



Accessible Button Components for Mobile Device Screen

Jaeryeong Choe

BACHELOR'S THESIS
April 2024

Degree Programme in Media and Arts

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media and Arts

Choe, Jaeryeong:
Accessible Button Components for Mobile Device Screen

Bachelor's thesis 44 pages, appendices 0 pages
April 2024

As the European Accessibility Act (EAA) came into effect in 2019, web accessibility has been gaining more attention among designers. This thesis will help designers provide accessible button components for mobile applications that meet the most recent Web Content Accessibility Guidelines (WCAG) 2.2 released in October 2023.

This thesis was done with a thesis partner, Cargotec Finland Oy, which is a world leading corporation providing cargo and load handling products. Overall, the thesis consists of two parts, research and practical implementation in order to apply research findings into a mobile application project of Cargotec Finland Oy. The purpose of the thesis was to research how to improve accessibility of user interface (UI) buttons for mobile applications and create reusable components in a prototyping tool, Figma, according to WCAG 2.2, existing design systems, literature, and journals.

The results suggest that accessibility of UI buttons can be affected by six aspects, target size, color contrast, color use, states, visual hierarchy, and composition. In terms of practical implementation two features of Figma, component properties and variables, were mainly adopted. Component properties were applied to produce flexible button components in an efficient way. As a result, buttons in the project were kept consistent, being capable of supporting variants in diverse types, states, and structure. Also, variables feature allowed to build color tokens and streamline creating multiple UI color modes.

Key words: UX and UI design, web design, accessibility, UI button.

CONTENTS

1	INTRODUCTION	5
2	BACKGROUND RESEARCH	6
2.1	Development of accessibility	6
2.2	History of buttons.....	7
2.3	Types of buttons	8
3	ACCESSIBILITY OF BUTTONS	12
3.1	Target size	12
3.2	Color contrast	13
3.3	Color use.....	14
3.4	Button states.....	14
3.4.1	Focus state	15
3.4.2	Hover state.....	16
3.5	Visual hierarchy	17
3.6	Composition.....	19
3.6.1	Accessible composition	20
3.6.2	Interaction area.....	20
4	IMPLEMENTATION.....	22
4.1	Button component in Figma	23
4.2	Size	27
4.3	Color.....	29
4.4	State.....	33
4.5	Visual hierarchy	34
4.6	Composition.....	35
5	DISCUSSION	39
	REFERENCES	41

GLOSSARY

ADA	Americans Disability Act
APCA	Advanced Perceptual Contrast Algorithm
CSS	Cascading Style Sheets
FAB	Floating Action Button
SC	Succession Criterion
User agent	Any software that retrieves and presents web contents for end users or is implemented using web technologies.
UI	User Interface
UX	User Experience
WCAG	Web Contents Accessibility Guidelines
W3C	World Wide Web Consortium

1 INTRODUCTION

The purpose of the thesis is to research how to improve accessibility of user interface (UI) buttons and create reusable UI button components for a mobile application according to the most recent Web Content Accessibility Guidelines 2.2(WCAG 2.2), existing design systems, literature, and journals.

This thesis was done with a thesis partner, Cargotec Finland Oy which is a world leading corporation providing cargo and load handling products. The thesis consists of research and practical implementation to apply research findings into a project of Cargotec Finland Oy. Considering the specific business field of the thesis partner, accessibility of the project is critical, because accessibility aligns with the safety of its users. While some may perceive accessibility as being primarily for people with disabilities, the term ‘disability’ in the context of accessibility can be broadened. In relation to accessibility, ‘disability’ doesn’t only refer to permanent disabilities but also includes situational disabilities, where individuals may be temporarily disabled due to certain situational circumstances (Bureau of Internet Accessibility 2023). Examples of situational disabilities of target users include when they use mobile phones outdoors under inclement weather conditions or reflected direct sunlight on their screen. Furthermore, they might face urgent situations that their vision and dexterity are limited.

Research question of the thesis is “How to make digital services more accessible by improving accessibility of UI buttons”. Subsequent research questions are below:

- What are the aspects of UI buttons that can affect accessibility?
- How those aspects can be designed to comply with WCAG 2.2?
- How to implement accessibility guidelines efficiently in a design prototyping tool?

Practical implementation is conducted on Figma (the current version in 2023) by Adobe which is a widely used prototyping design software among user experience (UX) and UI designers.

2 BACKGROUND RESEARCH

2.1 Development of accessibility

A definition of accessibility is “The quality of being able to be entered and used by everyone, including people who have a disability” (Cambridge dictionary n.d.). Web Content Accessibility Guidelines (WCAG) is international guidelines that describes about how to make web contents more accessible for people with disabilities (W3C n.d.). It offers a set of criteria in four guiding concepts, perceivable, operable, understandable, and robust that create accessible web contents (Pabba n.d.). Also, it consists of three levels of conformance:

- Level A - Minimum requirements.
- Level AA - All level A and level AA. This level is commonly aimed by many organizations.
- Level AAA - Enhanced requirements, including all level A, level AA, and level AAA (W3C n.d.).

As Web Content Accessibility Guidelines 2.1 (WCAG 2.1) are source of the official requirements for The European Accessibility Act (European Commission 2023), it is widely adopted to make inclusive digital service for people of all backgrounds (Smith 2022).

However, there were milestones that helped to shape the current state of web accessibility. In 1973, the US Rehabilitation Act of 1973 was passed in the US that forbids discrimination against digital users with disabilities by organizations or individuals tied to federal government (Audioeye 2022). Subsequently in 1990, the Americans with Disability Act (ADA) was legislated which is one of the best-known laws for people with disabilities. In 1998, section 508 of the US Rehabilitation Act of 1973 was updated to guarantee digital access of users with disabilities by all organizations tied to federal government. (Sapega 2020).

Following these milestones, the first version of WCAG was released in 1999 by Tim Bernes-Lee group, W3C, that ensures to comply with the ADA (Audioeye 2022). In 2018, WCAG 2.1 was released that is in line with official requirements for the European Accessibility Act (European Commission 2023). WCAG 2.1 started to pay attention to mobile devices from previous versions by providing

more guidelines on mobile web and responsive designs for web contents (Pressbooks 2019).

In October 2023, during this thesis writing process, the newest version WCAG 2.2 was released. While the newest version is expanded from the previous version, required accessibility succession criterion (SC) in the new version is compatible with the earlier versions, WCAG 2.0 and 2.1 (WCAG 2.2 2023). With that said, the thesis will reference mainly WCAG 2.2.

An aspect that the thesis wants to emphasize is the fact that accessibility benefits everyone, making services easier and accessible for all users. Disability doesn't only refer to permanent disabilities, but also includes situational disabilities where people are temporarily disabled due to certain situational circumstances (Bureau of Internet Accessibility 2023). Bureau of Internet Accessibility (2023) clarifies examples of situational disabilities:

- When you must use only one hand due to other hand being injured or tied to hold something else.
- In the middle of the day, when there's strong reflection of sunlight on your screens and hinders to read texts.
- While driving, you want to review and reply to an urgent text message relying on your voice assistant. (Bureau of Internet Accessibility 2023)

WCAG 2.1 also states in the abstract that the guidelines help to make web contents more accessible for people with various disabilities, but it also often makes digital services more usable to all users in general (WCAG 2.1 2023).

2.2 History of buttons

Brief research on history of buttons is conducted as background research here. I believe that understanding advances of buttons in our history will help UX and UI designers create intuitive buttons that users feel friendly to use based on our previous experience with buttons in the real world. The term, button, is originally from a French word, Bouton, which means a flower bud, pimple, projection or thrusting (Wills 2021). A primary push button was introduced by John Homes, an English inventor of the early electric system, in 1884 (Abernathy 2011). Following

this invention of the push button, a transition from lever to button happened. Between 1900 to 1950s, push buttons were applied to numerous mechanical and electronical machines, such as doorbells, radios, cameras, telephones, and vehicles (DeRouchev 2010).

Around 1980s, as computers with graphic user interfaces (GUI) became commercially available, buttons were placed inside of screens being operated by a mouse. However, it wasn't an easy adaption for users in the beginning. (DeRouchev 2010). Therefore skeuomorphism, reflecting real-world objects on digital counterparts, became popular in order to make user interfaces more familiar and understandable. (Big Human 2023). Skeuomorphic style is recognizable with digital object reflecting analog objects, 3-dimensional shapes, and gradients and shadows. (Big Human 2023).

In 2007, Apple announced the first iPhone which was equipped with a touch screen instead of physical hardware buttons (Križanović 2020). Despite of relatively new adaptation of touch screen, skeuomorphic design made the interface familiar and intuitive. Nonetheless skeuomorphism didn't last long as people became familiar with touch interactions. Buttons and interfaces have changed to be simple, minimal, and flat. (Muldrew 2019).

2.3 Types of buttons

In modern UI, there are multiple types of buttons featuring their own purposes and different appearances. In the research on types of buttons, I noticed that names of button types describe either what the button does or how the button looks like. Also, depending on the perspective of categorization, a button with a certain appearance can be called with multiple names by different organizations. For example, there are many existing design systems including guidance on buttons. Among them, Material Design 3 by Google contains a type of button, filled Button, defined to have filled background and high visual impact, while another design system, Carbon Design System by IBM includes a button with very similar appearance and role that is called by a different name, primary Button (Material Design 3 n.d. ; Carbon Design system 2023). However, regardless of

variety in button type categorization, there are types of buttons commonly used in digital user interfaces.

One of commonly used buttons is filled button. Filled buttons denote buttons with distinctive background color. They convey high emphasis and are used for the most important actions or the most common actions. (Material design 3 n.d.). In comparison, there is outlined buttons that conveys medium emphasis with only outline of the container. Outlined buttons are used for relatively important but not primary actions, such as a secondary action pairing with a filled button that contains a primary action. (Material design 3 n.d.).

Floating action button (FAB) is used for the most crucial action in a service for users. They are designed to be spotted and placed on the most top level of screen elevation remaining it's defined position. Due to purpose of containing the most important action, FAB requires highlighted appearances that puts itself on the top of visual hierarchy of the page. (Material Design 3 n.d.)

Another type of button found frequently from all digital interfaces is text button. Text buttons are text labels navigating users to an action that the text describes (Material Design 3). While text buttons are often used in the website header presenting content sections of the site, they are also used for secondary interactive options that do not divert from other main actions. (Tubik 2019). In contrast, there are icon buttons. Icon button means a button that accommodates only an icon (Shoelace n.d.). Icon buttons are assigned to minor actions. It's important to use icons with a clear meaning to describe the icon button's action. (Material Design 3 n.d.) Figure 5 shows example of buttons mentioned above.

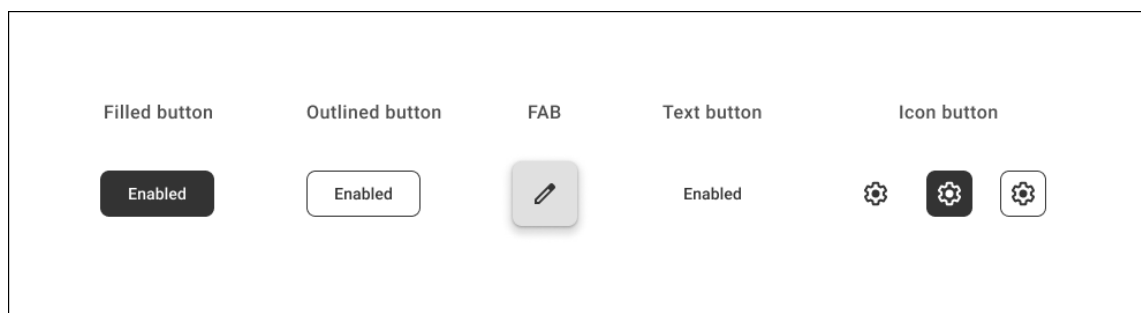


FIGURE 1. Example of types of buttons.

There are also buttons including more actions than a single action. Dropdown menu button is a group of buttons that displays a temporary surface with a list of actions in order to serve more options for users (Material Design 3 n.d.). A selected item can present the chosen value in a form or filtered content. (Carbon Design System 2023). In comparison, another group of buttons commonly seen in menu bars is split buttons (Figure 2). Laubheimer (2019) defines the split button as a button with two components, a label and an arrow icon. Unlike the dropdown menu button, a split button has distinct two parts, commonly the left side with a default option, and the right side with an arrow icon used to open a dropdown list and change the default option. Also, options in a split button are usually related commands that you can group. (Laubheimer 2019).

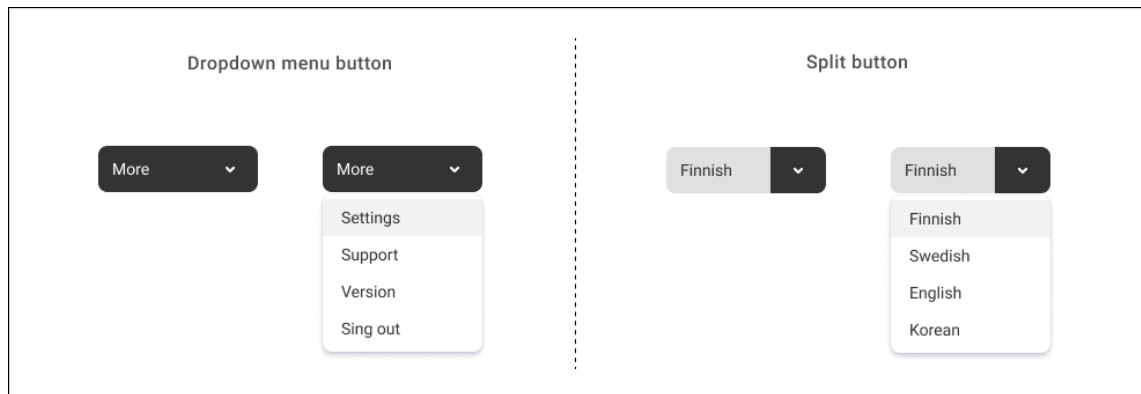


FIGURE 2. Dropdown menu button and split button.

Another example of a group of buttons is toggle button. It holds two related options but only one action can be activated at once. Toggle button is often used for on/off switch that user can change settings. (Carbon Design System 2023). Hence toggle button is also called as switch button or toggle switch buttons in different design systems. Material Design 3 by Google include switch, whereas Atlassian Design System contains toggle Buttons. However, how they appear and functionate are almost identical (Figure 3) (Material Design 3 n.d.; Atlassian Design System 2023).



FIGURE 3. Comparison of two buttons, Switch from Material Design 3 and Toggle from Atlassian Design System.

3 ACCESSIBILITY OF BUTTONS

This chapter mainly follows WCAG 2.2 to examine how to achieve accessible UI buttons. The criteria and materials researched in this chapter lay the design foundation of the implementation in the following chapter of this thesis, as well as providing accessibility evaluation standards.

3.1 Target size

WCAG 2.2 provides two succession criteria (SC) for target size of interactive elements. One of them is SC 2.5.8: Target Size (Level AA) stating that the target size for pointer input should be a minimum of 24 x 24 CSS (Cascading Style Sheets) pixels except for specific situations. Pointer input here refers to devices that enable aiming and selection, such as mouse, pen, or finger touch contact. And specific situations include the following five cases:

- Spacing: If the target is smaller than the specified minimum size but maintains a 24x24 gap between targets.
- Equivalent: If the same functionality is ensured from other targets on the same page.
- Inline: If the target is placed within a sentence or its size is constrained by the height of the line.
- User agent control: If the target size is determined by the user agent.
- Essential: If the presentation of a specific target is crucial or legally required. (WCAG 2.2 2023)

Another success criterion is SC 2.5.5: Target Size (Level AAA). The standard tells that the target size for pointer input should be a minimum of 44 x 44 CSS pixels except for specific situations. The specific situations for exclusion are the same as mentioned earlier, except that the spacing criterion does not apply in this case. (WCAG 2.2 2023).

One point to note here is that the unit for the target size is with the unit of CSS pixel. According to WCAG, CSS pixel is described as a standard unit that is density independent within CSS for all lengths and measurements. However, according to Shechter, he argues that there is no such thing as CSS absolute

units. He states that CSS measurement units were traditionally classified as absolute units based on physical units like pixels, centimeters, inches. However, in the current web perspective where high-density screens prevail, all CSS absolute units have become disconnected from physical units and have transformed into different kinds of relative units. Through a simple experiment, he discovered that CSS units do not align with actual physical units. (Shechter 2021).

Looking at other resources providing recommendations for the target size beyond WCAG, Harley (2019) claims that target size must be minimum 1 cm X 1 cm (0.4in X 0.4in). Also, Harley points out that measuring dimensions in pixel doesn't work effectively with touch screens of various densities. (Harley 2019). Additionally, Material Design by Google describes that on-screen element for click or touch interaction should be at least 48 X 48 pixels which results in a physical size of approximately 9mm in various screen sizes. (Material Design n.d.).

3.2 Color contrast

Use of color contrast is very crucial to make contents perceivable for users including users with disabilities (WebAIM n.d.). WCAG 2.2 includes two standards about color contrast which evaluate ratio of relative luminances between foreground and background colors. W3C defines that "relative luminance is the relative brightness of any point in a color space, normalized to 0 for darkest black and 1 for lightest white" (W3C 2021).

To meet SC 1.4.3 Contrast (Level AA), text should have a contrast ratio of at least 4.5:1, while large text can have a contrast ratio of 3:1. WCAG 2.2 determines that large text refers to 18pt (24 pixel) or larger for normal weight font type, and 14pt (18.5 pixel) or larger for bold font type. Another enhanced standard, SC 1.4.6 Contrast (Level AAA) requires a contrast ratio of 7:1 from normal text and 4.5:1 from large text. Also, contrast ratio of 3:1 is a requirement for graphical object except those that the graphic is essential to understand the context. (WCAG 2.2 2023). As most of buttons include multiple visual elements, such as text, icon, background or outlining element and so on, contrast ratio between varied elements should comply each guideline accordingly. For instance, the contrast between button background and text should be 4.5:1 (unless large text) and the

contrast between button background and surface of the screen should be 3:1 (ButtonBuddy, n.d.).

However, questions have been raised on the unclear cases of contrast ratio values achieved from the current formula. Hollywood-based filmmaker with visual impairment, Andrew Somers, has consistently raised awareness about issues related to WCAG 2 contrast ratio. Eventually he is currently developing Advanced Perceptual Contrast Algorithm (APCA). APCA recognizes that human perception varies based on various factors and introduces a new approach by focusing on areas where brightness and darkness are not perceptually weighted. APCA emerges as a new alternative to provide better accessibility for color contrast. (Muth, 2022).

3.3 Color use

WCAG 2.2 includes SC 1.4.1 Use of color (Level A). The standard establishes that color shouldn't be used alone for conveying information, indicating an action, prompting a response, or distinguishing a visual element (WCAG 2.2 2023). In other words, color alone is not enough to be indicator of meanings for users who is visually impaired. In addition to colors, other elements such as clear text, labels, or graphic elements like underline, chevron, or icon help communicate the meaning that the element is trying to deliver. (University of South Carolina n.d.).

3.4 Button states

Material Design 2 defines states as "visual representations used to communicate the status of a component or interactive element". Also, it mentions that different states of a component shouldn't visually alienate the component, while it must have explicit visual aid that users can notice and understand different states of interactive components (Material Design 2 n.d.). States type given to UI buttons and explanations are below:

- Enabled: interactive state
- Disabled: non-interactive state
- Hovered: cursor staying over an interactive element
- Focused: highlighted state by a keyboard or voice command

- Selected: selected state by user's choice
- Pressed: pressed state with user's tap

Selected state is specifically used for UI buttons, such as radio and toggle switch buttons letting users select and present the selected value or state (Material Design 3 n.d.). Also, hovered state is not applicable for mobile elements, since mobile user interfaces do not have hover (Antunes 2021).

3.4.1 Focus state

Focus state is a highlighted state by a keyboard or voice command (Material Design 3 n.d.). Focus state needs high emphasis to let users be aware of which element they are interacting now (GOV.UK Design System n.d.). While some people think focus state is equivalent of hover state, focus state is much more important than hover state for people relying on keyboard or voice to navigate (House 2022). WCAG 2.2 SC 2.4.7. Focus Visible (Level AA) ensures that all interfaces operatable with keyboards include at least a mode where focus indicators are shown. It clarifies that enabling focus mode does not only help people relying on the keyboard, but also benefits people with attention limitations, short term memory limitations, or limitations in executive processes. (WCAG 2.2 2023).

Despite of existing assumption that using a default focus state from browsers would be a safe option to fulfill non-text WCAG contrast standard, default focus state from many browsers fails the accessibility conformance (House 2022). Another guideline, SC 2.4.13: Focus Appearance (Level AAA) explains how to make the focus state easily spotted with visualized examples. The standard clarifies two factors, minimum size and color contrast. According to WCAG 2.2, "Focus indicators must

- be at least as large as the area of a 2 CSS pixels thick perimeter of the unfocused state.
- be a contrast ratio of at least 3:1 between the same pixels in the focused stated and unfocused state".

To fulfill the minimum size of focus state, the easiest and most common way the guideline suggests is using a solid outline that is thicker than 2 CSS pixels. (WCAG 2.2 2023).

WCAG 2.2 includes visualized example to elaborate this standard (Figure 4). In the Figure 2, the first three focus indicators pass the criteria. Last one fails because the focus indicator doesn't pass the minimum area requirement. Also, the success criterion includes information about other shapes than rectangles, detailed description of meeting the contrast ratio required. (WCAG 2.2 2023).

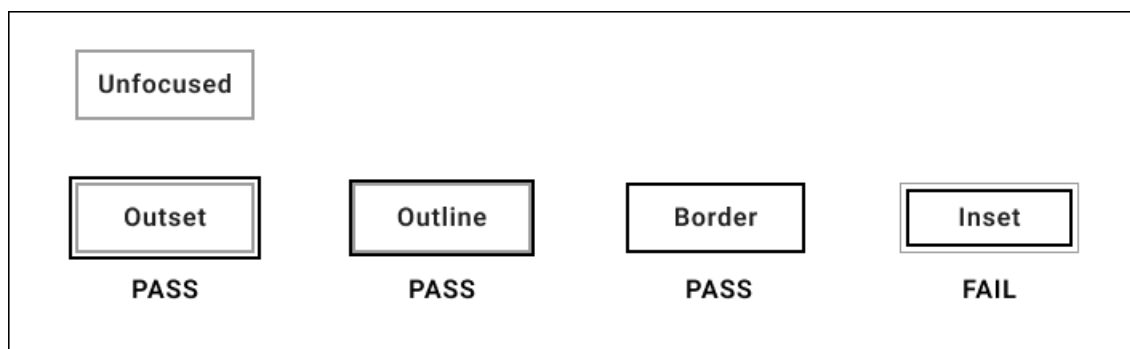


FIGURE 4. 4 Examples of focus state indicator from WCAG 2.2.

Another example of visualization of focus state can be from GOV. UK Design System that contains two colors with a high contrast ratio to be accessible with both light and dark background colors. In addition, they differentiate visual representations depending on elements. For example, link and input elements utilize multiple properties, such as double outlining, background colors, and use of underlining. (GOV.UK Design System n.d.).

3.4.2 Hover state

As mobile devices with touch screen don't have hover state, usage of hover states for digital services has been a topic of discussions. According to Wecreate.digital (n.d.), reasons why hover state should not be used are elaborated in terms of five perspectives.

- **Accessibility:** Mouse-hover is not available on touchscreen devices. Also, some physically impaired users may have difficulty to operate hover. Even users without any impairments can unintentionally trigger hover by just moving their cursor around.
- **Speed:** If a navigation menu is using hover, cursor moving away from a hover submenu will reset the navigation process and restart again. This will slow down users missing an item from the bottom of a submenu.

- User behavior: As clicking is used to apply diverse actions, clicking is more innate and natural web behavior. Besides, research has shown that people prefer clicking than hovering.
- Compatibility: Some web browsers, Internet Explorer and Microsoft Edge, have compatibility problems with hover states. Also, for touchscreen devices, double tap issues might happen to embrace hover.
- Futureproofing: Hovering can limit the capacity of future change and maintenance. If a submenu with hover changes to a new version without hover in the future, users would expect a submenu to appear when it's hovered. More importantly, hovering makes analyzing user behavior using heatmaps harder compared to clicks. (Wecreat.digital n.d.).

Kapanoglu (2022) recommends using hover state for enhancing the quality of the user's experience, never for main interaction flow (Kapanoglu 2022). As many people are using mobile devices with touch screen every day that doesn't have hovering interactions, I think hover states in desktop devices has been also decreased in comparison to earlier time.

3.5 Visual hierarchy

The visual hierarchy refers to structuring UI elements in order of importance of elements to effectively convey information and meanings of content. Visual hierarchy helps users navigate to the most important contents and allow users to understand flow of contents faster and easier. (Philips n.d.). Especially, if a user has a difficulty to stay focused due to attention deficit disorder, contents with clear visual hierarchy will help quickly understand the page, find key objects and complete tasks (Accessibility for Teams n.d.). Various visual factors utilized to create a visual hierarchy are followings:

- Size
- Color
- Contrast
- Typography
- Alignment and proximity
- Visual consistency (Loop 2023).

Color and contrast make certain elements stand out while others recede. Elements of higher importance should be designed to stand out to easily capture the user's attention. Changes in color, such as contrast and saturation, are frequently used to guide the importance of elements. (Gordon 2021).

Size is a significant factor building the visual hierarchy that communicates the relationship between elements and eventually meanings to users. For instance, Philip demonstrates that when there is a picture of a big bird sitting next to two small birds, the variation of size used here immediately communicate that they are parent and children, and family. (Philip n.d.). Larger elements tend to attract the user's attention more than smaller ones. Therefore making the most important element the largest can emphasize the most critical aspect of the content. (Gordon 2021). For best practice, Golden (2021) recommends keeping largest elements on a screen to maximum number of two to reinforce the hierarchy as well as using maximum three different sizes, small, medium, and large. (Gordon 2021).

Another element, alignment, serves to convey the orders between elements by connecting or separating multiple elements within a space. When an element deviates from the alignment dominating the space and disrupts the existing structure, it becomes noticeable and gains significance in contrast to the other elements. (Philips n.d.). Alignment is also closely linked to proximity. When related elements are aligned closely, and unrelated elements are relatively spaced apart conversely, it communicates the connection between elements (Cheung 2022). Golden (2021) mentions that grouping using proximity and whitespace helps users in navigating the screen easily and assists in identifying areas to focus or overlook (Golden 2021). There are more elements to create visual hierarchy in certain compositions, but the discussed elements were found repeatably among the reviewed sources.

In terms of the visual hierarchy in buttons, Material Design 3 provides a triangle to depict levels of button emphasis (Figure 5). High emphasis on the top is for the most important action on a screen. Medium emphasis in the middle is for important actions that doesn't attract attention from other elements. Lastly low emphasis on the bottom is for optional or supplementary options with the least importance among onscreen elements. And each level of button emphasis includes three button types. High emphasis contains expanded FAB, FAB, and filled button. Subsequently filled tonal button, elevated button and outlined button are sorted as medium emphasis. And text button, segmented button and icon button are in the low emphasis. (Material Design 3 n.d.).

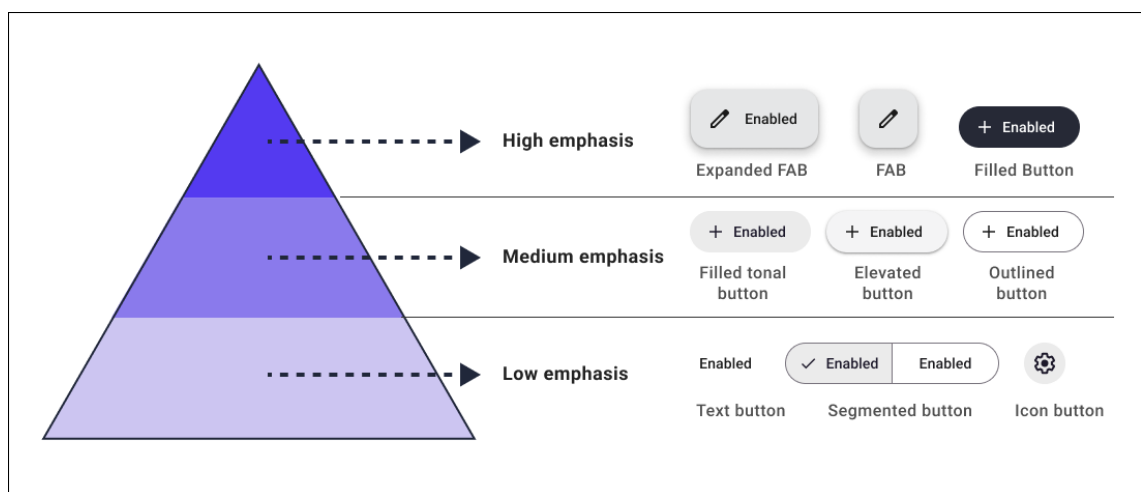


FIGURE 5. Button hierarchy by Material Design 3.

Material Design 3 recommends having only a single button for a primary action and communicate clearly different importance of buttons through hierarchy of on-screen elements (Material Design 3 n.d.).

3.6 Composition

According to O'Reilly media, composition is "a process of assembling a layout that consistently arranged components and contents on a page" (O'Reilly Media n.d.). This chapter will explore aspects that designers should consider in terms of accessible composition of UI.

3.6.1 Accessible composition

SC 3.2.3 Consistent navigation (Level AA) elaborates on how to structure navigation in page layouts. As buttons are involved in navigation, the guideline gives insights on how to place buttons and other UI elements accessibly. (WCAG 2.2 2023). The standard states that navigation repeated on many pages should be consistent with the same relative order. Here, the term "same order" does not mean that sub-navigation menus or secondary menus cannot be used. Instead, it intends to allow users to logically anticipate and interact more quickly by placing repeated content in the same location and same order. This guideline not only facilitates users employing magnification to easily navigate specific portions of the desired page but also enables other visually impaired users to quickly navigate through associative memory or visual cues. (WCAG 2.2 2023).

While this standard describes about navigation menus in headers and footers, SC 2.4.3 Focus order gives guidelines on how to build a logical order of websites. For example, when a website is navigated by using a tab key on a keyboard (which activates focus state mentioned in the chapter 3.4 Button states), interactive elements should be navigated in a logical order. If a website is configured to skip the middle content from the header and navigate directly to the footer when navigating via the tab key, it does not follow the logical flow of content. (WCAG 2.2 2023).

3.6.2 Interaction area

According to user experience design expert Steven Hoober, users prefer viewing content and engaging in touch interactions at the center of the screen. He emphasizes that the center of the screen is the most accurate area for touch interaction, while the outer edges or corners are the least accurate areas. Using a menu buttons placed in the outer corners as an example, he stated that many users touch around the button's center and some completely miss the touch target, leading to touch errors and inaccuracy. However, the mistakes can be reduced by spacing touch targets far apart from each other. (Hoober 2017).

Additionally, a research from Hooper(2017) that observed users operating a smartphone showed that users only tap after moving the object to the center of the screen. From this, he emphasizes two points:

- Place key content for reading or tapping in the center of the screen,
- Provide enough space at the bottom of the screen for scrolling so that users can interact with content in the center of the screen (Hooper 2017).

Another theory related to the interactive area is the Above the Fold theory. This theory originated from the early print industry, referring to the top half of newspapers or prints that were visible when folded in half. While it originally denoted the most visible portion, in modern web design it signifies the portion visible without scrolling. (Optimizely n.d.). Despite dynamic factors like devices, screen size, and resolution influencing this theory, Above the Fold remains crucial as a fundamental principle for effective communication in the digital space, especially in relation to the interactive area (Lara 2023). According to an eye-tracking study conducted by Nielsen Norman Group, out of 57,453 eye-tracking analyzed, significant amount of attention gap was observed depending on where contents are located. Contents allowed without scrolling received 102% more attention compared to contents required scrolling. (Schade 2015).

While there are plenty of resources emphasizing the importance of design in the Above the Fold area, there has been limited in-depth research on how this affects accessibility. However, a relevant aspect found is endless scrolling. Extensive scrolling with massive amount of content can cause issues for users with assistive technologies. When navigation and search options are primarily placed in a section that do not require scrolling, users utilizing screen readers or keyboard commands may face difficulties in navigating and searching for specific content. In such cases, to enhance accessibility, incorporating additional buttons that allow users to limit automatic infinite scrolling to view more content is recommended. Additionally, websites with page pagination are considered the most accessible option. (Bureau of Internet Accessibility 2021).

4 IMPLEMENTATION

Implementation of the thesis was to design accessible UI button components for a new mobile application project from Cargotec Finland Oy. I pursued to make reusable button components that can accommodate multiple variants in different style, states, and other configurations. I also assured to make button components that pass at least AA level of WCAG 2.2 in order to not violate EU (European Union) regulations. As the design system of the thesis partner was referenced by Material Design 2 and 3, implementation was complied with design principles from Material Design. I used a prototyping tool, Figma, for the implementation.

The concept of the project was to design a mobile application for crane operators where they can register their machine, get support on maintenance and operator targeted services. The application included comprehensive services, such as maintenance service notification for registered equipment, service location search, equipment error code search, instructional videos, practical checklist for maintenance, a link for fleet management service and online service chat.

As target users of the project were bound to use the mobile application in various environment where situational disability can be experienced, the service should be accessible. Therefore, I designed accessible UI buttons following the research findings. The service will be offered in two color modes, light and dark by utilizing color tokens. Details about color and multiple modes implementation will be included in 4.3 color chapter below.

4.1 Button component in Figma

To make buttons reusable and consistent throughout the project, I used a feature of Figma, component. Component is to make a reusable object that also allows to combine multiple variants into a single component. After generating a component, it can be duplicated to an instance. Instances have connection to its component. When changes are made to a component, instances will also get updates through the connection of its own component. (Figma Learn n.d.). Types of buttons that I implemented as components for the project are following:

- Filled buttons,
- Outlined buttons,
- Text button,
- Icon buttons,
- Dropdown menu buttons.

I designed a button component with component properties applied (Figure 4). The component properties are Figma's component features to operate components in a flexible way that a variant encompasses diverse properties settings. There are four kinds of component properties available:

- Variants
- Boolean
- Instance swap
- Text

By using them, the component can reduce the number of variants thus make it more convenient to use. (Figma Learn n.d.). In the Figure 6, it shows each variant involved in the button component. While the component includes only twelve variants, the number of buttons that can be made from the component is significantly more than the number of variants. Because diverse combinations of component properties are derived from a variant.

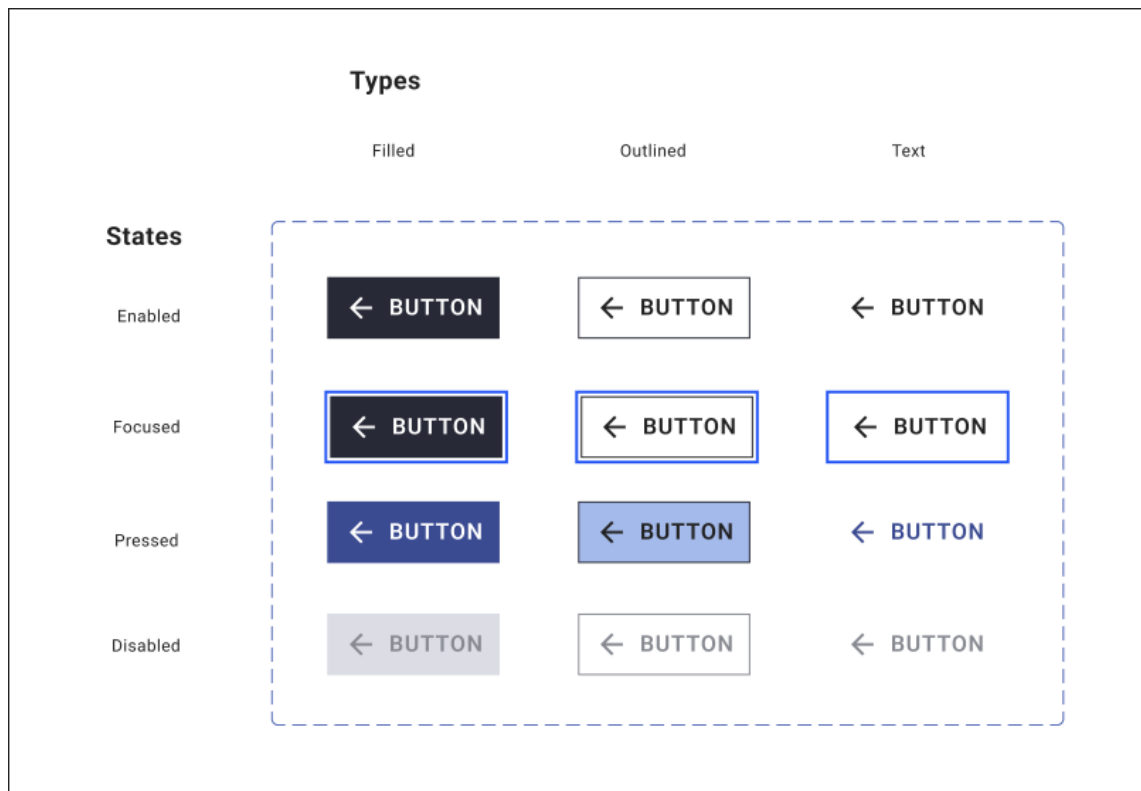


Figure 6. Button component including twelve variants.

Figure 7 shows how many component properties were applied for the button component. There are two variants, one is for types of buttons including three values, filled button, outlined button, and text button, and the other one is for states of buttons involving four values, enabled, focused, pressed, and disabled. I also set toggle property for text and icon which allows to show or hide them. Following that, I applied instance swap property that icons in variants can be switched to other icons without detaching instances from the component. Lastly there is text property for changing text of button easily to match the context.

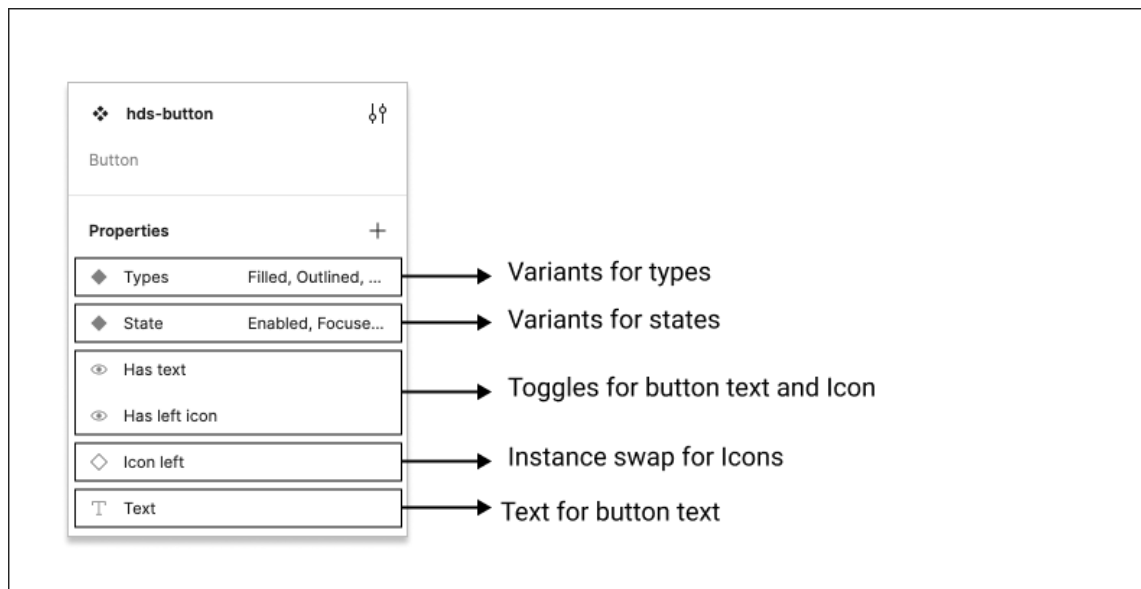


FIGURE 7. Component properties setting from the Button component.

Pre-defined options of component properties were easily accessible from each instance to tailor to different needs of buttons. In this way, icon buttons and icon buttons with text were generated from the same variant by configuring component properties. When I made an instance from the button component, the instance inherited a default appearance which is normally the first variant made. Then, I customized property settings. In practice, as many buttons were derived from a component, it streamlined to set identical values for paddings, colors, outlining, and size as well as assuring to keep buttons consistent.

Figure 8 demonstrates how component properties appeared in the settings of an instance. On the most left side, there is an instance from button component. On the right side, there is settings of the instance in where selected options can be replaced by other pre-defined options from the button component.

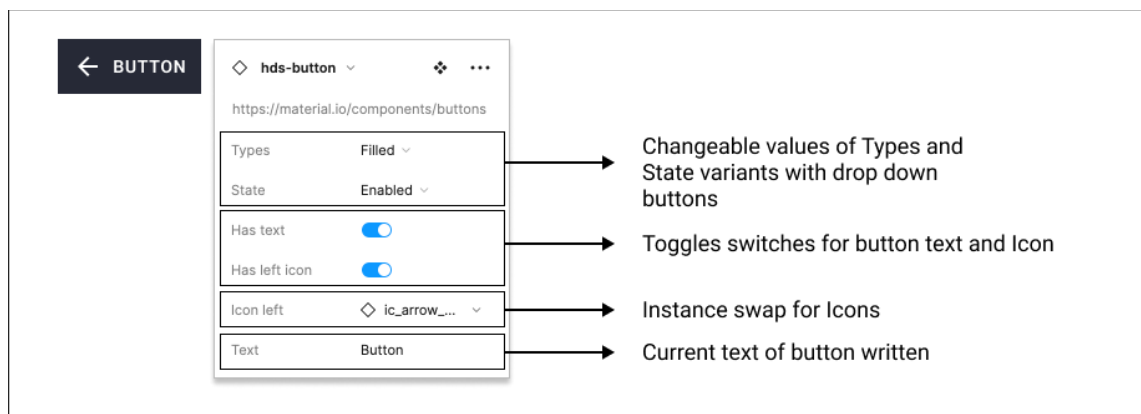


FIGURE 8. An instance of the button component.

Figure 9 shows a dropdown menu button consisting of two separate components for each button and list. The dropdown menu button component includes two groups of variants for types and states while the list component has variants for only types to easily support different number of list items.

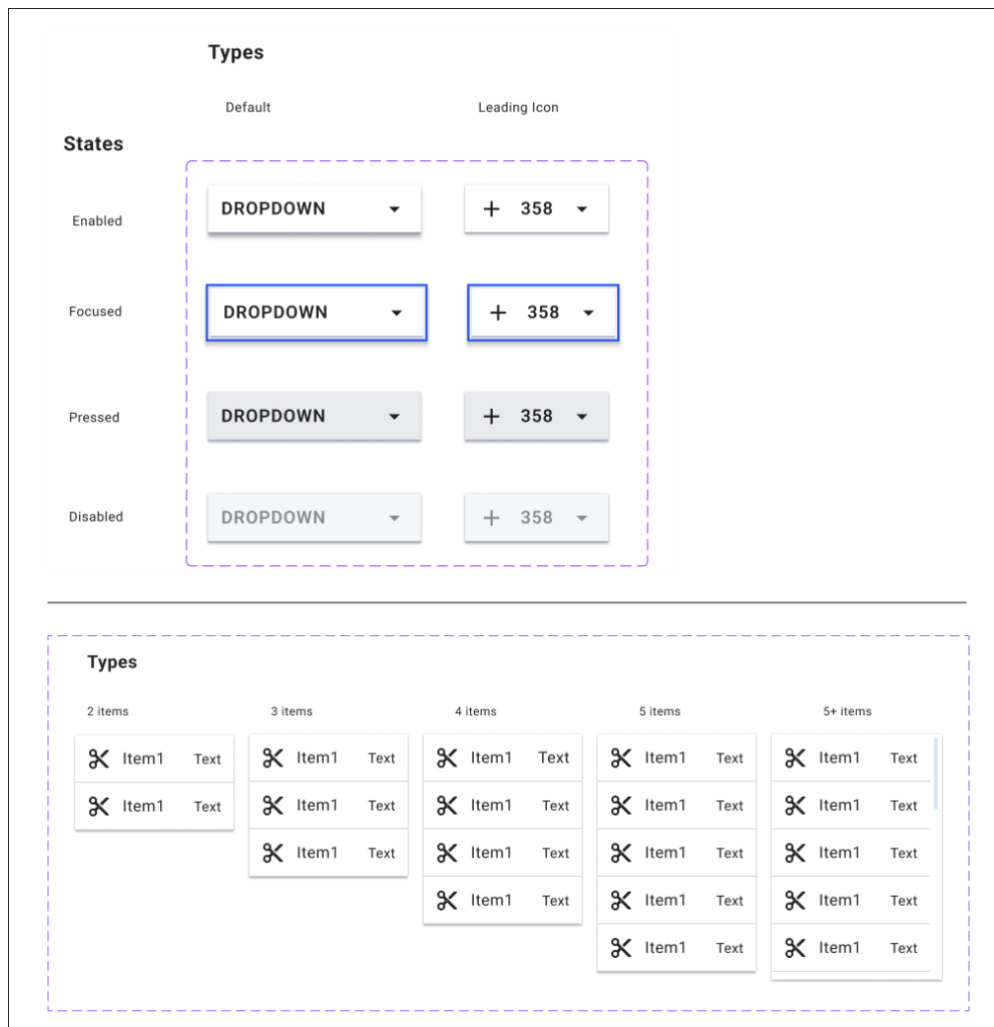


FIGURE 9. The dropdown button component and the dropdown button list component.

I designed the dropdown menu button the similar way as the button component by using component properties. Figure 10 demonstrates an example of the dropdown menu button and instance settings from both button and list. I created a dropdown menu button with three listed items by making an instance of the dropdown menu button and list components. To keep them attached, auto-layout function in Figma was applied.

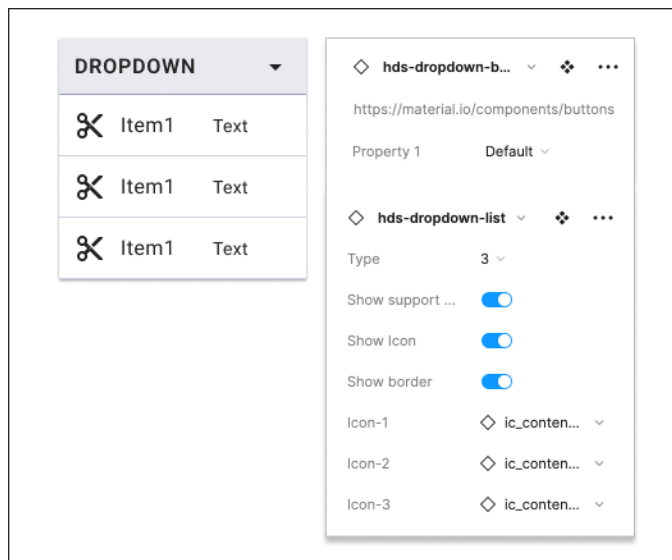


FIGURE 10. Examples of Dropdown menu button and instance setting from both Dropdown menu button component and Dropdown menu list component.

4.2 Size

To comply with WCAG2.2 SC 2.5.5: Target size, I assured to design buttons bigger than 44 X 44 CSS pixels. WCAG recommends applying even bigger size than the minimum size of the standard under given circumstances, for instance an applicable case is when an interactable object is often used or it is in difficult areas to reach. (WCAG 2.2 2023).

The size of buttons containing text meets the WCAG touch target size guidelines with a minimum dimension of 80 X 44 pixels. This includes 12 pixels padding on each side of the text. Consistent application of this padding value across all buttons aimed to ensure an adequate touch target size and promote a unified button design throughout the whole project. Additionally, icon buttons were sized as 48 X 44 pixels. The width of 48 pixels was happened due to the same padding values applied to the whole variants of the button component. As using identical padding values can provide a slightly more generous touch target size as well as maintaining the whole button consistency, icon buttons became rectangles with a size of 48 pixels by 44 pixels. (Figure 11).

The horizontal width of dropdown menu buttons can be adjusted based on content by setting the width configuration to fill container. The width of list items in the dropdown menu button can be adjusted to match the width of the dropdown

menu button, while the height follows the Material Design 3 guidelines, 48 pixels. I produced the initial chart format of Figure 9 with a Figma plugin, Spectral (Spectral n.d.).

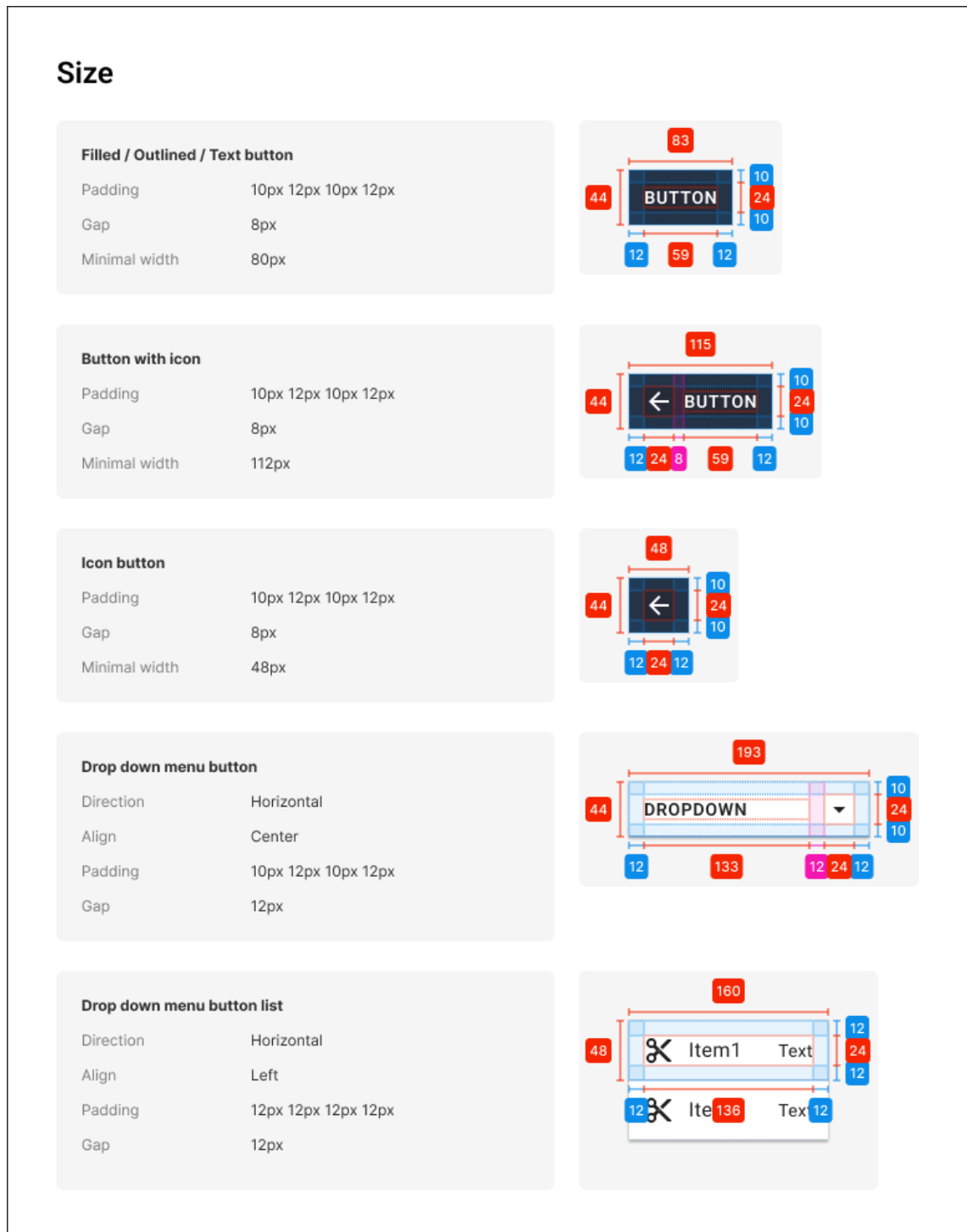


FIGURE 11. Size of each button.

4.3 Color

In terms of colors, a goal of this project was to create an accessible color palette for buttons in two color modes, light and dark. I designed the color palette for buttons to contain primary colors, cool gray colors, communication colors, and surface colors. The primary, cool gray, and communication colors were comprised of 12 colors each. Out of the 12 colors from each color palette, the darkest six shades were used for the light mode, while the remaining six lighter shades were used for the dark mode. I made sure that 12 colors must undergo color contrast ratios check with both text and background colors. Following WCAG guidelines, the contrast ratio must be verified between text and background, as well as between the foreground color of non-text graphic. Consequently, I designed each color palette to include two colors that only pass the AA level of color contrast standard, and all other colors that pass the AAA level of color contrast. The two shades that only meet the AA criteria are the brightest shades among the six for light mode and the darkest color among the six for dark mode (Figure 10).

In addition, the surface colors are a set of colors for the background, providing colors for background colors and background of UI elements in different states. I set 10 colors for each light mode and dark mode. To create background colors that harmonize with the primary colors, I selected colors by following a method provided in Material Design 3. The background colors also underwent contrast inspection between both high-contrast text color and medium-contrast text color. I designed the high-contrast text color to pass the WCAG AAA criteria for colors contrast, while the medium-contrast text color was guaranteed to meet at least WCAG AA criteria. (Figure 12).

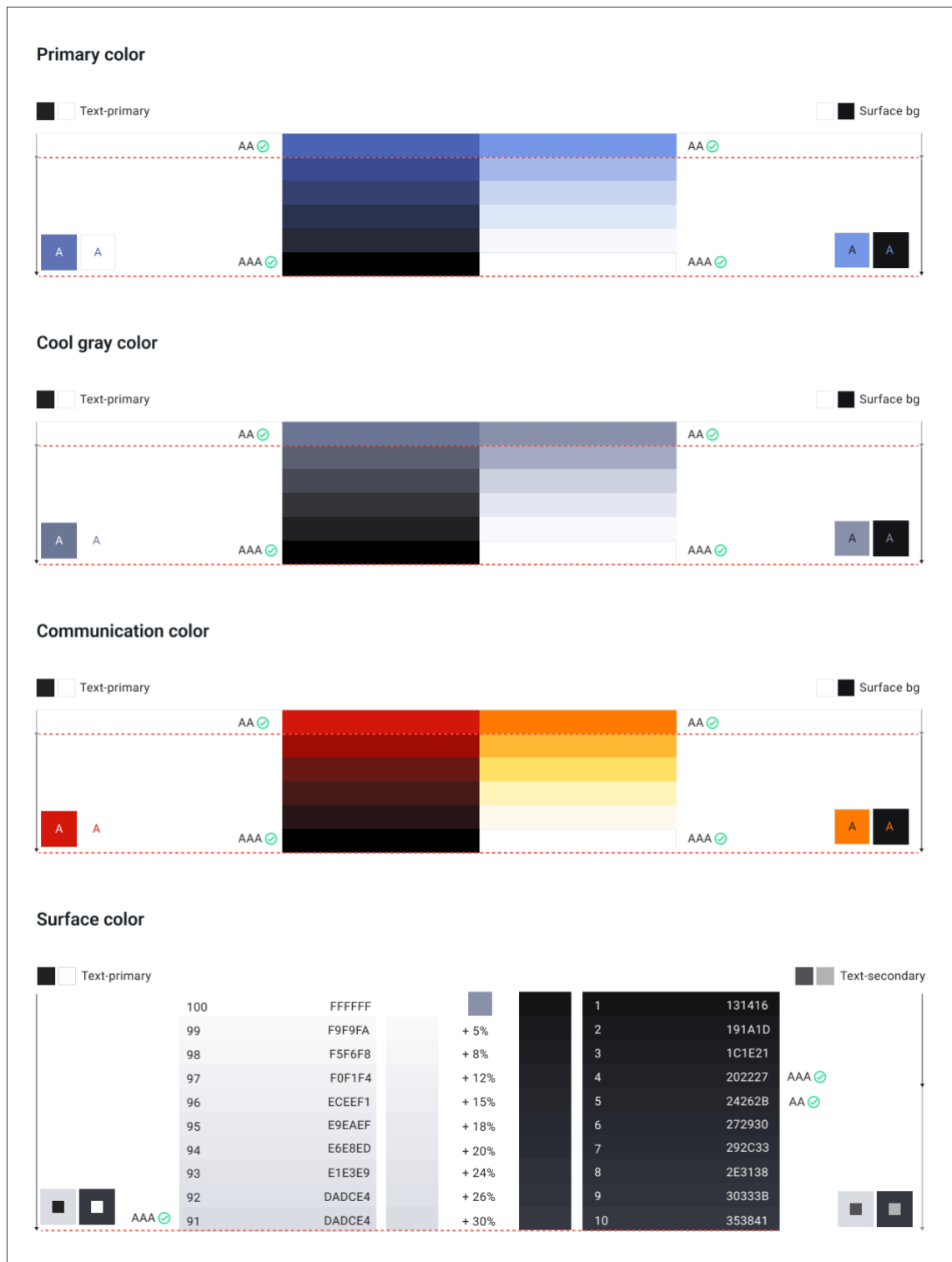


FIGURE 12. Color palette including primary color, cool gray color, communication color, surface color.

I applied variables, a feature of Figma, to efficiently implement multiple color modes (Figure 13). Figma states that “Variables in Figma store reusable values that can be applied to all kinds of design properties and prototyping actions” (Figma n.d.). Variables allowed not only to store sets of colors for multiple modes

but also to switch between different modes with a click. Figure 14 shows the button component in both light and dark modes. Figure 15 demonstrates the dropdown menu button and dropdown menu button list components in both light and dark modes.

Tokens		Name	Light	Dark	
All variables	77				+
text		ski-button-enabled	color/primary/p-10	color/primary/p-99	
surface		ski-button-focused	color/primary/p-10	color/primary/p-90	
border		ski-button-pressed	color/primary/p-40	color/primary/p-90	
button		ski-button-disabled	color/surface/s-92	color/surface/s-9	
chips		ski-button-border	color/primary/p-10	color/primary/p-99	
ui		ski-button-border-focus	color/primary/p-55	color/primary/p-55	
keyboard		ski-button-outlined-fill	color/primary/p-70	color/primary/p-20	
		ski-button-comms-enabled	...r/communications/c-50	...r/communications/c-50	
		ski-button-comms-border	...or/communications/c-0	.../communications/c-100	
		ski-button-comms-pressed	...or/communications/c-0	.../communications/c-100	

FIGURE 13. Variables for 2 modes.

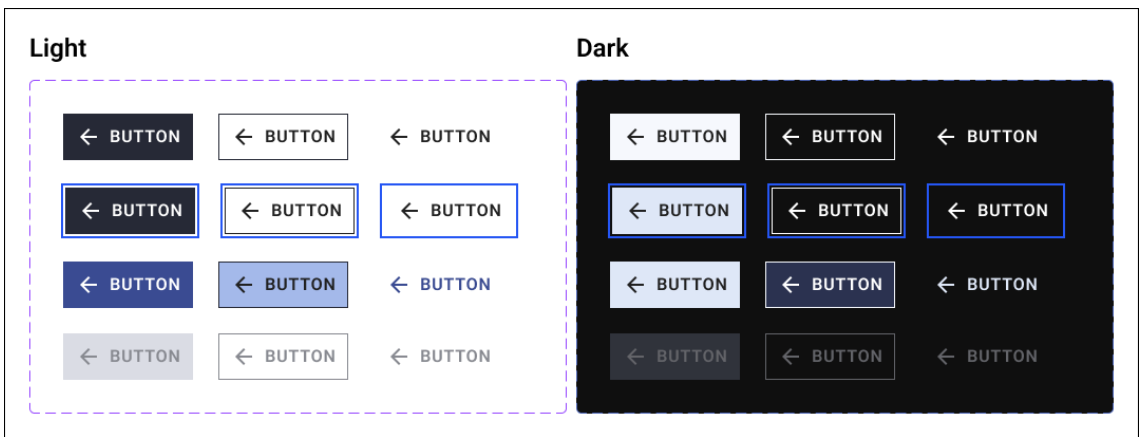


FIGURE 14. Button component in light and dark modes.



FIGURE 15. Dropdown menu button component in light and dark modes.

4.4 State

The button component included four states, enabled, focused, pressed, and disabled. I designed each state to have clear visual cues to indicate the state. Especially, as SC 2.4.13: Focus Appearance from WCAG 2.2 gives improved guidelines on how to design the appearance of focus states for compliance with succession of AAA level, focus state was designed accordingly by having a perimeter outline of two pixels thickness. Hover states were not included in the button component since hover doesn't exist in mobile applications. Also, selected status was not required for buttons designed for this project, but it is required for other buttons such as radio button or toggle switch buttons.

In the practice of designing focus states, I assured that the color contrast ratio between the same pixels from focused state and other states is more than 3:1 by using a Figma plugin, Contrast (WillowTree n.d.). As a result, I picked a vivid blue color as perimeter outline of focus state to solely use for indication of the focus state in all elements. Its contrast ratio with the same pixel of other status is more than 3:1 in the dark mode and 4.5:1 in the light mode (Figure 16).

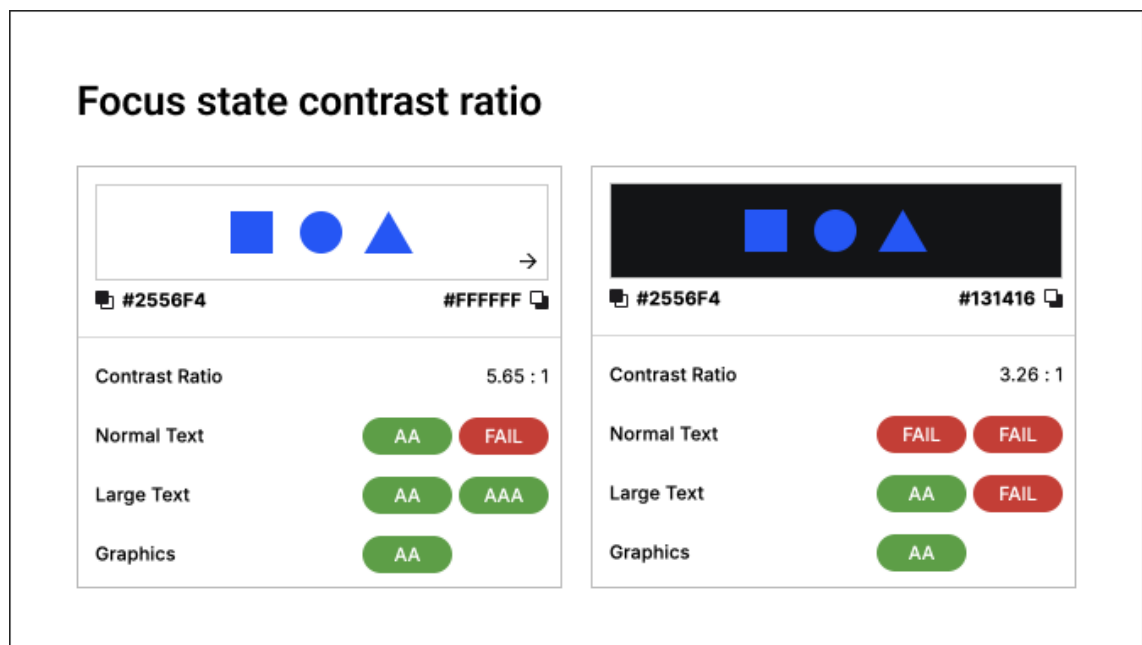


FIGURE 16. Focus state color contrast ratio passing 3:1 with both background colors in light and dark modes.

4.5 Visual hierarchy

Among the various methods to create visual hierarchy identified in the previous research, I firstly applied contrast on visual representation of buttons. A primary button, representing the most important action, was set as a filled button that has highly contrasting background color to make it the most noticeable. Additionally, when multiple buttons appeared on a page, only one action was emphasized as the primary button with obvious visual affordances. I decided styles and types for other buttons on the same page by reflecting the importance of buttons. For example, while filled button has primary action, outlined or text buttons that have lower emphasis contained secondary options.

Also, I differentiated the size of UI elements and texts. Varied sizes for text were used based on the importance of elements. By assigning scaling typography, I tried to navigate users to key text smoothly while ensuring readability of all texts. Lastly, I also applied alignment and proximity principle to distinguish related content to help users quickly grasp the relationship of content. For instance, when a button appears after a related contents, the button was placed closer to the contents than other UI elements following. Furthermore, I adopted proximity principle for arranging entire screens. (Figure 17).

Also, I applied consistent vertical and horizontal padding values between UI elements to build a coherent design throughout different pages. Basic vertical and horizontal gaps between components were set as 16pixel, and minimum 8pixel. When separated section occurs on a same page, 24pixel or 32pixel was used depending on the previous component.

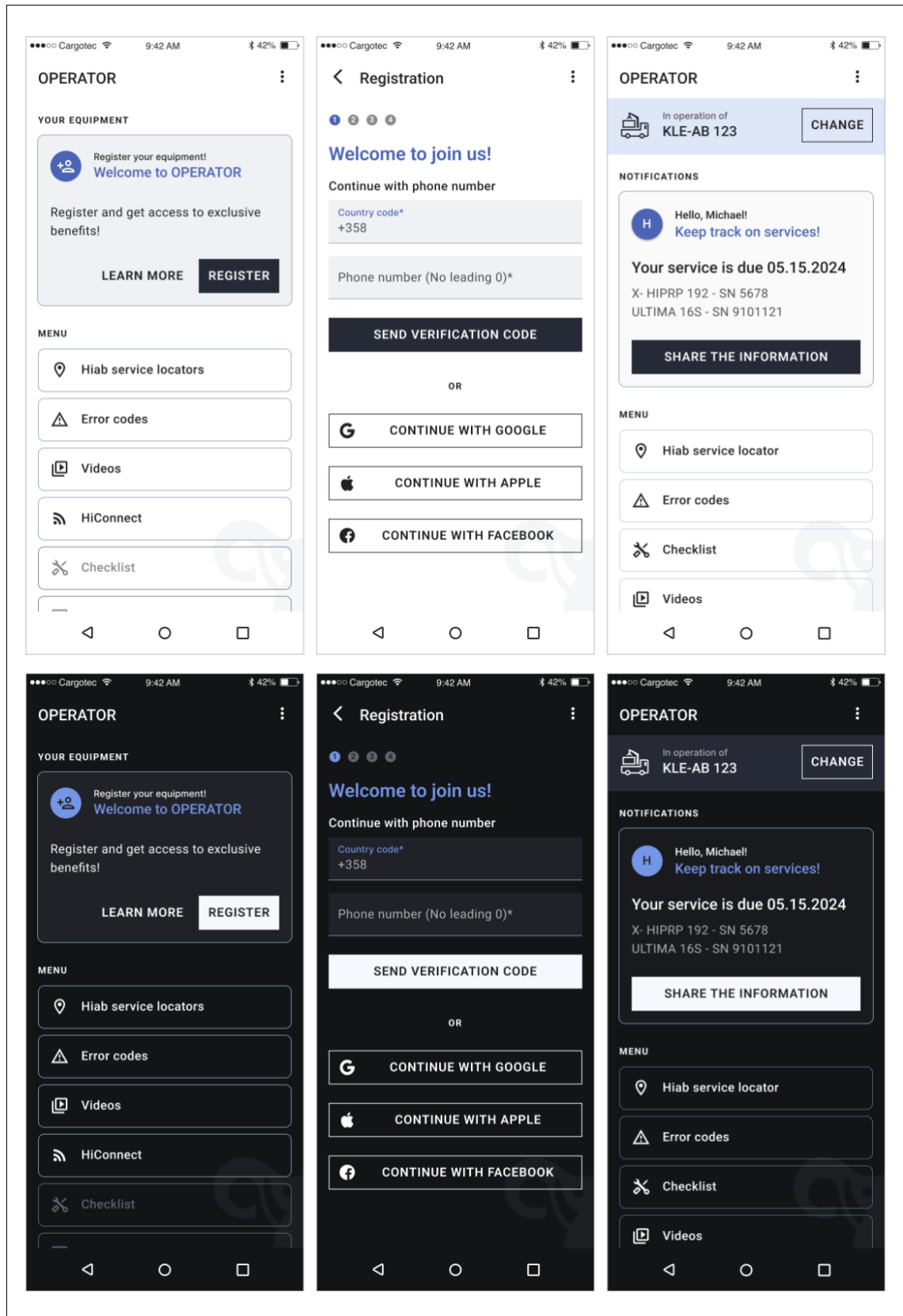


FIGURE 17. Clear button hierarchies for primary buttons in 3 different pages in both light and dark modes.

4.6 Composition

To comply with WCAG 3.2.3 Consistent Navigation (Level AA), I kept repetitive navigation appearing on multiple pages in an identical order (Figure 18; Figure 19). The main navigation menus in this mobile application was placed on the initial home screen with an icon menu button showing the dropdown menu consistently at the top app bar. I also ensured that another essential navigation button, the back button, was always positioned on the left app bar by purpose that users become familiar with it and be able to predict its location easily. Since the implementation of WCAG guideline 2.4.3 Focus order (Level A) is mainly handled from a development side, there wouldn't be significant design considerations in this regard.

Based on research on interactive areas, I made sure that key content was read or tapped from the center of the screen. Pages requiring scrolling provided sufficient space at the bottom to allow users to scroll the content to the middle of the screen for interaction. Additionally, in line with the Above the Fold principle, I intended to avoid pages that require excessive scrolling. Instead, I designed a pagination on the top of pages.

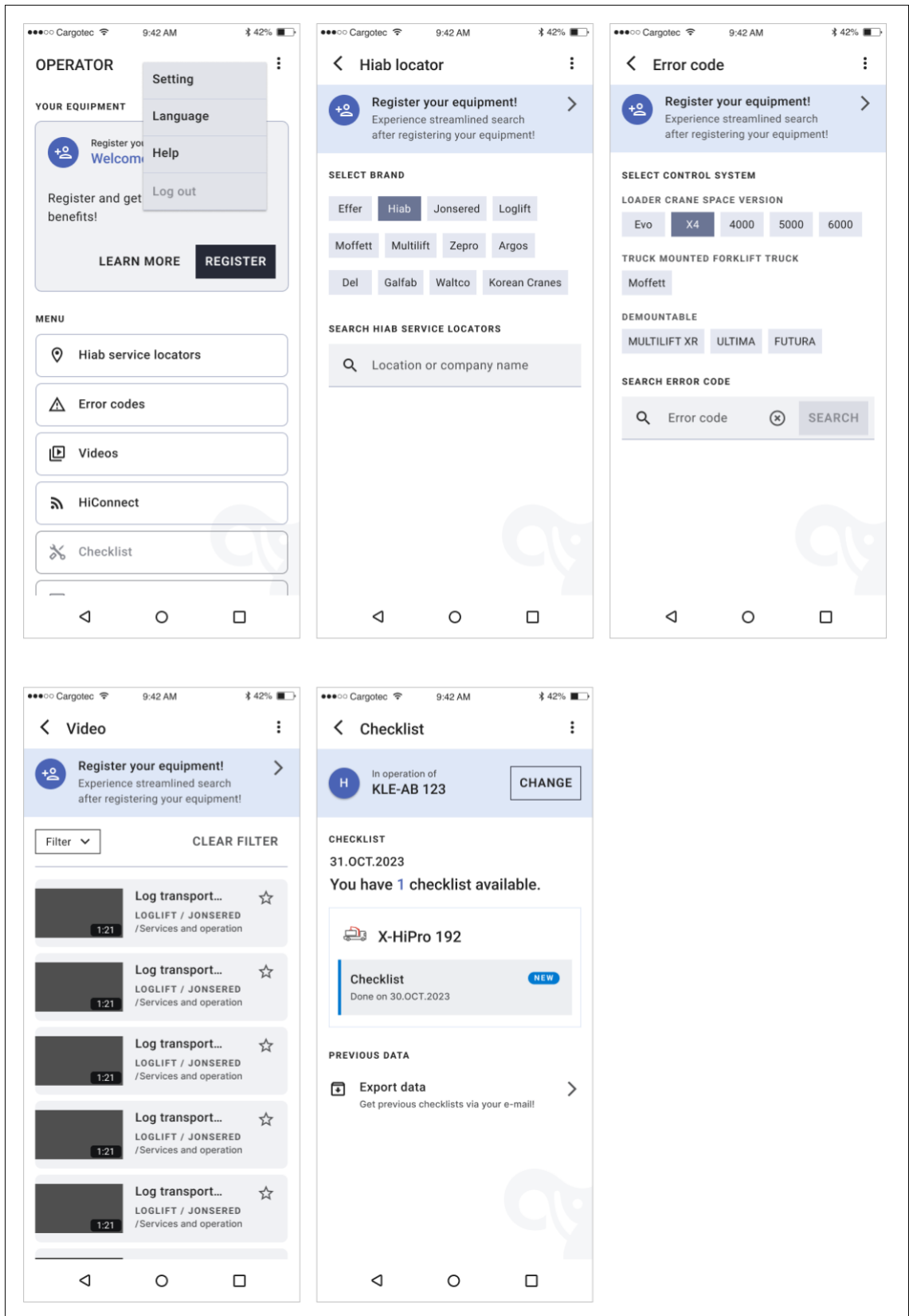


FIGURE 18. Consistent navigation of an app bar from different pages in the light mode.

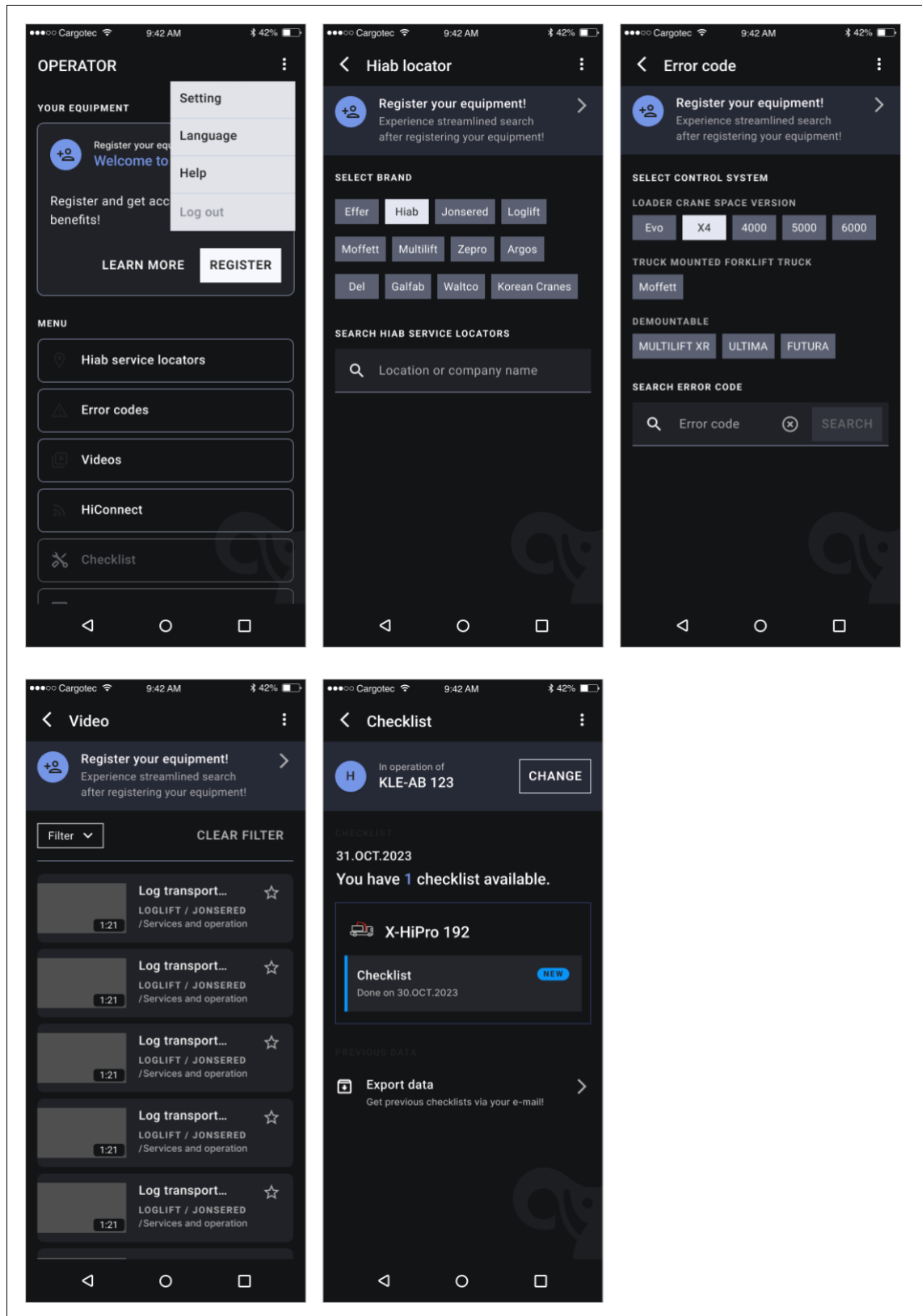


FIGURE 19. Consistent navigation of an app bar from different pages in the dark mode.

5 DISCUSSION

In the preliminary research, I examined a total of six aspects that could impact the accessibility of buttons and identified the corresponding WCAG criteria (If available). The implementation of buttons ensured compliances with at least WCAG AA level (Minimum) and aimed to achieve WCAG AAA level (Enhanced) if it was feasible.

To meet WCAG 2.2 SC 2.5.5: Target size (level AAA), I designed touch target size to be larger than 44 X 44 CSS pixels. In terms of color, I adopted variables, Figma's feature, to conveniently implement and maintain two modes, light and dark. Additionally, all colors applied to buttons were designed to comply with WCAG 2.2 SC 1.4.3. Contrast (Level AA), with a contrast ratio between text and background at 4.5:1, and between background and graphic objects at 3:1 or higher.

For the focus state, I assured to comply with guidelines, WCAG 2.2 SC 2.4.7. Focus visible (Level AA) and WCAG 2.2 SC 2.4.13: Focus appearance (level AAA). Regarding visual hierarchy, I could not find specific WCAG guidelines specifying about visual hierarchy. However, principles such as contrast, scaling, and alignment, were employed in various ways to help users capture key contents and eventually navigate smoothly according to prior research findings. Lastly, composition navigation menus were consistently placed to meet WCAG 2.2 SC 3.2.3 Consistent navigation (Level AA). I aimed to minimize endless scrolling and facilitate user interaction to occur at the center of the screen.

Utilizing component properties in Figma helped me produce consistent UI button components in an efficient way. Also, variables in Figma streamlined producing and maintaining multiple UI modes. As this thesis only includes how to enhance accessibility of UI buttons, subsequent research about enhancing accessibility of other UI elements is required in the future. Also, user tests that analyze improvement of accessibility based on the research findings will be beneficial for further studies.

The design discussed and thesis have been applied in a mobile application of my thesis partner. It will be launched in April 2024, and it will be available from Google Play and App Store as native mobile phone applications. Moreover, the design and the underlying theories became an essential part of their design system, which the division of my thesis partner use for their next designs across all screen-based user interfaces.

REFERENCES

- Abernathy E. 2011. A Brief History of the Push Button Light Switch. Released on 02.6.2011. Read on 12.4.2023. <https://www.historichouseparts.com/History-of-Push-Button-Light-Switch.html#:~:text=John%20Homes%2C%20a%20prolific%20English,a%20halway%20or%20room%20entrance>.
- Accessibility for Teams. n.d. Layout and hierarchy. Read on 2.12.2023. <https://accessibility.digital.gov/visual-design/layout-and-hierarchy/>
- Atlassian Design System. n.d. Toggle. Read on 2.5.2023. <https://atlassian.design/components/toggle/examples>
- Audioeye. 2022. A Brief History of Digital Accessibility. Released on 14.7.2022. Read on 2.4.2023. <https://www.audioeye.com/post/digital-accessibility/>
- Big Human. 2023. The History of Skeuomorphic Design. Released on 10.5.2023. Read on 11.12.2023. <https://www.bighuman.com/blog/guide-to-skeuomorphic-design-style>
- Bureau of Internet Accessibility. 2021. Is Infinite Scrolling Accessible? Released on 5.1.2021. Read on 14.12.2023. <https://www.boia.org/blog/is-infinite-scrolling-accessible>
- Bureau of Internet Accessibility. 2023. Situational Disabilities: Why Web Accessibility Impacts Everyone. Released on 14.9.2023. Read on 17. 11. 2023. <https://www.boia.org/blog/situational-disabilities-why-web-accessibility-impacts-everyone>
- ButtonBuddy n.d. Contrast Requirements for Buttons. Read on 10.12.2023. <https://buttonbuddy.dev/>
- Cambridge Dictionary. n.d. Read on 28.3.2023. <https://dictionary.cambridge.org/dictionary/english/accessibility>
- Carbon Design System. 2023. Button. Updated on 8.12.2023. Read on 20.6.2023. <https://carbondesignsystem.com/components/button/usage/>
- Carbon Design System. 2023. Dropdown. Updated on 20.6.2023. Read on 24.6.2023. <https://carbondesignsystem.com/components/dropdown/usage/>
- Carbon Design System. 2023. Toggle. Updated on 8.12.2023. Read on 14.12.2023. <https://carbondesignsystem.com/components/toggle/usage/>
- Cheung J. 2022. What is Visual Hierarchy in UX? 7 Key Principles. Updated on 26.8.2022. Read on 10.10.2023. <https://careerfoundry.com/en/blog/ux-design/what-is-visual-hierarchy/#connect-elements-with-alignment-and-proximity>

Contrast. 2023. WillowTree. Updated on 21.6.2023. Read on 9.4.2024.
<https://www.figma.com/community/plugin/748533339900865323/contrast>

DeRouchev B. 2010. History of the Button. Released on 28.3.2010. Read on 12.4.2023. <https://www.slideshare.net/billder/history-of-the-button-at-sxsw>

Equalize Digital. n.d. Text Too Small. Read on 13.7.2023.
<https://equalizedigital.com/accessibility-checker/text-too-small/>

European Commission. 2023. Web Accessibility Directive – Standards and Harmonization. Updated on 28.11.2023. Read on 8.12.2023. <https://digital-strategy.ec.europa.eu/en/policies/web-accessibility-directive-standards-and-harmonisation>

Fessenden T. 2018. Scrolling and Attention. Released on 15.4.2018. Read on 14.12.2023. <https://www.nngroup.com/articles/scrolling-and-attention/>

Figma Community. n.d. Spectral. Read on 14.12.2023.
<https://www.figma.com/community/plugin/1177722582033208360/spectral-designdoc>

Figma Learn. n.d. Create components to reuse in designs. Read on 14.12.2023.
<https://help.figma.com/hc/en-us/articles/360038663154>

Figma Learn. n.d. Explore component properties. Read on 14.12.2023.
<https://help.figma.com/hc/en-us/articles/5579474826519-Explore-component-properties>

Gordon K. 2021. Visual Hierarchy in UX: Definition. Released on 17.1.2021. Read on 10.10.2023. <https://www.nngroup.com/articles/visual-hierarchy-ux-definition/>

GOV.UK Design System. n.d. Understanding Focus State Styles. Read on 10.9.2023. <https://design-system.service.gov.uk/get-started/focus-states/>

Harley A. 2019. Touch Targets on Touchscreens. Released on 5.5.2019. Read on 10.10.2023. <https://www.nngroup.com/articles/touch-target-size/>

Hooper S. 2017. Design for Fingers, Touch, and People, Part 1. Released on 6.3.2017. Read on 27.12.2023.
<https://www.uxmatters.com/mt/archives/2017/03/design-for-fingers-touch-and-people-part-1.php>

House D. 2022. Focusing on Focus States. Released on 23.9.2022. Read on 10.9.2023. <https://iknowdavehouse.medium.com/focusing-on-focus-states-7c64089d805c>

Kapanoglu S. 2022. Hover is Dead. Released on 10.6.2022. Read on 25.9.2023. <https://ssg.dev/hover-is-over-5ca728a01cde>

Križanović I. 2020. Cell Phone History: From the First Phone to Today's Smartphone Wonders. Released on 4.8.2020. Read on 15.6.2023.
<https://versus.com/en/news/cell-phone-history>

Lara C. 2023. Above the Fold vs. Below the Fold: Does it Matter in 2023. Updated on 13.11.2023. Read on 14.12.2023.
<https://www.theedigital.com/blog/fold-still-matters>

Laubheimer P. 2019. Split Buttons: Definition. Released on 12.5.2019. Read on 14.12.2023. <https://www.nngroup.com/articles/split-buttons/>

Loop. 2023. Key Principles of Visual Hierarchy in UX Design. Released on 4.9.2023. Read on 10.10.2023. <https://www.loop11.com/key-principles-of-visual-hierarchy-in-ux-design/>

Material Design 2. n.d. States. Read on 5.9.2023.
<https://m2.material.io/design/interaction/states.html>

Material Design 2. n.d. Touch Target. Read on 10.10.2023.
<https://m2.material.io/develop/web/supporting/touch-target>

Material Design 3. n.d. All Buttons. Read on 15.10.2023.
<https://m3.material.io/components/all-buttons>

Material Design 3. n.d. Icon Buttons. Read on 22.6.2023.
<https://m3.material.io/components/icon-buttons/overview>

Material Design 3. n.d. Menus. Read on 24.6.2023.
<https://m3.material.io/components/menus/overview>

Material Design 3. n.d. States. Read on 5.9.2023.
<https://m3.material.io/foundations/interaction/states/overview>

Material Design 3. n.d. Switch. Read on 2.5.2023.
<https://m3.material.io/components/switch/overview>

Muldrew E. 2019. Design, How and Why it Evolves: Skeuomorphism to Flat UI. Released on 5.7.2019. Read on 15.6.2023. <https://blog.prototypr.io/design-how-and-why-it-evolves-skeuomorphism-to-flat-ui-a3a0f49d0f07>

Muth L. 2022. It's Time for A More Sophisticated Color Contrast Check for Data Visualizations. Released on 12.1.2022. Read on 10.12.2023.
<https://blog.datawrapper.de/color-contrast-check-data-vis-wcag-apca/>

O'Reilly. n.d. Composition Principles. Read on 12.11.2023.
<https://www.oreilly.com/library/view/designing-mobile-interfaces/9781449318451/ch01.html>

Optimizely. n.d. Above the fold. Read on 14.12.2023.
<https://www.optimizely.com/optimization-glossary/above-the->

W3C. 2021. Relative Luminance. Updated on 7.2.2021. Read on 8.12.2023.
https://www.w3.org/WAI/GL/wiki/Relative_luminance

W3C. n.d. Web Content Accessibility Guidelines (WCAG) 2 Level AAA conformance. Read on 12.3.2024. <https://www.w3.org/WAI/WCAG2AAA-Conformance>

W3C. Web Content Accessibility Guidelines (WCAG) 2.1. 2023. Abstract. Updated on 21.9.2023. Read on 28.12.2023. <https://www.w3.org/TR/WCAG21/>

WCAG 2.2 2023. Understanding SC 1.4.4: Resize Text (Level AA). Updated on 7.12.2023. Read on 15.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/resize-text>

WCAG 2.2. 2023. Understanding SC 1.4.1: Use of Color (Level A). Updated on 7.12.2023. Read on 12.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/use-of-color>

WCAG 2.2. 2023. Understanding SC 1.4.3: Contrast (Minimum) (Level AA). Updated on 7.12.2023. Read on 15.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum.html>

WCAG 2.2. 2023. Understanding SC 2.4.13: Focus Appearance (Level AAA). Updated on 7.12.2023. Read on 12.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/focus-appearance>

WCAG 2.2. 2023. Understanding SC 2.4.3: Focus Order (Level A). Updated on 7.12.2023. Read on 21.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/focus-order.html>

WCAG 2.2. 2023. Understanding SC 2.4.7: Focus Visible (Level AA). Updated on 7.12.2023. Read on 12.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/focus-visible.html>

WCAG 2.2. 2023. Understanding SC 2.5.5: Target Size (Enhanced) (Level AAA). Updated on 7.12.2023. Read on 14.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/target-size-enhanced.html>

WCAG 2.2. 2023. Understanding SC 2.5.8: Target Size (Minimum) (Level AA). Updated on 7.12.2023. Read on 14.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html>

WCAG 2.2. 2023. Understanding SC 3.2.3: Consistent Navigation (Level AA). Updated on 7.12.2023. Read on 21.12.2023.
<https://www.w3.org/WAI/WCAG22/Understanding/consistent-navigation.html>

WCAG 2.2. 2023. Understanding SC 2.5.5: Target Size (Enhanced) (Level AAA). <https://www.w3.org/WAI/WCAG22/Understanding/target-size-enhanced.html>

WebAIM. n.d. Contrast and Color Accessibility. Read on 10.12.2023.
<https://webaim.org/articles/contrast/>

Wecreate.digital. n.d. Menu Clicks VS Hover – Why Hover is A Concept You Should Not Use. Read on 27.12.2023. <https://wecreate.digital/blog/menu-clicks-vs-hover/>

Wills M. 2021. When the Push Button Was New, People Were Freaked. Released on 11.5.2021. Read on 12.4.2023. <https://daily.jstor.org/when-the-push-button-was-new-people-were-freaked/>

Wunder. n.d. Hover, Focus, Active. Read on 25.9.2023. <https://wunder.io/wunderpedia/accessibility/accessible-uis/hover-focus-active/>

Zumbrunnen A. 2015. Hover States. Released on 20.11.2015. Read on 25.9.2023. <https://azumbrunnen.medium.com/hover-states-6dc01b085e6>