

# 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] Yufei Chen, Han Wu, and Ruijie Wang. Assessing accessibility levels in mobile applications from figma templates. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 2024.

This paper audits mobile applications generated from Figma templates and demonstrates that many WCAG issues are inherited directly from template design choices. The study combines automated analysis with heuristic review. Credibility: CHI 2024, peer-reviewed and widely cited. Relevance: Reinforces my plan to perform design-time preflight checks since poor templates can institutionalize problems downstream. The combined automated and heuristic approach shows not only what tools flag but also what experts notice in context. A limitation is that user impact is inferred rather than measured, so I complement this with runtime evaluation using screen readers.

- [3] Wentao Duan, Yiheng Zhou, Chen Wu, and Xiang Anthony Cao. Generating automatic feedback on ui mockups with large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 2024.

This paper uses large language models to critique UI mockups, flagging issues such as contrast and spacing. The authors convert mockups into structured representations and evaluate LLM outputs against expert critiques. Credibility: CHI 2024, with empirical testing. Relevance: Serves as a baseline for my design-time preflight checks, showing what automated critique can and cannot achieve. Limitation: Model performance varies by prompt quality and cannot fully replace human accessibility review.

- [4] Wentao Duan, Yiheng Zhou, Chen Wu, and Xiang Anthony Cao. Uicrit: A ui critic agent and critique dataset. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '24)*. ACM, 2024.

The authors present UICrit, a dataset of expert UI critiques and a corresponding automated agent. The system outperforms baseline LLM critique tools by aligning with expert judgments. Credibility: UIST 2024, selective ACM venue. Relevance: Frames the limits of automated critique, reminding me to balance automation with manual inspection. Limitation: Dataset focus is on design quality broadly, not accessibility-specific issues.

- [5] Rony Ginosar, Hila Kloper, and Amit Zoran. Parametric habitat: Virtual catalog of design prototypes. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, page 1121–1133, New York, NY, USA, 2018. Association for Computing Machinery.

This paper reframes inclusive design as a practice centered on the lived experiences of marginalized users rather than as a checklist of requirements. Through case studies and design reflections, the authors argue that accessibility should be treated as a design philosophy emphasizing equity and empathy. Credibility: Presented at DIS, a top peer-reviewed HCI venue. Relevance: Helps me justify that my project is not just about guideline compliance, but about embedding inclusive design values early in workflows. The paper's strength is its practical framing: it translates values into concrete design moves that teams can adopt immediately. For my project, I will reflect these values by documenting accessibility rationale alongside each design token and component in Figma, then verifying that intent survives implementation. Limitation: The discussion is conceptual and does not evaluate specific tools, so I will supplement it with empirical studies.

- [6] Jingyi Huang, Walter S. Lasecki, and Liangjie Feng. Beyond the guidelines: Assessing accessibility in figma prototypes. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 2024.

The authors study methods for evaluating accessibility within Figma prototypes, showing how static design mockups can be scanned for issues like low contrast or missing labels. They emphasize that early-stage evaluation reduces costs and improves quality. Credibility: CHI 2024, a highly selective peer-reviewed conference. Relevance: Supports the design-time phase of my project

by showing how accessibility can be embedded before code is written. Methodologically, the work triangulates prototype artifacts with practitioner feedback to identify which checks are reliable at design-time. A limitation is that static mockups cannot capture interactive behaviors like focus order or announcements. I will plan a Phase 2 with code-level testing to cover those behaviors.

- [7] 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 '24)*, pages 316–321. ACM, 2024.

This paper examines mobile applications generated from Figma templates and evaluates their accessibility compliance. It finds recurring WCAG violations, especially with color contrast and missing labels. Credibility: PETRA 2024, a peer-reviewed assistive technology venue. Relevance: Supports my design-time preflight by showing how upstream template issues propagate to finished apps. Limitation: Relies heavily on automated evaluation, which may underrepresent experiential accessibility problems.

- [8] Wei Shi and Leah Findlater. “It Could Be Better, It Could Be Worse”: Understanding accessibility in ux practice with implications for industry and education. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, 2023.

Through interviews with UX professionals, this paper highlights barriers and opportunities for accessibility in real-world design practice. Findings show that organizational priorities, limited time, and lack of training often hinder accessibility integration. Credibility: Peer-reviewed CHI 2023 publication. Relevance: Connects directly to my project by showing how structural issues in industry justify stronger design- and code-time tools. Their interview sample provides rich detail on organizational pain points. A limitation is generalizability beyond the studied orgs; I mitigate this by grounding claims with my own defect dataset.

- [9] 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.