Andrew Nicholl

Portfolio of Recent Work

Callbot

An Interface With Personality

2

TNT Design System

Design system for a multi-brand ecosystem

18

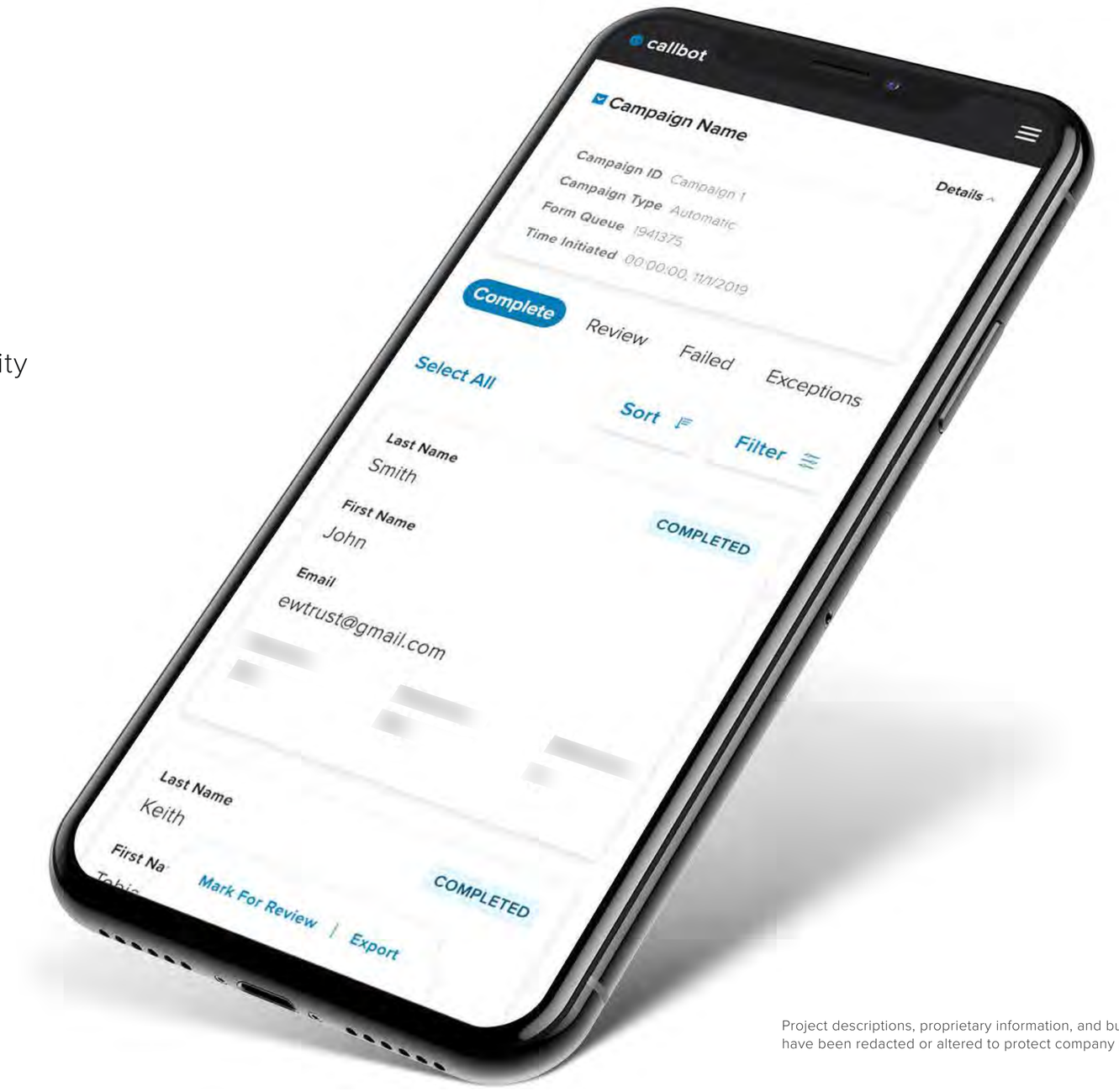
Da Vinci

Comprehensive Medical Survey

42

Callbot

An interface with personality



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

Summary

Callbot is an application is 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.

Approach

User Survey
User Personas
User Interviews
Information Architecture
Task Flows
Wireframes
Paper Testing
Design Prototypes
Usabillity Testing
Rapid Angular Prototypes

Team

Andrew Nicholl
Senior Product Designer

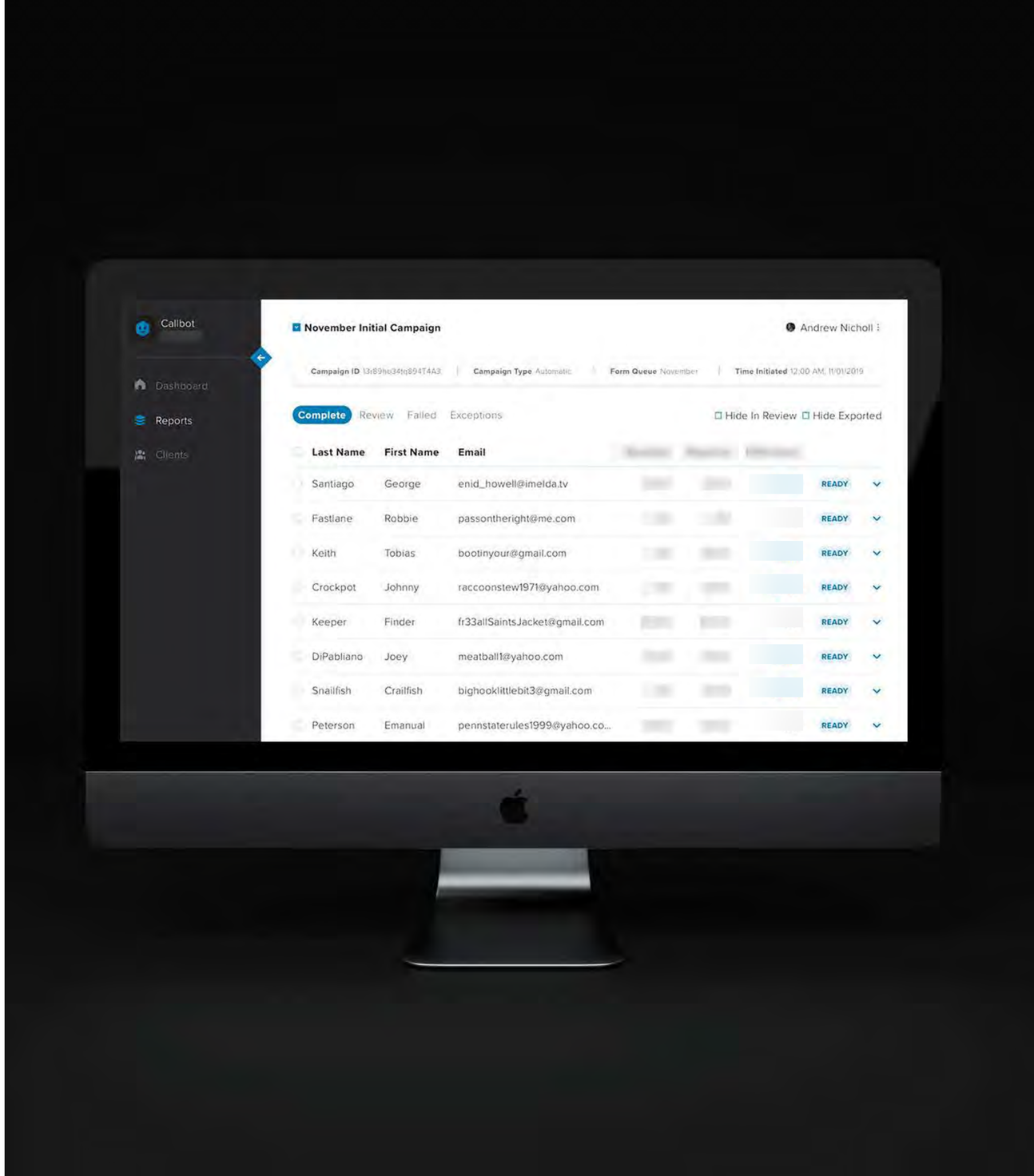
Jimbo Rountree
Product Designer

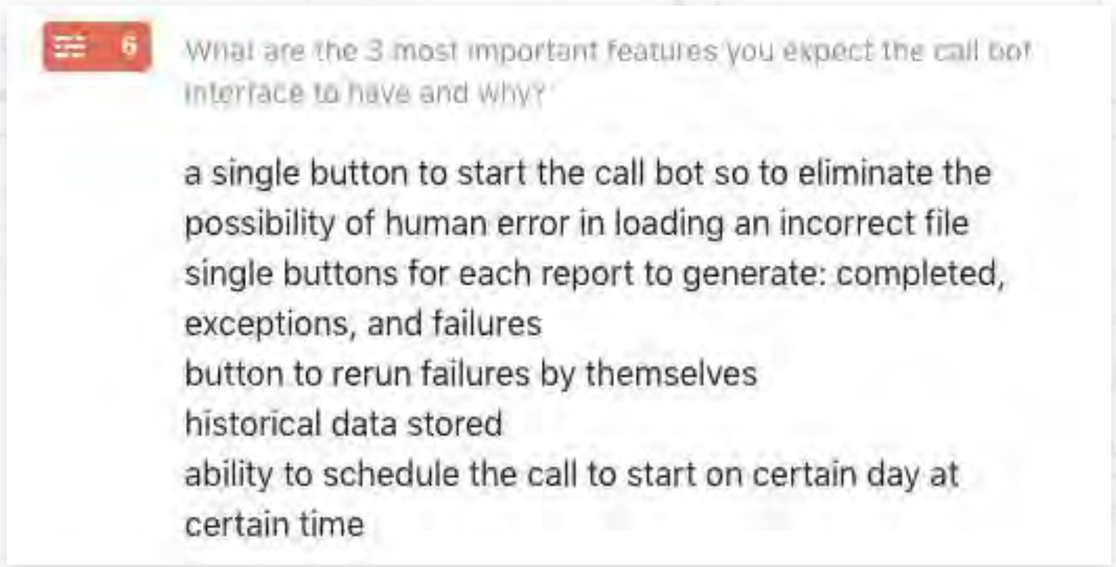
Hugo Ramos
Product Designer

Luke Pate
Lead Software Engineer

Lane Holcombe
Senior Software Engineer

Tory Minars
Project Manager





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.

Top Results

A single button to initiate the process within the application.

Other Essentials

Ability to easily toggle views between different reports

Nice to Have's

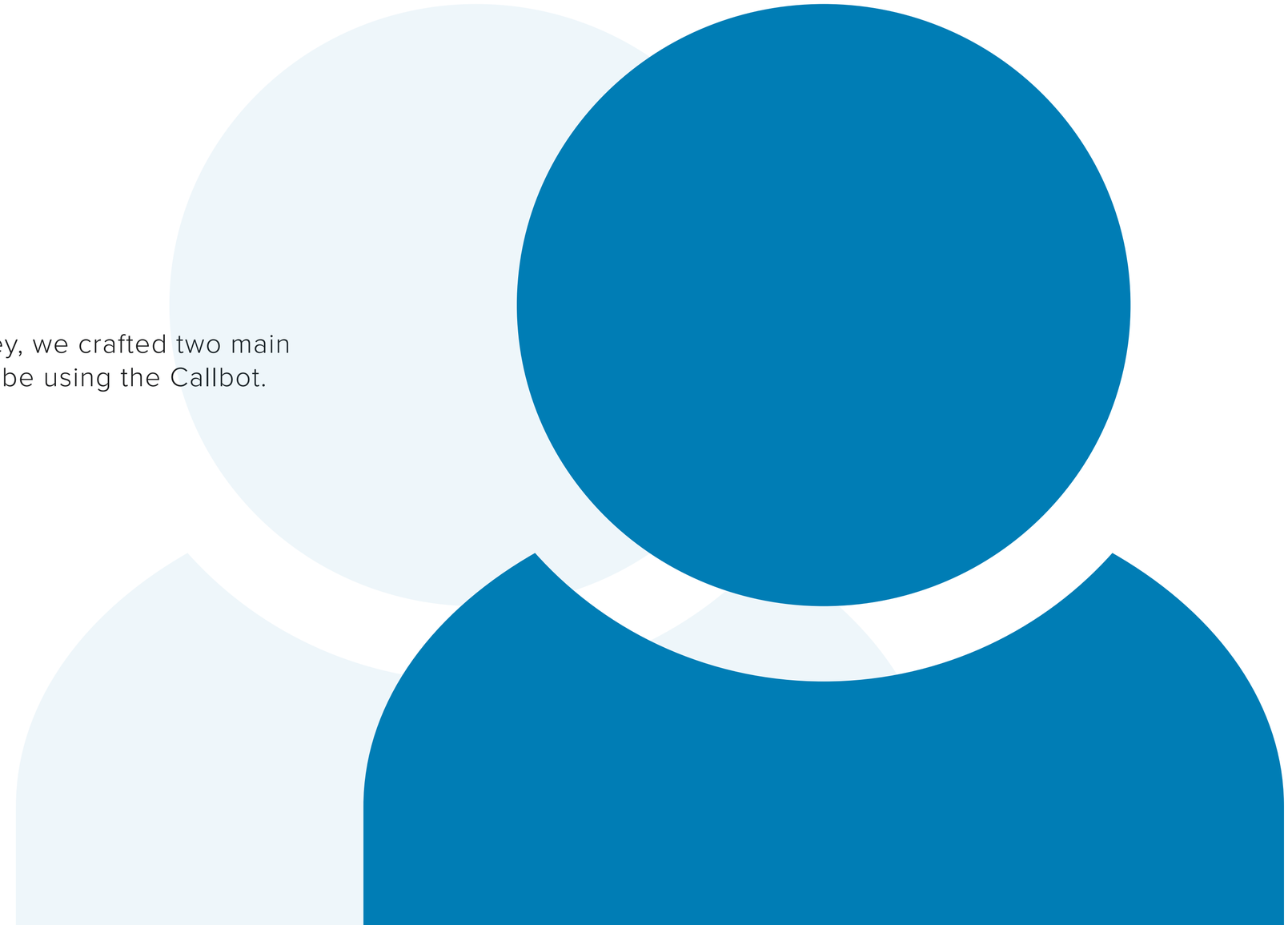
Ability to view all historical data

Ability to easily rerun process if failed

Ability to schedule process to happen at a future time and date

User Personas

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



“I want an easy button.”

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

Accaccurate 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

Readying 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)

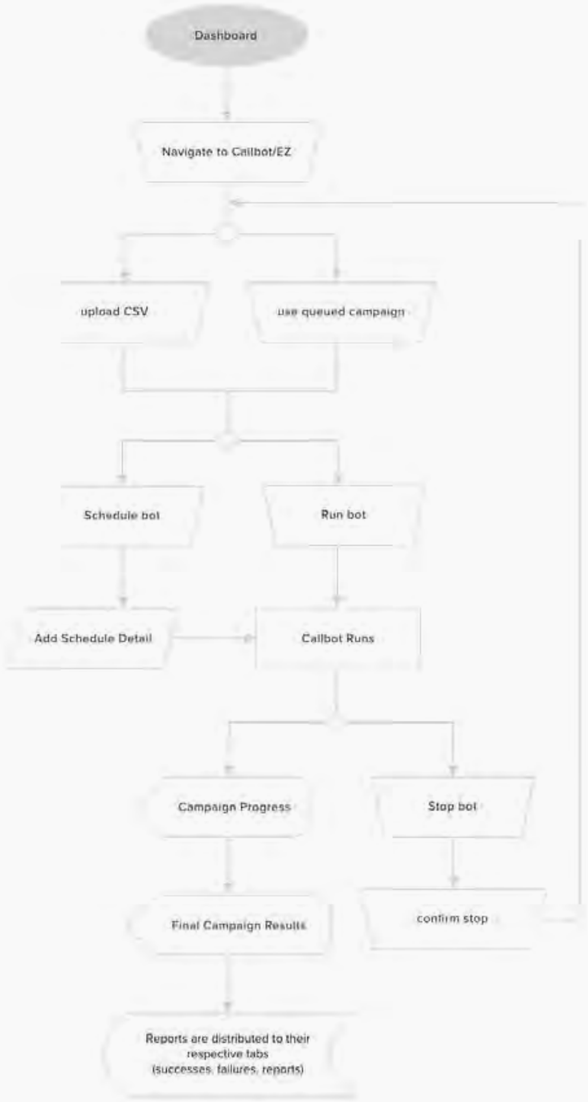
Windows Laptop (Home)

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

“As long as I can quickly see what's what I'll be happy.”



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.

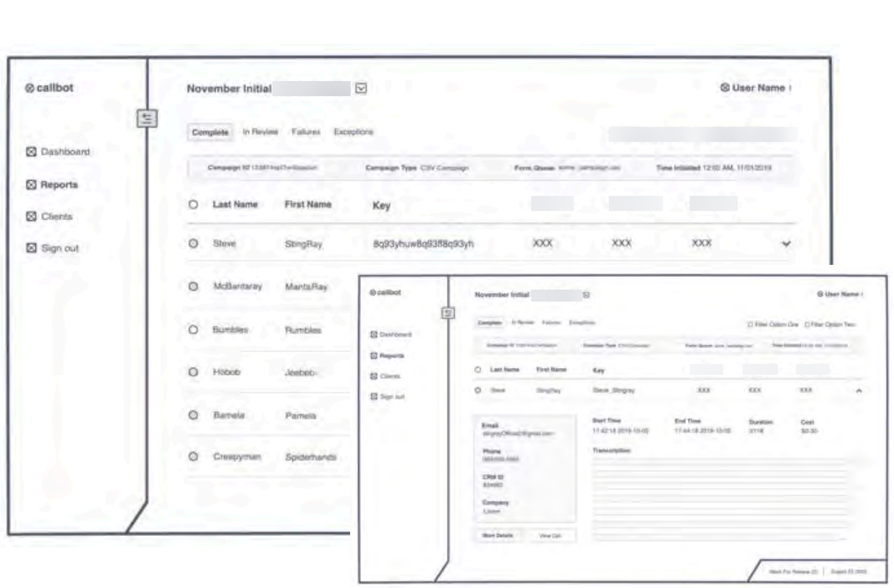
Design Lifecycle

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.

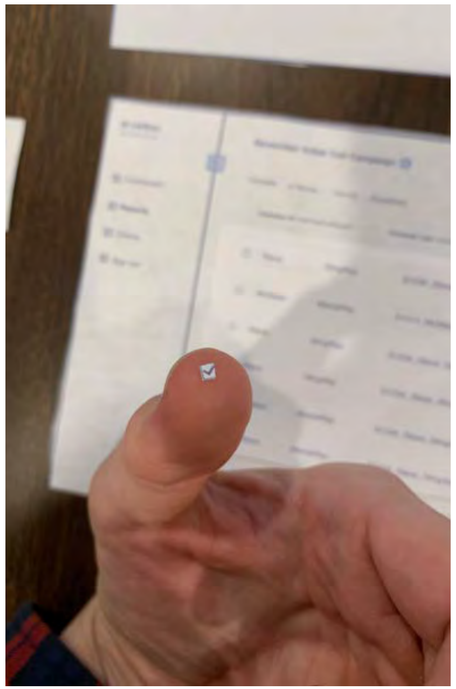
We then replicated our task flows using our wireframes to ensure that we were not missing any necessary steps or features.



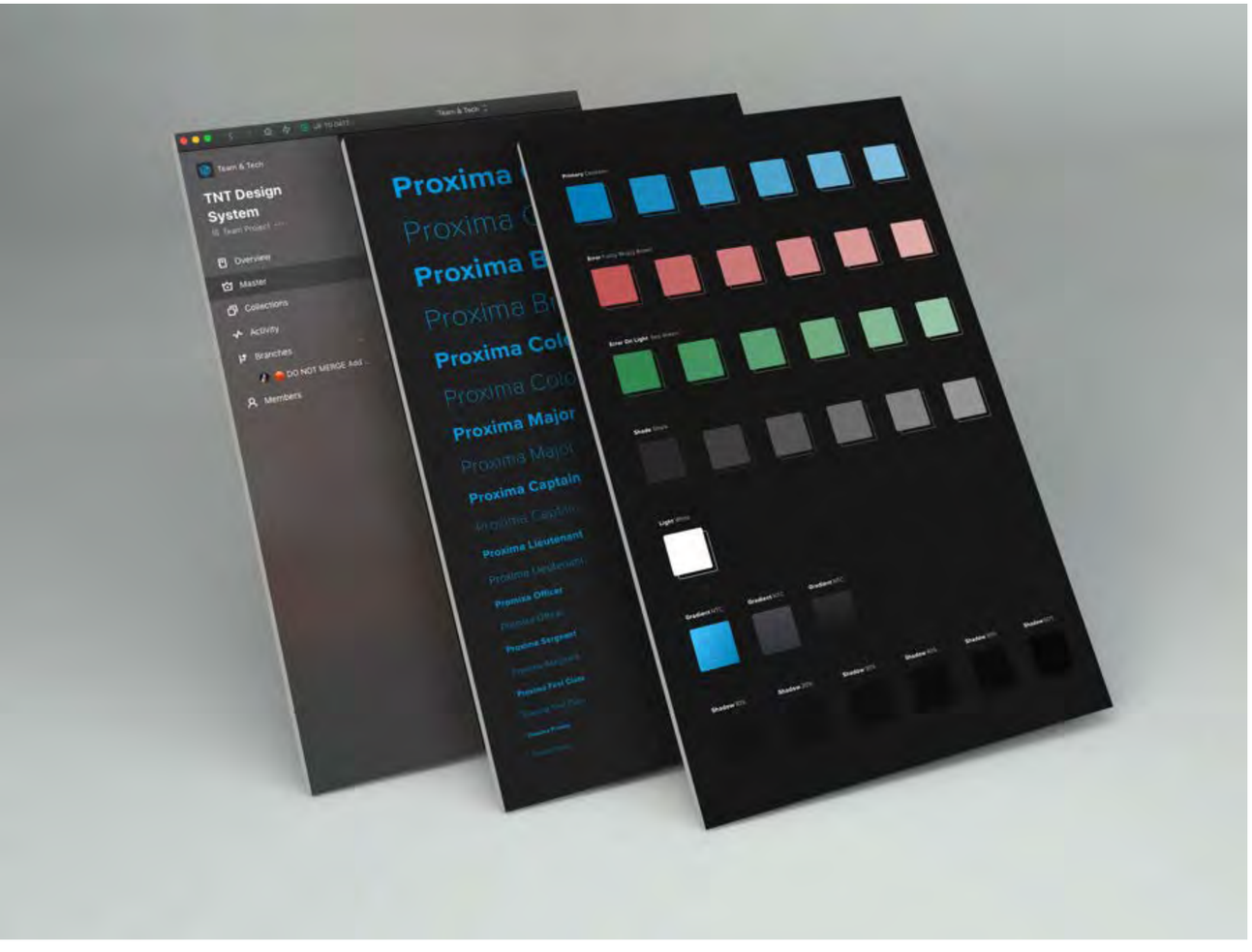
Paper Testing

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 got 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 [redacted] 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"

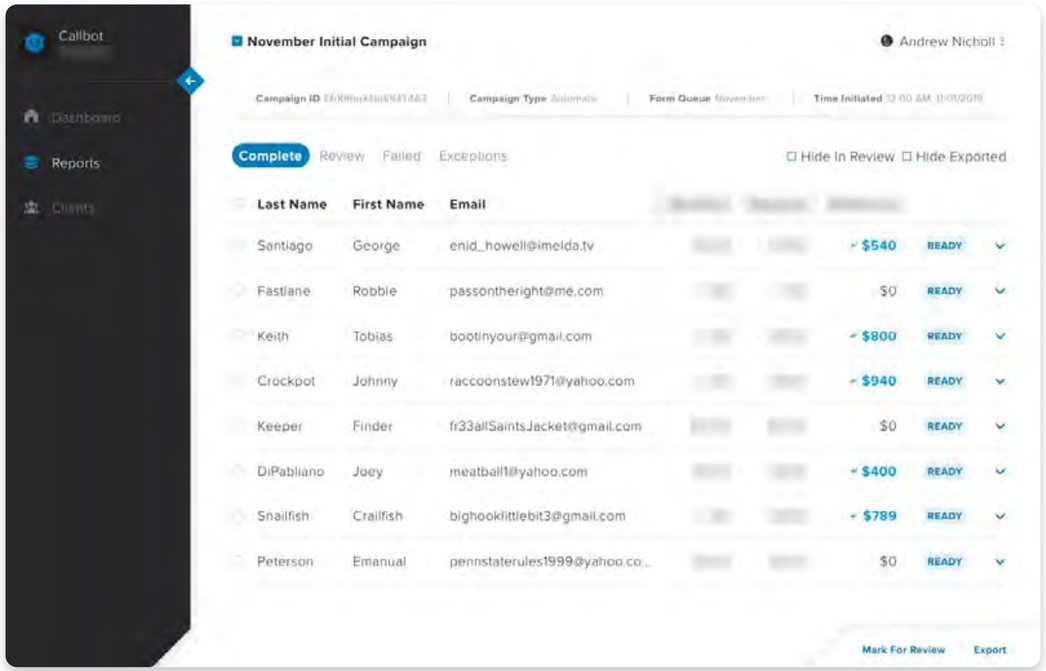


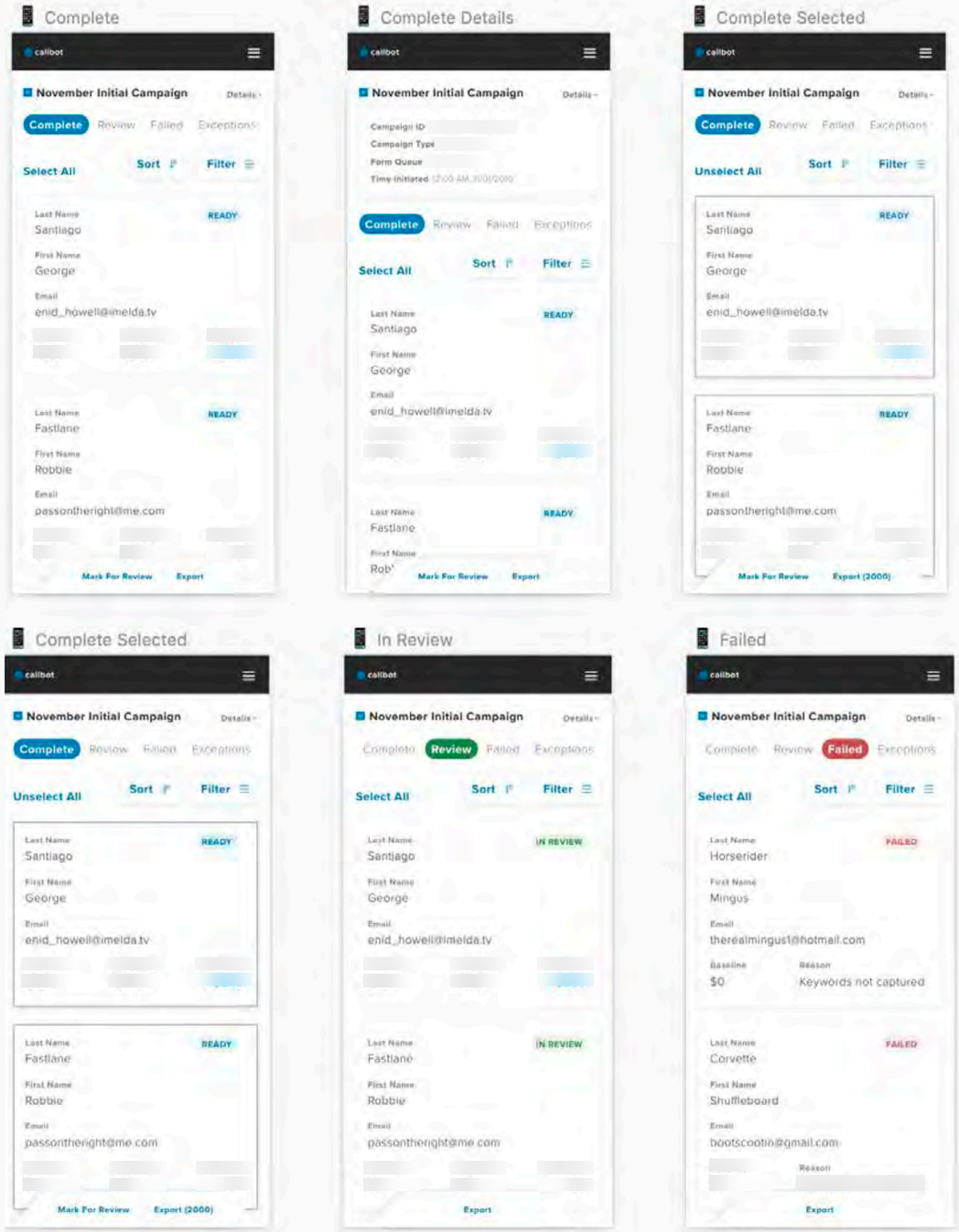
Design System Implementation

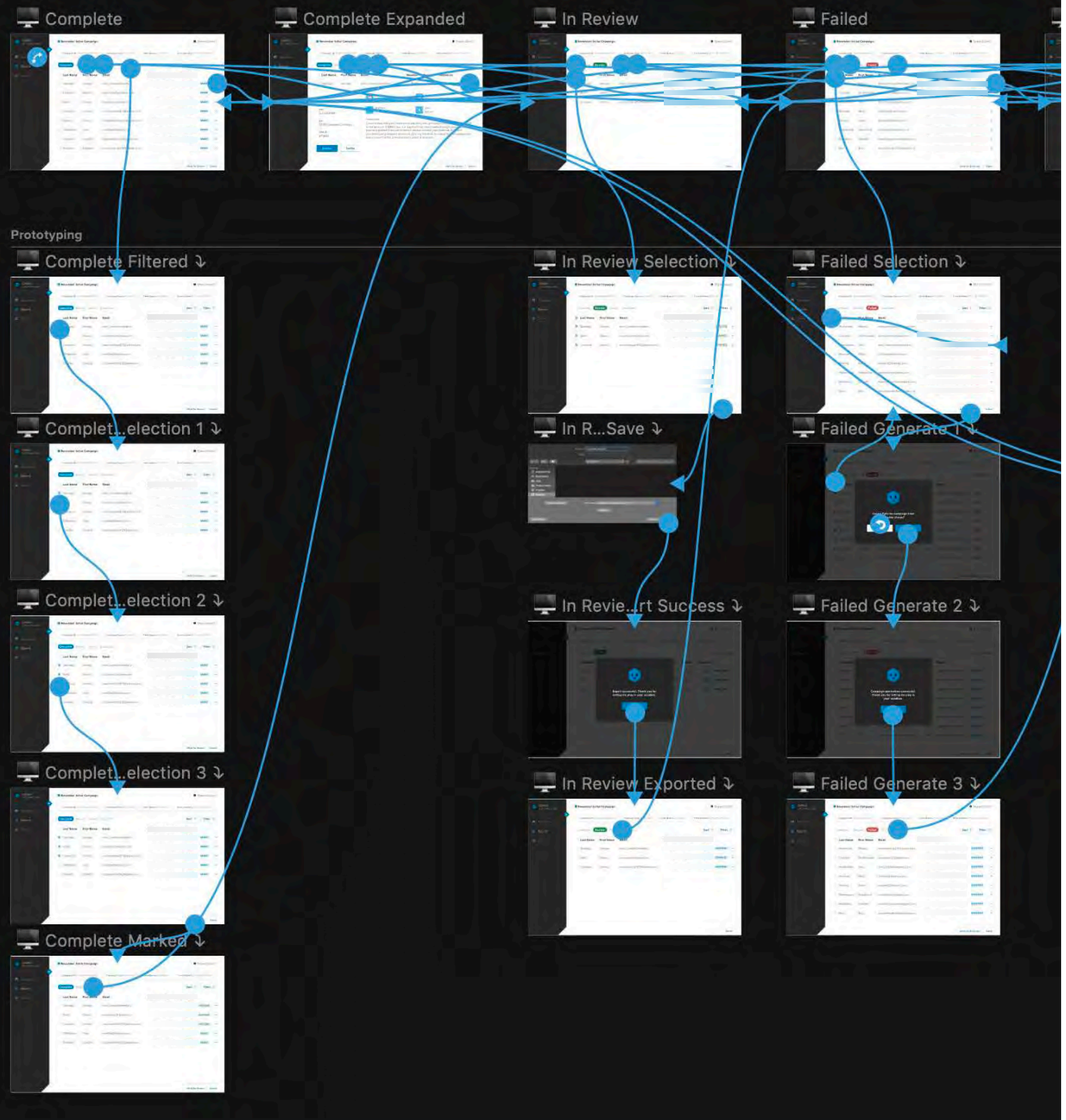
Using our design system for internal products we were able to come into this project with a predetermined set of minor components and styles. These included colors, icons and typography, which was extremely helpful in speeding up the design process, staying organized, and maintaining consistency throughout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





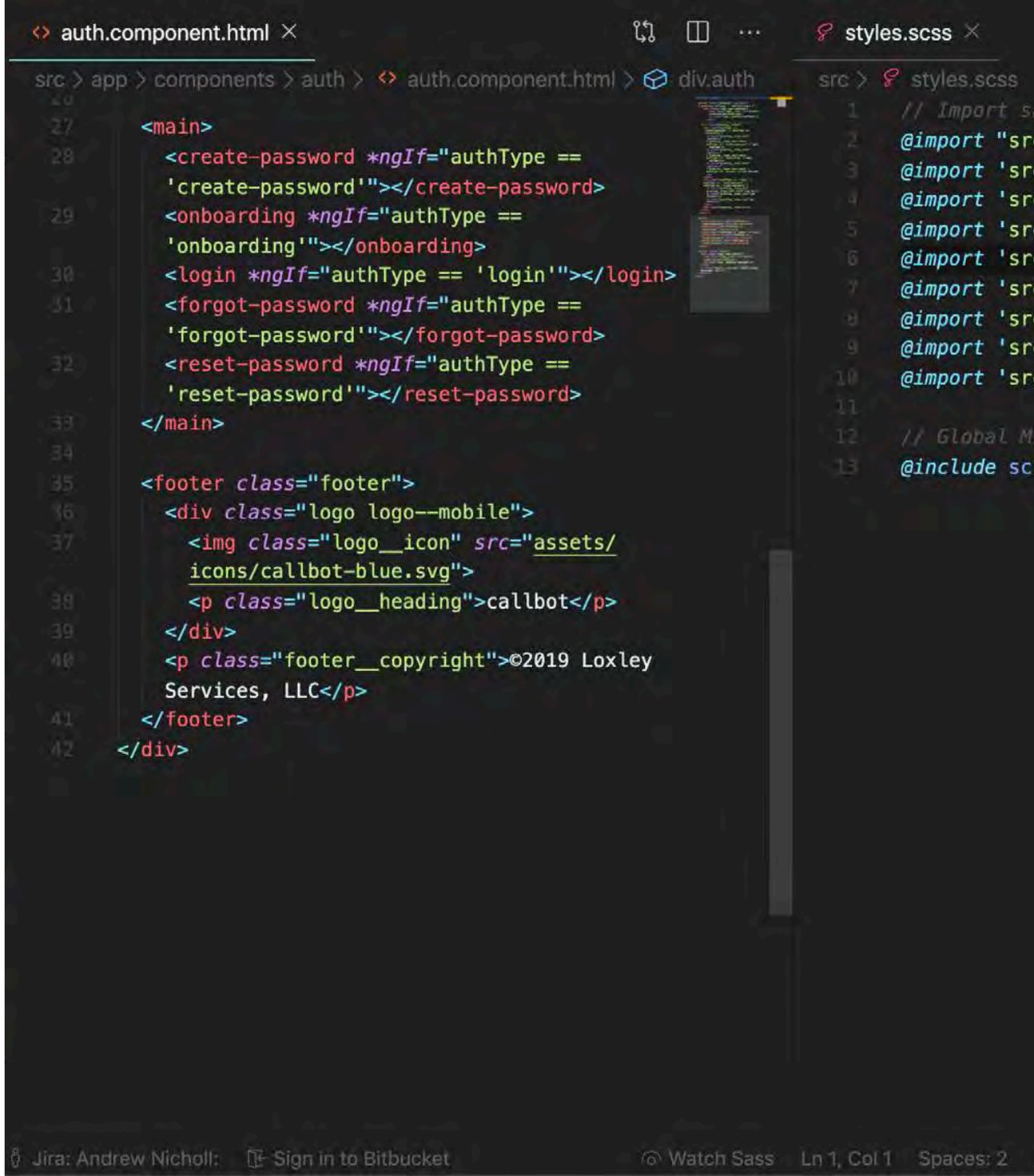


InVision Prototypes

For the more complicated components, we created prototypes using Craft + InVision and conducted usability tests with our primary users before developing prototypes in Angular.

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.



http://localhost:4200/#/

Elements Console 9

```
<!doctype html>
<html lang="en">
▶<head>…</head>
…▼<body class="user-is-not-tabbing" cz-shortcut-listen="true"> == $0
  <!-- Google Tag Manager (noscript) -->
  ▶<noscript>…</noscript>
  <!-- End Google Tag Manager (noscript) -->
  ▼<app-root _nghost-c0 ng-version="6.1.10">
    <router-outlet _ngcontent-c0></router-outlet>
    ▼<dashboard _nghost-c7 class="ng-star-inserted">
      ▼<div _ngcontent-c7 class="app-container">
```

html body.user-is-not-tabbing

Styles Computed Event Listeners

Filter :hov .cls +

```
element.style {
}
body {                                   <style>
  scrollbar-face-color: #007db5;
  scrollbar-track-color: #003248;
}
body {                                   <style>
  color: #282829;
  font-size: 1.6rem;
  line-height: 1.5;
  font-family: "proxima-nova", sans-serif;
  font-weight: 300;
  letter-spacing: 0.025em;
  -webkit-font-smoothing: subpixel-antialiased;
  -moz-osx-font-smoothing: grayscale;
}
body {                                   <style>
  color: #282829;
  font-size: 1.6rem;
  line-height: 1.5;
  font-family: "proxima-nova", sans-serif;
  font-weight: 300;
  letter-spacing: 0.025em;
  -webkit-font-smoothing: subpixel-antialiased;
  -moz-osx-font-smoothing: grayscale;
}
body {                                   <style>
  background: ▸ #ffffff;
  box-sizing: border-box;
```

Callbot

Dashboard

Reports

Clients

Sign Out

Campaign Name

John Doe

Campaign ID Campaign 1 **Campaign Type** Automatic

Form Queue 1941375 **Time Initiated** 00:00:00, 11/1/2019

Complete Review Failed Exceptions

Sort Filter

Last Name	First Name	Email				
Smith	John	ewtrust@gm...				COMPLETED
Keith	Tobias	bootinyour...				COMPLETED
Crockpot	Johnny	raccoonste...				COMPLETED
DiPabliano	Joey	meatball1@y...				COMPLETED
Peterson	Emanuel	pennstateru...				COMPLETED
Waits	Tom	heartofsatur...				COMPLETED
Blastoff	Jawana	maxfun100...				COMPLETED

Mark For Review | Export

Rapid Prototypes

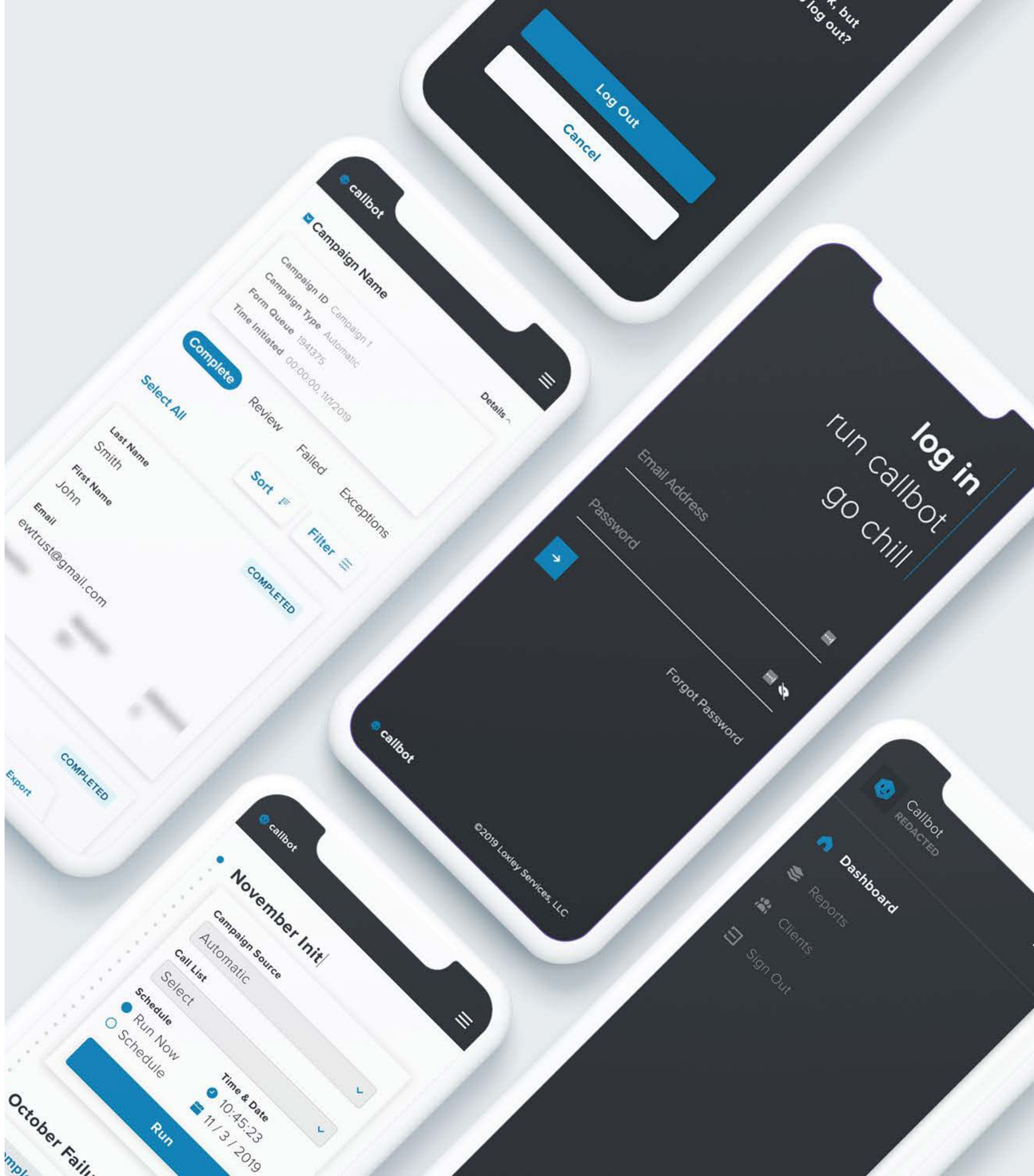
By having the product design team spearhead development, we were able to ensure that all interactions and design requirements were executed to spec and that our software engineers would have a solid front-end foundation to work from at the end of each sprint.

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.

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.



Callbot

Project Successes

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.

User Focused

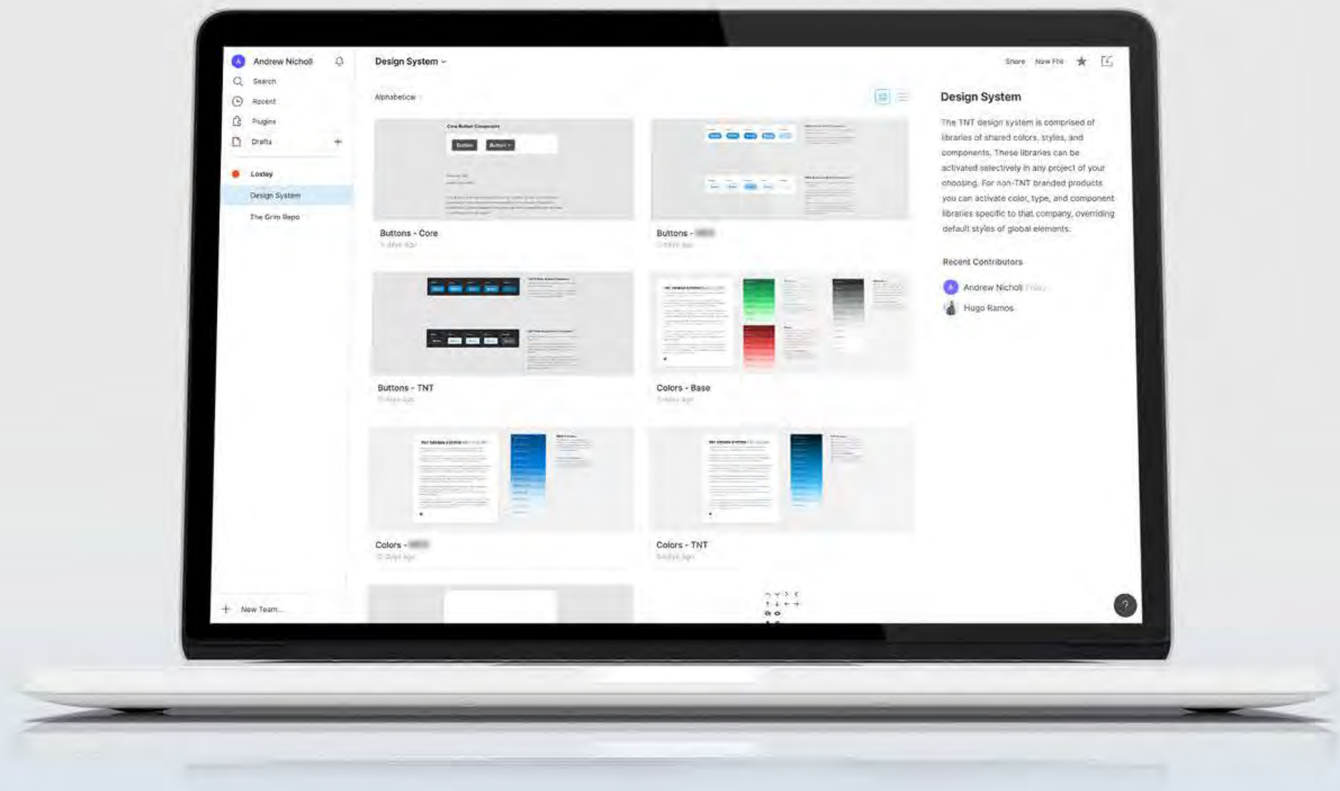
Not only did it keep work interesting, it made for an overall better product, as we were able to conduct usability test between sprints and quickly implement changes to the previous sprint's component and embrace new ideas for the next one.

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.

TNT Design System

A cohesive and scalable design system for a multi-brand ecosystem



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

Summary

The TNT design system is comprised of libraries of shared colors, styles, components, and their corresponding documentation and principles. We originally created this design system leveraging Sketch Libraries + Abstract for consumption by designers and for design spec handoff to developers.

This project involved auditing that design system, migrating it to Figma, creating documentation, and keeping our components up to date in dev.

Project Goal

Streamline the design and development process and allow for greater reusability between components across different brands within the company's ecosystem.

Team

Andrew Nicholl

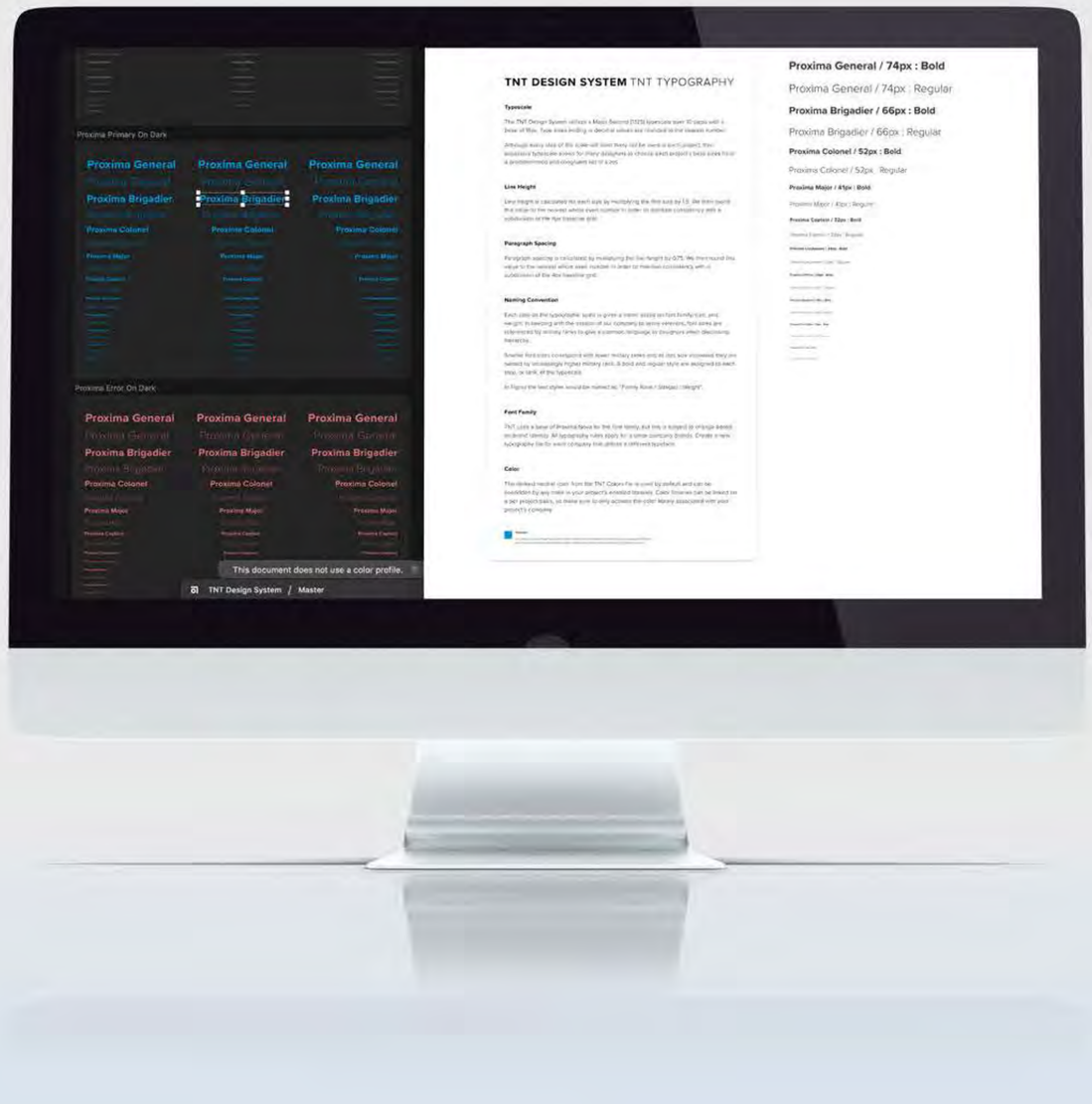
Senior Product Designer

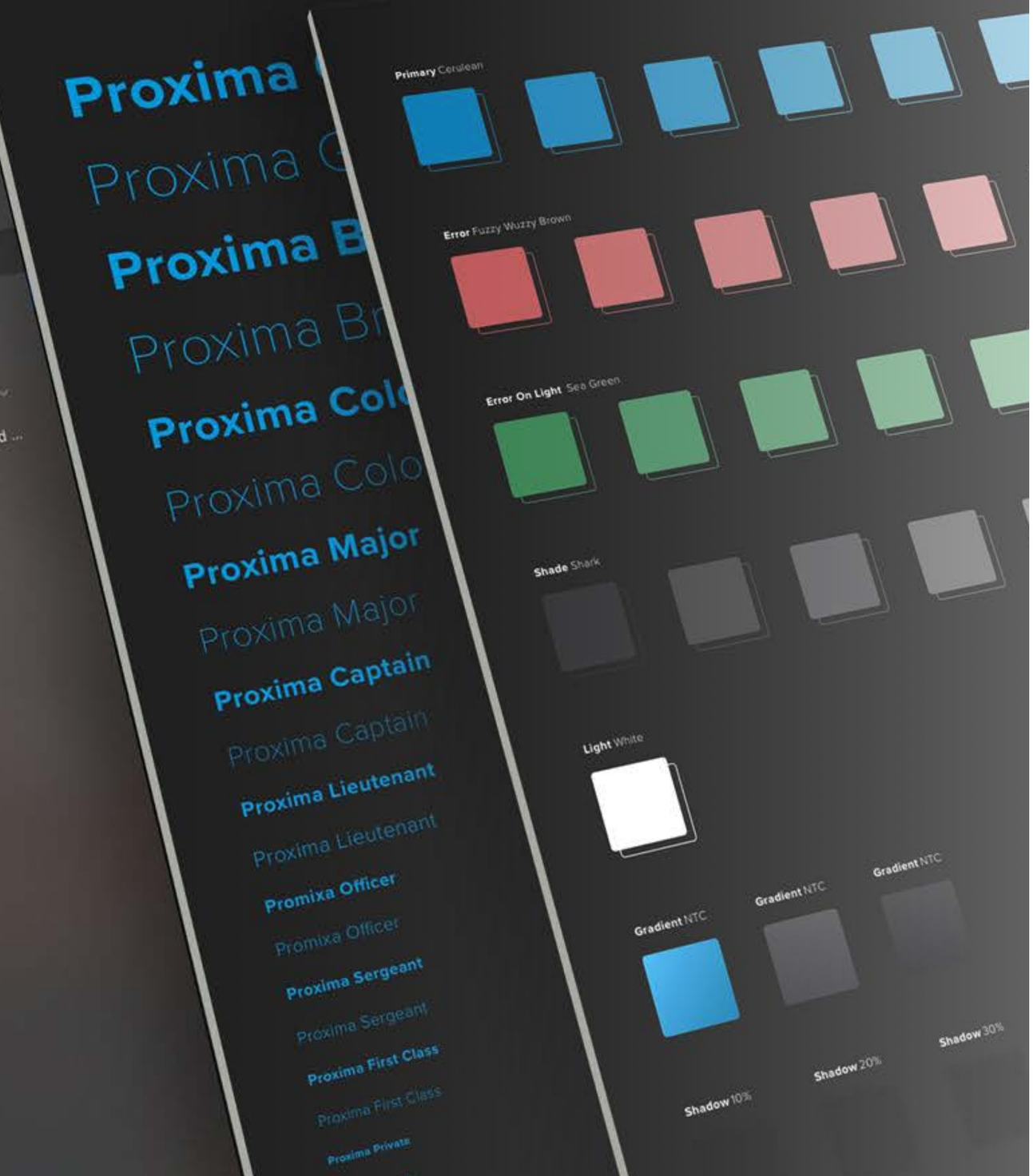
Jimbo Rountree

Product Designer

Hugo Ramos

Product Designer





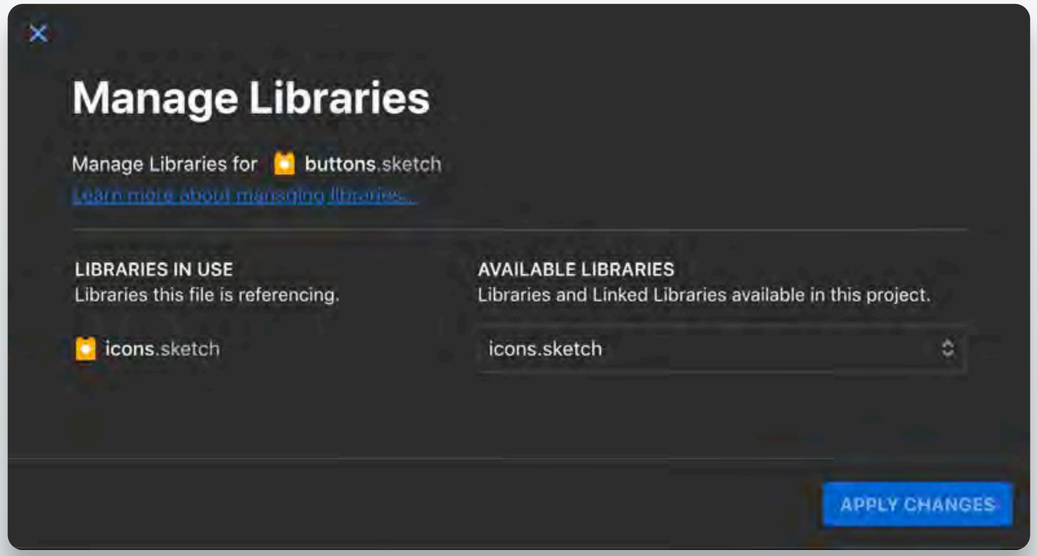
Design System Audit

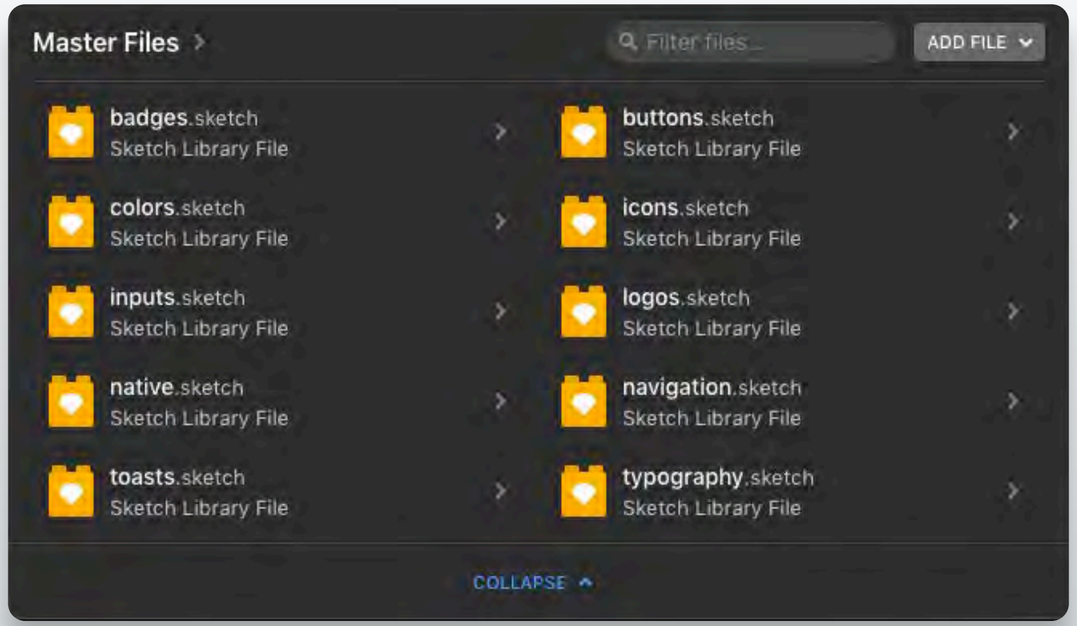
The decision to restructure our design system architecture was not made lightly. Although the product design team was attracted to trying new tools, we did not want to create technical debt that would not pay off in the long run. We decided to do a full assessment of our current system,

During our end of year design system assessment, we mapped out our current design system structure to see where the system was serving us well and where it was failing to meet our needs.

Sketch + Abstract

The first iteration of our design library was created in Sketch + Abstract. We utilized Sketch Libraries to house all of our symbols, which made them available to any linked files or additional libraries. Because we were using Abstract as our version control tool, we were able to link libraries selectively, rather than enable/disable them in Sketch's preferences.





Component Based Libraries

Our Sketch Library files were broken up into components which fed each other. This was the recommended approach from the team at Abstract, and ultimately won us over as opposed to atomic design.

Organizing components by context

Using this approach we were able to break our libraries down into the the following structure: colors, typography, icons, inputs, buttons, toasts, badges, and so on for all individual components.

In this model, we could duplicate each symbol, break their links to their style libraries, and reattach them to a new style library. his allowed us to see all of our buttons in one place, and make sure that they stay consistent with our sizing and spacing guidelines, although they may differ in branding elements or such as color or typography.

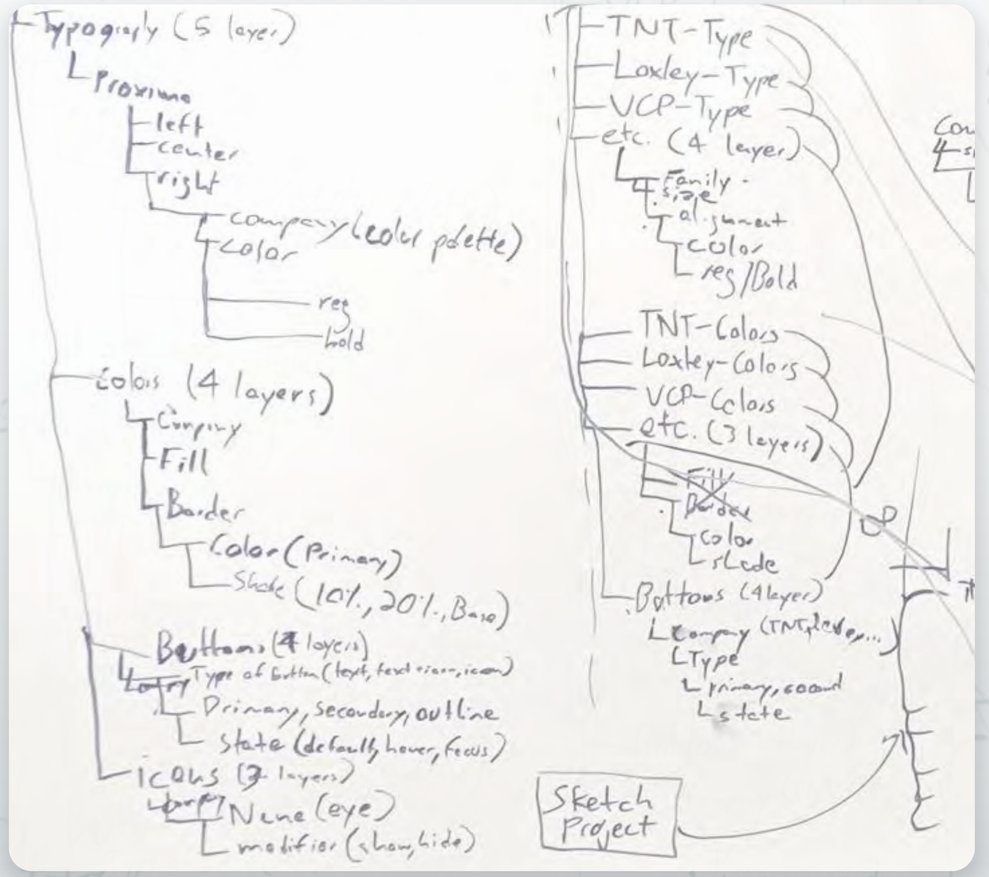
Making Connections

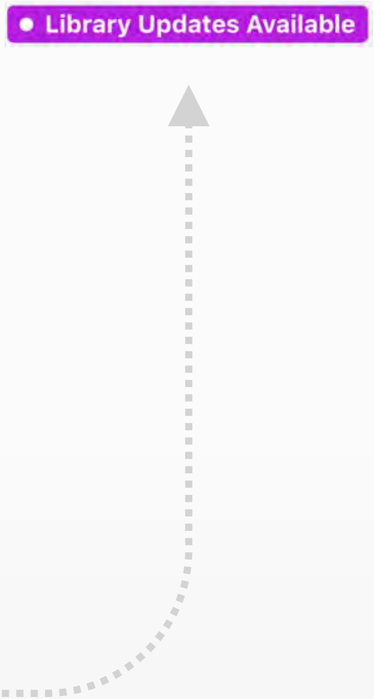
Our current system made these connections between companies by breaking up the companies into pages of the same "component.sketch" file.

Visualizing the Process

Surprisingly, our best alternative to this would be to break up the component files into company specific libraries and have the color and type libraries also live on the same layer as the parent components file. We could then use pages for each component, rather than a variant of a component.

After diagraming out how many layers were in each component/style library, we realized that this approach created less "library updates" between linked files. This meant that a change in one core area of the design system, such as color, would propagate to their components in a single library update, as opposed to a minimum of three update prompts (color -> typography -> icon -> component). While this did not allow us to open one file and see all of our companies buttons in one place, it did allow us to streamline updates when working on a project for a single company within our ecosystem.





But wait, that’s not the problem

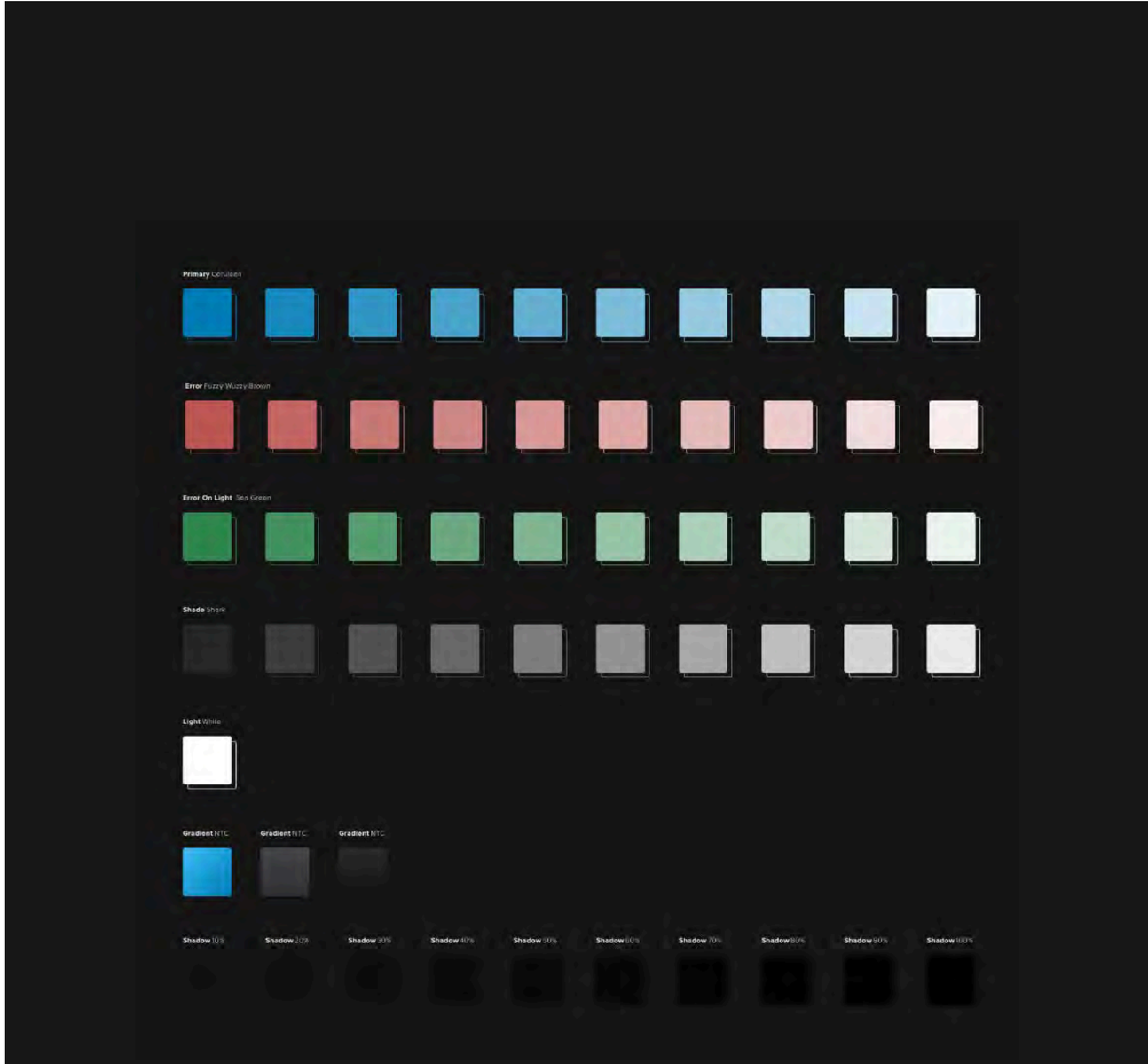
While all of these approaches would help us keep things organized and streamline library updates, it did not solve the problem that we were still not staying DRY in our components. The major issuse came down to Sketch's style system.

A Deeper Look

In order to know exactly what issues with the Sketch's style system we were trying to address, we had to take a closer look at what was costing us the most time and making us perform the most maintenance in our current model. We identified two main issues: colors and type.

Repeating Colors

A major limitation in Sketch's color styles in sketch is that they are inherently a fill or a border, which are tethered to properties such as opacity, fill mode, and border width. Similar to text styles, color styles end up being duplicated depending on use case. Another major issue was that neither fills nor border styles could be applied to text, causing a further duplication and maintenance of hex values in our design libraries.



Managing Typescales

All designers using Sketch will be familiar with the myriad rows of typography scales of different alignments and colors within their projects. This is due to the fact that Sketch does not allow for color styles to be applied to text without the assistance of some newer plugins that are not supported at an enterprise level, nor does it allow for alignment to be altered on a given text style.

This typically results in three rows of type (left, center, and right aligned) for each color of each font family in a design system. When taking accessibility into account for on-light and on-dark colors, this can get out of hand fast. It also means that if marketing needs to make a color change, UI designers must make that change in at least two libraries on multiple elements, leaving room for error and less time to actually design the applications these systems are meant to support.

The Big Question

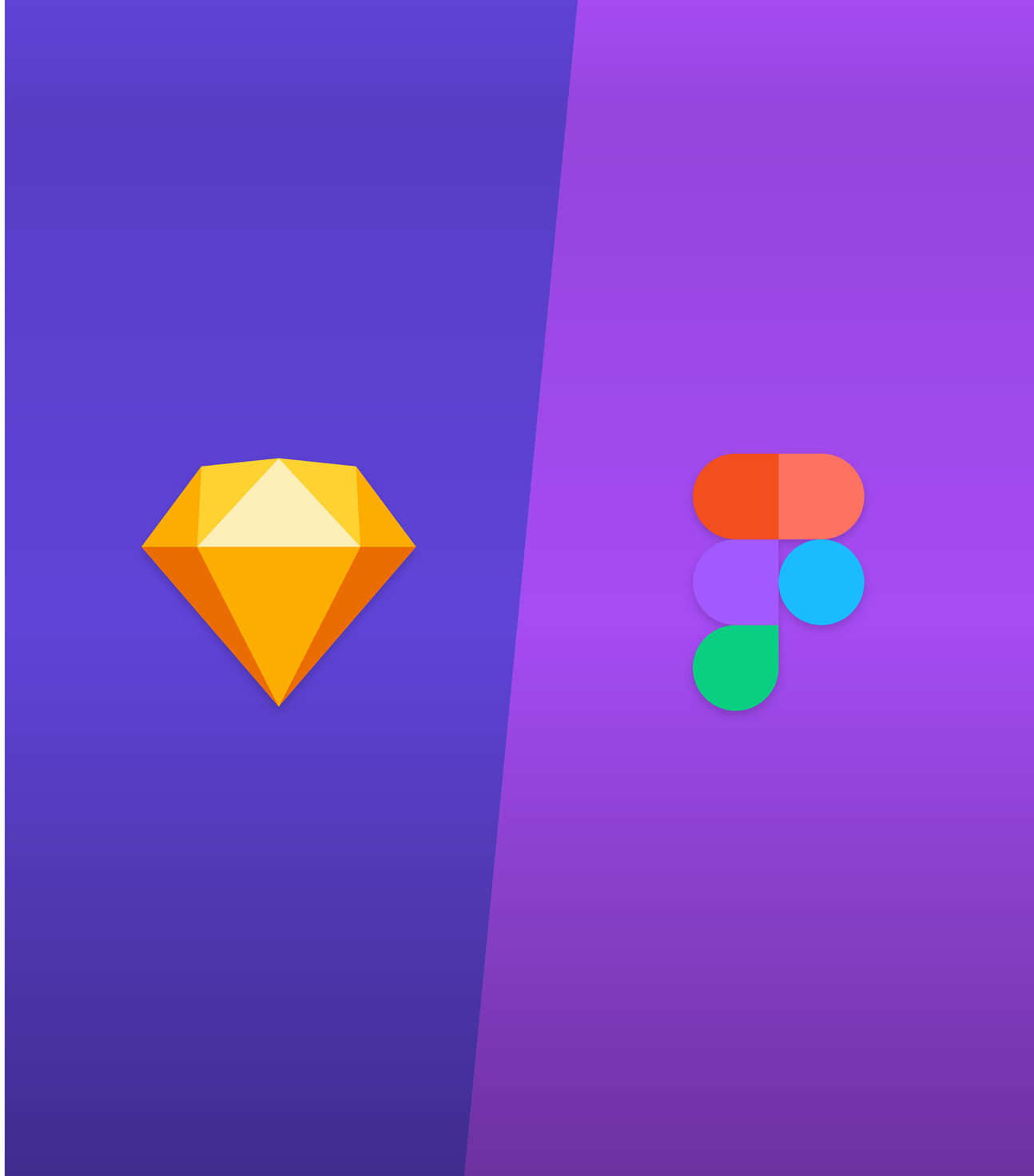
After creating a a diagram of our new library structure and identifying the main issues with our current platforms, we could see updating to a new pattern would not be a small task and some inherent software limitations would remain in place.

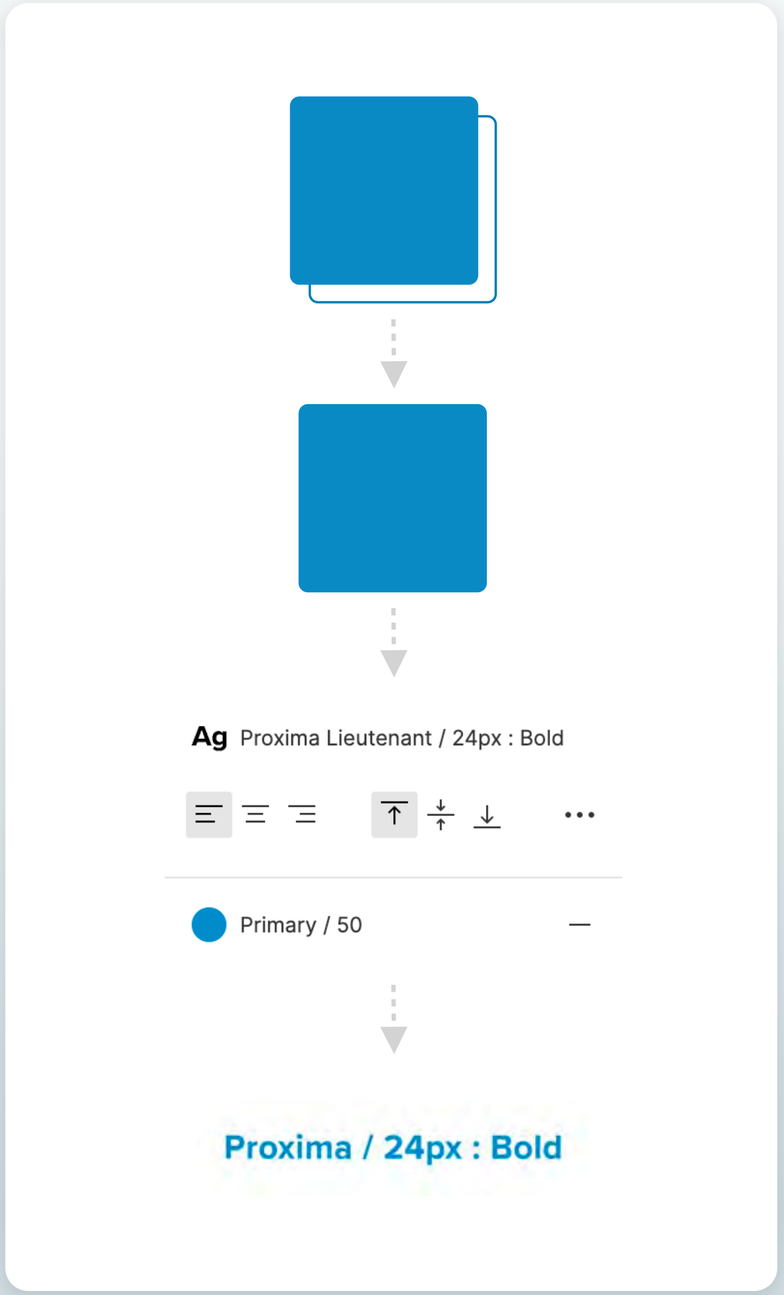
Shifting Our Current Pattern

This would involve symbol swaps, style swaps, and purges of unused styles within our libraries. Seeing the scope of work involved in the shift, we were forced to ask ourselves if we were solving a problem or fighting against a system that was no longer compatible with our workflow.

We Doin' This?

We had not wanted to drastically alter our design workflow if it meant any downtime for business, especially if we couldn't be convinced that a change would actually solve our problems. So we started researching other design platforms that may solve our issues in preparation for creating a proof of concept. After conducting small tests in multiple platforms, we ultimately decided that Figma looked the most promising.





Proof of Concept

After identifying color and text styles as the primary cause of bloat, repetition, and manual updates in our design files, we knew that this would be the first thing we needed to test in Figma.

Testing Colors

We started small. Very small. We created a Colors file in our Design System project and chose a primary color and a secondary color and created a color style out of each. We then published this library and activated it in our Playground file.

I drew a box and applied the primary color as a fill. Then I added a border and applied the same color. It worked. This was proof already that we could cut our color library maintenance in half.

Testing Typography

We then created a Typography file in our design system and brought our TNT Typescale (more on that later) into the project.

Then we applied our Primary color style to the text style. We then created a text style from it and changed the color to the secondary color. It did not appear to break the link between the text style and the line of text. I had to confirm. I changed the font size of the text style. The line of text updated. Holy. Bam.

Our decision

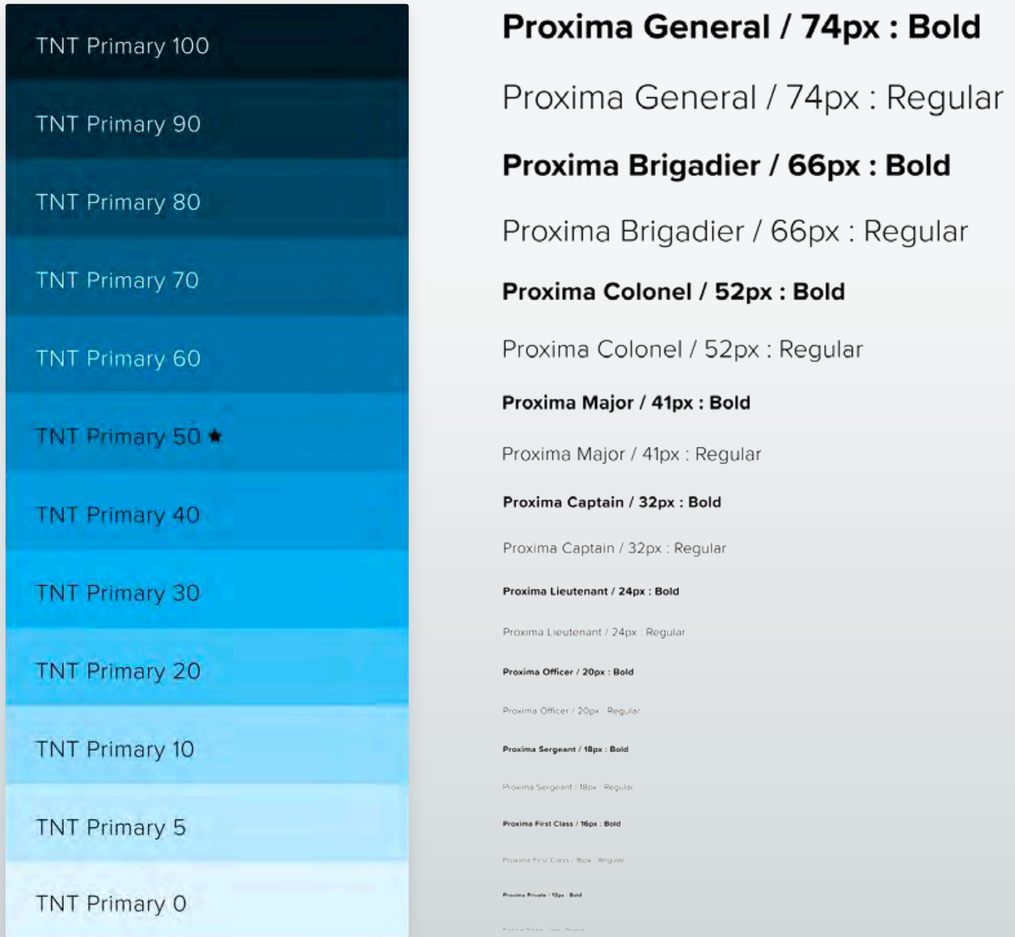
We expanded our proof of concept to encompass a full color scale, type scale, and a set of our core icons.

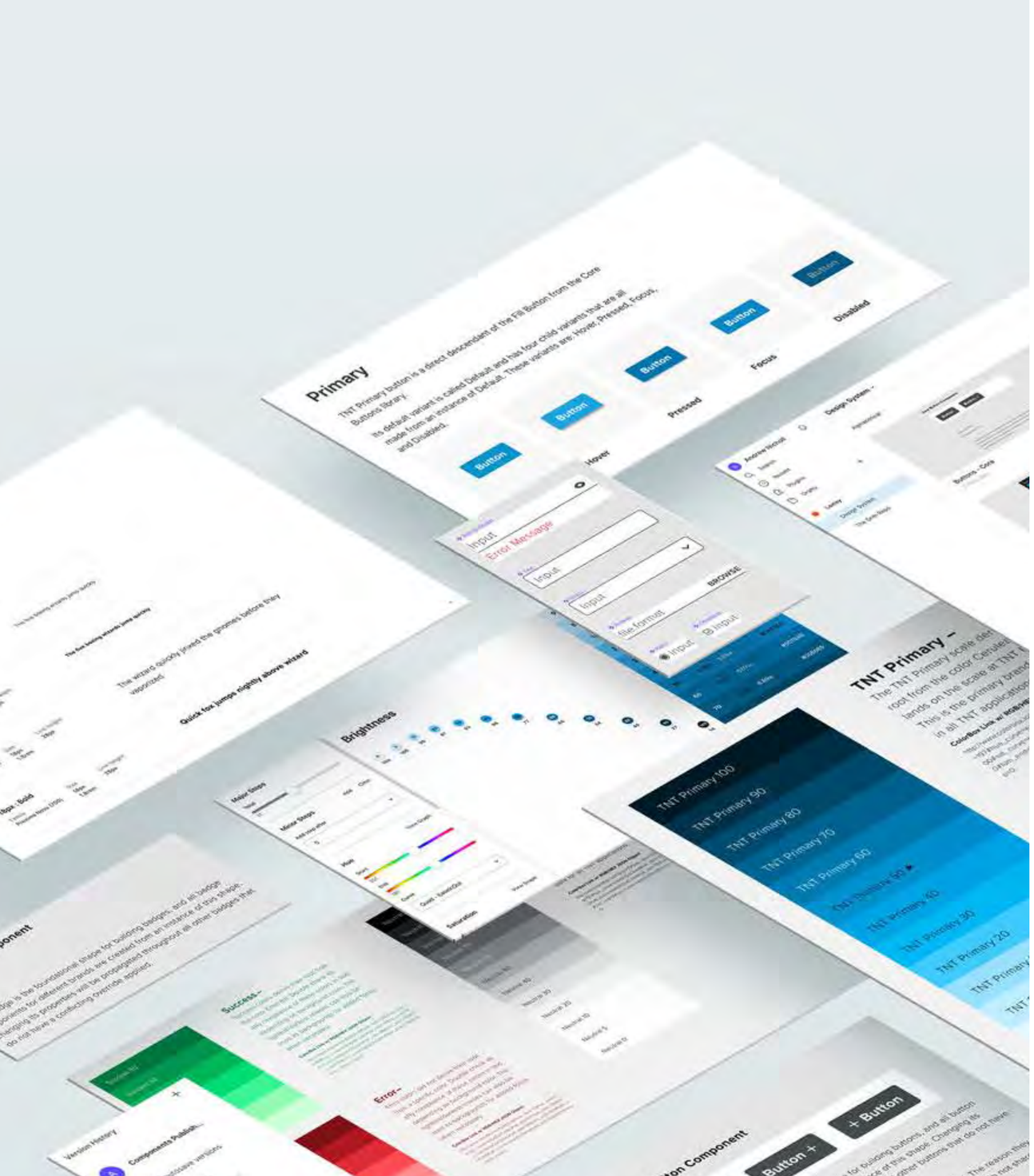
Rapid Implementation

In less than a day we were publishing libraries, turning them on and off, using components with color and text overrides to make more and more connected components, and keeping everything tied to the core components. This is the kind of thing that would have taken days, maybe weeks, and multiple plugins to accomplish using Sketch.

Setting a Goal

With how efficient Figma was feeling, we felt that we could make a full design system proof of concept in one or two sprints. If we hit that goal, it would be validation enough that Figma was an effective tool to use. We made the decision to go all in.





Making the Move

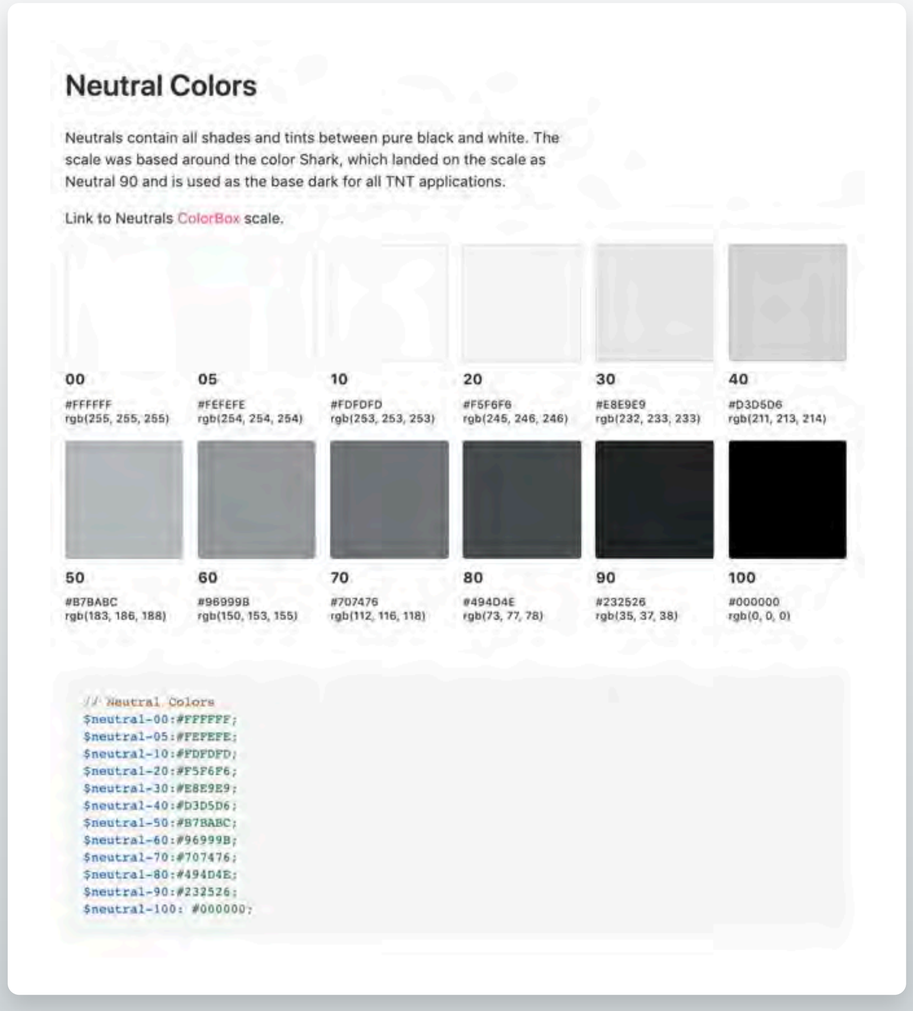
We used the opportunity of switching to a new tool to review each core piece of our design system as we migrated our libraries from Sketch to Figma and see how we could improve our system.

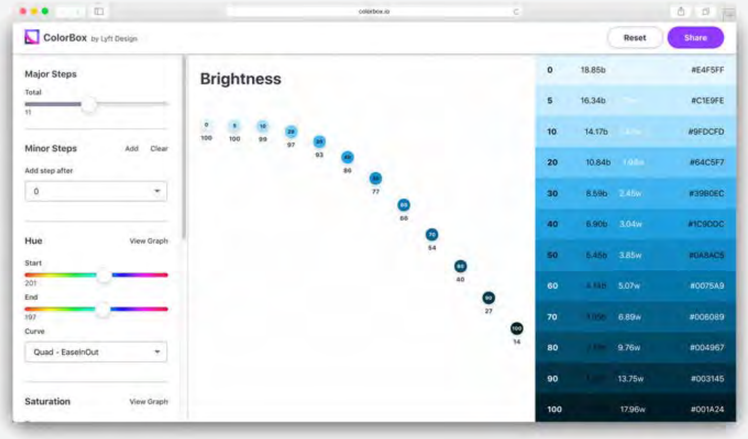
Migrating Colors

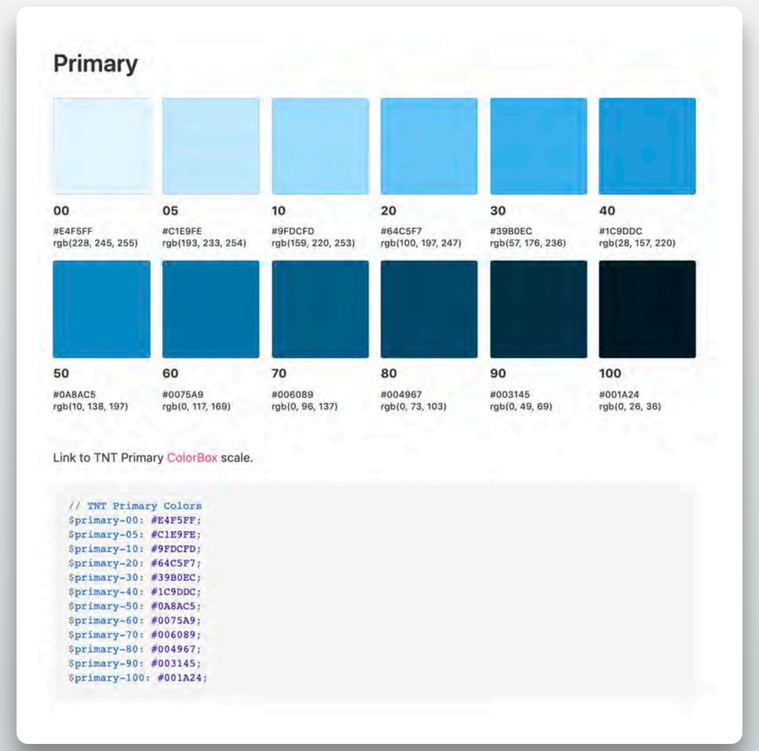
Since Figma allowed us to apply color styles to text, we knew that we would no longer need type scales of multiple colors with variants of each for ensuring accessibility compliance on light and dark backgrounds.

This meant we could reliably use shades and tints of our core colors and primary brand colors on typography, as long as the color scales followed a consistent naming convention that indicated their accessibility compliance with different color combinations.

This would make it possible for designers to have a full range of colors available for different design components, but provide a common and consistent language when communicating with development.







Color Conventions

After reviewing many color naming conventions, we decided upon the numbering conventions recommended by the Lyft design team that corresponds with their colorbox.io platform.

Staying on Brand

The platform allows you to lock in a hex value base your scale from, which we used as a way to maintain consistency with our brand colors. It then allows you to set different numbers of steps between colors and change the curve and rate at which you derive tints and shades from.

Ensuring Accessibility

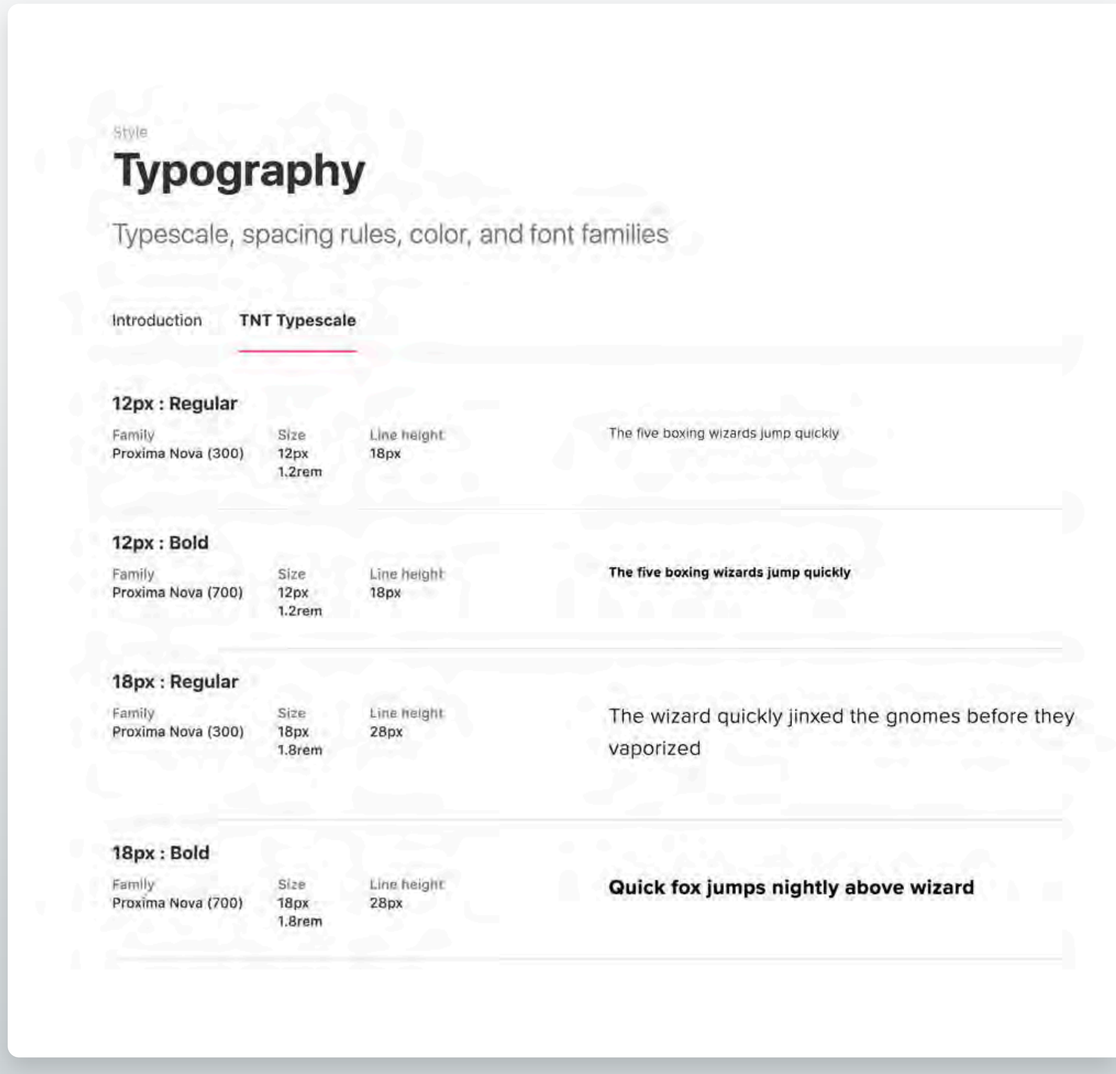
We decided to break the scale into 11 steps and ranges from 0-100 for Neutral and Primary Colors having a minimum of AA a11y compliance with white from 0-50 and with black from 60-100. For Success and Error Colors, we minimized scale is minimized to 6 steps, with a minimum of AA a11y compliance with white from 0-20 and with black from 30-50.

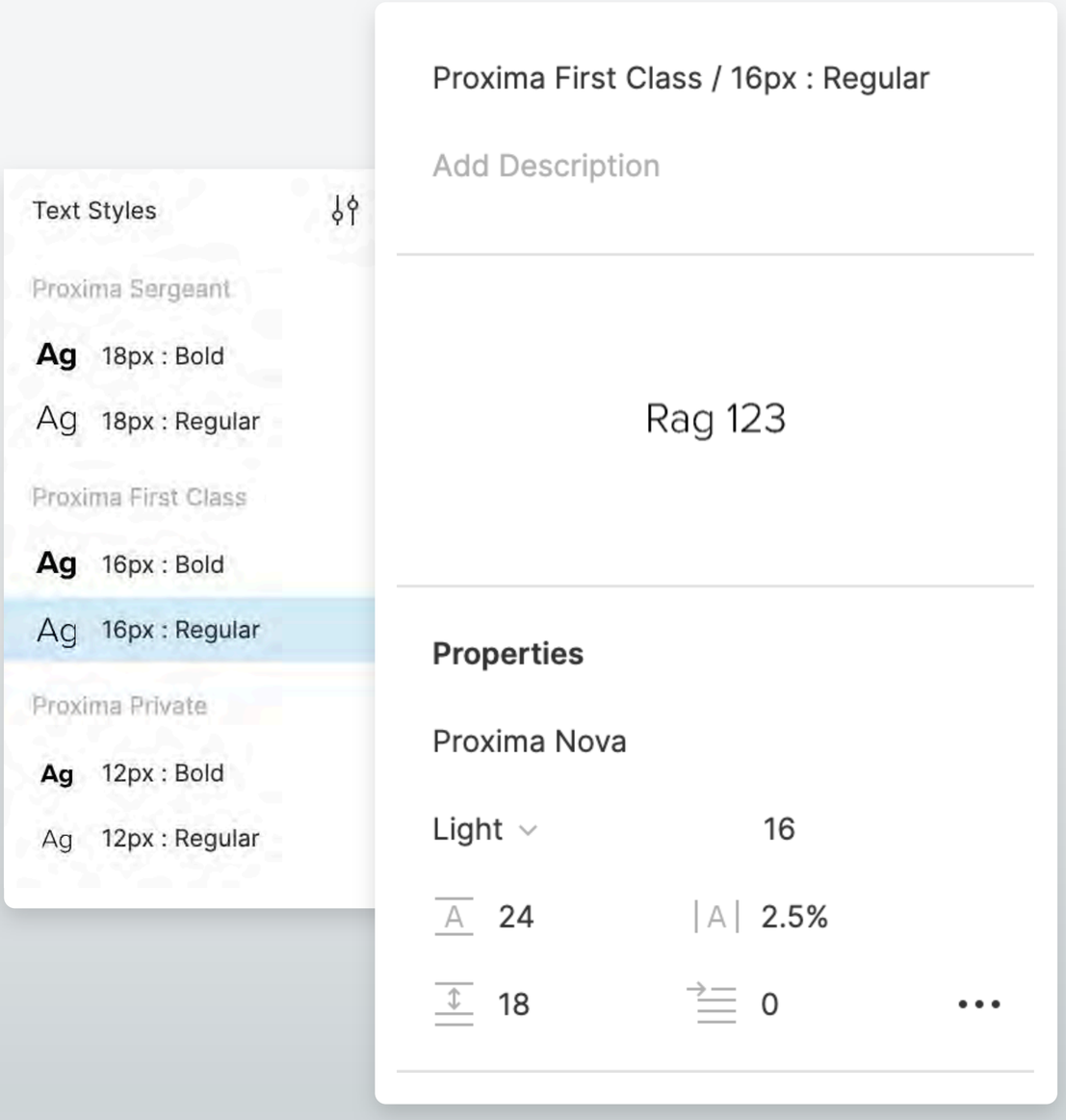
Naming Variables

In development, we named our Sass variables according to the same scale to ensure that all designers and developers were using the same language. We named each color scale of Primary and Secondary colors with the same variable names, but we separated the stylesheets. These stylesheets were published in a private NPM package in order to be available for import individually by any project, ensuring that we could use shared components amongst different brands without changing variable names.

Typography

We'd already brought the foundation of our typography while creating text styles for our main font family, Proxima Nova. We could now get down to the nitty gritty of adjusting line-height and paragraph spacing in order to stick to our 4px baseline grid.





Foundation

Our type scale utilized a Major Second (1.125) type scale over 10 steps with a base of 16px. Type sizes ending in decimal values are rounded to the nearest number.

Although every step of the scale will most likely not be used in each project, this expansive type scale allows for many designers to choose each project's base sizes from a predetermined and congruent set of sizes.

Line Height

Line height was calculated for each size by multiplying the font size by 1.5. We then round this value to the nearest whole even number in order to maintain consistency with a subdivision of the 4px baseline grid. Using this tactic, we could stick to our baseline grid while also providing the most accurate portrayal of how text will display in development, since we generally use 1.5em as our default line-height size.

Paragraph Spacing

Paragraph spacing was calculated by multiplying the line height by 0.75. We then round this value to the nearest whole even number in order to maintain consistency with a subdivision of the 4px baseline grid.

Naming Convention

Each step on the typographic scale is given a name based on font family, size, and weight. In keeping with the mission of our company to serve veterans, font sizes are referenced by military ranks to give a common language to designers when discussing hierarchy.

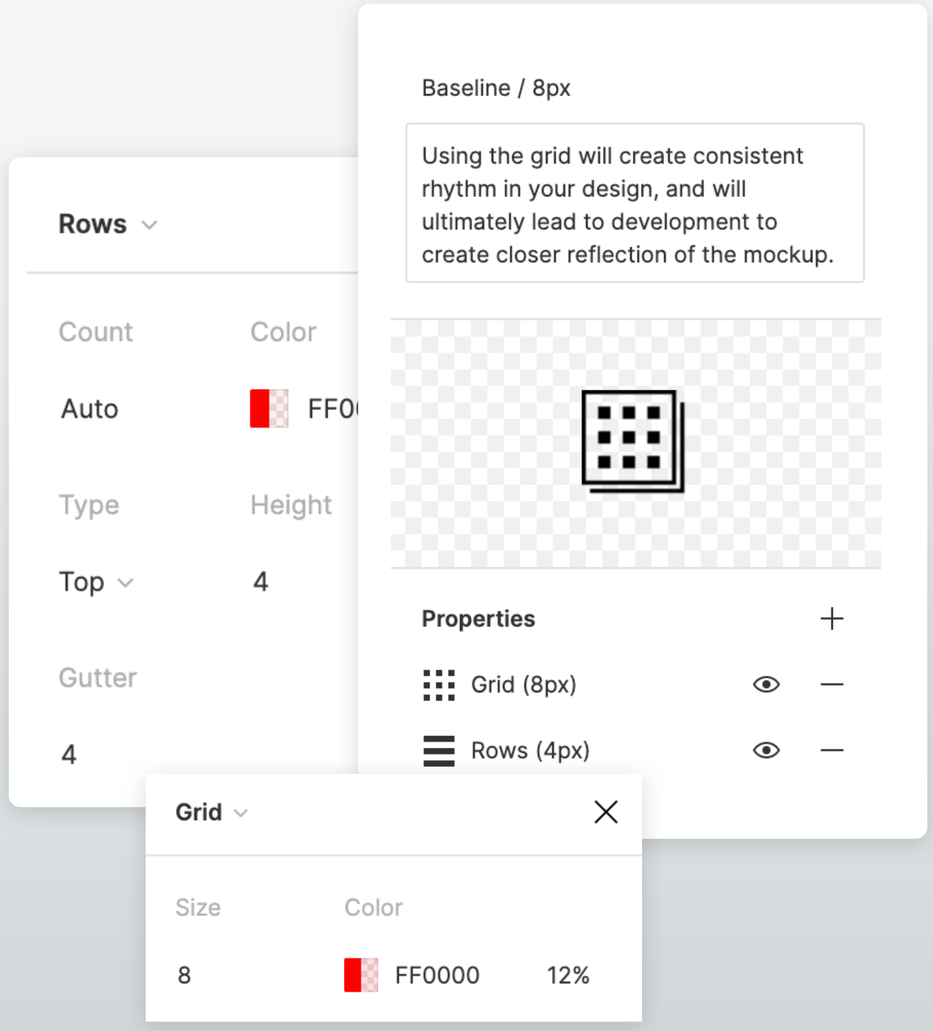
Smaller font sizes correspond with lower military ranks and as font size increases they are named by increasingly higher military rank. A bold and regular style are assigned to each step, or rank, of the type scale.

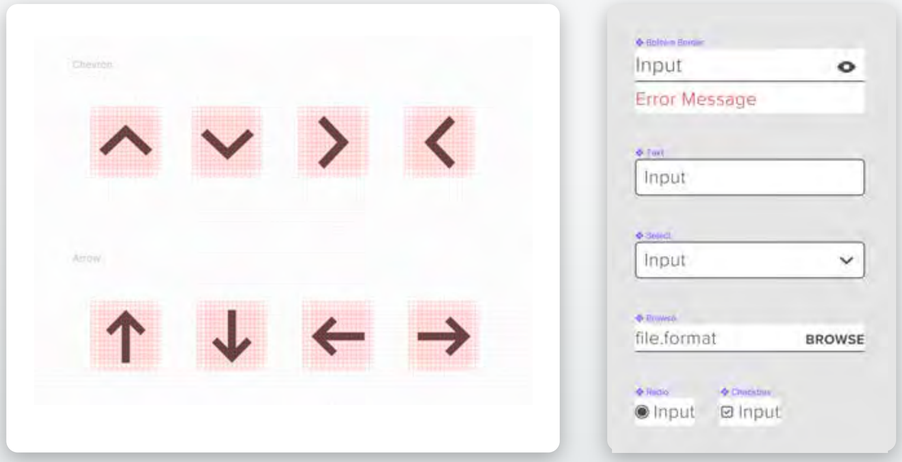
In Figma the text styles were named as: "Family Rank / Size(px) : Weight". This allowed us to see the name, have the pixel size for quick reference, and easily toggle between regular and bold weights.

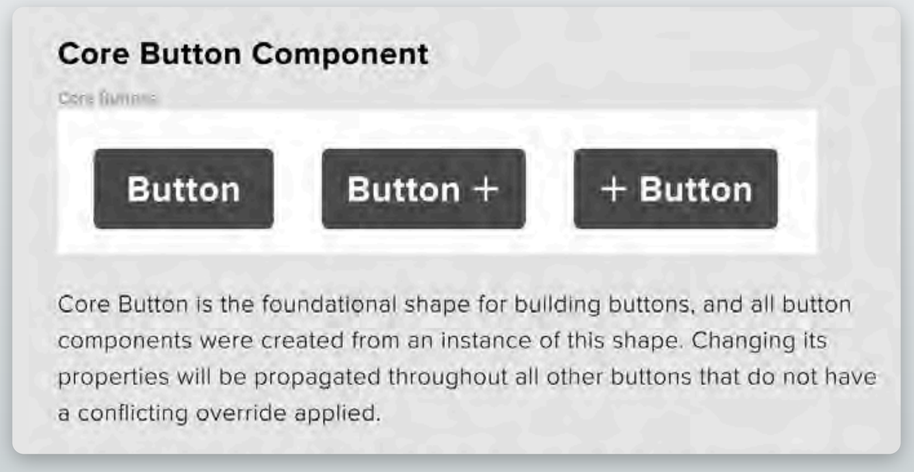
The 8pt Grid System

Our grid system is largely built upon the 8pt grid system made popular by Material Design amongst many others. We use this in conjunction with 4pt vertical baseline grid for typography allowing for more flexibility in line-height and paragraph spacing.

Figma made this very easy to recreate by allowing us to set up a grid style in our Grids library, which we could then consume in every project.







Core Components

Once our color, text, and grid styles had been set up, we could start in on building out our core components library.

In order to facilitate Component Driven Development (CDD). We started with our most essential icons and branched out into buttons, badges, inputs, toasts, etc.

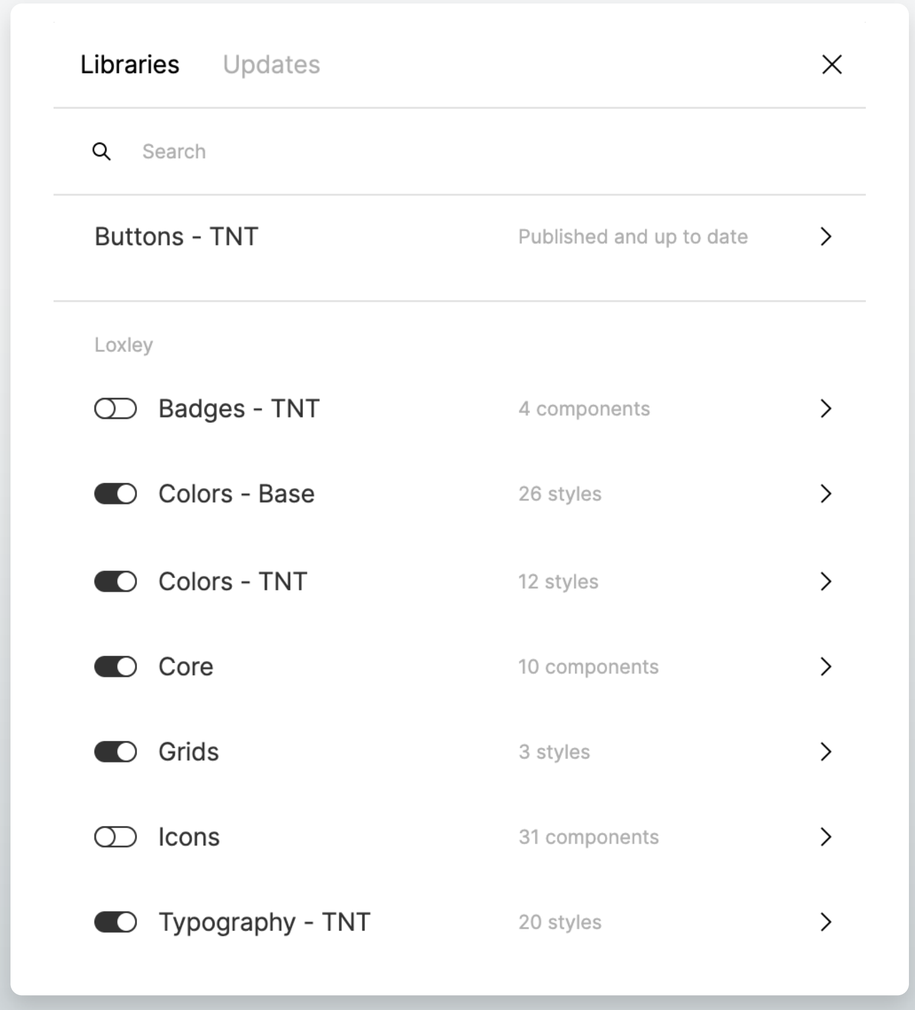
While they don't look flashy, they serve as the foundation for all of our individual components. All of our of libraries could use an instance of the their core component to build its branded components.

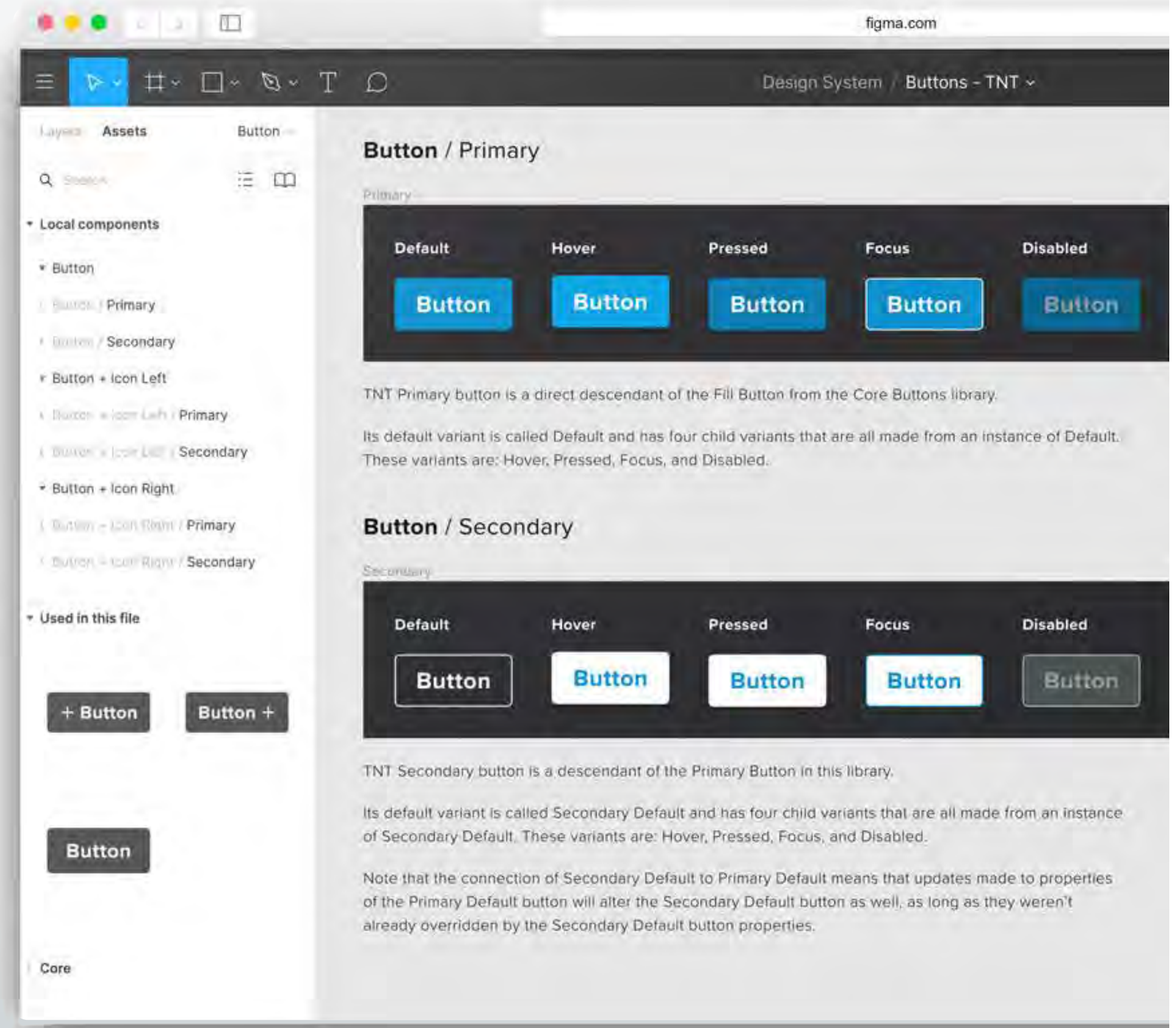
This means if anything was updated in the core, such as changing all button border radiuses from 4px to 8px, it would propagate to all of our buttons for every company!

Activating the Core

With the foundations of our core built, we could then make all core elements available in individual component libraries, allowing us to make modifications based on each company's branding.

The example shown is from our TNT branded buttons. This component library has access to TNT's typography and color libraries, as well as the base colors, grids, icons, and core components!





Bringing the Core to Life

After activating the core libraries, the real fun could start. Based from core components we created our primary components for each brand.

From the Primary component we could create all our variants, while keeping the variants attached to their parent component and the parent component attached to the core. It sounds like Inception, but in practice it was intuitive and allowed to iterate variants at a rapid pace.

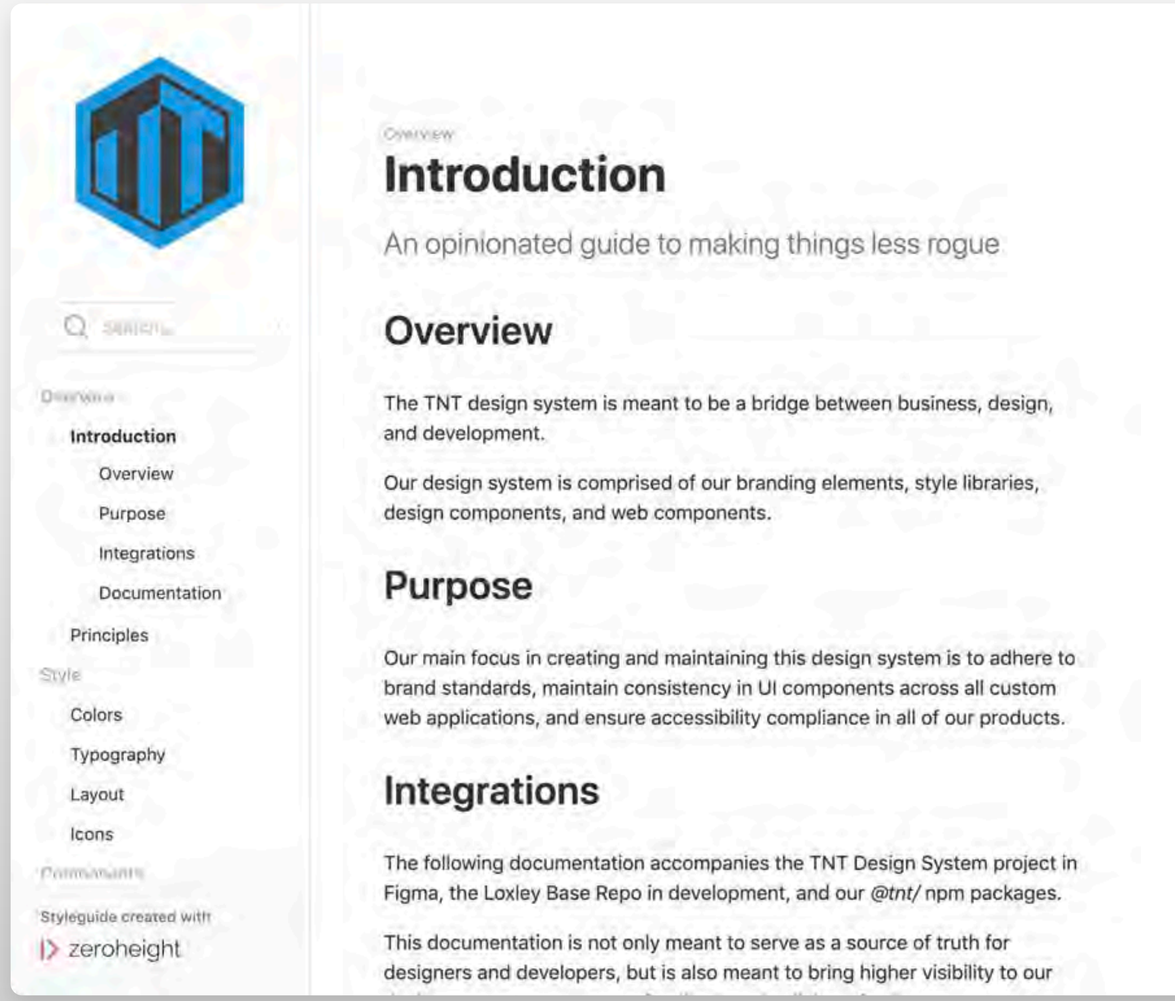
We embraced organizing our components the Figma way, using pages and frames instead of slash notation. This kept all of our layers clear and concise without losing the ability to toggle between instances of related components.

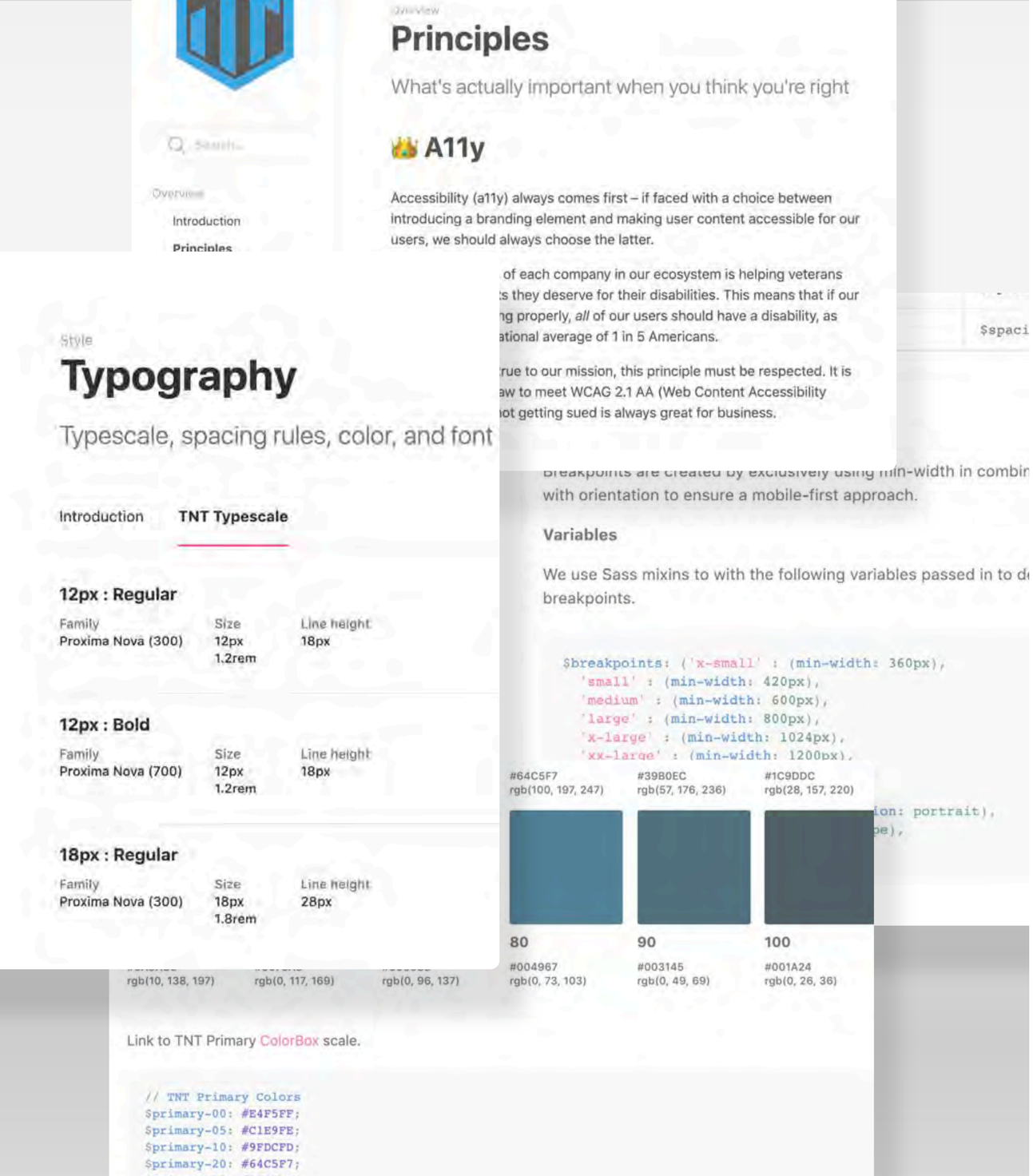
Living Documentation

A design system cannot be a true living system without proper and up-to-date documentation.

A developer on our team refers to this as “The Bus Factor”. If I were to have an unfortunate encounter with a bus tomorrow, would current or future team members be able to use and understand my work. To avoid the bus factor, we must have documentation.

We also needed to give some context to people outside of the department who would be interacting with the design system, such as marketing and product owners.





Zeroheight

After a lot of trial and error, Confluence graveyards, and some persistent research, we found a platform called Zeroheight that seemed to be an answer to all of our documentation woes.

Zeroheight automatically pulled in Figma components based on the file URL, so keeping figma components up to date was a simple click of the refresh button.

It also provides accompanying code snippets for any language, live previews of HTML/CSS/JS, and an integration with Storybook.

Using the platform we were able to add documentation extremely quickly and easily, as we built out our libraries and developed out products.

TNT Design System

Project Takeaways & Next Steps

Takeaways

Moving our design libraries from Sketch+Abstract to Figma ended up turning into so much more than a simple change of design platforms and library migration.

By challenging ourselves to rethink each principle and component of our design system, we ended up creating a much more robust product that could scale at the rate we needed.

Making our documentation more visible and ensuring that it could stay up to date with ease was another major win for the team.

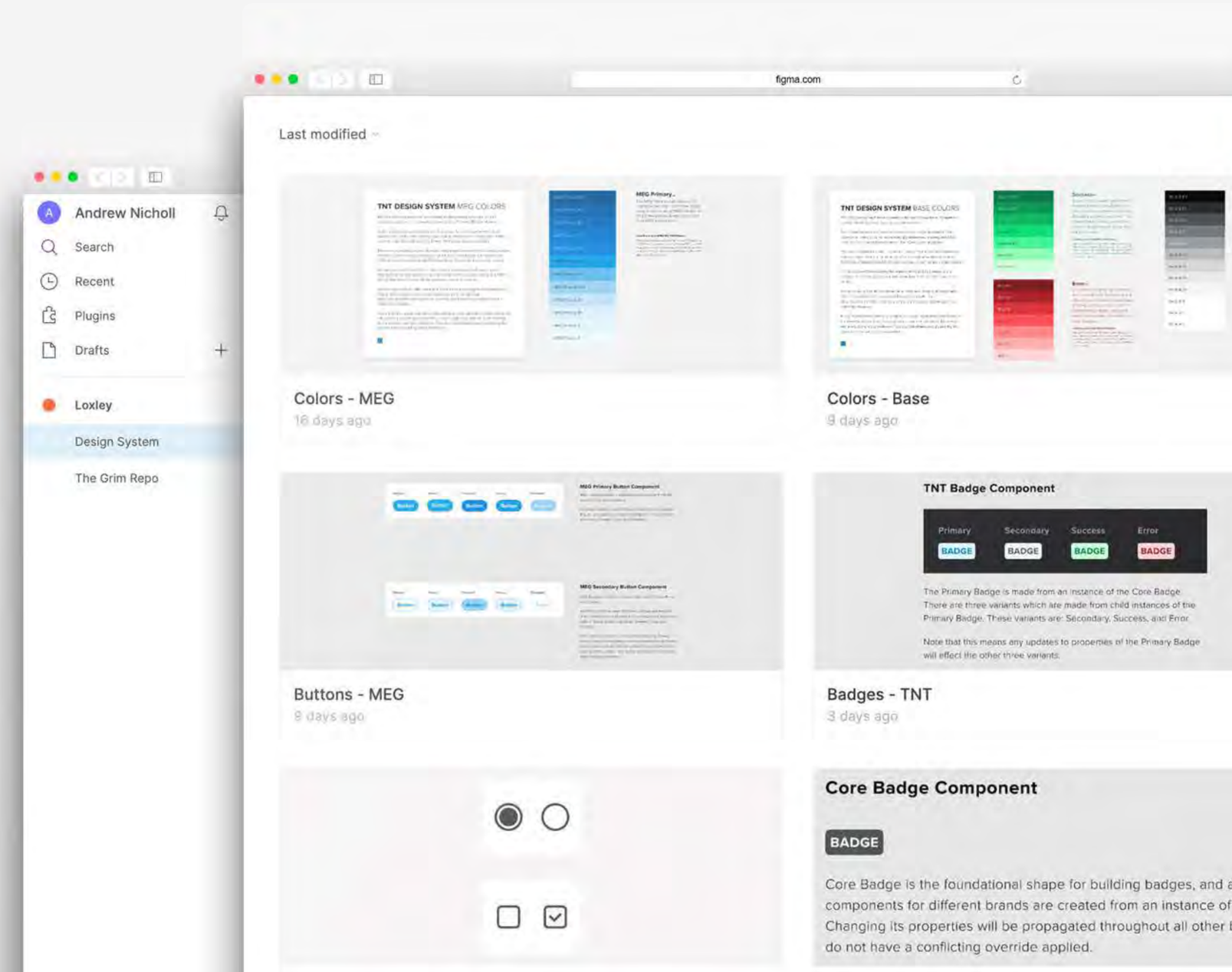
Teams outside of development can now see the core of our applications, become engaged with the system, contribute, and make it better.

Next Steps

Our next big goal is to finalize the the development of our design components.

The system is always growing and we are working on the most efficient way for development to stay up to date with design.

We are currently building out POC's using Bit and Storybook in combination with Angular components.



Da Vinci

Medical Survey

The following case study was assembled by 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.

Da Vinci Case Study

Summary

Da Vinci is a web based application designed to help military veterans obtain disability benefits and compensation by [redacted] through a comprehensive medical questionnaire.

Duration

3 months (6 sprints)

Team

Andrew Product Developer
Luke Lead Software Engineer
Allen Software Engineer
Jimbo Product Developer
Hugo Product Developer
Stephen Fearless Leader
Seth Software Engineer
BettyJo Software Engineer
Arthur Product Manager
Lane Software Engineer
Keiran Software Engineer
Ben Software Engineer

Design Approach

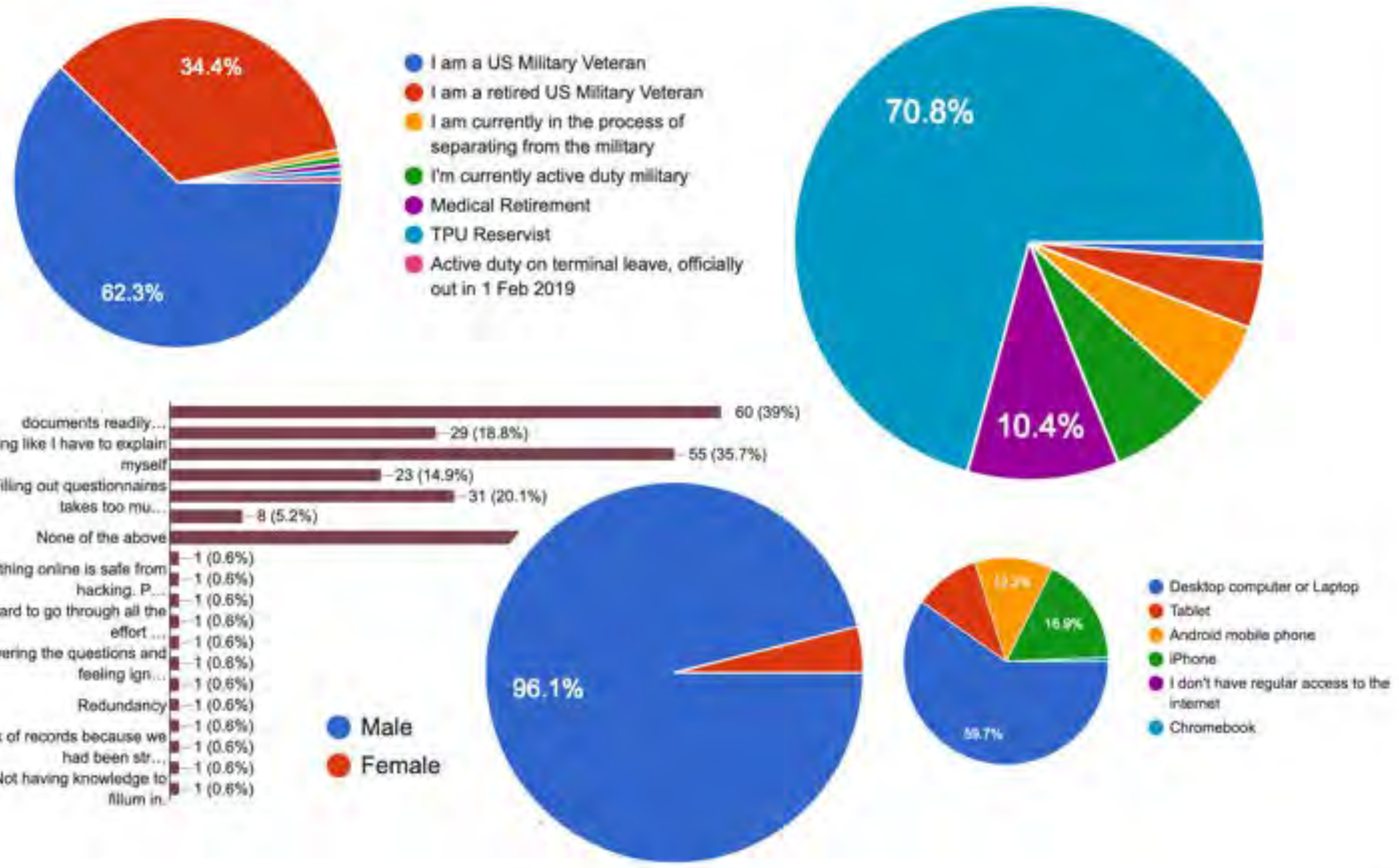
- User Research
- Market Research
- User Personas
- Process Flows
- Use cases/Scenarios
- Wireframes
- Usability Testing
- Desirability Testing
- Rapid Prototyping
- Back-end Development
- Front-end Development

Problem Statement

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.

Research Phase

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 and received key insights that would help us understand the problem from the user's perspective.



Quantitative Data Results

96% of respondents are male.
71% of users are age 55 and older.
59% of respondents use a desktop computer to access the internet.
30% of respondents use their premium e-benefits account to access their VA Ratings breakdown.
30% of respondents were not sure how to find their VA Ratings breakdown.
28% of respondents say they don't have any service connected disabilities.
24% of users access their VA ratings breakdown via physical records.
67.5% of respondents said they were interested in using the Da Vinci concept.
79.9% of respondents said they can accurately define the symptoms they're experiencing.
76.6% say they can accurately recall the medications they're currently taking.

Qualitative Data Results

36.4% are willing to dedicate up to 30 minutes to fill out a questionnaire.
31.8% are comfortable submitting personal medical information to an online questionnaire.
44.2% say they'd have to know more about the company to be comfortable submitting personal medical information to an online questionnaire.
16.2% say they'd feel uncomfortab;e discussing personal medical information to anyone other than their personal doctor.
37.6% feel they are not knowledgable about their current service connected disabilities.
34% feel they are knowledgable about their current service connected disabilities.
39% say the most frustrating aspect of the process is not having medical documentation readily available.

User Personas

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.

During our persona work, we discovered a resource called “Veterans Affairs UX Guide” made specifically for product teams designing digital products for military veterans which allowed us to leverage a vast library of data which helped us immensely in our process.

Vietnam Vince Primary

“ I rather deal with a live person than a computer screen, I am atleast keyboard literate.”

Gender: Male

Age: 55

Marital Status: Unmarried

Military Status: Retired Veteran

Patient Conditions: High Blood Pressure, Depression and PTSD, Diabetes (Agent Orange Exposure)

Closest Relationships:

Grand children

Girlfriend

Other Veterans

Characteristics: Cautious, Opinionated, Loyal

Technology Devices:

Iphone (Smartphone)

PC (Home Desktop)

Landline

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

VA Services Utilized:

I want to be in control of my medications

Make sure somone responds to me (Secure messaging and appoitments)

Let me see all of my health records

My Technology Pain Points:

Don’t feel that apps and websites are secure

Problems with sign in

Too much information on VA websites

Fed Up Fred Secondary

“I’m fed up with trying to make progress with the VA. I just want to be there for my family and feel like myself again.”

Gender: Male

Age: 35

Marital Status: Married

Military Status: Veteran

Patient Conditions: PTSD, Anxiety, Substance abuse management (Alcohoholism)

Closest Relationships:

Spouse

Children

Other Veterans

Characteristics: Tough, Unsatisfied, Loyal

Technology Devices:

Personal Laptop

Xbox One X (Gaming System)

Samsung Galaxy

iPad (Tablet)

VA Services Utilized:

VA Mental Health Counseling

eBenefits Premium account

Veteran Crisis Line (text messaging)

My needs:

Let me see all of my health records

Help me care for myself

Make sure someone responds to me

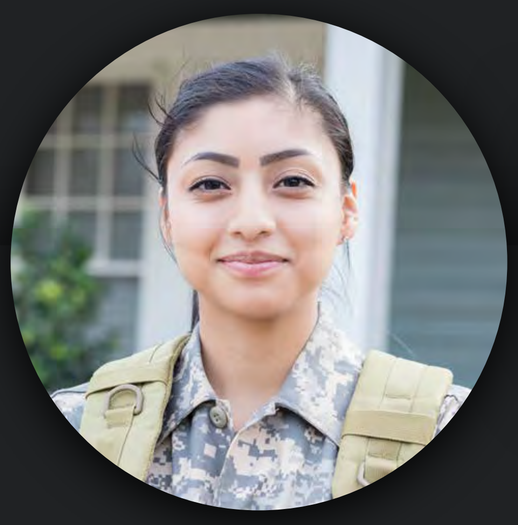
Help me connect with other Veterans with the same health issues

Pain Points:

Trying to use websites that are not optomized for mobile.

Constraints:

Doesn't have enough time outside of work to balance family life and also focus on the VA disability process.



Social Samantha Tertiary

"I want to be a career person and I want to accomplish things and feel like I'm contributing to society, my community, and my family."

Gender: Female

Age: 35

Marital Status: Married

Military Status: Veteran

Patient Conditions: Depression, Anxiety, Traumatic Brain Injury

Closest Relationships:

Spouse and Children

Close friend

Parents

Worries & Struggles:

Constraints and Principle

Technology Devices:

Iphone (Smartphone health & Productivity sapps)
Macbook (Personal Laptop)
iPad Mini
Fitbit Charge (Fitness Tracker)

VA Services Utilized:

VA Mental health counseling services

eBenefits

PTSD Coach

Veteran Crisis line

Pain Points:

VA websites are not optimized for mobile use

Can't upload my personal health data to the VA

Wifi is not always available at VHA facilities

User Needs:

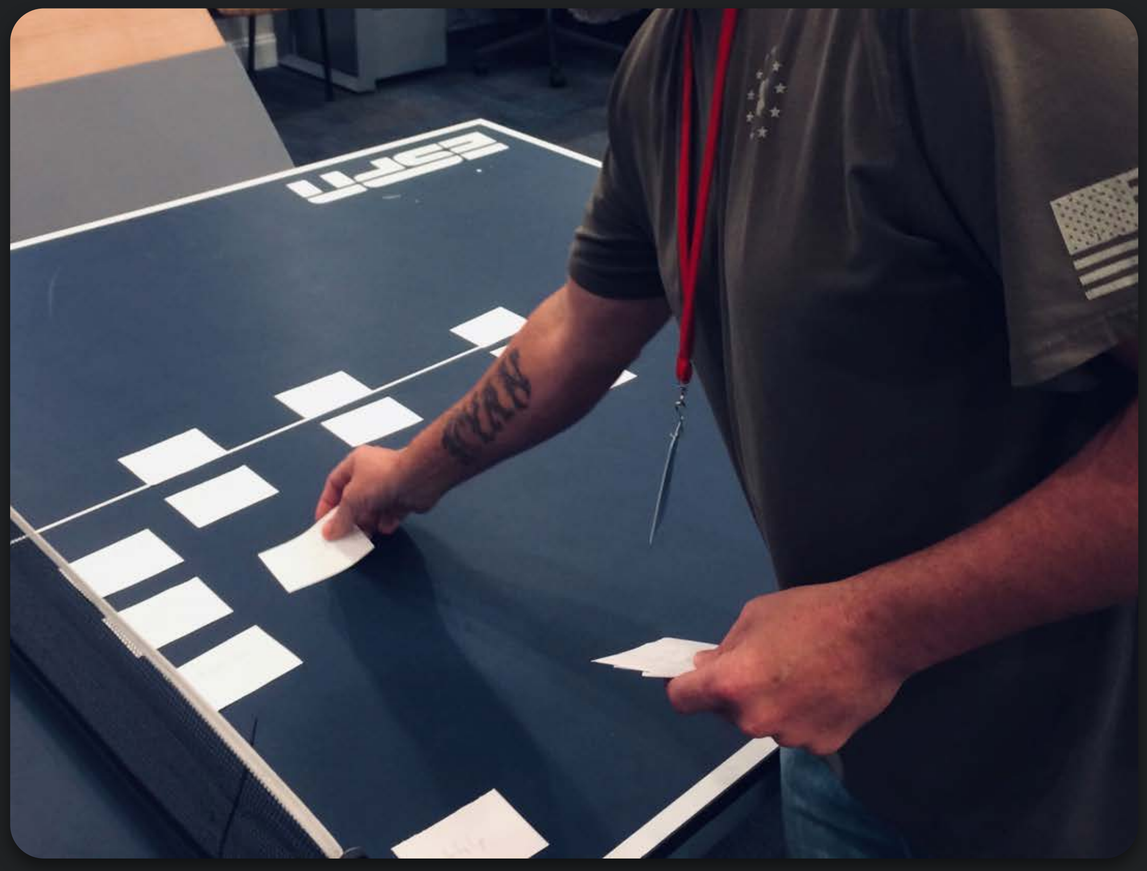
Time management

Make it easy for me to send my provider information

coordination with neurology and psychiatry

Information Architecture

At a high-level, information architecture “focuses on organizing, structuring, and labeling content in an effective and sustainable way.” (Usability.gov) Our goal was to dive into this part of our design process in order to create a logical structure that would bring meaning to the site’s content. In order to accomplish this goal we carried out severeal different processes led by industry rules of engagement that dictates how users would engage with our content.

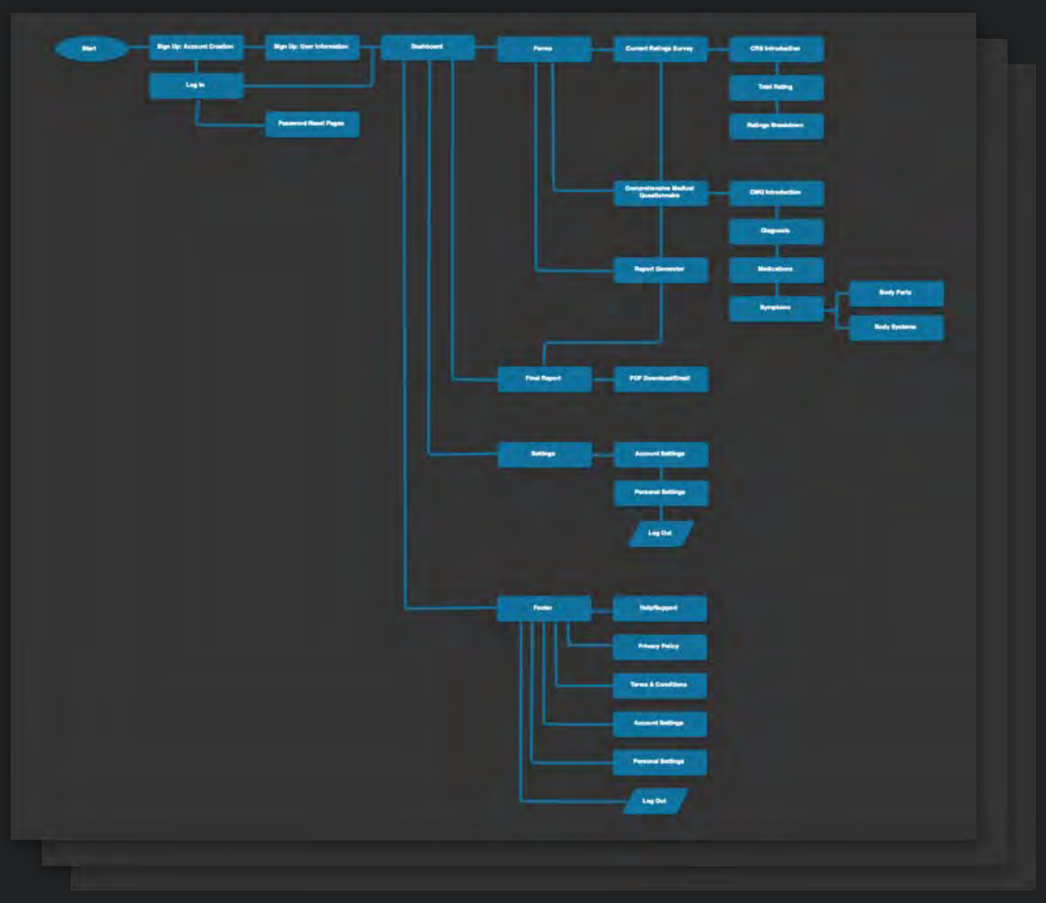


Card Sorting

With the help of some fellow employees we decided to conduct a brief card sorting to learn how users would expect to see content organized on our site. We then utilized the feedback results to construct the information architecture of the platform in a way that resonated with the mental model of our user.

Site map and User Flow

In preparation for our first concept sketch, we generated a user flow diagram via Sketch.io referencing all user scenarios and requirments. Our objective was to map out each user requirement in the app process ensuring that each task was achievable in the shortest amount of time and with the least amount of steps. After several rounds of improvements we were ready to then begin putting these diagrams into our first wireframe.



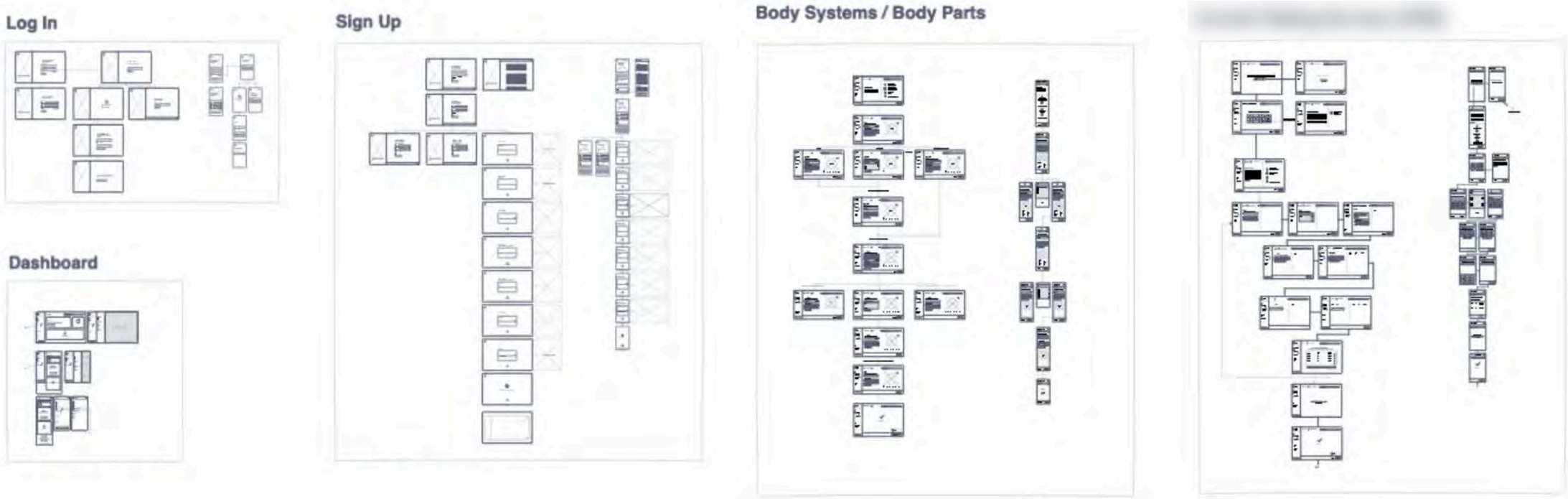
User Cases and Scenarios

After our card sorting session, we wrote in-depth case scenarios for each user persona detailing all of the possible use cases and scenarios. For instance, a majority of users are 55 and older, so we made sure our product was easily accessible, intuitive and easy to use. This crucial step in our design process allowed us to prioritize our features and eliminate all possible confusion that may arise.

Epic	Story	As a (role)..	I want (feature)...	Because (reason)	Scenario	Tasks
Auth Pages	Sign Up *Page 1*	Vietnam Vince	feel secure while creating my account	I want to know my potential disabilities	Vietnam Vince is at home on his desktop co	entering an
		Social Samantha	Be able to start this process now, even though I'r	I'm short on time at home and usually	Social Samantha is on the bus using Faceb	Redirecting
		Fed up Fred	Get started and get the answers he needs as soo	He is tired of lengthy and confusing pr	Fed Up Fred is on his lunch break at work u	Enter accou
		Fed up Fred	Get to the Sign Up page	because I'm on the Log In page but I d	Fed Up Fred accidentally ended up on the L	Click sign u
	Sign Up *Page 2*	Fed up Fred	Quit the sign up process	I've entered my email and password b	Fed Up Fred has completed the first page of	First page t
		Fed up Fred	Feel safe finishing the sign up process	More sensititive personal info is requir	Fed Up Fred completed page 1/2 of the Sign	Entering inf
		Vietnam Vince	Feel secure finishing the sign up process	More sensititive personal info is requir	Vietnam Vince completed page 1/2 of the Si	Entering inf
	Testimonials	Fed up Fred	Know that this service has helped other veterans	I don't want to waste my time or risk m	Fed Up Fred lands on the Sign Up page and	Navigate to
	Log In	Vietnam Vince	easily access my account from my computer	I need to check the status of my eligibi	Vietnam Vince is at home on his desktop computer and v	
		Social Samantha	Log into my account easily and securly from my p	I want to create an account to keep my	Social Samantha is waiting on her children a	Enter in use
		Fed up Fred	Get back into my account quickly	I is feeling motivated at home and war	Fed Up Fred has some time at home to finis	Enter usern
		Fed up Fred	Get to the Log In page	because I'm seeing the Sign Up page	Fed Up Fred accidently landed on the Sign U	Click log in
		Fed up Fred	Retrieve my password	I have entered it incorrectly multiple tin	Fed up Fred is trying to log back in to Da Vir	Click forgot
	Terms of Service	Fed up Fred	To know upfront what information I am required t	I want the security and certainty of knc	Fed up Fred is registering on our site for the	Click Terms
Dashboard	Starting Forms	Vietnam Vince	To get the process started and get my potential r	I need to provide my personal medical	Vietnam Vince has logged in for the first time	Navigate to
		Fed up Fred	Know where to start when I log in for the first time	Every website is different and I want to	Fed Up Fred is on his laptop and arrives at t	Navigate to
	Revisiting Forms	Social Samantha	I already been through the CRS however I need	I dont want mis-information to be subm	After logging back in Social Samantha reme	From the da
		Fed Up Fred	See all my ratings again	I want to make sure I didn't miss anyth	Fed Up Fred has completed the CRS. He wa	Navigate to
	Account Settings	Vietnam Vince	update his contact information	recently purchased a new smart phone	He recently got rid of his flip phone purchase	Find Accou
		Social Samantha	Update mailing address	I recently moved and want to update m	Social Samantha recently moved into a new	Find setting

Low Resolution Wireframes

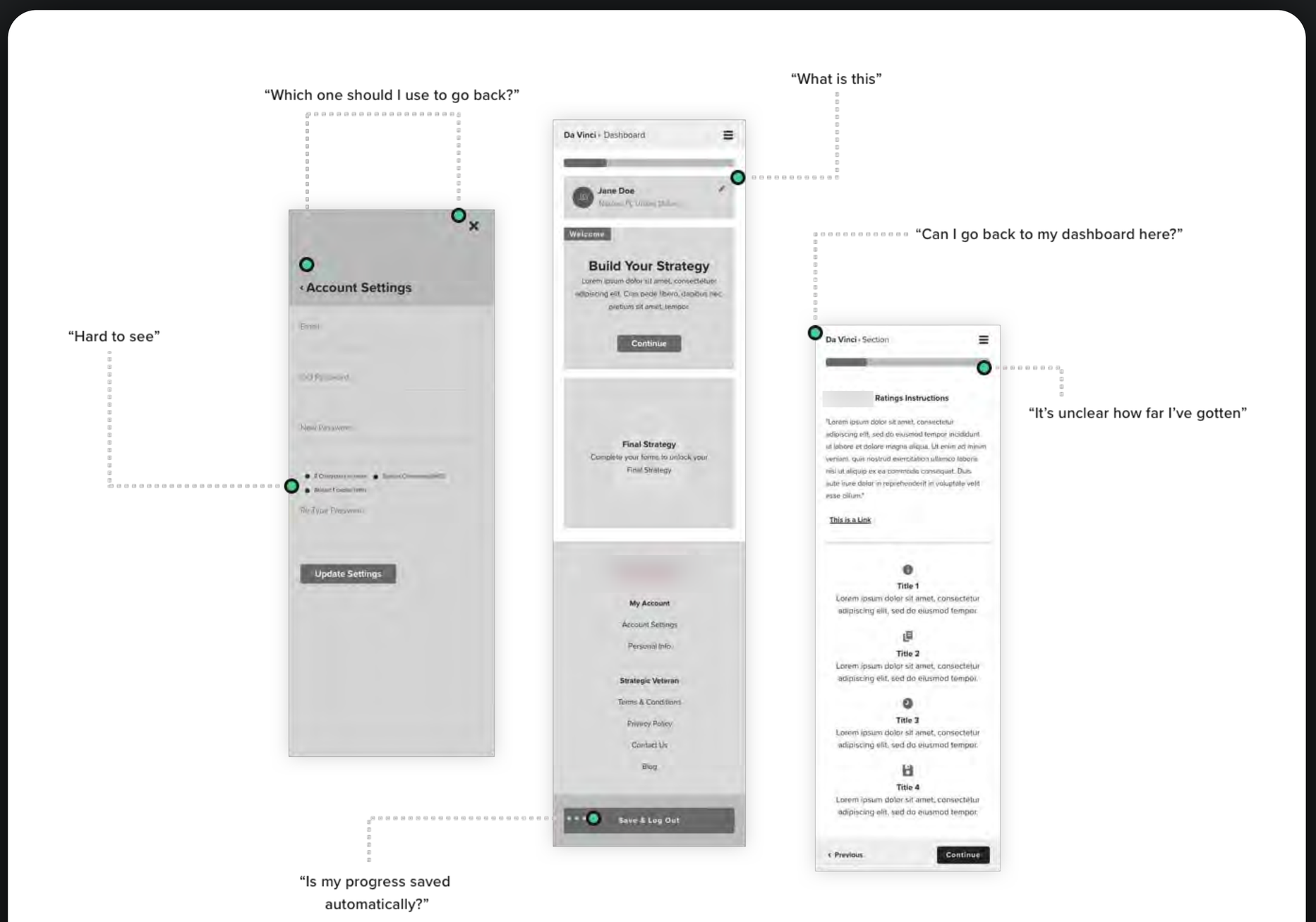
We began creating the general layout of the application based on specific requirements and must-haves that were imperative for the first version of Da Vinci. We collaborated on Invision's Freehand for the first round of concept iterations and made sure to follow Usability and human design centered principles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



Medium Resolution Wireframes

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.

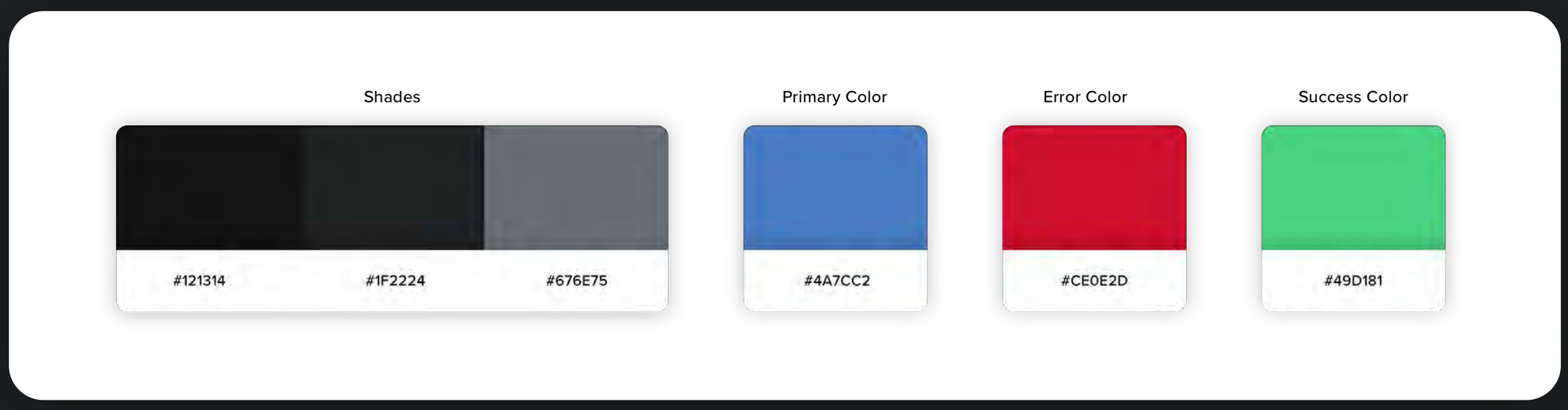


Color Theory and Visual Identity

In regards to the visual identity of Da Vinci, the goal was to find a balance between existing company branding and adhering to color theory guidelines and principles, in an effort to do so, we performed a desirability test to find the proper color scheme that would create the sense of dependability and security we hoped to evoke in users while interacting with our product.

We built our own library of styles including colors, icons and typography which was extremely helpful in staying organized and maintaining consistency throughout the application.

After exploring several options and getting feedback from users, stakeholders and our in-house marketing team we decided to proceed with the following color scheme.



Prototype

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 and function 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.

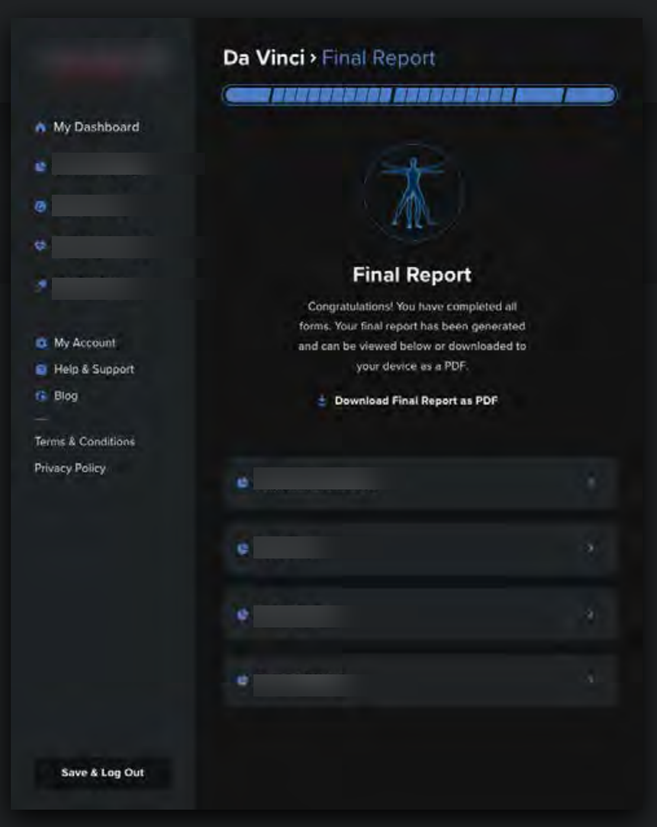
Design Outcome

Easy to use questionnaire

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

Propriety Algorithm

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



Generate Final Report

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

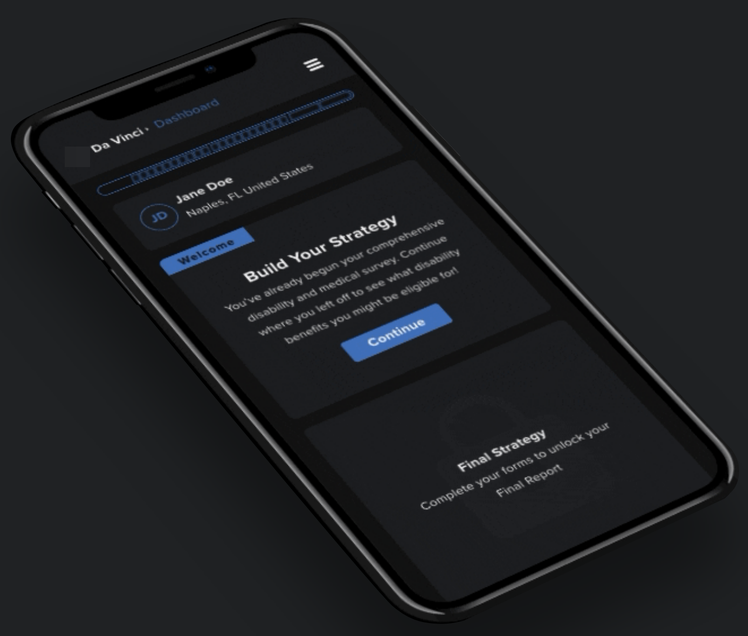
Accessibility

Upon completion of the first fully functional version of Da Vinci, we decided to begin the accessibility portion of our design process. Prior to this decision, we realized how imperative it is to our users that our product meets the highest accesssibility standards, especially being that most if not all of them have some type of physical disability that may or may not hinder their ability to use our product. With that being said, we decided to keep accessibility at the forefront of our design process moving forward with other projects.

We started by educating ourselves on the latest accessibility standards and best practices by taking courses on the subject and decided to document our own unique auditing process. This allowed us to establish consensus between everyone in the team including designers, developers, project managers and stakeholders which continues to allow us to track and ensure our progress as we continue to grow and create accessible products for our users.

Outcome

We carried out our accessibility auditing process on the Da Vinci site with the help of several third party plugins and software and performed our own manual analysis after realizing several of these tools have limitations that do not allow them to pick up on certain things that humans can. The primary objective in the way we constructed our audit was to ensure that our website was totally accessible in the highest standard for all types of disabilities; despite dramatic improvements to our site we are still working hard to achieve that goal.



Next Steps in our Design Process

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 user experience for our users.