

# SHAUN KELLY | HCI CAPSTONE SPRING 2013

## MILESTONE 4 – THE PROTOTYPE

### Yard Genius

Yard Genius is an application to help homeowners maintain their yards by keeping track of maintenance tasks and suggesting appropriate plants and vegetation based on the user's context.

The main user group for my system will be homeowners with yards and gardens to maintain. Users who will benefit most from the application are those with limited experience and knowledge about yard and garden maintenance.

The three main tasks for the application identified during user research are:

1. Determining suitable plants for a given location
2. Determining which tasks (planting, mowing, fertilizing, pruning, etc.) are required based on what items are planted
3. Design a pleasing layout for plants in the yard

### Prototyping Strategy

The strategy I chose for prototyping was an HTML/JavaScript-based, digital prototype.

From a user-testing perspective, I felt that as a tablet application, being able to test the interactions on an actual device was important.

From a development perspective, I am very comfortable developing in HTML/CSS/JavaScript from previous experience and from what I learned in HCI 573. Additionally, with the assortment of open-source licensed frameworks for CSS and JavaScript, I could develop the application almost as quickly but with better fidelity to the actual interactions than in a prototyping tool such as Axure or JustInMind.

I used the following open-source frameworks to assist in rapid prototype development:

- Twitter Bootstrap - for CSS, including typography and layout and for some interactions, such as modal dialogs.
- jQuery - for additional interactions and behaviors such as adding or updating text.

The prototype must be developed to meet the needs of the usability testing. Cooper (2007) recommend testing:

late enough in the process that there is a substantially complete design to test, and early enough to allow adjustments in the design and implementation

Buxton (2007) says:

[T]he main drawback of conventional sketching has to do with its limitations in capturing time, dynamics, phrasing - the temporal things that lie at the heart of experience.

It's these transitions that Buxton refers to that I am most interested in testing. And these transitions can't be properly captured through paper prototyping.

The prototype strategy I used would be characterized by McMurdy et al. (2006) as a "mixed-fidelity" prototype. They identify five areas of fidelity: visual, breadth of functionality, depth of functionality, richness of interactivity, and richness of data-model to argue that a high/low fidelity continuum is not an accurate way to describe prototype fidelity. My prototype has higher fidelity in functionality and interactivity and lower fidelity in visual and richness of data-model. This level of fidelity meets the needs of testing a high fidelity of interaction while not requiring substantial development that would be difficult to change.

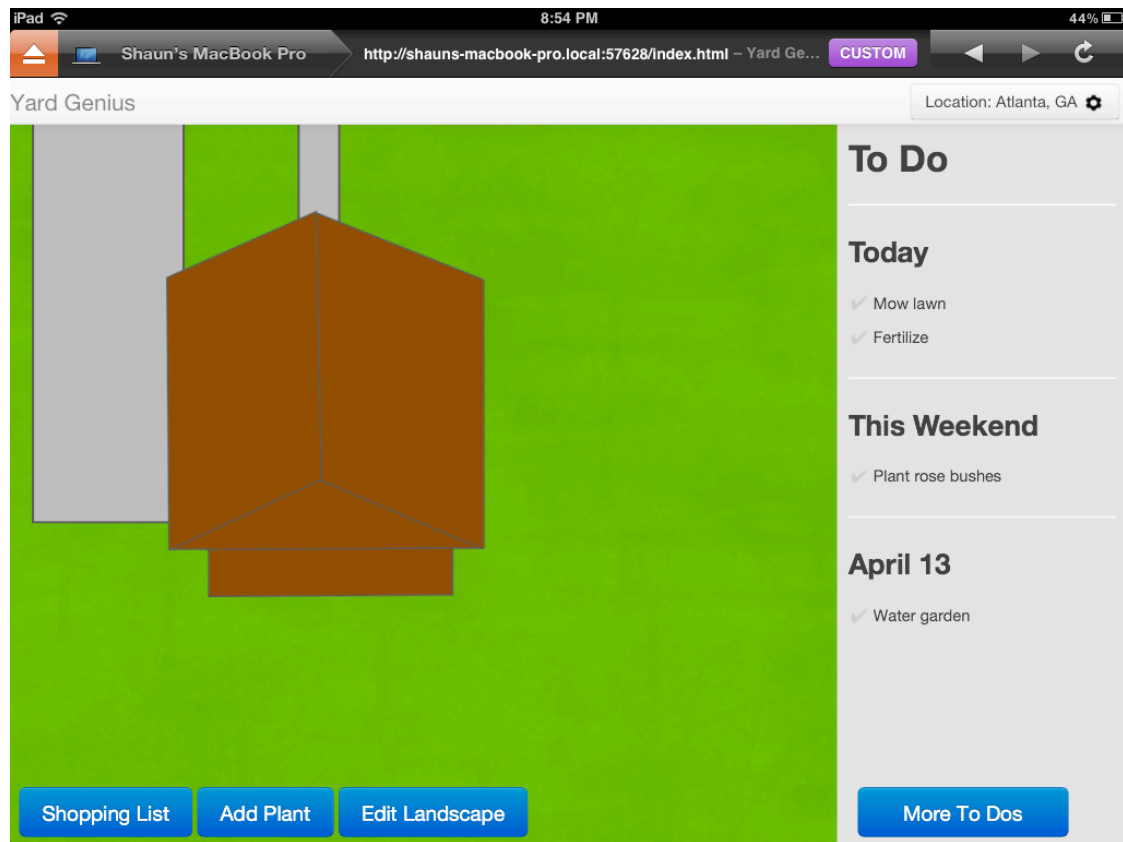
## Description of Prototype

The complete prototype is available at the following URL:

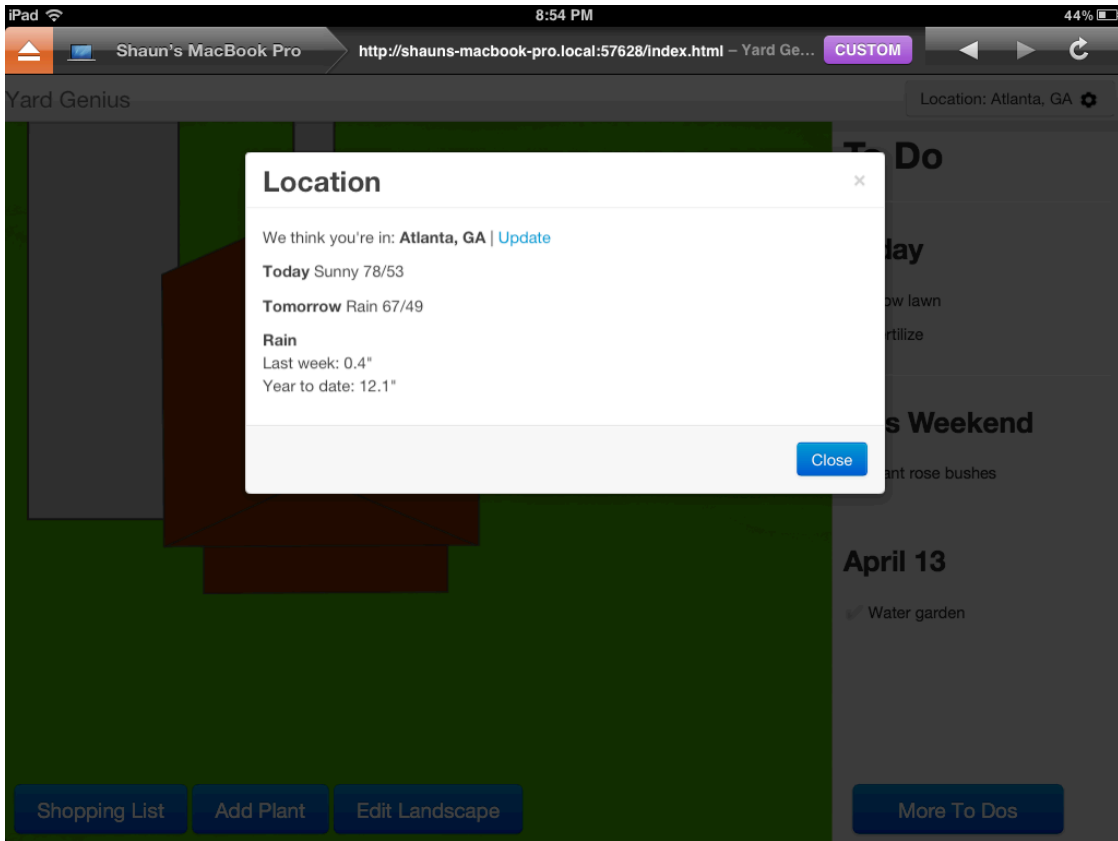
<http://superawesomemegood.com/ygproto>

While available on desktop browsers, it is best viewed in a 10" tablet browser as it is a prototype for a table application.

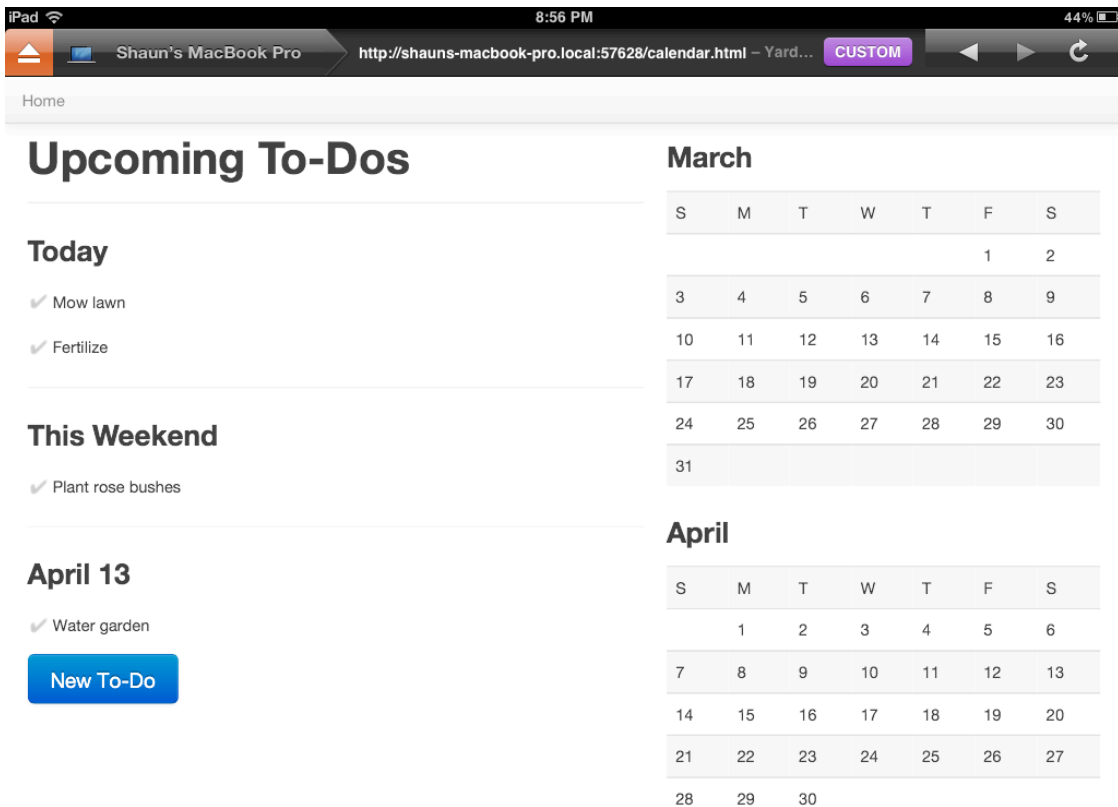
The following image shows the home screen:



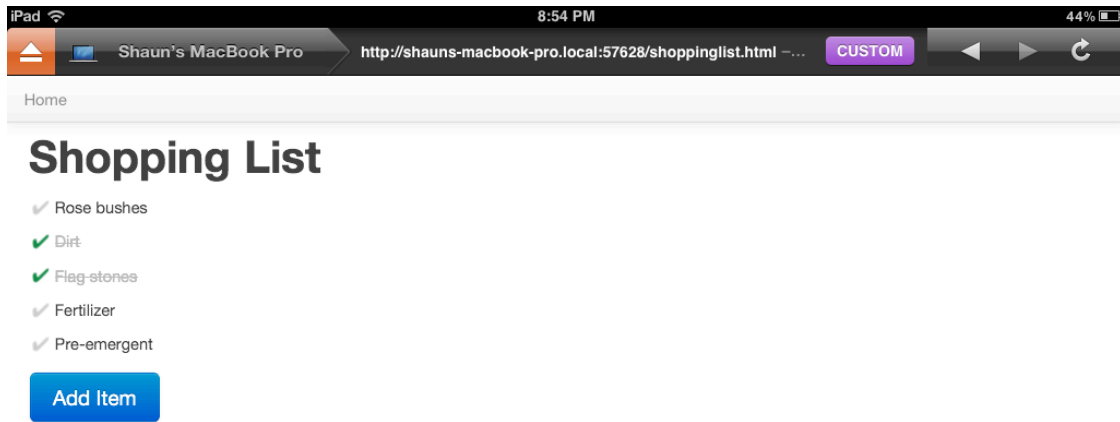
All functions are available from the home screen. Click the top right button to view/edit location context:



You can tap to-do items from the home screen to mark as complete or click More To Dos to get a complete list of to-do items:

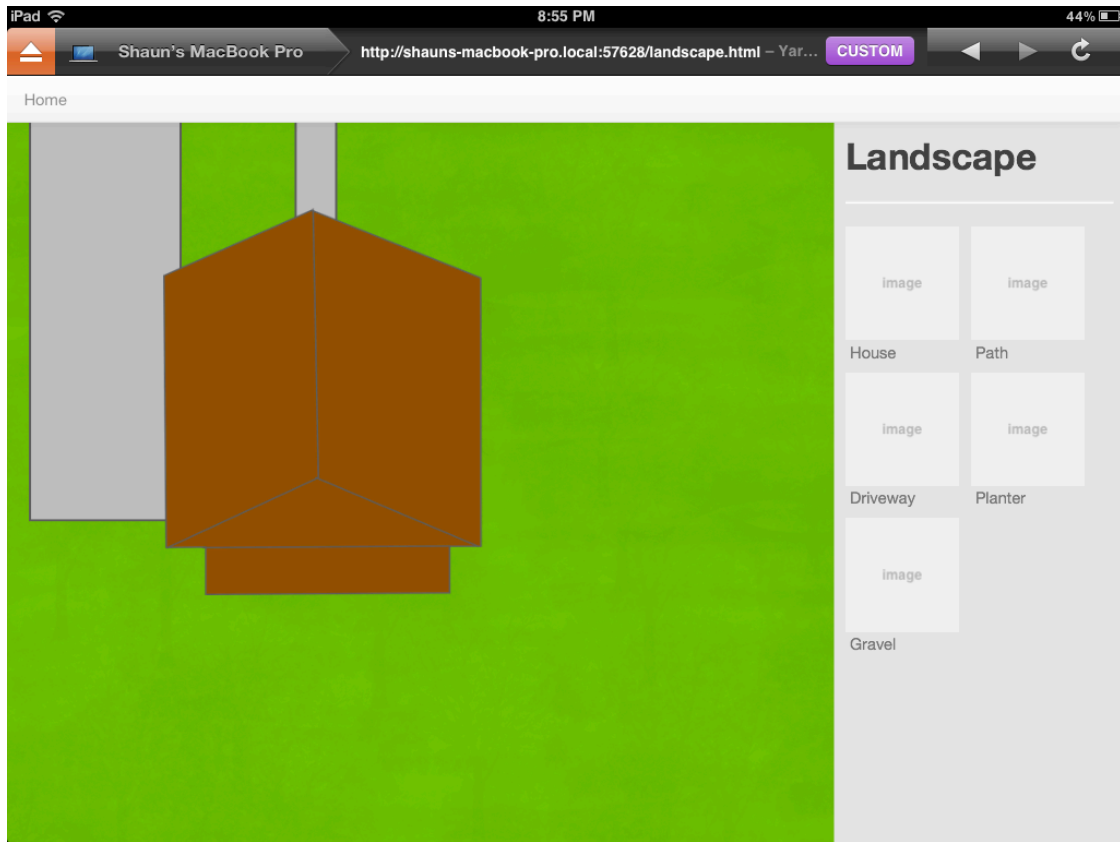


The following shows the shopping list screen which would be auto-populated with items needed for auto-generated tasks:



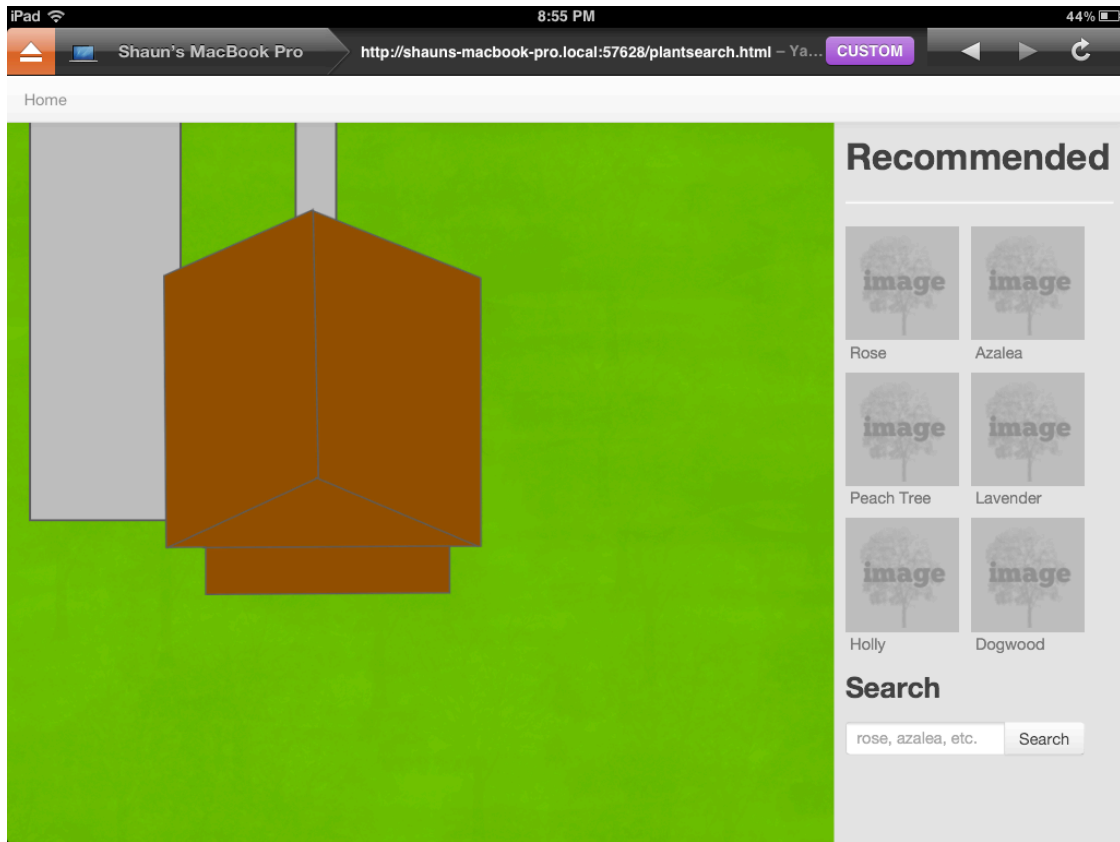
Check off the items as purchased or click Add Items to add custom items.

The following shows the edit landscape screen:



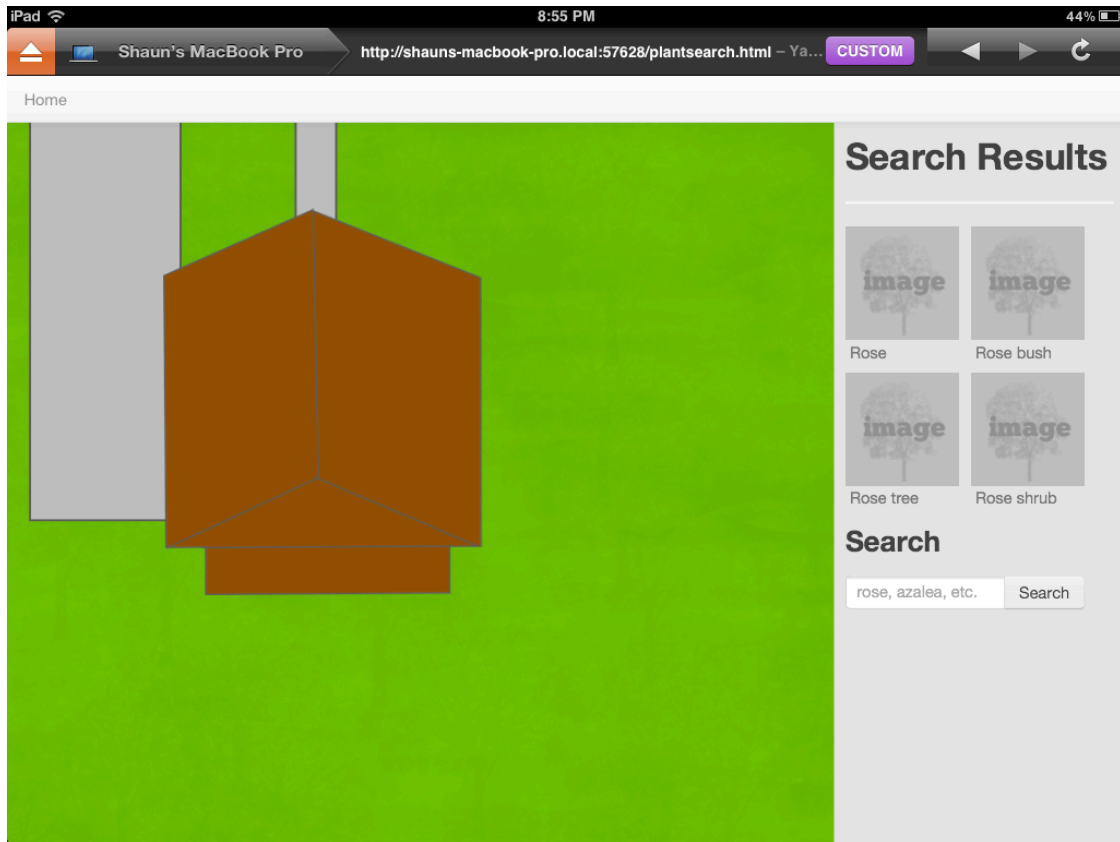
New landscape items can be added to the landscape by dragging from the pallet on the right.

The following shows the Add Plant screen:



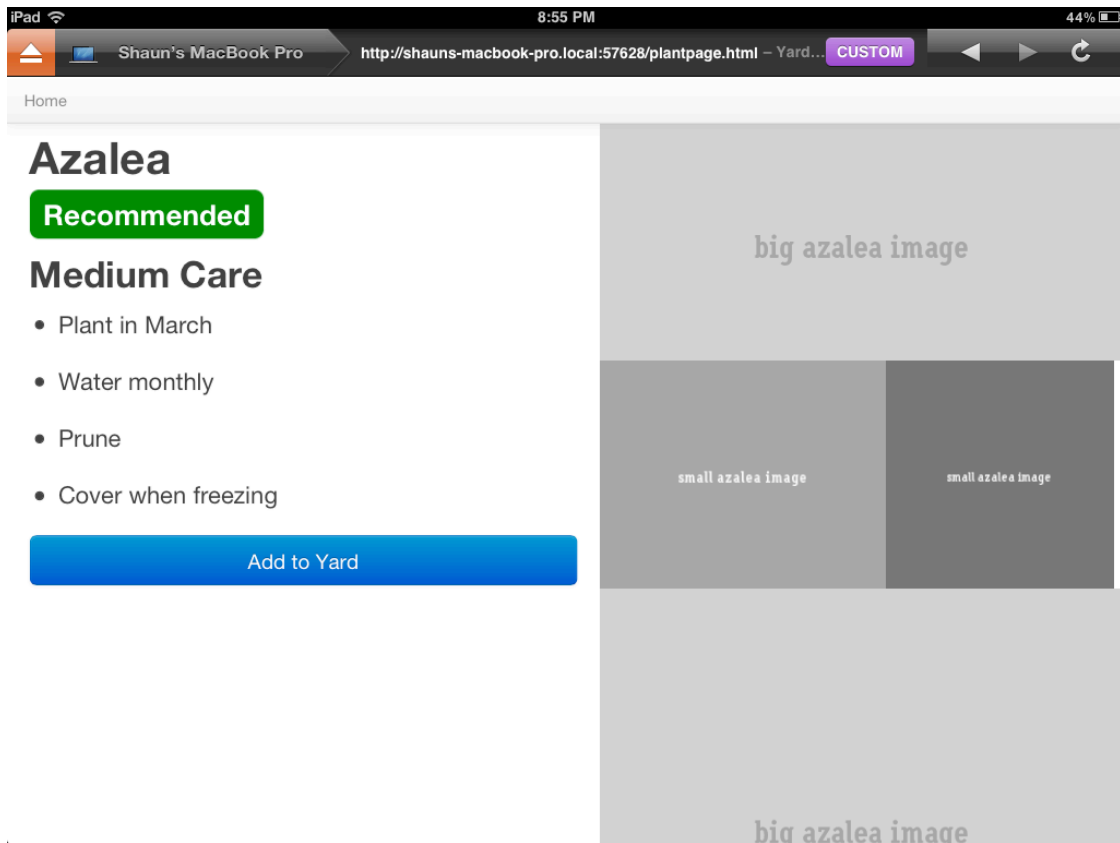
By default, this screen shows a list of plants recommended based on location. Plants can be added to the landscape by dragging from the palette similar to the landscape items. You can search for additional plants using the search option on the bottom of the palette.

The following shows the search results:



Click on any plant image in the search results to display more information about the plant:





The plant detail screen offers information about care requirements and inspiration pictures with the option to add the item to the landscape.

The main features not supported in the prototype are the auto-generation of tasks and shopping list items based on layout and context. This would be far too complex to implement in a prototype, so these areas are pre-filled with similar data. As the testing for a single user will take place on a single day, it's not important that new tasks or shopping list items be auto-generated if the prototype is pre-filled with relevant data.

## Reflection

One of the challenges of the strategy I chose was that the libraries used to speed the development process constrain the look and feel of the application. In order to develop an application that doesn't have the clear look and feel of the Twitter Bootstrap library, I would have to spend a good deal of additional time developing a custom look and feel. Using hand-drawn paper prototypes would give you greater control over look and feel. Prototypes developed in a prototyping tool would have a similar issue with using libraries as most prototyping tools allow you to choose a library of widgets.

This applies to interactions, as well. For example, the display of modal dialogs is built into Bootstrap. To use custom animations other than the default would require significant work. Drag and drop is built into jQuery, but has default behaviors which require extensive programming to override. Thus, it's easiest to rely on the defaults. This may subtly affect any design decisions made while converting sketches into code.

Of course, paper prototypes don't let you test animations and interactions with the same fidelity. Clickable greyscale wireframes might give you some sense of how interactions work, but more complex animations and drag and drop style interactions don't work well in that medium.

Working in digital to create the prototype also allowed the freedom to quickly test and iterate on minor refinements in the prototype design in a way that would be challenging or impossible with paper prototyping where a minor refinement might involve an entire new sketch. For example, I quickly realized that I needed to add a top navigation bar to allow users to return to the home screen, otherwise they would have no way back. I could quickly add this using a Bootstrap component. In this way, prototyping in code might still be considered "sketching" in the way that Buxton defines the concept.

I did underestimate the amount of time this process would take. Though I have experience with HTML/CSS/JavaScript, there was a good bit more to learn about the frameworks which slowed down my early work. That is something I would have to consider in the future. However, developing a pattern library of components for an application could significantly speed up an HTML/CSS prototyping process over time as much of the CSS and JavaScript could be reused.

In addition, one of the purported benefits of prototyping in code is the ability to reuse the work for the prototype in production. However, when prototyping for a native tablet application in HTML/CSS/JavaScript, I would not be able to reuse any of the code.

# Citations

Cooper, Alan; Reimann, Robert; and Cronin, David. (2007). *About Face 3: The Essentials of Interaction Design*. Indianapolis: Wiley.

Michael McCurdy, Christopher Connors, Guy Pyrzak, Bob Kanefsky, and Alonso Vera. (2006). "Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, Rebecca Grinter, Thomas Rodden, Paul Aoki, Ed Cutrell, Robin Jeffries, and Gary Olson (Eds.). ACM, New York, NY, USA, 1233-1242.

DOI=10.1145/1124772.1124959 <http://doi.acm.org/10.1145/1124772.1124959>

Buxton, Bill. (2007). *Sketching User Experience: Getting the Design Right and the Right Design*. San Francisco: Morgan Kaufman.