Design System & Prototyping

Co-funded by the European Union

shift4T

Funded by the European Union. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or Agency for Mobility and EU Programmes. Neither the European Union nor the granting authority can be held responsible for them.

Design System & Protoyping

- Summary
- Introduction to Design System
- Conducting a Design System Project
- Tips and Tricks
- Different types of UI
- Ressources
- Additional Reading

Summary

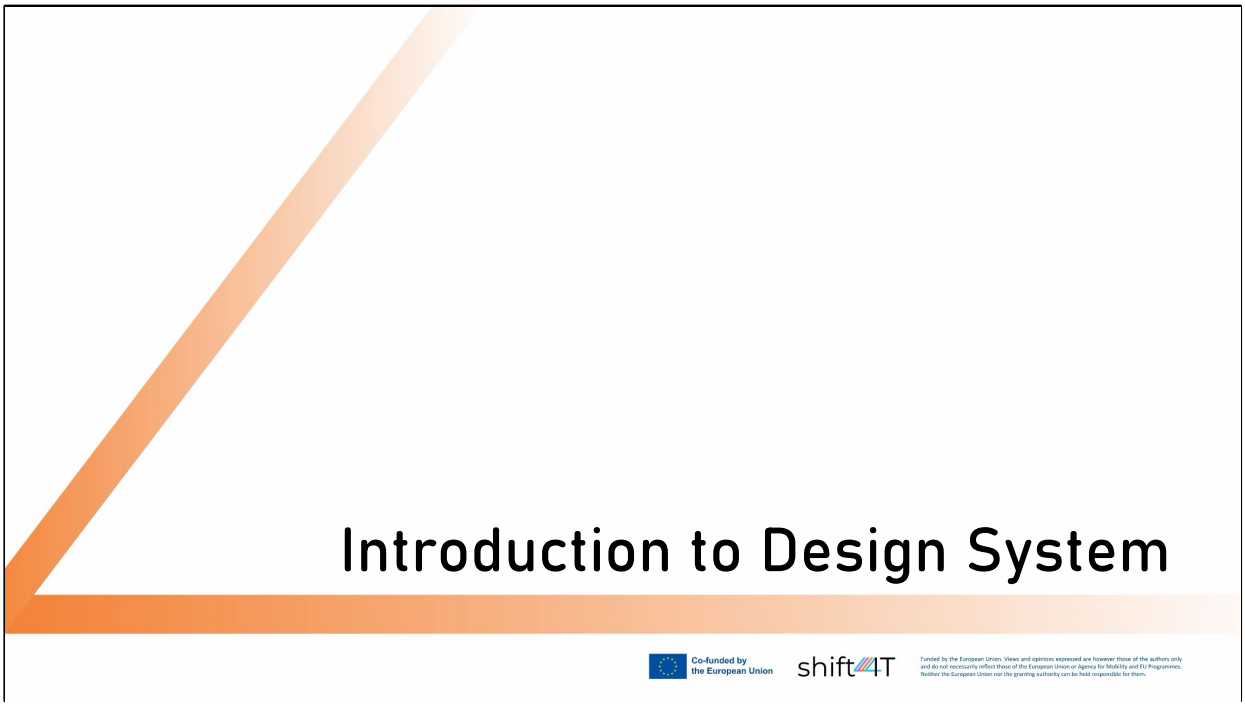
As we wrap up this module, you **will take** a step back and **recognize** how far you **will have come**.

In the upcoming **Design Systems** sessions, you **will learn** how to create consistency across a product by building reusable components, setting clear visual guidelines, and maintaining design coherence. These systems **won't** just be about aesthetics — they **will support** smoother collaboration and more efficient design workflows.

In the **Prototyping** part of the course, you **will explore** how to bring your ideas to life. From wireframes to high-fidelity prototypes, you **will practice** how to test and validate your designs before they ever reach development.

You **will learn** how to communicate user flows, gather feedback, and iterate quickly. Together, these skills **will become** powerful tools to help you design smarter, communicate better, and build products users truly value.

Keep practicing, keep testing — and always design with purpose!



📌 **Objective:** Introduction to Design Systems and Prototyping in UX/UI.

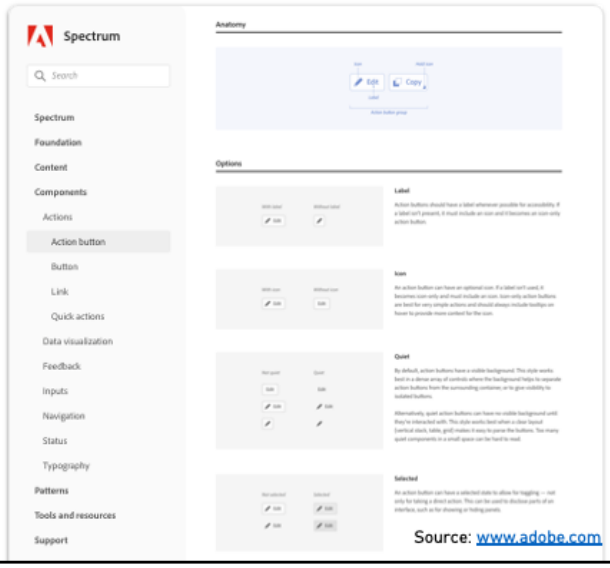
🔷 **Key Takeaways:**

- A **design system** is a structured set of **guidelines, best practices, and reusable UI components**.
- It ensures **consistency**, reduces **redundancy**, and creates a **shared language** for design and development teams.
- A well-implemented **design system** improves collaboration and efficiency.

What is a Design System ?

The design system translates into a set of rules, best practices, reusable elements and resources that are made available to the different teams. We can see the design system as a large toolbox common to all teams that would bring ease and consistency when building interfaces.

A design system is a set of standards to manage design at scale by reducing redundancy while creating a shared language and visual consistency across different pages and channels.



📌 **Objective:** Define what a design system is and its core components.

🔷 **Key Takeaways:**

- A design system is a **toolbox** for product teams, providing:
    - **Standardized components** (buttons, inputs, typography).
    - **Guidelines for branding, UX, accessibility, and development.**
    - **Patterns for consistent user experiences.**
- It facilitates **cross-team collaboration** and scales design efficiently.

🔗 **Examples:**

- Polaris (Shopify)
- Material Design (Google)

The benefits of the design system

- Consistency of the user experience
- A reinforced identity on all media and platforms
- Save time and money
- Centralized maintenance
- Simplified communication within the team (see slide n° 10)
- Encourages co-creation
- A system validated by the various experts (UX / UI / PO / Tech / Accessibility)
- It's a gain in terms of Consistency, Time and Money

📌 **Objective:** Highlight the advantages of using a design system.
🔷 **Key Benefits:**
✅ **Consistency** across products and platforms.
✅ **Time and cost savings** (reducing redundant work).
✅ **Faster development cycles** with reusable components.
✅ **Centralized maintenance**, making updates easier.
✅ **Improved communication** between designers, developers, and stakeholders.
🔗 **Additional Reading:**

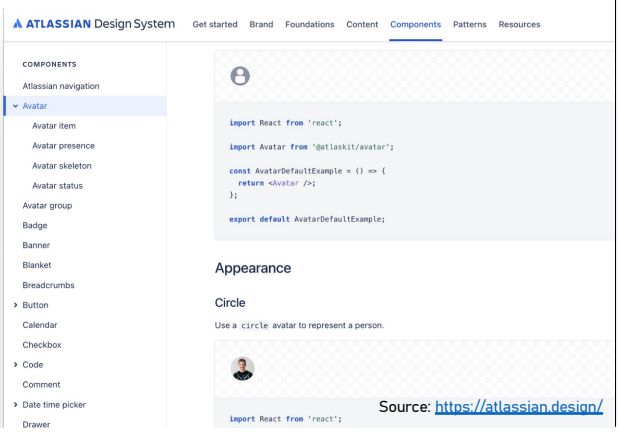
- Design System ROI Calculator https://www.designsystems.com/

Examples

Atlassian

- https://atlassian.design/
- Rules of use, design, code, editorial

- Components
- Patterns
- Brand
- Foundations
- Thrilled



Source: https://atlassian.design/

📌 **Objective:** Showcase real-world design systems.
🔷 **Key Takeaways:**

- Large companies maintain public design systems for **scalability and efficiency**.
- Example systems:
- **Atlassian** – Focuses on **foundations, components, design tokens**.
- **IBM Carbon** – Emphasizes **accessibility, development, and data visualization**.
- **Shopify Polaris** – Built for **e-commerce experiences**.
- Design tokens are **scalable variables** for colors, typography, and spacing.

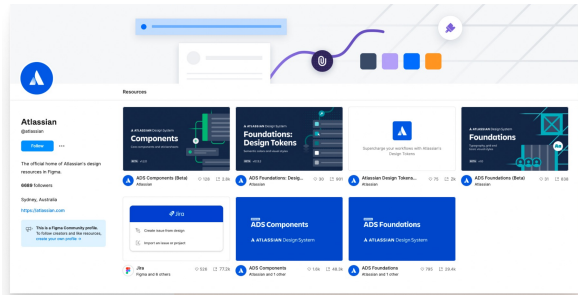
🔗 **Additional reading:**
https://atlassian.design/

Examples

Atlassian

- Foundations
- Components
- Design tokens

- https://www.figma.com/@atlassian

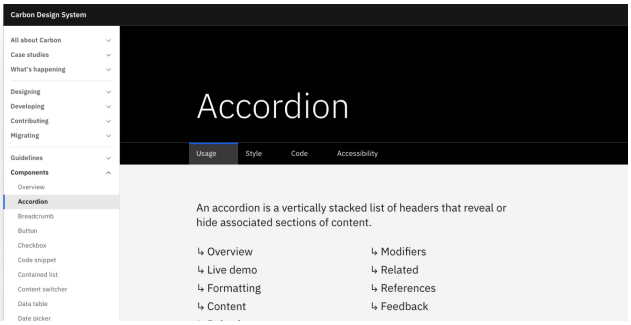


Source: https://www.figma.com/@atlassian

Examples

Carbon

- Rules of use, design, code, accessibility
  - Designing
  - Developing
  - Contributing
  - Migrating
  - Guidelines
  - Components
  - Patterns
  - Community assets
  - Data visualization

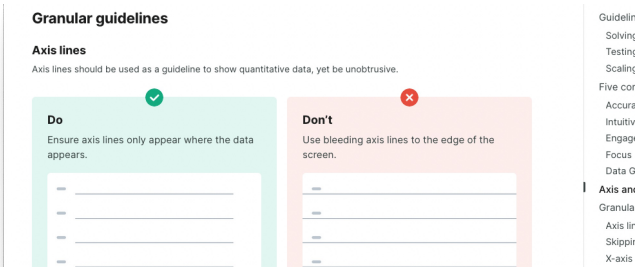


Source: https://www.figma.com/@ibmorg

Examples

Polaris (Shopify)

- https://polaris.shopify.com/



Source: https://polaris.shopify.com/

Some principles of a design system

- The design system is never finished, it is built and evolves with the product. To be successful, you have to start small.
- The design system is not just about designers, it is a full-fledged product that needs a team, referents, a vision and rules.
- Everyone's collaboration is the key point that defines a good design system.

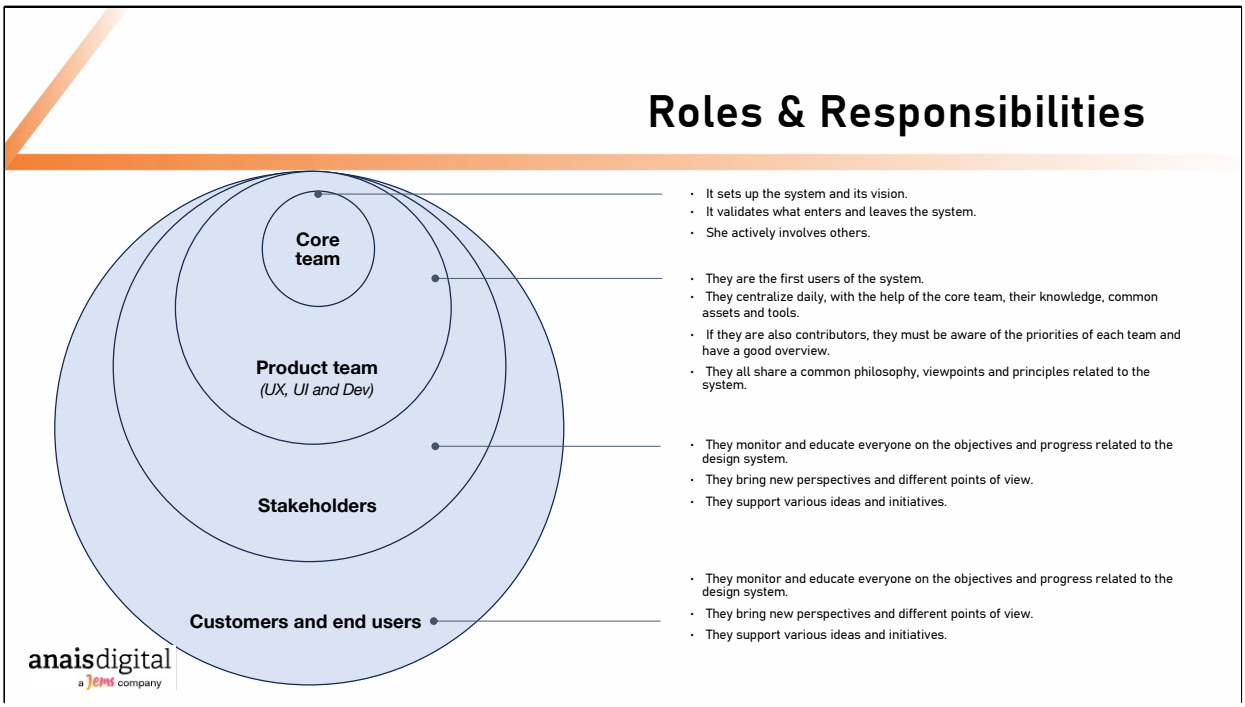
📌 **Objective:** Explain the core principles behind a design system.

🔷 **Key Takeaways:**

- A design system is **never truly finished**; it evolves with the product.
- It requires **dedicated ownership**, including UX, UI, developers, and product managers.
- Collaboration and governance are essential for **long-term adoption and consistency**.

🔗 **Recommended Reading:**

- Atomic Design by Brad Frost https://bradfrost.com/blog/post/atomic-web-design/

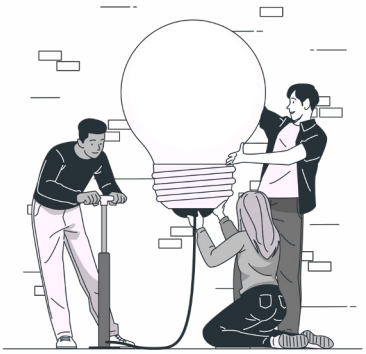


📌 **Objective:** Identify key stakeholders in a design system.
🔷 **Key Roles:**
👤 **Core Team:** Defines the system, validates new components, maintains consistency.
👥 **Product Team:** First adopters of the system, contribute to improvements.
👷 **Developers & Designers:** Implement, iterate, and refine components.
📢 **Stakeholders (PMs, Accessibility Experts):** Ensure system adoption across teams.
🔷 **Why is this important?**

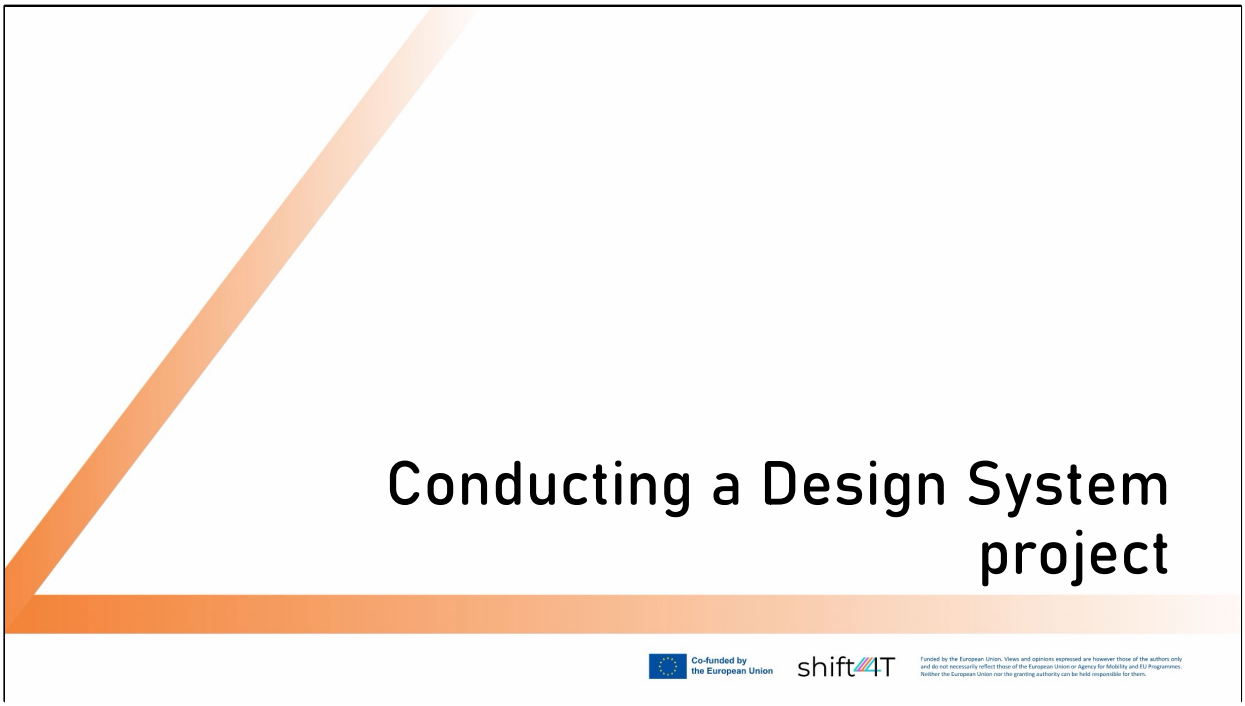
- Having **clear ownership and processes** ensures **scalability and consistency**.

Who should be part of your core team?

Roles?	Why?
Product Owner	Keeping a product vision
Developers	Validate technical feasibility
UX Researcher	Maintain consistency with the results of user studies
UX Designer	Producing what works through multiple feedbacks
Accessibility expert	Ensuring compliance with accessibility criteria
Communication Team	Guarantor of the corporate branding



anaisdigital
a Jems company



📌 **Objective:** Outline the process of creating a design system.
🔷 **Step-by-Step Approach:**
1️⃣ **Inventory of existing components & branding assets**
2️⃣ **Definition of experience principles & alignment across teams**
3️⃣ **Prioritization of components based on value & impact**
4️⃣ **Implementation & documentation**
5️⃣ **Measuring success with KPIs**
🔗 **Helpful Resources:**

- Interface Inventory Guide by Brad Frost
  https://bradfrost.com/blog/post/interface-inventory/

1. Alignment

Inventory

- To have an overview of the patterns (components, recurring functionalities in your system) based on existing product and chosen baseline framework.
- To collect all branding assets (recurring branding elements, icons, colors…) most used in the world of products and then prioritize them.
- Needed to organize workshops with the different teams

1. Alignment

Experience principles

- To have an overview of the patterns (components, recurring functionalities in your system) based on existing product and chosen baseline framework.
- To collect all branding assets (recurring branding elements, icons, colors…) most used in the world of products and then prioritize them.
- Needed to organize workshops with the different teams

1. Alignment

Roles & process

- Identify and understand each person's roles precisely and their associated tasks.
- When we talk about design systems, we often distinguish between 4 types of roles: "sponsors", "guarantors", “contributors” and "consumers".
- The roles will be distributed following the chosen implementation process.
- Result : we know how you want to work & manage you design system

1. Alignment

Prioritization of components

- Prioritize components by value (most frequent and easy to develop) in order to get the product backlog and a prioritization of the elements to be documented in the design system.
- We will either re-use & transform some existing components or build new ones essential to compose your design system MVP (Minimum Viable Product).

1. Alignment

How do you measure your success ?

- Define the main objectives of the design system, whether internal or external, and then associate them with precise measurement indicators.
- Define the key success factors of your design system is paramount. It is thanks to this that you will then be able to measure the value and success of the system with the right indicators.

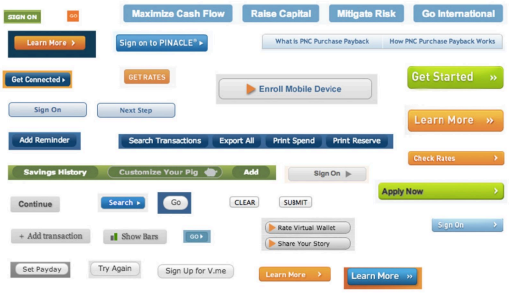
2. Inventory and file structure

Inventory

Make an exhaustive list of interface elements

Deconstruct current/future interfaces into smaller pieces

Check the consistency of interface elements / styles



Source: https://bradfrost.com/blog/post/interface-inventory/

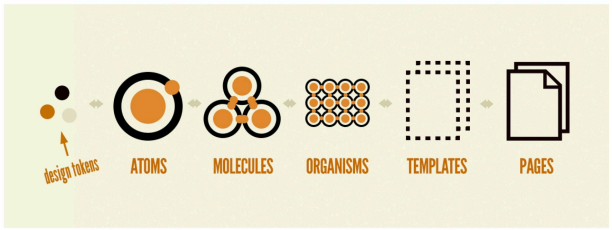
2. Inventory and file structure

Prepare a blank file (in any tool to paste screenshots into)
By screen/feature, take screenshots of interface elements
Categorize screenshots (eg: buttons, form elements, headers, ...)

https://bradfrost.com/blog/post/interface-inventory/

2. Inventory and file structure

Breaking down interfaces



Source: https://bradfrost.com/blog/post/atomic-web-design/

EXERCISE

- Make the inventory of a website homepage

Students are free to chose a homepage

Example: zalando

Organizations

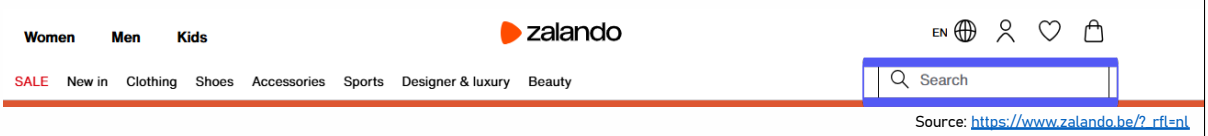
Modules /
Features

Source: https://www.zalando.be/?_rfl=nl

Example: zalando

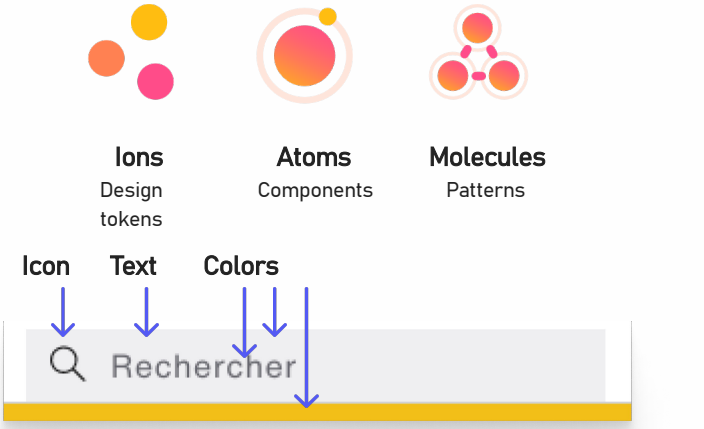
Organizations

Modules /
Features



Source: https://www.zalando.be/?_rfl=nl

Example: zalando



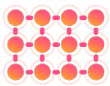
Source: https://www.zalando.be/?_rfl=nl

Hierarchy

Ions
Design tokens

Atoms
Components

Molecules
Patterns



Organizations
Modules
Features

Templates

Pages / Products

anaisdigital
a JEMS company

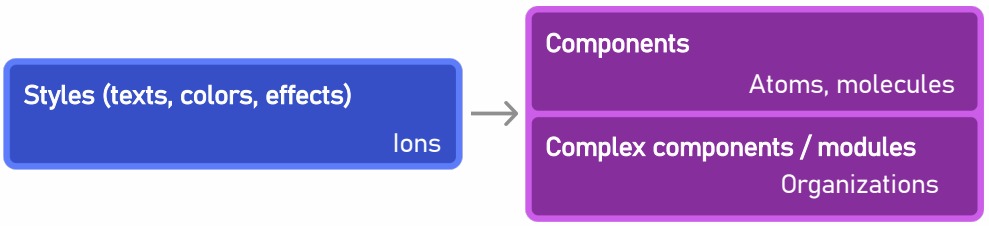
File structure of your DS

Styles (texts, colors, effects)
Ions

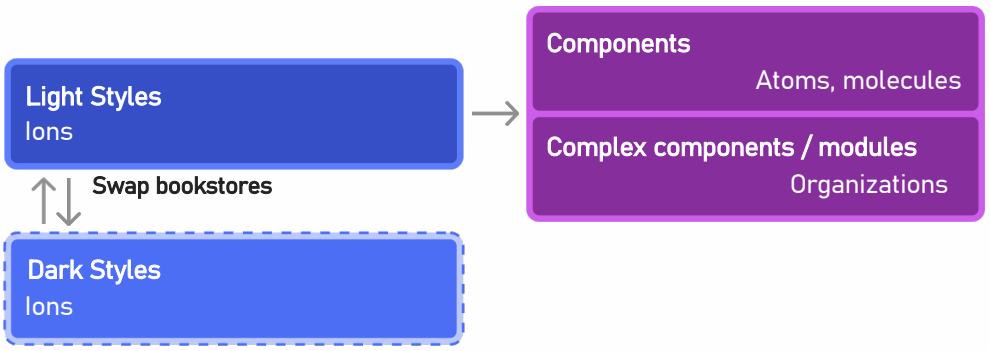
Components
Atoms, molecules

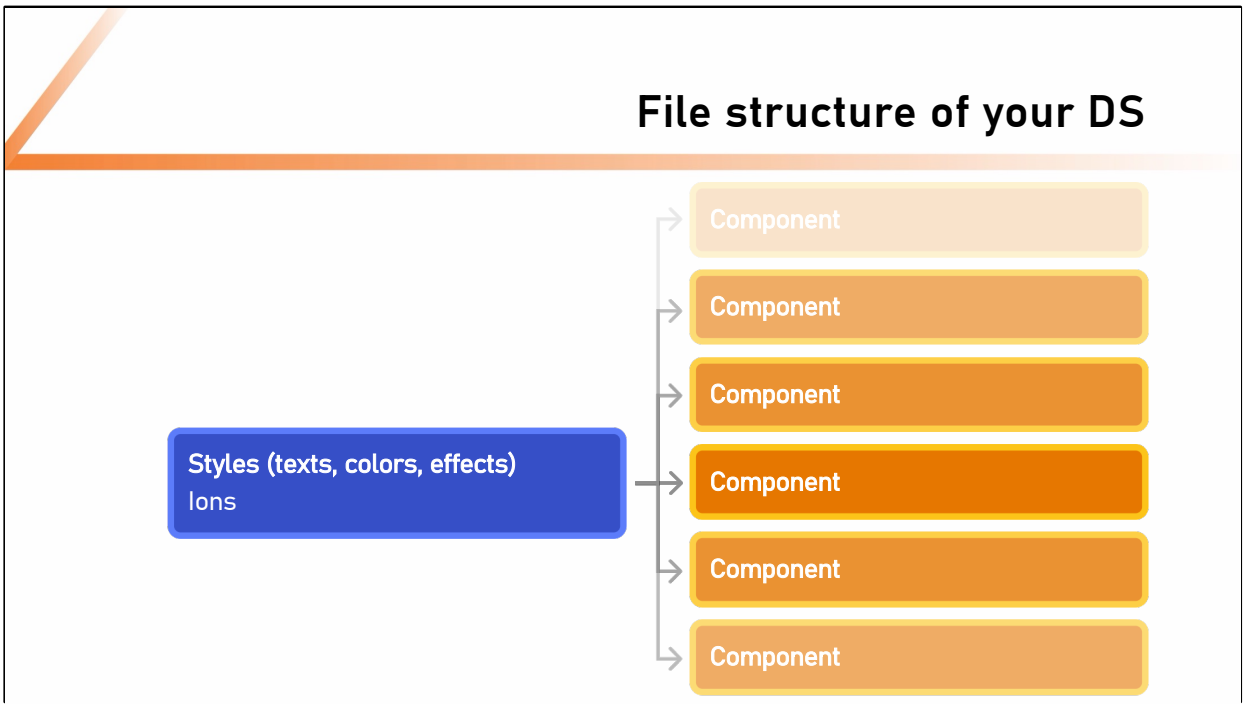
Complex components / modules
Organizations

File structure of your DS



File structure of your DS




File structure of your DS
Styles (texts, colors, effects)
Icons
Component
Component
Component
Component
Component
Component

File structure of your DS

- 1 file only: Ease of navigation in Figma
  - Only one file to watch
  - all in one place
  - **Small to medium project**
- 1 style file + 1 component file
  - Possibility of "swapping libraries" of styles (branding / light & dark mode etc.)
  - Clarity of separate files
  - **Small to medium project**
- 1 style file + 1 file per component
  - Less risk when pushing for changes
  - Focus on one thing at a time
  - Easier for versioning
  - Best performance
  - **Large project**

📌 **Objective:** Best practices for organizing a design system in Figma.

🔷 **Common Structures:**

- **Single File:** For small projects, everything in one place.
- **Style + Component Files:** Separate files for scalability.
- **Library System:** Swap styles (e.g., light/dark mode, branding variations).

🔷 **Why is this important?**

- Helps **reduce complexity** and **increase maintainability**.

Nomenclature

"*There are only two hard things in Computer Science: cache invalidation and naming things*."

Phil Karlton

📌 **Objective:** Define best practices for naming components and maintaining documentation.

🔷 **Key Guidelines:**

✅ **Use semantic names** instead of generic names (e.g., Primary Button instead of Button 1).

✅ Group elements by **functionality** rather than style.

✅ Maintain **comprehensive documentation** for developers and designers.

🔗 **Naming & Documentation Resources:** https://designsystemchecklist.com/

Examples / Links

- Decathlon https://www.decathlon.be/nl/splashpage/
- Gov.uk https://www.gov.uk/
- Material Design https://m3.material.io/
- There is a Figma plugin to generate a palette based on their principles
- Proxyclick
- Luis on color tokens https://youtu.be/uiCGvhI7Vwo?si=yrelRecf0K1BrTh-
- THE Druids colors which (subjectively) offer a good balance

Icons

- Not necessarily relevant to put all the icons in variants
- Some have more meaning to be than others, to change them more easily (example: chevron up / down)
- To facilitate component swap, ensure that component layers are identical (except for one exception: 1 vector layer named the same for each icon)
- Ability to provide icon size variations for stricter usage
- Possible to manage it with a “base component” but requires more maintenance anyway.

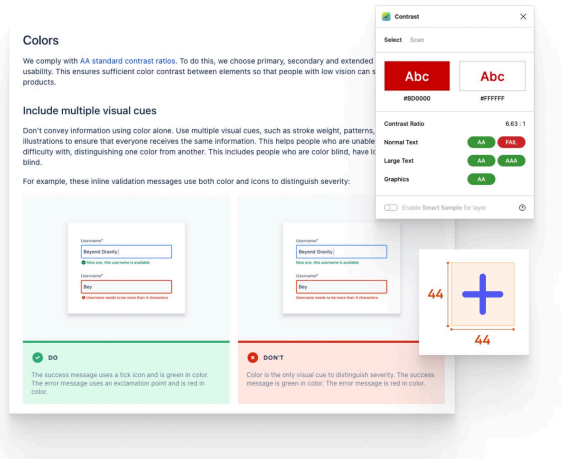
Components

- Too many variants/properties can harm the usability of the DS
- They can be more difficult to use/discover
- Example: all inputs in variants would not allow to quickly search and use an input date
- Separate by use Example: if a badge with icon is always used in case A and the one without icon in case B, these are 2 different components
- Fewer properties = more consistency
- “Base components” can be useful for managing large numbers of variants/components
- Use “unpublished components” to facilitate maintenance without cluttering user choices
- Component discoverability: Luis - “Is this a component?”
- Example of tabs that may be too complex ?

Documentation: accessibility

Like all other concepts, the Design System also allows us to centralize (in part) accessibility.

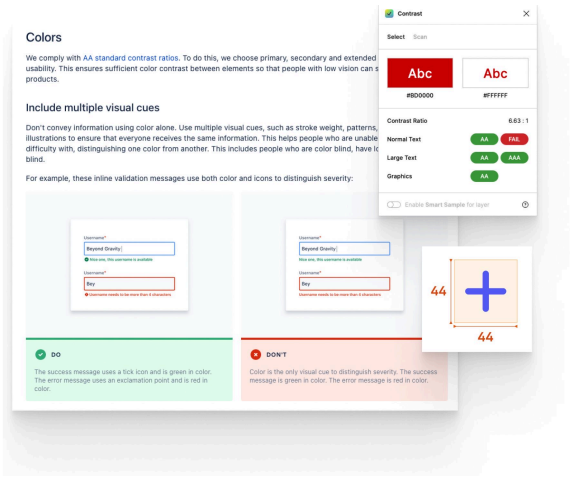
- Color contrasts
- Target Size
- Text size
- Don't rely on colours alone
- Specific behaviors
- Tab-index
- Increased contrast mode
- ...



Documentation: accessibility

Plugins

- Contrast
  - Check Contrasts - Be careful, not as reliable as using “true contrast checkers”.
- Adee
  - Accessibility toolbox (contrasts, target size, color blindness simulation, …)



Documentation Tool

Source: https://zeroheight.com/login

Your own platform?

Source: https://storybook.js.org/

Tips and tricks

Co-funded by the European Union

shift4T

Funded by the European Union. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or Agency for Mobility and EU Programmes. Neither the European Union nor the granting authority can be held responsible for them.

Tips and tricks: styles

- Don't create too many unnecessary styles
- If styles exist but are not Never used, they only clutter the DS
- Choosing a consistent nomenclature
- To make them easy to find and use
- Use descriptions to make usage easier
- Group them in a coherent manner

Tips and tricks: nomenclature

- It all depends on the context, who is using the DS and how.

- If it is not necessary, do not complicate
- Getting closer to the code
- Use human-readable names
- Use semantic (rather than generic) names
- Grouping together like with like
- Use descriptions to complete what cannot be said in the name
- (see point on documentation)

Tips and tricks : changes

Make a change

- Tip: place an instance of the component with different overrides next to the main in order to see live if things break.

Push a change(s)

- By selecting the relevant component
- Via asset / library
- Comment on your “push”

Accept a change(s)

- Accept all/some components
- Verify that the changes are safe

Tips and tricks : annotation

- Use descriptions (makes it easier to search and understand)
- Annotate when necessary

Tips and tricks : branch system

- As in development (and other disciplines), it is possible to create branches.
- Branches allow you to test changes (purely aesthetic or functional) on your side without involving large-scale modification.
- https://www.figma.com/best-practices/branching-in-figma/

Tips and tricks : move styles

- Sometimes it is necessary to create a style outside of the design system (it is sometimes easier to create a style in its context). It is then necessary to move it into the design system to continue to centralize them.

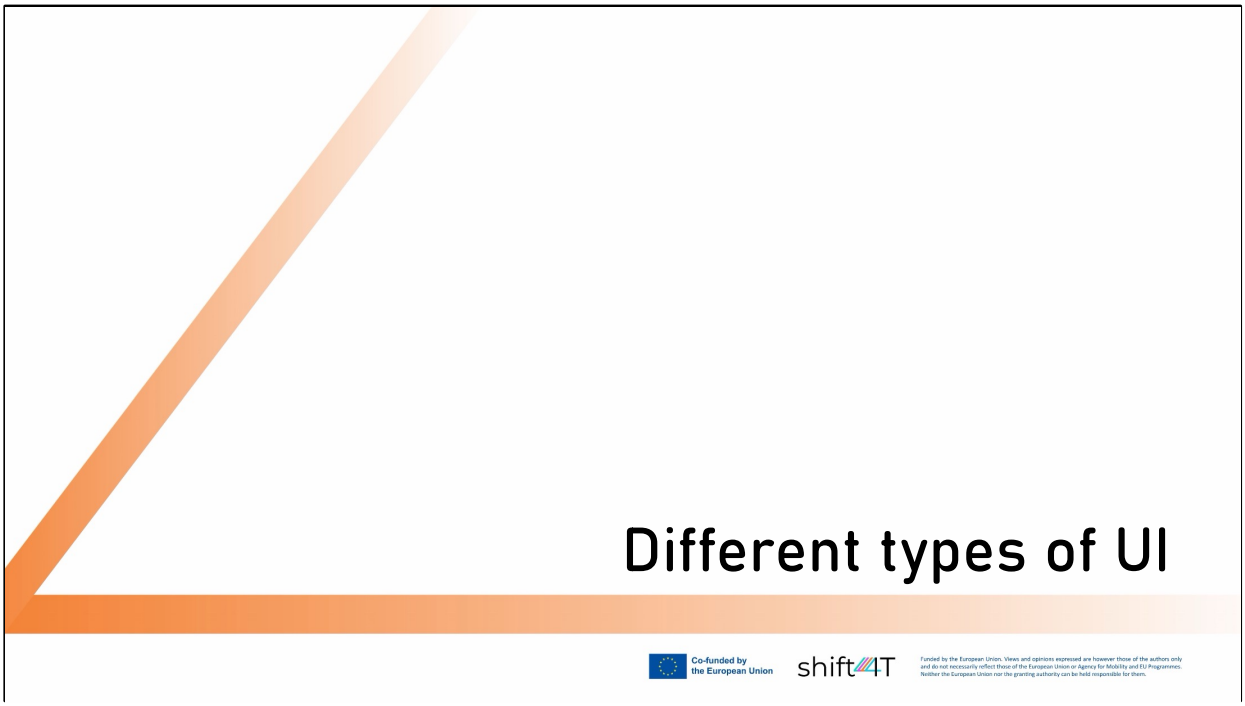
- https://help.figma.com/hc/en-us/articles/360039820134-Manage-and-share-styles#Move_styles

Tips and tricks : moving components

- Even more often than styles, it is sometimes easier to create a component in its context.
- It may be interesting to keep a component local, if it is specific to a feature. In other cases, it must be brought back to the design system.
- https://help.figma.com/hc/en-us/articles/4404848314647-Move-published-components

Tips and tricks : slot components

- In some cases, especially when following the Atomic Design method, the need arises to have flexible components that can accommodate several types of components.



📌 **Objective:** Explain different types of prototyping in UX/UI.
🔷 **Types of Prototypes:**
1️⃣ **Low-fidelity wireframes** – Quick sketches for early concept validation.
2️⃣ **High-fidelity wireframes** – More detailed, focusing on layout and structure.
3️⃣ **Interactive prototypes** – Simulated user interactions and flows.
🔷 **Why is this important?**

- Helps **validate ideas** before development, reducing costly rework.

•🔗 **More on Prototyping:**
•Figma Prototyping Guide

DISCUSSION

- What types of UI formats do you know?
- Which ones to use in which circumstances?

Zoning

- Spatial schematization of the interface
- Focus: order of elements, occupation of format, structure
- Reco. tools: paper, board, post-it
- Interaction: static

Zoning

Benefits

- Economical ++
- Fast ++ (preparation / creation / modification)
- Designer more inclined to change ++
- Character in progress acquired for stakeholders
- Ideal for co-design

Disadvantages

- Not sufficient in itself
- Discussions still rely heavily on imagination
- Feedback not always integrable / realistic
- Be comfortable with a pencil

Low-fi wireframe

- Black & white model taking into account the functional scope.
- Focus: role of zones, proportions, hierarchy visual, quantity of content, completeness
- Reco. tools: paper, wireframing software
- Interaction: static or sequential

Low-fi wireframe

Benefits

- Economical +
- Fast + (edit)
- Less pressure on testers
- Designer more inclined to change +

Disadvantages

- Feedback not always integrable / realistic
- Stakeholders may think this is a final deliverable
- Important Testing Presentation ++

Low-fi wireframe

- Detailed model very close to the rendering final visual.
- Focus: understanding of the content, brand fit, hierarchy visual
- Reco. tools: wireframing software
- Interaction: static or sequential or animated

High-fi wireframe

Benefits

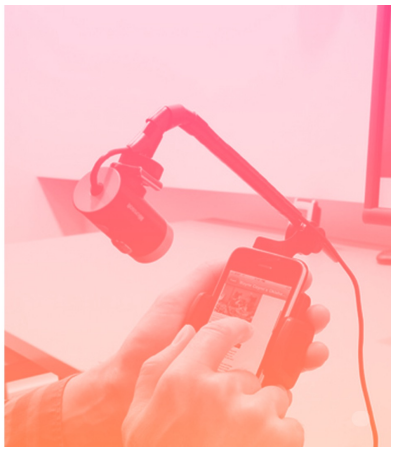
- Reliable results +
- Flow Integration +
- More realistic tester behavior
- Refining phase: no worries about the realization for the designer

Disadvantages

- Designer not inclined to modification
- Preparation of testing material +
- Approximate simulation of interactions
- Important Testing Presentation +

Interactive prototyping

- Model more or less close to the rendering final visual and functional.
- Focus: understanding flow, interactions, affordance, validation of scenarios
- Reco. tools: prototyping software, HTML
- Interaction: sequential and/or animated



Interactive prototyping

Benefits

- Reliable results +
- Light testing contextualization (if High-Fidelity) +
- Refining phase: no worries about the realization for the designer

Disadvantages

- Designer not inclined to modification
- Need resources (Dev) to simulate interactions if we want to go further
- Risk of bug
- Presentation/validation of important testing +

What ? When ?

Take into account

- The subject to be tested
- The work context
- His personal skills
- His personal preferences

Take into acccount

- Lo-Fidelity Interactive : Crucial and inexpensive step.
- Interactive High-Fidelity : Essential on large-scale projects before any development

Note: It is not mandatory to combine all types of prototypes on the same project.

Practice

- Go to figma file and practice in Figma

pages Figma, base, option, type of interactions => demonstration of the basics in Figma then
page Exercise 1 : Tell us a story using shapes only (please) => manipulation of basic shapes in Figma to create a story to present
pages overlays => demonstration then
page Exercise 2 : create something (a screen) with overlays
pages interactive components, variables => demonstration then
page Exercise 3 : we just ask you to create links for a prototype

RESSOURCES

Wise
Branding with a focus on accessibility and articles from the design team

Liferay
Also contains articles from the design team

GEL – Westpac group
A DS that encompasses several brands

Luna
Another DS that encompasses several brands. Possibility to change theme (brand) live

OpenBridge
DS specific to maritime interfaces

Wanda
Wonderflow.ai DS

RESSOURCES

Similar layer
Select layers based on common criteria

Styler
Create, detach, apply styles in groups

Batch Styler
Modify styles in one go (change color, font, etc.)

Typescales
Generate text scales based on a few inputs

Tailwind color generator
Generate color palettes based on a base color

Design lint
Define and verify rules that designs must respect

RESSOURCES

Self documentation

Generate documentation components automatically

EightShapes Specs

Create specs automatically based on instances

Propstar

Generate and show all possible variants of a component

RESSOURCES

Design Systems
Articles, best practices and list of open design systems

The component gallery
List of components and how some DS handle them

Luis (Figma)
Tweets often about DS, best practices, organization, ...

Molly Hellmuth
“Teaching what I know about Figma & Design Systems“

Design System Checklist
Checklist of tasks for creating a design system

DS List
List created by Vitaly Friedman of design systems by country

RESSOURCES

Design System ROI Calculator

A tool to calculate the returns on investment that a DS can bring

Building a Cross-Platform DS

How Booking built its multi-platform DS (Design language, tokens, “design API” ...)

ADDITIONAL READING

- https://www.designsystems.com/
- Atomic Design by Brad Frost
  https://bradfrost.com/blog/post/atomic-web-design/
- https://www.figma.com/best-practices/branching-in-figma/
- https://help.figma.com/hc/en-us/articles/360039820134-Manage-and-share-styles#Move_styles