

Jar: Low-fi Prototype and Test

Michael C. - Team Manager
Evan L. - Software Engineering
Tessera C. - Usability Testing
David M. - Design

Introduction

We're all tired of living with people who don't clean up after themselves. Use **Jar** to manage household tasks and motivate each other to get things done.

Jar is an app that assists roommates with keeping track of household tasks and managing the assignment of communal chores. It provides an easy-to-use interface and notifies users when the deadline for any of their assigned tasks approaches. For any unfinished tasks, money is withdrawn from that task's owner -- as a result, members of a household are motivated to complete all tasks that contribute to the well-being of their home. In other words, Jar's value proposition is: "Motivating the completion of communal tasks, a dollar at a time."

Concept Sketches

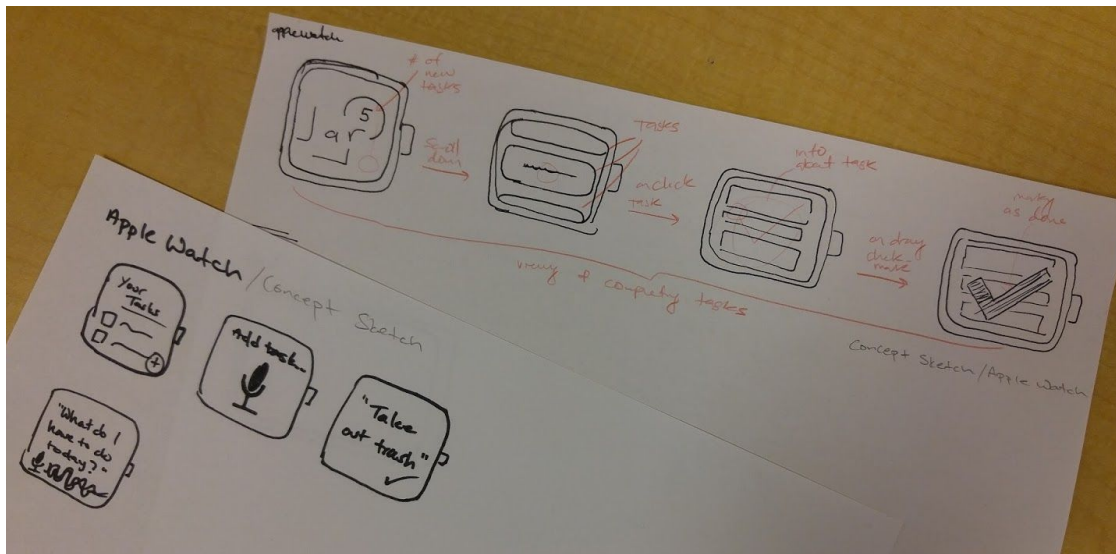


Fig. 1. Apple Watch concept sketches

We introduced the idea of using an Apple Watch because we appreciated the portability of it. These concept sketches show a simple UI design where the main tasks would be checking of items on the task list, or dictating speech to the watch in order to create tasks.

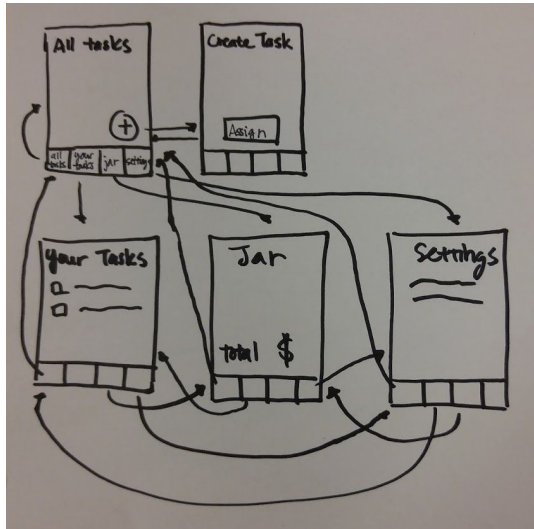


Fig. 2. iPhone concept sketches

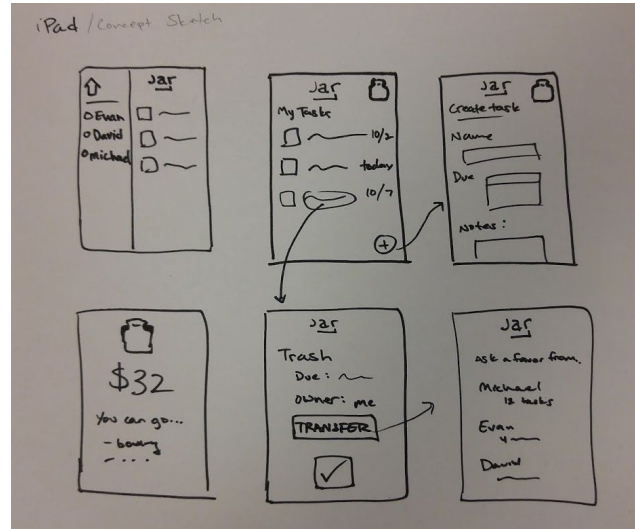


Fig. 3. iPad concept sketches

These iPhone concept sketches focused mainly on having three different tabs to switch between different screens showing the user's tasks, all tasks, and the money jar.

We wanted to explore the possibility of using an iPad, since they are often used more at home. This interface design fits more information per screen, and takes advantage of additional screen real estate by showing different information simultaneously.

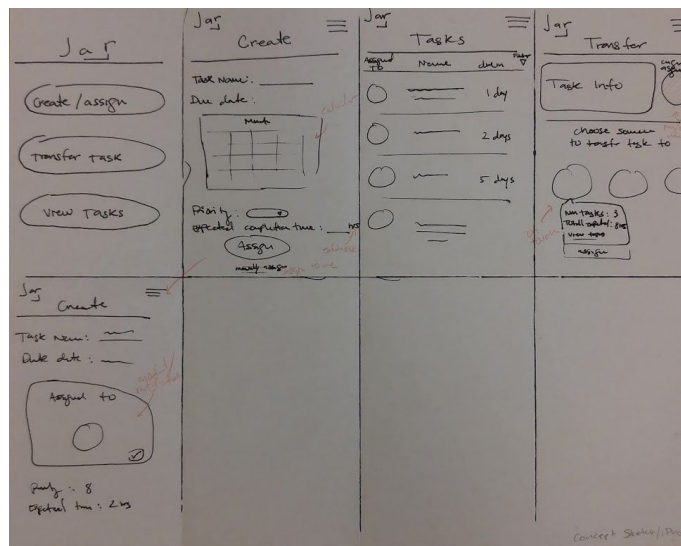
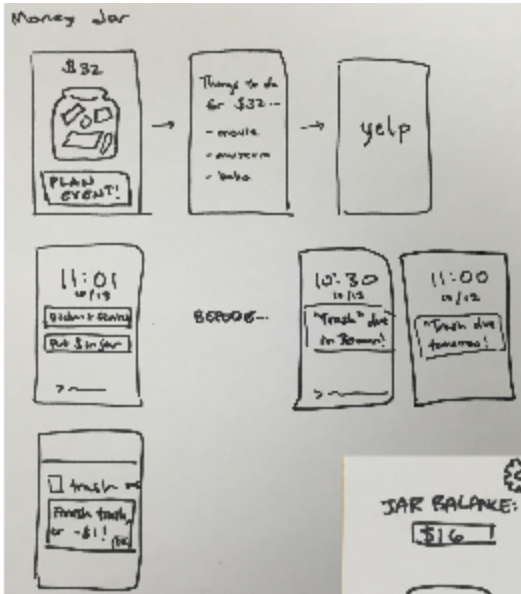


Fig. 4. Additional iPhone concept sketches

These additional iPhone concept sketches fleshed out the idea of using pictorial representations for users, either in the form of profile pictures, or bubbles with users' initials in them in order to easily identify different members of the group within the app.

Top two designs
iPhone



JAR	JAR	JAR
\$602	EVAN \$500 DAVID \$67 TESSERA \$35 MICHAEL \$0	EVAN
EVAN \$500 DAVID \$67 TESSERA \$35 MICHAEL \$0	\$602	Task Date dishes 6/2/16 dishes 6/2/16 practice 2/1/16 clean room 3/1/16 left bed 1/20/16

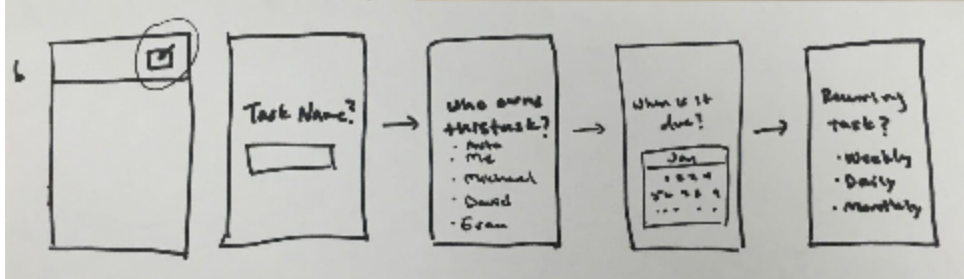
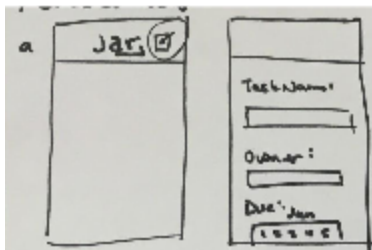
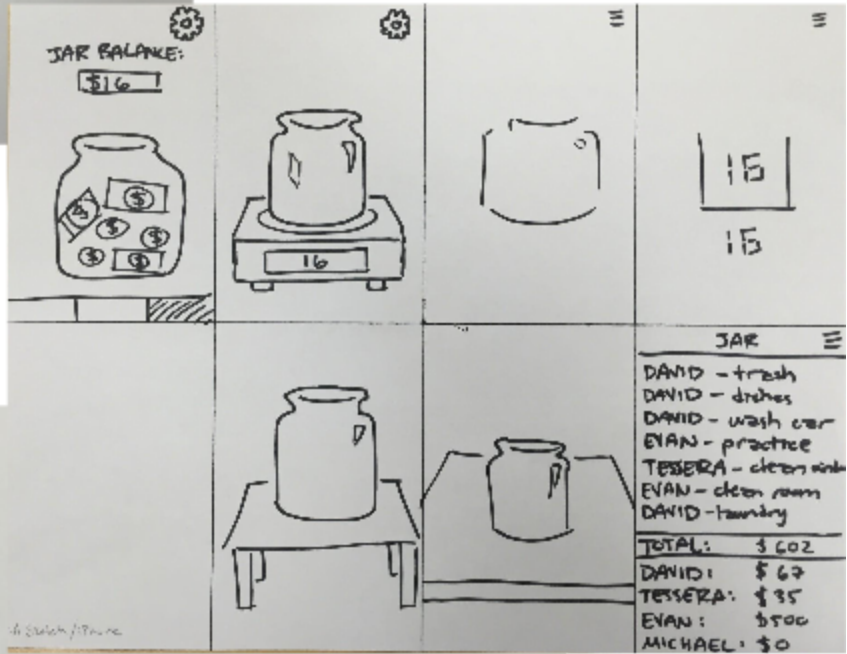


Fig. 5. iPhone UI sketches

iPad

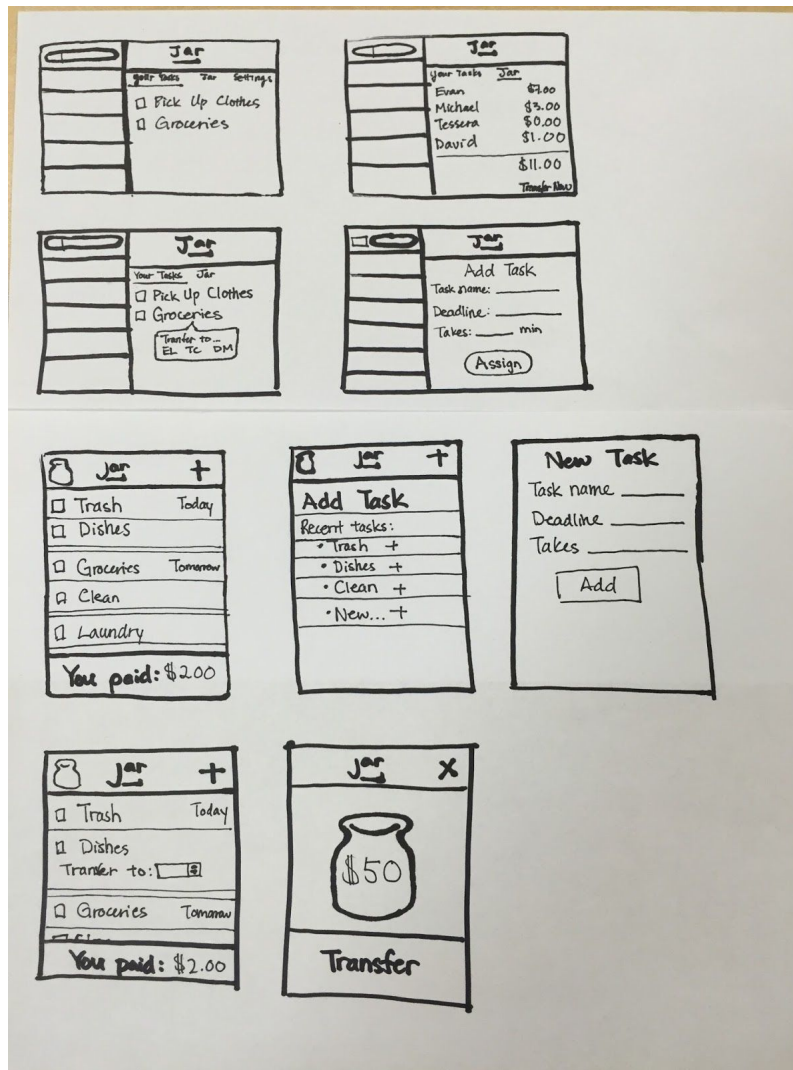


Fig. 6. iPad UI sketches

Our Selected Interface Design

Our iPhone design featured very simple interfaces and task flows. This took advantage of the iPhone's small, portable screen and allowed us to design slimmed-down flows that weren't cluttered or packed with too much information. We also experimented with the skeuomorphic idea of display actual money in a jar. This simple idea came across more naturally on an iPhone screen than on a large iPad screen. The iPhone also made a lot of sense for us to use, given the necessary portability of using the app to pay for some group event. However, the smaller screen is a double-edged sword, because it does significantly decrease screen real estate, and requires information to be spread onto multiple screens. This can create longer task flows, and make the app seem like it is slower.

Our other interface design was for iPad, which has a much larger screen and is inherently easier to read. In our sketches for the iPad, we took advantage of this extra screen real estate in order to simultaneously show multiple pieces of information to the user. This avoids the task of having to repeatedly switch back and forth between screens, since it made sense to always have the user's list of tasks available to them on the screen. However, in trying to design simpler, more bare-bone UIs, we began to find that the iPad screen was quite empty. We wanted our app to feel quick, not overly bloated with information, and portable enough to be used on the go, so our designs for the iPhone stood out to us.

UI Storyboards

Task 1: Create/assign tasks

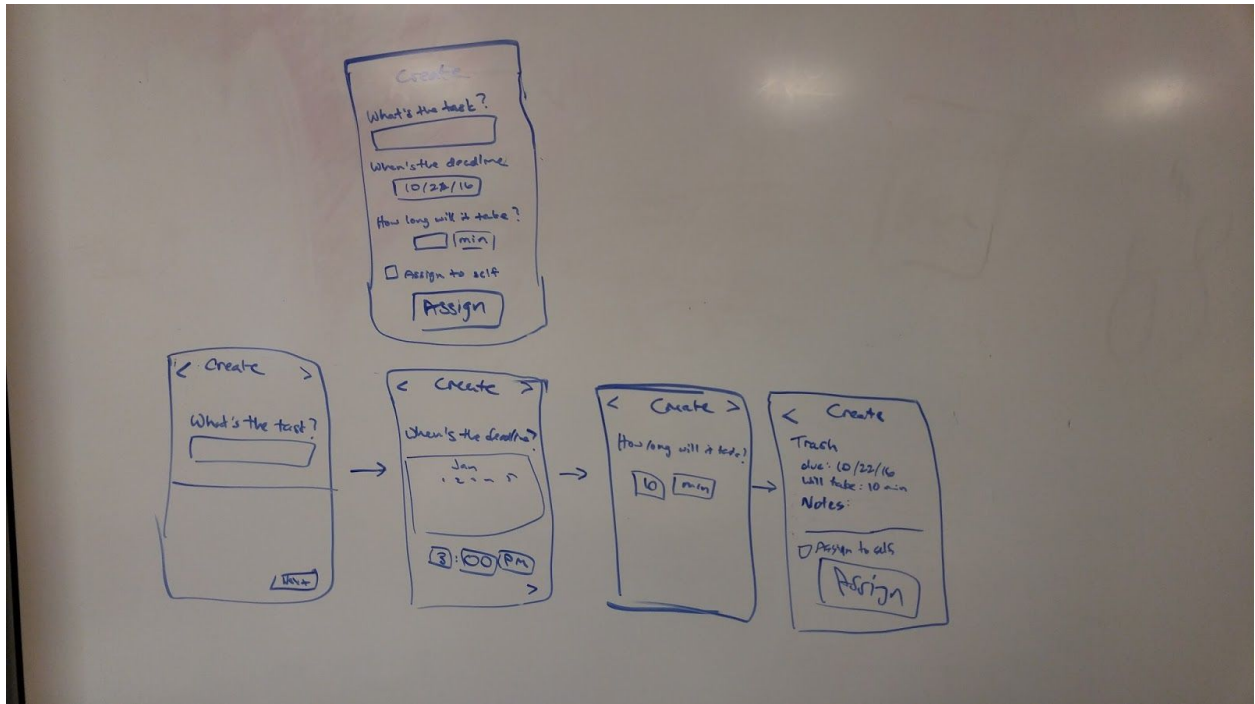


Fig. 7. Task flow for creating/assigning tasks

Task 2: Transfer Task

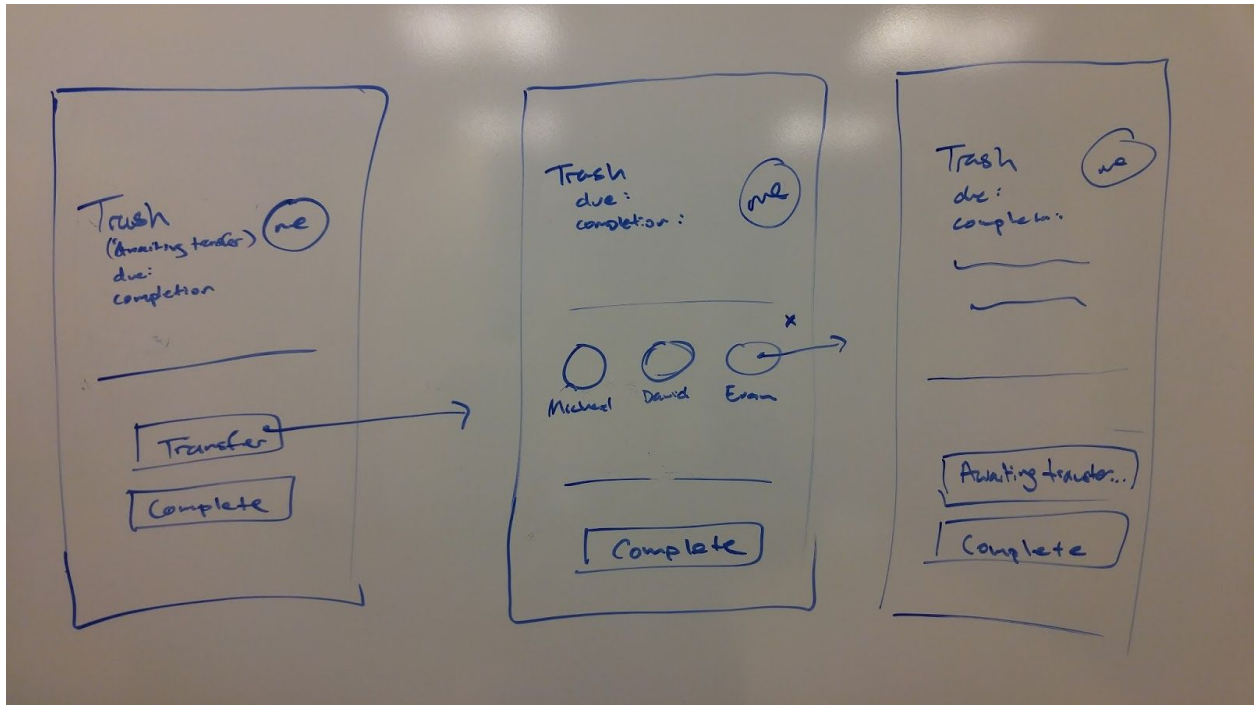


Fig. 8. Task flow for transferring tasks

Task 3: Use money jar

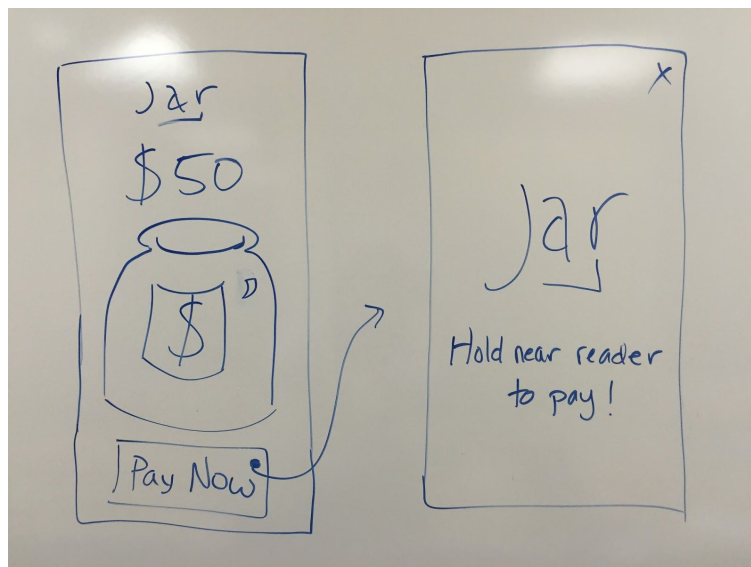


Fig. 9. Money jar task flow

Prototype

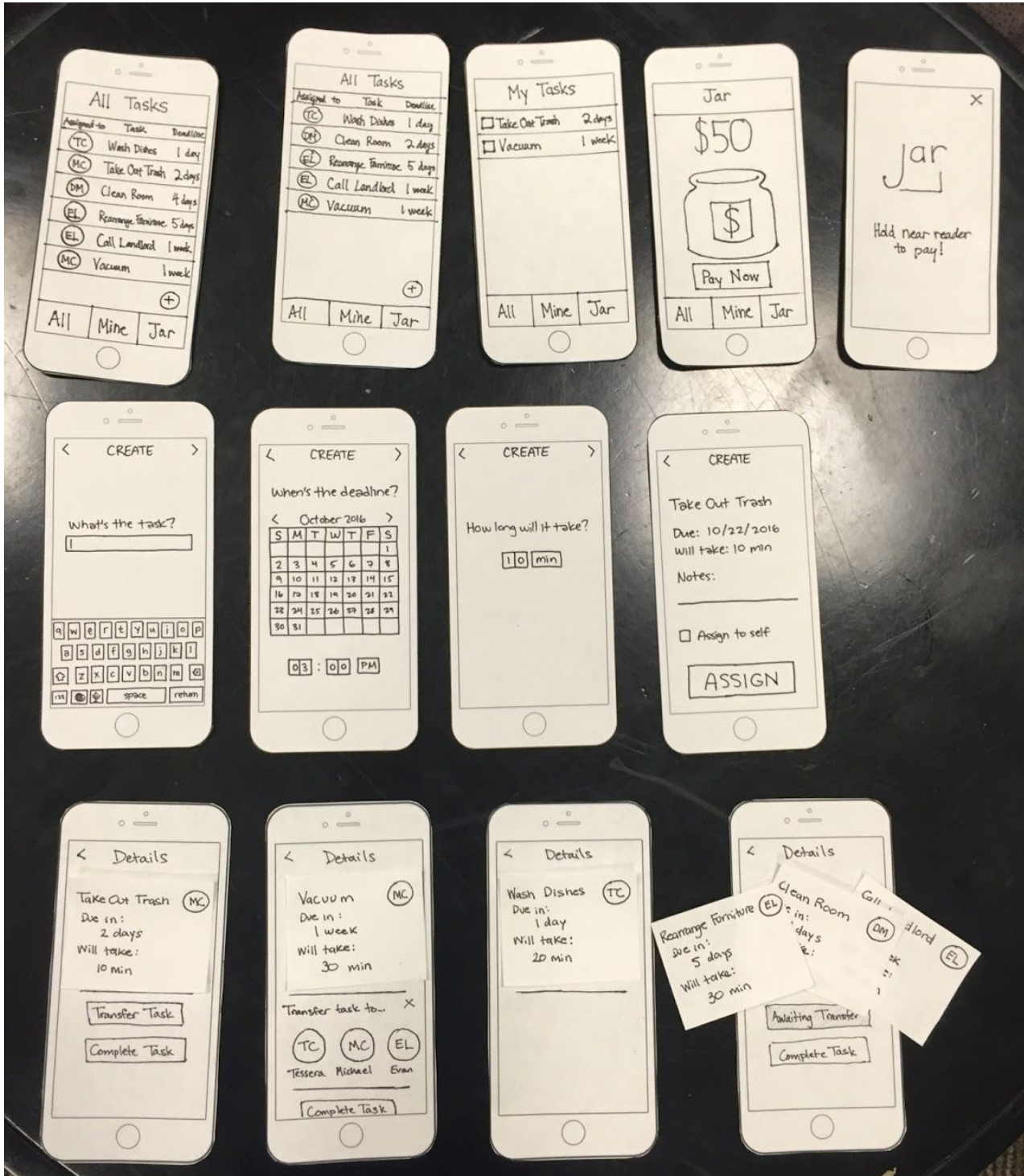


Fig. 10. Complete low-fi prototype

Our design for the low-fi prototype ended up borrowing aspects from multiple concept sketches. While we do envision the final product to have more features, we tried to simplify this prototype as much as possible so that we could focus on our three main tasks. Much of the prototype is inspired by Apple's design paradigms -- the style of the task list and the checkmark were inspired by Apple's Reminders app, the bubbles with people's initials were inspired by iMessage, the calendar and time UI were inspired by iOS's default date/time scrollers, and the Jar e-pay system was inspired by Apple Pay. We believed that drawing inspiration from these applications did not make us uncreative; rather, it allowed us to come up with a design that can be seamlessly integrated into the entire iOS experience.

Method

For our low-fi prototype testing, we recruited three participants -- we will call them participants one, two, and three. Participant one was a 21-year-old female Stanford student, participant two was a 23-year-old female, and participant three was a 22-year-old male. We found participant one by asking students in Old Union whether or not they were interested in helping us test our prototype, and whether or not they had experienced living with roommates. For participant one, we tested in a quiet room with a large table and multiple chairs. We recruited participants two and three at Teaspoon, where we asked random young adults to help us test. Again, we made sure that they had lived with roommates before. For these two participants, we tested our prototype on the tables at the dessert bar. None of our participants were compensated; they all graciously volunteered to donate their time and efforts to the development of our app. Before each participated tested our prototype, we gave them consent forms (refer to appendix) to sign.

We tested three tasks: one simple, one medium, and one complex. Our simple task was adding and assigning a household task, our medium task was transferring one's task to another roommate, and our complex task involved using the money accumulated from uncompleted tasks to pay for something.

As a team of four, we split up the roles as follows -- Tessera was responsible for observing the participants and taking notes on a laptop, David was responsible for timing each task and taking photos, Evan was the computer and managed the low-fi mocks and counted the number of errors per task, and Michael was responsible for greeting participants and facilitating the testing procedure.

For each participant, Michael would greet them and make sure they were comfortable. He would then facilitate our test by reading the script that our team wrote together (refer to appendix). No other team member spoke, and each did their job smoothly and quickly. By splitting up different observational tasks between the team members (Evan and David collected bottom-line data, and Tessera collected process data). For process data, we kept track of any interesting comments participants made, and also noted any difficulties that they may have had while trying to accomplish one of our tasks (refer to appendix for raw notes).

Results & Discussion

The process data from our testing exposed a major flaw in our interface design: when creating a task, users are taken through a task creation flow, where each screen prompted the user to fill out some information (task name, task deadline, and how long the task would take). In order to navigate through screens, we put left and right arrows (< and >, respectively) at the top of each frame. However, all participants made errors here: it looked like they were waiting for the screens to change automatically, and when it didn't, they tried things like swiping the screen and tapping on the keyboard's return key.

Based on the process data alone, we learned that we needed to fix the arrows in our task creation flow. However, the bulk of our suggested improvements came out of our debriefing sessions. To summarize, while none of the participants had to self-assign any tasks, they were all confused by the checkbox that allowed for this. Additionally, participants were confused about how transferred tasks were actually accepted, and also wanted a way to edit tasks after they were created. They also mentioned that the separation of all tasks and their own tasks were not necessary (we made two separate tabs for them). Due to the concerns brought up by the process data and the debriefing, our team decided that we needed to change a few things in our user interface.

First, we need to change the task creation flow. We came up with two solutions: we could either consolidate the entire flow onto a single page form, or we could replace the arrows with "Back" and "Next," and also automatically move the user ahead to the next step in the flow as soon as he finishes a step -- if they need to go back, they can always press the "Back" button. By implementing these changes, we can get rid of the confusion regarding navigating between flows while trying to create a task, and we can also guide and streamline the entire experience, resulting in a faster and smoother user experience. One con with the second solution is that if a user wanted to edit something from an earlier step, they would need to press "Back" multiple times to get there. Whether or not we decide to keep the flow will depend on how much we value the multi-step experience.

Next, we had an easy solution for self-assigned tasks: get rid of them. While users might want to use Jar to manage their own personal tasks, it has nothing to do with the main vision of Jar: to manage *communal* tasks. In addition, having this functionality adds another layer of complexity that our app can do without -- especially for an early prototype.

Additionally, we need a way to edit created tasks, since that was something multiple participants wanted. In order to avoid the unneeded chaos caused by multiple people editing the same tasks, we decided to assign the role of "head of the household" to a single roommate. The head would have editing permissions, and would be able to edit deadlines and the amount of time something takes to complete. In addition, to avoid any unfairness, the role of "head of the household" could be rotated amongst roommates.

Finally, we have to change how we display all tasks. A current solution we have in mind is to display all tasks in chronological, and perhaps highlight a user's own tasks or display them in a different color. We could also offer an option to filter the list so that the user can only see their own tasks. Implementing this change would allow users to more easily see what tasks are coming up and how the scheduling of their tasks fit in with others'.

We also mentioned that participants' concerns included not knowing how transfer requests were accepted. Unfortunately, accepting transfers was not part of the original three tasks, and could not have been tested by the participants. However, it is definitely an important feature, and will definitely be included in the next iteration of our prototype.

Appendix

Word Count: 1768 words

Script:

Intro: We've created an app that manages communal tasks, and our target audience is young adults with roommates. Basically, how our app works is as follows: when something needs to be done, such as taking out the trash, you create a task. During task creation, you also note the deadline (when it needs to be finished by), and an estimate of how long it'll take. Then, you add to the list of all tasks, and the app will take care of assigning it. Since there is a rough estimate of how long each task will take to complete, the app will use this information to evenly distribute household chores. Finally, here's the catch -- in order to ensure that all roommates complete their tasks, each person who fails to complete a task by its deadline is penalized a dollar, which is added to the group "jar." Eventually, if the jar gets filled up with enough money, you and your roommates can treat yourselves to something nice, like buying boba for everyone.

Task 1: In this situation, you notice that the trash can is filling up. It definitely needs to be taken out by the next two days. How would you create and assign this task?

Task 2: It's your turn to take out the trash, but shoot -- the deadline is tomorrow, but you'll be at home with your parents for the weekend. How would you transfer the task?

Task 3: You've been using the app for a while now. You and your roommates are about to pay for boba, and you realize that the jar has enough money to pay for everyone. How would you use the money accumulated in the jar?

Raw Notes:

0 = non issue

1 = cosmetic issue

2 = minor problem

3 = major problem

4 = catastrophe

Positive feedback

I wish...

Participant 1

Create/assign task (took 34 seconds, made 3 errors)

plus sign!

“what are the arrows for”

have next button float with the keyboard?

or label the arrows

same arrows btw nav and calendar is confusing

cool. choose. does this give me hours as well? chill.

can i assign to other people?

“assign to self, then click assign”

“what’s the point of assigning to yourself? so your housemates know you did something?”

under “assign to self”, write “(private task)”

Main task page

“can i also check this checkbox to complete”

“unify the way to complete task”

“just have a filter on the main task page for mine vs all instead of a different tab”

Transfer task (took 13 seconds, 0 errors)

“i click on a task”

“transfer task”

“tap michael. why is the complete task still here?”

awaiting task: “does that mean i’m waiting for him to accept it?”

curious about how allocating works

Money jar task (took 9 seconds, 0 errors)

“wait what’s the jar again?”

michael explains jar concept again

Need a loading screen or popup to explain the jar, perhaps when they first make their account
hold close to radar screen "cool!"
qr code for coupon "chill"
"Does this transfer the money to another account?"

Participant 2

Main task page

"do the days until due stay updated?"
"have a communal pot initially, then take a dollar from that "

Create/assign task (46 seconds, 4 errors)

"use the plus sign!"
have a popup instead of a new screen to make new task
"hit return when done typing"
have everything in one screen
"wait what day is it today?"
swipe?
progress bar
people can cheat the system by lying about how long this will take
how to take into account certain conveniences that would make some tasks easier for some people (e.g. someone works near grocery store)

Transfer task (16 seconds, 0 errors)

"i'd click on the name of the task"
"click on MC"
"i have to go to the task details again?"
"can i just tap on the task and have the names pop up?"
"how do i know if something needs my approval?"
i'll give her a dollar if she does the task for me
do swipe for navigation

Money jar task (11 seconds, 0 errors)

wish list
wow pay now! so convenient
"hold what near reader"
how is the money actually stored in the jar?
"pay now makes it sound like i have to pay the jar"

General feedback

whoever had to pay the most money, call them a sucker!
I want to see who's the most responsible, so give me a breakdown of who did what

"Public shaming is good, things need to get done, it's your job, if you don't do it you suck"

too many screens

can be done so much faster

apps like momentum have everything on one page

click something, have options pop up on the same screen

on home screen, show my tasks on the top

be able to edit due date

charge more for more important tasks

trade-off to no edit: has to be very simple to create

calendar is overkill, you only need relative days from today

for time task will take: have categories: medium, short, long

Participant 3

Create/assign task (29 seconds, 2 errors)

when enter (on the keyboard) is pressed, should go to next screen

would prefer the flow to automatically go forward

be able to select commonly selected tasks

maybe a way to set recurring tasks?

maybe could have a window pop over when creating task

adding to a list usually top right corner

ambiguity with assigning task to self (weird that it's a checkbox)

can you modify anything on confirmation page? it would be annoying to go all the way back

Transfer task (9 seconds, 0 errors)

(no comments)

Money jar task (7 seconds, 0 errors)

maybe cash out instead of pay now

the X on the jar page doesn't match with the rest of the interface

General feedback

clarify with when to penalty

maybe be able to change time it takes / deadline

maybe rotating head of the house who can edit deadlines/time it takes to complete tasks

have a button to toggle between everyone and yours so everything is in chronological order

maybe your tasks can be a different color so you can easily pick it out

Consent Form

The Jar application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of Jar. Data will be collected by interview, observation and questionnaire.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Evan Lin, Tessera Chin, Michael Chung, and David Morales) or with Professor James Landay, the instructor of CS 147:

James A. Landay
CS Department
Stanford University
650-498-8215
landay at cs.stanford.edu

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the Jar experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time.

Name _____

Participant Number _____

Date _____

Signature _____

Witness name _____

Witness signature _____