Increasing Accessibility at the Worcester Art Museum Via A Mobile Application

A Major Qualifying Project Proposal
submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the degree of Bachelor of Science

Submitted To:
Professor Rodica Neamtu
Professor Lane Harrison

Submitted By:
Michael Zeolla,
Marko Vila

Date:
4/25/2023

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see http://www.wpi.edu/Academics/Projects.
Abstract

In 2022 the Worcester Art Museum (WAM) hosted an exhibition with tactile sculptures for those with visual impairments while also analyzing practices for making the museum more accessible. However, in most cases, traditional exhibits do not support blind or low-vision guests, providing the WAM with an opportunity to explore different methods of supporting these guests besides specialized exhibitions. One potential solution decided upon by the WAM was a mobile app that would be a useful accessibility tool for visually impaired persons (VIPs). To this end, the museum partnered with Worcester Polytechnic Institute (WPI) to develop a mobile application to enhance the experience at the WAM for all types of VIPs.

The Worcester Art Museum App aims to empower visually impaired people to access all art pieces and exhibits in the WAM. One of the main goals is to produce an application for all users, widely accessible to the public, and to accomplish this, the app was developed using React Native. React Native allows for development on dual platforms, Android and iOS, allowing the app to reach a larger audience while also taking advantage of built-in accessibility features. The major features of the app included the ability to have the phone read out UI options present on the screen, read out audio files, use modified touch gestures to navigate the screen, the ability to customize the app color palette, and give the user the option to control the app with their voice. The long-term goal of the WPI team was to work with the museum to add physical alterations to the exhibit to improve accessibility and integration with the app, most notably in the form of NFC tags and QR codes. By providing supplemental information about the museum’s exhibit art pieces through text and audio descriptions, the role of the application is to expand access to exhibits to people with visual impairments.
Acknowledgments

The WPI development team would like to thank Professor Rodica Neamtu and Professor Lane Harrison for their guidance and support during the project, as well as Marnie Weir and Laura Taylor for representing the interests of the Worcester Art Museum throughout this past academic year, and Harry Duchesne from Audio Journals for their continuing advice throughout the project. Finally, we would also like to thank all the application testers who spent their time providing valuable feedback and criticisms on the project.
Authorship

* Michael Zeolla and Marko Vila began working on the project alongside other team members in late August. They both made equal contributions to the project alongside other team members. Following the end of C term on March 6th 2023, Marko Vila and Michael Zeolla will continue working on the MQP until the end of D term in May 2023 without the remaining group members. Their work will continue building features for the app, potentially standardizing the method of creating accessibility centered apps for future WPI teams. This is the final report containing both the work produced by the entire WAM team, including Michael and Marko, and the extended work completed by Michael and Marko as part of their double major requirement.

<table>
<thead>
<tr>
<th>Section</th>
<th>Primary Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Acknowledgment</td>
<td></td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>Daniel Rabinovitz</td>
</tr>
<tr>
<td>Chapter 2: Background</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>The Worcester Art Museum</td>
</tr>
<tr>
<td>2.2</td>
<td>Audio Journal</td>
</tr>
<tr>
<td>2.3</td>
<td>Visually and Print Impaired Persons</td>
</tr>
<tr>
<td>2.4</td>
<td>VIPs Challenges in Museums</td>
</tr>
<tr>
<td>2.5</td>
<td>Common Accessibility Features/Practices</td>
</tr>
<tr>
<td>2.6</td>
<td>Other Museums’ Solutions</td>
</tr>
<tr>
<td>2.6.1</td>
<td>The Madrid Typhlological Museum</td>
</tr>
<tr>
<td>2.6.2</td>
<td>American Association for State and Local History</td>
</tr>
<tr>
<td>2.6.3</td>
<td>Manchester Museum</td>
</tr>
<tr>
<td>Chapter 3: Methodology</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Project Goal</td>
</tr>
<tr>
<td>3.2</td>
<td>Organizational Process</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Project Information Structure</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Agile Development Process</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Project Management System</td>
</tr>
<tr>
<td>3.3</td>
<td>Proposed App Features and Research</td>
</tr>
<tr>
<td>3.3.1</td>
<td>WAM Minimum Viable Product</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Accessibility Navigation Research</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Color Contrast Navigation Research</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Voice Prompt Improvements</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Near-field Communication Research</td>
</tr>
<tr>
<td>3.4</td>
<td>User Interface/User Experience</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Application UI Mockup</td>
</tr>
<tr>
<td>3.4.2</td>
<td>iOS Accessibility Guidelines</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Android Accessibility Guidelines</td>
</tr>
</tbody>
</table>

**Chapter 4: Implementation**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Project Timeline and Ticket Management</td>
<td>Vrandol Perez</td>
</tr>
<tr>
<td>4.2</td>
<td>Implemented Application Features</td>
<td>Michael Zeolla</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Audio Player</td>
<td>Michael Zeolla</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Color Customization</td>
<td>Marko Vila</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Voice Prompts and Voice Navigation</td>
<td>Marko Vila</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Accessibility Navigation Support</td>
<td>Daniel Rabinovitz</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Deep Linking and Near-field Communication Tags</td>
<td>Michael Zeolla</td>
</tr>
<tr>
<td>4.2.5.1</td>
<td>Deep Link Challenges</td>
<td>Michael Zeolla</td>
</tr>
<tr>
<td>4.2.5.2</td>
<td>Deep Linking and NFC App Implementation</td>
<td>Michael Zeolla</td>
</tr>
<tr>
<td>4.2.5.3</td>
<td>NFC Tag Scanner Bottom Sheet</td>
<td>Michael Zeolla</td>
</tr>
<tr>
<td>4.3</td>
<td>Application User Interface/Experience</td>
<td>Marko Vila</td>
</tr>
<tr>
<td>Section</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>4.3.1 App Design Guidelines in Practice</td>
<td>Zane Carey</td>
<td></td>
</tr>
<tr>
<td>4.3.2 App Page Architecture</td>
<td>Zane Carey</td>
<td></td>
</tr>
<tr>
<td>4.3.3 Common Application Flows</td>
<td>Zane Carey</td>
<td></td>
</tr>
<tr>
<td>4.4 Backend Implementation</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>4.4.1 Outsourced Database Hosting</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>4.4.2 The WAM App Backend</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 5: User Study and Results</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1 User Study Procedure</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>5.2 Testing Results</td>
<td>Zane Carey</td>
<td></td>
</tr>
<tr>
<td>5.2.1 Visual and Print Impaired Tester Results</td>
<td>Daniel Rabinovitz</td>
<td></td>
</tr>
<tr>
<td>5.2.2 Non-Visually Impaired Testing Results</td>
<td>Vrandol Perez</td>
<td></td>
</tr>
<tr>
<td>5.2.3 Overall Testing Results</td>
<td>Zane Carey</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 6: Future Works</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1 Additional Future Work/Features</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>6.2 Recommendations for Future Teams</td>
<td>Marko Vila</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 7: Conclusions</strong></td>
<td>Everyone</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 8: Extended Work</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.1 Brief Project Recap and Final Steps</td>
<td>Marko Vila</td>
<td></td>
</tr>
<tr>
<td>8.2 Generalizing Accessible Applications</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>8.3 Applied Generalized Process and Ideas</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>8.3.1 EcoTarium Background Information</td>
<td>Marko Vila</td>
<td></td>
</tr>
<tr>
<td>8.3.2 EcoTarium Early Development Process</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>8.3.3 Feature Customization and Implementation</td>
<td>Michael Zeolla</td>
<td></td>
</tr>
<tr>
<td>8.3.4 EcoTarium Live Map User Assistance</td>
<td>Michael Zeolla and Marko Vila</td>
<td></td>
</tr>
<tr>
<td>8.4 Extended Works Conclusion</td>
<td>Michael Zeolla and Marko Vila</td>
<td></td>
</tr>
</tbody>
</table>
1. Introduction

Visual impairments come in a variety of forms such as color blindness, blurry vision, and total blindness, with each different type requiring different accommodations. The visually impaired community has had the creativity to overcome their unique challenges. Solutions often involve the liberal use of touch and sound. American art museums have demonstrated a storied history of creating exhibits that support all visitors through tactile sculptures, even as far back as the early 1970s, decades before the Americans With Disabilities Act (Haines, 2021). The WAM recently continued this history by creating tactile sculptures for a non-Visually focused exhibition. The museum was also looking for ways to provide additional support for visitors with vision or print impairments.

The accommodations required for VIPs in museums fall into two main categories. The first category involves navigation. Many museum layouts that are simple for a fully sighted person to navigate provide safety hazards for VIPs. VIPs unfamiliar with a museum’s layout will often require assistance from a companion visitor, service dog, or museum staff. Technological solutions for low-vision navigation is a developing field that is beyond the scope of this project. The second category is print and visual media. There are a variety of disabilities, both visual and non-visual in nature, that require accommodations such as a text reader, large text, or a close-up view. A mobile app is well-suited to provide print and visual media accommodations.

The WPI team implemented a mobile app for the Worcester Art Museum (WAM) with accessibility features modeled on the previous Audio Journal iOS app with some additional museum-specific features. Reception to the WAM app, called “Beyond the Visual” (BV), has been positive in user testing.
I I. Background

In the past, WPI partnered with Audio Journal, a nonprofit in Central Massachusetts that serves the visually impaired by broadcasting readings of news publications, stories, and live radio theater (“Mission and History,” n.d.). Students at WPI developed the Audio Journal App, an iOS application to make Audio Journal broadcasts more accessible (Grigolia, Marison, and Doyle, 2020). Before this project, WAM had independently collaborated with both Audio Journal and WPI to produce accessible experiences for the visually impaired (Staff, 2018). The collaboration of these three organizations led to a partnership in which WPI students created a mobile app that assists visually impaired visitors throughout the Museum. The initial app targeted a small community, but it would be beneficial for them if there was an accessible app for the entire Museum that was similar in nature to the one they have been using.

2.1 The Worcester Art Museum

The Worcester Art Museum is a museum located in Worcester, MA with a mission to connect people, communities, and cultures through the experience of art. Founded in 1896, the WAM houses a public collection of over 38,800 art pieces, representing 51 centuries of creative spirit and cultural heritage containing world-renowned European and American art, fine Roman mosaics, Japanese prints, and much more (Welcome! | Worcester Art Museum, n.d.). In its mission of connecting everyone in its community, the WAM wanted to appeal to visitors who have visual impairments. They approached this hurdle through the development of a mobile application for both iOS and Android devices, intended to reach a broad audience.
2.2 Audio Journal

Audio Journal is an organization based in Central Massachusetts to “connect individuals with a visual impairment, or an inability to access print material, to their communities through the broadcasting of local news, information, and entertainment, with exclusive programs and content” (“Mission and History,” n.d.). Audio Journal was first introduced as a Major Qualifying Project (MQP) in 2021 when WPI undergraduates gathered their talents to create a radio reading service for the blind and visually impaired (Grigolia, Marison, and Doyle, 2020). The team focused on broadcasting local news to the visually impaired community in Central Massachusetts and made their content accessible to users by creating an iOS application called the Audio Journal App. As an MQP, BV underwent several iterations of development and testing stages over the course of 21 weeks. This project was later taken on by another team of WPI students who expanded on the accessibility features (Grigolia, Marison, and Doyle, 2020).

In 2022 the Audio Journal’s second MQP team developed new features which advanced the usability of the application. These features included an improved media player with playback features, as well as advancements in searching for programs (Grigolia, Marison, and Doyle, 2020). They also conducted extensive testing with current users of the Audio Journal app to improve upon the features they had developed during the original period in the prior MQP.

2.3 Visually and Print Impaired Persons

Both the Audio Journal and Worcester Art Museum app focus on providing a valuable experience to users with visual impairments. However, it is important to understand what visual impairments are and how they can impact everyday life. Visual impairment is defined as a decrease in the ability to see to a certain degree which causes problems not solvable by usual means. Visual impairments are not to be confused with blindness, which is defined as the state of
being unable to see due to injury, disease, or genetic conditions (Blind vs. Visually Impaired: What’s the Difference? | IBVI | Blog, 2022). Visually Impaired Persons or Print Impaired Persons (PIPs) have a variety of challenges consuming visual forms of media. Media accommodations for VIPs are necessary everywhere - in Massachusetts, for example, approximately 3% of the population is visually impaired (National Federation of the Blind, 2019).

Visual impairments and blindness come in several forms and can be easily confused at times. There are four main terms used to refer to visual impairment and blindness levels: partially sighted, low vision, legally blind, and blind. A partially sighted individual has partial vision in one or both eyes. An individual with low vision has a severe visual impairment in which visual acuity is 20/70 or worse in the better-seeing eye and cannot improve with glasses or contacts. A legally blind individual has a corrected vision of 20/200 in their best-seeing eye. Finally, complete blindness refers to someone that has complete loss of sight in both eyes (Blind vs. Visually Impaired: What’s the Difference? | IBVI | Blog, 2022). However, visual impairment extends to other conditions, such as color blindness, which does not fit into the above terms.

Print-impaired persons are separate from the visually impaired. Print-impaired individuals are unable to read printed material due to dexterity problems, learning disabilities, or cognitive impairment. Dexterity problems are potentially caused by diseases like Parkinson's, multiple sclerosis, and arthritis. Thus, making it extremely difficult to hold printed material. Furthermore, individuals with learning disabilities such as dyslexia are challenged when attempting to read and interpret printed materials. Lastly, print impairments are also caused by cognitive impairments, including dementia and other brain injuries.

Furthermore, it is evident that visually and print-impaired individuals have difficulty accessing and reading printed materials. Materials such as newspapers, books, magazines, and
other forms of printed materials become a challenge to access without alternative methods such as audiobooks or audio journals. Thankfully, most libraries have audiobooks, and a majority of the larger news companies broadcast on the radio. While these two methods allow visually and print-impaired persons to access their content, there are still gaps in accessibility for VIPs. Local news channels, newspapers, and broadcasts remain mostly inaccessible.

More recently, companies such as Audio Journal, a non-profit organization based in Worcester, Massachusetts, are attempting to provide assistive technologies for VIPs and extend the accessibility of niche content, such as local news and broadcasts, to those that are visually and print impaired. Audio Journal records and broadcasts local news specific to the visually and print-impaired people in the city of Worcester. In the effort to continue this expansion of material for VIPs, the Worcester Art Museum decided to implement similar practices within their museum with the hope of reaching a wider visually impaired audience.

2.4 VIPs Challenges in Museums

In addition to companies that attempt to aid VIPs, many institutions are starting to implement their own accessibility measures. Museums are an example of these institutions that seek to make the best experiences for VIPs when they visit. Evelity, an app for guiding users of all disabilities, conducted a study in France in 2021 where they sent out questionnaires to find out what troubles VIPs were having in museums (Mathieu. (2021). The scope of the study included a total of 53 museum attendees, where 9% were under 29 years old, 51% were between the ages of 30 and 59 years old, and 40% were over 60 years of age (Mathieu. (2021). Their study identified 6 main challenges VIPs face which usually prevent them from visiting. These challenges are lack of adapted content, lack of accessibility inside the museum, lack of guided tours (for VIPs), lack of accessibility in the museum’s surroundings, lack of information about
the content, and lack of accessibility by public transport (Mathieu. (2021)). These results of the study were outlined in their infographic, seen in Figure 1 below. Their results show the biggest issue regarding museum visitation was the lack of adapted content; that is, content that VIPs can interact with. The next two largest issues were identified as a lack of accessibility inside the museum and adapted guided tours.

![Figure 1: VIPs reasons for not visiting museums](image)

### 2.5 Common Accessibility Features/Practices

When developing an accessibility application, it is important to keep in mind the group you are representing, in this case, the VIPs. To that end, the application developed for the WAM included features that support the VIP community rather than limit them. Apps focusing on VIPs should rely heavily on audio prompts, as opposed to the more common method of visual prompts. Accessibility is not just about making information available to people with disabilities but about making said information available to everyone, regardless of their capabilities or situation. With this in mind, many companies, including Apple, and Samsung, have created several guidelines to assist developers in creating useful and accessible apps.
2.5.1 VoiceOver

One of the most prevalent features in terms of audio prompts comes in the form of reading written text to the user. Apple’s VoiceOver and Androids TalkBack feature allow the user to hear audible descriptions of the content on their screen, aiding said users to navigate through applications when sight proves difficult. VoiceOver now has more detailed descriptions of persons, things, text, and graphs than ever before. With the use of a Bluetooth keyboard or simple motions on a touchscreen or trackpad, you can browse a screen with ease thanks to audio explanations of its components. Additionally, you can make information like webpages simple to view by using special rotor movements that work like a dial on touchscreens and trackpads. VoiceOver has hundreds of voices to pick from and is available in more than 60 languages.

To properly implement these design schemes, developers are required to provide their descriptions for all the visual components they include in the application. The most effective VoiceOvers don’t only provide the obvious details, such as the self-explanatory report that an able-bodied individual would give, but should focus on the information disclosed by the image. With this as part of the app, the VIPs can navigate through the app with ease since this feature utilizes accessibility information from elements to assist users in discerning element locations.

2.5.1.1 VoiceOver and Braille

Even in braille, VoiceOver conveys exactly what is seen on a screen. You may type braille directly on the touchscreen using Braille Screen Input or use a refreshable braille display that is Bluetooth-connected to an Apple device. For sighted persons to follow along with VoiceOver explanations, Braille is immediately translated to text in a caption panel on Mac. Apple Watch is compatible with Bluetooth-enabled braille keyboards and refreshable braille
displays. Users may select the best braille experience for them among the more than 70 refreshable braille displays that are compatible with iOS and iPadOS and more.

2.5.2 Text Display

The next large design goal when building accessibility apps is Dynamic Sizing also referred to as Dynamic Type. An application implementing Dynamic Type reflects any changes made to an app’s layout by the user in the form of adapting font sizes and letting users pick what’s best for them. On devices such as iPhones or iPads, this tool is easily enabled under the accessibility text sizes in the system settings. According to Apple, as the font size increases, it’s best to keep truncation within the content minimum (Apple Inc.). Another helpful tip is to adjust the layout of the application as the font size increases so that the text is readable. To illustrate, on a normal scale, the text might be fine along with images, but as the font size is enlarged the text might become unreadable. The size of interface icons will also need to change along with the font for their importance to stay intact for those in need of larger visuals.

2.5.3 Application Color and Effects

There are a handful of rules to follow when using colors to help the VIPs. First, developers should not rely on color to differentiate between elements or infer that something is important. A rule of thumb for developers is to integrate system colors within the app so that it may work in conjunction with contrast and inverted colors. Additionally, color combinations should be avoided when differentiating between two states or values since colorblind individuals have difficulty distinguishing between specific color combinations. In terms of color combinations, it is important to note that strongly contrasting colors may help improve readability. On the other hand, animations within the application should be kept to a minimum since this can unintentionally confuse visually impaired users (Apple Inc.).
2.6 Solutions Adopted By Other Museums

When developing any application, it is important to draw research specific to the topic at hand and understand the history behind the research. In this case, relevant information would be concerning museums of equivalent size to WAM. Across the world, there is a storied history of accommodations for the blind. The world’s first school for the blind, The Institute for Blind Youth, was established in Paris in 1786. The braille alphabet was developed during the period from 1821-1824 by Louis Braille, an adolescent student at The Institute for Blind Youth (American Foundation for the Blind, n.d.). The Institute for Blind Youth inspired Americans to charter a similar institution in 1829, the Perkins School for the Blind, as the first school for the visually impaired in the United States (Perkins School for the Blind, n.d.). However, it took decades for museum accessibility to enter the public consciousness. The first recorded exhibit for the visually impaired population in America was a tactile exhibit of taxidermied animals at the American Museum of Natural History in New York in 1909, but the idea of art exhibits for VIPs took hold in the 1960s and 1970s alongside the disability rights movement (Haines, July 2021).

The general layout of museum exhibits for VIPs at the time was to create art, usually, sculptures, designed to be touched by museum visitors and to have museum staff act as guides for any VIPs at the exhibit. Often the artists themselves would be visually impaired. Seeing and non-seeing visitors positively received these early exhibits (Haines, 2021). However, as technology and disability rights have progressed, museums' ways to accommodate visitors have changed. Most museums focus on adding physical accommodations such as ramps, handrails, and braille or audio versions of content to existing exhibits rather than designing new exhibits specifically for VIPs. However, the practice of creating exhibits or even entire museums specifically for the VIP community is still alive and well.
2.6.1 The Madrid Typhological Museum

The Typhological Museum (El Museo Tiflológico) in Spain is a museum designed entirely for VIP visitors. The museum has four main exhibit showcases: an exhibit of dioramas of world monuments works of art by VIP artists, an exhibit of tactile writing systems and documents from throughout history, and rotating temporary exhibits related to blindness (Madrid Office of Tourism, n.d.). The museum has developed a unique accessibility technology called beepcons (Phoneia.com, 2017), depicted in Figure 2 below.

![Figure 2: Beepcon Example (Diariocrítico.com, September 2017)](image)

Beepcons are small, battery-powered Bluetooth beacons that can connect to the Beepcon mobile app. Beepcons are placed near key locations such as elevators and are labeled in the app with the name of the key location. When selected on the mobile app beepcons beep loudly to help the user ascertain their direction and distance from the desired location. However

The Madrid Typological museum has a standardized placement of QR codes (Rabinovitz, 2022). Each QR code is placed in the top right corner of the sign for each item, accompanied by
either small bumps or a texture change around the corners of the QR code.

Figure 3: Typhlological Museum QR code example

Another notable accommodation was the museum’s use of a guiding trail on the floor of the building. The transitions between exhibit locations would be marked by a tiled path while exhibit spaces would use carpeted flooring. The combination of Beepcons, tactile exhibits, QR codes, audio tours, and other accessibility tools are combined for a great experience for VIP visitors. However, the beepcon technology is still new and difficult to find outside of Spain (Phoneia.com, 2017).

2.6.2 American Association for State and Local History

The American Association for State and Local History (AASLH) is a nonprofit focused on assisting history museums with growth and professional development. The AASLH partners with dozens of museums across the United States of varying needs and sizes. (ASSLH, n.d.). Many of these partner museums have small budgets and thus require cost-effective accessibility solutions that place little strain on the already limited hours of museum staff and volunteers. The AASLH has recommendations for VIP accessibility solutions for member institutions with small budgets (Klein et al., n.d.).
The first solution is to have museum materials such as exhibit labels and exhibit descriptions in a downloadable format on the museum website. This allows visitors to customize their accessibility experience at home before attending the museum. The second solution is to implement QR codes across the museum. The utilization of QR codes allows visitors to scan an image with their phone camera to access audio playback of exhibit information or even to access interactive experiences. The AASLH also recommends standardizing the placement of QR codes across all museum pieces, such as placing a QR code in the corner of every display sign.

2.6.3 Manchester Museum

The Manchester Museum is the United Kingdom’s largest university Museum and serves hundreds of thousands of visitors each year. The museum exhibits focus on archaeology, anthropology, and natural history (Blake, November 2015). The Manchester Museum’s Visitor Team has implemented a combination of both NFC tags and QR codes in their past exhibit “Nature Through Roman Eyes” (Ludolini, August 2018).

The Manchester Museum used NFC tags to implement a variety of accessibility options such as giving users audio playback of museum text on their phones, showing foreign visitors museum text in other languages, and more (Ludolini, August 2018). However, NFC tags have a strict limitation on distance, and some smartphones still cannot interface with NFC tags. As a result, QR codes were sometimes paired with NFC tags to provide another vector of accessibility. Visually impaired visitors at the Manchester Museum gave stellar reviews of the NFC tag experience. However, unlike the WPI team, the Manchester Museum chose to create a website instead of a dedicated mobile app.
II. Methodology

The methodology section of the report outlines the different methods and practices the development team implemented during their time working. It covers the goal of the project, how to reach that goal, and what organizational practices were implemented. This section also covers any additional research which was conducted on different topics.

3.1 Project Goal

The Worcester Art Museum App aims to empower visually impaired people to access the Worcester Art Museum exhibits. One of the main goals of the project is to produce an application for all users, widely accessible to the public. To reach a wider audience, BV was developed using the development framework, React Native. This development framework is ideal for mobile apps with an easy-to-follow user interface. Additionally, using React Native allows the Worcester Art Museum app to be placed on both the iOS app store and the Android app store. This allows the application to reach a larger target audience that doesn’t rely on the device operating system.

3.2 Organizational Process

During the development phase, the team took several steps toward creating a comprehensive system for managing and completing team goals. These steps included creating a specific structure for meetings, minute-keeping, and goal assessment for long-term and short-term goals (over 2 weeks). The team implemented multiple software, practices, and methodologies to improve the team's overall effectiveness. Additionally, to help productively facilitate effective work there was a general implementation of the AGILE development pattern.
3.2.1 Project Information Structure

When developing any large software project, there is often a large amount of information, most of which is critical to the development process. In particular, this information presented itself in the form of research, meeting minutes, meeting agendas, project documentation, powerpoints, and ticket tracking. One specific technology implemented by the team was OneDrive, which was used to create a system for storing such information. This OneDrive includes all of the team notes from the group, advisor, and sponsor meetings, as well as all other miscellaneous documents that may in some way reference or aid the project.

From there, the team created a GANTT chart, a bar type that illustrates a project schedule, for them to plan out the timeline for the deliverables. To illustrate, the GANTT chart is split into three sections, one for each term. To effectively implement the GANTT chart the team defined a set of deliverables and dates. Figure 3 shows a section of the team's GANT chart, which was used to measure progress along with their deliverables, and completed deadline.

![GANTT Chart](image-url)
3.2.2 Agile Development Process

The WPI team also implemented a common software development methodology known as Agile development, which is a flexible and responsive design method. The process was originally proposed in the 1990s and has been the industry standard ever since. This development process helps developers solve issues in a timely yet effective manner while offering the freedom to work as they please (Krancher, 2020). The Agile workflow spans short quarters, known as sprints, which are typically two weeks long and encompass multiple different phases, of which the most important ones are planning, development, and testing. On top of sprints, the team incorporated daily check-ins, also known as standups. These standups help foster collaboration between the team and track progress. The WPI development team held a standup meeting every day of the week, with the majority of the meetings in the afternoon. The team also met with the Worcester Art Museum frequently to ensure that the sponsors had their expectations met. The team also hosted weekly check-in meetings with the WPI advisors for guidance on what they could improve and reassurance that the team was on pace for completion.

3.2.3 Project Management System

Another important step in the development process is maintaining an accurate record of the project’s progress, typically in the form of what needs to be completed, what has been completed, and what is actively being worked on. A proper management process/system can help keep the project on track and manage expectations. Most commonly, different tasks for the project are stored as a “ticket” which describes the objective, who is working on the ticket, and the priority of the ticket. A popular online content management system is Jira, which implements a system for creating, assigning, and tracking project tickets. Jira allowed team members to keep
tabs on the overall project progress for themselves. Another valuable feature Jira provides is the
ability to organize sprints. Figure 4 displays a Jira dashboard with tickets and a Jira sprint board.

Figure 5: Jira Ticket Visual

3.3 Proposed App Features and Research

3.3.1 WAM Minimum Viable Product

When developing, it is common to design a Minimum Viable Product (MVP) for an application, a set of essential features. The features included in the MVP for the WAM app include the ability to play audio, the ability to display text associated with the art piece, accessible voice navigation, and color palette customization. The list was intended to highlight the essential features, good-to-have features, and stretch goals for the application. All of these features have currently been implemented in the WAM app.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed Audio Player</td>
<td>A simplistic designed audio player within the app. Play, pause, skip and replay the audio,</td>
</tr>
<tr>
<td>Simplistic App Views</td>
<td>Easy to use and understand application views for those VIP users.</td>
</tr>
<tr>
<td>Accessibility Navigation</td>
<td>Native Android and iOS support for TalkBack and VoiceOver respectively.</td>
</tr>
<tr>
<td>Color Palette Customization</td>
<td>Ability for the user to customize the theme based on a preset list of color options.</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Voice Prompted Navigation</td>
<td>Ability to speak into the application, and immediately navigate to the spoken page.</td>
</tr>
<tr>
<td>NFC Tags + QR Codes</td>
<td>Deep Linking within the application, accompanied by NFC tags and QR codes.</td>
</tr>
</tbody>
</table>

![Figure 6: Application Features Table](image)

### 3.3.2 Accessibility Navigation Research

For any application focusing on accessibility, a crucial feature is the integration of easy and accessible navigation. There are two main components for accessible navigation: interactivity and voice overlay. Interactivity is commonly accomplished by providing different actions based on how the view was clicked, and voice overlay is accomplished by reading text out loud to the user. For iOS and Android devices, voice overlay is handled via built-in packages, such as Voice Over for iOS and Talkback for Android. Both packages are meant to provide VIPs with the same amount of information that is present to sighted application users (Celusnak, 2015 & Gregorio, Palomba, Nucci, et al., 2022). The WAM application relied heavily on implementing the proper practices to create a seamless experience for accessible navigation.

React Native provides an easy wrapper for implementing the same native voice accessibility. To accomplish this, you must assign hidden properties to each element visible on the screen. One of the most common types of hidden properties is hidden text, known as the accessibility label and accessibility hint (Vinkle, 2021). The accessibility label is used to inform the user about the action they should take on that view, and the accessibility hint is used to inform the user what will happen when they select that view. For example, to the naked eye, the only information on the screen is a button with text stating “Help” and the sighted users can infer this would navigate them to the Help screen. However, VIPs cannot infer the same information
and instead rely on the hidden accessibility hint read aloud, stating, “Press this button to navigate to the Help page” (Vinkle, 2021). Another useful characteristic for accessible navigation is assigning an accessibility role and an accessibility value.

3.3.3 Color Contrast Customization

The WPI development team implemented a color palette functionality to prioritize the accessibility of color-blind users. This approach involved thorough research on optimal color combinations, drawing insights from studies linking color and color blindness. In particular, a study by Penn State in 2022 (Penn State, 2022) flagged color pairs such as red and green, black and white, and red and black as problematic. Fortunately, a study found by Ginley and Nelson (Ginley, Nelson, 2013) produced a collection of healthy color palettes, which the WPI team successfully incorporated into their application, as illustrated in Figure 6.

*More rainbow-like palettes, similar to the original Adobe XD button colors in vibrancy

![Color Blind Simulator](image1.png)

![Color Blind Simulator](image2.png)

Figure 7: Accessible Color Palettes

3.3.4 Voice Prompts Improvements

While VIPs can gain value from interacting with the app via touch, voice is another fruitful avenue. Voice prompts focus primarily on providing a seamless experience reliant on voice instead of visuals. It is common for applications to include some type of voice-related interactions within their apps (Podsiadło, Shweta, 2016). To comply with this commonality, the
WAM app allows VIPs to use their voice to prompt essential actions within BV. The overarching idea was to implement text-to-speech (TTS) and speech-to-text (STT) for users. TTS is a helpful tool for transcribing any text on the screen to the user, similar to a screen reader.

On the other hand, STT allows users to speak into an application and render the speech as text. This feature was particularly important given its ability to provide an alternative method for entering information into the app besides typing (Hoffmann, 1970). VIPs gain significant value from STT since they might not be able to interact with the native keyboard. Within the application, STT can also be implemented in reference to navigation since it can allow users to navigate to a page within the app by using their voice.

### 3.3.5 Near-field Communication Research

In the report above, it is mentioned that both touch and voice can be used to alleviate stress when navigating; however, the best solution would be to stop the user from navigating via the app interface entirely. To meet that demand, the MQP development team researched alternative methods for visual and voice navigation. One major consideration was Near-field Communication (NFC) tags. Near-field communication is a low-cost wireless communication centered on providing convenience to daily life activities. NFC tags have already been widely implemented in everyday life, including, but not limited to, medical care and commerce (Cao, et al., 2019). One of the largest uses for NFC tags is associated with cardless payments between a phone and a cashier, as well as paying for public transport in major cities.

The most important facts surrounding NFC tags are that they require no additional power source, are a secure system of communication, and are supported in almost every modern smartphone. The NFC technology can be triggered by a small touch, and the connection ends once the phone is moved away (Coskun, et al., 2015). In terms of the WAM application, NFC
tags provide a unique opportunity for visually impaired users. The goal of NFC tag implementation was to allow users to tap their device against a physical NFC tag at the WAM, triggering some response in the app. The most common use case would be to reduce the amount of time spent navigating the app and instead ease the process by opening the correct page of the app based on the NFC tag tapped. Similar uses of NFC tags have been tried and tested in reference to physical navigation within an indoor location (Ozdenizci, et al., 2015). Some places use NFC tags to help guide indoor residents around a specific building, and this concept is similar. However, instead of only physical navigation, NFC tags can help navigate the virtual world as well. NFC tags, along with deep linking within the app allow for navigation by the tap of a phone, adding dimension to the app besides voice and visuals.

3.4 User Interface/User Experience

One of the interesting problems the team needed a solution for was the application's user interface. Since the app would primarily focus on people with minimal to no visibility, the team determined that the focus would be on functionality over visual aesthetics. Therefore, a more simplistic layout than what is common in other mobile apps was implemented in the final product. The different stages of the app UI development process were the creation of an initial mockup using Figma, implementing a near replica of the mockup flow in the app, and finally, adjusting the UI to work with any additional application features.

3.4.1 Application UI Mockup

Before beginning the project's development phase, the WPI team decided to draft a few mockup designs to conceptualize ideas. The team started by creating an initial plan using Figma, a collaborative web application for interface design (The Power of Figma as a Design Tool, 2022.). Figure 10 shows the developed mockup, which varies from the final product. Essentially,
there would be six main different pages, the most important ones being the homepage and the detail view page. From the homepage, users could select from a large set of buttons with clear text, prompting them to navigate to a new page. The detail page can be navigated by clicking on a specific art piece in the list. Ultimately, the fundamental navigation process in the mockup is present in the final app, although the visuals were altered. The base ideas present in the mockup UI eventually evolved into a full-scale application.

![Mockup Diagram](image.png)

Figure 8: MVP Figma Mockup

### 3.4.2 iOS Accessibility Guidelines

As part of the research process and designing the user interface, the development team analyzed the different accessibility design principles created by Apple, the creators of iOS. These principles were meant to be used as a guide and referenced when developing an accessibility-focused application. The guidelines provide valuable insight into best practices and
methodologies when developing any accessible app. Specifically, Apple has five basic accessibility guidelines (Apple Inc., 2020):

1. **Design with accessibility in mind:** Accessibility is not just about making information available to people with disabilities — it’s about making information available to everyone, regardless of their capabilities or situation.

2. **Simplicity:** Enable familiar, consistent interactions that make complex tasks straightforward to perform.

3. **Perceivability:** Making sure that all content can be perceived whether people are using sight, hearing, or touch.

4. **Support personalization:** Design an app to support the accessibility features people use to personalize how they interact with their devices.

5. **Audit and Test the app for accessibility:** Testing helps you ensure that everyone can complete the most important tasks in an app, no matter how they interact with their devices.

These principles should act as a guide while developing an application, regardless of the platform. During development, the team strived to address each of these very important design aspects. Specifically, each of these principles was implemented within the app, and development process, each in its own unique way.

### 3.4.3 Android Accessibility Guidelines

Similarly, Android has published many documents describing the best accessibility practices for Android devices. However, in a general sense, many of the Android and Apple guidelines overlap with one another. Each of them emphasizes the importance of the same
characteristics but in their unique way. Specifically, Android has five main accessibility guidelines to follow when developing (Android, 2020)

1. **Describe user interface controls:** Provide content descriptions for user interface components that do not have visible text.

2. **Enable focus-based navigation:** Make sure users can navigate screens using hardware-based or software controls (i.e. keyboards and navigation gestures).

3. **Custom view controls:** If you build custom interface controls for an application, implement accessibility interfaces and provide content descriptions.

4. **No audio-only feedback:** Audio feedback must always have a secondary feedback mechanism to support users who are deaf or hard of hearing.

5. **Test:** Test accessibility by navigating an application using directional controls, and using eyes-free navigation with TalkBack enabled.

When developing the WAM team kept each of the Apple and Android guidelines and, and focused primarily on simplicity, scalability, and testing. Each different view layout underwent extensive testing and design iterations based on feedback and best practices.
I V. Implementation

The implementation section of the report outlines the development practices and features that were implemented within the application. It covers a detailed timeline of the development process, the UI design implemented in the final version, any design choices decided by the development team, database implementations, and each feature implemented described in detail.

4.1 Project Timeline and Ticket Management

When beginning the project work, the team developed a project schedule and timeline for the development stages of the project. To keep track of progress and ensure success, an Excel worksheet and Jira were used for each step of the process. The timeline was broken into three 7-week-long terms, which correspond to the weeks in which the student will be actively at university. For each term, the progress was tracked for Report Writing, App Development, and App Testing. Ultimately, the timeline was updated to reflect the realistic accomplishment dates for the project. The figure below shows the final descriptive project timeline.

During the first term of the project, A Term, the majority of the focus was on research, report drafting, and basic app development. During this time the team researched the required information for the project, including development best practices, accessibility features, and more. An important part of the A term stage was understanding the needs of the VIP audience, and understanding the current market standards. Each of the different topics researched, and their respective findings, are included in the methodology section of this report. Additionally, during this term, the very basic MVP version of the app was built. It acted as a building block for the development team to use when the main stage of development begins in B term.

During the second term of the project, B term, the largest action items were related to completing a working prototype of the WAM application. Using the research and MVP built-in
A term, the team continued to implement the projected features. By the end of B term, all of the features of the application were completed, and tested, not accounting for any bugs which occurred in the future. Specifically, the largest features were Backend Integration, NFC Integration, and Accessibility Navigation. User Testing started midway into B term, focusing on getting feedback on the UI and features.

![Figure 9: Project Deliverable Schedule](image-url)
4.2 Implemented Application Features

The main function of BV is to let users PIPs play audio prompts provided by the WAM concerning individual exhibits and artworks. As such, the WPI team implemented a variety of features and changes to features specifically to create ease of use for VIPs.

4.2.1 Audio Player

An essential feature of the application was the implementation of an audio player for users to play audio files provided by the WAM. To accomplish this, a custom component was created to allow for easy customization for VIPs. By building a custom component, as opposed to using the built-in Android and Apple audio player, the team was able to create a view more suitable for VIP users. For example, the audio player view features larger buttons, easier controls, and more contrast than other audio players. The audio player view contains five buttons and a progress “seek bar”, which displays the current progress of the file, along with the time remaining and the current progress. The user can also interact with the seek bar and alter the progress of the audio file. The simplest button implemented in the audio player is the play/pause button, which swaps functionality on selection.

Two buttons are responsible for controlling the speed of the audio, which are centered on the left and right segments of the audio player. As either of the buttons is selected, it will decrement or increment the speed of the video by 0.25. Whenever the values reach the edge case of x2 or x0.25, the speed cannot be incremented or decremented, respectively. Also, whenever the playback speed changes, the user is prompted with audio letting them know the new changed value and the next value. The speed of the audio player is displayed at the bottom center of the audio player in large text. There are another two buttons in the center of the audio player, which are responsible for skipping 10 seconds ahead or behind. All of the buttons in the audio player
support Voice Over and Talk Back support via aria-labels, and the labels are spoken to the user if they have enabled the correct phone settings.

![Audio Player View](image)

Figure 10: Audio Player View

**4.2.2 Color Customization Implementation**

As mentioned earlier in the report, one good-to-have feature for the application is the ability to alter the application’s color palette. Essentially the feature allows the user to switch between a wide range of preset color palettes to make the app easier to view. The colors implemented were chosen based on the research conducted in the methodology Section 3.3.3. Every element of text coloring, background coloring, and button coloring is then updated to reflect the selected palette. This feature can increase the contrast of colors within the app, allowing some users to easily distinguish different elements. Since color-related visual impairments, i.e. color blindness, come in a wide range, allowing the user to select from a wide range of preset palettes is one of the best approaches to counteract the color blindness concerns. Users can select a different color palette from the app and choose which one works best based on their unique color blindness, allowing for a better user experience.

Within the app, a unique user interface was implemented for users to select the different color palettes. On the homepage of the application, there is a Floating Action Button, a button separate from the other elements on the screen in the bottom left of the page, which can be used to open the Color Customization bottom sheet. Within the bottom sheet, the user can choose to
view the previous color palette option, the next color palette option, listen to the currently selected color palette, confirm the color selection, or cancel the color selection. Each different element within the bottom sheet is integrated with Talkback and VoiceOver, meaning these elements will be read out loud when the user has the native accessibility features enabled. When designing the color customization view, a bottom sheet was implemented because this allows the user to see how the different color options can alter the background in real-time. When a new color option has been selected the buttons in the background are visible and display what the app looks like with those colors.

![Image of Color Palettes Selector Sheet]

Figure 11: Color Palettes Selector Sheet

4.2.3 Voice Prompts and Voice Navigation

Another two integral features for the WAM application were the implementation of seamless text-to-speech, as well as speech-to-text. TTS helps the users to play different text
elements via audio, which is valuable if the user is unable to see the text information. While, STT within the app was used to implement a unique method of navigation.

The main locations TTS was implemented were on the descriptive page for each object in the museum, and on the help page. Each of these pages includes a clear button, which, when selected, will play the appropriate text to the user. For example, if the user was on the help page and would like to re-hear the text played to them, they could select the TTS button. Within the details page the TTS button plays the description associated with each art piece. In particular, this TTS button is valuable because it provides an alternative method of listening to the audio file for the art piece if it is not loaded due to internet connectivity issues. Assuming the user is unable to play the pre-recorded audio file, they can rely on the TTS button, which will play the same content as the audio file, but in a less enjoyable fashion.

Figure 12: Text-to-Speech Implementation

The WAM application implemented STT to serve two main purposes, one is to alleviate the stress when navigating the app, and the other is to search for information in a list. In the first case, on the application's homepage, there is a microphone icon on the header that, when clicked, prompts STT. The STT prompt is provided via a bottom sheet, visible in figure ###, and the
sheet includes a button associated with listening via STT. When clicked, the app will begin to listen for voice inputs, and render the spoken information to the user. Once the user is done speaking they will be prompted to navigate to the anticipated page, based on the spoken text.

From this point there are two application flows based on the spoken text. If the spoken information matches one-to-one with one of the pages within the app they will automatically navigate to that page. However, if there is no one-to-one matching, but instead multiple pages which could be a match, the user is guided to an intermediary page which lists all the possible pages matched with the input. For example, if the user inputs “Helmet in the form”, and there are two art pieces with a title including that text, “Helmet in the form one” and “Helmet in the form two”, then the intermediary page will include both options via a list and allow the users to select which is the intended target. Figure ### shows the flow of selecting the proper page from a filtered list of potential pages.

Figure 13: Speech-to-Text Implementation

Figure 14: Intermediary Navigation Page
Within the app, the other implementation for STT is used to search a list by voice input as opposed to typing text. Throughout the application, multiple sections allow the user to input text to help search a list of items. For example, a large list of museum art pieces can be filtered by typing text into the search bar. SST can be used to listen for voice input and then filter the list by the spoken words. This feature implementation is valuable for users which have difficulty using a digital keyboard within the app. Assuming the user cannot use the native keyboard, they can instead rely on STT to search. For example, a user could say “Mona Lisa” and the list of objects present at the WAM would be filtered by that text accordingly.

![Speech-to-text for List Search](image)

Figure 15: Speech-to-text for List Search
4.2.4 Accessibility Navigation Support

People with high levels of visual impairments may require a drastically different navigation experience. TalkBack on Android and VoiceOver on iOS, which will be referred to collectively as “alternative navigation,” provide that different experience.

From the user’s perspective, when altered navigation is enabled the operating system of the device will automatically enact TTS for all content and UI elements that are on screen. Content refers to any audio-free media such as text or images, and UI elements can refer to page headers, buttons, or other pieces of an application implemented required to navigate a mobile app. Additionally, a variety of new swiping and tapping gestures are introduced to allow the user to be able to navigate through UI elements without having to know exactly where elements are physically on the screen. Such gestures include swiping down to have the app read all elements and differentiating between a single tap and a double tap for activating a button.

On the developer side, any content or UI elements with non-text components will require an accessibility label in the codebase. For example, the standard UI for audio playback that was implemented in the app doesn’t use any text. Therefore each UI element requires an accessibility label that describes what that icon can be used for. For example in Figure ### below a standard arrangement for audio playback buttons includes very little text. Each action of the icon buttons can be inferred by the icon itself. However, VIPs cannot visually infer the same information from the icon, so each button would require a TTS accessibility label. The TTS accessibility would include information about what action will be taken assuming the user selects that given button. In Figure ### the accessibility label is represented by red text.
As developers, the main method of testing the UI required two primary steps. The first step was to find the links between each application page, e.g. each page accessible from the landing page. The next step was to try to navigate those pages using altered navigation while being blindfolded to simulate the experience of the VIPs who would be using these features.

### 4.2.5 Deep Linking and Near-field Communication Tags

Deep Linking is the practice of using hyperlinks within a website or app, to improve user engagement and navigation. Deep linking provides additional functionality to applications by allowing developers to simplify their navigation. As mentioned above, NFC tags can be a useful tool for promoting actions on a device, specifically a modern phone, and by combining the usage of deep linking and NFC tags a developer can generate a user experience within the app based on the tag content. Specifically, the NFC tag will include a hyperlink referring to a specific app, and then prompt some action within the app using deep linking. In the WAM app, deep linking was used alongside NFC tags to create an accessible form of navigation. Users could tap their phone against the NFC tag, which encodes a hyperlink, in turn opening the app to the correct screen.
4.2.5.1 Deep Linking Challenges

While there are inherent benefits to basic deep linking, there are also significant constraints that must be addressed. For example, deep linking works great if the app is installed on the device, but what happens if the app is not installed? The correct approach would be to open the app in the app store and prompt the user to download the app; yet how do you do this if the app isn’t installed? The solution is to use an external hosting site that handles the deep hyperlink, and in turn, acts accordingly based on the operating system of the device, and the current state of the app on the phone. In the same scenario above, if the app is not installed the hyperlink will lead to the app store, and if it is installed, the hyperlink will work as intended, opening to some specific page within the app. For this project, the development team opted to use Firebase Dynamic Links to solve the constraints mentioned above. Firebase Dynamic Links is an externally hosted service provided by Google meant to properly handle all scenarios related to deep linking (Qutbuddin, 2022). In most scenarios, it is best to use an external company to handle these issues since it will cost less, and offer well-rounded solutions, rather than implementing them individually. The team chose the Firebase alternative since it is an easy system to integrate, while also offering major benefits at a low price.

4.2.5.2 Deep linking and NFC Tag App Implementations

Within the app there are two main flows associated with NFC tags and the deep linking accommodations. The first method of using deep linking and NFC tags is when the app is closed or not installed. When the NFC tag is scanned the user will be prompted to navigate to a webpage hosted by Firebase Dynamic Links. On this webpage, the site will determine if the app is installed and if not navigate to the app store. In the scenario that the app is installed, but closed, the phone will open the app, and automatically navigate to the correct page. The page is
encoded in the hyperlink stored on the NFC tag. For example, if the user tapped the NFC tag associated with the “Mona Lisa” art piece, the app will specifically open to that page. If the associated encoded page does not exist within the app the user is prompted to let someone at the WAM know, since it is possible the NFC tag was tampered with or is not an official tag.

The second, and more likely flow, is when the app is already installed, opened on their device, and they use the in-app NFC scanner screen. In this case, since the app is already opened, and installed there is no need to navigate to the Firebase Dynamic Link site, so instead, a bottom sheet is used to give directions to the user. The directions will also be read back to the user via the native iOS and Android TTS features. However, some other scenarios are associated with this NFC flow once the NFC tag scanner button in the app is selected. Specifically, the user's device might not support NFC tags or the user's device might not have NFC tags enabled. In either case, the app will notify the user, and prompt them to act accordingly. Unfortunately, if the device does not support NFC tags there is no way for this application flow to work within the app. On the other hand, if NFC tags are supported, but not enabled, the app will describe how to enable these permissions and navigate them to the respective settings screen via a button. Finally, if NFC tags are supported, and enabled, the user will be prompted a bottom sheet, which directs them on how to scan an NFC tag.

4.2.5.3 NFC Tag Scanner Bottom Sheet

When integrating NFC tags within the application as described above it was important to implement a simple view for scanning the tags. Similarly to all other UI aspects of the app, this new NFC view would need to be simple and easy to use. The team opted to implement a bottom sheet view, which would appear whenever the NFC Scanner button was pressed. IOS devices by default come with an already developed NFC scanner bottom sheet. Apple provides this view out
of the box, and it cannot be customized. Figure # shows the iOS NFC tag scanner bottom sheet, which is used in the app for iOS devices. On the other hand, Android devices provide no built-in NFC scanner implementations. This meant that the team needed to build their own frontend UI bottom sheet. Ultimately, this was a major advantage, since implementing a custom view allowed the developers more creative freedom, and the ability to implement a more accessible view. Figure # shows the views for each of the Android flows in the bottom sheet.
4.3 Application User Interface/User Experience

As with any mobile application, the User Interface (UI) plays an integral role in the enjoyment of the user. Typically, this implies creating very complex, and aesthetically pleasing views, however, the WAM mobile application focused on providing VIP users with a valuable experience. To accomplish that goal the WAM app branches off from more common design patterns, and leans into the accessibility guidelines produced by Apple and Android developers. Specific details pertaining to the best guidelines for an accessibility application can be read in methodology section 3.4.

4.3.1 App Design Guidelines In Practice

As mentioned above, when designing the application specific principles were kept in mind. Guidelines created by both Apple and Android were used as a blueprint for success in the accessibility landscape. Specifically, Accessible Design was the main goal for developing the application UI. The team focused on making each visual element, including text, accessible for all users, via sight, touch, or hearing. The next principle, Simplicity played an integral part in the UI design process, with each element being simplified as much as possible. Overall, many of the main pages of the app were redesigned to focus on impaired users, and stray from common non-accessibility focused implementations. Personalization was implemented via the color customization options as well as font resizing, and NFC tag support. This guideline was quite valuable in engaging the user, and making them comfortable using the app. Lastly, Testing was done frequently throughout the design process, including development testing and user testing. Each of the testing groups helped to solve development issues/bugs, solidify good design, and overall provide feedback on the use cases for the application.
4.3.2 App Page Architecture

The interface of the end deliverable application followed some very specific design patterns and choices. The app needed to be easy enough to navigate for everyone, regardless of their degree of visual impairment. For most users, the application's visuals were likely not a concern, but the app's ease of use was crucial. When developing, the team kept this in mind while considering navigation throughout the app, focusing on making it straightforward and intuitive. Each page needed an easy layout to understand, with each component of the views being easily interacted with. In particular, the user should be able to hear the Voice Over or Talk Back prompts and take the appropriate action with ease. Making the views in the app the least bit complex was the route to success when developing since it would reduce clutter and prompts.

One significant element of this was the choices regarding the application buttons. Throughout the application, the team opted to use large buttons with clearly indicated text as the main form of prompting navigation. The design choice for this came from the previous Audio Journal App layout and navigation setup. These large buttons are designed with large contrasting colors so that users with low visibility can distinguish them from the background and other elements on the page. The buttons also feature large text, which also helps those with low vision ascertain what action will be taken when the button is selected. Figure 11 below shows the Audio Journal implemented button navigation system.
The WAM app was structured similarly to the Audio Journal, in the format of a list of buttons. It is valuable to structure the apps similarly since this will reduce the learning curve for VIPs, and increase ease-of-use for users. Following a simple, and similar format will translate the majority of the same implementations across applications, so it was recommended for the WAM app. This allows for a simple user experience where a user only has a few ways to navigate a page. The team tested how voice commands work on both iOS and Android and found that navigation can be tricky and slow. Therefore, the team wanted the list of buttons and locations to be as accessible as possible. For those who cannot use the buttons, the app can use voice commands and built-in voice assistance to control navigation. For example, a user could say “Press Sculpture X” after being read a list of the sculptures, and the phone would navigate them to the correct screen. On top of that, each screen was designed to be almost identical, with the only exception being the detail view page, which let users get familiar with the designs.
4.3.3 Common Application Flows

A user flow is used to describe how a user can interact with the application. The first and most crucial application flow is the ability for the users to locate different artworks via a list displayed to them. Essentially, users should be able to open the app, find an art piece they would like to know more about and play the audio file associated with that artwork. For this flow, there were a total of three different pages possible for the user to navigate to, those being the page for selecting the artwork location, a page for viewing all the relative pieces of art within the WAM matching that location, and lastly, a detail page which showed specifics about the artwork. The detail page also includes the UI for playing audio files, which can be seen in Figure 12. Screen A below shows the basic homepage of the application. From Screen A the user can navigate to Screen B by selecting the “Exhibits” button. Screen B displays a secondary filter for the Art Pieces which allows you to filter art pieces via art piece location. From there the user can open Screen C, which displays the different art pieces which follow that designated location. Finally, the user can navigate to Screen D, which is a detail view of the art piece, by selecting one of the art pieces from the presented list of items. One change made in relation to this flow, was the renaming of the “Exhibits” button to “Art Pieces” to reflect the naming scheme at the WAM.
Figure 20: Application Flow Screens, Screen A, B, C and D
The second flow for the WAM application is associated with users being able to find and view relative information about the app itself, the museum, and other miscellaneous actions. This flow pertains to any page not included in the main flow above, for example, the Help Page, or the NFC scan page. The second flow is generally less complex than the first since it involves less user interaction. The main home screen allows users to initiate this flow by selecting the page button options. Next, the app would navigate to that screen and show the relative information, users can also opt to navigate back to the previous page. For example, a user would first see a set of options to select from, such as Information or Help, and selecting an option would navigate them to the information or help page.

Figure 21: Help Page View and Detail View
4.4 Backend Implementation

An integral component of any application with dynamic data is the backend. The backend can help store, retrieve and modify data in a nonstatic, safe and productive manner. In this sense, non-static refers to the ability to change or add new data to the data set since static data cannot be altered easily. There were two main components to storing data for the WAM app: the visual text information for the art pieces and the audio files played by the users. The text information is relatively small and easy to store, while the audio files are much larger. To overcome the storage challenges, MongoDB and Firebase Storage were implemented within the app to retrieve data.

4.4.1 Outsourced Database Hosting

While the main features of storing data in a database might seem a solely virtual affair, there is a physical component associated with it. The data is stored on a physical computer via some internal storage system; storing, maintaining, and securing such a system can be costly and unsafe. To counteract this, companies such as Google and Facebook have created alternative hosting services to store the data safely and securely on their system. The downside of this approach is that it removes the security responsibility from the developers to whatever platform is chosen. However, for the WAM app, none of the data is sensitive information, and outsourced database hosting is a valuable and reliable solution.

4.4.2 The WAM App Backend

For the application backend, the development team implemented two cloud-hosted database platforms, MongoDB and Firebase Storage. This decision was made after considering the costs of maintaining and producing a custom database hosted by the WAM. Ultimately, it is more effective to host the service externally since this allows the team to focus on developing without spending time setting up the server and additional requirements. Unfortunately, adding
an external backend component creates a reliance on the internet within the app since network requests are required. However, this is standard for most modern mobile applications, and the WAM provides public internet access for users, in turn mitigating this issue.

Within the app, there are two main views in which asynchronous actions take place. The first and main location is when the app is first opened. Each time the app is opened, the data is fetched for the first time for each user. This gives the app a baseline for the information displayed to the users. The information fetched on the first load is all the art piece details stored in the MongoDB server. This data is then segmented into smaller lists by either type, location, or date created and as the user navigates the app. The art piece data is also displayed on the detail pages for each item, including the name, description, creator, and more.

The second main location within the app where asynchronous database connections are implemented is on the details page. While the text information displayed on the details page is from the initial backend fetch when the app is first opened, the audio files are only loaded when the users view a specific art piece. Audio files for the WAM app are stored within a Firebase Storage server hosted by Google, and when the user opens the detail page, this audio is fetched from the server. To accomplish this, a public link is generated to download and play the file, which is then loaded and integrated into the app. Due to the large nature of audio files, loading this data can take time, and while the file is loading, a text message and a loading spinner are displayed. Once the file is loaded, it can be interacted with via the UI, as shown in Section 3.4.

V. User Study and Results

During the testing phase of the project two user studies were conducted to test the implementation of the WAM Android and Apple apps. The testers were recruited both in part
with Audio Journal and with the Worcester Art Museum. Overall the development team tested with 20 different testers and collected their feedback. The WPI team eventually used the study feedback to suggest and implement additional changes within the app. This feedback was critical to the development process since none of the development team members were VIP.

5.1 User Study Procedure

When undergoing testing with participants the WPI development team followed a strict procedure for collecting samples and feedback. To begin testing, the team completed the required Institutional Review Board (IRB) forms, which are included in Appendix A, B, and C. The IRB consent form was required to be signed by all participants as the document outlines the goals and risk factors associated with the study. The team also generated a detailed Questionnaire, visible in Appendix B, to ask participants which the IRB approved. The questionnaire included both numerical responses on a rating-scale as well as open-ended responses. The team implemented both Apple Test Flight and generated an Android APK file to conduct the user testing. Apple Test Flight is a tool for apple developers to test applications before releasing them on the Apple App store. The team relied on their designated devices for testing, either iOS or Android, and downloaded the app respectively. On the designated testing days, the phones were provided to testers, depending on if they were familiar with an iOS or Android device. In the typical testing environment, phones were provided to testers to reduce the time spent downloading the app on every personal device. However, the team assisted if some testers preferred to download the application on their device.

During the testing phase, the team conducted two main sessions at two locations to reach a wide range of people, the WPI campus and the WAM. Audio Journal helped to recruit visually impaired testers for the WPI session. The second testing session was located at the WAM; to
recruit testers the WPI team approached museum visitors and asked them if they would like to engage in the study. The majority of the testers at the WAM were non-Visually impaired. At the end of each testing session, the team distributed a questionnaire via Google Forms to collect feedback which is in Appendix B. If the testers could not complete the survey individually a team member was assigned to verbally read questions and record the tester’s responses.

5.2 Testing Results

In general features were rated favorably by their target demographics. When users were asked to rank their app experience from one to five (one being “unintuitive” and five being ‘very intuitive’), more than 85% of users reported the app as being ‘very intuitive.’ This gave the WPI team confidence that the accessibility features included throughout the app were more than sufficient to support users of varying accessibility needs.

5.2.1 Visual and Print Impaired Tester Results

Results for VIP and PIP testers were largely positive. The average rating for the overall app’s intuitiveness was 4.6 out of 5. However there are some caveats; the small sample size of VIP/PIP testers was small with only seven VIP/PIPs in the testing pool and only one PIP. The intuitiveness ratings are also reliant on having a person experienced with the app to assist users with learning the user interface.
App features such as TalkBack/VoiceOver and Voice Control were received positively with high mean ratings of 4.5 and 5 respectively. Among the testing samples, the majority of visually impaired users used TalkBack and/or VoiceOver and were familiar with accessibility technologies from their day-to-day lives. Menu navigation was also seen as intuitive with a mean score of 4.5. The three features with the lowest results were the audio player, voice search, and color palettes with mean scores of 3.6, 3, and 2.33 respectively.

Qualitative feedback from testers indicated that the buttons for the audio player were in a confusing order for VIPs. For a sighted user, symmetrical placement of buttons with the most important buttons in the center is intuitive, but for a VIP it’s most intuitive to put buttons in the order that native accessibility features would read labels. VIPs also generally do not use the playback bar due to not being able to see where the slider is located. The app received updates to reflect the qualitative feedback.
Old:

![Old Audio Player](image1)

New:

![New Audio Player](image2)

Figure 23: Comparison of old and new audio players

VoiceOver and TalkBack read UI components from top to bottom and left to right. The playback bar’s accessibility label also reads off the current timestamp of the audio played, making the audio player’s accessibility label take longer to read than other buttons. Therefore the navigation buttons were placed above the audio player. The play button was placed in the top left corner because it is the most important button for operating the audio player.

The voice search button was moved to the bottom of the home page for similar reasons. With the new position, it is easier for the screen reader to interpret the item on the screen. The voice search button was rated as moderately unintuitive with a mean rating of 3.0 and is not explicitly necessary for navigation. Therefore the voice search button was seen as less important and put in the bottom right corner of the page to be read last.
While the feedback form did not have a space for the following point of feedback, users did verbally communicate that the accessibility labels for certain buttons were too long. For example instead of “color palette” the label reads “this button lets you change the color palette.” Long accessibility labels significantly slow down the user experience because the accessibility label is read out verbally using text-to-speech with TalkBack and VoiceOver. Accessibility labels were shortened across the app to the minimum amount of text that would be displayed.

Another goal of testing for the application was the NFC tags. NFC tags had favorable qualitative responses as can be seen below. However, the user testing session for VIP visitors was at the WPI campus in a large classroom and NFC tags were only featured in a small demo in a setting void of a museum exhibit context. One major conclusion drawn from the testing is that many users, including older testers, were unfamiliar with NFC tags. There are a variety of non-technical implementation challenges regarding NFC tags and QR codes such as placements and help for older visitors that will need to be adjusted during the app’s implementation at the WPI campus.
WAM. Moving forward the WPI development team worked closely with the WAM staff to plan the placement of NFC tags and QR codes.

<table>
<thead>
<tr>
<th>Did you use the NFC tags to navigate the application?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFC tags used and very helpful</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes useful</td>
</tr>
<tr>
<td>Yes, they were useful</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes very useful, but need more nfc examples to get a better sense of what the museum will actually be like.</td>
</tr>
</tbody>
</table>

Figure 25: Qualitative Responses For NFC Tags

5.2.2 Non-Visually Impaired Testing Results

The “Voice Search” feature, which received a score of 3.56, was considered the least intuitive feature by non-VIP testers. This may be because many users didn’t see the need for this feature since the UI was simple enough for them to navigate through the app easily. Another point of feedback was that in noisier environments the voice recognition element of voice search was less reliable. To help mitigate these concerns we clarified the UI for editing voice search.

On the opposite side of the spectrum, the color palette feature received a perfect score of 5. This was most likely due to the fact that the change in scenery and color was likable by the non-VIP testing party. Generally, the menu followed the same pattern as other features within apps so it remained intuitive for non-VIP testers.
Next, “Voice Control” received a score of 4.10 out of 5, and “Text-To-Speech” received a 4.22. The Text-To-Speech feature was not used as much by the non-VIP testers due to their lower accessibility needs. Similar to the Voice Search feature, Voice Control had issues interpreting the text spoken by the user; however, these are not concerns the WPI team can mitigate since voice recording is handled via the native phone software. Non-VIP users were also less experienced with using these features which would have influenced their results negatively. The “Audio Player” setup received a stellar score of 4.60 and was considered a simple feature for non-VIP users. However, it was important to note that following the testing, the layout of the Audio Player was adjusted to reflect the feedback from the VIP tester results. Still, the WPI team does not anticipate any significant changes in feedback scores for the UI from non-VIP users. Lastly, “Menu Navigation”, which landed 4.30 out of 5, shows that the navigation layout for both VIP and non-VIP testers was simple and intuitive. When speaking to these testers, they mentioned the UI was similar to other apps they used and was highly simplistic.

All non-VIPs unanimously stated that they would use the app in the future. The few that did try to give us additional advice thought of adding QR codes to the app. The average score of all features was an esteemed 4.34, while the average score for the entire app was an even better 4.60. In conclusion, the app was generally rated favorably among sighted users.
5.2.3 Overall Testing Results

While testing the app with both VIPs and non-VIPs, the WPI team found that certain parts of the application remained somewhat unintuitive. For instance, the term “NFC” (Near Field Communication) caused some confusion among a variety of testers. This is due to the term being new to them and a more technical way of describing the ability of two devices to communicate with each other when close to each other. Apple Pay is the most common and widespread technology for contactless payments that utilizes NFC tags. To explain to testers what NFC tags were, the team explained how NFC tags act as the true magic behind Apple Pay and other contactless forms of payment. For the majority of testers that were familiar with Apple Pay, this explanation was productive in explaining the concept.

During the initial planning process the WPI team opted to choose React Native to build the WAM app instead of Swift and Java, the native languages for iOS and Android development respectively. Since Android and iOS share the market for mobile devices, React Native was
selected as an alternative for building one app on both systems. The result of this decision was that the WPI team was able to create, build, and test the app more thoroughly across both iOS and Android devices. As part of the questionnaire, the WPI team wanted to ascertain if this development decision to use React Native was worth the effort. The results showed that 58% of testers used an iOS device and 35% of testers used an Android device, and 5% used both device types. The team believes that, given the results, the decision to use React Native was optimal given the ability to reach a wider range of guests at the WAM.

![In your day-to-day life do you use an iOS device, Android device or Both?](image)

Figure 27: Android vs iOS Testing Demographic

One change that stemmed from the testing results is related to the accessibility of the color palette button since it was only accessible on the homepage of the app. Some testers also confused the graphic we had on the help page for a usable button. The team decided to increase the accessibility of the button by turning the already present visual icon into a functional button for changing the color palette.
Two more features influenced by feedback were the Voice Search and Voice Control functions. These features received the most negative feedback and received the lowest overall rating of 3.5 out of 5 in terms of intuitiveness for users across all demographics. As discussed briefly in Section 5.2.2, it seemed as though users had trouble knowing when the microphone button was on and listening versus when it was inputting text. The WPI team saw this as an opportunity to redesign the button. This redesign made it easier for users to realize when the button, shaped like a microphone, is listening and when it is inputting text. The team also opted to include the edit box within the view so the user could decide to edit the text.
Additionally, to continue attracting ample feedback from users, the team integrated a support page found within the help tab. This made it easier for users to find and report any bugs/problems found while interacting with the app. Users could send an email via the app to the development team, which can address the feedback over time.
Lastly, another difficulty testers faced was reading the text due to the font size, specifically on the art piece pages. Given that the art piece text descriptions were a centerpiece of the app we added additional features to increase readability. The team increased font sizes throughout the entirety of the app as well as added integration for a device’s native font-scaling settings. As shown in the figure below, font scaling reflects the appropriate size font as preset on the user’s device. The app will scale all text visible on the screen based on the enabled font scaling settings within the device's settings. Figure X below shows the two different homepage views, one with font scaling set to detail, and the other to the max, resulting in larger text.

![Figure 31: Font Scaling Screenshot Comparison](image)

**VI. Future Work**

After working on the project the team developed a list of recommendations for both the WAM and future WPI development teams. The recommendations cover technical ideas, social
concepts, project planning, and future app ideas. The team hopes that readers and developers alike can learn from the recommendations and build a better application.

6.1 Additional Future Work/Features

The end status of the project is a working mobile app for the WAM focusing on accessibility. The app includes multiple accessibility features mentioned in the Methodology section of the report, however, there are many other features the team recommends become developed. Unfortunately, the team was limited by the timeline for the project, and could not research and implement all proposed features during drafting. Below are suggested future works:

1. Research costs and time constraints associated with building an app for non-VIP users at the WAM. When testing a large number of users were not visually impaired, and many indicated that they would enjoy using the app when they visit the WAM themselves. However, there is a disconnect between a UI focusing on VIP users and non-VIP users. This means that the UI and features may vary differently in an app focusing on non-VIP users, leading to an uncomfortable experience for VIP users.

2. The second recommendation is to implement Firebase Analytics within the application. Firebase Analytics is a tool that is useful for tracking key statistics in the app, for example, the number of crashes, the number of active users, and more. This tool could be useful to perform a more data-centered analysis for the app. Furthermore, since the WAM app already integrates Firebase it should be a simple integration within the current app.

3. Currently, the app text and the WAM voice recordings are all in English. One potential feature could be to implement multilingual functionality within the app. This feature would require recording additional audio files, and support different text languages.
4. Lastly, one future feature for the project is to build a simple UI, independent from the app, for members of the WAM to input and update app art pieces data. After speaking in depth with the WAM team it became clear that data within the museum changes frequently, with art pieces being added, moved, or removed from display. In the current setup, it might be challenging for the WAM to maintain an accurate record of art pieces. Therefore, it might be fruitful to implement a simple UI, i.e. website or a different interface, for the WAM to view, add and update art piece records.

6.2 Recommendations for Future Teams

After the WPI team’s development and testing period, the team developed a series of recommendations that could aid future teams working on similar projects. These recommendations have been informed by the development process of the WAM app, and aim to provide guidance for future teams working on similar projects.

The WPI development team’s first suggestion is to thoroughly research how disabilities affect an app’s demographic before starting development. It is crucial to conduct thorough research on accessibility and how it impacts the target demographic before beginning development on the app. In this case, the WPI team used the research done by the previous MQP teams, in addition to individual research found to get an understanding of what accessibility development tools exist and what could be leveraged for development.

In addition to the previous recommendation, it is important to test the app for usability in terms of VIPs and non-VIPS during the development process. One effective method for doing this is to enable accessibility settings on the testing device and test the app's features while simulating visual impairment, such as by using blindfolds. This type of testing can provide insights on how to improve the app's functionality and ensure that the app is simple to navigate.
The third recommendation revolves around code maintenance. Maintaining clean and organized code is an essential aspect of the development process. To achieve this, it is recommended to establish a system for managing tasks and code early on, such as using Agile development. Simple measures such as using consistent naming conventions, providing clear comments, and using descriptive variable names can greatly improve the readability and maintainability of the code. This not only benefits the current development team but also facilitates understanding and referencing for any future teams that may need to work with the code. Additionally, regular code reviews can also aid in keeping the code clean and organized, as this can ensure that the code adheres to the established coding standards and conventions. Furthermore, utilizing version control systems such as Git can also help to keep the code organized and make it easier to track changes and revert to previous versions if needed.

When it comes to testing an app, the team recommends getting a stable and working version as soon as possible. This will allow you to recruit more users earlier on and faster. User input is a crucial part of the development process, and having users test an app can help find bugs that you might have not been aware of. Ideally, it would be beneficial to recruit as many visually impaired users to test the app and provide feedback on the accessibility and usability of the different features. This will help ensure that the app meets the needs of the target audience. Overall, incorporating user testing early on and continuously throughout the development process will greatly improve the end result and enhance the user experience for visually impaired individuals visiting the art museum.
VII. Conclusions

Overall, the team felt that the project was a success. The app was submitted for approval to the Apple App Store and Google Play store on February 1st and was approved for release by February 2nd. The QR codes to the respective Android and iOS play story can be seen in figure 32, scan these options to download the app. The development and debugging, along with the user testing processes went smoothly. The team integrated Apple TestFlight and Android APK files for testing with different users, which made the experience easy and simple. The feedback provided via the user study gave valuable insights and edits into the app, which ultimately produced a better end product. The WPI team hopes that Beyond the Visual will inspire future endeavors in the intersection between technology and accessibility.

Figure 32: QR Codes for App Play Store
VIII. Extended Work

As mentioned in the Authorship section of this report Michael Zeolla and Marko Vila continued developing and working on the project following the conclusion of C Term 2023 (March 1st). The work conducted focused on advancing the WAM app by adding more features, finishing the development process at the WAM, and generalizing the application for different usages, i.e. other company implementations.

8.1 Brief Project Recap and Final Steps

At the conclusion of the project in C Term 2023, the application was released on both the Android and Apple app stores, and accessible for all visually impaired visitors at the WAM. However, while the application was released there were still internal processes that needed to be completed. Most critical was the installation of the NFC Tags at the WAM across multiple exhibits and art pieces, as well as the creation of media audio files for the different art pieces.

To help alleviate the process for the WAM, the WPI development team, consisting of Michael Zeolla and Marko Vila, recorded a series of informational videos and explanations describing the different processes for maintaining the application in the future. The most impactful steps in maintaining the application were actively updating the WAM MongoDB database, adding audio files to the Firebase Storage media repository, and properly installing NFC Tags at the WAM. At the same time as the WPI team was working on this step of the project, the WAM internally procured NFC Tag displays for public access, which contained the NFC Tags in a safe and secure casing, as seen in Figure 33. Once the WAM ordered and received all the NFC Tag displays the WPI team helped encode the NFC tags with the proper app Deep Links. Moving forward, due to the efforts of the WPI development team, the WAM had all the necessary information at their disposal to maintain the application.
8.2 Generalizing Accessible Applications

After developing the WAM application, the team realized that it would be more useful for the future to create a well-researched and developed design and implementation pattern which could be generalized to work for multiple different application use cases relating to VIPs. There are three major reasons why generalization in the accessibility space would be valuable:

1. Familiarity and Consistency: As mentioned previously, simplicity can be critical to accessibility applications, since it can reduce clutter and difficulty navigating the app. More complex, and common designs, tend to add difficulty for users to the views of the app. This is especially true for VIPs, since they cannot infer the same visual context many mobile applications rely on for everyday users.

2. Repetition: One important factor for implementing similar accessible designs across multiple applications is that it introduces repetition. Repetition references the fact that if a
visually impaired user is familiar with the accessibility layout in one accessible app, then there will be less confusion when using other, similar, accessibility apps. For example, the Audio Journal app, mentioned in the background section of this report, and the WAM app are similar to one another in reference to the different screen layouts and features, and this was implemented to increase repetition. Users that use the AJO app are likely the same users that will use the WAM app, since they are both VIPs, and both organizations exist in the same local region, Worcester, MA.

3. Reusability: Lastly, generalization introduces reusability to the project, and this directly benefits developers. The more generic, and reusable the codebase, the more undemanding the development process, since assets and principles can be implemented easily.

8.3 Applied Generalized Process and Ideas

During the later months of this project (February 2023) another local Worcester, MA organization, the EcoTarium, submitted a request for building an accessibility app to Professor Neamtu and the WPI development team. The EcoTarium hoped to develop an accessibility application for their visually impaired guests at the EcoTarium. Professor Neamtu and the WPI team saw this as an opportunity to apply their generalized ideas and development structure.

8.3.1 EcoTarium Background Information

The EcoTarium is a museum which offers a variety of exhibits and programs focused on science and nature, including a planetarium, a wildlife sanctuary, and interactive exhibits on topics such as energy, weather, and ecosystems. The EcoTarium also hosts a number of special events and educational programs throughout the year. The EcoTarium exhibits and programs are designed to inspire curiosity and a love of learning, while also raising awareness about environmental issues and encouraging visitors to take action.
When it comes to development of the app, the EcoTarium, as an organization, is similar to the WAM in the sense that they have exhibits which display different information pieces about a topic. However, the EcoTarium shows differences in the fact that each exhibit has a different focus and information, whereas at the WAM the mode of delivery and information displayed is always the art pieces. The EcoTarium’s displays have a wider variety of intractability, which posed an initial issue when considering how to best approach the app.

8.3.2 EcoTarium Early Development Process

To begin the development process of the new EcoTarium application the WPI development team first met with everyone at the EcoTarium. The team discussed expectations, goals, potential conflicts, and features desired from the application. The consensus was that the WAM app would be a great starting point for the new app. Following this, the WPI team began to map out the next steps and the overall plan for the project. Two vital limitations placed on the project were that the WPI team must focus on finishing the WAM application first and foremost, and secondly, that the WPI team would not release the EcoTarium app during this MQP cycle. Instead, the work completed by this WPI MQP team could act as a guide and reference point for a future MQP team, in 2023-2024, which could continue and finish the project in its totality.

In contrast to the first steps in the original WAM project, the initial development time for the EcoTarium was focused on modifying the WAM codebase to reflect the EcoTarium. This significantly increased productivity and decreased time spent on writing code, due to the original WAM app being used as a generic model. Much of the time spent researching design patterns and best practices was circumvented since the design for the WAM app was already tried and tested. Instead, this time was used to better flesh out new ideas and features, such as live map tracking, which would otherwise be improbable given the typical MQP timeline.
Following the modification of the basic generic frontend application from the WAM app, the next step was to integrate real information into the EcoTarium application. Each application requires a different database architecture depending on the information being stored. Therefore, within the confines of the EcoTarium app, the WPI development team created a MongoDB and Firebase backend for the application to begin. Since the information stored in the database may differ from exhibit to exhibit, a NoSQL database, such as MongoDB offers large-scale flexibility.

For the initial implementation of the EcoTarium application, one basic exhibit was created, and the information for this exhibit was extrapolated from an EcoTarium pamphlet. This exhibit was the “Vibram All Persons Trail” at the EcoTarium. The idea was to create a basic application, from which more exhibits could be added later on. In future iterations of the project, more exhibits could be added via the MongoDB and Firebase backend. Figure 34, shows an example section from the pamphlet, from which the exhibit information was taken.

Figure 34: EcoTarim All Persons Trail Example

### 8.3.3 Feature Customization and Implementation

The underlying user interface for the EcoTarium mirrors the WAM application almost one-to-one, as seen in Figure 35 below. Many of the screens are identical in functionality, such as the home page, help page, and list views. Due to the functionality being identical to the previous
WAM app, the WPI team decided to keep the views the same, since this would allow for redundancy and simplicity. Each of the views implements the same TalkBack and VoiceControl features as mentioned in the WAM app, to help alleviate navigation issues. Also, the more specific views, like the information help page, were updated to reflect the EcoTarium's details.

Figure 35: EcoTarium Application Screens
For the more unique and customizable exhibit screens, the WPI team relied on the EcoTarium to provide input into the design. Therefore, the EcoTarium provided a basic mockup for the different exhibit views. The mockup, as seen in Figure 36 allows the user to first select their exhibit, and then choose to view different informational pieces, such as different plants, animals, and more. The WPI development team took the mockup and chose to modify the designs to better support a visually impaired user. The largest change in the proposed UI, versus the implemented UI, was the movement of the arrow buttons from the bottom to the top of the view. The arrow buttons are responsible for allowing the user to swap between different “stops” within the exhibit and as such control a major part of the application flow. Moving the buttons to the top, as seen in Figure 36, allows users with TalkBack and VoiceOver to easily navigate the “stops”, without needing to swipe to the bottom of the view.

Figure 36: Original EcoTarim UI

As mentioned above, the EcoTarium app also relied on an information system that could provide text and audio information to the end user. Initially, MongoDB and Firebase were used, respectively, the same as the WAM app. Altering the backend for the new EcoTarium app was as simple as swapping out credentials since the WAM implementation could be replicated with
ease. Furthermore, additional features for the database, such as aggregation and filtering, could be easily added within the application on an app-by-app basis, depending on the requirements.

During the development process, a member of the EcoTarium proposed swapping the audio file hosting from Firebase to DreamHost. DreamHost is a simple web hosting platform, which offers free tiers for non-profit organizations, such as the EcoTarium. Due to the modularity of the database setup, it was a simple change within the application to swap the backend hosting platform. This database swap shows the power of using generic well-written code, which introduces modularity. Overall, it was simple to add support for another database type within the code, since the code for accessing the databases within the UI was abstracted. In the future, if a sponsor desires to implement other backend services, like Back4App or AWS, the change can be implemented easily and without significant hassle. However, in the end, Firebase was chosen to host the app audio files over DreamHost, due to data limitations, delays, and other factors.

Another custom feature that was introduced to the EcoTarium application was the concept of seasonal context. Within the Ecotarium, many exhibits and informational pieces depend on the season, for example, Winter versus Summer. Therefore, one idea proposed by the EcoTarium was the functionality to swap the content of the app based on the season, as determined by the app. To accomplish this, the data structure in the database was modified to include sub-contextual informational pieces. For example, the Vibram All Persons Trail has alternative text fields based on the plants that are in season at the time. At runtime, the app determines the season based on the date (month) of the phone, and other contextual information then fetches and displays the correct text from the database. This feature was a simple addition to the application but exemplifies the concept of customization on a need-by-need basis.
8.3.4 Live Map User Assistance

One of the major features desired for the EcoTarium and the Worcester Art Museum applications was the addition of a Live Map service for guiding users around the grounds. This feature was deemed out of the scope of the initial WAM app and was moved to be an idea for next year's team. Due to the reduced project timeline (only consisting of D term), the team and the EcoTarium agreed that this would be a future improvement for the app, completed by another WPI MQP team. However, the current MQP team conducted research as to the best practices, and concepts behind in-door user mapping, to help guide both future apps.

A common approach to tracking a user in real-time relies on Geolocating the device of the user, for example, a mobile device, using the Global Positioning System (GPS). This method relies on the use of satellites which ping transmissions to the device, and depending on the delay in receiving back a signal the device position can be approximated (Maddison and Cliona, 2009). Using GPS the location, speed, and direction can be contextually inferred. GPS works wonders for applications that rely on outside movements, like Google Maps, and Geofencing but is not reliable for indoor usage. The largest blocker to GPS for in-door spaces is the limited satellite signaling and responses. Essentially, when the device is inside, the satellites that GPS relies on cannot ping the device via a signal, due to signal losses when colliding with building structures and thus cannot approximate the user's location (Simões et al., 2020). While there have been minor improvements in this field, it is still not a reliable and robust system for adoption.

To overcome this challenge in-door tracking requires alternative methods of positioning, rather than relying on the traditional GPS. The indoor positioning systems can be broken into three main categories (1) self-positioning, (2) infrastructure positioning, and (3) assisted by self-directed infrastructure (Simões et al., 2020). Each of these types of in-door positions
provides a similar, but slightly different, approach to solving the issues. Of the three options mentioned above, the most applicable to this project is an option (2). Infrastructure positioning is the process of using externally set-up devices in the environment to approximate the position of the user's device. For example, using Bluetooth devices or Wifi connectors in the environment to help identify the current position of the user based on the signal strength from these devices. Other implementations would require more internal setup from the sponsor, which might not be possible. For example, another approach for in-door tracking involves computer vision for identifying and alerting guidance systems, i.e. the user's phone (Simões et al., 2020).

![Diagram of signal-based in-door tracking](image)

**Figure 37: Signal Based In-door Tracking (Simões et al., 2020)**

Positioning systems that follow the infrastructure approaches often rely on signal range and strength as the prediction measures. The signal propagation time between the transmitter and the receiver, and the intensity of the received signal are all factors that lead to proper identification of position (Correa et al., 2014). Figure 37 above shows a sample setup for tracking users by signal strength. Essentially, this method relies on the trilateration method between three nodes and the distance associated with them. The center of the distances is the anchor point, which can be used to approximate the user's position (Correa et al., 2014).
Time-based positional approaches, such as the one mentioned above, can result in accurate position estimates, making them a proper solution to the current problem at hand. However, further research is required in referring to sponsor willingness and associated costs.

In reference to the WAM and EcoTarium applications, in-door tracking using option (2), infrastructure positioning, would involve the small setup of internal environmental signals around the location, which are used to position the user's device. Within the app, it would be possible to then overlay a custom map depending on the location, integrated with Google Maps. Then, as the device's position was approximated using the infrastructure approach, update the position on the map in real-time. Furthermore, many React Native wrapper libraries could be used to integrate with the Google Maps API, such as React Native Maps. Once again, the benefit of using React Native in this scenario is that it allows for dual development, such that the implementation of one operating system works on others (Luschi, Alessio et al. 2022).

The recommended approach for implementing the map component within the app is to use the Google Maps SDK. The Google Maps SDK is an external tool for connecting and implementing Google Maps services, which can be used to display a custom Google Map, similar to the Google Maps app. In particular, Google provides a free open platform to connect to their map servers and track the user's location, which can be easily paired with React Native via their web interface. By pairing the live map tracking described above, with the custom Google Map component in the app, the user can track their position in real-time. In addition, this approach would require the WAM or EcoTarium to create a highly-accurate map of their location grounds. Within the confines of an accessibility app, the best method for adding the map component would be to add a Map button to the home page; which would allow users to navigate from the home screen to the map view within the app in a simple and accessible manner.
One stretch goal for the live-map user tracking would be to offer directions to the user based on their current position. For example, within the app UI, the user could specify a specific Art Piece or Gallery they would like to visit, and the app could guide them to that approximate location. A unique approach to user guidance was recently implemented for a medical facility, which relied on the usage of a custom Google Map, depicting the hospital layout, and an infrastructure approach using environmental guides (Luschi, Alessio et al. 2022). One concern for this implementation is that it would require the WAM and EcoTarium to keep a running directional database of potential paths. For example, at the WAM there are many times when certain hallways and rooms are closed to visitors, and the guidance system must account for this.

Based on the research conducted it can be concluded that a Live User assistive technology is possible for both the WAM and EcoTarium. However, it is important to note that it would require support from the sponsor in terms of purchasing and setting up infrastructure-based devices for guidance, as well as app UI alterations to include the map. Moving forward the team recommends that the next WPI development team continues the research conducted, researching more specifics on individual navigation types, as well as open a dialog with the sponsor about the requirements for such an undertaking.

8.4 Extended Work Conclusion

At the end of the project scope, the team felt that the extended project work was a success for both the WAM and EcoTarium applications. As mentioned in Section 7, the WAM app was submitted for approval to the Apple App Store and Google Play Store on February 1st and was approved for release by February 2nd. While the EcoTarium application was not publicly released to visitors, the work conducted in prototyping, researching, and structuring the project assisted in getting the project started. Future WPI MQP teams for the 2023-2024 season can use
the EcoTarium prototype and research as a beginning for the complete release. In conclusion, the work completed during the extended section of this project proves that generalization is possible for accessibility apps. The generic UI implemented for the WAM app was easily replicated for the EcoTarium, and can even be applied to other institutions. Reducing the burden on early prototyping and development by using a proven generalized approach can help to reduce the development time, and allow developers to implement more complex app-specific features.
Resources


https://ibvi.org/blog/blind-vs-visually-impaired-whats-the-difference/


https://confidentials.com/manchester/revealed-manchester's-most-popular-tourist-attractions

https://www.diariocritico.com/supercor-el-corte-ingles-accesible-beepcons


“Firebase Dynamic Links | Firebase Documentation.” Google, Google, 2016,


[link.springer.com/chapter/10.1007/978-1-84628-867-8_14](link.springer.com/chapter/10.1007/978-1-84628-867-8_14)


https://www.nytimes.com/2014/03/20/arts/artsspecial/what-comes-next-after-the-troops-are-dismissed.html


https://www.esmadrid.com/en/tourist-information/museo-tiflologico-de-la-once

Mathieu. (2021). *Study on the accessibility of museums to visually impaired people*.  
https://blog.evelity.com/en/study-accessibility-museums

https://audiojournal.org/about/mission/


https://accessibility.psu.edu/color/colorvisiondetails/

Perkins School for the Blind. “Perkins Milestones,” perkins.org,  
https://www.perkins.org/our-history/perkins-milestones/


medium.com/firebase-developers/firebase-dynamic-links-guide-76552c0f292e.


https://worcester.ma/2018/02/free-to-read-audio-journal-helps-visually-impaired-see-the-world/

https://www.shopify.com/partners/blog/react-native-accessibility

Retrieved September 26, 2022, from

https://www.afb.org/about-afb/history/online-museums/life-and-legacy-louis-braille


https://www.worcesterart.org/information/
Appendix A: IRB Consent Form

Informed Consent Agreement for Participation in Worcester Art Museum Accessibility App Study

Student Investigators: Michael Zeolla, Zane Carey, Marko Vila, Daniel Rabinovitz, Vrandol Perez

WPI Faculty Advisors: Rodica Neamtu, Email: rneamtu@wpi.edu, Lane T. Harrison, Email: ltharrison@wpi.edu

Student Investigators Alias: gr-wpi-wam-mqp@wpi.edu

Title of Research Study: Worcester Art Museum App User Study

Sponsor: Worcester Art Museum

Introduction:
You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the Study:
The Worcester Art Museum is working with WPI to develop a mobile app to make the museum more accessible for visually impaired visitors. We are the student team responsible for developing and testing the application on Apple iOS and Android devices. Our overall goal is for users to navigate the app effectively, and that all the features work as intended. Visually impaired people are the focus of this study, but we would also like people with no or minimal visual impairments to participate in the study. We will be writing a report and analysis of our findings.

Procedures to be followed:
- We will instruct you (in person or by email) on how to open the test version of the app (bundle).
- You will be asked to person some tasks regarding navigating the app.
- If you participate asynchronously, you will be provided a week to do the testing.
- At the end, you will be asked to answer a set of questions.
- Your identity and responses to individual questions will be kept confidential.
- We will provide iOS devices to use in-person, and those with Android devices will be instructed on how to open the test version of the app.
- You will be asked to perform some tasks regarding navigating the app while navigating a simulated museum exhibit.
- You will be asked to state out loud your thought process for navigating the app, as well as opinions on varying features of the app.
- Your identity and responses to individual questions will be kept confidential.
Risks to study participants:
Standard risks associated with using smartphones and navigating rooms.

Benefits to research participants and others:
You will be helping the Worcester Art Museum be more accessible for all visitors.

Your participation in this research is voluntary.
Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact:
WPI Faculty Advisor: Rodica Neamtu, Email: rneamtu@wpi.edu or
WPI IRB Manager: Ruth McKeogh, Tel. 508 831-6699, Email: irb@wpi.edu or
WPI Human Protections Administrator: Gabriel Johnson, Tel. 508 831-4989, Email: gjohnson@wpi.edu

By signing below, you acknowledge that you have been informed about the study, and consent to be a participant in the study described above. If you are under the age of 18, your parent or legal guardian must also sign this form. Make sure that all of your questions are answered before signing this form. You are entitled to retain a copy of this consent agreement.

Printed Name __________________________
Study Participant Signature: ___________________________ Date:
Investigator Signature: ___________________________ Date:
Appendix B: Questionnaire

The Worcester Art Museum App’s user testing study would consist of two testing pools. The first pool would test in person during a one-day in-person trial with approximately 15 volunteers supplied by local nonprofits for the visually impaired such as Audio Journal and The Perkins School. The second pool would be an asynchronous testing stage designed to test the app’s user interface functionality. Pool one’s test would be focused on testing the app’s practical usage in a museum environment, as well as the placings and usages of NFC tags and QR codes. Pool two’s test would be focused solely on the effectiveness of the application’s user interface. Both pools would include people with a variety of levels of visual impairment.

For pool one we would have volunteers in a small portion of the Worcester Art Museum or a simulated museum environment and ask volunteers to navigate the site while using the app. The app will allow the testers to get additional information about the simulated art pieces. For pool two, volunteers would be able to complete the testing at their leisure using a mobile device. All volunteers will be given a task list and a survey about their experience to be completed after the task list. No volunteer will be asked to spend more than 20 minutes of their time, and risks associated will be standard to operating a smartphone and navigating public spaces.

Task List:

- Open the application
- Navigate to the app’s help section and read/listen to the material provided
- Change color palettes in the app via the built-in color customization bottom sheet.
- Find a few specific museum items in the app's user interface
  - Play the audio file associated with a museum item
  - Use the Play, Skip, Pause, and Seek Bar audio controls
- Scan NFC tags as placed around the simulated exhibit
- Scan QR codes as placed around the simulated exhibit
- Navigate the app menus through the voice control bottom sheet
- Test the native Talkback or VoiceOver implementations
- Use the Speech to Text and Text to Speech features
  - Listen to audio descriptions as read off by Text to Speech
  - User Speech to Text to filter a list of art pieces

Question List:

1. Which accessibility options did you use when navigating the app? Check all that apply
   - TalkBack or VoiceOver
   - Text to Speech
   - Audio Descriptions
2. In your day-to-day life do you use an iOS device, Android device or Both?
   - iOS
   - Android
   - Both

3. Have you been to the Worcester Art Museum before?
   - Yes
   - No
   a. If yes, what was your method of learning about art pieces? Check all that apply
      - Audio Descriptions
      - Smart phone
      - Website link
      - Tour guide
      - Braille text

4. Have you used the Worcester Art Museum website to access information about the museum or different art pieces?
   - Yes
   - No

5. When testing the application did you use an iOS device or an Android device?
   - iOS
   - Android

6. Have you previously used other accessibility apps, such as the Audio Journal app?
   - Yes
   - No
   a. If you answered yes above, then please list which apps you have used in the past:

7. While navigating the application did you use NFC tags or QR codes to move around the app? If you are asynchronous, please check that option below.
   - Yes
   - No
   - Asynchronous

8. What is your level of visual or print accessibility needs?
   - No additional needs beyond glasses
   - Color blindness
   - Partial blindness
   - Full blindness
   - Partial print impairment
Numerical Ratings:
Rate the following app features on a scale from 1 to 5, with 1 meaning the feature was unintuitive and difficult to use and 5 meaning the feature was intuitive and easy to use.

<table>
<thead>
<tr>
<th>Unintuitive</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Intuitive</th>
</tr>
</thead>
</table>

1. If you used accessibility features, rate the ease of use and quality of our app’s integration of those features.
   a. Android TalkBack or iOS VoiceOver
   b. Voice Control
   c. Text to Speech
2. Finding art pieces via the app menu
3. Listening to audio provided by the Worcester Art Museum
4. Using voice search
5. List filtering
6. Switching color palettes
7. Using the help page
8. Using QR Codes
9. Using NFC tags
10. The overall app experience

Open Response Questions:
1. Did the app features work like you expected? If not, which features did not work like you expected?

2. Did you have any difficulty using the app? If yes, what was difficult?

3. Did you use the NFC tags to navigate the application? If so, were they useful additions in terms of accessibility and navigating the app?

4. (Only answer the next question if you used TalkBack or VoiceOver when testing the app) When navigating the app with TalkBack or VoiceOver, did all the buttons have descriptive labels? If not, which labels would you change?

5. Are there any features you feel are missing? If so, what would you add?

6. Will you use the app in the future?
Appendix C: IRB Covid Background Information

Covid-19 Study Background Information

What is COVID-19?
COVID-19 is a respiratory virus spread by respiratory droplets, mainly from person-to-person. This can happen between people who are in close contact with one another (less than 6 feet). It is also possible that a person can get COVID-19 by touching a surface or object (such as a doorknob or counter surface) that has the virus on it, then touching their mouth, nose or eyes.

Can COVID-19 be prevented?
Current ways to minimize the risk of exposure to COVID-19 include “social distancing” which is a practice to decrease the potential for direct exposure to others who may have been exposed to COVID-19, for example by avoiding large gatherings or refraining from shaking hands with others. It is important to understand that since study participation may include increased travel outside of your home and increased exposure to others within a research site it may increase your exposure to COVID-19.

What are the risks of COVID-19?
For most people, the new coronavirus causes only mild or moderate symptoms, such as fever and cough. For some, especially older adults and people with existing health problems, it can cause more severe illness, including pneumonia. While we are still learning about this virus, the information we have right now suggests that about 3 of 100 people who are infected might die from the virus.

Who is most at risk?
Individuals over 60 and with chronic conditions such as cancer, diabetes and lung disease have the highest rates of severe disease from the infection.

What do we do to minimize risk for research participants?

- We ask every research participant if they have the symptoms of COVID-19 or have been in close contact with anyone who has or had COVID-19. If you are suffering from symptoms, please refrain from studying.
- During your research visits, we try to reduce the time you are exposed to other people as much as possible.
- During the study visit, the study staff must adhere to physical distancing guidelines. You will be seen in an area that always allows 6 feet of separation, except when contact is necessary to complete the study procedures for your visit.
- We are following the current clinical guidelines for cleaning rooms and equipment.
- Please feel free to bring your own face mask to wear during the study. Face masks will also be provided during the visit for those who request them.