2021 IEEE Symposium on Visual Languages and Human-centric Computing (VLHCC2021), 10-13 October, St Louis, USA

A Toolkit for Building More Adaptable User Interfaces for Vision-impaired Users

Calvin Luy, Jeremy Law, Lily Ho, Richard Matheson, Tracey Cai,
Anuradha Madugalla, John Grundy
Department of Software Systems and Cybersecurity, Monash University, Australia
{calvinluy, jeremylaw3535, lilyyho98, richardmatheson.rm, Traceykycai}@gmail.com
{anu.madugalla, john.grundy}@monash.edu

***Abstract*—Most prior research into adaptable and adaptive user interfaces primarily focuses on facilitating consistent user experiences across devices and pays less attention to facilitating universal access for diverse end users. We address this shortcoming by developing adaptable user interface components for a category of diverse users, the vision impaired. This paper presents a framework that supports run time adaptation of web components to suit vision impaired users by using a set of adaptable, reusable widgets. We developed a prototype using these components and evaluated its effectiveness. Our results show that an such an adaptable user interface provides significant benefit in many key W3C accessibility areas, helping to make the web more accessible for diverse end users.**

***Index Terms*—Human computer interaction, Adaptive user interface, Accessibility, Visual impairment, Personalised UI**

I. Introduction

Universal access to digital technology has been a long term challenge [1]. *Adaptive user interfaces (Adaptive UI)* automatically tailor the interface to user preferences based on their predicted intent, by capturing and analysing user inputs and behaviour [2]–[5]. *Adaptable user interfaces (Adaptable UI)* allow users to tailor the UI components to their needs [6], [7]. *Diverse users* include users are those with vision impairment, cognitive impairments, aging population as well as users from socially and culturally diverse backgrounds [8]–[10]. Frameworks to support this include Intelligent User Interfaces [4], Adapt-UI [11] and OOUIC [6]. These provide insight into a design model for automatically adapting the user interfaces at runtime to allow for dynamically changing context of the user. Developing adaptable UIs for vision impaired users has received significantly lesser attention. Existing solutions include responsive web designs (RWD), adaptable interfaces and promoting best practices such as following accessibility guidelines. However, poor implementations have led users to experience greater dissatisfaction when navigating sites than with conventional designs [12], [13]. Adaptable interfaces allow users to select from a range of predefined configurations that best suit them. Accessibility guidelines provide guidance on common difficulties vision impaired users face, but can only guide developers who generally lack understanding of the vision impaired [14], [15].

We address this problem by developing and evaluating a set of reusable, adaptable web interface components for a subset of users *people with vision impairments,* aiming to improve accessibility. In this study, we focus on users with color blindness, low vision as well as users with dyslexia. By integrating our framework to an existing website, we provided a high level of adaptation especially using adaptable color schemes, font settings, layout and image settings. We evaluated our framework via a user study using personas of people with varying levels of vision impairment.

As user interfaces (UIs) become increasingly complex, developing a single user interface with a "one size fits all" approach is no longer sufficient [3], and leads to compromises due to conflicting user requirements [16]. As a result, usability aspects of most software tend to focus on homogeneous characteristics common to majority of users and neglect features and support required by special users [16]. A possible solution is to use OS specific adaptable features, such as iOS font, colour customisation [17]. However, this changes entire devices' font and colour settings and is not user-friendly. Another would be to use web browser plugins e.g. for colour filtering [18]. Their performance varies greatly between web sites, and impacts whole web sites. Therefore, much more nuanced, user controlled UI adaptation is needed.

We analysed the popular Zomato website which aggregates information and reviews on restaurants and food delivery providers [19]. We evaluated its website against the WCAG 2.0, which are the standard accessibility evaluation guidelines, using the WAVE Web Accessibility Evaluation Tool [20]. 92 errors were identified, such as 'low colour contrast' and 'small fonts' issues. These demonstrate the limitations this site had for accessibility limitations of vision impaired users. The view of this website for a user with Deuteranopia is shown in Figure 1, highlighting colour contrast and small text issues. It can be argued that browsers inbuilt zoom-in feature would be of assistance to the small text issue. But as shown in Figure 2, this would reduce the visible area leading to navigation problems and disorientation [12]. We tried out browser plug-ins and laptop and mobile device accessibility settings to try and support users with these challenges. While some support was helpful, much failed or didn't directly address key problems. This was due to (i) lack of user control over many settings; (ii) failure of the adapted application to display properly e.g. layout, interaction; and (iii) inability to control individual interface parts. Some web site components did not seem built to support accessibility at all.

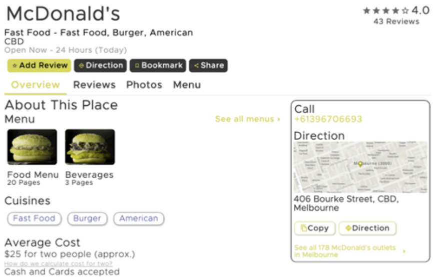

Fig. 1: Deuteranopia filtered Zomato – green-blind user view

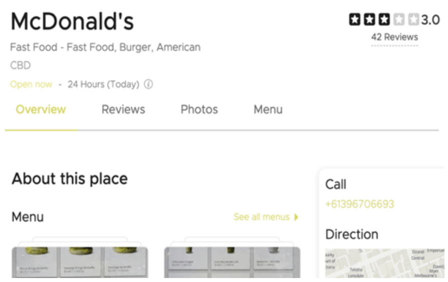

Fig. 2: Zoomed via browser leading to layout issues

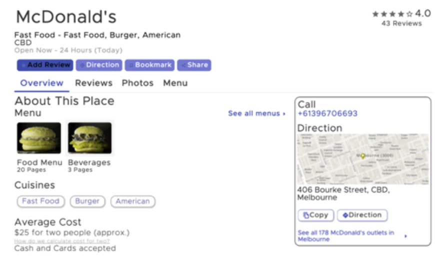

Fig. 3: Adaptive Zomato – text size increased, colour changed

II. Methodology

We wanted to answer the following questions: RQ1: What limitations surround current frameworks that support adaptive and adaptable user interfaces? RQ2: How can we determine the degree and type of accessibility required by diverse end users? RQ3: When used, does our adaptable web application impact user accessibility?

A. Our Approach

The UI adaptation requirements for the vision impaired vary with the degree of vision issues, resulting in an overwhelming number of combinations to be catered for [14]. Therefore in our work, we focused only on two key customisations: colour and font. We chose these since they have the highest impact and are most visible. We provide a customised set of UI components in open-source libraries and an architectural framework. We designed and implemented a framework that grants the user autonomy to modify their UI based on their individual needs. Contrary to the automated *adaptive* interface approach, we created a manual *adaptable* interface where users are able to set fine-grained options that modify the UI to better suit their individual needs.

B. Adaptable UI Framework and Components

We utilised Flutter as the basis of our adaptable framework [21]. Flutter is an open source software development toolkit that uses the Dart programming language to programmatically build front end UIs. This is an SDK that is increasingly becoming popular among web developers due to it's ease of use. Therefore our solution explores the possibility of using this SDK to ensure adaptability. Flutter interfaces are composed of components, called 'widgets'. Widgets are built when the application is first loaded, and are rebuilt when state changes. We leveraged this rebuilding ability to develop "adaptable Widgets" which can be used as replacements for some commonly used basic Flutter widgets. Composition of Flutter applications using our new suite of "Adaptive Widgets" and Adaptable Settings create an adaptable interface model. User (or in future, also automatic, adaptive) changes to the settings will modify the appearance of the UI at run time. The changes in the Adaptable Settings by the user in the Presentation Layer of the application, triggers changes of state in the Adaptive View Models that track the state of various accessibility settings. All Adaptive Widgets are designed to listen to changes in View Models, leading to an automated re-build of the Adaptive Widget in the case of changes.

C. Our Novel Adaptive Flutter Widgets

We built a set of Adaptive Widgets that change automatically based on user Adaptable Settings changes (which could also be changed automatically in future). They include *Adaptive Text, Adaptive Button, Adaptive Text Field, Adaptive Link*. We added functionality to listen to the View Model and incorporated changes to the widgets to account for the settings that are applicable for them. When building our Adaptive Text widget, we added 3 listeners (bold text, dyslexic friendly font and font size) to the Adaptive Widgets View Model. For example, if the state for bold text is changed to true, logic is applied within the widget modifying the existing text style. In our rebuild of the Button widget, we added a differentiate without colour listener by enabling rebuild of the widget to have all buttons underlined. Upon changing the state of Dyslexic Friendly font, the Adaptive Text in the Adaptive Button is modified to contain a dyslexic friendly font. Our adaptable widgets are used as one to one to replacements for and can be mixed with non-adaptive Flutter widgets.

D. Adaptable Settings Menu

Guided by W3C accessibility standards we selected Colour Themes, Font Settings and Image Colour Filters as the main sections of our accessibility menu. These menu options dictate the behaviour of our Adaptive Widgets.

1) Colour Themes: We implemented colour theme change for pages which automatically updates elements to a specific colour theme. *Predefined Colour Vision Deficiency Themes* improve the experience for those with a specific colour vision deficiency. There are three major types of color blindness with many sub-types [22]. We predefined four colour themes – red, green, blue and total colour-blind friendly [23]. When selecting one of these themes, the background, accent and text colour are automatically applied to all elements on the website, as shown in Figure 6. *Custom Colour Theme Selection* gives users full control to define their own custom colour theme. To assist users with selecting high contrasting colours, the accessibility menu provides a contrast ratio based on the selected primary and accent colours. The contrast ratio is

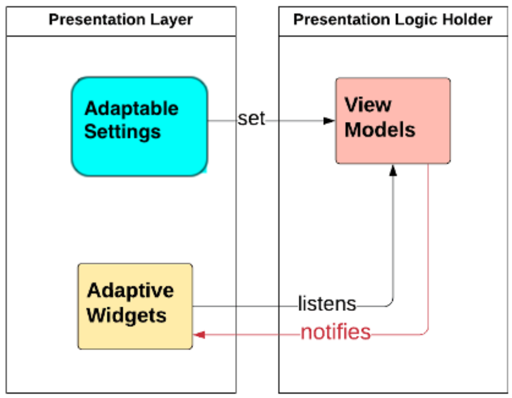

Fig. 4: Our Adaptable UI Framework

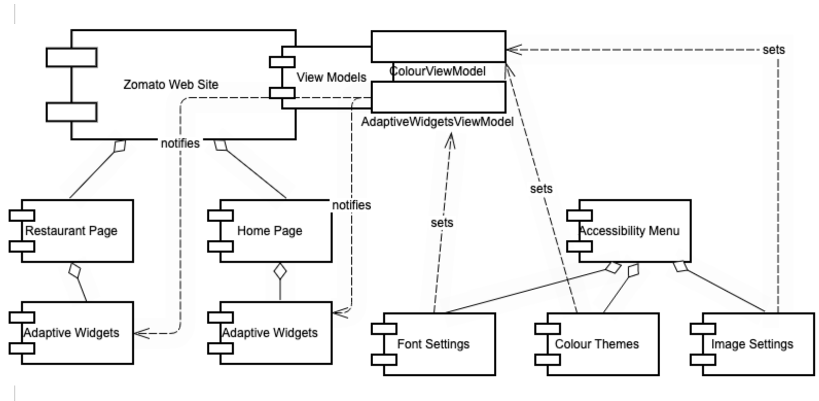

Fig. 5: Widget Tree of Adaptive Zomato Prototype

a measure of the difference in luminance of two colours, expressed as a ratio. The minimum contrast ratio required by the W3C standards is 4.5:1 for normal text.

2) Font Settings: Text Colour settings help people with colour vision deficiency who are often unable to read text on certain colour backgrounds [24]. We implemented the capability to change text colour of page body based on user preference.*Text Size* is another problem where it's difficult to perceive text and distinguish between different page elements. We implemented customisable font settings targeted at users with low vision, allowing them to bold text and adjust the default font size. We also allows users with better visual acuity to see more information by reducing font size. *Font type* focus on improve readability for users with common dyslexia symptoms. It enables applying a dyslexia friendly font: typeface OpenDyslexic, which provides weighted bottoms for each letter and unique letter shapes [25].

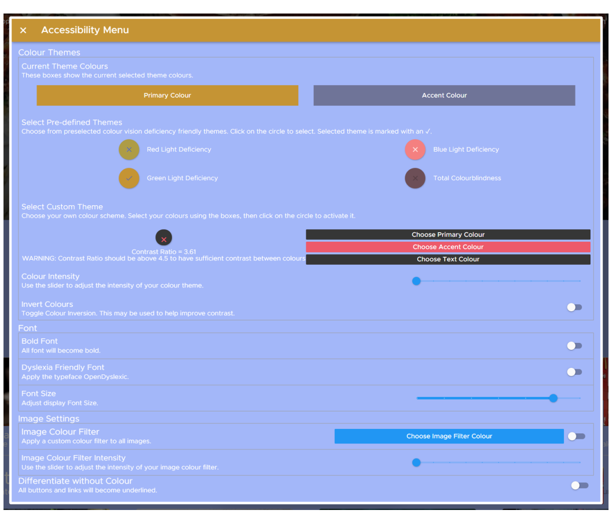

Fig. 6: Menu with predefined colour theme applied

3) Differentiate Without Colour: caters to those who cannot distinguish colours. We provide a switch to toggle the "differentiate without colour" option, making buttons and text links easily distinguishable with underlining.

4) Image settings: Our Image settings adaptations allow users to select an image filter colour to apply to all images, helping to better distinguish colours in images. This works by applying the inbuilt Flutter method BlendMode.modulate() which multiplies the colour components of the selected colour and the source image to produce the filtered image. We also allows users to adjust the intensity of the filter.

E. Adaptable UI Prototype

Figure 3 shows our rebuilt adaptive Zomato web site in use after colour, text and image adaptation configurations set via the menu in Figure 6. It shows text colour changed to a colour more suitable for a green-blind user. The stronger contrast is more suited to the user. Similarly, adjusting text size enables the user to see smaller text more easily without website layout being compromised. The user is still able to see the full page and its contents with larger fonts in each widget. The code can be downloaded from https://github.com/anukmd/toolKitAdaptiveUI

III. Evaluation

A. Method

Our evaluation, approved by our University Ethics Committee, used participants recruited by the authors from an advanced software engineering class. Each participant was assigned an author who conducted an evaluation session. The intended audience of this tool was people with colour blindness and reduced vision. However, due to COVID-19 and unavailability of working closely with the intended participants, we worked with participants with normal colour vision. We simulated colour blindness for them through the use of colour filter plugins. Fourteen participants partook in the evaluation. Table I show the distribution of participants and their simulated type of vision impairment.

To evaluate the accessibility of our prototype adaptable application, personas were assigned to each participant. As this application caters to a variety of colour blind users, the personas varied in their type of colour blindness, and variation between normal and shortsightedness. To aid personas, a Google Chrome colour vision simulator *Colorblindly* was

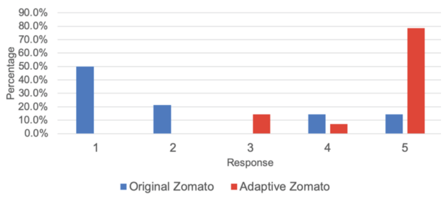

Fig. 7: Participant responses evaluating Contrast

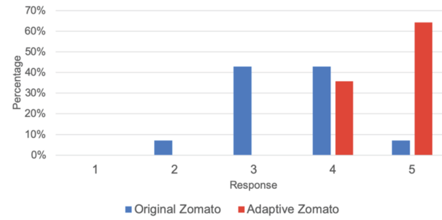

Fig. 8: Participant responses evaluating Colour

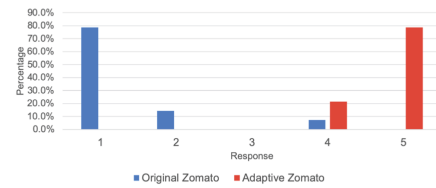

Fig. 9: Participant responses evaluating Text Size

applied to the web browser, allowing the participant to view the websites as their persona would.

TABLE I: Summary of Personas Participants Simulated

Type of Colour Blindness	Type of Partial Sightedness	Participants
Normal	Normal	4
Deuteranopia	Normal	4
Protanopia	Normal	2
Tritanopia	Normal	2
Monochromacy	Normal	2
Normal	Short Sighted	4
Deuteranopia	Short Sighted	4

A questionnaire was designed for the evaluation based on heuristic evaluation method. This was selected as the testing method for this study, as it helps to find usability and accessibility problems in UI designs [26]. The selected heuristics correspond to the W3C accessibility guidelines [27]

B. Results

Results from our heuristic evaluations indicated that the users perceived the recreated adaptive Zomato website to be much more accessible than the original website for almost all W3C vision impairment guidelines. Low scores in *Line Spacing, Line Length* and *Letter Spacing*, could be attributed to side effects of the available Flutter settings. For example, changing the font size will increase the line spacing, and enabling the dyslexia friendly font would increase letter spacing. *Capitalisation, Hyphenation, Justification* and *Margin and Borders* scores need further evaluation with more users to better understand the cause.

Contrast: Ability for a user to customise colours of web elements helps to distinctly differentiate elements by colour. Users had the option to choose between predefined colour blind friendly themes or picking their own colour combinations. Figure 7 results indicate that the original Zomato website performed poorly, with a mode value of 1. This was consistent across the respondents, as 70% of them rated the contrast 1 or 2. By comparison, our adaptive version of Zomato outperformed the original in terms of contrast, with approx 80% of respondents rating it a 5. This indicated that the adaptive interface was highly effective in allowing users to set background and text colour in order to improve contrast.

Differentiate without Colour: The ability to convey information, indicate an action or distinguish certain visual elements without a reliance on colour is a core W3C Accessibility guideline. When asked if "colour is not the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element" [27], the original Zomato website performed at an acceptable standard. As seen in Figure 8, 3 and 4 received highest responses with each of them making up 42.9%. This shows that it performed slightly above average in this category, and therefore, did not rely solely on colour to convey information. However results demonstrate that our adaptive Zomato outperformed the original with 64.3% rating it 5, and 36.7% as 4.

Text Size: Our menu allows users to change the font size of the screen text, without having to zoom in or out. Participants were asked to evaluate whether or not they could "change the text size of text, without zooming the entire interface" [27]. There was a large difference between the results, as illustrated in Figure 9. For the original Zomato website, 78.6% of the participants down voted the original website's performance. In contrast, our adaptable version showed the opposite results, with 78.6% rating the website a 5.

IV. Limitations

One of our study's limitations was the lack of real user involvement. This can be overcome in the future by conducting user studies with actual visual-impaired users. It would also help to overcome the limitation of possible bias in current participants since they were from a software engineering class. Once such a future work is completed, a comparison of task completion between real vs. simulated user groups may reveal interesting insights.

V. Summary

We prototyped a framework that grants the user autonomy to make detailed modifications to their web application interface based on their preferences. We designed, implemented and evaluated a set of web SDK components on top of Flutter to implement the framework. Participants found the functionality provided by our accessibility settings menu helpful for their simulated vision deficiency, enabling them to more easily view certain elements on the website. Future work needs to incorporate a range of vision impaired participants in order to gain a more accurate understanding of their perspectives when using adaptive user interfaces.

Acknowledgements

Grundy is supported by ARC Laureate Fellowship FL190100035, and Madugalla is supported by ARC Transformation Hub IH170100013.

References

[1] E. Machado, D. Singh, F. Cruciani, L. Chen, S. Hanke, F. Salvago, J. Kropf, and A. Holzinger, "A conceptual framework for adaptive user interfaces for older adults," in *2018 IEEE Int. Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2018, pp. 782–787.

[2] M. Peissner, A. Schuller, and D. Spath, "A design patterns approach to adaptive user interfaces for users with special needs," in *Int. Conference on Human-Computer Interaction*. Springer, 2011, pp. 268–277.

[3] K. Montague, V. L. Hanson, and A. Cobley, "Adaptive interfaces: a little learning is a dangerous thing..." in *Int. Conference on Universal Access in Human-Computer Interaction*. Springer, 2011, pp. 391–399.

[4] L. Rothrock, R. Koubek, F. Fuchs, M. Haas, and G. Salvendy, "Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms," *Theoretical issues in ergonomics science*, vol. 3, no. 1, pp. 47–84, 2002.

[5] D. Benyon and D. Murray, "Experience with adaptive interfaces," *The Computer Journal*, vol. 31, no. 5, pp. 465–473, 1988.

[6] L. Zhang, Q.-X. Qu, W.-Y. Chao, and V. G. Duffy, "Investigating the combination of adaptive uis and adaptable uis for improving usability and user performance of complex uis," *Int. Journal of Human–Computer Interaction*, vol. 36, no. 1, pp. 82–94, 2020.

[7] C. De Los Rios Perez, "Adaptable user interfaces for people with autism: A transportation example," in *Proc. Internet of Accessible Things*, 2018, pp. 1–2.

[8] H. Petrie, F. Hamilton, and N. King, "Tension, what tension? website accessibility and visual design," in *Proc. 2004 Int. cross-disciplinary workshop on Web accessibility (W4A)*, 2004, pp. 13–18.

[9] L. Levanthal, B. Teasley, D. Stone, A.-M. Lancaster, A. Marcus, B. Nardi, J. Nielsen, M. Kurosu, and R. Heller, "Designing for diverse users: will just a better interface do?" in *Conference companion on Human factors in computing systems*, 1994, pp. 191–192.

[10] J. Grundy, H. Khalajzadeh, and J. Mcintosh, "Towards human-centric model-driven software engineering." in *ENASE*, 2020, pp. 229–238.

[11] E. Yigitbas, S. Sauer, and G. Engels, "Adapt-ui: an ide supporting model-driven development of self-adaptive uis," in *Proc. ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2017, pp. 99–104.

[12] T. C. Nogueira, D. J. Ferreira, S. T. Carvalho, and L. O. Berreta, "Evaluating responsive web design's impact on blind users," *IEEE MultiMedia*, vol. 24, no. 2, pp. 86–95, 2017.

[13] T. do Carmo Nogueira, D. J. Ferreira, S. T. de Carvalho, L. de Oliveira Berretta, and M. R. Guntijo, "Comparing sighted and blind users task performance in responsive and non-responsive web design," *Knowledge and Information Systems*, vol. 58, no. 2, pp. 319–339, 2019.

[14] J. A. Jacko and A. Sears, "Designing interfaces for an overlooked user group: Considering the visual profiles of partially sighted users," in *Proc. 3rd Int. ACM conference on Assistive technologies*, 1998, pp. 75–77.

[15] A. P. Freire, C. M. Russo, and R. P. Fortes, "A survey on the accessibility awareness of people involved in web development projects in brazil," in *Proc. 2008 Int. cross-disciplinary conference on Web accessibility (W4A)*, 2008, pp. 87–96.

[16] J. Sauer, A. Sonderegger, and S. Schmutz, "Usability, user experience and accessibility: towards an integrative model," *Ergonomics*, no. just-accepted, pp. 1–23, 2020.

[17] "Use display and text size preferences on your iphone, ipad and ipod touch – apple support," shorturl.at/hluCI, (Accessed on 04/23/2021).

[18] "Colorblindly - chrome web store," shorturl.at/fnM35, (Accessed on 04/23/2021).

[19] "Mcdonald's, cbd, melbourne," shorturl.at/asxK6, 2008-2020, (Accessed on 12/07/2020).

[20] "Wave web accessibility evaluation tool," https://wave.webaim.org/, 2020, (Accessed on 12/13/2020).

[21] "Flutter - beautiful native apps in record time," https://flutter.dev/, 2020, (Accessed on 12/07/2020).

[22] "Types of color blindness, national eye institute," https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness/types-color-blindness, (Accessed on 04/23/2021).

[23] M. Geissbuehler and T. Lasser, "How to display data by color schemes compatible with red-green color perception deficiencies," *Optics express*, vol. 21, no. 8, pp. 9862–9874, 2013.

[24] J. T. Nganji and S. H. Nggada, "Disability-aware software engineering for improved system accessibility and usability," *Int. Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 47–62, 2011.

[25] J. J. Wery and J. A. Diliberto, "The effect of a specialized dyslexia font, opendyslexic, on reading rate and accuracy," *Annals of dyslexia*, vol. 67, no. 2, pp. 114–127, 2017.

[26] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proc. SIGCHI conference on Human factors in computing systems*, 1990, pp. 249–256.

[27] "Accessibility requirements for people with low vision," https://www.w3.org/TR/low-vision-needs/, March 2016, (Accessed on 12/06/2020).