



ARIELLE CERINI

MFA
PhD

Creative Problem Solver & Strategic Thinker

Zenith Design System

[VIEW PORTFOLIO](#)

PHONE
[\(631\) 655-6245](tel:(631)655-6245)

EMAIL
arielle@acerinidesigns.com

WEBSITE
acerinidesigns.com

ADDRESS
411 South Street Mechanicville, NY 12118





Zenith

Design System

BAM Technologies

Worked to develop a design system to support an enterprise platform that was using a low-code development software called Outsystems. The design system was built to be completely themeable and with the intent of supporting all projects worked on at the organization.

[VIEW PROJECT PAGE](#)

Position

UX Manager & Platform Strategist

Role

Project Lead & Design System Architect

Tools & Technology

Adobe XD & Outsystems



Design Goal

Development teams are tasked with creating intuitive user experiences that are consistent across the platform. Presently, there is a lack of continuity throughout the platform, as well as a disconnect between the created mockups and the executed designs.

How can we empower teams to better adhere to platform standards and deliver the best end-user experience possible?

Assumptions:

The lack of consistency occurs within the platform because of the following: there is no on-boarding process for new developers; there is a lack of comprehensive documentation on what tools exist for the development teams; there is no central location where best practices and documents exist; there is a lack of awareness of the design system and what it is; the definition of done does not take the business language and user experience into consideration.

If teams were to have the information and guidance needed, we would see an increase in consistency across the platform.

The reason why repeat functionality exists within the platform is that teams were unaware that the tools they needed already existed.



Problem Statements

Throughout the company, there are the following:

- a lack of design resources available for individual project teams
- repeat functionality and UI Elements being built
- a lack of resources to resolve problems previously solved by others
- a lack of clearly defined standards and best practices

Within the platform, there are the following:

- valuable resources being wasted to rebuild already existing functionality
- a lack of time and resources devoted to maintaining and implementing a design system (both in design and development)
- a lack of knowledge on how to use the design system
- a lack of clearly defined best practices for UI design and front-end implementation



Solution

1. Create a fully searchable and dynamic style guide with clear documentation for using the design system.
2. Develop a company resource site that contains the best practices, resources, etc., for all disciplines working on development teams. This site should serve as an easy-to-use resource hub for the development teams; it is the source of truth.

The creation of this shared internal resource is intended to help support the growth and development of all people at the company and set employees up for success. Through the creation of this platform, it is believed that the company will be able to simplify the development process for creating new features, allowing a greater focus to be spent on ensuring teams are solving the right problems. In addition, creating a central resource hub for the company will allow for knowledge to be shared across the different projects. This will help to create more of a 'community' within the workplace and prevent time being spent resolving issues that the company has already invested time and resources into solving.



Scope

Phase 1

Creating and implementing design system in low-code environment

It is proposed that the initial focus for the team be on creating the design system for the low-code environments because that is where many of the company's high-priority projects exist. Although the emphasis will be placed on low code, all development work should be done so that it can be easily translated into high-code.

Phase 2

Expanding system to be used in high-code environments

In the second phase of this project, the focus will shift toward the high-code environments. Working in conjunction with teams currently using Angular, the design system will be translated into Storybook so that it can easily be consumed by their projects.



Outcome

There are several key outcomes for this project.

1. An intuitively designed design system that focuses on continuity, reusability, and communication, drastically decreasing the time taken to develop and implement new designs.
2. Creating a fully themeable design system that can be applied to all pre-existing and future projects.
3. The creation of readymade screen templates for all frequently used layouts and workflows, allowing developers to drag in a template and tie it directly into the backend.
4. The development of design resources and documentation
5. Custom internal libraries and resources to improve upon the design and development process



Metrics (Measuring Success)

Success can be measured through the analysis of the following :

- **Decrease in the time spent on stories in DevOps**

It is hypothesized that by providing teams with a well-established design system and accompanying documentation, there will be a **decrease in the time needed to create and implement a design solution**.

- **Decrease in the number of high-fidelity mockups**

It is hypothesized that by investing time and resources into the design system, we will be able to construct a library of ready-made template layouts and blocks that will **decrease the number of high-fidelity mockups required**, allowing development teams to shift their focus to solving the user's problems.

- **Time spent analyzing**

It is hypothesized that having a well-established design system will allow for an **increase in the time teams spend analyzing user needs** and **decrease the time focused on UI and front-end development**.



Team Composition

A team of designers and developers whose sole focus is working on the design system.



UI Developer
(QA Emphasis)



UI Developer
(High-code Focus)
Phase 2



Junior UX Designer
(Emphasis on UX writing)



UI Developer
(Low-code Focus)



Junior UI Developer
*Ideally Phase 1, but can
be Phase 2*



Junior UI/UX Designer
(Emphasis on UI design
and Accessibility)
Phase 2



Solution

Part One: Design System Structure

- Properties
- UI Element Breakdown
- Libraries

Part Two: Live Style Guide

- Micro Service Architecture
- Information Architecture
- Site Design



Part 1: Design System Structure

Properties



Design System Structure

Properties

[CSS_Lib](#)
[View in Browser](#)

[View in New Tab](#)

Color Theming

Color Theme One

	-color-primary:	-color-secondary:
Solid Colors	#FF0000	#FF174D
Color Tints	<ul style="list-style-type: none"> -color-primary-00: #FF2525 -color-primary-05: #FF3333 -color-primary-10: #FF4444 -color-primary-20: #FF5555 -color-primary-30: #FF6666 -color-primary-40: #FF7777 -color-primary-50: #FF8888 -color-primary-60: #FF9999 -color-primary-70: #FFAA99 -color-primary-80: #FFB299 -color-primary-90: #FFC399 -color-primary-100: #FFD499 	<ul style="list-style-type: none"> -color-secondary-00: #FF174D -color-secondary-05: #FF174E -color-secondary-10: #FF174F -color-secondary-20: #FF1750 -color-secondary-30: #FF1751 -color-secondary-40: #FF1752 -color-secondary-50: #FF1753 -color-secondary-60: #FF1754 -color-secondary-70: #FF1755 -color-secondary-80: #FF1756 -color-secondary-90: #FF1757 -color-secondary-100: #FF1758

Color Theme Two

	-color-primary:	-color-secondary:
Solid Colors	#FF1111	#FF1400
Color Tints	<ul style="list-style-type: none"> -color-primary-00: #FF2222 -color-primary-05: #FF3333 -color-primary-10: #FF4444 -color-primary-20: #FF5555 -color-primary-30: #FF6666 -color-primary-40: #FF7777 -color-primary-50: #FF8888 -color-primary-60: #FF9999 -color-primary-70: #FFAA99 -color-primary-80: #FFB299 -color-primary-90: #FFC399 -color-primary-100: #FFD499 	<ul style="list-style-type: none"> -color-secondary-00: #FF1400 -color-secondary-05: #FF1401 -color-secondary-10: #FF1402 -color-secondary-20: #FF1403 -color-secondary-30: #FF1404 -color-secondary-40: #FF1405 -color-secondary-50: #FF1406 -color-secondary-60: #FF1407 -color-secondary-70: #FF1408 -color-secondary-80: #FF1409 -color-secondary-90: #FF140A -color-secondary-100: #FF140B

Color Theme Three

	-color-primary:	-color-secondary:
Solid Colors	#FF0000	#FF174D
Color Tints	<ul style="list-style-type: none"> -color-primary-00: #FFA5A5 -color-primary-05: #FFB3B3 -color-primary-10: #FFC1C1 -color-primary-20: #FFD1D1 -color-primary-30: #FFE1E1 -color-primary-40: #FFEBEB -color-primary-50: #FFEBE8 -color-primary-60: #FFEBE5 -color-primary-70: #FFEBE2 -color-primary-80: #FFEBE0 -color-primary-90: #FFEBE8 -color-primary-100: #FFEBE5 	<ul style="list-style-type: none"> -color-secondary-00: #FF174D -color-secondary-05: #FF174E -color-secondary-10: #FF174F -color-secondary-20: #FF1750 -color-secondary-30: #FF1751 -color-secondary-40: #FF1752 -color-secondary-50: #FF1753 -color-secondary-60: #FF1754 -color-secondary-70: #FF1755 -color-secondary-80: #FF1756 -color-secondary-90: #FF1757 -color-secondary-100: #FF1758

[View in New Tab](#)

Shadows (for reference)

Drop Shadow Sizes

Size	Components associated with this size
xx-small	Button, Alert, Actions Menu (open)
x-small	Card
small	
medium	Floating Action, Pop Up, Modal
large	
x-large	Side Panel, Bottom Panel
xx-large	Top Navigation

Using Drop Shadow

Each component has its own level on the elevation scale. Start at +0 as small, and increase in level once interacted with. The closer the component is to the user the bigger its shadow.

Drop Shadow: Responsive Elevation

Pressed	+ one level
Hover & Focus	+ one level
Active	+ two levels
Dragged	+ three levels
Disabled	Remove all shadow

Text, Media Element, Link, Tags, Container Alert, Nav Card, Form Piece

Inset Shadow Sizes

Size	Components associated with this size
x-small	Pieces (Input radio checkbox)
small	Scroll Boxes (tables etc)
medium	TBA Larger View/Scroll Areas
large	

Using Inset Shadow

Input pieces that are interactive use inset shadows to show a user input "goes into them". This can also be applied to components that contain a view of an item, similar to a view master

Drop Shadow: Responsive Elevation

Hover & Focus & Active	+ one level
Disabled	Remove all shadow

[View in New Tab](#)

Typography

Font CSS Variables

Heading Sizes (Desktop)	Content Sizes	Font Weights
<ul style="list-style-type: none"> -font-size-display: 36px -font-size-h4: 18px -font-size-h1: 32px -font-size-h5: 16px -font-size-h2: 26px -font-size-h6: 16px -font-size-h3: 22px 	<ul style="list-style-type: none"> -font-size-base: 16px -font-size-s: 14px -font-size-m: 16px -font-size-l: 12px 	<ul style="list-style-type: none"> -font-light: 300 -font-regular: 400 -font-semi-bold: 600 -font-bold: 700

Typography Rules

Line Height	Justification	Hyphens
Line height for all font sizes: 1.5rem	Text should never be justified	Hyphens, for word breaks, are allowed, but should be used sparingly. If hyphens are present in a block of text, then it should be
Character Spacing	Alignment	The majority of text used within the Myvector Platform should be left-aligned.
Header Text: 1.2em	Paragraph Text: 0.4em	Long blocks of text should always be left-aligned.
Paragraph Spacing	Header Text: 1.5em	Text within tables: left aligned
Header Text: 1.2em	Body Text: 1.0em	Button Text: centered
Paragraph Text: 0.4em		Right- and center-alignment should be used sparingly, and should never be applied to large blocks of text.



Design System Structure

Properties

context	design attribute	property	state/varient	hierarchy
body	color	background	—	—

\$body-color-background

Default background color for the whole app.



Design System Structure

Properties

Design System Master List

[View in Browser](#)

context	design attribute	property	state/variant	hierarchy	Token	Use Cases
body	color	background			\$body-color-background	Default background color for the whole app.
field	color	text	disabled		\$field-color-text-disabled	text color for disabled fields
field	color	background	disabled		\$field-color-background-disabled	Background color for disabled fields
field	color	border	disabled		\$field-color-border-disabled	Border color for disabled fields
field	color	background			\$field-color-background	background color for fields/input content
field	color	border			\$field-color-border	border color for fields/input content
field	color	text			\$field-color-text	text color for fields/input
field	color	border	active		\$field-color-border-active	border color for active fields/input content
field	color	border	hover		\$field-color-border-hover	border color for hovered fields/input content
field	color	border	error		\$field-color-border-error	border color for error validation fields/input content
focus	color	border			\$focus-color-border	focus border
header	color			1	\$header-color-01	main header text color
highlight-search	color	background			\$highlight-search-color-background	
icon	color			1	\$icon-color-01	Primary Icons
icon	color			2	\$icon-color-02	Secondary Icons
icon	color			3	\$icon-color-03	icons displayed inline with text
image-overlay	color	background			\$image-overlay-color-background	Color of mask overlay that sits on top of an image when text is present.
inverse	color	background			\$inverse-color-background	Default background color for dark portions of the app (like Stage Left or tooltips).
inverse	color	background	light		\$inverse-color-background-light	Light variant of COLOR_BACKGROUND_INVERSE.
link	color				\$link-color	Primary links
link	color		hover		\$link-color-hover	
link	color		active		\$link-color-active	color of link that is active
link	color		disabled		\$link-color-disabled	color of link that is disabled
link	color		focus		\$link-color-focus	color of link that is being focused on
link	color		visited		\$link-color-visited	color of a link that has been visited
login	color	background			\$login-color-background	background color for the login page
notification	color	background	new		\$notification-color-background-new	background color for a new notification item
overlay	color	background			\$overlay-color-background	color that goes over content when you are in a modal state
post	color	background			\$post-color-background	background color for a post



Part 1: Design System Structure

UI Element Breakdown



Design System Structure

UI Element Breakdown: Atomic Design



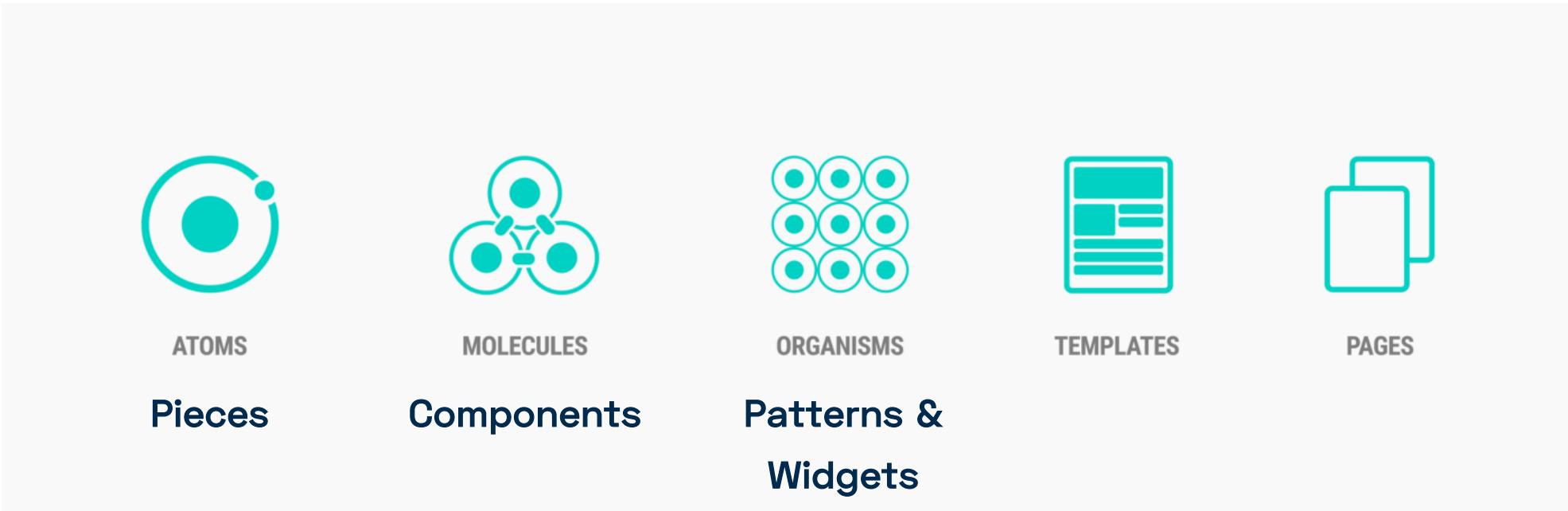


**Initially tried Atomic Design, but
development teams had a tough time
understanding the terminology.**



Design System Structure

UI Element Breakdown: Atomic Design (Inspired)





Design System Structure

UI Element Breakdown: Atomic Design (Inspired)

Properties

Pieces

Components

Patterns

Widgets

Templates



Design System Structure

UI Element Breakdown

Properties: the design tokens, classes, and principles the element uses

Pieces: the elements that make up the component; pieces can never stand alone

Components: a unique UI element with no intended purpose; for it to exist in a design, it needs to be given a purpose for being there

Patterns: a unique UI element with a specific purpose

Widgets: a UI element or series of UI elements that are tied to business logic

Templates: a single page or series of pages that represent a complete business solution

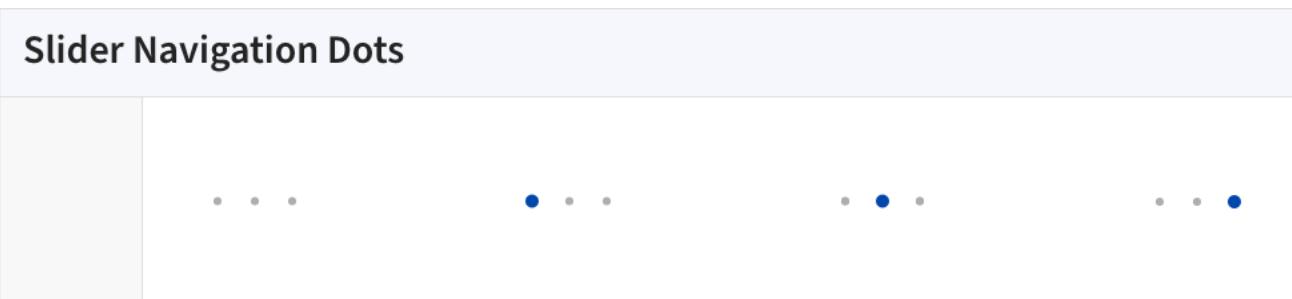


Design System Structure

Pieces

Pieces are localized to a specific library and are necessary for building other UI Elements but cannot stand on their own.

An example of a piece is a Slider Navigation Dot. The navigation dots are repeated throughout different elements but will never be used on their own and aren't needed outside of the context of a slider. Creating a reusable piece makes it easier to maintain consistency across different elements and streamlines the process of updating them.





Design System Structure

Components

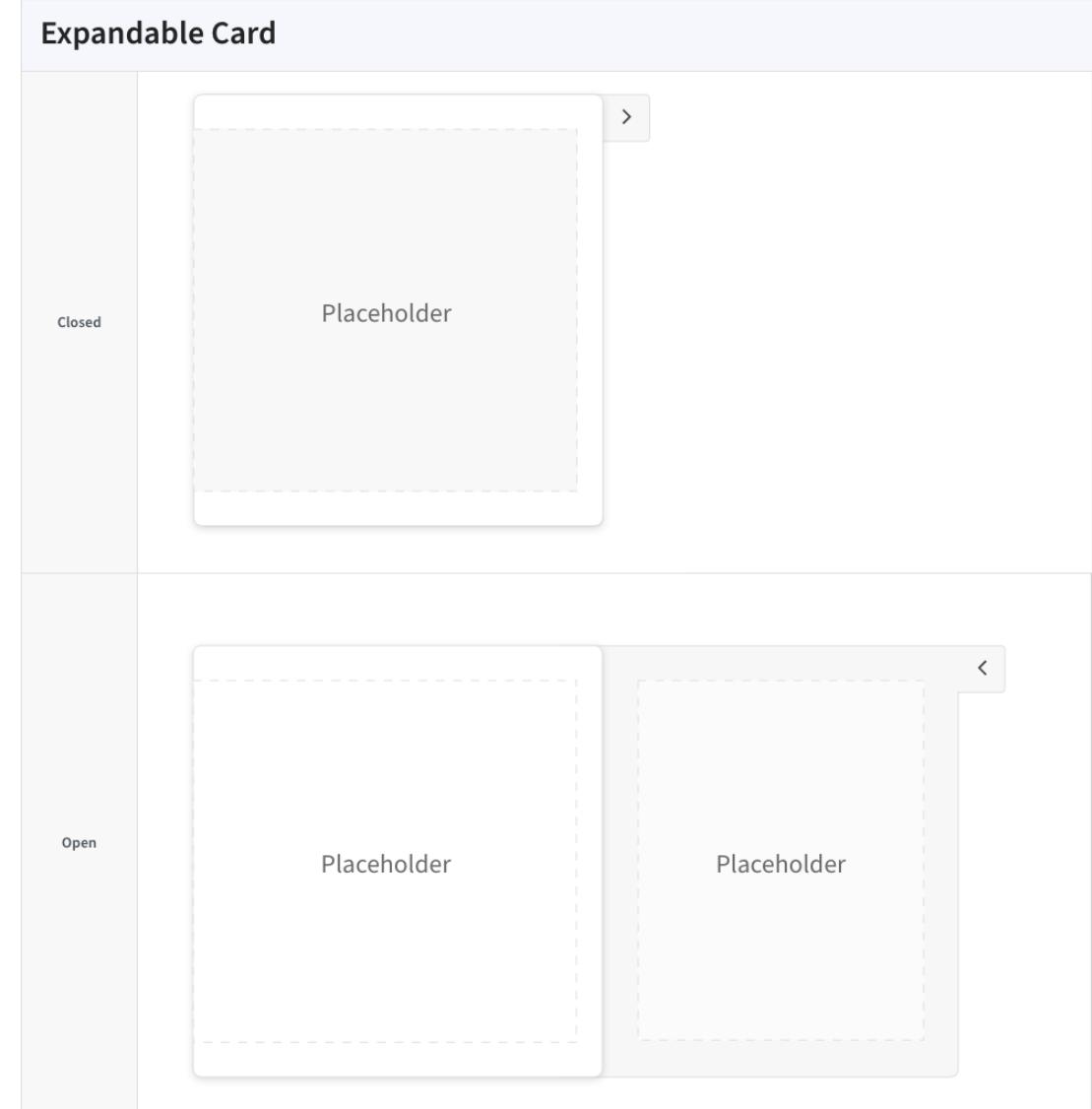
Components are distinct variations of a base UI Element. An example of a component is an Expandable Card.

The base UI Element is a Card, and the distinction is the fact that the card is expandable.

Naming conventions:

Distinction + Base UI Element

UI Flow: Plural form of the Base UI Element





Design System Structure

Patterns

Patterns are versions of components that have a fixed purpose. An example of a pattern is a Process Tracker Expandable Card. The purpose is to track a process, the distinction is that the card is expandable, and then the base UI element is a card.

Naming conventions:

Purpose + Distinction + Base UI

UI Flow: Plural form of the Distinction + Base UI

Element

Process Tracker Expandable Card

The screenshot shows two views of the Process Tracker Expandable Card. The left column is labeled 'Default' and the right column is labeled 'Example Content'. Each view consists of a large central card with a header placeholder and a description placeholder, and a smaller expandable section on the right containing a list of application dates. In the 'Example Content' view, the main card includes course title, category, descriptions, completion status, and enrollment details, along with a 'Review Training' button.

View	Content
Default	Process Heading Placeholder Process Description Placeholder
Example Content	Completed Course Title Course Category A description about this specific training can go here and A description about this specific training can go here Completed Online Completed 30 March 2022 Enrolled 20 March 2022 Review Training

Default View Content:

- Process Heading Placeholder
- Process Description Placeholder

Example Content View Content:

- Completed
- Course Title
- Course Category
- A description about this specific training can go here and A description about this specific training can go here
- Completed Online Completed 30 March 2022
- Enrolled 20 March 2022
- Review Training

Expandable Section Content:

- Application Date: Completed 12 December 2021



Part 1: Design System Structure

Libraries



Design System Structure

Libraries



Design System Breakdown

[View in Browser](#)

Taking an inventory of the system to determine how best to organize the UI Elements.



Design System Structure

Libraries

ID	Name	UI Layer (later)	UI SubLayer (later)	Description (keep)
1	CoreUI	Base Foundation	BottomLayer	HTML elements, stylistic components, and basic interactive elements.
2	DynamicDataUI	Content	SecondaryLayer	Patterns and components related to data visualizations. This is where the form and information UI is combined.
3	FormUI	Content	BaseLayer	Form input patterns and pre-made form templates.
4	InformationUI	Content	BaseLayer	Tags, alerts, typesetting, and other patterns and components related to displaying/encoding information.
5	LayoutUI	Base Foundation	SecondaryLayer	Sections, containers, cards, and other patterns and components related to the layout of the page.
6	ThemePatterns	Theme Structuring And Styling	TopLayer	Patterns constructed to be used within pre-defined containers and layouts. This includes page layout patterns, application progress patterns, header patterns, etc.
7	ResourceUI	Base Foundation	BottomLayer	JavaScript libraries containing basic functionality required throughout the platform. (In the near future, this will be merging with ReusableClientActions_Lib)
8	ReusableClientActions	Base Foundation	BottomLayer	Client actions that are not contained within basic OutSystems UI functionality. These are required throughout the platform to ensure consistent formatting of information.
9	CSS	Theme Structuring And Styling	TopLayer	Reusable CSS classes that are built using CSS variables so they can be repurposed across OutSystems projects.



Design System Structure

Libraries

Primarily Components

Core UI

Layout UI

Primarily Patterns

Information UI

Form UI

Widgets, Templates, and Patterns
(Not Themeable)

Theme Patterns

Dynamic Content



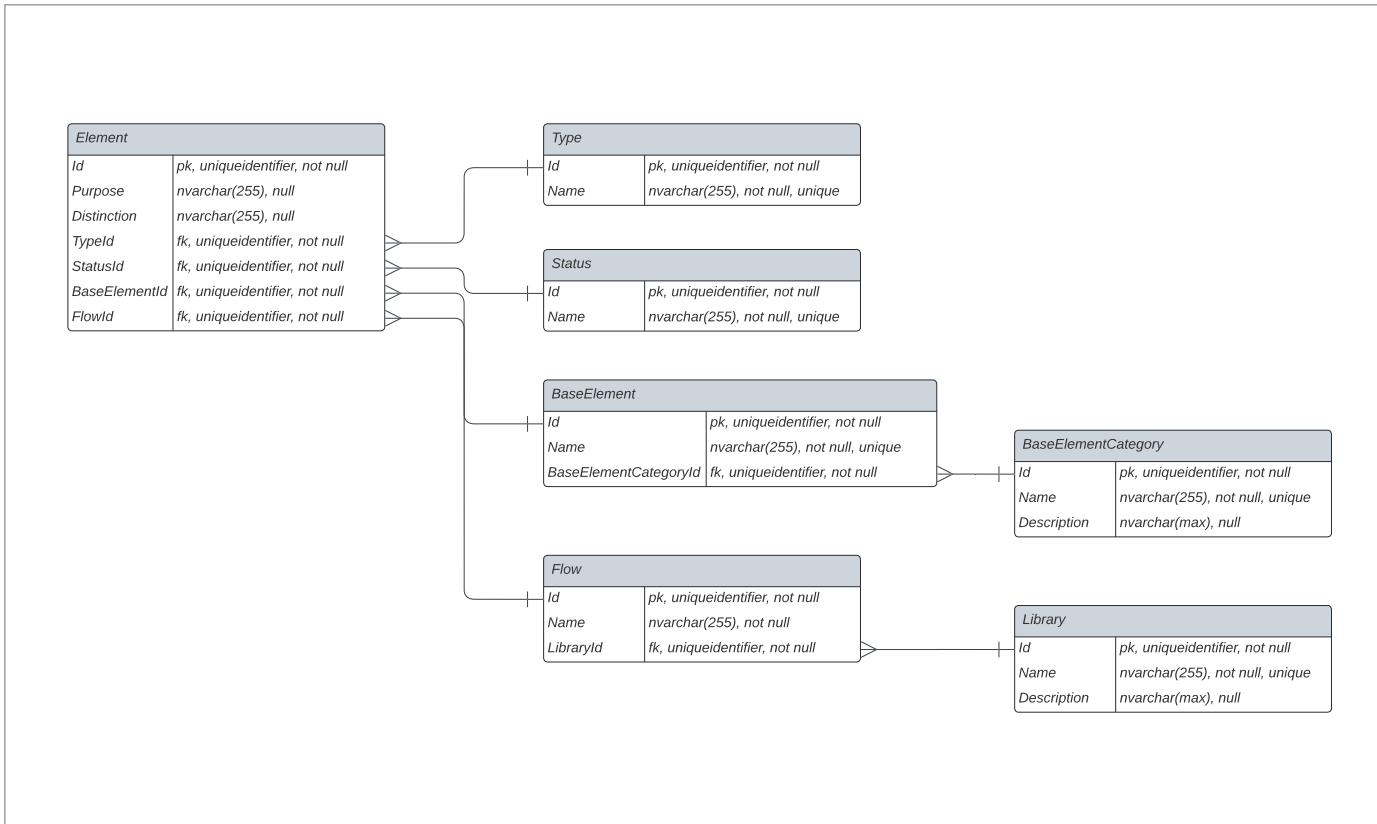
Part 2: Live Style Guide

Micro Service Architecture



Live Style Guide

Micro Service Architecture





Live Style Guide

Micro Service API

Element	Type	BaseElementCategory
<pre>{ "id": 1, "purpose": "Progress", "distinction": "Bar", "type": { "id": 1, "name": "Component" }, "status": { "id": 1, "name": "In Development" }, "baseElement": { "id": 1, "name": "Accordion", "baseElementCategory": { "id": 1, "name": "Input Controls", "description": "Description goes here" } }, "flow": { "id": 1, "name": "In Development", "library": { "id": 1, "name": "CoreUI", "description": "Description goes here" } } }</pre>	<pre>{ "id": 1, "name": "Component" }</pre>	<pre>{ "id": 1, "name": "Input Controls", "description": "Description goes here" }</pre>
		Status
	<pre>{ "id": 1, "name": "In Development" }</pre>	Flow
		<pre>{ "id": 1, "name": "In Development", "library": { "id": 1, "name": "CoreUI", "description": "Description goes here" } }</pre>
	BaseElement	Library
	<pre>{ "id": 1, "name": "Accordion", "baseElementCategory": { "id": 1, "name": "Input Controls", "description": "Description goes here" } }</pre>	<pre>{ "id": 1, "name": "CoreUI", "description": "Description goes here" }</pre>



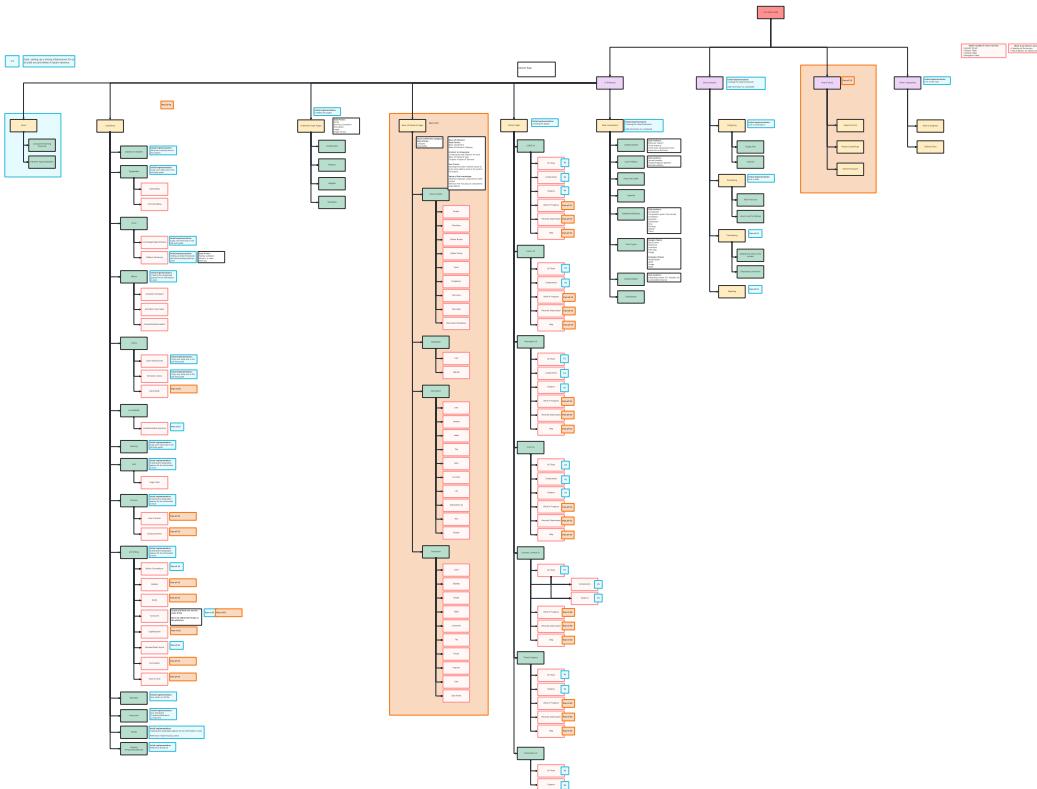
Part 2: Live Style Guide

Information Architecture



Live Style Guide

Information Architecture



Live Style Guide IA

[View in Browser](#)



Part 2: Live Style Guide

Site Design



Live Style Guide

End Design

The screenshot shows the Zenith Live Style Guide interface. On the left is a dark sidebar with navigation links: About, What's Happening, Designing, Developing, Contributing, Migrating, Guidelines, Components, Overview, and Text (which is currently selected). The main content area has a header "Components / Core Library" and a dropdown menu set to "Text". A sub-menu for "Text" is open, showing "Heading 1" as the selected item, along with "Heading 2", "Heading 3", "Heading 4", "Heading 5", "Heading 6", "Body Paragraph", "Small Paragraph", and "X-Small Paragraph". To the right of the sub-menu, there is a large text block with the heading "Text hierarchy" and a sub-section titled "Headings". Below this is a detailed description of text hierarchy: "Text hierarchy helps the end-user understand the relationships between the content on the page. In addition, screen readers use text hierarchy to understand how content on the page relates to one another. Think of the heading values like bullet points. H1 is the highest level, next h2. You would never go from the highest bullet point to the third or fourth indented value, the same is true for heading elements." At the bottom of the page, there is a section titled "Usage Guidelines" with the following text: "Every page should contain an h1 element. There should never be more than one h1 element on a page. The h1 element should always be the first text item on the page." On the right side of the main content area, there are two download buttons: "Download UI Kit" (COREUI.Lib.XD) and "Download Font Files" (SourceSans-VF.ttf), each accompanied by its respective icon.



Final Design

Implemented Solution

Project Page

[View Project Page](#)

Additional process documentation is available in the shared folder.

Other Details:

SASS followed a 7-in-1 architecture.

The same naming conventions were followed for both design and development work.

A series of classes were constructed using mix-ins that allowed developers to figure out what classes existed in the system quickly.

position: absolute -> .position-absolute

Audits were conducted to help with the following initiatives:

- defining iconography
- creating repeat patterns i.e. person cards
- creating a series of reusable dashboard layouts

...