

Le cahier des charges fonctionnel : les fonctionnalités clés et les spécifications

Table des matières

I. Spécifications dans le cahier des charges	3
A. Spécifications fonctionnelles et techniques	3
B. Comment sont utilisées les spécifications fonctionnelles	4
C. Dans le cahier des charges ou livrable séparé	6
D. Livrable à mettre à jour	8
II. Exercice : Quiz	9
III. Formaliser des spécifications fonctionnelles	10
A. Matérialisation des écrans	10
B. Définir les bons critères de description selon votre projet.....	12
C. Prendre en compte les cas d'erreur.....	15
D. Envisager l'administration du site.....	16
E. Réaliser des spécifications à l'aide d'Axure	18
IV. Exercice : Quiz	19
V. Essentiel	21
VI. Auto-évaluation	21

I. Spécifications dans le cahier des charges

Contexte

Le **cahier des charges** constitue une **référence tout au long du projet** : il en précise les **enjeux**, décrit la **solution envisagée**, sert de **référentiel d'exigences** durant toute la production et permet de **vérifier la conformité** des travaux à l'issue du projet. Il est donc **inévitables** qu'il contienne une **description précise de la plateforme digitale attendue**.

Les **détails** de cette réalisation sont portés par un document particulier appelé « **spécifications fonctionnelles** ». C'est en grande partie pour cela que votre cahier des charges est dit « **fonctionnel** » : il détaillera **les interactions possibles** entre les utilisateurs et l'application et indique **comment cette dernière doit fonctionner**.

Les spécifications doivent être **précises**, elles doivent **lever toutes les ambiguïtés**. Elles nécessitent aussi d'être **complètes**, pour **ne laisser transparaître aucun manque** dans la description des éléments attendus. Elles sont **la synthèse de tout votre travail de conception** et **déterminent les contours** de la commande passée par le porteur du projet.

Les spécifications sont donc **la pierre angulaire de votre cahier des charges fonctionnel**, et cette partie devra être **tout particulièrement soignée**. Dans ce cours, nous allons vous présenter ce que devraient être des spécifications fonctionnelles **utiles et efficaces** avant d'approfondir ensemble **la manière de les rédiger**.

Les spécifications constituent toujours un **référentiel d'exigences**. Quel que soit le contexte, elles **décrivent des règles à respecter**, des **processus** ou des **fonctionnements** attendus. Vous pourrez rencontrer des spécifications pour un produit électronique, une construction, etc. ou bien entendu **une réalisation digitale**.

Traditionnellement, vous serez confronté à **deux types de spécifications** dans un cahier des charges : les **spécifications fonctionnelles** et les **spécifications techniques**.

A. Spécifications fonctionnelles et techniques

Ces deux types de spécifications sont différents mais **complémentaires**.

Les **spécifications fonctionnelles** décrivent **comment** les différents écrans (de votre site ou de votre application mobile) seront réalisés. Elles **précisent leur structure** et détaillent également **les fonctionnalités attendues avec précision**. Elles n'ont pas vocation à **décrire à quoi vos écrans vont ressembler** : ça, se sera l'objet de vos **maquettes** ou de votre **charte graphique digitale**. Elles se concentrent **uniquement sur leur composition** : quels **types d'éléments** sont attendus à quel endroit, **sans présager de leur mise en forme**.

Elles ne détaillent **pas non-plus les contenus** qui seront utilisés, même si elles peuvent en **définir des caractéristiques** comme le nombre de caractères, le format, etc.

En revanche, elles **décrivent soigneusement comment tel ou tel élément de votre application devra fonctionner**.

Fondamental

Les spécifications fonctionnelles **décrivent et documentent la réalisation du point de vue de ses usages et de son fonctionnement pour l'utilisateur** et/ou l'administrateur.

Les **spécifications techniques** découlent en grande partie des spécifications fonctionnelles. Elles préciseront **comment** doit être développée la solution et les **modes de fonctionnement « techniques »** que l'utilisateur ne verra pas.

Elles peuvent indiquer les **langages** utilisés, préciser des **règles à suivre par les développeurs** dans leurs développements, le **fonctionnement d'API** (*Application Programming Interface* – Interface de programmation applicative) à utiliser, des **procédures informatiques automatiques** comme des **sauvegardes** et des **mise à jour**, indiquer des **flux de données** et leur **traitement** attendu, détailler des **procédures de mise en ligne**, de **synchronisation** des réalisations, de **tests**, etc.

Fondamental

Si les spécifications fonctionnelles décrivent ce qu'on va faire, **les spécifications techniques précisent comment on va le réaliser.**

Ce sont ces spécifications que vous allez retrouver en grande partie dans votre « **cahier des charges technique** ».

Rassurez-vous : en tant que chef de projet, si vous devez produire un cahier des charges technique, **vous ne serez concerné que par la rédaction de spécifications techniques générales.** Les spécifications techniques **détaillées**, elles, seront **nécessairement rédigées par un développeur** qui maîtrisera les notions et concepts présentés.

Ces deux types de spécifications peuvent se rencontrer dans un projet digital. **Les premières (fonctionnelles) décrivent le « quoi », les secondes (techniques) précisent le « comment ».**

B. Comment sont utilisées les spécifications fonctionnelles

Les **spécifications fonctionnelles** décrivent **la construction d'un écran** et **les fonctionnalités** qu'il contient. Le plus souvent, on se base sur les **wireframes** pour **présenter visuellement** les types d'écrans attendus. On présentera donc une **maquette fonctionnelle** assortie de **descriptions**.

Chaque élément est **identifié par un numéro de référence.** Dans un **tableau** à proximité de votre wireframe, vous **décrirez chacun des éléments.**



#	Fonctionnalité	Type	Interactions	Administrable BO
1	Logo	Image	Au clic, renvoi à la page d'accueil	Oui, téléchargement
2	Menu de navigation	Texte	Liste de liens renvoyant vers les pages de tête de rubrique	Non, automatique
3	Moteur de recherche	Formulaire	Déploiement au clic Validation par touche « Entrée »	Non
4	Zone d'accueil vidéo	Vidéo	Autoplay, boucle	Oui, téléchargement
5	Zone de titre animée	Texte	-	Oui, saisie texte
6	Sous-titre (optionnel)	Texte	-	Oui, saisie texte
7	Sone actualité	Mixte	-	Non, automatique
8	Titre de zone	Texte	-	Oui, zone de saisie
9	Bloc actualité	Image + Texte	Au clic, lien vers la page actualité correspondante	Oui, zone de saisie + téléchargement visuel
10	Zone témoignages	Mixte	Défilement automatique de la gauche vers la droite toutes les 3 secondes	Non, automatique
11	Titre de zone	Texte	-	Oui, saisie texte
12	Photo utilisateur	Image	Au survol, le visuel grandit	Oui, téléchargement
13	Navigation	Bouton	Au clic sur le pictogramme, défilement des visuels vers la gauche ou la droite	Non

Sur le principe, rien de plus simple. Toute la subtilité sera dans **la manière dont vous allez rédiger ces descriptions.** N'oubliez pas : il faudra être **précis et exhaustif.**

Exemple

En travaillant sur le projet *Trot'Bidulle*, le **site de vente en ligne de trottinettes électriques**, votre collègue **Martin** a décrit de cette manière la **fonctionnalité de recherche du blog** :

« *Champ de recherche permettant de trouver des contenus sur le site.* »

C'est **sommaire**, mais il décrit bien un moteur de recherche... Mais **nous manquons là cruellement de précision**. Quel **type de contenu** doit être recherché ? Dans **quelles sections** ? Que se passe-t-il **ensuite** ?, etc.

Si on voulait être plus précis, on pourrait reprendre la proposition de Martin de cette façon :

« **Champ de recherche plein texte** permettant de **retrouver des billets** contenant les **termes indiqués** parmi **l'ensemble des publications** sur le blog. **Lors de la saisie**, les **titres des billets** comportant les mots recherchés **sont proposés**. Un **clic** sur une **suggestion d'article** renvoie l'utilisateur **vers le billet concerné**. Lorsque l'utilisateur **valide sa recherche par la touche « Entrée »**, il est dirigé vers la **page de résultats**. »

C'est plus long, certes... **Mais beaucoup plus précis**. Les designers, comme les développeurs, **comprendront mieux** le fonctionnement attendu et ce qu'ils doivent réaliser.

Les **spécifications fonctionnelles** sont utilisées dans **trois séquences clés** du cycle projet :

En premier lieu, **au tout début du projet**, elles permettent de **décrire les attendus**, de **figer les exigences**. Votre commanditaire, qu'il soit interne ou externe à votre organisation, **devra valider les spécifications fonctionnelles**. On aura ainsi « **verrouillé** » les attendus.

Votre périmètre fonctionnel peut indiquer que le site devra contenir un **moteur de recherche**. Cette fonctionnalité pourra très bien être **imaginée de façon très différente** par vous, vos équipes et surtout **votre commanditaire**.

Les spécifications décriront **où se trouve** ce moteur de recherche, **comment il fonctionnera**, quels **types de résultats** il renverra, etc.

Vous aurez ainsi précisé « ce qui est prévu ».

En validant les spécifications fonctionnelles, votre client et vous **vous mettez d'accord** sur **ce qui est prévu** dans le périmètre de réalisation. Vous **supprimez les éventuelles divergences de points de vue** au moment de la livraison.

Fondamental

Les spécifications fonctionnelles **permettent de définir précisément le périmètre de réalisation** et de **dissiper les divergences de point de vue** sur les réalisations à produire. Elles **définissent un point d'accord** sur « **ce qu'il y a à faire** ».

Exemple

Imaginez que vous allez dîner dans une **pizzeria**. Parmi les pizzas proposées, vous choisissez la « *reine* ». La carte indique qu'**elle contiendra une base tomate, de la mozzarella, des champignons de Paris et du jambon blanc**. Vous demandez au serveur d'y ajouter **un œuf**, et précisez que vous ne la désirez « **pas trop cuite** ».

L'ensemble de ces éléments constitue votre commande. **Ce sont les spécifications** dont vous convenez avec le serveur.

Pendant la réalisation, les **équipes** auront besoin de **savoir ce qui est envisagé**. Les spécifications **guideront tout autant les créatifs**, qui ont besoin de savoir ce qui doit être présent sur chaque type d'écran, **que les développeurs** qui doivent **assurer le fonctionnement** attendu de la solution.

Les spécifications doivent donc **donner suffisamment d'informations** pour qu'à leur lecture, **les ressources n'aient pas à revenir vers vous** avec l'éternelle question : « *ce truc, là, comment c'est censé marcher ?* ».

Notons au passage que ce serait déjà un bon réflexe de venir vous demander.

Bien souvent, un fonctionnement attendu **peut être compris de façon très intuitive...** Mais **pas toujours comme il était prévu dans l'esprit du concepteur.**

Les spécifications détaillent donc **comment** doit fonctionner l'élément en question.

Fondamental

Les spécifications fonctionnelles **détaillent sans ambiguïté les fonctionnements attendus.** Elles servent de **guide de réalisation pendant les phases de production.** Toutes les ressources productives vont **s'y référer.** Elles servent de **référence** pour développer **les interactions attendues.**

Exemple

Dans la **pizzeria**, le pizzaiolo va **se référer au ticket de commande** remis par le serveur pour **savoir quelle pizza** il doit préparer. Il prend note de vos **demandes spéciales.** Comme il est **débutant**, il va se référer également **à la carte** pour savoir précisément ce que contient une pizza « *reine* ».

Par cette démarche, **il consulte les « spécifications »** qui vont lui indiquer **comment** il devra réaliser la pizza que vous avez commandée.

Le projet **se termine par une phase de recette.** C'est-à-dire une séquence pendant laquelle vous, dans un premier temps, puis le commanditaire du projet ensuite, **réceptionneront le résultat des développements.** Cette recette peut avoir lieu **de nombreux mois après la définition du projet.**

On se base alors sur **les documents de description** pour **évaluer la conformité** de la livraison avec les éléments attendus, **convenus en début de projet.**

On se réfère donc aux spécifications fonctionnelles pour **vérifier** que ce qu'on remet au client est bien **conforme** à ce qu'il avait commandé.

Fondamental

Les spécifications fonctionnelles servent de **support à la vérification de conformité de la réalisation livrée.**

Exemple

Avant d'apporter votre plat jusqu'à votre table, **le serveur va jeter un œil au ticket de commande.** Il pourra ainsi **vérifier que ce qu'il s'apprête à servir correspond bien à la commande.**

C'est une **phase de recette**, qui se base sur les **spécifications prédéfinies.**

Vos spécifications fonctionnelles constituent donc un **document de référence depuis la définition** du projet, **jusqu'à sa livraison.**

C. Dans le cahier des charges ou livrable séparé

Vous l'avez compris, **les spécifications fonctionnelles sont souvent un document conséquent**, qui va vous demander **une charge de travail importante** que vous ne devriez **surtout pas négliger.** Mais il arrive qu'elles ne soient **pas rédigées au moment où vous devrez formaliser votre cahier des charges.** Vous pouvez aussi envisager que les spécifications ne soient pas portées par un document rédigé mais par un **prototype commenté**, qui sera **un des livrables de votre projet.**

Dans ce cas, elles apparaîtront **dans la liste des livrables attendus** en cours de projet. Vous en préciserez **la période de rédaction**, et, dans votre **planning**, vous indiquerez les **dates de livraison et de validation** par le commanditaire.

Dans la mesure où elles **définissent en détails les attendus**, vous aurez tout intérêt à **conditionner les développements à la validation** de ces spécifications. En tous cas, dans une approche projet descendante traditionnelle.

La rédaction de ce document représentant **une charge importante**, elle est parfois considérée (à tort) comme **superflue**. Et cela est alimenté par **deux constats** que vous pourrez faire facilement lors de votre carrière :

- **Le cahier des charges est rarement lu en entier** par le commanditaire, comme par les équipes de réalisation,
- **Les spécifications fonctionnelles ne sont que survolées par le commanditaire** avant qu'il les valide.

Et c'est souvent vrai, et pour plein de bonnes raisons. **Dans un déroulement projet sans accroc**, cela ne sera **pas problématique**. En revanche, lorsqu'**au moment de la réception du projet**, des **désaccords** surviendront **sur la manière dont telle ou telle fonctionnalité avait été envisagée** par vous et par votre commanditaire, pouvoir **se reporter à une documentation de référence** permettra de **désamorcer des situations qui pourraient vite devenir conflictuelles**.

Même si vos spécifications font l'objet d'un livrable en cours de projet, même si vous optez pour **un format moins rédigé** que le formalisme classique wireframes + tableaux détaillés, prenez soin de **bien constituer un référentiel fonctionnel**.

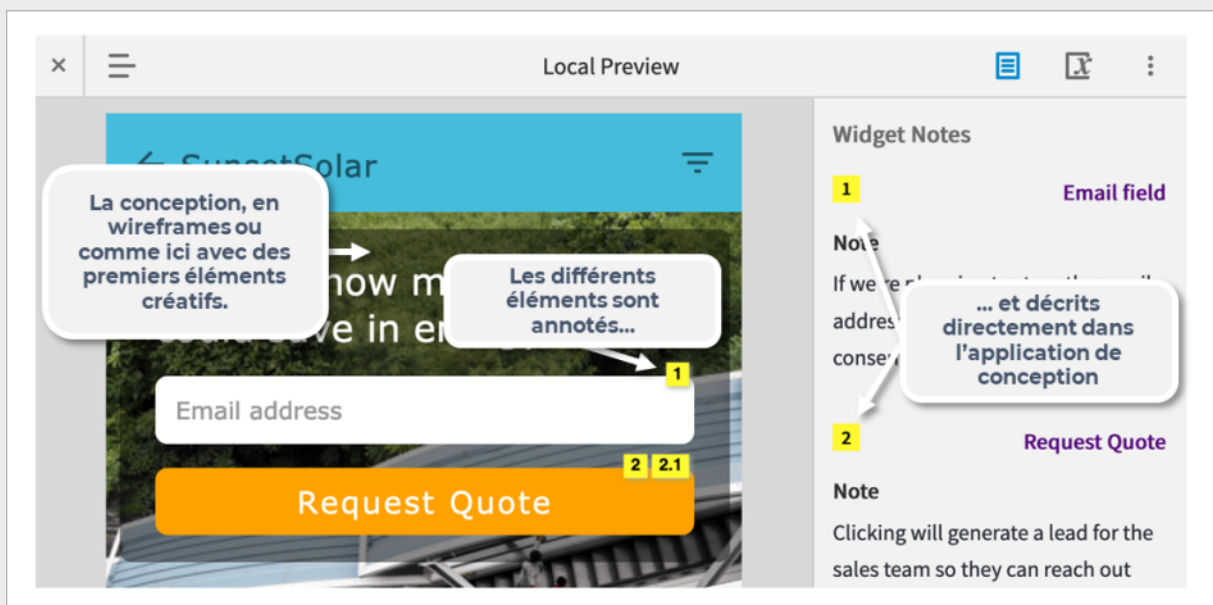
Un **prototype commenté** pourra facilement faire l'affaire, mais **ne faites jamais l'impasse sur ce document de référence**.

Exemple

Les **logiciels de conception**, en particulier lorsqu'ils **intègrent un fonctionnement collaboratif**, vous permettent de **documenter vos conceptions** et d'**éviter la rédaction de spécifications** fastidieuses à produire comme à consulter. Cela bien évidemment **seulement** si la fourniture de spécifications en bonne et due forme n'est **pas requise** par votre commanditaire.

Voici un exemple issu de la documentation en ligne du **logiciel de conception Axure**.

Chaque élément présent dans votre conception pourra être **décrit avec précision**. Le simple fait de commenter **génère l'apparition d'une note numérotée** automatique, à laquelle **n'importe quel acteur** consultant le prototype pourra se référer.



Un **prototype fonctionnel**, ainsi **documenté**, permettra facilement de **se projeter par la démonstration** dans le fonctionnement attendu. Les **précisions utiles** pourront être **ajoutées en commentaires**.

Complément

Si les spécifications sont **généralement basées sur des wireframes** (car on a souvent besoin de **connaître le fonctionnement des interactions** proposées pour réaliser un design adapté), il se peut que vous ayez **déjà les créations à votre disposition**. Si tel est le cas, afin de **faciliter la compréhension** de tous, et en particulier de votre commanditaire, vous pourrez tout-à-fait **réaliser vos spécifications fonctionnelles sur la base des créations finalisées**.

D. Livrable à mettre à jour

Les spécifications fonctionnelles sont le document qui **décrit comment fonctionne votre produit**. Elles en constituent **donc l'essentiel de la documentation**. Ce sera d'autant plus important si votre **contrat de prestation** prévoit **la fourniture de cette documentation à votre commanditaire**.

Il est essentiel que cette documentation reste à jour. Pour de multiples raisons, vous pourrez avoir à vous y référer **à plus ou moins longue échéance**.

Or, en cours de projet, il **est fréquent que le besoin évolue** ou soit précisé par le commanditaire. Des impératifs de livraison pourront amener à **des arbitrages** qui conduiront à **l'abandon de fonctionnalités**. Au contraire, des **tests utilisateurs** pourront inciter à **développer des interactions complémentaires** ou à **modifier** des fonctionnements établis... Bref, **plusieurs détails de votre projet seront modifiés avec le temps**.

Votre documentation, et donc **vos spécifications fonctionnelles, devront refléter ces évolutions**. N'oubliez pas qu'elles **serviront de référentiel au moment de la livraison** et toute différence pourra entraîner une **déclaration de non-conformité** de votre réalisation. Il vous faudra alors **reconstituer l'historique de vos échanges**, rassembler les prises de décision pour **démontrer** que vous avez bien produit ce qui, finalement, a été convenu.

Par ailleurs, **une fois votre projet terminé**, votre réalisation entrera dans sa **phase de vie**. Probablement qu'avec le temps, et en particulier **si vous êtes chargé de la maintenance de la solution**, vous serez amené à **la faire évoluer**. Il ne s'agit pas de refonte, mais de **petites mises à jour** ou de **simples optimisations** qui vont se cumuler.

Si dans vos équipes vos développeurs, par exemple, ont changé et doivent intervenir sur la solution **ils auront besoin de se référer à la documentation** pour savoir comment chaque élément doit fonctionner. Si elle n'a **pas été mise à jour**, ils seront **incapables**, par **manque d'historique**, de savoir si le **comportement attendu** est celui constatable sur la solution en place ou celui décrit dans les spécifications.

Exemple

Après trois années de collaboration avec votre commanditaire, vous avez **largement fait évoluer son application mobile**. De **légères modifications** ont été implémentées au fil du temps de façon à **améliorer les parcours utilisateurs ou la conversion**. À l'issue d'une réunion de suivi avec votre client, vous apprenez qu'une **consultation va être organisée** pour un **vaste projet de refonte** de l'application existante.

Comme il s'agit d'une refonte, votre client souhaite **joindre aux documents transmis aux soumissionnaires les spécifications fonctionnelles et techniques** de la version actuellement exploitée. Il s'agit de **tenir compte de l'existant** pour ne pas tout redévelopper. Cette documentation doit être **partagée** pour que **tous les répondants** à l'appel d'offre soient sur **un pied d'égalité**.

Votre client vous demande donc de lui **fournir les spécifications à jour**, ce qui devrait être rapide puisqu'elles **font partie des éléments à fournir prévus dans votre contrat de prestation**.

Deux situations sont envisageables :

- **Vous vous êtes assuré des mises à jour régulières des spécifications** après chaque intervention mineure, cela vous a demandé **un effort infime** à chaque fois, et vous pouvez **répondre à la demande du client dans l'heure**, en regrettant simplement une nouvelle mise en compétition après trois années de collaboration paisible...

- Vous avez **concentré vos efforts sur l'action, au détriment de la documentation**. Vous allez donc devoir **mobiliser vos équipes** pour **produire a posteriori** des spécifications fonctionnelles correspondant à l'état actuel de l'application. Ce travail risque d'être **très fastidieux** et nécessitera **une bonne semaine de travail** (non facturable, etc.), pour fournir la documentation due à votre client.

Pour toutes ces raisons, vous aurez tout intérêt à produire vos spécifications fonctionnelles à l'aide d'**un outil permettant des mises à jour faciles**. Ce sera le cas pour vos **prototypes commentés**, mais également si vous utilisez **un outil de conception comme Axure** qui vous propose de **produire vos spécifications fonctionnelles en deux clics** de souris, directement **à partir de votre prototype**.

Vous le comprenez, **les spécifications fonctionnelles sont au cœur de votre documentation projet**. Elles sont **essentielles** tant pour en **définir précisément les attendus** que pour **servir de socle à la validation** de vos réalisations. Cela en fait un **contenu clé** de votre cahier des charges fonctionnel.

Exercice : Quiz

Question 1

Voici plusieurs propositions à propos de l'utilité des spécifications fonctionnelles dans un cahier des charges. Indiquez celles qui sont exactes : (Plusieurs réponses possibles)

- ☐ Les spécifications fonctionnelles documentent la structure des écrans de la future application.
- ☐ Les spécifications fonctionnelles décrivent le fonctionnement des fonctionnalités proposées.
- ☐ Les spécifications fonctionnelles présentent le design graphique.
- ☐ Les spécifications fonctionnelles précisent les langages informatiques à utiliser.

Question 2

Quelle est **la façon la plus efficace de présenter des spécifications fonctionnelles** dans un cahier des charges ? (Une seule réponse juste)

- ☐ En rédigeant un texte concis pour chaque fonctionnalité attendue, par exemple dans un tableau Excel.
- ☐ En ajoutant des commentaires rapides, fléchés sur des maquettes graphiques.
- ☐ En formalisant un tableau décrivant chaque fonctionnalité de façon détaillée, associé à un wireframe numéroté.

Question 3

Pourquoi les **spécifications fonctionnelles** sont-elles très utiles **en début de projet**, et donc dans le cahier des charges fonctionnel ? (Plusieurs réponses possibles)

- ☐ Elles permettent de verrouiller le périmètre projet en définissant un point d'accord sur ce qu'il y a à faire.
- ☐ Elles offrent une perception unifiée des parcours des utilisateurs clés, essentielle en début de projet.
- ☐ Elles permettent de lever les ambiguïtés et les incompréhensions possibles avec le commanditaire sur les fonctionnalités attendues.
- ☐ Elles donnent toutes les informations utiles aux concepteurs pour imaginer le design fonctionnel.

Question 4

« Les **spécifications fonctionnelles** doivent **obligatoirement** figurer dans un cahier des charges. » **Vrai ou faux ?**
(Une seule réponse juste)

- ☐ Vrai
- ☐ Faux

Question 5

Voici quelques affirmations concernant **les spécifications fonctionnelles**. Indiquez celles qui sont **fausses** :
(Plusieurs réponses possibles)

- ☐ Les spécifications fonctionnelles constituent l'essentiel de la documentation du projet ; elles peuvent donc être mises à jour pendant le déroulement du projet.
- ☐ En tant que document de référence, les spécifications fonctionnelles pourront être mises à jour après la fin du projet.
- ☐ Les spécifications fonctionnelles sont un livrable de référence permettant de sécuriser le périmètre projet. Elles empêchent donc toute évolution du besoin en cours de projet.
- ☐ Les spécifications fonctionnelles seront très utiles durant les phases de maintenance de la solution issue du projet.
- ☐ Si vous avez la charge de la maintenance de la solution issue du projet, il ne sera pas nécessaire de mettre à jour les spécifications fonctionnelles pour des évolutions mineures.

III. Formaliser des spécifications fonctionnelles

Dès lors que vos spécifications **doivent être intégrées au cahier des charges** (ou lorsqu'elles seront détaillées dans **un livrable rédigé spécifique**), vous devrez **respecter un certain formalisme** pour vous assurer que votre document remplira bien ses fonctions. Cela passera par **une bonne matérialisation des écrans** présentés, des **critères de description adaptés** à votre environnement projet, et le **détail des cas d'erreurs** envisageables.

A. Matérialisation des écrans

Dans la **structure de votre cahier des charges**, la description des différents écrans viendra **après la présentation globale de votre environnement** projet. Votre lecteur aura normalement déjà à ce stade **une vue complète de la solution** que vous envisagez. Vos **spécifications** apporteront alors **des précisions sur la manière dont seront mises en scène les fonctionnalités attendues**. La matérialisation des écrans aura donc toute son importance.

Si vous utilisez des **wireframes**, il serait préférable de présenter **des maquettes fonctionnelles détaillées**, puisque vous allez en préciser **les détails pour chaque fonctionnalité**. Les simples croquis, s'ils peuvent permettre une compréhension globale de la réalisation attendue, ne seront pas assez précis pour porter vos descriptions sans générer de confusion.

Exemple

Voici **un exemple de wireframe** susceptible d'accueillir des **spécifications fonctionnelles détaillées**. Le **niveau de définition est suffisamment élevé** pour présenter **en situation** l'ensemble des précisions attendues. Il est ainsi plus **facile de se projeter dans la future réalisation**, avant même d'en avoir travaillé la création graphique.

[FAQ](#) | [Mentions légales](#) | [Conditions Générales d'Utilisation](#) | [Crédits](#) | © MonProjet, 2020

Cette construction des pages ne **doit pas être considérée trop strictement**. Il se peut qu'au moment de produire la création, vous vous aperceviez que **des ajustements sont nécessaires**. S'ils sont **importants**, vous **adapterez vos wireframes** pour les faire correspondre aux écrans définitifs et **préserver une documentation projet utilisable**.

Vous pourriez également **utiliser vos créations finales** comme **socle de vos spécifications**. L'important, c'est d'**afficher un niveau de détail suffisant** pour présenter vos descriptions **sans ambiguïté**.

B. Définir les bons critères de description selon votre projet

Dans vos spécifications fonctionnelles, vous devrez **décrire l'ensemble des fonctionnalités attendues**. **L'exhaustivité** est essentielle, mais il vous faudra dans le même temps **éviter les redondances**. En effet plusieurs fonctionnalités peuvent apparaître dans différents écrans de votre réalisation. Vous chercherez à **ne les décrire qu'une seule fois**. L'objectif est triple :

- **Limiter le volume du document** qui selon votre projet peut vite devenir important,
- **Faciliter l'accès aux informations** pertinentes,
- **Simplifier les mises à jour** de vos spécifications.

Si un composant est présent sur plusieurs écrans, **vous ne le décrirez qu'à un seul endroit, facilement identifiable**. Ce sera évidemment le cas pour les **éléments récurrents** comme par exemple un **en-tête**, un **pied de page**, un **module de chat bot** ou d'assistance présent en permanence. Mais vous pourrez également appliquer ce principe pour **des blocs fonctionnels** envisagés sur plusieurs écrans différents, comme un **module d'abonnement à une newsletter** par exemple.

Exemple

Dans un outil de conception comme **Axure**, vous aurez la possibilité de travailler avec des « **masters** ». Ce sont des **blocs fonctionnels** que vous pourrez **réutiliser à n'importe quel endroit de votre conception**. Les **informations de description** qui y sont rattachées ne seront alors **présentées qu'une seule fois** dans votre document de spécification et pourront **facilement être mis à jour**.

En outre, si vous avez besoin de **modifier ces masters**, sachez que **les changements seront automatiquement reportés sur l'ensemble des écrans** qui y font appel, ce qui vous fera **gagner un temps précieux**.

Si vous réalisez vos spécifications de bout en bout sans recourir à un logiciel de conception, vous pourrez alors **isoler ces descriptions dans une section** intitulée « *fonctionnalités récurrentes* » par exemple. Si vous avez au préalable réalisé un **périmètre fonctionnel**, vous avez normalement **déjà identifié ces fonctionnalités transversales**.

Veillez toutefois à **la clarté de votre document** : il est essentiel d'en **faciliter la consultation** régulière. Utilisez **un système clair de renvois**. Les fonctionnalités de votre traitement de texte pourront vous faciliter la tâche pour cela.

Il vous faudra maintenant **détailler chaque élément avec soin**. C'est un travail qui est plus simple à réaliser si vous décrivez vos fonctionnalités **au moment de la conception**. Reprendre dans un second temps l'ensemble de vos écrans pour les commenter peut en effet s'avérer assez long et pénible.

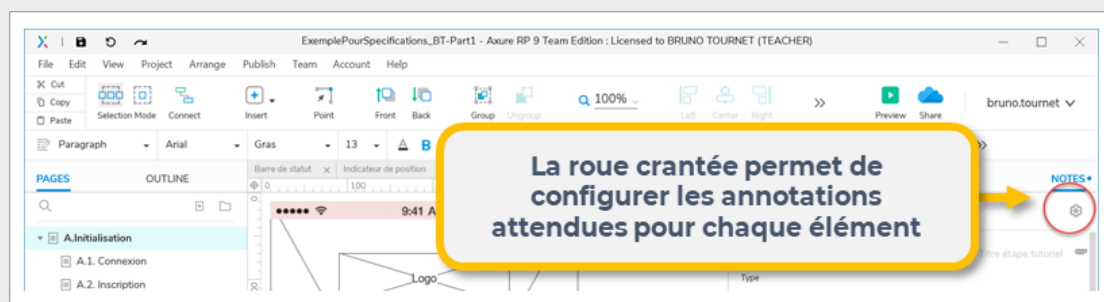
Votre **mode de description** dépendra de votre organisation et de vos **besoins spécifiques**. Mais **quelques informations sont structurantes**, quel que soit votre environnement. Vous retrouverez ainsi pour chaque fonctionnalité :

- **Un nom**, court et explicite, permettant de l'identifier facilement,
- **Un type**, si vous avez besoin de catégoriser les éléments décrits,
- **Une description**, plus longue, pour décrire le comportement attendu,
- **Une cible**, c'est-à-dire une destination, si l'utilisateur est renvoyé vers une autre page ou un autre écran,
- **La localisation des fichiers sources** éventuellement nécessaires pour vos développeurs.

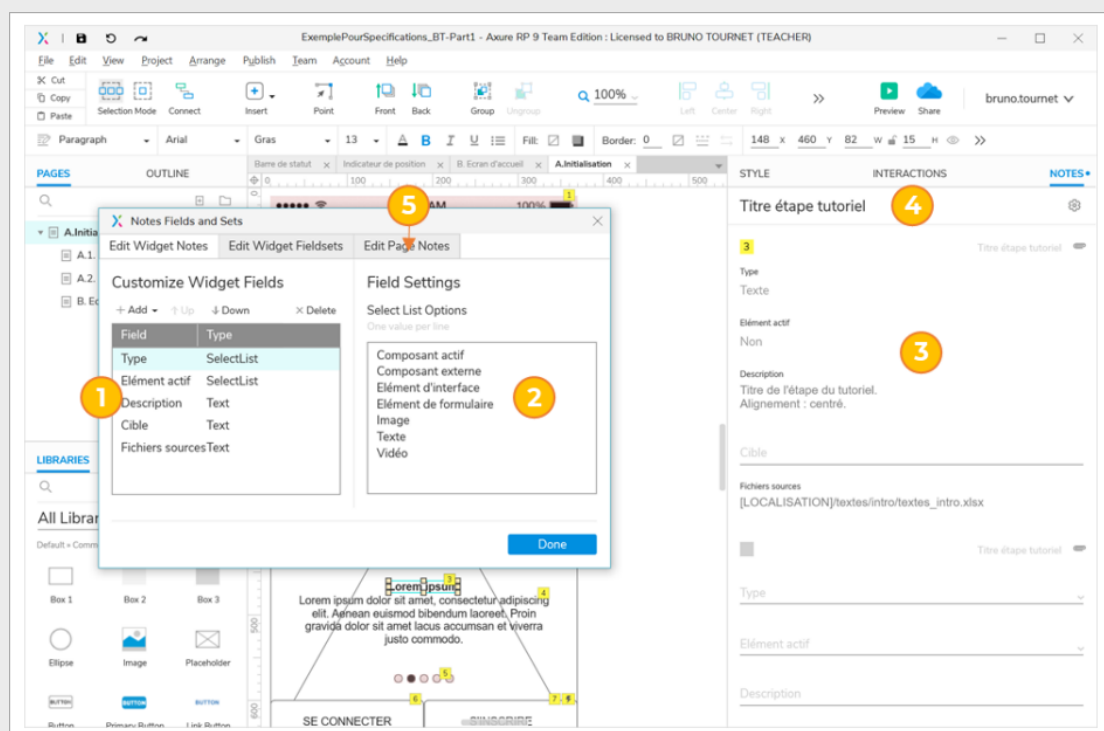
Toutes ces informations seront **précieuses** dans vos spécifications.

Exemple

Dans le logiciel **Axure**, vous avez la possibilité de **systématiser les modes de description** utilisés. Cela se paramètre grâce à **l'onglet note**, dans la partie droite de l'interface.



Vous pouvez **définir les champs attendus** pour la **description de chaque élément** de votre conception dans la fenêtre d'édition. **Chaque champ** vous sera alors proposé pour **tous les composants** que vous utiliserez.



1. Vous pouvez **ajouter autant de champs que nécessaire** dans la zone « *Customize Widgets Fields* » (personnaliser les champs de composant). On retrouve ici les **catégories** citées précédemment : **type**, **description**, **cible** et **fichiers sources**. Chaque champ peut être un **texte libre**, une **liste de sélection** ou un **nombre**.
 2. Pour les **listes de sélection**, vous pouvez déterminer les options proposées dans la zone « *Fields settings* » (paramétrage des champs). Elles sont utiles car vous n'aurez qu'à **pointer l'item pertinent** sans avoir à le réécrire. Dans cet exemple, on considère que les éléments de l'interface peuvent être :
 - Des **composants actifs**, c'est-à-dire proposant une interactivité,
 - Des **composants externes**, comme une carte Google map par exemple,
 - Des **éléments d'interface** nécessitant une description ,
 - Des **éléments de formulaire**,
 - Ou des médias dont on précise la nature (**image**, **texte** ou **vidéo**).
 3. Ces choix apparaîtront **sous forme de menu déroulant** dans cette **zone d'annotation**, comme l'ensemble des autres champs prévus. Vous pourrez alors **facilement commenter** tout élément que vous ajouterez dans votre conception.
 4. Même sans définir de champs spécifiques, la zone de notes vous proposera **un champ pour nommer l'élément sélectionné** et une zone d'annotation par défaut.
 5. Vous pouvez remarquer la présence de l'**onglet « Edit Page Notes »** (éditer les notes de page). Il fonctionne de la même manière que les champs de composants mais il vous propose de **définir des champs de description particuliers** qui s'appliqueront **à l'écran dans son ensemble**. En effet, il est souvent utile de **décrire globalement un écran** dans vos spécifications avant d'entrer dans le détail des items qui le composent.
- Si vous n'utilisez pas Axure, vous pourrez **reporter ces éléments de description dans un simple tableau** dans lequel vous décrierez chaque élément **en respectant le même formalisme**.

Vos descriptions doivent être précises pour indiquer **comment** vous envisagez **le comportement** de chaque élément. Imaginez **l'écran d'ouverture d'une application mobile**. Vous souhaitez afficher **une vidéo en arrière-plan** de votre procédure de découverte, lancée à la première ouverture de l'application. Votre description ne pourra être une simple « **vidéo en arrière-plan** ». Vous devez **détailler son comportement et les interactions attendues**.

Exemple

Voici comment vous pourriez rédiger la spécification de votre vidéo d'arrière-plan :

« Vidéo de présentation de l'application (**silencieuse**). La vidéo **se lance automatiquement et boucle**.

Les **contrôles vidéo** classiques **ne sont pas accessibles**.

Au **toucher-glisser** sur toute la surface de la vidéo, l'utilisateur peut **faire défiler les étapes du tutoriel**.

En fin de série (à droite comme à gauche) le **défilement est bloqué** et signifié par un effet de **rebond**.

L'arrière-plan vidéo et le logo **restent fixe**, seules **les étapes texte** du tutoriel (notes 4 et 5) défilent en lien avec les **indicateurs de position** (note 6). »

Même des éléments qui ne sont pas interactifs doivent être spécifiés. Par exemple, vous pourriez très bien décider que votre page d'accueil présentera **les derniers commentaires client**. Vous aurez défini une **mise en forme type** pour chaque commentaire, reprenant en vignette **l'avatar** du client, son évaluation sous forme d'étoiles, les **premières lignes de son commentaire** et son **prénom**.

Ce bloc sera **reproduit plusieurs fois** avec des informations différentes pour chaque commentaire. **Vos spécifications détailleront ce motif de répétition** : combien de commentaires doivent être affichés simultanément, comment s'animent-ils, se renouvèlent-ils, à quelle fréquence, etc. Même si **l'utilisateur n'interagit pas directement** sur ces éléments, vous devrez **préciser comment ils doivent être générés**.

Bien entendu, **le niveau de détails** de vos spécifications devra être **adapté à vos besoins**.

C. Prendre en compte les cas d'erreur

Tous les éléments actifs de votre solution ne pourront **pas toujours fonctionner idéalement**. Vous devrez prévoir ce qu'on appelle des « **cas d'erreur** ». Selon les fonctionnalités considérées, ils peuvent être très différents.

Vous aurez à décrire les **cas d'erreur « généraux »** qui **s'appliquent à votre application en général**.

Exemple

Dans vos spécifications, vous devrez **préciser ce qui doit se passer** par exemple lorsque l'utilisateur **demande une ressource qui n'existe pas** (page 404), lorsqu'il **tente d'accéder à un contenu** auquel il n'a **pas les droits d'accès**, ou lorsque votre application se trouve dans une situation où l'utilisateur **n'a pas de réseau**.

Vous aurez aussi à envisager **les cas d'erreur dans l'utilisation de certaines fonctionnalités**. Souvent, cela s'appliquera à des **formulaires**.

Exemple

Ces erreurs apparaissent par exemple lorsque votre utilisateur **ne saisit pas les types d'informations attendus** dans un formulaire : un signe « @ » oublié dans l'adresse e-mail, un **mauvais format** de numéro de téléphone, deux champs de mots de passe **incohérents**, etc. Ou lorsqu'il **omet** de compléter un champ requis.

Vous devrez alors **spécifier comment votre application va se comporter** : quels **tests** allez-vous effectuer sur ces données, comment vous **identifiez visuellement** les champs erronés, quels **messages explicites** vous allez afficher, etc.

Voici un exemple de cas d'erreur. Ces spécifications ne concernent par une fonctionnalité en particulier mais **sont détaillées au niveau de l'écran** dans son ensemble :



Dans cet exemple, on traite le cas d'une **déconnexion de compte** ou d'une **nouvelle installation** de l'application pour un utilisateur enregistré. Ce sont des « **cas d'erreur** » car ils décrivent une **utilisation non régulière** de l'application.

Préciser ces cas d'erreur présente l'avantage de **ne pas laisser aux développeurs le soin d'imaginer des parades** à toutes les erreurs possibles. Les indiquer dans vos spécifications fonctionnelles permettra également à vos **équipes créatives** de **ne pas oublier la mise en forme** de ce type de messages.

Les cas d'erreur peuvent également concerner des **comportements automatiques** non liés à des interactions utilisateurs. Si nous reprenons l'exemple des **motifs de répétition** cités précédemment, vous devrez envisager ce qui doit se passer **si aucun commentaire n'est encore publié**, si vous décider de les afficher par 3 mais que **seulement 4 sont disponibles**, etc.

D. Envisager l'administration du site

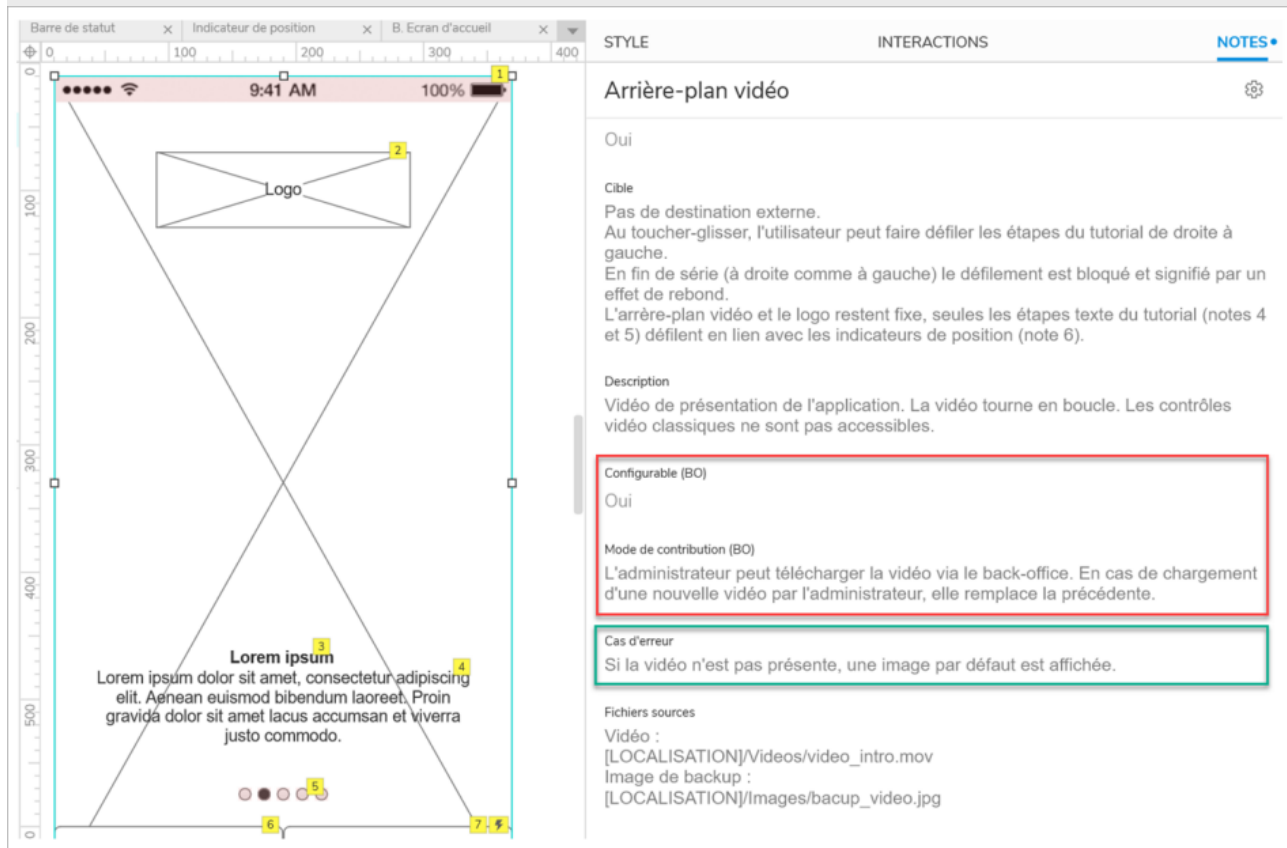
Toujours **dans une logique d'exhaustivité**, il sera très utile de décrire dans vos spécifications **comment les différents éléments de votre application pourront être gérés par l'administrateur**.

Un contenu texte, un logo, une image, un document à télécharger, etc. Tous ces éléments constitutifs de votre réalisation ne sortiront pas de nulle part : ils seront soit **intégrés « en dur »**, c'est-à-dire **par les équipes de développement**, soit **administrables**. Cela signifie que **l'administrateur** du site **pourra les modifier ou les ajouter par lui-même**, via son interface dédiée. Dans ce cas, vos spécifications devront préciser **les types d'informations attendues**, et éventuellement **les types de fichiers** acceptés.

Exemple

Vous pourrez avantageusement **compléter votre grille type de spécification**. Que ce soit dans votre logiciel de conception ou dans vos tableaux de description, pensez bien à **intégrer ces informations liées à l'administration de chaque élément**.

Voici un exemple de comment **détailler la gestion d'un élément en back-office**, ici encadré en rouge :



Nous indiquons ici que la vidéo utilisée en fond est **administrable en Back-Office (BO)**. Il suffit à l'administrateur d'en **télécharger une nouvelle** pour qu'elle **remplace** la vidéo existante.

Vous noterez également la **gestion d'un cas d'erreur**, ici encadrée en vert : **si aucune vidéo n'est présente**, alors l'application devra afficher **une image de remplacement**, dont nous avons précisé le chemin d'accès dans le champ « *fichiers sources* ».

Nous aurions même pu être **plus précis** en indiquant que cette image de remplacement s'affichera également **si le format de la vidéo téléchargé n'est pas pris en charge**, ceci afin d'**éviter la perception d'un dysfonctionnement** côté utilisateur.

Si vous avez bien déterminé **les utilisateurs cibles** de votre produit, vous avez normalement défini au moins **un persona administrateur**. Celui est **porteur de besoins spécifiques** qui se traduisent par des **parcours très particuliers**.

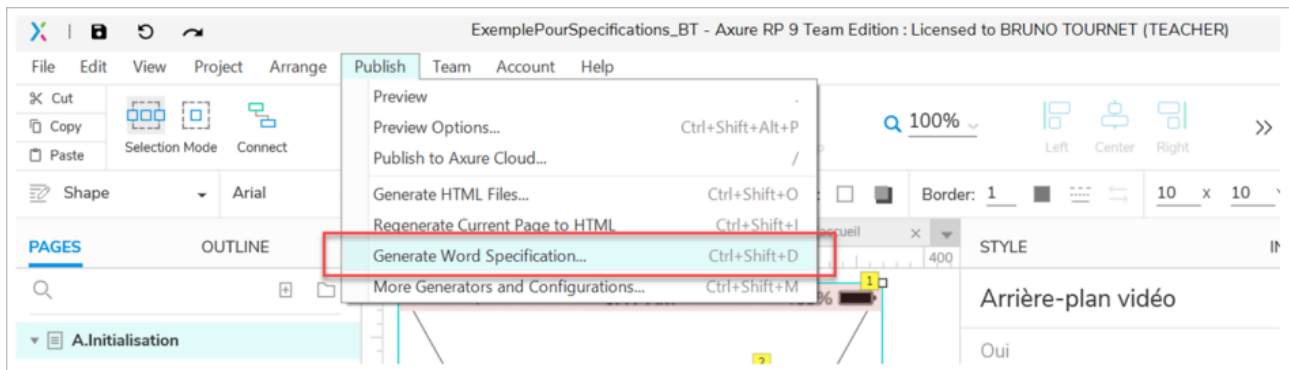
Il se peut que le fonctionnement par défaut de votre **interface d'administration** (le back-office) ne soit **pas suffisant** et **nécessite des développements spécifiques**. N'oubliez d'**inclure dans vos spécifications** la présentation de ces écrans et des **fonctionnalités propres** envisagées.

Tout comme votre projet comporte une facette « *publique* » et une autre destinée à votre commanditaire, **vos spécifications fonctionnelles ne peuvent se borner à décrire l'interface utilisateur**. Elles devraient aussi **intégrer la présentation des fonctionnalités attendues en back-office**.

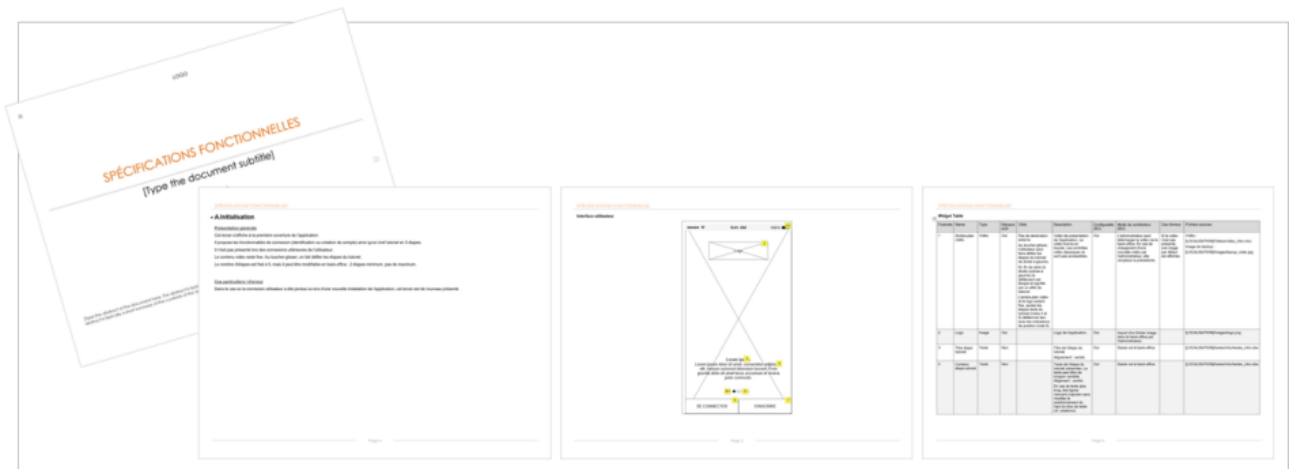
E. Réaliser des spécifications à l'aide d'Axure

Dans ce cours, si vous ne le connaissiez pas encore, vous avez pu découvrir **Axure** au travers des exemples présentés. Il existe bien entendu **de nombreux outils de conception**, et vous pouvez tout-à-fait produire des spécifications fonctionnelles **sans avoir besoin d'autres logiciels qu'un simple traitement de texte** tels que Word ou Google docs. Toutefois, **Axure** vous simplifiera grandement la tâche quand il s'agira de **formaliser un document complet de spécification**.

Une fois vos wireframes réalisés, vous pourrez via le menu « *publish* » (publier) utiliser la fonctionnalité « *Generate Word Specification...* ».



Le logiciel reprendra alors **chacun des écrans que vous avez conçus**, les **commentaires** et **spécifications** que vous avez apportées à **chaque élément** et **générera un document Word complet et structuré**. Vous y retrouverez **tous vos wireframes, tous vos commentaires**, les **masters** seront **isolés** pour **préserver l'unicité des descriptions** et vous obtiendrez un document de qualité en quelques secondes.



Vous **gagnerez énormément de temps** dans la formalisation de ce type de livrable, en particulier pour la réalisation des **tableaux de spécification** :

Footnote	Name	Type	Élément actif	Cible	Description	Configurable (BO)	Mode de contribution (BO)	Cas d'erreur	Fichiers sources
1	Arrière-plan vidéo	Vidéo	Oui	Pas de destination externe. Au toucher-glisser, l'utilisateur peut faire défiler les étapes du tutorial de droite à gauche. En fin de série (à droite comme à gauche) le défilement est bloqué et signifié par un effet de rebond. L'arrière-plan vidéo et le logo restent fixe, seules les étapes texte du tutorial (notes 4 et 5) défilent en lien avec les indicateurs de position (note 6).	Vidéo de présentation de l'application. La vidéo tourne en boucle. Les contrôles vidéo classiques ne sont pas accessibles.	Oui	L'administrateur peut télécharger la vidéo via le back-office. En cas de chargement d'une nouvelle vidéo par l'administrateur, elle remplace la précédente.	Si la vidéo n'est pas présente, une image par défaut est affichée.	Vidéo : [LOCALISATION]/Videos/video_intro.mov Image de backup : [LOCALISATION]/Images/bacup_video.jpg
2	Logo	Image	Oui		Logo de l'application.	Oui	Import d'un fichier image dans le back-office par l'administrateur.		[LOCALISATION]/Images/logo.png
3	Titre étape tutorial	Texte	Non		Titre de l'étape du tutorial. Alignement : centré.	Oui	Saisie via le back-office.		[LOCALISATION]/textes/intro/textes_intro.xlsx
4	Contenu étape tutorial	Texte	Non		Texte de l'étape du tutorial présentée. Le texte peut être de longueur variable. Alignement : centré. En cas de texte plus long, des lignes viennent s'ajouter sans modifier le positionnement du haut du bloc de texte (cf. créations).	Oui	Saisie via le back-office.		[LOCALISATION]/textes/intro/textes_intro.xlsx
5	Indicateur de position tutorial	Élément d'interface	Non		Indique le nombre total d'étapes du tutorial et la position de l'étape de tutorial actuellement présentée.	Non			
6	Bouton de connexion	Composant actif	Oui	A.1. Connexion	Bouton d'accès à l'écran de connexion.	Non			[LOCALISATION]/textes/intro/textes_intro.xlsx

Complément

Dès lors que vous êtes **étudiant**, vous pouvez **télécharger gratuitement** la version éducation du logiciel **Axure**. Pour plus d'information, vous pouvez consulter la page¹. Si vous n'êtes pas étudiant, une **version d'essai de 30 jours** est également disponible.

Vos spécifications ainsi formalisées, **votre cahier des charges ne laissera plus aucune zone d'ombre sur le fonctionnement attendu pour votre application**. Vous pourrez ainsi sereinement **sécuriser votre périmètre de réalisation** tout en offrant un **référentiel de qualité** aux équipes de production.

Exercice : Quiz

Question 1

« Des **wireframes en basse fidélité** sont suffisants pour servir de socle aux spécifications fonctionnelles, en particulier lorsque les délais projet sont serrés. » **Vrai ou faux ?** (Une seule réponse juste)

- ☐ Vrai
- ☐ Faux

Question 2

¹ <https://www.axure.com/edu>

Parmi les **critères de spécification** suivants, indiquer les ceux qui vous semblent **pertinents** pour **décrire** une fonctionnalité : (Plusieurs réponses possibles)

- ☐ Le nom de la fonctionnalité.
- ☐ L'action attendue au clic.
- ☐ L'action attendue au survol.
- ☐ La localisation des fichiers sources.
- ☐ Les fonctions javascript à utiliser.
- ☐ Les mots-clés à utiliser pour le référencement.

Question 3

« Un élément qui n'est **pas interactif** ne sera pas décrit dans les spécifications fonctionnelles. » **Vrai ou faux ?**
(Une seule réponse juste)

- ☐ Vrai
- ☐ Faux

Question 4

Imaginez que lors d'une création de compte, l'utilisateur doivent **saisir son adresse de courriel** dans un formulaire. Comment pourriez-vous **décrire les cas d'erreur** correspondants dans vos spécifications ? (Plusieurs réponses possibles)

- ☐ Champs de saisie e-mail, obligatoire.
- ☐ Si l'adresse e-mail n'est pas complétée lors de la validation du formulaire ou que le format n'est pas correct, afficher un message d'erreur.
- ☐ Si l'adresse e-mail n'est pas complétée à la perte de focus sur le champ, indiquer visuellement le champ en erreur et afficher le message « *Ce champ est obligatoire* ».
- ☐ Si l'adresse e-mail saisie n'est pas au format *xxx@domaine.ext* à la perte de focus sur le champ, indiquer visuellement le champ en erreur et afficher le message « *Utilisez un format d'e-mail valide* ».
- ☐ Si l'adresse e-mail n'est pas complétée à la perte du focus, afficher le champ en rouge et indiquer le message d'erreur correspondant.

Exercice

Quels types d'informations relatives à **l'administration du site** pourriez-vous indiquer dans vos spécifications fonctionnelles ? (Plusieurs réponses possibles)

- ☐ Indiquer que l'élément considéré est administrable en back-office.
- ☐ Préciser le motif de répétition d'un contenu dynamique, par exemple concernant l'affichage de vignettes produits.
- ☐ Préciser comment l'administrateur du site pourra mettre à jour un élément d'interface tel qu'un visuel d'illustration.
- ☐ Aucune de ces propositions. Les spécifications fonctionnelles ne doivent pas décrire les fonctionnalités d'administration.

V. Essentiel

Votre cahier des **charges** constitue un **référentiel permanent** depuis dès le début du projet et jusqu'à son achèvement. **Les spécifications y tiennent logiquement une place de choix** en décrivant **dans les détails** la solution attendue en fin de réalisation.

Les spécifications se déclinent en **spécifications fonctionnelles**, qui visent à **décrire la construction des différents écrans**, les **fonctionnalités attendues** et les **comportements** des éléments dynamiques. Les **spécifications techniques** vont préciser **comment, techniquement**, chaque fonctionnalité sera développée. Elles constitueront l'essentiel de ce qu'on appelle le **cahier des charges technique**.

Le plus souvent, **on présente les spécifications fonctionnelles au travers de wireframes commentés**, mais elles pourront également être **rédigées à partir des créations finalisées** si elles sont disponibles. Elles seront **utiles en début de projet** pour **formaliser les engagements** de réalisation, **en cours de réalisation** pour **guider les équipes de production** mais aussi **en fin de projet** en servant de **référentiel lors de la phase de recette**.

Si elles ont toute leur place dans le cahier des charges, les spécifications fonctionnelles peuvent également **faire l'objet d'une livraison spécifique** en cours de projet et en constituer **un des livrables clés** dont les **échéances de remise et de validation** devront être précisées.

Dans tous les cas, elles devront **refléter les évolutions** probables de la solution et être **mises à jour chaque fois que nécessaire**.

Lorsque vous réaliserez vos spécifications fonctionnelles, vous prendrez soin d'**utiliser des représentations d'écran suffisamment détaillées** pour **faciliter la compréhension** des éléments commentés. Des **critères de descriptions systématiques** faciliteront à la fois **la rédaction des spécifications** et leur **consultation**. Ainsi, **chaque élément d'interface** sera décrit par un **nom**, un **type**, une **description** et une **cible** éventuelle. Ces descriptions doivent être **précises** et **éviter toute omission ou ambiguïté**.

Des spécifications fonctionnelles complètes tiendront compte des **cas d'erreur éventuels** et des **principes d'administration** nécessaires.

Il est bien entendu possible de **produire des spécifications fonctionnelles sans l'aide d'aucun outil spécifique**, un simple **traitement de texte** vous le permettra. Néanmoins, des logiciels tels qu'**Axure** vous permettront de **produire en quelques secondes** des documents de spécifications **structurés** qui **sécuriseront votre périmètre** de réalisation tout en fournissant **un document de référence** essentiel à l'équipe de réalisation.

VI. Auto-évaluation

Exercice 1 : Quiz

Question 1

À l'aide **quels outils** pouvez-vous produire des spécifications fonctionnelles ? (Plusieurs réponses possibles)

- ☐ Un logiciel de conception spécialisé, comme Axure.
- ☐ Un logiciel de traitement de texte comme Word ou Google Docs.
- ☐ Un logiciel de présentation comme Powerpoint ou Google Slides.

Exercice

« Il est plus simple de **formaliser** les spécifications fonctionnelles **après** avoir réalisé la conception de l'ensemble des écrans. » **Vrai ou faux** ? : (Une seule réponse juste)

- ☐ Vrai
- ☐ Faux

Question 3

Voici plusieurs affirmations à propos des **spécifications fonctionnelles**. Indiquez celles qui sont **exactes** : (Plusieurs réponses possibles)

- ☐ Dans des spécifications fonctionnelles, on ne décrit qu'une seule fois chaque fonctionnalité, même si elle est présente dans plusieurs écrans.
- ☐ Seules les fonctionnalités clés et/ou complexes doivent être décrites dans les spécifications fonctionnelles. Les fonctionnalités évidentes à comprendre ne seront pas décrites.
- ☐ Ne décrire qu'une seule fois chaque fonctionnalité facilite l'accès aux informations utiles et simplifie la mise à jour de ce document de référence.
- ☐ Contrairement aux fonctionnalités présentes dans l'en-tête ou le pied de page, des éléments présents sur quelques pages seulement doivent être décrits dans chacune de ces pages.

Question 4

Dans quelles **grandes phases du projet** les **spécifications fonctionnelles** peuvent-elles être utilisées ? (Plusieurs réponses possibles)

- ☐ Dans la phase avant-projet.
- ☐ Au début du projet.
- ☐ Pendant le déroulement du projet.
- ☐ Durant la phase de recette.
- ☐ Pendant la phase de vie du projet.

Question 5

« Les spécifications techniques découlent des spécifications fonctionnelles. » **Vrai ou faux** ? : (Une seule réponse juste)

- ☐ Vrai
- ☐ Faux

Exercice 7

Complétez le texte suivant :

Quand un cahier des charges contient la description détaillée des fonctionnalités attendues, on parle alors de cahier des charges . la présentation de la construction des différents écrans, des fonctionnalités attendues et des comportements des éléments dynamiques constituent les . on les présente généralement sur la base de commentés, associés à un tableau comportant les détails des proposées. si la réalisation évolue, elles devront être mises à jour.

Exercice 8

Dans vos spécifications fonctionnelles, vous gagnerez à standardiser le mode de représentation. Ainsi chaque élément pourra être décrit en apportant les précisions attendues pour chaque catégorie de description. Encore faut-il y indiquer des informations cohérentes.

Nous vous présentons ci-dessous 5 catégories type que vous pourriez utiliser pour vos spécifications. Associez les commentaires proposés aux bonnes catégories de spécification.

L'ordre d'affichage peut être défini via l'interface d'administration par le rôle administrateur.

Image

Élément d'interface

<https://www.google.fr>

La vidéo se lance automatiquement à l'affichage de la page et tourne en boucle ; les boutons de contrôle de la vidéo ne sont pas affichés.

connexion.html

Page de rubrique produits outdoor

Si la photo de l'article est manquante, afficher l'image par défaut : « *missingpicture.png* ».

Au clic, l'utilisateur est redirigé vers la page produits outdoor.

Lorsque le focus quitte le champ et si son contenu est vide, identifier visuellement le champ en erreur.

Le code produit est renseigné par le contributeur au moment de la création d'un nouveau produit.

En cas de perte de connexion, afficher l'écran d'attente de synchronisation.

Au survol, une infobulle comportant le texte « *Recherche* » apparaît.

L'administrateur peut télécharger la vidéo via l'interface d'administration ; chaque nouveau téléchargement remplace le précédent.

Texte

Type	Cible	Description	Mode de contribution en back-office.	Cas d'erreur