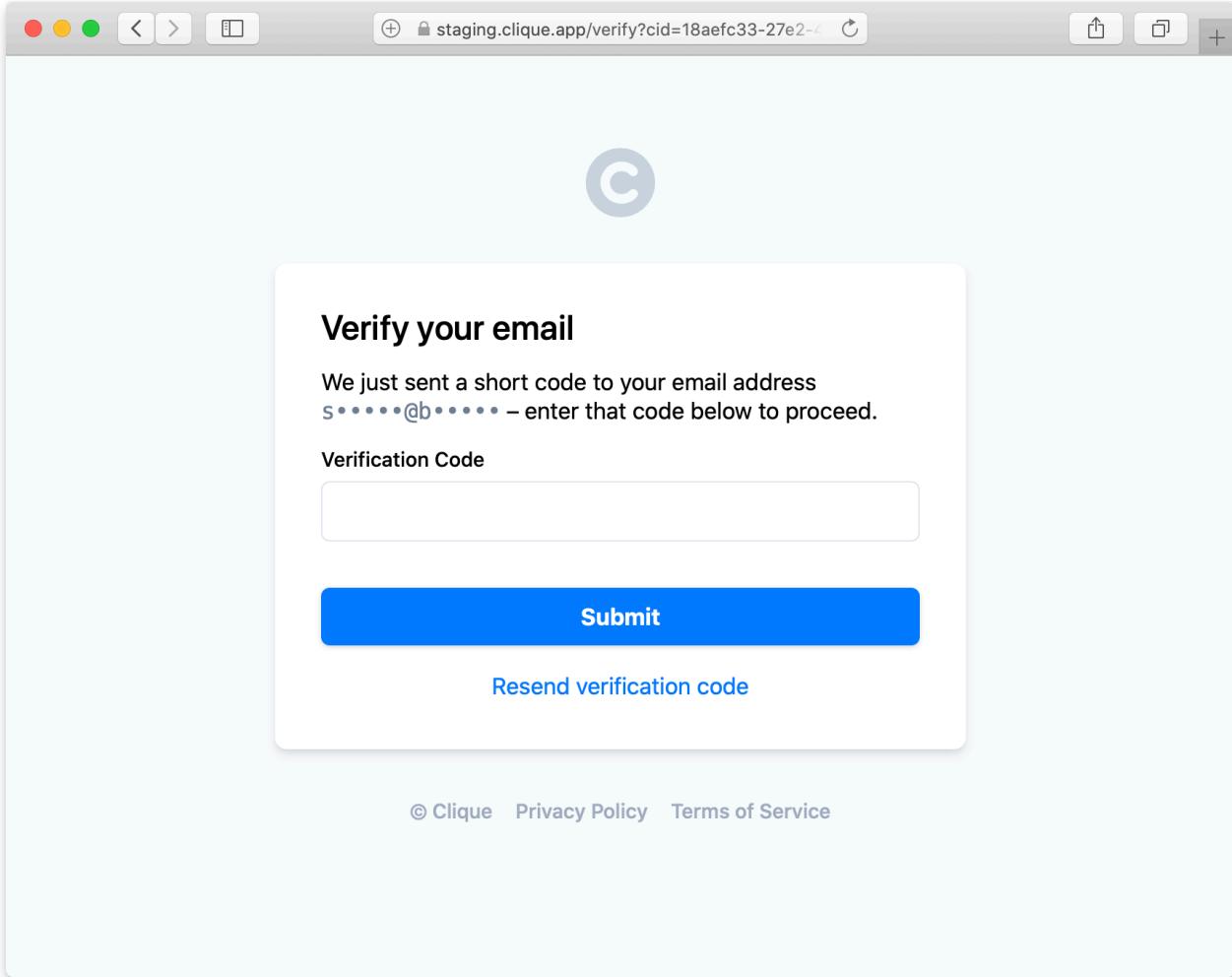


Login screen for Clique, a private social network.

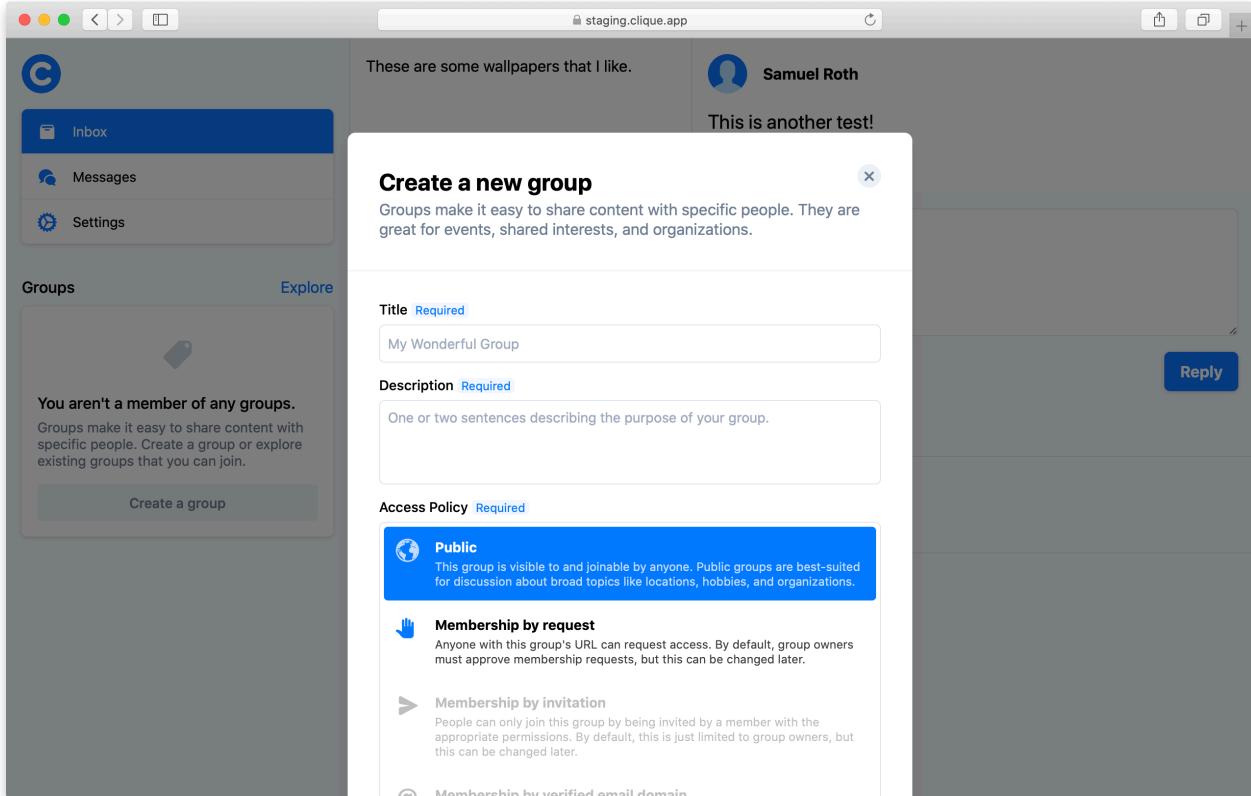
Building Compose, a design system for Vita services

This and other screens from Clique are built using an internal framework called Compose. Being a solo founder and developer, I found myself spending a lot of time duplicating various interface elements across the many iterations of Clique's MVP. I've saved myself a substantial amount of time and effort by slowly extracting and refining components into a separate NPM package. Every component is strongly-typed, extensively tested, and built with accessibility in mind.

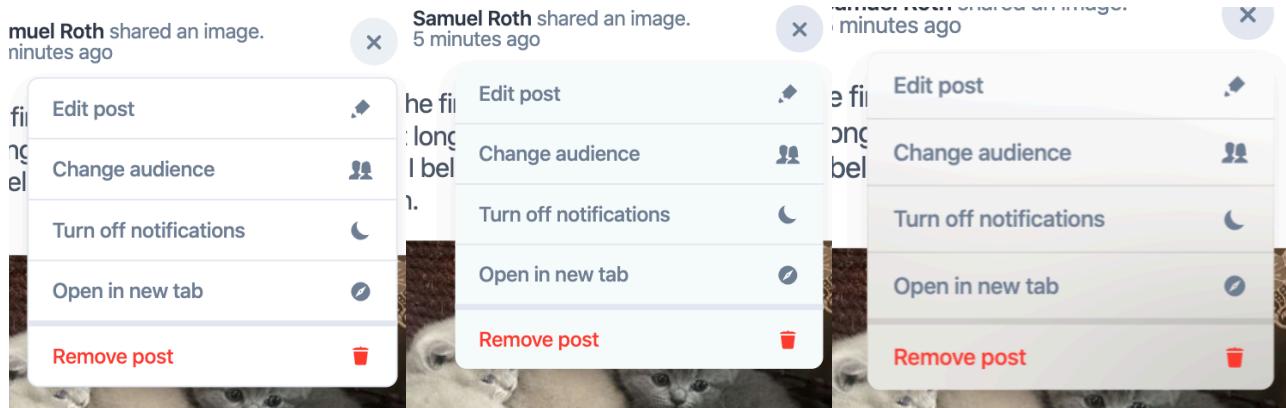
The web implementation of Compose is built with TypeScript and React, and I'm actively working on a SwiftUI port for Clique's iOS application.



Multi-factor authentication screen for Clique; supports email and SMS verification. Web Authn in progress.



Example modal screen, using the same Compose input elements from the authentication flow.



Experiments with popover menus.

The screenshot shows a web browser window with the URL ironlang.org/#types. The page title is "Types". On the left, there's a sidebar with links: "Introduction", "Iron at a Glance", "Types" (which is highlighted in red), "Functions", "Protocols", and "Modules". The main content area starts with a heading "At the foundation of the Iron type system is a set of *primitives* from which all other types are derived." Below this is a large red box containing the letter "Fe". Under "Booleans", it says: "The Boolean type has only two possible values: `true` or `false`. It works just like boolean types in other languages, used primarily in conditional expressions, which we will dig into later." Under "Numbers", it says: "Numeric types are available in many sizes, in both signed (-2^{N-1} - 1 to $2^N - 1$) and unsigned (0 to $2^N - 1$) varieties." A table follows:

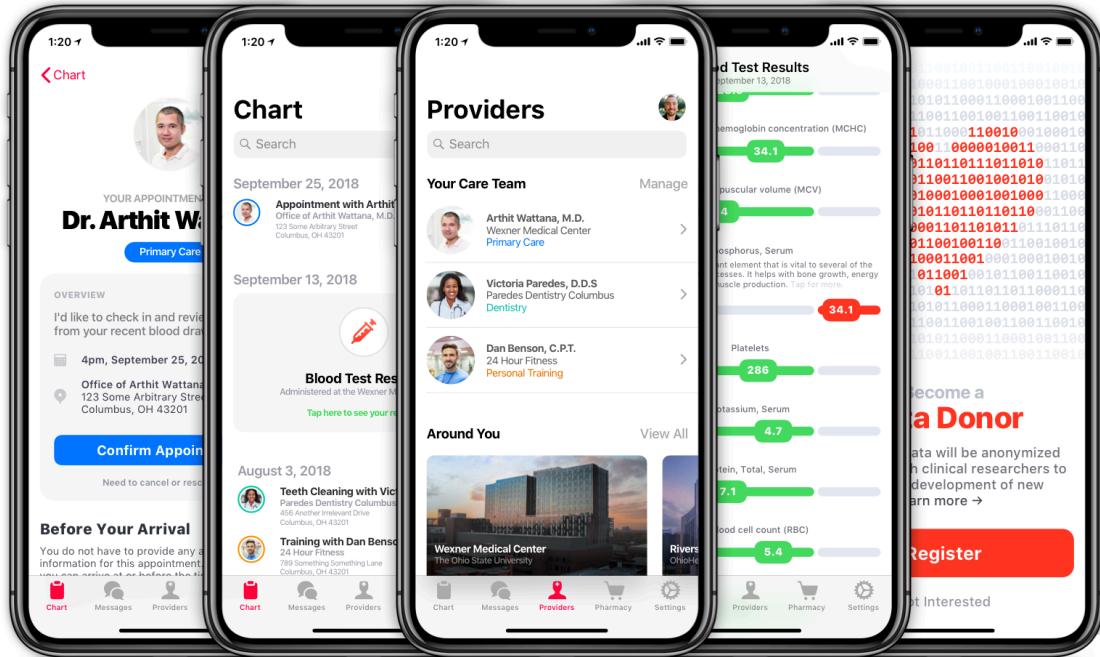
Size	Signed	Unsigned	Floating Point
8 bits	I8	U8	
16 bits	I16	U16	
32 bits	I32	U32	F32
64 bits	I64	U64	F64
128 bits	I128	U128	F128
Pointer size *	Int	Uint	

* Either 32 or 64 bits depending on the system architecture.

The website for the Iron programming language.

Testing

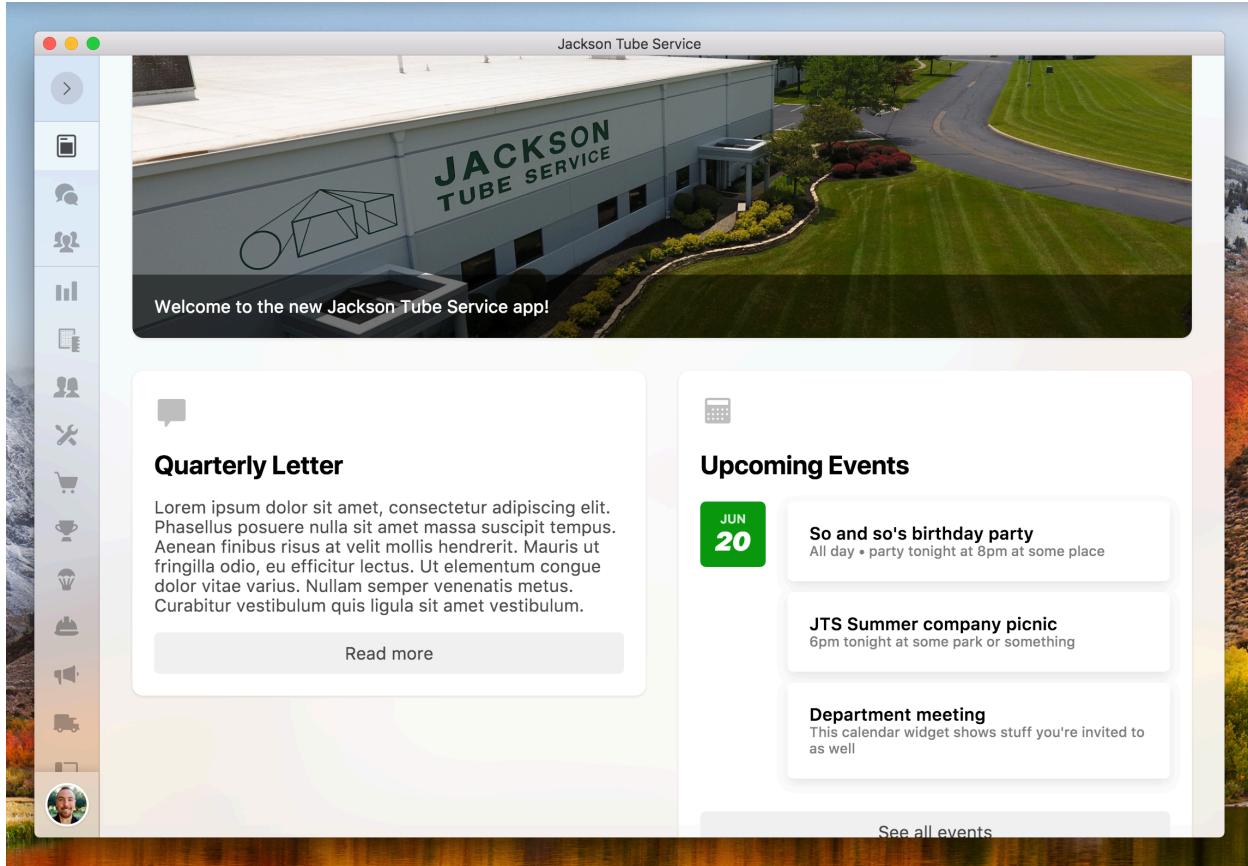
blah



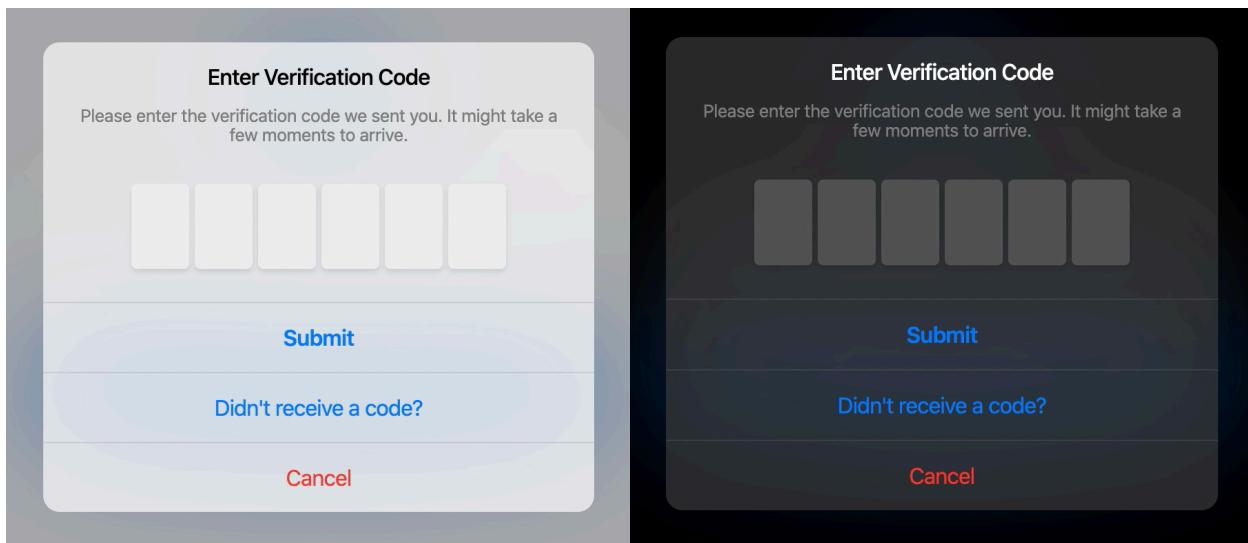
Screens from a separate project in the healthcare space.



Exploratory icons for that project's macOS application.



Proof of concept for an Electron desktop application for Jackson Tube Service, a steel tubing manufacturer.



Exploratory UI for multifactor code input (React + TypeScript).