ViaSight: An Accessibility Android app for Visually Impaired Individuals

Aayush Jain*, Aditi Rao†, Vaishnavi Mishra‡
Information Technology, National Institute of Technology Karnataka,
Surathkal, Mangalore, Karnataka, India 575025
*Email: aayushjn11@gmail.com
†mail: aditiraops@gmail.com
‡Email: vaishnavimishra050@gmail.com

***Abstract*—In today's world, the previously common three-dimensional interfaces of buttons on cellular devices are nonexistent, removed in favor of highly visual two-dimensional touch screens. While revolutionizing the ease of interaction for most this poses a great challenge in accessibility for visually impaired users. Most apps made today are not specially made for the visually impaired, so they create much confusion and result in problematic user experience. Unlike apps that provide visual stimulus through special sensors or other sensory feedback, we aim to substitute sight with both auditory and haptic feedback and command interfaces making smartphones more accessible in a non-intrusive and discreet way to the visually impaired.**

I. Introduction

With the current popularity of smartphone/handheld technology, many of us use touch-based interfaces frequently. However, such devices do not consist of any physical buttons and rely on the user's sense of sight to complete certain tasks. With over 253 million people suffering from moderate to serious visual impairment or blindness [1], there is a need to make life easier for them. A rise in the number of visually impaired is predicted due to the high prevalence of visual impairment amongst the elderly and large aging populations in countries across the globe. Accessibility had always proved to be a challenging problem while developing for mobile devices mainly because unlike in the past when three-dimensional buttons were available for interaction; they have been replaced by flat touch screens. All interaction is now highly visual through two-dimensional graphics, making commonly available smartphones hard to use for those with visual impairment.

Recent advances in the instrumentation technology of sensory substitution have presented new opportunities to develop human-machine interfaces by substituting one sense with another [2]. Sensory substitution has been the most common way to allow for increased accessibility, and voice assistants such as Google Assistant and Siri currently utilize this. ViaSight aims to include an additional dimension of interaction and substitution through gesture commands and haptic feedback, allowing for a more efficient and fulfilling experience. Voice commands alone reduce the ability to have a degree of privacy while using a smartphone, even when using earphones for audio output. In this light, ViaSight provides touch-based commands to allow for more discreet access to commonly used functionality in smartphones.

II. Literature Survey

Most modern attempts at improved smartphone accessibility have been through voice commands and personal assistant style applications. Google Assistant and Siri form the primary means of voice-based accessibility interaction with Android and iOS devices. However, these approaches do not allow for any haptic feedback or touch-based interaction with the device. Gesture-based application launchers do exist, but so far do not have widespread usage with voice commands. Due to the large availability of affordable smartphone technology, such devices hold a lot of promise in the field of *low vision rehabilitation* [3]. Various steps have been taken to improve the quality of life for the elderly using smartphones. One such step was *PhonAge* [4], an adapted smartphone interface for elderly people.

Many tools have been proposed for navigation using the many sensors available on smartphones to make life easier for the blind. These include, but are not limited to, *BlindNavi* [5] and *PERCEPT-II* [6]. Specific research done on spatial processes in blind and visual impairment has proven to be effective in relieving some of the unintentional stresses arising due to the inherent nature of touch-based technology [7].

Since the exponential growth of IoT based systems since 2010, the technology market has seen a large number of wearable devices for various purposes. Wearable technology can prove very useful in providing an alternate "view" of the world by using the many sensors embedded in these objects. A large amount of research has been done in the field of smartphones as "wearables" to minimize the costs involved in manufacturing such devices (PCB design, sensor integration). Since smartphones have multiple sensors and are made with ever-improving technological standards, it is optimal to use such devices. Specific research on this topic has been done concerning visually impaired people [8].

III. Problem Statement

Commercially available solutions for improving accessibility do not have a large focus on integrating haptic and auditory feedback for effective use of smartphones. To overcome these shortcomings, we aim to create an accessibility application catering to the needs of the visually impaired, combining haptic and auditory feedback with voice controls to allow for a non-intrusive, discreet and efficient means of interaction.

A. Objectives

ViaSight launcher must:

- Allow for visual sensory substitution through touch and speech.
- Application must allow for discreet and non-intrusive interaction with the device.
- Minimizing the number of commands to access functionality and execute commands.
- Must enhance the experience of visually impaired persons with their smartphones.

IV. Methodology

ViaSight was implemented as a **launcher style Android application**, acting as a layer to translate voice and gestures into similar actions a user would normally perform. It acts as a replacement of the default system launcher and later relies on system-wide accessibility options to complete the experience.

Sensory substitution was done by providing different **vibration patterns**, as well as assigning **swipe gestures** to trigger various functionalities. Each pattern holds a different purpose. It helps to train the user to rely on haptic feedback during certain events. These events (in the case of the app) are incoming notifications, positive TTS (Text-To-Speech) responses and negative TTS responses.

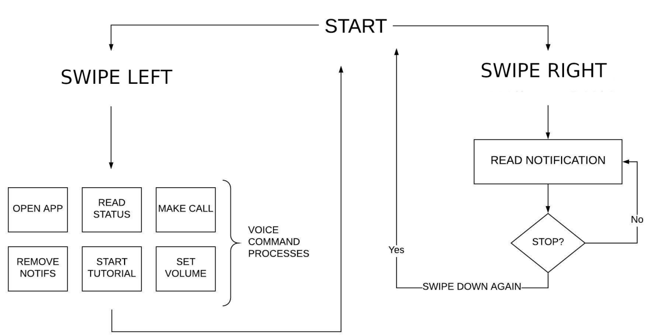

Fig. 1. App flowchart

Swipe gestures implemented include:

- Swipe left to enable voice commands (as mentioned below)
- Swipe right once to read notifications
- Swipe right again to stop reading notifications

ViaSight currently supports voice commands with the following syntax:

- ***open*** <application_name>
- ***what*** is the *date/time/battery percentage*?
- ***call*** <contact_name>
- ***remove*** *notifications*
- ***play*** *tutorial*
- ***set*** *volume silent/vibrate/normal*

The above is accomplished through acquiring permissions to set vibration, record audio, make calls and read contacts. However, all processing occurs entirely on the device, keeping personal data secure. Moreover, the system does not rely on internet access (does not rely on Google Assistant APIs) and thus can be used **offline** with ease.

Since the app was made for visually impaired people and allows for near complete control of the device, the user interfaces involved are practically empty and only show incoming notification information. The rest of the app relies on audio, sensory and vibrational responses.

V. Results and Analysis

Due to Android's implementation of "dangerous permissions", the user must grant the app access to **contact information**, **notification information** and **make calls**. The app's simplistic design is because it is primarily targeted at those who have a visual impairment (not necessarily blind), and thus the only visual cue added is the notification information in large text as in 2.

The rest of the app is aimed towards bolstering the user's reliance on auditory or haptic feedback from the device. Each time a notification is "posted", the device vibrates with a pre-set waveform that helps uniquely identify notifications. Additionally, a sound is made to ensure that the user hears that there is a new notification. The notifications displayed are of the format:

1) **Package:** This tells which Android app the notification is from
2) **Title:** This is the title of notifications visible in the notification bar
3) **Text:** This is the content of the notification as seen in the notification bar
4) **Time:** This is the time at which the notification was posted

The results of the various voice commands are discussed below:

- ***Open*** *<application_name>*: This triggers a method to retrieve all installed apps on the system. It then filters out apps that match the processed result of speech synthesis. If a match is found, then the system says "Launching <app_name>" and launches the corresponding app. Otherwise, it asks the user to say it clearly by saying that the app was not found.
- ***What*** is the *date/time/battery percentage*?: This is one of the commands that are multi-functional. It begins with the keyword *what* and is followed by either of the previously mentioned commands. The response for each is as follows: time in **24-hour** format, date in **DD Month YYYY** format, and the battery percentage is told as **87%** (for instance).
- ***Call*** *<contact_name>*: This triggers a method to retrieve all saved contacts. It then filters out contacts that match the processed result of speech synthesis. If a match is

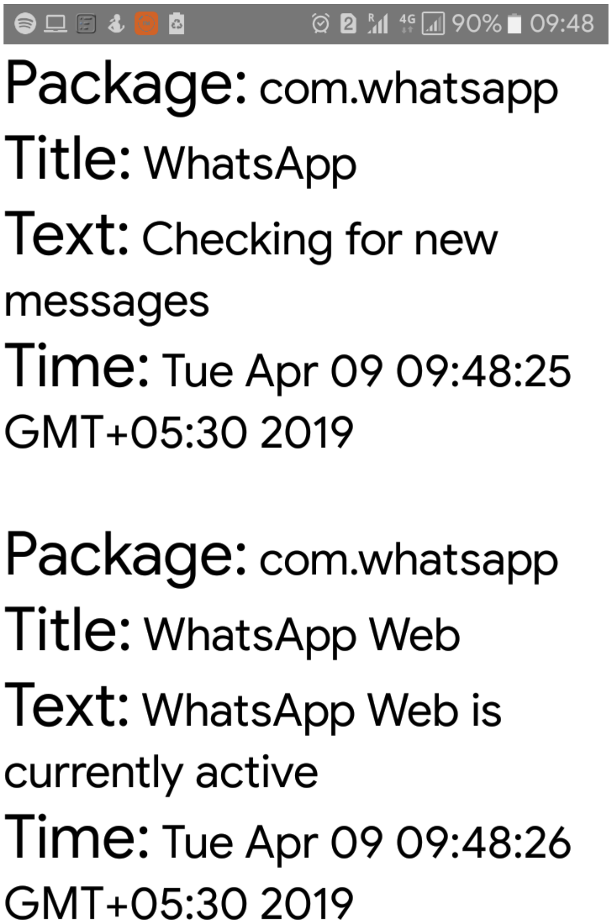

Fig. 2. App screenshot

found, then the system places a call to that contact. Otherwise, it asks the user to say it clearly by saying that the contact was not found.

- ***Remove*** *notifications*: This command clears all the pending notifications and makes the screen blank. It also invokes the TTS component to inform the user that the notifications have been cleared.
- ***Play*** *tutorial*: Upon hearing this command, the system invokes the TTS component, and it reads out the different possible commands supported by the system and the gestures that can be used.
- ***Set*** *volume silent/vibrate/normal*: This command does exactly what it says. It sets the system volume to either of the three possible configurations.

Each speech-based commands that respond with auditory feedback are linked with a sensory substitution that enables the user to pre-empt the response from the device. Positive responses from the system are linked to a "positive vibrational waveform", whereas negative responses (app not found/contact not found/command not recognized) are linked with a "negative vibrational waveform".

As mentioned in IV, swiping to the left invokes the speech commands. On swiping to the left, a sound is played signifying the start of audio recording, and the same audio is played later to mean the end of audio recording. Thus the user has audio cues to know when to start speaking. On swiping to the right, the system retrieves all the notifications and passes them to the TTS for speech. If no new notifications exist, the TTS responds by saying "You do not have any pending notifications". If the user, swipes right again after swiping once, the TTS is paused if it was speaking. Then the user can swipe to right once more to listen to the notifications.

The speech recognition system is pretty robust, and the app also handles any possible edge-cases that could arise like *incomplete voice commands* (e.g., what is, call, etc.), and even *unknown commands* (e.g., turn off the device, etc.). Additionally, if any notifications are unread when the app is closed (due to launching a new app or shutting down the system or closing the screen), it saves them to memory and then restores them when needed.

The app design was done keeping in mind a few key requirements:

- App must be lightweight (APK size=2.9 MB, download size=2.5 MB)
- App must be fast and responsive (Specialized APIs were used to ensure that the models were serialized faster than traditional Java objects)
- App must reflect modern standards and must use the best-intended practices (App was written in Kotlin and uses special features available in the programming language. It also uses the latest libraries and targets Android Marshmallow to Android Pie)
- App does not waste system resources and does not require heavy RAM usage (All acquired resources are released within the lifecycle of the app itself)

The major issues that the app faces are as follows:

- Minimal command set.
- Every notification that is posted is read. This includes download progress, configuration changes in the system and constant barrages of repetitive notifications.
- Lack of a richer UI

These issues can be addressed easily:

- The command set can easily be expanded to include more features. However, certain features like sending SMS to contact are not possible as they violate Google's updated privacy policy.
- This issue is slightly harder to address. A filter can be set to ignore system notifications, but most of this would require some processing of the content (machine learning models) which could raise security and privacy concerns.
- The UI was intentionally kept minimal as the system does not rely on visual aids. Its main purpose was to substitute visual aids with sensory, audio and haptic

feedback mechanisms.

VI. Conclusion

ViaSight is a holistic application aimed at bridging the gap between the differently abled and the real world. ViaSight is an immersive Android application aiming to improve smartphone accessibility for the visually impaired. By using haptic and auditory feedback, we aim to simplify the working of the mobile phone and expand its scope to a larger audience. Since smartphones are becoming increasingly common with many people owning multiple such devices, it is important to give the differently abled a chance to be able to utilize the same technologies with a similar approach.

Individual contribution

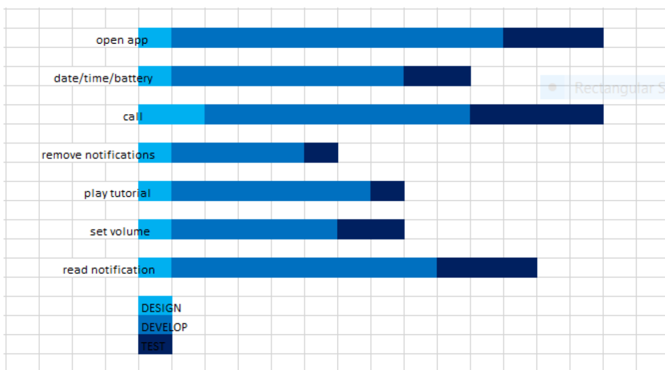

Fig. 3. Gantt Chart

Contributor	Contribution
Aayush Jain	App implementation, design
Aditi Rao	App design, testing, feature proposals
Vaishnavi Mishra	Idea proposal, feature proposals, user feedback analysis

TABLE I
Individual contribution

References

[1] W. H. Organization, "Blindness and vision impairment," https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, October 2018.

[2] P. B. y Rita and S. W. Kercel, "Sensory substitution and the humanmachine interface," *Trends in Cognitive Sciences*, vol. 7, no. 12, pp. 541 – 546, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661303002900

[3] D. Irvine, A. Zemke, G. Pusateri, L. Gerlach, R. Chun, and W. M. Jay, "Tablet and smartphone accessibility features in the low vision rehabilitation," *Neuro-Ophthalmology*, vol. 38, no. 2, pp. 53–59, 2014. [Online]. Available: https://doi.org/10.3109/01658107.2013.874448

[4] F. Arab, Y. Malik, and B. Abdulrazak, "Evaluation of phonage: An adapted smartphone interface for elderly people," in *Human-Computer Interaction – INTERACT 2013*, P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 547–554.

[5] H.-E. Chen, Y.-Y. Lin, C.-H. Chen, and I.-F. Wang, "Blindnavi: A navigation app for the visually impaired smartphone user," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '15. New York, NY, USA: ACM, 2015, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/2702613.2726953

[6] A. Ganz, J. M. Schafer, Y. Tao, C. Wilson, and M. Robertson, "Percept-ii: Smartphone based indoor navigation system for the blind," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2014, pp. 3662–3665.

[7] Z. Cattaneo, T. Vecchi, C. Cornoldi, I. Mammarella, D. Bonino, E. Ricciardi, and P. Pietrini, "Imagery and spatial processes in blindness and visual impairment," *Neuroscience & Biobehavioral Reviews*, vol. 32, no. 8, pp. 1346 – 1360, 2008, special section: The European Workshop in Imagery and Cognition: Neurocognition and Visual Imagery. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0149763408000638

[8] H. Ye, M. Malu, U. Oh, and L. Findlater, "Current and future mobile and wearable device use by people with visual impairments," in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 3123–3132. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557085