CS194H: FINAL REPORT

Project: Petflix & Chill
Make meaningful connections through bonding over dogs

Team: Scott Buckstaff (Development)
Ken Chhan (Development)
Austin Chustz (Development & Product)
Hailey Spelman (Design & Usability)

Class: CS194H, Winter 2016

Website: http://web.stanford.edu/class/cs194h/projects/petflix+chill/
http://hci.stanford.edu/courses/cs194h/2016/wi/projects/petflix+chill/

I. Problem and Solution Overview

Many University graduates experience a stark transition from their student lives to their professional lives. They often go from four years of having been constantly surrounded by familiar faces and community to living in a new city trying hard to make new friends. Most graduates have a hard time meeting people they would like to see again outside of their place of work. Bars, gyms, and other places they frequent have proven to be poor places to make meaningful connections. This struggle to form lasting connections in a new community is not unique to college graduates and can be observed in older adults as well. We have found (and other research has shown) that man's best friend helps man make friends. However, many people have restrictions on their ability to own a dog for various reasons.

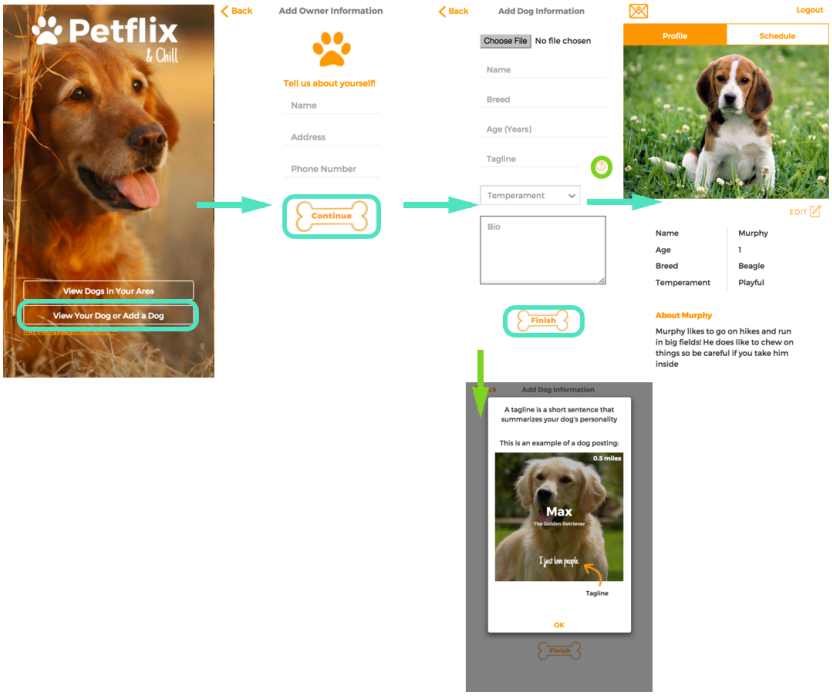
Petflix and Chill is a dog-sharing platform that connects volunteer walkers with local dogs and their owners. Dog owners who don't always have the time to give their beloved pets the time and attention they deserve can list their dogs on the service and have their dog walked for free by a member of the Petflix and Chill community. Walkers can browse the dogs in their area and message their owners to initiate a conversation and work out the logistics for a walk. While out on their walks, the walkers will benefit from the dogs' approachability and the ease with which they attract other people to interact with them. The drastic increase in the number of social interactions a walker is likely to experience in the duration of a single walk increases the chances that they will meet someone they will want to interact with again. Walkers can also build rapports with individual owners if they repeatedly walk their dogs. Petflix and Chill helps people make meaningful connections through bonding over dogs.

II. Tasks

We didn't begin with three representative tasks in mind for Petflix and Chill because it wasn't one of the focuses of CS 247. This quarter we determined that our three tasks would be the following:

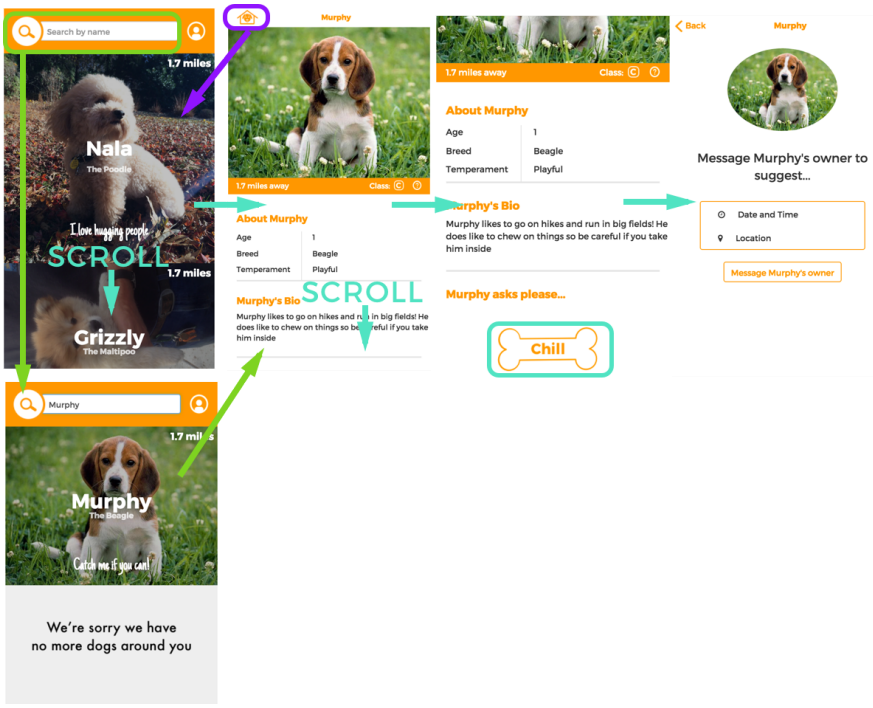
1. Signing up as an owner and listing a dog on the service (simple)
2. Searching for and selecting a dog (moderate)
3. Scheduling the time and place for handoff (complex)

We chose listing a dog on the service as our simple task because it was the most straightforward. The entire action is completed with only two forms.

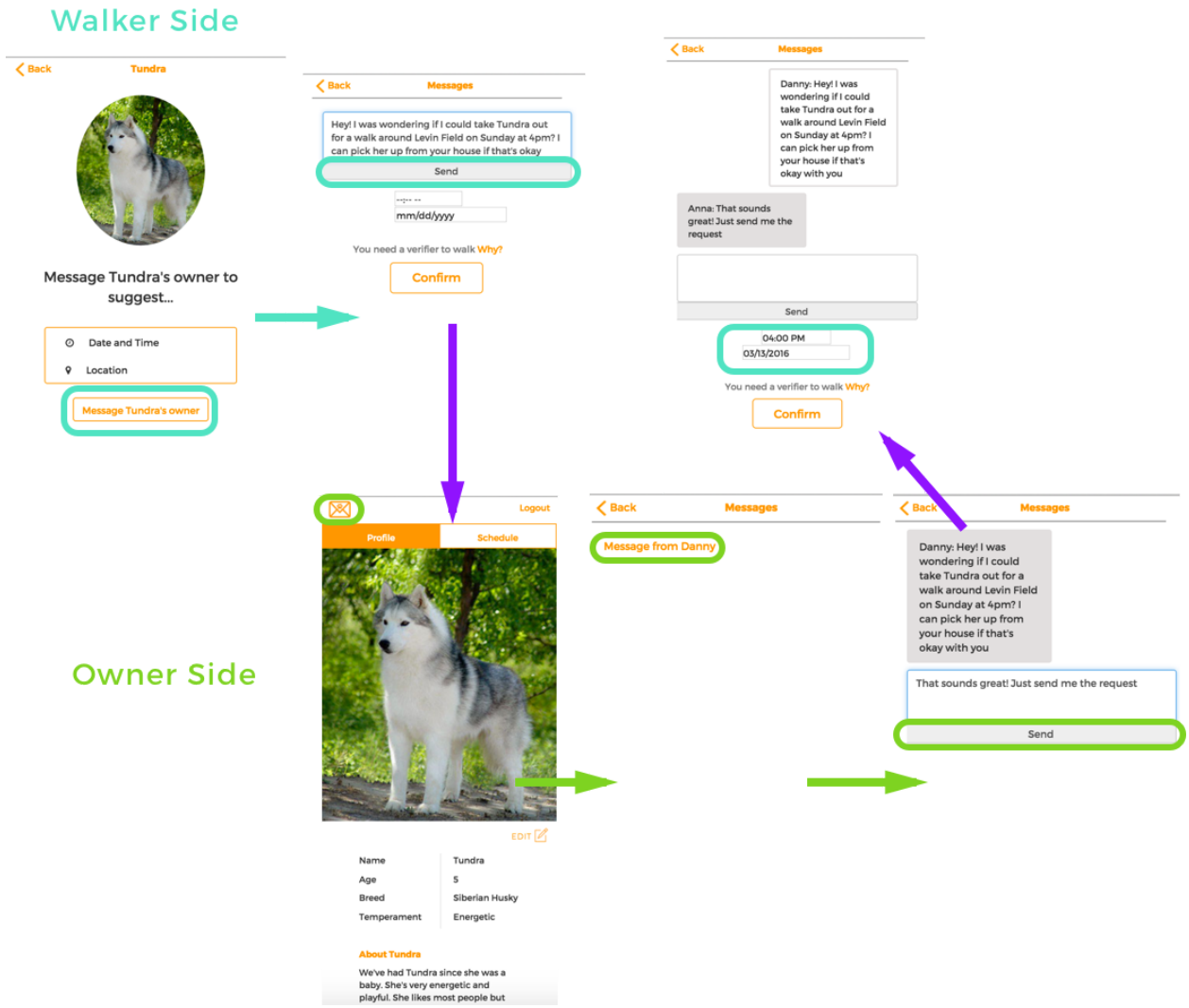


**Green route represents the option of clicking on the information pop-out for the Tagline

We chose search and select as our moderate task because it encompasses the walker's flow for chilling with a dog. The walker can search for a dog by name or just browse the list view to view a dog's profile.



We chose scheduling the time and place for handoff as our complex task because it was logistically the toughest part of the app. We tried to address this complexity multiple ways throughout both quarters and we found that the simplest way to address it was actually through in-app messaging. The walker and owner can decide on a time and place on their own through the messenger and then the walker sends a request for the owner to confirm for that time.



**Purple arrows are transitional arrows between Walker and Owner while blue arrows represent Walker actions and green arrows represent Owner actions

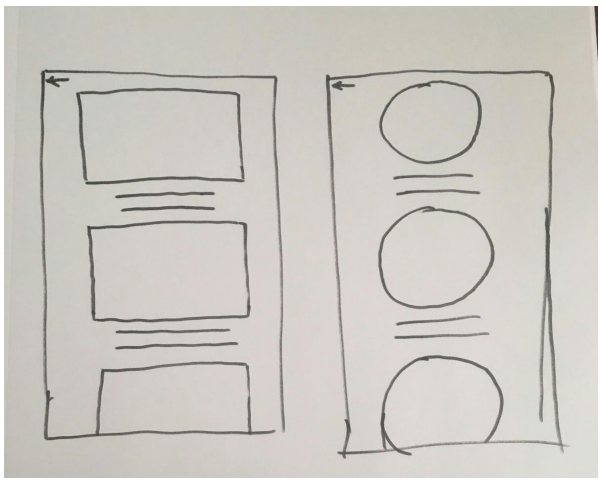
III. Design Evolution

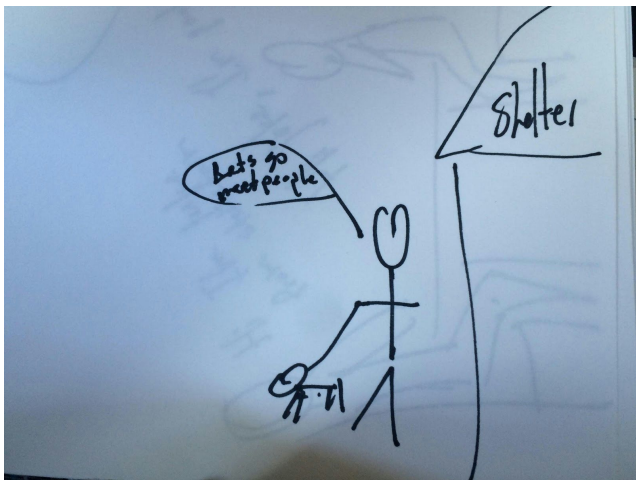
The idea of Petflix and Chill came from a fairly extensive ideation session. The area of our focus was urban computing and we had framed our main question that we wanted to find a way to fix as "how can we help newcomers to a community meet people and integrate more easily?" For the majority of the session we were spouting out traditional ideas that didn't seem to have any spark. We were at the end of our allotted time in the ideate part of the Design Thinking process and one of the requirements for our next deliverable was to prototype a dark horse idea. Scott had been jokingly talking about making a product for dogs all quarter and just wrote Petflix and Chill on the whiteboard. From there our group decided that since we needed a dark horse prototype any way that we would humor Scott and just prototype it.

We didn't really give the idea much thought at first and just sort of prototyped it to get the assignment done. Our first sketches were only of what would eventually become our list view. The original idea was that you could rent a dog for a time period. We were also very interested in working with pet shelters to help people find dogs. Our first notes on the topic were simple:

- Dark Horse -- Petflix & chill (share your dogs)
    - "rent a dog"
    - Shelters make it possible to walk and play with animals (dogs) while meeting new people

Paper Prototype

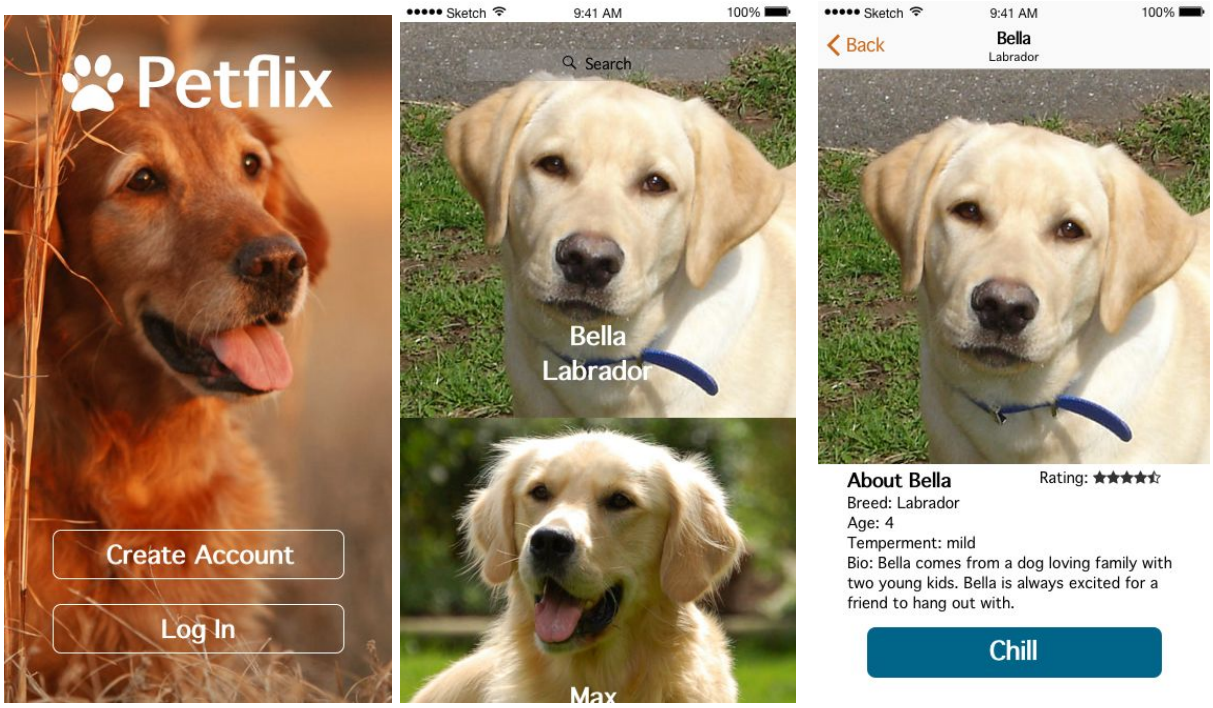




When it came time to decide what product our group actually wanted to build out we decided that it would be best to build a neighborhood exploration platform. However, Michael Bernstein called us into his office hours to express his concerns that our idea was too mainstream. He told us that our group was too creative, too unique, and too different to make such a traditional project. So with a bit of encouragement we decided to build out Petflix. Our first live prototype was incredibly simple, but it had the core functionality that Petflix revolves around: searching for and walking a dog. We were also advised that it would be easier to avoid using shelter dogs because of an array of legal complications

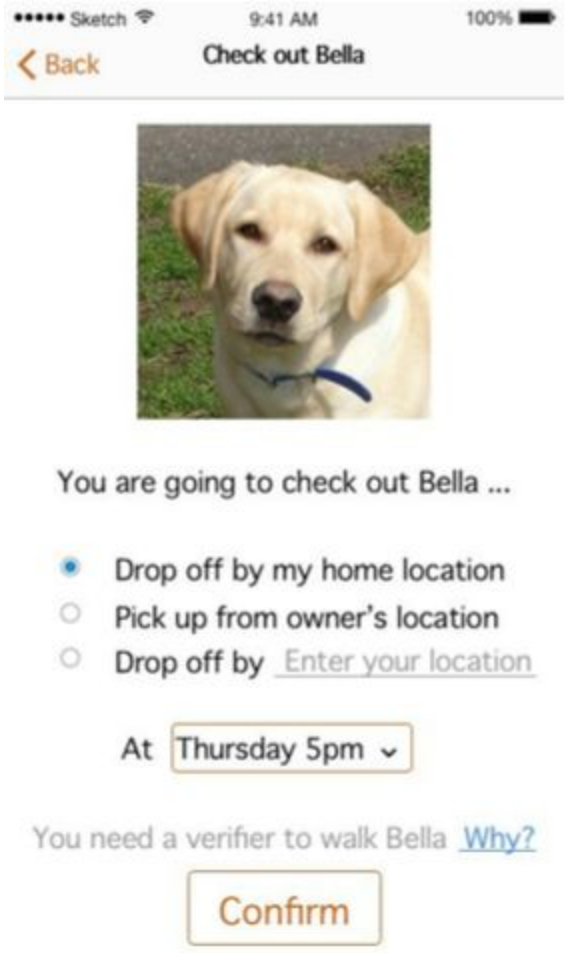
revolving around liability. One of the biggest UI improvements we made was the dog profile. We added in a bio for the dog and some basic info on the dog.

Medium-fi Prototype



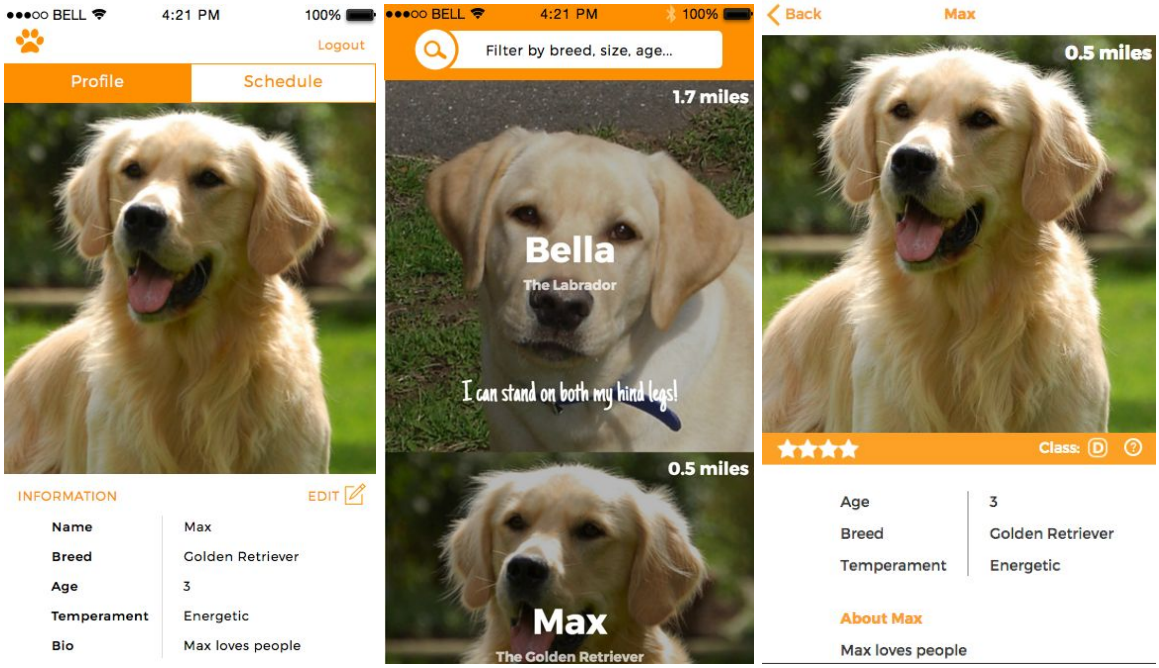
When we went out and tested our first prototype we learned a ton from our test users. The walkers loved the idea. Their only suggestions were that they wanted a way to schedule with the owner and that they felt that the app needed a facelift. The owners weren't as sold on the idea as the walkers were. Their main question was: "Why should I trust this app and the random person that is walking my dog?" This feedback sent us back to the whiteboard to try and design solutions to these problems. Our solution was our dog walker verifiers, people who certify individuals to ensure that they are able to walk dogs and to add in scheduling. We also decided that we wanted to have two sides to the app: a walker side and an owner side. From here it was time to build a functioning hi-fi prototype. Our hi-fi prototype made our screens from our medium-fi prototype live as well as additional pages with scheduling and information about dog walker verifiers.

Hi-fi #1 Prototype additional screens



At the start of CS194h we had functioning front-end for the walkers but we had no idea how we wanted to implement the owner side of the app, how we wanted to implement the backend, or how we were going to deal with user accounts. Almost all of our data was hard coded. We decided that owners and walkers should have different accounts and we basically decided to create two apps that were connected through the backend. For our hi-fi prototype #2 we did a significant facelift as well as add an owner side. Our largest user interface changes were adding in the ability to log in as an owner or a walker, adding the ability to add dogs to the database, and made the app feel modern. We added little things all over the app that help walkers get a better feel for the dog that they are walking (such as comments on the dogs and the tagline on the dogs image in the listview.

Hi-fi Prototype #2 design changes and added owner profile view

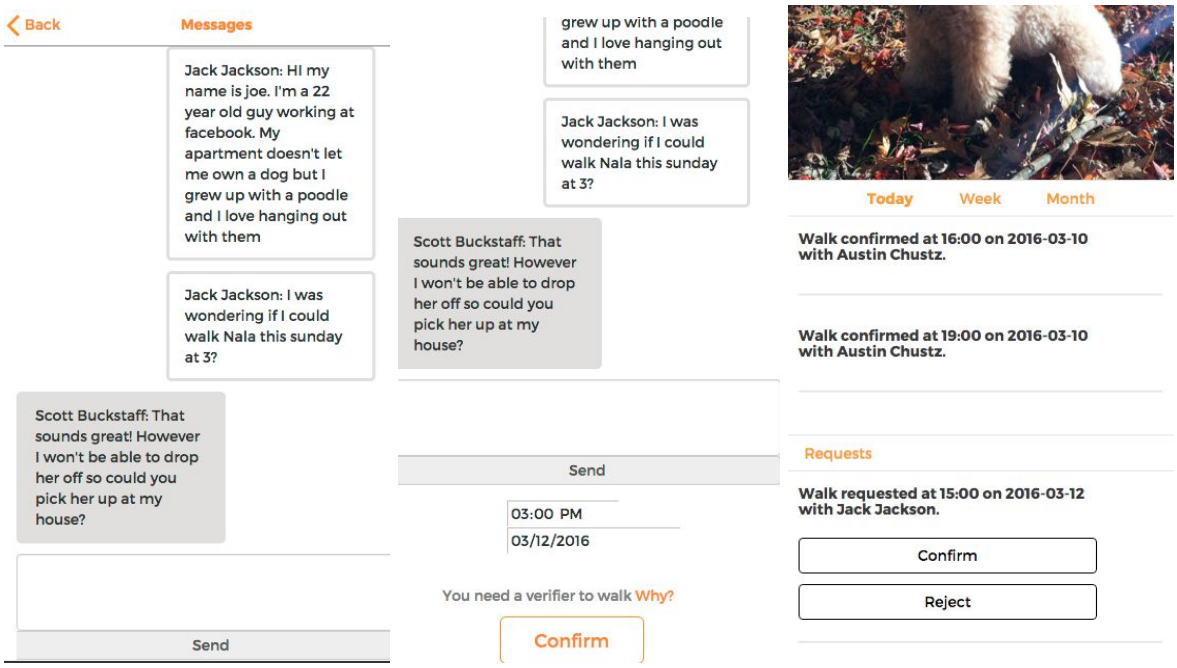


From here we felt confident in the design of the app and felt that our main challenge left was removing all wizard of oz techniques from the app. When we went out and tested with sample users we found that users may want to be both a walker and an owner. We decided that we would rejoin the two split users groups. We came up with the idea that when a user signs up for the app they are neither a walker nor an owner. They then would have the choice to enter the walker side of the app and sign up as a walker or enter the owner side of the app and sign up as an owner and add their dog. The user always has the option of going back to the main landing page and signing up for the other section of the app. This functionality is clearly defined in the final interface section.

We also had feedback from owners that the verifiers were not enough for them to feel comfortable with strangers walking their dog. Instead of trying to solve the issue of owners not feeling comfortable with strangers we reframed the problem to “how can we make owners not feel that walkers are strangers?” At this point we did extensive industry research and we decided to model the airbnb approach where the owners and visitors have extensive talks with each other before agreeing on a transaction. We modeled this through our messaging system where we encourage owners and walkers discuss scheduling and where to meet as well as simple info about each other (or the dog in the owner's case) to add in a degree connection. This had the added benefit of making connections for the walker in their new community. We also heard that owners wanted the option to accept or reject a request. So we added in scheduling functionality that allows users to see their scheduled walks and gives the owner the option to reject or confirm requests from walkers.

To accomplish this we added in a schedule page for the owner and we added in a walker profile that included the walker's upcoming schedule.

Hi-fi Prototype #3 Messaging and Scheduling UI



Over the course of the two quarters we found that in person testing was the most efficient way of getting feedback on our project. The process that we went through changed depending on what we wanted feedback on. If we were looking for design and UI feedback having five to six users test our app and getting their feedback as well as watching where they got stuck really helped us streamline the user flow. On the other hand when we were trying to figure out what would make people use our app we would interview individuals on whether they would use the app under different conditions. For example when we were trying to figure out what would get owners to put their dog on the app we found it much easier to interview them than to actually give them a live demo.

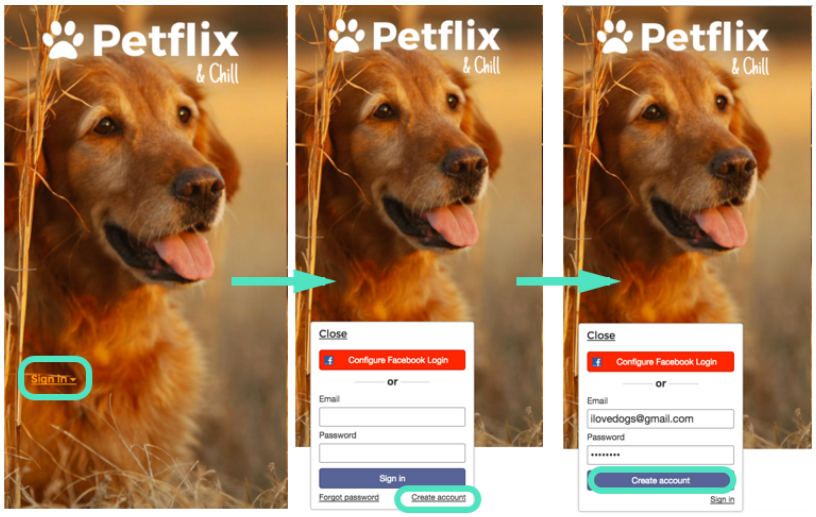
IV. Final Interface

When designing Petflix and Chill, we broke the functionality down into four main categories: sign-up, dog selection, messaging, and scheduling. While all of these elements are intertwined within our tasks, they are distinctly separate operations.

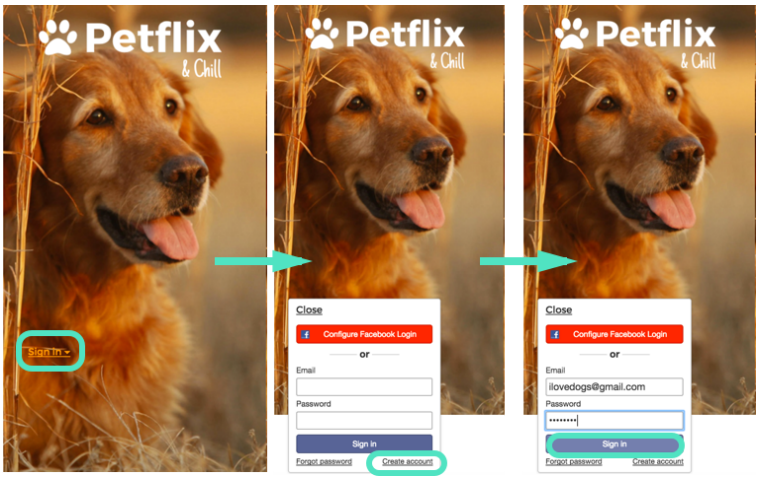
Sign-up/Login

Our key goal for sign-up (which is tied to logging in) was to make the process both streamlined and flexible. In this, we opted for a single login and "create account" area for both walkers and owners. Once the user has logged in or signed up with their email address and password, they are presented with two options: to "View Dogs in Your Area" or "View Your Dog or Add a Dog." The first of these two options represent the Walker side of the app and the second represents the Owner side of the app. Having both of these options available to the user makes the app more flexible to allow a single user to be both a Walker and an Owner. After clicking either of these options, the user is prompted to fill out information about themself. If the user is on the Owner side of the app (meaning they clicked "View Your Dog or Add a Dog"), they are then prompted to fill out information about their dog. One important decision we made on that page was to include an information pop-out that explained what a "Tagline" is for your dog and showed a sample dog listing in order to clarify the purpose of that information feature. At the end of this process, a user on the Walker side would be directed to the list view of all available dogs while the Owner side directs to the Owner's dashboard. Given the small number of screens the user must pass through and the straightforward nature of these screens, we believe that we made it a streamlined and effective process. The user interface design of this process can be seen below in the following diagrams.

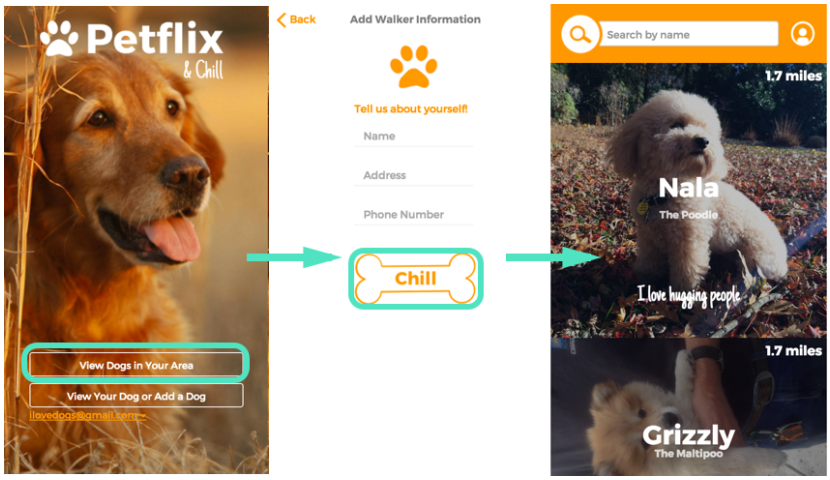
First time Sign-up



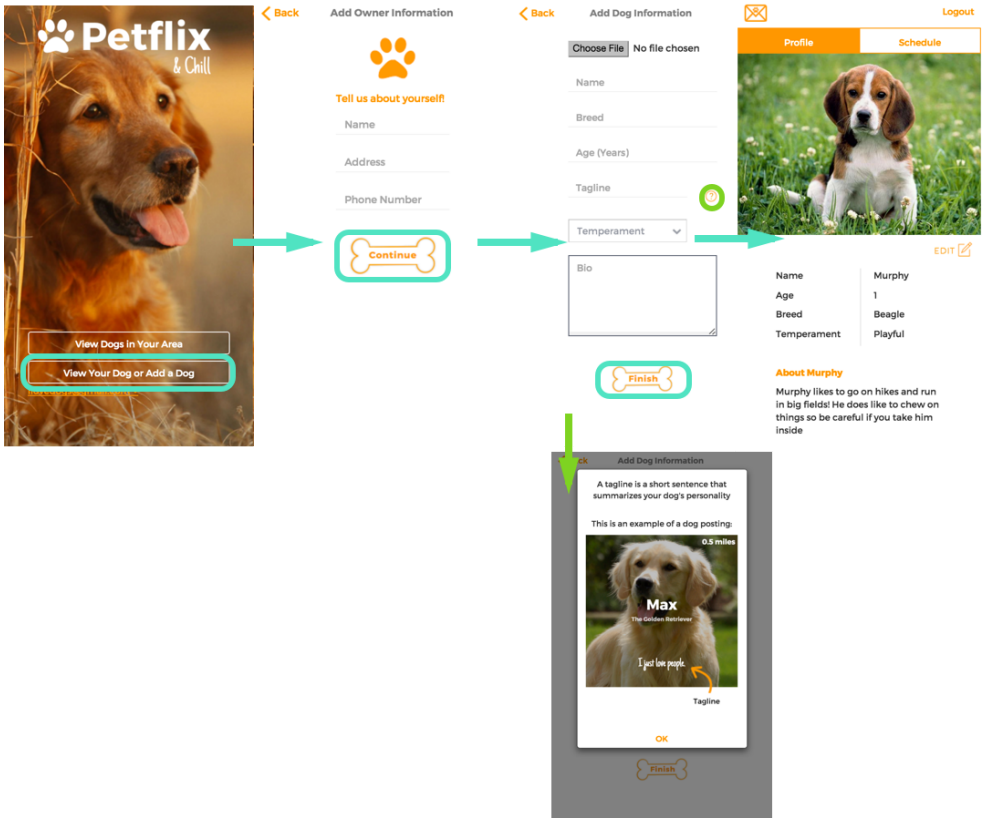
Sign-in



Walker Sign-up (continued from either signing in or creating an account)



Owner Sign-up (also continued from sign-in/create account)

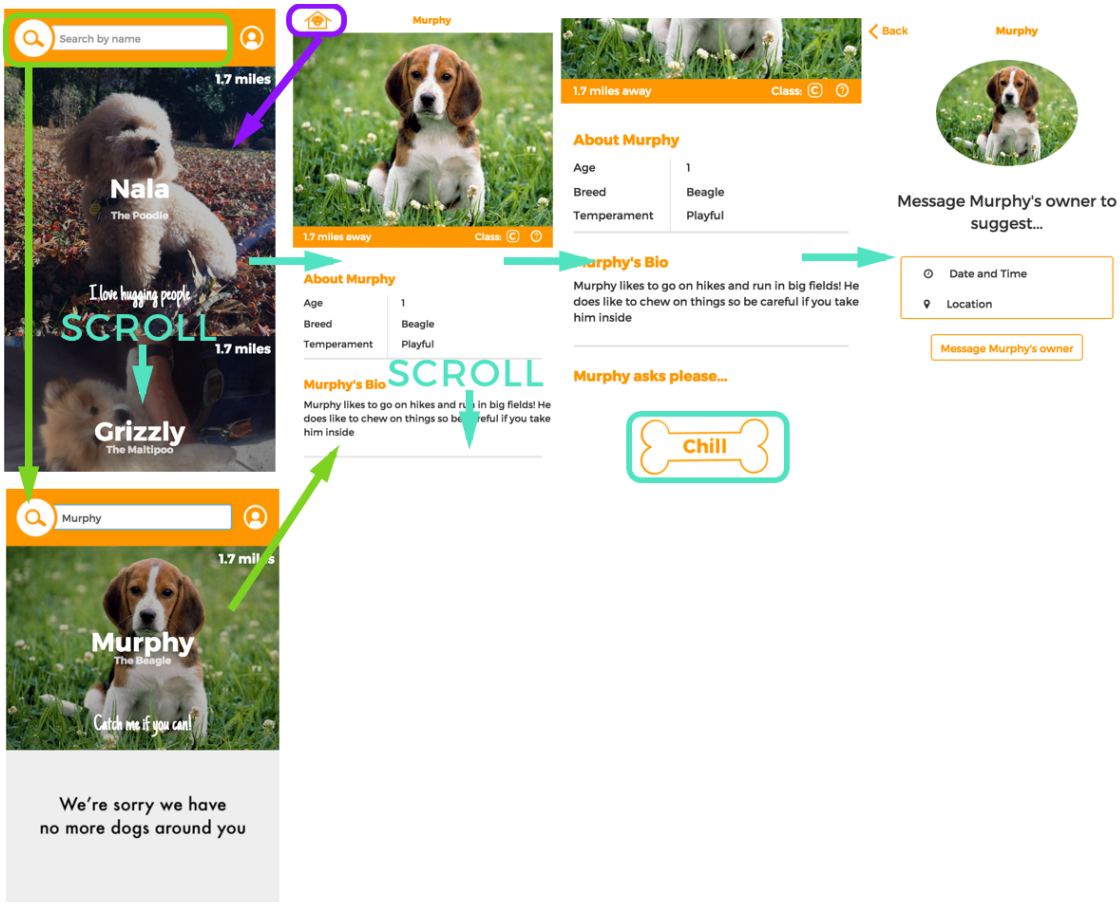


**Green route represents the option of clicking on the information pop-out for the Tagline

Dog Selection

Dog selection is the core functionality aspect for the app's purpose: to connect people with dogs to walk (in the hopes that it will help them integrate into a new community). Thus, we spent the majority of our iterative design process making it as clear, informative, and enjoyable as possible. For the sake of clarity, we have all available dogs to walk in a scrolling list-view structure where each dog has a photo, their name, their breed, the distance the dog is from you at the moment, and a short tagline that describes the dog. From this list-view, the user has the option to either scroll down and browse all of the available dogs or search for a specific dog by name (seen in the green arrow route in the diagram below). Once the user reaches the dog's profile (through either browse or search), they are presented with more detailed information about the dog-- such as temperament and a short a bio. From there, the user can select the dog by scrolling down the profile and clicking to "Chill" with it, or the user can return to the list-view by pressing the home button in the top left corner of the profile (exemplified by the purple track on the diagram below).

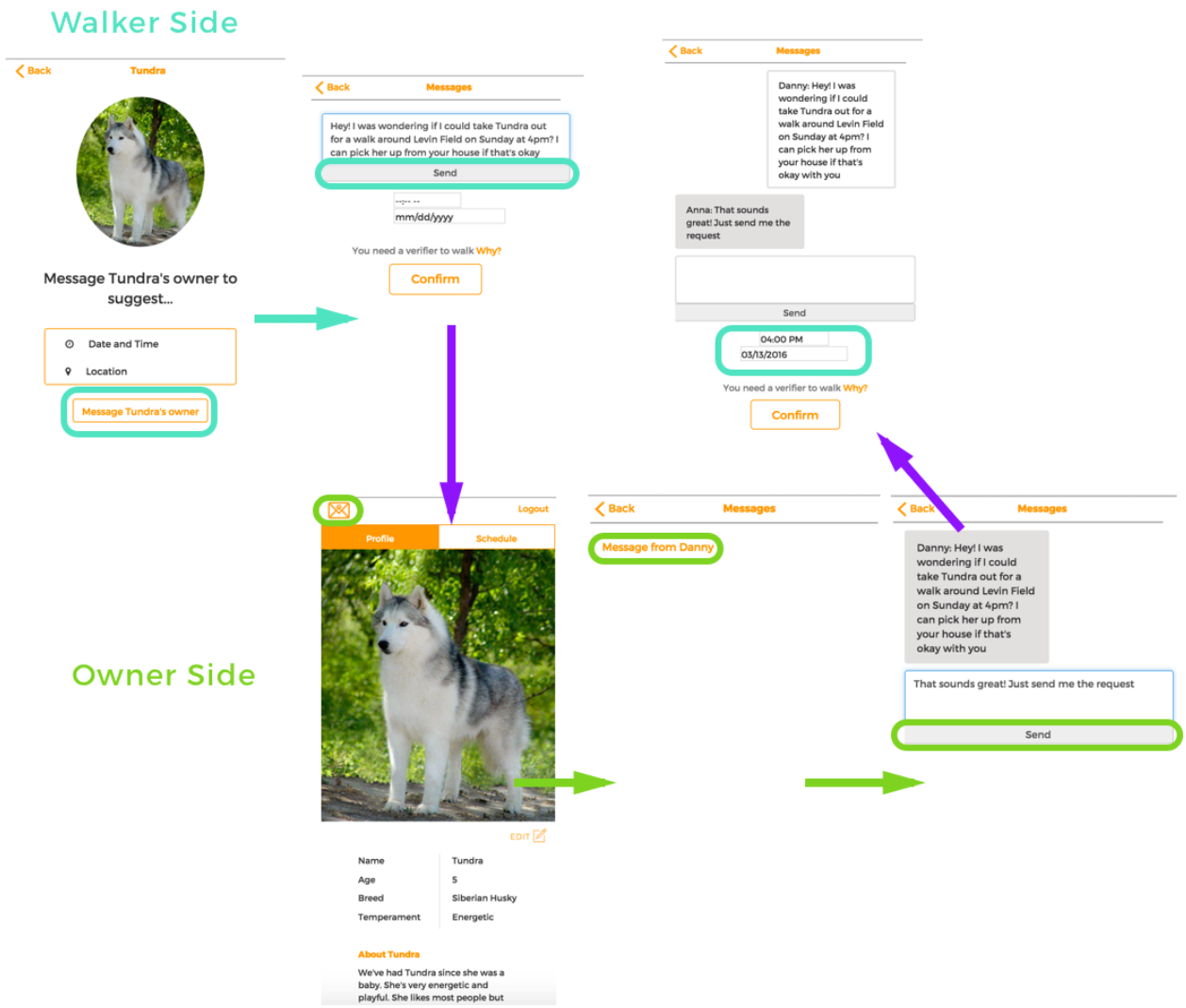
Dog Selection



Messaging

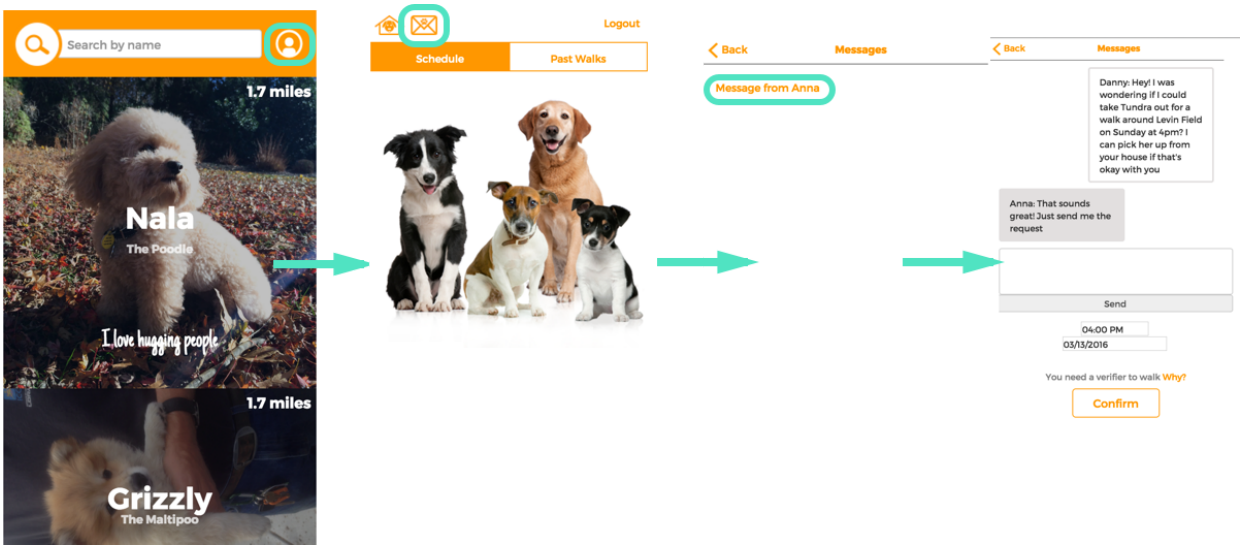
We decided to implement messaging late in our design process as a solution both for deciding on a time and place of dog handoff as well as a for a way to initiate long-term involvement with the dog (by establishing a talking relationship with the owner for repeated walks). The main design goal of this function was to facilitate easy scheduling by allowing the owner and walker to message to decide on a time and place rather than forcing the owner to input their availability every week. In the diagram below, you can see that the Walker initially accesses messaging by selecting a dog to chill with and proceeding forward to send a message to the dog's owner to schedule the handoff of the dog. The Owner then replies to the Walker to confirm that time and place and the Walker can then send a schedule request to the Owner. Both the Walker and the Owner can access their messages through the message icon on their dashboard (seen for the Owner in the diagram below and for the Walker in the second diagram), while the Walker has the option of staying on the post-Chill button page to continue messaging. This design makes it so that messaging has a heavy involvement in scheduling.

Messaging Example between Walker and Owner



**purple arrows are transitional arrows between Walker and Owner while blue arrows represent Walker actions and green arrows represent Owner actions

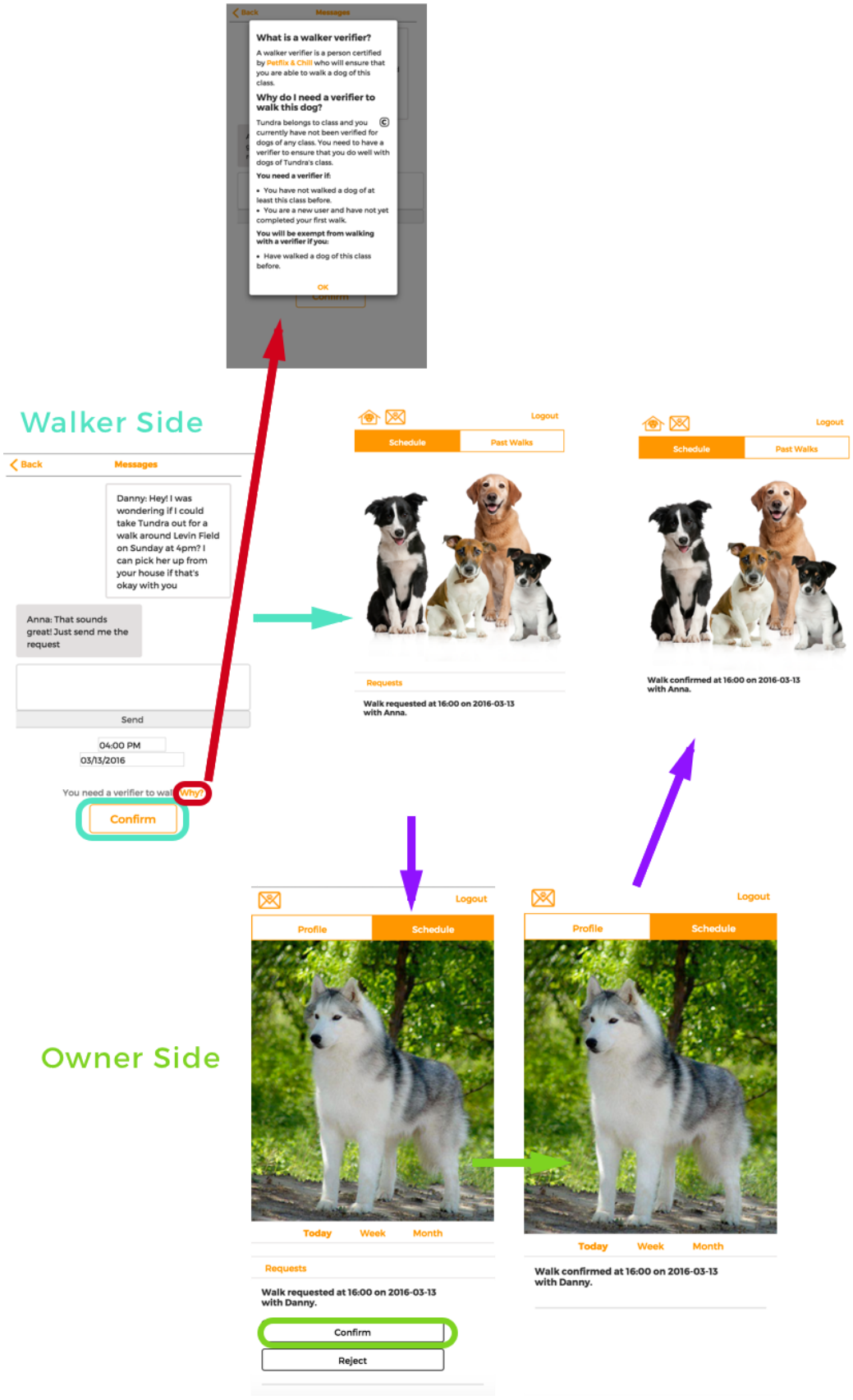
Walker Side Access to Messaging through Dashboard



Scheduling

As mentioned above, scheduling is facilitated through messaging. Scheduling is meant to be simple for both the Walker and Owner and thereby needs little explanation. After messaging with the Owner, the Walker sends a walk request at a specific time and date by clicking “Confirm” at the bottom of the Walker messaging screen. The request then appears on the Owner’s dashboard under the “Schedule” tab and the Owner chooses to accept it or not. This decision then appears as a response on the Walker’s “Schedule” tab. In this process, the Walker is introduced to the “Walker Verifier” feature that requires walkers of a new dog class be verified before they can walk that class (seen through the red path in the diagram below).

Scheduling Example between Walker and Owner (continued from Messaging above)


Back
Messages
What is a walker verifier?
A walker verifier is a person certified by Petflix & Chill who will ensure that you are able to walk a dog of this class.
Why do I need a verifier to walk this dog?
Tundra belongs to class and you currently have not been verified for dogs of any class. You need to have a verifier to ensure that you do well with dogs of Tundra's class.
You need a verifier if:
• You have not walked a dog of at least this class before.
• You are a new user and have not yet completed your first walk.
You will be exempt from walking with a verifier if you:
• Have walked a dog of this class before.
OK
Walker Side
Back
Messages
Danny: Hey! I was wondering if I could take Tundra out for a walk around Levin Field on Sunday at 4pm? I can pick her up from your house if that's okay with you
Anna: That sounds great! Just send me the request
Send
04:00 PM
03/13/2016
You need a verifier to wal
Why?
Confirm
Logout
Schedule
Past Walks
Requests
Walk requested at 16:00 on 2016-03-13 with Anna.
Logout
Schedule
Past Walks
Walk confirmed at 16:00 on 2016-03-13 with Anna.
Owner Side
Logout
Profile
Schedule
Today
Week
Month
Requests
Walk requested at 16:00 on 2016-03-13 with Danny.
Confirm
Reject
Logout
Profile
Schedule
Today
Week
Month
Walk confirmed at 16:00 on 2016-03-13 with Danny.

Things Left Unimplemented

Given our time and work limitations, our final app was left with many things unimplemented. Firstly, we didn't write any code to validate neither walker nor owner signup information except for empty input checking. We also left some dog owner profile features left unimplemented, namely dog profile editing, specific "please do/do not…" requests for walking a dog, removing confirmed walks, and marking walks as confirmed. Furthermore, currently each dog can only have one photo and walkers cannot comment on a dog regarding their dog walking experience. In terms of dog walker implementations, implementing filtering for the dog list view was harder than expected because it required restructuring our entire data structures, and was thus left unimplemented. We also would've liked to have been able to implement a feature to share the location of every dog walker with each other.

V. Making it Real

Our team of four consisted of three full-stack engineers and one pure UI/UX designer. Our project was built using Meteor. Meteor is a JavaScript framework that produces cross-platform code (web, Android, and iOS). Since the front-end was made using HTML5/CSS3/JavaScript and the back-end was done using MongoDB/Meteor Collections, all of our developers were fit to make both front-end and back-end contributions and bring our designer's mockups into fruition. Meteor was also an ideal choice because it has the power to compile a mobile-web app for iOS and android.

Currently, there are dog walking apps that are succeeding purely on monetizing dog walking services (e.g. Wag). Our project provides the same service, but the walkers can instead be thought of as being paid by using the dog to meet new people. Being an alternative dog walking service that still benefits dog owners who are too busy to walk their dogs and dog walkers, we feel our project would be able to gain and maintain popularity and traction.

All of our users can be considered our customers. Our two target groups, dog owners and dog walkers, are both providing and receiving a service when using our app. Dog owners receive the benefit of having their dog walked, while they allow the walker to use their dog to catalyze forming new relationships. Conversely, dog walkers are providing a dog walking service while benefitting from the new connections they make while walking a dog.

We can make a reasonable estimation of our market size by first looking at other dog walking services to approximate how many dog owners are too busy to walk their dog. Then, it may be reasonable to extrapolate that number to a larger size because we are providing the same service, but for free. Furthermore, we can approximate the number of walkers from age demographics, particularly in the age 22-26 range (i.e. recent college graduate ages).

The expected costs of our project would stem from server costs and maintenance, dog walker verifier training, marketing, advertising, and possible lawsuits. We would start monetizing by placing in-app advertisements. After our dog database grows large enough, we will start using targeted advertising to advertise products and services for specific dog ages, sizes, breeds, and temperaments. Finally, we would charge a fee on dog walker verification training and provide dog insurance services for while dogs are walked.

In making this product, we wanted to lower the barrier for making new meaningful relationships, and found that dogs are great mediators for that. Dog owners will be able to get their dogs walked, free of charge, and dog walkers will have an easier time making neighborhood friends, creating tighter communities.

VI. Summary

Our key innovation was finding identifying two target audiences that would mutually benefit from each other: dog owners lacking time to walk their dogs and relocated new graduates trying to make new meaningful connections. Dog owners will be able to share their dogs to be walked by new graduates. By using dogs to facilitate and catalyze forming new relationships, new graduates will be able to meet new people at a much faster and effective rate than without a dog, while owners benefit by having their dogs walked. Using this model, we can eliminate the need to monetize the primary function of our project. We hope to drastically lower the barrier for a newcomer to effectively assimilate into a new neighborhood and effectively create greater tight-knit communities.