**Periphery Deliverable 4**

Isaac Engler, Olivia Pooler, Jack Bisson, Markus Hiyazaki, Ebban Pudden

University of Ottawa, Department of Engineering

GNG 1103, Group 5

Professor Knox

October 13, 2024

**Table of Contents**

[Abstract 2](#_Toc1071815854)

[Introduction 2](#_Toc1074409395)

[Subsystems 3](#_Toc505671244)

[Solutions 5](#_Toc1601817574)

[Evaluation of Purposed Solutions Against Design Criteria Using a Selection Matrix 7](#_Toc866153578)

[Conclusion 9](#_Toc1156353907)

[Appendix 9](#_Toc1480984746)

# Abstract

This report contains the contents of our ideate stage. It also contains the three solutions we’ve developed and plan on presenting to Shabodi during our meeting on the 21st of October.

# Introduction

This focus of the is project is to showcase the capabilities of Shabodi’s NetAware API by creating a product that can have a positive impact on those with visual impairments. The goal of this project is to create a smart glass product that through visual and auditory cues and feedback, allows those living with visual impairments to navigate their daily lives with less difficulties.

# Subsystems

**Table 1:** Inputs (Isaac)

|  |  |
| --- | --- |
| **Feature** | **Capabilities** |
| Camera | * Object recognition * Facial recognition |
| Microphone | * Auditory input (Voices, etc.) |
| Sensors | * IR distance sensor |

**Table 2:** Outputs (Olivia & Markus)

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Type** | **Purpose** | **Feasibility** |
| Object detection | Auditory | When an obstacle or individual gets within a certain distance of the user, a verbal message should be heard to indicate both distance and what is in front of the user. | API’s such as OpenAI, Microsoft Azure, and Eden AI offer access to image recognition software. However, most API’s (except for OpenAI) are not open source, nor written in Python. |
| Object detection - visualisation | Visual | The same concept as the subsystem above, however a vector and text should be displayed around the object and/or individual to both highlight and identify what the user is looking at. Moreover, the colour of the vectors should be customizable to suit the preference of the user as well as adjust to ambient lighting (i.e., increase in colour saturation when in low light lighting). | Same feasibility as the subsystem above. |
| GPS notifications | Visual | The user should receive notifications which state both directions and depict an image of an arrow (provides a rough idea of where to go). | Feasible, however GPS software will have to be integrated into Shabodi’s API. |
| Hazard alert | Auditory and/or visual | In cases of poor weather, traffic accidents, or even criminal activity (i.e., shootings) the user should receive a notification which displays a highly saturated background alongside displaying text which is then read to the user, stating what the warning is. | Although, it would be feasible. It depends on what SIM provider the users have, as only certain SIMs are compatible with Alert Ready’s warning system. |
| Text recognition | Auditory | Scan things like menus, signs and displays (virtual?) to provide text-to-speech description | This technology is widely used already. Therefore, can most likely find a library of code to use. |
| Different languages | Visual/auditory | Provides auditory and visual cues in different languages | Use a translational program that is already developed. |
| Translation | Auditory | Can translate verbally given information into language of user’s choice | Use a translational program that is already developed. |
| Facial recognition | Visual/auditory | Can recognize and provide visual or auditory cues for faces (maybe they must be scanned into the system beforehand?) | This technology is widely used already. Therefore, can most likely find a library of code to use. |
| Emergency response contact | Visual/auditory | Will alert chosen emergency contact upon receiving specific hand gesture | This would be simply to code |

**Control of outputs**

* Users can control which outputs they will use based on the settings in the app
* Having too many notifications or outputs will be too overwhelming, and we must remember that all visually impaired people have different strengths and weaknesses so being able to customize outputs depending on the user and what they are doing will be helpful

**Table 3:** User Experience / Interface (Ebban)

|  |  |  |
| --- | --- | --- |
| **Feature** | **Subcategory** | **Method** |
| Home Screen | Interface | * Dashboard: Displays time, battery life, and notifications. * Large Icons: Key functions like Messages, Navigation, and Music with clear labels. * Voice Activation: Small microphone icon for hands-free control. |
| Navigation (Potential External Device) | Interface | * Swipe Left/Right: Switch between features easily (e.g., Notifications, Settings). * Scroll Down: For more options within a selected feature. |
| Icons | Experience | * Simple Design: Use easily recognizable icons (bell for notifications, gear for settings). * Feedback: Visual or haptic cues when options are selected. |
| Quick Access Bar | Interface | * Bottom Control Bar: Offers shortcuts to frequently used features (e.g., Music, Calls). * Customizable: Users can select which features to include. |
| Settings | Interface | * Easy Navigation: Group settings into categories (Display, Connectivity). * Toggles: Simple switches for adjustments. |
| Help | Experience | * Onboarding Tutorial: A quick guide for first-time users. * Help Icon: Always accessible for tips and FAQs. |
| User Flow | Experience | * Put on Glasses: Home screen appears. * Swipe to Navigate: Quickly access different features. * Select Options: Tap an icon or use voice commands. * Control Bar: Manage music or calls easily. * Adjust Settings: Simple menu for customization. * Access Help: Use the help icon anytime. |
| Design Focus | Experience | * Minimalist: Keep the interface clean and straightforward. * High Contrast Colors: Ensure readability in various lighting conditions. |

**Table 4:**  Security (Jack)

|  |  |
| --- | --- |
| **Feature** | **Method** |
| User verification | Fingerprint reader needed to access service |
| File encryption | Files sent to phone to be saved locally  Any file sent outward will be encrypted |
| Camera use indicator | Any time camera is in use a light of some sort must be turned on to warn those being recorded of it being done |
| Microphone disabled while not in use | Needs physical switch or verbal signal to begin recording |
| Vision coverage maximum | Ensuring the user can constantly see their surroundings, preventing them from accidentally running into their surroundings |

# Solutions

**Table 5:** Chosen subsystems for Solution #1 (Daily-aid model)

|  |  |
| --- | --- |
| **Subsystem** | **Features** |
| Input | Camera:   * Object recognition |
| Output | Visual:   * GPS Notifications * Hazard Alert   Auditory:   * Objection detection * Hazard Alert |
| User Interface | * Dashboard * Minimalistic |
| Security | * File encryption * In-use indicator (light) for camera * Regulation of pop-ups for safety and visibility |

**Table 6:** Subsystem physical and software component requirements for Solution #1

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| Camera | * Visible light camera   + Can detect objects the user cannot see |
| Microphone | * Speech detection   + Can repeat what people have said to allow the user to listen |
| Display | * Very limited display (more focused on auditorial than visual) * High contrast |
| Speakers | * Clear, easy to understand * Able to sound human * Able to connect to hearing aids |

**Table 7:** Chosen subsystems for Solution #2 (Technological advanced and high-end model)

|  |  |
| --- | --- |
| **Subsystem** | **Features** |
| Input | Camera:   * Facial recognition * Object recognition   Infrared sensor:   * Depth perception |
| Output | Visual:   * Object detection * Hazard alert * Provides output in different languages * Facial recognition   Auditory:   * Object detection * Hazard alert * Provides output in different languages * Facial recognition |
| User Interface | Home screen:   * Dashboard * Voice activation   Design Focus:   * High contrast |
| Security | * Fingerprint sensor * File encryption * In-use indicator (light) for camera * Microphone only activated by key word |

**Table 8:** Subsystem physical and software component requirements for Solution #2

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| Camera | * Visible light camera * Able to read barcodes + QR codes * Google lens capabilities * Infrared sensor * Measures distance and stops user from walking into things |
| Microphone | * Voice activated assistant (Jarvis)   + Capable of doing calls * Translation model   + Allows user to translate what others are saying |
| Display | * Looks cool   + Sci fi futuristic star trek * External display to show off how cool you are |
| Speakers | * Tuned for music * Able to connect to earbuds/speakers through Bluetooth |

**Table 9:** Chosen subsystems for Solution #3 (Severe visual impairment model)

|  |  |
| --- | --- |
| **Subsystem** | **Features** |
| Input | Camera:   * Object recognition * Facial recognition |
| Output | Visual:   * Hazard Alert   Auditory:   * Objection detection * Text recognition * Hazard Alert * Facial recognition |
| User Interface | * Minimalistic (little to no visual cues) * Voice activation |
| Security | * File encryption * In-use indicator (light) for camera |

**Table 10:** Subsystem physical and software component requirements for Solution #3

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| Camera | * Visible light camera   + Able to detect objects * Infrared camera   + Capable of reading distance of objects |
| Microphone | * Speech detection   + Can repeat what people have said to allow the user to listen |
| Display | * Not quite needed for visually impaired users |
| Speakers | * Clear, easy to understand * Able to sound human * Able to connect to hearing aids |

# Evaluation of Purposed Solutions Against Design Criteria Using a Selection Matrix

Each of the subsystems will be rated from 1 to 3 based on how to fit the design criteria, with 3 being the best (fits criteria the most) and 1 being the worst (fits criteria the least). The score will then be totaled to produce a final rating of each solution.

**Table 11:** Selection Matrix for Solution #1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Solution #1**  **Ratings** | | | | | |
| **Design Criteria** | Input | Output | User Interface | Security Features | **Total rating (/12)** |
| User-Friendly Interface | 3 | 1 | 2 | 2 | **8** |
| Compatibility | 2 | 1 | 2 | 1 | **6** |
| Reliability | 3 | 3 | 3 | 1 | **10** |
| Security | 1 | 1 | 2 | N/A | **4** |
| Customization Options | 2 | 2 | 2 | 2 | **8** |
| Support and Maintenance | 3 | 3 | 2 | 2 | **10** |
| Scalability | 1 | 3 | 2 | 2 | **8** |
| Integration Capabilities | 1 | 2 | 1 | N/A | **4** |
| **Final Rating (/90)** | | | | | **58** |

**Table 12:** Selection Matrix for Solution #2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Solution #2**  **Ratings** | | | | | |
| **Design Criteria** | Input | Output | User Interface | Security Features | **Total rating**  **(/12)** |
| User-Friendly Interface | 1 | 3 | 1 | 1 | **6** |
| Compatibility | 3 | 3 | 3 | 3 | **12** |
| Reliability | 1 | 1 | 1 | 2 | **5** |
| Security | 3 | 3 | 1 | N/A | **7** |
| Customization Options | 3 | 3 | 3 | 3 | **12** |
| Support and Maintenance | 1 | 1 | 1 | 1 | **4** |
| Scalability | 3 | 2 | 3 | 1 | **9** |
| Integration Capabilities | 3 | 3 | 3 | N/A | **9** |
| **Final Rating (/90)** | | | | | **64** |

**Table 13:** Selection Matrix for Solution #3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Solution #3**  **Ratings** | | | | | |
| **Design Criteria** | Input | Output | User Interface | Security Features | **Total rating**  **(/12)** |
| User-Friendly Interface | 3 | 2 | 3 | 3 | **11** |
| Compatibility | 1 | 2 | 1 | 2 | **6** |
| Reliability | 2 | 2 | 2 | 3 | **9** |
| Security | 2 | 2 | 3 | N/A | **7** |
| Customization Options | 2 | 1 | 1 | 1 | **5** |
| Support and Maintenance | 2 | 2 | 3 | 3 | **10** |
| Scalability | 2 | 1 | 1 | 3 | **7** |
| Integration Capabilities | 2 | 1 | 2 | N/A | **5** |
| **Final Rating (/90)** | | | | | **60** |

# Conclusion

There were several ways you could have approached this project, and we had many ideas. Between our design criteria and what Shabodi have set out to do, we believe that Solution 3 is the best, and the one we will be going forward with.

# Appendix

**Table 10:** Rough subsystem ideas

|  |  |
| --- | --- |
| **Device** | **Chosen Idea** |
| Cameras | * Visible light camera |
| Microphone | * Can detect loud noises and translate text |
| Display | * Explain what objects are in a legible format |
| Speakers | * Inform user of object detection |

References

Eden AI. (n.d.). Top 10 object detection apis. <https://www.edenai.co/post/top-10-object-detection-apis>