

Selected projects 2015-2023

Bartosz Dębicki

mobile :+48 508 343 705

email :bartosz.debicki@zoin.pl

linkedin :/[in/bartoszdebicki](#)

github :/[bdebicki](#)

behance :/[earldebicki](#)

Merge properties panel unification

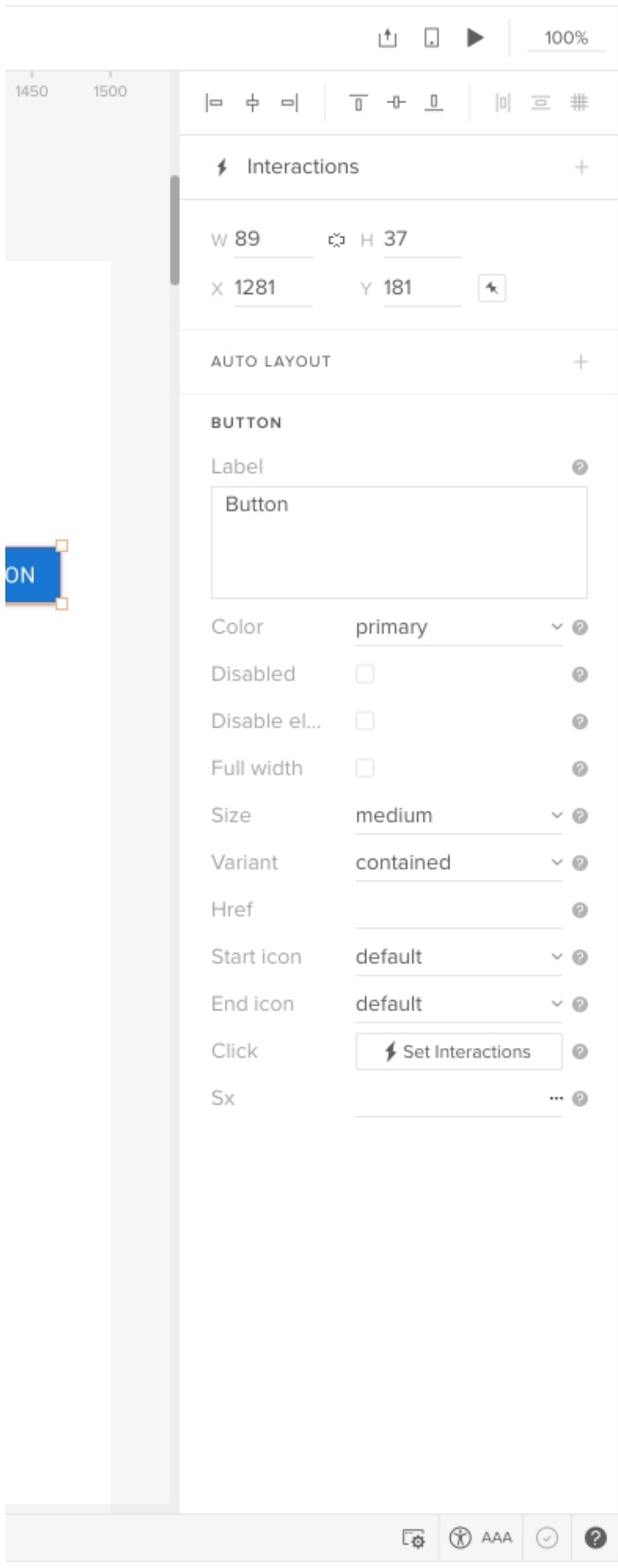
From feedback from our users, we find out that properties panel for merge components is not clear to them. The biggest issue was that they had to learn new, different concept than for regular components.

On the other hand we would like to create experience that makes feel that Merge components are something natural and consistent with UXPin editor.

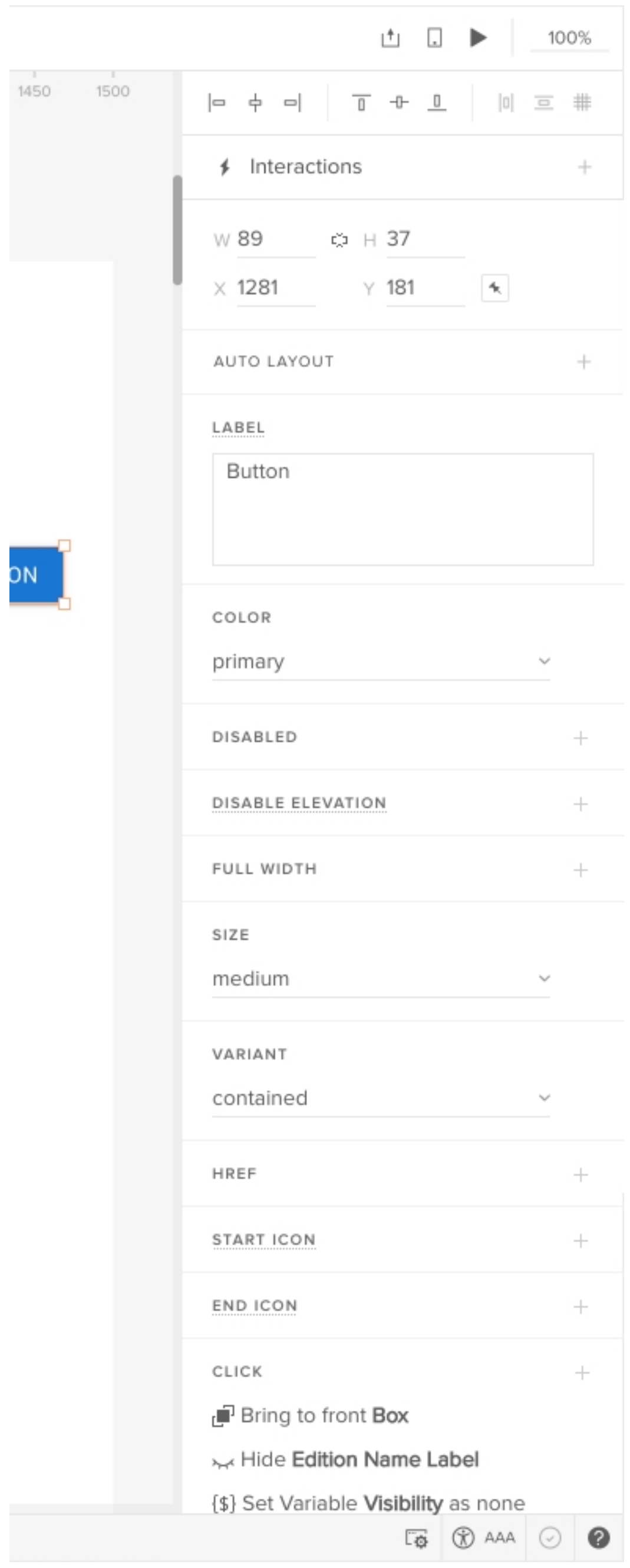
We decided to unify properties panel and align Merge components to use same concept as UXPin's classic elements.

2023 | UXPin
role :design

Previous and unified properties panel



old Merge components properties panel



unified Merge components properties panel

Merge components sections states

collapsed

null / undefined / "" (string) / false (boolean) property value

expanded

with value / true (boolean) property

SECTIONS

textfield control
JSONNodePropertyHandler
NodePropertyHandler
TextFieldPropertyHandler

LABEL +

LABEL
Button

click on plus: showing textarea and focus in field
click on clear: clean property value and hide textarea

select control
UnionLiteralPropertyHandler

COLOR +

COLOR
primary

click on plus: showing select, should be open and display available options
click on clear: clean property value and hide select

boolean control
BooleanPropertyHandler
BooleanSwitcherPropertyHandler

DISABLED +

DISABLED
True False

click on plus: show switcher with selected true value
click on clear: set up false value and hide switcher

string control
JSONStringPropertyHandler
StringPropertyHandler
NumberPropertyHandler (with InputMinMax)

HREF +

HREF

click on plus: show input and focus field
click on clear: clean up value and hide input

function control
FunctionPropertyHandler

CLICK +

CLICK +
Bring to front Box
Hide Edition Name Label
{s} Set Variable Visibility as none

click on plus: opens add new interaction form
click on interaction: opens edit interaction form
click on clear: remove interaction

code control
CodePropertyHandler

SX +

SX
...

click on plus: show code control with opened code editor
click on clear: clean up value and hide control

color control
ColorPropertyHandler

COLOR +

COLOR
#FFC7C7

click on plus: show color control with opened color picker
click on clear: clean up value and hide control

PROPERTY HAS DESCRIPTION

DISABLE ELEVATION +

DISABLE ELEVATION
True False

if property has description:
- property heading has dotted underline
- cursor when hovering over a heading is question mark (the lp)
- description is displayed in a Tooltip component; default position of the tooltip should be top

DEFAULT VALUE HAS BEEN OVERRITTEN

default is empty

TITLE
Toggle element visibility

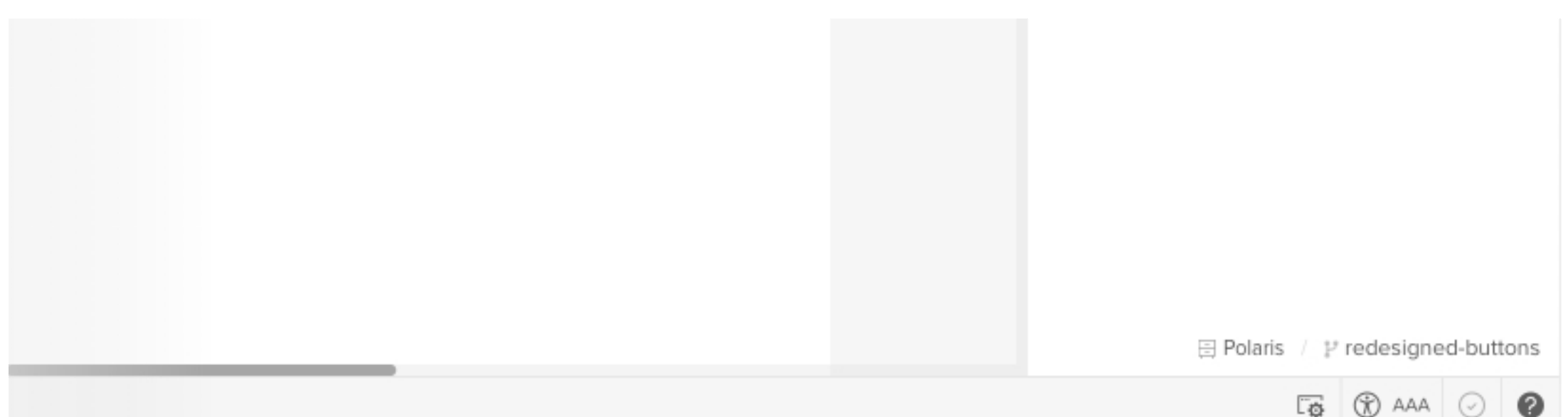
after click in reset icon:
- value should be reset to default
- control could be collapse if there was no value or default value was false (for boolean)

default has value

ERROR MESSAGE +

after click in reset icon section should be expanded with default value

Information about component's library in properties panel



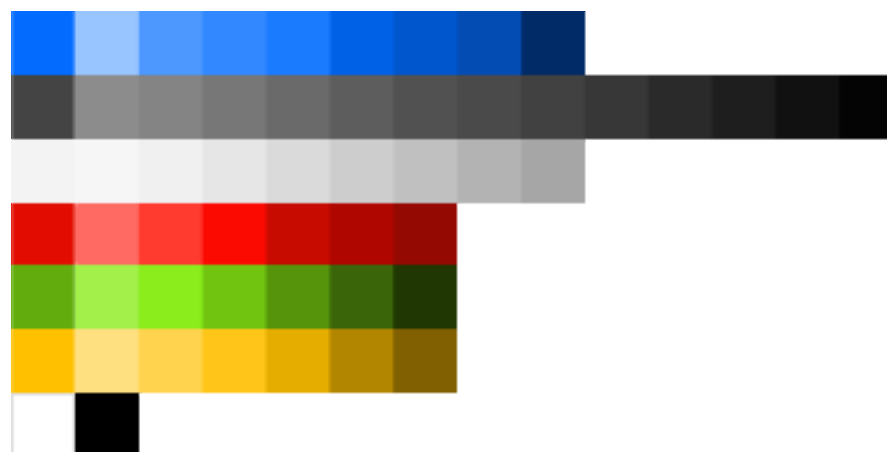
UXPin Design System

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

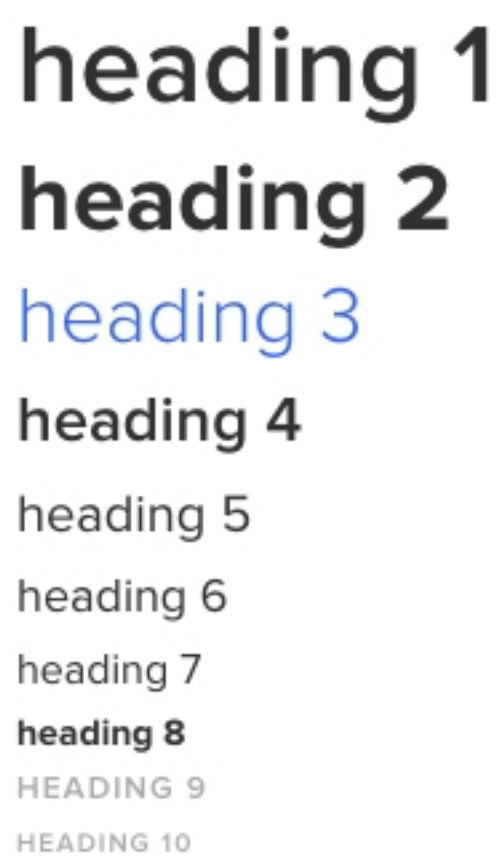
color palette

Based on colors used in the UXPin application I've prepared palette 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.



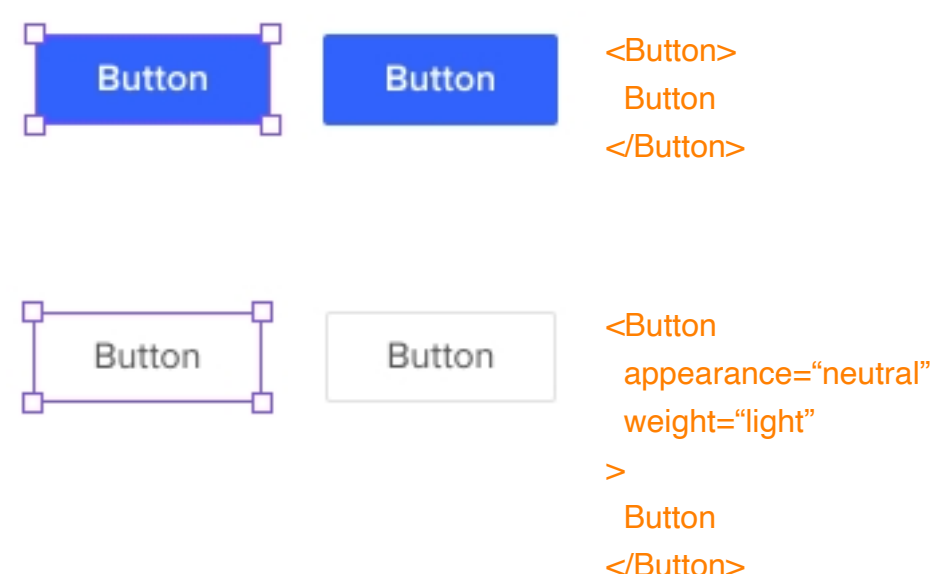
icons

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

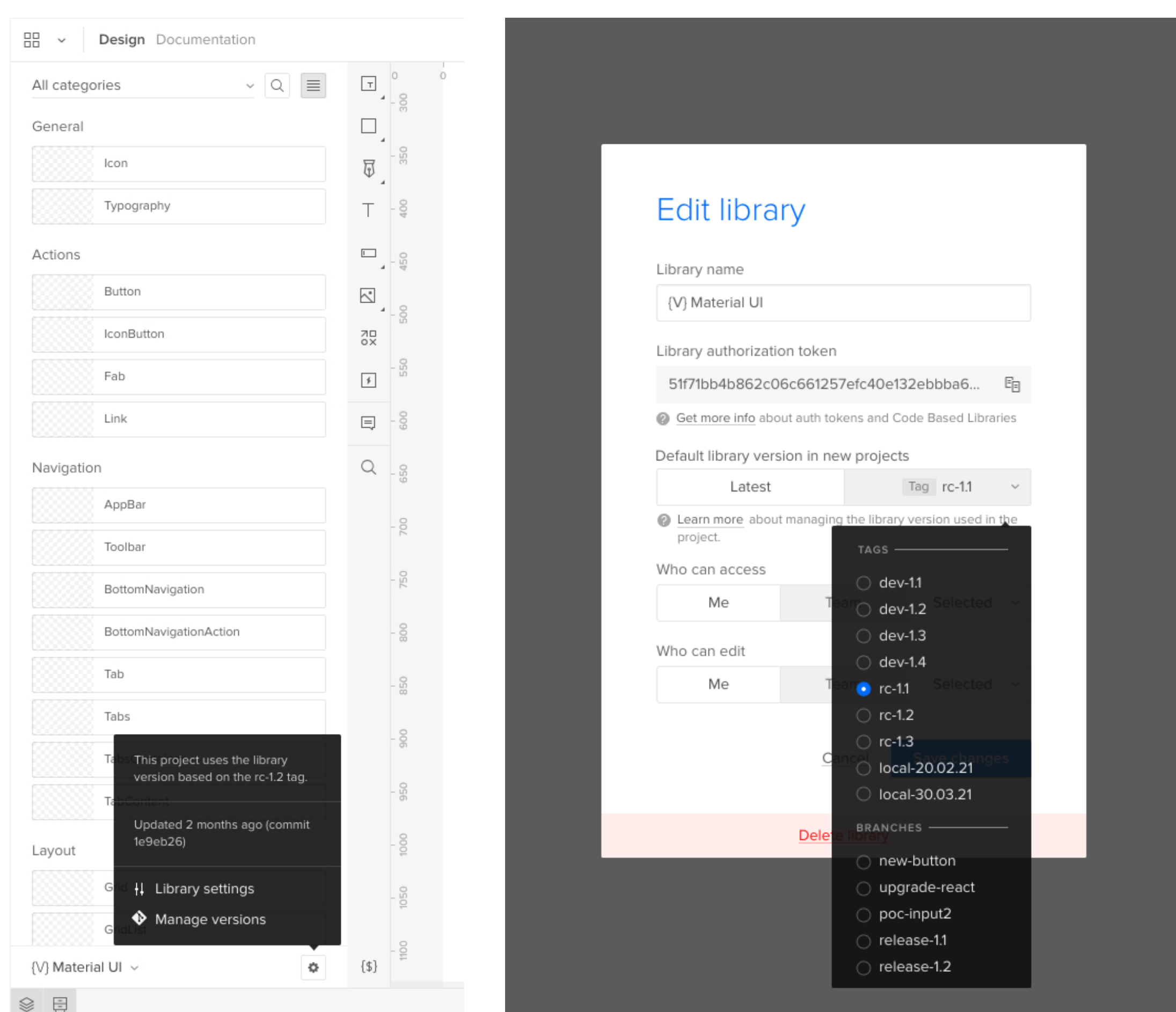


Merge Libraries Version Control

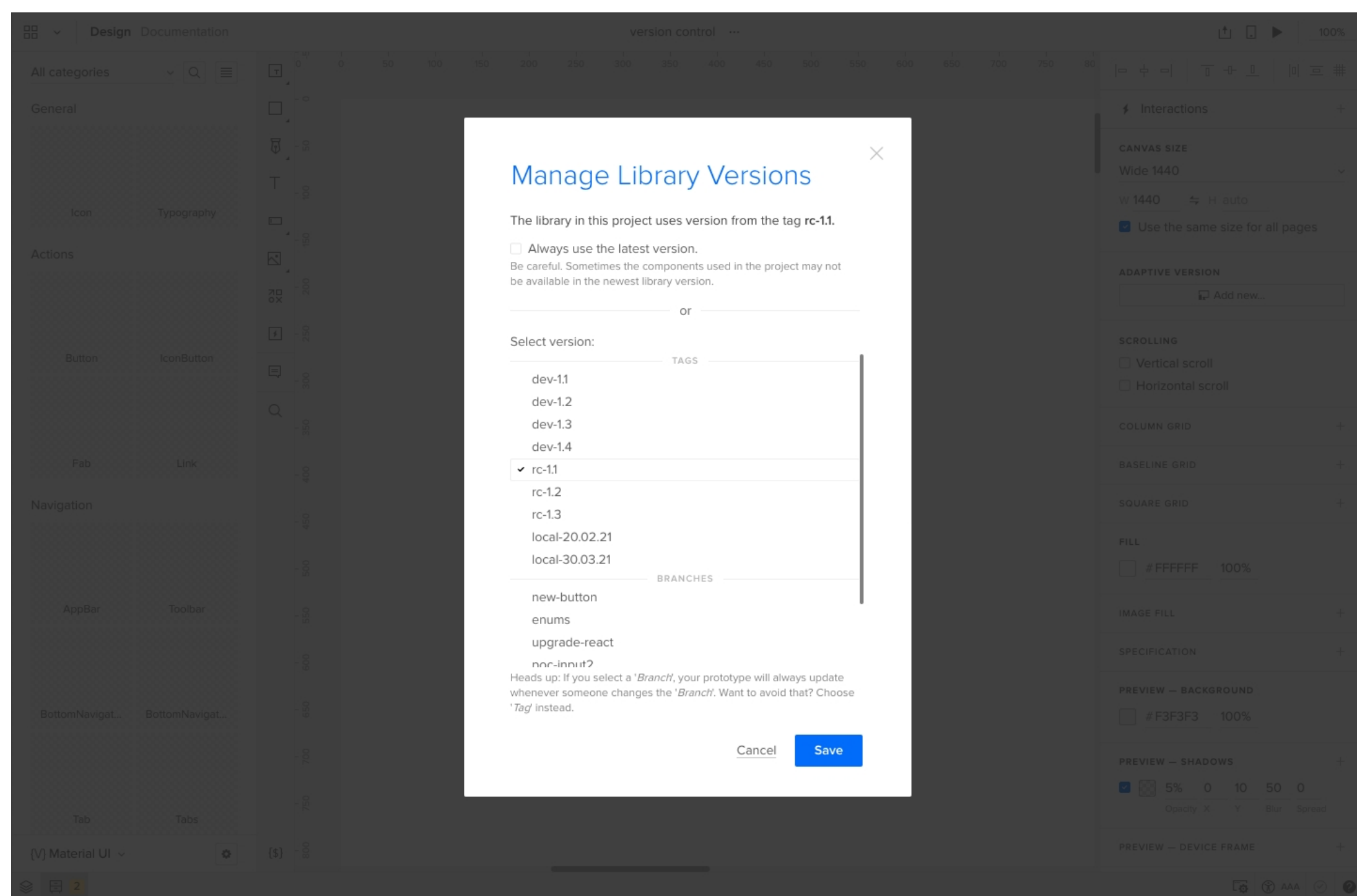
Based of feedback from our users, how they are implementing Merge to their organisation we realise there's need to handle multiple versions of the Merge library. Some cases was working on redesign, some was refactoring. Different projects might need different versions on components for testing purposes.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

2021 | UXPin
role :code + design

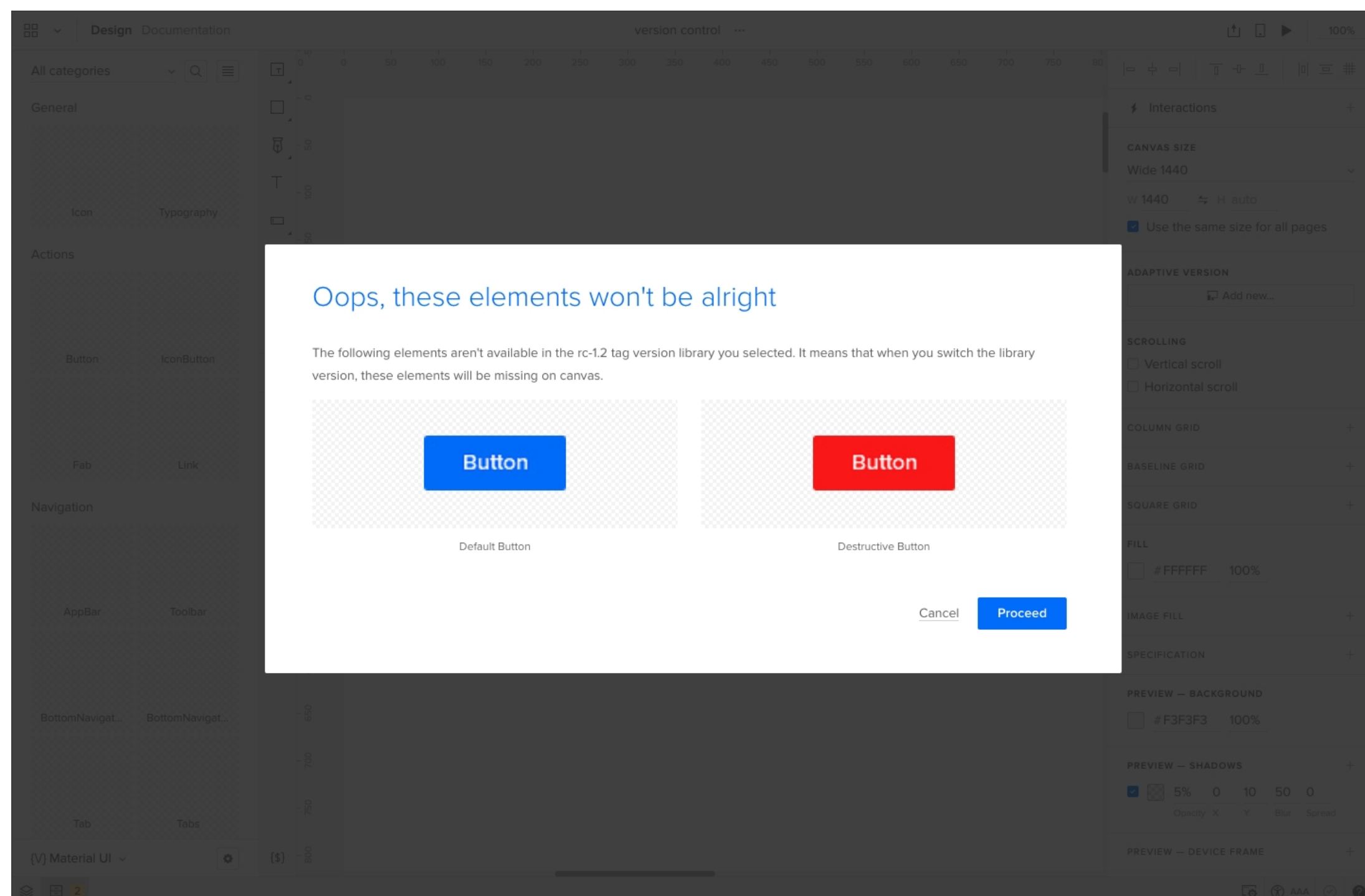
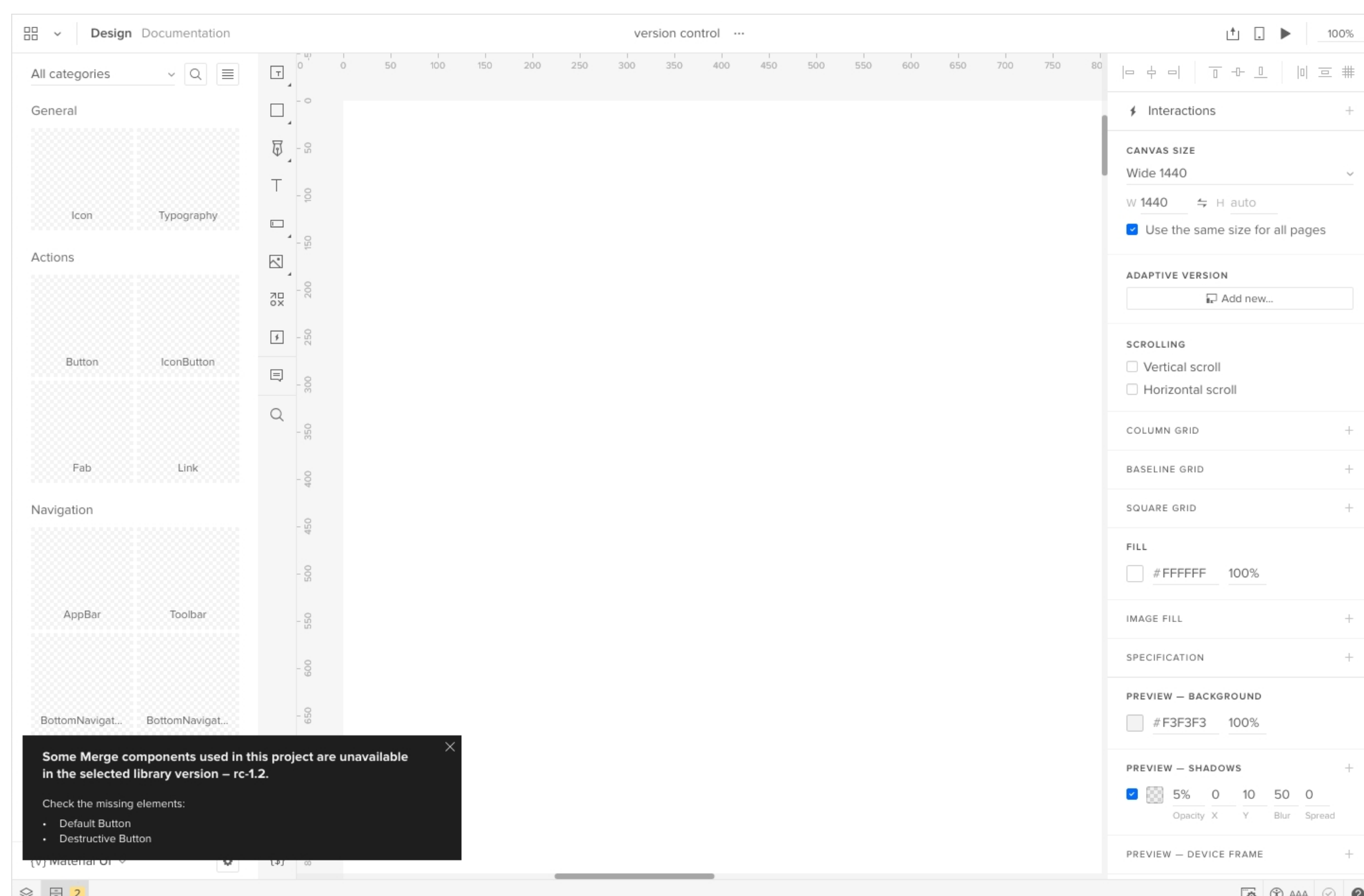
seeing default version of the library for all projects



setting library version on project level



version conflicts



Adele Design System 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

Adele by UXPin



The Github design system is called Primer

website <https://primer.github.io/>
repository <https://github.com/primer/primer/>

WHAT DOES GITHUB'S DESIGN SYSTEM INCLUDE?

Technology

Code Depth	HTML/CSS	<input type="checkbox"/>
Components	✓	
JS Library/Framework	×	
Typescript	×	
Web Components	×	
Tests	AVA	
Linters	StyleLint	
CSS	Sass	
CSS in JS	×	
Design Tokens	Scss	
Bundle Manage	Primer Module Build Script	
Distribution	npm	

Design

UI Kit	×	
Brand Guidelines	×	
Color Naming	natural (e.g. gray-300)	<input type="checkbox"/>
Contrast Analysis	×	
Typography	✓ Go to Typography	
Icons	Octicons (SVG)	
Space / Grid	✓ Go to Space / Grid	
Illustration	×	
Data Visualization	×	
Animation	×	
Voice & Tone	×	

Documentation & guidelines

Accessibility Guidelines	×	
Design Principles	×	
Documentation Website	×	
Code Documentation	Markdown	<input type="checkbox"/>
Storybook	✓ Go to Storybook	

DO YOU NEED HELP WITH YOUR SYSTEM?

Evangelize Design System with a Free Template!



40+ Slides ready for action. Perfect for team and stakeholder presentation. Available as a powerpoint and keynote deck.

[Learn more & download](#)

UXPin: Prototype and Manage Your Design System



Join thousands of companies (including PayPal, Sapient and HBO) and optimize your design process with UXPin.

[Start a free trial now!](#)



2010 – 2020 UXPin INC

Maintained by Marcin Treder. Inspired by [Awesome Design Systems](#) by Alex Pate. Great design systems resources: [Styleguides.io](#)

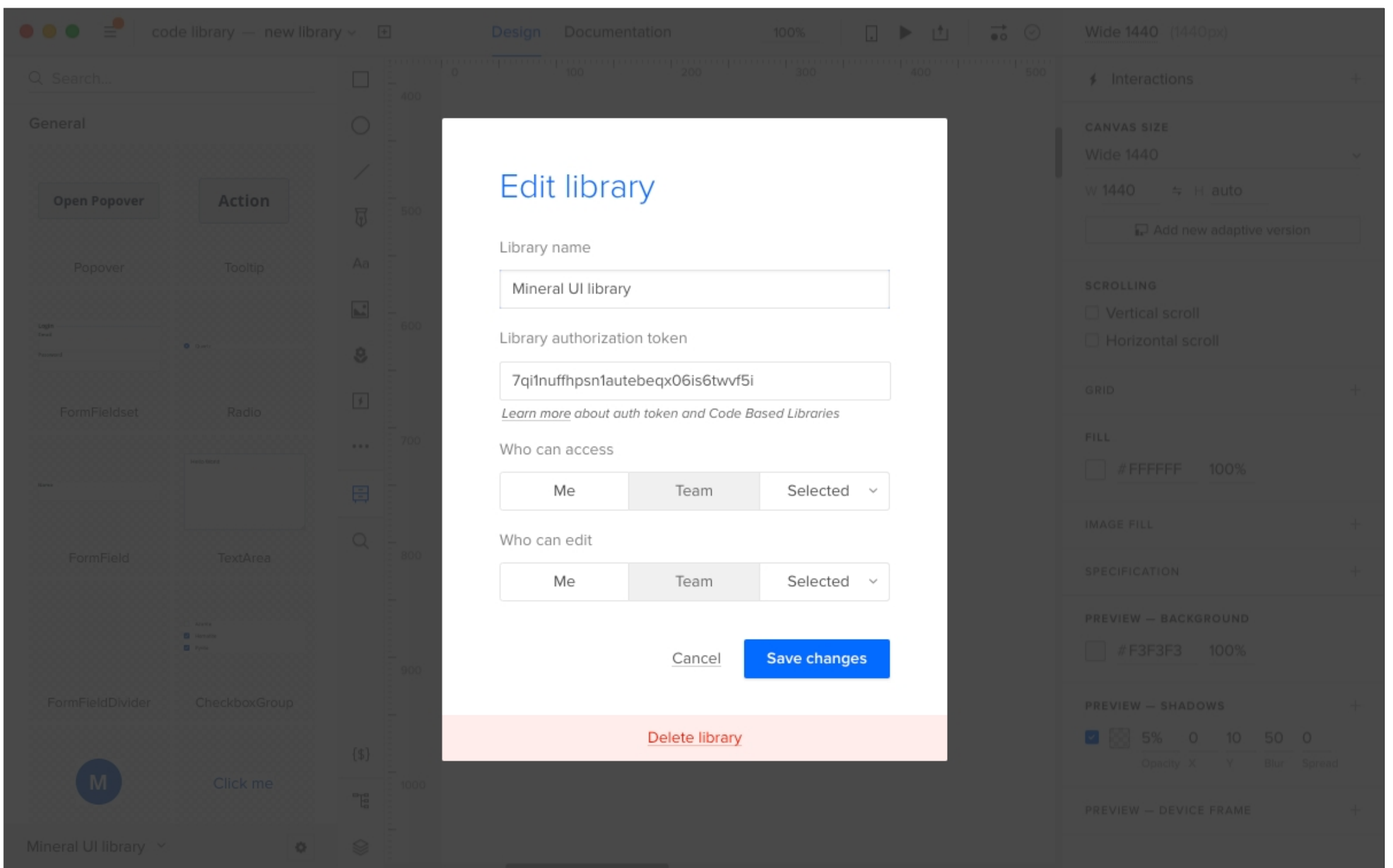
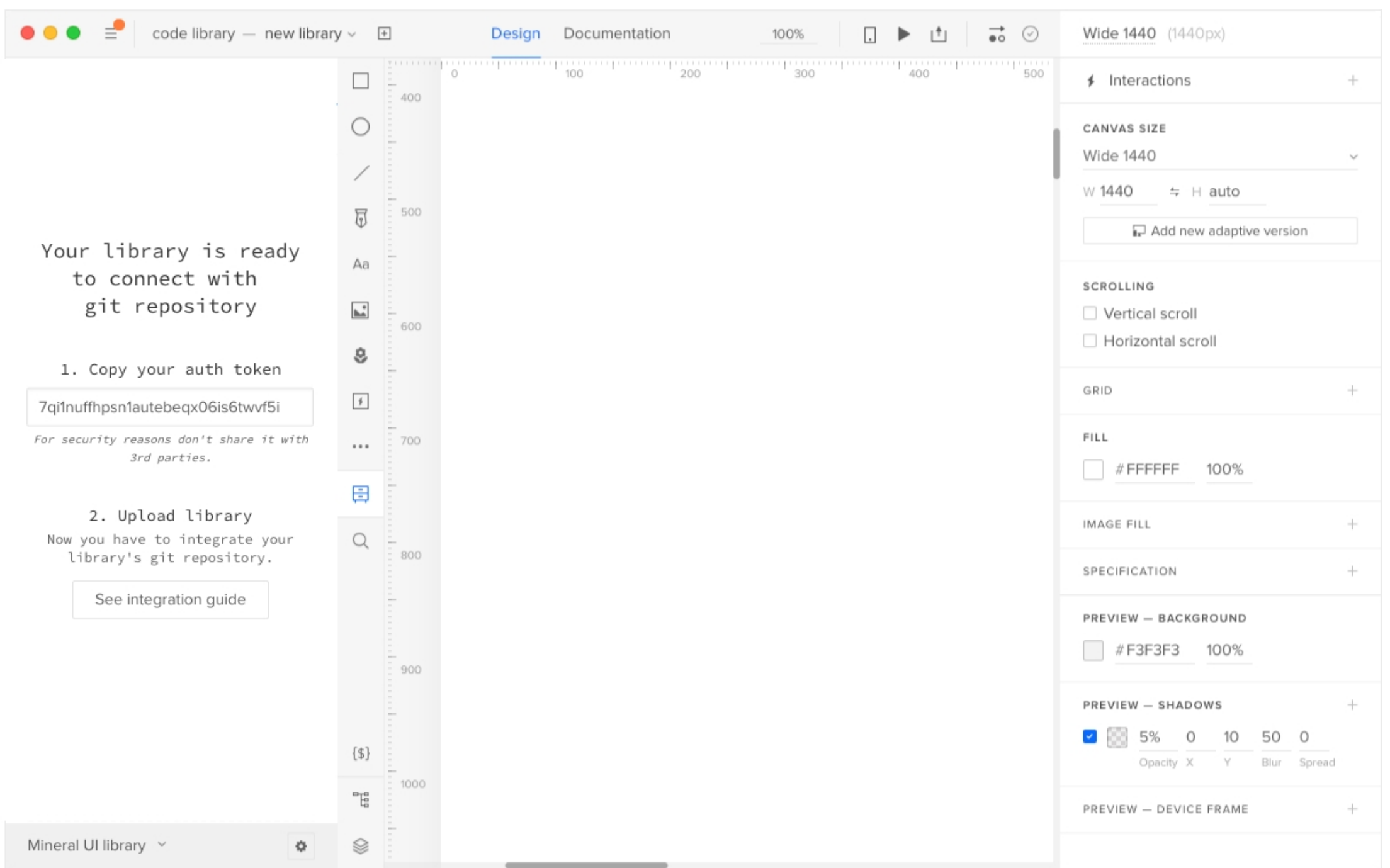
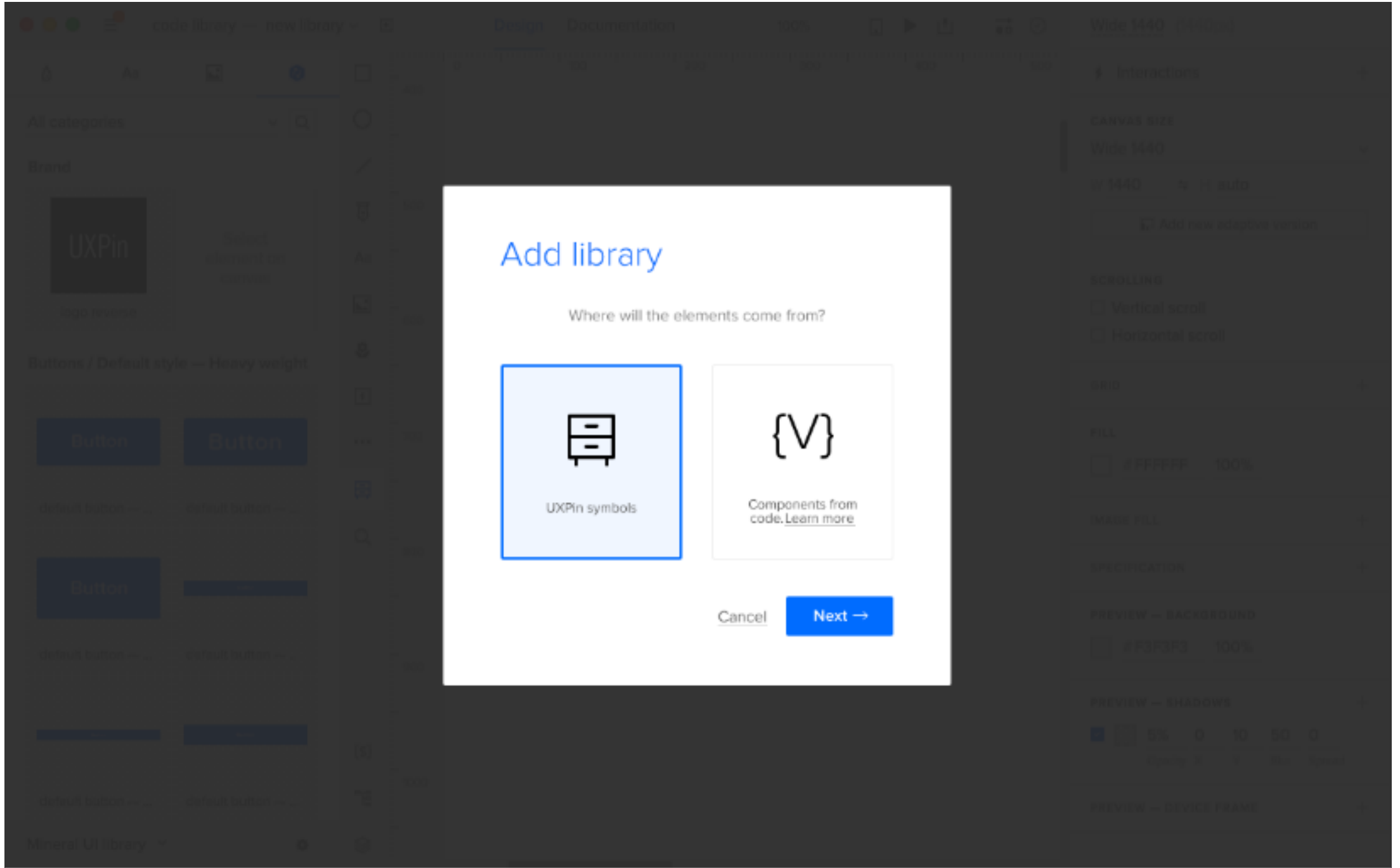
UXPin Merge

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. Check out more about {V} from one of UXPin's webinars.

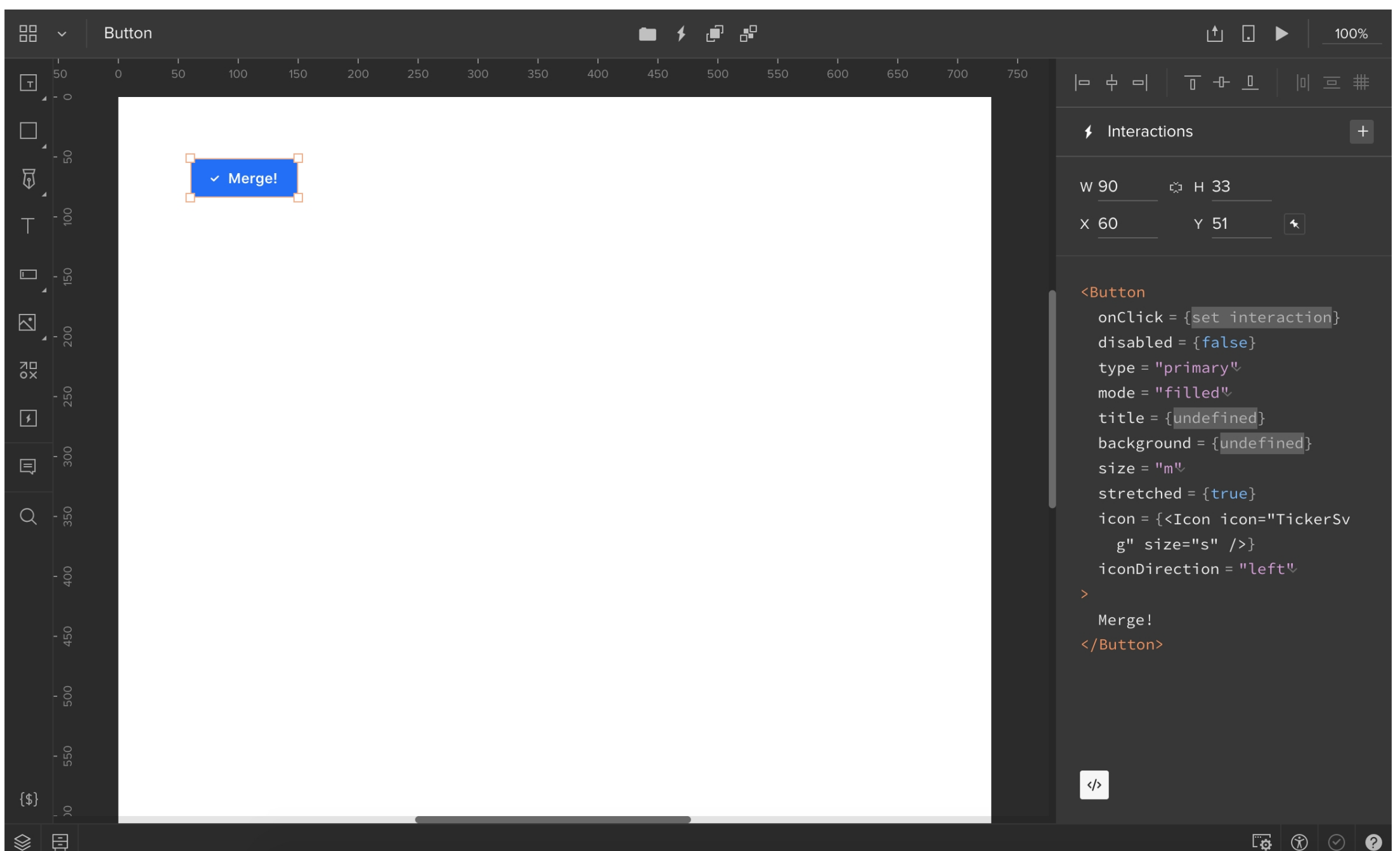
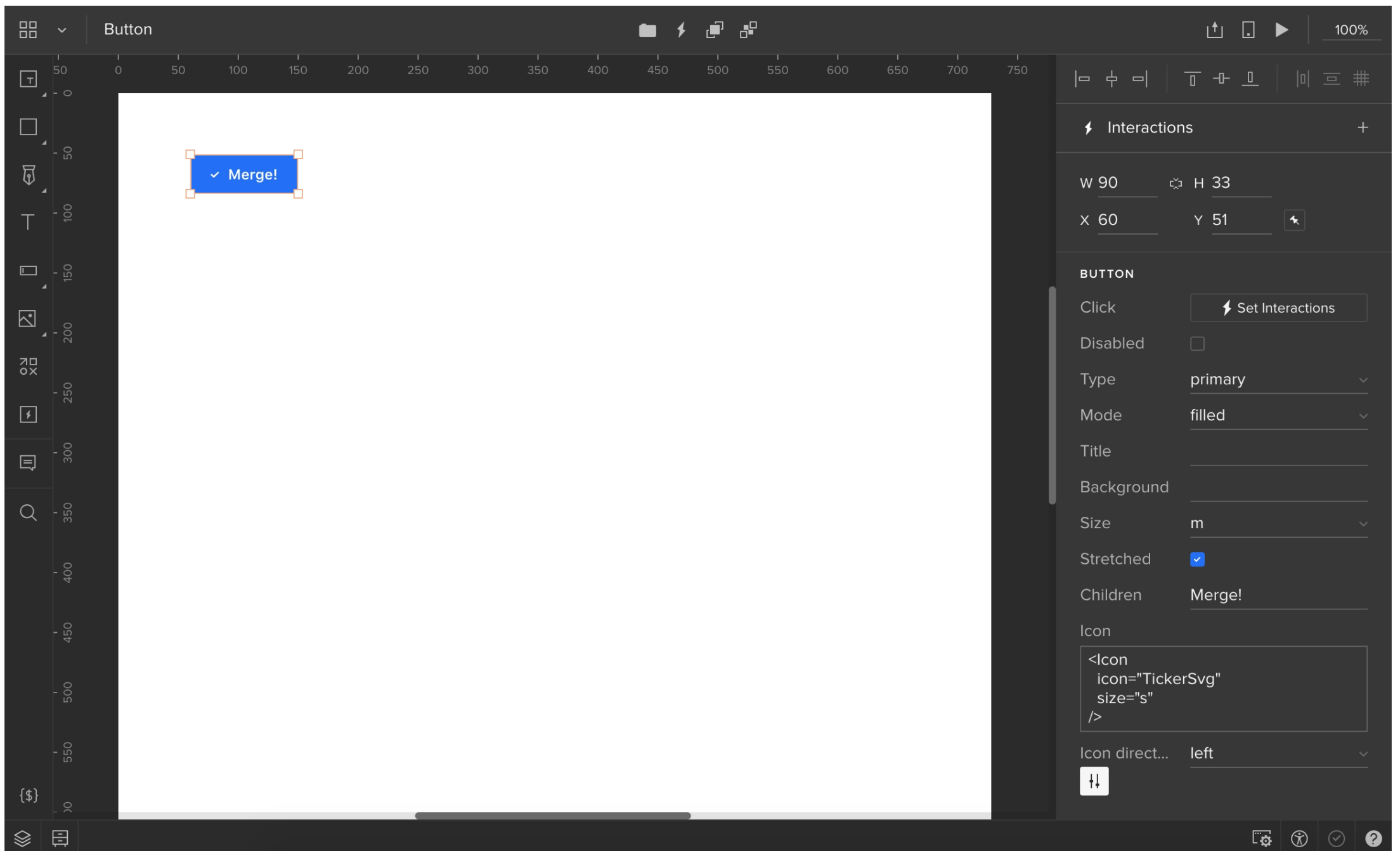
2021 | UXPin
role :product lead / ux design

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.

creating new library flow



code components properties panel

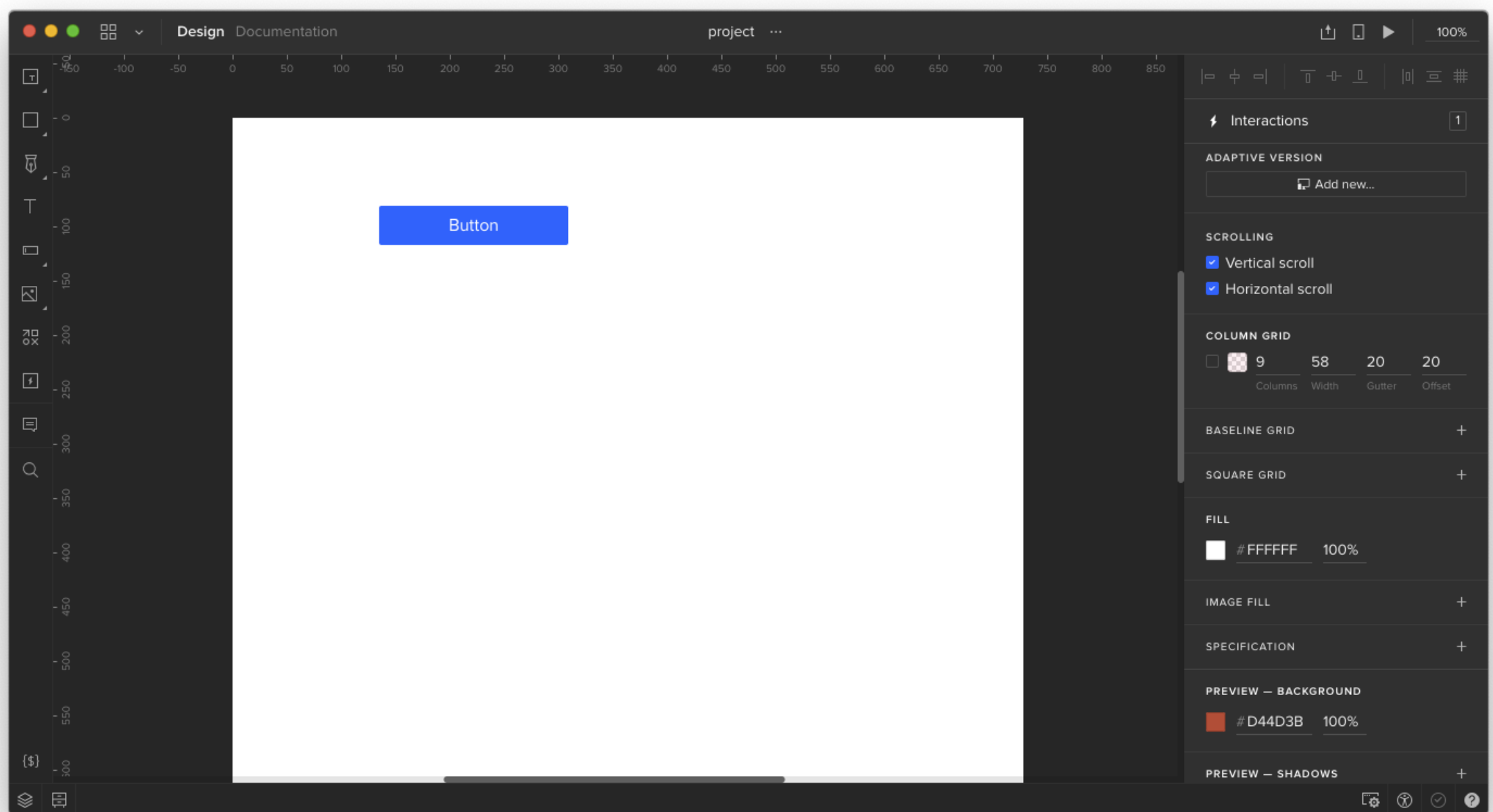


UXPin Desktop App

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

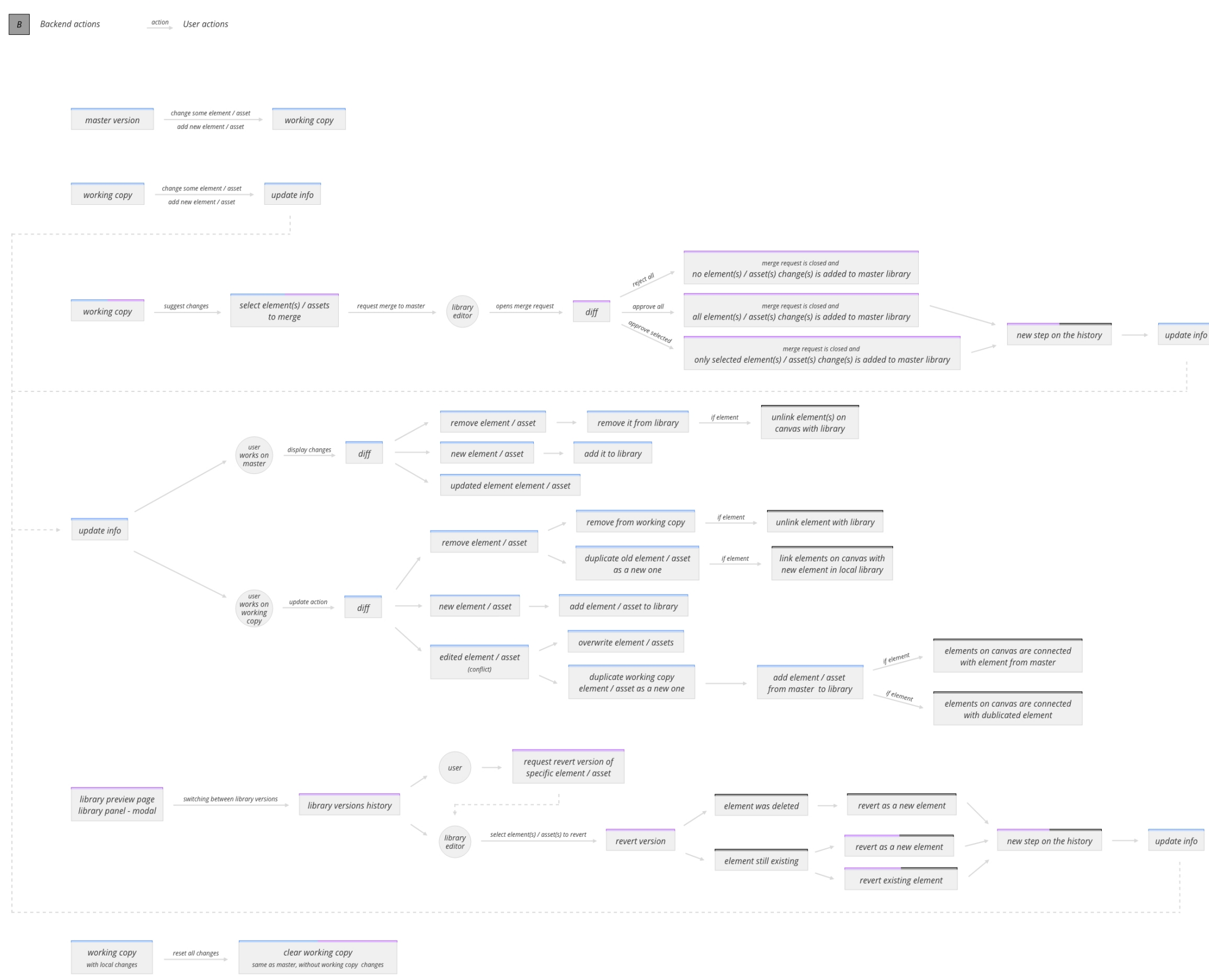


Design System version control

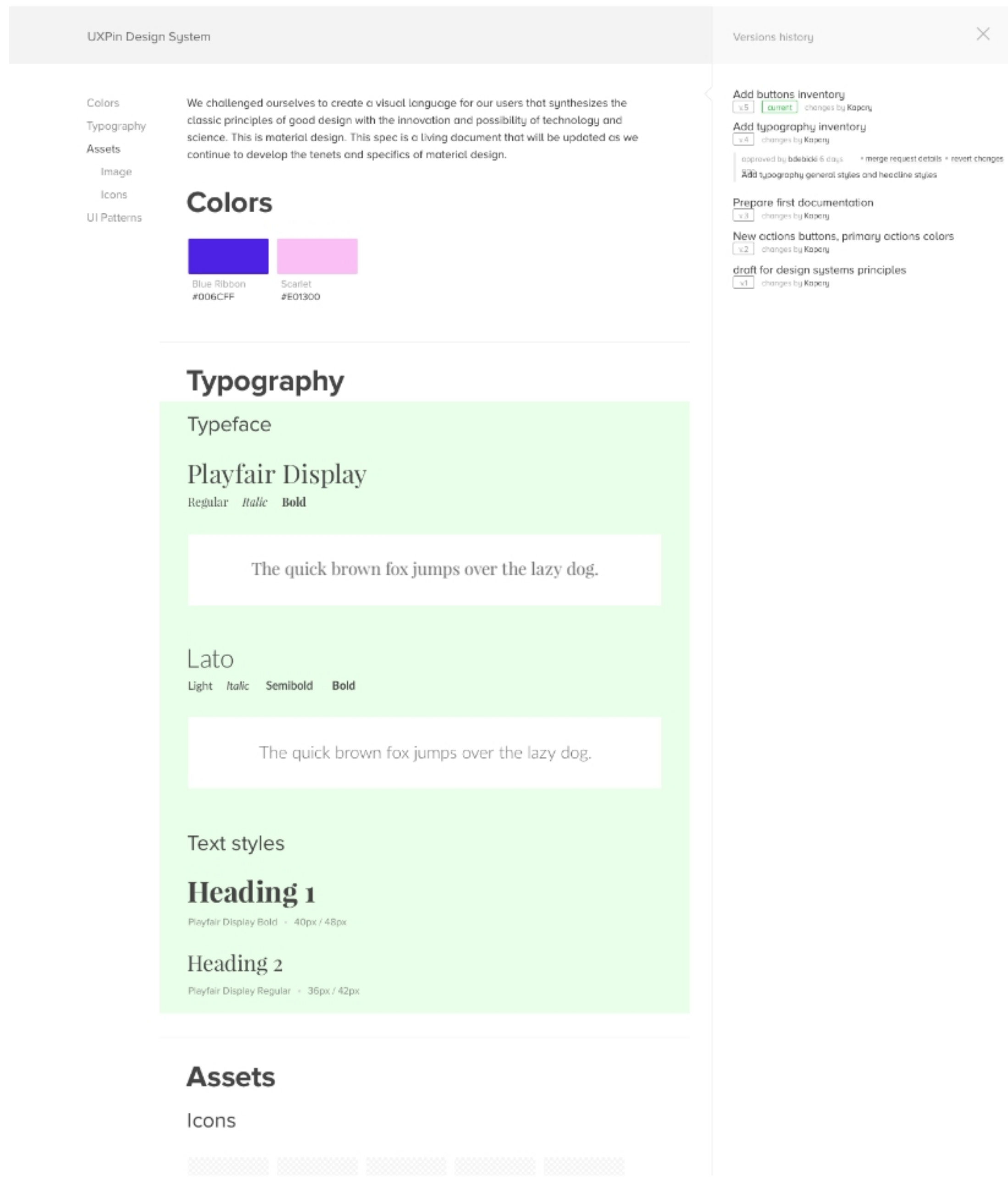
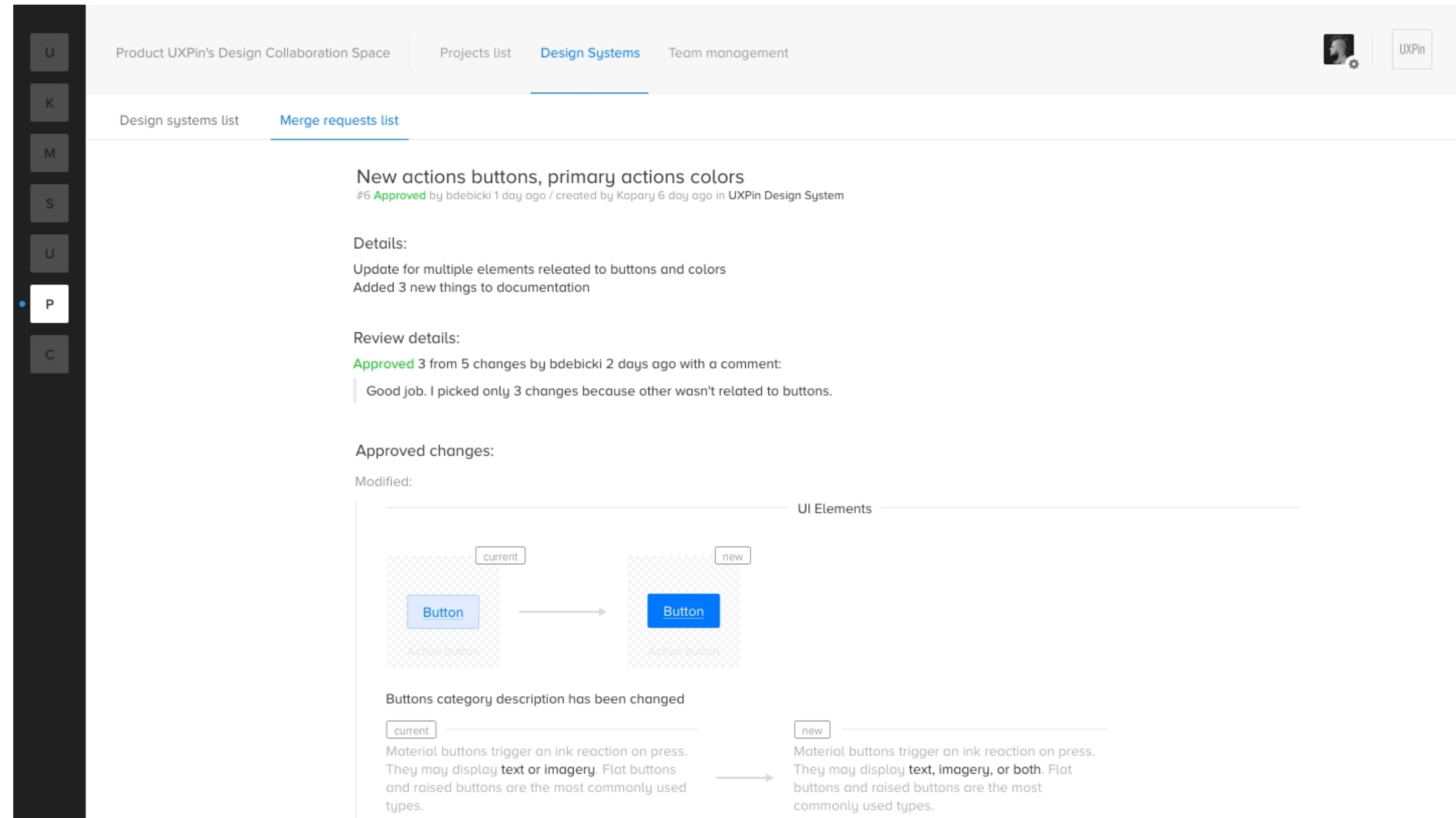
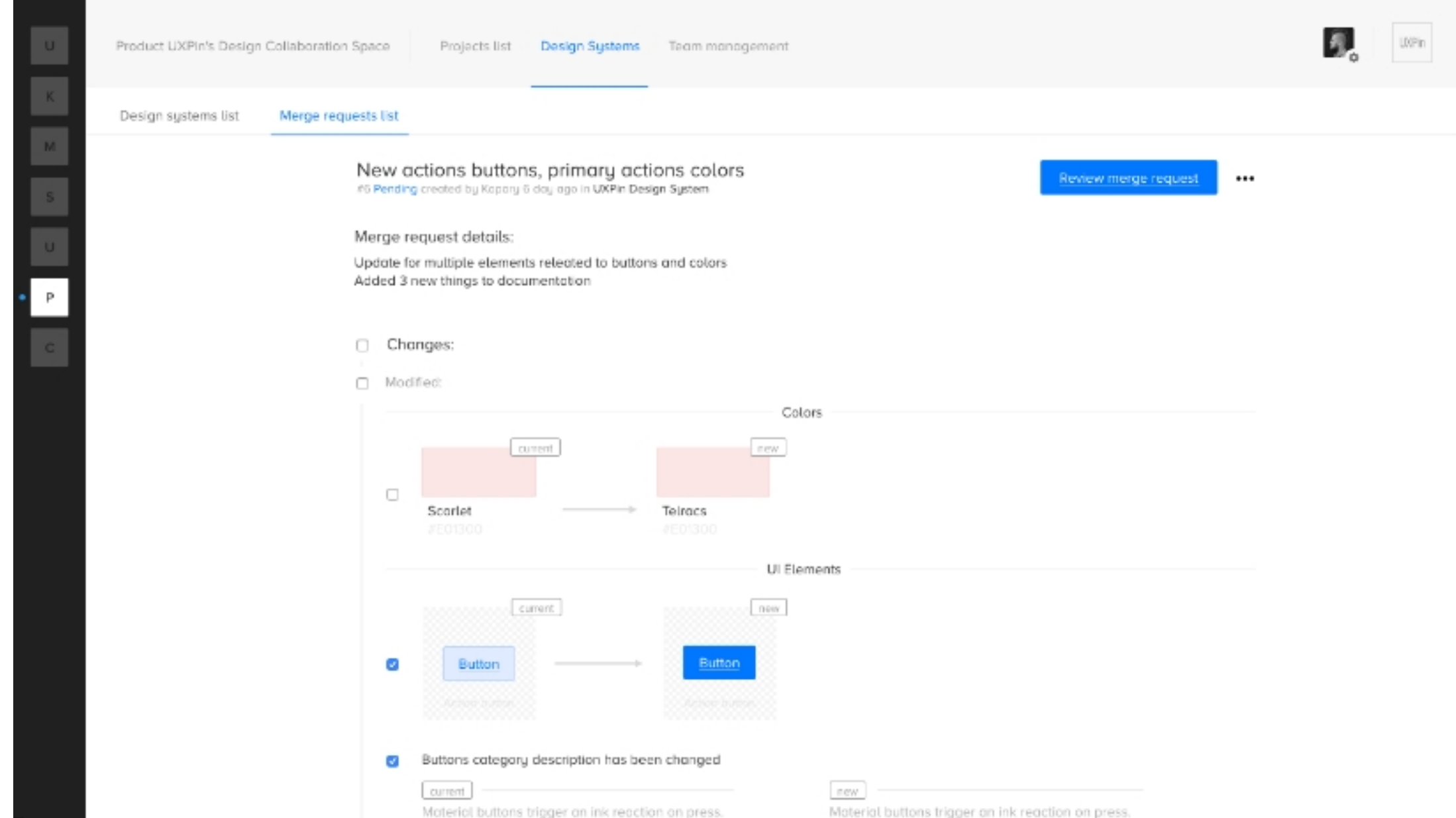
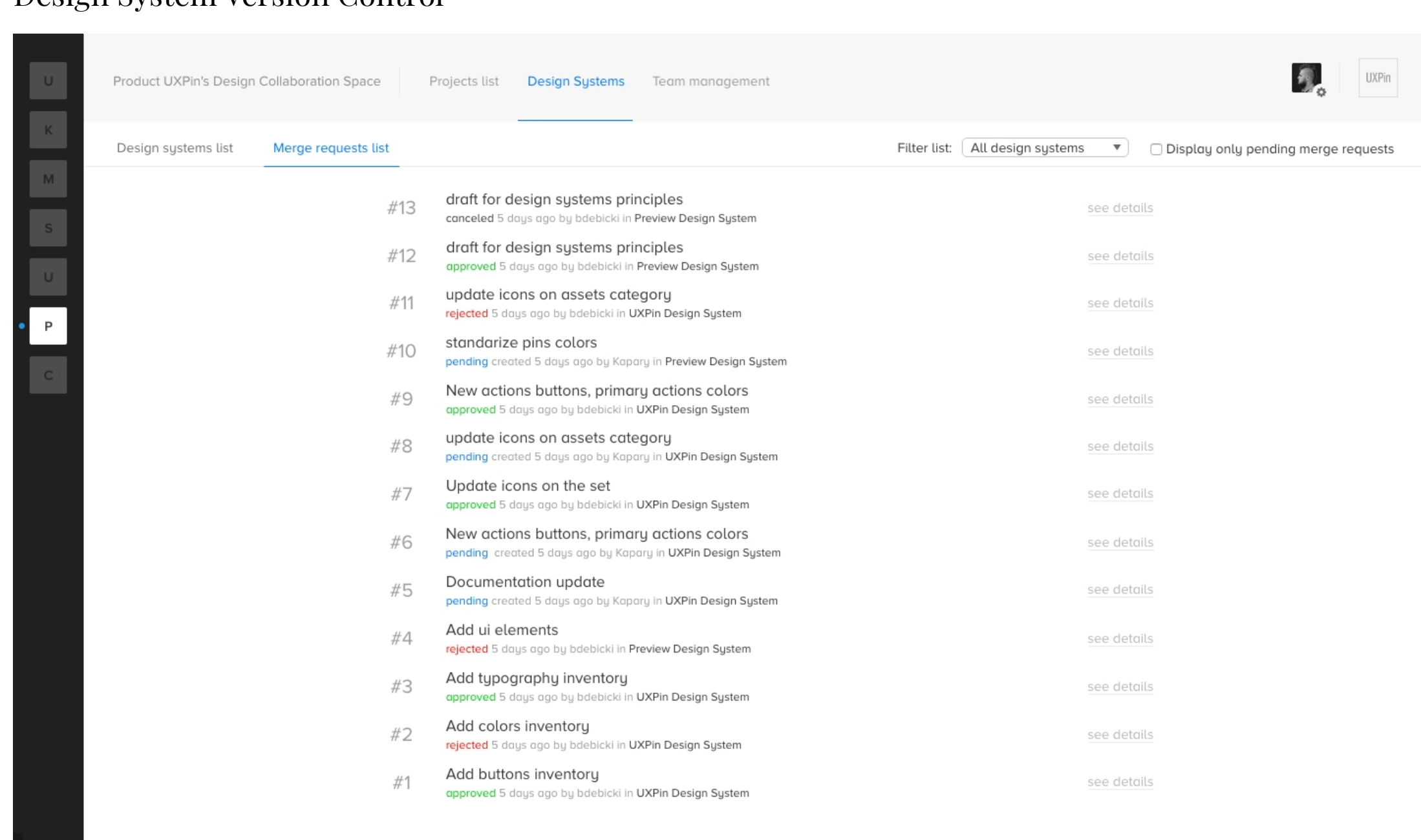
Right after the release of Design Systems and Design Libraries features in UXPin next step supposed to be version control of those areas. The project was very technical centered and complex because a lot of actions happened in backend hidden by the interface.

2017 | UXPin
role :product lead / ux design

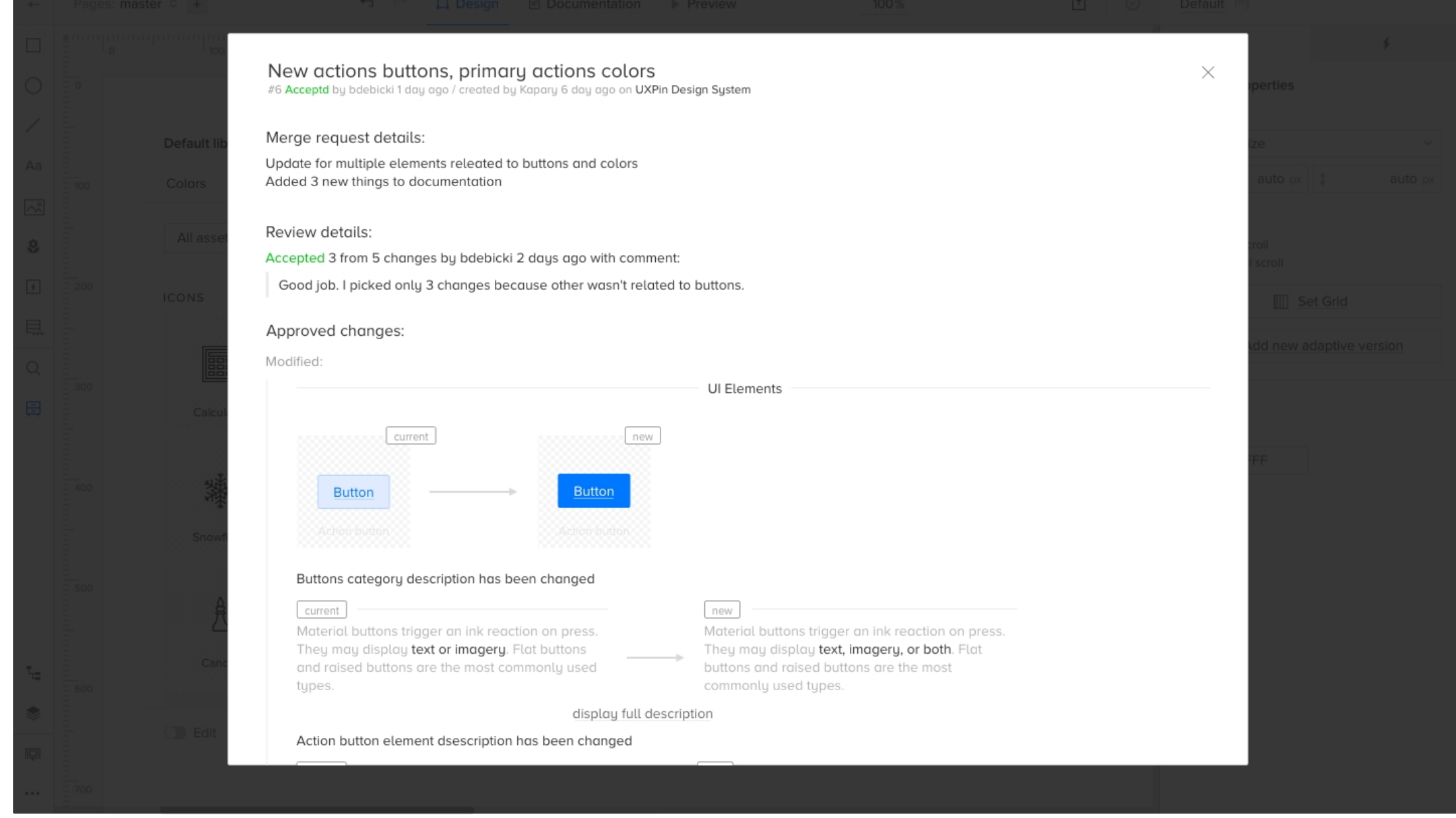
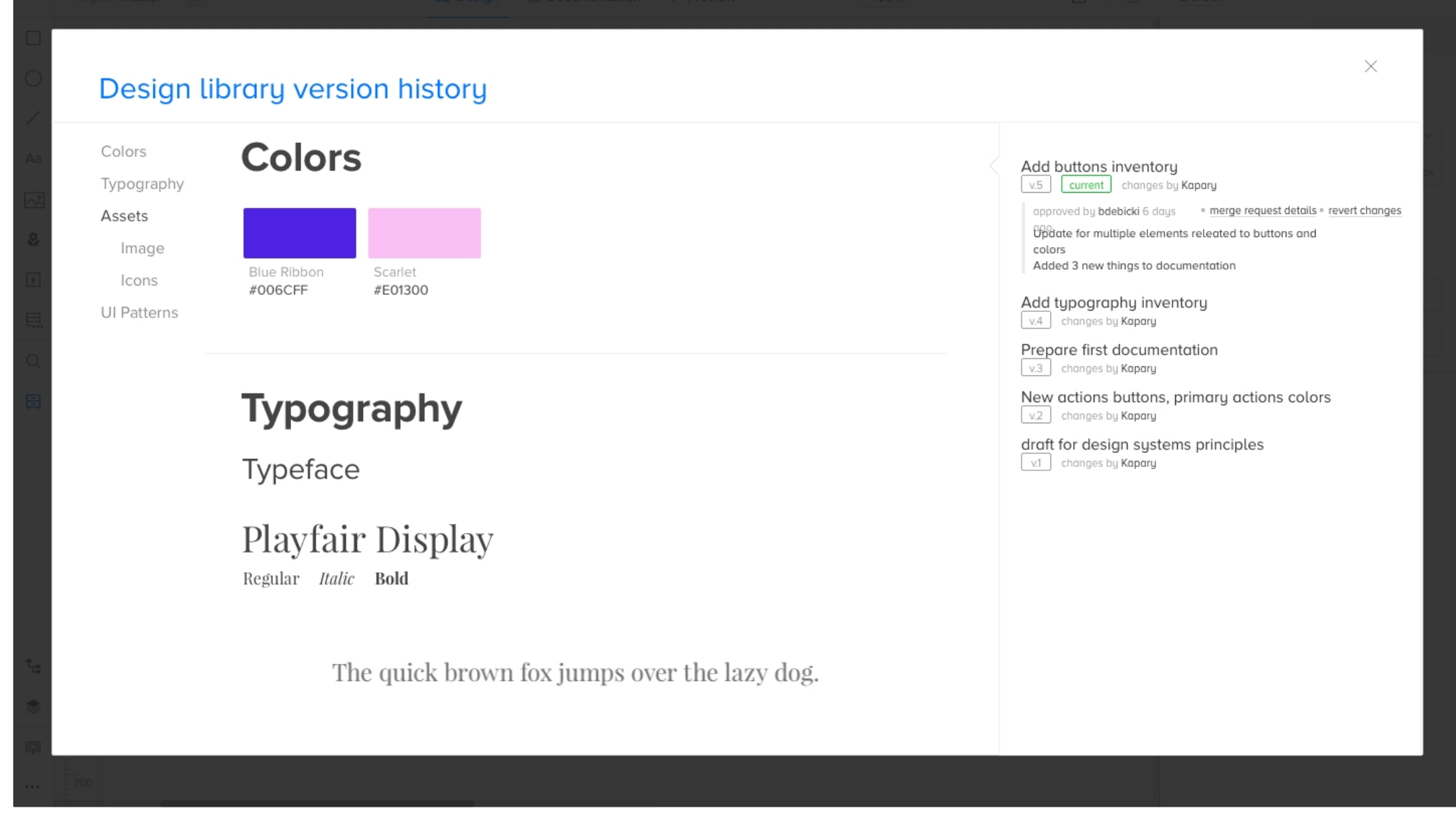
The Flow



Design System Version Control



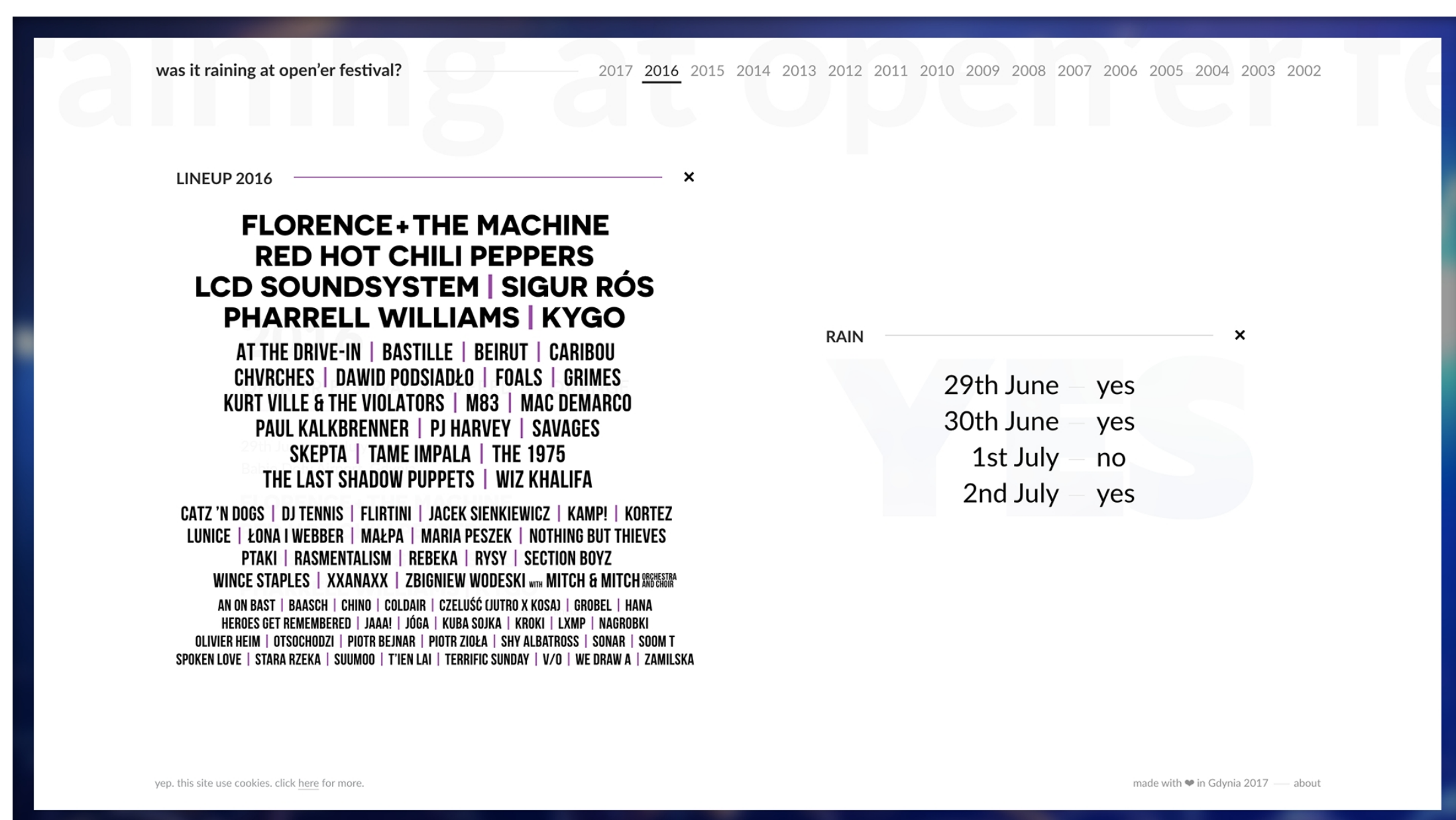
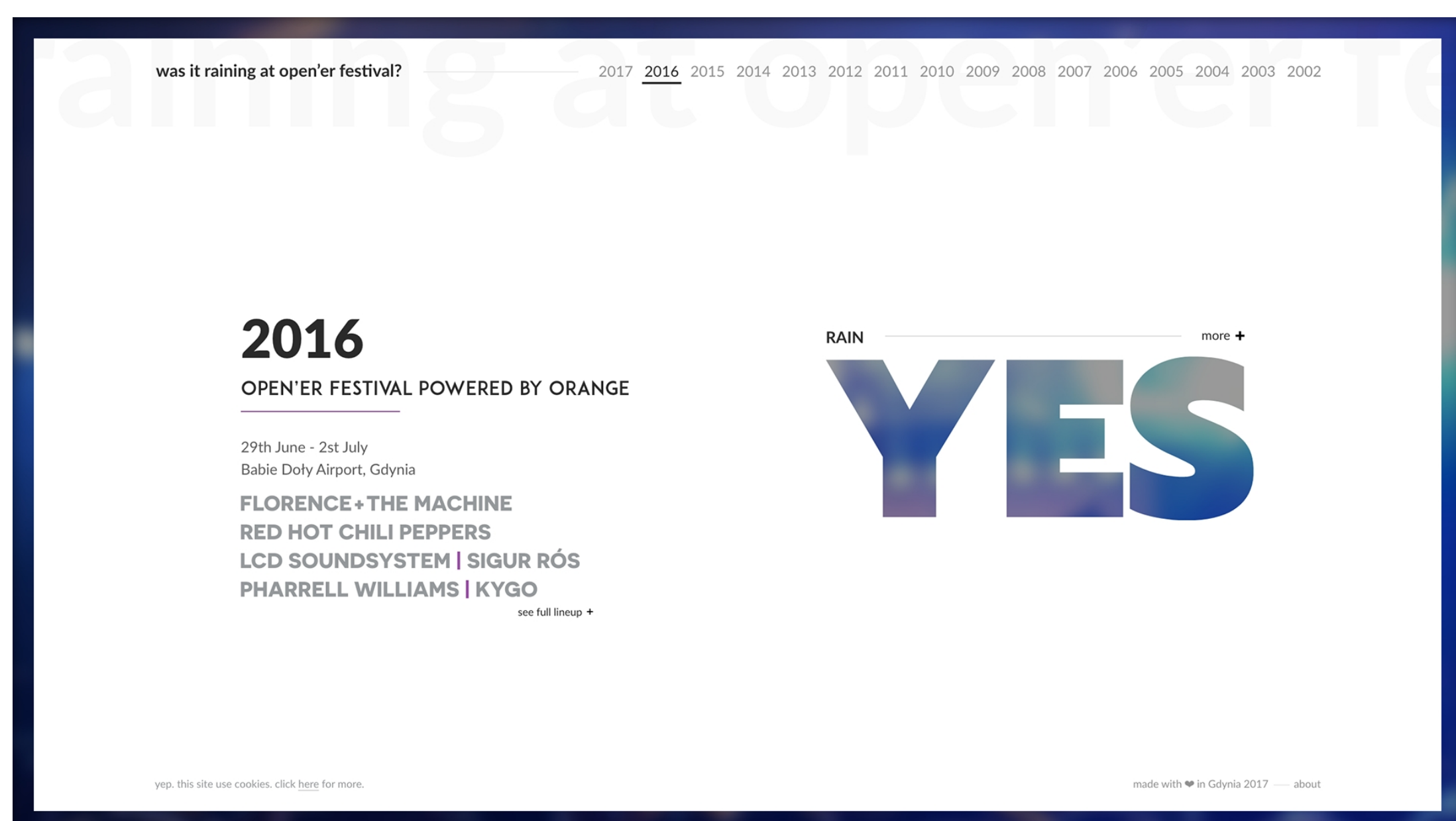
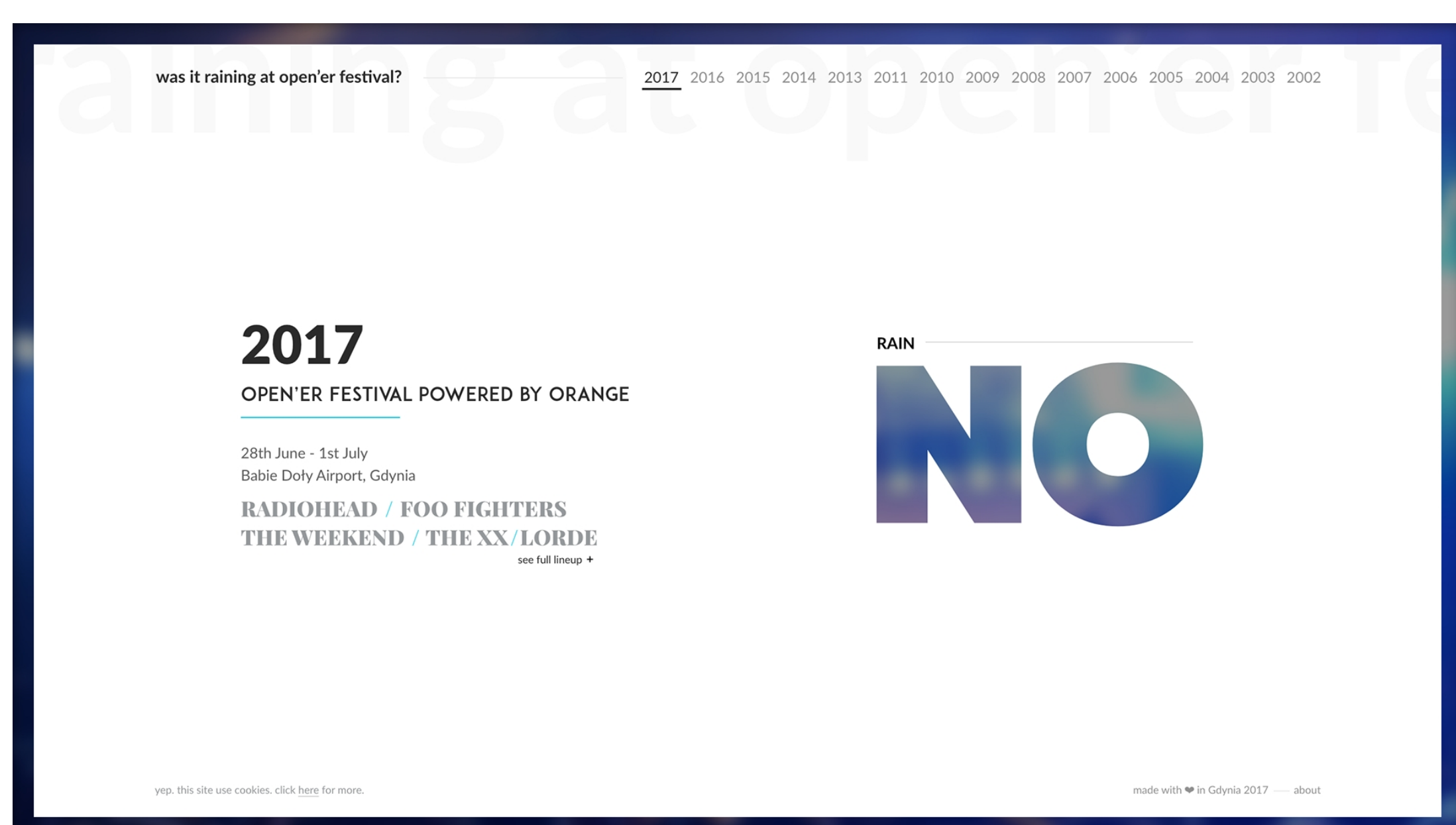
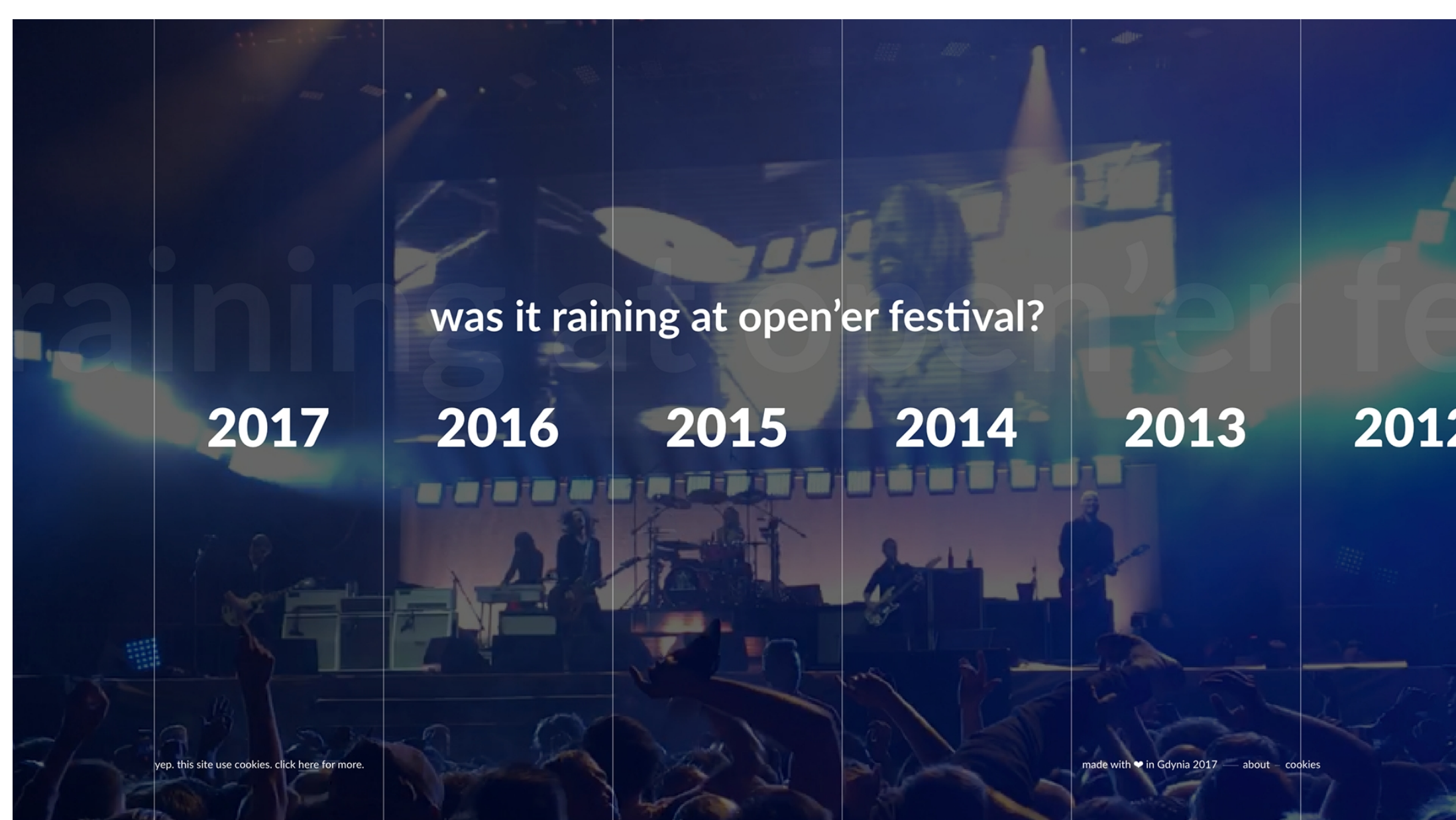
Library Version Control



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



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

Simplify editor

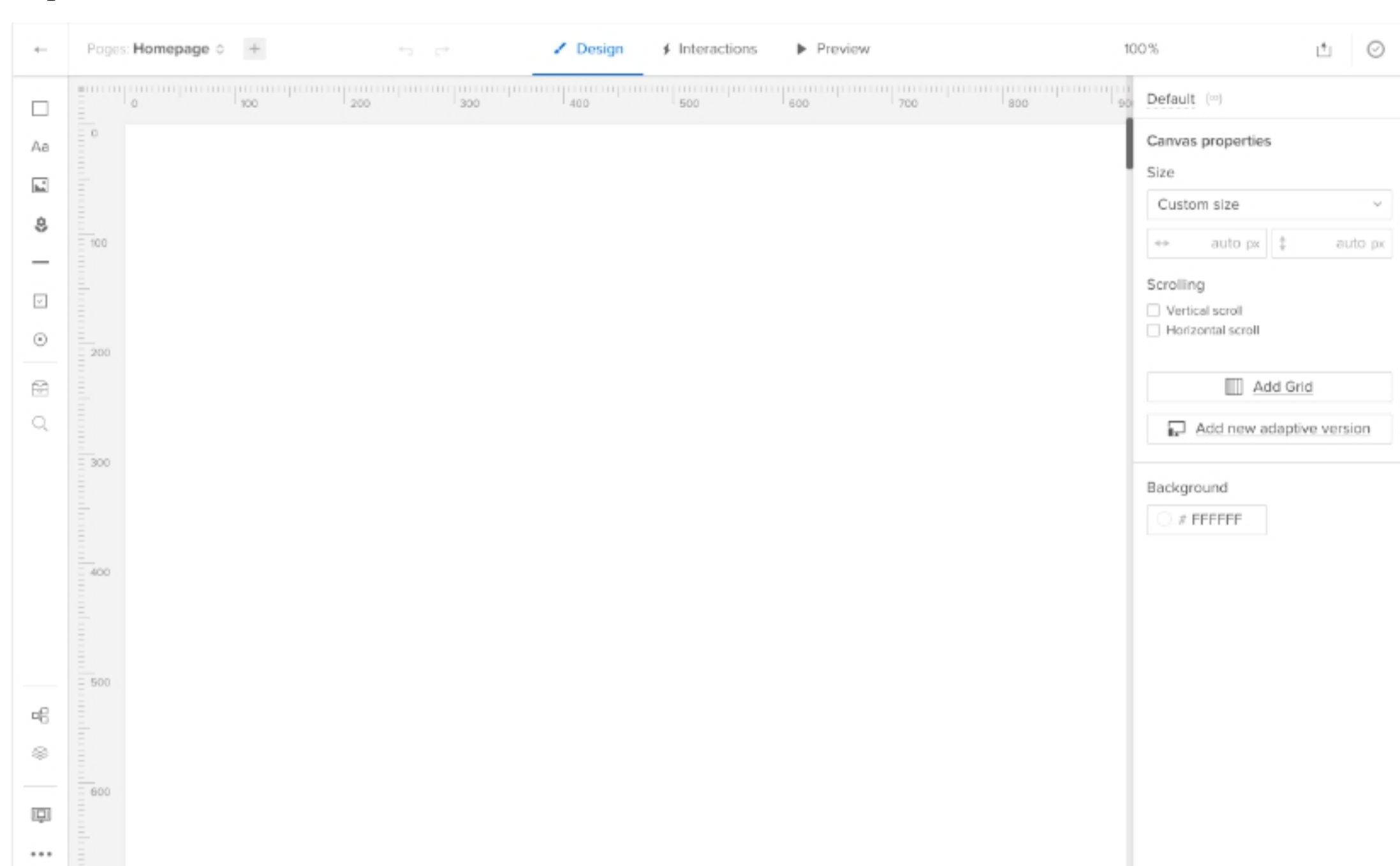
In 2017 in UXPin we did a huge usability test with our users about the condition of our application. Based on the research we decided to prepare a package of improvements in working with elements on the canvas and small redesign of UI.

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).

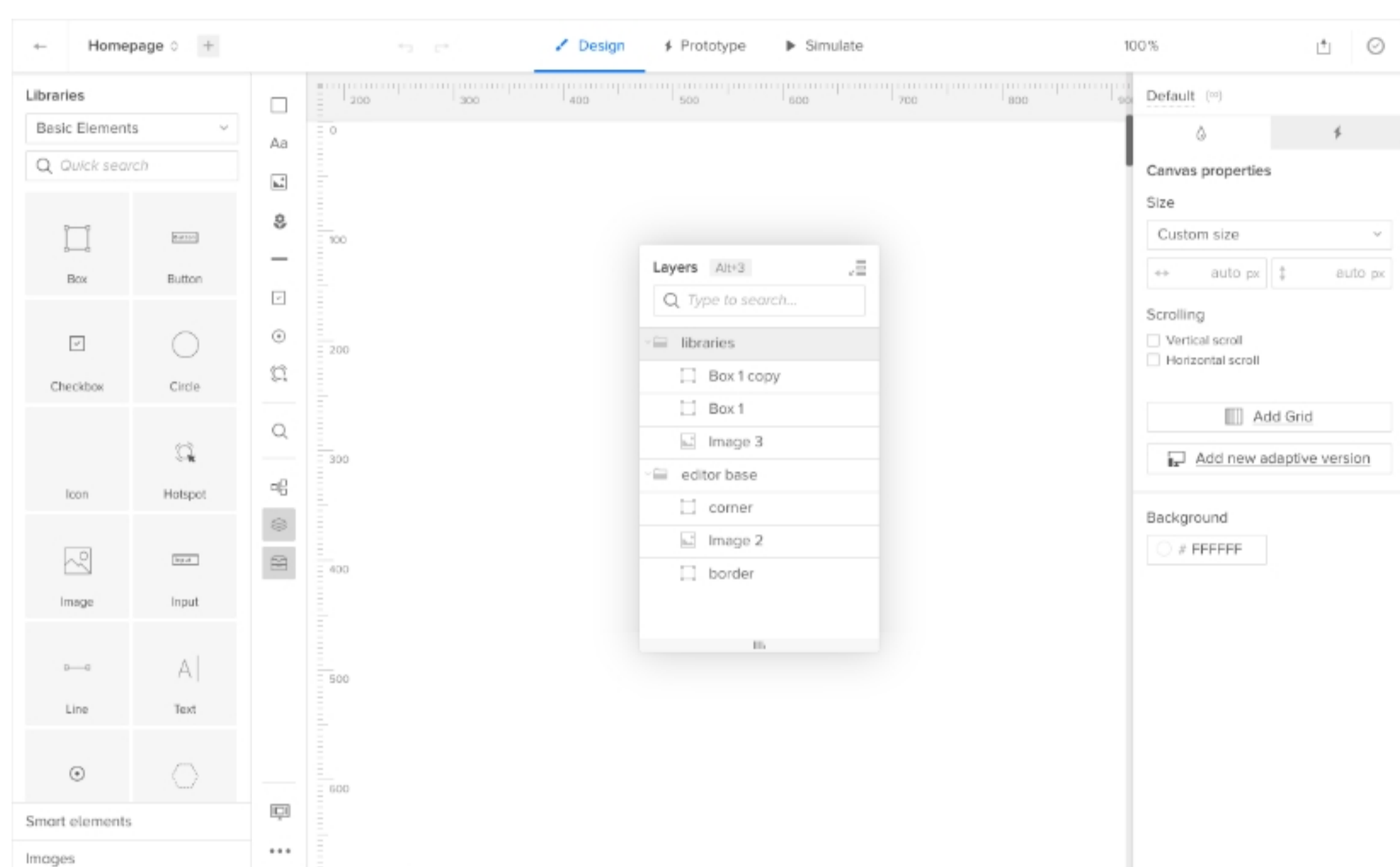
2017 | UXPin

role :product lead / ux design

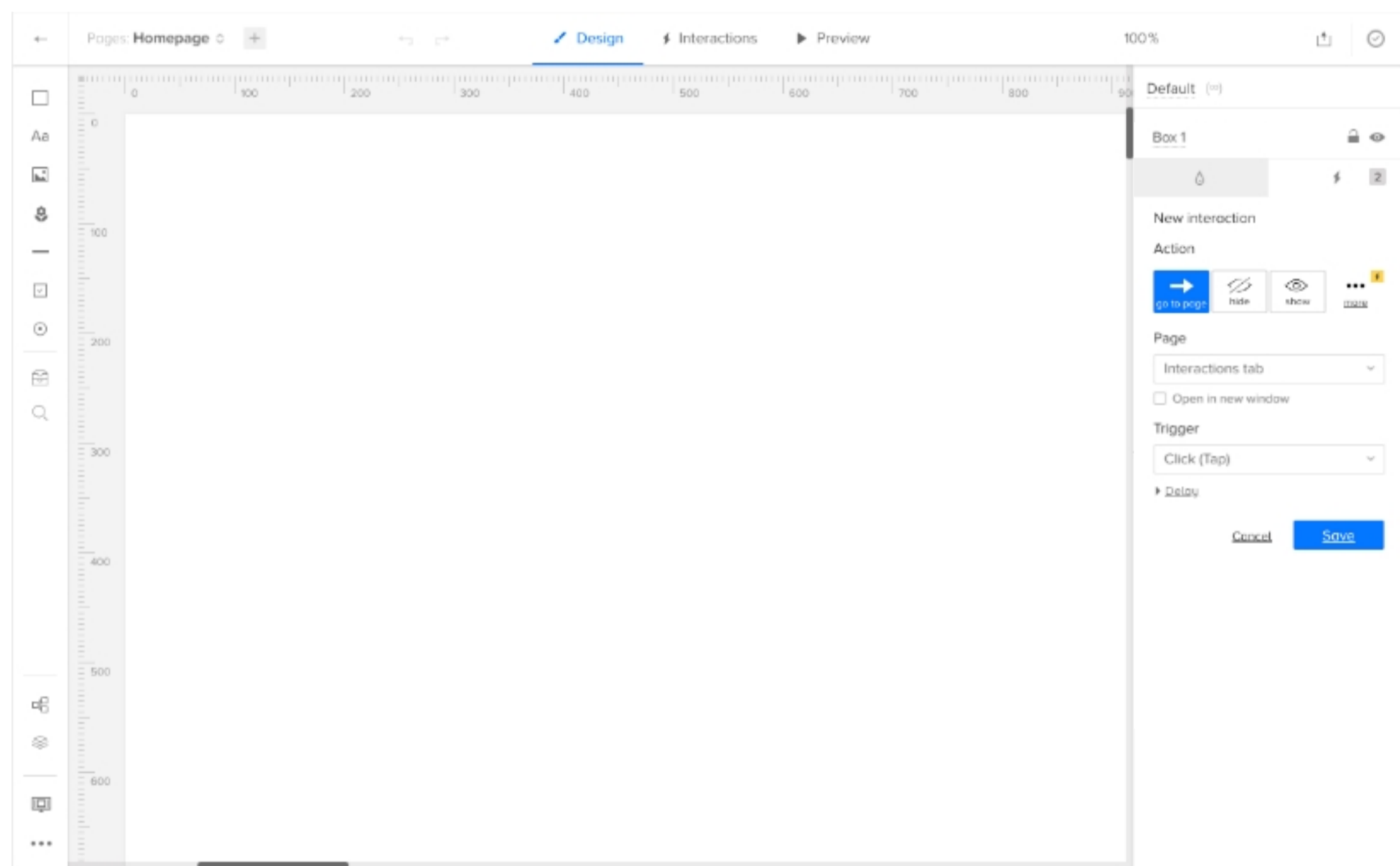
Top bar and toolbar



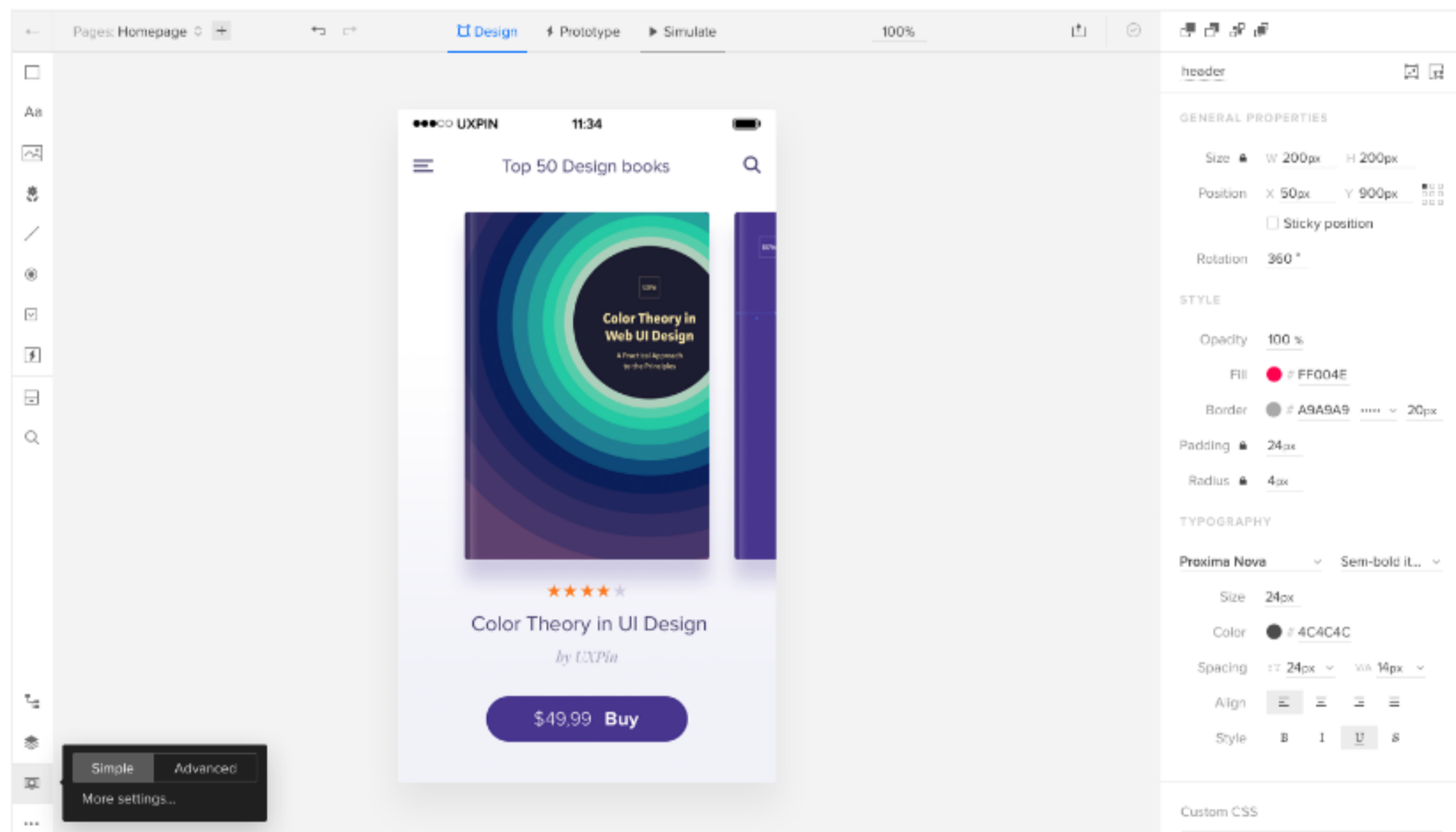
Left panels and floating layers



Interactions panel



Final redesign effect



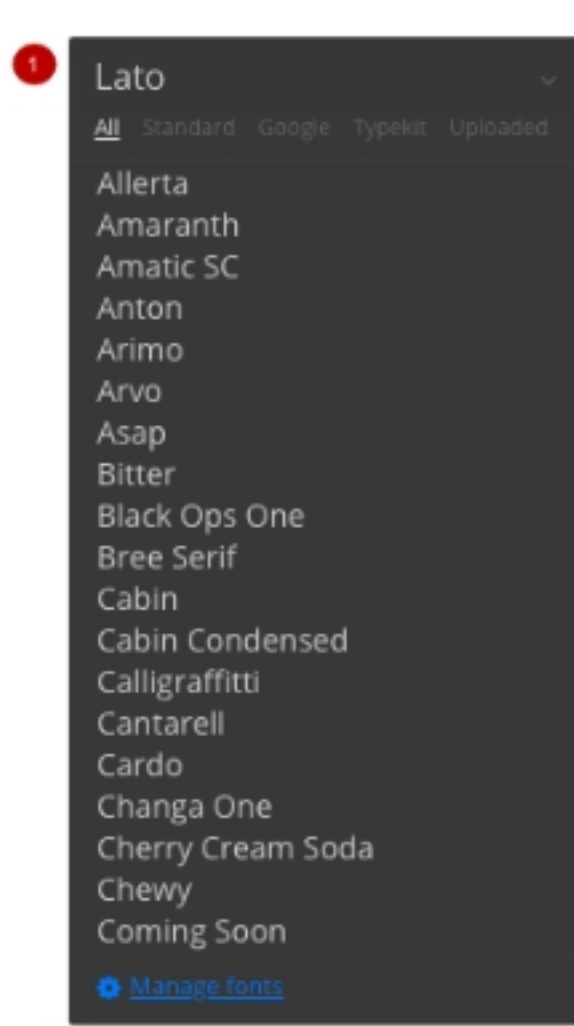
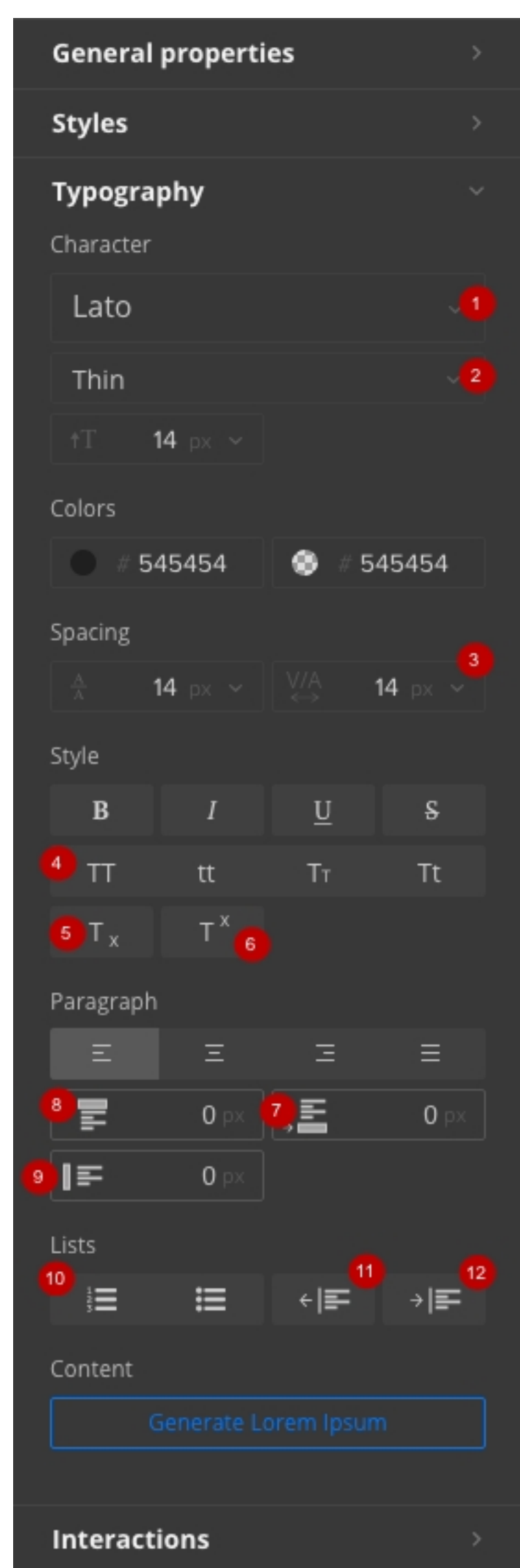
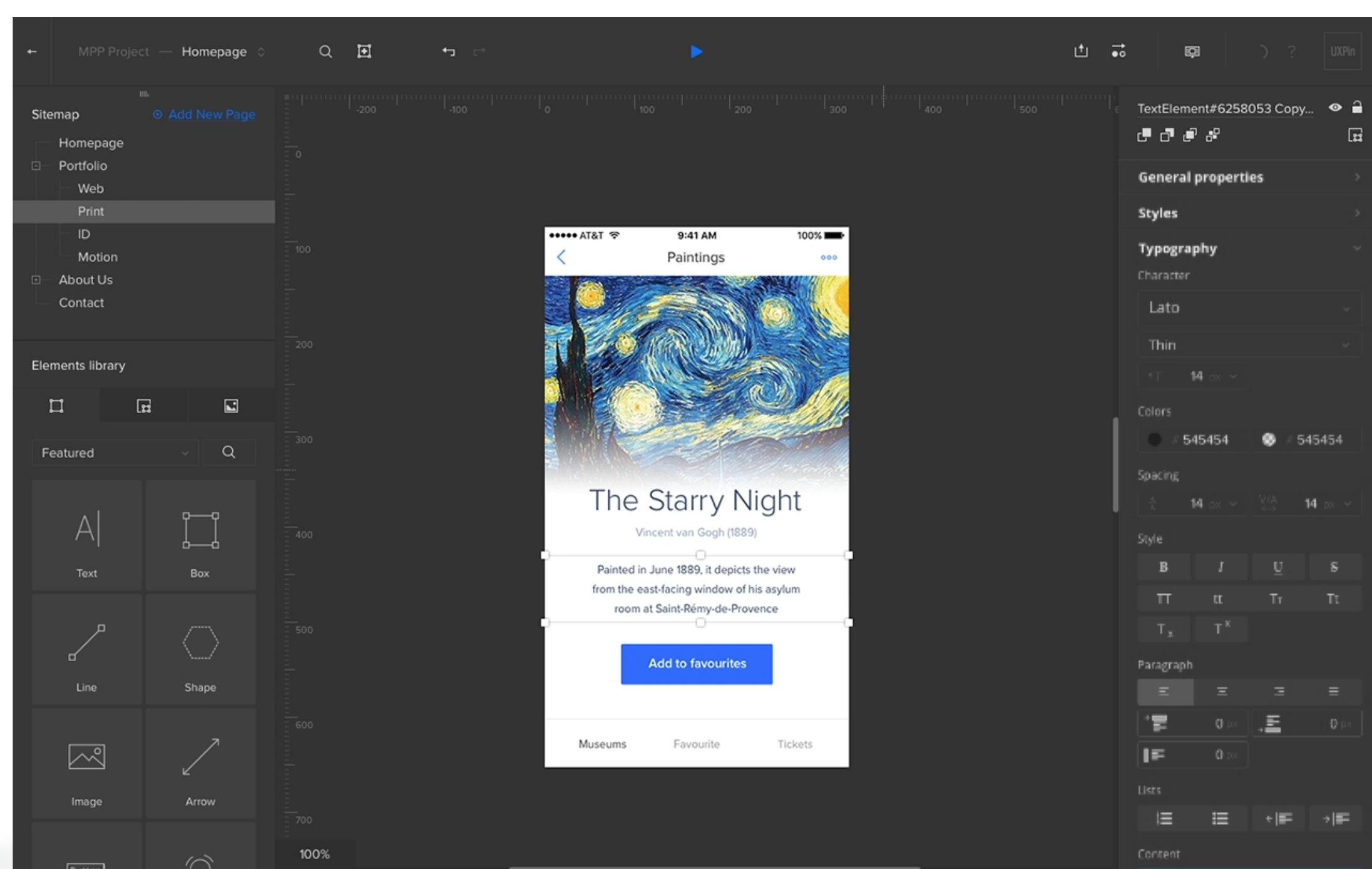
Visual design made by Sebastian Witman

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

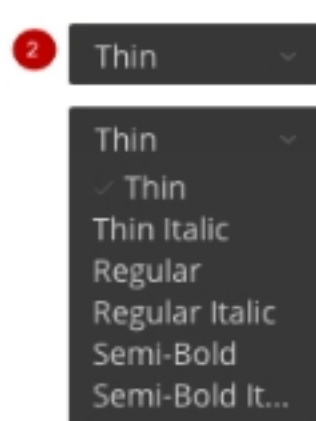


Fonts list; list of all fonts selected by user in fonts manager; list is saved for specific users, every user can have selected different set of fonts;

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

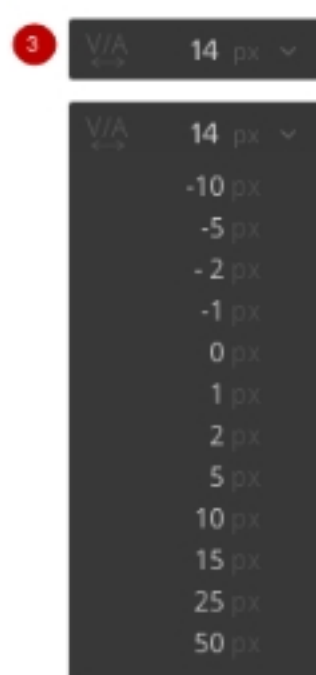
Selected category in list should be saved

By click in input with name of font you can type font name and search it on list.
(ex. <http://recordit.co/Zs1ULnHKA>)



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

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



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

0 is default
0 = auto



Text transformations; change case sensitive; none or only one option can be selected; based on css property *text-transform*

TT - uppercase (*text-transform: uppercase;*)
tt - lowercase (*text-transform: lowercase;*)
Tt - small-caps (*font-variant: small-caps;*)
Tt - capitalize (*text-transform: capitalize;*)



Subscript; create subscript from selected word; we should use CKEditor function for this



Superscript; create superscript from selected word; we should use CKEditor function for this



Bottom paragraph margin; value should work for lists and paragraphs; value should be saved for future paragraphs (same as color, of font size)



Top paragraph margin; value should work for lists and paragraphs; value should be saved for future paragraphs (same as color, of font size)



Left paragraph margin; value should work for lists and paragraphs; value should be saved for future paragraphs (same as color, of font size)



Lists; change paragraph to list (*ordered* or *unordered*); none or only one option can be selected; we should use CKEditor list plugin

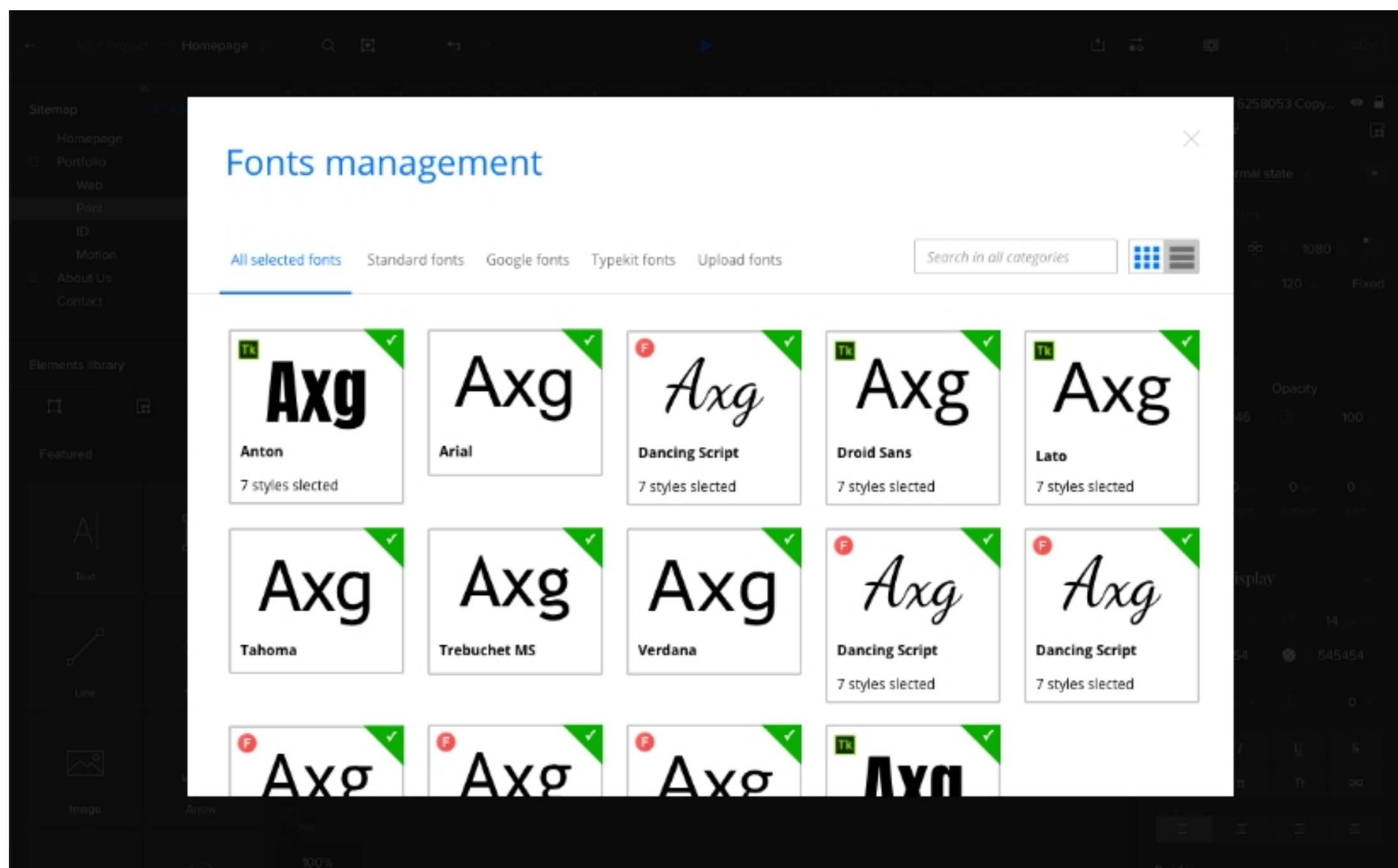


Remove indent; remove indent from group and remove nesting groups; we should use CKEditor feature



Add indent; add indent and nesting list in list; the size of indent is set in left paragraph margin; we should use CKEditor feature

Fonts manager



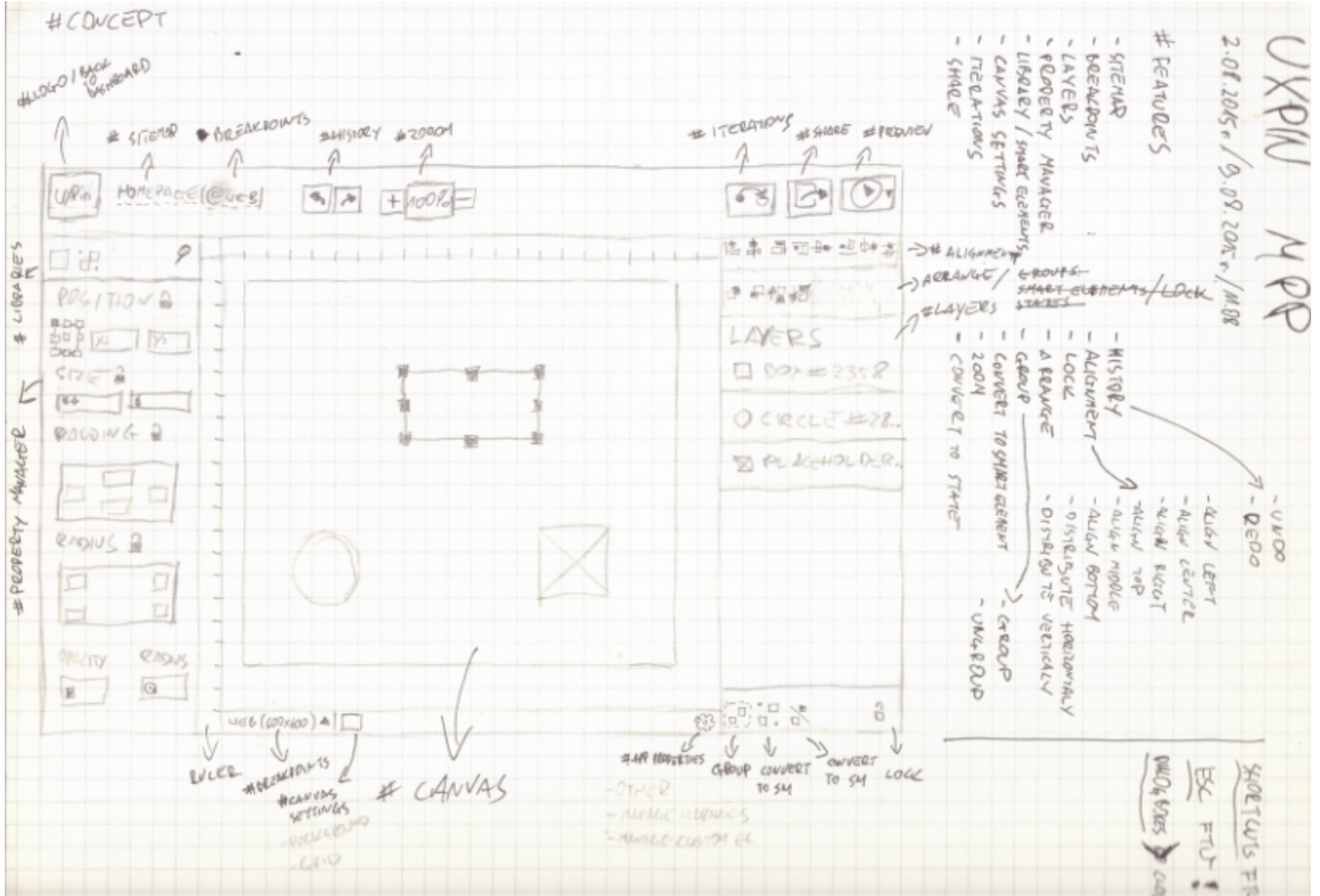
UXPin 3.0

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.

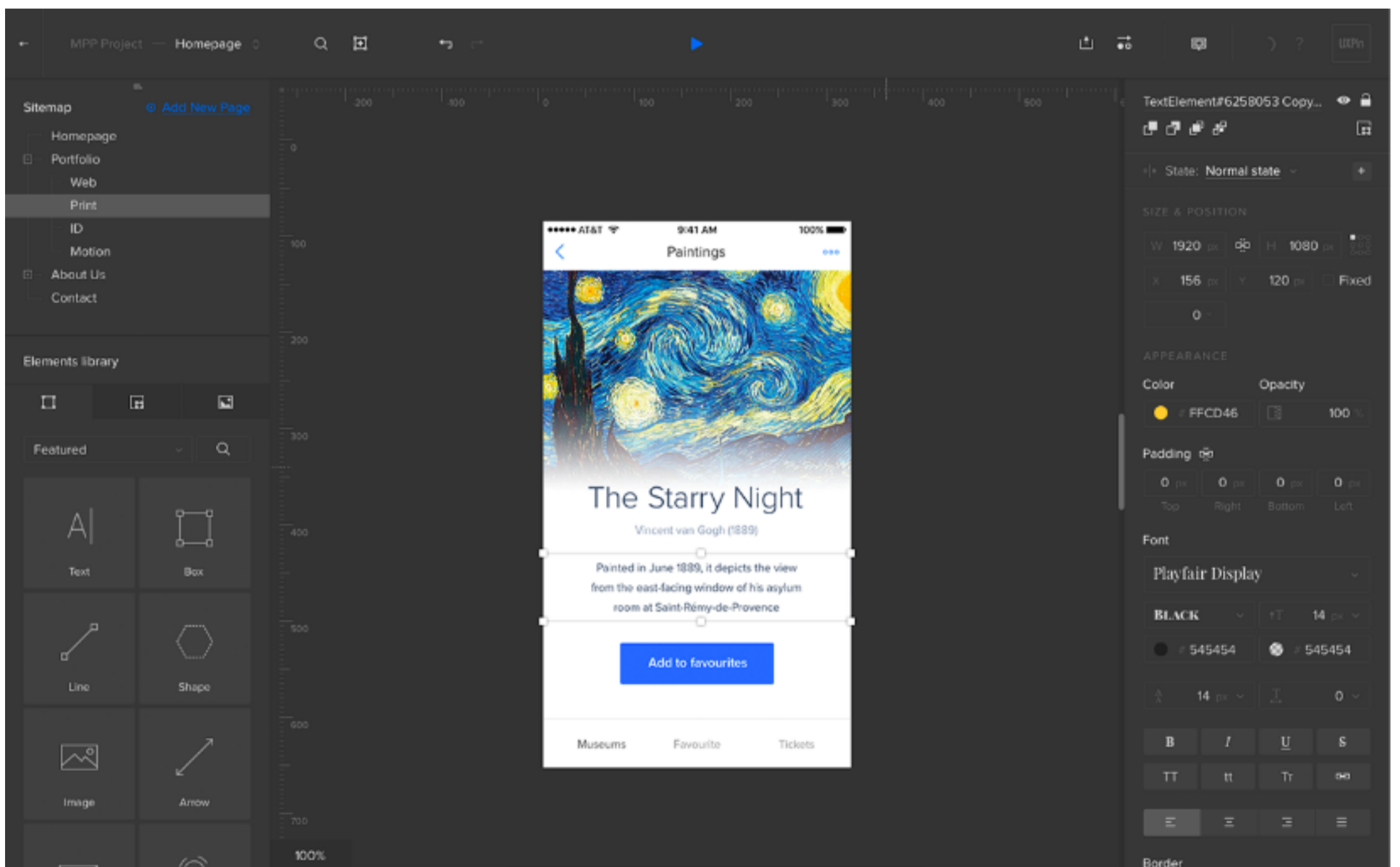
2015 | UXPin

role :product lead / ux design + code

Te one of conners for the MPP Editor.



Final design for UXPin 3.0



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