Sprite Final Report

Bring your smart home device to life.

CS 147 Fall 2018

Team:




Amanda S, Julea C, Clara K, Chloe T

We are a team of students in the Transforming Living Spaces studio.

Problem and Solution Overview

Smart home device owners rarely take advantage of their smart home device's full capabilities. Sprite is an app that personifies your smart home device, making it easier and more engaging to use your device. Your Sprite depends on you, fostering a sense of responsibility and creating a relationship between you and your device.

Therefore, we thought an appropriate mission statement for our mobile app to be: ***make interacting with a smart home device engaging and fun*** and our value proposition to be: ***bring your smart home device to life***.

Figure 1: *High Fidelity Sprite Prototype showcasing Sprite's moods.*
As you interact with your smart home device, your Sprite becomes happier.

Tasks and Final Interface Scenarios

Simple: Create your first Sprite

We labelled this task as simple because a user would have to complete the tutorial associated with first-time use of the app. Many first-time users were confused about the purpose of the app, so we decided to create a high-level description of the app, displayed

the first time they open the app. The onboarding screens (Figure 2) cover creating a Sprite (by pairing a smart home device), discovering new features, and earning badges. To create their first Sprite, the user swipes through the onboarding screens shown in Figure 2. Then, they are prompted to select what smart home device they'd like to pair with the app through the "create new sprite" screen. Once the pairing is confirmed, they are able to rename the Sprite and customize its color to complete the setup process.

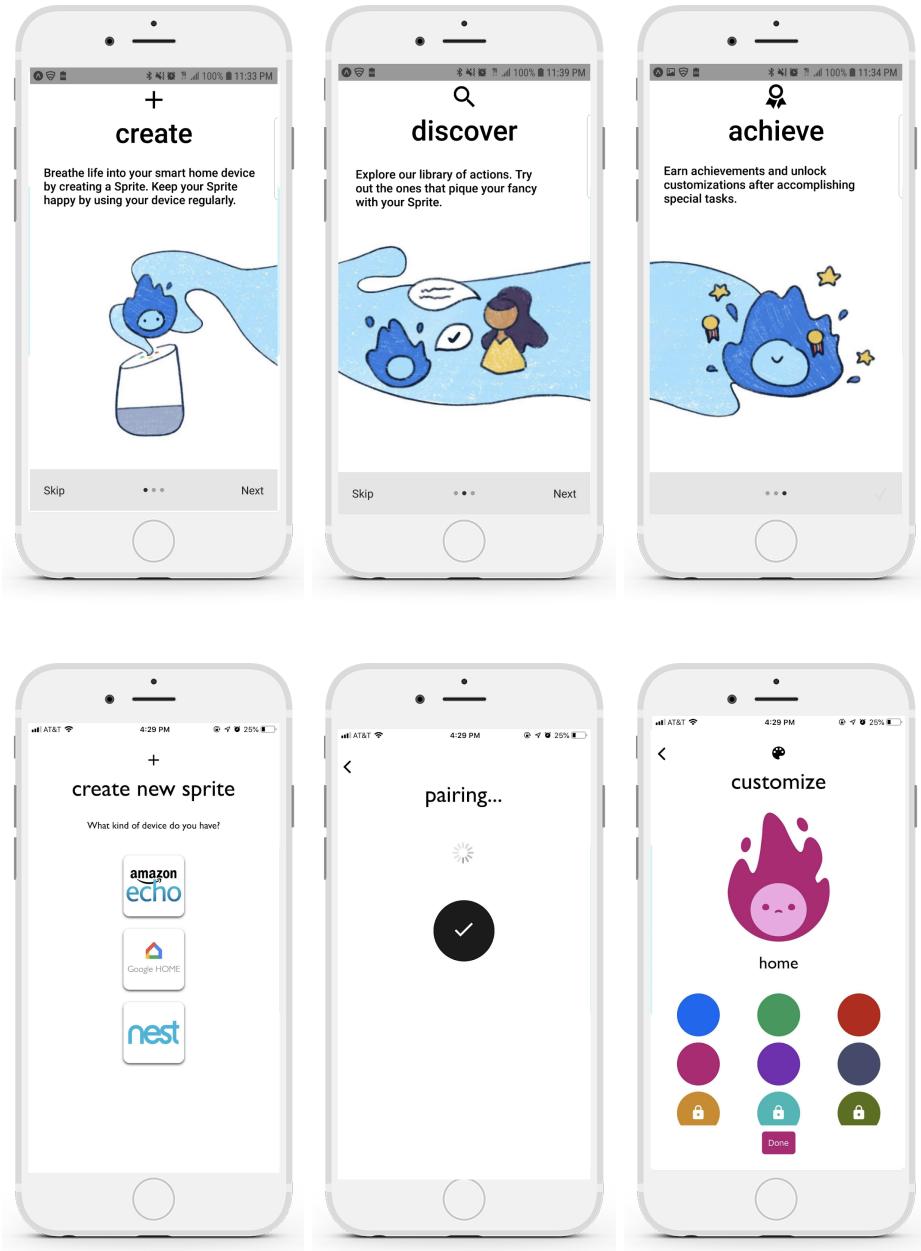

Figure 2: *Instructional onboarding screens and create new Sprite sequence.*

Moderate: Learn a new action

From the discover screen, users can explore actions sorted into five categories (for you (recommended), saved, trending, lifestyle, fun) or search for a particular action in the search bar. Once an action is selected, the user can swipe among a brief description of the action and various chat bubbles containing voice instructions to activate the action. After the physical smart home device registers the user's command, a check mark will appear on screen to indicate that the Sprite recognized the successful execution of a command. Users can then return to the home screen and see that their Sprite's mood has improved, motivating them to continue exploring new features to maintain the Sprite's happiness.

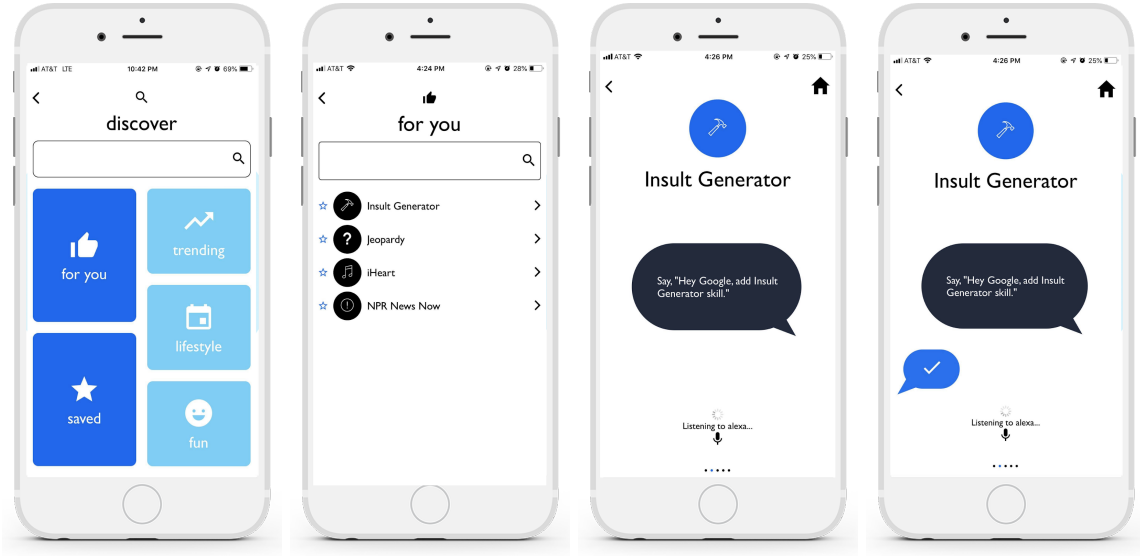

Figure 3: *Learn a new action task flow.*

Complex: Unlock a new badge

From the achieve screen, users can view all of the badges they've earned and the ones remaining to unlock. Unlocked badges have a star displayed, and locked actions have a lock displayed. When the user clicks on a locked action, a popup will appear telling them what actions they'll need to complete to unlock that badge. As the user unlocks more badges, more customization options will be available for their Sprite, motivating them to keep using and exploring the app.

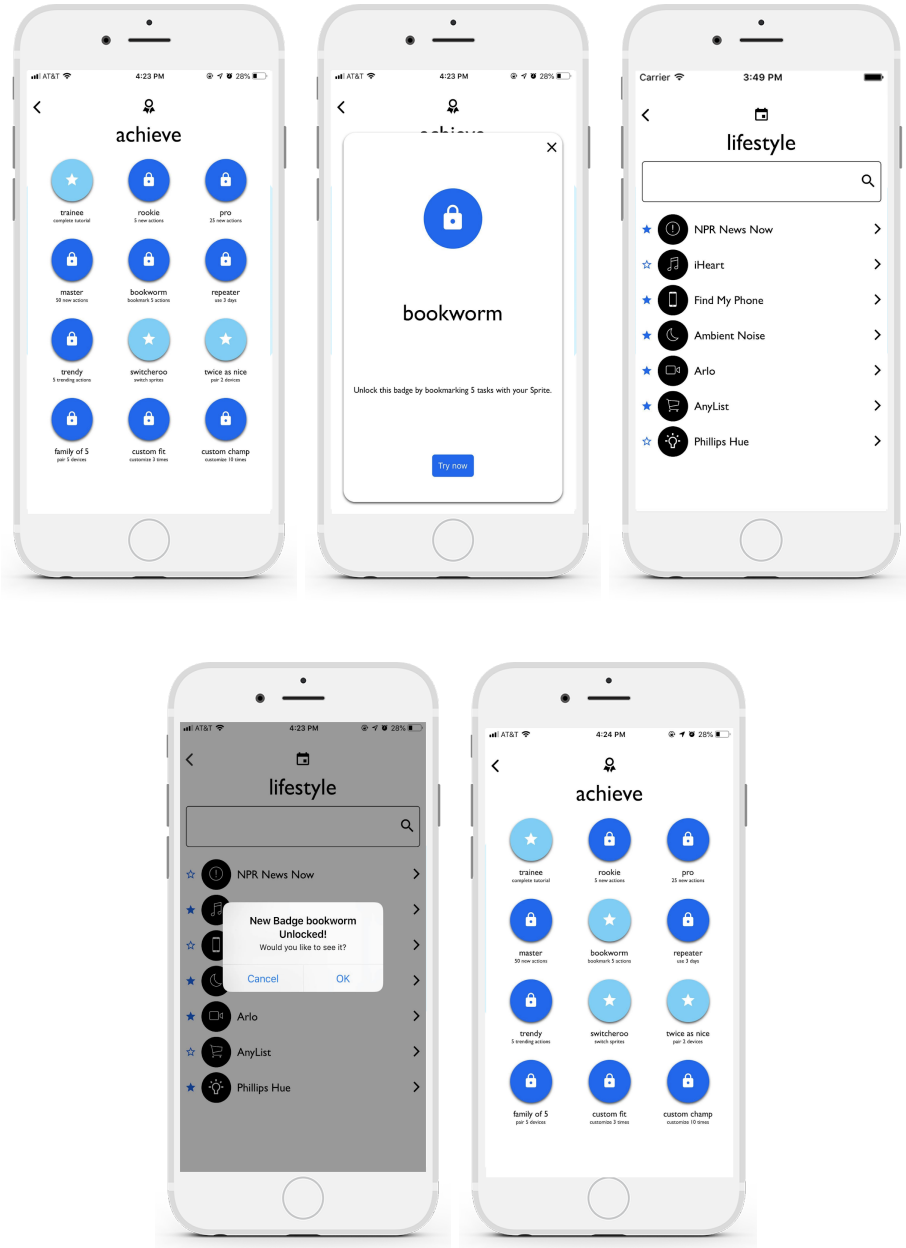

Figure 4: *Unlock a new badge task flow.*

Simple: Switch to Nest Sprite

Our last task centers around pairing multiple smart home devices to Sprite. When users tap the switch button on the top left of the home screen, they're shown their collection of Sprites, and can switch between them. Alternatively, users can also swipe left and right to switch between Sprites from the home screen.

Figure 5: *Switch to Nest Sprite task flow.*

Design Evolution

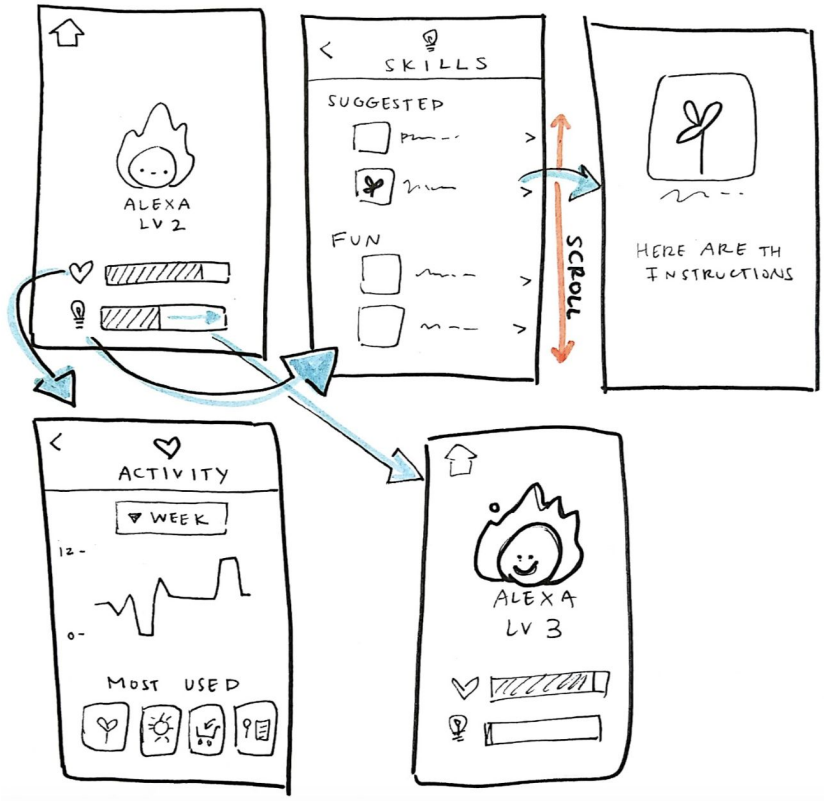

Figure 6: *Initial UI sketch*

Lo-fi Prototypes:

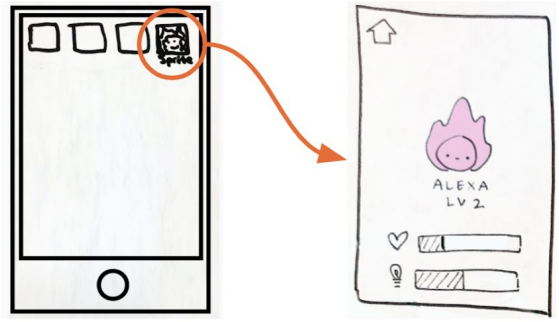

Figure 7: *UI flow for our simple task (checking your Sprite's status)*

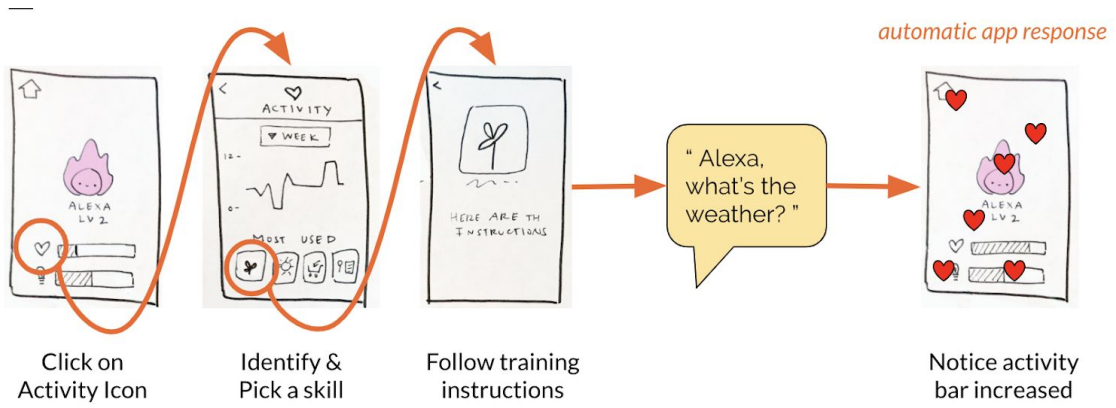

Figure 8: *UI flow for our moderate task (increasing your Sprite's happiness)*

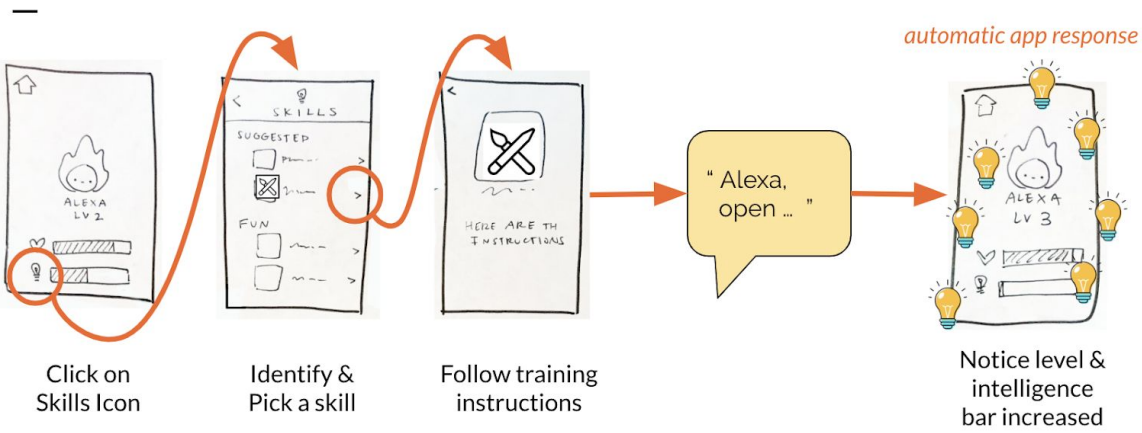

Figure 9: *UI flow for our complex task (increasing your Sprite's intelligence)*

For our initial UI sketch, we decided to emphasize the personification of a smart home device into a pet-like creature by placing the Sprite at the center of the home screen. We wanted to encourage people to interact with and teach their Sprite, so we came up with the concept of happiness and intelligence. Your Sprite's happiness level, depicted by the heart icon, increases as you interact more with your Sprite. Your Sprite's intelligence level, depicted as a lightbulb, increases as you use new skills with your smart home device. We also had screens for the user to explore new skills, check their past activity, and learn how to use each skill. We kept this design for the lo-fi prototype.

Based on feedback from our lo-fi testers, we adjusted our medium-fi prototype in a couple different ways:

1. Changing the Home Screen

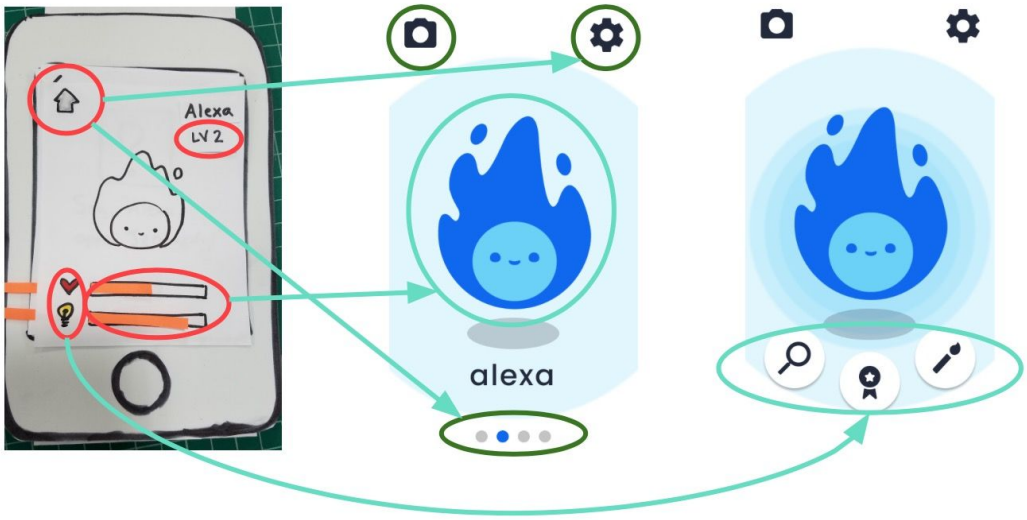

Figure 10: *Before & After: Home screen*

We noticed that users kept clicking on the home button when they didn't know how to achieve an action (even when the button wasn't relevant to the task), so we removed the home button. We replaced the home button's functionality with the ability to swipe between Sprites and a settings button in the top right corner. Also, users were confused about the purpose and importance of progress bars. We removed the progress bars and show the Sprite's emotional state purely through facial expressions. Users were confused by the heart and light bulb icons and didn't recognize them as buttons, so we removed the heart and light bulb icons and replaced them with 3 buttons that appear on tap. We

introduced a 3 button functionality (discover, achieve, customize), instead of the 2 previously (happiness and knowledge) because most users didn't understand what the heart icon meant during testing.

2. Skills to Actions

Figure 11: *Before & After: Skills/Actions*

We found during user testing that although the skills screen itself was intuitive to use to find a particular skill, this layout would not be scalable to a library of several hundred or several thousand skills. Additionally, the idea of a “skill” is confusing, so we opted to drop the word entirely. Each feature is now considered an “action”. To accommodate a larger number of actions and to optimize finding actions relatable to the user, we switched to an app-store like format. This is a better alternative to a sectioned list because it no longer requires the user scroll long distances to find a particular category. We also wanted to add a search bar to allow for users to find desired actions more easily, which aligns with reframing our app as a fun and useful instruction manual.

3. Action Screen

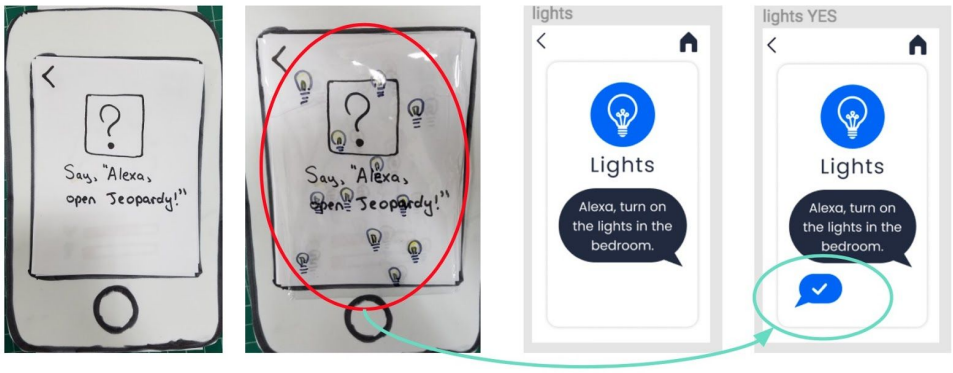

Figure 12: *Before & After: Action screen*

In the lo-fi prototype, once the user said the correct command prompt (e.g., "Alexa, open Jeopardy!"), light bulbs would rain down on the screen and the user would be automatically redirected to the home screen. In our new action screen, we put the voice command prompt in a speech bubble to more closely simulate a real life conversation. We introduced a check mark that appears once your smart home device has registered your command. Once completed, the user can return to the home page through the new home icon on the top right corner.

4. Activity to Achieve

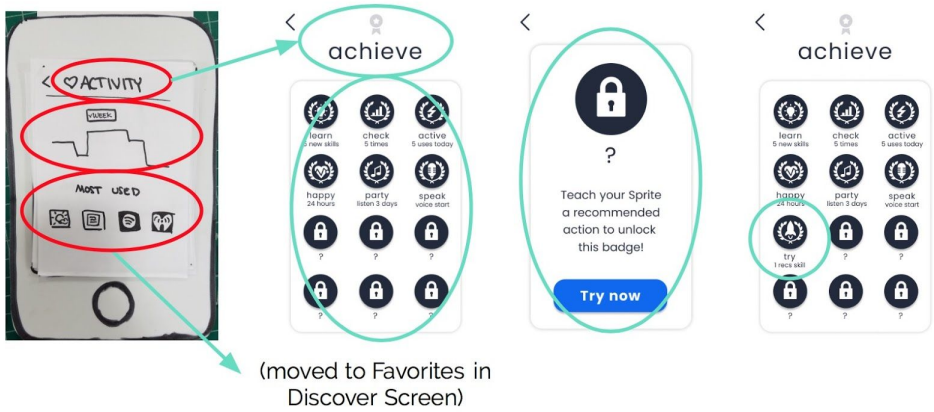

Figure 13: *Before & After: Activity/Achieve*

During user testing, we noticed that people didn't understand why looking at their past activity would be helpful, or why using their smart home device more would be helpful. Therefore, we reframed "Activity" as "Achievements". We also removed our original

activity graph completely, thinking that a badge system would be more motivational to achieve specific tasks and actions. We created an achieve screen that would host all your badges, and as you achieved more actions, you can unlock more badges. If the user clicked on a locked badge, there would be a description on how to unlock that badge, and then a “try now” button directly prompting you to try a specific action. We wanted to make it clear what the user had to do and reduce the amount of memorization necessary. Once that’s completed, the badge would be unlocked.

Also, we reframed Most Used skills as Favorites and moved its functionality to the Discover Screen. During user testing, we saw that users didn't see the link between viewing past activity and using a most used action, so we decided to lump all features that involved using an action together and more clearly separated them from past activity.

5. Switch Sprites

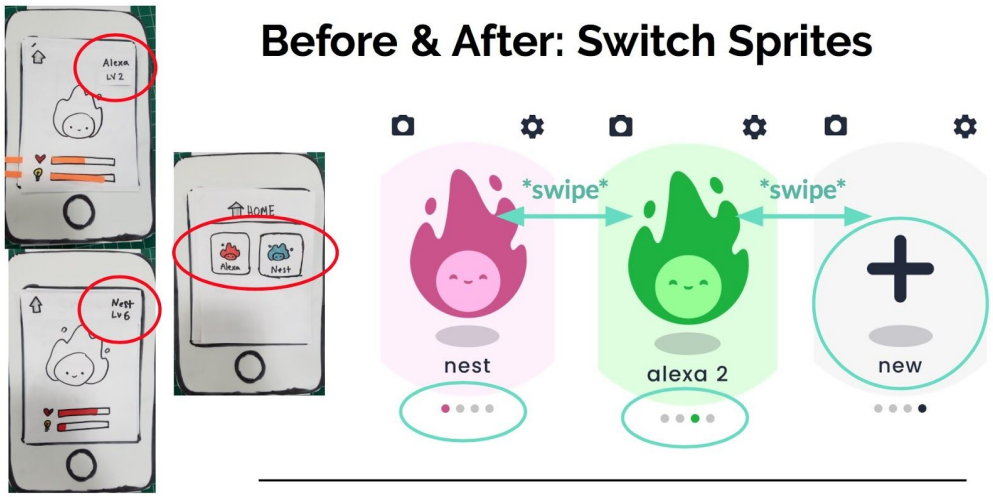

Figure 14: *Before & After: Switch Sprites*

Before, to switch Sprites, the user had to tap on the home icon, and then select the corresponding Sprite button. Although users usually did not have trouble with this task during testing, we felt that it took a long time to complete this, and wanted a quicker and easier way to switch between Sprites. One user suggested a swiping feature to switch between Sprites, which we implemented in our medium-fi prototype. As you swipe between your Sprite devices, the colored circle on the bottom indicates what home screen you’re on. We also added a screen on the far right to allow you to register a new smart home device.

Med-fi Prototypes:

You can see how our task flows evolved from our lo-fi prototype to the medium-fi prototype below.

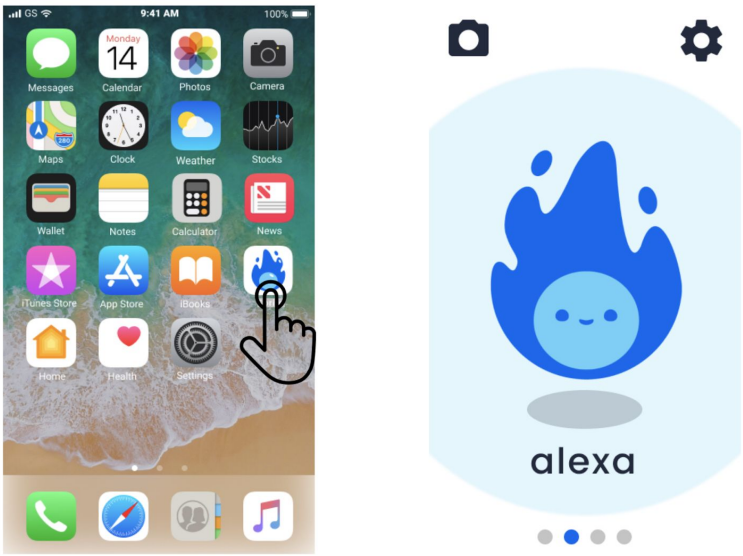

Figure 15: *Simple Task - Check in on your Sprite*

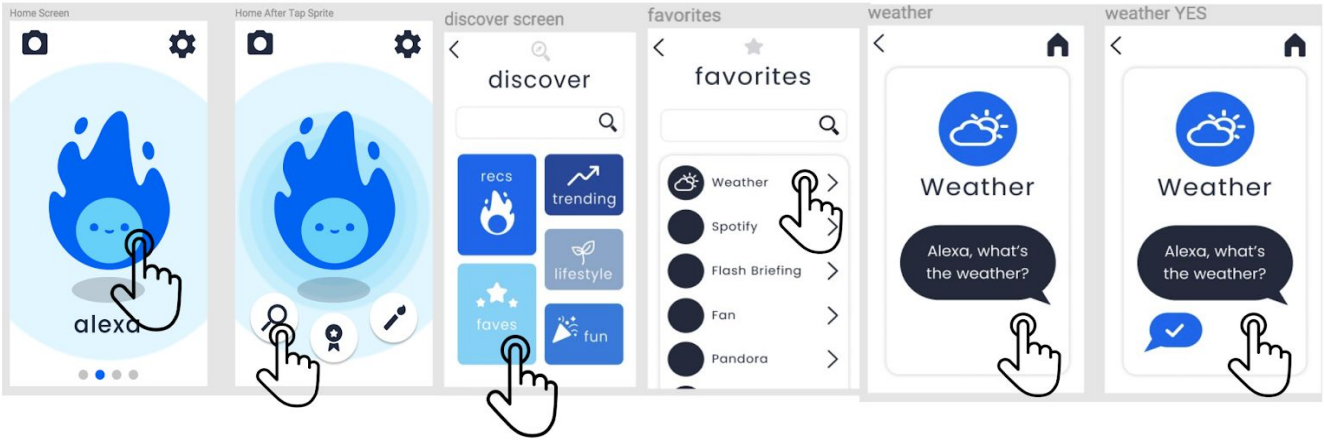

Figure 16: *Moderate Task - Use a favorite action*

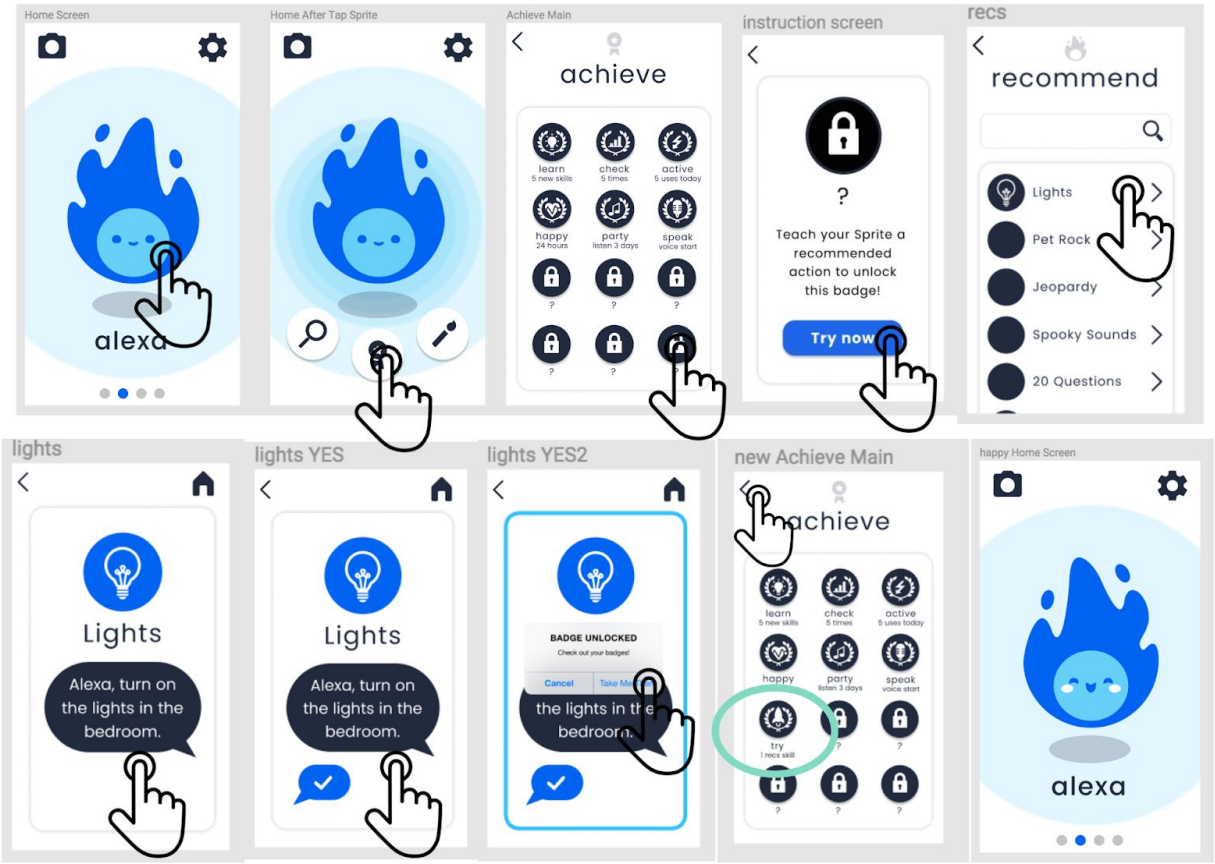

Figure 17: *Complex Task - Unlock a new badge*

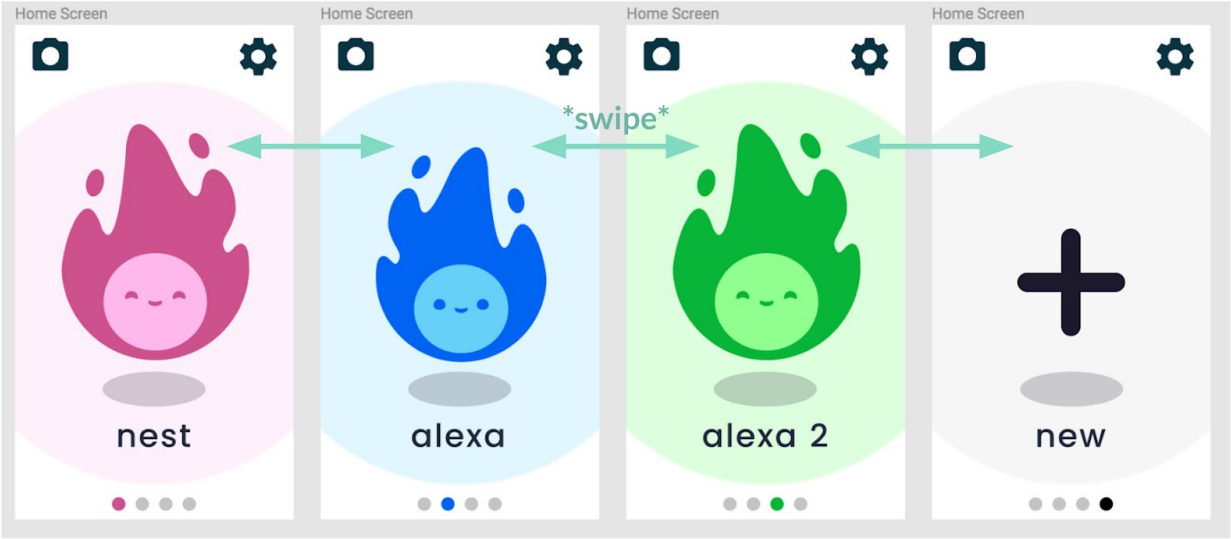

Figure 18: *Simple Task - Switch to Nest Sprite*

Major Usability Problems Addressed:

Our major usability problems (levels 3 and 4) are listed below. The medium fidelity prototype is represented by the black iPhone, and the high fidelity one is represented by the white iPhone.

1. Not intuitive to click on the Sprite on first open of app (Severity 4)

Figure 19: *Medium Fidelity vs High Fidelity Home Screen*

In our medium-fidelity prototype, we received feedback that users wouldn't know to tap on the Sprite when first opening the app, so we got rid of the tap action completely in the high fidelity prototype (Figure 19). We animated the Sprite as well. We also added an onboarding screen (Figure 2) for our high-fidelity prototype that explains some of the app's features.

2. Unclear how to pair smart home device and app (Severity 3)

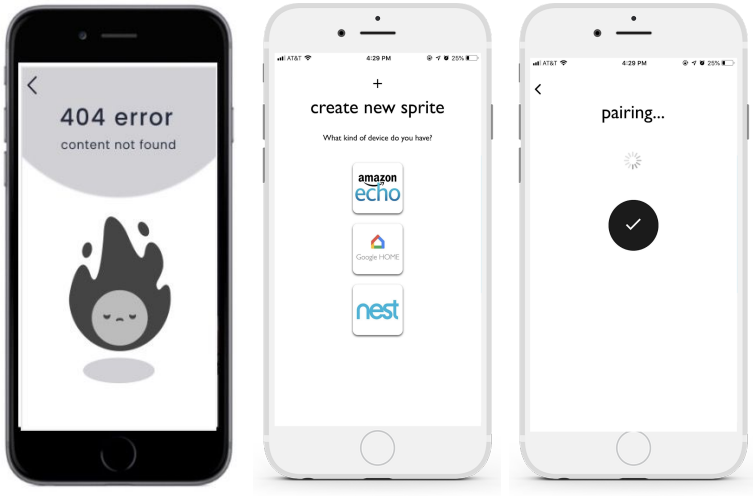

Figure 20: *Medium Fidelity vs High Fidelity Pairing Screen*

Our medium-fidelity prototype didn't show the process of pairing a smart home device, so we added a setup process (Figure 20). This process uses Wizard of Oz techniques to simulate the process of connecting a smart home device without actually doing so.

3. "Achieve" screen buttons are not clear if clickable (Severity 3)

Figure 21: *Medium Fidelity vs High Fidelity Achieve Screen*

We noticed that users didn't know that the achieve icons were clickable, so we designed the icons with a shadow to emphasize that they were buttons.

4. Create better icons on “Discover” screen - don’t use Sprite icon elsewhere besides home screen (Severity 3)

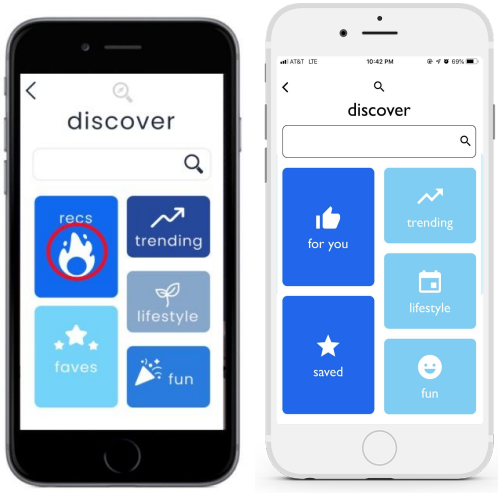

Figure 22: *Medium Fidelity vs High Fidelity Discover Screen*

We noticed that users were confused by inconsistent colors, text placement, and icons on the discover screen. We made the buttons more uniform by placing the text underneath each logo and only using two button colors. We also changed the "recs" button to "for you" and used a “like” icon instead of an image of the Sprite. We also renamed “faves” to “saved”.

5. Did not like swiping between Sprites - would prefer to see them all listed on a “collection” of Sprites screen (Severity 4)

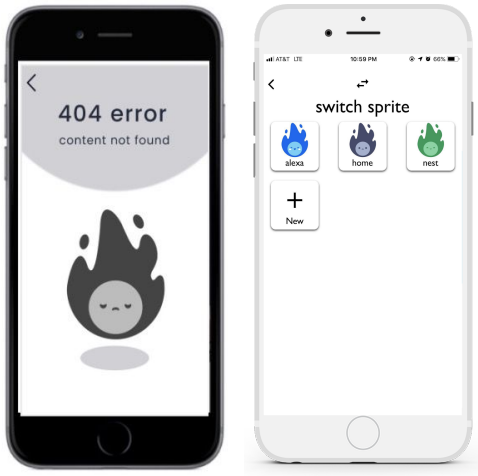

Figure 23: *Medium Fidelity vs High Fidelity Switch Sprite Screen*

The heuristics evaluators very much disliked swiping between the Sprites as a way to switch Sprites. They preferred our low fidelity prototype's way to switch Sprites - having them all on a collection screen, rather than purely swiping on the medium fidelity prototype. Therefore, we implemented both ways to switch Sprites in the high fidelity version. Users can choose between swiping between Sprites or referencing the Switch Sprite screen.

Prototype Implementation:

We used React Native and Expo for our High-Fidelity prototype. Expo allowed us to iterate quickly, since we could easily test each version on both a simulator and each of our phones. React Native provided a simple web-like format that helped us create layouts to cater to each screen size. We tested most thoroughly on Pixel 3XL and iPhone 7+.

However, React Native and Expo had a few downsides. React Native, while very flexible for creating prototypes quickly, is a web-based language more like Javascript than the languages our team knows best - C++ and Java. As such, our team had limited experience with these tools, so the majority of the technical app implementation fell on Clara, who took CS47 (Introduction to React Native) concurrently to learn the basics. Additionally, Expo's development framework prevented us from using Native Components (software libraries designed for Android or iOS, but not both). Without Native Components, we could not feasibly incorporate voice recognition, connection to Amazon and Google servers and thus actual Smart Home Devices, as well as other limitations listed below. We decided the increased flexibility Expo allows for developing and distribution was worth giving up those features, given they may not even be feasible.

The data for the app is stored using a library called AsyncStorage provided by Expo, allowing a user to return to the app in the same state and use it without an internet connection. The navigation framework of the app is an implementation of React Navigation's Stack Navigator. The Sprite character is a creation of Julea Chin's for this project. The actions used for the instruction library are selected from Amazon Alexa Skills library. The app is accessed through the expo distribution network via the link expo.io/@spritecs147/Sprite. An .ipa and an .apk are available on our website, but they are not the best way to experience our app.

We also used a few Wizard of Oz techniques:

- Sprite App does not log user into Amazon, Google, or Nest accounts, rather user

can tap on email or password fields to autofill data. No real device is paired, the entire connection experience is simulated.
- Sprite App does not connect to Smart Home Device for voice recognition, rather the instructions are on a 10 second timer. No audio is recorded or analyzed.

Also, almost all of our data was hard-coded:

- All actions are added by us and not downloaded to from on online database, but they are based on existing Amazon Alexa skills.
- Categories are not generated by user preferences, they are pre-selected Sprite "mood" is not dynamic over time, rather changes with number of instructions used.

Many features are missing from the high-fidelity prototype. If we had time, we would add the following:

- No user profile or login based on their Sprite information. All data is local.
- Only a limited number of Alexa-specific skills are available.
- There are a limited number of customization color options and achievement badges.

We did not add these features because they were not integral to our tasks, and we either did not have time or the background knowledge necessary to implement them.

Summary:

Most people only use a few of their smart home device's capabilities. Finding relevant new features is frustrating and time-consuming. Sprite gamifies the process of using a smart home device. With Sprite, exploring new features is fun and engaging. Throughout the quarter, we've met with users and iterated on our application's design to be more engaging and intuitive. This class has allowed us to explore the whole design process. Our team looks forward to gaining more design experience in the future.