plantr.

Team

Jen Jianfen Zhang: *project manager; user interface design*

Jory Rice: *interactive prototype*

Paul Bartell: *ideation; writing; prototype integration*

Whitney Schmidt: *ideation; writing; website*

Problem & Solution

Plants can improve mood and health. Many people want to have plants in their home or workplace, but struggle with properly caring for their plants.

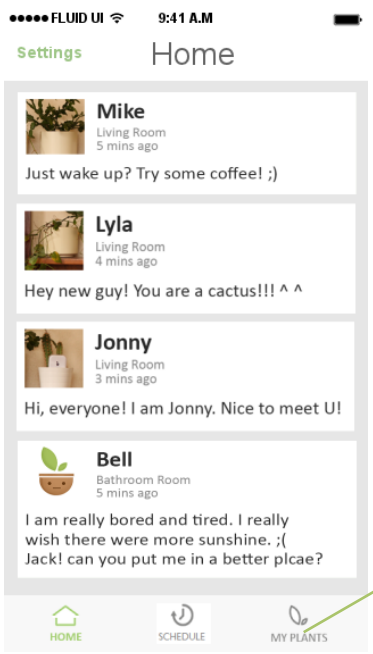
Plantr is a smartphone application that encourages proper plant care by facilitating fun interaction between plants and plant owners. Plantr anthropomorphizes plants, letting them talk to the user like a person would by displaying messages on the owner's phone. Plants within the same household can also 'talk' to each other in entertaining conversations that are shown on the user's smartphone.

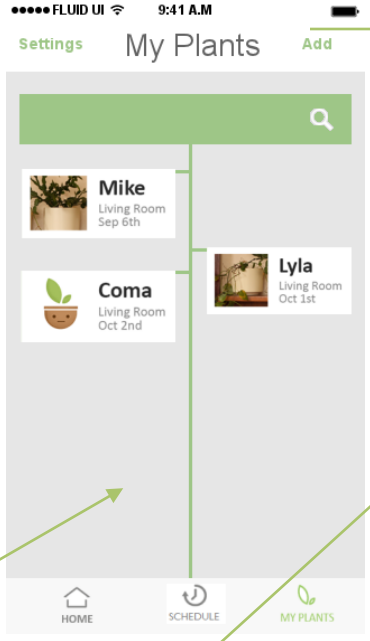
A feedback system between sensors in the plant's soil and the owner's smartphone would allow the plant to 'talk' to its owner when it needs care. Sensors track the plant's basic conditions, like temperature and soil moisture. Users can view their plant's basic condition from their phones. Plantr also has a plant care and disease database, and supports contacting nearby nurseries.

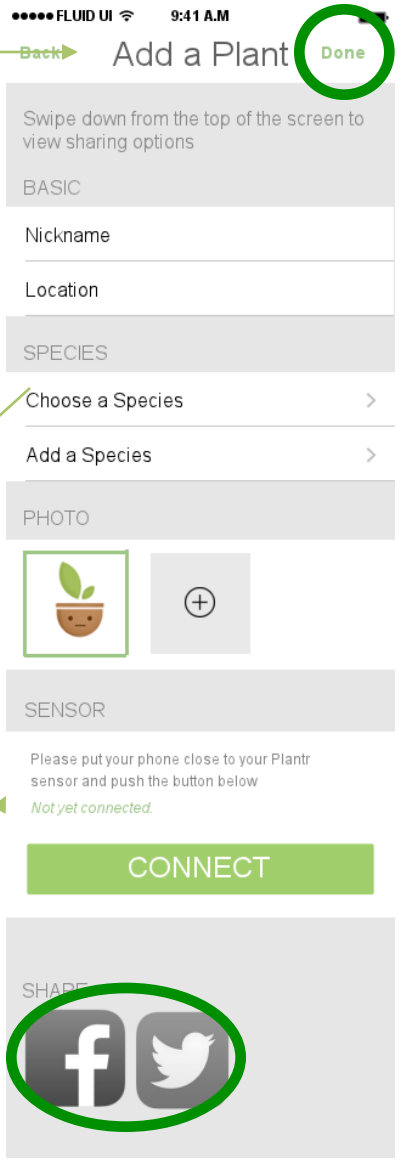
Tasks

difficult task - adding a plant to Plantr

Adding a plant changed from our easiest task to our most difficult task. In user testing, participants struggled with adding a plant because it was a multi-step process that went through several screens. For this task, the user navigates to the MyPlants page and taps the 'Add' button to go to the add a plant screen. On the add a plant screen, the user has to enter basic information about the plant, navigate through the plant selection screen to choose the plant's species, sync their smartphone with the plant's sensor, and choose whether or not to broadcast buying a new plant on social networks. In addition to being complex, this task will not be performed very often, since plants only need to be added to Plantr right after they have been bought.





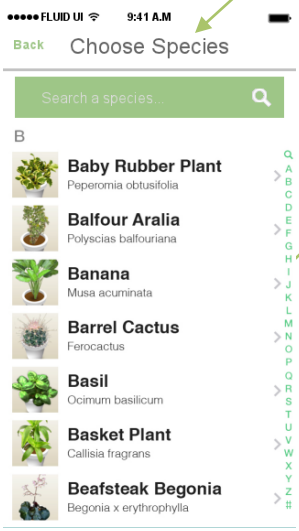


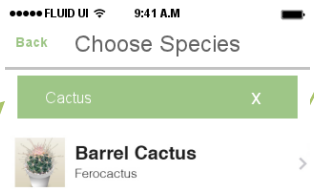
1. First the user selects the My Plants Button.

2. Then “Add”.

3. The user then fills out the Nickname and location of their plant.

4. Then they select “Choose a Species”.





5. After Searching for their plant's species, the user selects the closest species. This takes them back to the Add a Plant screen.

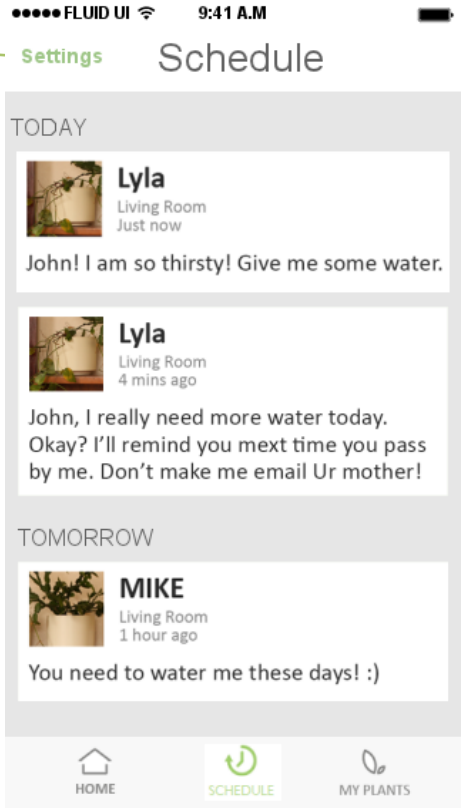
6. Finally, the user chooses whether or not to share that they have a purchased a new plant to Facebook and Twitter.

7. To finish adding the plan, the user selects “Done”.

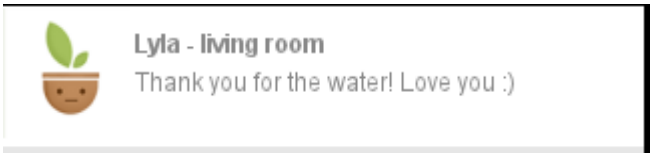
easy task - remembering to care for a plant

We have chosen to keep the same task used in our previous prototypes: helping the user remember to care for a plant. This is an easy task because users are familiar with alerts and reminders on their smartphones. It is also the task that requires the least interaction with Plantr's electronic interface, and has the last number of steps. If a plant needs care, it will send an alert to the user's phone, asking for attention. After the user has viewed the alert, the alert will disappear from the user's phone screen. When the user waters the plant, the plant will send a thank-you alert to the user. Users can also go the MyPlants screen on Plantr to view the current care needs of their plants. Viewing the MyPlants screen is simple: the user must open up Plantr, then tap on the MyPlants button on the main menu at the bottom of the screen.

1. The user receives a notification that Lyla is in need of water



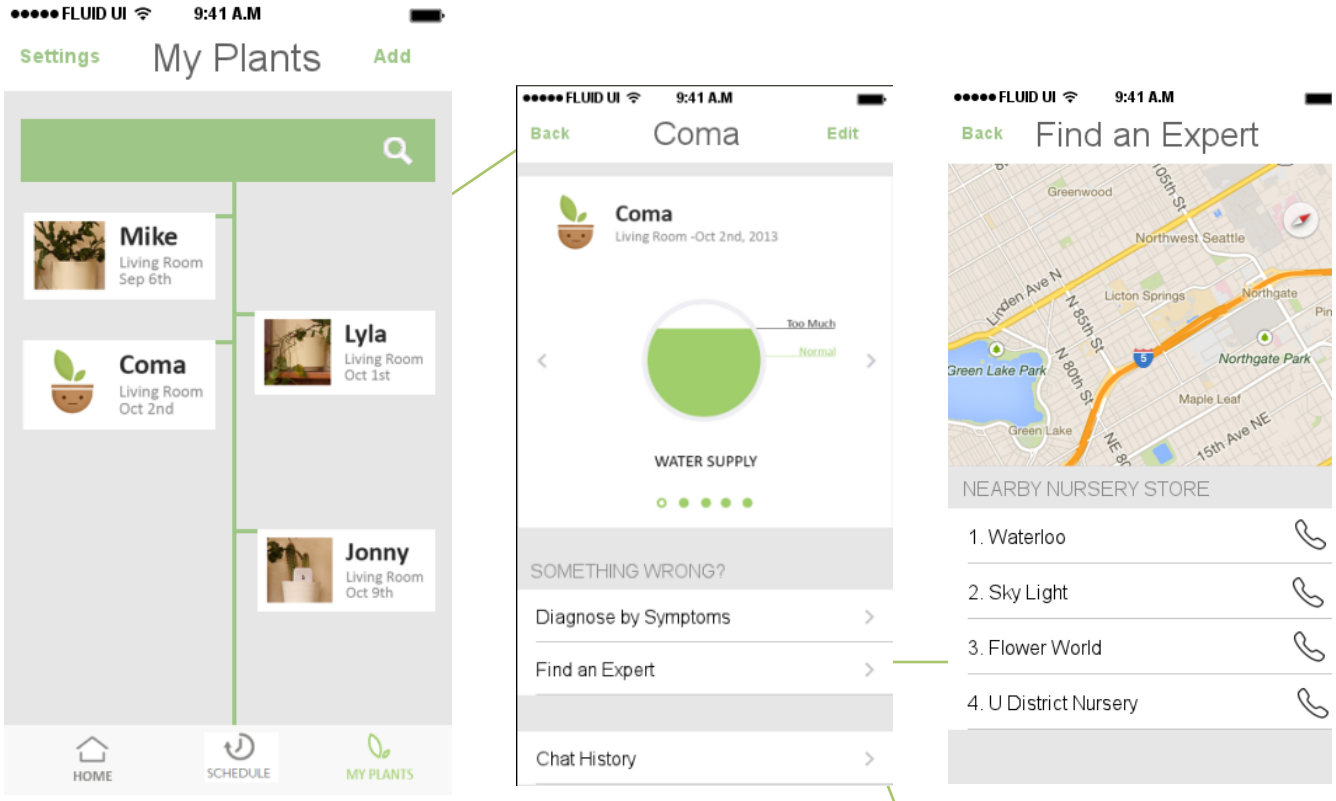
2. The user taps on the notification, taking them to the Schedule page with more details

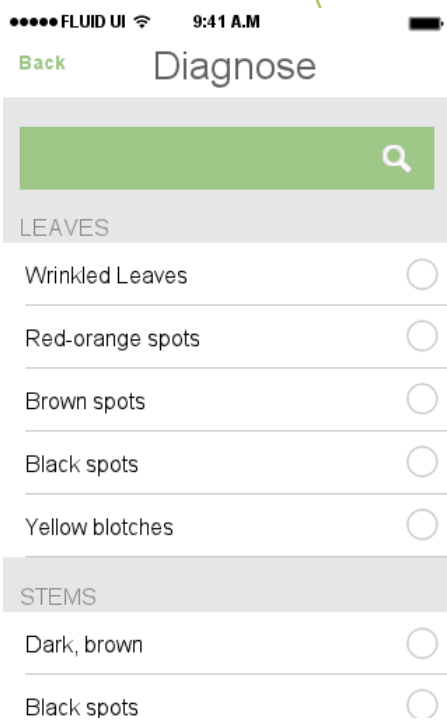


3. A few minutes after watering their plant, the user receives another notification, with the plant thanking them for the water

moderate task - identifying and managing causes of plant distress

Our second task required users to utilize Plantr's 'something is wrong' feature. This is a moderate task because it requires navigating through several screens. Additionally, because use of the 'something is wrong' feature is expected to be infrequent, the process will not be very familiar to users. There are two main parts to this task: the user can choose to check off symptoms and have Plantr attempt to 'diagnose' the problem, or the user can 'find an expert' and contact a nearby nursery.





The last task is somewhat open-ended. If someone realizes that their plant is not doing well, Plantr can help find out what is wrong.

1. First, the user navigates to the My Plants screen

2. Then, the user can choose either the Diagnosis route or Find an Expert route

3. Both lead to congratulations when a diagnosis item is selected or a nursery is selected

Interface Revisions

Pairing a Sensor

Feedback from our subjects during the paper prototype testing as well as from teaching assistants and other students during studio time made it clear that our previous method for pairing a sensor was confusing and unclear. We decided to re-invent this feature, imagining that the sensor would pair with Plantr using near-field communication or another close-proximity technology. This removed the need to have the user choose which sensor they were going to pair with from a list on screen, instead relying on an easier hardware-based solution. There is a significant difference between Figure 2 (the original UI) and Figure 1 (the revised version).

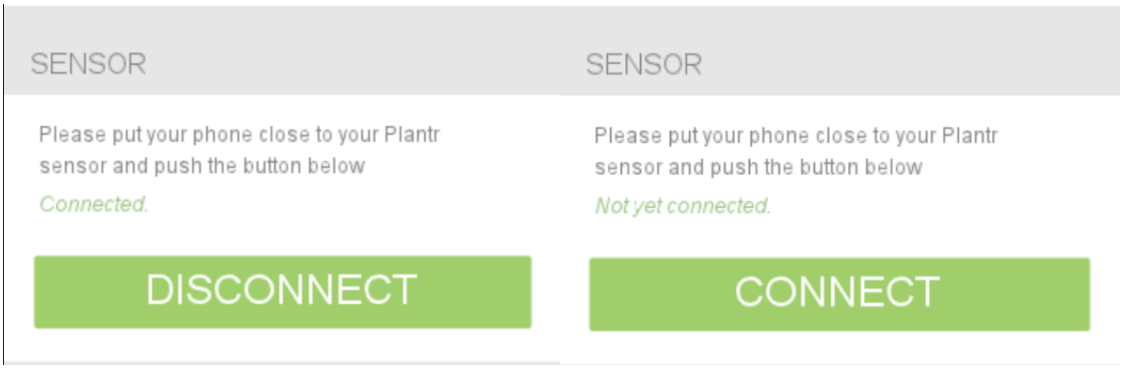


Figure 1: The revised Sensor part of the “Add a Plant” screen.

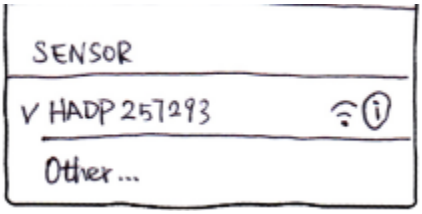


Figure 2: The original sensor section from our paper prototype.

Social Media Sharing

We made some slight changed in the “Share” part of the add a plant task. We changed from a list with check-boxes to a greyed-out versus colored indicator for sharing information about you new plant to Facebook and twitter. While this is less consistent with iOS7 UIs, it is makes it easier to see the Facebook and twitter logos. This will hopefully make it simpler for users to share their important Plantr moments with friends. The difference in cleanliness of the UI from Figure 3 to Figure 4 was a large portion of the decision for the change. Both figures show Facebook selected.

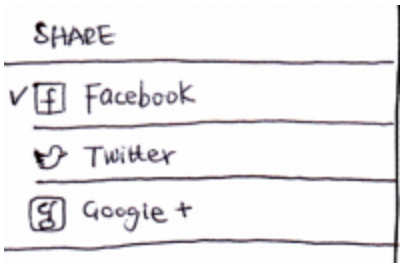


Figure 3: The Original Social Media UI

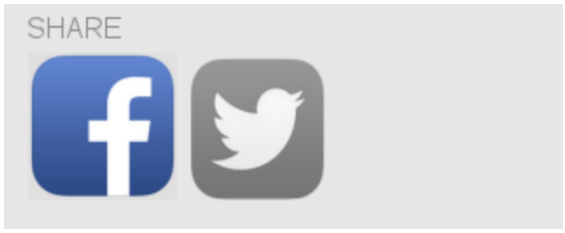


Figure 4: The revised social media portion of the add a plant screen.

Something is Wrong

While we combined the two “Something is Wrong” screens into one after our user study, it became clear that this was not a very good way to approach the problem. We ended up returning to the layout we had in our paper prototype for this part, clearly delineating the difference between the “contact and expert” feature and the “diagnose by symptoms” feature, rather than having both features on the same screen.

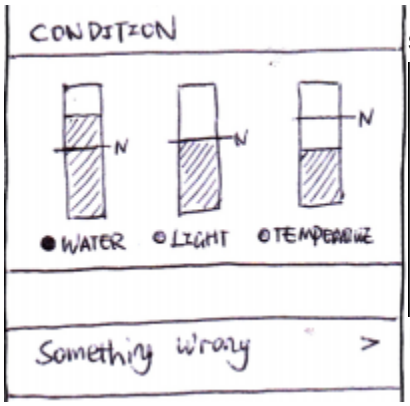


Figure 5: The Original Something is Wrong button.

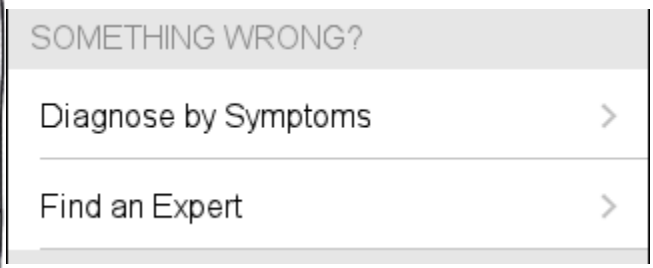


Figure 6: The revised “Something is wrong” section.

Plant State Display

In the low-fidelity prototype, we chose to show each of the different sensors as a separate bar as in Figure 5, but this was hard to translate into a realistic UI that was aesthetically pleasing. We opted to instead show only one type of data at a time with arrows and a swipe-effect for moving between the different types of data, as in Figure 7.

Figure 7: The revised method for visualizing sensor data.

The Prototype

Overview of Implementation

Overall, the changes from our paper prototype were pretty minimal, but the jump from a low-fidelity to high-fidelity, interactive prototype required a great amount of expansion in terms of graphic design. We also spent a fair amount of time on the informational website portion of this project in addition to the actual interactive prototype.

Scenarios

Our interactive prototype constructs three scenarios, which map directly to the tasks described earlier: adding a plant (Figure 1), remembering to care for a plant (Figure 2), and identifying and managing causes of plant distress (Figure 3).

In our first scenario, the user must imagine that they purchase a new barrel cactus, nicknamed Johnny, and want to add Johnny to Plantr as a new plant.

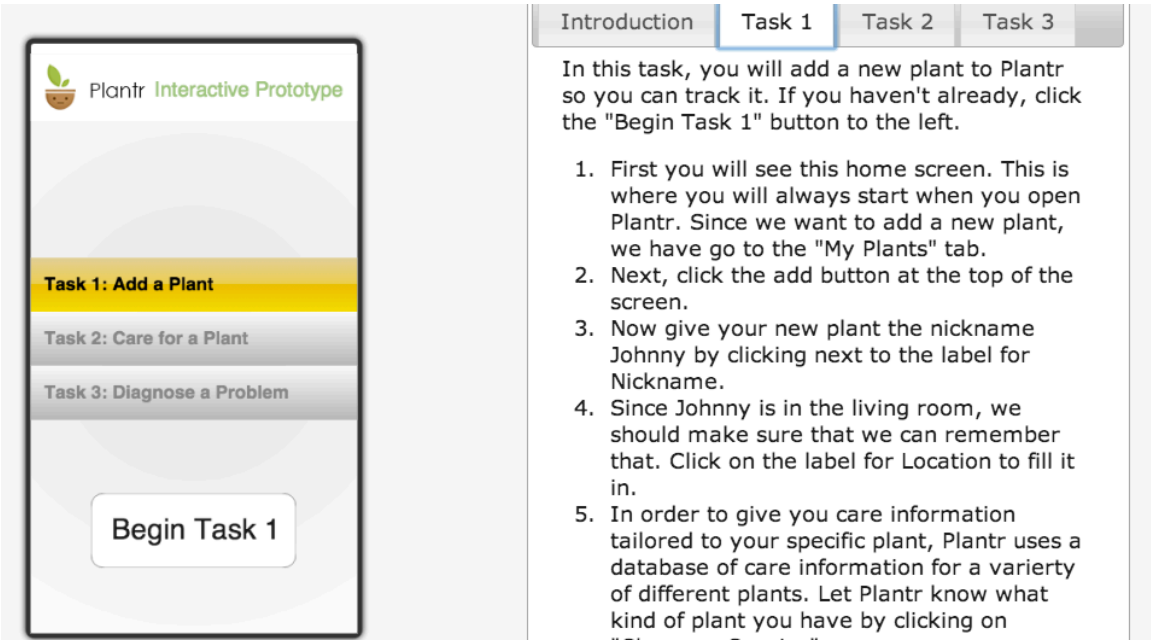


Figure 1

Our second scenario requires the user to imagine that they own a plant named Lyla, and some sort of container and water source which could be used to water Lyla. In this scenario, Lyla sends the user a notification because she is thirsty. The user is required to 'water' (imagine watering) Lyla.

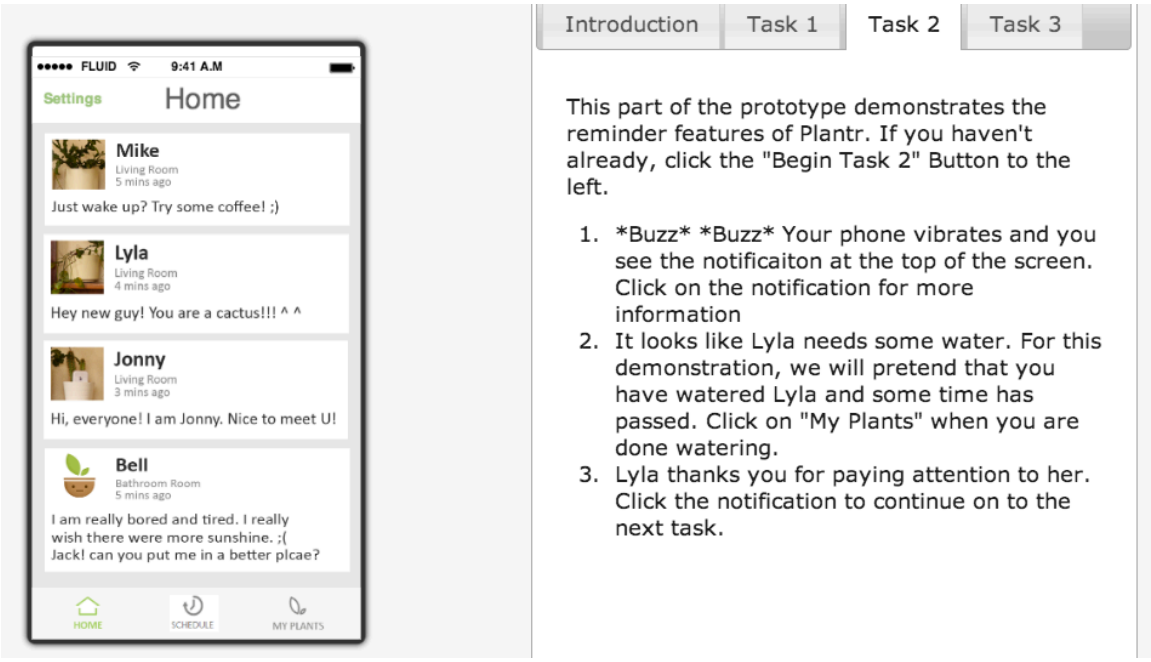


Figure 2

In our third scenario, the user has an orchid named Coma. Coma is suffering from an unknown ailment which causes her leaves to turn grey. We gave the user a series of steps which allowed them to figure out how to make Coma healthy again, either by talking to an expert or using Plantr's diagnose function.

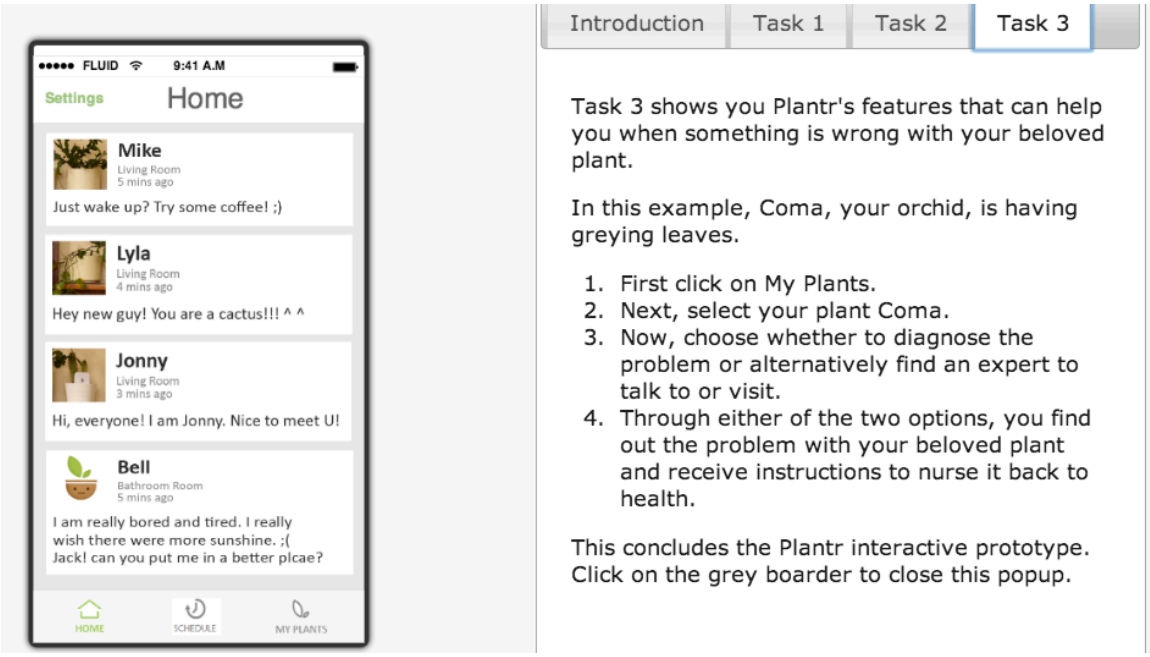


Figure 3

Tools Used

The primary tool for building both our website and interactive prototype was Adobe Photoshop. This allowed an easy transition from the design phase to creation and interaction. A library of iOS7 widgets was very helpful for creating the prototype in a manner that conformed to the platform's UI norms. Our mockup images ended up becoming a large part of our interactive prototype.

Initially, we tried to use Intel's App Framework, a UI toolkit for mobile HTML5 apps. While the parts we made with this tool provided a very interactive and authentic experience for the user, it required a lot more time to make the UI consistent with what an iOS 7 native app would look like. The documentation for this framework was relatively sparse, making it even harder to tweak spacing and other minor formatting issues.

After struggling with App Framework for a bit, we decided to use FluidUI instead, a rapid prototyping tool used for creating image-based mock-ups of mobile apps. Fluid UI was useful because we were able to use images from our UI mock-up, done in Photoshop, to accurately represent our interface. It provided a quick and easy way to simulate interaction within the prototype. Fluid UI, however, had some limitations. It did not provide a way to let the user input text, and it did not support functions such as

searching through a list or receiving a notification. Because of this, some features had to be faked for the interactive prototype. FluidUI allows exporting the entire prototype as a combination of HTML, Javascript, images, and CSS. Using a combination of jQuery, Javascript, CSS and in-line frames, we integrated the prototype generated by FluidUI into the rest of our website.

Portions Not Implemented

We chose not to implement the “add a species” functionality shown in the figure below. While one potential user requested this feature when we were running through the paper prototype with them, we found that it confused many users, so we decided not to make it part of any of the tasks for our interactive prototype.

We also chose not to implement customizing the photo of a plant while adding it. We felt that providing an intuitive way to take and/or choose a photo would be tough and distract from the main focus of the app. Similarly, we did not implement the “Chat History” feature or the “Remove a plant” feature because they were not major parts of any of our current tasks.

Limitations of the Prototype

While our interactive prototype gives users a pretty good sense of the user interface, we did have to simplify some parts. In order to create a high-quality interactive prototype in a short amount of time, we had to give the user a less realistic experience with less room to experiment. Since FluidUI doesn't support text entry, we automated the task by having the user click on the text field where they would have entered the relevant text. Similarly, for a search box, we auto-filled the text and filtered after one tap on the search box.

In the first task, adding a plant to Plantr, we simplified the pairing with a sensor portion of the task in order to make it easier for the user to connect a specific sensor with a plant they are adding. Our new way of pairing a sensor requires some amount of imagination from the user as far as when they are pairing the sensor. Due to constraints introduced by the way scrolling works in FluidUI, after pressing the “Connect” button, users are returned to the top of the page rather than to the same part of the page as when they initially pressed the button. This is inconsistent with how a true app would perform.

Conclusion

Overall, our website and interactive prototype provide a good overview of the functionality of Plantr, without requiring large amounts of tedious data entry. While it is not perfect, every effort was made to make the experience as authentic as possible within the capabilities of the tools used.

Appendix I: The Website

SOCIAL INTERACTION

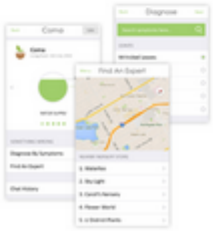
Plantr allows your plants to talk to you and each other. Imagine that you buy a new plant. Your other plants say hi and welcome the new plant.

Your plants can express happiness, sadness, and other emotions!

REMINDERS

Plants will send you messages via Plantr when they need care.

When your new plant needs to be fertilized, it sends you a message telling you that it's hungry. When you fertilize your plant, it sends you a thank-you message.



SOMETHING IS WRONG

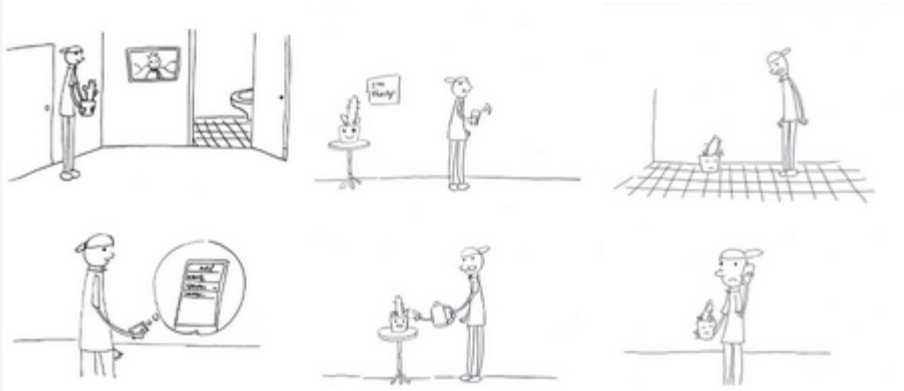
Plantr can help you take care of plants even when something goes horribly wrong.

Imagine that one of your plants has brown spots on its leaves. You check its water supply, pH, temperature, and light exposure, but nothing appears to be wrong. You can use Plantr to diagnose the problem and figure out a solution or contact a nearby nursery store for more help.

Scenario Video

0:00

Scenario Sketches



Team Members

JEN JIANFEN

Graduate Student in MHCID

I am a designer who studies at the U dub. I am mainly responsible for the user interface design of Plantr.

Contact me: jianfenz@uw.edu

WHITNEY

Undergraduate Student in Computer Science & Creative Writing

I am a UW student studying software engineering and writing. I am responsible for a large part of the writing in Plantr.

Contact me: wps5@uw.edu

PAUL

Undergraduate Student in Electrical Engineering

I'm an EE student at UW focusing on embedded systems and controls. I'm mainly responsible for iterative design, interactive prototyping and hardware.

Contact me: pbartell@uw.edu

JORY

Undergraduate Student in Computer Science

I have a background in software engineering. I am mainly responsible for task development and ideation.

Contact me: joryn@uw.edu

Download Documents

PROJECT PROPOSAL	CONTEXTUAL INQUIRY	TASK ANALYSIS	VIDEO REPORT	USER TESTING

PRESENTATION ONE	PRESENTATION TWO	PRESENTATION THREE

Appendix II: The Interactive Prototype