Design System (WeLive)

A Design system is a language that builds trust, transparency, and speed in design/development process.

Role - Lead Designer / Platform - iOS/android

The Welive design system

When I first joined the WeLive team in 2017, there was an MVP app that was live (that I had worked on the year before). The existing app had been made so hurriedly, that there was no design system or consistency in code or components. It was clear the WeLive App needed consistency, and the creation of our team in 2017 allowed us to start over with the codebase and design system. So I embarked on a process of approvals from the Brand Team to establish colors, type, and other basic elements so that I could hit the ground running.

Color and typography

The atoms of any design system are color and typography. What I had to start with was a single reddish color and a set of grays, along with system fonts.

Color was something that came from WeLive's print design standards. But for technology, you need a range of a single color for interactions. So I took the reddish color and used an HSL scale to darken and lighten the color systematically. Then, realizing the reddish color looked amazing in print, but literally burned your eyes on a digital device, I went looking for a color to balance the reddish color.

Inspiration came from the space. I lived at WeLive prior to joining the team, and I had furniture that came in a variety of colors. Trying to replicate the colors on the screen, I went through hundreds of swatches and arrived at a green color. It balanced well with the reddish color, and after using the HSL scale, I came to a solution that lead me to making it the primary color in the app. The reddish color would remain as an accent or for areas of pop.

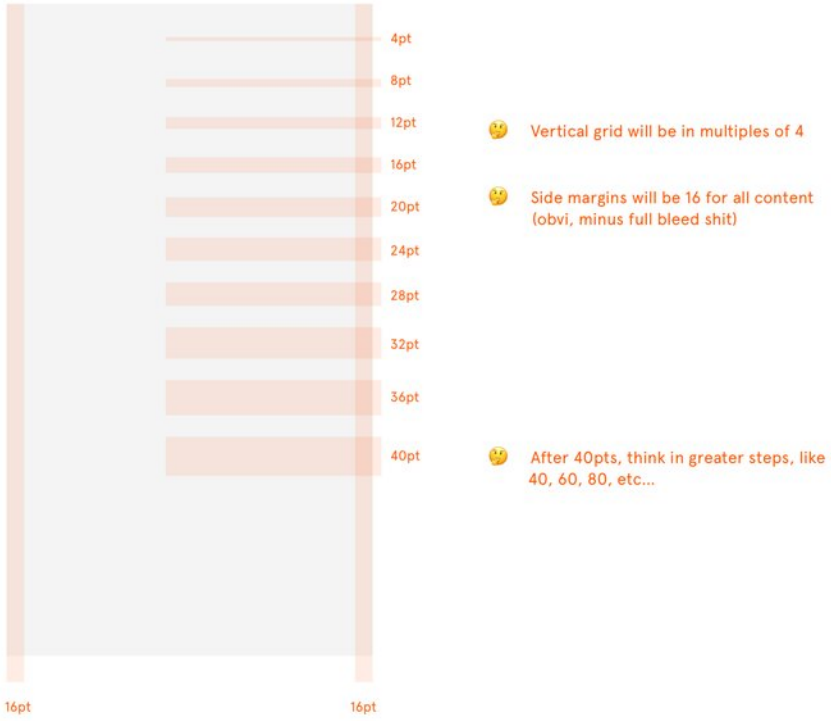
Then for the typography, I initially didn't intend on changing the typefaces. System fonts are completely fine. But after talks with Brand, there was a push to unify the brand of this app within the living space. And discussing with the developers, we planned on making a unified system between iOS and the Android app we were about to build. Both systems would use Apercu, and as we created components, or web views, the visual experience would be similar throughout the user journey.

Green50, welive50, and gray colors on the HSL scale.

Type scale for Apercu to be used arcoss iOS and Android

The grid

The grids for WeLive all worked on a 4pt/4dp grid. This allowed all the designs to fit most phone sizes available on the market, made math easy (yay math!), and cleaned the code and design spacing.

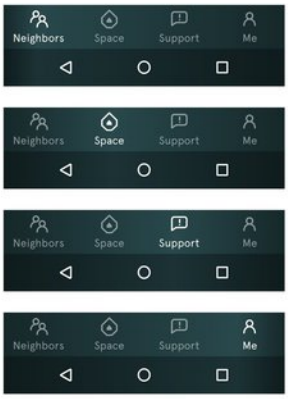


Navigation

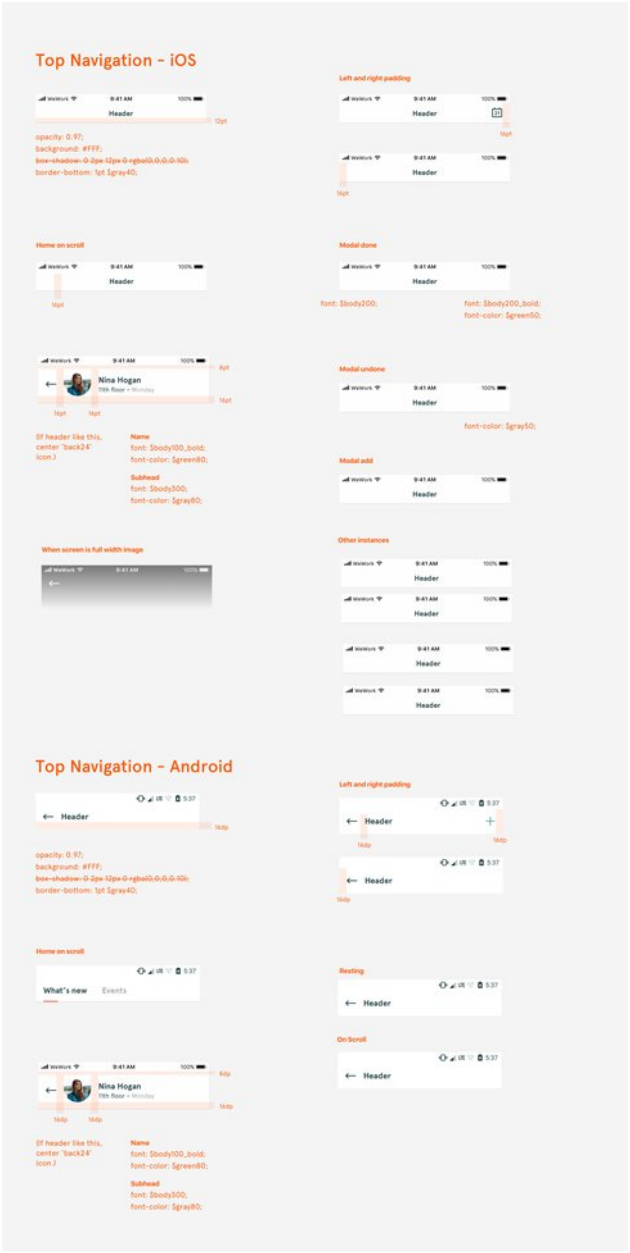
There are parts to a system that may never translate 100% from iOS to Android or vise versa. Although global design standards have brought patterns and trends closer together over time, these platforms are themselves their own beast. This is most clear when you look at navigation patterns.

Navigation

(Green glow highlights active tab)



(Green glow highlights active tab)

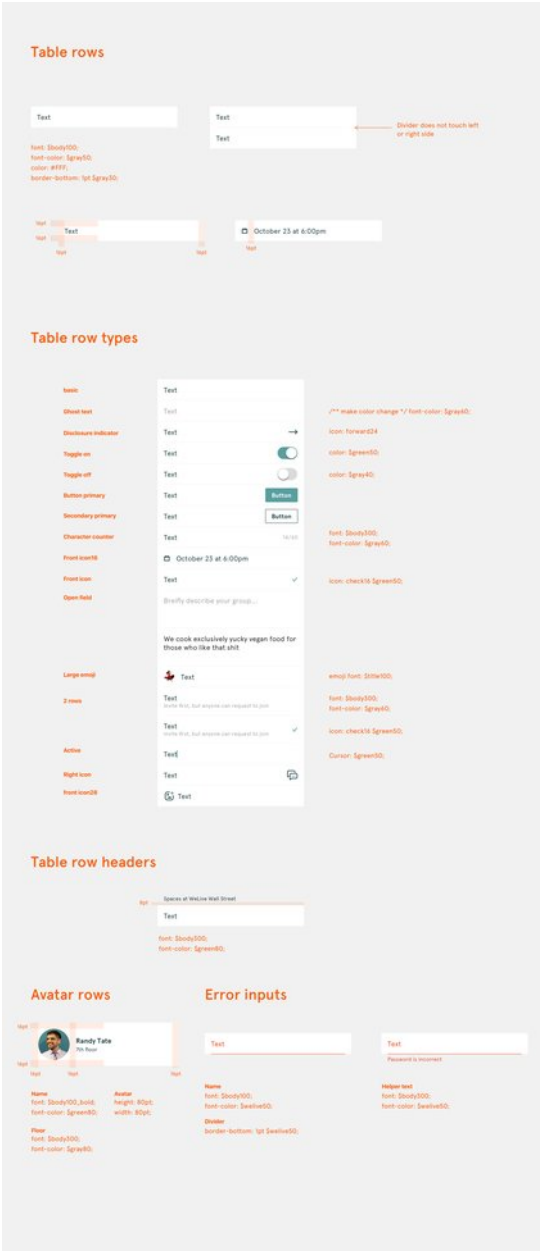


Icons

For this design system, I created a set of icons based on some of the older icons on the original WeLive app, but refreshed to fit the vibe of the current styles I had been experimenting on. Additionally, building them on the 4pt grid everything was built on.

Cells and table rows

One thing the developers and I agreed on doing was to keep components as consistent as possible between platforms. Since we were using Apercu and the same icon set for both platforms, and the same 4pt grid, most components could be identical.

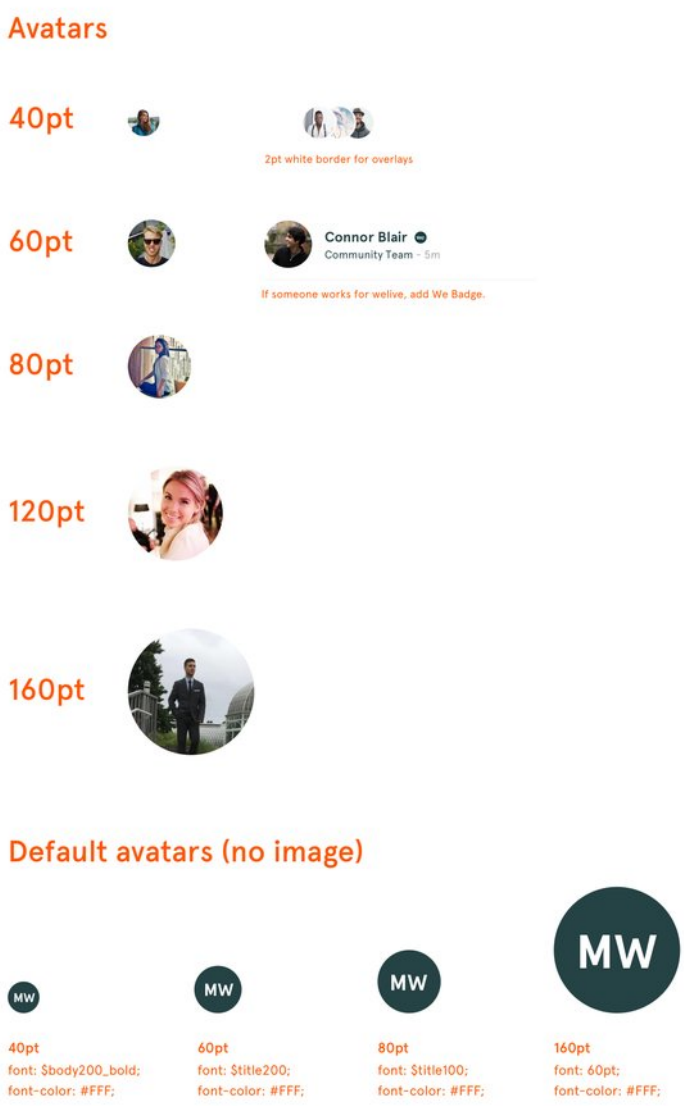


Buttons

Buttons are critical for any product. I created two sizes, a primary-tertiary set, as well as a couple of rules surrounding button usage. A button loader was another important part, and I created an animation using Haiku exported to Lottie. You can see the loader in actionhere.

Images

Controlling how images are shown on your app helps the predictability and recognition of ui elements. If there is a cell with a 40pt avatar image on one page, then it better act or display similar information on another page.



final outcome

As you probably noticed, there is a common language formed in all these examples. A way for designers to talk to developers, and a way to keep design hand-off clean and clear. Ideally, you wouldn't need to 'spec' every feature you hand off. That's what Zeplin and Figma are good at. This system was made before our team had fully transitioned to Zeplin.