

Selected projects 2015-2024

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.

2022-2024 | Adverity

role :product owner / code

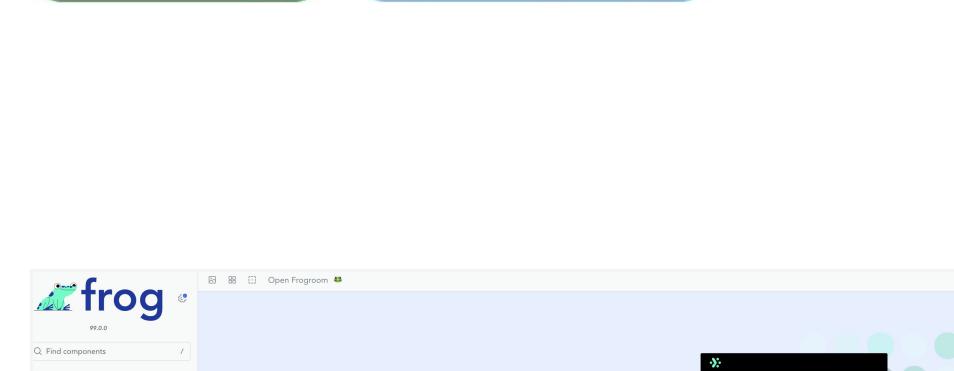
Tokens

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.

```
export const colors: Colors = {
  brand: '#1839A0',
  brandHighlighted: '#2950E3',
  brandActive: '#4071F7',
  brandSecondary: '#F4F8FF',
  brandSecondaryHighlighted: '#E5EEFF',
  brandSecondaryActive: '#D1E1FF',
  brandText: '#182447',
  brandAccent: '#00FFC5',
```

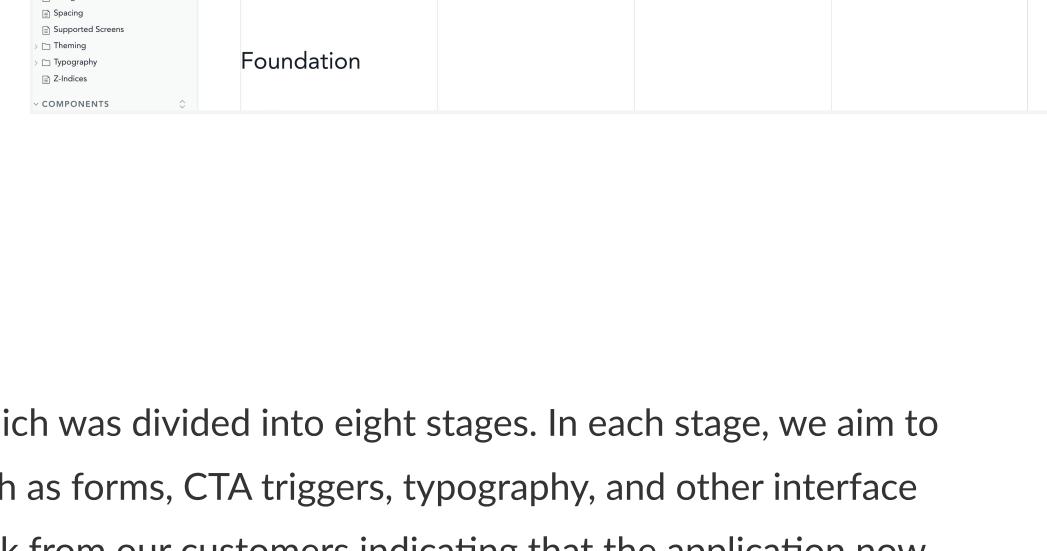
Components

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.

This screenshot shows the Adverity platform interface before the redesign. It features a top navigation bar with links like 'Default', 'Connect', 'Enrich', 'Transfer', 'Explore', 'Present', 'Reveal', and 'What's new'. On the left, there's a sidebar with sections for 'Datastreams', 'Authentications', 'Statistics', 'Managed Onboards', 'All Extracts', and 'All Issues'. The main content area contains a large data table with columns for 'Type', 'Workspace', 'Name', 'Updated', 'Created', 'Last used', 'Billing objects', 'Datastreams', and 'Access Granted'. The table lists various datastreams and their details. A search bar at the top of the table allows filtering by name.

Adverity UI before redesign

This screenshot shows the Adverity platform interface after the redesign. The overall aesthetic is darker and more modern. The sidebar on the left is simplified, containing 'CONNECT DATA' (Datastreams), 'MANAGE DATA' (Enrichments, Data Dictionary), 'USE DATA' (Data Explorer, Dashboards, Destinations, Data Shares), and 'Administration'. The main content area has a clean layout with a header 'Datastreams' and a sub-header 'Datastreams are how you collect data from your chosen data source in Adverity. See how it works'. Below this is a table with columns for 'Datasource', 'Destinations', 'Name', 'Workspace', 'Issues', 'Last Run', 'Next Run', 'Edited', and 'Creator'. The table lists several datastreams with their respective details. A 'Create datastream' button is located in the top right corner of the main content area.

Adverity UI after redesign

UXPin Tokens

benchmarking various tools on the market, such as Tokens Studio and Interplay, and analyzed the draft of the Design Tokens Standard from the Design Tokens Community Group.

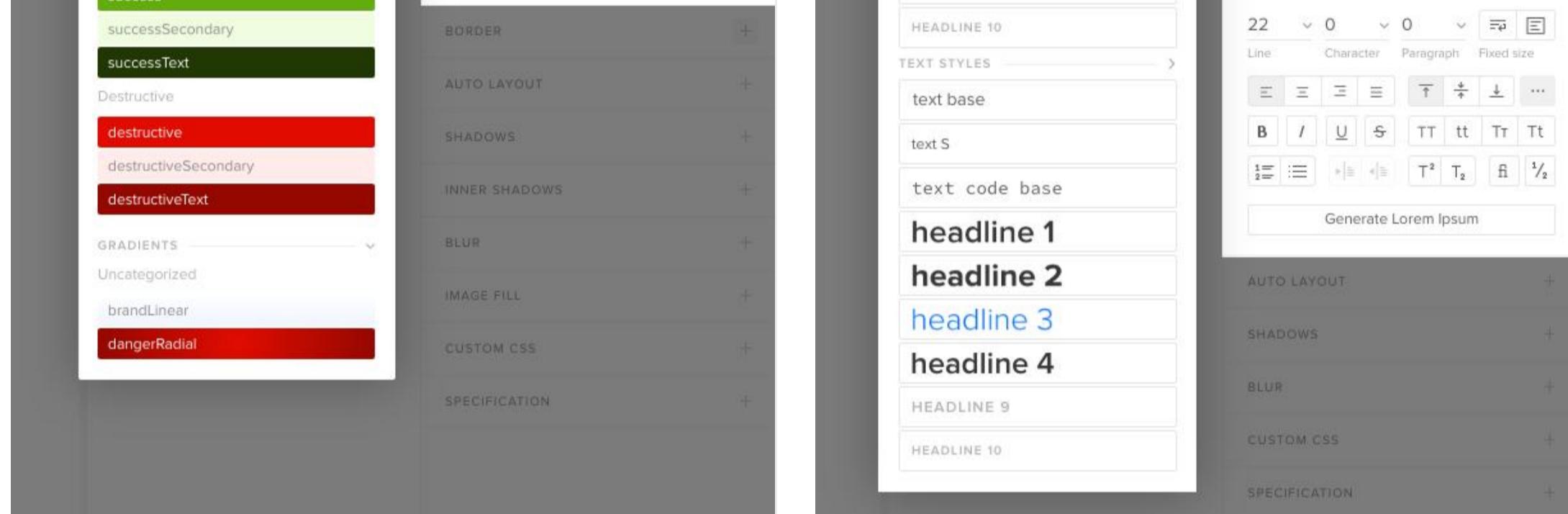
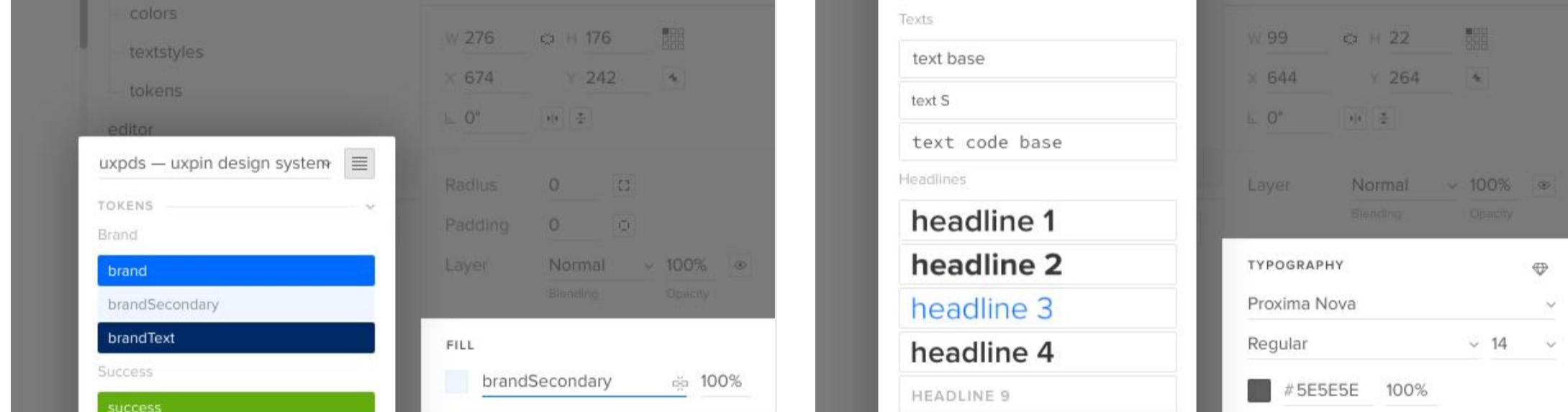
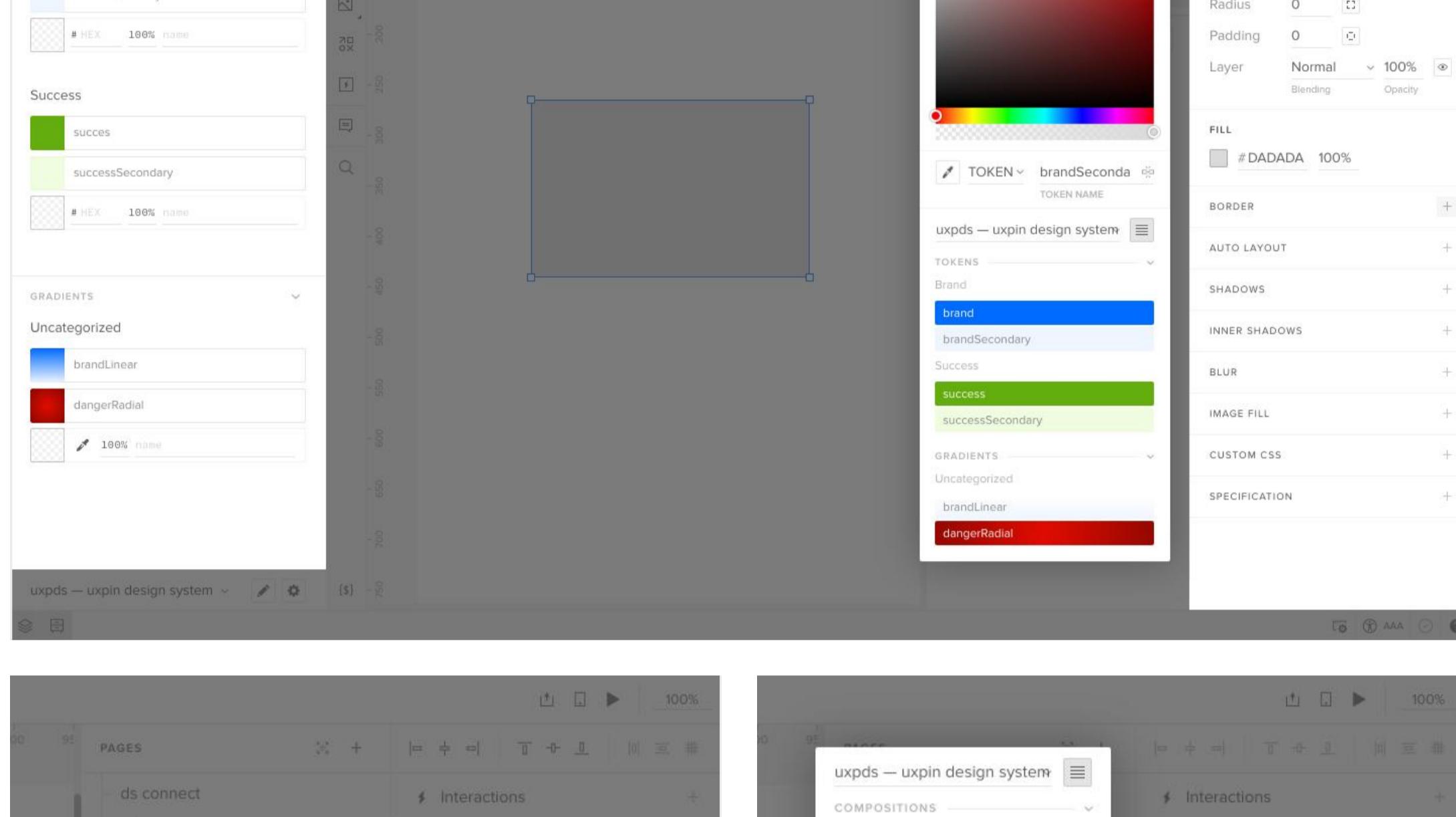
In addition to designing and conceptualizing new features, I also contributed to the technical analysis by sharing knowledge about design tokens and their standards and patterns.

Unfortunately, after thorough technical analysis and cost estimation, the

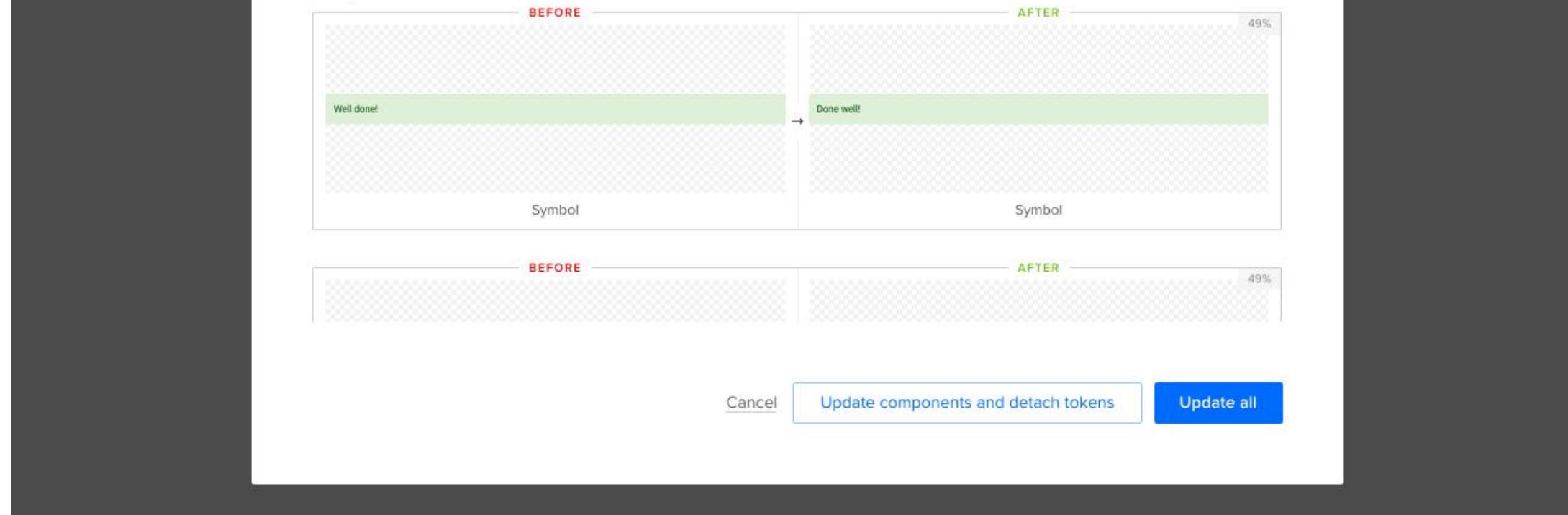
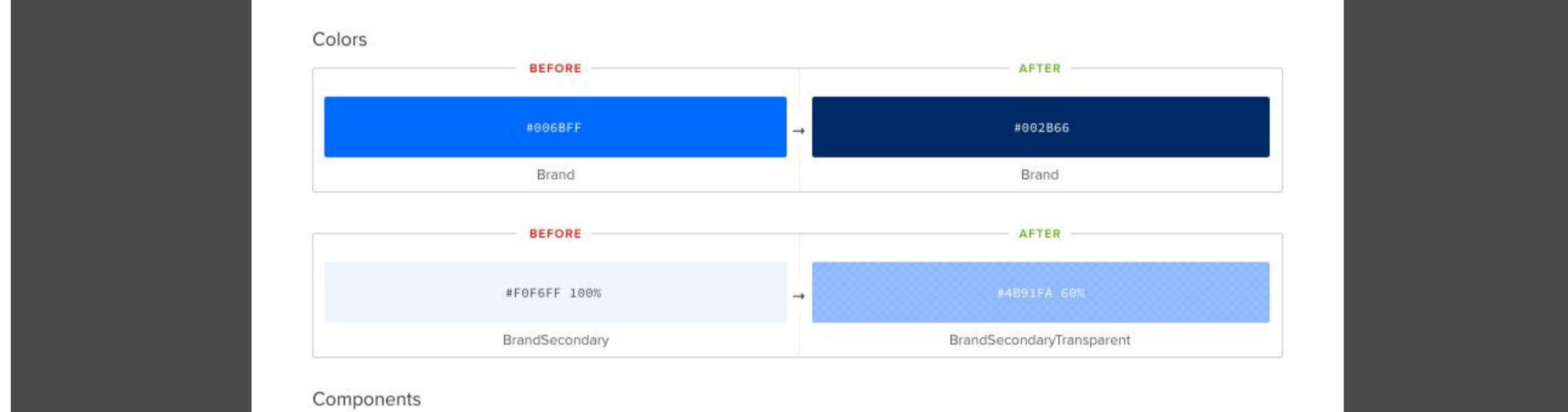
Unfortunately, after thorough technical analysis and cost estimation, the project was canceled due to development costs and limited resources.

Library and properties panel

Brand



1



Merge properties panel unification

From feedback from our users, we find out that properties panel for merge components is not clear to them. The biggest issue was that they had to learn new, different concept than for regular components.

On the other hand we would like to create experience that makes feel that Merge components are something natural and consistent with UXPin editor. We decided to unify properties panel and align Merge components to use same concept as UXPin's classic elements.

2023 | UXPin

role :design

Previous and unified properties panel

The image shows two side-by-side screenshots of the UXPin editor interface, comparing the 'old Merge components properties panel' on the left and the 'unified Merge components properties panel' on the right. Both panels are for a 'Button' component, showing its properties like color, size, and click behavior. The unified panel on the right has a more consistent and modern design, while the old panel on the left looks like a separate control panel.

old Merge components properties panel

unified Merge components properties panel

Merge components sections states

This diagram illustrates the states of various Merge component sections. It shows how sections like 'textfield control', 'select control', 'boolean control', 'string control', 'function control', 'code control', and 'color control' transition between collapsed and expanded states, often triggered by a plus icon. It also shows how sections like 'PROPERTY HAS DESCRIPTION' and 'DEFAULT VALUE HAS BEEN OVERWRITTEN' are managed.

Information about component's library in properties panel

This diagram provides details on how components are listed in the properties panel library. It shows sections for 'default is empty' and 'default has value', each with its own specific behaviors and descriptions.

UXPin Design System

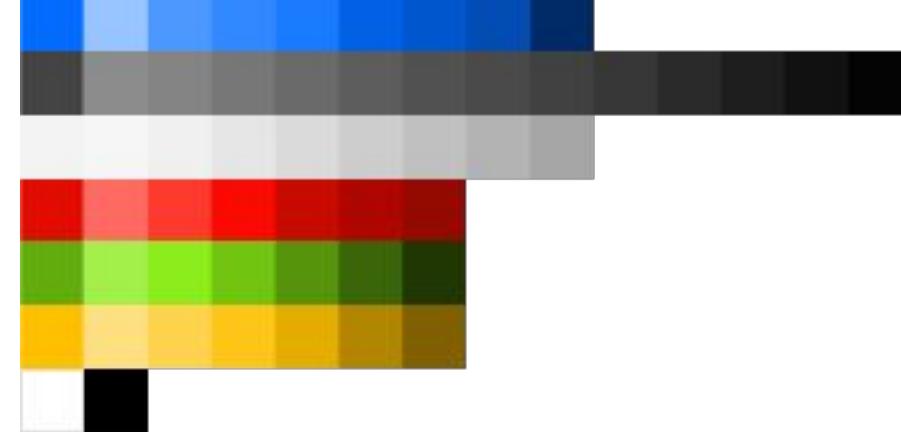
Basing on the actual design of the product I have prepared a design system. The goal was to clean and organize and simplify colors, typography, iconography, and interface elements. During this work, I was in close contact with the design and development team.

2017 - 2022 | UXPin
role :leader / code + design + evangelisation

Color palette

Based on colors used in the UXPin application I've prepared palate of colors. ~5500 colors defined in code as hex has been limited to 55 defined colors. Each color has its own base and number of lighten / darken accents.

Except for colors itself, I have prepared the naming pattern.



Typography scale

Similar to colors I have check all text styles used in the product and prepare a close list of text styles.

In addition, we decide to use only 2 font families - Proxima Nova and Source Code Pro. Each text or heading can behave in one of the available 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

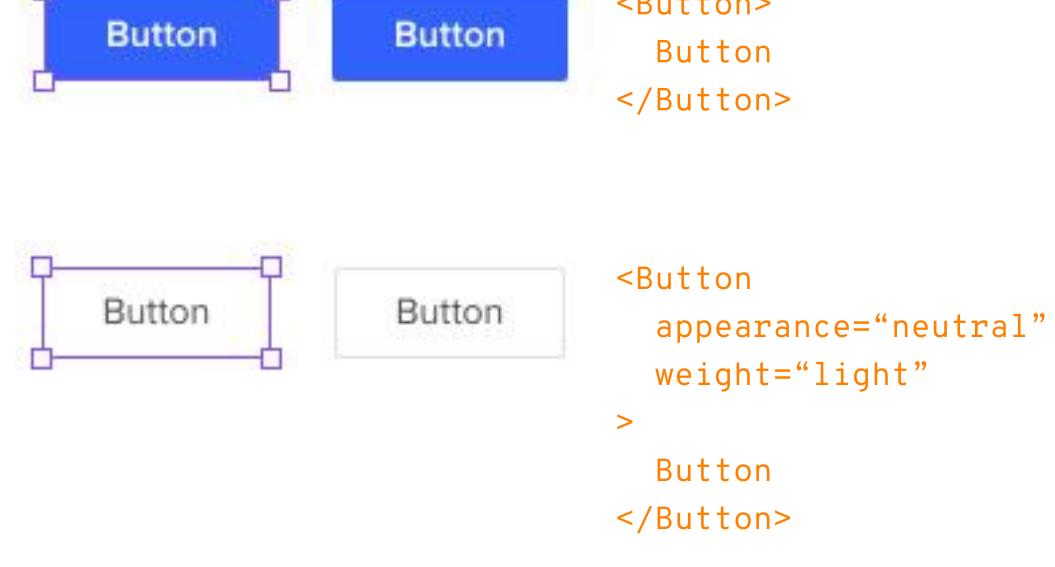
In UXPin was 291 used icons. A lot of those icons were duplicate and similar. During cleaning, I decrease the number of icons to 172.



Components

The fourth part of the design system is components. Similar to previous areas I have check each version of each component and normalize it to one with few versions. Currently, in the UXPin Design System library, we designed and developed 22 components.

Components were recreated as reusable UXPin symbols and coded as a React component. Each of the components has detailed documentation with use cases and examples



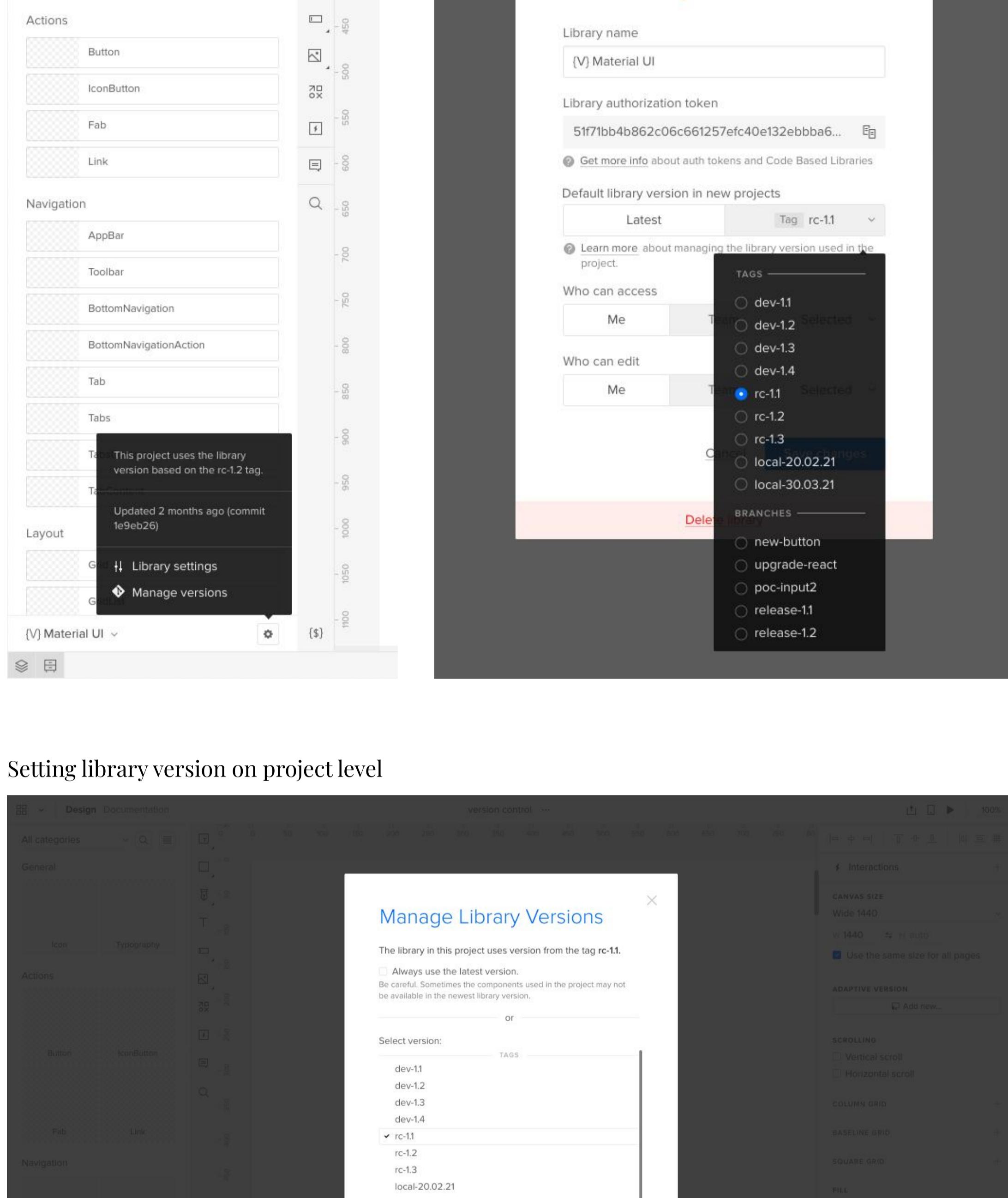
Merge Libraries Version Control

Based on feedback from our users, how they are implementing Merge to their organisation we realise there's need to handle multiple versions of the Merge library. Some cases was working on redesign, some was refactoring. Different projects might need different versions on components for testing purposes.

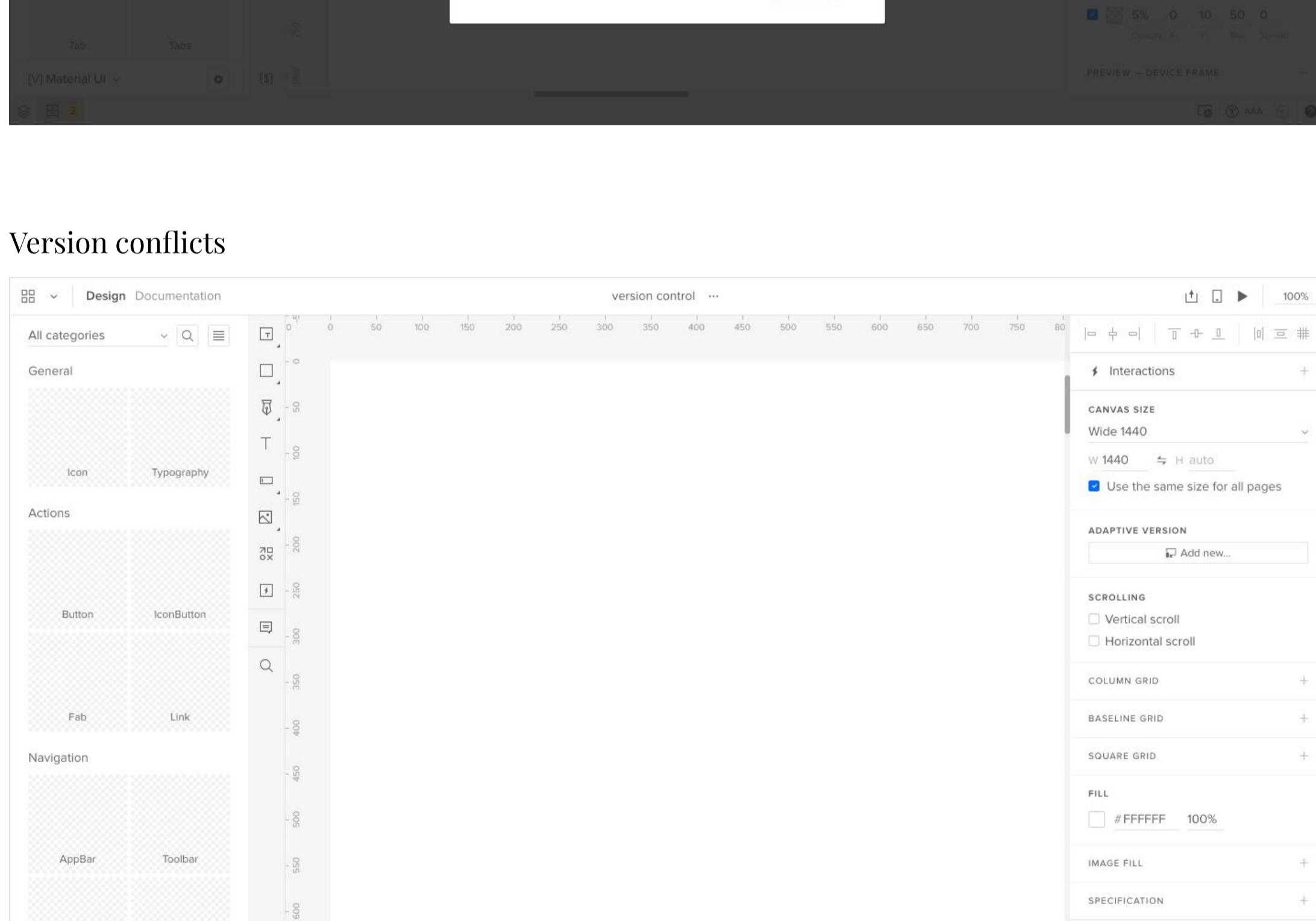
We decided to based on Git branches and tags.

2021 | UXPin
role :code + design

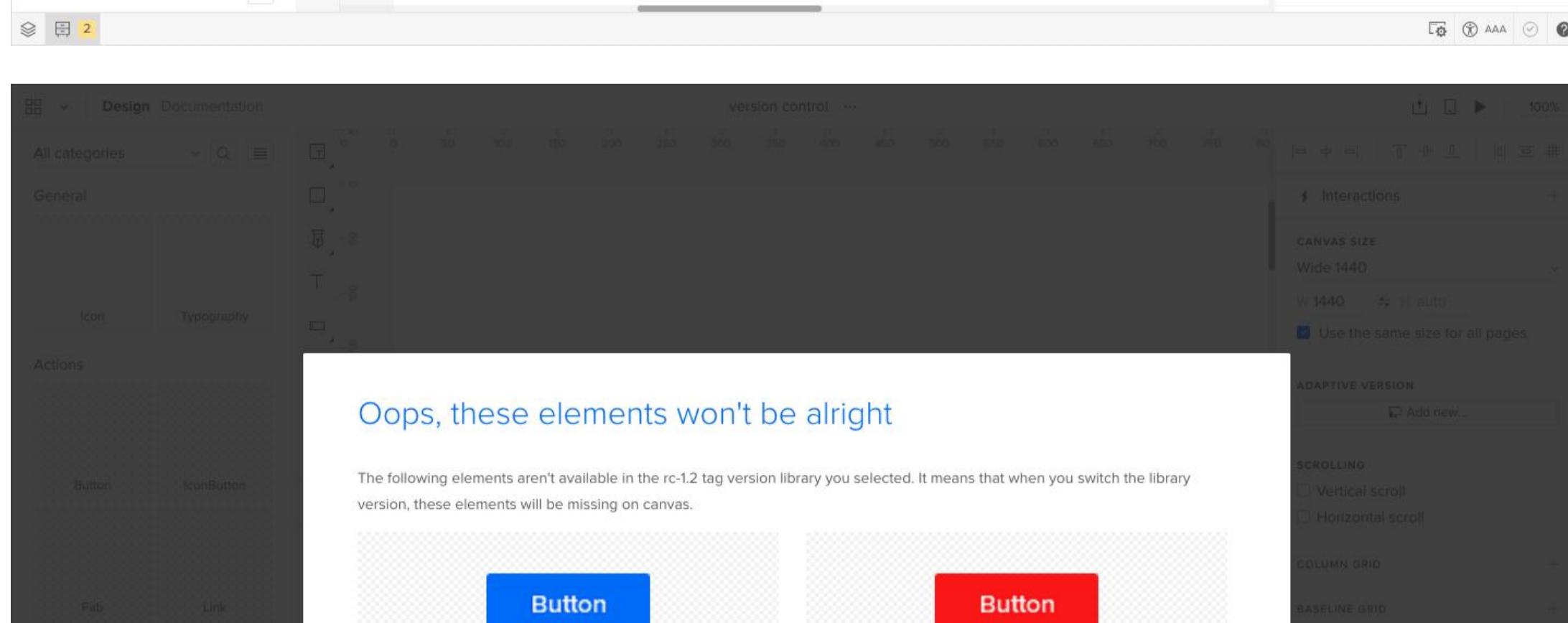
Setting default version of the library for all projects



Setting library version on project level



Version conflicts



Oops, these elements won't be alright

The following elements aren't available in the rc-1.2 tag version library you selected. It means that when you switch the library version, these elements will be missing on canvas.

Default Button

Destructive Button

Cancel Proceed

page 6 – Bartosz Dębicki Selected Projects 2015-2024

Adele Design System details page

The project was related to creating a dedicated page for each of the design systems from the list. The goal was clear – increase SEO on Adele to provide traffic to UXPin.

2020 | UXPin
role :design

The screenshot shows a landing page for the GitHub Primer design system. At the top, there's a navigation bar with 'Adele' and 'UXPin'. Below it is a large title 'The Github Primer' with a subtitle 'design system is called'. A callout box provides links to the 'website' (<https://primer.github.io/>) and 'repository' (<https://github.com/primer/primer/>). The main content area is divided into three sections: 'Technology', 'Design', and 'Documentation & guidelines'. Each section contains a table comparing various design system components across different technologies like Code Depth, HTML/CSS, and UI Kit. The 'Technology' section includes rows for Components (✓), JS Library/Framework (✗), Typescript (✗), Web Components (✗), Tests (✗), Linter (StyleLint), CSS (Sass), CSS In JS (✗), Design Tokens (Scss), Bundle Manage (Primer Module Build Scriptno), and Distribution (npm). The 'Design' section includes rows for UI Kit (✗), Brand Guidelines (✗), Color Naming (natural e.g. gray-300), Contrast Analysis (✗), Typography (✓ | Go to Typography), Icons (Octicons (SVG)), Space / Grid (✓ | Go to Space / Grid), Illustration (✗), Data Visualization (✗), Animation (✗), and Voice & Tone (✗). The 'Documentation & guidelines' section includes rows for Accessibility Guidelines (✗), Design Principles (✗), Documentation Website (✗), Code Documentation (Markdown), and Storybook (✓ | Go to Storybook). At the bottom, there's a section for 'Evangelize Design System with a Free Template!' featuring a yellow square icon, and another for 'UXPin: Prototype and Manage Your Design System' featuring a wireframe icon. Both sections include calls to action: 'Learn more & download' and 'Start a free trial now!'. The footer contains the UXPin logo and copyright information: '2010 - 2020 UXPin INC' and 'Maintained by Marcin Treder. Inspired by [Awesome Design Systems](#) by Alex Pate. Great design systems resources: [Styleguides.io](#)'.

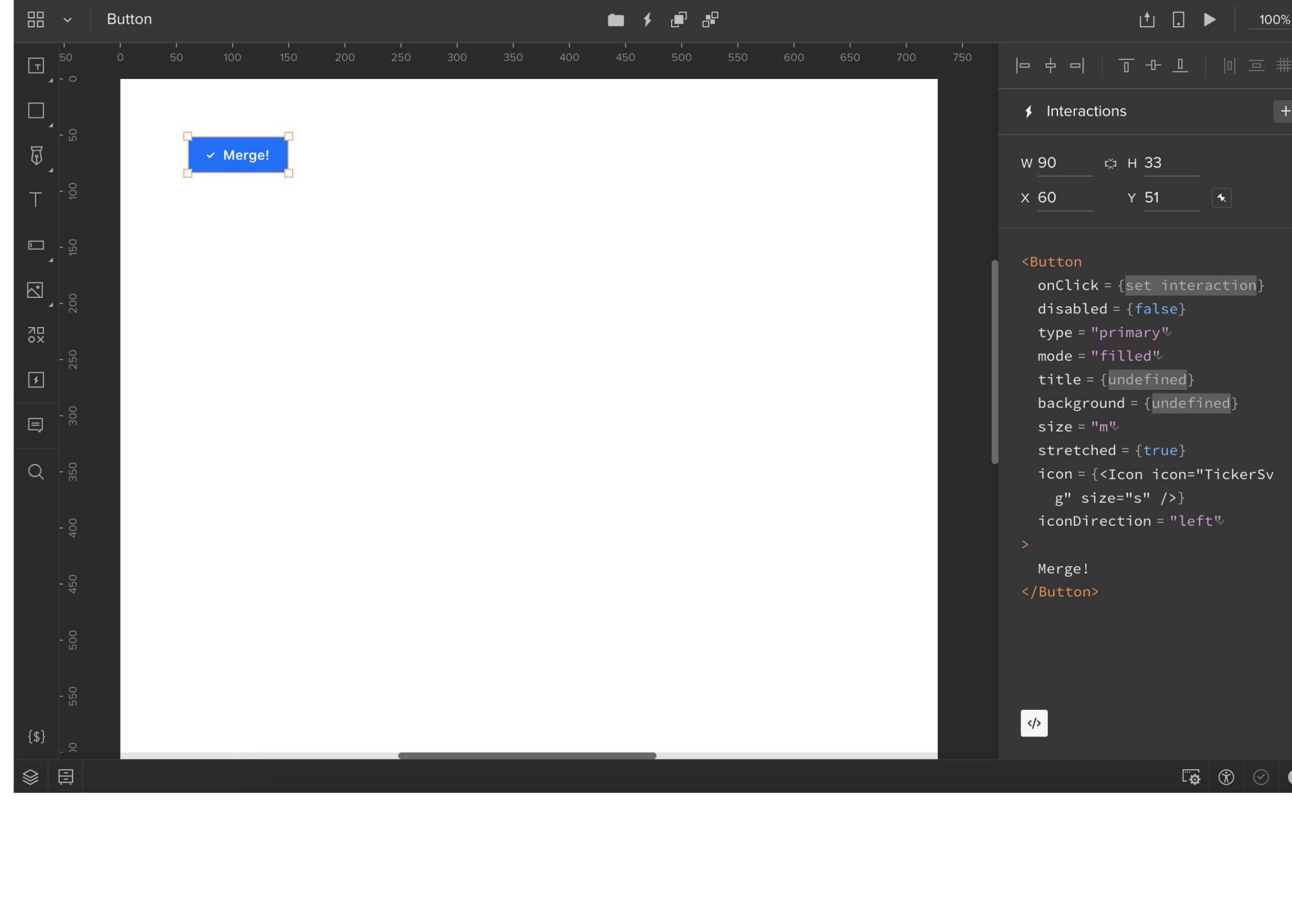
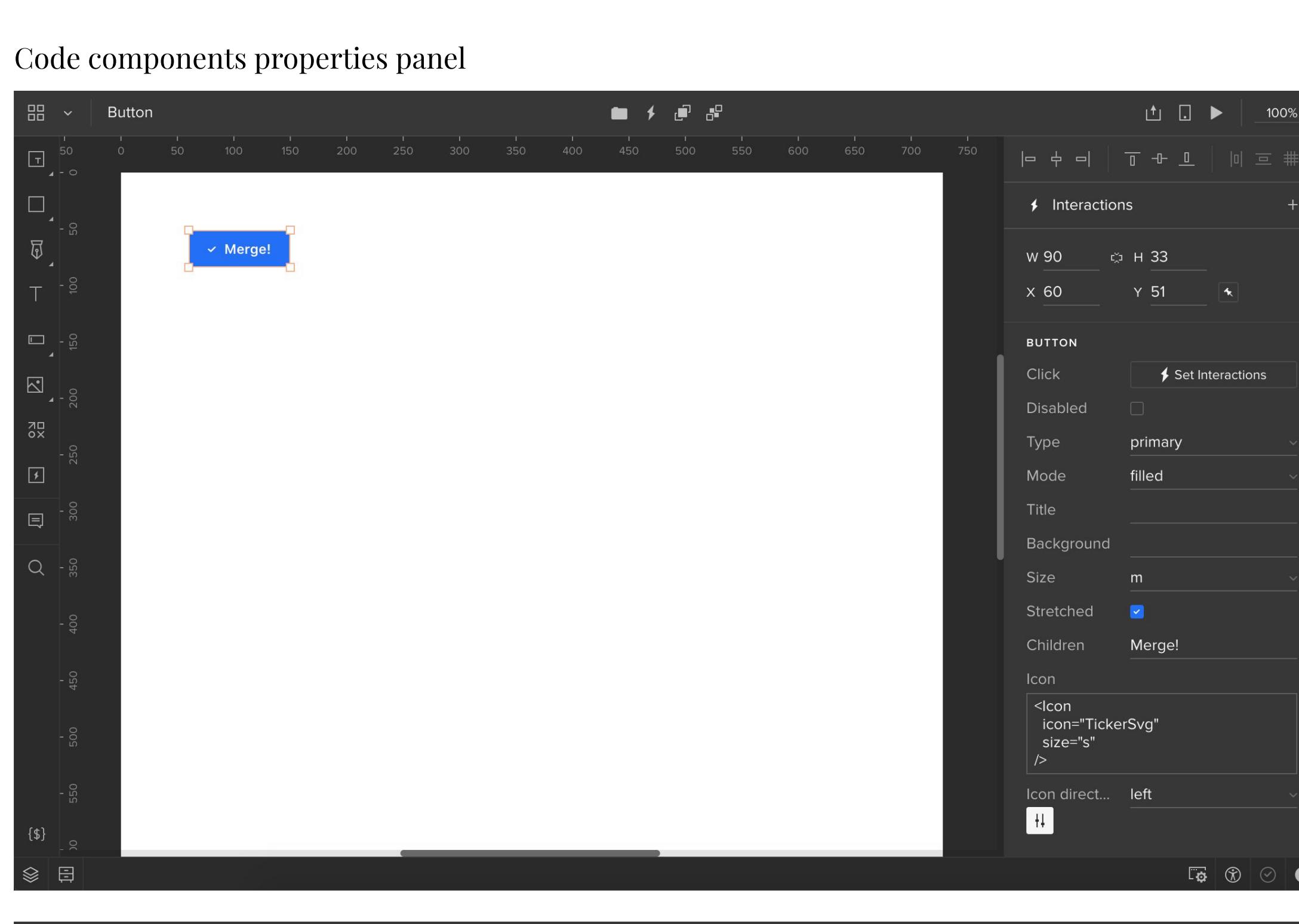
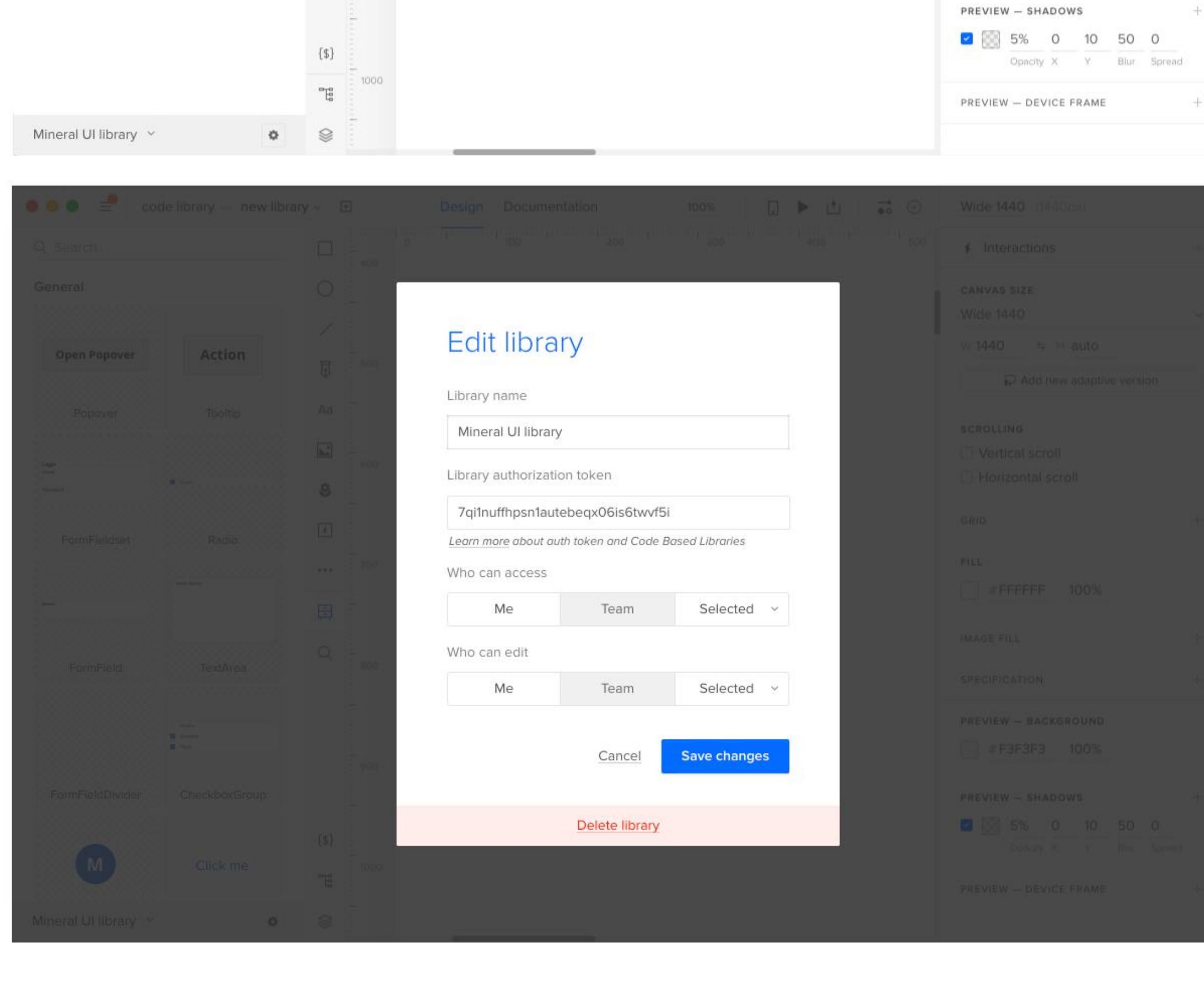
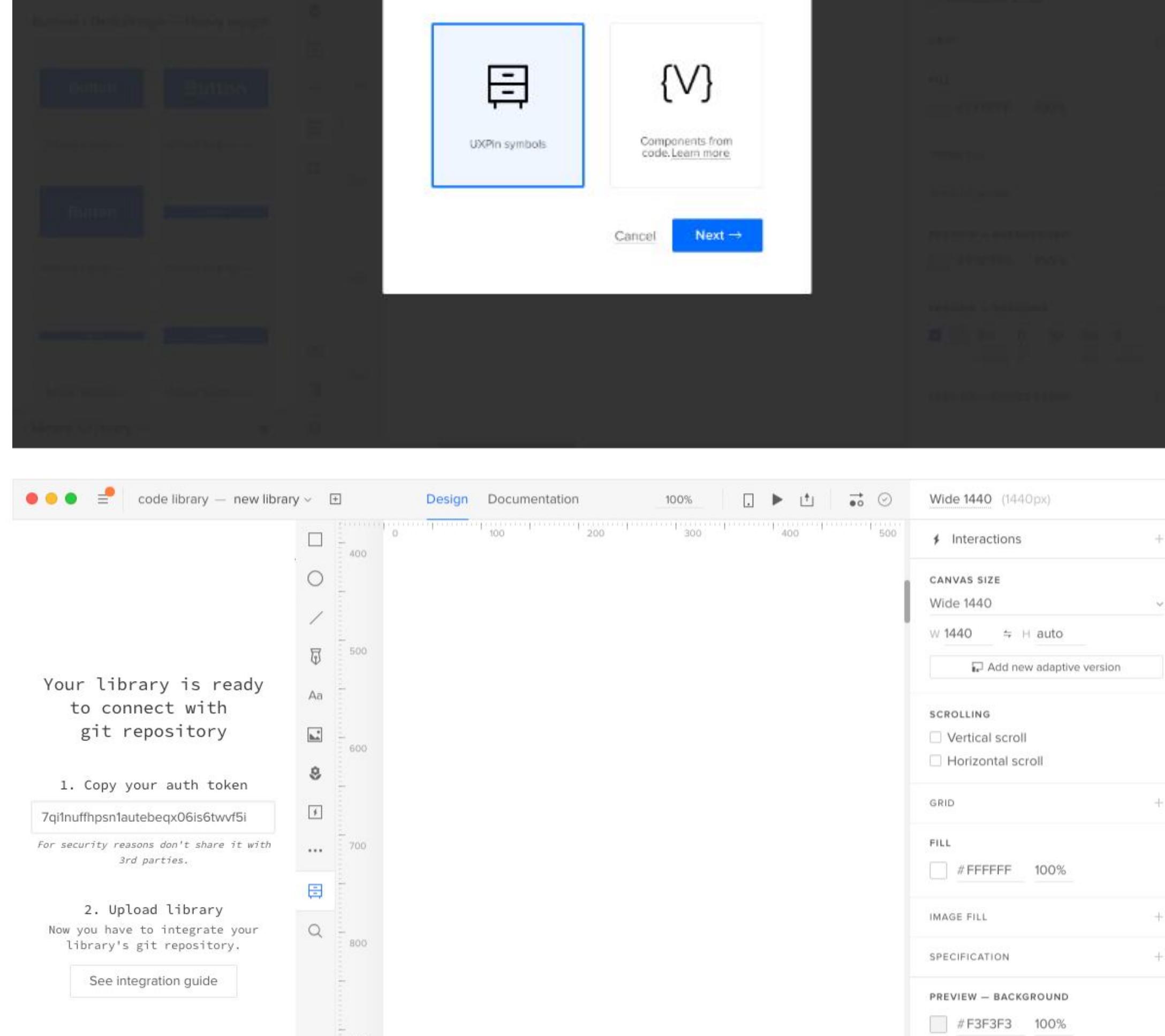
UXPin Merge

UXPin Merge is a technology that bridges the gap between design and technology. Technology gives the possibility to import real production react components into UXPin and use them in designs. This approach gives developers and designers the possibility to work on the same elements and use a single source of truth – the git repository.

When I was a Product Lead in the Merge project except scoping, creating product roadmap, and supporting early access users and I was responsible for a project user experience and the design.

2021 | UXPin
role :product lead / ux design

Creating new library flow



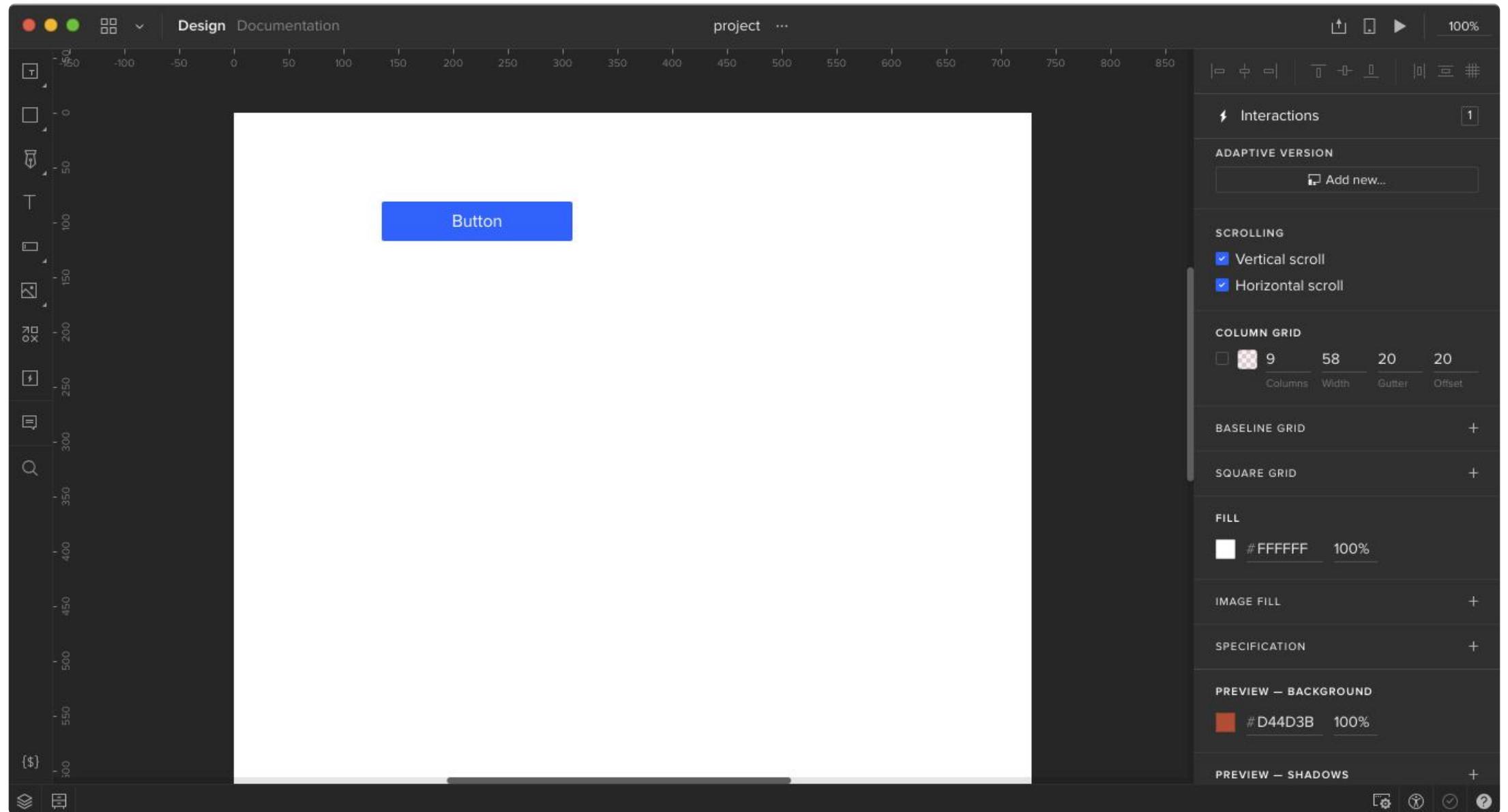
UXPin Desktop App

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



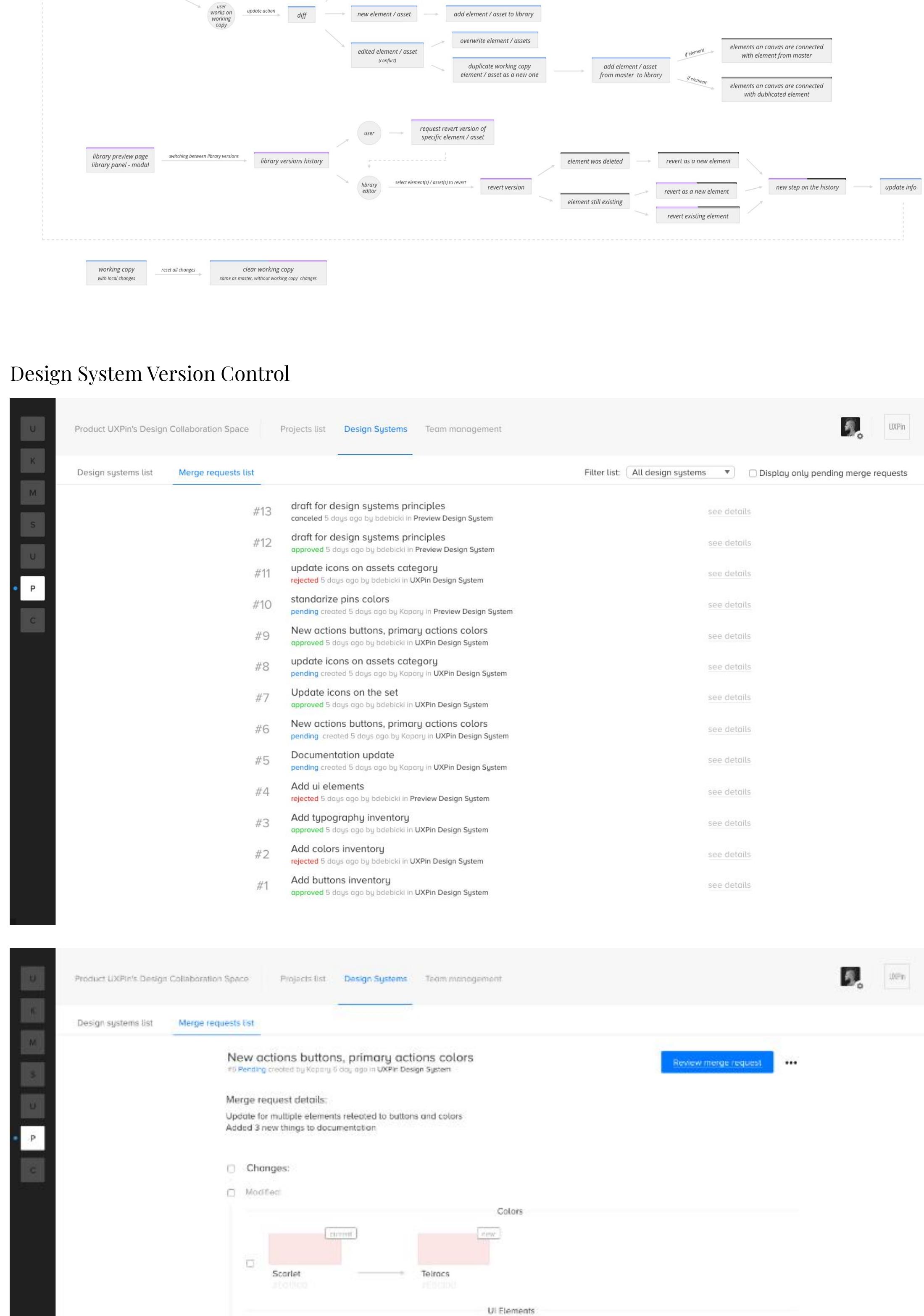
Right after the release of Design Systems and Design Libraries features in UXPin next step supposed to be version control of those areas. The project was very technical centered and complex because a lot of actions happened in backend hidden by the interface.

The diagram illustrates the flow between the Editor and Dashboard areas. It shows two main components: 'actions on Editor area' (blue box) and 'actions on Dashboard / New view area' (purple box). Below these, 'Backend actions' (grey box) are shown on the left, connected by an arrow labeled 'action' to 'User actions' (grey box) on the right.

```
graph LR; E["E  
actions on Editor area"] -- "action" --> D["D  
actions on Dashboard / New view area"]; B["B  
Backend actions"] -- "action" --> U["User actions"]
```

The diagram illustrates a two-step workflow:

- Step 1:** A "master version" box is connected by an arrow labeled "change some element / asset" to a "working copy" box.
- Step 2:** A "working copy" box is connected by an arrow labeled "change some element / asset" to a "update info" box.



M	New action
S	#6 Approved by [REDACTED]
U	Details:

• P
C

Update for multiple elements related to buttons and colors
Added 3 new things to documentation

Review details:
Approved 3 from 5 changes by bdebicki 2 days ago with a comment:
Good job. I picked only 3 changes because other wasn't related to buttons.

Approved changes:

Modified:

UI Elements



Buttons category description has been changed

current → new

Material buttons trigger an ink reaction on press.
They may display **text or imagery**. Flat buttons
and raised buttons are the most commonly used
types.

Material buttons trigger an ink reaction on press.
They may display **text, imagery, or both**. Flat
buttons and raised buttons are the most
commonly used types.

[View full description](#)

UXPin Design System

Versions history X

Colors

Typography

Assets

Image

Icons

UI Patterns

Colors



Blue Ribbon #006cff

Scarlet #E01300

We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design. This spec is a living document that will be updated as we continue to develop the tenets and specifics of material design.

Add buttons inventory v5 **current** changes by Kopyng

Add typography inventory v4 changes by Kopyng

Approved by bdebicki 6 days ago merge request details revert changes

Add typography general styles and headline styles

Prepare first documentation v3 changes by Kopyng

New actions buttons, primary actions colors v2 changes by Kopyng

Draft for design systems principles v1 changes by Kopyng

Typography

Typeface

Playfair Display

Regular *Italic* **Bold**

The quick brown fox jumps over the lazy dog.

Lato

Light *Italic* Semibold **Bold**

The quick brown fox jumps over the lazy dog.

Text styles

Heading 1

Playfair Display Bold • 40px / 48px

Heading 2

Playfair Display Regular • 36px / 42px

Assets

Icons

Library Version Control

The screenshot shows the Figma interface with the 'Design' tab selected. A modal window titled 'Design library version history' is open, showing a list of versions for an asset named 'Colors'. The first version is highlighted as 'current'. The modal includes a 'Add buttons inventory' button and a note about changes by Kapary. At the bottom, there are links for 'merge request details' and 'revert changes'.

- Image
- Icons
- UI Patterns

A screenshot of a design system interface. At the top left, there's a navigation bar with icons for search, file, and settings. The main header is 'Typography' with a subtitle 'Typeface'. Below that is a section for 'Playfair Display' with three font styles: 'Regular', 'Italic', and 'Bold'. A sample text 'The quick brown fox jumps over the lazy dog.' is shown in the regular style. On the right side, there's a sidebar with several documentation items: 'Prepare first documentation' (v.3), 'New actions buttons, primary actions colors' (v.2), and 'draft for design systems principles' (v.1). The bottom half of the screen shows a dark workspace with a floating modal window. This modal contains a merge request for 'New actions buttons, primary actions colors'. It shows the request was accepted by bdebicki 1 day ago and created by Kapary 6 days ago. It also lists 'Merge request details' (update for multiple elements related to buttons and colors, added 3 new things to documentation) and 'Review details' (Accepted 3 from 5 changes by bdebicki 2 days ago with comment: 'Good job. I picked only 3 changes because other wasn't related to buttons.').

was it raining at open'er festival?

If you are a fan of the Open'er Festival or live in the Tricity Poland you probably know jokes about the weather during the festival. I decided to analyze historical data and refute or confirm the myth that it always rains during the Open'er. The project is still in development.

2017 | private project
role :design + code



was it raining at open'er festival? [2017](#) [2016](#) [2015](#) [2014](#) [2013](#) [2012](#)

yep. this site use cookies. click [here](#) for more.

2017
OPEN'ER FESTIVAL POWERED BY ORANGE

28th June - 1st July
Babie Doły Airport, Gdynia

RADIOHEAD / FOO FIGHTERS
THE WEEKEND / THE XX / LORDE

[see full lineup +](#)

RAIN

NO

yep. this site use cookies. click [here](#) for more.

made with ❤ in Gdynia 2017 — [about](#) — [cookies](#)

was it raining at open'er festival? [2017](#) [2016](#) [2015](#) [2014](#) [2013](#) [2012](#) [2011](#) [2010](#) [2009](#) [2008](#) [2007](#) [2006](#) [2005](#) [2004](#) [2003](#) [2002](#)

2016
OPEN'ER FESTIVAL POWERED BY ORANGE

29th June - 2nd July
Babie Doły Airport, Gdynia

FLORENCE + THE MACHINE
RED HOT CHILI PEPPERS
LCD SOUNDSYSTEM | SIGUR RÓS
PHARRELL WILLIAMS | KYGO

[see full lineup +](#)

RAIN

YES

yep. this site use cookies. click [here](#) for more.

made with ❤ in Gdynia 2017 — [about](#)

was it raining at open'er festival? [2017](#) [2016](#) [2015](#) [2014](#) [2013](#) [2012](#) [2011](#) [2010](#) [2009](#) [2008](#) [2007](#) [2006](#) [2005](#) [2004](#) [2003](#) [2002](#)

LINUP 2016 [x](#)

**FLORENCE + THE MACHINE
RED HOT CHILI PEPPERS
LCD SOUNDSYSTEM | SIGUR RÓS
PHARRELL WILLIAMS | KYGO**

AT THE DRIVE-IN | BASTILLE | BEIRUT | CARIBOU
CHVRCHES | DAWID PODSIADEŁO | FOALS | GRIMES
KURT VILLE & THE VIOLATORS | M83 | MAC DEMARCO
PAUL KALKBRENNER | PJ HARVEY | SAVAGES
SKEPTA | TAME IMPALA | THE 1975
THE LAST SHADOW PUPPETS | WIZ KHALIFA

CATZ 'N DOGS | DJ TENNIS | FLIRTINI | JACEK SIENKIEWICZ | KAMP! | KORTEZ
LUNICE | ŁONA I WEBBER | MAŁPA | MARIA PESZEK | NOTHING BUT THIEVES
PTAKI | RASMENTALISM | REBEKA | RYSY | SECTION BOYZ
WINCE STAPLES | XANAXXA | ZBIGNIEW WODESKI with MITCH & MITCH ORCHESTRA
AN ON BAST | BAASCH | CHINO | COLDAIR | CZELUŚCJUTRO X KOSA | GROBEL | HANA
HEROES GET REMEMBERED | JAAAI | JÓGA | KUBA SOJKA | KROKI | LXMP | NAGROBKI
OLIVIER HEIM | OTSCHODZI | PIOTR BEJMAR | PIOTR ZIOLA | SHY ALBATROSS | SONAR | SOOMT
SPOKEN LOVE | STARA RZEKA | SUUMO | T'NIEN LAI | TERRIFIC SUNDAY | V/O | WE DRAW A | ZAMILSKA

RAIN [x](#)

29th June	yes
30th June	yes
1st July	no
2nd July	yes

yep. this site use cookies. click [here](#) for more.

made with ❤ in Gdynia 2017 — [about](#)

Spoiler alert: for 14 of 18 editions of the festival rain was noticed]:->

Simplify editor

In 2017 in UXPin we did a huge usability test with our users about the condition of our application. Based on the research we decided to prepare a package of improvements in working with elements on the canvas and small redesign of UI.

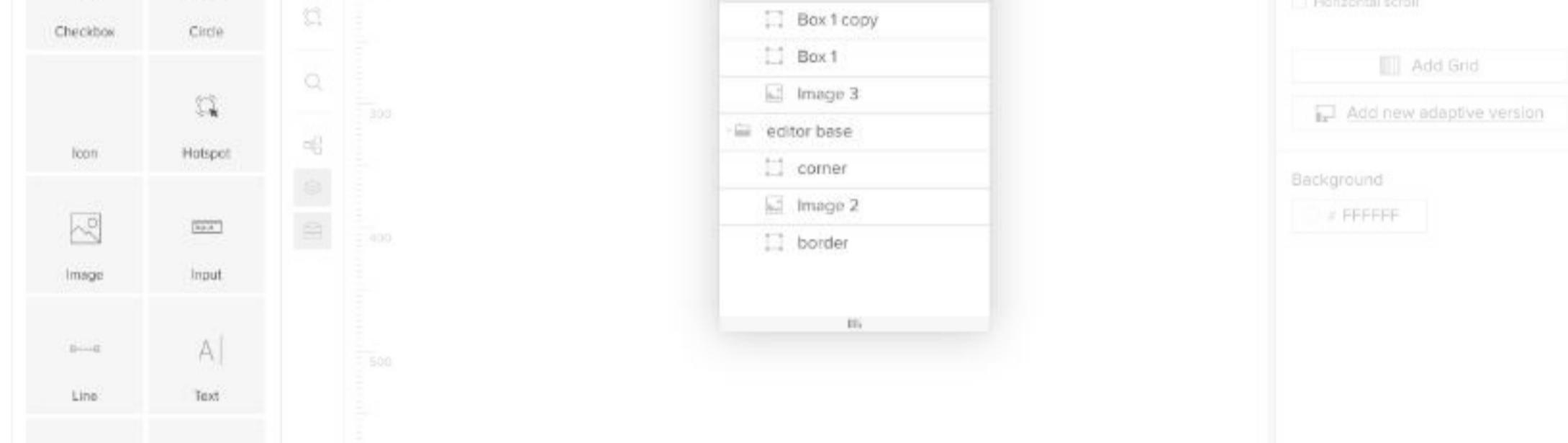
UI changes contain redesign of the top bar and release toolbar with quick access to the most popular elements and floating layers panel. Interactions flow was also simplified and enriched by predefined actions (based on the most popular ones).

2017 | UXPin
role :product lead / ux design

Top bar and toolbar



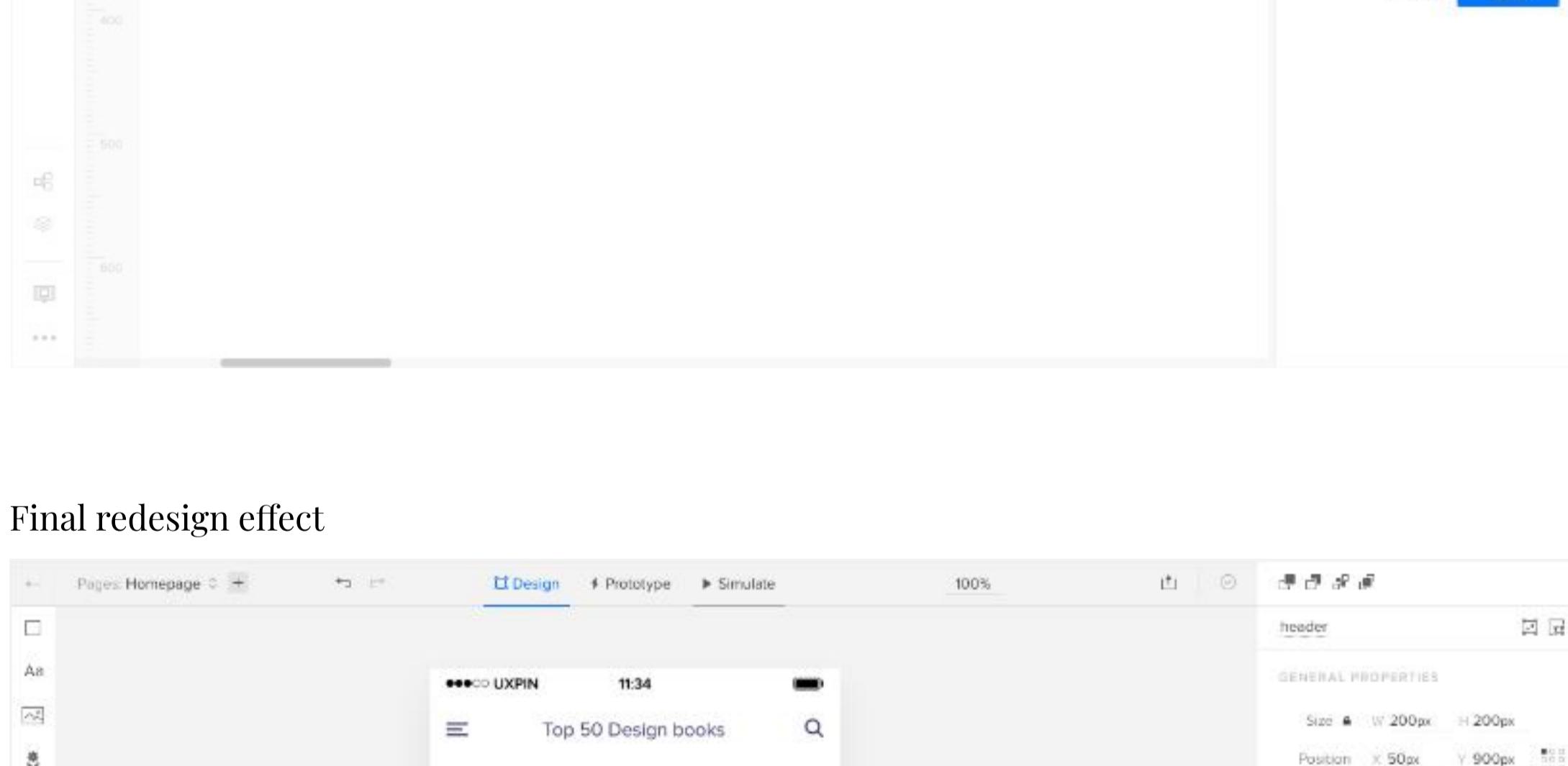
Left panels and floating layers



Interactions panel



Final redesign effect

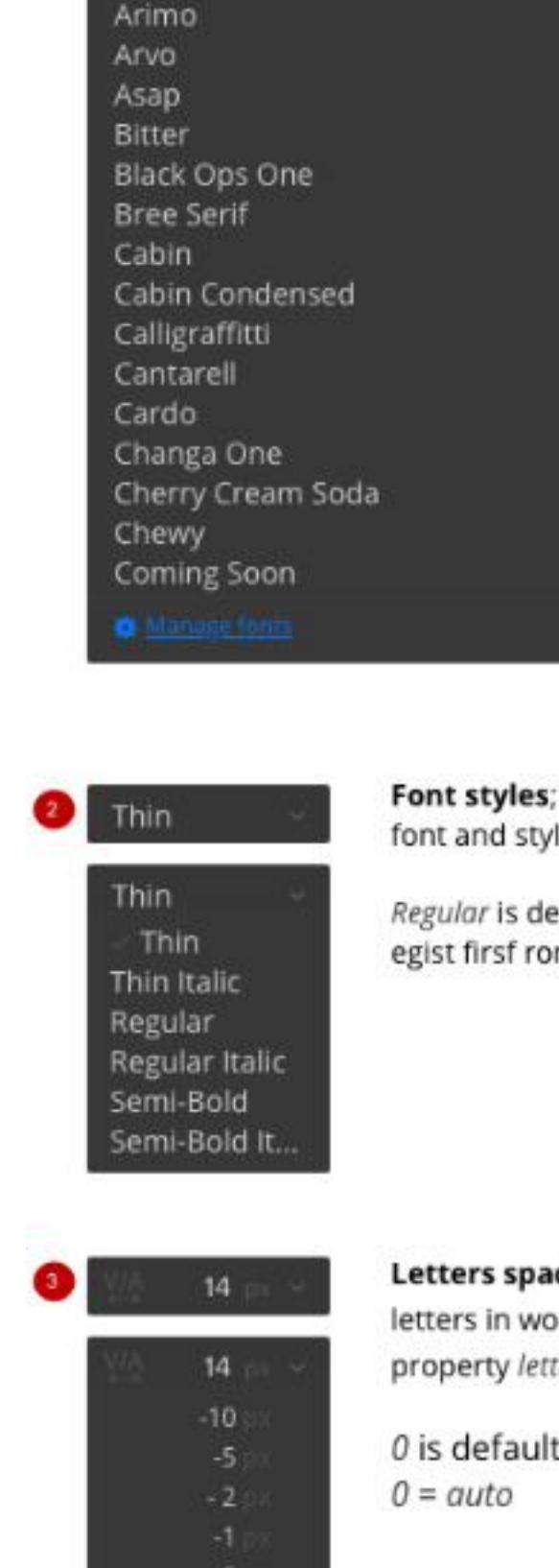
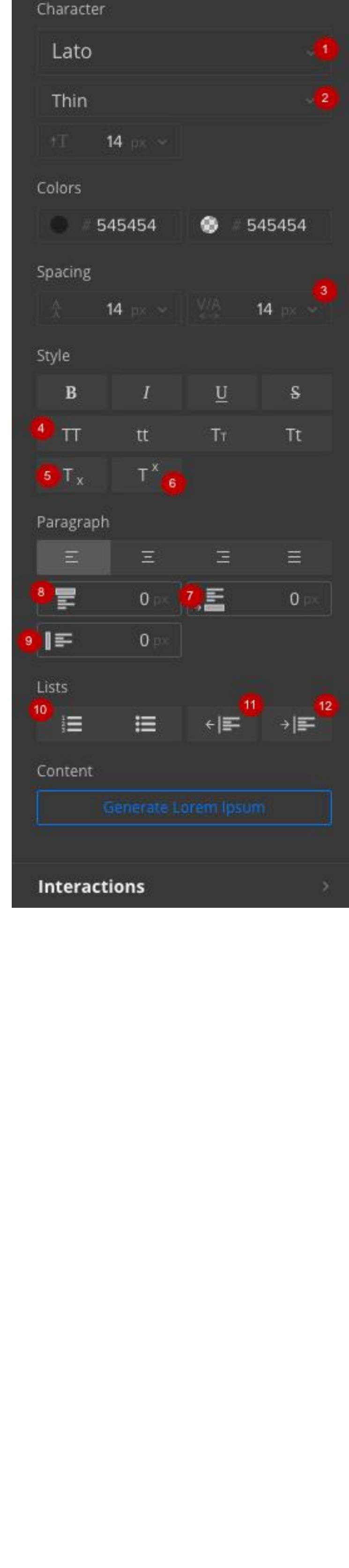
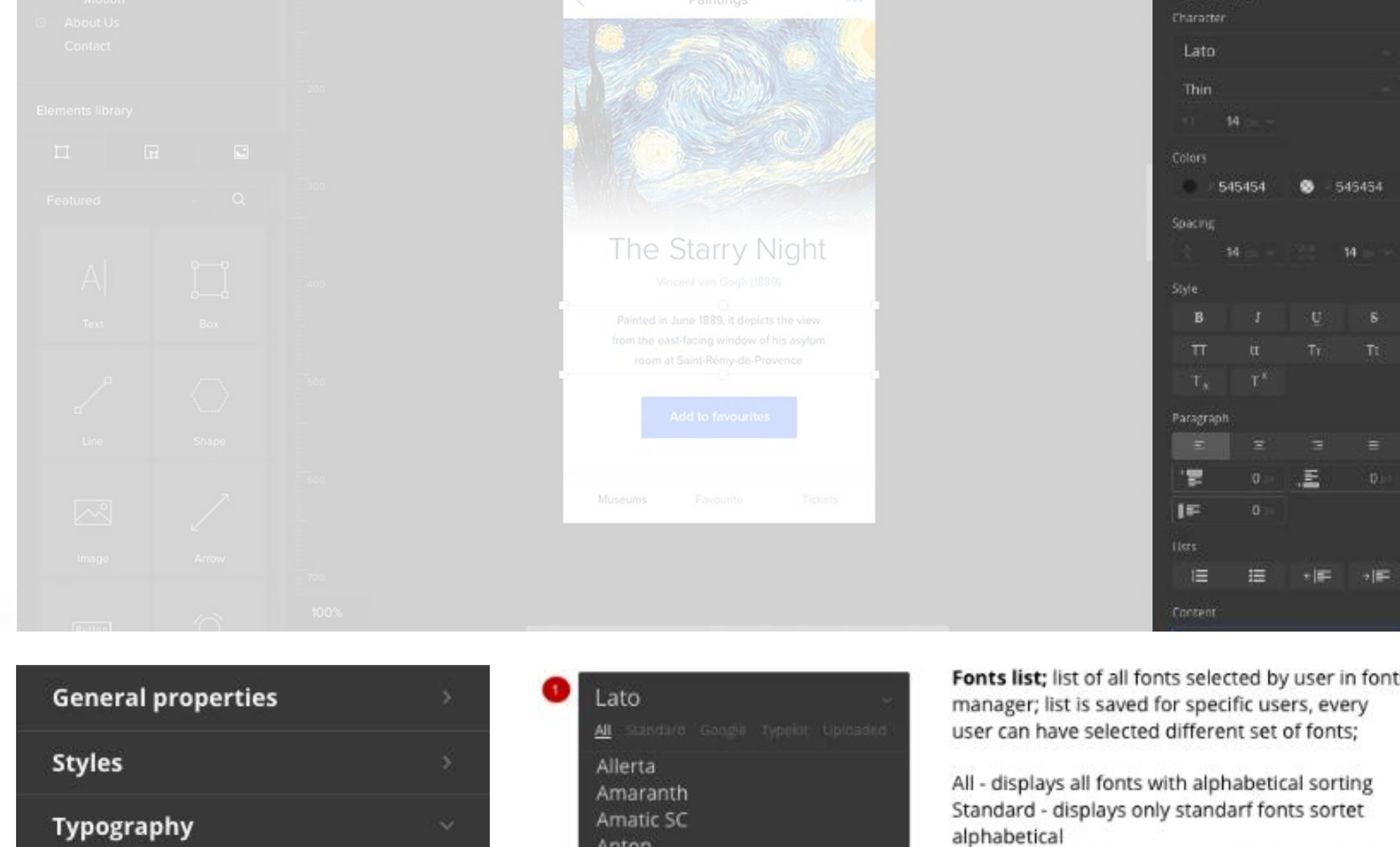


Advanced typography

The project concentrate on improve UXPin's text element. Whole work has been split into 4 releases. Text element property improvements (letter spacing, text transformations, paragraph spacing, and lists), integration with Google fonts, Typescript integration, and upload own fonts.

2016 | UXPin
role :product lead / ux design

Text element properties

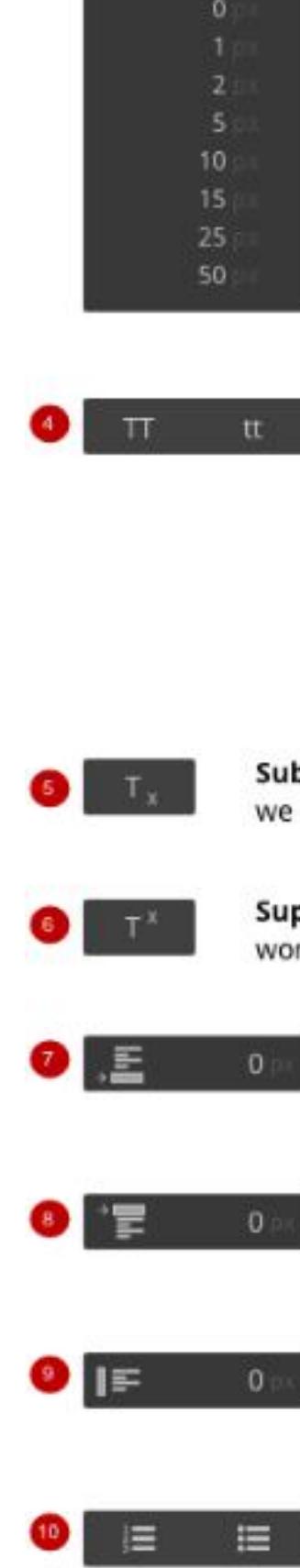


Fonst list: list of all fonts selected by user in fonts manager; list is saved for specific users, every user can have selected different set of fonts;

All - displays all fonts with alphabetical sorting
Standard - displays only standarf fonts sorted alphabetical
Google - displays onlu fonts from google fonts
Typekit - displays only fonts from typekit
Uploaded - displays fonts uploaded by user

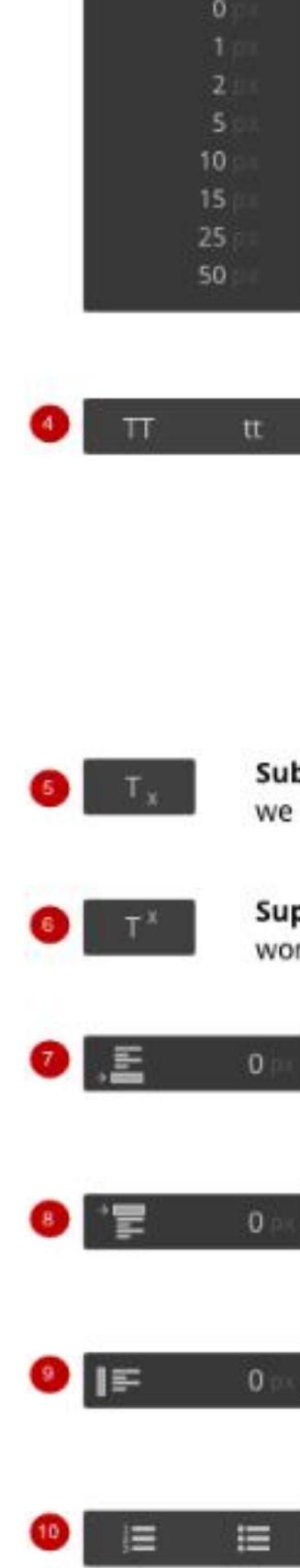
Selected category in list should be saved

By click in input with name of font you can type font name and search it on list.
(ex. <http://recordit.co/Zsi1ULnHKa>)



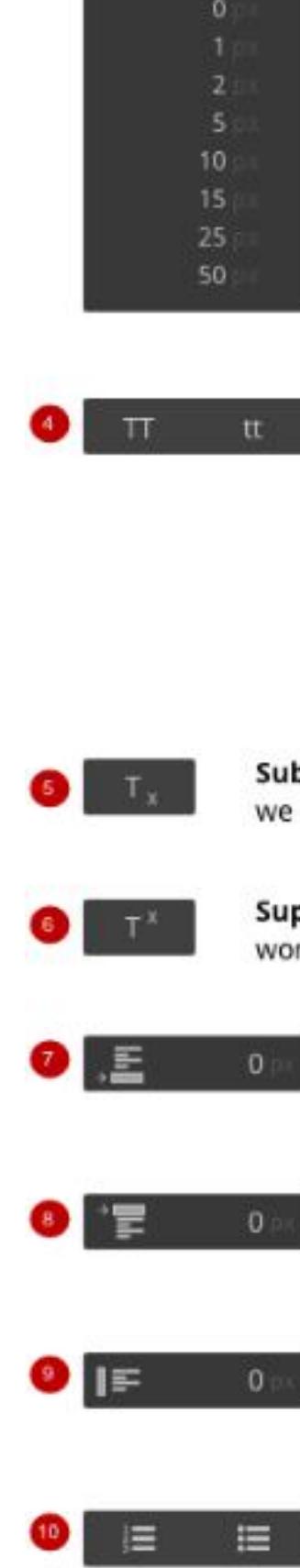
Font styles: list based on styles of font and styles added by user

Regular is default; if Regular doesn't exist first rom top is default



Letters spacing: space between letters in words; based on css property *letter-spacing*

0 is default
0 = auto

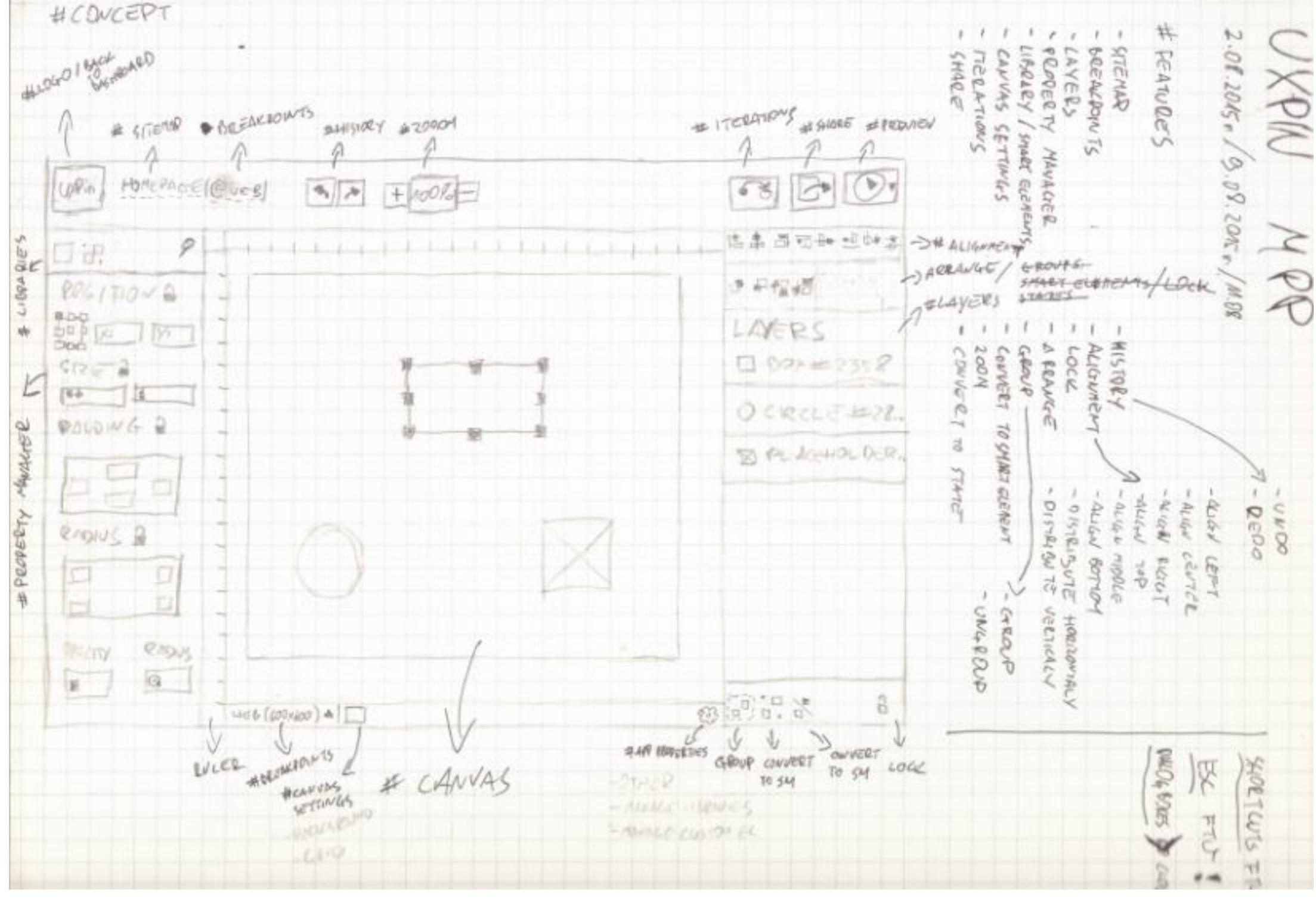


UXPin 3.0

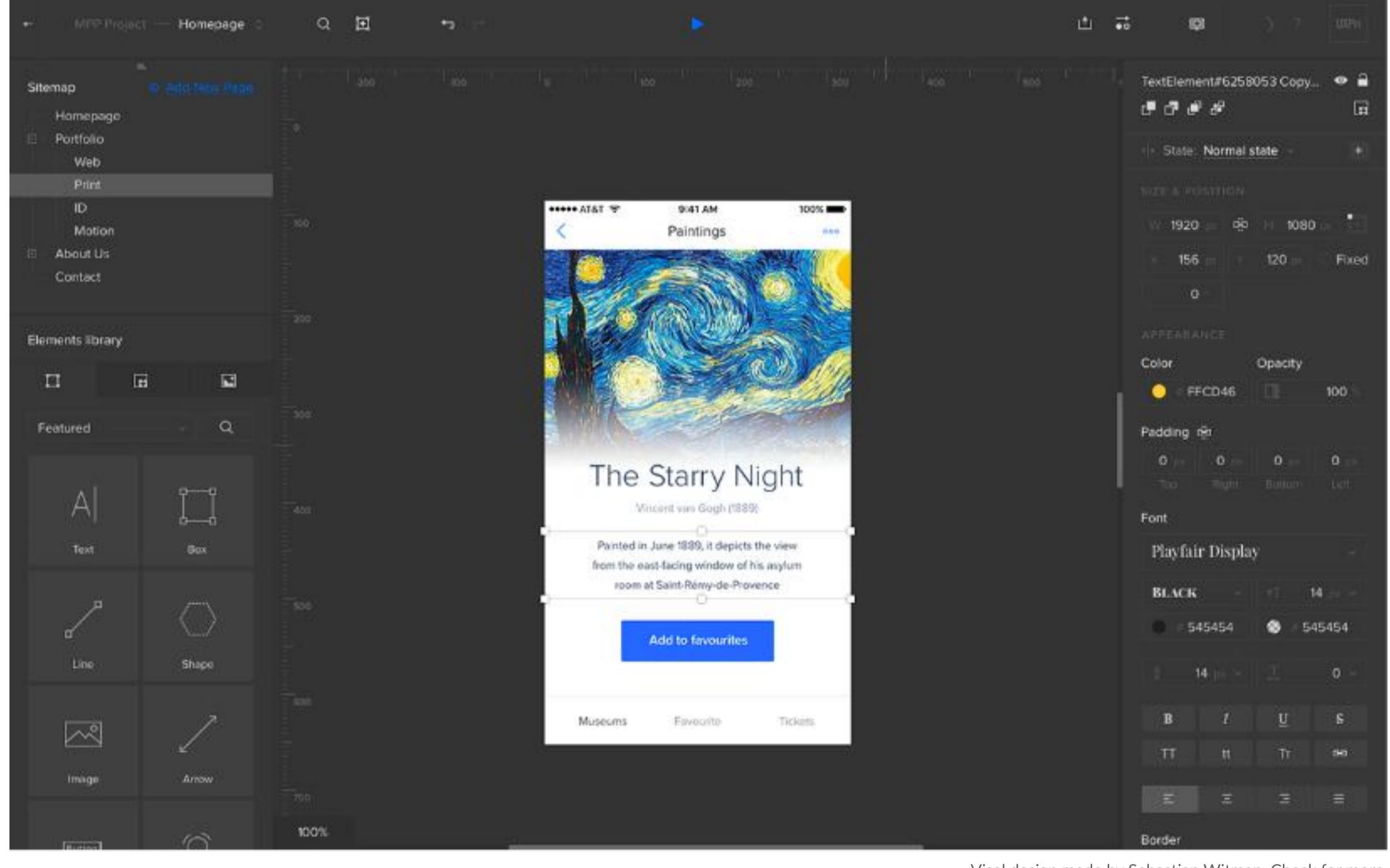
The project started by 1 email sent by CEO to some folks from the product team - “MPD - Minimum Pride Product”. Then it happened. We started working on a new interface for the UXPin’s editor. The time was perfect because we were rewriting UXPin’s heart - the engine. Collectively we designed the new editor. I was a product lead who was leading a team in the development. Besides that, I was analyzing beta User’s feedback and creating scope for future sprints.

2015 | UXPin
role :product lead / ux design + code

Te one of conners for the MPP Editor.



Final design for UXPin 3.0



Visal design made by Sebastian Witman. [Check for more.](#)