

CREATING ACCESSIBLE MOBILE APPLICATIONS: A CASE STUDY OF CHALLENGES AND LESSONS

Rachael Baumann, Jaclyn K. Schwartz, Roger O. Smith
University of Wisconsin-Milwaukee
Rehabilitation Research Design and Disability (R₂D₂) Center

INTRODUCTION

The accessibility of mobile applications is a vital issue for people with disabilities. Use of the internet enhances quality of communication and sense of self-determination and independence for people with disabilities due to access to more and better information (Dobrasky & Hargittai, 2006). However, only 54% of individuals with disabilities have internet access (Harris Interactive, 2010). This is likely due at least in part to financial disparity (National Council on Disability, 2011). A less expensive way for people with disabilities to obtain access to the internet is through mobile devices. For this reason, not only phones but the information displayed on them need to be accessible. There were more than one million apps in both Apple and Android app stores by the end of 2013 (Apple Press Info, 2014; Kameka, 2013). With this pace of app development, it is essential for developers to know how to make them accessible. When the Access Ratings for Buildings (AR-B) research and development team sought to create an app for users with disabilities, accessibility was a main priority (Schwartz & Smith, 2013; Schwartz, O'Brien, Edyburn, Ahamed & Smith, 2013; Edyburn, Schwartz & Smith, 2013). While there is much information about creating accessible websites and devices, a paucity of information is available regarding methods to assure mobile app accessibility. The present study investigated the challenges involved in creating accessible mobile apps. The study centered around experiences and knowledge gained during the development of mobile applications through the Access Ratings for Buildings (AR-B) project.

METHODS

To better understand app accessibility resources for developers, we engaged in a two phase process; a literature review followed by key informant interviews. The literature search, included searches of relevant scholarly literature, programming guides from mobile device manufacturers (Apple and Android), and web content guidelines (W3C). Next, three AR-B team software developers/programmers were interviewed regarding their knowledge and opinions on app accessibility: Developer 1 was the lead developer for

the AR-B Trained Rater app; Developer 2 was the lead developer creating the AR-B Consumer app; and Developer 3 was the computer programmer developing the AR-B Web app.

RESULTS

Literature Review

A review of the literature revealed that resources for developing accessible apps are widespread and contain extensive detail. Current standards for mobile accessibility are included in those for web content. The most commonly referenced standards are the World Wide Web Consortium's (W3C) guidelines, which recommend that web and mobile content be perceivable, operable, understandable, and robust. Specifically, to make content perceivable, developers should provide text alternatives for non-text content, provide captions and other alternatives for multimedia, create content that can be presented in different ways, and make it easier for users to see and hear content. To make it operable, make all functionality available from a keyboard, give users enough time to read and use content, do not use content that causes seizures, and help users navigate and find content. For content to be understandable, developers should make text readable and understandable, make content appear and operate in predictable ways, and help users avoid and correct mistakes. To be robust, design should maximize compatibility with current and future user tools. Although there are standards that apply generally to software applications and documented techniques for each OS, the steps necessary to implement them widely varies dependent on the mobile platform.

Inconsistency Across Platforms

The participants reported that a main challenge in developing accessible apps is the lack of versatility within operating systems. Web applications, Apple's iOS, Google's Android, and other operating systems all provide different frameworks for programming accessibility features. Each framework provides capacity for different types of information and levels of detail. As a result, participants noted that developers must create separate applications for different

operating systems, each of which has distinct accessibility frameworks. Cross-platform compilers and responsive design have been introduced as solutions to this problem. Cross-platform operating systems enable developers to create applications for multiple platforms using a single code base (e.g. Appcelerator). Also, responsive design provides custom layouts to devices based on the size of the browser window (Robbins, 2012). Although cross-platform and responsive design resolve the need to create separate apps for different operating systems in theory, they are restricted in their ability to implement device-specific accessibility information. This is due to the unique complexity of each operating system. For example, Developer 2 noted that where Android enables a developer to label what an interface feature is, iOS also enables a developer to “add explanation” on what a feature does through what are termed accessibility hints. The extended capacity for information through the hint attribute also provides space for more advanced accessibility features, such as Equivalent Text Descriptions, which are written descriptions of graphic elements (Rehabilitation Design and Disability Center). By contrast, Android's accessibility programming interface does not provide space for descriptions of images. Developer 1 explained that an advantage of developing with the Android operating system is that the team has been able to gain access to several device features that the Trained Rater app will utilize, whereas this has not been the case for iOS.

Table 1: Characteristics of Development Platforms

Platform	Characteristic
iOS	<u>Accessibility Programming Interface:</u> - Capacity for detailed information - Consistency across apps due to limited SDK options and access to features - Straight-forward - Developers can provide accessibility information through the following attributes: Label, Trail, Hint, Frame, and Value <u>Programming Considerations</u> - Limited design options may restrict app functionality
Android	<u>Accessibility Programming Interface:</u> - Confusing - Provides developers with

	access to all features - Developers can provide accessibility information through content descriptions with the android:contentDescription XML layout
Cross-Platform	While they attempt to implement complete cross functionality, they are not seamless with device-specific implementations that can facilitate accessibility
Responsive Design	Does not utilize device-specific features that facilitate accessibility

Lack of Support and Resources

The developers reported that they received minimal to no training in the area of accessibility, and must rely primarily on dense standards guides, as well as input from disability experts if available. Resources available for creating accessible apps provide little guidance for developers. Programming guides from manufacturers such as Apple and Google are the most comprehensive app-specific resources available, but do not offer procedural guidance and may be confusing (Apple Developers; Android Developers). Developer 2 explained that developers typically rely on the software development kit (SDK) for guidance, but programming at this level provides for only basic accessibility. Developer 3 expressed that W3C standards and guidelines have similar limitations, with “very specific, low level detail” and a paucity of examples. In addition, there is minimal support from companies to assist developers throughout the process. With minimal support and resources, developers undergo a complex process to create app accessibility.

Complexity of App Development

The participants reported that the process of creating accessibility is complicated and frequently relies on trial and error rather than a stepwise process. In general, approaching app accessibility begins with an understanding of what an accessible app looks like and what features are entailed. A significant amount of interpretation is involved when using standards guides as a resource. A developer must apply each standard to the app they are developing with the options available for each development platform. Varying degrees of flexibility and interpretation are involved depending on the specific operating system or other platform used. Developer 2 noted that the SDK for each operating system provides a basic level of guidance for implementing accessibility features. For

example, most SDK's prompt the developer to insert descriptive information for on-screen features that can be conveyed to blind users via screen readers. However, the capacity for detail varies based on the operating system. Since web application development involves few restrictions, Developer 3 explained that applying accessibility standards frequently requires starting from scratch. The process of trial and error becomes increasingly important as the app becomes more customized, noted Developer 2. Developer 1 explained that a developer may seek third-party or open-source algorithms to implement a desired feature or overcome restrictions of a specific operating system's SDK.

DISCUSSION

Standards and Procedures Are Needed

Existing standards for creating accessible software is exceedingly dense, requiring a painstaking effort on the part of developers to read and understand the content. There is a lack of easy to use examples of accessibility features within the standards, making it difficult to apply them to apps under development. Furthermore, since the best current standards were designed for web content, standards directed specifically towards mobile applications may be more clear and of greater use. The development of accessible apps could be significantly aided by the creation of stepwise procedures, which would avoid the current need to utilize a trial and error process when implementing accessibility.

Consistency Across Platforms is Needed

Achieving greater consistency across platforms would significantly reduce the amount of time and effort necessary to create accessible apps. Currently, developers must create a separate app for each operating system and understand the accessible features build into each one's accessibility programming interface. Consistency across platforms would simplify this process.

Address Lack of Support and Resources

Developers, practitioners and policy makers must respond to the lack of support and resources for creating accessible apps. Developers should be aware of the time and effort needed to make an app accessible, given the lack of support, resources, and procedures available to them. Practitioners should be aware that the majority of apps available are not accessible, and that alternative methods of

access may be necessary for people with disabilities to use some applications. This may take place through the use of assistive technology or other methods. Last, it is essential for policy makers to respond to this lack of support by overseeing procedures to enhance consistency across platforms and providing greater support for application development.

CONCLUSION

This study highlights the complexity of creating accessible mobile applications as a process requiring trial and error strategies on the part of developers. It is further complicated by the fact that operating systems are very different from one another, and that there are minimal resources and support available to assist developers. These findings highlight a need for the creation of standards and procedures, for increased consistency across platforms, and for policy makers, practitioners, and developers to respond to these needs. These actions will help construct a more fluid process for creating accessible apps.

ACKNOWLEDGEMENTS

The AR-B Project is supported in part by the Department of Education, National Institute on Disability and Rehabilitation Research (NIDRR), grant number H133G100211. The opinions contained in this poster do not necessarily represent the policy of the U.S. Department of Education, and you should not assume endorsement by the Federal Government.

REFERENCES

- Android Developers. Accessibility. Retrieved from: <http://developer.android.com/guide/topics/ui/accessibility/index.html>
- Apple Developer. Accessibility in iOS. Retrieved from: <https://developer.apple.com/technologies/ios/accessibility.html>
- Appcelerator Inc. Titanium SDK. Retrieved from: <http://www.appcelerator.com/titanium/titanium-sdk/>
- Apple Press Info (2014). App Store Sales Top \$10 Billion in 2013. Retrieved from:
- Dobra sky, K. & Hargittai (2006). The Disability Divide in Internet Access and Use. *Information, Communication, and Society*, 9(3), p. 313-334.
- Edyburn, K., Schwartz, J., Smith, R.O., (2013). A case study: Development of Access Ratings for Buildings "Consumer" mobile app. Paper presented at the Rehabilitation Engineering and Assistive

Technology Society of North America 2013
Conference, Bellevue, Washington, USA.

Harris Interactive Inc. Lou Harris Poll on Disability.
Retrieved from:

http://www.google.com/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CCwQFjAA&url=http%3A%2F%2Fwww.aboutdisability.com%2Fresources%2FHARRIS.DOC&ei=VqtzUteVA_KssASGI4GoBA&usg=AFQjCNGwLmPtH0g57O0fCCmCDTzEXoLcjw&sig2=M0sXbUaLfA2ILJ6xHIM8aA&bvm=bv.55819444,d.dmg

Kameka, A. (2013). 1 million Android apps are now available in Google Play. Retrieved from: <http://www.mobileburn.com/21859/news/1-million-android-apps-are-now-available-in-google-play>

National Council on Disability (2011). National Disability Policy: A Progress Report. Retrieved from: http://www.ncd.gov/progress_reports/Oct312011

Rehabilitation Research Design and Disability Center. Equivalent Text Descriptions. Retrieved from: Retrieved from: <http://access-ed.r2d2.uwm.edu/EqTDs/>

Robbins, J. N. (2012). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media: Sebastopol, CA

Schwartz, J. K., & Smith, R. O. (2013). Access Ratings for Buildings: Measuring Building Accessibility in the Community Environment Paper presented at the Second Annual Occupational Therapy Summit of Scholars, Chicago, IL.

Schwartz, J., O'Brien, C., Edyburn, K., Ahamed, S.I., Smith, R.O. (2013). Smartphone based solutions to measure the built environment and enable participation. Paper presented at the Rehabilitation Engineering and Assistive Technology Society of North America 2013 Conference, Bellevue, Washington, USA.

World Wide Web Consortium. Mobile Accessibility. Retrieved from: <http://www.w3.org/WAI/mobile/#covered>