

Selected projects 2015-2023

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)

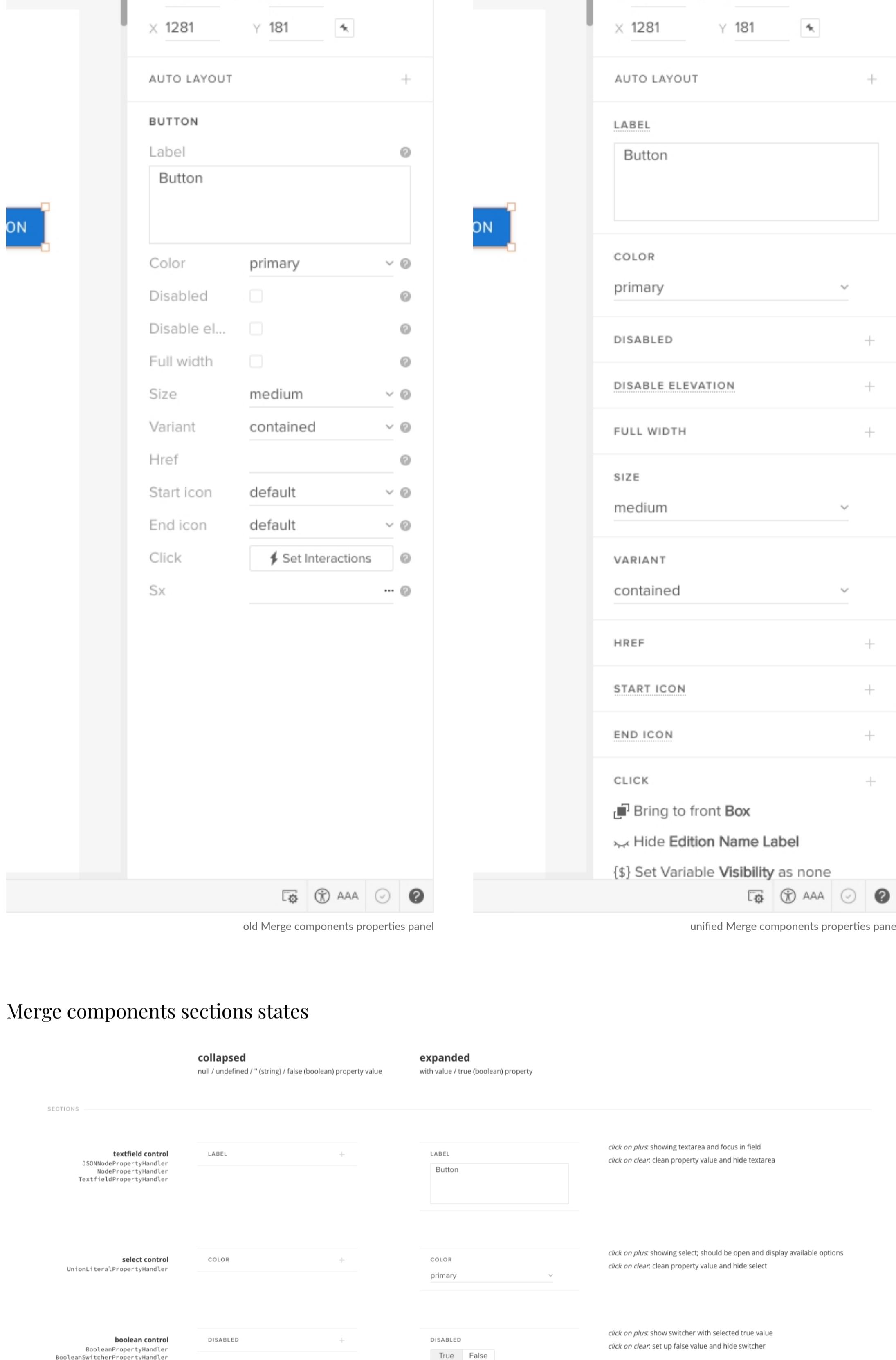
Merge properties panel unification

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

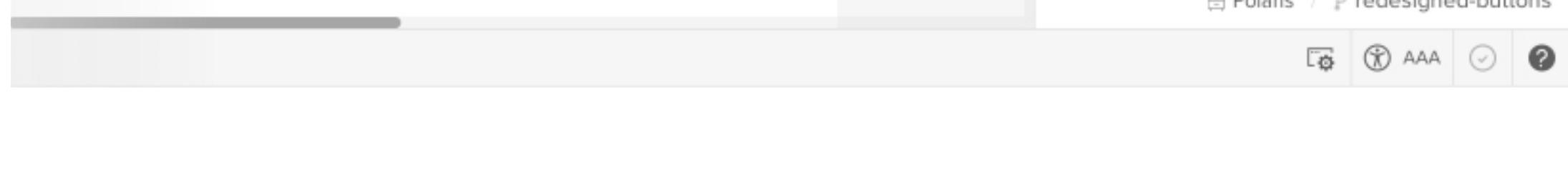


Merge components sections states

This diagram illustrates the various states and controls for different Merge component types:

- collapsed**: null / undefined / "" (string) / false (boolean) property value
- expanded**: with value / true (boolean) property
- SECTIONS**:
 - textfield control**: JSONNodePropertyHandler, NodePropertyHandler, TextFieldPropertyHandler
 - LABEL: Button
 - select control**: UnionLiteralPropertyHandler, BooleanPropertyHandler, BooleanSwitcherPropertyHandler
 - COLOR: primary
 - boolean control**: BooleanPropertyHandler, BooleanSwitcherPropertyHandler, NumberPropertyHandler (with InputMinMax)
 - DISABLED: True / False
 - string control**: JSONStringPropertyHandler, StringPropertyHandler, NumberPropertyHandler
 - HREF: href
 - function control**: FunctionPropertyHandler
 - CLICK: Bring to front Box, Hide Edition Name Label, Set Variable Visibility as none
 - code control**: CodePropertyHandler
 - SX: sx
 - color control**: ColorPropertyHandler
 - COLOR: #FFC7C7
- PROPERTY HAS DESCRIPTION**:
 - if property has description:
 - property heading has dotted underline
 - cursor when hovering over a heading is question mark (help)
 - description is displayed in a Tooltip component; default position of the tooltip should be top
- DEFAULT VALUE HAS BEEN OVERWRITTEN**:
 - default is empty
 - default has value: ERROR MESSAGE

Information about component's library in properties panel



UXPin Design System

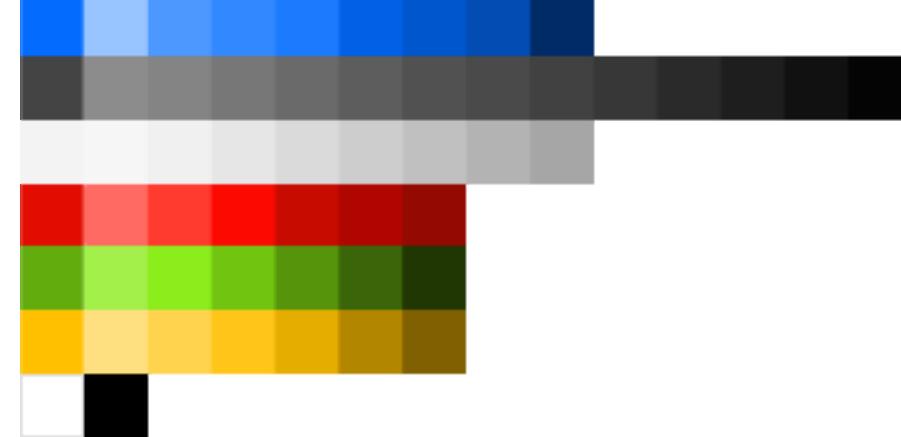
Basing on the actual design of the product I have prepared a design system. The goal was to clean and organize and simplify colors, typography, iconography, and interface elements. During this work, I was in close contact with the design and development team.

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

color palette

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



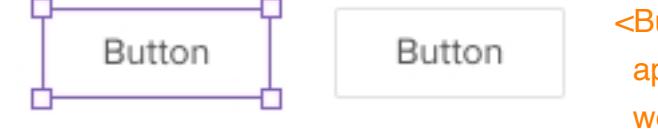
components

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



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



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

Merge Library Manager

Merge Library manager is an evolution of the Merge integration process. The best description for the feature is that Merge Library Manager is the User Interface layer for Merge integration. Because classic integration is based on a GIT repository, it forced users to be code-know. New integration process based on the NPM package. This approach doesn't force the user to have access to the git repository and does not force him to prepare any configuration from the code side. This approach opens Merge integration to non-coding users.

I am responsible for the design and front-end development.

2021 - 2022 | UXPin
role :code + design

merge library manager for nom and git integrated libraries

The screenshot shows the Merge Library Manager interface for editing a 'Button' component. On the left is a sidebar with categories like General, Actions, Navigation, Layout, and Card. The 'Actions' section has 'Button' selected. The main area shows a preview of a blue 'SIGN UP' button and a properties panel. The properties panel includes fields for 'Color' (Color), 'Property name' (Color), 'Display name' (Color), 'Description' (empty), 'Property type' (one of), 'Property control' (select), 'Options' (default, inherit, primary, secondary), 'Default value' (primary), 'Children' (disabled), 'Disabled' (disabled focus ripple, disabled ripple), and 'Full width' (disabled). Buttons for 'Discard changes' and 'Save changes' are at the top right.

merge tab in dashboard

The screenshot shows the Merge dashboard with the 'Merge' tab selected. On the left is a sidebar with categories like Projects, Design Systems, Merge, and Manage your team. The 'Merge' tab has sections for 'Components from code' (with a 'Create library with components from code' button) and 'Components from Storybook' (with a 'Create library with components from Storybook' button). A note at the bottom says 'To learn more about revolutionary Merge technology, visit our Docs.'

The screenshot shows the Merge dashboard with the 'Merge' tab selected. On the left is a sidebar with categories like Projects, Design Systems, Merge, and Manage your team. The 'Merge' tab has sections for 'Components from code' (with a 'Create library with components from code' button) and 'Components from Storybook' (with a 'Create library with components from Storybook' button). A note at the bottom says 'To learn more about revolutionary Merge technology, visit our Docs.'

adding new component flow

The screenshot shows the Merge component editor for a 'Button' component. The sidebar on the left lists categories like General, Actions, Navigation, Layout, and Card. The 'Actions' section has 'Button' selected. A central modal window titled 'Add new components' contains a text input for 'List import for components you would like to use in your library' with placeholder code: 'import { Button, Input } from '@your-library';'. It also has a 'select category' dropdown and a 'Cancel' button. Below the modal is a properties panel for 'Size' (11.5) and 'Variant' (latest). A 'Great work!' message at the bottom indicates 3 new components were added.

page 3 – Bartosz Dębicki Selected Projects 2015-2021

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 cede to based on Git branches and tags.

2021 | UXPin
role :code + design

seeing default version of the library for all projects

The screenshot shows the UXPin interface. On the left, the 'Design' tab is selected, displaying a library catalog with categories like General, Actions, Navigation, Layout, and Tabs. A tooltip on the 'Tabs' category indicates it uses the 'rc-1.2' tag. In the center, a modal window titled 'Edit library' is open, showing fields for 'Library name' (set to 'Material UI') and 'Library authorization token'. It also displays the 'Default library version in new projects' dropdown set to 'Latest' and a 'Tag' dropdown set to 'rc-1.1'. A note below explains managing library versions. On the right, a sidebar shows 'Who can access' (Me) and 'Who can edit' (Me), both with 'rc-1.1' selected. Below this is a 'Tags' section with 'dev-1.1' through 'dev-1.4' and a 'BRANCHES' section with 'new-button', 'upgrade-react', 'poc-input2', 'release-1.1', and 'release-1.2'. A 'Delete' button is visible in the branches list. At the bottom of the modal, there are 'Cancel' and 'Save' buttons. The background shows the library catalog with various components like Icon, Typography, Button, IconButton, Fab, Link, AppBar, Toolbar, Tab, and Tabs.

setting library version on project level

The screenshot shows the 'Manage Library Versions' dialog. It states the project uses version 'rc-1.1'. There are two tabs: 'TAGS' (selected) and 'BRANCHES'. Under 'TAGS', 'rc-1.1' is checked. Under 'BRANCHES', 'new-button', 'enums', and 'upgrade-react' are listed. A note at the bottom says: 'Heads up: If you select a 'Branch', your prototype will always update whenever someone changes the 'Branch'. Want to avoid that? Choose 'Tag' instead.' At the bottom are 'Cancel' and 'Save' buttons. The background shows the library catalog with components like Icon, Typography, Button, IconButton, Fab, Link, AppBar, Toolbar, Tab, and Tabs.

version conflicts

The screenshot shows a warning message: 'Some Merge components used in this project are unavailable in the selected library version – rc-1.2.' It lists missing elements: 'Default Button', 'Destructive Button', and 'Input'. It also shows a warning message: '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.' It shows two buttons: 'Default Button' (blue) and 'Destructive Button' (red). At the bottom are 'Cancel' and 'Proceed' buttons. The background shows the library catalog with components like Icon, Typography, Button, IconButton, Fab, Link, AppBar, Toolbar, Tab, and Tabs.

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

The page features a large, bold title "The Github Primer" with "Github" above "Primer". Below the title, it says "design system is called". At the bottom left, there are three sections: "Technology", "Design", and "Documentation & guidelines". Each section has a table with various design system components and their status (checkmark or X). At the bottom right, there's a call to action for UXPin.

Technology

	Code Depth	HTML/CSS
Components	✓	
JS Library/Framework	X	
TypeScript	X	
Web Components	X	
Tests	AVA	
Linter	StyleLint	
CSS	Sass	
CSS In JS	X	
Design Tokens	Scss	
Bundle Manager	Primer Module Build Scriptno	
Distribution	npm	

Design

UI Kit	X
Brand Guidelines	X
Color Naming	natural (e.g. gray-300)
Contrast Analysis	X
Typography	✓ Go to Typography
Icons	Octicons (SVG)
Space / Grid	✓ Go to Space / Grid
Illustration	X
Data Visualization	X
Animation	X
Voice & Tone	X

Documentation & guidelines

Accessibility Guidelines	X
Design Principles	X
Documentation Website	X
Code Documentation	Markdown
Storybook	✓ Go to Storybook

Evangelize Design System with a Free Template!

UXPin: Prototype and Manage Your Design System

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

[Learn more & download](#)

[Start a free trial now!](#)

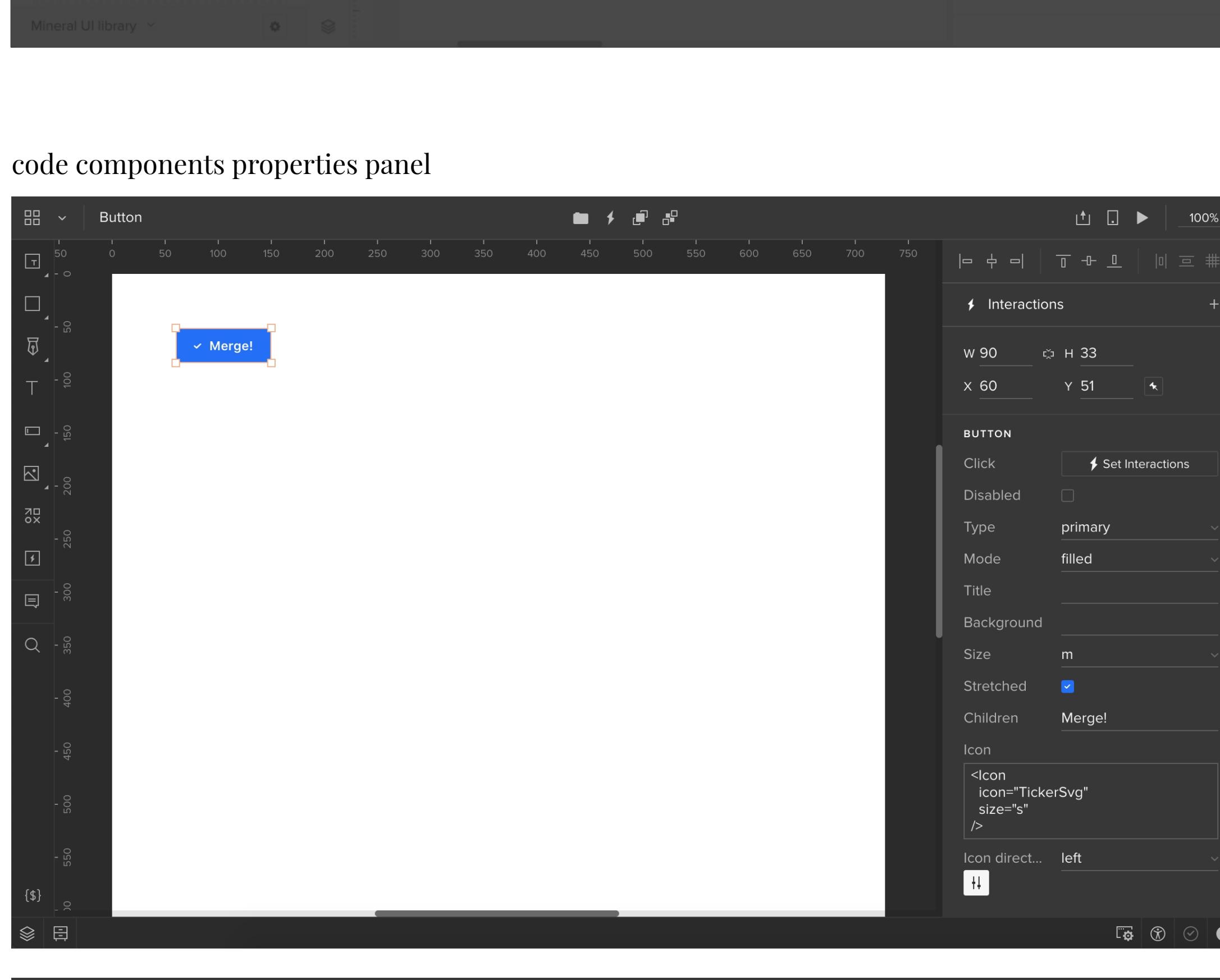
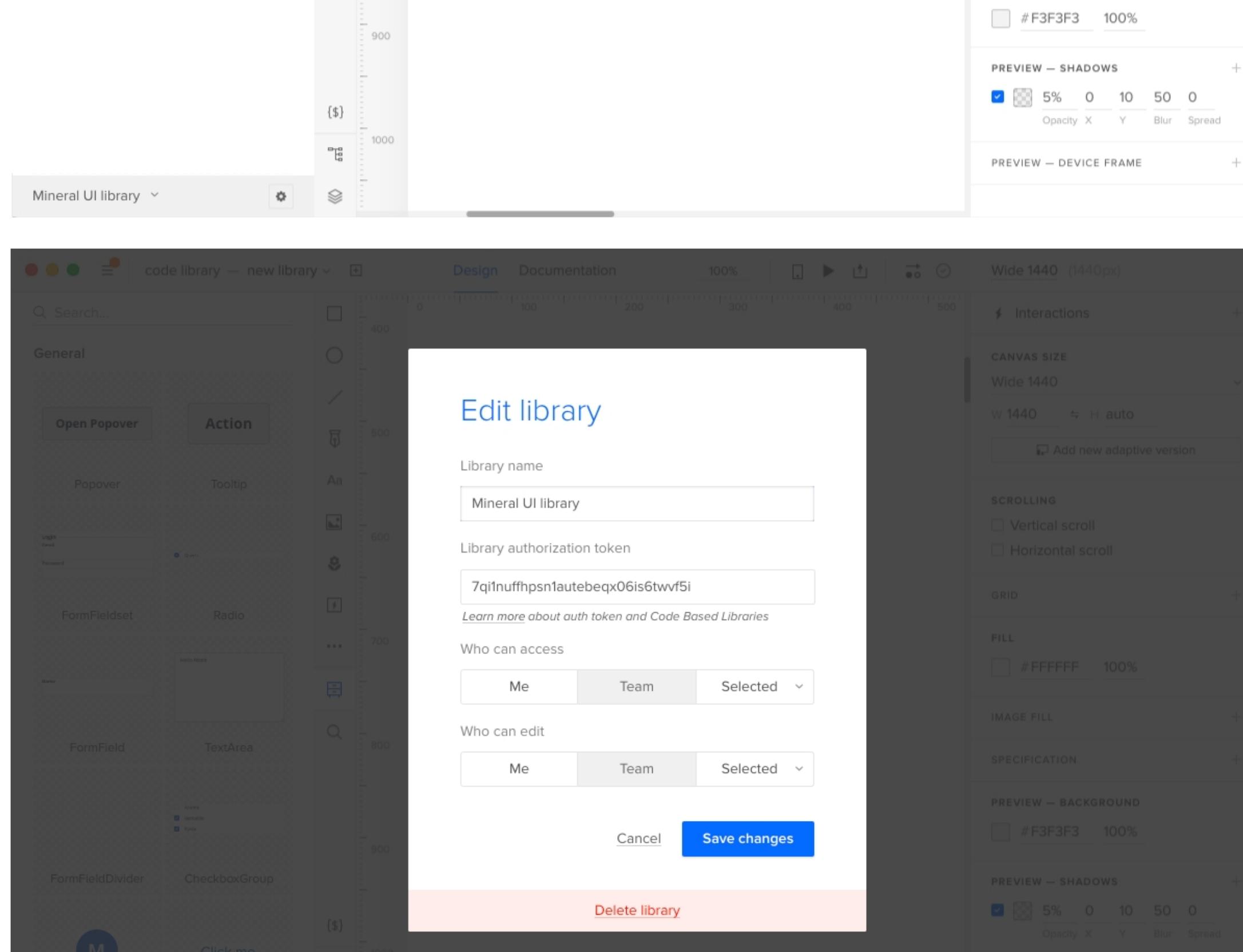
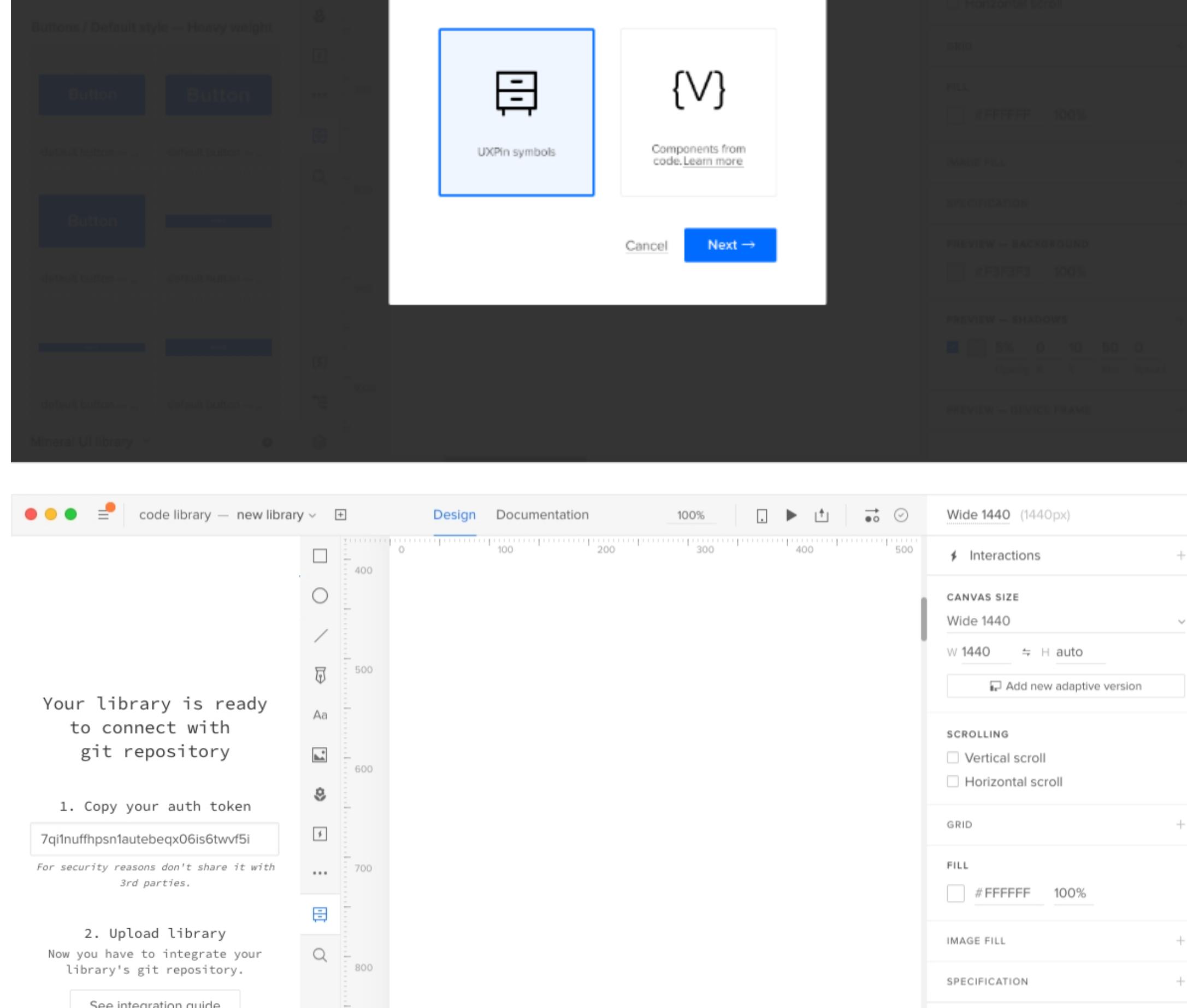
UXPin Merge

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.

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



code components properties panel



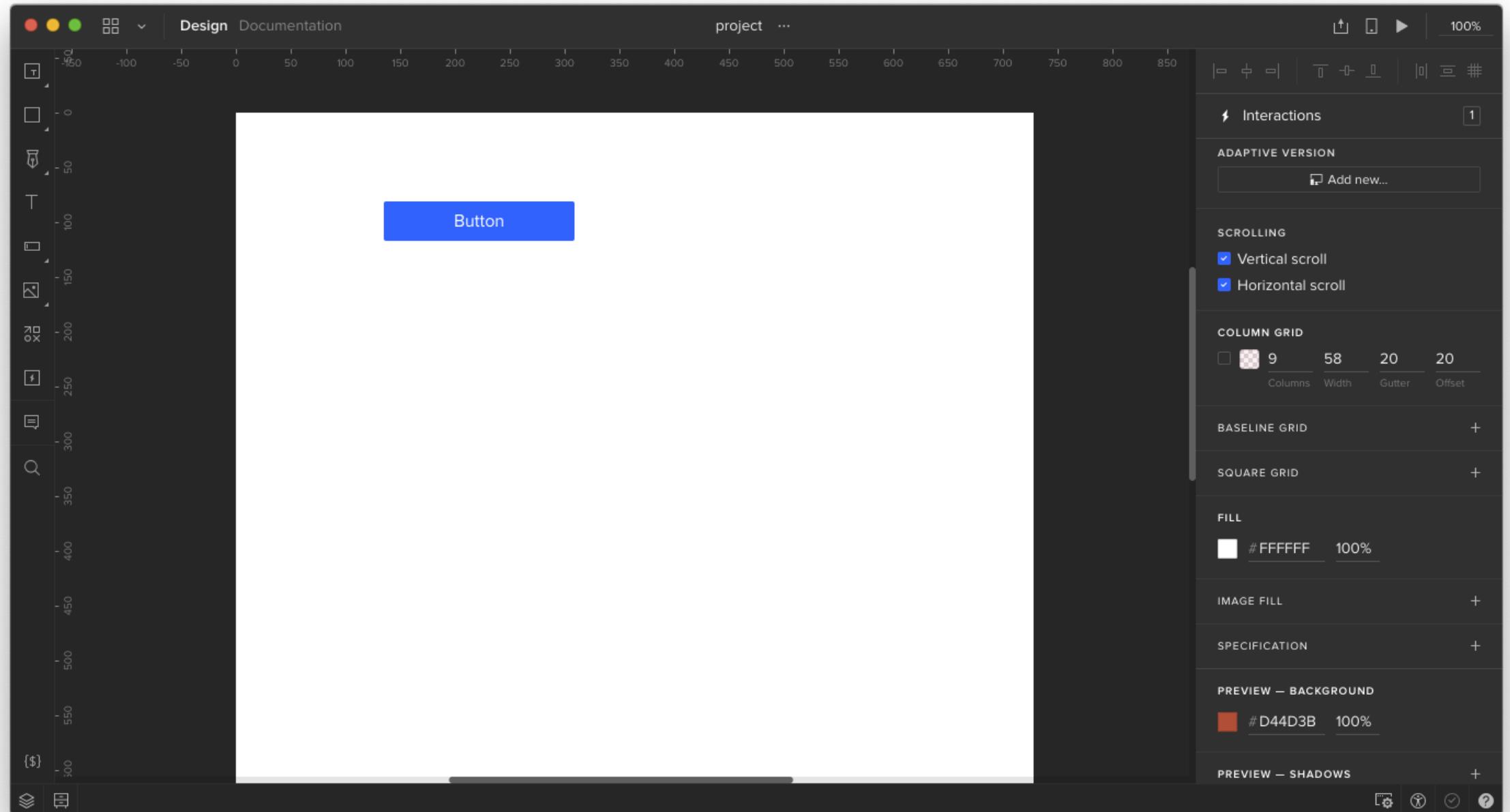
UXPin Desktop App

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



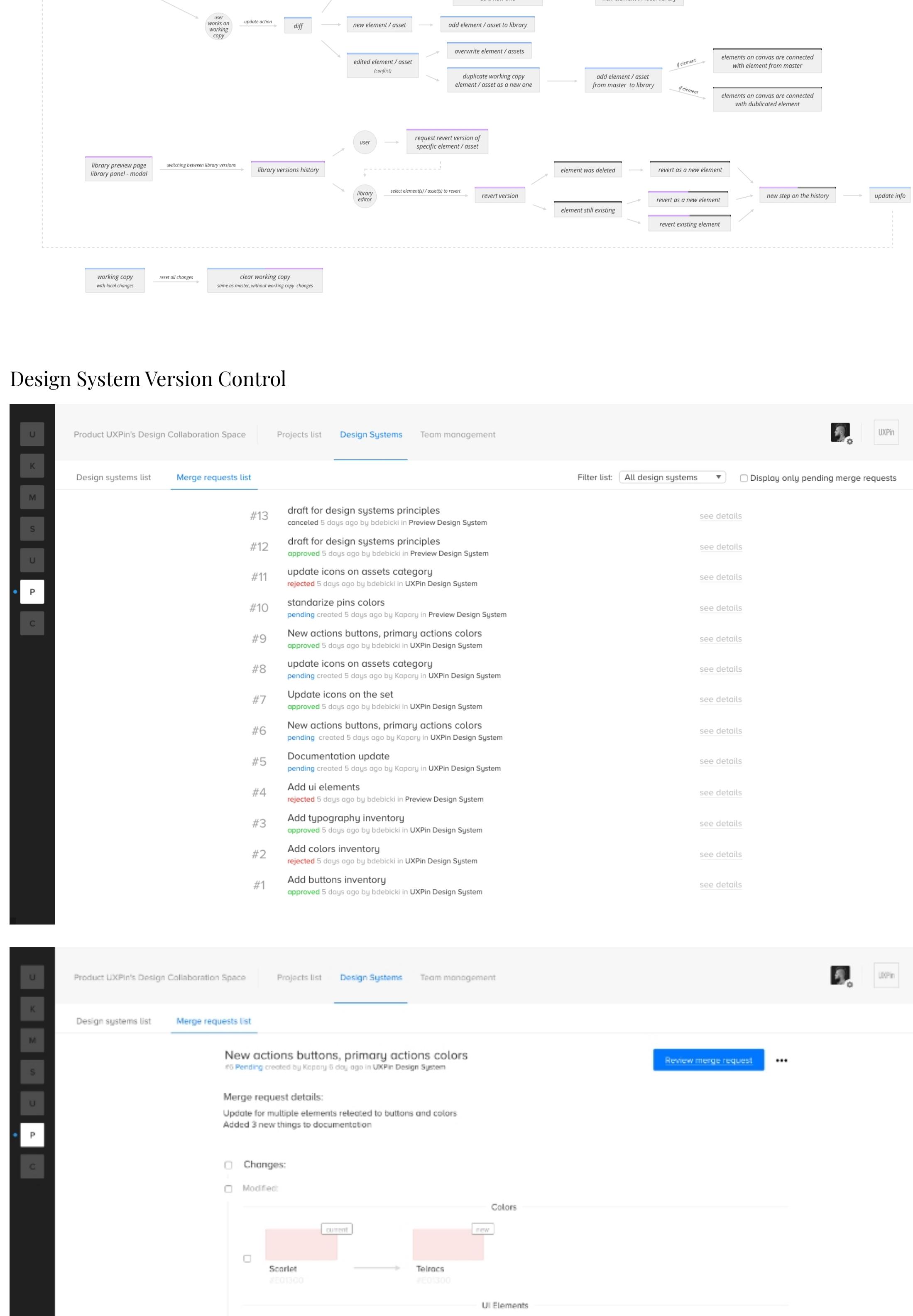
Design System version control

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.

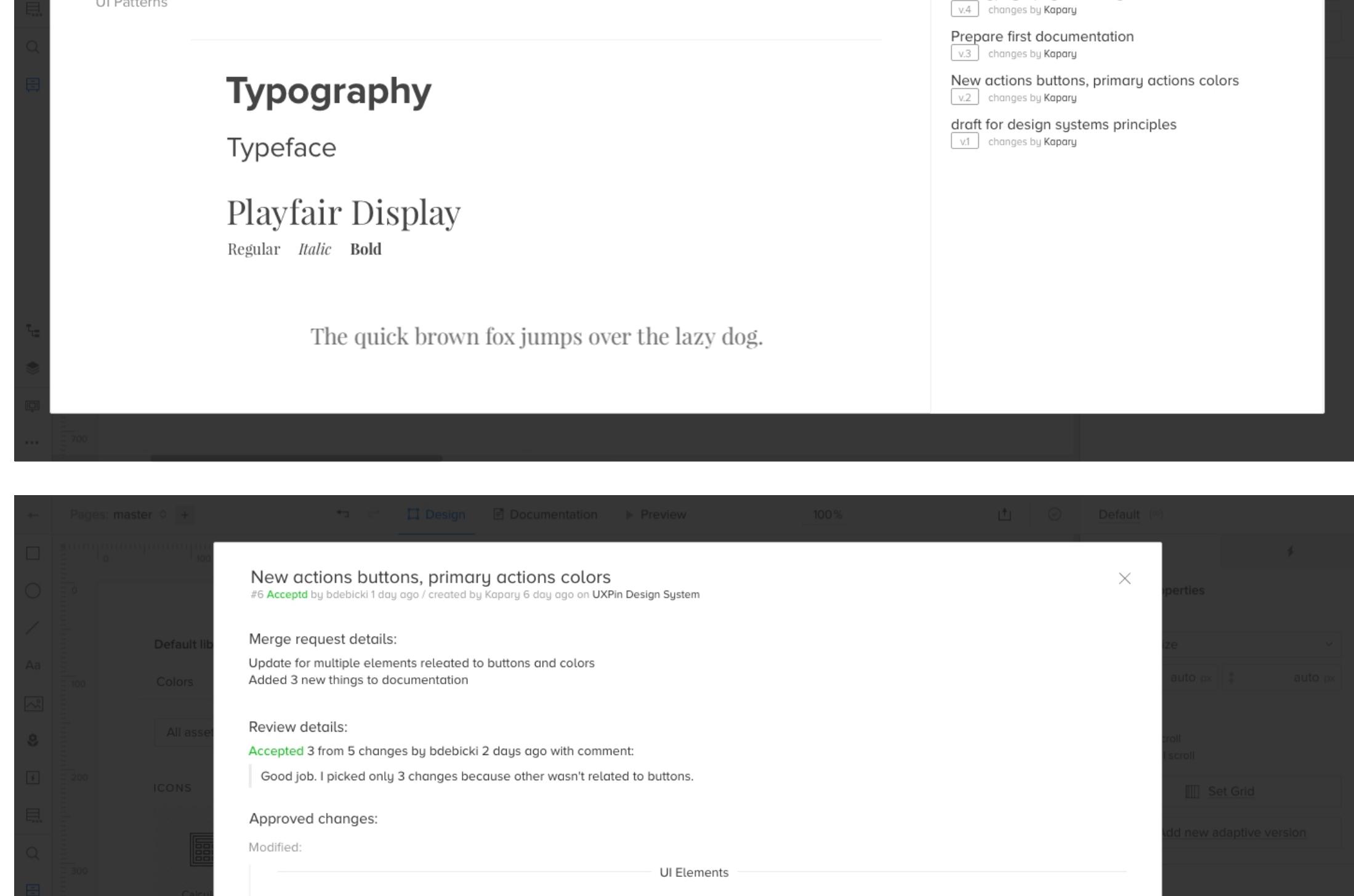
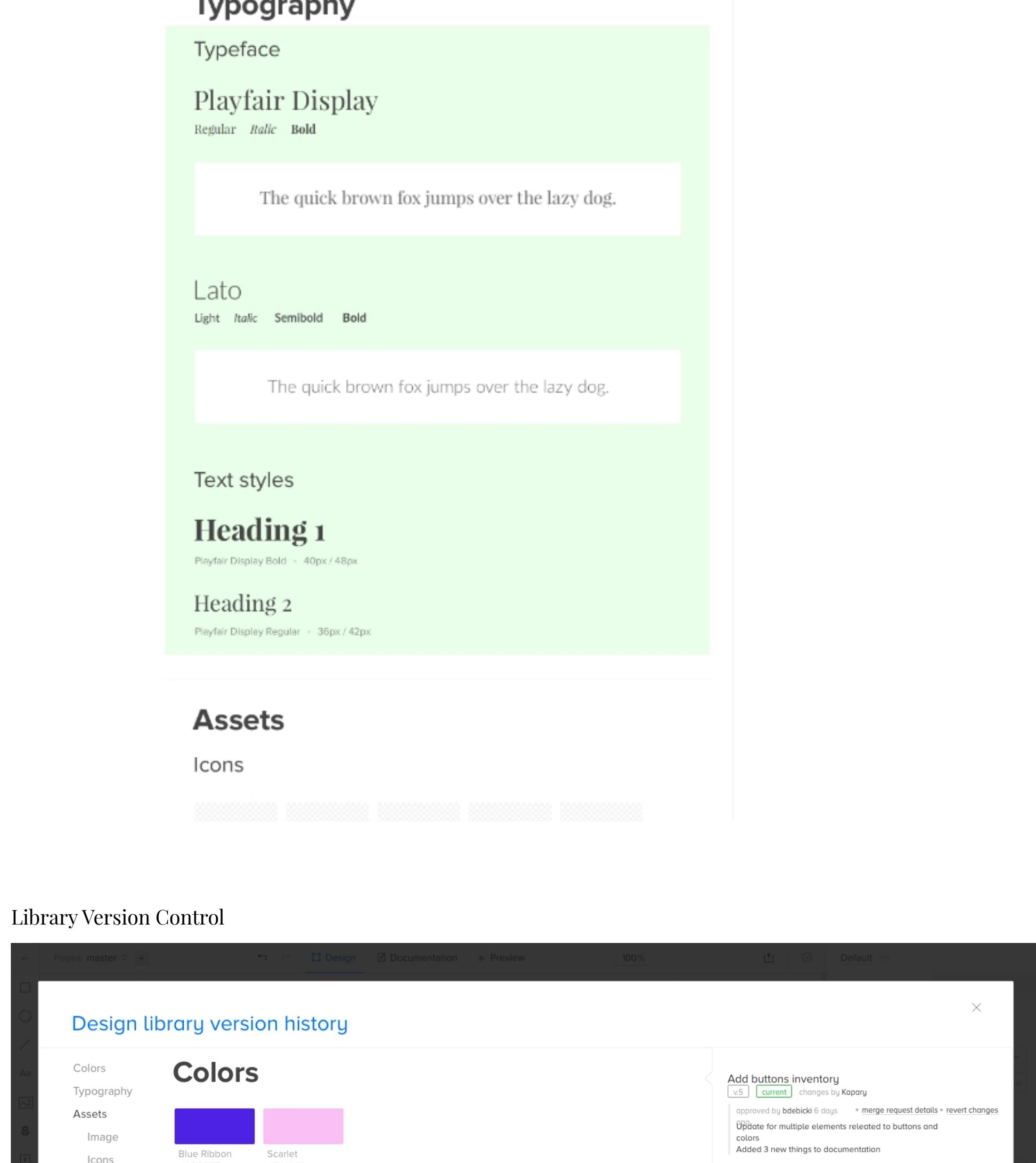
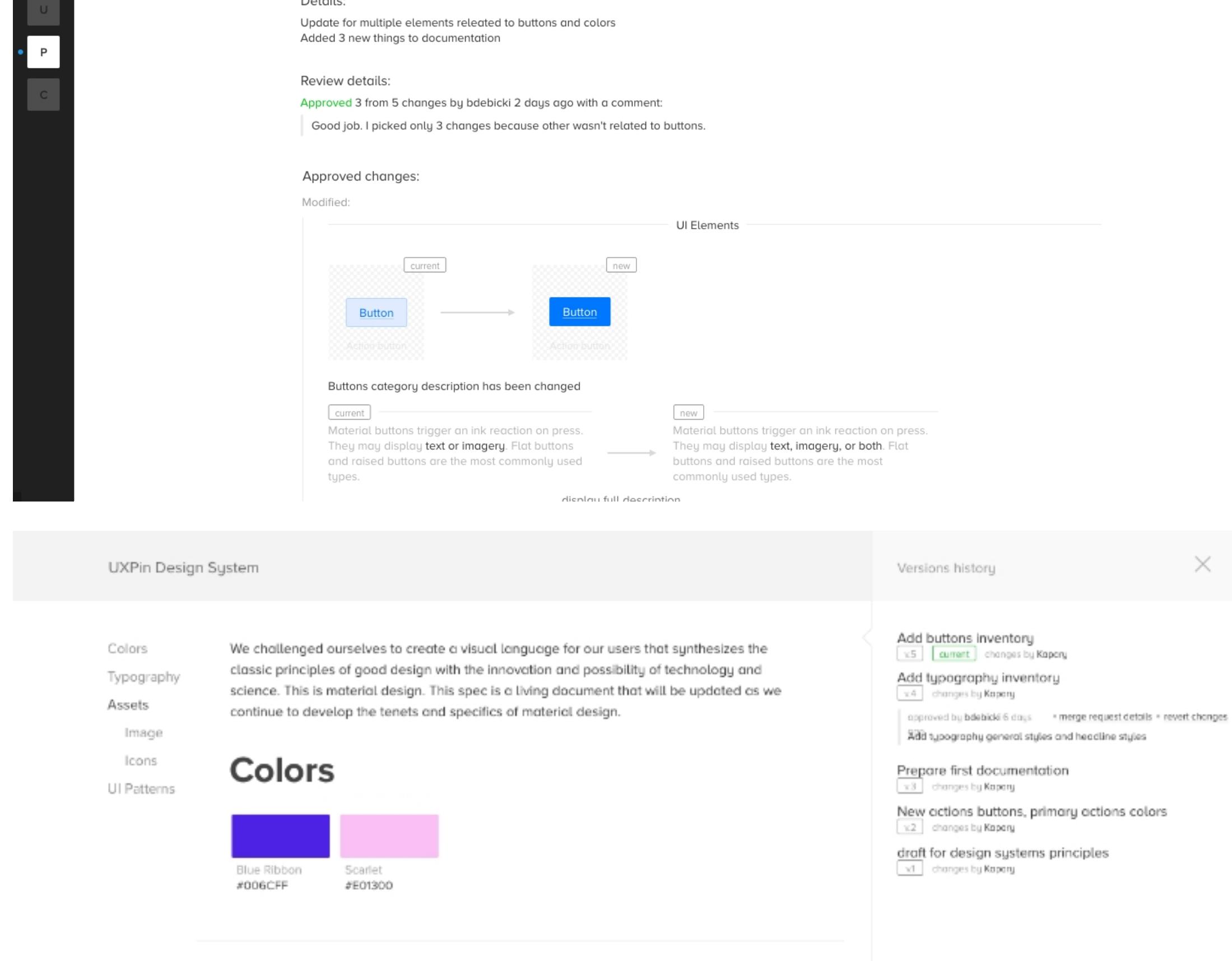
Finally, after technical analysis and user research, the project has been paused.

2017 | UXPin
role :product lead / ux design

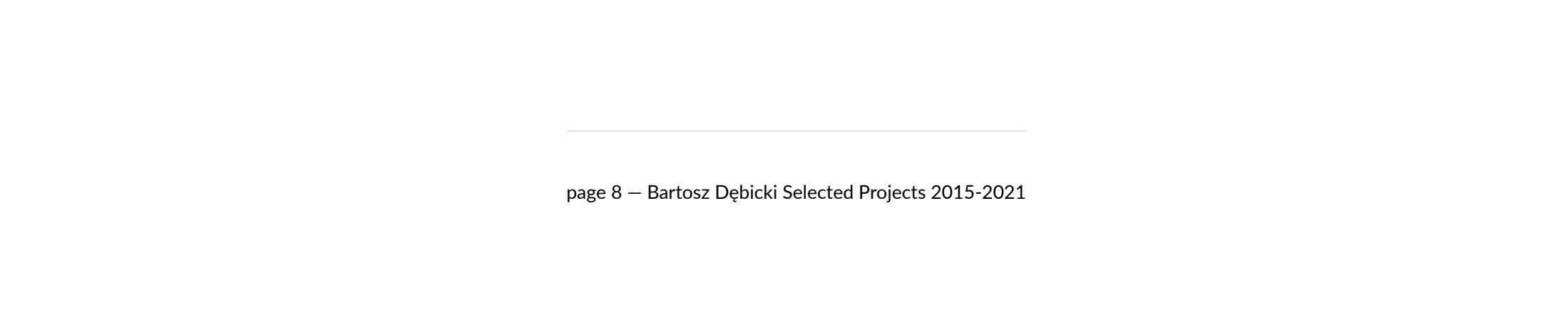
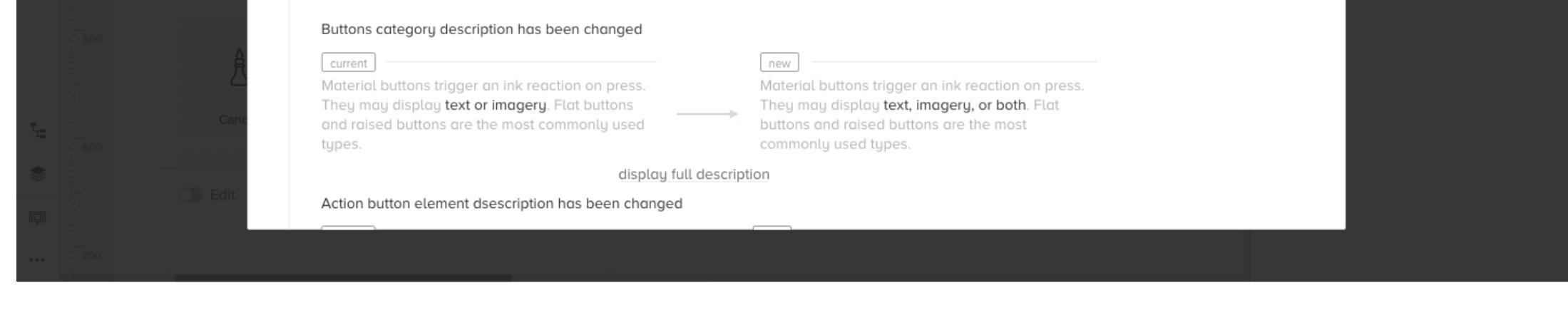
The Flow



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



rainning at open'er fe

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

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

rainning at open'er fe

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

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

rainning at open'er fe

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

LINEUP 2016

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 | WEBBER | MAŁPA | MARIA PESZEK | NOTHING BUT THIEVES
PTAKI | RASMENTALISM | REBEKA | RYSY | SECTION BOYZ
WINCE STAPLES | XANAXX | ZBIGNIEW WODESKI with MITCH & MITCH ORCHESTRA
AN ON BAST | BAASCH | CHINO | COLDAIR | CZELUŚC LIUTRO X KOSA | GROBEL | HANA
HEROES GET REMEMBERED | JAAAI | JÓGA | KUBA SOJKA | KROKI | LXMP | NAGROBKI
OLIVIER HEIM | OTSCHODZI | PIOTR BEJMAR | PIOTR ZIOŁA | SHY ALBATROSS | SONAR | SOOM T
SPOKEN LOVE | STARA RZEKA | SUUMO | T'NIEN LAI | TERRIFIC SUNDAY | V/O | WE DRAW A | ZAMILSKA

RAIN

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.

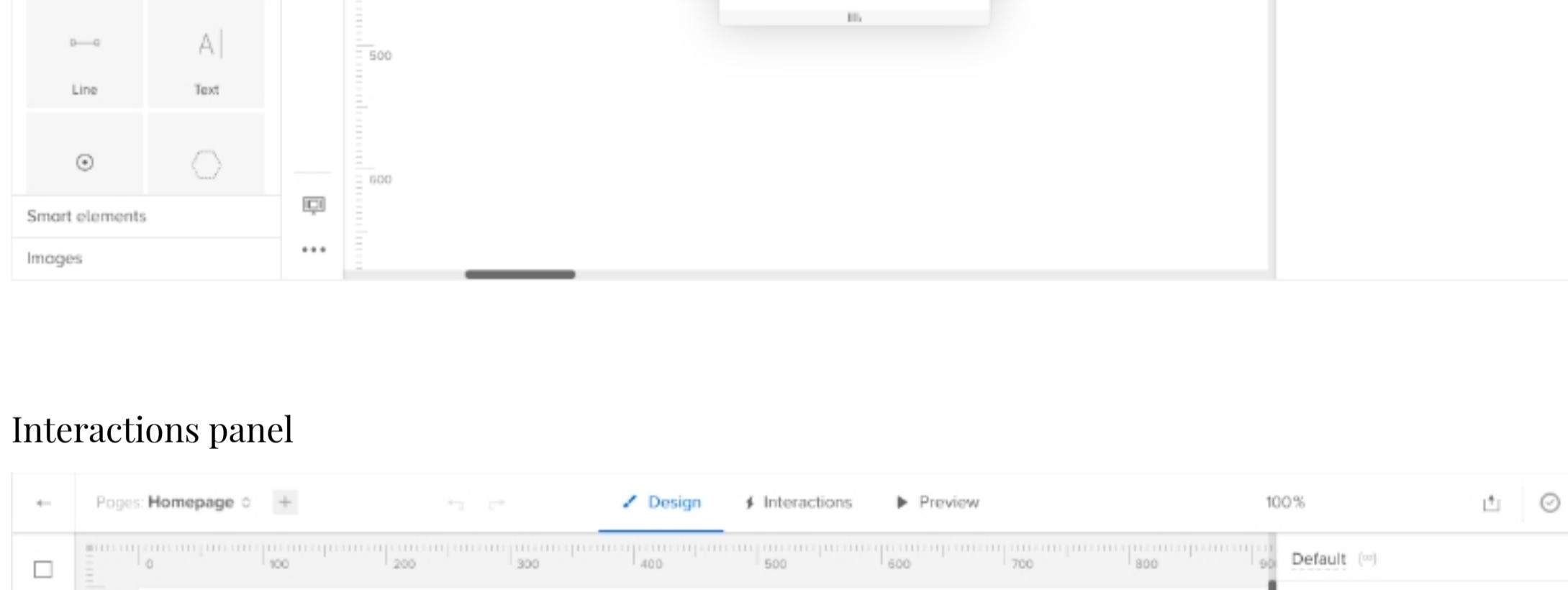
UI changes contain redesign of the top bar and release toolbar with quick access to the most popular elements and floating layers panel. Interactions flow was also simplified and enriched by predefined actions (based on the most popular ones).

2017 | UXPin
role :product lead / ux design

Top bar and toolbar



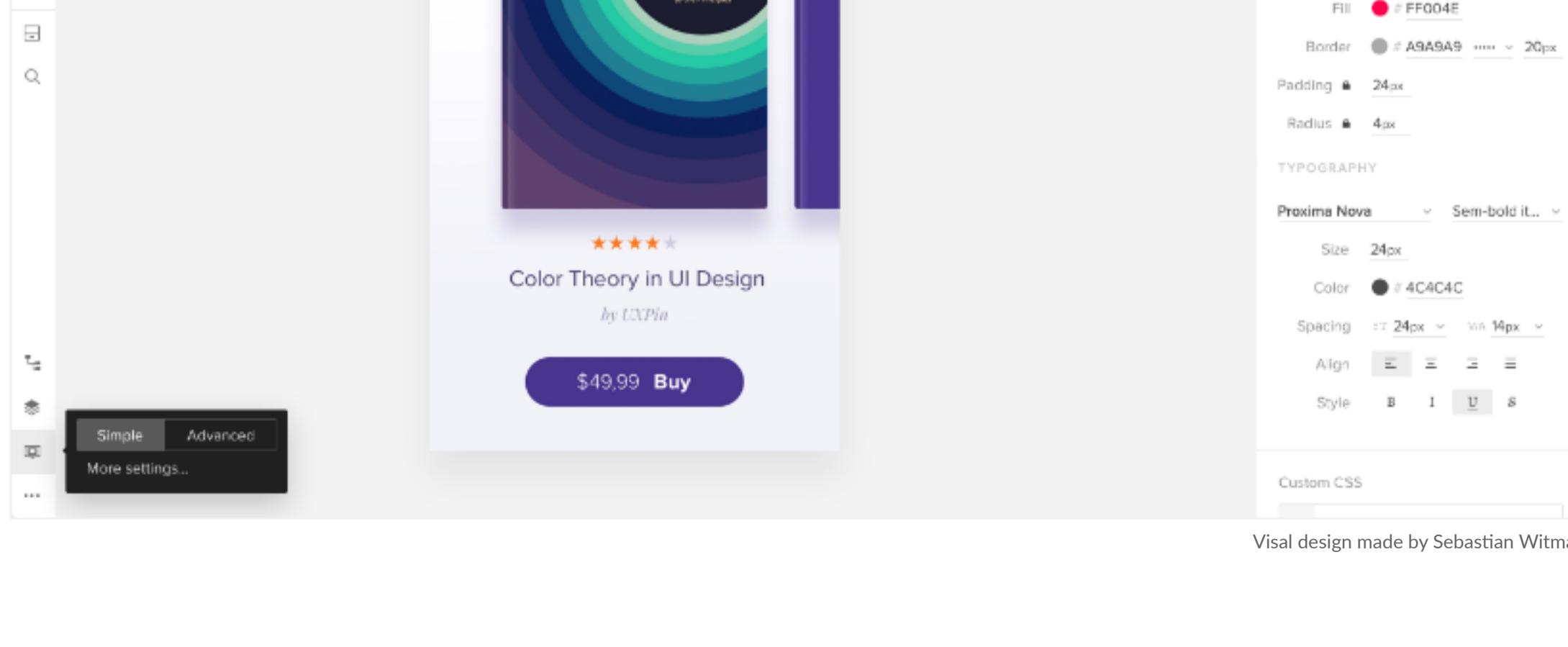
Left panels and floating layers



Interactions panel



Final redesign effect

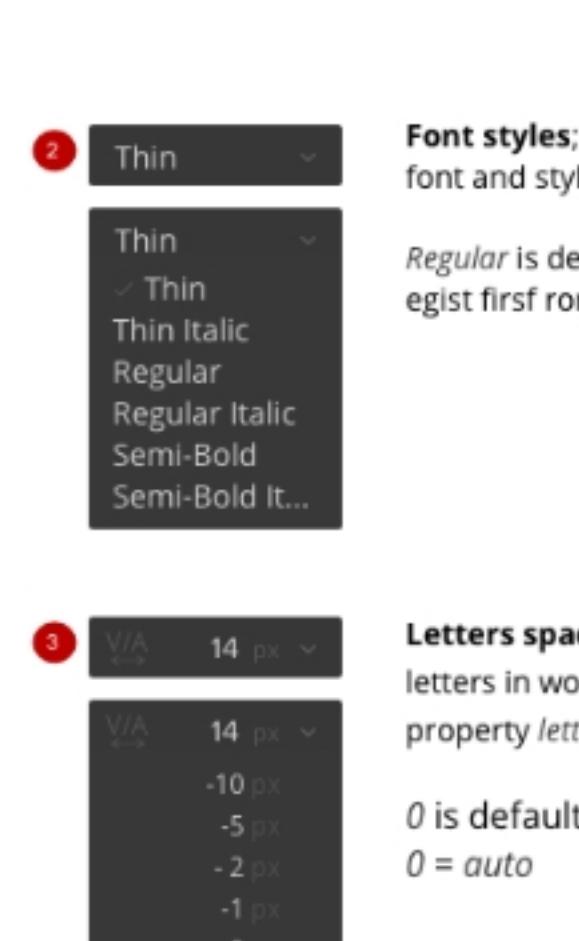
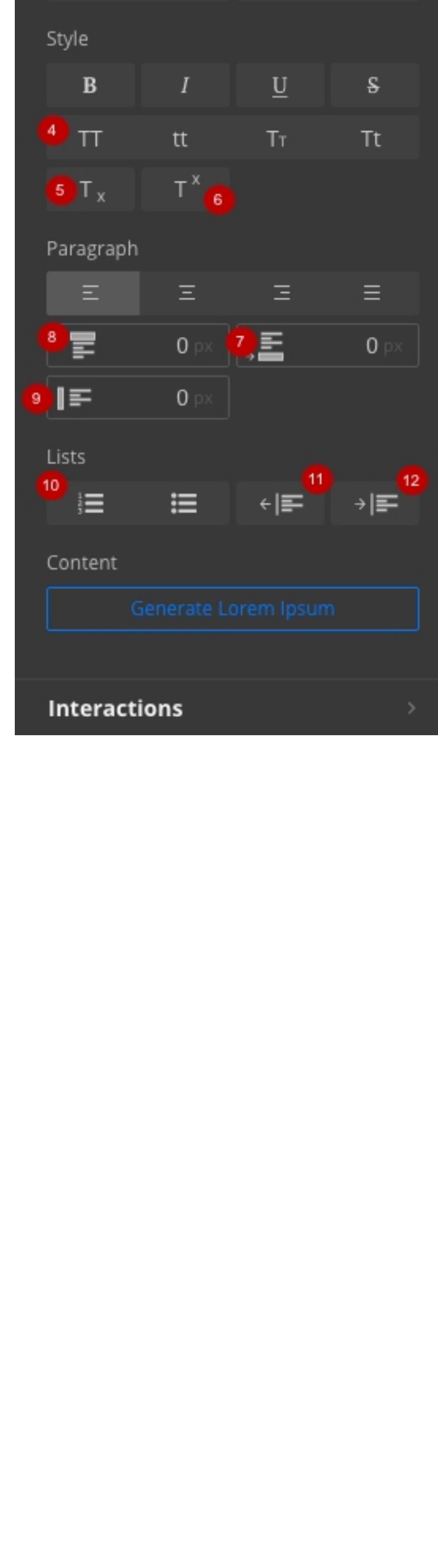
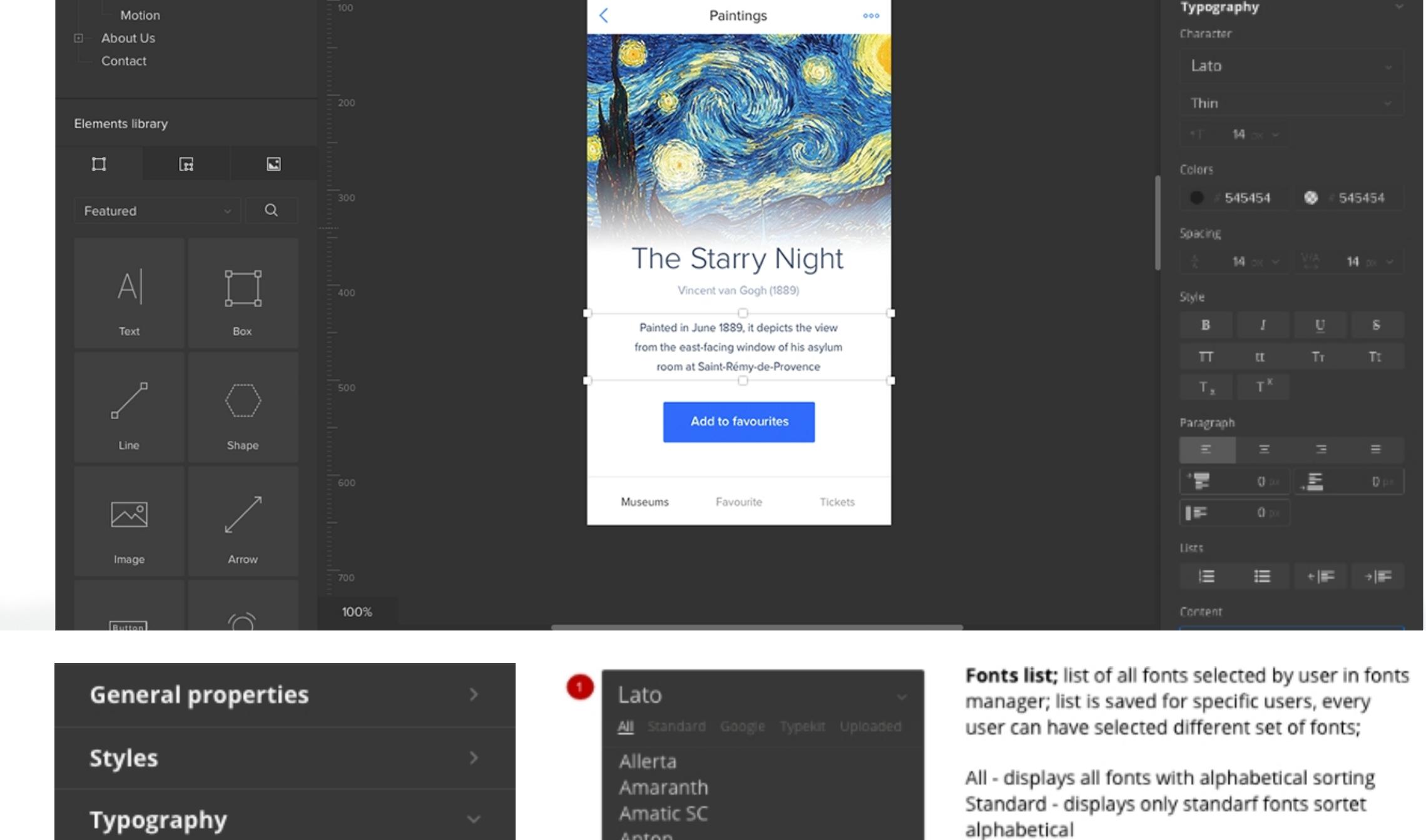


Advanced typography

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

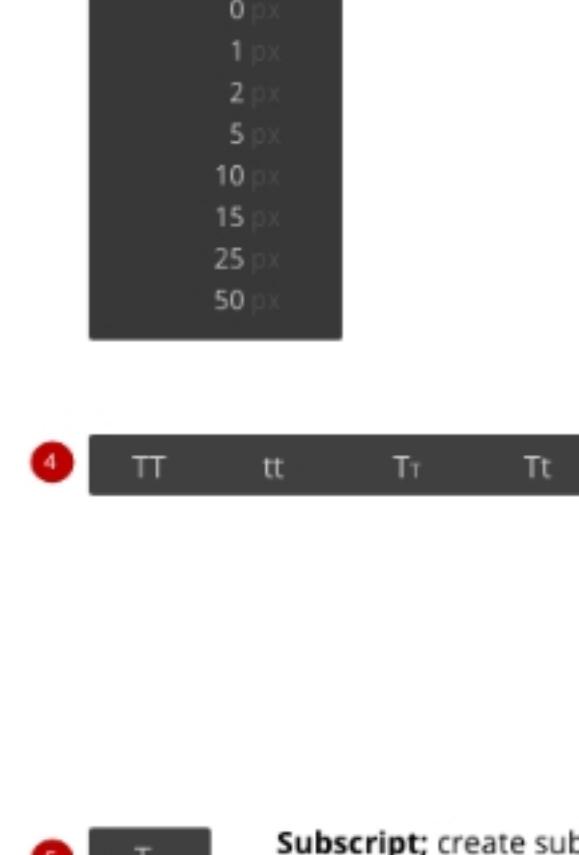


Fonst list; list of all fonts selected by user in fonts manager; list is saved for specific users, every user can have selected different set of fonts;

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

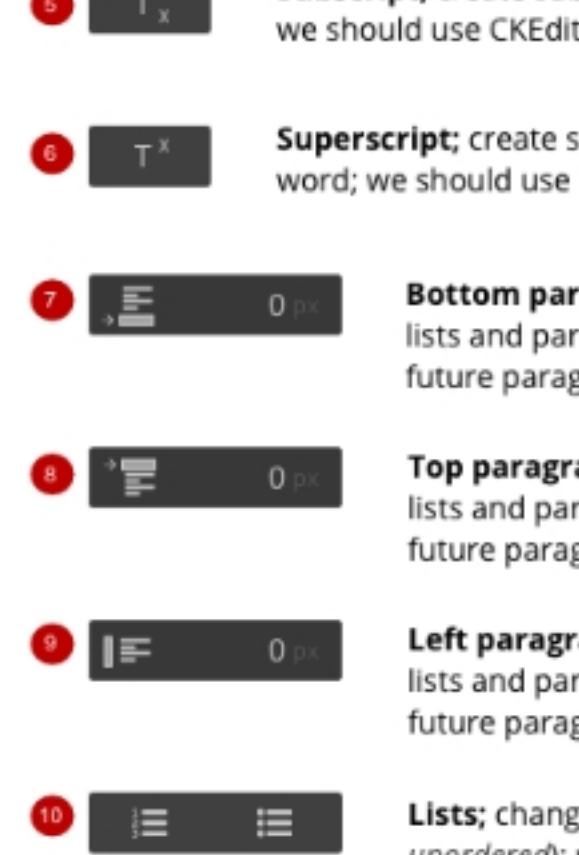
Selected category in list should be saved

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



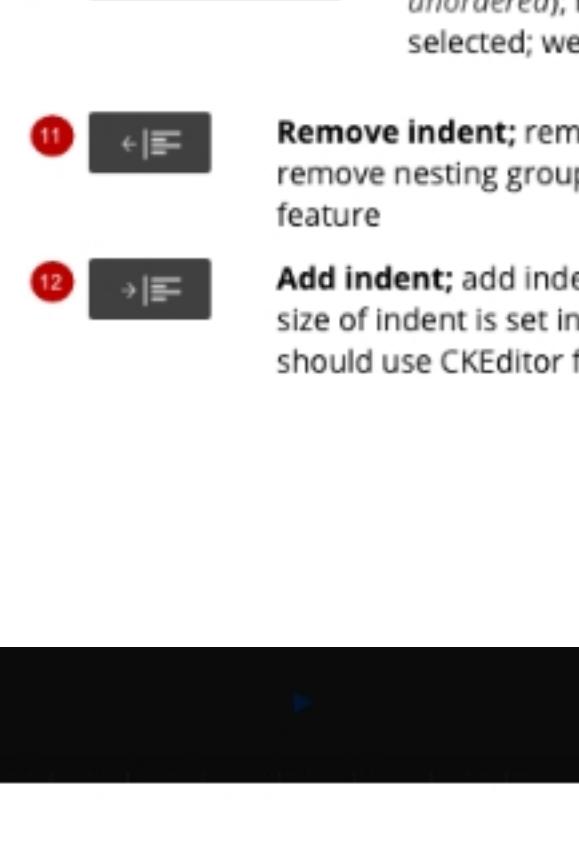
Font styles; list based on styles of font and styles added by user

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



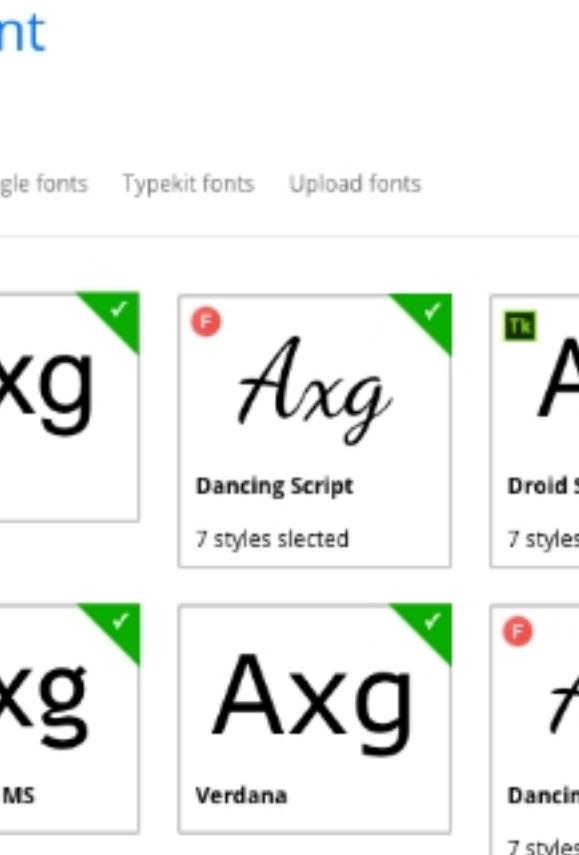
Letters spacing; space between letters in words; based on css property *letter-spacing*

0 is default
0 = auto

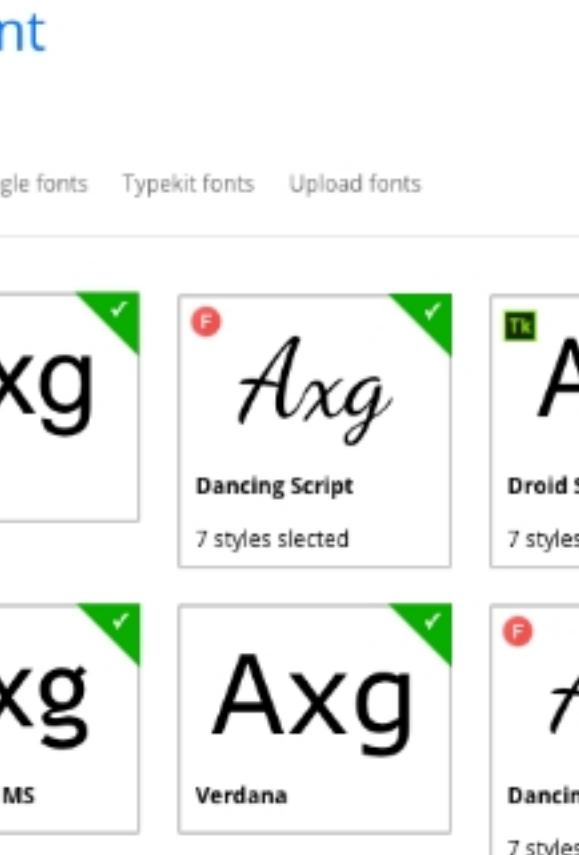


Text transformations; change case sensitive; none or only one option can be selected; based on css property *text-transform*

TT - uppercase (*text-transform: uppercase;*)
tt - lowercase (*text-transform: lowercase;*)
Tr - 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

Fonst manager

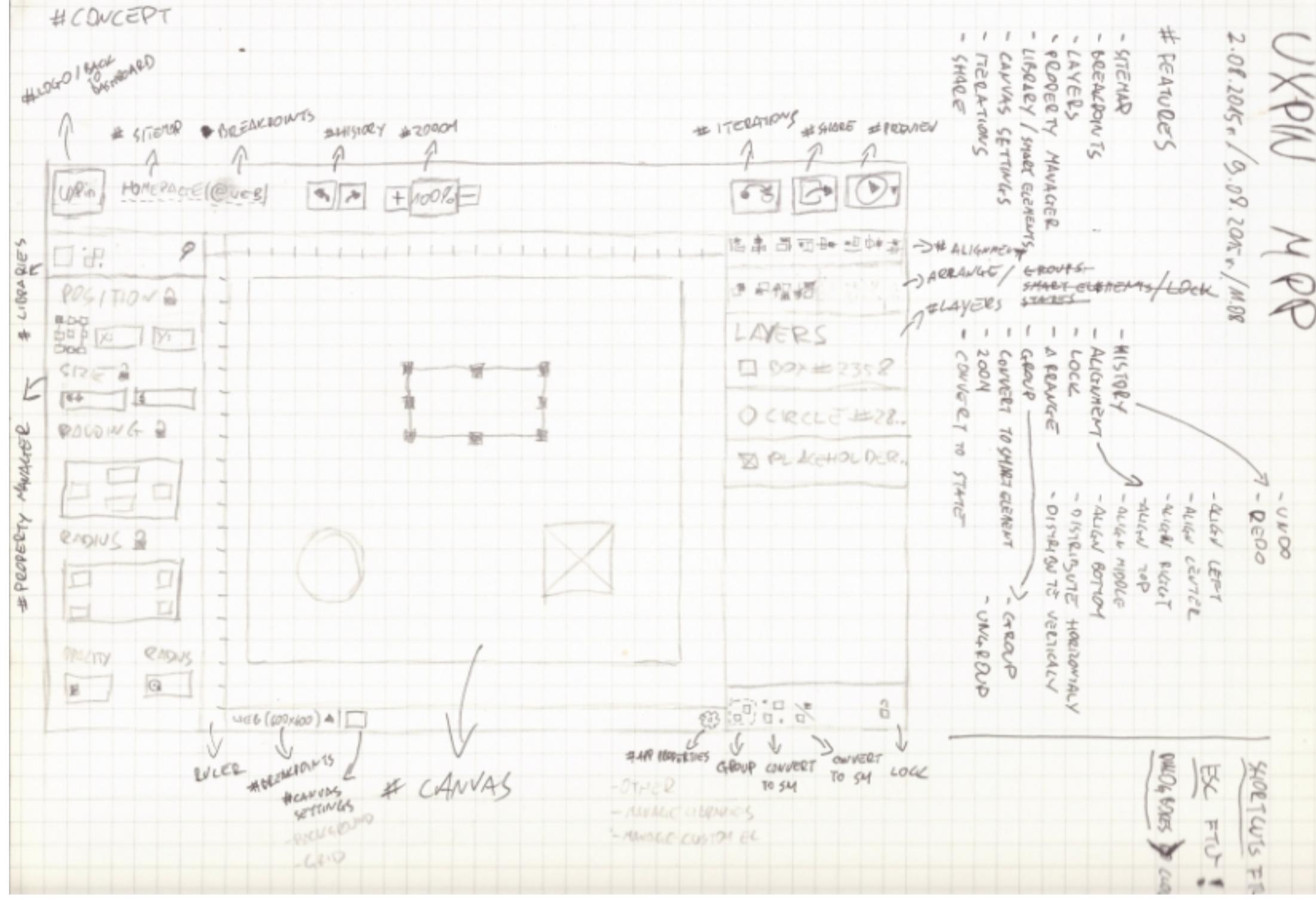


UXPin 3.0

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

