

Annotated Bibliography: Accessibility UX/UI Design

Noelynn Faith Batalingaya

September 25, 2025

References

- [1] Abdulaziz Alshayban, Iftekhhar Ahmed, and Sam Malek. Accessibility issues in android applications: State of affairs, sentiments, and ways forward. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE)*, pages 1323–1334. IEEE/ACM, 2020.

This paper studies accessibility problems in more than one thousand Android applications. It also uses surveys and user reviews to understand what developers and users think about these problems. The authors found many common issues, such as missing labels and poor support for screen readers. The International Conference on Software Engineering is a top computer science conference, so the source is credible. For my project, this shows why accessibility problems are still common in real-world applications and why I need to include both design-time and runtime testing.

- [2] Lian Duan, Zhi Chen, Hao Liu, Xiang 'Anthony' Chen, and Yang Li. Mockup to feedback: Leveraging large language models for accessible design feedback. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 2024.

This study tests how large language models can give automatic feedback on design mockups. It shows how artificial intelligence can help catch possible accessibility and usability problems before the coding process begins. Since the paper is from the CHI Conference on Human Factors in Computing Systems, one of the top human-computer interaction conferences, it is very credible. For my project, it shows how automation can help early design reviews, but also reminds me that human checks are still needed because artificial intelligence can miss important details.

- [3] Lian Duan, Bowen Xu, Mingrui Zhang, Xiang 'Anthony' Chen, and Yang Li. Uicrit: A dataset and critique agent for user interface design feedback. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 2024.

This paper introduces UICrit, a dataset and tool for giving structured feedback on user interface designs. The goal is to make automated critiques more consistent and useful. The study is published at the ACM Symposium on User Interface Software and Technology, another respected human-computer interaction conference, so it is credible. For my project, it helps me understand both the potential and the limits of automated design critique. It supports my plan to combine automated checks with manual testing.

- [4] Júlia Holanda Muniz, Daniel Mesquita Feijó Rabelo, and Windson Viana. Assessing accessibility levels in mobile applications developed from figma templates. In *Proceedings of the 17th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, pages 316–321. ACM, 2024.

This article looks at mobile applications built from Figma templates and finds many built-in accessibility problems, such as poor color contrast and missing labels. It shows how design templates can build in problems that later carry over to the final applications. The International Conference on Pervasive Technologies Related to Assistive Environments is a peer-reviewed accessibility conference, so it is credible. This connects directly to my research, because it shows why checking Figma designs for accessibility issues is an important first step.

- [5] Hanxiao Zhang, Anhong Guo, Xiang 'Anthony' Chen, and Yang Li. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 2021.

This study shows how computer vision can recognize user interface elements and create accessibility metadata such as roles and labels. The authors argue that automated recognition can help screen readers describe applications better. The paper is from the CHI Conference on Human Factors in Computing Systems, a leading human-computer interaction conference, so it is credible. For my project, it shows why code-time validation is important, because some problems can only be found when the application is running, not just in the design phase.