

Selected projects 2018-2024

Bartosz Dębicki

mobile :+48 508 343 705
email :bartosz.debicki@zoin.pl

linkedin :[/in/bartoszdebicki](https://in/bartoszdebicki)
github :[/bdebicki](https://github.com/bdebicki)
behance :[/earldebicki](https://www.behance.net/earldebicki)

Frog Design System

Adverity's design system provides a comprehensive toolkit that includes tokens, components, and detailed guidelines. The design system has been used by over 25 engineers (frontend and full-stack) and 7 designers on a day-to-day basis.

My role was responsible for building a typescript and react-based library of components and tokens, analysing consumer feedback, and prioritising work for the team in short-term and long-term roadmaps.

Tokens

Design system based on an 8px grid with tokens describing almost every single piece of interface. This includes size, scale and spacing, animations, and borders. Additionally, it provides a harmonized color palette and defined typography styles.

2022-2024 | Adverity
role :product owner / code

```
export const colors: Colors = {
  brand: '#1839A0',
  brandHighlighted: '#2950E3',
  brandActive: '#4071F7',
  brandSecondary: '#F4F8FF',
  brandSecondaryHighlighted: '#E5EFF',
  brandSecondaryActive: '#D1E1FF',
  brandText: '#182447',
  brandAccent: '#00FFC5'}
```

Components

The Design System boasts a comprehensive library of over 70 components and hooks. It is provided in both React and Figma libraries. This extensive collection enables developers and designers to maintain consistency and efficiency whether they are building digital products or creating design prototypes.

Documentation

After analyzing the usage of our documentation and gathering consumer feedback, I have initiated and designed a new documentation page with improved architecture. We have implemented several minor visual enhancements and introduced a new navigation architecture and component page documentation architecture. We have received positive feedback from our users, who find the new documentation clearer and more effectively focused on the essential information they need.

Platform redesign

The team has had the opportunity to drive the platform redesign, which was divided into eight stages. In each stage, we aim to implement redesigned changes that affect specific product areas such as forms, CTA triggers, typography, and other interface elements. After completing the first two stages, we received feedback from our customers indicating that the application now appears more modern and that the navigation has improved significantly—users reported a reduction in confusion and easier navigation.

Adverity UI before redesign

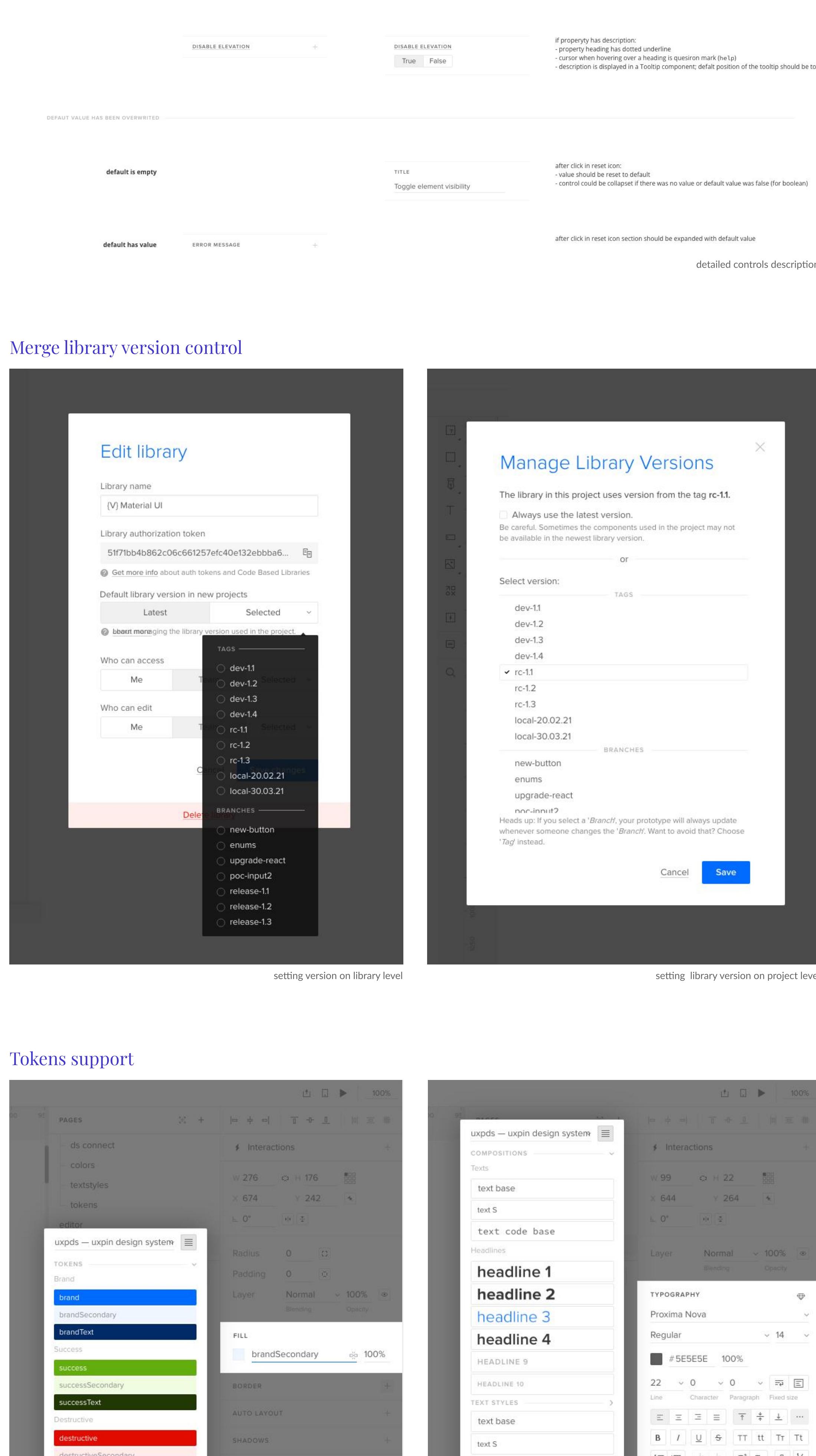
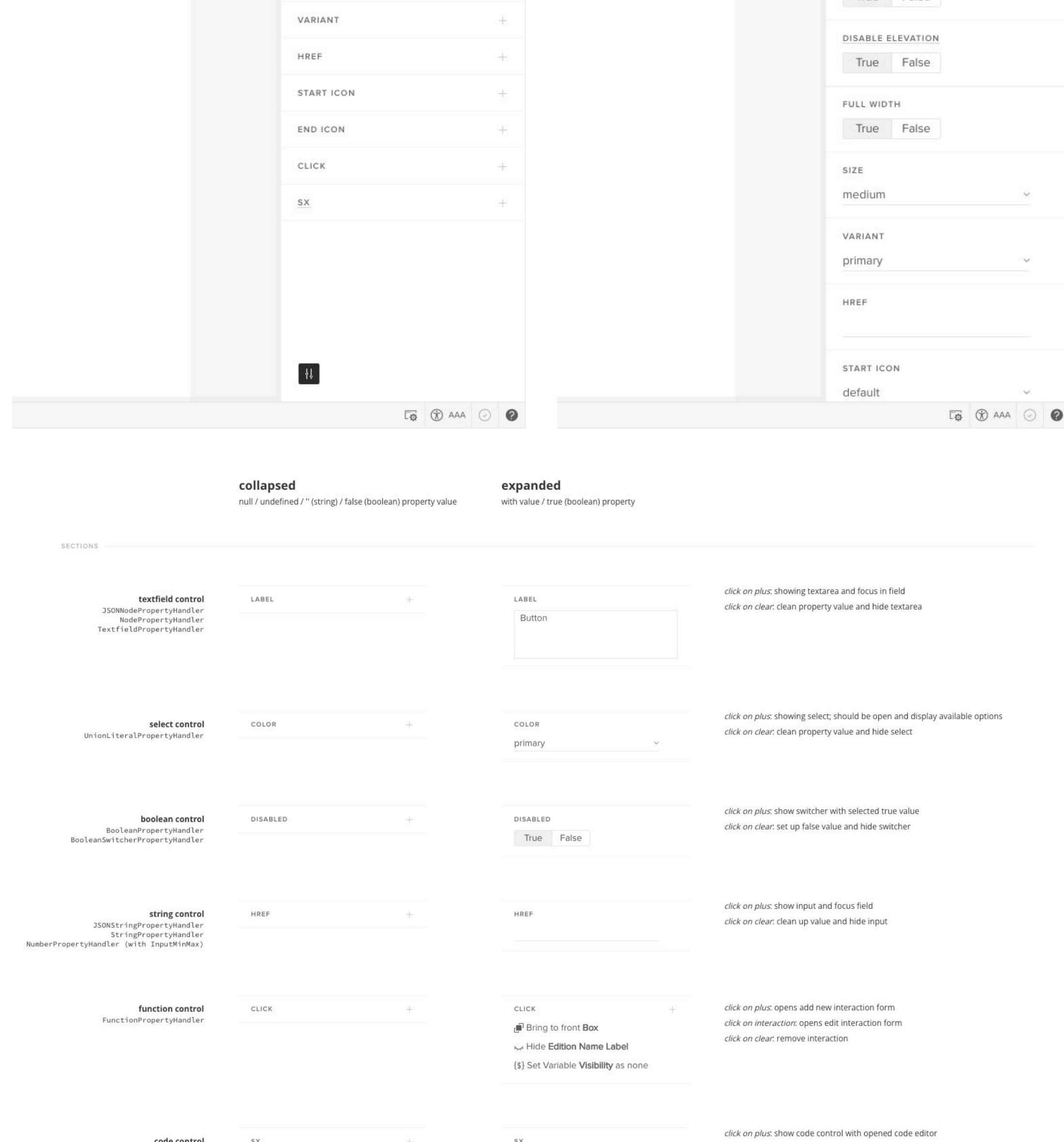
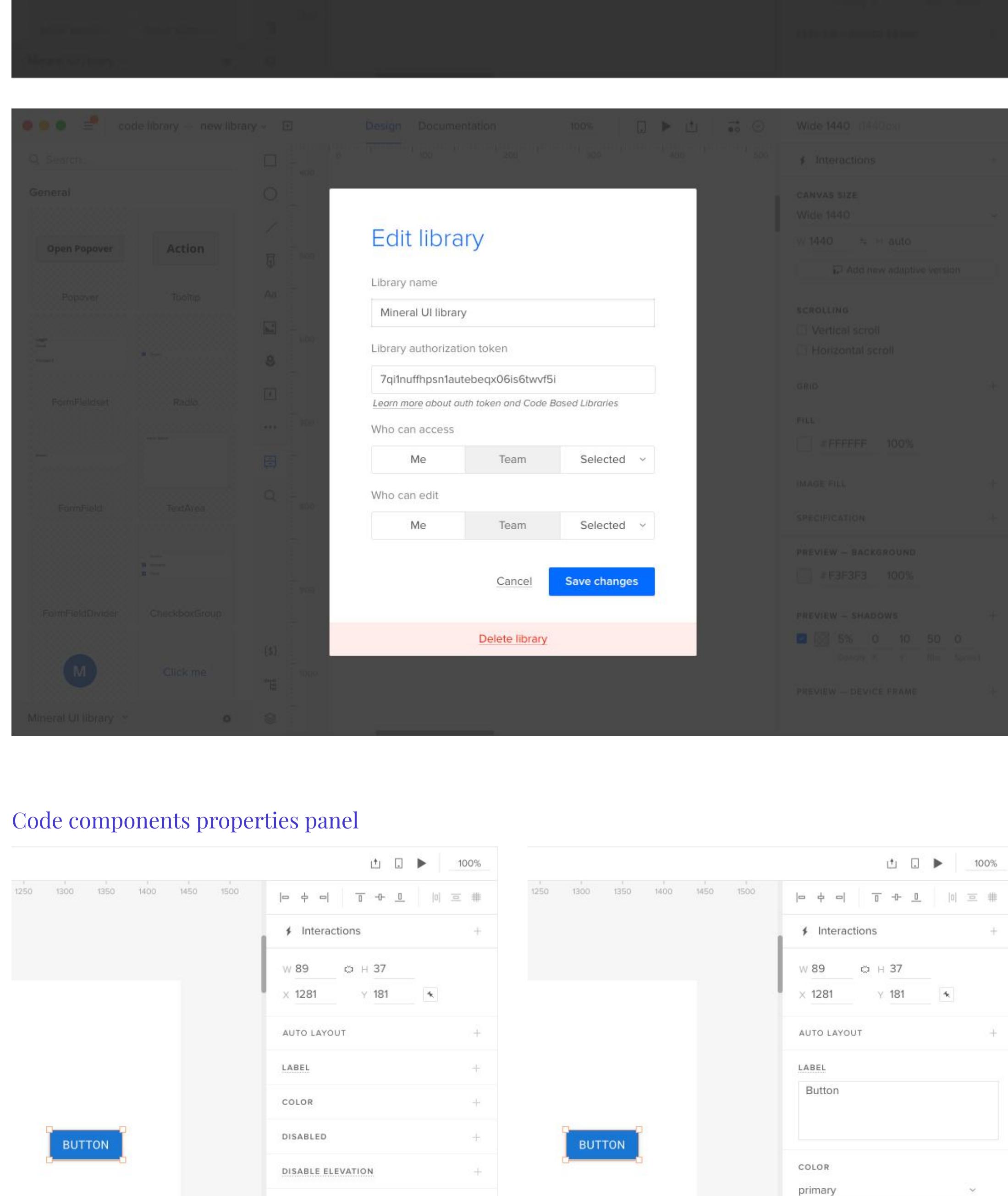
Adverity UI after redesign

UXPin Merge is a technology that bridges the gap between UXPin and Figma. It allows the import of ready-made components from Figma into the UXPin editor, enabling their use in designs. This integration makes it easier for designers to work with the same components across both tools.

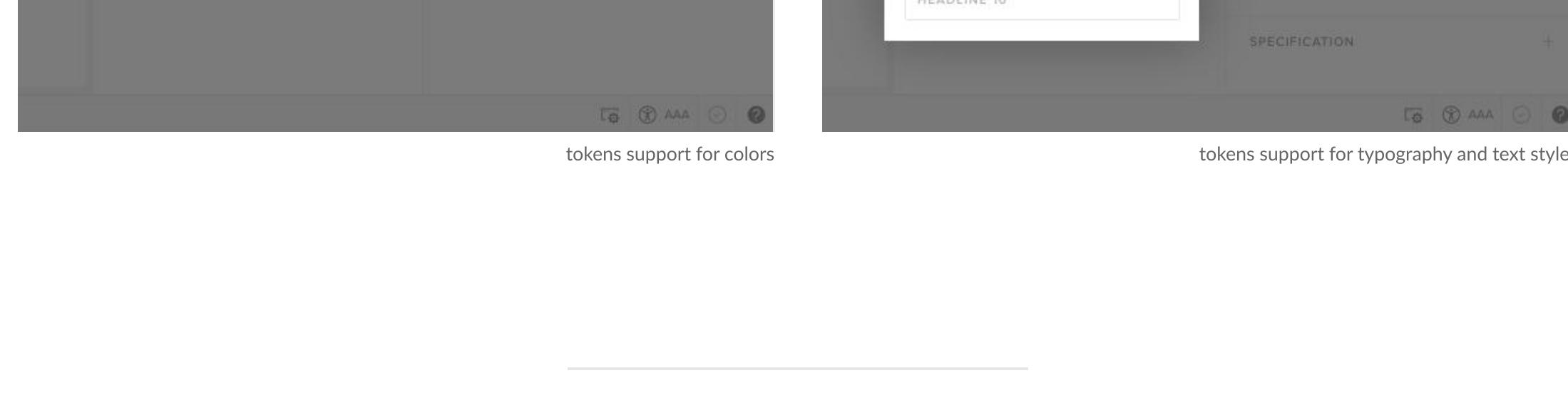
project and library levels, all managed by Git branches or Git tags. When I was a Product Lead in the Merge project, in addition to scoping, creating the product roadmap, and supporting early access users, I was responsible for user experience, design, and UI engineering in React.

A screenshot of the UXPin application interface. On the left, there's a dark sidebar with various icons and a search bar. The main workspace shows a dark-themed wireframe of a user interface. A prominent modal window is centered over the workspace, featuring a white background and a blue header bar with the text 'Add library'. Below the header, the text 'Where will the elements come from?' is displayed. At the bottom of the modal, there are two large, light-blue rectangular input fields. The first input field contains a small house icon, and the second is empty.

—



dangerRadial



UXPin Design System

Based on the current product design, I created a streamlined and cohesive design system.

The goal was simple: tidy up, organize, and simplify the colors, typography, iconography, and interface elements. Throughout this journey, I worked closely with both the design and development teams, resulting in a polished and efficient system. Together, we created a sustainable process for maintaining and growing the design system within the company.

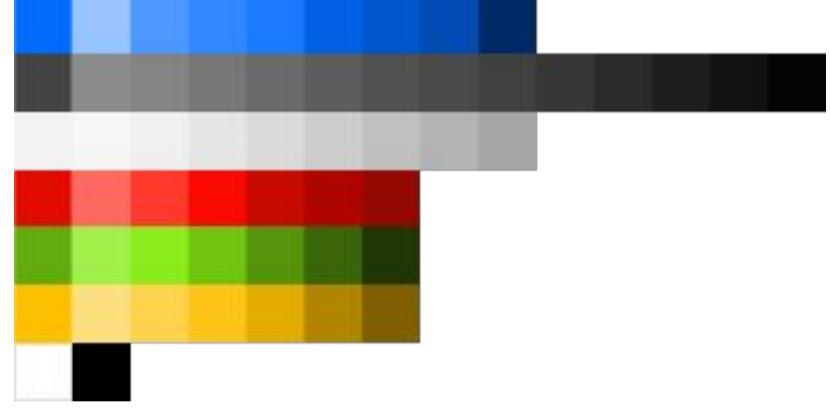
2017 - 2022 | UXPin

role :leader / code + design + evangelisation

Color palette

Based on the colors used in the UXPin application, I created a new color palette. The original 5500 colors defined in code as hex values were streamlined to just 55 key colors. Each of these colors has a base and a range of light and dark accents.

In addition to refining the colors, I developed a consistent naming pattern for them.



Typography

Similar to the color palette, I reviewed all text styles used in the product and compiled a concise list of text styles.

Additionally, we decided to use only two font families: Proxima Nova and Source Code Pro. Each text style or heading can be applied using one of these font families.

heading 1

heading 2

heading 3

heading 4

heading 5

heading 6

heading 7

heading 8

HEADING 9

HEADING 10

text XS: The quick brown fox jumps over the lazy dog.

text S: The quick brown fox jumps over the lazy dog.

text M: The quick brown fox jumps over the lazy dog.

text L: The quick brown fox jumps over the lazy dog.

Icons

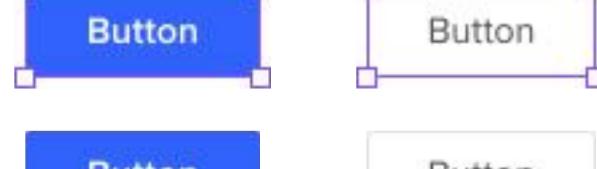
In UXPin, there were 291 icons in use, many of which were duplicates or very similar. By cleaning up and streamlining them, I created a refined icon set of 172 icons.



Components

The fourth part of the design system was components. Similar to the previous areas, I reviewed each version of every component and standardized them, reducing the number of variations. The UXPin Design System library now includes 22 carefully designed and developed components.

These components were recreated as reusable UXPin symbols and coded as React components. Each component is accompanied by detailed documentation, including use cases and examples.



```
<Button>
  Button
</Button>
```

```
<Button
  appearance="neutral"
  weight="light"
>
  Button
</Button>
```

UXPin Desktop App

The project started after one of UXPin's internal hackathon. After POC done on the hackathon we decided to invest some additional time and release the desktop client. During this project, I was leading a team that builds the application. We used the electron.js library.

I was responsible for scoping, user experience, and coding.

2018 | UXPin

role :product lead / ux design + code

