"I tend to view ads almost like a pestilence": On the Accessibility Implications of Mobile Ads for Blind Users

Ziyao He
ziyaoh5@uci.edu
University of California, Irvine
Irvine, California, USA

Syed Fatiul Huq
fsyedhuq@uci.edu
University of California, Irvine
Irvine, California, USA

Sam Malek
malek@uci.edu
University of California, Irvine
Irvine, California, USA

ABSTRACT

Ads are integral to the contemporary Android ecosystem, generating revenue for free-to-use applications. However, injected as third-party content, ads are displayed on native apps in pervasive ways that affect easy navigation. Ads can prove more disruptive for blind users, who rely on screen readers for navigating an app. While the literature has looked into either the accessibility of web advertisements or the privacy and security implications of mobile ads, a research gap on the accessibility of mobile ads remains, which we aim to bridge. We conduct an empirical study analyzing 500 ad screens in Android apps to categorize and examine the accessibility issues therein. Additionally, we conduct 15 qualitative user interviews with blind Android users to better understand the impact of those accessibility issues, how users interact with ads and their preferences. Based on our findings, we discuss the design and practical strategies for developing accessible ads.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in accessibility**; • **Social and professional topics** → *Assistive technologies.*

KEYWORDS

Android, Accessibility, Advertisement, Screen Reader

ACM Reference Format:
Ziyao He, Syed Fatiul Huq, and Sam Malek. 2024. "I tend to view ads almost like a pestilence": On the Accessibility Implications of Mobile Ads for Blind Users. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3597503.3639228

1 INTRODUCTION

Mobile apps play a vital role in people's daily lives, with millions of apps available in the market and billions of users relying on them. However, one crucial quality often overlooked by developers in the implementation of mobile apps is their accessibility. Approximately 15% of the global population has some form of disability [74], and a substantial number of them rely on assistive technologies to use mobile apps. An assistive technology, such as a screen reader, depends on the implementation of apps to function properly (e.g., availability of proper labels for images). As a result, it is imperative for developers to ensure their apps are accessible to all users, regardless of their abilities or preferred mode of interaction.

Several empirical studies investigating real-world mobile apps have highlighted the prevalence of accessibility issues within them [4, 16, 55]. These impede effective app usage for people with disabilities. Fortunately, recent years have seen an increased awareness of app accessibility, largely due to the growing independence of people with disabilities using mobile apps and advocacy in professional and online spaces [37]. Government mandates, such as Section 504 and 508, coupled with the rise in accessibility-related lawsuits, have drawn attention to the importance of app accessibility [20]. In response to these concerns and to mitigate legal risks, technology companies such as Google and Apple have published accessibility guidelines to assist mobile developers in creating more inclusive and accessible apps [8, 12].

Despite these efforts to improve the overall accessibility of mobile apps, an important factor that has not been the focus of prior studies is the accessibility of mobile ads. In the mobile app ecosystem, there are two ways of provisioning apps from a financial standpoint: pay-to-install and free-to-use. The latter generates revenue through in-app purchases, advertisements, or a combination of both. Advertisements have become a crucial component of this ecosystem and are estimated to be a multi-billion-dollar market [63].

Ads have been examined for their privacy risks [13, 14, 31], security aspects [22, 25, 43], and user perceptions [29, 30, 33]. However, little research has focused on how ads affect people with disabilities, especially blind users. To the best of our knowledge, only two studies have investigated the impact of ads on blind users. In a study that predates smartphones, Thompson and Wassmuth [67] found that more than half of the sampled ads in online newspapers have no ALT tag. Their empirical study only examined the accessibility issue of labeling within web ads while neglecting to address other accessibility concerns [72]. Another study by Nengroo and Kuppusamy [50] used questionnaires to gauge screen reader users' preferences and challenges with ads on the web. Their study did not involve real-time observation of how blind users interact with ads, potentially leading to the loss of crucial clues and details.

While these previous studies have primarily focused on the accessibility of web ads, there is a lack of research examining the accessibility implications of mobile ads. There are several factors that collectively create distinctly different ad-related experience for blind users on mobile devices than on the web. A previous study found that mobile screen reader users take more time to complete tasks on mobile apps than on equivalent web/desktop applications due to finger-based navigation [40]. Another study pointed out

ICSE '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0217-4/24/04.
https://doi.org/10.1145/3597503.3639228

that users can block web ads using ad-blockers [50], but no equivalent solution is available for mobile apps. Additionally, there are distinctions in how web and mobile ads are represented. Web ads are typically displayed using the `Document Object Model (DOM)` when integrated into web pages. In contrast, mobile ads can be implemented using native elements specific to the mobile platform or a `WebView` component. Therefore, considering the different ad-related experiences for blind users on mobile devices, this paper aims to shed light on the accessibility implications of mobile ads for blind users.

Overall, the paper makes the following contributions:

- A publicly available tool, called AdMole, that can assist developers with detecting accessibility issues of ads on Android apps [11];
- An investigation into 5 accessibility issues across 500 ad screens, comparing the accessibility in different ad formats and libraries;
- Insights from 15 qualitative user interviews about the impact of mobile ads on blind users' navigation;
- An analysis of the design and financial implications of addressing accessibility of mobile ads.

The remainder of this paper is structured as follows: Section 2 provides a motivating example. Sections 3 and 4 discuss the methods and findings of the empirical and qualitative studies, respectively. Section 5 synthesizes findings of the two studies and provides practical implications. Section 6 presents a review of prior research in this domain. Section 7 discusses threats to validity, and the paper concludes with Section 8.

2 MOTIVATING EXAMPLE

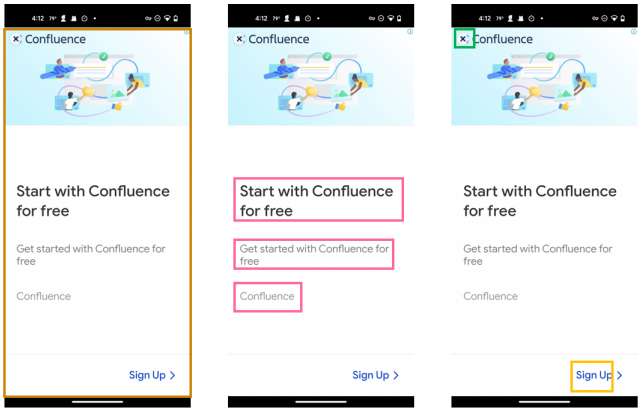

Figure 1: The navigation process for an interstitial ad.

This section illustrates the challenges faced by Android blind users when interacting with mobile ads using screen readers. Consider a scenario where a blind user downloads a news app but is unfamiliar with its features. Upon opening the app, an interstitial ad appears, occupying the entire screen. Since TalkBack, the Android screen reader, is activated, it automatically focuses on the first element, announcing *"web view"* to the user. Here, the screen reader is announcing the type of view that is displaying the ad, which in this case is a `WebView` component, and indicated by the brown box in Figure 1(a). The blind user becomes confused by the announcement of *"web view"*, since the user is unaware that it is an ad. Based on her mental model, she expects to find an option that allows her to select different news categories. Consequently, she employs linear navigation, swiping right to explore all the elements on the current screen one by one, hoping to find the news category option. During her exploration, TalkBack focuses on a `TextView` with the text *"Start with Confluence for free"*. This only adds to her confusion. She performs two more quick swipes to the right, and TalkBack focuses on the remaining two `TextView` elements, indicated by the pink box in Figure 1(b), announcing their textual descriptions. None of the elements announced by TalkBack align with the blind user's expectation. Determined to understand the purpose of the page, she continues exploring. Another swipe right leads TalkBack to focus on the yellow box in Figure 1(c), announcing it as *"Sign Up"*. The blind user assumes that she needs to sign up for an account in order to browse different news categories.

To confirm her assumption, she performs another swipe right. This time, TalkBack focuses on the last element of the screen, as shown by the top-left green box in Figure 1(c). Unfortunately, this element is improperly labeled, resulting in TalkBack only announcing it as "button" to the blind user. She becomes aware that the button is unlabeled, based on her past experiences of encountering unlabeled buttons. Due to her uncertainty about where the unlabeled button will direct her, she decides to navigate back by performing a swipe left gesture. TalkBack returns to the *"Sign Up"* `TextView`. To ensure that she does not miss any elements on the screen, the blind user navigates back until she encounters the `WebView` again. Realizing that she has explored all the elements on the screen, she starts navigating forward until she reaches the *"Sign Up"* `TextView`. Now, she firmly believes that signing up for an account is necessary to browse the news.

She double-taps on the screen (which is the gesture in TalkBack for selecting a focused element) to select the *"Sign Up"* `TextView`, expecting a sign-up form. But to her surprise, it opens up her browser and directs her to a web page. TalkBack announces *"Chrome Browser"* upon opening the web page. After exploring several elements on the web page, she finally realizes that she had encountered an ad upon opening the app, and clicking the *"Sign Up"* `TextView` directed her to the web page. Frustrated, she closes the browser and, after additional navigation, selects an unlabeled button to dismiss the ad and access the app screen.

The example above is, in fact, collected from one of five preliminary user studies we conducted with blind users. These users used TalkBack, Android's screen reader, to navigate apps. For each study, we provided 5 different apps with 5 types of ads [35]. We examined how they interacted with these ads and found that, similar to the illustrated example, blind users face considerable challenges in noticing the presence of ads. Furthermore, when accessibility issues were present within an ad, it led to confusion and required additional time and effort to dismiss the ad. These insights have informed the empirical and qualitative studies for this work.

3 EMPIRICAL STUDY

To investigate the properties and accessibility issues of mobile ads, we conducted an empirical study on apps from Google Play Store.

Table 1: The unique identifier for each ad library.

Ad Library	Unique Identifier
AdMob	**meta-data**: com.google.android.gms.ads.APPLICATION_ID
Meta Audience	**activity**: com.facebook.ads.AudienceNetworkActivity
AppLovin	**meta-data**: applovin.sdk.key

3.1 Methodology

We evaluated the accessibility of 500 ad screens using a customized automated accessibility evaluation tool. This tool was developed based on existing literature and tailored specifically for assessing the accessibility of mobile ads.

3.1.1 ***Study Subject.*** We opted to focus on the Android platform for our empirical study due to its widespread usage in the current mobile landscape [68]. Our approach involved crawling the Google Play Store across 28 app categories and applying several filters. Firstly, we filtered out apps that did not display any ads, relying on the presence of the *"Contains ads"* label on the Google Play Store page. We also filtered apps based on their installation numbers, excluding those with less than 100,000 installations. This process resulted in 1,079 apps.

We aimed to study the accessibility of mobile ads from the top 3 ad libraries, as reported by previous literature [2]. We chose the top 3 ad libraries, because each of the remaining ad libraries, as identified in that study, had a presence rate of no more than 7% in the app ecosystem. The top 3 ad libraries are Google AdMob, Meta Audience Network, and AppLovin.[1] To ensure that apps in our dataset incorporated at least one of these libraries, we inspected the apps' `AndroidManifest` files for evidence of ad library integration, a technique suggested by [52]. We used the integration guidelines offered by the three ad libraries, which contained unique identifiers, as shown in Table 1. For instance, if an `AndroidManifest` file contains a meta-data entry with the name `applovin.sdk.key`, it indicates that the app incorporates the AppLovin ad library.

To achieve the filtering process described above, we downloaded all 1,079 APKs from Google Play Store and pulled the APK files from a rooted physical device to our laptop using the ADB command. We then parsed the AndroidManifest files using the *AAPT2* tool [7]. If an APK file did not contain any of the *unique identifiers* listed in Table 1, we excluded that app from our dataset. As a result, we obtained a final dataset of 545 apps.

3.1.2 ***Accessibility Evaluation Tool – AdMole.*** To evaluate the accessibility of mobile advertisements inside Android apps, we built upon an existing automated accessibility evaluation tool, called Groundhog [61]. We decided to build upon Groundhog as it outperforms existing accessibility evaluation tools, such as Google Accessibility Scanner [6]. Such tools primarily assess an app's compliance with accessibility guidelines, many of which are irrelevant for screen reader users (e.g., color contrast and text size issues that do not affect a blind user using a screen reader). By contrast, Groundhog specifically detects accessibility issues that manifest when the app is navigated using a screen reader, and demonstrates high accuracy [61]. In order to align Groundhog with the scope of this study — mobile ads — we made modifications to specifically assess the accessibility of ad elements. We further extended the tool's functionality to detect two additional accessibility issues that were identified in previous studies [4, 16, 59]. A successor of GroundHog, we name the new tool AdMole.

Identifying Ad Elements. When an ad occupies the whole screen, there is no need to differentiate between the ad elements and app elements since all the elements on the screen are related to the ad. However, if an ad occupies only a fraction of the screen, such as a banner ad, it becomes necessary to extract elements specifically related to that ad. To address this, we conducted a pilot study where we selected 30 apps from each ad library and identified the characteristics of the ad elements. After analyzing the pilot study results, we observed that the resource ID (a unique resource name for an element) of an ad element typically contains keywords such as *"_ad"*. For the complete set of conditions used to extract ad elements, please refer to our companion website [11]. It is worth noting that all 90 apps from the pilot study were excluded from the subsequent empirical study to avoid introducing any bias.

Supported Accessibility Violation Detection. AdMole is designed to detect five types of accessibility issues, the first three of which are already supported by Groundhog:

- **(I1) Unlocatable Element - Linear Navigation.** This issue pertains to elements on the screen that cannot be located or focused on when using TalkBack Linear Navigation. TalkBack Linear Navigation requires blind users to swipe right or left, allowing TalkBack to focus on the next or previous element on the current screen.
- **(I2) Unlocatable Element - Touch Navigation.** This issue refers to elements on the screen that cannot be located or focused on when using the TalkBack explore-by-touch strategy. When utilizing this strategy, blind users can touch a specific spot on the screen, and TalkBack should then focus on the element precisely at the coordinates of that touch.
- **(I3) Ineffective Action.** This category encompasses situations where TalkBack is unable to perform an action, such as double-tapping a button, resulting in no response or desired outcome.
- **(I4) Unlabeled Element.** Previous studies have identified the absence of labels for visual elements as one of the most prevalent accessibility issues in mobile apps [4, 16, 47]. In this context, if an element lacks the content or the textual description, it is categorized as an unlabeled element.
- **(I5) Excessive Interaction.** Previous literature has highlighted concerns regarding excessive interactions when using TalkBack [44, 59], which refers to situations where blind users are required to perform multiple swipes to accomplish a particular task. In our study, we consider interactions excessive if blind users need to swipe multiple times to reach the close button of a full-screen ad or to get out of the entire non-full-screen advertisement. To establish a threshold, we set the limit at 15 swipes, aligning with a previous study [59].

AdMole utilizes a client-server architecture. The server side is responsible for sending commands to the client side and is hosted on a Windows laptop equipped with an Intel Core i7-7700HQ, 2.80GHz CPU, and 24GB of RAM. The client side is used for interacting with

[1] MoPub was originally considered the third most popular ad library. But since it was acquired by AppLovin and merged into the AppLovin Network on January 3, 2022 [28], we replaced MoPub with AppLovin as the third ad library for our study.

the subject apps to assess the accessibility of ads. It is installed on a rooted physical device, specifically a Pixel 4a running Android version 13 and TalkBack version 13.1.

3.1.3 ***Data Collection and Analysis.*** We adopted a hybrid approach, combining random selection with the round-robin algorithm, to guarantee a well-balanced representation of both app categories and ad libraries in our study. This approach resulted in 100 apps selected from our dataset, representing all 28 app categories. We included 34 apps integrated with AdMob, 33 apps utilizing the Meta Audience Network, and 33 apps employing AppLovin. We then assessed five screens for each app that displayed advertisements using AdMole. Upon launching the app, we initially checked for the presence of an ad. If an ad was found, we captured a UI screenshot of that screen. Then, the automated analysis of the captured UI screenshot would begin. In cases where no ads were initially found, various actions were conducted, such as scrolling, typing, and clicking, to determine if these actions would reveal additional ad screens. The actions ceased once we had collected 5 ad screens for a specific ad library. Please note that the selection of the five ad screens is tied to a specific ad library, with the sequence of the round-robin process determining which ad library is chosen.

In addition, we aimed to investigate the influence of ad formats on the accessibility quality of mobile ads. We conducted a thorough review of the documentation provided by the top 3 ad libraries. We found that each ad library supported its own distinct set of ad formats. For instance, while AdMob supported the App-Open format, the Meta Audience Network had not yet implemented support for it. Nevertheless, we identified certain similarities between some ad formats upon examining the interaction guidelines. For instance, banner ads from Google AdMob and MREC ads from AppLovin exhibited striking resemblance, occupying a small portion of the screen and refreshing their content at approximately 30-second intervals. Similarly, rewarded ads from AppLovin and interstitial ads from Meta Audience commandeered the entire screen, requiring users to locate a dismiss button to return to the app's flow. Inspired by these observations and guided by the ad library documentation, we mapped the various ad formats to higher-level categories:

- **Interstitial Related Ad.** This category includes ads that occupy the entire screen. Users have to dismiss those ads in order to proceed.
- **Banner Related Ad.** These ads occupy only a small portion of the screen, and the content within the ad is automatically refreshed every 30 seconds or so.
- **Native Ad.** Native ads also occupy a small portion of the screen. However, unlike Banner Related Ads, Native Ads employ native Android elements to display their ad content. For instance, instead of using a WebView to wrap the ad content, Native Ads might use `LinearLayout` or other native UI elements. Additionally, Native Ads do not refresh the ad contents automatically.

Another challenge we encountered in our empirical study was determining the ad format and the specific ad library employed during runtime. This challenge arose because an app might integrate multiple ad libraries to maximize revenue [35, 57]. To address this challenge, we drew inspiration from previous studies that used ad requests to map the ad library and corresponding ad format [22, 64]. We utilized HTTP Toolkit [53] to intercept ad requests. We also created three mock Android apps, each integrating one of the three ad libraries, to establish the mapping between ad requests and specific ad libraries and formats. Due to space limitations, we only present the ad request mapping for the interstitial-related ad format, as shown in Table 2. For the complete list of mappings, please refer to our companion website [11].

Table 2: The interstitial ad request mapping.

Ad Library	Base URL	Interstitial-Related Ad
AdMob	https://googleads.g.doubleclick.net/mads/gma	format = interstitial_mb
Meta Audience	https://graph.facebook.com/network_ads_common	PLACEMENT_TYPE = interstitial or rewarded_video or rewarded_interstitial
AppLovin	https://prod-mediate-events.applovin.com/1.0/event/load	applovin-ad-format = INTER or REWARDED or APPOPEN

Once the ad library and format were determined, we executed AdMole on the current screen. The tool automatically detected and assessed the accessibility of the ad screen. To mitigate potential bias, we implemented a data normalization process for the collected data from screens. This normalization step was crucial because screens with a larger number of UI elements related to ads have a higher likelihood of encountering accessibility issues. To that end, for most of the experiments we report the *inaccessibility rate*, which we define to be the number of ad elements on a screen exhibiting certain accessibility issues by the total number of ad elements present on that screen. For example, the inaccessibility rate for Unlabeled Elements on an ad screen is determined by dividing the number of ad elements without a label to the total number of ad elements on the screen.

In our study, we conducted scans on a total of 165 ad screens from the Meta Audience Network, 165 ad screens from AppLovin, and 170 ad screens from AdMob.

3.2 Results

This section presents the results of our empirical study. We examined the prevalence of accessibility issues across all 500 scanned ad screens. The Mean and Standard Deviation presented in Table 3 indicate a rightward skew in the distribution of the overall inaccessibility rate. This suggests that, while the overall inaccessibility rate is relatively low, with an average of 3.96%, a majority of mobile ads suffer from accessibility issues, with 84.4% of ad screens (422 ad screens) exhibiting some form of accessibility problem.

Table 3: The statistics of accessibility issues.

Issue	Frequency	Mean	Std	Max
Unlabeled Element	335 Screens	8.81	9.17	66.67
Unlocatable Element - Touch	318 Screens	6.15	8.47	61.54
Unlocatable Element - Linear	229 Screens	4.55	8.01	66.67
Excessive Interaction	7 Screens	0.19	2.06	31.44
Ineffective Action	5 Screens	0.08	0.9	15
All issue types		3.96	4.44	35.39

3.2.1 ***Comparison by Issue Type.*** Table 3 looks into the statistics of the five types of accessibility issues on the 500 ad screens. The most frequent accessibility issues are: *Unlabeled Element*, *Unlocatable Element - Touch*, and *Unlocatable Element - Linear*.

Table 4: The accessibility comparison of different ad formats.

Ad Format	Total Ad Screens	Without Issue	Issue-Specific Inaccessibility Rates					Overall Inaccessibility Rate
			I1	I2	I3	I4	I5	
Interstitial-Related	167 Screens	27 Screens	**6.73**	7.88	0	8.36	0.03	**4.6**
Banner-Related	167 Screens	**28 Screens**	5.48	**8.22**	0.09	7	**0.53**	4.27
Native Ad	166 Screens	23 Screens	1.42	2.34	**0.16**	**11.09**	0	3

(I1) Unlocatable Element - Linear Navigation, (I2) Unlocatable Element - Touch Navigation, (I3) Ineffective Action, (I4) Unlabeled Element, and (I5) Excessive Interaction.

Table 5: The accessibility comparison of different ad libraries.

Ad Library	Total Ad Screens	Without Issue	Issue-Specific Inaccessibility Rates					Overall Inaccessibility Rate
			I1	I2	I3	I4	I5	
AdMob	170 Screens	5 Screens	4.79	5.64	**0.24**	9.97	**0.46**	4.22
Meta Audience	165 Screens	**53 Screens**	1.06	2.98	0	1.97	0	1.2
AppLovin	165 Screens	20 Screens	**7.79**	**9.84**	0	**14.46**	0.1	**6.44**

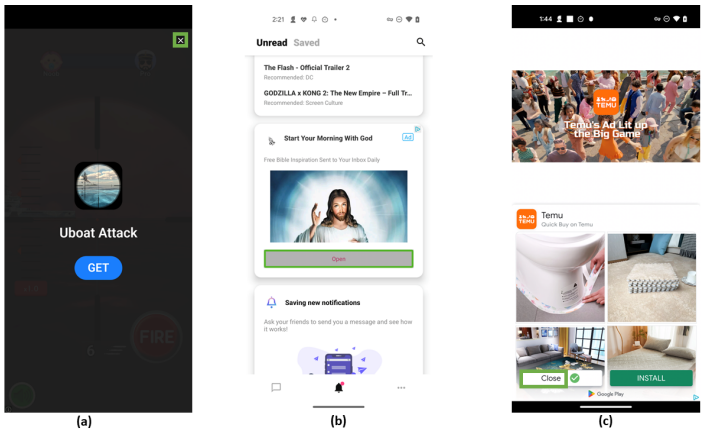

Figure 2: Illustration of accessibility issues in ads. (a) Unlocatable Element; (b) Ineffective Action; (c) Excessive Interaction.

In Figure 2, we show examples of accessibility issues found in our study, **bar I4 - Unlabeled Button**, which has previously been illustrated in Section 2. Figure 2(a) reveals issues **I1 and I2 - Unlocatable Elements**. The close button, denoted by the green box at the top right, cannot be focused on by TalkBack, neither through linear navigation nor explore-by-touch. This can result in blind users not being able to close the ad and return to the app. Figure 2(b) showcases issue **I3 - Ineffective Action**, which pertains to a `Button` object with the text "Open" responsible for opening the Google Play store page. When blind users employ TalkBack to click on this button, nothing happens. Figure 2(c) demonstrates issue **I5 - Excessive Interaction**, on the close button located at the bottom left. Since TalkBack can only focus on one UI element at a time, blind users must navigate through the `WebView`, the name of the advertised app, and 15 product images before reaching the close button. This results in an arduous process, requiring the user to swipe more than 20 times to close the ad.

3.2.2 ***Comparison by Ad Format.*** As mentioned in Section 3.1.3, we categorized ads into three formats: *interstitial-related ads*, *banner-related ads*, and *native ads*. We present the comparison in Table 4 and highlight the highest value for each column.

Table 4 reveals that all three ad formats have a similar number of screens without accessibility issues. Native ads, in general, demonstrate better accessibility compared to the other formats, boasting the lowest overall inaccessibility rate and outperforming others in three average inaccessibility rates, except for Ineffective Action (I3) and Unlabeled Element (I4). Indeed, native ads exhibit the highest average inaccessibility rate for the Unlabeled Element issue, suggesting a recurring problem in this specific ad format. Regarding Interstitial-Related and Banner-Related ads, they exhibit similar accessibility levels, as indicated by the comparable overall inaccessibility rate. The most frequent accessibility issues in Interstitial-Related ads are Unlabeled Element (I4) and Unlocatable Element-Touch Navigation (I2), while in banner-related ads, the most common issue is Unlocatable Element-Touch Navigation (I2).

Furthermore, certain accessibility issues can be specific to particular ad formats. For instance, all instances of Ineffective Action (I3) occurred in banner-related ads and native ads, while instances of Excessive Interaction (I5) were only observed in banner-related ads and interstitial-related ads.

3.2.3 ***Comparison by Ad Library.*** Table 5 provides an accessibility comparison among 3 ad libraries, highlighting the highest value for each column. Notably, within the Meta Audience Network, 53 ad screens were entirely free of accessibility issues. This accounts for a significant portion of all observed accessible screens — 53 out of 78, equating to 67.95%.

We visualize this data in Figure 3 for better comparison. As observed, AppLovin tends to exhibit the highest inaccessibility rates, especially for Interstitial and Banner-related ads. This suggests a need for targeted improvement in these specific ad formats by AppLovin. Among interstitial-related ads from AppLovin, a staggering 92.7% (51 out of 55) suffer from Unlocatable Element issues (I1 and I2). Notably, in these ads, crucial elements like the ad-choice icon and the close ad button are not locatable using either linear or touch navigation methods. However, AppLovin's performance drastically improves with Native ads, surpassing those of AdMob and Meta Audience. This emphasizes the role of native elements in reducing inaccessibility in ads.

AppLovin ads suffer from the Unlabeled Element Issue (I4), consistently exhibiting the highest inaccessibility rates for I4 across all ad formats. While AdMob generally performs better than AppLovin, it also suffers from I4, especially for Banner-related and

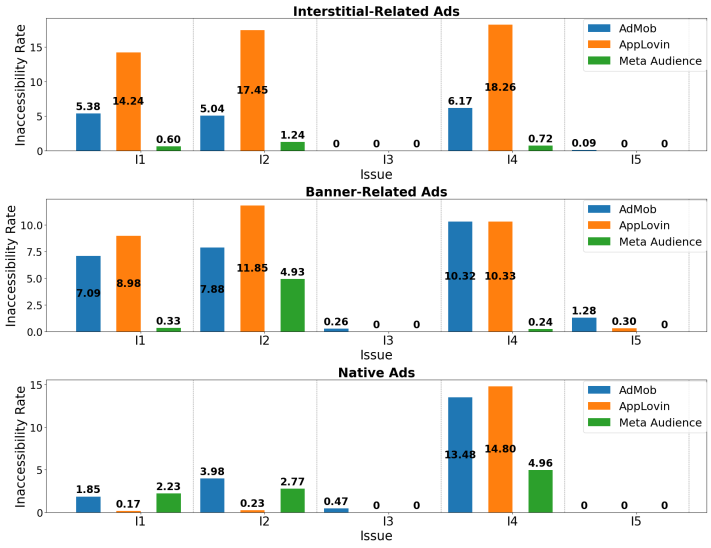

Figure 3: Comparing inaccessibility rates across different ad formats and ad libraries.

Native ads. I4, affecting all native ads from AppLovin and 98.21% (55 out of 56) of native ads from AdMob, underscoring a critical area for improvement in accessibility for both these ad libraries.

Meta Audience showcased the fewest ads with accessibility issues, the lowest overall inaccessibility rate and superior accessibility quality in both Interstitial-Related and Banner-Related ads. Based on the overall inaccessibility rates, the ad libraries can be ranked from most to least accessible as follows: *Meta Audience Network* > *AdMob* > *AppLovin*.

4 QUALITATIVE STUDY

Our empirical study has revealed a range of accessibility issues present in mobile ads. We conduct 15 qualitative user interviews with blind Android users to further investigate how these issues impact their experience, along with how they interact with and what their preferences are for mobile ads.

4.1 Methodology

For our qualitative study, we conducted user interviews, a combination of user studies and semi-structured interviews. The user study was conducted using open-source Android apps, modified to include different types of advertisements with varying degrees of accessibility. We observed how users navigated through, around or out of the ads, and how those ads affected their navigation. In the semi-structured interview, the same participants were asked about their experiences and opinions of mobile ads, both from the study and in general.

4.1.1 ***Artifacts:*** Among the 15 user interviews, we conducted the first 10 to assess the impact of *inaccessible* ads on blind users. It is crucial to ensure that the participants experienced all 5 accessibility issues. As mentioned in Section 3.2.3, we have identified specific patterns within each ad library. For instance, 98.21% native ads from AdMob were found to have Unlabeled Elements, and 92.7% interstitial-related ads from AppLovin contained Unlocatable Elements. By leveraging these patterns, we ensured that participants encountered inaccessible ads from different categories.

While we could not identify similar patterns for the accessibility issues related to excessive interactions and ineffective action, we found alternative approaches to present those forms of inaccessibility in ads. As discussed in Section 3.1.3, we created three mock Android apps to establish a mapping between the ad requests and the ad formats. During this process, we discovered that developers had control over the native ads displayed in their apps. They could customize the layout, hide specific ad elements, add textual descriptions, and even render certain buttons ineffective. In our study, we modified the install button of the AdMob native ad to be ineffective, replicating the issue of ineffective action. To simulate excessive interactions, we developed mock ads that captured the essence of such ads. We discovered an interstitial ad from AdMob that contained 15 product images, with the close ad button positioned at the bottom of the screen. TalkBack users would potentially have to navigate through all the product images before reaching the close button. To replicate this inaccessible ad experience, we developed an Android `Activity` that mimics the behavior and we ensured that all the texts, content descriptions, and traversal orders of the elements remained the same as the original ad.

One criterion that influenced our choice of apps is their universal functionality. We selected apps that are likely to be used daily by users. Two open-source Android apps from GitHub meet the requirement [45, 49]: Markor, a note-taking app, and Money Manager, an expense-tracking app. We modified the source code of these apps to inject inaccessible ads into their interfaces. Each open-source app was utilized for 5 user studies, with the first 5 studies conducted using the Markor app and the subsequent 5 using the Money Manager app. Within each app, we introduced inaccessible ads encompassing all five issue categories. In our preliminary study from Section 2, we observed that the same accessibility issue can result in different impacts based on how it is presented. Therefore, the inaccessible ads in the two apps varied in the form of presentation. For instance, in the Markor app, the unlabeled elements were integrated into a native ad, while in the Money Manager app, an unlabeled element was embedded within an interstitial ad.

For the final 5 user interviews, our objective was to observe the impact of *accessible* ads on blind users. We wanted to observe whether and how ads, even if accessible, can affect the usability of apps for blind users. As observed in Section 3.2.3, ads from Meta Audience tended to exhibit a higher level of accessibility compared to the other two ad libraries. As a result, we have chosen three ads from Meta Audience, each representing a different ad format, to be utilized for the subsequent injection. Additionally, we took steps to improve the accessibility of the AppLovin native ad by adding missing content descriptions (labels) to its elements. We then injected all those 4 accessible ads into the second open-source app, namely Money Manager. The core contents and functionalities of the app remained unchanged, with the exception of the accessibility quality of the ads integrated within the app.

4.1.2 ***Participants and Protocol:*** For our study, we recruited blind participants who relied on screen readers to use their Android phones. We utilized the Fable platform [39], an organization that

connects technology companies with disabled users for accessibility evaluation. Specifically, we used their User Interview feature, a one-hour long free-form interview session where we could ask the participants to install apps and share their screens as they navigate them. This facilitated our mixed-method approach.

We conducted 18 individual sessions, 3 of which were discarded as the ads did not display properly. Due to the platform's limited roster of testers, there were repeat participants. However, no single tester was given the same set of ads more than once, enabling distinct user studies. To prevent familiarity with the app to cause bias, for our last five user studies, we asked Fable to recruit testers who were not part of the previous five.

Each session was divided into two parts. First, for the user study, the participant was given the ad-injected app and was asked to complete specific in-app tasks while sharing their mobile screen on Zoom. The participants were not informed of the injection of ads to prevent preconceived biases. Each task was designed so that the participant would encounter an ad while completing it. They were instructed to think aloud [70] as they performed their assigned tasks. When encountering ads, participants were asked specific questions such as "How did you recognize or fail to recognize an ad?" and "Did this ad introduce any obstacles?" The same tasks were then repeated on an ad-free version of that app.

The second part contained a semi-structured interview to gather more insight into the participants' responses to ads. They were asked about their experience in that session, regarding the varied intrusiveness of the different ads and their preferences. We also posed questions about their general experience with mobile ads.

It is important to note that all participants in our study used TalkBack as their screen readers and indicated a high level of familiarity with its usage. Among the 9 unique participants, 7 were male and 2 were female.

4.1.3 ***Coding and Analysis:*** After conducting 15 user interviews, we transcribed each recording. Each transcript included the conversations from think-aloud sessions and interviews, along with textual descriptions of critical incidents and user actions (e.g., when an ad popped up, what the participant did to close the ad, etc.).

We conducted qualitative coding to analyze the transcripts, following the coding guidelines from Saldañ [58]. We utilized the online coding software Delve [24], which facilitated the two authors to code concurrently and compare codes after completion. We employed the line-by-line open coding method [65] and interleaved it with axial coding [66] to establish connections and hierarchies between labels generated during the open coding phase.

To determine theoretical saturation, we followed a similar procedure as Breukelen *et al.* [69], comparing newly created codebooks to versions prior. Throughout the process, we maintained a list of codes generated from each transcript. We selected transcripts from all three phases in a round robin fashion. We chose the codebook generated from six user studies (two from each phase) as our initial version. We set the stopping criteria as two additional transcripts with a threshold of 95%. We compared the initial version of the codebook with the codebook after the eighth transcript. If the similarity of codes was 95% or more, we considered data saturation to have been reached. Otherwise, we proceed to code an additional transcript and compare the new codebook with the previous version. We observed data saturation was reached after the tenth transcript.

4.2 Results

The qualitative coding yielded 233 codes, presented with examples in our companion website [11]. In synthesizing the individual codes into higher level categories, three distinct perspectives emerged. First, how users interacted with the ads using Talkback. Since the users had not been informed of the injected ads, we were able to note their natural process of recognizing and navigating around ads. Second, how this interaction affected their navigation of the app itself, observing the impact the issues can cause. Lastly, based on these experiences, what are blind users' preferences, and what their overall opinions are on mobile ads.

4.2.1 ***How blind users interact with ads.*** More often than not, users were able to recognize an ad, along with its format — whether full screen, embedded[2] or including video. Users utilized multiple clues to recognize an ad. The most common one was *context clues*: elements associated with an ad layout. These include the close and install buttons, the "ad app icon", mentions of Google PlayStore, and ad action control buttons for muting, unmuting or playing media. See, e.g., **Table 6 Row 1**.

The second most common method of recognition was *recurrence*. When users repeated an assigned task that included an ad, they immediately recognized the ad. This corresponded with their dependence on *familiarity*; once they were familiar with the app's content, after multiple uses, they tended to navigate it more easily, as they memorized the transition of pages and placement of elements, including that of ads. See, e.g., **Table 6 Row 2**.

Familiarity also helped them identify unusual or unexpected page layouts, which in turn was used for recognizing ads. Unusual elements like random numbers and letters, lots of images, and no other actionable elements other than a close button indicated that it was not the app's content. See, e.g., **Table 6 Row 3**.

The textual content of the ads, like the advertised product's name and description, the name of the business, identified by its dissimilarity with the app's name, or promotional copy, helped users realize that the content they were reading was an ad. See, e.g., **Table 6 Row 4**.

Another common way of ad recognition was the appearance of `WebViews`. Most ads are presented to users with web components, contained in a `WebView`. On the other hand, the content of the app is presented via native Android components. When the user focuses on a WebView, TalkBack announces that, indicating that they entered an ad. See, e.g., **Table 6 Row 5**.

In the cases where the users could not recognize the ad, multiple factors played a role. For instance, inability to distinguish an ad from app content due to the user's unfamiliarity with the app, overlooking ad content due to focus on the task assigned, inability to comprehend ad content due to accessibility issues, and more.

Once the users realized that they were navigating inside an ad, their primary goal was to *skip past or exit it immediately*. For embedded ads, they would quickly swipe past the ad or go up

[2]For the sake of brevity, in the remainder of this section, we refer to ads that do not occupy the entire screen as embedded ads, i.e., banner-related and native ads.

using backward navigation, thinking the ad indicated the end of the screen. See, e.g., **Table 6 Rows 6-7**.

For interstitial ads, which they needed to close to progress, they either pressed the *back button* or looked for and pressed the *close button* offered by the ad. The former was sometimes substituted by the *back gesture*, a feature offered by TalkBack, in the form of a horizontally mirrored "L". See, e.g., **Table 6 Row 8**.

In case they could not detect a close button, they would *explore by touch*. As a last resort, they would *restart* the app, hoping the ad would not reappear. See, e.g., **Table 6 Rows 9-10**.

> Screen reader users recognize ads based on context clues, their familiarity with the UI, and text and layout that are inconsistent with the app's own. Upon recognition, their initial goal is to exit the ad via a close or back button, or swipe past it, without consuming the ad's content.

4.2.2 ***How ads impact blind users' experience.*** We observed various levels of negative impact, ranging from inconvenience to complete blockage. *Inconvenience* was primarily present with **accessible ads**. The sudden appearance of ads *confused* the user, *broke their flow*, and *slowed them down*. However, the clearly labeled elements made it easy to close the ad. See, e.g., **Table 6 Row 11**.

For inaccessible ads with unlabeled elements, sometimes the issue was amended by TalkBack's *Optical Character Recognition (OCR)* feature, which announced "X" or "cross" when encountering unlabeled close buttons. However, OCR was not always correct, and users said that they did not like to rely on it. Furthermore, this feature is only available from Android 13 onwards, so not all the participants could benefit from it. See, e.g., **Table 6 Rows 12-13**.

Unlabeled elements in the ad created *obstructions*. Users were confused about the content and how to exit the ad, usually needing hints to do so. Sometimes they could guess that a button might be used to close the ad, but were *concerned* about clicking an unlabeled button, as they were unsure where it would take them. They feared that it would redirect them to a webpage, which posed privacy and security issues. For instance, they were concerned that their browser would keep track of them visiting the page and would later be *bombarded by similar ads*. See, e.g., **Table 6 Rows 14-15**.

When encountering ads with excessive interaction issues, in most cases, users were unable to recognize it as an ad, confusing it as *part of the app's content*. Therefore, they ended up swiping through all of the elements, trying to complete the assigned tasks, making their navigation *less efficient*. On second encounters, they recognized the ad instantly and used explore by touch to quickly reach the close button. Users stated that more than 6-10 swipes or two minutes spent on an ad is *irritating*. See, e.g., **Table 6 Row 16**.

Users showed significant concern over ads that included videos, with or without audio. On ones without audio, the lack of clue on what was happening created *confusion* about the content. On the other hand, audio *disrupted* their navigation as it played over TalkBack's announcement, creating an auditory overload. See, e.g., **Table 6 Rows 17-18**.

The most problematic ad for all the users was an interstitial ad with unlocatable element. These ads took over the whole screen

Table 6: Quotes from blind participants.

Row	Quote
1	*"When I explored by touch at the bottom, you know, I noticed that there was an advertisement and it said 'Flood-It!', it said 'Install', it said 'Google Play', so I realized that, okay, that's definitely an advertisement right there."* - Study 6 (S6)
2	*"Typically, with an application I get very familiar with, and I kind of know where the advertisement are on the screen."* - S9
3	*"As I was moving through, it was just saying a bunch of gibberish, like basically development tags for whatever. I kinda, it took a second, and I'm like, yeah, this is probably another ad thing."* - S7
4	*"You know, I kind of realize by the name of [the ad] at the top, that's not the Markor name, so I think it's an advertisement."* - S3
5	*"TalkBack read out 'Webview', which to my mind, reads as an advertisement. Typically when I open an application, and the first thing I see is TalkBack saying Webview, there is a very high likelihood of an advertisement."* - S9
6	*"I've noticed the little bit of text, you know, so I think it was an ad. [Once I realized it was an ad,] I just wanted to skip over it, like I just wanted to pass it."* - S1
7	*"Well, I thought, I'm used to advertisements that are on the end of the, or at the right edge of the app, so that's why I started to navigate back when I encountered [the ad]."* - S10
8	*"[To close an ad] I'm gonna have to go through like we did, either with the back arrow, or go and close the ad down using the ad close button."* - S3
9	*"I end up doing a little bit of exploring by touch to basically see if I can move my TalkBack cursor to, you know, close to where I need to go, and then swipe to find the [ad close] button."* - S1
10	*"Oh, dear, I'm afraid the only way I know of at this point to X out of this ad is to close the app entirely and relaunch it."* - S2
11	*"I think, so [this accessible ad] is not something that I couldn't get over, but it definitely does add extra stuff to deal with."* - S11
12	*"I think TalkBack was able to automatically recognize the image and provide smart analysis."* - S8
13	*"I don't know, but it's certainly running the latest version, but sometimes it reads descriptions, and it says 'detected, X', and then other times it doesn't."* - S10
14	*"[For unlabeled elements,] I have to waste my time trying to get out of the ad, closing the ad, and not being sure what I clicked. If it's not labeled, I was like, do I risk it? Because sometimes, you can click on something that lead you further into, or, hopefully, not make a purchase, but generally, it can cause you to get further into the ad."* - S14
15	*"Situations like that [unlabeled elements] do concern me because I don't want the app store mistakenly assuming that I'm interested in this app just because some ads sent me to that web page or app store screen. I don't want them adding that to my search history as, 'oh, he's interested in a gambling app.' No, I'm not."* - S13
16	*"But if I have to swipe more than, I would say that, 10 is a pretty reasonable number to expect. In fact, more than 10 times, it gets me irritated very quickly."* - S9
17	*"Many times, ads consist of a video with only visual content and little to no auditory content where I have no idea what it's advertising because all this is playing is music, whereas [it's maybe showing] clever animations."* - S15
18	*"When it's a full-screen advertisement, it's like coming in over everything, I wanna block it. I don't want video and audio to start just playing because then I can't hear the TalkBack announcement."* - S7
19	*"Yeah, especially the video ad. I mean, it's crazy because it takes up the whole screen, and then you can't get out of it because you can't find the close button for that ad."* - S5
20	*"When I notice an ad, my immediate thought is to avoid it. I tend to view ads almost like a pestilence."* - S8
21	*"I still would rather get rid of them, even though they're quite accessible. I'm very aggressively opposed to advertising."* - S15
22	*"I would [opt for the ad-free version] if I were to find out that the app itself was accessible. That wouldn't but incline me toward making that purchase to remove the ads."* - S2
23	*"I feel like the app needs to be about 85 to 95% accessible for me to consider paying for a subscription."* - S6
24	*"But if it's something that I'm using a lot, then I will get the ad-free version because the ads are still kind of annoying."* - S11
25	*"So generally, if it's not a big price [for the ad-free version], I'd rather just get rid of them, I didn't have to deal with them."* - S3
26	*"As long as the app has other features besides removing ads in the subscription, I would use that [with a monthly subscription]. But just for removing ads, I would pay a one-time thing, you know."* - S12
27	*"That's the other thing I like about the advertisements at the bottom of the screen because a lot of times I don't notice them. When they're at the bottom of the screen, I get it more easily."* - S10
28	*"I would prefer the ad that just shows up, of course not during app launch, and then I could tap the back button and be done with it because it's very clear that it is an advertisement. I have to scroll through unnecessary information to find the information that I am looking for when the advertisement is in the middle of the screen."* - S9
29	*"I prefer the ones which only took up part of the screen, so I was able to successfully navigate either around or past to utilize the rest of the app."* - S15
30	*"[Ads are] clean, it's just noise. I like, I don't know, there is something in my brain to automatically switch it off [whenever I encounter an ad]. I don't even know what that thing was trying to advertise, it's just noise."* - S7
31	*"To me, [ads are] not necessarily evil because I know, especially when an app is not being paid for by the users, the advertisement sort of has to be there, right?"* - S7
32	*"Many times I suspect [developers are] using a third-party ad agency to deliver the ads, and they may not have much control over what ads show up, how they're presented."* - S2

with a `WebView`, hiding the navigation bar at the bottom, which removed the user's ability to close the ad via the back button. The back gesture would not be registered either. While a close button exists on the ad for a sighted user to tap on, it is unlocatable with Talkback, hence unusable. Any action on the ad would instantly redirect the user to the advertised app's Google Play Store page. Lastly, the user would try closing the app and relaunching it, but the ad would reappear. Therefore, this ad created a complete *blockage* as users could not access the app. See, e.g., **Table 6 Row 19**.

> Ads can obstruct and even completely block screen reader user's navigation due to unlabeled elements, audio and video content, and layouts that overtake Android's default UI. While proper labeling reduces interference, their appearance in the middle of app content creates confusion and hampers smooth navigation.

4.2.3 ***What blind users' preferences are on ads.*** All participants have responded negatively to ads, being *aggressively opposed to ads*, both in the context of the study and in their general experience. In most cases, they would rather *not use the app entirely* because of the intrusive ads, regardless of the level of obstruction they faced. The only exception was accessible ads that were easy to close or skip past. But even for those, most users would prefer to *avoid ads* and *opt for the ad-free version*, while a minority are willing to tolerate the ads. See, e.g., **Table 6 Rows 20-21**.

They stated multiple factors in choosing to pay. The most important being the *accessibility of the app itself*. Regardless of ads, if the app itself contained accessibility issues or was incompatible with TalkBack, users would not pay. See, e.g., **Table 6 Rows 22-23**.

The second factor was the *usage frequency or user needs*. If the app provided important features to the users that they needed regularly, then they would pay to remove the ads. See, e.g., **Table 6 Row 24**.

Pricing was another factor in choosing to pay. Their preference leaned towards a *one-time payment* rather than *monthly subscriptions*, solely for the purpose of removing ads. Additionally, they emphasized that the ad-free version should remain reasonably priced. They indicated a willingness to consider monthly subscriptions only if the app's premium version not only removes ads but also introduces new features. See, e.g., **Table 6 Rows 25-26**.

While their first preference regarding ads was to remove them entirely, we also observed their preferences for different ad formats. The most popular ad format was *embedded ads at the bottom of the screen*, which places the ad after all of the app's content. This prevented confusion and extra swipes while navigating the app's features, enabling ease and efficiency. See, e.g., **Table 6 Row 27**.

Among the participants, there were different preferences between embedded ads in the middle of the screen and interstitial ads. Those who preferred *interstitial ads* favored its recognizability and consequent ease of closing. It is easier to distinguish the ad from the app's content as the whole page is taken over. And it is easy to get back to the app's content by simply clicking on the close button. See, e.g., **Table 6 Row 28**.

On the other hand, participants favored *embedded ads*, even if these were placed in between app content, because they only need to swipe a few times to move past it. By contrast, if the interstitial ad contains an inaccessible design, it is much harder to navigate around it as they need to correctly locate the close button. Interstitial ads also tend to contain videos, which participants pointedly disliked. See, e.g., **Table 6 Row 29**.

In terms of the ad content, users have shown no preference or interest. They view ads as *background noise* to the app's actual content. They *retained no information* about the advertised product or service. Moreover, some stated that they tend to view the advertised product negatively if being promoted. See, e.g., **Table 6 Row 30**.

Users have also shown awareness of the commercial aspect of ads. They *empathize with the developers*, as ads are necessary to make profits. They are aware that developers do not construct the ads themselves, hence its accessibility is *out of their control*. They view entities that create and distribute the ads to be more accountable. However, users would still prefer if developers tested the ads' accessibility more before releasing, choosing ones that are more accessible. See, e.g., **Table 6 Rows 31-32**.

> Users respond very negatively towards ads, opting to switch to alternative apps or pay for an ad-free version, if the app itself is accessible. In terms of ad categories, users prefer embedded ads at the bottom of the screen, followed by interstitial ads with properly labeled close buttons.

5 DISCUSSION

In this section, we synthesize the findings from the empirical and qualitative studies to understand the practical implications of the different forms of mobile ads.

Based on how inaccessible ads impacted user navigation, we **assess the accessibility issues** identified in Section 3.1.2. Ineffective Action (I3), rendered as buttons on native ads, caused the least harm as users skipped past the affected elements without noticing them. Excessive Interactions (I5) caused annoyance, where users had to painstakingly navigate through the ad elements, and confusion, in trying to differentiate between app and ad contents. Unlabeled Elements (I4) caused obstruction to user navigation as they could not identify how to safely exit out of the ads. Unlocatable Elements (I1 and I2) resulted in blockages, as users found no way to exit out of the ad, preventing them from accessing the app's features.

Preference on ad formats can also be elucidated from the findings. Users who preferred interstitial ads favored it because they were familiar with the format. They did not get confused distinguishing between ad and app content, unlike with embedded ads. All they needed to do was to identify and activate the close button, or use their device's back button, to exit out. However, this depended on the proper labeling of the close button, which, as evident from the empirical study, is not consistent. This is why some other users preferred embedded ads, where they can simply swipe past the ad and it does not obstruct their navigation.

These insights provide two design implications for ads. Firstly, **ad elements must be properly labeled**. This prevents blind users from getting stuck in interstitial ads and from being confused when understanding the functionalities of ad elements. While the Optical Character Recognition (OCR) feature [10] on TalkBack 13.0 can sometimes generate missing labels, we observed that it is not

consistent and users prefer not to rely on it. A more definitive solution involves assigning accurate labels to these elements when the ad is programmed. Tools like Google Accessibility Scanner can assist by scanning app screens and annotating UI elements lacking labels. App developers can then access the Android layout files for native ads (since they have control over native ads) and insert the necessary labels. Similarly, entities responsible for creating or distributing ads should use these annotations to label elements appropriately on their end.

Second, **context clues must be provided** so that users can distinguish between ad and app content. This can be incorporated using textual cues that indicate an "ad" is being displayed at the beginning of the ad, so that the screen reader can announce it upon entry. Sometimes, for sighted users, context clues come in the form of audio and/or video playing. However, these are severely intrusive for blind users. A muted video provides them zero context of new content while an audio causes sensory overload, where they cannot hear screen reader announcements. A solution for this is to start video ads on mute by default and provide textual content as clues, as mentioned previously.

Placement of ads is another important aspect that we observed. For instance, users preferred if ads did not overlap with app contents. Embedded ad situated at the bottom of the screen minimizes interruption, as it would be announced at the end of the app's content. For interstitial ads, users objected to ads showing up during app launch, especially for new apps. Since they are not accustomed to the app itself, an ad at launch confuses them.

Another aspect of ad construction is the use of **Android Native elements over `WebViews`**. Webviews are popular because they add dynamism and interactiveness to ads, enabling more user engagement. However, we observed in our empirical study that interstitial ads implemented using `WebView` render essential elements, like the ad close button, unlocatable, thereby creating a blocker. Furthermore, according to the interviewed users, `WebViews` caused the most problems to their navigation. On the other hand, ads constructed with Android Native elements performed best both empirically and qualitatively, with the lowest inaccessibility rate and easier user navigation. This contributed to Meta Audience Network's higher ad accessibility, as a significant portion (86%) of the Interstitial-Related ads are implemented using Android Native elements. Ads on AppLovin, though generally less accessible than other ad libraries, outperformed others in Native ads.

Discourses on software accessibility are often challenged by its **financial implication** [37]. This is especially relevant when discussing the accessibility of ads, as ads are primary tools for generating revenues for apps. How should ads be constructed to promote accessibility without compromising profit generation? We can utilize our findings so far to achieve a balance between the two. Developers opt for ads that generate the highest revenues, which are interstitial-related and native ads [51]. As discussed, with interstitial ads, the close button must be properly labeled and the ads should not be displayed intrusively. Intrusive ads, defined as those that greatly disrupt content and slow down browsing due to improper placement in apps, are to be avoided. As per the Better Ads Standards, one way to prevent ads from being intrusive is to ensure that they are not displayed during app launch or immediately following a user's action [19]. Instead, interstitial ads should be integrated during natural page transitions. Our empirical study suggests that developers might prefer Meta Audience for their superior accessibility performance in interstitial ads. For other parts of the app, outside of page transitions, ads with native elements are recommended. According to our study, there is no significant difference in accessibility performance among native ads from different ad libraries, giving developers flexibility in their choice of ad provider.

6 RELATED WORK

6.1 Accessibility of Mobile Apps

Previous research has conducted empirical studies regarding Android app accessibility, primarily focusing on labeling issues [27, 48, 56, 62]. However, some studies have taken a broader view, examining a range of accessibility concerns in Android apps, including labeling, touch target size, color contrast, and more [1, 4, 71, 73, 75]. It's worth noting that many of these concerns, like touch target size and color contrast, pertain to individuals with motor disabilities or color blindness rather than blind users.

Besides empirical studies, various tools have been developed to identify and address accessibility issues in mobile apps. These tools typically evaluate adherence to accessibility guidelines [6, 9, 17, 26, 34, 38]. Some recent studies have integrated assistive technologies into mobile app accessibility evaluations [3, 5, 46, 59–61]. However, these tools often exclude mobile advertisements from their assessments, citing their versatility and uncertainty during run-time, which is recognized as a limitation. Although overall app accessibility has been examined, an important aspect that has been overlooked in previous research is the accessibility of mobile ads.

6.2 Mobile Advertisement

Previous researchers have studied different aspects of mobile ads. For example, several researchers investigated the privacy risks introduced by ad libraries. Grace *et al.* and Book *et al.* [14, 31] found that most ad libraries collected private information from users. Another study by Book *et al.* [13] analyzed a sample of 114,000 Android apps and concluded that many of the permissions used by ad libraries pose threats to users' privacy. Stevens *et al.* [64] studied 13 pervasive Android ad libraries and found that many leveraged permissions were not declared in their documentation.

Researchers have also investigated the security aspects of mobile ads. Cho *et al.* [18] implemented an automated click generation tool that could bypass the security policies of 75% of the ad libraries. Automatic tools such as MAdFraud [22], FraudDroid [25], MadDroid [43], and MAdLife [15] are used for detecting fraudulent behavior and devious contents in mobile ads. DAPANDA [42] is an automated tool for detecting aggressive push notifications in Android that entice users to download potentially harmful APKs.

Other studies have delved into the user perspectives on mobile ads. Gao *et al.* and Gui *et al.* [29, 33] examined app reviews and identified various ad-related issues reported by users. Furthermore, Gao *et al.* [30] developed RankMiner, a tool that quantifies user concerns about ads based on app reviews. Their findings indicated that users are particularly worried about the additional battery consumption caused by mobile ads. Similarly, Gui *et al.* [32] discovered that ads consume a significant amount of system resources, such as

CPU and battery, and the presence of these hidden costs can have a detrimental impact on app ratings.

To the best of our knowledge, no studies thus far have specifically examined the accessibility of mobile ads and their implications. Thompson and Wassmuth [67] found that more than 50% of ads in online newspapers lacked alternative text. Another study by Nengroo and Kuppusamy used questionnaires to gauge screen reader users' preferences and challenges with ads on the web [50]. Our study focuses on the mobile platform, using a mixed method that combines an empirical study to understand the state of accessibility in mobile ads and qualitative studies to understand the impact of inaccessible ads on blind users.

7 THREATS TO VALIDITY AND SOUNDNESS

External Validity. One potential threat to validity is the representativeness of the analyzed 500 ad screens. To address this concern, we adopted several mitigation strategies. Firstly, we selected subject apps from all the 28 app categories and ensured installation numbers greater than 100,000. Moreover, we analyzed a similar number of ad screens for each ad format and library to conduct fair comparisons. This balanced approach helps minimize bias and ensures that each ad format and library is adequately represented.

Another potential threat arises from the ambiguity in determining whether an accessibility issue originates from the ad itself or is due to a bug in TalkBack. Determining the exact cause often requires source code analysis, which was not possible since the apps we chose are not open source.

Internal Validity. Any errors introduced by the employed tool could negatively impact our research findings. To mitigate this concern, we built upon the tool used in previous literature, known as Groundhog [61], which has been reported to exhibit precision and recall rates greater than 83%. Despite this strong foundation, the modifications made to Groundhog, such as the detection of two additional accessibility issues and the identification of UI elements related to ads, could introduce defects. We ensured our implementation's reliability through extensive testing and validation, using a small, separate set of apps to confirm the accuracy of our results before our empirical study.

We adopt Lincoln and Guba's [41] framework to examine the soundness of our qualitative research, using three validity strategies advised by Creswell [21]: triangulation, prolonged engagement in the field, and peer debriefing.

Credibility. Our selection of ads for the user study was informed by the empirical study, thus ensuring triangulation. We aimed to incorporate all the accessibility issues and ad formats we had identified into apps tested. To minimize inaccuracy in displaying the inaccessible ads, we employed real world advertisements. Only for Excessive Interaction, which could not be displayed consistently, we simulated with an Android Activity. However, to retain user behavior, we replicated the labels and traversal orders of ad elements observed in our empirical findings.

To construct a protocol without bias or ambiguity, we conducted 5 pilot studies with testers with blindness. We also employed peer debriefing on the protocol, especially for perfecting its mixed style of conversation and user actions. The interviews had been conducted with blind Android users with reported proficiency in TalkBack. With their prolonged experience in this platform, they were able to report nuanced insights about the issues and recontextualize their general experience with that of the user study. During analysis, the two authors independently conducted parallel coding to mitigate the potential impact of author bias on the qualitative code. Disagreements were resolved through collaborative discussion.

Transferability. To ensure our user study has broad generalizability, we chose apps that are likely to be used daily by users, as evident in [23, 36, 54]. To maintain the integrity of our study's findings, we addressed potential biases by designing tasks that required blind users to engage with the full suite of the app's features, thereby preventing skewed results from a narrow use of the apps. Additionally, we evaluated ads of various types and with varying levels of accessibility to comprehensively assess their effect on the blind user experience.

8 CONCLUSION

This paper reports on the first empirical investigation into the accessibility of mobile ad screens for blind users, analyzing 500 ad screens from 3 popular ad libraries. The results highlight a notable prevalence of accessibility issues, with 84.4% of the analyzed ad screens demonstrating some form of accessibility problem when navigated with a screen reader. The most common issues found within mobile ads include *unlabeled elements*, *unlocatable elements when using explore-by-touch*, and *unlocatable elements when using linear navigation*. Subsequently, 15 qualitative user interviews were conducted with blind users to understand their interaction with mobile ads. All participants have responded negatively to ads and tend to view ads like a pestilence. The results highlighted various impacts of inaccessible ads, from interruptions to complete blockage, shedding light on how users identified and exited ads.

Lessons that have emerged from this work will guide our future research. Notably, we will explore whether the automated tool constructed to assist our empirical analysis can be adapted to help businesses provisioning ad libraries to vet ads for accessibility. We will further investigate how ads may impact users with other forms of disability, such as those with motor impairment, who use other types of assistive technology (e.g., switch access) to navigate apps.

Our research artifacts are available on the companion website [11].

ACKNOWLEDGMENTS

This work was supported in part by award numbers 2211790 and 2106306 from the National Science Foundation. We are grateful to Fable for granting us access to its platform, which allowed us to conduct extensive user studies. We appreciate the anonymous reviewers of this paper for their detailed feedback, which helped us improve the work.

REFERENCES

[1] Patricia Acosta-Vargas, Luis Salvador-Ullauri, Janio Jadán-Guerrero, César Guevara, Sandra Sanchez-Gordon, Tania Calle-Jimenez, Patricio Lara-Alvarez, Ana Medina, and Isabel L Nunes. 2020. Accessibility assessment in mobile applications for android. In *Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2019 International Conference on Human Factors and Systems Interaction, July 24-28, 2019, Washington DC, USA 10.* Springer, 279–288.

[2] Md Ahasanuzzaman, Safwat Hassan, Cor-Paul Bezemer, and Ahmed E Hassan. 2020. A longitudinal study of popular ad libraries in the Google Play Store. *Empirical Software Engineering* 25 (2020), 824–858.

[3] Ali S Alotaibi, Paul T Chiou, and William GJ Halfond. 2022. Automated Detection of TalkBack Interactive Accessibility Failures in Android Applications. In *2022 IEEE Conference on Software Testing, Verification and Validation (ICST)*. IEEE, IEEE, Virtual, 232–243.

[4] Abdulaziz Alshayban, Iftekhar Ahmed, and Sam Malek. 2020. Accessibility issues in Android apps: state of affairs, sentiments, and ways forward. In *2020 IEEE/ACM 42nd International Conference on Software Engineering*. ICSE, Virtual, 1323–1334.

[5] Abdulaziz Alshayban and Sam Malek. 2022. AccessiText: Automated Detection of Text Accessibility Issues in Android Apps. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Singapore, Singapore) *(ESEC/FSE 2022)*. Association for Computing Machinery, New York, NY, USA, 984–995. https://doi.org/10.1145/3540250.3549118

[6] Android. 2022. *Accessibility Scanner*. Google. Retrieved June 27, 2023 from https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=en_US

[7] Android. 2023. *AAPT2*. Google. Retrieved June 27, 2023 from https://developer.android.com/tools/aapt2

[8] Android. 2023. *Build more accessible apps*. Google. Retrieved June 27, 2023 from https://developer.android.com/guide/topics/ui/accessibility

[9] Android. 2023. *Improve your code with lint checks*. Google. Retrieved June 27, 2023 from https://developer.android.com/studio/write/lint?hl=en

[10] Android. 2023. *What's new with TalkBack 13.0*. Google. Retrieved November 8, 2023 from https://support.google.com/accessibility/android/answer/12345706?hl=en

[11] Anonymous. 2023. *Advertisement-Accessibility*. Anonymous. Retrieved June 27, 2023 from https://github.com/AdvertisementAccessibility/Advertisement-Accessibility

[12] Apple. 2023. *Building accessible apps*. Apple. Retrieved June 27, 2023 from https://developer.apple.com/accessibility/

[13] Theodore Book, Adam Pridgen, and Dan S Wallach. 2013. Longitudinal analysis of android ad library permissions. *arXiv preprint arXiv:1303.0857* (2013).

[14] Theodore Book and Dan S Wallach. 2015. An empirical study of mobile ad targeting. *arXiv preprint arXiv:1502.06577* (2015).

[15] Gong Chen, Wei Meng, and John Copeland. 2019. Revisiting mobile advertising threats with madlife. In *The World Wide Web Conference*. 207–217.

[16] Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhu, and Guoqiang Li. 2020. Unblind Your Apps: Predicting Natural-Language Labels for Mobile GUI Components by Deep Learning. In *2020 IEEE/ACM 42nd International Conference on Software Engineering*. ICSE, Virtual, 322–334.

[17] Sen Chen, Chunyang Chen, Lingling Fan, Mingming Fan, Xian Zhan, and Yang Liu. 2021. Accessible or Not An Empirical Investigation of Android App Accessibility. *IEEE Transactions on Software Engineering* 48 (2021), 3954–3968.

[18] Geumhwan Cho, Junsung Cho, Youngbae Song, and Hyoungshick Kim. 2015. An empirical study of click fraud in mobile advertising networks. In *2015 10th International Conference on Availability, Reliability and Security*. IEEE, 382–388.

[19] Coalition. 2023. *Better Ads Standards*. Coalition. Retrieved November 8, 2023 from https://www.betterads.org/standards/

[20] Ada Site Compliance. 2022. *A Recap of 2022 Website Accessibility Lawsuits*. Ada Site Compliance. Retrieved June 27, 2023 from https://adasitecompliance.com/recap-2022-website-accessibility-lawsuits/

[21] John W Creswell and Cheryl N Poth. 2016. *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

[22] Jonathan Crussell, Ryan Stevens, and Hao Chen. 2014. Madfraud: Investigating ad fraud in android applications. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 123–134.

[23] Dataintelo. 2023. *Note Taking App Market Report*. Dataintelo. Retrieved November 8, 2023 from https://dataintelo.com/report/global-note-taking-app-market/

[24] Delve. 2023. *Delve: Qualitative Data Analysis Software*. Retrieved June 27, 2023 from https://delvetool.com/

[25] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Tegawendé F Bissyandé, Tianming Liu, Guoai Xu, and Jacques Klein. 2018. Frauddroid: Automated ad fraud detection for android apps. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 257–268.

[26] Marcelo Medeiros Eler, José Miguel Rojas, Yan Ge, and Gordon Fraser. 2018. Automated accessibility testing of mobile apps. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation*. ICST, Västerås, Sweden, 116–126.

[27] Raymond Fok, Mingyuan Zhong, Anne Spencer Ross, James Fogarty, and Jacob O Wobbrock. 2022. A Large-Scale Longitudinal Analysis of Missing Label Accessibility Failures in Android Apps. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[28] Adam Foroughi. 2022. *AppLovin's Acquisition of MoPub Has Officially Closed*. AppLovin. Retrieved June 27, 2023 from https://www.applovin.com/blog/applovins-acquisition-of-mopub-has-officially-closed/

[29] Cuiyun Gao, Jichuan Zeng, David Lo, Xin Xia, Irwin King, and Michael R Lyu. 2022. Understanding in-app advertising issues based on large scale app review analysis. *Information and Software Technology* 142 (2022), 106741.

[30] Cuiyun Gao, Jichuan Zeng, Federica Sarro, David Lo, Irwin King, and Michael R Lyu. 2021. Do users care about ad's performance costs? Exploring the effects of the performance costs of in-app ads on user experience. *Information and Software Technology* 132 (2021), 106471.

[31] Michael C Grace, Wu Zhou, Xuxian Jiang, and Ahmad-Reza Sadeghi. 2012. Unsafe exposure analysis of mobile in-app advertisements. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. 101–112.

[32] Jiaping Gui, Stuart Mcilroy, Meiyappan Nagappan, and William GJ Halfond. 2015. Truth in advertising: The hidden cost of mobile ads for software developers. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 100–110.

[33] Jiaping Gui, Meiyappan Nagappan, and William GJ Halfond. 2017. What aspects of mobile ads do users care about? an empirical study of mobile in-app ad reviews. *arXiv preprint arXiv:1702.07681* (2017).

[34] Shuai Hao, Bin Liu, Suman Nath, William GJ Halfond, and Ramesh Govindan. 2014. PUMA: programmable UI-automation for large-scale dynamic analysis of mobile apps. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM New York, NY, USA, Bretton Woods, New Hampshire, USA, 204–217.

[35] Boyuan He, Haitao Xu, Ling Jin, Guanyu Guo, Yan Chen, and Guangyao Weng. 2018. An investigation into android in-app ad practice: Implications for app developers. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2465–2473.

[36] Malikberdi Hezretov. 2021. *Budget Tracker Highly Customizable Budgeting Mobile Application*. Ph. D. Dissertation.

[37] Syed Fatiul Huq, Abdulaziz Alshayban, Ziyao He, and Sam Malek. 2023. #A11yDev: Understanding Contemporary Software Accessibility Practices from Twitter Conversations *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 217, 18 pages. https://doi.org/10.1145/3544548.3581455

[38] KIF. 2023. *Keep It Functional - An iOS Functional Testing Framework*. Retrieved June 27, 2023 from https://github.com/kif-framework/KIF

[39] Fable Tech Labs. 2023. *Fable | Digital accessibility, powered by people with disabilities*. Fable Tech Labs. Retrieved June 27, 2023 from https://makeitfable.com/

[40] Barbara Leporini, Marina Buzzi, and Marion Hersh. 2023. Video Conferencing Tools: Comparative Study of the Experiences of Screen Reader Users and the Development of More Inclusive Design Guidelines. *ACM Transactions on Accessible Computing* 16, 1 (2023), 1–36.

[41] Yvonna S Lincoln and Egon G Guba. 1985. *Naturalistic inquiry*. sage.

[42] Tianming Liu, Haoyu Wang, Li Li, Guangdong Bai, Yao Guo, and Guoai Xu. 2019. Dapanda: Detecting aggressive push notifications in android apps. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 66–78.

[43] Tianming Liu, Haoyu Wang, Li Li, Xiapu Luo, Feng Dong, Yao Guo, Liu Wang, Tegawendé Bissyandé, and Jacques Klein. 2020. Maddroid: Characterizing and detecting devious ad contents for android apps. In *Proceedings of The Web Conference 2020*. 1715–1726.

[44] Nutnicha Maneesaeng, Proadpran Punyabukkana, and Atiwong Suchato. 2016. Accessible video-call application on android for the blind. *Lecture Notes on Software Engineering* 4, 2 (2016), 95.

[45] Markor. 2023. *Text editor - Notes & ToDo (for Android) - Markdown, todo.txt, plaintext, math, ..* Markor. Retrieved June 27, 2023 from https://github.com/gsantner/markor/tree/master

[46] Forough Mehralian, Navid Salehnamadi, Syed Fatiul Huq, and Sam Malek. 2022. Too Much Accessibility is Harmful! Automated Detection and Analysis of Overly Accessible Elements in Mobile Apps. In *2022 37th IEEE/ACM International Conference on Automated Software Engineering*. IEEE, ACM New York, NY, USA, Rochester, Michigan, USA, 13 pages.

[47] Forough Mehralian, Navid Salehnamadi, and Sam Malek. 2021. Data-driven accessibility repair revisited: on the effectiveness of generating labels for icons in Android apps. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM New York, NY, USA, Virtual, Athens, Greece, 107–118.

[48] Lauren R Milne, Cynthia L Bennett, and Richard E Ladner. 2014. The accessibility of mobile health sensors for blind users. In *International technology and persons with disabilities conference scientific/research proceedings (CSUN 2014)*. 166–175.

[49] Nada Nasser. 2023. *Money Manager App*. Nada Nasser. Retrieved June 27, 2023 from https://github.com/Nada-Nasser/Money-Manager-App

[50] Ab Shaqoor Nengroo and KS Kuppusamy. 2019. 'Advertisements or adversetisements?'—An accessibility barrier for persons with visual impairments. *Comput. J.* 62, 6 (2019), 855–868.

[51] Meta Audience Network. 2023. *4 reasons native ads are important for app monetization*. Meta Audience. Retrieved November 8, 2023 from https://www.facebook.com/audiencenetwork/resources/blog/4-reasons-native-ads-are-important-for-app-monetization

[52] Pranav Kalyan Panage. 2017. Analyzing Android Ad-libraries.

[53] Tim Perry. 2023. *Intercept, debug & mock HTTP with HTTP Toolkit.* HTTP Toolkit. Retrieved June 27, 2023 from https://httptoolkit.com/

[54] Ramesh Reddy. 2023. *10 Best Budgeting Apps of 2023: A Comprehensive Comparison.* TECHPP. Retrieved November 8, 2023 from https://techpp.com/2023/11/05/best-budgeting-apps/

[55] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O Wobbrock. 2017. Epidemiology as a framework for large-scale mobile application accessibility assessment. In *Proceedings of the 19th international ACM SIGACCESS conference on computers and accessibility.* ASSETS, Baltimore, MD, USA, 2–11.

[56] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O Wobbrock. 2018. Examining image-based button labeling for accessibility in Android apps through large-scale analysis. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility.* 119–130.

[57] Israel J Mojica Ruiz, Meiyappan Nagappan, Bram Adams, Thorsten Berger, Steffen Dienst, and Ahmed E Hassan. 2014. Impact of ad libraries on ratings of android mobile apps. *IEEE Software* 31, 6 (2014), 86–92.

[58] Johnny Saldaña. 2021. The coding manual for qualitative researchers. *The coding manual for qualitative researchers* (2021), 1–440.

[59] Navid Salehnamadi, Abdulaziz Alshayban, Jun-Wei Lin, Iftekhar Ahmed, Stacy Branham, and Sam Malek. 2021. Latte: Use-case and assistive-service driven automated accessibility testing framework for android. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* ACM New York, NY, USA, Virtual, Okohama, Japan, 1–11.

[60] Navid Salehnamadi, Ziyao He, and Sam Malek. 2023. Assistive-Technology Aided Manual Accessibility Testing in Mobile Apps, Powered by Record-and-Replay. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* 1–20.

[61] Navid Salehnamadi, Forough Mehralian, and Sam Malek. 2022. Groundhog: An Automated Accessibility Crawler for Mobile Apps. In *37th IEEE/ACM International Conference on Automated Software Engineering.* ACM New York, NY, USA, Rochester, Michigan, USA, 1–12.

[62] Leandro Coelho Serra, Lucas Pedroso Carvalho, Lucas Pereira Ferreira, Jorge Belimar Silva Vaz, and André Pimenta Freire. 2015. Accessibility evaluation of e-government mobile applications in Brazil. *Procedia Computer Science* 67 (2015), 348–357.

[63] Statista. 2021. Mobile advertising and marketing worldwide. https://www.statista.com/topics/5983/mobile-marketing-worldwide/#editorsPicks. (Accessed on 08/24/2022).

[64] Ryan Stevens, Clint Gibler, Jon Crussell, Jeremy Erickson, and Hao Chen. 2012. Investigating user privacy in android ad libraries. In *Workshop on Mobile Security Technologies (MoST)*, Vol. 10. 195–197.

[65] Anselm Strauss and Juliet Corbin. 1998. *Basics of qualitative research techniques.* Citeseer, Chapter Open Coding.

[66] Anselm Strauss and Juliet Corbin. 1998. *Basics of qualitative research techniques.* Citeseer, Chapter Axial Coding.

[67] David Thompson and Birgit Wassmuth. 2001. Accessibility of online advertising: a content analysis of alternative text for banner ad images in online newspapers. *Disability Studies Quarterly* 21, 2 (2001).

[68] Ash Turner. 2023. *How Many Android Users Are There? Global and US Statistics (2023).* Bank My Cell. Retrieved June 27, 2023 from https://www.bankmycell.com/blog/how-many-android-users-are-there

[69] Sterre van Breukelen, Ann Barcomb, Sebastian Baltes, and Alexander Serebrenik. 2023. " STILL AROUND": Experiences and Survival Strategies of Veteran Women Software Developers. *arXiv preprint arXiv:2302.03723* (2023).

[70] Maarten Van Someren, Yvonne F Barnard, and J Sandberg. 1994. The think aloud method: a practical approach to modelling cognitive. *London: AcademicPress* 11 (1994), 29–41.

[71] Christopher Vendome, Diana Solano, Santiago Liñán, and Mario Linares-Vásquez. 2019. Can everyone use my app? an empirical study on accessibility in android apps. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME).* IEEE, 41–52.

[72] W3C. 2023. *Guideline 2.1 Keyboard Accessible.* W3C. Retrieved June 27, 2023 from https://www.w3.org/TR/WCAG21/#keyboard-accessible

[73] Bruce N Walker, Brianna J Tomlinson, and Jonathan H Schuett. 2017. Universal design of mobile apps: Making weather information accessible. In *Universal Access in Human–Computer Interaction. Design and Development Approaches and Methods: 11th International Conference, UAHCI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017, Proceedings, Part I 11.* Springer, 113–122.

[74] WHO. 2011. *World report on disability.* World Health Organization. Retrieved June 27, 2023 from https://www.who.int/disabilities/world_report/2011/report/en/

[75] Shunguo Yan and PG Ramachandran. 2019. The current status of accessibility in mobile apps. *ACM Transactions on Accessible Computing (TACCESS)* 12, 1 (2019), 1–31.