SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING

Date: May 24, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Luis Abraham Millan
David Blake Tsuzaki
Yang Li

ENTITLED

Portable Reading Assistant Headset for the Visually Impaired

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING
BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING

Thesis Advisor

Thesis Advisor

Department Chair

Department Chair

Portable Reading Assistant Headset for the Visually Impaired

by

Luis Abraham Millan
David Blake Tsuzaki
Yang Li

Submitted in partial fulfillment of the requirements
for the degrees of
Bachelor of Science in Computer Science and Engineering
Bachelor of Science in Electrical Engineering
School of Engineering
Santa Clara University

Santa Clara, California
May 24, 2017

Portable Reading Assistant Headset for the Visually Impaired

Luis Abraham Millan
David Blake Tsuzaki
Yang Li

Department of Computer Engineering
Department of Electrical Engineering
Santa Clara University
May 24, 2017

ABSTRACT

Most products in the domain of improving the visually impaired textual understanding focus on the direct translation or dictation of text that is in front of a user. Seldom focus on any type of textual understanding that goes beyond literal translation. In this report, we are documenting the implementation of a novel wearable device that allows the visually impaired to have a better understanding of the textual world around them by having a system learn a textual understanding for them and then providing more significant and natural feedback based on a users semantic queries regarding the text at their gaze.This document includes the requirements, design, use cases, risk tables, workflow and our projected development timeline for this device we are developing. This report also provides a way for us to review our progress and justify decisions we make as we advance in the development phase.

Table of Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Current Solutions . . . 1
1.3 New Solution . . . 2

2 Requirements **3**
2.1 Functional Requirements (*The system will...*): . . . 3
2.2 Non-functional Requirements (*The system will be...*): . . . 4
2.3 Design Constraints . . . 4

3 Use Cases **5**

4 Activity Diagram **9**

5 Technologies Used **11**
5.1 Tesseract Optical Character Recognition Engine: . . . 11
5.2 TensorFlow: . . . 11
5.3 OpenCV: . . . 11
5.4 NumPy: . . . 11
5.5 scikit-learn: . . . 12

6 Architectual Diagrams **13**

7 Design Rationale **15**
7.1 Architecture . . . 15
7.2 Technologies . . . 16

8 Test Plan **18**
8.1 Alpha Testing . . . 18
8.2 Beta Testing . . . 18

9 Risk Analysis **19**

10 Sustainability **21**

11 Development Timeline **25**

List of Figures

3.1 Use Cases 6

4.1 Activity Diagram 10

6.1 Level 0 Block Diagram 14
6.2 Level 1 Block Diagram 14

11.1 Use Cases 26

List of Tables

9.1 Risk Table 20

Chapter 1

Introduction

1.1 Motivation

Communication is the hallmark of our interactions as humans, occurring across different media, platforms, and entire paradigms. Much of the information we consume on a daily basis is provided in very specialized media, such as a newspaper or a billboard, that cater to only one specific sense, such as vision. This presents a particular challenge for individuals with sensory disabilities. Every day people absorb visual, sonic, and touch information from their surroundings. The visually impaired rely on a heightened sense of sound and touch to obtain information and are hindered from being able to easily obtain data from visual texts, such as posters, newspaper, fliers, etc. This hindrance not only affects the ability to obtain important textual information, such as warning or caution signs, but it also statistically raises the likelihood of unemployment.

1.2 Current Solutions

The Braille alphabet has been one of the most common aids in bridging the gap between textual information and people with visual disabilities. However, one of the problems with Braille is that it depends on text being translated and printed on a medium that the user can touch. More importantly, Braille suffers from a low literacy rate within the visually impaired community because it requires teachers with specialized training, a luxury that is not always available at public schools in the U.S.

There are many products on the market that serve as an alternative to Braille. One example is the FingerReader by the MIT Media Lab, an electronic device that dictates the text the user touches in a document. However it can only read font that is of 12-point that the user can physically touch.

Another example, the OrCam MyEye, is a dedicated headset that also performs live text dictation. However, it is priced at $2,500 and still requires a gesture input that assumes the user has some visual capability. There is a general issue with both products directly feeding the user everything from the text content. People who are not visually impaired can skim the content or skip around to get a brief idea of the content; the visually impaired must finish listening to the entire passage to understand the whole content.

1.3 New Solution

To address this problem, we are proposing to design an assistive Optical Character Recognition (OCR) system consisting of a portable headset that captures a feed of the users surroundings, a front-end mobile application that performs live OCR of the text within this feed, and a back-end framework for building a model of textual understanding. This system is capable of reading aloud the key points of the text the user is positioned to gaze at. By using haptics and computer vision, we address the shortcomings of current solutions by allowing the user to focus on text both as near as a handheld newspaper and as far as a billboard. By using a mobile phone for processing, rather than dedicated hardware, we address another key shortcoming of the current solutions by keeping the cost of the device down and allowing the system to build a model for better dictation in the future. With this system, we hope to address one of the biggest daily challenges of individuals who are visually impaired, a very underserved segment of our society.

Chapter 2

Requirements

The Requirements section presents a categorized and itemized list of project requirements. Categories include Functional and Non-functional Requirements and Design Constraints. Functional requirements define what must be done, while non-functional requirements define the manner in which the functional requirements need to be achieved. Both categories have sub-categories, determined by the importance of a given requirement. Design Constraints are similar to non-functional requirements but constrain the solution and not the problem.

2.1 Functional Requirements (The system will...):

- Critical
  - have sensors to communicate with the user and detect the users surroundings
  - communicate with the user through haptic feedback and voice dictation
- Recommended
  - have a learning model to tag specific instances of text and symbols and improve overall recognition

2.2 Non-functional Requirements (The system will be...):

- Critical
  - easy and intuitive to recognize text in the users environment and dictate it to the user
  - light and untethered from a large computing system
  - conform to the Federal Communications Commission guidelines on wearable devices[1]
- Recommended
  - maintainable for future usage and/or upgrades
- Suggested
  - generally aesthetically pleasing so that it does not draw too much attention from the users surroundings

2.3 Design Constraints

- The main device is a wearable headset whose hardware is self-contained
- The main devices main communication with the outside world is through a smartphone
- The main device communicates primarily through sound and touch, and not through vision

[1]https://www.fcc.gov/general/ingestibles-wearables-and-embeddables

Chapter 3

Use Cases

The Use Cases section defines specifics regarding the main, expected interactions between users and the system.

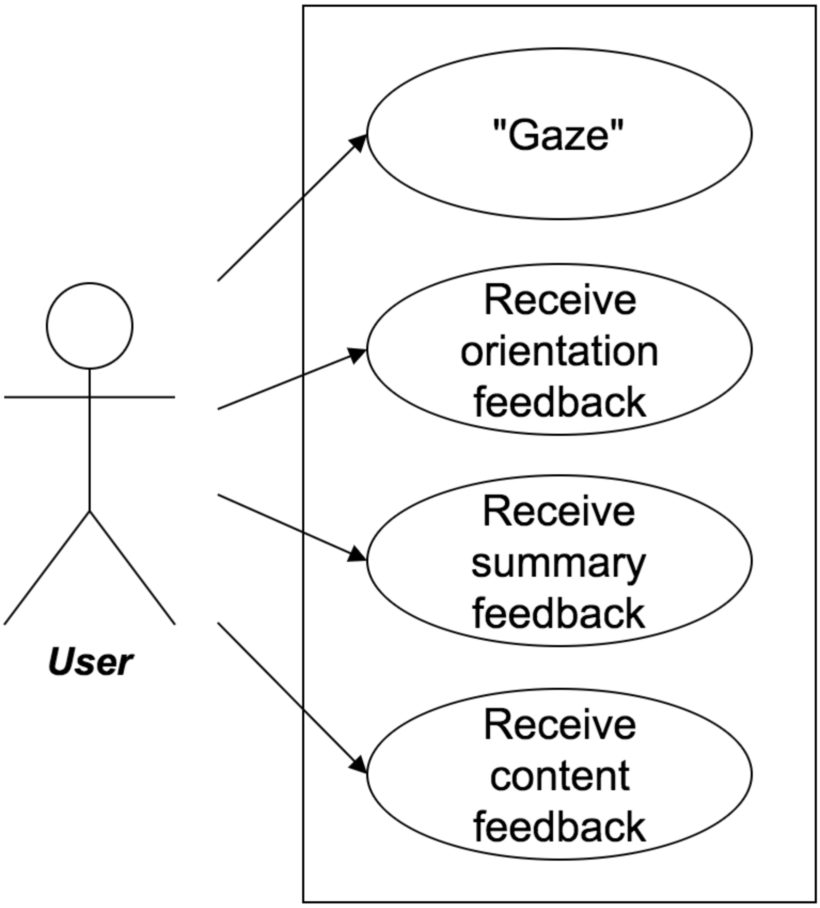

Figure 3.1: Use Cases

"Gaze"

Goal Identify and obtain graphical input

Actor User

Preconditions Device must be turned on

Steps User stare at the document he will to know

Postconditions Graphical input pass to optical character recognition engine

Exceptions Unstable input, such that the motion of the headset is outside the threshold

Receive orientation feedback

Goal Inform the user to change the way he is looking at the document, such as tilt his head

Actor User

Preconditions Graphical input pass to optical character recognition engine

Steps Follow the instructions

Postconditions Graphical input pass to optical character recognition engine

Exceptions User choose not to recognize the text

Receive summary feedback

Goal Provides the user with a summary of the document

Actor User

Preconditions Document recognized by OCR and processed by text summarization engine

Steps Auto output to user after successful scan for document

Postconditions Request user for content feedback

Exceptions User choose to abort

Receive content feedback

Goal Provides the user with the content of the document

Actor User

Preconditions User received summary feedback and wished more detailed content

Steps After the user listen to the summary feedback, he can press the control button to initiate the action

Postconditions N/A

Exceptions User choose to abort

Chapter 4

Activity Diagram

Figure 4 is the activity diagrams that show the flow of actions of the user.

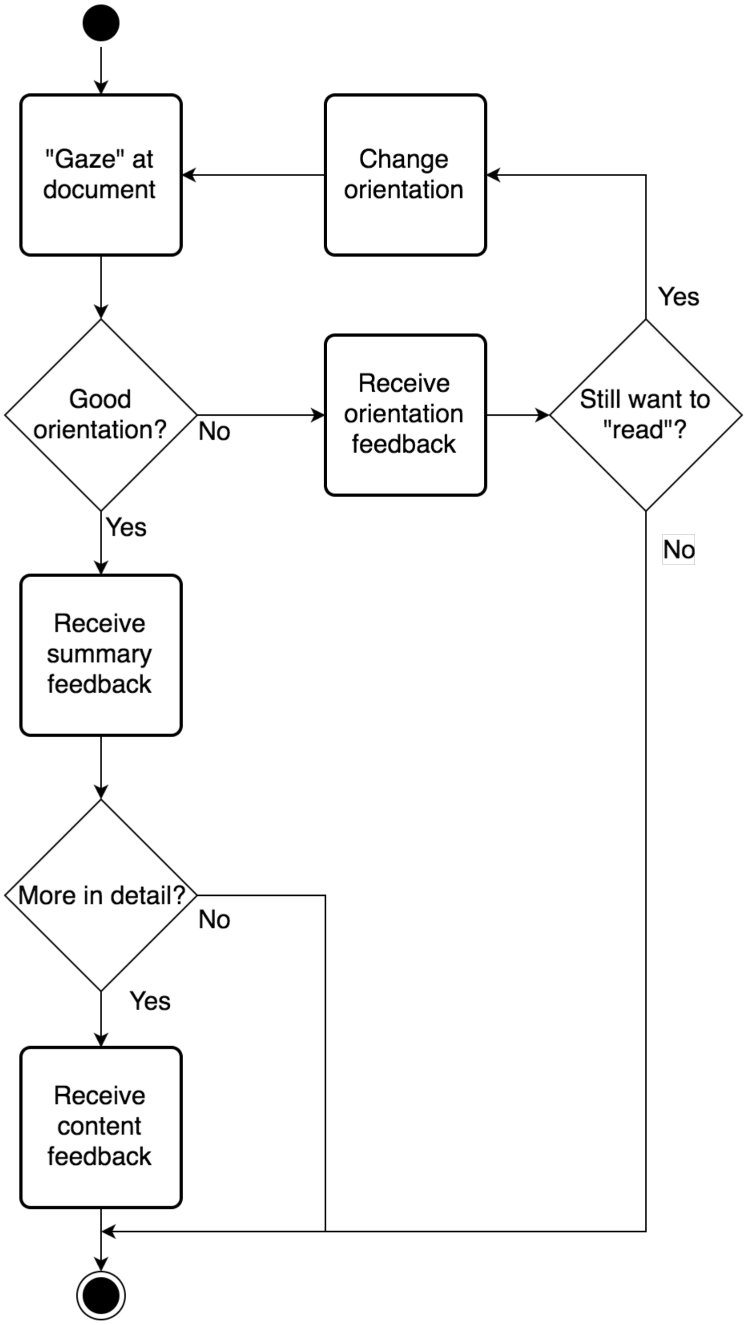

Figure 4.1: Activity Diagram

Chapter 5

Technologies Used

5.1 Tesseract Optical Character Recognition Engine:

An open source OCR engine licensed under Apache License, Version 2.0[1]. It is one of the most accurate open source OCR engines currently available.

5.2 TensorFlow:

An open source library of programing functions for machine learning. Licensed under Apache License, Version 2.0[1].

5.3 OpenCV:

An open source library of programing functions mainly aimed at real-time computer vision. Licensed under BSD license[2].

5.4 NumPy:

An extension (numerical and scientific library) to the Python programming language, it adds support for large, multidimensional arrays and matrices. Licensed under BSD-new license[2].

[1] https://www.apache.org/licenses/LICENSE-2.0.txt

5.5 scikit-learn:

A free machine learning library for Python programming language, designed to interoperate with NumPy and SciPy. Licensed under BSD license[2].

[2]https://en.wikipedia.org/wiki/BSD_licenses

Chapter 6

Architectual Diagrams

This system has a data-flow architecture. The reason of choosing this architecture is because there are constant inputs received by the system, and the inputs in general goes through the same route within the system. The data-flow architecture also has build in concurrency, which can speeds up the process, especially the platform of the product is embedded.

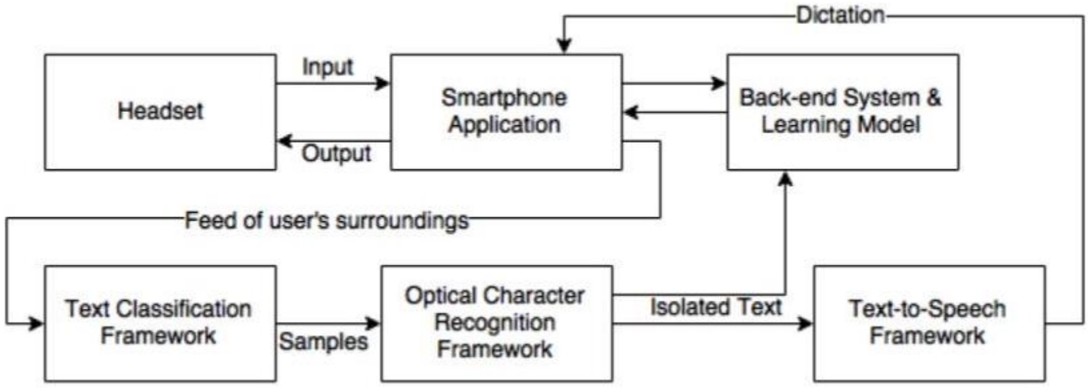

Figure 6.1: Level 0 Block Diagram

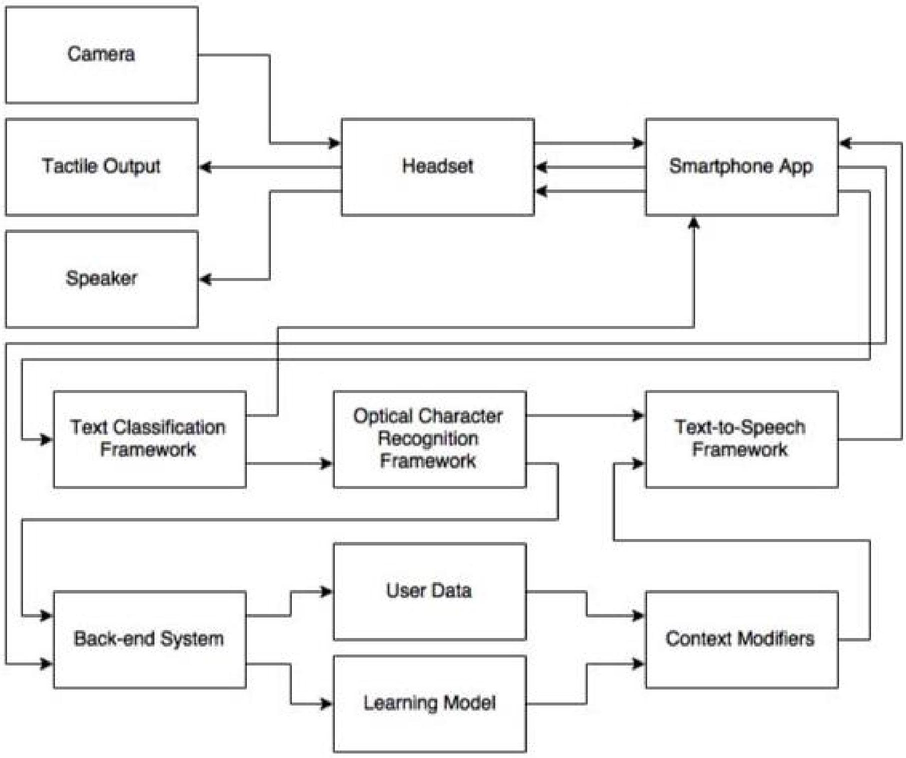

Figure 6.2: Level 1 Block Diagram

Chapter 7

Design Rationale

7.1 Architecture

- Hardware
  - Wired video module to the phone
    * Low-Latency, and therefore improving the systems responsiveness when providing feedback to user
  - Choice of Haptics + Sonics Feedback
    * Haptics allow us to better communicate spatial information regarding their gaze and important text or a users text of interest
    * Sonics are natural way of communicating textual information to a user
- Software
  - Textual Classification Framework before Optical Character Recognition Framework
    * Current Tesseract OCR Engine does poorly with text that is skewed, and therefore the Textual Classification will allow us to preprocess the video frame to first know whether there is text in the frame to be interested about and secondly to deskew the image as best as possible to improve performance of the OCR engine.

7.2 Technologies

- Tesseract Optical Character Recognition Engine
  - Currently, the most accurate open source solution to optical character recognition.
  - Highly documented with academic papers written about its architecture
  - Constantly being developed by a community of developers, so if we run into problems we have a community to ask questions to
  - Has both a C (python wrapper) and a C++ API
  - Supported by Google
- OpenCV
  - De Facto standard software libraries for computer vision
  - Lots of developer support
  - Open source
  - All of its data structures are compatible with Tesseracts API
- TensorFlow
  - Supported by Google
  - Scalable to distributed systems
  - Machine Learning Lead has most experience with this technology
  - Many examples of state of the art models implemented in TensorFlow making it easy to build upon, expand and even customize models to an applications requirements
- Numpy
  - Computationally efficient way of storing and computing operations involving vectors and matrices. Useful in our application because video frames and words are commonly represented and manipulated via vectors and matrix operations

- scikit-learn
  - Offers state of the art plug and play machine learning models that we can use for our initial designs and provide a baseline for the performance of our system

Chapter 8

Test Plan

The following describes how the product will be tested.

8.1 Alpha Testing

Function Testing

During function testing, each part of the system will be tested individual. These tests are also done during development time. The following are some examples of function testing:

- Graphical input from camera
- Corresponding OCR input and output
- Summary feedback to user

System Testing

The system will be tested as a whole during this phase of testing. The test will focus on where all modules and functions within the system cooperating with each other, and whether the system functions as a whole.

8.2 Beta Testing

The product will be tested by visually impaired students on campus of Santa Clara University for actual user feedbacks.

Chapter 9

Risk Analysis

The Risk Analysis table defines a potential set of risks that our group could face, as setbacks to timely progression towards our finished project. For each risk, there are two potential consequences, a probability value ($0 \rightarrow 1$), a severity value ($0 \rightarrow 10$), an impact value (impact=probability*severity), and two potential mitigation strategies. The risks are ordered from greatest to least impact value (top-to-bottom).

Table 9.1: Risk Table

Risk	**Consequences**	**Probability**	**Severity**	**Impact**	**Mitigation Strategies**
Illness	- Work distribution becomes uneven - Project progress is delayed	0.5	4	2	- Get proper sleep/good hygiene - Stay hydrated
Loss of code	- Significant additional work - Notable setback in project progress	0.2	8	1.6	- Careful organization - Use version control system (i.e. GitHub)
Planned implementation failed to work	- Mid-cycle design decisions/ changes required - Additional coding required	0.2	7	1.4	- Careful planning - Good communication
Schedules do not mesh	- Pace of work is potentially slower - Communication (of changes/ plans) is not as fluid/immediate	0.7	2	1.4	- Careful planning - Earnest communication
Designed product does not meet expected product	- Deduction in grade - Inter-team disappointment/ resentment	0.1	7	0.7	- Careful planning - Frequent project check-up meeting

Chapter 10

Sustainability

Sustainability is an essential factor in delivering a product that will benefit peoples lives and create value. During the design of PRAHVI, we have identified multiple points of sustainability through the lifecycle of the product: from its environmental costs, to its user interaction, and its economic viability. By evaluating these facets, we can draw a concrete triple bottom line to evaluate the effectiveness of the product in the context of an actual business model.

As a product, PRAHVI is designed primarily using off-the-shelf components that have been developed at scale both to reduce costs and to minimize its environmental footprint. Its construction consists mainly of a PCB board, a camera module, a plastic enclosure, and a cable to connect the product to the users smartphone. The board used in the current version is known as a Raspberry Pi Zero, a lightweight, low-power board designed for mobile applications. This means that PRAHVI operates using less than 100mW and can be powered entirely by a smartphones accessory port. The electrical components are manufactured in compliance with the global Restriction of the Use of certain Hazardous Substances (RoHS) regulation. Additionally, each company we have sourced components from have publicly pledged their products to be free of conflict materials and use minimal amounts of rare-earth metals. This allows us to minimize the environmental impact of both construction and disposal of the product by accounting for its individual materials and following federal and international procedures. The enclosure is constructed using a 3D printer with ABS plastic material. Although this material is not biodegradable, using a pure ABS filament and designing the product to be modular allows a recycling entity to separate the components and recycle the ABS plastic. Overall, we anticipate a single PRAHVI unit to last the life cycle of the users smartphone, typically 2-3 years. This accounts for future feature upgrades and the general durability of the product against normal wear and tear. By accounting for each of these factors during the design phase of our project, we can minimize PRAHVIs overall environmental footprint.

In terms of social sustainability, we have chosen a very clear demographic that has been largely untapped for innovation making PRAHVI a promising product for this field. PRAHVI is designed to assist users with visual disabilities navigate their visually-oriented environments, from casual reading to discovering signs and posters. The use cases it presents are context-specific but very familiar to those who struggle with these tasks on a daily basis. Because we are tailoring the interface of the product to individuals with partial or full visual disability, making the product intuitive presents a unique challenge for us. By crafting an interface that communicates primarily through sound and haptic feedback, we believe that the product will be intuitive and useful for the user. Furthermore, we hope to begin testing the product through usage exercises and potentially real-world user tests. However, creating a product for this niche also draws the factor that users will create an implicit dependence on the product. Should the user begin entering high-risk environments on their own, such as navigating a busy street, the stakes for failure could mean the difference between life or death. Therefore, for the first few iterations of the product, we would advise users only to use the product in a controlled environment with minimal hazards and many safeguards. Overall, we hope that PRAHVI can add significant value to a users daily life with the objectives we have set, using the technologies and sense of interface we have developed.

The final metric for a products viability involves the economic sustainability, particularly in a world saturated with electronic assistance devices. By utilizing off-the-shelf components and relying on a smartphone that users in this demographic typically already have, we have made strides in minimizing the cost of the product to well below current solutions. Our target cost of the product is to be less than \$200, which is a fraction of the closest competing product, OrCam, which is priced at \$2500. Much of the cost for such devices is in the software and the processing unit, as these devices are typically designed to be standalone. During the design phase of our project, we studied the demographic of individuals with visual disabilities and found that many typically own and use a smartphone on a daily basis. This means that we can safely trade off a small measure of convenience with a standalone device for the cost savings of using a processing unit users already own. Additionally, this drives down the cost and frequency of future upgrades, as the devices processing power is upgraded for free with every new smartphone a user purchases. This means less revenue is spent on developing and manufacturing new PRAHVI processing modules. As a product, most of the revenue would go to the material cost of the components and development and testing of the software. Additionally, the retail cost to the end user can typically be augmented by support from their insurance providers. In the future, PRAHVI may be manufactured entirely using a custom PCB board and custom hardware that, manufactured at scale, would further drive down costs while delivering a more integrated product. From an economic standpoint, PRAHVI is an effective product in this segment, especially compared to competing products, by using a careful mix of tradeoffs that overall benefit users.

As a product, we hope that PRAHVI meets or exceeds the triple bottom line to remain a fully-sustainable product. By identifying a key niche ripe for innovation, then sourcing parts and development in an environmentally and economically responsible way, we envision a life cycle that helps the product remain viable for many iterations. With each iteration, we also hope that the product can execute fixes and improvements that make it more useful for users and even expand its target audience. These goals overall help create the framework for a transition from a simple design project into an actual product.

In a world with increasingly limited natural resources and a larger focus on industrial impact on our environment, sustainability is a significant part of research and development. During the design of PRAHVI, we have identified processes, components, and the sourcing of these components to evaluate its environmental impact. As a product that spans the resources of multiple industries, we recognize that our products lifecycle includes many stakeholders and resources, from manufacture, to daily usage, and final disposal.

The construction of PRAHVI begins with its components, their sourcing, and overall assembly. The primary component of the device is its printed circuit board (PCB). For research purposes, we have used an off-the-shelf board known as the Raspberry Pi Zero. This board consists of plastic for the board itself, laser-etched copper traces, and components with varying amounts of copper, silicon, and gold. In compliance with the global Restriction of the Use of Certain Hazardous Substances (RoHS) regulation, the Raspberry Pi is manufactured without the use of conflict materials and minimal use of rare-earth elements. In addition, the camera module manufactured by Sony Inc. is manufactured under stricter regulations that replace many materials, such as those that go into the imaging sensor, with more environmentally-friendly, albeit slightly more expensive alternatives. We found that the small form factor of the Raspberry Pi not only makes the product more portable, but holistically uses fewer materials to manufacture, further driven down by the large scale of the Raspberry Pi, while still meeting our quality and performance requirements. Although other boards and modules were evaluated, most are manufactured under small-scale operations that use more resources or did not meet our requirements. The case of the product is manufactured using a 3D printer with ABS plastic material. This material is not biodegradable, and must be recycled at the end of the products lifecycle. During the design phase of this project, we selected what we believe are the optimal components and materials for PRAHVI from a performance and environmental standpoint.

As a holistic product, PRAHVI introduces challenges to managing energy consumption from manufacture, to delivery, and daily use. The parts used in PRAHVI are sourced primarily from China. With careful design and planning, we consolidated our parts orders into three stages and from a single supplier to ensure that we minimized the impact of transportation in the product. The case, which is manufactured using a MakerBot Replicator 2X, is the only part we manufacture ourselves. For research purposes, using a 3D printer significantly reduces the energy and resource overhead of a professional manufacturer, while providing a representative component of the final product. During use, we anticipate that PRAHVI will be powered entirely using a smartphone device, removing the need for a separate battery. Its small form-factor and ARM processor allows PRAHVI to operate with minimal power use. We further these energy savings by defining clear contexts in which PRAHVI is in a passive sleep mode and when it is in an active scanning mode. These practices combined help to minimize the overall energy footprint of PRAHVI as a product.

Finally, although PRAHVI is designed with longevity of the product in-mind, we incorporated its transition out into the design process. PRAHVI is made with standard components, each of which can be easily replaced. We anticipate that PRAHVI can be used throughout the standard lifecycle of a smartphone, around two years. This accounts for component failures, the likelihood of damage resulting in a system failure, and required feature upgrades for new versions of the smartphones software. At the end of the devices lifecycle, the components of the Raspberry Pi Zero can be easily extracted and recycled through standard protocols. In addition, the case is entirely constructed from ABS plastic that can also be recycled and repurposed. These considerations help ensure that PRAHVI is built to last with the users needs as well as transition out of use in a sustainable manner.

Chapter 11

Development Timeline

The Development Timeline presents a general outline, in Gantt chart form, of when various project steps will be completed and by which project member(s). Steps are divided into three sections: Requirements, Design, and Implementation. A legend provides a reference for the utilized color-coding.

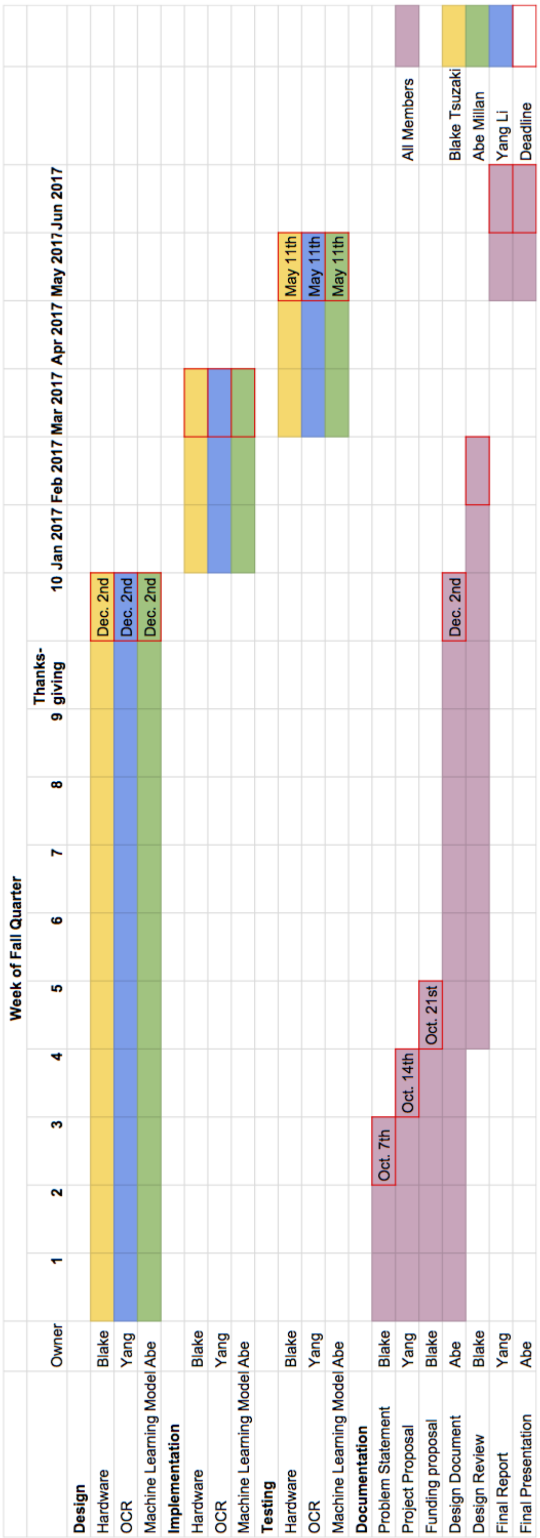

Figure 11.1: Use Cases