

# Principes d'interface (WIP)

## Principe de navigation





L'application Smart-RX cible exclusivement les pharmaciens français à l'heure où ce document est écrit. Dans notre culture, partir vers la droite symbolise la progression, tandis que partir vers la gauche symbolise le retour en arrière (c'est par exemple le modèle mental des navigateurs internet). Nous pouvons adopter ce principe de progression dans le produit et en tirer quelques règles :

- "Avancer" dans l'activité en cours du produit signifie que l'on se déplace vers la droite, ou que le contenu se déplace de la droite vers la gauche.
- Corollaire, "Revenir en arrière" dans l'activité en cours du produit signifie que l'on se déplace vers la gauche, ou que le contenu se déplace de la gauche vers la droite.

Ce principe est largement répandu, les fil d'Ariane, par exemple, modélise l'avancée comme une progression de la gauche vers la droite.

## Principe d'interaction

Les éléments activables doivent informer l'utilisateur du résultat de l'action qui va être entreprise, de ce qui va se passer. La tableau suivant montre ainsi le symbole à afficher sur pour une section "ouvrable" type accordéon.

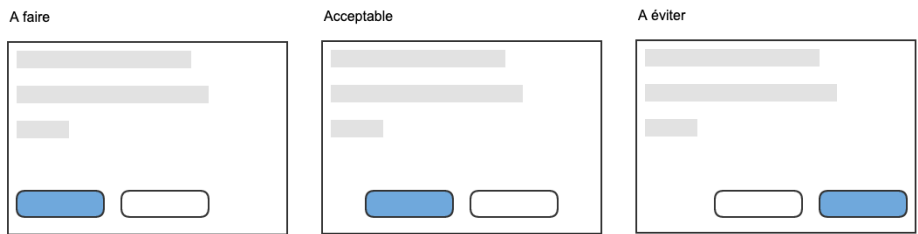
Situation de départ	Afficher	Ne pas afficher
L'accordéon est fermé		
L'accordéon est ouvert		

Notez que la couleur est juste là pour illustrer ce qui est préconisé et ce qui ne l'est pas.

## Placement des boutons type Valider/Annuler, etc

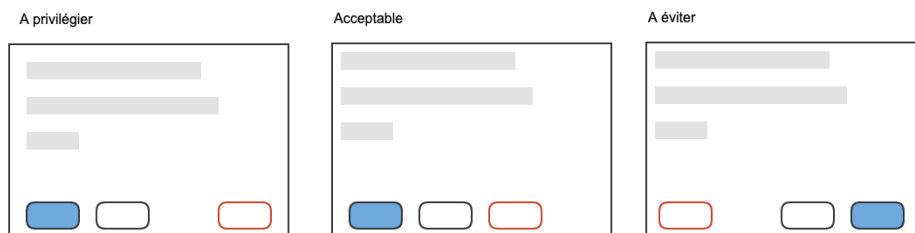
Les boutons qui terminent une étape de travail doivent privilégier le chemin le plus court pour le doigt ou la souris en vertu de la [loi de Fitts](#) (voir aussi [cet article](#) très complet sur le sujet).

Dans le dessin ci-dessous, le chemin le plus direct est de descendre directement vers le bouton, pas d'aller totalement vers la droite. La disposition à droite est celle qui implique le plus long trajet, et impliquera des incohérences si nous la maintenons.



## Action positive en premier

Dans le même esprit que le placement des boutons ci-dessus, l'accès privilégié doit être donné à l'action positive (valider, ajouter, etc), l'action négative (fermer, supprimer, etc) sera alors plus éloignée. Ce fonctionnement est celui qui se rapproche le plus de notre cerveau qui pense "positif" en premier lieu.



## Label des boutons

Sauf cas particuliers, les labels des boutons d'actions sont des verbes, si possible avec un complément pour précisément décrire ce qui va se passer quand on va activer le bouton. On évitera de mettre OK ou des mots trop généraux. La précision oriente l'utilisateur et réduit les possibilités d'erreur. [Un article](#) très intéressant sur le sujet.

## Activable versus Activé

De la même façon que les contrôles doivent illustrer ce qui va se passer lors de l'activation (et non l'état courant), pour les contrôles concernés (tab, navigation, etc) on mettra en évidence les éléments activables au détriment de l'élément activé. Ce principe renforcera la cohérence de nos interfaces.



Dans l'exemple ci-dessus, on illustre que le "2tab" est activé, mais on matérialise clairement que les "1 tab" et "3 tab" sont activables.

## Règles pour les tableaux

Articles qui liste [les bonnes pratiques pour les tableaux](#).

Une maquette XD est en cours pour présenter les tableaux Smart Rx, quand cette maquette sera plus avancée, quelques principes de tableaux seront décrits ici.

## ➖ Ce qui suit n'est PAS à jour

### Autres règles

- Des options de changements d'état doivent être des adjectifs, ce ne sont pas des actions.
- PAS de menu hamburger, autant que possible
- Ne montrer que le nécessaire au moment où l'utilisateur en a besoin. Exemple : ne montrer que les raccourcis clavier activables sur l'écran en cours.
- Ne jamais discriminer une information sur la seule couleur. Les personnes atteintes de déficience de perception des couleurs seraient incapables de différencier les différentes informations. Voir l'exemple de Trello ici : [Interfaces](#)