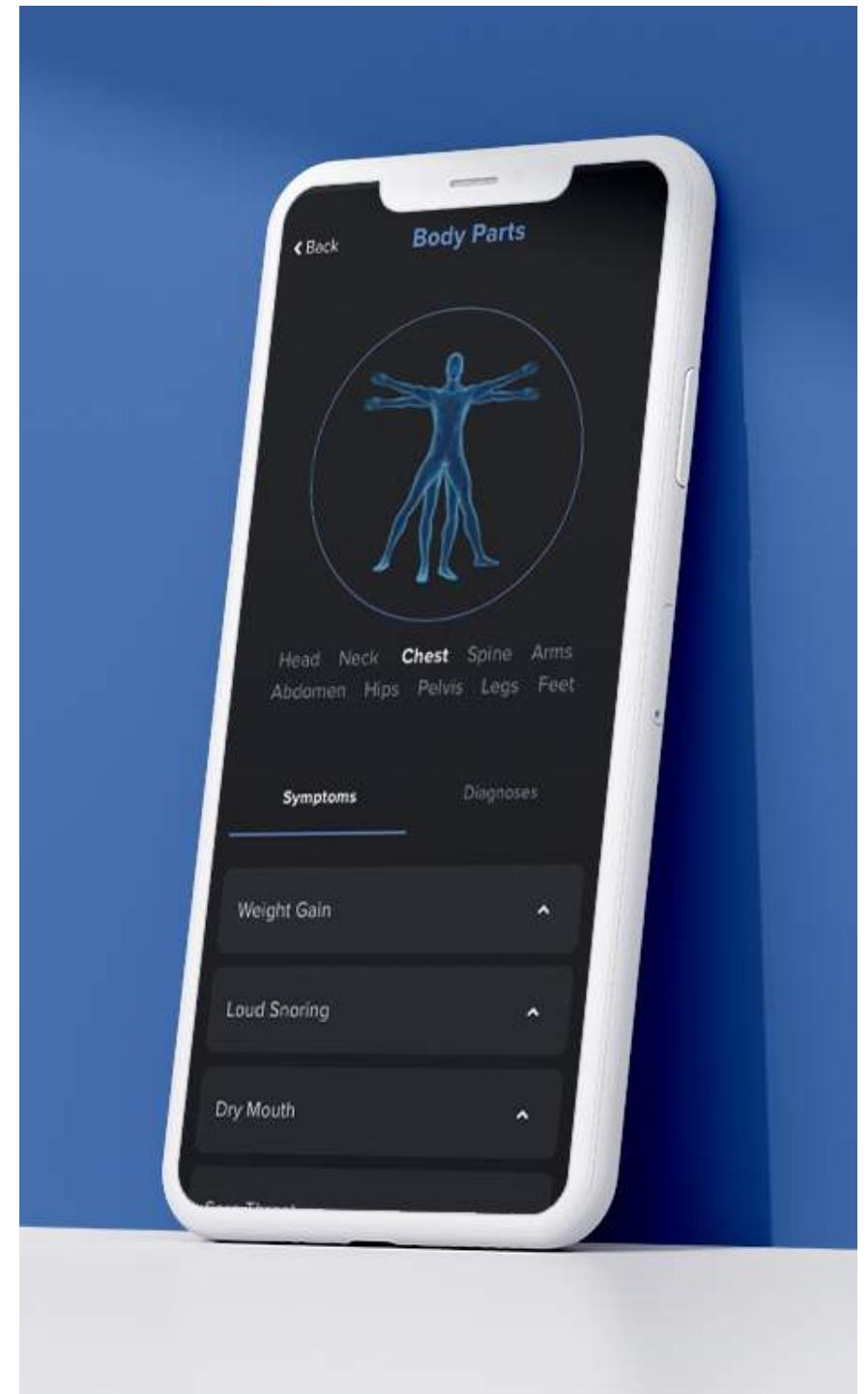


James Rountree

I am a Product Designer and Developer passionate about solving complex user experience challenges.

JimboRountree.com
LinkedIn@James-Rountree
James.D.Rountree@Gmail.com
352 328 6777



Da Vinci

U.S military veterans need help obtaining monetary compensation for disabilities sustained from their war efforts. In hopes to create an effective solution for this problem, we created Da Vinci: a scalable web based platform intended to help Veterans obtain the medical benefits they deserve in the shortest period of time.

The following case study was assembled by myself and my teammate, Hugo Ramos

His work can be seen at hugoramos.co



Challenges and Goals

UX Challenge

Some of the medical questionnaires clients must fill out can be hundreds of questions long. One of the biggest challenges is breaking up the long monotonous forms with medical terms. Another significant challenge is creating a central hub where clients can check and intuitively update disabilities for further service connections.

UX Goals

Translate and display complex medical terms into laymens terms so its easier for users to comprehend having no medical background. Creating a concise and easy to comprehend summary of potential connected disabilities that users can use for reference in filing for future benefits.

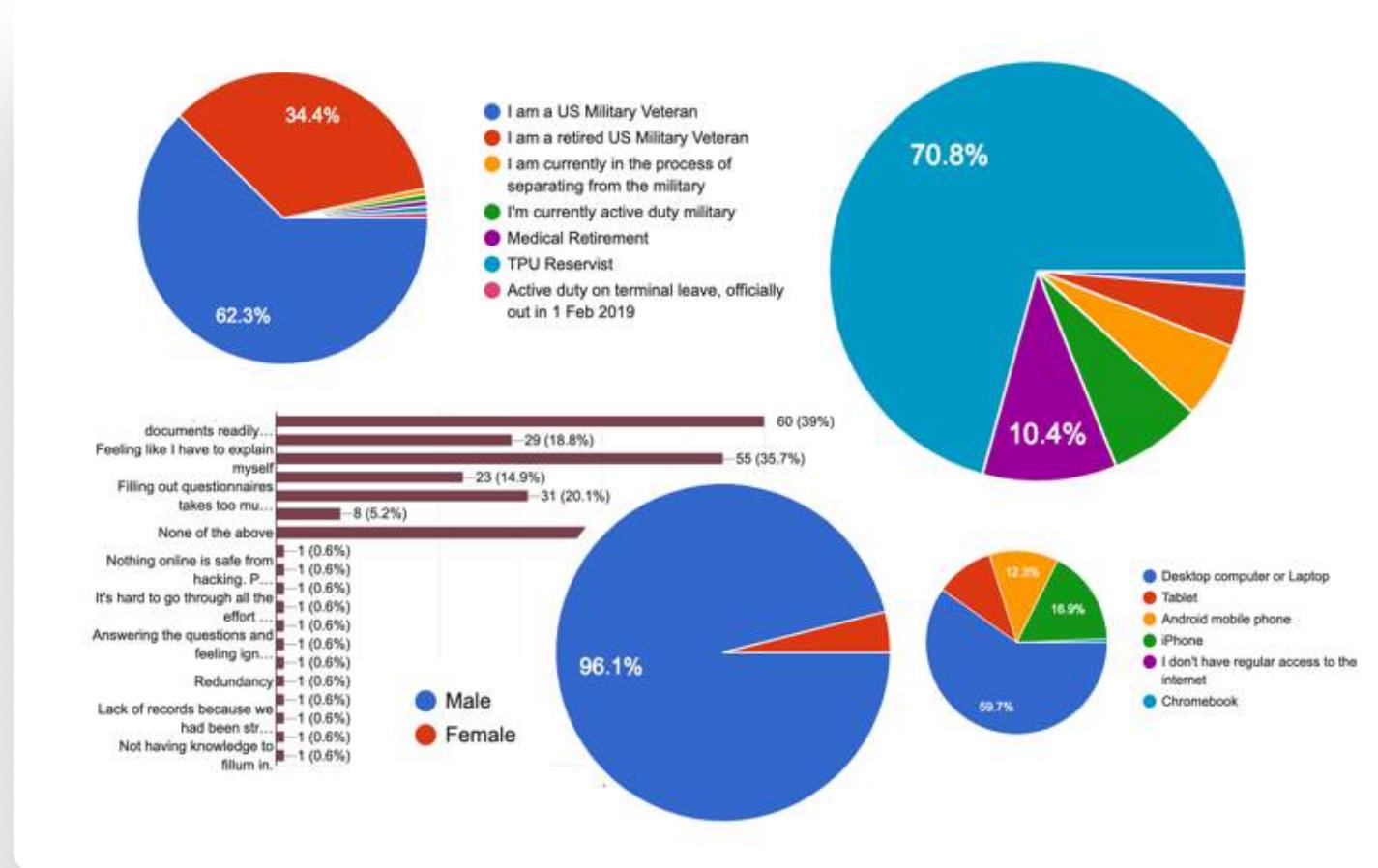
Responsive First

When meeting with product owners one core feature was for veterans to have full function support on mobile devices. From the initial wireframes we began designing with the responsive first approach, accounting for screen sizes everything from an iPhone 5 all the way up to a full television.

The screenshot shows the Da Vinci application interface. On the left is a dark sidebar with navigation links: 'My Dashboard', 'Current Ratings' (which is the active page), 'Body Parts', 'Body Systems', 'Workplace', 'My Account', and 'Help & Support'. The main content area has a light gray header with the text 'Da Vinci > Current Ratings' and a progress bar. Below the header is a table with columns 'Disability', 'Rating', and 'Date'. A text input field says 'Please enter your first [Disability]' with a placeholder 'Say something like "Tylenol" or "None"' and a dropdown arrow. Below this is a section titled 'Or choose from the common disabilities below:' with a grid of buttons for 'Plantar Fasciitis', 'Pes Planus', 'Erectile Dysfunction', 'Tinnitus', 'Degenerative Joint Disease', 'Depression', 'Post-traumatic Stress Disorder', 'Sleep Apnea', 'Knee', 'Gastroesophageal Reflux Disease', 'PTSD', and 'Diabetes'. To the right of the table is a dark sidebar with sections for 'Disability' (set to 'Sleep Apnea'), 'Rating' (set to '%'), and 'Effective Date' (set to 'MM / DD / YYYY').

Research

To learn more about our user base, we sent out a brief survey to gather qualitative and quantitative data to help form the basis of our design process. We obtained 154 responses from US military veterans and received key insights that would help us understand the problem from the user's perspective.



User Empathy

Based on our research, we decided to create user personas to form a deeper understanding of our user-base in order to meet their specific needs. Predicated upon demographics, personality type, and other psychographic information, we came up with three archetypes that met the criteria of the user we're designing for.



Vietnam Vince

Retired US Veteran

"I'm at least keyboard literate, so I rather deal with a live person than a computer screen."

Gender: Male

Age: 55

Military Status: Retired Veteran

Patient Conditions:

- High Blood Pressure, Depression, PTSD, Diabetes (Agent Orange Exposure)

VA Services Utilized:

- My HealtheVet: secure messaging, Rx refill, blue button, lab results
- Peer Counseling Services
- Smoking cessation services
- Treatment for Diabetes from Agent Orange Exposure

Characteristics:

- Cautious, Opinionated, Loyal, and Skeptical

Desires:

- I want to be in control of my medications
- Make sure someone responds to me (Secure messaging and appointments)
- Let me see all of my health records

Closest Relationships:

- Grand children
- Other Veterans

Technology Devices:

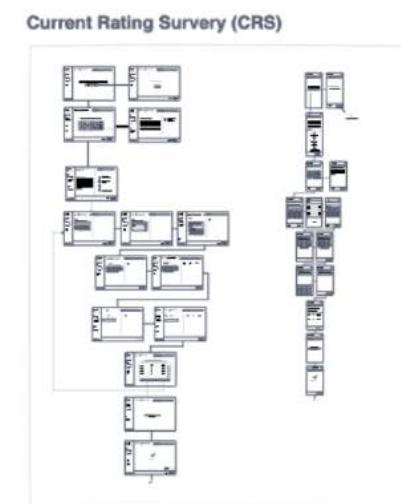
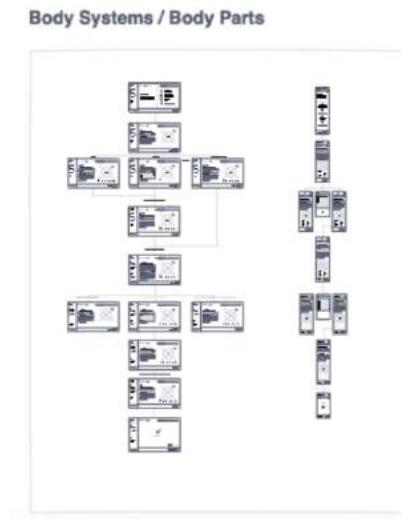
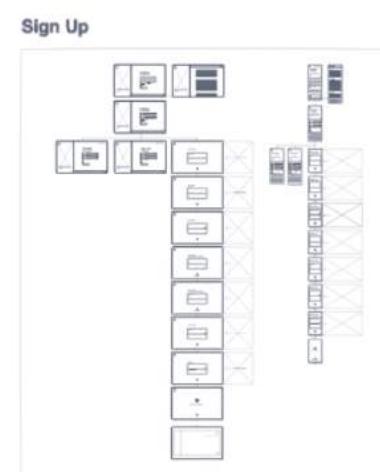
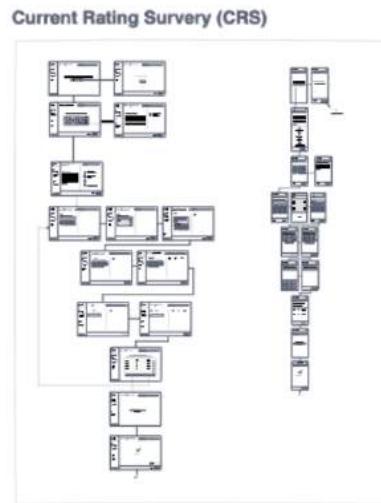
- iPhone5 (Smartphone)
- PC (Home Desktop)
- Landline

Technology Pain Points:

- Don't feel that apps and websites are secure
- Problems with sign in
- Too much information on VA websites

Quick Concept

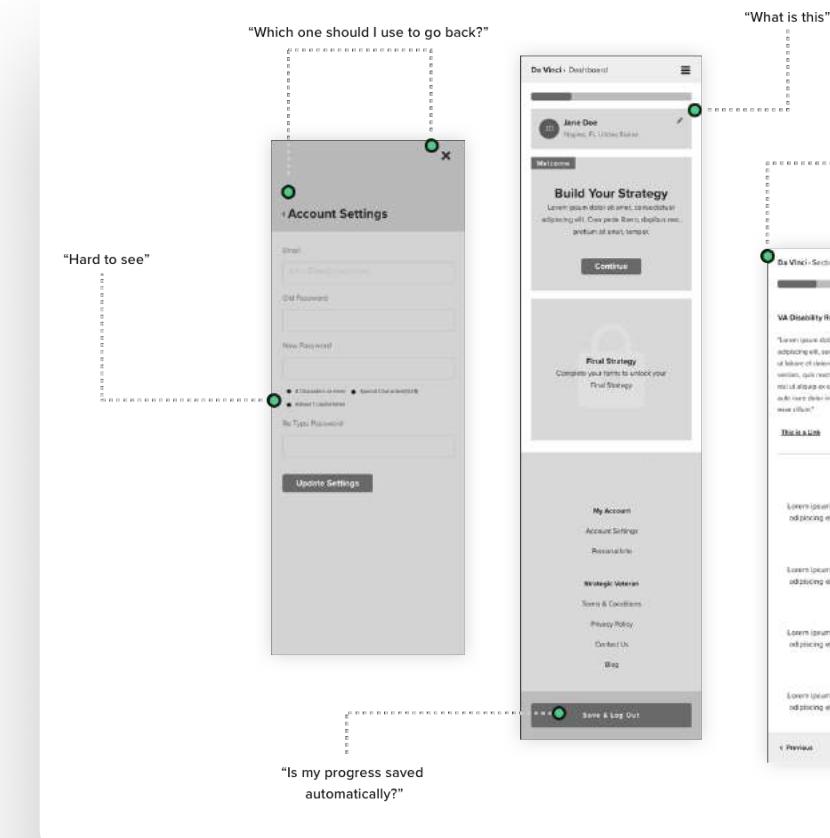
We began creating the general layout of the application based on specific requirements and must-haves that were imperative for the first version of Da Vinci. We collaborated on Invision's Freehand for the first round of concept iterations and made sure to follow Usability and human design centered principles and best practices along the way.

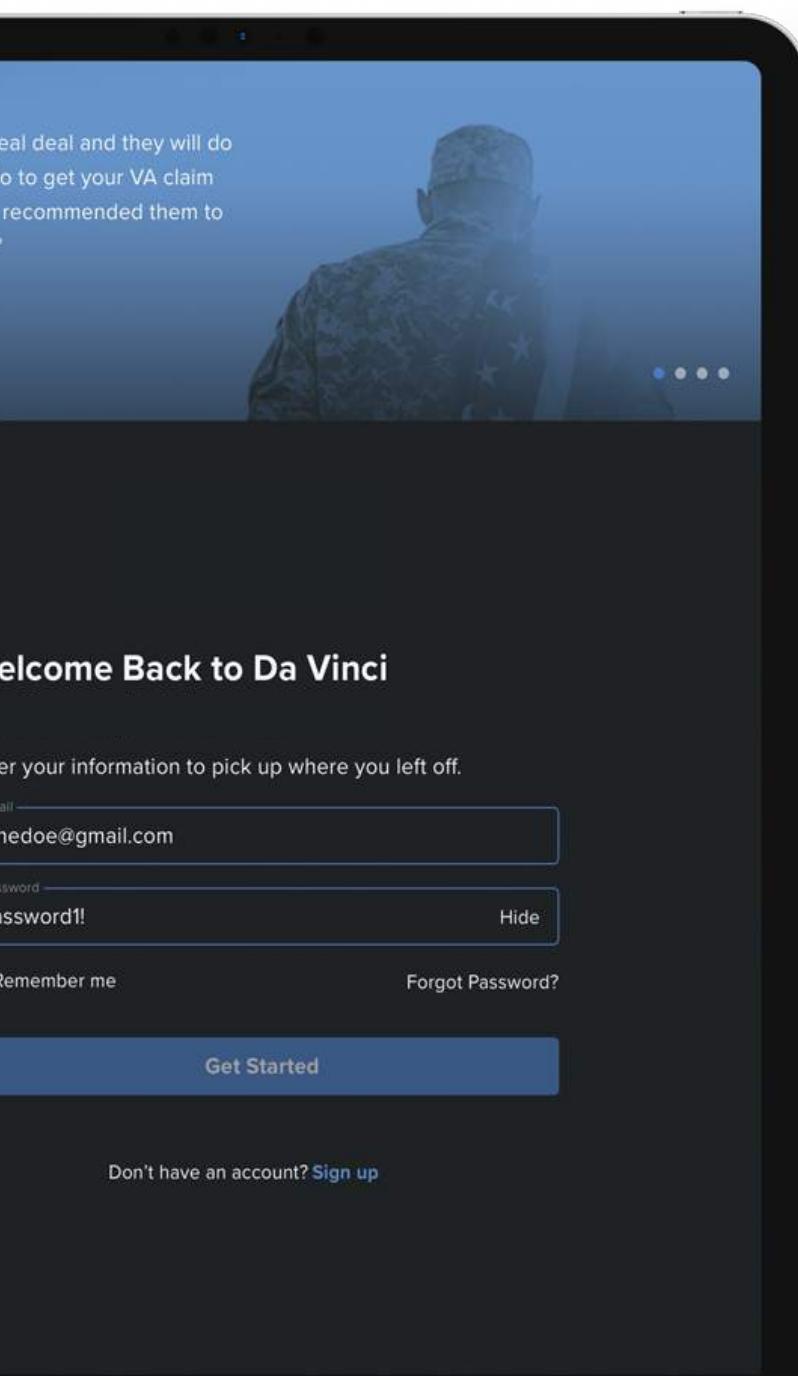


Renovation

After ensuring the low-fidelity wireframes met all user and product requirements and scenarios, we converted our initial concept drawings into medium resolution wireframes in preparation for our first usability test.

In following a strict agile process, we made all necessary changes before committing time and energy into adding styles, icons and visual elements to the application. However, we did create symbols out of the elements that were validated from our testing sessions, so when we were ready to move into styling it was done globally, saving us a great deal of time and energy.





Outcome

After creating a high resolution prototype using Sketch and Flinto, we conducted in depth usability tests to further improve the application. It was imperative for us to gain feedback from stakeholders, customer service and sales representatives, and other key personnel within our organization that have hands on experience helping military veterans every day. This was to ensure that every feature made sense and adhered to and/or improved upon the processes that is followed on a daily basis while helping military veterans get the disability compensation they deserve.

Proprietary Algorithm

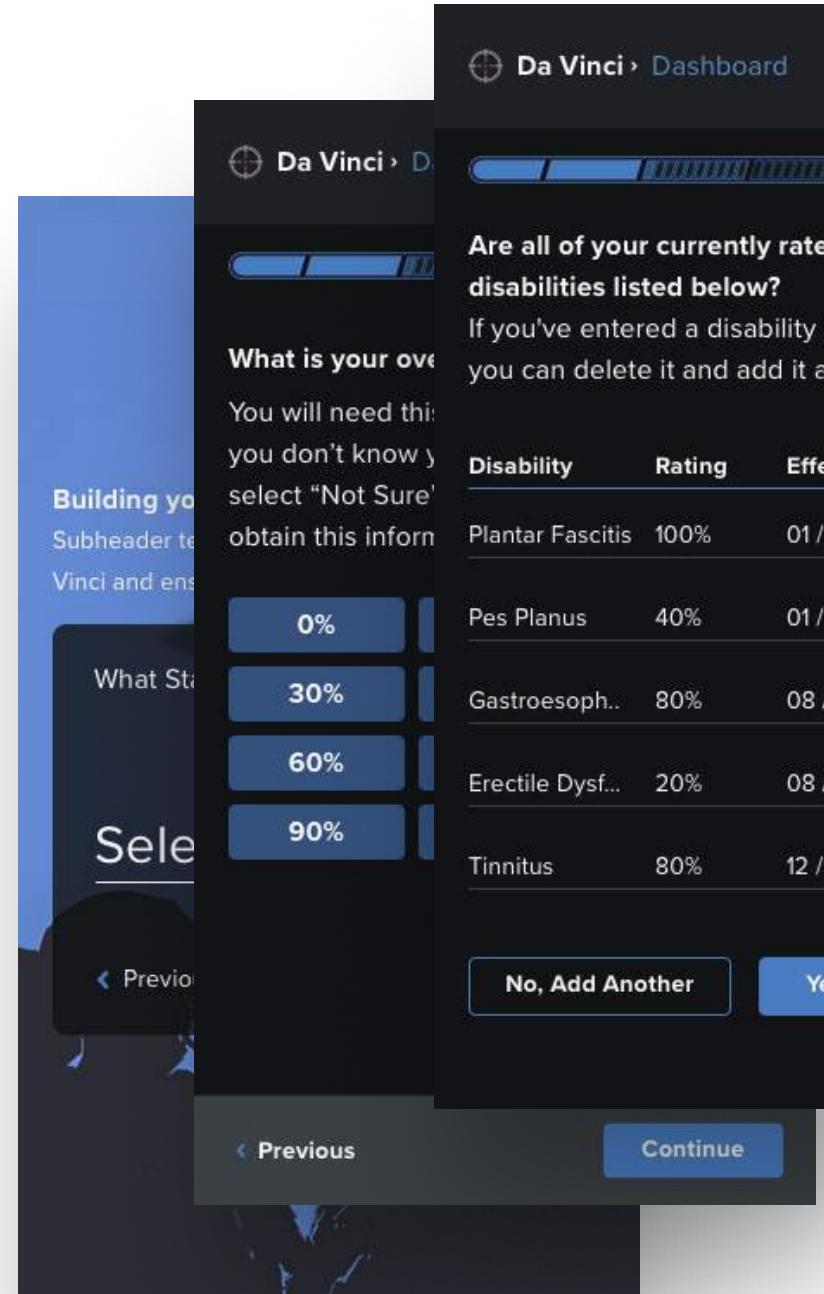
Our proprietary algorithm allows us to help more veterans by simplifying and speeding up the discovery process in a scalable way.

Easy to use Questionnaire

Our intuitive questionnaire simplifies the process for the user by auto generating the top 10 most common answers.

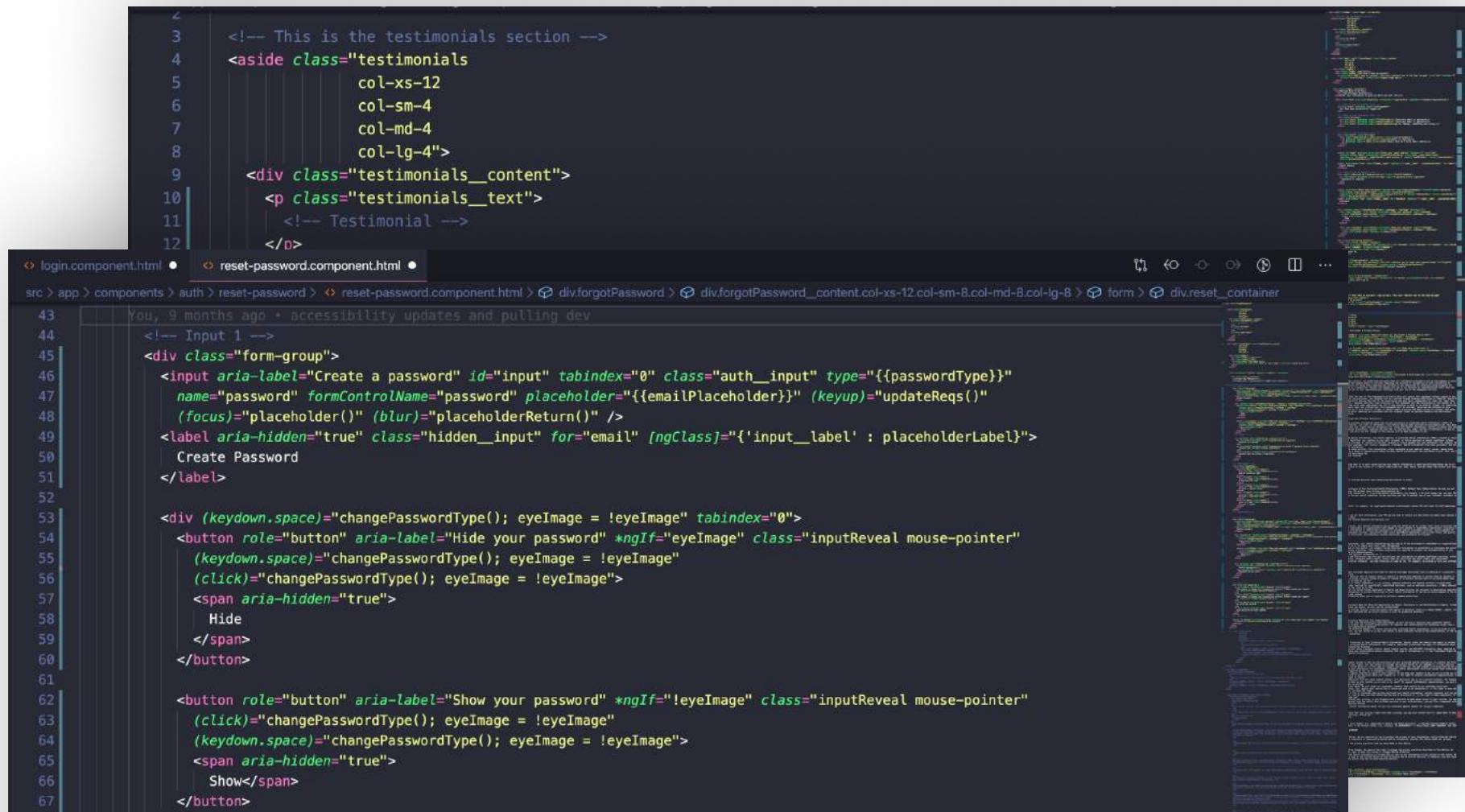
Generate Final Report

After receiving input from users, our application synthesizes all the data and automatically generates a report of potential ratings that can be downloaded as a PDF.



Development

We began development after extensive user testing via clickable prototypes. Utilizing Angular Cli 6 and using separate components we were able to independently work on parts of the application without overriding each others styling and functions. With global styling files in SCSS we were able to use variables and Mixins so that we were not recreating styling and classes that might appear in multiple parts of the application already written by our team. It made development faster and the front end styling more consistent, it was incredibly beneficial because we never had to rewrite visual components.



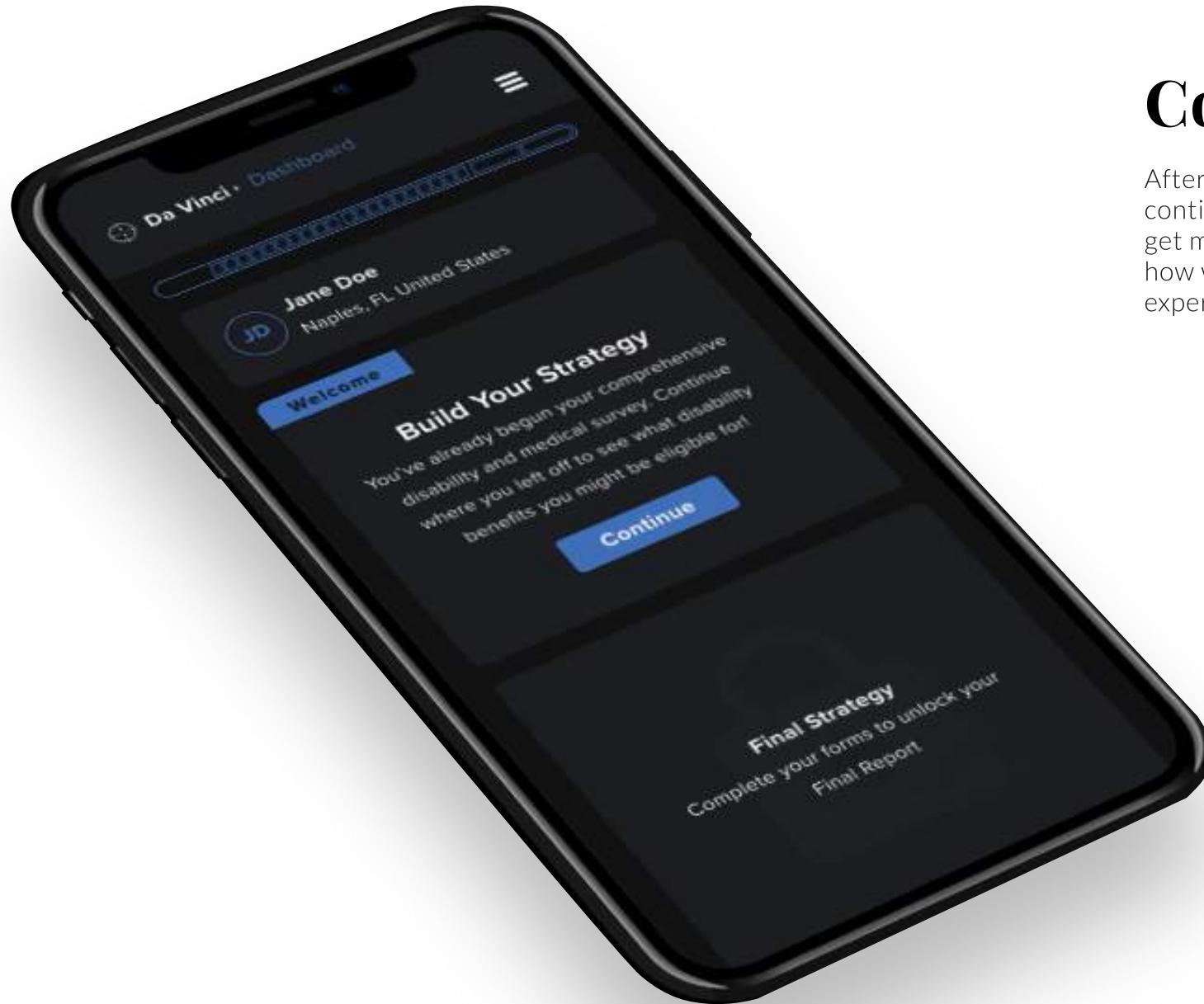
A screenshot of a code editor displaying two files: `login.component.html` and `reset-password.component.html`. The code is written in HTML and includes some CSS-like styling using Angular's class binding syntax (e.g., `[ngClass]`). The editor has a dark theme and shows line numbers on the left. The code is related to user authentication, specifically password creation and visibility.

```
<!-- This is the testimonials section -->
<aside class="testimonials"
      col-xs-12
      col-sm-4
      col-md-4
      col-lg-4>
  <div class="testimonials__content">
    <p class="testimonials__text">
      <!-- Testimonial -->
    </p>
  </div>
</aside>

You, 9 months ago * accessibility updates and pulling dev
<!-- Input 1 -->
<div class="form-group">
  <input aria-label="Create a password" id="input" tabindex="0" class="auth__input" type="{{passwordType}}"
  name="password" formControlName="password" placeholder="{{emailPlaceholder}}"
  (keyup)="updateReqs()"
  (focus)="placeholder()"
  (blur)="placeholderReturn()"/>
  <label aria-hidden="true" class="hidden_input" for="email" [ngClass]="'input_label' : placeholderLabel">
    Create Password
  </label>

  <div (keydown.space)="changePasswordType(); eyeImage = !eyeImage" tabindex="0">
    <button role="button" aria-label="Hide your password" *ngIf="eyeImage" class="inputReveal mouse-pointer"
    (keydown.space)="changePasswordType(); eyeImage = !eyeImage"
    (click)="changePasswordType(); eyeImage = !eyeImage">
      <span aria-hidden="true">
        Hide
      </span>
    </button>

    <button role="button" aria-label="Show your password" *ngIf="!eyeImage" class="inputReveal mouse-pointer"
    (click)="changePasswordType(); eyeImage = !eyeImage"
    (keydown.space)="changePasswordType(); eyeImage = !eyeImage">
      <span aria-hidden="true">
        Show</span>
    </button>
  </div>
</div>
```



Conclusion

After the initial release, we will continue researching and testing to get more feedback and learn more on how we can create the most optimal experience for our users.

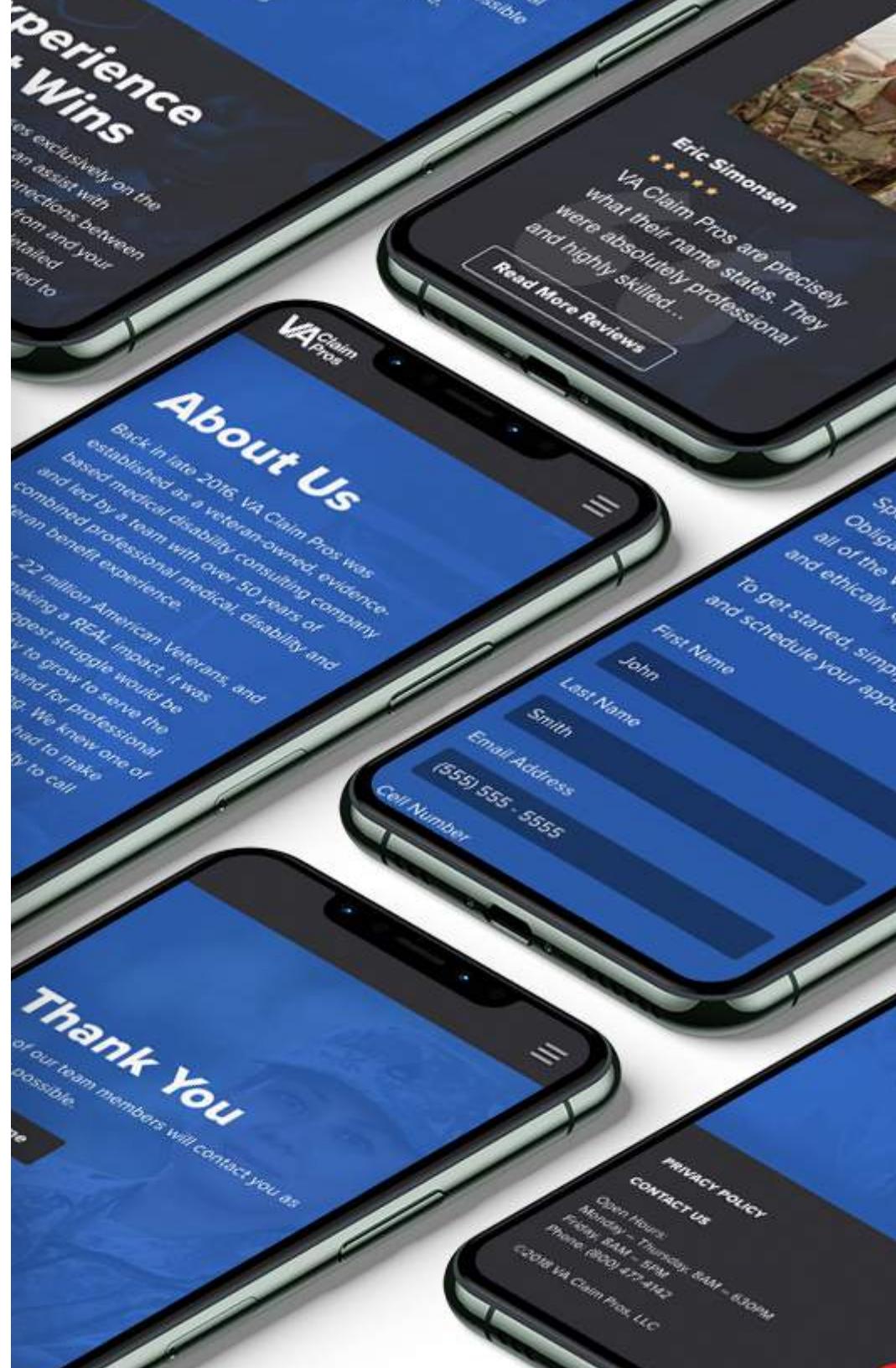
VACP

VA Claim Pros serves United States military veterans by providing them with professional medical disability consultation services. The purpose of this site is to serve as a landing page for potential clients and existing clients to learn more about the company and set appointments to meet with a VACP representative.

The following case study was assembled by myself and my teammate, Hugo Ramos

His work can be seen at hugoramos.co

Project descriptions, proprietary information, and business assets have been redacted or altered to protect company privacy.



Process

○ 1. Research

Industry Research
User Research

○ 2. Iteration

Rapid Prototyping
High fidelity

○ 3. Testing

Accessibility
Usability Testing

○ 4. Validation

User Testing
KPI's

Problem Statement

"Veterans need help obtaining the medical disability benefits they were promised, however they feel misled and ignored. They need a reliable, responsive resource to help them achieve their goals."

Project Duration

From initial scope meeting to final product was three months in total. User acceptance testing continued for two additional weeks.

Team

Hugo Ramos

Product Designer

Andrew Nicholl

Sr. Product Designer

James Rountree

Product Designer

Luke Pate

Lead Software Engineer

Tory Minars

Project Manager

Lane Holcombe

Sr. Software Engineer

User Journey Diagram

User journey diagrams allow us to be more empathetic while designing for our user because we have a better understanding of the specific frustrations they go through before they discover our product and provides us with more awareness of frustrations that may potentially arise in the future.

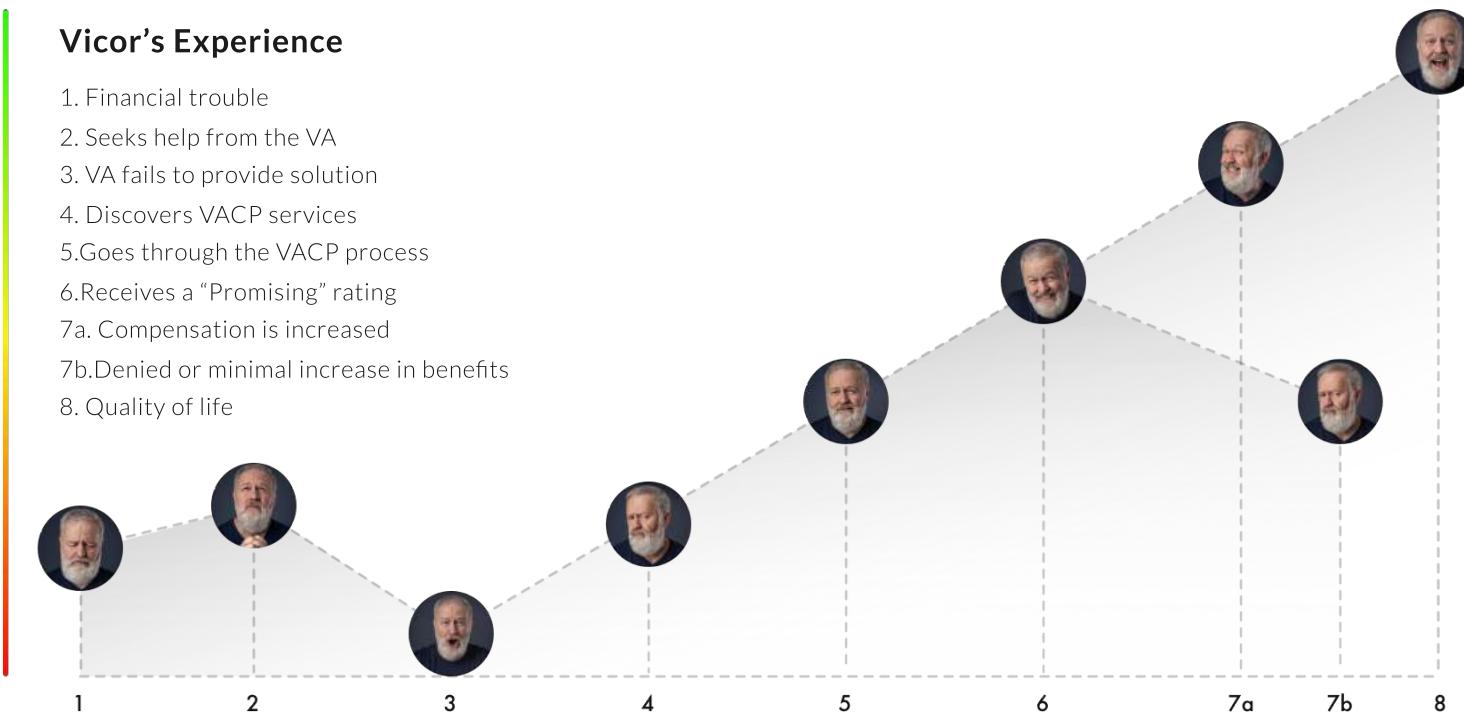


Goals and Expectations

Victor's primary goal is to increase his income so he can take care of himself and his family after service. He expects to receive the help a veteran who has served their country deserves

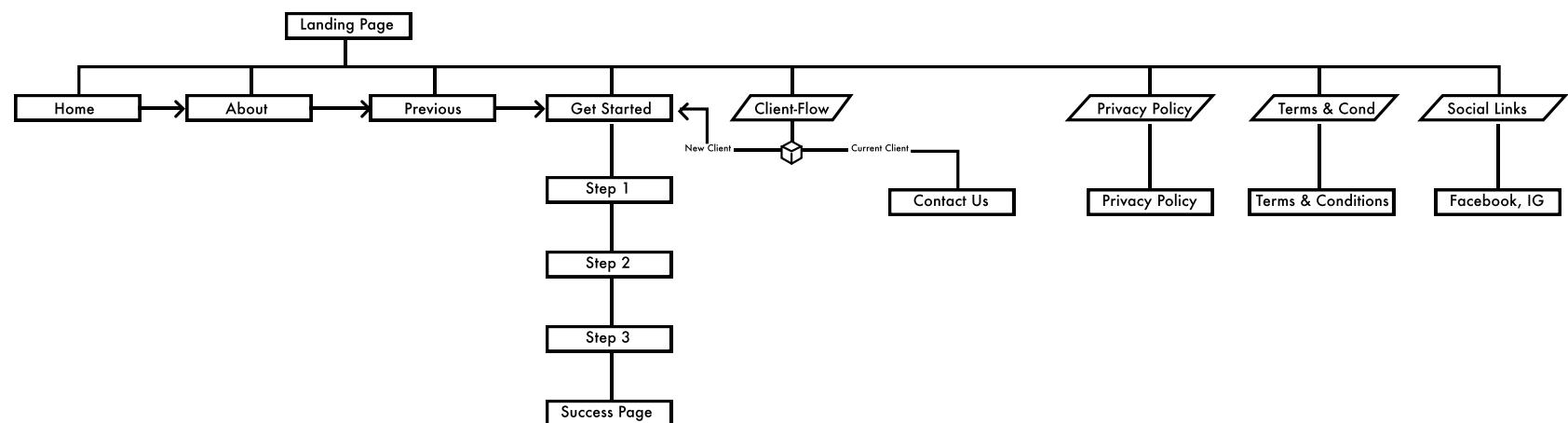
Victor's Experience

1. Financial trouble
2. Seeks help from the VA
3. VA fails to provide solution
4. Discovers VACP services
5. Goes through the VACP process
6. Receives a "Promising" rating
- 7a. Compensation is increased
- 7b. Denied or minimal increase in benefits
8. Quality of life



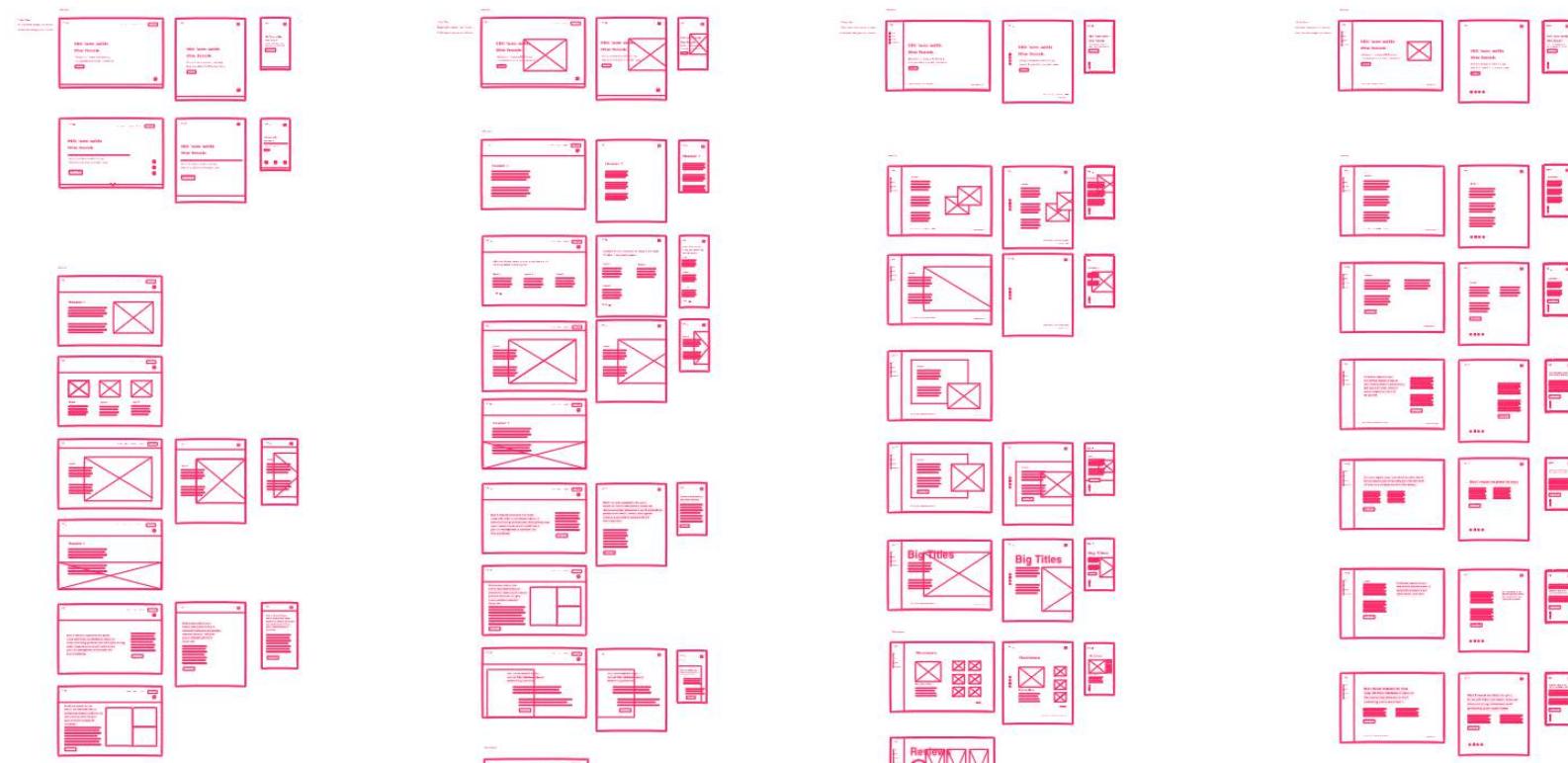
Site Structure

Before our brainstorming session, we decided to map out a simple site structure highlighting the intended user flow. This flow would allow the user to get acquainted with the company and schedule an appointment with a representative.



Quick Concept

After laying out the site structure, we decided to immediately begin iterating several design alternatives. We then regrouped to decide which layout would work best for the desired outcome from both a user and business perspective. Although there were many changes in our approach, this prototyping session was helpful in allowing us to explore all possible ideas in order to determine the most viable option.



Renovation

Before we started this project, we knew there were certain components that needed to be created based on stakeholder requirements. After creating them, we agreed upon which low-fidelity wireframes would work best, then we started converting them into medium resolution. In order to save time and energy we created symbols out of all individual components and linked them to a separate style guide which we could update to effect all components in the future.

The image shows a collage of three screenshots from a website, likely a landing page or a series of pages from a larger site. The top right screenshot features a large headline "Hit 'em with the hook" in bold, black, sans-serif font. Below it is a sub-headline "And bring it on home with the tagline" and a "Learn More" button. The middle screenshot shows an "About Us" section with a bio of a character named "The Fiddler". The bottom right screenshot shows a "Privacy Policy" section with a large heading and several paragraphs of text.

VA Claim Pros

Are you a current VA Claim Pros client?

Yes **No**

Hit 'em with the hook

And bring it on home with the tagline

Learn More

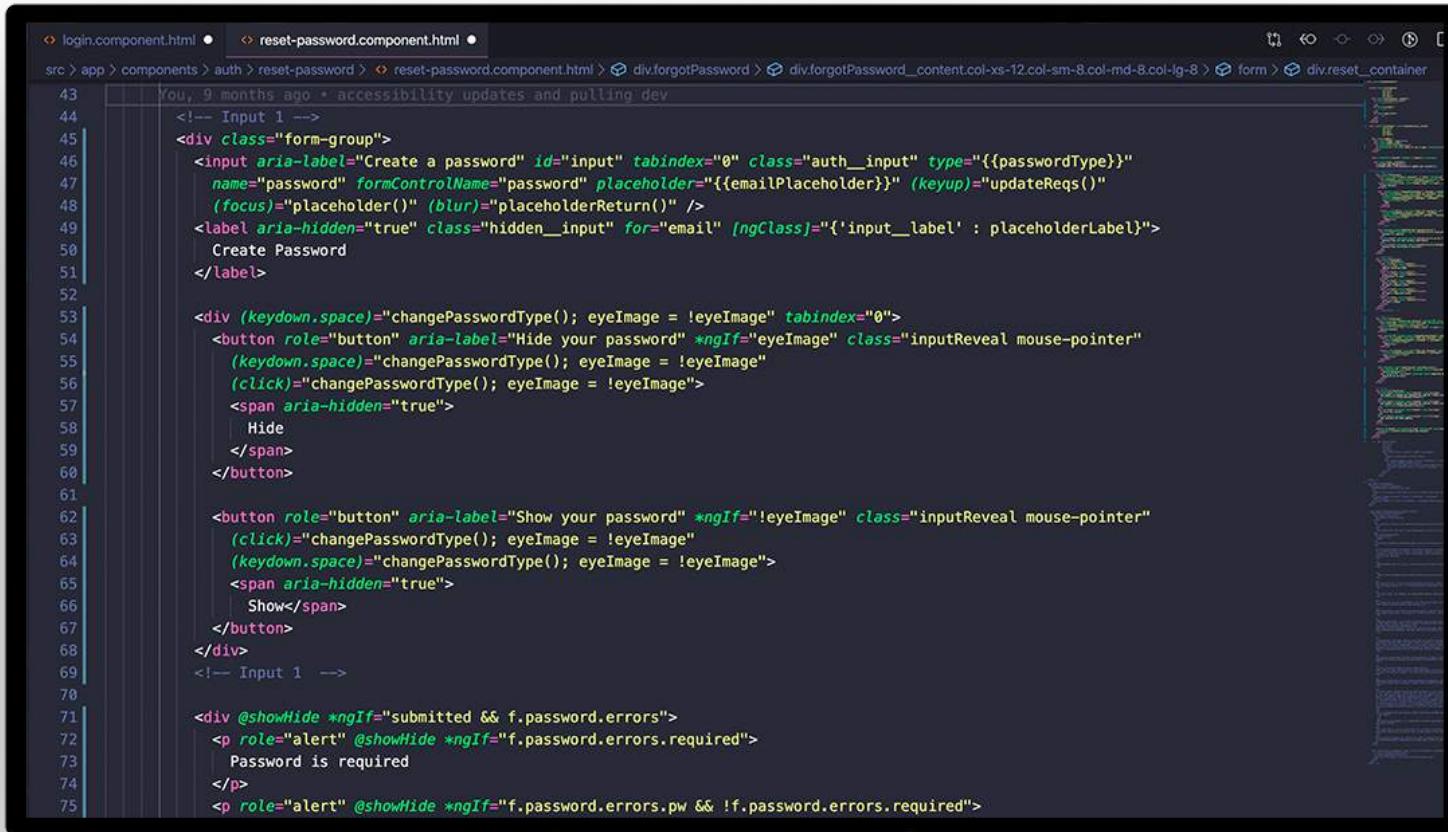
VA Claim Pros

About Us

Privacy Policy

Development

Before our brainstorming session, we decided to map out a simple site structure highlighting the intended user flow. This flow would allow the user to get acquainted with the company and schedule an appointment with a representative.



The screenshot shows a code editor with the file `reset-password.component.html` open. The code is written in Angular template syntax. It includes a form group for a password input, a button to toggle password visibility, and validation messages for required fields. The code uses `aria-label`, `tabindex`, and `placeholder` attributes, along with `keyup`, `click`, and `ngIf` directives.

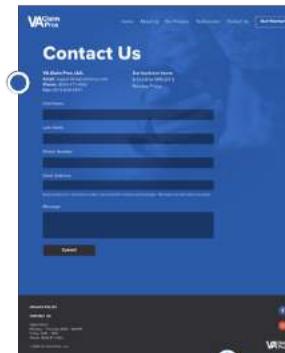
```
43 You, 9 months ago * accessibility updates and pulling dev
44 <!-- Input 1 -->
45 <div class="form-group">
46   <input aria-label="Create a password" id="input" tabindex="0" class="auth_input" type="{{passwordType}}"
47   name="password" formControlName="password" placeholder="{{emailPlaceholder}}" (keyup)="updateReqs()"
48   (focus)="placeholder()" (blur)="placeholderReturn()"/>
49   <label aria-hidden="true" class="hidden_input" for="email" [ngClass]={'input_label' : placeholderLabel}>
50     Create Password
51   </label>
52
53   <div (keydown.space)="changePasswordType(); eyeImage = !eyeImage" tabindex="0">
54     <button role="button" aria-label="Hide your password" *ngIf="eyeImage" class="inputReveal mouse-pointer"
55     (keydown.space)="changePasswordType(); eyeImage = !eyeImage"
56     (click)="changePasswordType(); eyeImage = !eyeImage">
57       <span aria-hidden="true">
58         Hide
59       </span>
60     </button>
61
62     <button role="button" aria-label="Show your password" *ngIf="!eyeImage" class="inputReveal mouse-pointer"
63     (click)="changePasswordType(); eyeImage = !eyeImage"
64     (keydown.space)="changePasswordType(); eyeImage = !eyeImage">
65       <span aria-hidden="true">
66         Show</span>
67     </button>
68   </div>
69   <!-- Input 1 -->
70
71   <div @showHide *ngIf="submitted && f.password.errors">
72     <p role="alert" @showHide *ngIf="f.password.errors.required">
73       Password is required
74     </p>
75     <p role="alert" @showHide *ngIf="f.password.errors.pw && !f.password.errors.required">
```

User Testing

We decided to conduct user testing sessions at various points in our process between medium and high res to validate our design decisions. In each session, we realized that there were many opportunities to improve the experience for the user. After organizing all feedback into a Trello board, we prioritized all of the changes to be made and implemented them into the final version of our high res design while adhering to stakeholder requirements.



"Add a phone number to the contact page to allow users to call the office directly."



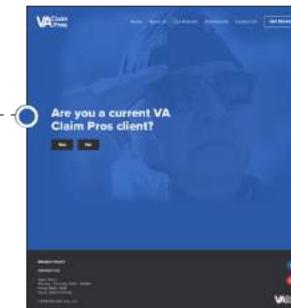
"I expect to receive a turn around time for when someone should reach back to me."



"There should be a link to the privacy policy page so users can be more informed."



"It's frustrating that the contact page takes me here first."



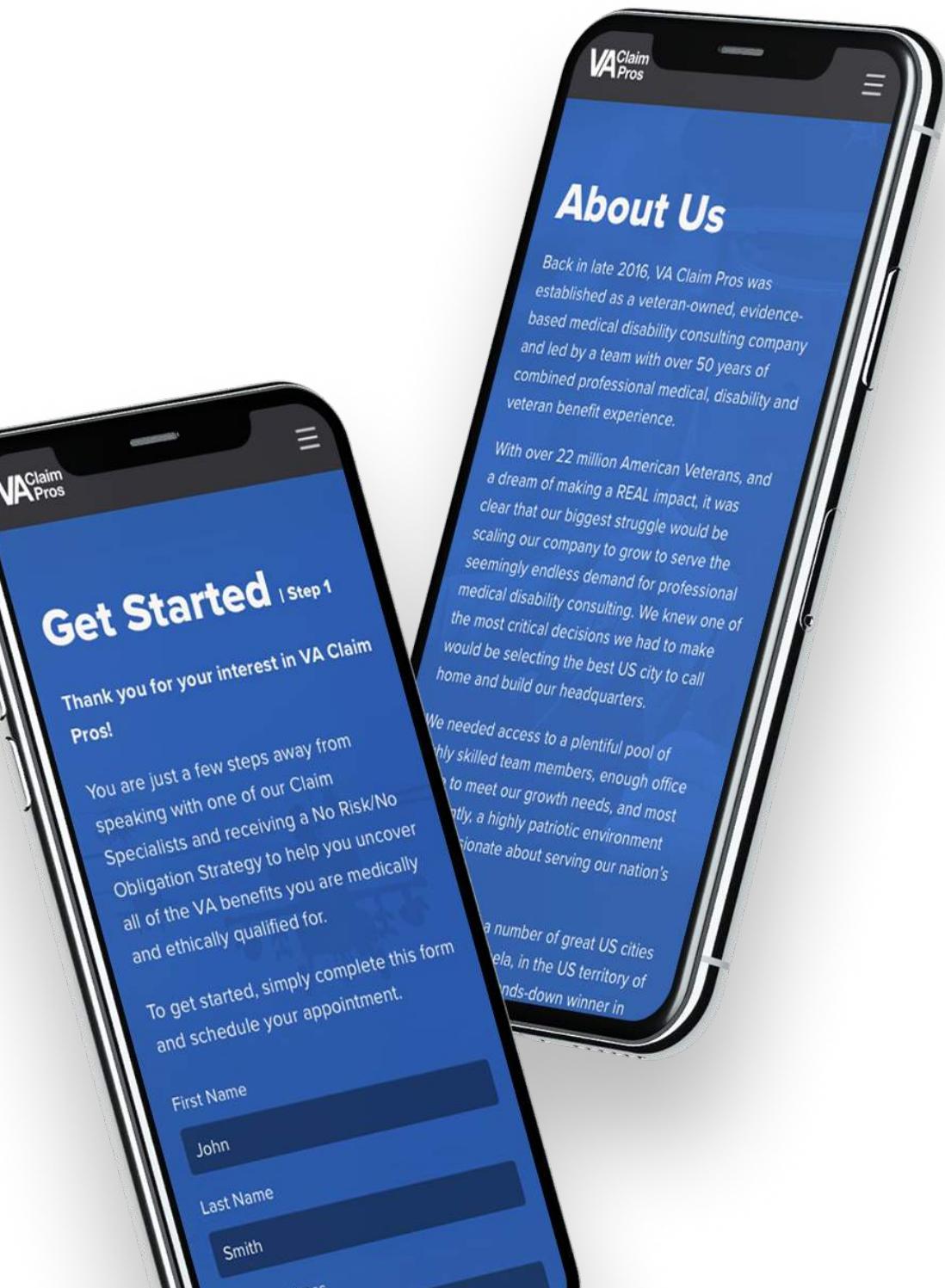
Home Page

US Military Veterans appreciate doing business with established companies they can trust and believe in. The about page was designed to build rapport and credibility with the user base.

Our Process

In hopes to increase disability benefits for US Military Veterans on a wide scale, the site is designed to take users through a flow that leads them into scheduling an appointment with one of our representatives. Current clients have the ability to reach out directly to client support.

The screenshot shows the VA Claim Pros website. At the top right, there's a navigation bar with links for Home, About Us, and Contact Us. The main header features the company logo and the tagline "Win the VA disability benefits you've been promised." Below the header is a large blue banner with the text "VA Claim Pros". On the left side of the banner, there's a "Learn More" button. The main content area has a dark background with white text. It starts with a section titled "Our Process" which includes a brief description of their services and a "Get Started Today" button. To the right of this is a testimonial from a user named Eric S., featuring a photo of two veterans and a 5-star rating. Further down, there's a "About Us" section with a brief description of the company's history and expertise, followed by a "Read Our Story" button.



Get Started

In hopes to increase disability benefits for US Military Veterans on a wide scale, the site is designed to take users through a flow that leads them into scheduling an appointment with one of our representatives. Current clients have the ability to reach out directly to client support.

About Us

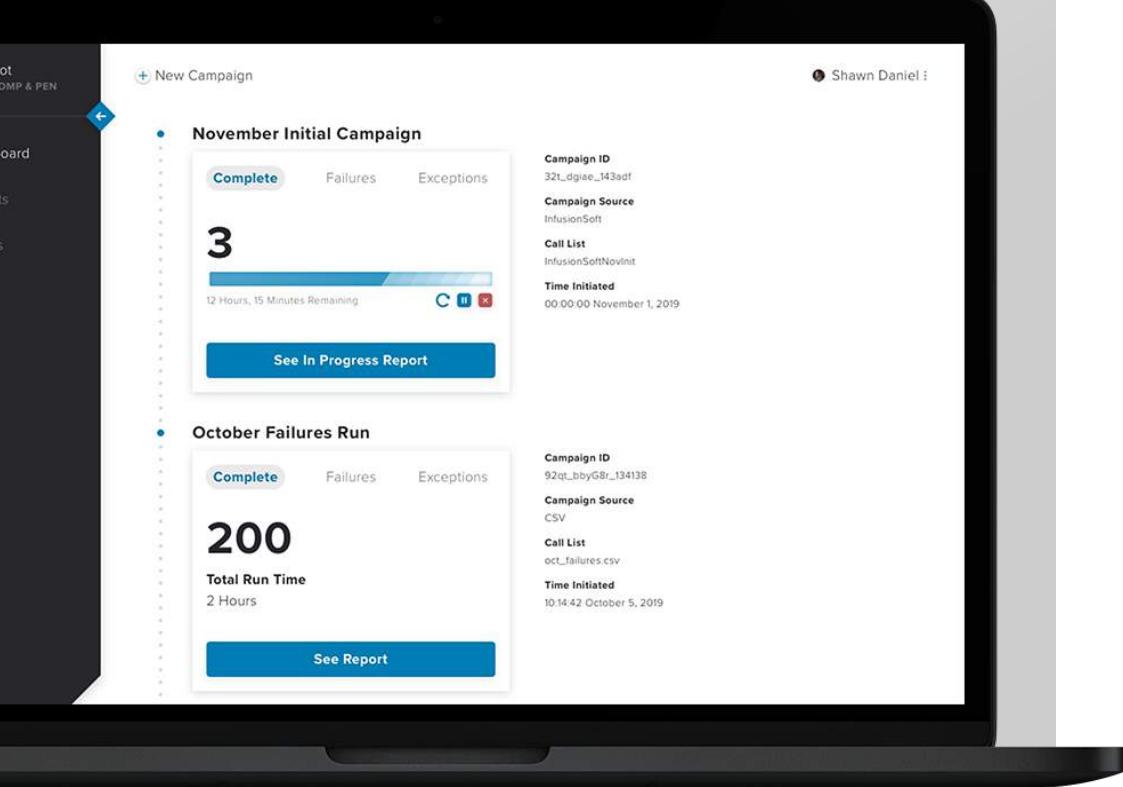
US Military Veterans appreciate doing business with established companies they can trust and believe in. The about page was designed to build rapport and credibility with the user base.

Style Guide

This style guide was created in a modular fashion based on components from our sketch libraries. Using abstract we were able to keep all UI components up to date across multiple variations of the design and reuse these components to populate each individual element in our style guide. This allowed us to keep the style guide up to date automatically based on edits to the library master files. Abstract also allows for peer review and critique with pull requests.

The screenshot displays two pages of a style guide in the Abstract app:

- Top Page (Elements):** Shows a sidebar with "VACP Style Guide", "Components", "Elements", "Header", "Colors", and "Assets". The main content area is titled "ELEMENTS" with a sub-section "Primary UI elements documented below include buttons and inputs." It shows four button states: Primary Button (Normal), Secondary Button (Normal), Primary Button (Hover), and Secondary Button (Hover). The properties panel on the right shows an Artboard width of 1448px, height of 11949px, and fill color #18181a.
- Bottom Page (Colors):** Shows a sidebar with "VACP Style Guide", "Components", "Elements", "Header", "Colors", "Typography", and "Assets". The main content area is titled "COLORS" with a note about color naming conventions. It lists four base colors: \$color-st-tropaz (#2255AA), \$color-tuna (#223333), \$color-white (#FFFFFF), and \$color-gray-chateau (#BDBDBD). The properties panel on the right shows an Artboard width of 1448px, height of 11949px, and fill color #18181a. It also includes sections for "COLORS" (with a color palette) and "TYPOGRAPHY" (listing ProximaNova Regular, 16px, 24px, 12px, 18px, 8px, 21px, 24px, 48px, 32px, 16px, 21px, 32px, 72px, 36px, 56px) and "LINE HEI..." (listing 24px, 16px, 28px, 16px, 32px, 72px, 56px).



CallBot

Callbot is an application built to run or schedule processes for an internal team. The goal of developing a user interface for Callbot was to allow employees to view campaign results, manage client statuses, and view historical records in an attempt to democratize the process of running the application and bring in non-developers to use the tool.

The following case study was assembled by my teammate, Andrew Nicholl

His work can be seen at nicholldesign.com

Process

○ 1. Research

Industry Research
User Research

○ 2. Iteration

Rapid Prototyping
High fidelity

○ 3. Testing

Accessibility
Usability Testing

○ 4. Validation

User Testing
KPI's

Problem Statement

"As a user I need to be able to "

Project Duration

From initial scope to completion of Phase 1 was set for 3 sprints (6 Weeks)

Team

Hugo Ramos

Product Designer

Andrew Nicholl

Sr. Product Designer

James Rountree

Product Designer

Luke Pate

Lead Software Engineer

Tory Minars

Project Manager

Lane Holcombe

Sr. Software Engineer

Research Phase

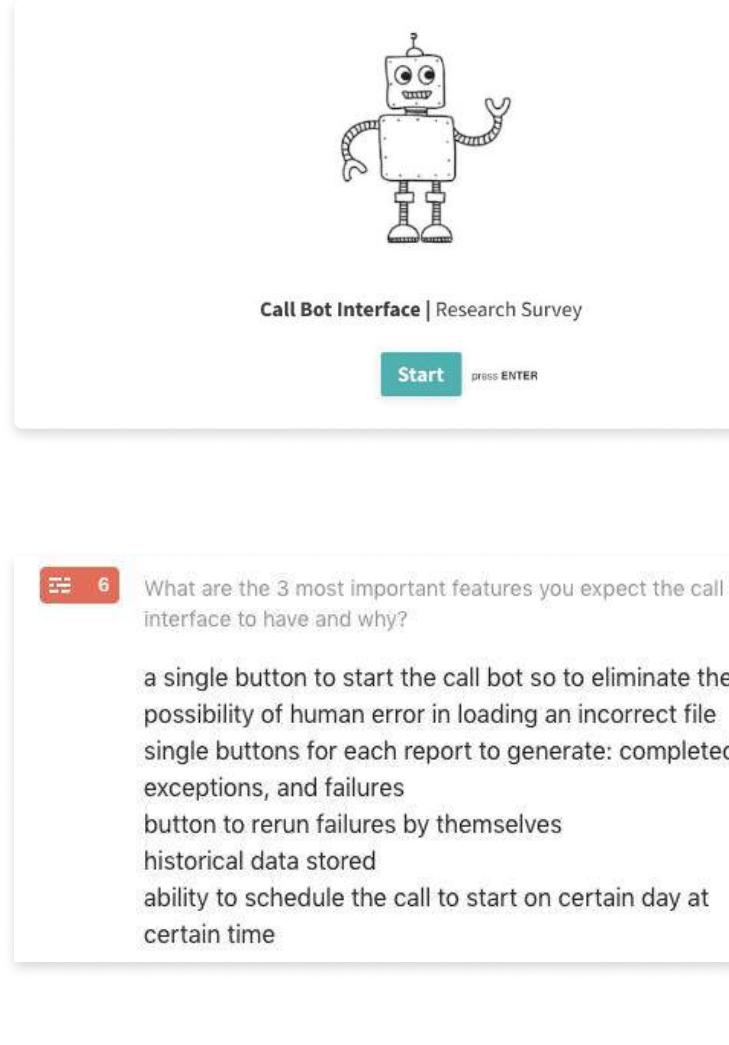
In order to figure out which types of research we needed to conduct, we first needed to understand what kind of information we were hoping to learn. We used a research brainstorming template to get all of our questions down on paper.

User Survey

We decided to conduct a survey on the handful of people within our organization that would use Callbot to understand their experience with the current system and their hopes for the future.

Key Takeaways

A single button to initiate the process within the application, ability to easily toggle views between different reports, ability to view all historical data, ability to easily rerun process if failed, ability to schedule process to happen at a future time and date.



The screenshot shows a user interface for a 'Call Bot Interface | Research Survey'. At the top right is a small, friendly-looking robot icon. Below it, a green button labeled 'Start' with the instruction 'press ENTER' is visible. A question card is displayed with the number '6' in a red box and the text: 'What are the 3 most important features you expect the call interface to have and why?'. To the right of the question, several responses are listed in a vertical stack:

- a single button to start the call bot so to eliminate the possibility of human error in loading an incorrect file
- single buttons for each report to generate: completed exceptions, and failures
- button to rerun failures by themselves
- historical data stored
- ability to schedule the call to start on certain day at certain time

User Personas

Using the results of our survey, we crafted two main personas who were going to be using the Callbot.



Matthew
Primary User

Personal Info

Easy going, passionate leader, meticulous.
 Gender: Male
 Age: Mid 40's
 Marital Status: Married
 Technological Aptitude: Average

User Needs

Efficiency in running business processes
 Accurate historical records

Technology Devices

Android (Personal)
 Windows Desktop (Work)
 Windows Laptop (Home)

Pain Points

Importing and exporting excel spreadsheets takes too much time and is prone to formatting errors
 Not familiar with the command line and cannot run the application in it's current state without assistance from dev ops
 Readyng data in plain text format and looking through large json objects is tedious and details are often missed



Busy Bethany
Secondary User

Personal Info

Focused, gets the task done, juggles a million things at once.
 Gender: Female
 Age: Late 20's
 Marital Status: Single
 Technological Aptitude: Low to Average

User Needs

Clearly see different results between different clients
 Have similar results be organized together
 Ability to export lists of results

Technology Devices

iPhone (Personal)
 Windows Desktop (Work)
 Windows Laptop (Home)

Pain Points

Having to update spreadsheet names with different versions is hard to keep track of
 There's no way to see how far into the process I am until I'm nearly done

Site Architecture and Task Flows

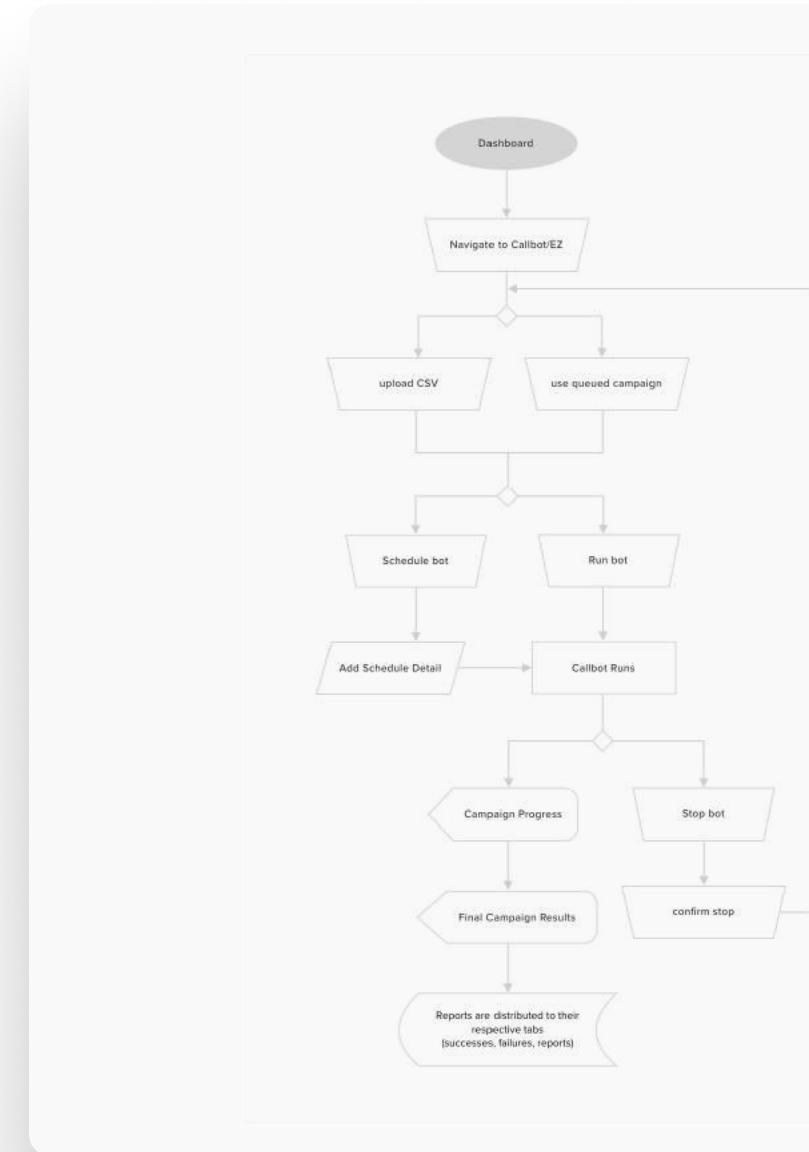
Based on the needs of our users we determined the basic site architecture of Callbot and began making task flow for each section of Callbot's interface. After we had the core structure and flow of our application in place, we moved on to the next stage in the design lifecycle.

Agile Design Sprints

In order to put out a testable component at the end of each design sprint, we broke the project up into it's major components and focused one of the major components per sprint.

Design Lifecycle

This meant going through an entire design sprint cycle of wireframing, usability testing, creating hi fidelity design mockups, and developing rapid prototypes in Angular to use for a final usability test before continuing to the next component.





Wireframes

At the beginning of each sprint, we decided which major component we were tackling and dove into sketching and creating low-fidelity wireframes in InVision Freehand. We then replicated our task flows using our wireframes to ensure that we were not missing any necessary steps or features.

Paper Testing

We then printed out these wireframes, cut them up, and conducted usabilities tests using these paper prototypes. Our notes from these paper testing sessions were used to rapidly update wireframes and conduct more tests. After three to four rounds of tests for each component, we were ready to move into designing mockups in Sketch.



- Run the call bot task - "If you wanted to start a new one, would go to campaigns or run bot?" - assumed the side navigation would just run the bot, expected a task instead of navigation ("home" instead of run bot)
 - 💡 Made it "Dashboard"

- Start a new campaign task - "from here I would click run bot to clear this out" (click on new campaign)
 - 💡 Fixed by above task

- Run new campaign task - "Expected the navigation was steps, instead of actual navigation" (Copy)
 - 💡 Changed their names, bolded active tab

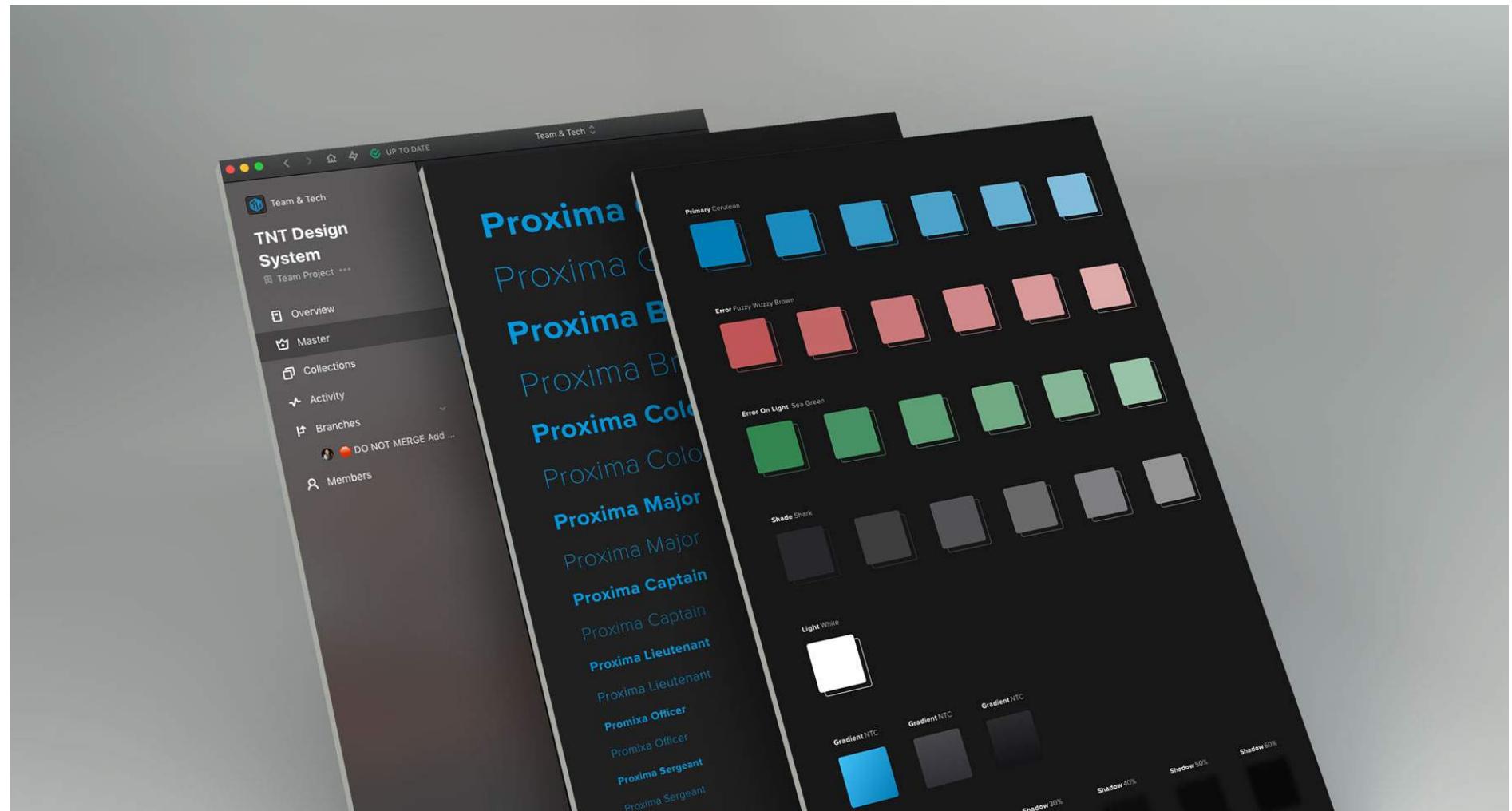
- Campaign type what would you expect to happen - "Infusionsoft campaign means I have to upload the CSV" - "Can we automate the pulling of the CSV at a certain time of the day?" - "We need to pull that campaign the night before around midnight, and I would expect to schedule it and it would automatically pull the latest CSV"
 - 💡 Changed "Campaign Type" to "Campaign Source"

- Changing to upload csv TASK - "you would check the dropdown and highlight CSV", "Choose to me is vague, it seems like I already loaded the CSV", It should be "Browse" that makes sense to me, like I have to browse to find the right one"
 - 💡 Changed to "Browse"

- Change other form queue TASK - "Just click the drop down", "I want to have to pick the one I want so that way someone has to make sure they choose the right option"

Design System Implementation

Using our design system for internal products we were able to come into this project with a predetermined set of minor components and styles. These included colors, icons and typography, which was extremely helpful in speeding up the design process, staying organized, and maintaining consistency throughout the application.



Design Mockups

After ensuring the low-fidelity wireframes for each major component met all user and product requirements and scenarios, we converted our initial concept drawings into medium resolution mockups. We then styled them and added all reusable components using the design system.

The image displays six medium-resolution mockups of a campaign management interface, arranged horizontally. Each mockup shows a list of contacts with columns for Last Name, First Name, Email, and Status (e.g., READY, IN REVIEW, FAILED). The interface includes navigation buttons (Complete, Details, Sort, Filter), search fields, and action buttons (Mark For Review, Export).

- Complete:** Shows a list of contacts for the "November Initial Campaign". All contacts are marked as "READY".
- Complete Details:** Shows the details for the "November Initial Campaign". It includes a "Complete" button and a "Details" dropdown.
- Complete Selected:** Shows the selected contacts for the "November Initial Campaign". It includes a "Select All" button, a "Sort" button, a "Filter" button, and a "Details" dropdown.
- In Review:** Shows the contacts for the "November Initial Campaign" marked as "IN REVIEW". It includes a "Select All" button, a "Sort" button, a "Filter" button, and a "Details" dropdown.
- Failed:** Shows the contacts for the "November Initial Campaign" marked as "FAILED". It includes a "Select All" button, a "Sort" button, a "Filter" button, and a "Details" dropdown.
- Failed:** Shows the details for the "November Initial Campaign". It includes a "Complete" button, a "Review" button, a "Failed" button, and a "Exceptions" button.

Callbot

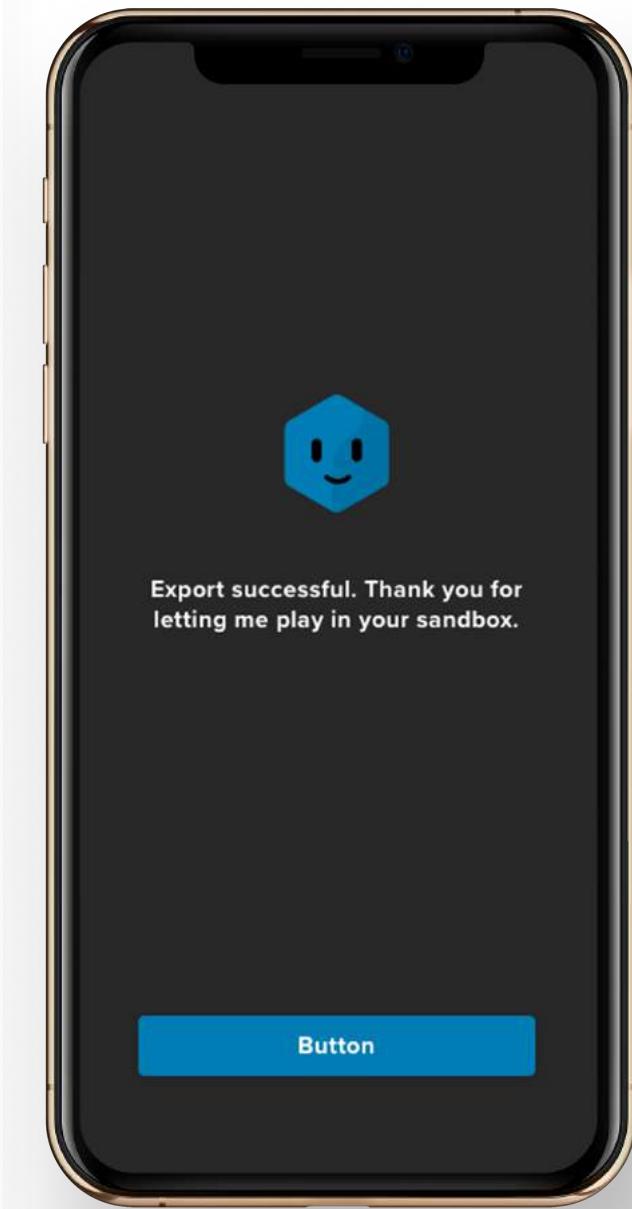
Dashboard Reports Clients

◀ November Initial Campaign Andrew Nicholl :

Campaign ID 13r89hq34tq894T4A3 | Campaign Type Automatic | Form Queue November | Time Initiated 12:00 AM, 11/01/2019

Complete Review Failed Exceptions Hide In Review Hide Exported

Last Name	First Name	Email	Amount	Status	Action
Santiago	George	enid_howell@imelda.tv	\$540	READY	▼
Fastlane	Robbie	passontheright@me.com	\$0	READY	▼
Keith	Tobias	bootinyour@gmail.com	\$800	READY	▼
Crockpot	Johnny	raccoonstew1971@yahoo.com	\$940	READY	▼
Keeper	Finder	fr33allSaintsJacket@gmail.com	\$0	READY	▼
DiPabliano	Joey	meatball1@yahoo.com	\$400	READY	▼
Snailfish	Craifish	bighooklittlebit3@gmail.com	\$789	READY	▼
Peterson	Emanual	pennstaterules1999@yahoo.co...	\$0	READY	▼



Maintainable

By sticking to our design system and keeping up our documentation as one would in a traditional design to developer handoff, we were also able to ensure that it would be easy to add additional components and functionality in the future, while sticking to the design and style guides we have in place.

User Focused

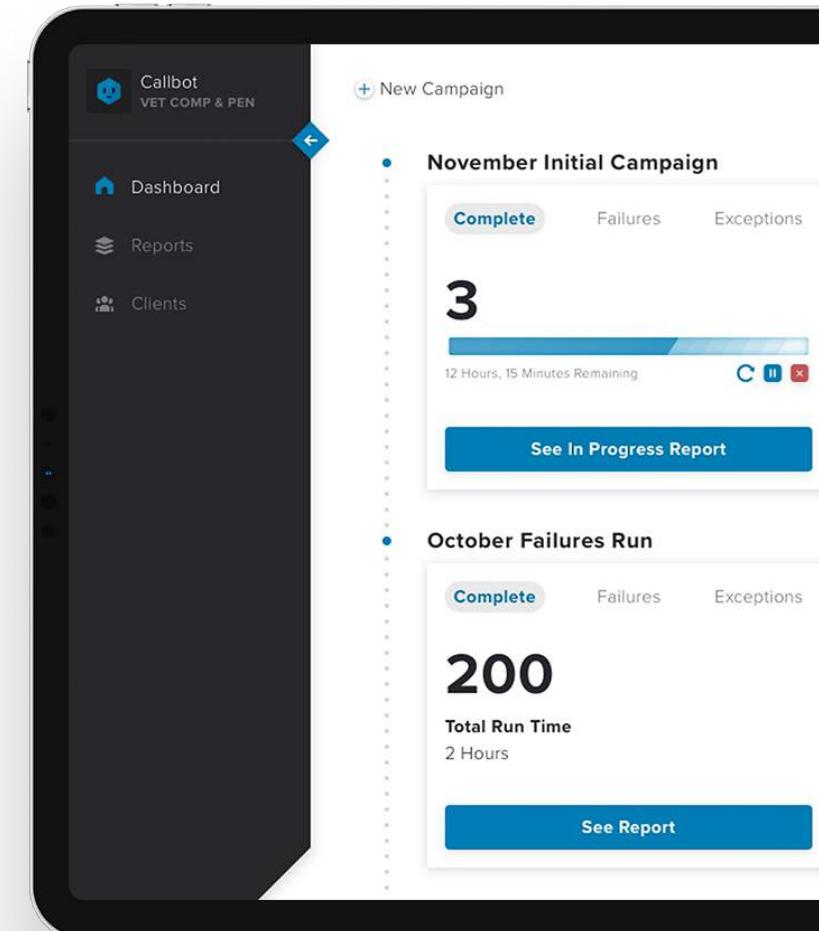
Not only did it keep work interesting, it made for an overall better product, as we were able to conduct usability test between sprints and quickly implement changes to the previous sprint's component and embrace new ideas for the next one.

Accessibility

Having the product development team spearhead front-end development also helped ensure accessibility in our product from the outset, by writing semantic html and ensuring WCAG and ADA compliance.

Agile

By embracing agile methodologies and building out each major component per design sprint, we were able to produce results fast and keep the development process moving in tandem with our design process.

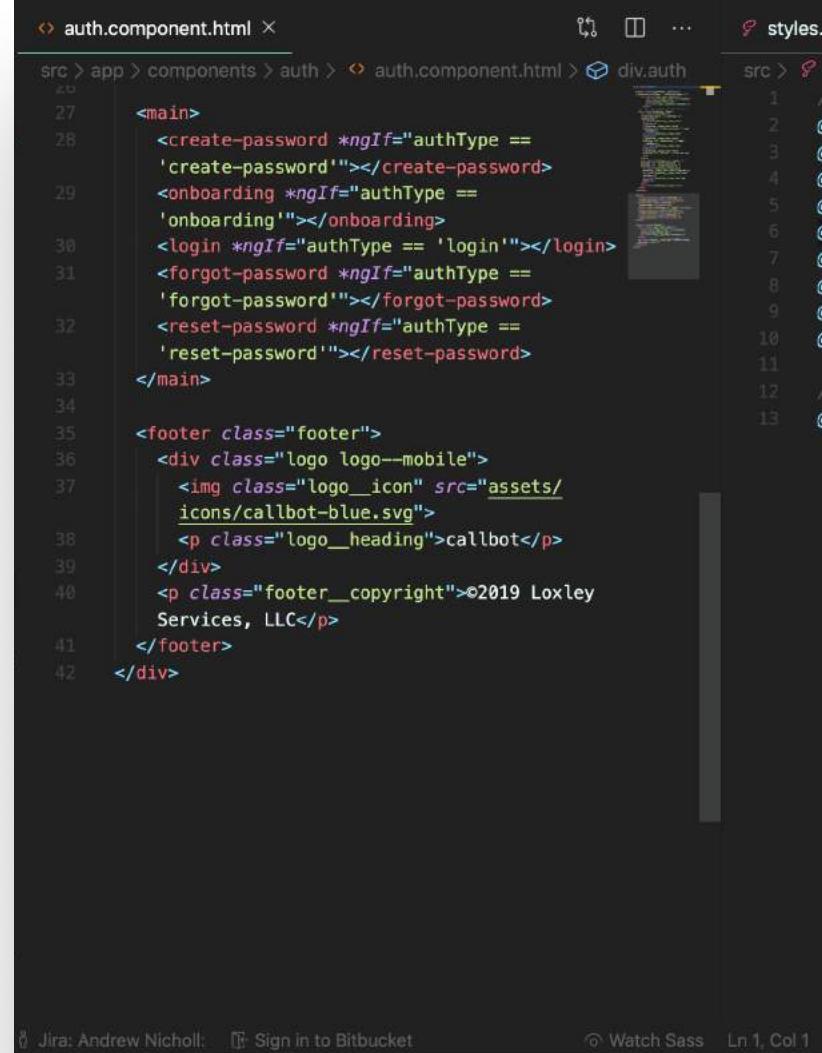


Development

Our team utilized Angular as our development framework, so we began development by laying the frontend foundation for our Angular project using HTML, Sass (SCSS), and Typescript.

Usability Tests & Demo

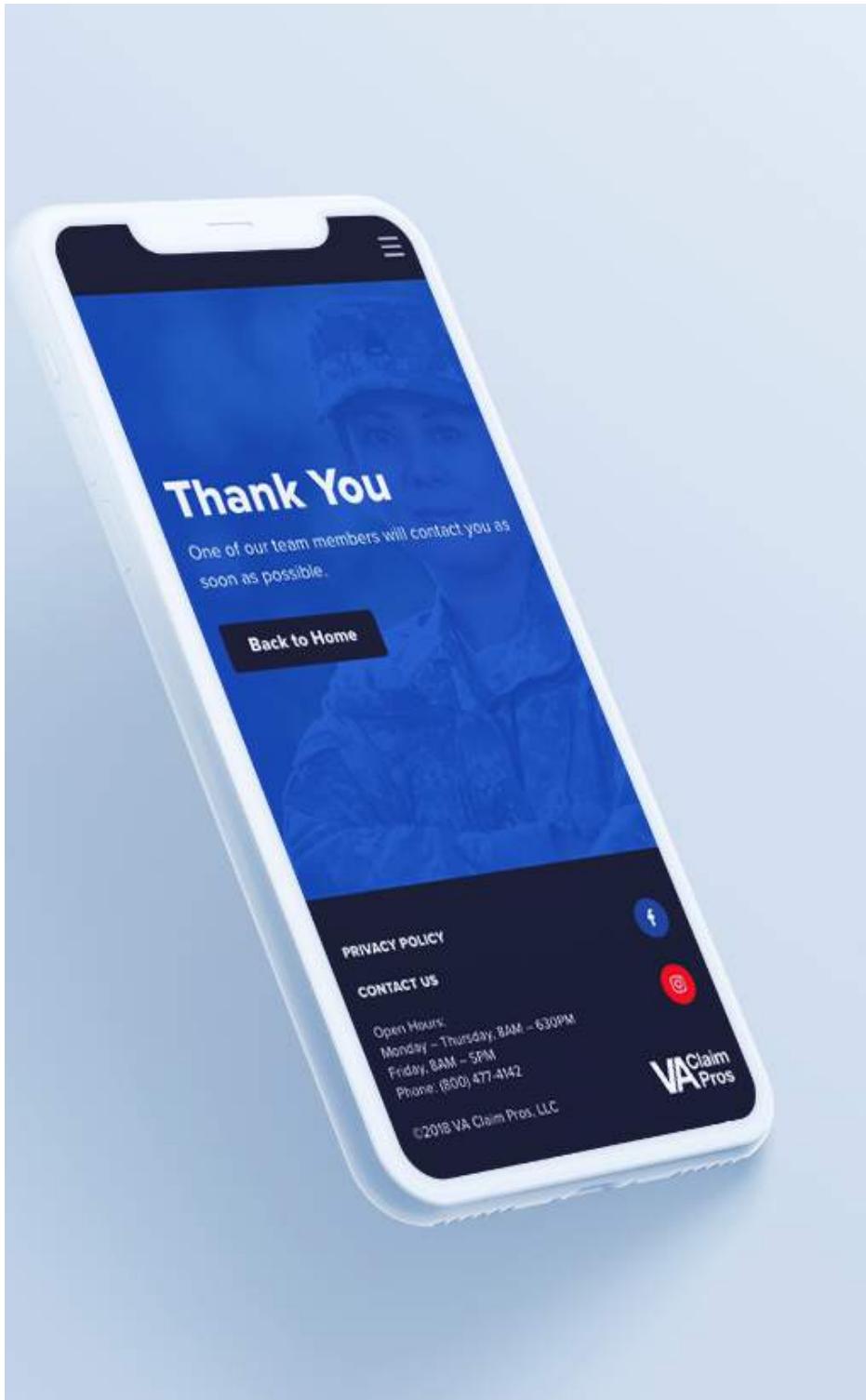
At the end of each sprint, we conducted a final usability test of the component and made any necessary updates before holding out demo for product owners, stakeholders, and the development teams.



A screenshot of a code editor showing the file `auth.component.html`. The code is an Angular template with several conditional directives (`*ngIf`) based on the value of `authType`. It includes components for `create-password`, `onboarding`, `login`, `forgot-password`, and `reset-password`. Below this, there's a `main` section and a `footer` section containing a logo and copyright information for Loxley Services, LLC. The code editor has a dark theme with syntax highlighting for HTML, CSS, and JavaScript.

```
<main>
  <create-password *ngIf="authType == 'create-password'"></create-password>
  <onboarding *ngIf="authType == 'onboarding'"></onboarding>
  <login *ngIf="authType == 'login'"></login>
  <forgot-password *ngIf="authType == 'forgot-password'"></forgot-password>
  <reset-password *ngIf="authType == 'reset-password'"></reset-password>
</main>
<div class="footer">
  <div class="logo logo--mobile">
    
    <p class="logo__heading">callbot</p>
  </div>
  <p class="footer__copyright">©2019 Loxley Services, LLC</p>
</div>
</div>
```





Lets Connect!

Thank you for taking the time to view some of the highlights of my work.

JimboRountree.com
LinkedIn@JohnSmith
James.D.Rountree@Gmail.com
352 328 6777