like the Android Toast notifications that are the popups to notify the user about events

Prototype Implementation

Tools

We used proto.io to sketch out the mockups and programmed the app in Android Studio using Java SDK. The mockups were helpful when designing the Android layout but proto.io's interface was so clunky and inconvenient that it may have been more efficient to simply have made the mockups in Sketch or Publisher. The assets used in our app were made in Sketch which helped us create high quality assets very quickly. Coding in Android SDK had both advantages and disadvantages. The advantage was that the layout part was simple to implement and coding only required knowledge of Java. It was also much easier to upload and set up than iOS simply because of Apple's developer restrictions. The disadvantage with coding in Android is that the code is significantly more complicated and difficult to find the write functions to use. As well, some functions are not backwards compatible and thus some functions used in earlier Android versions did not work on the newer versions. A significant amount of time was spent trying to find the right functions for our purposes. For example, it is extremely difficult take photos in Android and return the image back to the activity. We also had to perform hacks to get around some Android restrictions for purposes such as making toast notifications faster. Android Studio is also still in beta so the IDE is quite buggy. In addition, we used Coolors.co to discover some matching colors for our color scheme.

Wizard of Oz and Hard-Coded Data

Our final app was mainly just front-end design and linking activities together with action items. We had to Wizard of Oz the backend server including storing tasks, login information, and payment information. There was also no client side so all the tasks were pre-made. The monetary amounts were determined by comparing with market price as well. Since there is no client, the actual results from completing tasks are also not stored. There are also only basic validation checks for the two-handed tasks. The login page is simply a validation check to see if the email address contains a “@” symbol and the password has to be longer than a certain number of characters. The monetary amounts in the payment section are hard-coded numbers to imitate the experience for users and the bank account information are also pre-set. The bank information would also not be stored if the user enters a bank account. The settings tab is mostly static to provide the user with an idea of what it would look like. The functionality will be implemented in future improvements.

Future Improvements

There are four areas we would focus on in the future. Due to the time constraints for this project, it was unfeasible for us to implement a complete app. However, we do believe that there is a lot of potential in this idea and may continue to iterate on this in the future. The four areas are listed in the following.

1) **Fully-functional backend** - We would implement a backend server that will store user information (i.e. tasks completed, payment types, current balance), communicate results to the clients, and perform validation checks to ensure users are properly using the app. The full functionality would provide a much more comprehensive experience.

2) **More variety in tasks** - Ideally we would like to have at least 10 tasks in each category so it would simulate a real experience of completing multiple tasks in a span of time. Once the client side is matched, there would usually be hundreds if not thousands of tasks available at once for users to complete (referenced from Mechanical Turk’s traffic). We would focus on creating tasks that are different and interesting so that users don’t feel burdened when doing the tasks. If possible, we would even implement a game-like system where users can advance levels or obtain scores for completing tasks. We may implement tests that users can take in order to perform more advanced (thus better-paying) tasks.

3) **Clients and real tasks** - We would like to reach out to companies and individuals that would be willing to pay for people completing their micro-tasks. This would allow us to have a better understanding of what types of tasks are in most demand and how much clients are willing to pay. We would also implement a client interface for them to develop tasks. We would incorporate design advice and samples for clients so they can tweak their task designs to make them more appealing to users. The interface would also give clients the option to filter users and accept/reject results. Similarly, users would be able to rate clients so there would be a mutual rating system.

4) **Better design and user experience** - Even though we have already iterated our design multiple times, it is by no means complete. There has not been extensive user testing so there is still a lot of potential for improvement. Given the time constraints, we were unable to focus on the design components of the app as much as we would have liked. Our goal would be to design a clean, appealing, and intuitive interface. This is what differentiates us from Mechanical Turk and would be crucial in making our app

more than just a job platform. We are hoping to incorporate this app into people’s lives and become a productive option to fill up spare time.