Motor Impairment Keyboard

Project Proposal

MIKey

Jackson Hagood

Andrew Imwalle

Brittany Jenkins

Connie Liu

Abhishek More

Max Smith

Department of Computer Science

Texas A&M University

9/12/2022

Table of Contents

1 Executive summary 3
2 Introduction 3
2.1 Needs statement 4
2.2 Goal and objectives 4
2.3 Design constraints and feasibility 4
3 Literature and technical survey 4
4 Proposed work 6
4.1 Evaluation of alternative solutions 6
4.2 Design specifications 6
4.3 Approach for design validation 9
5 Engineering standards 10
5.1 Project management 10
5.2 Schedule of tasks, Pert and Gantt charts 11
5.3 Economic analysis 14
5.4 Societal, safety and environmental analysis 14
5.5 Itemized budget 14
6 References 15
7 Appendices 15
7.1 Product datasheets 15
7.2 Bios and CVs 15

1 Executive summary

A collection of devices has been designed to allow those with disabilities to use computers uninhibited. This has allowed them to have careers in computing and provide them with newfound abilities in communication and independence. Considerable resources have gone into the development of these devices, but there is still room to improve. Those with motor function impairments, for example, are left without a device designed specifically for their needs. They have been forced to choose between options to radical for their needs and keyboards that do not provide any assistance.

To help solve this problem, MIKey, a team of six Senior Computer Engineering students at Texas A&M University, seek to develop a Motor Impairment keyboard. MIKey's goal is to design a device to allow those with motor function impairments, such as Parkinson's, to more easily and efficiently interact with their computer with a physical keyboard. This keyboard will need to be standard enough such that anyone used to a normal QWERTY keyboard can easily make the transition. For the product to be viable. The keyboard must be cost-effective, portable, and sufficiently customizable. Satisfying these goals should mean the device will fill a whole in the market and improve the lives of those with motor function impairments.

Existing solutions to related problems were analyzed and several potential solutions were proposed to fulfill these needs. MIKey has narrowed in on a design that can meet these objectives and solve this problem. This Motor Impairment Keyboard (MIK) is centered around two key features. First, the keyboard will be oversized with large keycaps. This means that users will have less difficulty getting to the precise location to press the key. The keyboard will also be designed such that the individual keycaps are sunken into the housing to allow for better accuracy when typing. The second feature of the MIK is the auto-complete functionality. Three LCD's present on the top of the keyboard will display auto-complete options dependent on what the user has typed to that point. This necessitates the need for a micro-controller present in the keyboard.

To accomplish this implementation, development is split into three main categories: software, hardware, and modeling. For the software, hardware signals will be mapped to key representations. From there, those signals will be sent through USB to the computer. If an alphabetic key was pressed, the auto-complete algorithm will be run with the current buffer. This algorithm will incorporate a Trie data structure to allow for efficient determination of word candidates. The hardware design is centered around PCB to map the key switches to a binary signal for the micro-controller (a Raspberry Pi 0). Auto-complete options output from the micro-controller will be linked to three LCD's on the top of the keyboard. Finally, modeling will be done for the keyboard's housing and keycaps. Time will be needed to print these materials in PLA. Altogether, the cost of such a prototype will not exceed $200, with the aim for a mass-produced device to be sold under $100.

Assuming a solution, the result should be a device to allow those with motor function impairments better control in their interactions with a computer. The benefit of this design is that those without motor function impairments can also benefit from a more capable device. Furthermore, with the device designed to be modifiable, versions could be conceived for other languages and specific disability needs. Such a design is best verified with real users, so MIKey is coordinating with outside organizations.

2 Introduction

The world is in a digital age where usage and navigation of computers has become increasingly important for the population. In 2018, the American Community Survey (ACS) found that 92 percent of households had a computer, and this number has been trending upwards [4]. Unfortunately, not everyone is able to efficiently use computers, as they are not designed for those with physical disabilities.

2.1 Needs statement

Those with tremors or motor function impairments (such as Parkinson's) experience trouble when interacting with traditional computer peripherals (namely a keyboard). Therefore, as use of technology becomes more prevalent, there exists a need for a simple assistive technology that provides better computer hardware accessibility to individuals with limited motor function. Existing assistive technologies are often too restrictive for those with mild to moderate motor function impairments (such as eye-trackers) and tend to sacrifice efficiency for the user. As a result, there is a need for a more traditional form of interaction that is easier for those with mild impairments to utilize. There also exists a need for a more affordable solution to this problem, as assistive devices can cost upwards of thousands of dollars, making them unattainable for a large section of the population.

2.2 Goal and objectives

The goal of a potential solution would be to make the typing experience easier for those who have motor function impairments. The solution should be cost-effective, accessible, and should also improve the experience of interacting with their computer (namely typing or otherwise inputting text). Priority must be placed on ensuring the solution is easy to learn, as a large percentage of those with motor function impairments are typically older in age and likely to be less proficient with computers. Therefore, avoiding the need to install device drivers or other software is preferable. The solution should also be generic enough to be customizable depending on the specific needs of each user (like variable actuation force or form factor). The design should not compromise any of the abilities present in a normal keyboard. Ideally, a solution would add to a normal keyboard's capabilities. The aim of these capabilities should be to reduce the user's opportunity to make mistakes while typing and improve their typing speed. Another important objective is to keep the solution as cost effective as possible in order to keep the product in reach of the general user. A final objective is to follow a "helping the disabled helps all" philosophy. Many devices for those with disabilities end up helping others in unintended ways. A solution to this problem could end up having uses for a wide variety of circumstances, even for those without motor function impairments.

2.3 Design constraints and feasibility

Several constraints presented in the needs statement involve the price, functionality, and compatibility of the product. In terms of price, we aim to deliver a final product that does not exceed $200, though development of a prototype will likely exceed this amount. Additionally, users will ideally be able to set up and take advantage of core functionalities of the product on their own. We also want the user to be able to use this product with any computer in a variety of settings, so it must be reasonably portable and compatible across Linux, Windows, and Mac. To do this, our solution will be based on existing technologies with the form factor of a standard USB-A, QWERTY American-English keyboard. Lastly, our development time is constrained at 10 weeks. We have determined that a majority of these constraints are feasible given our technical expertise, but our biggest challenges will likely be the final cost of the product and limitations on development time.

3 Literature and technical survey

When analyzing the market for existing relevant technologies, three main categories are present. First, alternative forms of computer interaction are common for those with more severe motor function impairments (such as an eye tracker or a mouse stick). While these are not completely in line with the intentions of the proposed design, they are useful for considering accessibility features. More in-line with the project's goals are more tradition keyboards. These include oversized keyboards, or sunken key keyboards. The last type of relavant technology is existing auto-complete and auto-correct software which could potentially be leveraged to better a potential solution.

Those with sever motor function impairments often use devices that fall under alternative forms of computer interaction. One of the most well-known of these is an eye-tracking device. This device allows users that cannot interact directly with their hands to select on-screen items using their eyes. This device is often used by those with ALS, muscular dystrophy, or spinal cord injuries [5]. Another device under this category is a mouth stick. This device is placed into the user's mouth where they use it to blow onto on-screen items. Again, this device is intended for those who do not have use of their hands. Other devices in this category follow the theme of finding alternative avenues for users to interact with their screen, most of which make use of an on-screen keyboard. While these items do not fulfill all the objectives specified in Goals and Objectives, they do provide some insight into how devices are developed for those with disabilities. All these devices, for example, feature a relatively high level of customizability. The mouth stick, for example, can come in a plethora of shapes and sizes depending on the user's needs, and some eye trackers have modes for stationary and mobile heads. It can be surmised that when developing devices for those with motor disabilities, it is important to keep the solution customizable and expandable as all users have a specific set of circumstances.

Many with more mild motor impairments use more traditional keyboard peripherals. One of these devices is what is known as an oversized keyboard. The largest manufacturer of this device is BigKeys with nearly two decades in the market. These keyboards feature large keycaps and a wide form factor that makes it easier for the user to avoid selecting the wrong key. The users of these keyboards include children and those with motor function impairments. By grouping these categories together, however, many with motor function impairments may feel put-off by the child-oriented design. Furthermore, the keyboard is "bare-bones" (no auto-completion) but costs over $100 [1]. With a comparable, standard-sized keyboard costing as little as $10, this device feels lacking. Another type of more traditional keyboard is a sunken key keyboard. In this design, the keys are sunken into a casing with guards between them. This makes it so users who often press neighboring keys on accident can no longer press the wrong key. This fits in well for those with Parkinson's disease. The largest issue with this keyboard, however, is its availability. Only one model is easily found online, and it costs greater than $200 [7]. Both keyboards seem like substantial solutions to the Needs Statement, but their price and availability place them out of reach for many with motor-function impairments.

The last type of relevant technology is auto-complete software. For years, auto complete has been quite universally used on smartphones to allow users to type faster. Auto-complete solves two major problems with typing on smartphones. First, the difficulty of pressing the correct key on a small keyboard is alivieted by having to type less when suggestions are chosen instead of typing the entirety of the word. The second problem is improving the speed at which users type on their smartphones, with most typing far slower on their phone than a full-sized physical keyboard. While auto-complete was not designed for those with motor function impairments, its benefits could be leveraged for these users. In the same way the average smartphone user has difficulty selecting the correct key on their phone's keyboard, many with motor function impairments have trouble typing the intended key on their physical keyboard. Furthermore, the speed improvements provided by auto-complete on a smartphone could potentially be applied to a physical keyboard experience. Therefore, it stands to reason that auto-complete could greatly assist those with motor function impairments to type on a physical keyboard.

Looking at these three categories of devices, three main lessons can be learned. First, devices designed for those with disabilities should be customizable to fit the needs of each specific user. It is therefore best to strive for a design that can be deconstructed or configured without destroying it. Second, existing keyboards for those with motor function impairments are too expensive for a lot of people. A useful design should therefore be relatively affordable for the average person. Finally, auto-complete has significant potential to improve the typing experience of those with motor-function impairments. The design proposed in this document will be less expensive, more customizable, and more advanced than existing alternative keyboards.

4 Proposed work

4.1 Evaluation of alternative solutions

One solution to the problem outlined in the Needs Statement is simply to use a large touch-screen display. This device obviously already exists and may be a good fit for some users. The large screen can enable the user to avoid typing the wrong key, and on-screen auto-complete is a feature in many such devices. This device has two problems, however. First is the price, with many touch-screen computers or displays costing thousands of dollars. The second problem is the lack of tactile response. Many with motor function impairments may benefit from tactile feedback that their keystroke has registered.

Another alternative solution to our problem is more theoretical and does not exist at the moment. This solution is essentially the same as the touchscreen solution proposed earlier, but fixes the missing feedback issue by placing multiple haptic motors underneath the screen. In this way, the keys that show up on the screen can be wholly configurable to the user's preferences while also providing the user with a better typing experience with haptic feedback. However, one drawback to this design is the cost, as well as the limitation of our team's technical experience.

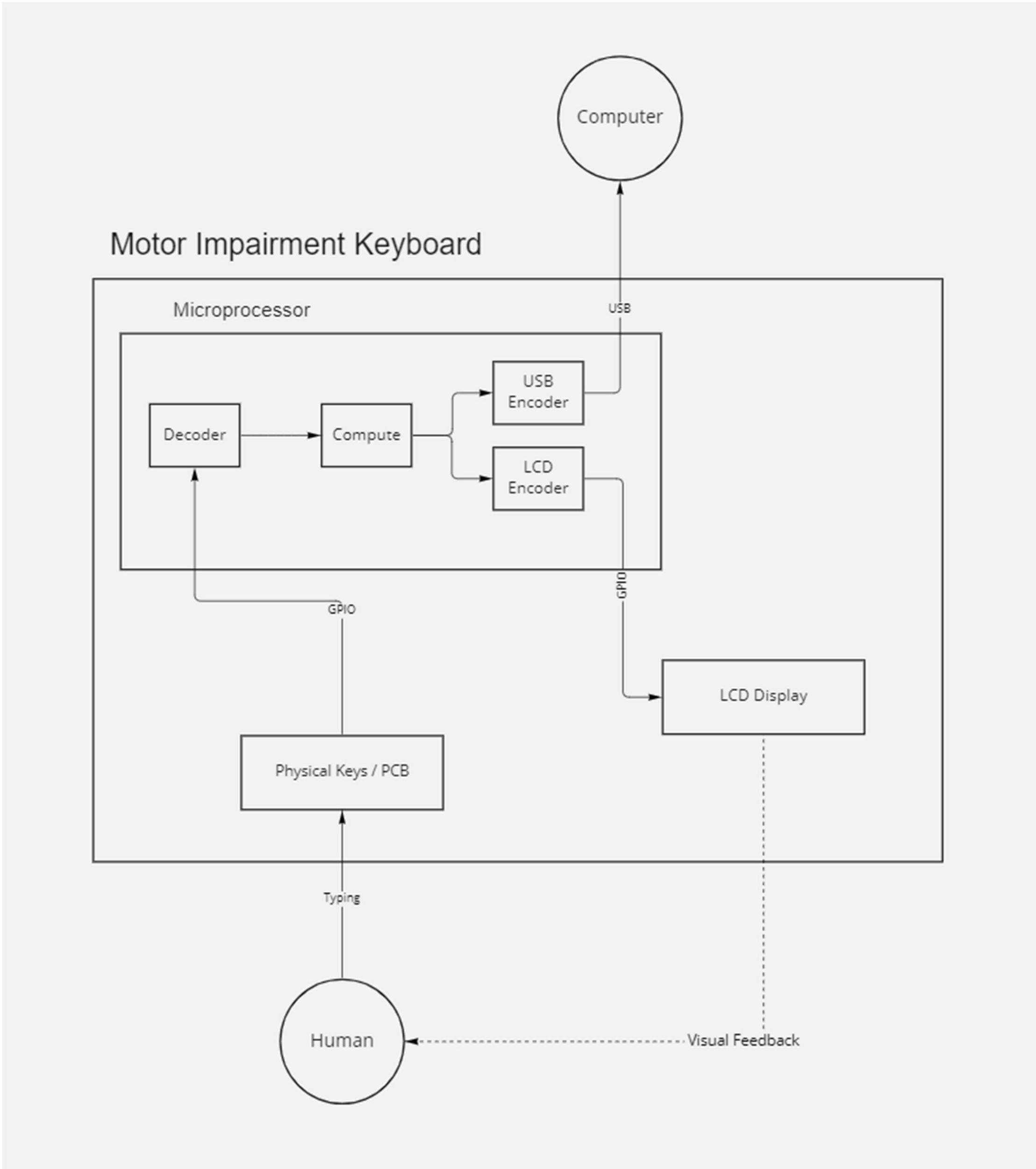
An additional solution is using an eye-tracking device alongside an on-screen keyboard, but also providing some auto-complete or text suggestion software. This would increase the user's typing efficiency compared to simply using an onscreen keyboard. The downside of this solution is the high barrier to entry. Eye trackers recommended for use with the Windows Eye Control software are sold at a minimum of $250 [2], which exceeds the budget of our audience. Additionally, the user's typing speed will still be slower compared to some of the other solutions.

An alternative option that was considered was incorporating speech to text as a feature on a physical keyboard. While this could be a good stretch goal for our project, it would not be a more efficient solution for users with some function in their hands. As described above, we are aiming to help improved computer interaction for people with mild to moderate motor function impairments. Speech to text is not a feasible solution for our target users, and we are looking to solve an issue where we see a gap in the market.

Another means of reducing the amount of typing that a motor-impaired user must do to complete a given task is to allow for customizable keyboard macros to be set. To the user, this means that a single button press can be set up to complete a complex task, such as opening a browser and navigating to a specific website. If used for multiple frequently-executed tasks, this feature has the capacity to greatly reduce the amount of typing and mouse navigation required for a user to carry out regular activities on a computer. The downside to this solution is that it would require an additional software component that users may have difficulties both installing and navigating.

4.2 Design specifications

Below is a high-level block diagram of the system prototype. The Motor Impairment Keyboard (MIK) is surrounded by an input and output. To start off the input stage, the user will type their desired text on the keyboard. The MIK then outputs through a USB signal to a computer. The three main components of the MIK are the physical keyboard/PCB, the microprocessor, and the LCD display.



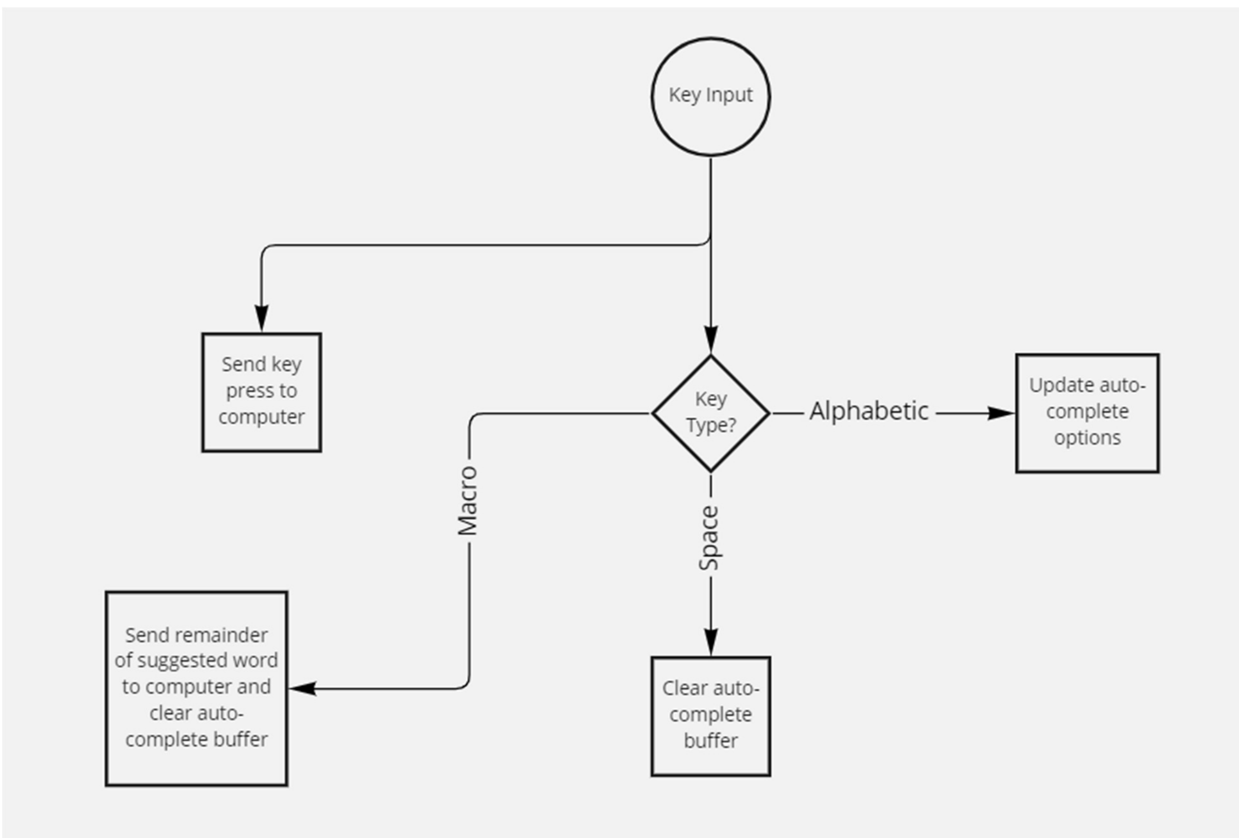
Hardware

The first part of the MIK is the physical keyboard. The keyboard will be in a QWERTY grid layout (as opposed to staggered in a standard keyboard). The keyboard will be a custom layout without a numpad but with a function bar (closest to a 75% keyboard). The keyboard and individual keys will be oversized to assist with accuracy. The keys will be sunken into the keyboard housing, similar to a sunken keyboard, also to assist with accuracy and speed. The keys will be mechanical to allow for a greater level of customization (users can choose a specific switch type, add O-rings, etc.). The keycaps will be a slim design as to not make the keyboard too tall, which is an aesthetic as well as functional decision. Finally, three special buttons will be placed underneath the space bar where the user's thumbs will likely rest so the user can select auto-complete options. Key presses will be read in by the microprocessor's general-purpose input pins.

The LCD display will be built into the keyboard and will receive GPIO signal input from the microprocessor. It will display up to three suggestions that that the user can select. As the user continues to type, the LCD will update. The font will be large and high contrast to ensure that the user can read the suggestions clearly.

3D printing and modeling will be central to the completion of the MIK. The keyboard housing and keycaps will both be designed in CAD software and printed using PLA on low-fidelity 3D printers. Using PLA as filament will maintain a strong form factor while ensuring a high-quality print. The PCB is another component that will have to be custom made to account for the over-sized keycaps. Other components will be bought off the shelf (like the Raspberry Pi and mechanical switches). The MIK will be a contained single device where the housing will hold all components. There will be a balance between the size of the keyboard to ensure ease of typing and to ensure portability.

Software



When the signal for a key press is received, the microprocessor decodes the input based on the combination of signals it received from the physical keys. The key press is then passed to the computer. The microprocessor then runs the compute algorithm required to calculate text suggestions and encodes the output in order to send data to the LCD display and, ultimately, the user's computer. The microprocessor will receive input from the keys via the GPIO pins and will decode the digital signal input to the proper key mapping.

The compute engine receives text input and determines the best suggestion for the next word. This requires the microprocessor to contain a dictionary of the English language, which takes up 14 MB of storage space. At runtime, this module will build a Trie data structure with every word in the dictionary, which allows for amortized constant time text suggestion. Once the processing is complete, the compute engine will encode the top three suggestions into a data format which is readable by the LCD display. Users can select from these word suggestions by pressing one of three special keys on the keyboard. Whenever a macro key press is detected, the suggested word will be sent to the USB encoder as a stream of key presses. This encoder will allow for the compute engine to send key stroke data, via USB, to the computer.

Looking at the software approach for solving the auto-complete problem, C++ seems like a good fit. In order to keep track of Trie data structures, object-oriented programming is a necessary feature of our chosen language. At the same time, speed is vital for the usability of the product, so a language like Python will not work. Other candidate languages, such as Java, are fast, but its memory usage may exceed the resources of the system. C++ is both fast and memory efficient while still providing us with object-oriented programming capabilities. Furthermore, it is a common choice in embedded systems thanks to its proximity to the hardware. Another reason for choosing C++ is the team's past experience with the language.

When programming in C++, best practices will be exercised in order to conserve all possible time and memory resources. For example, while tree-like data structures lend themselves well to recursive BFS/DFS algorithms, recursion taxes the device's memory to a scale that we do not want for our product. Iterative approaches, such as using an explicit stack, will therefore be taken to implement all functions. Other techniques, like excessive object-oriented programming and unnecessary use of libraries, will be avoided for similar reasons.

Summary

The features that have been decided on for the MIK are the most feasible options for supporting low cost, portability, and ease of use. We also considered our time constraints for development of our solution. By 3D printing our parts and designing our PCB we will be saving time and money on the prototype. By choosing an embedded system, it will increase ease of use by supporting plug-and-play functionality for the basic MIK. The algorithm languages, structures, and styles have been considered based on our skill sets and the projects memory requirements to be the best for our needs, which will serve the project by having quicker overall computation and therefore better feedback for the user.

4.3 Approach for design validation

Testing of the system must determine whether or not the design makes typing easier for users with motor function impairments. In order to test this, the team will coordinate with people who suffer from motor function impairments, such as Parkinson's, to observe their usage of the design. Ideally, this group will be observed typing using both MIK, as well as an unmodified keyboard, in order to establish whether or not there is an advantage in typing efficiency offered by using MIK. Users will be given a short period of time to familiarize themselves with using MIK, and then they will be asked to type blocks of text using both their unmodified keyboard and MIK. This will allow for the gain in typing efficiency to be calculated.

Additionally, short learning time is an objective of the design, making it necessary to gauge how long it takes users to become familiar enough with MIK to be able to type effectively. The amount of time it takes for a user to become familiar with the system can be tested by meeting with motor-impaired users multiple times in order to administer the aforementioned benchmark. Doing so allows the development team to determine whether or not the product is designed such that users are able to quickly improve their typing efficiency.

Given the short duration under which the product is to be developed and limited resources of the development team, customizability may be more difficult to validate than the other objectives. Testing customizability of the product will require prospective users to make selections on a variety of attributes and separate prototypes to be constructed. Conversely, determination of whether or not the cost objective has been met is quite trivial.

Lastly, the product must be completely usable by non-motor-impaired users without hampering their ability to type. Similar to the proposed validation for typing efficiency improvement, non-motor-impaired users will be administered the same benchmark in order to ensure that MIK does not reduce the ability to type when compared to an unmodified keyboard.

5 Engineering standards

5.1 Project management

The responsibilities of the team have been divided such that each person has at least one technical and non-technical responsibility. Everyone on the team has software experience, so the hardware technical roles were divided among those who have the most hardware experience. This responsibility will fall on Andrew, Max, and Abhishek. Connie, Brittany, and Jackson will be the team members who will prioritize software development. Finally, due to the nature of our product, 3D Printing and Modeling is another technical role that must be filled. Fortunately, Andrew and Jackson have experience and skills to fill this role.

Technical Role	**Members Involved**
Hardware Team	Andrew, Max, Abhishek
Software / Embedded	Connie, Brittany, Jackson
3D Printing / Modeling	Andrew, Jackson

Non-technical duties have also been divided among the team. Brittany has been the point of contact for outside resources for mentorship, assistive technology, and disabilities resources, and she will continue to do so. Brittany also has experience with project management and will do the planning for the team. The team has decided to utilize agile software development, and Andrew and Jackson will be the SCRUM masters due to each being on either the hardware or software team. Connie and Max will handle the majority of the technical reports and the meeting notes because they also span both the hardware and software teams. Finally, Abhishek will handle the finances and purchasing for the team's prototype.

Non-Technical Role	**Members Involved**
Project Manager	Brittany
Hardware SCRUM Master	Andrew
Software SCRUM Master	Jackson
Communications	Brittany
Finance	Abhishek
Technical Reports / Meeting Notes	Connie, Max

As previously mentioned, the team will be taking an agile software development approach for this project. The team will have weekly meetings during the designated lab time for the course. To maintain organization, the team will use Jira to facilitate the agile development.

Discord will be the primary communication for the team. The team has also been divided into a hardware and a software team. There are multiple channels in the discord server, for hardware team, software team, modeling, scheduling, and a general chat. This will keep the group communications organized and reduce bloated chats.

To share files the team will use a shared OneDrive folder that has been set up. This allows the team to work on documents at the same time and share documentation easily. For version control, the team will use GitHub. This will streamline the coding and development process and allow the team to work on different parts of the same project.

Coordination Tools
GitHub
Jira Site
OneDrive
Discord

5.2 Schedule of tasks

After creating some user stories for the project, we were able to identify a few Epics, which is Jira's grouping for related tasks. These are broken down below, and identify hardware, software / embedded, and Integration user stories.

I. Epic: Design the physical keyboard and procure hardware.
    a. User Stories:
        i. As a user, I want to be able to plug my keyboard in to any computer and have it work immediately because I do not know how to install complicated drivers/software.
        ii. As a user, I want to be able to type using all 10 of my fingers even if I am unable to press firmly with some of them.
        iii. As a stakeholder, I want the keyboard to be affordable for everyday users, so I will be able to sell more and turn a larger profit.
        iv. As a user, I do not want extraneous movement in my hands to cause me to mistype a key.
        v. As a user, I want the keyboard to be portable and no more than 5 lbs.

II. Epic: Design an embedded algorithm for predictive typing
    a. User Stories:
        i. As a user, I want to be able to be offered predictions based on my typing to save me key strokes.
        ii. As a developer, I want to save memory by using one language because the system is embedded and will have a finite amount of storage.

III. Epic: Integrate the embedded hardware components with the software algorithms
    a. User Stories:
        i. As a developer, I want the solution to be secure and protect the user from cyber-attacks, like injections.
        ii. As a technician, I want the code to be easily maintainable through commenting and documentation because I will not always be the one who wrote the code that needs to be fixed.
        iii. As a user, I want there to be little to no lag between typing and seeing my text appear on the screen because that could cause confusion.
        iv. As a user, I want a seamless experience with immediate feedback.

From the User Stories we were able to break them into different types of tasks and sub-tasks, as shown below. The number listed at the beginning of each task is the story points our team assigned from 1-5, where 1 is the easiest to complete and 5 is the most complicated and time consuming to complete. Below the software and hardware tasks have research sub-tasks that are prerequisites to completing some of the more technical tasks. Our biggest dependency will be between ordering and receiving parts to test and assemble the MIK. That is why we plan to order parts as soon as possible in order to mitigate risks if any parts we need take longer than expected to arrive. Because of this, we will also be focusing on designing the software in earlier sprints to allow time for the hardware to be delivered.

Development tasks:

I. Hardware:
    a. Research
        i. (1) Key Actuation (how it works, components needed, parts ordering)
        ii. (1) Create Parts list and begin purchasing
        iii. (1) key sizing vs. Missed keys
    b. Design of digital circuitry
        i. (3) Key press decoding circuitry
    c. PCB creation
        i. (5) Design PCB
        ii. (2) Print PCB
    b. Construction of physical circuitry
        i. (1) Ordering components
        ii. (3) Construct physical circuit
II. Software:
    a. Research
        i. (1) Research existing open-source algorithms for predictive text
        ii. (2) Calculate how much memory is needed for storage
        iii. (1) Download any dictionaries, databases, etc. needed to run algorithm
    b. Signal decoding:
        i. (4) get raw input from GPIO
        ii. (3) Key mapping
        iii. (1) Send key signals to encoder and key decision maker
    b. Key decision maker:
        i. (2) If alphabetic key, compute suggestions and update LCD
        ii. (2) If space, clear buffer
        iii. (2) If macro, send remaining buffer and clear buffer
        iv. (3) Shortcut handling with context (add apostrophe to buffer)
    c. Auto-complete algorithm:
        i. (2) Trie class skeleton code
        ii. (4) Trie constructor with dictionary (stored)
        iii. (4) Trie compute suggestion function
    d. USB encoding:
        i. (5) Encode provided keys to USB protocol and send signal
    e. LCD encoding:
        i. (4) Encode suggestions to LCD signal
II. Modeling:
    a. 3D modeling
        i. (4) Creating housing
        ii. (3) Creating keycaps
    b. Printing
        i. (2) Printing housing
        ii. (2) Printing keycaps
        iii. (2) Assemble the MIK
III. Integration
    a. Security
        i. (2) Check code for known vulnerabilities

ii. (3) Stress test prototype

b. Debug

i. (4) Unit Testing

IV. Marketing

a. Research

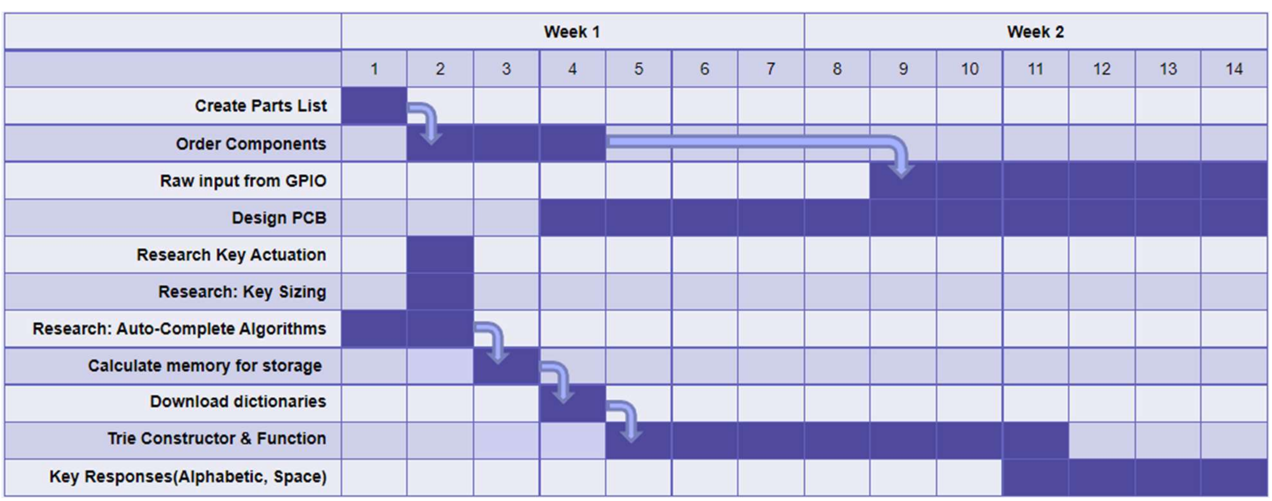
i. (1) prototype pricing vs. Alternative Final Product materials

The schedule for our sprints is outlined below. As mentioned, in earlier sprints we plan to focus on software implementation, in the middle sprints we will focus on hardware design and unit testing. Finally, for the last two sprints we will work solely on integration, testing, and security for the MIK. This will likely include any improvements that need to be made in the firmware or hardware before finalizing our product.

Iteration	Dates	Goal
Sprint 1	Sept. 20 – Oct. 4	Software design and implementation, hardware purchases
Sprint 2	Oct. 4 – Oct. 18	Circuitry design and build, unit testing for hardware and software
Sprint 3	Oct. 18 – Nov. 1	Software and hardware integration, system testing, 3d modeling
Sprint 4	Nov. 1 – Nov. 15	User testing and feedback, improvements
Sprint 5	Nov. 15 – Nov. 27	Security review, improvements

We plan to do a final one-week sprint where we finalize documentation, a demonstration, and presentation for our project and turn in all deliverables. This sprint will be from Nov. 27 – Dec. 5.

Gantt Chart (Sprint 1)



The Gantt chart analyzes dependencies and an estimated time frame for Sprint 1 to be completed by. IT offers a few days to start receiving parts after the initial order is placed. And it allows time for some research to be done before starting any of our software from scratch. Because Agile is the method we chose to build our project, we will need to reevaluate our expected time frame for the next sprint before creating a Gantt chart analysis. The time between ordering and receiving components is currently unknown, and therefore, may be underestimated in our Gantt Chart which could push us back when completing sprint 1.

5.3 Economic analysis

For the end user, our keyboard will cost around $100 dollars. This is a standard price for any quality keyboard, and far cheaper than other existing assistive technology. Eye trackers recommended for use with Windows Eye Control software cost at-least $250, while some assistive keyboards cost upwards of $400. Compared to the estimated prototype cost of $200, we expect the production cost to be $100

For the commercial hardware of the product, we are planning to use plastic for the keyboard components. There are many manufacturing companies that will create this for us, given a three-dimensional model. We will have to design a PCB, which can be mass produced for commercial use. Lastly, we will need some processor on the keyboard, which will handle text suggestion and auto correct computation. There are several options for this, including Raspberry Pi and Arduino.

Our product will require the same maintenance as a regular keyboard: occasional dust removal depending on usage. We are also planning to develop a non-essential software that can be used to customize the keyboard. This will contain software / firmware updates that we would want to extend to customers.

5.4 Societal, safety and environmental analysis

As with any engineering project, our proposed idea raises several repercussions that we must consider before, during, and even after development when it comes to our impact on society, safety, and the environment.

First and foremost, because our project is mainly targeted towards those with Parkinson's or similar motor function impairments, the societal impact of our project will likely be rather small in regards to the scale of our intended market. However, if we do well to fully represent our target audience in all stages of the project development, our end result will likely encompass and serve their needs to a considerable degree. This focus on creating a very good solution for a small portion of the population (as opposed to a mediocre solution for widespread use) introduces a promising improvement to the quality of life for those target users. Obtaining an end-result that improves typing accuracy and speed for motor-impaired individuals will provide a boost to their overall autonomy and independence. One potential harm to society of our solution can be a threat to privacy; regardless of how we implement our solution, marketing our end product as a "smart" keyboard is bound to raise some alarms for users who are wary of their personal data being stolen or sold. To ensure our users' privacy, we will not be collecting usage data in any form outside of optional user experience surveys and feedback.

Although the end product likely won't raise any safety concerns for our users, we will certainly encounter several safety concerns during the development of our prototypes. These include, but are not limited to, the development of our custom PCB, power delivery to our hardware components, and safety hazards when 3D printing the housing components. The best way to overcome these safety hazards will be to stay alert while following all outlined safety protocols and trainings.

Because the volume of production of our device will likely be fairly small, the environmental impact of our product will also be reasonably manageable. One decision we can consider to decrease our environmental impact is to look towards using recycled materials to create our product.

5.5 Itemized budget

Detailed budget of all costs expected to be incurred during the project (e.g., parts, fabrication services).

Product Name	Cost	Purchase Site
Raspberry Pi 0	79.99	Amazon

LCD Displays	15.99	Amazon
Keyboard Switches	47.98	Amazon
O-rings	6.99	Amazon
PLA Filament	18.99	Amazon
PCB Material	~90	jlcpcb.com
Total	259.94	

6 References

"BigKeys keyboards," *BigKeys Company*, 2017. [Online]. Available: http://www.bigkeys.com/. [Accessed: 14-Sep-2022].

"Get started with eye control in Windows," *Microsoft Support*, 2022. [Online]. Available: https://support.microsoft.com/en-us/windows/get-started-with-eye-control-in-windows-1a170a20-1083-2452-8f42-17a7d4fe89a9. [Accessed: 17-Sep-2022].

Kinesis, "Why get a programmable keyboard," *Kinesis*, 04-Oct-2019. [Online]. Available: https://kinesis-ergo.com/programmable-keyboards/. [Accessed: 18-Sep-2022].

M. Martin, "Computer and Internet use in the United States: 2018," *census.gov*, 2021. [Online]. Available: https://www.census.gov/content/dam/Census/library/publications/2021/acs/acs-49.pdf. [Accessed: 19-Sep-2022].

"Motor Disabilities Assistive Technologies," *Motor Disabilities*, 12-Oct-2012. [Online]. Available: https://webaim.org/articles/motor/assistive. [Accessed: 17-Sep-2022].

S. J. Davies, "Our new keyguard alternative," *LimeTech*, 29-Jan-2022. [Online]. Available: https://limetech.uk/keyguard-alternative/. [Accessed: 16-Sep-2022].

T. Rudnicki and T. ATI, *Eye Control Empowers People with Disabilities*, 2022. [Online]. Available: https://www.abilities.com/community/assistive-eye-control.html. [Accessed: 19-Sep-2022].

7 Appendices

7.1 Product datasheets

Include product datasheets that may be particularly relevant to your proposed work. Say you want to use a certain type of microcontroller, because it has just the right combination features (e.g., types of I/O ports, or power consumption, etc.). You would then attach datasheets for this product.

NOTE: including a data sheet does not replace the need for explaining how the component works and how it will be integrated in the system (section 4.2).

Raspberry Pi 0 Datasheet

LCD Display Datasheet

7.2 Bios and CVs

Jackson Hagood

I am a Senior Computer Engineering student studying at Texas A&M University. I have worked as a software engineering intern at JPMorgan Chase & Co., a teaching assistant at Texas A&M University, and a coding tutor at Code Ninjas. Upon graduation in May of 2023, I am looking to enter the industry as a Software Engineer. Through my work experience and coursework, I have worked with and been exposed to a wide range of technologies and programming languages. I am most proficient with C++, Java, JavaScript, and Python, but have experience with other languages like Haskell, Assembly, and Lua. I have worked as a back-end, front-end, and full-stack developer and have experience with designing web-applications from the ground up. I also have some low-level programming and hardware experience. Some of my past projects include a full-stack application at JPMorgan Chase & Co., a flashcard image reader web-application, and an audio message encryptor / decryptor.

JACKSON **HAGOOD**

Computer Engineering Student & Prospective Software Engineer

hagoojac@tamu.edu

College Station, TX

www.jacksonhagood.com

github.com/JacksonHagood

Profile

Senior Engineering student at Texas A&M University with a 4.0 GPA. Worked as Software Engineering Intern at JPMorgan Chase in Summer 2022. Currently work as a teaching assistant in the CSE department at Texas A&M. Seeking a Software Engineering position upon graduation in May 2023.

Education

BS Texas A&M University *8/2019—Present*

- Bachelor of Science in Computer Engineering (Computer Science track, CECN)
- Minor in Mathematics
- 4.0 GPA, 113 hours completed (senior classification)
- Dean's Honor Roll
- Member of Tau Beta Pi (Texas Delta Chapter)
- Expected graduation May 2023

Work Experience

Software Engineering Intern at JPMorgan Chase *6/2021—8/2021*

- Worked as full-stack software engineer on a project to be deployed into production
- JavaScript React / Redux front-end
- Java Spring Boot back-end
- Cassandra database
- Deployment with Jenkins
- Agile development with Jira and Bitbucket

Undergraduate Teaching Assistant for CSCE 121 *9/2021—Present*

- In charge of leading weekly lab sections
- Hold frequent office hours to assist students with lab exercises and homework
- Help formulate course materials including exam questions
- Grading responsibilities
- CSCE 121 covers object-oriented programming, dynamic memory, debugging, algorithms, and software development

Code Sensei at Code Ninjas *6/2018—12/2021*

- Worked at Code Ninjas The Woodlands and Code Ninjas Magnolia
- Primarily instructive and tutoring work with students aged 7—16 in JavaScript, Python, and C#
- Served in a lead role which included advising other locations and training new Senseis
- Developed documentation for JavaScript curriculum used at several locations across the United States
- Designed and implemented a introduction to Python camp

Eagle Scout *2013—2018*

Programming

C++

- Experience with OOP, algorithms, abstract data types, and computer systems
- Teaching experience as CSE TA

Java

- Experience with OOP, abstract data types, and multi-threading
- **Spring Boot** API development

JavaScript

- Asynchronous programming, OOP, and **Node JS**
- **React JS / Redux JS**
- **HTML / CSS**

Python

- Experience with OOP and mathematical modeling (**NumPy**)
- **Flask** full stack development
- Code Ninjas camp development

Other Technologies

- **Cassandra**
- **PostgreSQL**
- **Haskell**
- **ARM Assembly**
- **Lua**
- **MATLAB**
- **Git**
- **Jenkins**
- **Jira / Bitbucket**

Skills

Electrical Engineering

- Circuit design and analysis
- Signals and systems
- **Verilog**
- **SPICE** and **Multisim**

Mathematics

- Cryptography
- Calculus / differential equations
- Linear algebra and applied math
- Discrete math

3D Modeling & CAD

- **Blender, Maya, 3DS Max**, and **AutoCAD**

Computer Hardware

Andrew Imwalle

I am a Senior Computer Engineering student studying at Texas A&M University. I have worked as a systems engineer developing firmware on a security chip for servers at Hewlett Packard Enterprise. I am currently working at the Fischer Engineering Design Center at Texas A&M University where I manage requests for 3D Printing, Laser Cutting, and Printed Circuit Boards. Through my experience at my various jobs and courses, I have learned a wide variety of modeling software and coding languages. I have worked as a back-end, front-end, and firmware developer and enjoy taking on a project manager role. Some of my past projects include a fire staff, a videogame using Unity, firmware development for a security chip, and an audio message encryptor / decryptor.

Andrew Imwalle
College Station, TX | 469-247-1283 | animwalle@gmail.com

EDUCATION

Texas A&M University, College of Engineering — College Station, TX
Bachelor of Science in Computer Engineering (CECN) — Expected May 2023
GPA: 3.718

Blinn College, Texas A&M Engineering at Blinn — College Station, TX
General Engineering — May 2020
GPA: 4.0

The Woodlands High School — The Woodlands, TX
GPA: 4.23 — May 2019

EXPERIENCE

Hewlett-Packard Enterprise — Spring, TX
Firmware Engineer — June 2022 – August 2022
- Created a scalable testing framework for a server security chip
- Automated the testing to upload the results of the tests directly to GitHub

Fischer Engineering Design Center — College Station, TX
Student Worker — June 2021 - Present
Prototyping Center Volunteer — February 2020 – May 2021
- Processed request for 3D prints, Laser Cutting, and Printed Circuit Boards
- Advise Clients on printing specifications and troubleshoots concerns
- Maintained and serviced a variety of 3D printing machines
- Assisted in coding database system that tracked requests
- Created models in CAD software to laser cut for requests
- Offered advice and information to customers and updated databases

H.E.B. Grocery Store — The Woodlands, TX
Cashier — May 2017 - December 2018
- Worked an average of 20 hours a week as a Cashier supporting customers in buying Groceries
- Received and processed payments by cash, check, and credit card from customers
- Served the community during Hurricane Harvey and was awarded the title 'Hurricane Harvey Hero'

RELEVANT SKILLS

- Coding Languages: Rust, Java, Html, CSS, Python, C#, and C++
- Computer Animation: Maya, Mudbox, Blender, 3Ds Max
- Videogame development using Unity Engine
- Modeling for 3D prints: Blender, AutoCAD, Solidworks
- Operation and use of a personal Creality Ender 3D printer
- Modifications and repair for personal 3D printer
- Experience building and upgrading multiple personal desktop computers

PASSION PROJECTS

Individual:
Creating and implementing a videogame using Unity Engine using C# — Spring 2021 – Present
Using Unity game engine and coding in C#

Collaborative:
Designed, coded, and built a staff that shoots fire when brandished — Summer 2020

EXTRACURRICULAR

Texas A&M Gymnastics Club — August 2019 – Present
- Practiced and perfected skills on a variety of events including Parallel Bars and Pommel Horse
- Raised money for the Texas A&M Gymnastics Club by participating in Flips for Tips
- Coaching team mates to be able to perform new skills

Brittany Jenkins

As a senior in Computer Engineering at Texas A&M University, I have been able to develop my skills and interests for a wide range of topics. I have always been interested in accessibility in computing for people with visual, neurological, and hearing impairments, and I am looking forward to working with this team to design and build an assistive technology. Most of my experience is in cybersecurity concerning Linux systems and network administration. I have experience developing system management tools in Python and Perl. I plan to start work as a systems administrator after I graduate in December of 2022. Currently, I am continuing to develop my skills through the Texas A&M Cybersecurity Center, where I work on server maintenance and configuration, Virtual Machine deployment, and help with Lab design for computer science and cybersecurity classes.

BRITTANY CAPE JENKINS

(979) 479-5065 • bcjenkins32759@tamu.edu • LinkedIn: https://www.linkedin.com/in/brittanycapejenkins/

Clearance

Current Top Secret / SCI Clearance (May 2021)
CI Polygraph (April 2021)
(Both Last Active: August 2022)

Certifications

Security+ Certification,
CompTIA, 07/2021 - 07/2024

Work History

Software Engineering Co-op 05/2022 to 08/2022
L3Harris Technologies – Greenville, TX

- Developed automated software inventory program on over 200 machines
- Installed, configured, and maintained hardware and software equipment including switches, virtual machines, raid storage servers, and user workstations across 5 labs
- Collaborated with systems administrators, systems engineers, software engineers, and IT staff to learn their daily tasks and complete my tasks successfully
- Oversaw communication with corporate IT department to discuss Microsoft SharePoint 2007 migration project
- Ensured proper account creation and permissions for users

Texas Cyber Range Employee 03/2022 to Present
Texas A&M Cybersecurity Center – College Station, TX

- Maintain type 1 hypervisors to support cybersecurity labs and research for Texas A&M university
- Work with servers, switches, firewalls, racks, and software to ensure availability of services
- Provide service and customer support to clients on the range

Cybersecurity Apprenticeship Program 09/2021 to 04/2022
Texas A&M Security Management and Incident Response Center – College Station, TX

- Develop skills for analyzing possible network intrusions including using Splunk, reading firewall logs, and utilizing Linux tools

Apprenticeship Program Intern 06/2021 to 08/2021
U.S. Marine Corps Forces Cyberspace Command – Fort Meade, MD

- Successfully completed rigorous Industrial Control System training program
- Collaborated to detect, remediate, and recover from virtualized live cyber attacks
- Gained practical knowledge of offensive and defensive cyber skills including penetration testing, SIEM tools, device and network hardening, and python scripting

Student Security Analyst 10/2020 to 05/2021
Texas A&M System Security Operations Center – College Station, TX

- Monitored AI ticketing system and investigated cybersecurity incidents for more than 21 Texas A&M system universities and agencies
- Understand indicators of compromise and document malicious & benign activity

Professional Skills

Intrusion Detection	Project Management	Problem Solving	Wireshark
Kali Linux	Java Programming	Network Analysis	Time Management
C++ Programming	Python Scripting	Elastic Stack	Interpersonal Skills

University Education

Bachelor of Science: Computer Engineering from Texas A&M University - College Station, TX 08/2018 to 12/2022

- GPA: 3.41
- Minors in Mathematics and Cybersecurity

Affiliation

Defense Cyber Leadership Development Program, '20-'21 Cohort 09/2020 to present

- Competitively selected for NSA-sponsored developmental program designed to prepare students to fill vital cybersecurity roles within the DoD

Texas A&M Cybersecurity Club, Member 09/2020 to present

National Cyber League Competition, Silver Bracket 10/2020, 11/2021

Institute of Electrical & Electronics Engineers, IEEE National Member 09/2019 to 05/2020

Aggies Invent: Nuclear Security, Participant 09/2019

Connie Liu

I am a senior Computer Engineering major at Texas A&M University. Last summer, I worked at Cirrus Logic as an intern on the applications engineering team, where I was able to gain a lot of experience in Python scripting and development. A majority of my technical experience falls under front-end software development, but I do have an interest in learning more about full-stack development and embedded systems. Throughout my coursework and past internship experiences, I have acquired many skills in programming languages and technologies such as C++, Python, Java, HTML/CSS, and ReactJS. Once I graduate in Spring of 2023, I hope to begin working full-time as a software engineer at a company based in Austin, TX.

Connie Liu

(806) 317-5356 | connie.liu@tamu.edu | https://github.com/conconloo

EDUCATION

Texas A&M University, College Station, Texas — **Aug 2019 - May 2023**

B.S. in Electrical and Computer Engineering — GPA: 3.897

WORK EXPERIENCE

Cirrus Logic (Austin, TX) — **May 2022 - Aug 2022**

Applications Engineer Intern

- Converted python scripts to from old Cirrusvue to new SCS API for customer-specific audio codec
- Created Audio Precision and python test scripts and procedure documentation for future FACR tests
- Upgraded toolchain software for XMOS USB audio chip used by Cirrus AudioHub

BNSF Railway (Fort Worth, TX) — **Jun 2021 - Aug 2021**

Technology Services Intern

- Facilitated software transition of BNSF employee portal metrics for over 41,000 daily users
- Consolidated cross-department documentation for BNSF Connect app upgrade process
- Engaged in various networking and interest organization opportunities

The Association of Former Students (College Station, TX) — **Feb 2021 - Dec 2021**

Web Design Student Assistant

- Provided support to Web Design Team by building custom pages with HTML/CSS/Javascript
- Supported WordPress sites for over 100 Classes, Clubs, and Constituent Networks within A&M System
- Contributed to and participated in various Association-wide activities and Aggie traditions

PAST PROJECTS

Raspberry Pi Modern Record Player

- Utilized RFID/NFC cards and Waveshare PN542 NFC HAT to create an R-Pi modern record player
- Integrated Spotify API & authentication for user song playback with Spotipy and Raspotify
- Coordinated addition of reactive LED lights and 3D printed housing components completed by teammates

Pocket-Watch Safety App

- Created a web app to inform users of real-time weather hazards, crime rates, and safety training
- Built front-end UI and included interface accessibility features with HTML/CSS and ReactJS
- Worked with teammates in code sprints using Agile methodology, delivering webapp with 5 separate APIs

LEADERSHIP

Sling Health, Texas A&M University Chapter

Co-President — **May 2021 - Sep 2021**

- Oversaw yearly recruitment and curriculum initiatives within Sling Health
- Ensured on-target progress for event planning and preparation of Design Reviews and presentations
- Over $2.45 million raised by former teams in outside funding

Vice President of Engineering Integration — **May 2020 - May 2021**

- Primary liaison between Sling Health leadership board and external academic/industry relations
- Provided advice to teams on designing and prototyping process
- Organized and hosted Design Reviews for team progression feedback throughout academic year

Chinese Student Association, Texas A&M University Chapter

Household Parent — **Aug 2019 - May 2022**

- Head of household; responsible for fostering friendly relationships and team building efforts among members
- Organized and coordinated weekly family dinners and study sessions throughout academic year
- Collaborated with two other household parents to organize, plan, and budget for bi-weekly large events

SKILLS

Experienced in: Python, C++, Java, Java Swing GUI development, HTML/CSS, ReactJS, PostgreSQL, Verilog, ARM Assembly
Familiar with: Git version control, Agile development, OOP principles, Linux/Unix and Windows environments

QUALITIES

Flexible, out of the box thinker • Solid communicator and team player • Highly responsible and reliable • Attentive and focused • Efficient time management

Abhishek More

I am a Senior Computer Engineering student studying at Texas A&M University. This summer, I worked as a software engineer intern at Amazon, using Golang and TypeScript to develop new features for AWS EC2. I am currently on the tech team at TAMUhack, the organization that runs the biggest hackathon in Texas. My responsibilities include creating the annual event webpage and managing the registration system. This year, I created a new virtual mentoring software using a React / Next.js frontend and a Node.js / Express backend.

Abhishek More

abhishekmore@tamu.edu
abhishekmore.com

EDUCATION

Texas A&M University — Expected May 2023
B.S Computer Engineering — *College Station, TX*
Minors: Cybersecurity, Mathematics — *GPA: 3.933 / 4.0*

- Brockman full-ride Scholarship
- National Merit Scholarship from 3M

COURSEWORK

Data Structures and Algorithms, Discrete Structures, Differential Equations, Linear Algebra
Circuit Theory, Machine Learning, Cloud Computing, Human-Computer Interaction

EXPERIENCE AND PROJECTS

TAMUHack / *Hackathon Development Director* — **Jan 2021 - Present**

- Used **Golang** to build the TAMUHack registration system backend, supporting **4000+** users.
- Handled authentication, data, and file storage with **Firebase**, **PostgreSQL**, and **AWS S3**.
- Built TAMUhack's virtual mentoring software with **Next.js / Tailwind**.

Amazon / *Software Engineer Intern* — **May 2022 - Aug 2022**

- Developed crash detection software for AWS App2Container using Go and **TypeScript**.
- Worked with AWS Serverless architecture, including **Lambda**, **CloudWatch**, and **DynamoDB**.
- Increased **Java** unit test line coverage 10%, up to over 85% total line coverage (starter task)

Systematrix Solutions / *Software Engineering Intern* — **May 2020 - Aug 2020**

- Used **Python** and MongoDB to scrape and store geospatial data of businesses world-wide.
- Created data visualizations in **Tableau** and presented to upper management teams weekly.

Asciifi / *TAMUHack Grand Prize Winner* — **Jan 2021**

- Built a low-latency video-calling platform with **React.js** and **OpenCV** which converts video to ASCII characters, yielding speeds **300%** faster than Zoom, Skype, etc.

PROGRAMMING SKILLS

- Languages	Javascript, Go, Python, C++, Swift
- Formatting	HTML, CSS, TailwindCSS, Latex
- Web	React.js, Node.js, Next.js, Express
- Databases	PostgreSQL, MySQL, MongoDB

AWARDS

- Winner of **6 hackathons,** including:
  - TAMUHack 2021
  - UIUC HackThis 2020
  - HackOHIO 2019
  - CBUSHack 2019

ACTIVITIES

TAMUHack Director	**Jan 2021 - Present**
Google Developer Student Club	**Jan 2020 - Present**
Aggie Coding Club	**Sept 2019 - Present**
Texas A&M Orchestra	**Sept 2019 - Present**

Max Smith

I am a senior Computer Engineering student at Texas A&M University. Currently, I am working as an undergraduate teaching assistant for Texas A&M's Computer Science and Engineering Department and have additional experience teaching mathematics. Through my coursework, I have gained experience in operating systems, computer architecture, hardware design, and a multitude of skills and technologies including C, C++, Python, Java, Cadence, and SPICE simulators. Past projects include an audio message encryption/decryption system, a political education game, and designing a single-cycle processor.

Max Smith

maxsmith271346@gmail.com

EXPERIENCE

Undergraduate Teaching Assistant August 2021–Present
Texas A&M Department of Computer Science and Engineering

- Hold weekly office hours to assist students with coursework, as well as help students improve problem-solving and programming skills
- Lead weekly C++ programming lab sessions focused on introductory programming concepts (problem solving skills, debugging, dynamic memory, object oriented programming, basic data structures)

Private Tutor June 2022–August 2022
iD Tech

- Designed lesson plans and taught lessons in mathematics and computer science to students aged 7-19

Food Service May 2019–July 2022

- Worked multiple positions in food service to help defray educational and living expenses
- Learned to effectively work and communicate with a team in a fast-paced environment

LEADERSHIP

Institute of Electrical and Electronics Engineers, Texas A&M Chapter Spring 2022–Present
Technical Education Committee Officer

- Organizing IEEE TAMU 2023 Micromouse competition and holding a series of lectures/workshops on related concepts
- Responsible for programming education initiatives
- Served as a team leader in the IEEE TAMU 2022 Micromouse competition, guiding a team in developing a maze-traversing robot

Secular Student Alliance, Texas A&M Chapter May 2022–Present
Treasurer

- Manage organization finances
- Organize weekly meetings, monthly social events, weekly tabling
- In charge of campus outreach

EDUCATION

Bachelor of Science, Computer Engineering Expected May 2023
Texas A&M University, College Station, TX
GPA: 4.0

SKILLS

Programming Languages: ARM-V8 Assembly, C/C++, Python, Java, Verilog
Tools: Cadence, Git, NI Multisim, Jira, Vivado
Mathematics: Linear Algebra, Discrete Mathematics, Calculus

PROJECTS

Message Encryption/Decryption System Spring 2022

- Worked with a team to construct device capable of sending and receiving encrypted messages through audio
- Prototype implemented encryption/decryption, error correcting code, audio processing, analog-to-digital conversion circuitry