# Internal Design System Projects

## Project: Design system expansion

Contributed to the enhancement of the internal design system at Amazon Business by developing and integrating numerous new components.

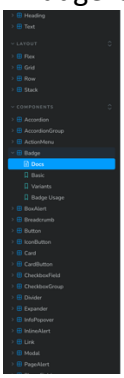
### Tech Stack

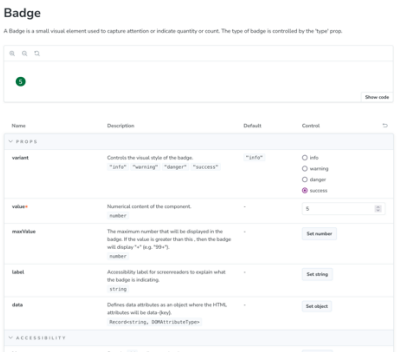
- React Ts
- Testing Library (Unit testing)
- Vanilla Extract CSS (Typescript CSS pre-processor)
- React Storybook
- TypeScript
- Design Tokens
- React-Aria
- WCAG 3/APAC accessibility standards

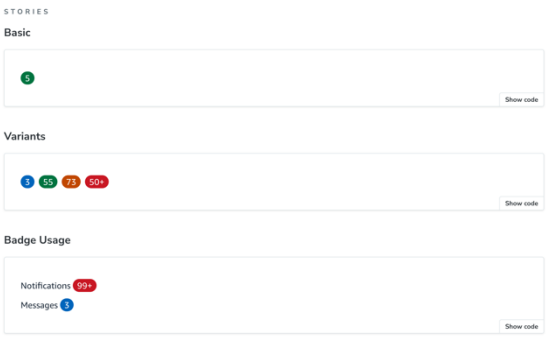
### Components developed & delivered

#### Badge Component

A Badge is a small visual element used to capture attention or indicate quantity or count.





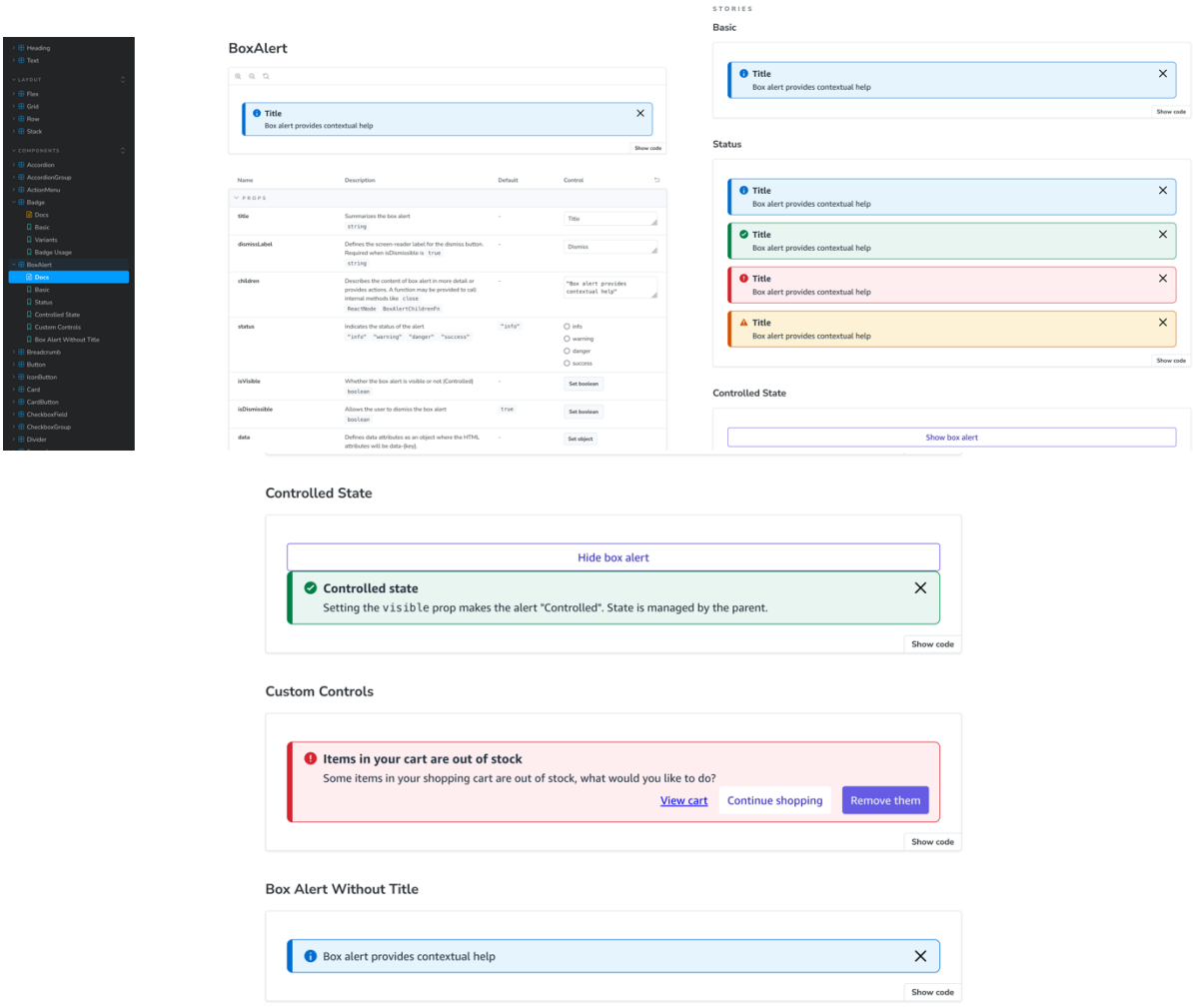

API:

| Name | Description | Default |
|---|---|---|
| **props** | | |
| **variant** | Controls the visual style of the badge.<br>"info""warning""danger""success" | "info" |
| **value*** | Numerical content of the component.<br>number | - |
| **maxValue** | The maximum number that will be displayed in the badge. If the value is greater than this , then the badge will display "+" (e.g. "99+").<br>number | - |
| **label** | Accessibility label for screenreaders to explain what the badge is indicating.<br>string | - |
| **data** | Defines data attributes as an object where the HTML attributes will be data-{key}.<br>Record<string, DOMAttributeType> | - |
| **accessibility** | | |
| **id** | Sets the id attribute on the element. | - |

| | | | |
|---|---|---|---|
| | Note: Use this prop for accessibility purposes only. Do not override component styles. Component internals may change at any time and are considered private.<br><br>string | | |
| **aria-describedby** | Identifies the element (or elements) that describes the object.<br>string | - | |
| **aria-details** | Identifies the element that provides a detailed, extended description for the object.<br>string | - | |
| **aria-label** | Defines a string value that labels the current element.<br>string | - | |
| **aria-labelledby** | Identifies the element (or elements) that labels the current element.<br>string | - | |

## Box Alert Component

The box alert component is used to provide contextual information or helpful tips to the customer. It is generally used inside a container or other components unlike page alerts.



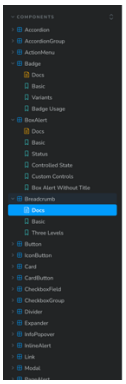

API:

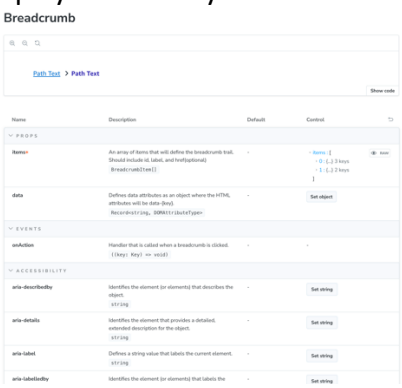
| Name | Description | Default | Control |
|---|---|---|---|
| **props** | | | |
| **title** | Summarizes the box alert<br>string | - | Title |
| **dismissLabel** | Defines the screen-reader label for the dismiss button. Required when isDismissible is true | - | Dismiss |

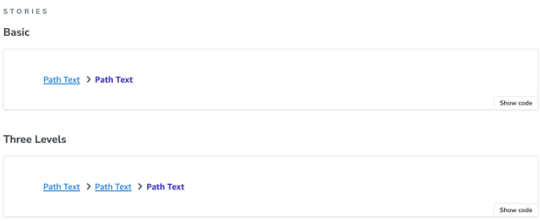
| | | | |
|---|---|---|---|
| | string | | |
| **children** | Describes the content of box alert in more detail or provides actions. A function may be provided to call internal methods like close<br>ReactNodeBoxAlertChildrenFn | - | "Box alert provides contextual help" |
| **status** | Indicates the status of the alert<br>"info""warning""danger""success" | "info" | infowarningdangersuccess |
| **isVisible** | Whether the box alert is visible or not (Controlled)<br>boolean | - | |
| **isDismissible** | Allows the user to dismiss the box alert<br>boolean | true | |
| **data** | Defines data attributes as an object where the HTML attributes will be data-{key}.<br>Record<string, DOMAttributeType> | - | |
| **events** | | | |
| **onDismiss** | Handler that is called when the dismiss button (x) is pressed<br>((ev: PressEvent) => void) | - | - |
| **onVisibleChange** | Callback when the visible state changes (Controlled)<br>((visible: boolean) => void) | - | - |
| **accessibility** | | | |
| **assertiveness** | Defines the priority of the box alert content when announced by screen readers. This is automatically set based on the status.<br>"off""assertive""polite" | - | offassertivepolite |

## Breadcrumb Component

Breadcrumbs display a hierarchy of links to the current page or resource in an application.





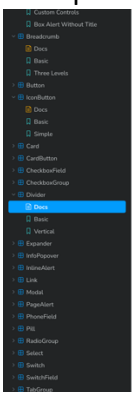

API:

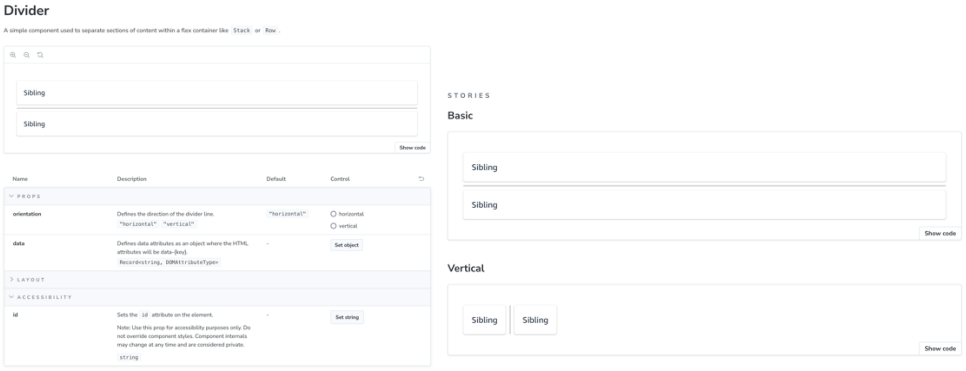
| Name | Description | Default | Control |
|---|---|---|---|
| **props** | | | |
| **items*** | An array of items that will define the breadcrumb trail. Should include id, label, and href(optional)<br>BreadcrumbItem[] | - | items : [<br>0 : {...}<br>3 keys<br>1 : {...}<br>2 keys<br>] |
| **data** | Defines data attributes as an object where the HTML attributes will be data-{key}.<br>Record<string, DOMAttributeType> | - | |
| **events** | | | |
| **onAction** | Handler that is called when a breadcrumb is clicked.<br>((key: Key) => void) | - | - |
| **accessibility** | | | |
| **aria-describedby** | Identifies the element (or elements) that describes the object.<br>string | - | |

| | | | |
|---|---|---|---|
| **aria-details** | Identifies the element that provides a detailed, extended description for the object.<br>string | - | |
| **aria-label** | Defines a string value that labels the current element.<br>string | - | |
| **aria-labelledby** | Identifies the element (or elements) that labels the current element.<br>string | - | |

## Divider Component

A simple component used to separate sections of content within a flex container.






API:

| Name | Description | Default | Control |
|---|---|---|---|
| | **props** | | |
| **orientation** | Defines the direction of the divider line.<br>"horizontal""vertical" | "horizontal" | Horizontal/vertical |
| **data** | Defines data attributes as an object where the HTML attributes will be data-{key}.<br>Record<string, DOMAttributeType> | - | |
| | **accessibility** | | |
| **id** | Sets the id attribute on the element.<br><br>Note: Use this prop for accessibility purposes only. Do not override component styles. Component internals may change at any time and are considered private.<br><br>string | - | |

# Project: Level Access Accessibility Audit

Conducted a comprehensive audit of all legacy design system components to assess their compliance with WCAG 3 and APAC accessibility guidelines. Identified and documented inconsistencies, assigning remediation tasks to ensure adherence to WCAG 3 standards.
Components assign to me for fixes/maintenance:

- Select box
- Badge
- Alerts
- Checkbox
  Icons

### Impact

This comprehensive audit ensured that all legacy internal design system components and associated websites adhere to WCAG 3/APAC accessibility guidelines. This optimization guarantees an enhanced user experience for **thousands of customers** utilizing assistive technologies such as screen readers.

## Project: Internal Design System Visual Regression Testing

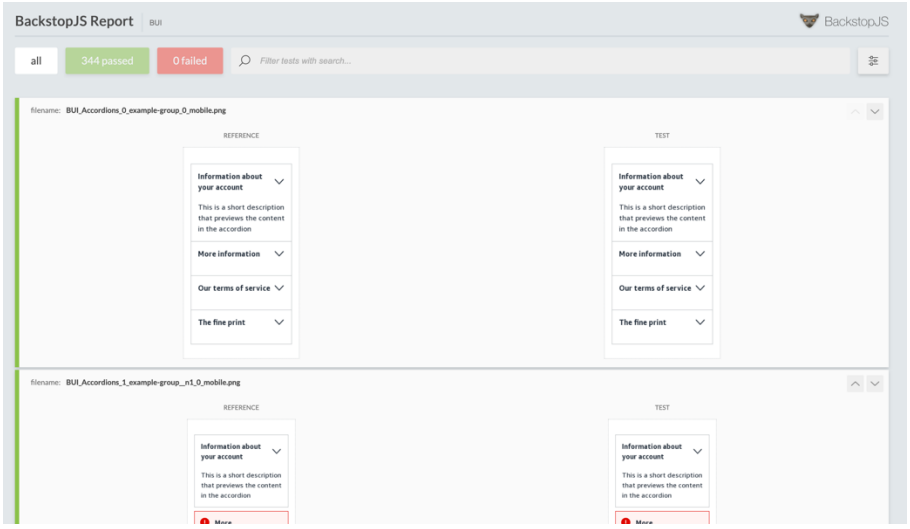
Regression testing is done whenever there are any code changes made by the developers to check if it has not broken any other functionalities of the software. In the same way, visual regression testing checks whether the code changes have not affected the software's visual interface.

Visual regression testing (VRT) helps to catch visual errors or defects that are caused due to improper styles, alignments, and fonts. Common visual issues include overlapping modules, hidden or missing elements, elements that render off-screen, etc.

In this project, I worked along with a colleague to create a VRT pipeline checkpoint in the main pipeline for the internal design system deployment that tests all the components and aborts deployment if there is any failures in the test results and notifies on slack in a channel set for all pipeline related notifications for the team so that it can be addressed timely.

### Tech Stack

- React Typescript
- Backstop JS
- Docker
- JSON
- Python
- AWS Services:
    - Lambda
    - CloudFront
    - API Gateway
    - Hosted Zones + Route 53
    - Simple Storage Service (S3) bucket



### Impact

Addition of this process as a middle step significantly mitigated risk for **500+ users** of the internal design system. By proactively identifying breaking changes resulting from design system updates through Visual Regression Testing, potential issues were caught before reaching production, ensuring a stable, robust and reliable user experience.

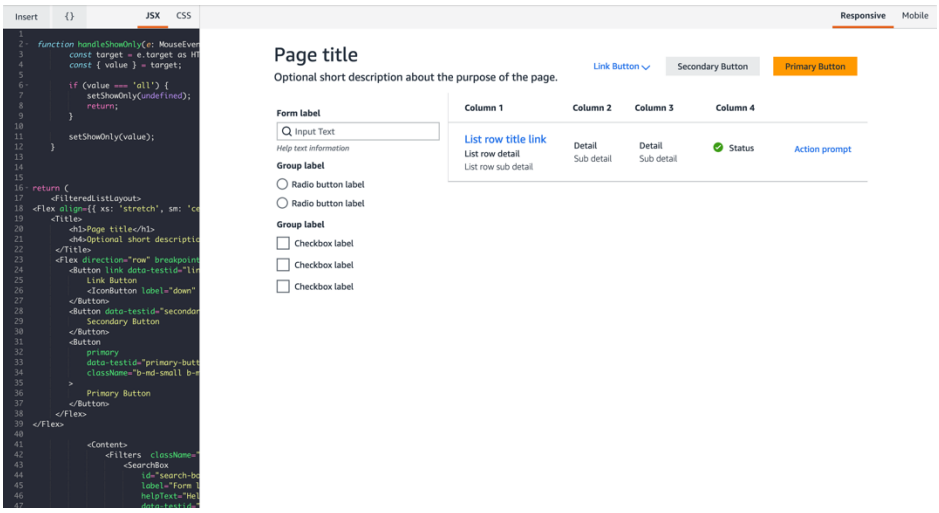
## Project: Developer templates for Design Patterns

To create molecular templates using the internal design system's atomic components to demonstrate their implementation for developers on the Docusite.

### Templates developed & delivered

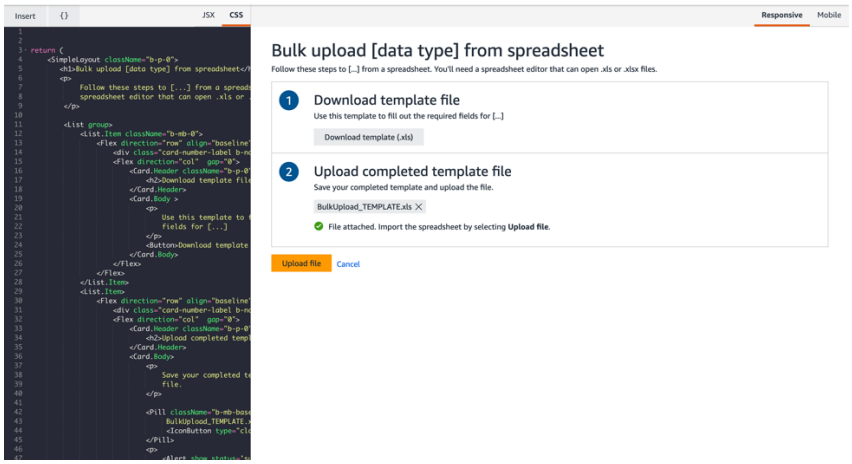
#### Filtered List layout

Primarily used while working with search functionality to display a large quantity of results with associated filters for ease of use.



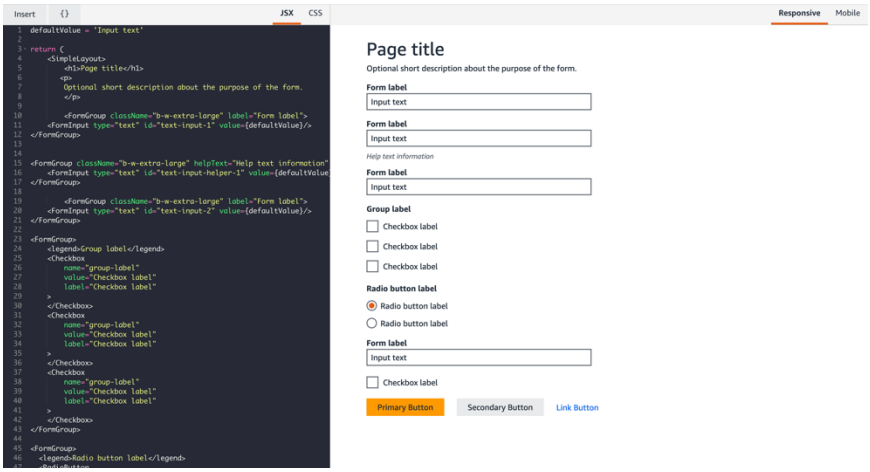

#### Bulk Upload layout

Used when the customer wants to upload details of multiple addresses/contact info etc in present in a single file rather than filling out everything manually using multiple forms



### Form layout

Used to gather details for registration etc.




### Impact

The introduction of layout patterns within the internal design system has empowered **500+ developers** to create more refined and efficient layouts. These patterns significantly reduce development effort by allowing direct import and customization, eliminating the need to build layouts from scratch using atomic components.

## Project: Documentation website content review

The internal design system docusite lists all the components of our current design system as well as the associated design and development guidelines. As a part of this project, the content of those guidelines for design and API was thoroughly reviewed for all components and re-worked in order to be complaint with WCAG 3 guidelines as well as align with the highest standards when compared to other design systems like Material UI, Atlassian DS, React-Aria etc.

### Impact

**20+ components' design and development guidelines** were revamped which resulted in all our component content being compliant with WCAG3/APAC guidelines and an overall better user experience for our customers as they no longer had to worry about their websites, built using our design system, not being WCAG3 compliant.