

# **UX Behavioral Patterns Checklist 2024-2025**

## **Patterns UX comportementaux modernes — web et mobile**

Version : FULL v3 — généré le 2026-02-04

Checklist opérationnelle : chaque pattern inclut une règle, un exemple concret, un anti-pattern, un tag plateforme et des références (liens cliquables).

Note : la sélection vise des patterns largement utilisés en 2024-2025 ; certaines sources sont plus anciennes mais restent des références normatives.

# Table des matières

- A. États & Feedback
- B. Flux utilisateur
- C. Interactions
- D. Information
- E. Confiance & Sécurité
- A. États & Feedback — 1. Loading States
- A. États & Feedback — 2. Empty States
- A. États & Feedback — 3. Error States
- A. États & Feedback — 4. Success Feedback
- A. États & Feedback — 5. Disabled States
- B. Flux utilisateur — 6. Navigation Patterns
- B. Flux utilisateur — 7. Onboarding
- B. Flux utilisateur — 8. Progressive Disclosure
- B. Flux utilisateur — 9. Wizard / Multi-step
- B. Flux utilisateur — 10. Search & Filter
- C. Interactions — 11. Forms
- C. Interactions — 12. Actions & Confirmations
- C. Interactions — 13. Selections
- C. Interactions — 14. Drag & Drop
- C. Interactions — 15. Gestures (Mobile)
- D. Information — 16. Data Display
- D. Information — 17. Notifications
- D. Information — 18. Help & Support
- E. Confiance & Sécurité — 19. Trust Patterns
- E. Confiance & Sécurité — 20. Privacy & Consent

## **Mode d'emploi (audit rapide)**

- 1) Parcourir les 20 sections et cocher la checklist rapide en bas de chaque section.
- 2) Pour chaque item non respecté, remonter au pattern correspondant et appliquer la correction.
- 3) Vérifier l'impact sur 3 parcours : première utilisation, tâche récurrente, et cas d'erreur/offline.
- 4) Refaire une passe accessibilité (clavier, lecteur d'écran, reduce motion) avant release.

# A. États & Feedback

## Objectifs clés

- Réduire l'incertitude (attente, erreurs) et fournir une récupération claire.
- Privilégier feedback local (zone impactée) plutôt que bloquer toute la page.
- Écrire des messages neutres : quoi / pourquoi / comment corriger.

Astuce d'audit rapide : parcourez d'abord les checklists, puis revenez sur les patterns en échec.

## A. États & Feedback – 1. Loading States

Objectif : réduire l'anxiété pendant l'attente, éviter les abandons, et rendre la latence tolérable sans mentir.

### 1.1 – Choisir le bon indicateur (spinner vs skeleton vs progress) [UNIVERSAL]

Règle : Spinner pour une attente courte et indéterminée ; barre de progression déterminée si vous pouvez estimer ; skeleton pour du contenu dense (listes, cards) quand l'attente dépasse le “moment”.

Exemple : Liste d'articles : afficher 6-10 skeleton cards (image, titre, 2 lignes) puis remplacer progressivement ; upload : barre déterminée + % ; action rapide : spinner inline dans le bouton.

Anti-pattern : Afficher un spinner plein écran pour une action locale ; ou afficher une barre déterminée sans estimation fiable (pourcentage “bloqué”).

#### Références :

- Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>
- Progress Indicators (Material Design) — <https://material.io/components/progress-indicators>
- Loading (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/loading>

### 1.2 – Seuils d'affichage et perception de performance [UNIVERSAL]

Règle : Ne pas afficher d'indicateur pour des réponses quasi instantanées ; au-delà, afficher un feedback avant que l'utilisateur doute que l'action ait été prise en compte.

Exemple : Clic “Enregistrer” : état du bouton passe en “Sauvegarde...” immédiatement ; si l'opération dépasse le seuil interne, afficher skeleton / progress dans la zone concernée.

Anti-pattern : Laisser l'UI figée sans changement visuel ; ou déclencher un loader trop tôt et provoquer du “flicker” (apparition/disparition très rapide).

#### Références :

- Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>
- Website Response Times (Nielsen Norman Group) — <https://www.nngroup.com/articles/website-response-times/>

### 1.3 – Skeleton efficace : anatomie, mouvement, et stabilité du layout [WEB] [ANDROID] [iOS]

Règle : Un skeleton doit refléter la structure réelle (zones, alignements) et préserver les dimensions finales pour éviter le layout shift ; animation subtile et non distrayante.

Exemple : Card produit : bloc image, titre 60%, meta 40%, prix ; shimmer léger ; respecter la grille et la hauteur finale des lignes.

Anti-pattern : Skeleton générique qui ne ressemble pas au contenu ; shimmer agressif qui attire plus que le contenu final ; layout qui saute à l'arrivée des données.

#### Références :

- Progress Indicators (Material Design) — <https://material.io/components/progress-indicators>
- Accessibility (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/accessibility>

## **1.4 – Optimistic UI pour actions réversibles [UNIVERSAL]**

Règle : Appliquer une mise à jour immédiate si l'action est généralement rapide et facilement annulable ; prévoir une stratégie de rollback explicite.

Exemple : Like : incrément immédiat + toast “Annuler” ; déplacement d'item : mise à jour immédiate + undo ; en cas d'échec, revert + message clair.

Anti-pattern : Optimistic UI sur actions irréversibles (paiement, suppression définitive) ; absence de rollback → incohérences et perte de confiance.

### **Références :**

- • Snackbars (Material Design) — <https://material.io/components/snackbars>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://www.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://www.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **1.5 – Lazy loading : listes infinies vs “Load more” vs pagination [WEB] [ANDROID] [iOS]**

Règle : Utiliser l'infinite scroll pour exploration ; préférer “Load more” ou pagination quand l'utilisateur doit se repérer, comparer, ou revenir précisément.

Exemple : Catalogue : infinite scroll + sauvegarde de position ; tableau admin : pagination + taille de page ; feed : “load more” au bas pour contrôle.

Anti-pattern : Infinite scroll sur pages où l'utilisateur doit atteindre un footer (liens légaux, contact) ou revenir à un item précis sans mémorisation de position.

### **Références :**

- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- • Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile-guides/layout-and-content/layout-and-nav-patterns>

## **Checklist rapide (DO/DON'T)**

- Feedback visible dès l'action (bouton/zone impactée) sans bloquer toute la page.
- Aucun “flicker” : loader affiché seulement si la latence dépasse le seuil interne.
- Skeletons reflètent le layout final (pas de layout shift).
- Optimistic UI uniquement pour actions réversibles avec rollback/undo.
- Le pattern de chargement (infinite/load more/pagination) correspond au besoin de repérage.

## A. États & Feedback – 2. Empty States

Objectif : transformer un “vide” en prochaine action claire (apprentissage, création, exploration) sans culpabiliser l’utilisateur.

### 2.1 – Structure standard : illustration + titre + explication + CTA [UNIVERSAL]

Règle : Un empty state utile explique l’état, le bénéfice, et propose une action primaire (et parfois une secondaire).

Exemple : Liste vide : “Aucun projet pour l’instant” + 1 phrase “Créez votre premier projet...” + bouton “Créer un projet” + lien “Importer”.

Anti-pattern : Message vague “Rien ici” sans action ; ou illustration massive qui repousse le CTA sous la ligne de flottaison.

#### Références :

- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>
- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### 2.2 – Tonalité : encourageante vs informative selon le contexte [UNIVERSAL]

Règle : Contexte “première utilisation” : ton encourageant ; contexte “filtre/recherche trop stricte” : ton factuel + suggestions.

Exemple : Recherche vide : “0 résultat pour ‘x’ — essayez moins de filtres” + bouton “Réinitialiser filtres” + suggestions de requêtes.

Anti-pattern : Ton infantilisant ou culpabilisant (“Vous n’avez encore rien fait”) ; ou absence de piste de récupération.

#### Références :

- 5 Proven UX Strategies for “No Results” Pages (Baymard) — <https://baymard.com/blog/no-results-page>
- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

### 2.3 – Empty state “permission-gated” (accès/notifications/localisation) [iOS] [ANDROID] [WEB]

Règle : Si le vide vient d’un manque d’autorisation, expliquer clairement pourquoi et offrir une action pour activer, sans forcer.

Exemple : Carte : “Activez la localisation pour voir les magasins proches” + bouton “Autoriser” + lien “Saisir une ville”.

Anti-pattern : Afficher un écran vide sans explication ; ou bloquer toute fonctionnalité tant que l’autorisation n’est pas accordée.

#### Références :

- Accessibility (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/accessibility>
- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

## **2.4 – “Zéro données” vs “zéro résultats” : distinguer les causes [UNIVERSAL]**

Règle : Différencier “vous n’avez rien créé” (zéro données) de “votre recherche ne trouve rien” (zéro résultats) avec des recommandations adaptées.

Exemple : Historique vide : CTA “Démarrer” ; filtre trop strict : CTA “Effacer filtres” ; erreur réseau : CTA “Réessayer”.

Anti-pattern : Réutiliser la même copie et les mêmes actions dans tous les vides, ce qui empêche la récupération.

### **Références :**

- • 5 Proven UX Strategies for “No Results” Pages (Baymard) — <https://baymard.com/blog/no-results-page>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **2.5 – Empty state comme onboarding : checklist de démarrage [UNIVERSAL]**

Règle : Utiliser une mini-checklist (2-4 étapes) pour guider l’utilisateur vers le “moment aha” et réduire l’abandon initial.

Exemple : App de gestion : (1) Créer un projet (2) Inviter un membre (3) Ajouter une tâche — chaque item cliquable.

Anti-pattern : Tour imposé et non skippable ; ou checklist trop longue et non contextualisée.

### **Références :**

- • Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

## **Checklist rapide (DO/DON'T)**

- Le vide explique la cause (pas juste le symptôme) et propose une action primaire.
- Le ton est adapté (first-use vs no-results vs permission vs offline).
- Les actions proposées permettent une vraie récupération (reset filtres, suggestions, import).
- L’illustration ne vole pas l’attention au CTA.
- Le vide déclenche une progression vers le “moment aha” (checklist courte).

## A. États & Feedback – 3. Error States

Objectif : aider l'utilisateur à comprendre, corriger, et récupérer rapidement (sans honte), tout en prévenant les erreurs futures.

### 3.1 – Canal : inline vs toast/snackbar vs modal (selon impact) [UNIVERSAL]

Règle : Inline pour erreurs de champ/contexte local ; snackbar/toast pour statut non bloquant ; modal seulement si bloquant, rare, ou à risque élevé.

Exemple : Formulaire : erreur sous le champ + focus ; sauvegarde échouée : Snackbar “Échec — Réessayer” ; paiement : modal explicite + options.

Anti-pattern : Modal pour une simple validation de champ ; toast pour une erreur qui nécessite une action précise (l'utilisateur ne sait pas quoi corriger).

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Snackbars (Material Design) — <https://material.io/components/snackbars>

### 3.2 – Message d'erreur : “quoi + pourquoi + comment corriger” [UNIVERSAL]

Règle : Un message efficace décrit le problème, la cause probable, et l'action de correction, avec langage neutre.

Exemple : “Mot de passe trop court (min 12). Ajoutez des caractères.” ou “Carte refusée : vérifiez le numéro ou essayez une autre carte.”

Anti-pattern : Messages hostiles (“Erreur utilisateur”), codes techniques, ou messages sans action (“Échec”).

#### Références :

- Hostile Error Messages (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Hostile\\_Error\\_Messages.pdf](https://media.nngroup.com/media/reports/free/Hostile_Error_Messages.pdf)
- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

### 3.3 – Timing de validation : éviter la validation prématuée [WEB] [iOS] [ANDROID]

Règle : Valider en ligne quand cela fait gagner du temps, mais éviter de flagger une erreur avant que l'utilisateur ait fini (ex: onBlur / after pause).

Exemple : Email : valider au blur ; mot de passe : indiquer les règles en temps réel sans rouge tant que l'utilisateur tape ; confirmer en submit.

Anti-pattern : Erreur rouge dès le 1er caractère ; ou laisser toutes les erreurs à la fin (frustration au submit).

#### Références :

- Usability Testing of Inline Form Validation (Baymard, Jan 2024) — <https://baymard.com/blog/inline-form-validation>
- Best Practices For Mobile Form Design (Smashing Magazine) — <https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>

### **3.4 – Retry + état offline : stratégie de récupération [UNIVERSAL]**

Règle : Toujours offrir une action “Réessayer” quand c'est possible, et un état offline explicite qui préserve ce que l'utilisateur a saisi.

Exemple : Upload échoue : conserver le fichier en file d'attente + bouton “Relancer” ; édition offline : badge “Hors ligne” + sync auto.

Anti-pattern : Perdre les données saisies ; boucles de retry silencieux sans feedback ; erreurs réseau identiques à erreurs métier.

#### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • Loading (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/loading>

### **3.5 – Prévention : contraintes + affordances avant l'erreur [UNIVERSAL]**

Règle : Réduire les erreurs en guidant avant la saisie (mask, exemple, contraintes), et en rendant l'état attendu visible.

Exemple : Numéro de carte : regroupement + validation Luhn ; date : datepicker ; limites : compteur de caractères.

Anti-pattern : Laisser l'utilisateur deviner le format ; ou masquer les règles jusqu'à l'échec.

#### **Références :**

- • Best Practices For Mobile Form Design (Smashing Magazine) — <https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>
- • Mobile Form Usability: Never Use Inline Labels (Baymard) — <https://baymard.com/blog/mobile-forms-avoid-inline-labels>

### **Checklist rapide (DO/DON'T)**

- Le canal d'erreur correspond à l'impact (inline/local vs toast/global vs modal/bloquant).
- Chaque message indique quoi, pourquoi, comment corriger (ton neutre).
- Validation inline non prématuée (pas d'erreur instant dès 1er caractère).
- Récupération possible (retry, offline state, conservation de saisie).
- Prévention en amont (formats, exemples, contraintes visibles).

## A. États & Feedback – 4. Success Feedback

Objectif : confirmer quand nécessaire, sans bruit, et donner un chemin de retour (undo) lorsque l'erreur serait coûteuse.

### 4.1 – Toast/snackbar : pour confirmation non bloquante + action [WEB] [ANDROID] [iOS]

Règle : Utiliser un feedback transitoire quand l'état n'est pas évident ou quand une action de récupération est utile (undo).

Exemple : “Élément archivé” + action “Annuler” ; “Lien copié” ; “Paramètres enregistrés”.

Anti-pattern : Toast pour une action critique (paiement) ; ou snackbars empilés qui masquent l’UI.

#### Références :

- • Snackbars (Material Design) — <https://material.io/components/snackbars>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://www.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://www.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### 4.2 – Inline confirmation : quand l'utilisateur doit continuer sur place [UNIVERSAL]

Règle : Préférer l'inline quand le succès fait partie du flux (formulaire, étape de wizard) et que l'utilisateur doit poursuivre.

Exemple : Après upload : badge “Téléversé” dans la ligne + icône ; champ email : coche verte discrète.

Anti-pattern : Redirection brutale ou popup de succès qui interrompt la tâche.

#### Références :

- • Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>
- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

### 4.3 – Micro-celebration : uniquement pour milestones [UNIVERSAL]

Règle : Réserver les animations de célébration aux événements rares (fin d'onboarding, objectif atteint) ; permettre la réduction d'animation.

Exemple : Fin de parcours : check animé + confetti léger ; option “Réduire animations” respectée.

Anti-pattern : Confetti à chaque clic (fatigue) ; animations longues qui bloquent l'utilisateur.

#### Références :

- • Peak-End Rule (Laws of UX) — <https://lawsofux.com/peak-end-rule/>
- • Accessibility (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/accessibility>

## **4.4 — Quand NE PAS confirmer : actions évidentes et instantanées [UNIVERSAL]**

Règle : Si le résultat est immédiatement visible et sans risque (toggle, tri), éviter les confirmations répétitives.

Exemple : Switch dark mode : changement immédiat ; tri : liste réordonnée sans toast.

Anti-pattern : Confirmer “Réglage appliqué” à chaque toggle → bruit et désensibilisation.

### **Références :**

- • Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>
- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

## **4.5 — Undo vs confirm : privilégier Undo pour actions réversibles [UNIVERSAL]**

Règle : Pour des actions fréquentes et réversibles (archive, remove), préférer Undo plutôt que demander confirmation avant.

Exemple : Swipe “Supprimer” : applique + snackbar “Annuler” ; suppression définitive : confirmation explicite.

Anti-pattern : Double confirmation pour chaque petite action ; ou absence d'undo après action destructive légère.

### **Références :**

- • Snackbars (Material Design) — <https://material.io/components/snackbars>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **Checklist rapide (DO/DON'T)**

- Snackbars/toasts uniquement si l'état n'est pas évident ou si Undo est utile.
- Succès inline pour flux continu (formulaires, wizards).
- Micro-celebrations réservées aux jalons, non aux actions fréquentes.
- Pas de confirmations pour actions déjà visibles/évidentes.
- Undo privilégié pour actions réversibles et fréquentes.

## A. États & Feedback – 5. Disabled States

Objectif : éviter la frustration “pourquoi je ne peux pas ?” en rendant la cause et la solution explicites.

### 5.1 – Désactiver vs cacher : garder la découvrabilité [UNIVERSAL]

Règle : Désactiver quand l'action existe mais n'est pas disponible maintenant ; cacher seulement si l'action n'est jamais pertinente pour cet utilisateur.

Exemple : Bouton “Exporter” grisé tant qu'il n'y a pas de données ; menu admin masqué pour non-admin.

Anti-pattern : Cacher un élément temporairement indisponible → utilisateurs pensent que la fonction n'existe pas.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>

### 5.2 – Toujours expliquer la condition de déblocage [WEB] [ANDROID] [iOS]

Règle : Un état désactivé doit indiquer la raison et comment l'activer (inline helper, tooltip, texte sous le bouton).

Exemple : Bouton “Continuer” grisé + texte “Ajoutez une adresse pour continuer”.

Anti-pattern : Bouton grisé sans explication ; ou tooltip uniquement sur hover (inaccessible sur mobile).

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Accessibility (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/accessibility>

### 5.3 – Progressive disclosure alternative : montrer l'étape suivante [UNIVERSAL]

Règle : Remplacer “disabled” par une étape intermédiaire quand possible (setup, requirements) pour guider l'utilisateur.

Exemple : Feature premium : bouton “Débloquer” → écran d'info + essai ; upload : bouton “Ajouter un fichier” au lieu d'un submit grisé.

Anti-pattern : Mettre un formulaire complet avec submit grisé et aucune guidance → abandon.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>

## **5.4 — Accessibilité : focus, contraste, et états non uniquement couleur [UNIVERSAL]**

Règle : Le disabled ne doit pas être uniquement une baisse d'opacité ; conserver lisibilité, état vocalisable, et raisons disponibles au clavier/lecteur d'écran.

Exemple : Texte de raison associé au contrôle ; composant accessible (aria-disabled, role) ; contraste suffisant.

Anti-pattern : Disabled trop pâle illisible ; focus qui se perd ; information uniquement via couleur/gris.

### **Références :**

- • Accessibility (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/accessibility>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **5.5 — Éviter le “dead-end” : proposer une action alternative [UNIVERSAL]**

Règle : Si une action est indisponible, proposer une alternative immédiate (ex: “Enregistrer brouillon”, “Contacter support”, “Voir docs”).

Exemple : Export indisponible : proposer “Télécharger CSV d'exemple” ou “Demander accès”.

Anti-pattern : État bloqué sans alternative ni explication → frustration.

### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

## **Checklist rapide (DO/DON'T)**

- L'état disabled est utilisé pour indisponible temporaire ; hide pour non pertinent permanent.
- La raison de désactivation est toujours visible (mobile compris).
- Quand possible, transformer un disabled en étape de setup (progressive disclosure).
- Contraste et accessibilité : pas “juste gris”.
- Alternative proposée (brouillon, docs, contact) pour éviter l'impasse.

## B. Flux utilisateur

### Objectifs clés

- Rendre le parcours prévisible : où je suis, où aller, comment revenir.
- Progression visible + reprise (brouillon) pour tâches longues.
- Recherche/filtre = formulation guidée + récupération quand 0 résultat.

Astuce d'audit rapide : parcourez d'abord les checklists, puis revenez sur les patterns en échec.

## B. Flux utilisateur — 6. Navigation Patterns

Objectif : permettre aux utilisateurs de comprendre où ils sont, où aller, et revenir sans perdre le contexte.

### 6.1 — Position de la navigation principale selon la plateforme [WEB] [ANDROID] [iOS]

Règle : Mobile : navigation primaire souvent en bas (thumb-friendly) ; web desktop : top/side selon profondeur ; respecter conventions de plateforme.

Exemple : App mobile : 3-5 destinations en bottom nav ; web SaaS : sidebar persistante + regroupement par domaines.

Anti-pattern : Mettre 8+ items en bottom nav ; mélanger navigation primaire et actions (ex: "+") sans hiérarchie claire.

#### Références :

- • Build Adaptive Navigation (Android Developers) — <https://developer.android.com/develop/ui/compose/layouts/adaptive/build-adaptive-navigation>
- • UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>
- • Fitts's Law (Laws of UX) — <https://lawsofux.com/fittss-law/>

### 6.2 — Hiérarchie : distinguer navigation, actions, et paramètres [UNIVERSAL]

Règle : La nav sert à changer de section ; les actions modifient l'état ; les paramètres sont secondaires. Séparer visuellement et structurellement.

Exemple : Sidebar : sections (Projets, Rapports) + bloc "Actions" + lien "Paramètres" en bas.

Anti-pattern : Boutons d'action dans la nav principale ; ou paramètres au même niveau que la tâche principale.

#### Références :

- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### 6.3 — Breadcrumbs : quand les utiliser [WEB]

Règle : Utiliser breadcrumbs pour hiérarchies profondes et navigation multi-niveaux (catalogues, docs, admin).

Exemple : Produit : Accueil > Catégorie > Sous-catégorie > Produit (cliquable).

Anti-pattern : Breadcrumbs pour une navigation plate (3 niveaux max) ; ou breadcrumbs non cliquables.

#### Références :

- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- • Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

## 6.4 – Back behavior et restauration de contexte [ANDROID] [iOS] [WEB]

Règle : Retour doit ramener à l'état précédent (scroll, filtres, onglet) ; préserver le contexte au retour des détails vers liste.

Exemple : Liste filtrée → détail → back : même position et mêmes filtres ; deep link : back mène à une page logique (pas une impasse).

Anti-pattern : Back qui renvoie en haut de la liste, ou réinitialise les filtres sans prévenir.

### Références :

- • Layouts and Navigation Patterns (Android Developers) —  
<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- • Response Times: The 3 Important Limits (Nielsen Norman Group) —  
<https://www.nngroup.com/articles/response-times-3-important-limits/>

## 6.5 – Deep linking + URLs stables + état partageable [WEB] [ANDROID] [iOS]

Règle : Les vues importantes doivent être partageables : URL stable (web) / deep link (apps) ; inclure l'état minimal (ex: filtre clé).

Exemple : “/orders?status=pending” ou lien “Ouvrir la fiche client” ; ouvrir dans l'app ou web fallback.

Anti-pattern : États non partageables (hash non stable), deep links cassés, ou écrans qui s'ouvrent sans contexte et bloquent l'utilisateur.

### Références :

- • Layouts and Navigation Patterns (Android Developers) —  
<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## Checklist rapide (DO/DON'T)

- Navigation primaire conforme aux conventions de plateforme (mobile vs desktop).
- Actions et navigation séparées (pas de mélange).
- Breadcrumbs utilisés uniquement quand la hiérarchie le justifie (web).
- Back restaure scroll/filtre/onglet et n'efface pas le contexte.
- Deep links / URLs stables pour les vues importantes.

## B. Flux utilisateur — 7. Onboarding

Objectif : amener rapidement l'utilisateur au bénéfice clé ("aha moment") avec friction minimale.

### 7.1 — Onboarder seulement quand nécessaire (complexité, risque, nouveauté) [UNIVERSAL]

Règle : Onboarding doit être proportionné : si l'UI est auto-explicative, laisser explorer et utiliser des aides contextuelles.

Exemple : App simple : pas de tour, mais empty states guidés ; app complexe : checklist + coach marks sur 2-3 actions.

Anti-pattern : Tour complet obligatoire dès le 1er lancement ; ou écrans marketing qui retardent la première action.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

### 7.2 — Progressive onboarding : au moment du besoin [UNIVERSAL]

Règle : Présenter une info juste avant qu'elle soit utile (just-in-time), plutôt que tout expliquer d'un coup.

Exemple : Premier filtre utilisé → tooltip "Vous pouvez enregistrer ce filtre" ; première export → explication de format.

Anti-pattern : Afficher 10 coach marks en cascade au 1er écran ; ou aide générique hors contexte.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>

### 7.3 — Coach marks : courts, actionnables, et skippables [WEB] [ANDROID] [iOS]

Règle : Coach marks doivent pointer une seule action, tenir en 1-2 phrases, et offrir "Ignorer / Plus tard".

Exemple : Sur bouton "+" : "Créez votre premier projet ici." + "OK" + "Ignorer".

Anti-pattern : Coach marks sans sortie ; ou qui bloquent l'UI tant qu'on ne les a pas tous cliqués.

#### Références :

- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>
- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/mobile-guidelines/layout-and-content/layout-and-nav-patterns>

## **7.4 — Permission priming : expliquer avant la demande système [iOS] [ANDROID]**

Règle : Avant la boîte de dialogue système, expliquer la valeur et donner le contrôle (Continuer / Pas maintenant).

Exemple : “Autoriser notifications pour vous rappeler vos tâches. Vous pouvez changer plus tard.” → bouton “Continuer”.

Anti-pattern : Demander la permission dès le launch sans contexte ; ou nagger plusieurs fois par session.

### **Références :**

- • Accessibility (Apple Human Interface Guidelines) —  
<https://developer.apple.com/design/human-interface-guidelines/accessibility>
- • Layouts and Navigation Patterns (Android Developers) —  
<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

## **7.5 — Skip option obligatoire + reprise plus tard [UNIVERSAL]**

Règle : Offrir “Passer” et permettre de retrouver l’onboarding (centre d’aide, checklists) ; ne pas pénaliser le skip.

Exemple : Bouton “Passer” + accès “Guide de démarrage” dans Profil.

Anti-pattern : Onboarding impossible à skipper ; ou masquer la fonctionnalité si non terminé.

### **Références :**

- • UI Design Dos and Don’ts (Apple) — <https://developer.apple.com/design/tips/>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **Checklist rapide (DO/DON’T)**

- Onboarding proportionné à la complexité (sinon exploration libre).
- Progressif (just-in-time) plutôt que “tour complet”.
- Coach marks courts, actionnables, skippables.
- Permission priming avant prompt système, avec “pas maintenant”.
- Skip + possibilité de reprendre plus tard.

## B. Flux utilisateur — 8. Progressive Disclosure

Objectif : réduire la charge cognitive en montrant l'essentiel et en révélant les options avancées au bon moment.

### 8.1 — Defaults intelligents + options avancées regroupées [UNIVERSAL]

Règle : Afficher les choix les plus probables en premier ; regrouper l'avancé derrière “Options avancées”.

Exemple : Formulaire : champs essentiels visibles ; “Options avancées” (accordéon) pour paramètres rares.

Anti-pattern : Montrer 20 options au même niveau ; ou cacher des options critiques trop profondément.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

### 8.2 — Accordéons : pour contenu scannable, pas pour cacher l'essentiel [WEB] [ANDROID] [iOS]

Règle : Utiliser accordéons pour sections indépendantes ; garder l'information clé visible sans interaction.

Exemple : FAQ : questions en accordéon ; réglages : catégories repliables avec résumé.

Anti-pattern : Cacher les champs obligatoires dans un accordéon fermé ; ou accordéons imbriqués (perte de repères).

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

### 8.3 — “Voir plus” avec prévisualisation (teaser) [UNIVERSAL]

Règle : Montrer un aperçu partiel et laisser l'utilisateur décider d'étendre (ex: 2 lignes + “Voir plus”).

Exemple : Commentaires : 2 premiers + “Voir les 18 autres”.

Anti-pattern : “Voir plus” sans indication de volume ; ou expansion qui fait perdre la position (jump).

#### Références :

- Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>
- Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>

## **8.4 — Information hierarchy : titres, résumés, et densité [UNIVERSAL]**

Règle : Structurez l'info avec titres explicites, résumés courts, et densité adaptée (compact/comfortable).

Exemple : Tableau : toggle “Compact” ; cartes : résumé + détails en drill-down.

Anti-pattern : Tout en texte continu ; ou densité fixe qui ne convient ni desktop ni mobile.

### **Références :**

- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- • Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

## **8.5 — Gestion de charge cognitive : chunking + reconnaissance [UNIVERSAL]**

Règle : Découper en blocs ; privilégier reconnaissance (choix visibles) plutôt que rappel (mémoriser).

Exemple : Filtres : chips visibles ; wizard : étapes nommées ; formulaires : sections avec titres.

Anti-pattern : Forcer l'utilisateur à mémoriser des règles ou des valeurs sans aide.

### **Références :**

- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- • Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>

### **Checklist rapide (DO/DON'T)**

- Essentiel visible, avancé regroupé.
- Accordéons utilisés pour sections scannables, pas pour cacher l'obligatoire.
- “Voir plus” avec aperçu et indication de volume.
- Hiérarchie explicite (titres, résumés, densité).
- Chunking et reconnaissance privilégiés.

## B. Flux utilisateur — 9. Wizard / Multi-step

Objectif : réduire l'abandon sur tâches longues en rendant le progrès visible, en sauvegardant, et en gérant les erreurs localement.

### 9.1 — Indicateur de progression : steps vs progress bar [UNIVERSAL]

Règle : Utiliser steps nommés si l'utilisateur doit comprendre les étapes ; progress bar si seul le progrès global compte.

Exemple : Checkout : étapes “Adresse / Livraison / Paiement / Confirmation”.

Anti-pattern : Wizard sans indication de longueur ; ou progress bar indéterminée sur un flux multi-step.

#### Références :

- Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>
- Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>

### 9.2 — Sauvegarde automatique + reprise [UNIVERSAL]

Règle : Auto-save entre étapes et possibilité de reprendre plus tard (brouillon).

Exemple : Création d'annonce : sauvegarde local/serveur à chaque étape + “Reprendre”.

Anti-pattern : Perte de données au back ou à la fermeture ; ou autosave sans feedback minimal.

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Loading (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/loading>

### 9.3 — Navigation back/forward sans punition [UNIVERSAL]

Règle : Permettre de revenir sans effacer ; conserver les choix ; prévenir si une étape invalide des données suivantes.

Exemple : Changer de plan à l'étape 2 : message “Cela réinitialise les options de paiement” + confirmation.

Anti-pattern : Back qui reset tout ; ou empêcher le back sans raison.

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

## **9.4 — Résumé final avant validation (review step) [UNIVERSAL]**

Règle : Avant action irréversible, afficher un écran de résumé avec liens “Modifier” par section.

Exemple : Paiement : récapitulatif adresse, livraison, méthode de paiement + “Modifier”.

Anti-pattern : Forcer à revenir manuellement à chaque étape pour vérifier ; ou résumé sans possibilité d'édition rapide.

### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • How Users Perceive Security During the Checkout Flow (Baymard) — <https://baymard.com/blog/perceived-security-of-payment-form>

## **9.5 — Erreurs par étape : localiser, prioriser, et guider [UNIVERSAL]**

Règle : Afficher les erreurs au niveau du champ + résumé en haut de l'étape si nécessaire ; focus sur la première erreur.

Exemple : Écran “Adresse” : message en haut “Corrigez 2 champs” + scroll/focus sur le premier champ en erreur.

Anti-pattern : Lister des erreurs sans lien ; ou afficher l'erreur seulement après navigation à l'étape suivante.

### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • Usability Testing of Inline Form Validation (Baymard, Jan 2024) — <https://baymard.com/blog/inline-form-validation>

## **Checklist rapide (DO/DON'T)**

- Progress visible (steps nommés ou bar) + estimation implicite de longueur.
- Auto-save/brouillon + reprise.
- Back/forward conserve les données et avertit en cas d'invalidation.
- Review step avant actions irréversibles.
- Erreurs localisées, priorisées, focusable.

## B. Flux utilisateur — 10. Search & Filter

Objectif : aider l'utilisateur à formuler une requête, à affiner, et à récupérer quand il n'y a pas de résultats.

### 10.1 — Autocomplete : guider la formulation, pas seulement accélérer [UNIVERSAL]

Règle : Les suggestions doivent aider à formuler une requête (terminologie, catégories, produits) et rester modifiables.

Exemple : Suggestion surlignée copiée dans le champ pour édition ; inclure catégories et résultats rapides.

Anti-pattern : Suggestions non éditables (pas copiées) ; ou suggestions sans hiérarchie (tout mélangé).

#### Références :

- Always Copy the Active Autocomplete Suggestion to the Search Field (Baymard) — <https://baymard.com/blog/copy-search-suggestion-to-search-field>
- Offer Relevant Autocomplete Suggestions for Misspellings (Baymard) — <https://baymard.com/blog/offer-autocomplete-suggestions-for-misspellings>

### 10.2 — Tolérance aux fautes : orthographe, synonymes, variantes [UNIVERSAL]

Règle : Supporter fautes de frappe et variantes ; proposer corrections et alternatives plutôt que zéro résultat.

Exemple : “ergomic” → “ergonomic” + résultats ; “tshirt” → “t-shirt”.

Anti-pattern : Effacer les suggestions dès qu'il y a une faute ; ou renvoyer “0 résultat” sans aide.

#### Références :

- Offer Relevant Autocomplete Suggestions for Misspellings (Baymard) — <https://baymard.com/blog/offer-autocomplete-suggestions-for-misspellings>
- 5 Proven UX Strategies for “No Results” Pages (Baymard) — <https://baymard.com/blog/no-results-page>

### 10.3 — Filtres : placement, lisibilité, et reset [UNIVERSAL]

Règle : Afficher les filtres accessibles (drawer, sidebar) et rendre l'état visible via chips ; offrir “Réinitialiser” clair.

Exemple : Chips de filtres actifs au-dessus des résultats + “Tout effacer”.

Anti-pattern : Filtres cachés sans signal ; reset qui efface aussi la recherche sans prévenir.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

## **10.4 — Tri : defaults sensés + explication implicite [UNIVERSAL]**

Règle : Choisir un tri par défaut cohérent avec la tâche (pertinence, récence) et rendre le tri courant visible.

Exemple : Marketplace : “Pertinence” par défaut, dropdown visible ; admin : “Date (récent)” par défaut.

Anti-pattern : Tri par défaut surprenant (ex: prix décroissant) ; ou tri appliqué sans indication.

### **Références :**

- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- • Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>

## **10.5 — No results : 5 stratégies de récupération [UNIVERSAL]**

Règle : Transformer l'échec en guidance : suggestions de catégories, requêtes alternatives, recommandations, contact, et reset.

Exemple : Page “0 résultat” : CTA “Effacer filtres”, suggestions synonymes, catégories proches, produits populaires, chat.

Anti-pattern : Page “0 résultat” comme impasse avec seulement des “tips” génériques.

### **Références :**

- • 5 Proven UX Strategies for “No Results” Pages (Baymard) — <https://baymard.com/blog/no-results-page>

## **Checklist rapide (DO/DON'T)**

- Autocomplete améliore la formulation et reste éditabile.
- Tolérance fautes (orthographe/synonymes) + suggestions.
- Filtres : état visible (chips) + reset clair.
- Tri par défaut cohérent + état visible.
- No-results : propose des chemins concrets (pas juste des tips).

# C. Interactions

## Objectifs clés

- Réduire l'effort (formes) et sécuriser les actions à risque (undo/confirm).
- Interactions “avancées” (drag/gestures) doivent rester découvrables et accessibles.
- Toujours montrer l'état courant (sélection, validation) et comment en sortir.

Astuce d'audit rapide : parcourez d'abord les checklists, puis revenez sur les patterns en échec.

## C. Interactions — 11. Forms

Objectif : réduire l'effort de saisie, prévenir les erreurs, et maximiser le taux de compléction sans surprises au submit.

### 11.1 — Labels : préférer labels persistants (pas inline placeholders) [UNIVERSAL]

Règle : Les labels doivent rester visibles pendant la saisie ; les placeholders servent d'exemples (format), pas de labels.

Exemple : Label au-dessus du champ ; placeholder “ex: nom@domaine.com”.

Anti-pattern : Inline labels qui disparaissent au focus ; ou placeholder utilisé comme seule indication.

#### Références :

- Mobile Form Usability: Never Use Inline Labels (Baymard) — <https://baymard.com/blog/mobile-forms-avoid-inline-labels>
- Best Practices For Mobile Form Design (Smashing Magazine) — <https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>

### 11.2 — Indiquer required vs optional de façon cohérente [UNIVERSAL]

Règle : Choisir une convention unique (marquer les champs optionnels ou requis) et expliquer la logique (ex: “\* requis”).

Exemple : Formulaire long : marquer “(optionnel)” pour limiter le bruit ; ou \* pour requis + légende.

Anti-pattern : Mélanger \* et “optionnel” sans logique ; ou laisser l'utilisateur deviner.

#### Références :

- Best Practices For Mobile Form Design (Smashing Magazine) — <https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>
- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### 11.3 — Validation inline : utile, mais non prématurée [UNIVERSAL]

Règle : Valider au bon moment (pause/onBlur) ; retirer l'erreur quand corrigé ; utiliser validation positive discrète.

Exemple : Mot de passe : checklist des critères qui se coche ; email : message d'erreur au blur.

Anti-pattern : Afficher erreur en rouge dès la première frappe ; ou garder l'erreur après correction.

#### Références :

- Usability Testing of Inline Form Validation (Baymard, Jan 2024) — <https://baymard.com/blog/inline-form-validation>
- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **11.4 — Auto-focus & clavier : respecter l'intention, pas surprendre [iOS] [ANDROID] [WEB]**

Règle : Auto-focus seulement si c'est clairement l'action principale ; sur mobile, éviter d'ouvrir le clavier sans contexte.

Exemple : Écran login : focus sur email ; page avec contenu : pas d'auto-focus. Tab order logique sur web.

Anti-pattern : Auto-focus sur un champ secondaire ; tab order incohérent ; focus qui saute.

### **Références :**

- • Best Practices For Mobile Form Design (Smashing Magazine) —  
<https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>
- • Accessibility (Apple Human Interface Guidelines) —  
<https://developer.apple.com/design/human-interface-guidelines/accessibility>

## **11.5 — Prévenir l'abandon : réduire champs, pré-remplir, et chunker [UNIVERSAL]**

Règle : Minimiser les champs, utiliser autofill, pré-remplir quand possible, et chunker les formulaires longs.

Exemple : Adresse : autocomplete ; profil : préremplir nom/email ; multi-step pour processus long.

Anti-pattern : Demander des infos non nécessaires ; formulaire long en une page sans repères.

### **Références :**

- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

## **Checklist rapide (DO/DON'T)**

- Labels persistants, placeholders = exemples.
- Convention required/optional cohérente et explicitée.
- Validation inline non prématurée + disparition quand corrigé.
- Auto-focus et tab order respectent l'intention.
- Formulaires minimisés, pré-remplis, chunkés.

## C. Interactions — 12. Actions & Confirmations

Objectif : permettre des actions rapides sans erreurs graves, en utilisant confirmation, undo, et garde-fous adaptés au risque.

### 12.1 — Actions destructives : confirmer seulement si irréversible [UNIVERSAL]

Règle : Confirmer avant uniquement quand l'action est irréversible ou à haut risque ; sinon préférer Undo après.

Exemple : Suppression définitive : modal “Supprimer” + conséquence ; archive : snackbar Undo.

Anti-pattern : Confirmation pour chaque micro-action ; ou suppression définitive sans confirmation ni undo.

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Snackbars (Material Design) — <https://material.io/components/snackbars>

### 12.2 — Undo : fenêtre de récupération courte et claire [UNIVERSAL]

Règle : Proposer Undo sur actions fréquentes et réversibles ; l'action doit être évidente et accessible.

Exemple : Snack “Message supprimé” + bouton “Annuler”.

Anti-pattern : Undo caché ou trop bref ; ou Undo qui n'annule pas réellement (effets secondaires).

#### Références :

- Snackbars (Material Design) — <https://material.io/components/snackbars>
- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### 12.3 — Libellés de boutons : verbes spécifiques, pas “OK” [UNIVERSAL]

Règle : Les boutons doivent nommer l'action (“Supprimer”, “Enregistrer”), et le bouton primaire refléter l'effet final.

Exemple : Modal : primaire “Supprimer la facture”, secondaire “Annuler”.

Anti-pattern : Boutons génériques “OK / Oui / Non” sans contexte ; ou ordre incohérent entre écrans.

#### Références :

- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>
- Hostile Error Messages (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Hostile\\_Error\\_Messages.pdf](https://media.nngroup.com/media/reports/free/Hostile_Error_Messages.pdf)

## **12.4 — Bulk actions : toujours montrer la sélection et l'impact [UNIVERSAL]**

Règle : Afficher le nombre sélectionné, permettre d'annuler la sélection, et résumer l'impact avant l'action.

Exemple : Barre contextuelle “3 sélectionnés” + actions “Déplacer / Supprimer”.

Anti-pattern : Appliquer une action de masse sans feedback de sélection ; ou pas de moyen de “deselect all”.

### **Références :**

- • Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **12.5 — Raisons de désactivation (submit) : expliquer au moment utile [UNIVERSAL]**

Règle : Si un submit est disabled, indiquer précisément la raison (champs manquants, erreur) et guider vers la correction.

Exemple : Bouton grisé + liste courte “Ajoutez : email, mot de passe”.

Anti-pattern : Submit grisé silencieux ; ou erreur seulement après tentatives répétées.

### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • Usability Testing of Inline Form Validation (Baymard, Jan 2024) — <https://baymard.com/blog/inline-form-validation>

## **Checklist rapide (DO/DON'T)**

- Confirmation seulement si irréversible/haut risque ; sinon Undo.
- Undo visible, fiable, fenêtre claire.
- Boutons libellés avec verbes spécifiques.
- Bulk actions : count + possibilité d'annuler sélection + impact clair.
- Disabled submit explique exactement quoi corriger.

## C. Interactions — 13. Selections

Objectif : rendre la sélection prévisible et contrôlable, surtout sur listes longues et opérations groupées.

### 13.1 — Single vs multi select : choisir le bon contrôle [UNIVERSAL]

Règle : Radio pour choix exclusif ; checkbox pour multi ; utiliser des patterns visuels distincts et des libellés clairs.

Exemple : Filtre “Taille” multi : checkboxes ; “Livraison” : radios.

Anti-pattern : Utiliser checkbox pour un choix unique ; ou mélanger radios/checkbox sans logique.

#### Références :

- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### 13.2 — Select all / deselect all : indispensable sur grandes listes [UNIVERSAL]

Règle : Offrir “Tout sélectionner” et “Tout désélectionner” quand la liste dépasse quelques éléments.

Exemple : Liste d’emails : checkbox en header + menu “Tout sélectionner sur cette page”.

Anti-pattern : Forcer la sélection item par item ; ou “select all” ambigu (page vs tout résultat).

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

### 13.3 — Range selection : support clavier (shift+click) [WEB]

Règle : Pour web desktop et tableaux, supporter shift+click pour sélectionner une plage.

Exemple : Tableau : click item 1, shift+click item 20 → sélection 1-20.

Anti-pattern : Sélection uniquement via checkboxes sans raccourci ; ou sélection de plage qui surprend (sans indication).

#### Références :

- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- Fitts's Law (Laws of UX) — <https://lawsofux.com/fittss-law/>

### 13.4 — Persistance de sélection à travers pages/filtrage [UNIVERSAL]

Règle : Si l’utilisateur navigue, préserver la sélection ou expliquer sa portée ; afficher un compteur persisté.

Exemple : Après filtre : garder les items déjà sélectionnés + bannière “3 sélectionnés (incl. hors écran)”.

Anti-pattern : Perdre silencieusement la sélection ; ou appliquer une action de masse sans clarifier le périmètre.

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>

## **13.5 — Indicateur de sélection : count + affordance de sortie [UNIVERSAL]**

Règle : Toujours montrer combien d'éléments sont sélectionnés et offrir "Annuler sélection".

Exemple : Bar "5 sélectionnés" + "Tout annuler".

Anti-pattern : Sélection active sans indication ; l'utilisateur ne sait pas pourquoi des actions apparaissent/disparaissent.

### **Références :**

- • Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>
- • Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>

### **Checklist rapide (DO/DON'T)**

- Contrôles adaptés (radio vs checkbox).
- Select all/deselect all disponible quand nécessaire + portée claire.
- Range selection sur web desktop (shift+click) pour tableaux.
- Sélection persistée ou portée explicitée lors navigation/filtrage.
- Count visible + action "annuler sélection".

## C. Interactions — 14. Drag & Drop

Objectif : rendre le drag-and-drop découvrable, sûr, et accessible (clavier, alternatives) sans erreurs de manipulation.

### 14.1 — Affordance : rendre le draggable évident [UNIVERSAL]

Règle : Utiliser poignées, icônes, curseur, ou instruction contextuelle ; sur mobile, long-press explicite.

Exemple : Liste réordonnable : handle “≡” + cursor grab ; tooltip “Glissez pour réordonner”.

Anti-pattern : Élément draggable sans indice ; ou drags déclenchés au scroll (mobile).

#### Références :

- Fitts's Law (Laws of UX) — <https://lawsofux.com/fittss-law/>
- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>

### 14.2 — Feedback pendant drag : preview + drop zones [UNIVERSAL]

Règle : Afficher un aperçu de l'objet et mettre en évidence les zones de drop valides ; indiquer interdits.

Exemple : Kanban : colonne surlignée ; placeholder d'insertion ; icône interdit sur zone invalide.

Anti-pattern : Aucun feedback ; ou drop accepté puis erreur après coup.

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

### 14.3 — Validation et annulation : drag safe [UNIVERSAL]

Règle : Permettre d'annuler (Esc, undo) ; sur actions destructives, demander confirmation ou proposer undo.

Exemple : Déplacer un fichier : undo ; supprimer via drop sur “corbeille” : confirmation/undo.

Anti-pattern : Drop destructif immédiat sans récupération ; ou pas d'annulation possible.

#### Références :

- Snackbars (Material Design) — <https://material.io/components/snackbars>
- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### 14.4 — Mobile : long press + haptics + seuils [iOS] [ANDROID]

Règle : Sur mobile, déclencher le drag via long press, utiliser haptics/vibration et seuil de déplacement pour éviter erreurs.

Exemple : Reorder : long press 300-500ms → vibration → drag.

Anti-pattern : Drag au simple touch ; ou conflits avec swipe/scroll.

#### Références :

- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>

## **14.5 — Alternative accessible : boutons “Monter/Descendre” [UNIVERSAL]**

Règle : Toujours offrir une alternative au drag (clavier, actions).

Exemple : Boutons ↑ ↓ ou menu “Déplacer vers le haut/bas”.

Anti-pattern : Interaction impossible au clavier/lecteur d'écran ; dépendance exclusive au drag.

### **Références :**

- • Accessibility (Apple Human Interface Guidelines) —  
<https://developer.apple.com/design/human-interface-guidelines/accessibility>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### **Checklist rapide (DO/DON'T)**

- Draggable évident (handle/cursor/instructions).
- Preview + zones valides visibles pendant drag.
- Annulation/undo disponible ; actions destructives protégées.
- Mobile : long press + haptics + seuil pour éviter erreurs.
- Alternative clavier/accessibilité (↑↓, menu).

## C. Interactions — 15. Gestures (Mobile)

Objectif : exploiter les gestes sans sacrifier la découvrabilité, la prévention d'erreurs, et l'accessibilité.

### 15.1 — Swipe actions : réserver aux actions secondaires [iOS] [ANDROID]

Règle : Les gestes cachés (swipe) doivent être secondaires ; l'action principale doit rester accessible via UI visible.

Exemple : Email : swipe archive ; action visible via bouton “Archiver”.

Anti-pattern : Mettre l'action critique uniquement en swipe ; ou swipe qui déclenche sans preview.

#### Références :

- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>

### 15.2 — Pull to refresh : signaler l'état et éviter les refresh invisibles [iOS] [ANDROID]

Règle : Pull-to-refresh doit afficher feedback (spinner) et conserver le scroll ; ne pas l'utiliser comme seul moyen de refresh.

Exemple : Liste : pull → spinner en haut ; bouton “Actualiser” en option.

Anti-pattern : Refresh silencieux qui change la liste sans feedback ; ou refresh qui ramène en haut.

#### Références :

- Progress Indicators (Material Design) — <https://material.io/components/progress-indicators>
- Loading (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/loading>

### 15.3 — Edge swipe (back) : ne pas créer de conflits [iOS] [ANDROID]

Règle : Éviter de placer des interactions horizontales cruciales au bord (car back gesture).

Exemple : Carrousel : marge de sécurité ; ou désactiver swipe au bord si nécessaire.

Anti-pattern : Slider/carrousel plein écran au bord qui entre en conflit avec back gesture.

#### Références :

- Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>

### 15.4 — Découvrabilité : hints progressifs [iOS] [ANDROID]

Règle : Si un geste est important, l'indiquer une fois (hint/coach mark) ou via affordance (icône).

Exemple : Liste : première fois, afficher “Glissez pour archiver” puis disparaître.

Anti-pattern : Aucun indice ; l'utilisateur ne découvre jamais le geste.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>

## **15.5 — Accessibilité : alternative aux gestes + respect reduce motion [UNIVERSAL]**

Règle : Toujours fournir des contrôles visibles pour les actions, et respecter les réglages d'accessibilité (reduce motion).

Exemple : Boutons visibles en plus du swipe ; animations désactivables.

Anti-pattern : Gestes sans alternative ; animation obligatoire.

### **Références :**

- • Accessibility (Apple Human Interface Guidelines) —  
<https://developer.apple.com/design/human-interface-guidelines/accessibility>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### **Checklist rapide (DO/DON'T)**

- Gestes réservés aux actions secondaires, actions principales visibles.
- Feedback pour pull-to-refresh ; pas seul moyen d'actualiser.
- Pas de conflit avec edge swipe back.
- Hints pour gestes importants (au moins une fois).
- Alternatives accessibles + respect reduce motion.

# D. Information

## Objectifs clés

- Choisir la structure d'affichage selon la tâche (comparer vs explorer).
- Éviter la surcharge de notifications : canal + fréquence + contrôle.
- Aide contextuelle + escalade humaine au bon moment.

Astuce d'audit rapide : parcourez d'abord les checklists, puis revenez sur les patterns en échec.

## D. Information — 16. Data Display

Objectif : présenter l'information dans la bonne structure (table/cards/lists) avec repérage, densité, et performances adaptées.

### 16.1 — Choisir table vs cards vs list selon la tâche [UNIVERSAL]

Règle : Tables pour comparaison multi-attributs ; cards pour exploration visuelle ; lists pour scan rapide d'items homogènes.

Exemple : Admin : table triable ; marketplace : cards ; messagerie : list.

Anti-pattern : Utiliser cards pour data dense (comparaison difficile) ; table sur mobile sans adaptation.

#### Références :

- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- • Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>

### 16.2 — Pagination vs infinite scroll vs load more [UNIVERSAL]

Règle : Pagination quand repérage/retour précis est clé ; infinite scroll pour exploration ; load more pour contrôle sans pagination stricte.

Exemple : Tableau d'audit : pagination ; feed : infinite ; recherche : load more.

Anti-pattern : Infinite scroll sans sauvegarde de position ; pagination cachée/peu visible.

#### Références :

- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- • Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>

### 16.3 — Indicateurs de tri/filtre : toujours visibles [UNIVERSAL]

Règle : Afficher l'état de tri et les filtres actifs ; permettre de les retirer facilement.

Exemple : Tri : flèche + libellé ; filtres : chips ; bouton "Effacer".

Anti-pattern : Tri appliqué sans indication ; filtres invisibles (utilisateur ne comprend pas les résultats).

#### Références :

- • Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### 16.4 — Options de densité : compact vs confortable [WEB] [ANDROID]

Règle : Offrir une densité adaptée au contexte (admin vs grand public) et mémoriser le choix.

Exemple : Tableau : toggle Compact/Confort ; sauvegarde par utilisateur.

Anti-pattern : Densité unique qui force scroll excessif ou rend la lecture difficile.

#### Références :

- • Layouts and Navigation Patterns (Android Developers) — <https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

## 16.5 — Tables responsives : patterns de reflow [WEB]

Règle : Sur mobile : transformer table en cards, utiliser colonnes prioritaires, ou permettre scroll horizontal avec headers sticky.

Exemple : Table “Commandes” : cards par ligne ; colonnes secondaires sous forme de détails expandables.

Anti-pattern : Table non lisible sur mobile ; colonnes coupées ; interaction de tri impossible.

### Références :

- • Layouts and Navigation Patterns (Android Developers) —  
<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### Checklist rapide (DO/DON'T)

- Structure choisie selon tâche (comparaison vs exploration vs scan).
- Pattern de chargement adapté (pagination/infinite/load more) + restauration.
- Tri/filtre actifs toujours visibles et manipulables.
- Densité ajustable quand nécessaire, préférence mémorisée.
- Tables mobiles adaptées (reflow/priority/scroll).

## D. Information — 17. Notifications

Objectif : informer sans saturer, préserver la confiance, et donner le contrôle (fréquence, canaux, silence).

### 17.1 — Typologie : transactionnel vs marketing vs système [UNIVERSAL]

Règle : Classifier et traiter différemment : transactionnel prioritaire, marketing opt-in, système lié à sécurité/compte.

Exemple : Confirmation de paiement (transactionnel), newsletter (marketing), alerte sécurité (système).

Anti-pattern : Mélanger promo et sécurité ; utiliser push pour tout.

#### Références :

- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>
- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

### 17.2 — Choisir le bon canal : in-app vs push vs email [UNIVERSAL]

Règle : In-app pour feedback contextuel ; push pour urgence/temps réel ; email pour récap/trace.

Exemple : Changement de mot de passe : email ; rappel de tâche : push ; succès d'action : in-app.

Anti-pattern : Envoyer push pour confirmations non urgentes ; ou email pour micro-feedback instantané.

#### Références :

- Response Times: The 3 Important Limits (Nielsen Norman Group) — <https://www.nngroup.com/articles/response-times-3-important-limits/>
- UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>

### 17.3 — Gestion de fréquence : batching et digest [UNIVERSAL]

Règle : Regrouper les notifications non urgentes (batch) et offrir digests (quotidien/hebdo) selon préférence.

Exemple : 5 commentaires → 1 notification “5 nouveaux commentaires”.

Anti-pattern : Notifier chaque micro-événement ; spam → désinstallation/opt-out.

#### Références :

- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- Peak-End Rule (Laws of UX) — <https://lawsofux.com/peak-end-rule/>

### 17.4 — Centre de notifications : historique et actions rapides [UNIVERSAL]

Règle : Fournir un endroit où retrouver l'historique et agir (marquer lu, paramétriser).

Exemple : Icône cloche + liste triée ; actions “Tout marquer lu”, “Paramètres”.

Anti-pattern : Notifications éphémères sans trace ; pas de moyen de gérer.

#### Références :

- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>

## **17.5 — Do Not Disturb et préférences granulaire [UNIVERSAL]**

Règle : Permettre de couper temporairement, choisir types, canaux, et horaires.

Exemple : Paramètres : toggles par type + plage horaire silencieuse.

Anti-pattern : Un seul toggle global “Notifications” ; ou nagger après opt-out.

### **Références :**

- • UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

### **Checklist rapide (DO/DON'T)**

- Types de notifications distingués (transactionnel/marketing/système).
- Canal choisi selon urgence et contexte.
- Batching/digest pour éviter spam.
- Historique accessible + actions de gestion.
- DND + préférences granulaires, pas de nagging.

## D. Information — 18. Help & Support

Objectif : réduire la demande de support en fournissant aide contextuelle, self-serve, et escalade humaine au bon moment.

### 18.1 — Aide contextuelle : tooltips et microcopy au bon endroit [UNIVERSAL]

Règle : Mettre l'aide près de l'endroit où la décision est prise ; préférer des explications courtes et actionnables.

Exemple : Icône “i” près d'un champ complexe + tooltip ; microcopy sous champ “Nous n'afficherons pas votre email”.

Anti-pattern : Aide dans une FAQ générale difficile à trouver ; tooltips trop longs.

#### Références :

- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>
- Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### 18.2 — Help center : recherche + articles courts orientés tâches [UNIVERSAL]

Règle : Le help center doit être searchable, structuré par tâches, et proposer des articles courts.

Exemple : “Comment exporter” + étapes + captures ; search autosuggest.

Anti-pattern : Articles trop longs, jargon, pas de recherche.

#### Références :

- Always Copy the Active Autocomplete Suggestion to the Search Field (Baymard) — <https://baymard.com/blog/copy-search-suggestion-to-search-field>
- Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

### 18.3 — Contact options hierarchy : self-serve → chat → humain [UNIVERSAL]

Règle : Proposer d'abord self-serve, puis chat/assistant, puis escalade humaine pour cas complexes.

Exemple : Lien “Besoin d'aide ?” → suggestions + bouton “Contacter support”.

Anti-pattern : Cacher le contact ; ou chat qui bloque l'accès au support humain.

#### Références :

- Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

## **18.4 — Chatbot : clarifier limites et offrir escalade [UNIVERSAL]**

Règle : Un chatbot doit annoncer ce qu'il sait faire, poser peu de questions, et proposer une escalade humaine quand bloqué.

Exemple : "Je peux aider pour facturation, compte, bugs. Sinon, je vous mets en relation."

Anti-pattern : Chatbot qui boucle ; demandes répétées d'info ; pas d'escalade.

### **Références :**

- • Hostile Error Messages (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Hostile\\_Error\\_Messages.pdf](https://media.nngroup.com/media/reports/free/Hostile_Error_Messages.pdf)
- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

## **18.5 — Proactive help : détecter blocages (erreurs répétées, abandon) [UNIVERSAL]**

Règle : Déclencher aide proactive uniquement sur signaux forts (erreurs répétées, abandon d'étape), et rester non intrusif.

Exemple : Après 3 erreurs de paiement : proposer "Besoin d'aide ?" + FAQ + support.

Anti-pattern : Popups d'aide agressifs sans signal ; interruption du flux.

### **Références :**

- • Peak-End Rule (Laws of UX) — <https://lawsofux.com/peak-end-rule/>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)

## **Checklist rapide (DO/DON'T)**

- Aide proche du contexte (tooltip/microcopy).
- Help center searchable, articles orientés tâches.
- Hiérarchie de contact claire + accès au support humain.
- Chatbot transparent + escalade quand bloqué.
- Aide proactive basée sur signaux, non intrusive.

# E. Confiance & Sécurité

## Objectifs clés

- Confiance = preuves contextualisées + transparence (prix, politiques, identité).
- Sécurité perçue : rendre les zones sensibles visuellement claires.
- Consentement et réglages : choix symétriques, contrôles granulaires, sans dark patterns.

Astuce d'audit rapide : parcourez d'abord les checklists, puis revenez sur les patterns en échec.

## **E. Confiance & Sécurité — 19. Trust Patterns**

Objectif : construire la confiance par preuves, clarté, et cohérence — sans faux signaux ni surcharge.

### **19.1 — Social proof : avis, stats, et preuves contextualisées [UNIVERSAL]**

Règle : Le social proof marche mieux quand il est spécifique au contexte (produit, région, segment) et vérifiable.

Exemple : Produit : avis triables + badge “Achat vérifié” ; onboarding : “+20k utilisateurs”.

Anti-pattern : Témoignages vagues et invérifiables ; chiffres énormes sans source.

#### **Références :**

- • Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>
- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

### **19.2 — Perceived security : regrouper et renforcer les champs sensibles [WEB]**

Règle : Les utilisateurs jugent la sécurité visuellement : encadrer les champs de paiement, ajouter microcopy rassurante et badges proches.

Exemple : Zone paiement : fond distinct + icône cadenas + texte “Paiement sécurisé”.

Anti-pattern : Badges de sécurité éloignés (footer) ; champs sensibles identiques au reste → baisse de confiance.

#### **Références :**

- • How Users Perceive Security During the Checkout Flow (Baymard) — <https://baymard.com/blog/perceived-security-of-payment-form>
- • Visually Reinforce Your Credit Card Fields (Baymard) — <https://baymard.com/blog/visually-reinforce-sensitive-fields>

### **19.3 — Transparence : prix total, frais, et conditions avant engagement [UNIVERSAL]**

Règle : Afficher le prix total tôt, préciser frais et conditions ; éviter surprises à la fin.

Exemple : Panier : total + livraison estimée ; abonnement : prix/mois + période d'essai + annulation.

Anti-pattern : Frais cachés jusqu'au dernier écran ; conditions difficiles à trouver.

#### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.ngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.ngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

## **19.4 — Company info accessible : “Qui sommes-nous”, contact, mentions [WEB]**

Règle : Rendre l'identité de l'entreprise, contact, et infos légales faciles à trouver, surtout pour décisions à risque.

Exemple : Footer : adresse, contact, politique, société ; page “À propos” claire.

Anti-pattern : Site sans contact clair ; infos cachées → suspicion.

### **Références :**

- • Presenting Company Information on Corporate Websites (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Presenting\\_Company\\_Information\\_on\\_Corporate\\_Websites\\_3rd\\_Edition.pdf](https://media.nngroup.com/media/reports/free/Presenting_Company_Information_on_Corporate_Websites_3rd_Edition.pdf)

## **19.5 — Garanties et politiques : rendre la récupération explicite [UNIVERSAL]**

Règle : Mettre en avant garanties, retours, et politiques (support) au moment où l'utilisateur hésite (checkout, pricing).

Exemple : Checkout : “Retours 30 jours” près du CTA ; pricing : “Annulation en 2 clics”.

Anti-pattern : Politiques dans un PDF caché ; ou garanties trompeuses.

### **Références :**

- • Peak-End Rule (Laws of UX) — <https://lawsux.com/peak-end-rule/>
- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>

## **Checklist rapide (DO/DON'T)**

- Social proof spécifique, vérifiable et contextualisé.
- Champs sensibles visuellement renforcés (perceived security).
- Prix total et frais visibles avant engagement.
- Company info et contact faciles à trouver.
- Garanties/retours explicités au moment clé.

## **E. Confiance & Sécurité — 20. Privacy & Consent**

Objectif : obtenir un consentement clair et durable, donner contrôle et transparence sur les données, et éviter les dark patterns.

### **20.1 — Consentement : choix symétriques et non coercitifs [WEB] [iOS] [ANDROID]**

Règle : Les choix doivent être équilibrés (accepter/refuser) et compréhensibles ; éviter le “tout accepter” comme unique option.

Exemple : Banner : “Accepter” et “Refuser” au même niveau + “Personnaliser”.

Anti-pattern : Cookie wall ; refus caché ; libellés trompeurs ; consentement pré-coché.

#### **Références :**

- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>
- • Hostile Error Messages (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Hostile\\_Error\\_Messages.pdf](https://media.nngroup.com/media/reports/free/Hostile_Error_Messages.pdf)

### **20.2 — Timing : demander au moment de valeur (permission priming) [UNIVERSAL]**

Règle : Demander accès (notification, localisation) quand la valeur est claire, pas au lancement systématique.

Exemple : Localisation demandée quand l’utilisateur cherche “près de moi”.

Anti-pattern : Prompt au démarrage sans contexte → refus massif.

#### **Références :**

- • Accessibility (Apple Human Interface Guidelines) — <https://developer.apple.com/design/human-interface-guidelines/accessibility>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>

### **20.3 — Transparence : “quelles données” + “pourquoi” + “combien de temps” [UNIVERSAL]**

Règle : Expliquer en langage simple ce qui est collecté, pourquoi, et comment supprimer/exporter.

Exemple : Page confidentialité : sections courtes + liens “Télécharger mes données” / “Supprimer”.

Anti-pattern : Texte légal opaque ; absence de contrôle utilisateur.

#### **Références :**

- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) — <https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>
- • Presenting Company Information on Corporate Websites (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Presenting\\_Company\\_Information\\_on\\_Corporate\\_Websites\\_3rd\\_Edition.pdf](https://media.nngroup.com/media/reports/free/Presenting_Company_Information_on_Corporate_Websites_3rd_Edition.pdf)

## **20.4 — Granularité des réglages : contrôle fin et réversible [UNIVERSAL]**

Règle : Permettre d'activer/désactiver par catégorie (marketing, analytics) et de revenir sur sa décision.

Exemple : Paramètres : toggles par type + “Réinitialiser” ; historique des consentements.

Anti-pattern : Un seul toggle global ; ou impossibilité de retirer le consentement facilement.

### **Références :**

- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>

## **20.5 — Anti-dark patterns : éviter nudges trompeurs [UNIVERSAL]**

Règle : Éviter les patterns qui manipulent : couleurs asymétriques, wording trompeur, friction au refus, “confirm shaming”.

Exemple : Boutons de même poids visuel ; texte neutre “Refuser”.

Anti-pattern : “Refuser” en gris pâle ; message culpabilisant ; multiples écrans pour refuser.

### **Références :**

- • Hostile Error Messages (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Hostile\\_Error\\_Messages.pdf](https://media.nngroup.com/media/reports/free/Hostile_Error_Messages.pdf)
- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>

## **Checklist rapide (DO/DON'T)**

- Choix symétriques (accepter/refuser) + personnalisation.
- Demande au moment de valeur (pas au lancement par défaut).
- Transparence : quoi/pourquoi/combien de temps + suppression/export.
- Réglages granulaires, réversibles, facilement accessibles.
- Aucun dark pattern (couleur, wording, friction, shaming).

# Références (liste consolidée)

- • Response Times: The 3 Important Limits (Nielsen Norman Group) —  
<https://www.nngroup.com/articles/response-times-3-important-limits/>
- • Website Response Times (Nielsen Norman Group) —  
<https://www.nngroup.com/articles/website-response-times/>
- • Progressive Disclosure (Nielsen Norman Group) — <https://www.nngroup.com/articles/progressive-disclosure/>
- • Heuristic #9: Help Users Recognize, Diagnose, and Recover from Errors (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Heuristic\\_9\\_help\\_users\\_recognize\\_diagnose\\_recover\\_from\\_errors.pdf](https://media.nngroup.com/media/reports/free/Heuristic_9_help_users_recognize_diagnose_recover_from_errors.pdf)
- • Hostile Error Messages (NN/g report PDF) —  
[https://media.nngroup.com/media/reports/free/Hostile\\_Error\\_Messages.pdf](https://media.nngroup.com/media/reports/free/Hostile_Error_Messages.pdf)
- • Presenting Company Information on Corporate Websites (NN/g report PDF) — [https://media.nngroup.com/media/reports/free/Presenting\\_Company\\_Information\\_on\\_Corporate\\_Websites\\_3rd\\_Edition.pdf](https://media.nngroup.com/media/reports/free/Presenting_Company_Information_on_Corporate_Websites_3rd_Edition.pdf)
- • Usability Testing of Inline Form Validation (Baymard, Jan 2024) —  
<https://baymard.com/blog/inline-form-validation>
- • Mobile Form Usability: Never Use Inline Labels (Baymard) —  
<https://baymard.com/blog/mobile-forms-avoid-inline-labels>
- • 5 Proven UX Strategies for “No Results” Pages (Baymard) — <https://baymard.com/blog/no-results-page>
- • Always Copy the Active Autocomplete Suggestion to the Search Field (Baymard) —  
<https://baymard.com/blog/copy-search-suggestion-to-search-field>
- • Offer Relevant Autocomplete Suggestions for Misspellings (Baymard) —  
<https://baymard.com/blog/offer-autocomplete-suggestions-for-misspellings>
- • How Users Perceive Security During the Checkout Flow (Baymard) —  
<https://baymard.com/blog/perceived-security-of-payment-form>
- • Visually Reinforce Your Credit Card Fields (Baymard) —  
<https://baymard.com/blog/visually-reinforce-sensitive-fields>
- • Hick's Law (Laws of UX) — <https://lawsofux.com/hicks-law/>
- • Fitts's Law (Laws of UX) — <https://lawsofux.com/fittss-law/>
- • Peak-End Rule (Laws of UX) — <https://lawsofux.com/peak-end-rule/>
- • Von Restorff Effect (Laws of UX) — <https://lawsofux.com/von-restorff-effect/>
- • Zeigarnik Effect (Laws of UX) — <https://lawsofux.com/zeigarnik-effect/>
- • Snackbars (Material Design) — <https://material.io/components/snackbars>
- • Progress Indicators (Material Design) — <https://material.io/components/progress-indicators>
- • Layouts and Navigation Patterns (Android Developers) —  
<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-and-nav-patterns>
- • Edge-to-edge Design (Android Developers) —  
<https://developer.android.com/design/ui/mobile/guides/layout-and-content/edge-to-edge>
- • Build Adaptive Navigation (Android Developers) —  
<https://developer.android.com/develop/ui/compose/layouts/adaptive/build-adaptive-navigation>
- • Use Material 3 Insets (Android Developers) —  
<https://developer.android.com/develop/ui/compose/system/material-insets>
- • UI Design Dos and Don'ts (Apple) — <https://developer.apple.com/design/tips/>
- • Loading (Apple Human Interface Guidelines) —  
<https://developer.apple.com/design/human-interface-guidelines/loading>
- • Accessibility (Apple Human Interface Guidelines) —  
<https://developer.apple.com/design/human-interface-guidelines/accessibility>
- • Thinking Like A UX Writer For Better Microcopy (Smashing Magazine, Sep 2024) —  
<https://www.smashingmagazine.com/2024/09/thinking-like-ux-writer-better-microcopy/>
- • Best Practices For Mobile Form Design (Smashing Magazine) —  
<https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>