

Comment optimiser le temps de production grâce au Design System ?

Note d'intention

Dans un contexte où les **produits numériques évoluent rapidement** et où les **contraintes de production** se renforcent, les équipes recherchent des solutions pour **gagner du temps**, **limiter les erreurs** et **maintenir une cohérence** sur l'ensemble de leurs interfaces. L'**UI Kit** et le **Design System** font partie des **outils** qui répondent à ces **enjeux**.

Mais au-delà de la recherche de **productivité**, ces outils sont de véritables **leviers de collaboration**. Bien utilisés, ils permettent de créer un **langage commun** entre designers, développeurs, responsables produit et autres parties prenantes, en **facilitant la communication** et en **réduisant les incompréhensions**.

Ce livre blanc propose d'explorer ces sujets en combinant **théorie, exemples concrets issus de projets réels** et **bonnes pratiques directement applicables**. L'objectif est d'aider les équipes à mieux comprendre **les différences entre un UI Kit et un Design System**, leurs complémentarités ainsi que les conditions nécessaires pour en tirer un réel bénéfice.

Le contenu s'adresse principalement aux professionnels impliqués dans la **création** et la **maintenance d'interfaces numériques** : designers, développeurs, product managers... mais il est également conçu pour celles et ceux qui souhaitent **découvrir ces notions** et **s'initier à leurs enjeux**.

L'intention est d'offrir une **base claire** pour **nourrir la réflexion**, **guider la mise en place** ou l'**évolution d'un Design System**, et **accompagner les équipes** dans une démarche plus **structurée, efficace** et **collaborative**.

Introduction

Hello chers lecteurs, chères lectrices,

Je m'appelle **Sylvie GAO**, et je suis ravie de vous accueillir dans ce livre blanc, réalisé dans le cadre de ma deuxième année de **mastère Digital Design & Creative Technologies** à l'**École Supérieure du Digital**. En parallèle, je suis en alternance chez **Creatiivity**, où j'occupe le poste d'**UX/UI designer** depuis maintenant plus de deux ans.

Durant cette période, j'ai eu l'opportunité de travailler sur différents projets tels que des **sites web**, des **applications mobiles** et des **plateformes web**. Chaque mission m'a permis de **progresser**, d'**affiner ma méthode de travail**, et de réaliser à quel point **poser de bonnes bases dans la création des maquettes** est aussi important pour **produire plus rapidement** ou pour **mieux collaborer avec les développeurs**.

Ce que j'aime dans ce métier, c'est que **chaque projet a ses besoins, ses contraintes, ses spécificités**. Rien n'est jamais figé, et c'est cette diversité qui rend le travail aussi **stimulant**, sans parler du **challenge créatif** propre à chaque univers.

Pourquoi ce sujet ?

Dès mon arrivée chez **Creatiivity**, on m'a formée à créer et à mettre en place des **UI Kits** sur chaque projet sur lequel je travaille. J'ai rapidement compris leur utilité : **gagner du temps, garder une cohérence graphique, faciliter les échanges entre designers et développeurs**. Et dans les projets sur lesquels j'ai travaillé, cet outil m'a été très utile.

Mais même quand un UI Kit est bien construit, avec ses composants et leurs différents états (default, hover, disabled...), il arrive un moment où **ça ne suffit plus**. Quand un projet **prend de l'ampleur** ou s'étend sur **plusieurs plateformes**, on peut vite **se perdre dans la construction**. Et pour un designer qui arrive **en cours de route**, sans repères clairs, c'est encore plus compliqué de s'y retrouver. C'est là qu'il devient essentiel de **poser des règles**, de **documenter les usages**, et de **s'appuyer sur une logique commune**, partagée par toute l'équipe.

C'est en prenant du recul, à travers la réalisation de ce livre blanc, que j'ai mieux compris ce qu'un **Design System** pouvait vraiment apporter. Pas seulement une **bibliothèque de composants**, mais aussi une **documentation claire**, des **règles d'usage bien définies**, et une **méthode de travail partagée** entre **designers** et **développeurs**.

Un mot d'honnêteté

Durant mon alternance, j'ai surtout travaillé sur la conception d'**UI Kits**. Mais j'ai aussi eu la chance de participer à la création d'un **Design System**, encadrée par mon manager, sur un projet de plus grande envergure. C'est cette **expérience** qui m'a permis de voir ce qui change vraiment quand on passe d'un UI Kit à un design system.

Ce **livre blanc** est pour moi l'occasion de prendre du recul, à la fois sur mes expériences concrètes et sur les grands principes qu'on retrouve dans la pratique du design aujourd'hui.

Problématique

Comment optimiser le temps de production grâce au Design System ?

Objectifs du livre blanc

À travers ce livre blanc, je souhaite :

- **Clarifier les différences** entre un UI Kit et un Design System,
- Montrer en quoi un Design System bien pensé peut devenir un **véritable levier de productivité**,
- Partager mes **retours d'expérience professionnels**,
- **Proposer des bonnes pratiques concrètes** pour les équipes qui veulent structurer leurs projets de manière plus efficace.



SOMMAIRE

01. UI KIT ET DESIGN SYSTEM : DEUX OUTILS, DEUX LOGIQUES 07

- 1.1 Qu'est-ce qu'un UI Kit ?
- 1.2 Quand un UI Kit ne suffit plus
- 1.3 Le Design System : plus qu'une bibliothèque, une méthode
- 1.4 Métaphore LEGO
- 1.5 Une logique qui dépasse le produit

02. LE DESIGN SYSTEM FACE À LA RÉALITÉ : UN GAIN DE TEMPS OU UN FREIN ? 16

- 2.1 Les promesses : gain de temps, cohérence, alignement
- 2.2 Les pièges : rigidité, dette, surcharge initiale
- 2.3 Un bon Design System, c'est quoi ?

03. COLLABORATION DESIGN/DEV : CE QUE CHANGE UN DESIGN SYSTEM 24

- 3.1 Les difficultés qu'on rencontre sans Design System
- 3.2 Quand le Design System fluidifie les échanges... ou pas
- 3.3 Outils partagés : Tokens Studio, Figma, Storybook

04. MON EXPÉRIENCE CHEZ CREATIVITY : ENTRE THÉORIE ET PRATIQUE 34

- 4.1 Mise en place d'UI Kits et apprentissages
- 4.2 Contribution à un Design System complet
- 4.3 Leçons tirées, limites rencontrées

05. COMMENT DÉCIDER ENTRE UI KIT ET DESIGN SYSTEM ? 41

- 5.1 UI Kit ou Design System : comment faire le bon choix ?
- 5.2 Les bonnes questions à se poser avant de se lancer
- 5.3 Les trois piliers : gouvernance, documentation, maintenance

06. RECOMMANDATIONS & BONNES PRATIQUES 45

- 6.1 Ce que j'ai appris sur le terrain
- 6.2 Conseils pour gagner du temps et éviter les erreurs
- 6.3 Conseils pour les designers et pour les devs

07. CONCLUSION & REMERCIEMENTS 48

08. RÉFÉRENCES & GLOSSAIRE 51



01.

**UI Kit et Design System :
deux outils, deux logiques**

1.1 Qu'est-ce qu'un UI Kit ?

Un **UI Kit** est une **boîte à outils graphique** contenant tous les éléments essentiels à la construction d'une interface : **boutons**, **champs de saisie**, **icônes**, **couleurs**, **typographies**, **composants interactifs**... C'est un outil très utile pour **gagner du temps en début de projet** et garantir une certaine **cohérence visuelle**.

Sur mes différents projets chez **Creatiivity**, je mettais systématiquement en place un UI Kit. Cela me permettait de **construire rapidement des maquettes avec des bases solides**, en utilisant des éléments communs, et de ne pas repartir de zéro à chaque nouvel écran.







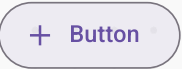
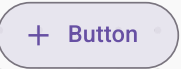
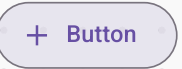
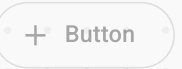
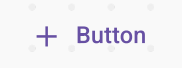
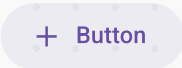
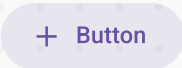
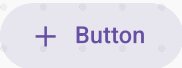

Exemple :

Dans un projet avec de nombreuses pages, je devais ajouter un nouveau bouton sur une vingtaine d'écrans. Ce bouton n'existait pas encore comme composant. J'ai dû faire la modification manuellement sur chaque écran, ce qui m'a pris du temps et a provoqué quelques erreurs (padding oublié, mauvaise taille de texte...). Rien de dramatique, mais évitable.

À l'inverse, sur un autre projet, un composant "card" était déjà bien structuré. Quand le client a demandé d'ajouter un bouton favori dans la "card", j'ai simplement modifié le composant principal et toutes les instances se sont mises automatiquement dans les maquettes. Gain de temps considérable, et cohérence garantie.

Astuce :

Dans un UI Kit, il est **essentiel de prévoir les différents états des composants**. Cela permet aux développeurs de savoir exactement comment chaque élément doit se comporter, et évite les oublis au moment de l'intégration. C'est un **réflexe très utile** qui évite beaucoup d'allers-retours avec les développeurs. **Tout est clair dès le départ**, même sans documentation.

	Enabled	Hovered	Focused	Pressed	Disabled
Filled					
Outline					
Text					

1.2 Quand un UI Kit ne suffit plus

Mais malgré tous ces avantages, l'UI Kit atteint vite ses limites, surtout quand :

- **Plusieurs designers** travaillent sur le même projet.
- Le **produit évolue rapidement**.
- Quand le **nombre de composants explose**.

Exemple :

Dans mon entreprise, on a toujours eu l'habitude de structurer les maquettes dès le départ, en créant des composants Figma. Cette rigueur nous permet déjà de gagner un temps précieux sur la production.

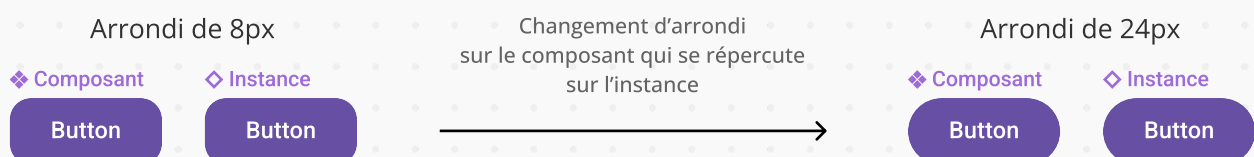
Mais j'ai vu que ce n'était pas toujours suffisant, notamment sur un projet auquel j'ai directement participé : une solution de gestion dédiée à la restauration, déclinée en trois plateformes (commandes, caisse, gestion).

Face à la complexité croissante du produit et au nombre important de composants, l'équipe en charge du projet dont mon manager a décidé de le moderniser et de le restructurer dans sa construction. C'était aussi l'occasion idéale pour mettre en place un Design System complet, dans l'objectif de clarifier les règles d'usage des composants et d'unifier les pratiques.

Ce Design System nous a permis de reposer des bases solides (typos, couleurs, espacements, tokens...) et d'assurer une vraie cohérence graphique durable entre les trois plateformes, tout en facilitant la maintenance pour les évolutions futures.

Définition :

- Un **composant** est un élément UI interactif et réutilisable, conçu une seule fois pour garantir la **cohérence entre les écrans** et éviter les **écarts visuels** (tailles, marges, couleurs...).
- Une **instance** est une copie de ce composant, utilisée dans les maquettes. Si le composant principal est modifié, **toutes les instances se mettent à jour automatiquement**.



1.3 Le Design System : plus qu'une bibliothèque, une méthode

Un **Design System** va bien au-delà d'un simple UI Kit ou d'un fichier de composants partagés. Ce n'est pas juste une **bibliothèque graphique**, c'est un cadre de travail structuré, conçu pour assurer la cohérence, la clarté et la maintenabilité du design dans un produit, que ce soit un site web ou une application.

Il ne s'agit pas seulement d'avoir des composants prêts à l'emploi, mais aussi de **savoir comment, quand et pourquoi les utiliser**. C'est cette **logique partagée** qui fait la vraie différence par rapport à un simple UI Kit.

“

Pour toi, quelle est la principale différence entre un UI Kit et un design system ?

Un UI KIT peut être plus facilement modifié qu'un design system qui lui possède des règles plus strictes en terme d'utilisation.

Valentin LE COZ - UX/UI Designer

”



Ce que ça montre :

Un **UI Kit**, c'est **pratique, souple, rapide**. Mais à force de liberté, **on perd parfois en cohérence**. Le **Design System**, lui, **impose un cadre plus strict** et c'est précisément ce cadre qui garantit la continuité quand le projet grandit.

Concrètement, un design system intègre :

- Une **bibliothèque de composants**, avec leurs **différents états et variantes** (normal, hover, désactivé, etc.),
- Des **règles d'usage précises** : dans quel contexte utiliser un bouton primaire ou secondaire, quel champ convient à quel cas d'usage, etc.,
- Un **système de nommage cohérent** appliqué à l'ensemble des composants, pour faciliter la lecture, la maintenance et le lien entre design et développement,
- Des **styles typographiques**, des **tokens de couleurs**, une **grille modulaire**,
- Une **documentation claire et partagée** pour l'équipe.

Dans le projet de gestion de restauration dont j'ai parlé plus tôt, la création du design system ne s'est pas arrêtée à la structuration graphique des composants.

Nous avons aussi mis en place une **documentation détaillée pour chaque élément**, directement dans **Figma**.

Exemple :

Par exemple, pour un composant bouton, la documentation inclut :

- une explication complète de toutes les possibilités : bouton avec icône devant ou derrière le texte, plusieurs couleurs (primaire, secondaire, etc.), différents styles (solid, faded, ghost), plusieurs tailles (small, medium, large...),
- une vue de tous les états associés à chaque style et chaque couleur : normal, hover, focus, désactivé, etc.,
- et des exemples concrets et visuels, pour que chacun comprenne clairement comment et quand utiliser chaque variation.

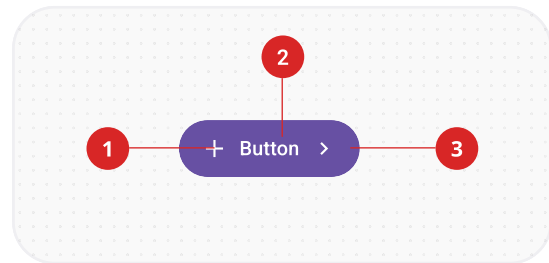
Cette documentation a été pensée pour anticiper tous les cas d'usage et éviter les approximations. On a vraiment préparé le terrain pour que les développeurs puissent intégrer les composants tels qu'ils sont conçus, sans malentendu. Tout a été prévu, illustré, nommé de façon logique, et expliqué dans le détail.

Résultat : les équipes gagnent du temps, travaillent avec plus d'autonomie, et le produit gagne en cohérence et en évolutivité.

Extrait de documentation : bouton

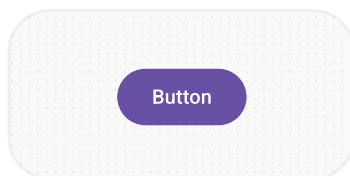
Anatomy

- 1 Leading Icon
- 2 Label Text
- 3 Trailing Icon

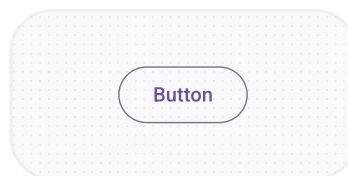


Variant

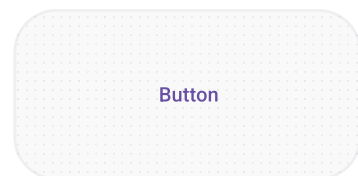
Filled



Outlined

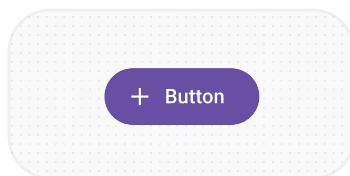


Text

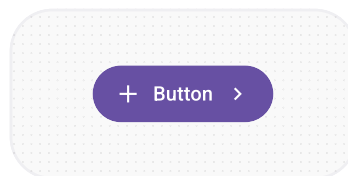


Icon

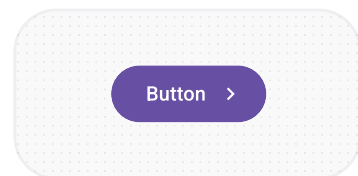
Leading



Both



Trailing



States

	Enabled	Hovered	Focused	Pressed	Disabled
Filled					
Outline					
Text					

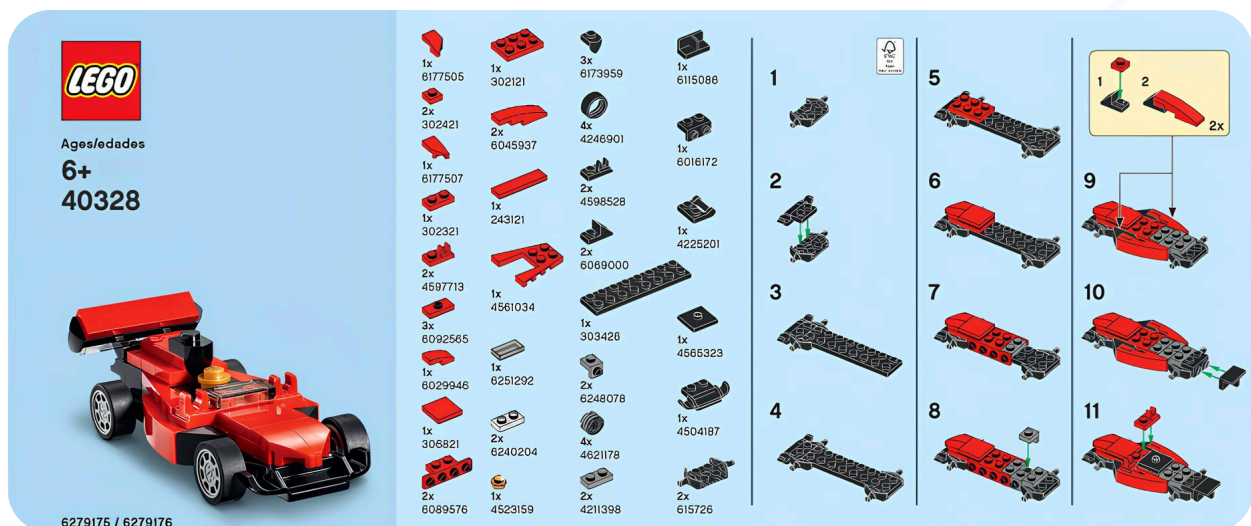
1.4 Métaphore LEGO

Pour expliquer simplement la différence entre un UI Kit et un design system, on peut prendre l'exemple d'une **boîte à LEGO**.

Un **UI Kit**, c'est comme si tu avais une **boîte remplie de pièces LEGO**, prêtes à l'emploi. Tu peux construire ce que tu veux, à ta manière, avec les règles que tu t'es fixées. Mais ces règles, **tu es souvent la seule à les connaître** : elles ne sont pas documentées, donc les autres peuvent **les interpréter différemment** ou **ne pas les voir du tout**.

Un **Design System**, c'est cette même boîte... mais avec **un manuel de construction clair**, des **règles écrites et visibles**, et une **organisation pensée pour être comprise par tous** : designers, développeurs, nouveaux arrivants. Il ne s'agit plus seulement de construire vite, mais de **construire de manière alignée, durable et cohérente**.

Dans ma pratique, l'UI Kit me permet de **travailler rapidement**, avec des composants bien pensés. Mais dès qu'on est plusieurs à intervenir, ou que le produit évolue, le **Design System devient essentiel**. Il **évite les malentendus**, **garantit une bonne utilisation des éléments**, et **simplifie les échanges** entre les équipes.



En résumé :

Un **UI Kit** permet de **construire librement**, en suivant ses propres repères.

Le **Design System** permet de **construire ensemble**, avec un **cadre commun**, et sans devoir réexpliquer chaque décision.

1.5 Une logique qui dépasse le produit

Le **Design System** est souvent associé à la conception d'interfaces sites web, des applications mobiles et des plateformes web. Pourtant, certains designers étendent cette logique à d'autres domaines du design, comme le **branding** ou la **communication visuelle**. Ils n'utilisent pas le design system uniquement comme un **outil de production**, mais comme une **méthode de pensée** : **structurer**, **systématiser**, puis **décliner**.

Certains designers l'adoptent dans leur pratique pour **gagner en cohérence**, **en efficacité**, mais aussi **en liberté créative**. D'autres y voient une **contrainte qui fige**. Cette section explore ces usages plus transverses, là où le Design System devient un **véritable outil méthodologique**, au-delà de la simple interface.

“

Est-ce que le design system a changé ta manière de travailler en branding ? Et qu'est-ce qui, selon toi, le différencie d'un design system pensé pour le design produit ?

Au lieu de me focaliser sur l'identité visuelle de manière globale, je me focalise plutôt sur sa déconstruction : comment retourner à l'essentiel pour ensuite scale et build des éléments visuels de plus en plus complexes. [...]

Il n'y a pas vraiment de différences techniques fondamentales à mon sens, ce sont surtout des médiums différents avec leurs propres règles et normes. En brand, on construit pour du marketing. Côté produit, le design system inclut aussi toute la partie dev, la documentation, la gestion des composants, etc.

Denis CONSTANT - Brand Designer

”



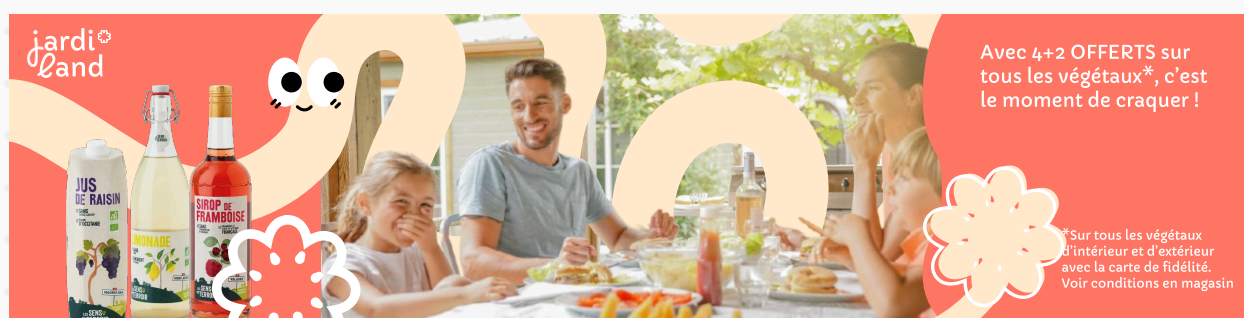
Ce que ça montre :

Même en dehors des produits digitaux, la logique du design system reste pertinente. Elle permet de **penser une identité visuelle** comme un **ensemble cohérent, organisé et déclinable**. Le **branding**, lui aussi, peut s'appuyer sur une structure qui aide à maintenir la cohérence, tout en répondant à des **objectifs différents** de ceux du produit.

Exemple d'un Design System utilisé en branding

Dans un **contexte de branding**, on cherche à **automatiser la création de visuels**.

Par exemple, on peut mettre en place une automatisation en termes de **couleurs**, d'**images**, ou de **motifs floraux**.





02.

**Le Design System face à la
réalité : un gain de temps ou
un frein ?**

2.1 Les promesses : gain de temps, cohérence, alignement

Le **Design System** est souvent présenté comme une **solution miracle** pour gagner en efficacité. Mais dans la réalité, il faut bien distinguer les promesses à long terme des efforts à fournir au départ.

D'abord, soyons clairs : **au début, ce n'est pas un gain de temps**. Mettre en place un Design System **demande du temps**, de la rigueur et de l'anticipation. Il ne suffit pas de créer des composants : il faut **penser à toutes les variantes, documenter leur usage, choisir une logique de nommage, et organiser une documentation claire et lisible**. Il faut aussi s'assurer que tous les composants respectent une **cohérence globale** : les tailles de texte, les espacements, les arrondis, la hiérarchie des éléments visuels... Ce sont des détails, mais mis bout à bout, ils donnent au produit sa lisibilité et sa solidité.

Dans mon expérience, la mise en place d'un vrai Design System **peut facilement prendre plusieurs mois**. Mais une fois bien en place, les **bénéfices sont concrets et durables**.

Voici **les avantages** que j'ai pu constater :

- Un vrai **gain de temps sur les modifications** : quand un composant évolue (ajout d'un pictogramme, ajustement de padding, changement de couleur...), il suffit de modifier le composant principal. Toutes les instances dans les maquettes **se mettent à jour automatiquement**. Sur des projets avec des dizaines d'écrans, c'est un vrai soulagement, on évite de tout refaire à la main et on limite les oublis.
- Une **meilleure qualité d'intégration** : comme tous les cas d'usage sont pensés à l'avance et bien documentés, les composants sont intégrés plus fidèlement. Cela limite les erreurs d'interprétation ou les incohérences (mauvais état, mauvaise taille, oubli d'icône, etc.).
- Un **point de référence partagé** : une fois le système en place, il sert à tout le monde même aux personnes qui l'ont conçu. En cas de doute ou d'oubli (sur un état, une règle d'usage...), la documentation **évite les doutes et fait gagner du temps**.
- Une **base commune entre plusieurs fichiers** : sur le projet de gestion de restauration, le design system a été publié en librairie Figma. Il est utilisé dans trois fichiers différents un pour chaque plateforme (commandes, encaissement, gestion). Chaque fichier reste autonome, mais tous se basent sur la même bibliothèque. Cela **garantit une cohérence visuelle** tout en facilitant les évolutions **à long terme**.

Exemple :

Sur certains de mes projets, lorsque je ne documente pas suffisamment mon UI Kit, je m'en rends vite compte. Quand je reviens sur un fichier après plusieurs semaines, je peux être un peu perdue : pourquoi ce composant a-t-il cette taille ? Est-ce qu'il y a une raison à ce style précis ? Est-ce que c'est une variation officielle ou juste un test ?

C'est dans ces moments-là qu'on comprend à quel point la documentation est un gain de clarté, même pour soi-même.

“

Constates-tu un vrai gain de temps ou une meilleure cohérence visuelle grâce au Design System ?

On pense toujours à la consistance et à la cohérence visuelle par ce principe. Lors de la création de nouveaux assets, on ne repart jamais de zéro, donc gain de temps, et assurance de la consistance visuelle.

Denis CONSTANT - Brand Designer

As-tu déjà travaillé sur un projet sans Design System ? Si oui, qu'est-ce qui change selon toi ? Et si non, que penses-tu que ça impliquerait de ne pas en avoir ?

Oui j'ai déjà travaillé sans, et cela est compliqué pour la partie design où on peut se retrouver avec des incohérences, mais également lors du développement cela évite les erreurs et les différences entre les maquettes et la production.

Valentin LE COZ - UX/UI Designer

Est-ce que vous ressentez un vrai gain de temps dans votre production grâce au Design System ?

Oui, dès lors de mon arrivée, cela m'a fait gagner un temps considérable en prenant en main plus rapidement le Design System.

Valentin LE COZ - UX/UI Designer

Est-ce que le design system te fait gagner du temps lors de l'intégration ? À quel moment le ressens-tu ?

Absolument ! tu le ressens au bout du 2e composant créé. Tout devient plus rapide.

Sylviane TRAN - Développeuse

”



Ce que ça montre :

Le Design System ne se contente pas de structurer les composants : il **rend le projet plus lisible, plus cohérent** et **plus rapide à prendre en main**, pour les designers comme pour les développeurs. Son absence complique l'organisation, sa présence donne un cadre qui fait gagner du temps à tous les niveaux.

En résumé :

Un Design System bien conçu permet :

- De **gagner du temps sur le long terme**, surtout pour les projets évolutifs ou multi-équipes,
- De **réduire les erreurs**, en cadrant les usages et en rendant les intentions plus claires,
- Maintenir **une cohérence visuelle et fonctionnelle dans le temps**, même quand les équipes ou les projets changent.

Créer un Design System, ce n'est qu'un début. Pour qu'il **reste efficace**, il faut le **maintenir à jour, ajuster ce qui pose question**, et s'assurer qu'il est compris et utilisé par toute l'équipe.

2.2 Les pièges : rigidité, dette, surcharge

Même si les bénéfices du Design System sont réels, il **peut aussi devenir une contrainte**, surtout s'il est **mal dimensionné** ou appliqué **sans discernement**.

Trop lourd pour des projets courts :

Je l'ai réalisé en travaillant sur un projet où je devais adapter le Design System de Material Design à un contexte client spécifique. Au début, j'ai commencé à documenter tous les ajustements que je faisais : styles, couleurs, règles d'usage... mais très vite, je me suis rendu compte que **je perdais du temps à trop détailler**, alors que le projet n'allait pas évoluer à long terme. C'était un cas où un simple UI Kit bien structuré aurait largement suffi. Le projet était ponctuel, sans réutilisation prévue. **Trop documenter était contre-productif.**

Trop de règles = moins de liberté ?

Un autre effet moins visible, mais réel, c'est que quand les **règles sont posées**, on a tendance **à ne plus les remettre en question**. Ce n'est pas forcément bloquant, mais j'ai remarqué que sur certains projets, une fois que le Design System est installé, **on ne pense pas toujours à explorer** d'autres alternatives même quand le contexte pourrait le justifier. Ce n'est pas qu'on ne veut pas innover, c'est juste que **les règles deviennent la norme**, et qu'on **évite de sortir du cadre**.

“

Penses-tu qu'un design system bien pensé peut aussi stimuler la créativité ? Ou est-ce que tu le vois plutôt comme un outil de structuration ?

Oui et c'est même son but à mon sens : la créativité naît de la contrainte. La structure permet d'avoir une base saine, d'être serein sur les futurs visuels qu'on aimerait créer, et donc d'apporter un cadre pour nourrir notre créativité.

Denis CONSTANT - Brand Designer

Quels sont, selon toi, les inconvénients d'un Design System ? Et si tu devais choisir entre UI Kit et Design System, lequel préférerais-tu et pourquoi ?

Lorsque le design system ne répond plus aux attentes du designer, cela peut créer des difficultés le temps que celui soit mis à jour. Je choisirais Design system, pour un gros gain de temps mais le gros point négatif est le manque de créativité. Revenir ensuite à de la créa sans, devient plus difficile.

Valentin LE COZ - UX/UI Designer

”



Ce que ça montre :

Selon les profils et les pratiques, le Design System peut être vécu **comme un tremplin** ou **comme une contrainte**. Pour certain-es, la structure **apporte de la clarté et libère l'esprit** pour se concentrer sur des choix créatifs forts. Pour d'autres, elle peut **figer les habitudes et rendre plus difficile** le retour à une **création "libre"**. Le Design System ne bloque pas la créativité en soi : tout dépend de la façon dont il est conçu... et de la place qu'on lui donne dans le processus.

Et parfois... on n'a tout simplement pas assez de recul

Il arrive aussi qu'on se lance dans la mise en place d'un Design System sans avoir encore assez de recul sur le produit. On veut bien faire, structurer proprement, **anticiper tous les usages possibles**... mais on le fait **trop tôt**, parfois sous pression ou dans une logique de rapidité, et sans attendre que le projet prenne forme.

En résumé :

On passe du temps à documenter ou organiser des choses... qui ne seront peut-être jamais réutilisées. Ce n'est pas une erreur grave mais c'est un apprentissage précieux : **tout ne mérite pas forcément un niveau de structuration maximal dès le début.**

Ce que je retiens de ces expériences

Il faut adapter l'effort de structuration à la nature du projet. **Tous les projets ne justifient pas un Design System complet.** Il vaut mieux **commencer simple**, avec un **UI Kit solide**, et ajouter des briques de Design System si le produit a le potentiel d'évoluer. On peut aussi s'inspirer de bons Design Systems existants, pour construire plus rapidement, en gardant une logique cohérente. Et surtout, il faut accepter que le Design System il **doit pouvoir évoluer**, se **simplifier**, ou se **réadapter en fonction des besoins réels**.

2.3 Un bon Design System, c'est quoi ?

Dans ma pratique, je ne dirais pas qu'il existe un "Design System parfait". Mais il y a des signes clairs qui montrent qu'un Design System est bien pensé, bien structuré, et réellement utile.

Ce que je remarque dans les bons Design Systems

C'est souvent au moment de rédiger la documentation que je me rends compte que le système fonctionne bien. Quand **tous les composants sont alignés dans leur construction** (mêmes règles de spacing, mêmes tailles de police, même logique d'icône ou de bordure), et que **chaque règle s'applique de façon constante**, alors je sens que le système est stable et fiable. Et quand le Design System est réutilisé sur plusieurs projets comme dans mon entreprise, où la même librairie Figma sert pour trois plateformes différentes on voit si les composants gardent une vraie continuité : même rendu visuel, mêmes logiques d'interaction, mêmes comportements attendus côté développement.

Ce n'est pas juste une question "d'apparence"

C'est le fait que **chaque élément fonctionne comme prévu**, peu importe où on l'utilise, et qu'il **respecte une même grille de lecture**. C'est ce qui renforce la cohérence globale du produit. Je le vois aussi quand j'observe les échanges entre les développeurs et mon manager : il y a **peu de questions liées au design lui-même**. Les discussions portent plus souvent sur des **détails d'implémentation technique**. Ça montre que le Design System **remplit bien son rôle de référentiel** : il rend les règles visibles, compréhensibles, et évite les malentendus.

Et un mauvais design system, c'est quoi pour moi ?

C'est **quand les composants ne sont pas construits avec les mêmes logiques** (espacements différents, arrondis incohérents, styles contradictoires), quand les éléments sont **mal nommés** ou **mal rangés**, ou tout simplement quand on **se perd dans le fichier**, sans savoir par où commencer. Ce n'est pas forcément un problème d'exécution, mais plutôt un **manque de structure claire ou de règles définies dès le départ** notamment celles qui assurent la cohérence entre les composants (tailles, marges, états...).

Un conseil évident... mais qu'on oublie facilement

Souvent, on veut bien faire et aller vite et ça peut amener à poser des fondations trop rapidement, sans avoir encore assez de recul sur le projet. Dès le départ, mon manager m'a donné un conseil qui m'a beaucoup aidée : "Va voir comment font les autres. Observe plusieurs Design Systems, **inspire-toi des logiques existantes**, et **prends le temps de structurer proprement**."

Ça m'a permis de construire avec plus de recul, en prenant de bonnes pratiques dès le départ. Et à chaque nouvelle création de composant, je fais des revues régulières avec lui, pour valider la construction, la cohérence et le respect des règles fixées.

Pour moi, un bon design system, c'est...

Une **cohérence dans la construction** : les composants sont construits avec les mêmes logiques, les **règles fixées sont respectées** dans chaque nouvelle création. Une **documentation claire et complète**, qui couvre tous les cas d'usage (états, tailles, variations...), sans devenir un bloc trop lourd à lire. Un **fichier organisé**, où l'on peut se repérer rapidement, même sans connaître le projet par cœur. Et j'ajouterais que chaque phase de création (composants, nommage, documentation...) demande du recul. C'est essentiel d'**avoir une vision d'ensemble**, et pas juste une logique locale.

Et côté équipe : qui en est responsable ?

Dans l'idéal, c'est une responsabilité partagée. Mais concrètement, je pense que le **designer** est la personne la plus légitime pour piloter le Design System. C'est lui qui est **au cœur de la construction des composants**, qui **maîtrise la logique graphique**, et qui peut faire le lien entre les besoins visuels et techniques. Le développeur code ce qui est prévu, le chef de projet suit l'avancement mais le designer, s'il connaît bien le projet, peut garantir la cohérence de l'ensemble.

Ce que je dirais à un nouveau designer

Si quelqu'un arrivait sur un design system déjà existant, je lui dirais d'abord : **"Commence par observer les composants existants, lis bien la documentation, et cherche à comprendre la logique mise en place."**

Ensuite, il faut :

- **suivre les règles de nommage,**
- **respecter les styles de couleurs** et de **typographie,**
- **ne pas hésiter à poser des questions** : au designer qui a construit le système si quelque chose te semble flou, ou au développeur si tu as un doute sur l'intégration.

Un Design System est un appui et non une barrière

Un **Design System** donne un cadre. Ce n'est pas un système figé : il doit **évoluer avec le produit, s'adapter aux besoins**, et parfois être simplifié. Il **ne bloque pas la créativité** : il permet de mieux canaliser les décisions, et d'éviter de repartir de zéro à chaque nouveau projet.

Dans la suite, je vais justement parler de cette relation clé entre design et développement et de ce que change (ou pas) la présence d'un Design System dans la collaboration.



03.

**Collaboration design/dev :
ce que change un Design
System**

3.1 Les difficultés qu'on rencontre sans design system

Même si aujourd'hui, dans mon entreprise, la mise en place d'un UI Kit (ou d'un Design System) est une habitude dès le début de projet, j'ai eu des expériences où ce n'était pas le cas.

Exemple 1 : Un projet sans composants, tout en manuel

Dans l'un de mes précédents stages, où je réalisais des maquettes sans composants formalisés, et où j'intégrais moi-même le design sur WordPress.

Sur le moment, je ne réalisais pas l'impact de ce fonctionnement. Mais avec le recul, je vois clairement les limites :

- Chaque bouton ou bloc devait être modifié manuellement sur chaque page.
- Il n'y avait pas de logique centralisée, donc des petits écarts se glissaient sans qu'on s'en rende compte (arrondis différents, espacements inégaux...).
- Et les changements de dernière minute devenaient longs et risqués, parce que tout était "en dur", sans automatisation.

Exemple 2 : Quand on attend trop pour structurer

Une autre expérience qui m'a aussi marquée, j'étais arrivée sur un projet déjà bien avancé, dans une phase où l'on faisait une première proposition de design pour un client. Comme rien n'était encore validé, le designer qui était sur le projet avait choisi de ne pas créer les composants tout de suite, pour éviter de structurer des éléments qui allaient peut-être changer. L'idée était de gagner du temps au départ.

Mais avec le recul, je pense qu'il faut savoir s'adapter à la situation. Même si ce n'est pas encore validé, il y a parfois un bon moment pour commencer à créer certains composants. Quand on sent qu'un élément revient souvent, ou qu'on commence à le modifier à plusieurs endroits, c'est un signal qu'il faut le formaliser, pour éviter de perdre du temps ensuite.

Dans ce projet, le client a finalement demandé plus de maquettes, et comme les composants n'étaient toujours pas créés, on a dû remettre tous les écrans au propre manuellement, ce qui a été très long. Avec un peu d'anticipation, on aurait pu commencer à structurer progressivement, même sans validation finale.

Même un UI Kit ne fait pas tout

Même avec un UI Kit bien construit, **des petites imprécisions peuvent subsister**. Quand on design, on pense d'abord à ce qui est nécessaire sur le moment. On prévoit par exemple les états d'un champ de formulaire (rest, error, success) et parfois, par automatisme, on ajoute hover ou disabled parce qu'on sait que ce sont des cas standards pour les développeurs.

Mais **il arrive aussi qu'un état soit oublié**, tout simplement parce qu'il n'était pas encore utilisé dans les écrans à ce moment-là. Les développeurs viennent alors parfois demander des confirmations très précises, du type : "Pour ce bouton secondaire, c'est bien le bleu clair en mode repos, et le violet en hover ?"

Ce sont des détails, mais qui montrent que **tant qu'un système n'est pas documenté, chacun peut interpréter à sa façon**, et **c'est là que les écarts commencent à apparaître**.

3.2 Quand le Design System fluidifie les échanges... ou pas

Un des apports les plus concrets du **Design System**, c'est **la facilitation des échanges entre designers et développeurs**. Quand tout est bien documenté, que les composants sont clairs, et que chacun parle le même langage, **les allers-retours sont largement réduits**.

Sur mes projets, j'ai constaté que cette documentation partagée avec les bons états, les bons styles, les bons noms permet aux développeurs d'intégrer plus rapidement les éléments sans se poser trop de questions. Même si je n'ai pas toujours les échanges directs avec les développeurs, j'observe cette **fluidité dans la collaboration** entre mon manager et les devs. Quand les composants sont bien pensés dès le départ, **les questions diminuent**, car les **réponses sont déjà dans la documentation**.

Mais **cela ne veut pas dire qu'il n'y a plus aucune zone floue**. Certaines questions ne peuvent apparaître qu'au moment de l'intégration : un comportement technique, une contrainte liée à la façon dont le composant doit être intégré côté développement, ou une animation qui ne fonctionne pas comme prévu. Et c'est totalement normal. Un bon Design System **réduit les risques d'erreurs ou d'oublis**, mais il ne les élimine pas entièrement. Ce que j'ai aussi remarqué, c'est que la qualité du design system dépend beaucoup de l'expérience et du profil de la personne qui le construit.

Par exemple, dans mon entreprise, mon manager a une approche très structurée : il prend le temps d'anticiper les points techniques dès la phase de création. Il arrive à se poser les bonnes questions : "Si j'étais développeur, qu'est-ce que j'aurais besoin de savoir ici ?" Mais ce genre de recul, je pense que ça s'apprend avec l'expérience, et aussi en fonction de ses connaissances en développement. On ne l'a pas forcément au début, et c'est normal.

Ce que je constate aujourd'hui, c'est qu'un Design System est **avant tout un outil de collaboration entre design et développement**, bien plus qu'un simple moyen d'aller plus vite. Il ne suffit pas de "faire un système" dans son coin. Ce qui fonctionne, c'est quand le système est pensé aussi pour ceux qui vont l'utiliser et l'implémenter. Et même dans ces conditions, il restera toujours des ajustements à faire. Mais au moins, l'équipe avance avec une **base commune**, qui permet de **réduire les imprévus**, et de **gagner en fluidité**.

“

Est-ce que le Design System a un impact selon toi sur la collaboration entre designers et développeurs ?

De la meilleure communication et moins d'aller-retours entre les différentes filières.

Valentin LE COZ - UX/UI Designer

Est-ce qu'avoir une documentation claire (avec variantes, états, cas d'usage) facilite ton travail d'intégration ? Et qu'est-ce que ça change quand il n'y a qu'un UI Kit sans documentation ?

C'est beaucoup plus facile de comparer et de tester quand tout est à plat. Quand il n'y a qu'un UI Kit sans documentation, c'est plus long comme il y a plus de communication entre départements.

Sylviane TRAN - Développeuse

”



Ce que ça montre :

Le Design System ne se contente pas de faire **gagner du temps** : il **favorise** aussi **les échanges entre designers et développeurs**.

Une **documentation claire** et **complète** permet aux développeurs de **travailler plus rapidement** et de façon plus **autonome**. À l'inverse, un simple **UI Kit** demande **davantage d'échanges** avec les designers, ce qui peut **ralentir l'intégration**.

3.3 Outils partagés : Figma, tokens, Storybook...

Un **Design System**, ce n'est pas juste un fichier bien organisé. C'est aussi et surtout des **outils concrets**, pensés pour **faciliter son utilisation**, **sa maintenance**, et **sa transmission**... tant que le produit continue d'évoluer.

Figma : mon outil du quotidien

Figma est l'**outil** que j'utilise chaque jour pour **construire** mes **maquettes**. C'est une plateforme très complète, qui évolue constamment pour améliorer le quotidien des designers. L'objectif est clair : nous aider à designer plus efficacement, tout en améliorant la collaboration avec les développeurs.

Dans mon entreprise, on utilise notamment :

- Les **variables de couleurs** : on définit une palette centralisée, où chaque couleur (violet, bleu, vert, etc.) possède ses déclinaisons du plus clair au plus foncé (par exemple violet-100 à violet-900). Cela permet d'**assurer des usages cohérents** dans tout le projet et de tout mettre à jour facilement en modifiant la variable source,
- Les **styles de texte** et les **effets** : pour éviter de redéfinir manuellement chaque taille ou chaque ombre, on applique des **styles prédéfinis**, ce qui **garantit une cohérence** dans tout le projet,
- Les **composants réutilisables** : comme les boutons, champs de formulaire, ou icônes. Une fois créés, ils sont utilisés dans toutes les maquettes, ce qui fait **gagner du temps** et **assure une continuité visuelle**,
- Les **composants de structure** : quand une même disposition revient souvent (par exemple une image à gauche, texte à droite), on en fait un composant. Cela permet d'**accélérer la production** tout en maintenant la même logique d'alignement, de marges et d'espacements.

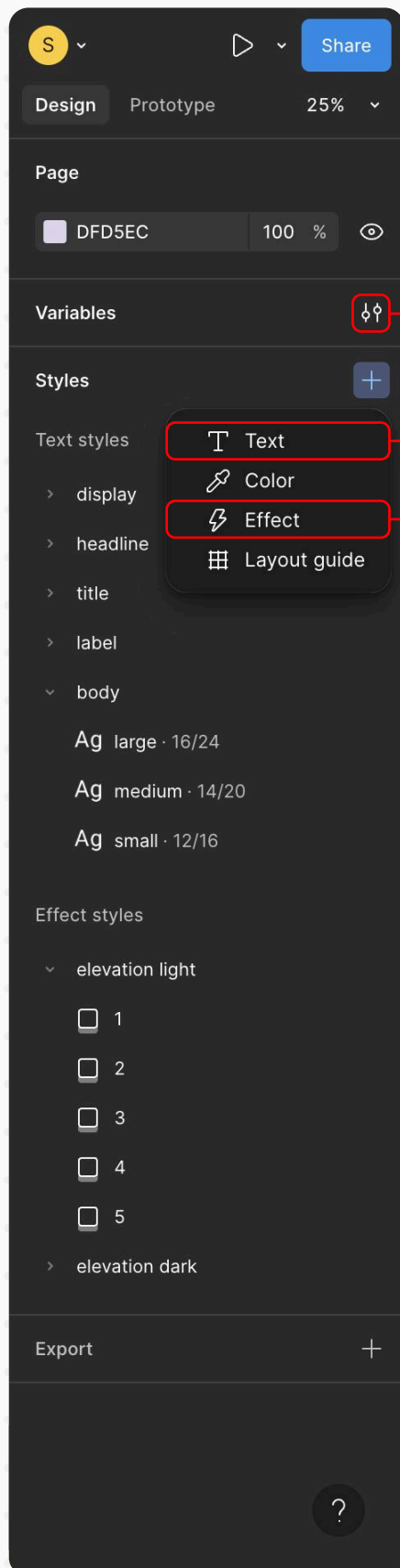
Figma propose aussi un mode Dev, que les développeurs peuvent activer lorsqu'ils doivent intégrer un composant. Ce mode affiche les dimensions, les marges, le padding, les couleurs, la typo, ou encore le border-radius, un peu comme dans un inspecteur de navigateur. Certains styles sont même directement proposés sous forme de code à copier ce qui limite les erreurs et fait gagner du temps.

Définition :

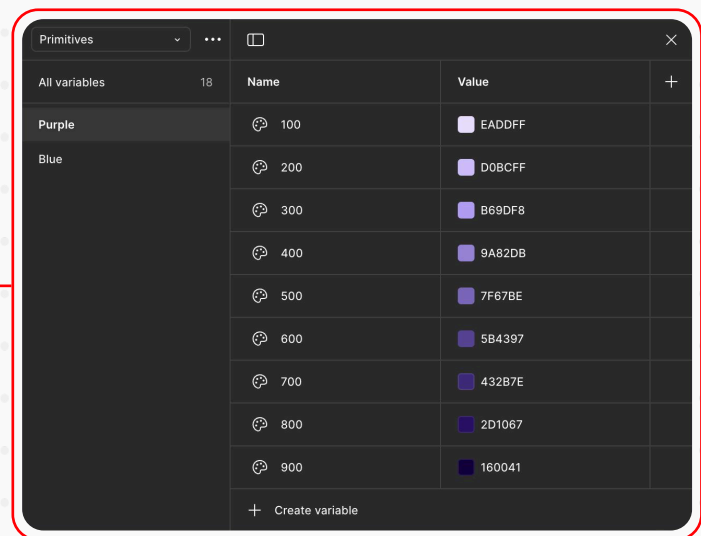
- Une **variable de couleur** est une valeur brute enregistrée en code hexadécimal, par exemple #6750A4 pour le violet. On l'applique là où on en a besoin. Si la variable change, tous les éléments qui l'utilisent se mettent à jour automatiquement.
- Un **style de texte** ou **d'effet** est un réglage enregistré (police, taille, ombre...) qu'on applique en un clic. Si on le modifie, tous les éléments liés se mettent à jour automatiquement.

Zone de création des variables de couleurs , styles de textes et des effets

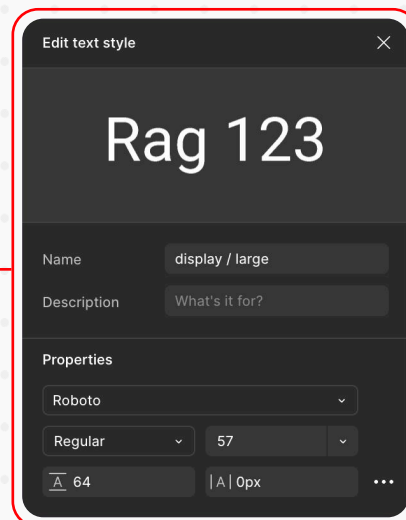
Voilet de travail



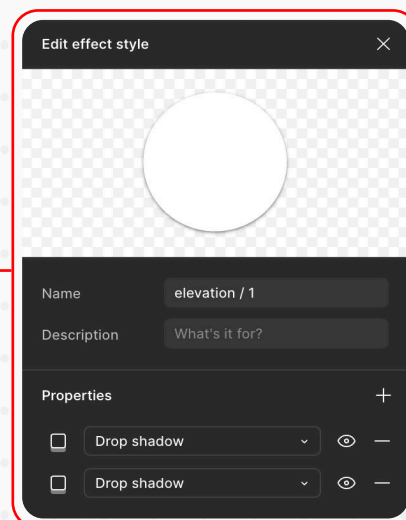
Panel de création de variables de couleurs



Panel de style de texte



Panel de style d'effet



La logique des tokens : structurer l'usage

Quand on met en place un Design System, on ne pense plus seulement en couleurs "visuelles", mais en **usages précis et cohérents**. C'est ce qu'on appelle la logique des tokens.

Voici comment nous l'avons mis en place sur le projet de gestion de restaurant :

1. Création des variables de couleurs :

Ce sont les couleurs "brutes" du projet, organisées selon leur intensité.

Par exemple :

- violet-100, violet-200, ..., violet-500
- bleu-100, bleu-200, ..., bleu-500

Ces variables ne portent pas encore de signification fonctionnelle : ce sont juste des valeurs.

2. Définition des tokens :

Ensuite, on définit des tokens, qui représentent des usages concrets dans l'interface.

Par exemple :

- background-primary → couleur du fond
- border-primary-muted → bordure discrète
- on-background-primary → couleur du texte sur fond principal

3. Association des tokens aux variables :

Une fois les usages fixés, on les associe à des variables :

- background-primary = violet-500
- on-background-primary = blanc

Si un jour on décide de changer le fond principal du violet au bleu, il suffit de modifier l'association du token (par exemple background-primary = bleu-500) sans toucher aux composants un par un.

Cette **séparation entre la valeur** (la couleur) **et l'usage** (le rôle dans l'interface) rend l'interface plus souple et plus cohérente dans le temps. On peut modifier une couleur à un seul endroit, tout reste aligné automatiquement, et on **évite les erreurs ou les ajustements manuels fastidieux**.

Dans un UI Kit, on peut aussi définir des couleurs, mais comme il n'y a pas toujours de normalisation claire, certains usages peuvent être interprétés différemment. Et si un autre designer reprend le fichier, ou si on oublie nos propres règles, des incohérences visuelles peuvent apparaître sans même qu'on s'en rende compte.

C'est pourquoi ce travail **demande une vraie vision d'ensemble** :

- **anticiper les cas d'usage** (titres, fonds, bordures...),
- **nommer les tokens de façon claire**,
- **conserver une logique constante** dans tous les écrans.

Ce qui est intéressant, c'est que le **nommage des tokens** côté design peut être **aligné avec les variables utilisées en code**. Cela crée un vrai **langage commun** entre designers et développeurs, et **limite les erreurs d'interprétation**.

Mise en place des tokens

Étape 1 : Création des variables de couleurs

Primitives	...				×
All variables	72	Name	Value		+
Purple		0	000000		↕ ↕
Blue		5	160041		
Red		10	21005D		
Grey		15	2D1067		
		20	381E72		
		25	432B7E		
		30	4F378B		
		35	5B4397		

Étape 2 & 3 : Définition et association des tokens

Tokens	...				×
All variables	21	Name	Value		+
		background-primary	Purple/40		
		background-primary-fa...	Purple/90		
		background-primary-hi...	Purple/30		
		background-primary-fa...	Purple/80		
		on-background-primary	Purple/100		
		border-primary	Purple/30		
		border-primary-faded	Purple/20		
		background-secondary	Blue/40		

Répercussion des modifications dans leur ensemble

background-primary = Purple/40

❖ Button

+ Button

❖ Icon Button

❖ Switch

Tokens

...

All variables

21

Name	Value		+
background-primary	Purple/40		
background-primary-fa...	Purple/90		
background-primary-hi...	Purple/30		
background-primary-fa...	Purple/80		↕
on-background-primary	Purple/100		
border-primary	Purple/30		
+ Create variable			

En modifiant la couleur associée au token, tous les éléments liés se mettent à jour automatiquement.

❖ Button

+ Button

❖ Icon Button

❖ Switch

Tokens

...

All variables

21

Name	Value		+
background-primary	Blue/40		
background-primary-fa...	Purple/90		
background-primary-hi...	Purple/30		
background-primary-fa...	Purple/80		
on-background-primary	Purple/100		
border-primary	Purple/30		
+ Create variable			

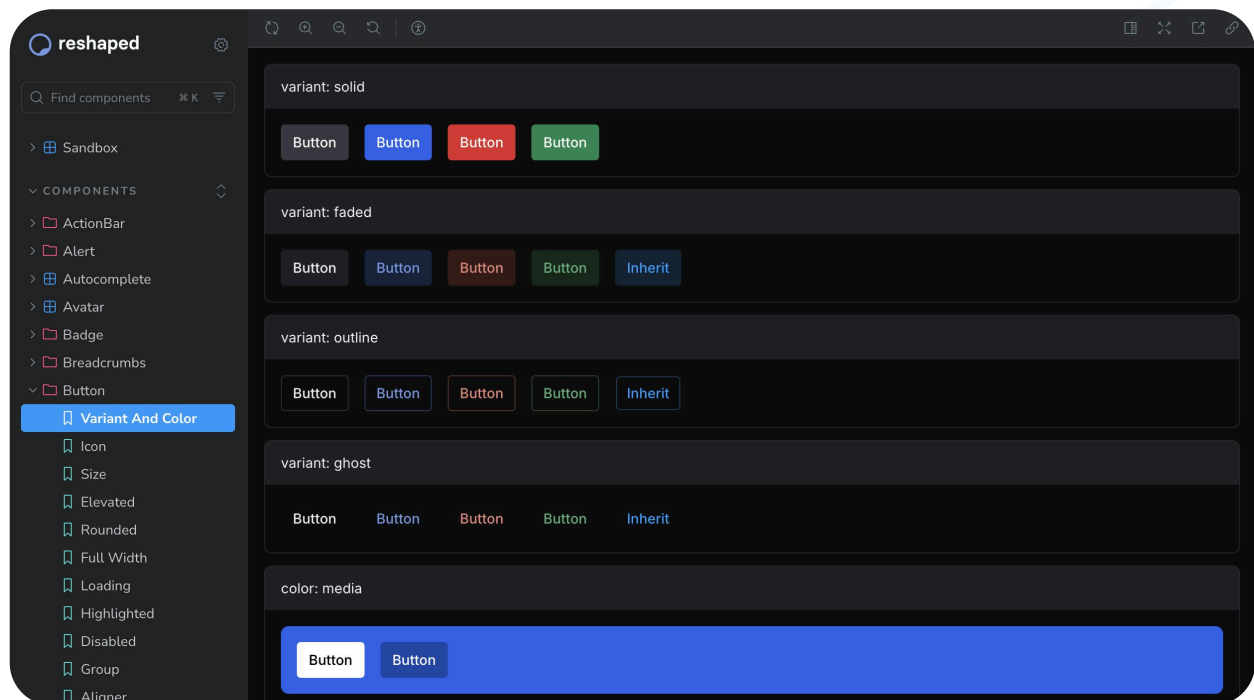
Storybook (ou Histoire) : la version développeur du design system

Sur ce même projet, les développeurs ont mis en place un outil appelé Histoire, équivalent à Storybook. C'est une **librairie interactive** de composants, **que les développeur utilisent pour valider et tester chaque élément de l'interface**.

Elle permet :

- de **voir tous les composants disponibles**,
- de **tester leurs différents états** (repos, hover, désactivé, etc.),
- de **vérifier** que **le rendu** correspond bien à celui prévu en design.

Même si je ne l'utilise pas directement, c'est un **excellent outil de validation croisée**. Il permet aussi de **repérer des oublis**, des cas mal gérés, ou tout simplement de mieux comprendre comment le composant se comporte réellement une fois en production.



Ce qui aide vraiment : la communication

Au-delà des outils, ce qui fait vraiment la différence, ce sont les échanges entre les équipes. Dans mon entreprise, on organise régulièrement des réunions inter-métiers pour que chacun puisse :

- **présenter son avancement**,
- **poser des questions** sur les besoins de l'autre pôle,
- **anticiper ensemble les décisions** liées à l'interface ou à l'intégration.

Un design system efficace repose à la fois sur des outils adaptés, une structure claire, mais surtout une vraie collaboration dès la phase de création.



04.

**Mon expérience chez
Creatiivity : entre théorie et
pratique**

4.1 Mise en place d'UI Kits et apprentissages

Mettre en place un UI Kit est devenu une étape systématique pour moi. Que ce soit pour des sites vitrines, des plateformes web ou des applications mobiles, c'est une manière de poser des bases solides avant d'entrer dans la production.

De la direction artistique à la structuration

Selon les projets, soit une direction artistique est déjà définie par le client avec une charte graphique transmise soit c'est à nous de proposer des pistes graphiques, à partir d'un brief, de moodboards fournis... ou même parfois en totale liberté créative. Dans tous les cas, on commence souvent par créer 3 à 4 écrans de proposition pour valider une direction. Dès cette phase, certains éléments apparaissent comme récurrents (boutons, blocs, icônes...), ou sont simplement des standards qu'on sait qu'on utilisera. On commence donc à les intégrer dès cette étape, pour structurer le travail et gagner en efficacité.

Les fondations d'un UI Kit efficace

Avant de créer les composants eux-mêmes, on pose les **bases du design** :

- La **palette de couleurs**, structurée avec des familles (primary, secondary, tertiary, neutral, etc.), chacune déclinée en nuances (100 à 900). En général, 500 est la couleur de référence,
- Les **styles de texte**, organisés en familles (display, title, label, paragraph) et déclinés en plusieurs tailles (small, medium, large) et graisses (regular, medium, bold),
- L'**iconographie**, pour assurer une cohérence visuelle dans tous les composants.

Ensuite, on construit les **composants de base**, ceux qu'on utilise dans presque tous les projets :

- les **boutons** (avec des variantes solid, faded, ghost, et des tailles différentes),
- les **champs de saisie**,
- les **éléments de navigation** (comme les onglets, les menus...).

Fondation d'un UI Kit

Typographies

Display

Display Large - Roboto 57/64 . 0

Display Medium - Roboto 45/52 . 0

Display Small - Roboto 36/44 . 0

Headline

Headline Large - Roboto 32/40 . 0

Headline Medium - Roboto 28/36 . 0

Headline Small - Roboto 24/32 . 0

Title

Title Large - Roboto Regular 22/28 . 0

Title Medium - Roboto Medium 16/24 . +0.1

Title Small - Roboto Medium 14/20 . +0.1

Label

Label Large - Roboto Medium 14/20 . +0.1

Label Medium - Roboto Medium 12/16 . +0.5

Label Small - Roboto Medium 11/16 . +0.5

Body

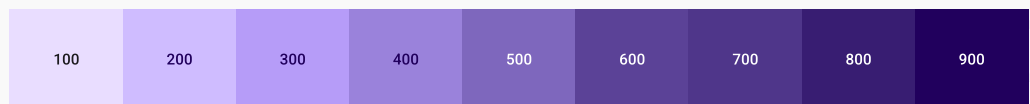
Body Large - Roboto 16/24 . +0.5

Body Medium - Roboto 14/20 . +0.25

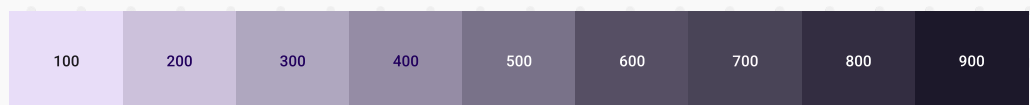
Body Small - Roboto 12/16 . +0.4

Couleurs

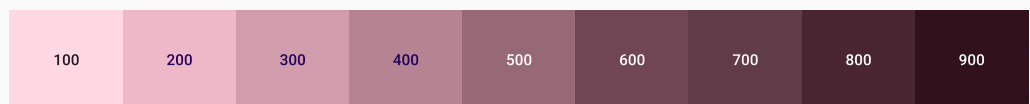
Primary



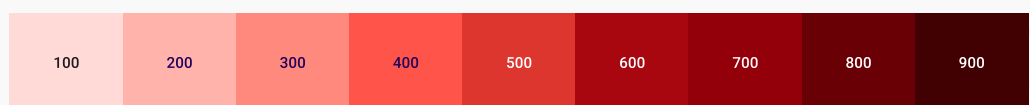
Secondary



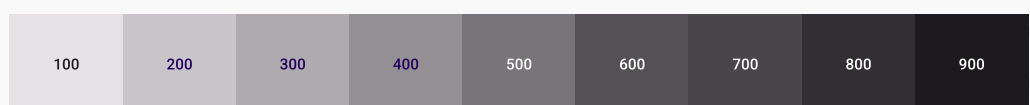
Tertiary



Error



Neutral



Neutral Variant



Une méthode transmise et répétée

Cette manière de structurer un UI Kit, je l'ai acquise progressivement grâce à une **transmission de savoir-faire** de mon manager. Aujourd'hui, elle est bien ancrée : à **chaque nouveau projet**, les étapes reviennent naturellement. Et même si je ne m'appuie pas systématiquement sur de grands Design Systems comme Material, je prends souvent des repères dans les projets que j'ai déjà réalisés. Il m'arrive de repenser à un composant que j'avais conçu dans un autre contexte, et de **rouvrir l'ancien fichier** pour **retrouver une logique**, une structure ou une idée que je peux **réadapter**.

Un cadre structuré, mais pas encore documenté

Ce que je construis dans ces UI Kits n'est pas un Design System mais j'essaie toujours de **garder une logique globale** : nommer les éléments avec cohérence, prévoir les variantes nécessaires, organiser les composants proprement. Même sans aller jusqu'à une documentation formelle, je fais en sorte que **l'ensemble reste clair et structuré**, pour **éviter les erreurs** ou **les redondances**.

Et pour un autre designer qui reprend le projet ?

C'est là que je vois la principale **limite d'un UI Kit**. Quand un autre designer arrive sur le projet, il doit souvent **prendre le temps d'analyser** ce qui a été fait :

- observer comment les composants ont été construits,
- repérer les logiques d'usage (quelles couleurs pour quels cas, quels espacements, quelles tailles de texte...),
- parfois poser des questions si certaines choses ne sont pas évidentes.

Dans un **Design System**, ces **règles sont documentées, illustrées, accessibles**. Le travail est déjà "pré-digéré" : on peut se référer à une fiche composant, voir les différentes variantes, et comprendre rapidement les intentions.

Dans un UI Kit, même bien structuré, on doit souvent analyser et déduire soi-même certaines règles. Le Design System, lui, les rend visibles et partagées, et c'est ce qui m'a donné envie de m'y intéresser davantage.

4.2 Contribution à un Design System complet

J'ai eu l'opportunité de participer à la création d'un Design System complet au sein de mon entreprise, sur un **projet de logiciel de gestion dédié à la restauration**. C'était la première fois que l'équipe design s'engageait réellement dans cette démarche et c'est aussi la première fois que j'y étais impliquée à ce point.

Pourquoi ce projet avait besoin d'un design system ?

Le produit existait déjà, avec une librairie partagée qui regroupait des composants utilisés sur les trois plateformes principales. Mais dans les faits, chaque plateforme avait ses composants regroupés à part, comme s'il s'agissait de trois UI Kits distincts réunis dans un même fichier.

Le projet étant en pleine évolution, c'était l'occasion de :

- **moderniser l'interface** (typographies, couleurs, espacements, arrondis...),
- **unifier les bases visuelles**,
- **tester une méthode plus durable** : le Design System.

Mon rôle dans la mise en place

J'ai participé à la **création des composants** dans Figma, en binôme avec mon manager. On se répartissait les composants à créer, et je faisais des revues régulières avec lui pour valider leur structure, leur organisation et leur nommage. J'étais également en charge de **documenter chaque composant** dans Figma, en y intégrant :

- l'anatomie (structure du composant),
- les variantes de style (solid, faded, ghost...),
- les tailles (x-small, small, medium, large, x-large),
- les apparences (primary, critical, positive, information...),
- les états (rest, hover, active, focus-visible),
- les configurations (avec icône à gauche, à droite, ou sans).

Chaque composant avait **sa propre page** Figma avec une **documentation visuelle complète**, directement consultable par les développeurs.

Les tokens de couleurs : un apprentissage progressif

Au départ, mon manager m'a laissée tester la mise en place des **tokens de couleur**, en m'appuyant sur des Design Systems comme Material Design ou Reshaped. Mais rapidement, il a repris la main, car cette phase demandait une **réflexion** plus **complexe** qu'il n'y paraît.

En effet, cela **exige une vision globale** avec du recul pour **anticiper tous les usages** des couleurs (fonds, textes, bordures, états...), ainsi qu'**une logique rigoureuse** dans le nommage des tokens, afin de garantir un sens clair et cohérent entre designers et développeurs, et des choix de couleurs pertinents et harmonieux.

Il m'a ensuite expliqué toute la logique qu'il avait mise en place, afin que je puisse m'y référer lorsque je crée les composants. C'est ce qui donne toute sa force au **Design System** : une **logique partagée, cohérente sur l'ensemble des composants**, et alignée à la fois dans l'usage des couleurs et dans la manière dont les développeurs les implémentent.

Ce que ça m'a appris

Cette expérience m'a vraiment permis de faire évoluer ma manière de travailler.

J'ai appris à :

- travailler avec des variables, au lieu d'appliquer des couleurs directes,
- penser à l'ensemble des cas d'usage dès la création,
- structurer les composants avec rigueur, dans leur forme comme dans leur nommage.

J'ai aussi compris que la mise en place d'un Design System demande beaucoup de temps, mais qu'elle permet :

- d'améliorer la collaboration avec les développeurs,
- de réduire les erreurs à l'intégration,
- gagner en efficacité sur le long terme, surtout dans les projets qui évoluent constamment.

4.3 Leçons tirées, limites rencontrées

Participer à la création d'un Design System m'a permis de prendre du recul sur ma manière de travailler, et de mieux comprendre ce qui fait la différence entre un bon UI Kit et un vrai système cohérent, partagé, documenté.

Ce que j'ai compris en pratiquant

Un **UI Kit contient déjà une structure claire** : on y retrouve des règles dans les tailles de texte, les couleurs, les espacements, les marges, etc. On pense à la cohérence, **mais souvent de manière intuitive, sans aller jusqu'à tout formaliser**.

Avec un **Design System**, on va plus loin :

- On rend ces **règles visibles et partagées**.
- On ajoute des **éléments structurants** comme les **tokens**, la **documentation** d'usage, une logique d'interaction plus poussée.

Dans un UI Kit, on applique souvent une logique de nommage qu'on a déjà utilisée dans d'autres projets, et qu'on continue par habitude. Mais sans cadre vraiment défini, on peut se retrouver avec des incohérences dans les noms ou dans la construction des composants, sans s'en rendre compte tout de suite.

Exemple :

Dans certaines situations, j'ai nommé la propriété liée à l'icône "visuel", et dans d'autres "icon". Ce n'est pas bloquant, mais si un autre designer arrive sur le fichier, il peut se demander si c'est une différence voulue ou juste un oubli de cohérence.

Un **UI Kit** donne souvent une **vue d'ensemble de tous les composants, rangés dans une même page**. C'est pratique pour tout voir d'un coup, surtout quand on cherche à repérer ce qui existe déjà.

Un **Design System**, lui, présente les choses composant par composant, avec tous leurs cas d'usage, variations et règles. C'est **plus précis** et **mieux documenté**, mais cela demande parfois un peu plus de temps pour se représenter la vision globale du système, car on découvre les composants un par un.

Pour moi, le **Design System est une version plus complète du UI Kit**, notamment grâce à la mise en place de tokens, à la documentation des composants, et à l'alignement renforcé entre le design et l'intégration. Il permet de transformer une logique personnelle en un outil partagé, robuste, et durable.

Une erreur que j'ai (failli) faire

Dans un autre projet, j'ai eu à utiliser un Design System déjà existant, que je devais adapter à un nouveau besoin. Par réflexe, j'ai commencé à documenter tous les changements que je faisais, par peur que les développeurs ne soient pas au courant. Avec le recul, je pense que ce n'était pas inutile mais qu'**il faut savoir adapter le niveau de détail à la réalité du projet**. Sur un projet court, une note claire ou un échange direct peut suffire. Ce qui compte, c'est que les informations soient transmises pas forcément formalisées dans une documentation complète.

Ce que j'essaie d'appliquer aujourd'hui

J'accorde plus d'attention à :

- **poser les bonnes bases dès le départ** : palette de couleurs, tailles et graisses des typographies, arrondis, etc.,
- **penser aux usages récurrents** pour ne pas attendre la validation finale avant de créer des composants,
- **prendre du recul avant de se lancer** dans la production pure.

J'ai aussi appris que vouloir aller vite peut parfois nuire à la clarté du système. Prendre un moment pour réfléchir à la logique d'ensemble, c'est un gain de temps sur le long terme.

Ce que je conseillerais à quelqu'un qui débute un design system

D'abord, s'inspirer des autres. Regarder des Design Systems bien construits (comme Material ou Reshaped) permet de repérer ce qui fonctionne : logique des tokens, règles d'espacement, structure des composants... Ensuite, **ne pas viser la perfection dès le début**. Il vaut mieux poser une base simple, cohérente et évolutive, plutôt que de chercher à tout figer trop tôt. Et surtout, penser à ceux qui vont utiliser le système, côté design comme côté dev. Un Design System n'est pas un simple fichier Figma. C'est **un outil d'équipe**, fait **pour faciliter le travail de chacun**.



05.

**Comment décider : UI Kit ou
Design System ?**

5.1 UI Kit ou Design System : comment faire le bon choix ?

Il n'y a pas de règle universelle pour savoir quand passer d'un UI Kit à un Design System. Mais avec l'expérience, on repère **certains signes**, certaines **situations** où il est temps d'organiser le projet de façon plus claire et solide. Dans mon contexte, je ne me pose pas toujours mille questions en amont parce que je sais déjà, avec les types de projets qu'on traite chez **Creativity**, que le Design System n'est pas toujours adapté. On travaille souvent sur des projets courts, qui n'ont pas vocation à évoluer sur le long terme. Dans ce cas, un bon UI Kit suffit largement. Et je sais aussi que mettre en place un Design System demande du temps, du recul, et que ce n'est pas toujours faisable selon les délais ou les contraintes. J'ai quand même repéré certains éléments qui aident à prendre la bonne décision dès le début.

Le projet est-il amené à évoluer dans le temps ?

Si c'est un **projet ponctuel**, comme un site vitrine qui ne sera pas mis à jour régulièrement : un **UI Kit bien construit suffit largement**. Mais si le projet est prévu pour **évoluer sur le long terme** avec des **ajouts réguliers**, des mises à jour, ou des **besoins de maintenance** alors il peut être utile de **penser Design System** dès le départ. Ou au moins de poser une base claire et structurée qui pourra évoluer dans cette direction.

Combien de designers travaillent dessus ?

Quand on est plusieurs, même brièvement, **il faut une logique claire et partagée**. Chez nous, on met très tôt un **UI Kit** en place, même sur des petits projets, pour **éviter les allers-retours inutiles**. Et quand un designer arrive en cours de route, c'est toujours **plus simple de lui transmettre un fichier bien structuré**, avec une logique déjà en place.

Est-ce que les développeurs auront besoin de détails précis à l'intégration ? Avec le temps, et les échanges réguliers avec les développeurs, on a appris à anticiper ce dont ils auront besoin : les différents états, les tailles, les règles d'interaction ou d'accessibilité... On intègre ces éléments directement dans le UI Kit, grâce à l'expérience transmise par mon manager, ce qui nous permet d'éviter des oublis ou des aller-retours en fin de projet.

5.2 Les bonnes questions à se poser avant de se lancer

Quand je démarre un projet, je ne me pose pas forcément mille questions techniques à l'avance. Mais avec l'expérience, j'ai fini par adopter une façon de faire assez claire, que je reproduis presque à chaque fois. Elle me permet de structurer mes fichiers Figma efficacement, sans perdre de temps plus tard.

Comprendre le contexte du projet

Tout commence par une **phase de brief**. Je regarde le **cahier des charges**, puis je vérifie :

- Est-ce qu'il y a une **direction artistique** ou une **charte graphique existante** ?
- Sinon, quel est l'univers visuel souhaité ? Le client a-t-il un moodboard, des exemples de style, des inspirations ?
- Y a-t-il des wireframes fournis, ou devons-nous concevoir tout le parcours UX ?
- Même si le besoin est centré sur l'UI, je reste attentive à l'expérience globale, et je propose des ajustements si certains éléments me paraissent incohérents.

Poser une base visuelle solide

Quand je travaille sur les premiers écrans clés (en général 3 à 4 propositions de maquettes), **je respecte les contraintes visuelles transmises** : charte graphique, typographies, couleurs... Je peux déjà intégrer une première version de la palette de couleurs, surtout si une teinte principale est imposée. Une fois la direction artistique validée par le client, je passe à l'étape suivante : j'itère la DA sur l'ensemble des maquettes, tout en posant les **fondations de mon UI Kit**.

Créer les fondations au fil de l'itération

Pendant la phase de design, je structure progressivement :

- les **variables de couleurs**, bien nommées et organisées,
- les **styles de texte** (avec des tailles et graisseurs cohérentes selon les niveaux),
- les **composants récurrents** comme les boutons, les champs, les tags...

Dès qu'une structure de mise en page revient plusieurs fois (par exemple une section avec des cards), je la transforme aussi en composant. C'est un réflexe qui me fait gagner beaucoup de temps, surtout lors des retours ou ajustements.

Structurer le fichier Figma

J'ai gardé une logique inspirée des projets précédents, transmise aussi par mon manager :

- **Une page "Foundation"** : couleurs, styles typographiques, icônes...
- **Une page "Composants"** : tout ce qui construit l'interface, du bouton à la section.

Certains composants sont imbriqués dans d'autres : par exemple un bouton placé dans une card, elle-même intégrée dans une section. Cette structure me permet de garder une cohérence dans tout le projet, et de faciliter les modifications globales.

Tout ça, ce sont des réflexes que j'ai développés avec l'expérience et avec les conseils de mon manager. Aujourd'hui, je sais où je vais, comment structurer un fichier proprement, et surtout comment poser des bases solides pour la suite du projet.

5.3 Les trois piliers : gouvernance, documentation, maintenance

Créer un Design System, c'est une chose. Mais pour qu'il reste utile dans le temps, trois éléments sont essentiels : la gouvernance, la documentation et la maintenance.

Qui a la main sur le système ?

Dans le cadre du projet de gestion de restauration, c'est le chef de projet et le lead design qui jouent ce rôle. En l'occurrence, c'était mon manager. C'est lui qui repère les ajustements nécessaires, qui valide les évolutions à apporter, et qui décide si un nouveau composant mérite d'être intégré dans le système. Parfois, c'est lui qui fait les modifications. D'autres fois, c'est moi qui m'en occupe. Mais aujourd'hui, c'est principalement lui qui entretient et supervise l'évolution du design system.

Documentation : pour transmettre plus que des maquettes

Lorsque j'étais activement impliquée dans le projet, j'étais aussi en charge de la documentation des composants. On documentait chaque élément avec une structure claire :

- L'**anatomie du composant** (ses parties principales),
- Les **variantes** (styles visuels, comportements),
- Les **états** (rest, hover, focus, disabled...),
- Les **tailles** disponibles (small, medium, large...),
- Les **propriétés configurables** (avec ou sans icône, texte aligné à gauche, etc.).

L'objectif était simple : que les développeurs puissent retrouver toutes les infos utiles en un seul endroit, et intégrer les composants exactement comme on les avait pensés.

Maintenance : faire vivre le système

Aujourd'hui, la maintenance du Design System est assurée par mon manager. Il est en charge de garder le système cohérent, à jour, et de l'ajuster en fonction des besoins du produit. De mon côté, je n'ai pas encore assez de recul pour dire comment le système évolue dans le temps. Mais je sais qu'un **Design System demande un suivi régulier** pour ne pas devenir obsolète. Sinon, il finit par ne plus être utilisé ou même peut devenir source de confusion.

C'est ce qui distingue un Design System "vivant" d'un fichier figé, il ne suffit pas de créer des composants. Il faut aussi décider qui les gère, comment on les transmet, et qui veille à leur évolution.



06.

**Recommandations &
bonnes pratiques**

6.1 Ce que j'ai appris sur le terrain

- **Créer un UI KIT devient plus intuitif avec l'habitude.** Aujourd'hui, je structure plus vite : je sais où placer mes variables, mes composants, et comment organiser le fichier.
- Commencer à **structurer dès les premières maquettes** est une bonne base. Même sans tout formaliser tout de suite, repérer les éléments récurrents permet d'anticiper leur construction.
- **Imbriquer les composants** (boutons dans cards, cards dans sections...) est très utile pour les changements globaux.
- Revenir sur d'**anciens projets** bien structurés me sert souvent de **point de repère** ou **d'inspiration** quand je démarre un nouveau kit UI.

6.2 Conseils pour gagner du temps et éviter les erreurs

- Un **composant ne vit jamais seul** : il doit s'inscrire dans une **logique commune** avec les autres, mêmes tailles, mêmes espacements, mêmes comportements. Sinon, même s'il fonctionne en apparence, il risque de casser la cohérence, et de semer le doute, pour les autres designers et les développer.
- La **documentation aide** aussi **à se relire soi-même**. Même un simple tableau ou une description visuelle peut éviter des oublis.
- Les **échanges avec les développeurs sont essentiels**, surtout pour comprendre ce qui leur manque. Cela permet de mieux anticiper leurs besoins au moment de l'intégration.
- Quand on veut aller vite, on peut négliger certains détails... Mais **prendre du recul** au bon moment évite souvent de tout revoir plus tard.

“

Est-ce que le design system te fait gagner du temps lors de l'intégration ? À quel moment le ressens-tu ? Et est-ce que ça t'est déjà arrivé de faire des retours ou des suggestions aux designers pour améliorer le design system ?

Absolument ! Tu le ressens au bout du 2ème composant créer, tout devient plus rapide. J'ai déjà conseillé des designers pour enrichir un design system surtout sur les différents states qu'il peut avoir par exemple sur des input et des boutons.

Sylviane TRAN - Développeuse

”

6.3 Conseils pour les designers (et pour les devs)

Pour les designers

- **Regardez comment font les autres** (design systems ou kits UI) : ça aide à structurer ses idées et à faire de meilleurs choix.
- **Commencez par des fondations solides** (palette, typos, styles...), même simples. Ça pose une base claire dès le départ.
- Pensez tout de suite aux états, aux tailles, aux usages répétés.

Pour les développeurs

- **Utilisez les composants comme référence** directe : ils ont été pensés pour être cohérents, même sans documentation complète.
- **Si un détail est flou, posez la question**. Il vaut mieux un échange rapide qu'une mauvaise interprétation.
- Faites-nous des retours aussi ! Savoir ce qui fonctionne bien ou ce qui manque nous aide à améliorer les projets suivants.

Ces bonnes pratiques, je les ai intégrées petit à petit, en observant, en testant, et en étant guidée par mon manager au fil des projets. Elles m'aident à gagner en efficacité, mais surtout à mieux collaborer avec le reste de l'équipe.

The background is a solid red color. It features several large, semi-transparent geometric shapes: a large circle in the top-left, a large square in the top-right, and a large circle in the bottom-left. These shapes overlap each other and the background. The text '07.' is positioned in the lower-left area, and the title 'Conclusion & Remerciements' is at the bottom center.

07.

Conclusion & Remerciements

Conclusion

Ce livre blanc m'a permis de faire une vraie **pause dans le rythme quotidien** de **prendre du recul** sur mes projets, mes méthodes, et surtout sur **mon évolution en tant que designer**.

En le rédigeant, j'ai réalisé à quel point **certaines choses étaient devenues des automatismes** : la création de composants, la structuration d'un kit UI, l'anticipation des états, des usages, des répétitions... Mais j'ai aussi pu mettre des mots sur des réflexions que je n'avais jamais vraiment formalisées. Et surtout, j'ai **clarifié pour moi-même la différence entre un UI Kit et un Design System**, et ce que chacun peut apporter à un projet.

Je sais que je suis encore en apprentissage et que chaque projet me permet d'affiner un peu plus ma méthode. Mais aujourd'hui, je me rends compte qu'**un bon Design System, c'est bien plus qu'un simple gain de temps**. C'est un cadre, une logique commune, un **outil de collaboration** qui **facilite le travail de toute l'équipe, sur la durée**.

J'espère que ce retour d'expérience, même s'il reste personnel, pourra servir à d'autres : pour structurer un fichier, gagner du temps, éviter les oublis... Mais surtout pour comprendre qu'**un projet bien organisé dès le départ, c'est souvent un projet plus fluide à construire et plus agréable à faire évoluer**.

Remerciements

Je tiens à remercier toutes les personnes qui m'ont accompagnée dans la réalisation de ce livre blanc, et plus largement tout au long de ma formation.

Merci à l'**école** et à l'**équipe pédagogique**, pour leur accompagnement bienveillant et leur disponibilité tout au long de l'année. Merci aussi de m'avoir donné l'opportunité de concevoir ce livre blanc dans un cadre à la fois libre et structurant.

Merci à **mes camarades de classe** pour leur soutien tout au long de l'année, les échanges, l'entraide, et les encouragements mutuels. Cette année a aussi été marquée par un vrai esprit de groupe, qui a rendu l'expérience plus motivante et plus enrichissante.

Merci à mon manager, **Nathan LIBOUTET**, pour sa confiance, ses retours toujours constructifs, et pour m'avoir transmis une méthode de travail solide, que je peux aujourd'hui appliquer avec assurance.

Merci à toute l'équipe de **Creatiivity**, pour m'avoir permis de travailler sur des projets variés, concrets, et d'expérimenter des méthodes directement applicables.

Merci aussi à **Denis CONSTANT**, **Valentin LE COZ** et **Sylviane TRAN**, pour leurs retours d'expérience et leurs contributions. Leurs témoignages ont enrichi ce livre blanc en apportant des regards complémentaires, ancrés dans le réel.

Enfin, merci à vous, **lectrices** et **lecteurs**. J'espère que ce document pourra vous servir de repère, vous inspirer ou vous donner envie d'explorer de nouvelles manières de structurer vos projets.

The background is a solid orange color with several large, semi-transparent geometric shapes. There are two large circles, one in the upper left and one in the lower left, and several triangles of different sizes and orientations scattered across the page, creating a modern, abstract design.

08.

Références & Glossaire

Références

Différences entre un UI Kit et un Design System

- <https://www.uinkits.com/blog-post/what-is-the-difference-between-a-ui-kit-vs-design-system>
- <https://medium.com/design-bootcamp/understanding-design-systems-and-ui-kits-building-a-cohesive-design-language-37a92c07ab9b>
- <https://www.beryl.design/fr/post/ui-kit-vs-design-system-comprendre-les-differences-cruciales>

Des Design Systems concrets à découvrir

- <https://m3.material.io/>
- <https://reshaped.so/>
- <https://polaris-react.shopify.com/>

Tokens & variables : les bases à connaître

- <https://m3.material.io/foundations/design-tokens/overview>
- <https://help.figma.com/hc/en-us/articles/18490793776023-Update-1-Tokens-variables-and-styles>

Visuels

Visuel des composants

<https://www.figma.com/community/file/1035203688168086460>

Visuel du storybook

<https://reshaped.so/storybook/index.html?path=/story/components-button--variant-and-color>

Illustration du LEGO

<https://pin.it/ykgxev3Mr>

Glossaire

- **UI Kit :**
Un ensemble d'éléments graphiques prêts à l'emploi (boutons, champs de saisie, icônes, couleurs, typographies...). Il sert à construire des maquettes rapidement, avec cohérence.
- **Design System :**
Une version plus poussée et complète qu'un UI Kit : il regroupe des composants, des styles, des tokens, des règles d'usage et une documentation. Il garantit la cohérence entre design et code et facilite la collaboration entre équipes.
- **Variable de couleur :**
Une couleur enregistrée sous forme de valeur brute (ex : #7F3FBF) à laquelle on donne un nom comme violet-500. Changer la variable met à jour tous les éléments qui l'utilisent.
- **Token :**
Un nom qui définit un usage précis (ex : background-primary) et qui est lié à une variable de couleur (ex : background-primary = violet-500). Cela permet d'assurer un usage logique et de tout modifier facilement si besoin.
- **Composant :**
Un élément interactif réutilisable (ex : bouton, champ de saisie) conçu une fois pour garantir la cohérence (tailles, marges, styles).
- **Instance :**
Un exemplaire d'un composant utilisé dans les maquettes. Si le composant principal est modifié, les instances se mettent à jour automatiquement.
- **Figma** (logiciel) :
Un outil de design collaboratif qui permet de créer, organiser et partager des maquettes et des composants avec toute l'équipe.
- **Librairie** (Figma) :
Un fichier partagé dans Figma qui contient des composants et des styles. Ces éléments peuvent être liés à d'autres fichiers pour être réutilisés facilement dans plusieurs projets.
- **Documentation :**
Un ensemble de consignes et d'exemples qui expliquent comment utiliser les composants, les tokens et les styles dans le projet.
- **State** (état) :
La variation d'un composant selon son interaction : normal, hover, focus, désactivé...
- **Naming** (noms) :
La logique dans la manière de nommer les composants, tokens et styles pour garder un système clair, compréhensible et aligné avec les développeurs.

Note de synthèse

À travers ce livre blanc, nous avons exploré **les différences, les complémentarités et les enjeux** d'un **UI Kit** et d'un **Design System**, en nous appuyant sur des exemples concrets et des retours d'expérience issus de projets réels.

Le Design System et l'UI Kit sont **deux outils complémentaires** mais **fondamentalement différents**. L'UI Kit fournit un ensemble d'éléments graphiques essentiels (boutons, icônes, typographies) permettant de construire rapidement des maquettes cohérentes. Il constitue un socle visuel précieux pour démarrer un projet et gagner du temps sur les premiers écrans.

Toutefois, **l'UI Kit montre vite ses limites** lorsque le produit évolue, que le nombre de composants croît, ou que plusieurs designers interviennent simultanément. Dans ces contextes, il devient difficile de maintenir la cohérence globale, de gérer les déclinaisons, et d'éviter les erreurs manuelles.

C'est dans ces situations que le Design System prend tout son sens. Au-delà d'une simple bibliothèque de composants, il offre un cadre structuré intégrant des règles d'usage, un système de nommage cohérent, des styles typographiques normalisés, des tokens de couleur, et une documentation complète partagée entre les équipes.

Un **Design System** bien pensé n'est pas uniquement un outil au service des designers. Il devient **un véritable levier de collaboration entre design, développement et produit**. En fournissant un **langage commun et des règles explicites**, il facilite l'alignement des pratiques et limite les allers-retours liés à des incompréhensions ou des approximations.

La mise en place d'un Design System ne se réduit pas à compiler des composants. Elle suppose un travail collectif, un effort de documentation rigoureux, et un engagement des équipes pour faire vivre et évoluer ce système au fil des besoins. Il doit être conçu pour ses utilisateurs : designers, développeurs, mais aussi product managers.

En résumé, le **Design System** apporte bien plus qu'une accélération des livraisons : il **structure la manière dont les équipes collaborent et conçoivent des produits** évolutifs et cohérents. S'il ne supprime pas les ajustements nécessaires en cours de route, il **offre une base commune qui fluidifie le travail et limite les imprévus**.

Summary note

In this white paper, we explore **the differences, complementarities and challenges** of a **UI Kit** and a **Design System**, drawing on concrete examples and feedback from real-life projects.

The Design System and the UI Kit are **two complementary** but **fundamentally different** tools. The UI Kit provides a set of essential graphic elements (buttons, icons, typography) for rapidly building coherent mock-ups. It provides an invaluable visual foundation for getting a project off the ground and saving time on the first few screens.

However, the **UI Kit quickly shows its limits** when the product evolves, when the number of components grows, or when several designers are involved simultaneously. In such contexts, it becomes difficult to maintain overall consistency, manage variations and avoid manual errors.

This is where the Design System comes into its own. More than just a library of components, it offers a structured framework incorporating usage rules, a consistent naming system, standardized typographic styles, color tokens, and complete documentation shared between teams.

A well thought-out **Design System** is not just a tool for designers. It becomes a **true lever for collaboration between design, development** and product. By providing a **common language and explicit rules**, it facilitates the alignment of practices and limits the back-and-forth caused by misunderstandings or approximations.

Setting up a Design System involves much more than simply compiling components. It requires a collective effort, rigorous documentation, and a commitment on the part of the teams to make the system live and evolve as needs arise. It must be designed for its users: designers, developers, but also product managers.

In short, the **Design System** does much more than speed up deliveries: it **structures the way in which teams collaborate and design evolving, coherent products**. While it doesn't eliminate the need to make adjustments along the way, it does **provide a common basis that makes work flow more smoothly and limits unforeseen events**.



Sylvie GAO

UX/UI Designer

Curieuse et active, j'aime apprendre et améliorer mes compétences en design. Lors de mes missions, j'aime mettre en avant vos valeurs et votre charte graphique. Je suis prête à m'impliquer au sein de votre équipe à l'occasion d'un **CDI à partir de janvier 2026**.

📞 07 81 48 93 96

🌐 linkedin

✉ sylvie.sgdesign@gmail.com

🌐 sylvie-gao.fr

FORMATIONS

◇ 2023 - 2025

MASTÈRE DIGITAL DESIGN

École Supérieure du Digital

Confronté à divers projets créatifs, j'apprends à maîtriser de nouveaux outils en peu de temps tout en produisant dans des délais restreints.

◇ 2020 - 2023

BACHELOR MANAGEMENT DE PROJET WEB

Epitech Digital School

J'ai acquis un esprit d'équipe et une méthodologie créative afin de mener à bien les différents projets.

◇ 2017 - 2020

BACCALAURÉAT GÉNÉRAL

Lycée Dorian

LOISIRS



Curieuse et ouverte au monde, j'aime découvrir de nouveaux paysages, une manière de me ressourcer et de m'inspirer.



Passionnée par la photographie, je prends plaisir à capturer la beauté du monde et à conserver des souvenirs précieux.



Attirée par les travaux manuels, j'apprécie ces moments de créativité qui m'aident à ralentir le rythme et à me recentrer.

EXPÉRIENCES

◇ JANVIER 2023 - SEPTEMBRE 2025

UX/UI DESIGNER

Stage et Alternance - Creativity

- Création de visuels pour les réseaux sociaux
- Création de maquettes web et mobile UX/UI
- Création de page sur Webflow
- Participation dans la création d'un design system
- Animation d'illustrations sur After Effects/Rive

◇ AVRIL - JUILLET 2022

GRAPHISTE

Stage - Institut Calebasse

- Création de visuels pour les réseaux sociaux
- Réalisation de supports de communication numérique et impression (flyer, brochure, poster, dossier de presse...)
- Mise en page et conception de template de newsletter
- Conception de maquettes web
- Création de pages web sur Wordpress avec Elementor

◇ JUIN-SEPTEMBRE 2021

WEBDESIGNER & GRAPHISTE

Stage - P'tit Bout d'Com

- Proposition d'axes d'amélioration aux différents sites internet entretenue par l'agence
- Création de visuels pour les réseaux sociaux
- Création de pages web sur Wordpress avec/sans Elementor

COMPÉTENCES

- Design Web / Mobile
- Wireframe / Maquette
- Benchmark design
- UX/UI Design
- Infographie
- Montage vidéo

OUTILS



Merci pour votre lecture !

 <https://sylvie-gao.fr/>

 <https://www.linkedin.com/in/sylvie-gao-a827a320a/>