

Équipe de projet du système de conception

Résumé du sprint 5

Du 13 décembre au 14 janvier 2022

Objectifs de sprint

1. Établir les flux de travail :
 - Essais de flux de travail a11y (automatiques et manuels)
 - Flux de travail d'intégration Figma
2. Décomposer une composante :
 - Composante du bouton :
 - i. Pochoir – variantes du bouton
 - ii. Façon dont nous classifions les choses
 - iii. Texte/étiquettes
 - iv. États – survol, actif, concentration

Exigences relatives aux boutons

Appliquer les pratiques exemplaires et la logique de conception pour améliorer l'expérience globale relative aux boutons ainsi que leur convivialité

Renseignements documentés :

- **Propriétés de style** – configurables ou non, et pourquoi
- **États et comportement de l'état** – survol, mise au point, inactif, actif
- **Propositions** – ce qu'il faut conserver, modifier, supprimer, ajouter

Modèle de données : catégories

Les composantes comportent trois catégories de propriétés :

- **propriétés universelles** : propriétés que possèdent toutes les composantes – elles suivent une convention d'appellation uniforme
- **propriétés propres aux composantes** : propriétés qui ne s'appliquent qu'au modèle unique de la composante
- **dérogations personnalisées** : propriétés qui remplacent les paramètres par défaut avec des valeurs personnalisées

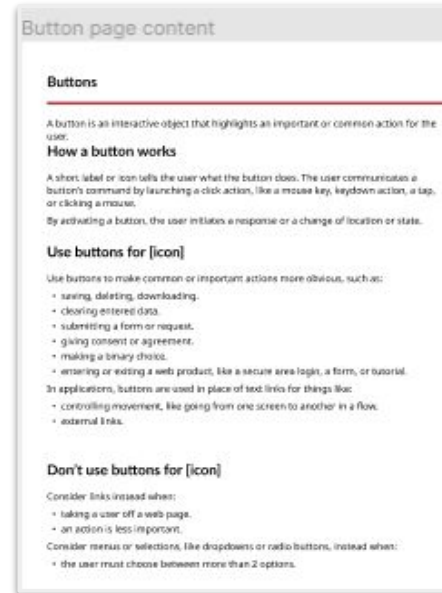
Propriétés universelles

- **type** : nom du comportement fonctionnel qui distingue une composante des autres composantes
- **tâche** : les principaux styles qui facilitent les différentes tâches d'une composante
- **variante** : les versions d'une composante dont l'aspect diffère des styles de tâches standard ou qui consomment des données différentes à une fin particulière
- **état** : options qui dictent l'état par défaut de la composante
- **langage** : langage qu'une composante doit charger

Projet de modèle de documents

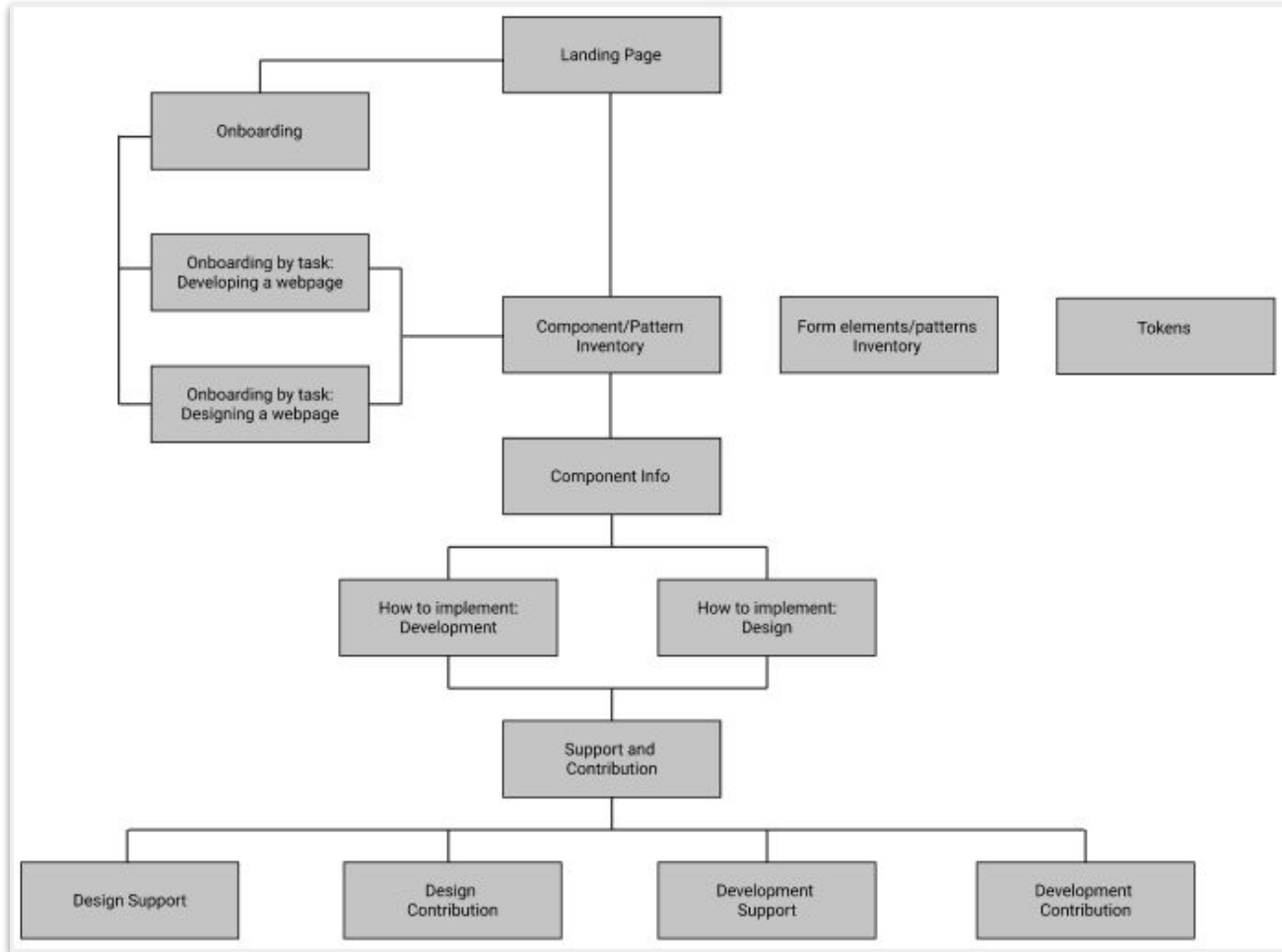
- Définition
- Fonctionnement d'un bouton
- Utiliser les boutons pour
- Ne pas utiliser les boutons pour
- Styles
 - solide
 - contour
 - texte seulement

- États
 - survol
 - actif
 - mise au point
 - inactif
- Rendre les boutons utilisables
 - étiquettes
- Rendre les boutons accessibles
 - auditif
 - cognitif
 - moteur
 - visuel



Carte de site

Élaboration conceptuelle du lien entre les sections



Essais d'accessibilité

Questions de recherche

- Le fait d'avoir des éléments interactifs, comme un bouton, dans le DOM fantôme crée-t-il un problème d'accessibilité?
- Si oui, comment peut-on y remédier?

Mise à l'essai

- Essais exécutés avec deux lecteurs d'écran (VoiceOver et NVDA)
- Utilisation d'un bouton HTML, du bouton POC (bouton gc-ds) et d'un autre bouton de composante Web (cadre ionique)

Chaque bouton était reconnu comme un bouton du lecteur d'écran.

Outils d'accessibilité

Outils CDS

- Application A11y-tracker – [GitHub](#)
- Action Git Hub A11y-checker – [GitHub](#)
- Action GitHub A11y-multiple-page-checker – [GitHub](#)

Essais manuels

Outils qu'un développeur peut exécuter dans son navigateur :

- [Accessibilityinsights.io](#)
- [Axe](#)
- [Barre d'outils WAVE](#)
- [WebAIM : Vérificateur de contraste](#)
- [Google Chrome – Lighthouse](#)

[Billet de blogue : Guide to accessible UI components \(en anglais seulement\)](#)

Essais automatisés – composants Web/pochoir

Pour les tests a11y :

- Le pochoir génère des essais unitaires et E2E (de bout en bout) pour chaque composante
- Utiliser les essais E2E pour les interactions
- Exécuter des tests sur les référentiels de composantes lorsqu'une demande de tirage est ouverte/modifiée

Émulateur de navigateur Web de niveau 0 du ~~CSS~~

Expérience réussie avec WET-BOEW

- Tire parti du langage CSS pur plutôt que de SASS/Less
- Permet aux auteurs d'utiliser le langage CSS pur à l'étape 0+ et de le tester en direct dans un navigateur

<https://wet-boew.github.io/wet-boew-experimental/src/sites/wb.dev-css/example.html>

Équipe de produit du système de conception du GC

[Slack de l'équipe de conception du GC](#) • dto.btn@tbs-sct.gc.ca