DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

PROJECT CHARTER
CSE 4316: SENIOR DESIGN I
FALL 2020

ALS EYE TRACKING GROUP
ALS EYE TRACKING APP

ANTHONY VARDARO
MICHAEL KOSTA
ZIXIU SU
THANHTHAO LE

Revision History

Revision	Date	Author(s)	Description
0.1	10.04.2020	AV, MK, KL, ZS	document creation

Contents

List of Figures

1 Problem Statement

There are many different kinds of diseases that affect people all over the world, some of which are incurable. While not all diseases make life impossible for the people who are affected, diseases such as ALS, Muscular dystrophy, and Myasthenia gravis; which cause a deterioration of muscle tissue, can make life become nearly impossible. The people affected by these types of diseases can have a difficult time communicating with anyone. The longer they have the disease, the harder it gets; eventually only being able to move your eyes. That is where the ALS Eye Tracking APP will be of use. While it might not be perfect, the people who use this APP will have a much easier time communicating with others.

2 Methodology

We will be building an application to go along with eye tracking technology, to help give a means of communication to people who cannot communicate. With the deterioration of muscle tissue caused by ALS, the only thing a person will eventually be able to move is their eyes. So, if they can only use their eye, it makes sense to use eye tracking technology. This technology has already been developed, but it is being primarily used for gaming. We are going to develop an app to go along with the already created hardware, that will make it easier for people with ALS to communicate.

3 Value Proposition

In the United States there are statistically 12,000 to 15,000 people who suffer from ALS. These are all people who either cannot, or eventually will not possess any way of communication and who would all benefit from this application. Not only this, but there are other types of people who could benefit from this as well. The number of people who suffer from limb loss is around 2.1 million. Those of them who do not have any arms may need to find a new way to type. Developing this application will not only help people with ALS, but also give people without arms more of a choice as to what type of electronic typing they use, whether it be verbal or visual.

4 Development Milestones

Here is a list of all the different documents, project features, and deadlines.

- Project Charter first draft, October 5, 2020
- Prepare for Second Sprint presentation, October 7, 2020
- Second Sprint plan presentation, October 9, 2020
- Demonstration of Tobii Eye Tracker Functionality, October 16, 2020
- Continuing Research on Predictive Text, October 23, 2020
- System Requirements Specification, October 26, 2020
- Third Sprint plan presentation, October 30, 2020
- Gather resources and research on GUIs, November 6, 2020
- Architectural Design Specification, November 16, 2020
- Fourth Sprint plan presentation, November 20, 2020

- Complete Peer Review, December 4, 2020
- Detailed Design Specification, December/January 2020
- CoE Innovation Day poster presentation, Spring 2021
- Demonstration of implementing a Dasher like program into a GUI, February 2021
- Demonstration of Predictive Text Algorithms, March 2021
- Demonstration of Test Code, April 2021
- Final Touch Ups, April/May 2021
- Final Project Demonstration, May 2021

5 Background

What is ALS, or amyotrophic lateral sclerosis, is a neurodegenerative disease that affects the nerve cells in the brain and spine. This disease can make it difficult for someone to communicate with other people, especially if they can only move their eyes. This is where the eye tracking software comes into play. While eye tracking is not a new technological achievement, it has become much more advanced in recent years. Eye tracking was originally invented in 1908 by Edmund Huey. Even though the current eye tracking software interface for the software is straight-forward, the interface may cause the user to have a hard time creating words, such as staring at the screen for too long. This can create a problem for the user since the user may have the need to blink or their eyes might get tired. Another problem is having their eyes stare at only a portion of the screen for a long time, mainly towards the right side. The eyes may need to take a break from staring at certain parts of the screen. A different version should be created to help the user have an easier time communicating to others while just using their eyes. Another product we will be getting ideas from, for our application, is dasher. Dasher is an input method and computer accessibility tool created in 2010. While this is a useful tool and a great reference point, there is still a lot of things that need to be fixed. Having a more stable screen, one that does not move with the cursor. Finally, we have spelling correctors. These were first created in 1961, by Les Earnest. At the time, they only held about 10,000 words. Eventually, they were adapted in the 1980s to be used on please personal computers. Now, you can find different websites and different hardware tools that can replicate the same effect. Even though using these will help in the creation of a predictive text algorithm, this alone will not be enough. We will still need to add enough code and have a better understanding of words, so that we can display the correct letter combinations. The existing relationship between the development team and the sponsor is students and instructor.

6 Related Work

While using eye tracking to help people with ALS is a fairly new concept, we are not the only ones who are attempting to work with or work to solve this problem. To start, lets look at some similar technologies. The first of these is the program which started the idea, Dasher. *Dasher is an information-efficient text-entry interface, driven by natural continuous pointing gestures* [2]. Dasher is a commercially available product, and while it performs similar to how we want our app to perform, it is not created to be used with an eye tracker, rather it was built to be used with a mouse and keyboard. Dasher also implements a predictive text algorithm, which is the next programming idea to look at. While there are not a lot of examples of predictive text algorithms found online, there is an abundance of similar technologies found called a spelling checker. A spelling checker is a commercially available product that enables you to correct any spelling mistake [3]. While correcting misspelled words and predicting the next letter of a word are similar, there is still a major difference between them. Finally, the last technology would be the eye trackers made by other companies. One company, called Gazepoint, has created their commercially available version of an eye tracker. They talk about how there is a general set of metrics used to measure eye tracking; Direction of gaze, number of fixations, time to first fixation, blink rate, blink duration, and pupil diameter [4]. While these are the standard for eye tracking, Tobii is still rated the top company for eye tracking technology. Next, lets take a look at different companies researching ways to help people with ALS communicate. An article from ALSNewsToday talks about how the implementation of eye tracking could really help ALS patients [5]. While they do mention multiple limiting factors to eye tracking, this article mainly seems to be just academic research. They also mention using tablets or touchpads, while they seem to understand the usefulness of eye tracking technology, they are greatly limited on their understanding of how to implement this technology. Finally, the last of the related companies is the Bridging APPs. *BridgingApps provides the access, education, and resources needed to effectively use mobile devices (smartphones and tablets) to help people with disabilities* *communicate, exceed educational goals, and reach their fullest potential* [1]. While using tablets can be useful for people who were just diagnosed with ALS, this would not be useful in patients who have had ALS for a long period of time. This cannot help patients who have completely lost the ability to move. While a lot of these products work well separately, they do not cover all the necessary requirements that are needed to make this product a viable solution.

7 System Overview

Our system is composed of a user interface, web api, and database.

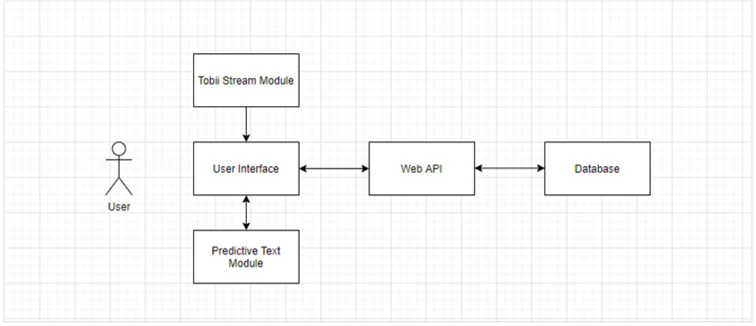
User Interface
The user interface (UI) is responsible for displaying the onscreen keyboard and collecting gaze information from the Tobii eye tracking device. The UI contains onscreen components that the user will interact with to maneuver the eye tracking software.

Keyboard
Visual display of the keyboard, and what the user looks at to communicate input. The keyboard emulates keypress events when the user gazes at individual keys along the keyboard.

Gaze Event Listener
This is a continuous real-time module that regularly gathers gaze input from the eye tracker and delegates that information to the Keyboard component. This module continuously gathers gaze data from the Tobii SDK in the form of a time-series and associates the data to positions on the screen.

Web API
This is simply a RESTful web api that collects user information and preferences and writes them to the corresponding database. This empowers the users to create account and manage it.

Database
Component for storing user info and other metadata.



The arrows indicate what directions data flows throughout the system. When a user traces their eye across the screen (signaling input), this data will be captured by the Tobii and sent to the user interface. The user interface will use the Predictive Text Module to estimate the users input given positional coordinates from the input stream and historical input. This exchange between the Tobii, UI, and Predictive Text Module will repeat each time the user confirms a letter input when spelling a word.

8 Roles & Responsibilities

Stakeholder: Christopher McMurrough, an instructor for Senior Design in Fall 2020.

Point of Contact: Professor McMurrough is also our contact for this project as well. He will be the one we contact to try and get guidance on the direction we are going, as well as, if we need to reach out to any costumers or test candidates.

The team members in this project are Steven Michael Kosta, Thanhthao Kelly Le, Zixiu Su, and Anthony Vardaro.

We will be switching roles between each of the separate scrums. These roles are product owners, scrum master, and research head. We have decided to have two product owners, these will be two team members who will have and use the two Tobii 5s, they will manage testing of the products and writing of the codes. We will also have a scrum master, who will maintain the scrum schedule and make sure the team is on track. Finally, we will have a research head, they will be the one who manages the separation and congregation of the research material between team members. We have decided to switch these roles up every scrum, so that if someone is unhappy with their role, they will not be in the same one forever; plus, this means we will all get equal turns and experience.

Product Owners: Anthony Vardaro and Zixiu Su

Scrum Master: Thanhthao Kelly Le

Research Head: Steven Michael Kosta

9 Cost Proposal

Since this project mainly focuses on software development, most expenses for this project will be used on Eye-Tracking hardware, such as Tobii Eye Tracker 5. There may be expenses on software products as well.

9.1 Preliminary Budget

Hardware component - $500 Software component - $200

9.2 Current & Pending Support

Current Support: - UTA CSE Department ($800)
Pending Support: None

10 Facilities & Equipment

This project requires at least one of the latest eye-tracking hardware and can be bought through amazon. The eye-tracker can be found on its official website as well. The main eye-tracking equipment that will be used is Tobii Eye Tracker 5. Due to limited budget, 2 Tobiis will be used instead of 4. Another equipment is some computers to code and execute the developing program with the Tobiis. This project can be worked at home for testing code; however, due to limited equipment, a lab space may be required for a testing area. We could also use a viable test candidate to help us to understand what constraints we need to work on and what possible settings we could add.

- Two Tobii Eye Tracker 5s

- One or Two Computers Used for Testing the Code
- A Space in the Labs to do Group Work
- Home Space for Research and Code Work
- Test Candidate

11 Assumptions

Critical assumptions related to the project:

- We can integrate Tobii Eye input with the User Interface.
  The success of this project is contingent on our success integrating the user experience and the Tobii Eye Tracker. The final product needs to be able to accept input via eye tracking movement and accurately relate that information to keys on a virtual keyboard. The final product will be unusable if this feature does not function properly.
- The Tobii Eye Tracker accurately captures input
  The Tobii Eye Tracker product is foundational to the success of our product. However, if the Tobii is unable to provide eye input to the quality that we require for our purpose, we will be forced to explore new options in capturing input. This can set the project timeline back.
- The Users hardware and Operating System is compatible with our product
  It is possible that the user may be using a device that does not support the Tobii device. The Tobii connects via USB however some devices (such as Apple laptops) do not include this port, and the user will have to accommodate this shortcoming with an adapter.
- The Predictive Text Model is helpful and does not restrain the user.
  We intend to add weightings to individual letters to estimate what the user intends to write next. This comes with the risk of selecting the incorrect letter and frustrating the user. We intend to use an existing library/model for this component, but the assumption prevails that this needs to be an improvement and not a disservice.
- The user has proper connectivity with the Internet
  It is critical that the user can connect to the Internet to interact with the external Web API. If they cannot, it may disconnect the user or the user has to reenter the word or phrase.

12 Constraints

Limited Constraints related to the project:

- Everyones schedule may overlap due to attending other classes and/or working at different jobs.
- Due to COVID, most interactions are met online. This event can reduce in-person interaction, especially when we need to test a program with limited equipment.
- Some equipment may not be working properly.
- Some equipment may be delayed.
- Communications may be delayed due to network issues or busy at the moment.

Risk description	**Probability**	**Loss (days)**	**Exposure (days)**
Delay when writing and trying to run the predictive text algorithm	0.40	15	6
Lab space is not available	0.30	7	2.8
Test candidate is not available	0.70	20	14
Unable to contact the Stakeholder	0.10	7	0.7
Work on the GUI has been delayed due to over complexity	0.50	14	7
Unable to meet due to personal problems	0.75	5	3.75

Table 1: Overview of highest exposure project risks

13 Risks

These are the possible risks we could think of that could affect our project.

14 Documentation & Reporting

14.1 Major Documentation Deliverables

14.1.1 Project Charter

Describe how this document will be maintained and updated (how often, under what circumstances, etc.). When will the initial version be delivered? When will the final version be delivered?

14.1.2 System Requirements Specification

Describe how this document will be maintained and updated (how often, under what circumstances, etc.). When will the initial version be delivered? When will the final version be delivered?

14.1.3 Architectural Design Specification

Describe how this document will be maintained and updated (how often, under what circumstances, etc.). When will the initial version be delivered? When will the final version be delivered?

14.1.4 Detailed Design Specification

Describe how this document will be maintained and updated (how often, under what circumstances, etc.). When will the initial version be delivered? When will the final version be delivered?

14.2 Recurring Sprint Items

The following items will be documented and maintained during each individual sprint. As above, remove this paragraph from your draft, but leave the heading.

14.2.1 Product Backlog

How will items be added to the product backlog from the SRS? How will these items be prioritized? Who makes the decision (product owner, group vote, etc.)? What software will be used to maintain and share the product backlog with team members and stakeholders?

14.2.2 Sprint Planning

How will each sprint plan be planned? How many sprints will there be (you need to look at the schedules for this course and previous Senior Design II courses during the appropriate semesters to figure this out).

14.2.3 Sprint Goal

Who decides the sprint goal? How will you involve your customer in this process?

14.2.4 Sprint Backlog

Who decides which product backlog items make their way into the sprint backlog? How will the backlog be maintained (collaboration software, a "scrum board", etc.)?

14.2.5 Task Breakdown

How will individual tasks be assigned from the sprint backlog? Will it be up to each team member to voluntarily claim a task, or will it come from the product owner? How will time spent on tasks be documented?

14.2.6 Sprint Burn Down Charts

Who will be responsible for generating the burn down charts for each sprint? How will they be able to access the total amount of effort expended by each individual team member? What format will the burn down chart use (include an example burn down chart below).

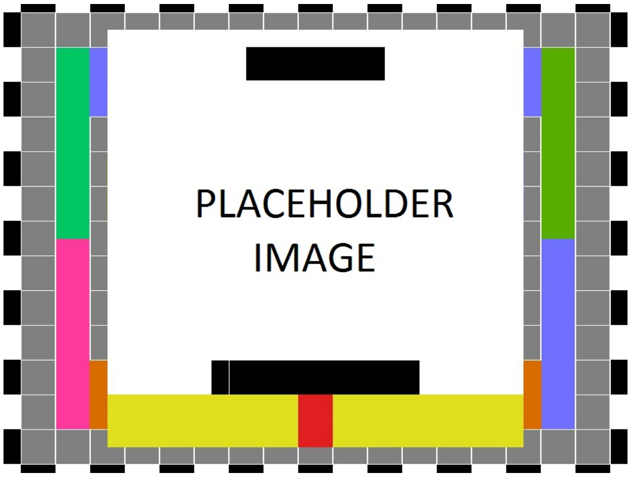


Figure 1: Example sprint burn down chart

14.2.7 Sprint Retrospective

How will the sprint retrospective be handled as a team? When will this discussion happen after each sprint? What will be documented as a group and as individuals, and when will it be due?

14.2.8 Individual Status Reports

What sort of status will be reported by each individual member, and how often will it be reported? What key items will be contained in the report?

14.2.9 Engineering Notebooks

How often will the engineering notebook be updated, at a minimum, by each team member? What is the minimum amount of pages that will be completed for each interval, and how long will that interval be? How will the team keep each member accountable? Who will sign of as a "witness" for each ENB page?

14.3 Closeout Materials

The following materials, in addition to major documentation deliverables, will be provided to the customer upon project closeout. Remove this paragraph from your draft, but leave the heading.

14.3.1 System Prototype

What will be included in the final system prototype? How and when will this be demonstrated? Will there be a Prototype Acceptance Test (PAT) with your customer? Will anything be demonstrated off-site? If so, will there be a Field Acceptance Test (FAT)?

14.3.2 Project Poster

What will be included on the poster, what will be the final dimensions, and when will it be delivered?

14.3.3 Web Page

What will be included on the project web page? Will it be accessible to the public? When will this be delivered? Will it be updated throughout the project, or just provided at closeout (at a minimum, you need to provide a simple web page at the end).

14.3.4 Demo Video

What will be shown in the demo video(s)? Will you include a B-reel footage for future video cuts? Approximately how long will the video(s) be, and what topics will be covered?

14.3.5 Source Code

How will your source code be maintained? What version control system will you adopt? Will source code be provided to the customer, or binaries only? If source code is provided, how will it be turned over to the customer? Will the project be open sourced to the general public? If so, what are the license terms (GNU, GPL, MIT, etc.). Where will the license terms be listed (in each source file, in a single readme file, etc.).

14.3.6 Source Code Documentation

What documentation standards will be employed? Will you use tools to generate the documentation (Doxygen, Javadocs, etc.). In what format will the final documentation be provided (PDF, browsable HTML, etc.)?

14.3.7 Hardware Schematics

Will you be creating printed circuit boards (PCBs) or wiring components together? If so, list each applicable schematic and what sort of data it will contain (PCB layout, wiring diagram, etc.). If your project is purely software, omit this section.

14.3.8 CAD files

Will the project involve any mechanical design, such as 3D printed or laser-cut parts? If so, what software will you use to generate the files and what file formats will you provide in your closeout materials (STL, STEP, OBJ, etc.). If your project is purely software, omit this section.

14.3.9 Installation Scripts

How will the customer deploy software to new installations? Will you provide installation scripts, install programs, or any other tools to improve the process? Will there be multiple scripts provided (perhaps separate scripts for the graphical front end and back end server software)?

14.3.10 User Manual

Will you customer need a printed or digital user manual? Will they need a setup video? Decide now what will be provided and discuss.

References

[1] Bridgingapps - bridging the gap between technology and people with disabilities.

[2] Dasher basics.

[3] Spell checker: check grammar and spelling for english texts.

[4] Eye tracking - learn more about eye tracker technology, Aug 2019.

[5] PhD Jose Marques Lopes. High-tech devices improve quality of life, should be standard care..., Aug 2018.