CSU SACRAMENTO
COLLEGE OF ENGINEERING & COMPUTER SCIENCE

Team 1
End of Project Documentation

Authors:
Jeremy Shaw
Alexander Leones
Angelica Smith-Evans
Jesse Aucelluzzo

Professor Tatro, Professor Thomas, & Professor Levine

EASR – Elderly Assistance Storage and Retrieval Robot

Date Submitted: April 27th, 2020

SACRAMENTO
STATE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

In America, the elderly demographic has been among the fastest growing age demographics. Due to this rise, there is greater demand for appropriate care solutions. To help mitigate the societal impact of deploying in home assistance on a large scale, we propose the non-invasive Elderly Storage and Retrieval (EASR) Robot. The EASR Robot aims to improve the independence of its owners.

This report details the development and rationale our team made for the EASR Robot. The report begins from the initial conceptual design of this project, moving into the development phase, and finally the testing results. This document covers the how Team 1 (EASR) came to focus on the team's chosen societal problem, the rise in demand for elderly care. This includes statistical research, definition of the current issues in this field, and the ideation of a solution. This report looks over the core concepts of Team 1's project design proposal, which explains the approach and rationale of the robot design. In depth details of each step taken to build the rig are included, covering a two-semester research & development period.

This report also details how the team allocated up the work between its members. In doing so, each member was able to focus on a specific aspect of the project and deliver on time. The specifics are in the Project Milestones and Work Breakdown Structure sections. This project is self-funded and cost as much as $450 by the end of the development period. This price fell within early estimates.

With every project, there are risks associated with it. The siloed approach taken by Team 1 was able to mitigate some of the proposed risks of Senior Project development. However, the totality and depth of the campus closure, due to COVID-19, was somewhat unexpected, though not unprecedented. This closure affected some aspects of the project development, though effective risk mitigation techniques allowed the team to complete the project and deliver on the promised feature set. Included in this report is the design philosophy, which shares the approach Team 1 took, and how this design solution may be applicable to people with mobility or cognitive impairments.

This project was chosen and developed, in part, to contribute to a slow global trend of personal, assistive robots. These digital companions are intended to aide people in their day-to-day lives. Market research shows that there is a gap in these market that this robot may be able to fill. Team 1 hopes to have made a positive impact in the field of elderly care with the EASR Robot.

ABSTRACT

The elderly population is rapidly growing in the United States. Team 1's robot solution, the Elderly Assistance Storage and Retrieval (EASR) Robot solves this. The EASR Robot is a complete system comprised of interworking parts, with contributions from all team members. Our team used research done on the rise in demand of elderly care, our chosen societal issue, to propose and develop a solution. This solution is a robot that has four major interworking parts: a graphical user interface (GUI), the pathing and logic control, the lifting arm, and the chassis/electrical system design. This was to modularize our project into areas of expertise. The expertise is divided between team members for ease-of–development, and to simulate real-life workflow. Our robot was designed to meet a specific set of self-made measurable metrics to gauge completeness. Through our period in senior design, our team was able to meet every one of these measurable metrics in our punch list.

Our current deployable prototype status is complete based on our punch list. Therefore, during this two-semester design period, we built an assisting robot that was able to store, catalog and retrieve items for the user with the goal or reducing injury and increasing elderly people's effective functionality. This prototype will be able to fill the niche market of elderly care that is consistently and economically available. This end of project report covers all the relevant tasks and effort involved in this project.

KEYWORD INDEX

Robotics, older adults, elderly, gerontology, autonomous robot, mobility, disability, caregiving, independence, ease-of-use, valet service, storage, retrieval, in-home assistance, senior citizens, baby boomers

I. INTRODUCTION

The goal of this documentation is to narrate the actions the team took through our ABET-accredited senior design course at California State University Sacramento (CSUS) from Fall 2019 to Spring 2020. In this time span, Team 1 (Aucelluzzo, Leones, Shaw, Smith) worked on the EASR robot, with a goal of serving the elderly population. This allows a traditionally vulnerable group, who often needs a higher level of constant care, to mitigate the need for persistent care. Included in this documentation is the full breadth of steps our team has taken to ensure the project's success. This report describes this process from the beginning, where the team discovered its chosen societal problem, to the completion of the final deployable prototype.

The formal report starts with background research of elderly care within the Societal Problem section. In this section, the factual, statistically backed reality of a growing elderly population in the United States is discussed. Additionally, the financial considerations of this demographic are addressed. This demographic has the constraint of lower income, which conflicts with the potential need for medical care. Senior citizens are most at-risk for sustaining injuries that result in mobility-impairments or cognitive-impairments. The statistics of age-related injuries in Injuries and Mobility Constraints and Cognitive Issues are also presented. It is at this point, where the importance of addressing the need for more round-the-clock care at a lower cost is raised. This is discussed within Demand for Care. Considering the rising elderly population, which demands care at relatively low costs, Team 1 proposed a technical solution, the EASR robot, which will aide this underserved population. The full depth of this is discussed in the following section.

Next, we discuss our Design Idea Contract. In this section, we propose the technical solution we believe will service the growing elderly population. We propose a semi-autonomous robot that will be able to store and retrieve relatively small objects. We discuss how this solution satisfies a simple and effective solution for those with basic technological experience. In Robot Function, we explain in detail how the robot operates. Next, we review the EASR robot's measurable metrics, which we sought to complete by our term's end. These measurable metrics gauge the success or failure of our end-of-term Deployable Prototype. Finally, we discuss how we met the measurable metrics of the following features: GUI and User Experience, Pathing, Lifting Arm, and Chassis and Electrical System.

Moving on, we address purchasing and the acquisition of the parts which comprised this build.

In the following Funding section, budgeting and personal expenses of the project is concisely explained. Included here is the itemized list of every relevant equipment purchase for the deployable prototype rig.

Thereafter, all major project milestones are discussed. Significant milestones are divided into two categories: project management and major build features. The former, project management, had hard deadlines and were class assignments. These milestones helped build the repertoire of our team's writing and management skills. The latter, major build features, and steps taken to obtain completion.

The follow-up section to this is the Work Breakdown Structure. This is provided as the divvying up of major project tasks. In this section we discuss in detail how our team tackled each aspect of the project. We discuss the completion of the four major aspects of the robot: graphical user interface (GUI), logic control and pathing, lifting arm design, and finally the chassis and electrical system. In addition to this, we also discuss the steps to completion for our major events: Fall 2019 Senior Project Showcase and the Spring 2020 Deployable Prototype submission. This section also represents how our team utilized logical and efficient tackling and problem-solving at each step of the design process.

Moving forward, we explain our team's project mitigation plan in the Risk Assessment section. Here we discuss the potential risk associated with the major aspects of the robot design.

As with any project of this width and breadth, we provided a completed risk-mitigation plan on potential project hazards. This turned out to be especially important, given that the last fourth of project development was severely affected due to the 2020 COVID-19 pandemic. The mandated Stay-at-home order severely limited our team activities, and we were forced to work on any team assignment remotely from self-isolation. This leads to us reflecting on things our team did right, and ways we could improve our planning. Our mitigation solutions are discussed in-depth within this section.

Next, we explain how we decided on our project vision in Design Philosophy. This section does not review the specific technical aspects of our design but will instead convey the big picture goals we had in mind for our design. With our design idea and philosophy laid out, we will prove our fulfilment of the measurable metrics through our testing reports. The goal of most of our tests is to see if the promised functionality was delivered at a dependable level. We wanted to build this robot with a specific user in mind, and we touch on that. Finally, it is explained why we chose a modular design for the robot.

In the last portion of the report, we discuss the marketability forecast of our project. Here we picture our project as a marketable product. In doing so, we provide research into the current market trends, our target audience, and finally potential market competitors.

Finally, the project documentation concludes with a reference listing, a glossary of technical terms, and various appendices. Within the appendices, all data not diegetic to the main report's narrative is presented. This includes the User Manual, technical details of the Hardware, an in depth reveal of the GUI Software's inner workings, and a brief overview of the team's 4 members.

II. SOCIETAL PROBLEM

In the next 50 years, there stands to be a major demographic shift in the United States. The number of Americans over the age of 65 is expected to double by 2060. This will increase the percentage of those in this age group from 16% to 23% of the general population. [1] With the substantial rise in average age, the demand for appropriate accommodation will also increase. As the demand for care services rises, increasing cost for care and shortages of caregivers is expected. For older adults, there is a need to carry out tasks that may carry risks, especially for those without ready and affordable access to aid. These tasks include simple services, such as grabbing one's medication or cell phone. Having these items on their person would decrease risks that may arise if they had not had them. For example, a person who is a diabetic may need extra help to bring them insulin or a sugary snack in an emergency. These tasks that may seem simple require the introduction of an autonomous device that can help in these situations. Our team took time to think about issues that may arise in the lives of our relatives, as well as future concerns of our lives and the lives of loved ones. For some, there will be a certain point in time in which they would lack the ability to move as freely. This is the main societal issue we attempt to address throughout our design process.

The following subsections provide an in-depth analysis of the issue of an aging population and the current demand for a solution to this issue. When conceptualizing one issue this presents, which is the lack of full-time, cost-effective senior valet servicing currently, it is important to look from this from a multitude of perspectives, i.e. a statistical point of view or a financial point of view. From the statistical analysis of the issue, as well as financial, and further multi-faceted analyses we can make an argument for a proposed solution. Our team will assert the need for an autonomous valet design, once the argument is made. It is especially important to recognize the factual data that is currently present in this field.

When considering the problems that arise from an aging population, it is important to analyze why the population age is shifting, and the unintended costs of this shift. We will address the costs, labor and financial considerations. After establishing a rise in this population, it is then necessary to describe the risks this population faces. These risks include cognitive impairment, diseases that cause cognitive impairment, such as Alzheimer's, and accidents that cause injuries, such as falling. Finally, we address the societal burden of caregiving from the family's perspective as well as from a societal perspective. Understanding multi-faceted perspectives will help prove the need for our engineered design.

A. An Aging Population

There is a global demographic shift currently taking place. Living standards have improved, medical care has expanded, and birth-rates are declining. In turn, elderly Americans are becoming a greater percentage of the U.S. population.

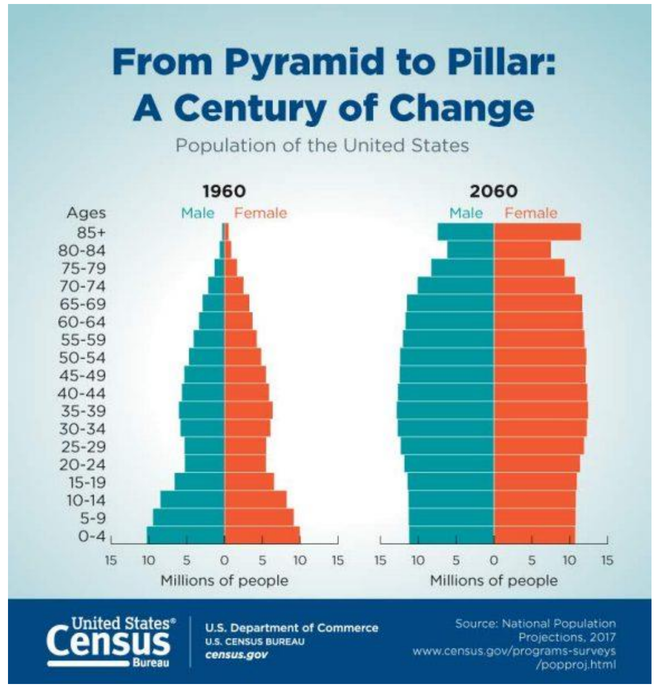

Figure 1. Projected population shift in the United States [2]

According to the Population Reference Bureau via the U.S. Census Bureau statistics, "the number of Americans ages 65 and older is projected to nearly double from 52 million in 2018 to 95 million by 2060, and the 65-and-older age group's share of the total population will rise from 16 percent to 23 percent [1]." As this population group grows, demand to accommodate their unique needs increases as well. Challenges include decreased mobility, increased prevalence of neurodegenerative disease, and a dramatic increase in the demand for nursing home care [1]. With the average age of a population shifting, there will be an increased need for care via nursing homes and similar care facilities. There will likely be a shortage of caregivers who can provide specialized care to patients without being overwhelmed with work. If a patient is not served in a timely manner, or if their caregiver is feeling overwhelmed, there can be an increase in the risk of accidents due to carelessness. The population of older adults is growing, and society will have to shift to accommodate. Along this vein of thinking, older populations also must consider the financial costs of reaching a retirement age.

B. Financial Considerations

Finances are also a concern for those over the age of 65. According to the U.S. Administration for Community Living, the average income reported by this group in the census was $24,224 and 9.2% were under the poverty level as of 2017 [3]. Furthermore, according to the Supplemental Poverty Measure the number of elderlies under the poverty level is 14.1% which is almost 5 percentage points higher than the official rate [3]. Out of pocket medical expenses is one of the main causes for this number according to the Supplemental Poverty Measure [4]. Finances and price must be considered when creating a solution to issues specific to older Americans. The solution should ideally be a cost-reducing solution compared to the other similar solutions available at the time of creation.

C. Injuries and Mobility Constraints

An aging population increases the threat of catastrophic injury, specifically those resulting from age-induced limited mobility. According to the Center for Disease Control, “Falls are the leading cause of fatal and nonfatal injuries among persons aged ≥65 years [5]". The CDC further notes “the rate of deaths from falls among persons aged ≥6 years increased 31% from 2007 to 2016 approximately one in four U.S. residents aged ≥65 years report falling each year. Fall-related emergency department visits are estimated at approximately 3 million per year. In 2016, a total of 29,668 U.S. residents aged ≥65 years died as the result of a fall, compared with 18,334 deaths in 2007 [5].” The risk factor is increased when an elderly person is alone at home, because “approximately 75% of all elderly falls occur at home,” [5] which can delay the response of medical assistance if the person is not utilizing an automatic emergency response system or in-home caretaker. One of the hardest fall-related injuries to recover from is a broken hip. Over 300,000 older people are hospitalized for hip fractures annually, with 95% of hip fractures resulting from a fall [7]. The chances of one fracturing their hip increases with age as many elderly people have osteoporosis. Osteoporosis is a disease that weakens the bones, leading to increased risk of fracture after any sort of trauma. After sustaining a fractured hip, for example, an elderly person will have to undergo a costly surgery and will likely lose the ability to walk during the recovery period, or even permanently. Only one half of surviving hip fracture patients regain the ability to walk without assistance [8]. A patient would also be at a far greater risk of contracting secondary medical complications such as blood clots, bed sores, urinary tract infections, and pneumonia. Hip fractures are associated with a mortality rate ranging from 14% to 36% within one year of hip surgery [9].

Falls are rapidly growing in number amongst older Americans and often tragically followed soon by mortality.

D. Cognitive Issues

Older People commonly face age-related neurodegenerative diseases such as Alzheimer’s disease and dementia. The lifetime risk for Alzheimer’s dementia at the age of 65 is 21.1% for women and 11.6% for men. Currently, there is an estimated 5.8 million Americans with Alzheimer’s dementia as of 2020 [10]. These diseases can drastically affect their lives through the onset of memory and thought loss. Some people with memory problems have mild cognitive impairment, or MCI. For those with MCI, most daily activities are not impaired, but some memory and movement difficulties are linked to the condition [11]. Simple daily tasks, like remembering where one placed their wallet, can become a challenge for those with these diseases. This is especially the case with progressive neurodegenerative diseases such as Alzheimer’s. Also, diseases such as Alzheimer’s can cause sufferers to place objects in illogical locations. Those afflicted with these diseases face serious challenges and will often require continual assistance for their daily routines.

E. Demand for Care

According to the Population Reference Bureau’s analysis of U.S. Census data, the increasing number of older Americans “could fuel more than a 50 percent increase in the number of Americans ages 65 and older requiring nursing home care, to about 1.9 million in 2030 from 1.2 million in 2017” [1]. Someone turning age 65 today has almost a 70% chance of needing some type of long-term care services and support, according to Health and Human Services [12]. Many are predicting that the rising demand for in-home care can trigger a health crisis amongst older adults. The rising demands are predicted to cause an increase in the cost of care, which could potentially price out many elderlies who, as briefly stated earlier, are often lacking full financial security. Medicare and Medicaid are the largest single payers for long term care in the United States, accounting for 43% of expenditures [12]. Medicare pays for home-based care, but only for a short duration following hospitalization [12]. In total, long term care accounts for 40 percent of Medicaid spending annually [12] placing a great strain on this publicly funded health care program. The disparities in what care these medical assistance programs provide force “at least 17.7 million individuals in the United States to provide care to an older parent, spouse, friend, or neighbor who needs help because of a limitation in their physical, mental, or cognitive functioning [13]”. The care provided is also not equal amongst all these cases. Some of those caring for older people provide daily care, and others only provide only the occasional care [13]. This can pose as a safety concern during the times older people in need of care do not have anyone to assist them. This method of implementing in-home care can function as a burden in the lives of caregivers. Compared to non-caregivers, family caregivers of older adults are more likely to experience emotional distress, depression, anxiety, or social isolation [14]. They also will also often fall victim to financial instability due to paying for the care out of pocket.

F. Societal Problem Conclusion

Today, as our society continues growing, our elderly population is also growing, along with all the unique needs that come with the growth. Traditional methods of elderly care are varied but are either not efficient or numerous enough to expand coverage to everyone in need. Challenges to any avenue of care include assisting those with decreased mobility and dealing with an increased prevalence of neurodegenerative disease. Those afflicted with these diseases face serious challenges and will often require continual assistance for their daily routines.

III. DESIGN IDEA

Our design idea is a storage and retrieval robot focused on reducing injuries and assisting those with neurological diseases and other disabilities. These afflictions can potentially make storing and retrieving every-day objects a life-threatening act. Our primary user group is the elderly population, as they most often face these challenges. We chose to focus on this small problem so our design solution would be distinct and simple to use. Ease of use is an important factor in being accepted and used by the elderly population, who are commonly hesitant to rely on technology for day-to-day activities.

The rising demands for assistance are predicted to cause an increase in the cost of care, both economically and socially. While many robotic aids exist on the consumer market, many are targeted as general-purpose family companions, with overtly broad features and functionality. It is our belief that our focused product can serve as the cornerstone of bridging gaps in existing care solutions.

In this segment, our team addresses the current issues relevant in elderly care, and propose our solution, a valet robot. We will discuss in-depth how the robot is intended to work, and we provide measurable metrics of the robot. These measurable metrics are detailed in four major aspects: graphical user interface, lifting arm, pathing, and finally the chassis and electrical system.

A. Addressing the Problem

Our solution is an object storage and retrieval robot that uses line pathing to bring items from a user’s common sitting location to a desired storage space and back. This addresses the problem of injuries caused by falling by reducing the number of times the elderly person will need to stand up and walk to retrieve their items, which poses a falling risk multiple times a day.

We wanted our design to be a clear, simple solution for a single problem that elderly people face. We believed this solution would be more effective than larger scale solutions that try to address several different problems. These larger scale solutions tend to be confusing for someone who is technologically limited. The elderly population is less likely to learn and effectively use these solutions. Home automation systems are not often marketed specifically towards the elderly population, and therefore are not designed with ease-of-use being the driving factor behind the design. Our design features a simple and intuitive Graphical User Interface (GUI) with audio cues. This simplicity was chosen to keep the learning curve of our robot low enough for anyone to learn. Because our design is easy to use and marketed towards the elderly population, we believe it is more likely to be accepted and effectively utilized by this group.

B. Robot Function

Our robot's function is a simple loop of storing, cataloging and retrieving items that are placed into a basket that can be attached to the arm of the robot and placed into storage cubbies. Once an item is placed in the basket attached to the robot's arm, the user can prompt the robot to store the object by selecting the button corresponding to the destination and cubby. To retrieve a stored item, one taps on the photograph of the item they wish to fetch. The photograph is taken by a camera mounted to the robot during the initial storage of the object. The robot will begin at the starting point near the user, travel to the cubby location to place or grab a basket, then return to the start point for both storage and retrieval operations. A more detailed list of operations will be included below. The arm of the robot has two hooks on the end which latch onto the basket when lifted and unlatch from the basket when lowered into a cubby. It operates in a forklift-like manner, driven by a stepper motor and controlled by Arduino code with limit switches that guarantee precise and reliable operation. Our first prototype used a lever arm implementation with a stepper motor, adding unnecessary weight. The current arm saves weight and operates with greater precision. The arm is not user-controlled but instead prompted by markings in the path it follows. The path of the robot is created using a tape path that contrasts with the ground. Using four infrared (IR) sensors to monitor this contrast, the robot detects if it is veering off course and redistributes power to the wheel motors for correction. It is advantageous to use a line-following implementation versus a pre-defined path because modifying the path layout can be done without modifying anything in the pathing logic code.

C. Features and Measurable Metrics

For our design to solve the societal problem, we needed to implement several major features. The main features of the physical build are the GUI, pathing, mechanical arm and drive motors and electrical/power system. Features required for the project management portion of the design course include reports, timelines, presentations, periodic status updates and meetings. Our team created a list of measurable metrics to gauge if we met our design idea contract terms. In this section, we will comprehensively review these metrics from both semesters of work and reflect on what features we satisfied or failed to deliver.

D. GUI and User Experience

Crucial to the success of computer-driven technology with older adults is usability. If it takes someone a month to use a product that is supposed to make their life easier, it is likely that they will lose confidence in their ability to rely on such a complex device. When creating the device idea contract in the first semester, we brainstormed different ways a user would be able to control the robot. Voice control, a GUI, and even a remote controller were all user input control options considered. We ultimately decided on using the GUI method of input. We understood that if we carefully designed the interface, it would only take a few buttons to operate the robot. We choose large input buttons and graphics in our implementation. We also include pre-programmed audio cues that explain what buttons have been pressed and the operation the robot is currently performing. The alternatives require hard to remember voice commands or having enough coordination to learn the operation of a specialized remote control. Furthermore, the technology behind voice commands and remote controlling gets expensive if you want it to be reliable. Alternatively, the GUI uses a relatively low-cost touchscreen with the low-cost Raspberry Pi micro-computer we were already using. Having the input come from a touchscreen also gives us virtually unlimited options of input methods. If it is possible to program it to appear on the screen and execute, it can be included in the GUI.

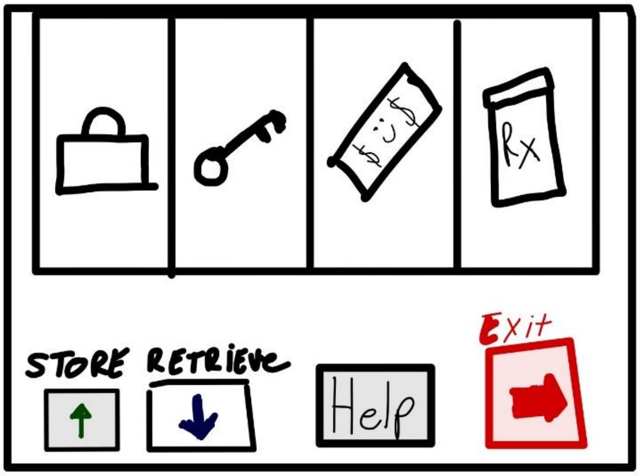

Figure 2. Initial Concept of the GUI [17]

The finished GUI displays a catalogue of 4 dynamic buttons which is hypothetically expandable based on the amount of storage you have at the destination. These buttons will initially have the number of the corresponding empty cubby on them. When a user wants to place an item in a cubby, they place the item inside the robot's basket and click one of the four empty buttons. The button will then change to show an image of whatever is inside of the tray.

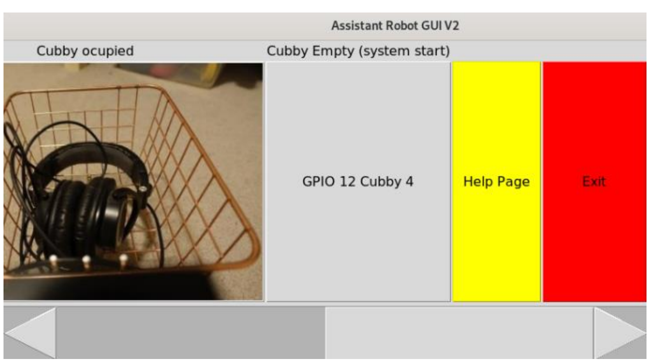

Figure 3: GUI program showing headphones stored in cubby three [17].

A web camera attached via USB to the Raspberry Pi will take the picture of the item automatically. When the user wants to retrieve an item, they just click on the catalogued image of whatever object is desired. The GUI has a two-way line of communication with the microcontroller controlling the robots pathing, arm and motors. The sends the signal telling the robot to begin storing in one of the four cubbies, or to retrieve from one of the four cubbies.

This method of input meets our two measurable metrics for the GUI included in the design idea contract. The first one is: *there will be an interface that the user will use to tell the robot which item it will like to bring back or put away in the shelf. User interface is driven by a photographical array of stored items. User selects an item to retrieve by selecting the relevant photograph.* And the second one being: *the robot will be able to record the items put away in the inventory. The robot does not know exactly what the items are specifically but will know which shelf the item is in and be able to inform the user via photograph of the item through the user software interface. The photograph is taken by the robot itself. These photographs drive the inventory control system. System informs user of inventory status.*

E. Pathing

Our goal from the start was to implement pathing in an easy and cost-effective manner. We decided that infrared (IR) sensors following a physical path would be the best implementation for our design. As discussed previously, a major benefit of the infrared line-following method is that the path can be rearranged easily. We acknowledge that there are some downsides to this implementation. The most obvious downsides being the tape path is an eyesore and not easily reconfigurable for somebody with mobility constraints. We feel that for the functionality the pathing provides, these downsides are relatively minor.

Our first measurable metric for the pathing was: *our robot will be able to travel along a black line path autonomously and reach the endpoints of the path. the endpoint of our path will be marked by a shelving unit and a predetermined starting point which represents where the user will interface with the robot. the path will be marked with duct tape or a similar material. the surface it* *will be able to travel on is a flat white surface, like riverside hall's tile flooring.* The first component we completed was the infrared sensors detecting a black tape line. This was a simple code that output "BLACK LINE" every time the infrared sensor LED turned off. Calibration required some adjusting of the infrared sensors' potentiometers. This was done four times make sure that each IR sensor worked. The next step was to make sure the DC Motors moved the chassis based on an input. Base code was written to move the DC Motors with the H-bridge component and drive the chassis forward, backward, left-forward, and right-forward for testing purposes. Finally, our newly developed logic control software integrated these components, along with this line following schema to create our line-following robot. Please refer to Appendix C-2 for this algorithm's logical flowchart. Essentially, there are four cases that exist when pathing. Both infrared sensors in the front either are both ON, left ON/right OFF, left OFF/right ON, and finally both OFF. Based on that logic, the DC Motors will either path the robot forward, towards the right, towards the left, or stop altogether.

Our second measurable metric for the pathing is: *the robot should be able to stop if it detects something in front of it. If it no longer detects something in front of it, it will proceed with the previous movement.* If someone or something is to collide with the robot, the robot will stop. It can continue moving along the path once the obstruction is removed.

F. Lifting Arm

While the lifting mechanism is relatively simple, its proper and accurate function is an integral part of the system. If the lifting mechanism fails to properly hook the basket and returns to the user without the basket, it may dramatically disrupt the normal function of the robot. Returning the robot to normal functioning would then be required by the user, which is not optimal, or the user would have to wait for someone else to come and fix the robot for them.

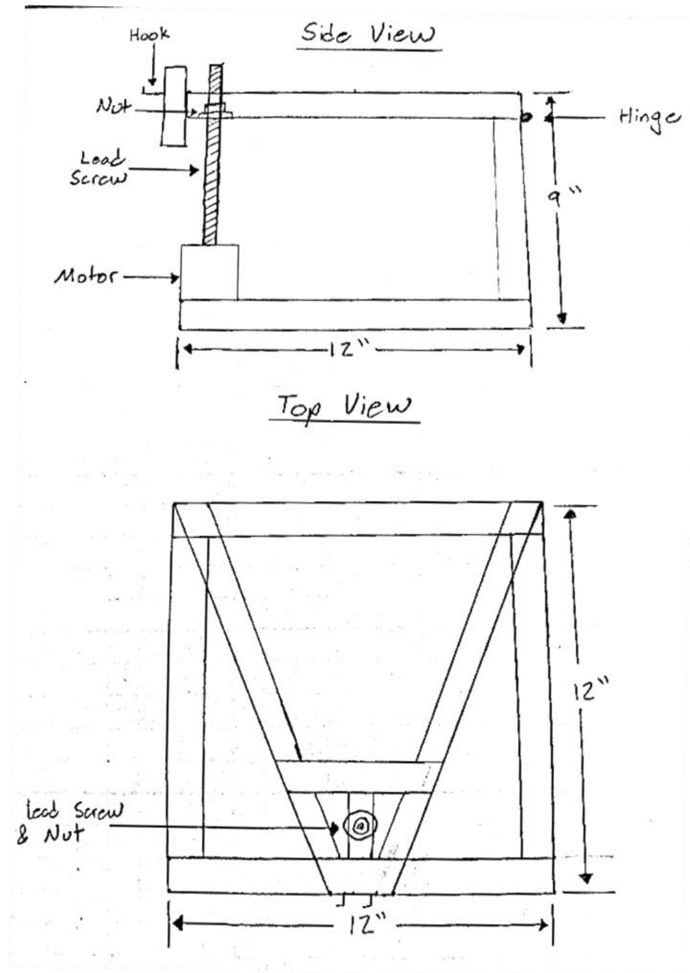

Figure 4. Lifting Arm initial blueprint [17]

The initial design consisted of a hinged lifting arm with hooks on the end. The arm was designed with a groove for the lead screw of a stepper motor to pass through. Two nuts on the lead screw were bound together with a piece of wire. When the stepper motor rotated, the nuts would push up or down on the hinged arm, which would raise and lower the arm. The arm worked for our laboratory prototype, but we wanted to improve it for two reasons. The first reason was because the design looked too hobbyist. We believed our design was clever, but it would be an eyesore if we tried to market it in its current state. The second reason we desired a redesign was the lack of feedback in the system. We relied entirely on the stepper motor lifting and lowering the arm the same distance indefinitely. While stepper motors are accurate, they will eventually misstep, and our robot would eventually fail to lift the basket and require a manual realignment of the height of the lifting arm.

With these design requirements in mind, we built the second design of our robot. We eliminated the bulky hinged lifting arm and attached the hooking mechanism directly to a linear stage actuator. We also added an adjustable limit switch to the lower end of the linear stage actuator's movement. This allows us to decide where the lower limit needs to be based on the height of the basket. When in use, the limit switch allows the system to remain constantly aligned and will make it so the system never needs to be manually adjusted. The second iteration of our lifting mechanism design is more compact, visually appealing, and powerful, providing a tangible upgrade over the first design.

G. Chassis and Electrical System

EASR consists of two 24VDC encoded drive motors which are driven by a L298N Dual Channel MOSFET and controlled by the main microcontroller (Mobility System; Logic) via PWM. The electrical system is considered as part of the chassis. It is driven by a single 24V/2A DC battery, the TalentCell PB240A1. The use of a battery pack, vs the earlier 24V DC adaptor, enhances the robot's mobility. The power is distributed as such: 24V bus attached to the main L298N driver, DC-DC Buck Converter, and TB6600 microstepping controller. The DC-DC Buck Converter has been adjusted to output 8V, which feeds into the Arduino's (Mobility System; Logic) DC input jack. While the USB link to the Raspberry Pi also provides 5V, the extra load on the Raspberry Pi's power system would require a more capable 5V regulator for the Raspberry Pi and would introduce another main failure point for the system. The 5.5mm VDC input jack is regulated by the Arduino's own LDO, which is slightly less efficient vs outputting the Buck Converter directly into the 5V rail of the Arduino. However, the Arduino's 5V rail also serves as the logic reference level for the L298N and TB6600. Furthermore, this Buck Converter-- >LDO setup allows for slightly tighter voltage regulation and helps suppress minor transient voltage spikes and voltage ripple from the DC-DC Buck Converter. The TB6600 drives the NEMA23 stepper motor. Although were no measurable metrics directly related to the chassis and electrical system in our design idea contract, every operation of the robot relies on a dependable build and power system to run. Significant workhours were put into designing, testing, and redesigning this portion due to every feature's total reliance on it.

IV. FUNDING

We received no outside funding for our project, instead using materials that we already had and paying for the rest out of pocket. The following table (Table 1) details all parts utilized directly by this project (ancillary tools such as laptops utilized in the software development, documentation, and report generation, are not included). Finally, we show the total amount of money spent on the project.

Table 1. Bill of Materials

Item	**Source**	**Quantity**	**Price Per Unit ($)**	**Order Cost ($)**
Arduino Mega UNO R3	Amazon	1	15.23	15.23
Plywood, ½" thick	Home Depot	2'x4'	12.00	12.00
Wheels	Amazon	4	0.75	3.00
DC motors	Amazon	8	12.00	96.00
IR sensors	Amazon	4	1.12	4.48
2"x4"x8' Douglas Fir Lumber	Home Depot	2	3.34	6.68
1"x2"x8' Whitewood lumber	Home Depot	1	4.96	4.96
3/4"'x4'x8' Sanded Plywood	Home Depot	1	28.18	28.18

Everbilt Utility Screw Hook	Home Depot	2	1.18	2.36
Everbilt M4-0.7x25mm Machine Screw	Home Depot	2	0.48	0.96
Everbilt M6-1.0x25mm Machine Screw	Home Depot	4	0.51	2.04
TB6600 4A 9-42 V Stepper Driver	Amazon	1	12.59	12.59
Rattmotor 100mm Travel Length Linear Stage Actuator	Amazon	1	109.89	109.89
Logitech C270 Webcam	Amazon	1	17.49	17.49
SmartPi TouchScreen holding case.	Amazon	1	24.99	24.99
7" Official Raspberry Pi Touch Screen Display	Amazon	1	64.00	64.00
Raspberry Pi 3B+	Amazon	1	40.00	40.00
Total	$444.85			

V. PROJECT MILESTONES

We marked specific points throughout the project timeline in which we aimed to complete major features and components of our project. These specific points, referred from here on out as milestones, were useful in gauging the completeness of our design. To keep track of these milestones, we created and regularly updated a Gantt chart using Microsoft Project. These major milestones are listed into two sections. Section A includes all major class assignments and documentation required throughout the design process. Section B includes all the major milestones reached building the robot and adding functionalities.

A. Class Assignments and Documentation

The project was conceived and developed within the CSUS Senior Design course. The major class assignments were:

- Design idea contract
- Work breakdown structure
- Project Timeline
- Risk management report
- Technical review
- Lab prototype completion
- Problem statement
- Device Test Plan
- Market Review
- Feature Presentation
- Deployable prototype presentation (video)

B. Major Build Features

Below, we describe our full list of major build feature milestones that our team achieved in our tenure:

- Creating an interface capable of displaying four input buttons.
- Making the buttons capable of triggering output and the GUI program taking inputs.
- Adding the capability of the GUI to capture and display photos of the stored items dynamically.
- Communication between GUI program and pathing/storage algorithm.
- The addition of audio prompts to the user interface.
- 2D robot movement
- DC motor movement
- Line following

- Advanced line following
- Avoiding collisions along the robot's path.
- Assembling the stepper motor arm
- Grabbing and precision placing a basket in a cubby
- Adding the arm control logic to the microcontroller.

VI. WORK BREAKDOWN STRUCTURE

Our entire project's schedule is planned by task, duration, and expected start and completion dates. These are provided within the work breakdown Gantt chart we created and worked on utilizing Microsoft Project. The tasks consist of the major milestones listed in Section V of this report. Included in this section is a summary of the hours worked in total by each team member on each task. This summary will be further divided by each major feature. To reiterate, the major features of our project are the GUI, pathing, mechanical arm, and the drive motors and electrical/power system. Also included will be all our class assignments including documentation such as the design idea contract, technical reviews, and device test plan. Our work on the project was mostly done modularly. Each member focused most of their efforts on one of the four major features of the design and then integrated them with the other features in the last phases of the development.

A. GUI

The GUI was developed in an incremental manner. Multiple builds of the GUI were created, with different features and appearances. This rapid prototyping allowed systems functionality testing before building the final interface finished development.

1) Interface Capable of Displaying Four Input Buttons

This was the initial work on creating the GUI. To complete this feature, we had to research the different GUI programs that were available to us. We eventually chose Tkinter, the GUI building library packaged with Python. Researching the various GUI programs such as Kivy, PyQT and Tkinter took about 10 hours. Once we decided on using Tkinter, it took about 15 hours of research and development in order to create a clean interface with 4 buttons. This work, a total of 25 hours of research and development was all done my Alexander Leones.

2) GUI Input and Output Capabilities

After creating the 4-button GUI program, we had to add functionality to the buttons and make it capable of taking input. This allowed us to add more functionalities later such as dynamically changing button images and buttons that change functionality based off the robot's current state. We ran into some bugs such as double outputs and freezing which took a few more hours to fix. In total, this portion of the project took about 24 hours of work to complete.

3) Displaying Images on the GUI

A major feature of our GUI program is that it's able to display an image of the item that have been stored in a catalog so users can easily remember and select the belongings they've stored or would like to retrieve. Adding this functionality took finding a reliable image capturing program, proper image file storage and updating, and of course, dynamically updating the GUI during runtime with the most current images of items stored away. This feature took about 30 hours of work to complete. Alexander completed a bulk of the work, about 26 hours, with Jeremey researching and assisting in the early stages of work for about 4 hours. This task took a considerable amount of time to complete because countless hours were spent on correcting any error or display issue we came upon. The GUI also had to be updated and optimized to display images in a way that is easy to scroll through and see. Even for someone without particularly good eyesight.

4) Communication between the GUI and the Pathing controller

Almost all interaction with the GUI controls the movement of the robot. The users must be able to rely on the 2 components to reliable communicate without the need of adjustment or troubleshooting. For this reason, we chose serial communication via USB. This sends raw binary signals with error corrective functionality. This task took about 6 hours total to implement. Alexander and Jeremy both worked 3 hours each on this task.

5) Audio Prompting

One thing we wanted to make sure of throughout the design was that as many people as possible, especially the elderly, were able to use our robot. We decided to add audio prompts confirming every action the user takes as a way to assure they do not command the robot to perform the incorrect operations due to lacking the ability to see the GUI screen or simple user error. This part took 2 hours to research how best to implement audio in a Tkinter program and 1 hour to create the audio prompts and integrate them into the program. After 2 hours of testing and optimizing, this task ended up taking 5 hours total. This work was all done by Alexander.

B. Pathing

Developed in two stages. Angel created the initial state machine and pathing concept. The pathing was initially done using black tape lines and two IR transceivers. It was a basic line following robot, with a state machine controlling the logic.

C. Lifting Arm

The Lifting arm was designed with input from the entire team and built primarily by Jesse with assistance from Jeremy. Each iteration of the lifting arm was built and tested individually before integration with the rest of the system.

1) Physical Build of Lifting Arm Frame

The lifting arm frame was fabricated out of wood. In order to build it without the robot present, we aimed to have the footprint of the frame match the footprint of the chassis. The frame was a 12”x12” square with an A-frame type lifting arm attached to hinges 9” above the lower frame. The stepper motor used to raise and lower the hinged A-frame arm is threaded through a metal channel attached to the arm.

2) Integration of the Stepper Motor

Once the frame was completed, we added the stepper motor with test code to ensure our design would function as we needed it to. We used a simple test code created using Arduino’s stepper library and ran the motor with an L298N H bridge driver. The stepper motor functioned, although it was loud, and the motor driver ran at a surprisingly high temperature.

3) Integration of lifting arm with robot

Although the function of the lifting arm was not ideal, we decided to perfect it after integrating it with the robot. We screwed the frame of the arm down to the top of the robot and eliminated the Arduino that tested the arm by moving the pins and code to the Arduino that controlled the robot. At this time, we realized that the Arduino Uno that we were using did not have enough I/O pins to contain our entire design and upgraded to an Arduino Mega. We also solved the heat problem by switching out the L298N H-bridge driver in favor of an A4988 stepper motor driver, which is more ideal for this application. This driver solved our heat problem and allowed the stepper motor to run smoother and quieter. With Jeremy’s help, the code was added to the state machine and revised to allow the stepper motor to move up and down an ideal amount that properly raised and lowered the basket. While we were revising the code, we attached the hooking mechanism Jeremy built from wood and metal hooks to ensure that the robot was able to properly hook the baskets. The lifting arm was now prepared for our fall senior design showcase.

4) Second Design of Lifting Arm

After our first semester of senior design, we recognized some shortcomings in our lifting mechanism. The main flaw was a lack of feedback in the system. With the lack of feedback, we were relying entirely on the stepper motor traveling the exact same vertical distance indefinitely for it to be able to hook the basket every time it runs. We recognized that the motor mis-stepping occasionally would eventually cause the lifting arm to be too low or too high to hook the basket properly and would require manual realignment by the user. The other shortcoming that we wanted to address were that the current arm lifted on a hinge instead of directly vertical, and the design was bulky and looked too “hobbyist” for a professional project. These were not necessary to fix, but helped our robot be more marketable.

5) Physical Build of Updated Lifting Arm

The physical build of the second lifting arm was much simpler than the first. We replaced the stepper motor with a linear stage actuator (LSA). The LSA utilizes a stepper motor and translates the rotational motion of the stepper into linear stages. We used the linear stage actuator’s built in mounting channels to attach a piece of wood vertically along the back of the LSA, then attached a piece of wood to the bottom of the vertical piece that lies horizontally along the chassis of the robot. We then built a hooking mechanism very similar to the first iteration. We attached a piece of wood to the lifting part of the LSA, then attached two screw-in hooks along with a smaller piece of wood to account for the angle at the vertical part of the basket. The final part of the build was to add a bracket on the side rails of the LSA to hold the limit switch. This bracket was made from wood and integrated in a way that we could adjust the height to allow us to set the lower limit when we begin the testing of our final system.

6) Testing of LSA and Limit Switch

Because the LSA still utilizes a stepper motor, the testing was similar to the first design with two key differences. The first change was the addition of a TB6600 stepper motor driver. This driver is designed specifically for higher power stepper motor applications. With some research on this motor and driver being used together, we decided to forgo the use of Arduino’s stepper library and manually specify the pulse width, rate, and duty cycle. This allowed for fine control of the stepper motor, resulting in smoother operation of the motor. The second major change was the integration of the lower limit switch. This was done with a simple change in the code. When the motor was being lowered, we had it run until contact was made with the limit switch instead of having it run for a set distance.

7) Integration of Second Lifting Arm

With a functional test code and a previous iteration of the lifting arm code already in the main system, the integration of the second lifting arm was relatively straightforward. We first attached the lifting arm system by screwing it onto the top of the chassis, replacing the first lifting arm system. We then wired the system onto the main Arduino and the updated power system. We then replaced the old code from the first lifting system with the new test code that was designed for the second lifting arm. The system was now ready for the final testing of the lifting arm.

D. Chassis and Electrical System

The chassis would be built on a wooded plyboard, 12” by 18”, and was built by Jeremy Shaw in September, with small tweaks into October and November. The electrical system was also developed by Jeremy in parallel with the chassis. Minor tweaks were made throughout the project as goals became more clearly defined.

1) Early development

The chassis design and build out started early. High mobility on a flat surface and a small overall size were two key considerations. This resulted in the utilizing a 3-wheel differential drive system, as commonly seen on smaller commercial robots, such as the iRobot vacuum. This 3-wheel system would have all three wheels axially aligned and positioned in a tripod orientation. The two main drive wheels would be on a “rear” axle (perpendicular to the primary direction of travel), slightly behind the center of gravity for the robot. The rear “axle” placement allows the independently driven motors to control the direction of the robot using differential drive. The front wheel was mounted to a 360º caster at the very front of the robot.

Other common arrangements in this class of robot were the tail dragger (inverted tripod, with the unpowered 360º wheel at the rear) and a pure differential drive system, with multiple bogies and tracks. The former was considered, however the overall length of the initial prototype (12" wide, 18" long) and the dynamics of having the line reading sensors on a long polar axis, allowed for finer control of the robot's movement. This was important when it came to the main sensor-logic feedback loop.

The latter system, using a tracked propulsion system was also considered. Since our project development process started with a large amount of research and project development paperwork, this team had time to interact with members of other teams from the prior semester. Conversations with these other teams ultimately revealed the these tracked systems had greater mechanical build requirements, along with a larger number of unique and proprietary parts, which were not readily available on the US commercial market. Given Team 1's composition and inexperience with mechanical engineering, the decision was made to proceed with the 3-wheel arrangement, with two rear drive wheels.

There exists a bit of rough napkin math, when it comes to scaling up a mobile design, specifically relating to motive power vs the weight of a design. For flying objects, double the weight may require eight times the power. For wheel vehicles, it may be a bit less, but the ballpark estimates are still relatively accurate. Early prototypes for this project, had failed to account for this, and vastly underspecced the entire motive system (drive motors and electrical network). This led to a constant cycle of reevaluation and significant increases in the power budget of the design, growing from a 6V chassis with 6V drive motors to a 24V chassis with 24V drive motors.

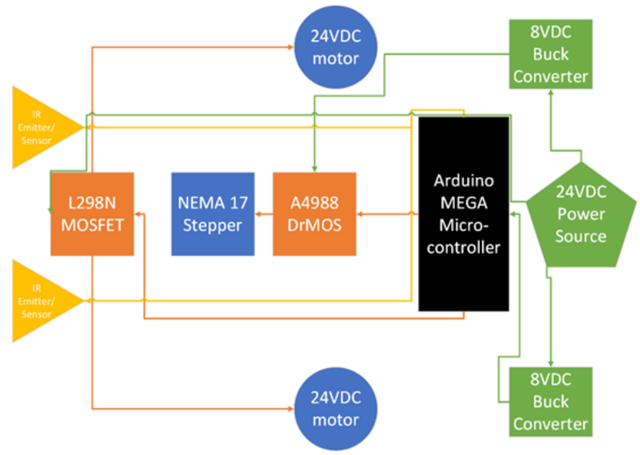

Figure 5. Early Electrical Schematic (obsolete) [17]

2) Preparation for 1st semester Senior Showcase

The semester traditionally ended with a Senior Showcase, where projects from 1st and 2nd semester Senior Design teams would present their projects. The showcase formed a natural deadline for the project, especially when it came to a presentable prototype of some sort. This required rapid integration of the logic and chassis' systems. Most of the integration and testing work was done in a single 3-hour session before the technical review meeting. The work relied on rapidly incorporating feedback from short test runs, and significant progress was made in that time frame. This would ultimately prove to be a recurring story throughout the rest of the project, where brief periods of project integration had provided much of the forward progress of various components of the project.

One major aspect not yet covered, is the integration of the microcontrollers. The initial work distribution of this project had siloed out early development work of each main feature to a member of the project team. The immediate result was 3 different microcontrollers for the various mechanical and logic aspects of the project. The main control logic and chassis were integrated into a single microcontroller early in the project, but the timing of the lifter arm's development had resulted in a second microcontroller remaining in the project. Earlier plans for the project had called for a much higher GPIO pin count. This was due to some uncertainty on the final controller arrangement, since the goals of the project were heavily driven and refined by the feedback we had received from our advisors and our internal testing. The night before Senior Showcase, against the better advisement of our Professors, the Logic/Chassis controller was integrated with the Lifter Arm controller. This single action tested the stability of the Logic code's main state machine and slightly simplified the electrical layout of the chassis. The integration proved successful and remained in use throughout the remainder of the project.

3) 2nd semester developments

After the senior showcase, the 2nd semester saw a renewed focus on delivering the rest of the project's feature list. Pertinent to the chassis and electrical system, a major, implicit requirement (though never explicitly listed) was battery power. The 24Vdc requirement of the chassis' electrical system eliminated several popular and simple battery layouts from consideration. Several previous and current teams were consulted on their approaches to this aspect. The project advisors also had appropriate feedback towards the power/weight ratio of successful projects from previous semesters. Ultimately, there were three primary choices for the robot's battery power system. These were a dual 12V sealed lead acid battery, wired in series. A custom battery pack made from Li-ion cells. A commercial 24V Li-ion battery, made from the Li-ion cells. Difficulties in attaining the appropriate battery cells and charge state controllers eliminated the custom battery pack from consideration, and the unfavorable power to weight ratio of the serial lead acid batteries - and unfavorable results from previous semester teams - ultimately discouraged that system's inclusion on this project.

The remaining option, a commercial battery pack, was the most expensive option and had the most limitations when it came to capacity (shipping regulations limit effective battery pack size to ~100Whr) and load limits (many regulated packs had low limits on their power outputs). The TalentCell 24V PB240A1 was chosen for its unregulated 29V-21V, ~3A output and ~82Whr capacity. The high voltage and current limits allowed for a margin of error in the project's development.

4) Final Chassis and Electrical considerations

The chassis and electrical systems were originally planned as a temporary development, with a soft plan to properly integrate the systems into a more cohesive, easy to modify, and aesthetically appealing design. This would include a centralized 24V power bus. Voltage converters were to be physically mounted in proximity to the systems which required them. Better labeling and color coordination of the wires and wire harnesses were planned, to avoid the potential of dangerous misconnections. Some aspects of this were partially implemented. A central harness was created and used for the primary 24Vdc rail. Signal and power wires were largely color coordinated between certain systems.

Unexpected impedances against productivity, due to many unforeseen complications stemming from the vast scheduling conflicts and physical distances between every team member, in addition to a massive, terminal campus closure for the remainder of the project, lead the temporary solution to become the final deliverable. This does not impact the quality of the project, since the wooden chassis with a bevy of wires, controllers, MOSFETs, and motors, was always intended to be a development prototype which successfully demonstrated all of Team 1's project goals and requirements.

E. Cubbies and Cubby Stand

The cubbies and cubby stand were built primarily by Jesse with assistance from Alex. Because the design of the cubbies was determined by the size of the basket, the design was straightforward.

1) Cubby Design and Build

The purpose of the cubbies is to hold four baskets that will hold the items that the user chooses to store. The dimensions of the baskets we used are 11" in length, 8.5" in width, and 5" in height. Our design aimed to fit the baskets with a slight wiggle room for any imperfections in our pathing design. We chose 9" x 9" x 9" for each cubby. This allows enough horizontal wiggle room for a slight error in our pathing, enough vertical room for the basket to be lifted out, and the depth allows the baskets to stick out slightly, allowing the robot to interact with the baskets without the cubbies interfering. The cubbies were built in one day with ¾" plywood and held together with 2" nails. The cubbies were well made enough for our design but could have been improved with the use of a table saw or circular saw instead of a jigsaw (for straighter cuts), and the use of screws instead of nails for a tighter fit and added stability. To seal off the back of the cubbies, we cut a heavy-duty cardboard sheet to size, and secured it to the back with a staple gun.

2) Cubby Stand Design and Build

The cubby stand was not initially planned, however when our first-semester robot design was ready for testing, we realized the cubbies were too short to be effectively used in their current design. We measured and decided that the cubbies would need to be around 9 inches higher to be effectively used and set out to build a stand that would allow this. As the build was done last minute, we decided on a simple stand made entirely of 2" x 4" wood. This stand has the same footprint as our cubbies, then has four 9" legs screwed into the side of the stand to give the cubbies the desired offset. The cubbies were set on top of the stand, and this design allowed us to properly test our completed robot.

VII. RISK ASSESSMENT

Risk assessment is a critical part to every engineering project. Risk mitigation became incredibly prevalent during our second semester. Our second semester of senior design coincided with the COVID-19 outbreak and pandemic. This was a profound, unprecedented, and unexpected complication to the workflow of our system design, but a great test of if our mitigation plan worked or not.

In this section, we will describe the risks we associated with our design. This risk will be understood through asking and answering four questions, and finally we will address if our risk mitigation worked. Risk assessment is based on four 'What If' questions: *What could go wrong? What is the likelihood that it would go wrong? What are the consequences?* And the risk mitigation question*: What can be done to reduce or eliminate risks?*

There were four major aspects of the project itself that were deemed critical to the success of the project. Implementing Robot Pathing, the Robotic Arm, the GUI, and the chassis itself. A risk assessment matrix allows for a visual presentation and analysis on the likelihood of an event vs its impact on the project's success.

On this "matrix," the horizontal axis represents the lowlihood of an event (either failure or presence) on our project, while the vertical axis represents the magnitude of that event on our project.

5			Robotic Arm		
4					
3				System Logic Pathing	
2		GUI			
1					
	1	2	3	4	5

Figure 6. Risk Assessment Matrix [17]

A. Implementing Robot Pathing

Implementing a pathing algorithm can present a multitude of issues during the development phase of our project. We imagined one complication that could spring up when implementing line-following was a misunderstanding of the algorithm. In addition to this, the inability to make a pathing algorithm work as intended could have set our project back. Another risk worth mentioning is a critical failure of any of our components, which would force us to order new components in a short amount of time. And finally, the last issue that may be prevalent is an inability to cohesively work as a team to put together the pathing for the robot. We will now discuss the likelihood of each event we calculated during the early phases of design. The likelihood that we will misunderstand the algorithm is medium to high.

This is something that is expected because we have no experience implementing line-following and will have to climb the learning curve if this becomes the main issue. The likelihood that we will have a complete inability to implement a pathing algorithm is low. The likelihood that any of our components will have a critical failure is low. The likelihood that our teamwork will hinder our chances of moving forward is medium to low.

We will now discuss the potential consequences in these events. If the algorithm is misunderstood or misapplied, the progress for moving forward with the project will be slowed as we attempt to implement the algorithm as intended. If any of our components have a critical failure, it will set our project behind schedule as we wait for replacement parts to come in. The consequence of not being able to work together as a team would be that our project will be incomplete due to miscommunication between team members. The final section is dedicated to risk mitigation of these potential risks. If our team misunderstands or misapplies the pathing algorithm, we can teach ourselves several possible algorithm techniques or educate ourselves by asking someone who has successfully implemented the pathing algorithm. If a component experiences critical failure, we can mitigate some risk by ordering multiples of a single component within our budget. In order to mitigate the risk of having team discord, our team should aim to be transparent and to continue to work closely with our project advisor when we encounter an issue.

B. Robotic Arm

Our previous design lifted the baskets using a stepping motor with an 8-inch lead screw pushing up on a hinged arm. The arm was mostly made of wood, and the hinges were two small 5/4" metal door hinges. The overall design was an extended c shape, with a piece of wood (2-5 inches depending on final force calculations) separating the base and arm to allow room for the stepping motor and lead screw.

This early had many potential failure points. The wood and hinges did not allow for precise movement. Failure of a hinge itself or its attachment to its wooden supports was likely, however the consequences of these weaknesses were minimal. We have extra door hinges and extra wood that we can use to quickly and easily replace a failed component. Failure of the wooden arm is possible. It can break due to weight or warp due to external conditions. The consequences would be that we would have to source new wood to replace a failed wooden arm. Another potential failure was the NEMA 17 stepper motor. The NEMA 17 was for the most part dependable but was slightly underpowered. A failure of the motor led us to buying a new motor. This was a preventative measure taken to avoid future failure of the lifting mechanism.

While our updated lifting arm is an overall improvement over the first, it still has some risks associated with it. With the addition of the linear stage actuator, the only wooden part directly associated with the lifting is the bracket of the hooking mechanism. With the way the weight is distributed on this mechanism, we do not believe the wood would fail unless it was exposed to extreme moisture and was severely warped. The LSA and updated stepper driver motor produced much less heat than the first design and we do not believe overheating will be an issue, but there is always a possibility of a random failure with these components. This could be slightly mitigated by only buying professional grade components, but this is out of our budget and the mitigation would be minimal, as these products would still be capable of random failures. The final risk associated with this updated design is the wooden mounting brackets. While stainless steel brackets and fittings would be an obvious upgrade, it is more difficult and requires more specialized tools to work with, and we did not have the budget or tools to make this happen. With the expected indoor conditions and forces that we defined our robot to work under, we do not expect the wooden design to pose an issue, but there is clearly more risk involved while using wood instead of aluminum or stainless steel. We believe our second design has mitigated the risks that were involved with the first design and has an acceptably low chance of a failure that would disrupt the normal operation of the system.

C. GUI

Tkinter, a Python library included within the Raspberry Pi's Raspbian OS, drives the main user interface. This Python package is a critical part of the system's design as it is the only way for the user to control the behavior of the robot. Python is a very versatile language. It is interpreted as it runs, versus a language such as c which is pre-compiled into low level machine code. This added the low-level risk that the Python GUI, if not coded efficiently, may be less responsive than intended. The robot need not perform actions in perfect real time speeds, so we felt that using python on a Pi would more than likely work exactly as intended. To mitigate this potential risk, we included and responsiveness in the list of things we tested for.

Another possible risk with using Python and Tkinter is that we would not be able to implement every single operation control. The two main potential causes of this shortfall were the potential lack of a software function necessary to carry out a user-controlled operation and our inexperience with Tkinter disallowing us from implementing all our intended features correctly by the deadline. The possible consequences included having a GUI not intuitive to use, having limited functions than promised, or worse case having a buggy GUI that does not work with our system control logic. To mitigate any of these potential issues, we focused on building a GUI that works flawlessly and de-prioritizing things such as aesthetic design and other parts that will cause overcomplication. All features we promised in the design idea contract were delivered, so our mitigation efforts seemed to work.

D. Chassis and Electrical System

The larger risks for the Chassis and Electrical system all centered around its functional abilities. These intertwined systems would serve as the central framework for the project, and thus became a critical bottleneck and gatekeeper. A major step taken in mitigating the risk of falling behind during systems development was to silo individual components and attach those components to individual members of the team. Development would then be able to happen in parallel while the chassis and electrical systems were being brought up to speed. While this would somewhat complicate integration work, it did allow for the different developmental rates and stages of each system to progress on their own and allowed for the Chassis and Electrical System to freely evolve to meet its own goals and requirements.

This sort of development is not without risk, however. The risk is further magnified by the physical and temporal distances between individual team members, with living situations and work schedules heavily impacting the ability for this team to work together, reconcile developments, and integrate the various systems of this project. In short, it makes project progress much harder to regulate and judge, since measurable project outcomes were largely only testable after systems integration.

E. Reflecting on COVID-19 Complications

Professor Tatro lectured our class on the potential effects a natural disaster could inflict on our projects. The semester before our first semester as designers, Sacramento State closed for eight days as a result of heavy smoke from the Camp Fire. Professor Tatro urged us to take precaution and have mitigation measures planned out for another closure of campus, mostly in case of a fire. He shared that teams during the fire could not access their builds as the laboratory storage they stored them in was completely locked up during the campus closure. Our team prepared for a temporary closure of this nature by discussing who would take home what parts of the build in event of a campus closure. We also discussed how we would work remotely, or where we would meet if campus was not an option.

In the team's last semester of the system design, Spring 2020, the team was cautioned again by Professor Tatro about the, then seemingly remote, possibility of a campus closure due to the coronavirus outbreak in China and parts of Europe. As it became clearer to us that the closure was in-fact eminent, we followed our mitigation plans and brought all of robot components home with us. Campus closed as expected, but we did not expect or plan for such a long term of social distancing. Thankfully, our build was in the finishing stages when we decided it was unsafe for us to meet as a team again, but much testing and continued integration of components still had to be done.

In retrospect, an even better mitigation method would have been to continuously integrate and perform greater testing throughout the entire system design period. The profound impact the COVID-19 pandemic has had on every part of society, including our project gives us a lot to consider going forward in our young engineering careers. We believe it calls for greater consideration of mitigation measures planned for large-magnitude natural disasters. On one hand, the COVID-19 pandemic was more impactful than anything we have seen since maybe the Spanish flu almost exactly a century ago. It makes us question, *"Is a risk this unlikely worth always accounting for?"*. On the other hand, the impact it has had on our workflow seems too great to permit us to ever overlook an outbreak or disaster of this magnitude again. Our ultimate takeaway is that from now on, we must consider and plan for the worst-case scenario, not matter how unlikely the causing risks may seem.

VIII. DESIGN PHILOSOPHY

The objective of this project was to develop a system capable of moving small objects around a roughly room-sized area. This is a somewhat grey area for existing commercial and academic suppliers, largely due to the rather unproved market for general assistance robots for personal use. Our philosophy was that we could fill this grey area with our best attempt at a robot that functions somewhere in between the commercial and academic designs that have come before our attempt.

Another factor which influenced our philosophy was the target users. While larger goals were present in the discovery process, the addressed Societal Problem favored a simpler user interaction model. This would limit the complexity of the project's interface while increasing the amount of development and testing work to catch edge cases. This drove the design away from larger, more complicated options and towards smaller systems that were dependable, intuitive for all to use, and well-engineered.

Regarding workload and delving out the tasks, we decided that we would work on each of the major features individually eventually reaching a point where every team member has a fully developed feature ready to be integrated into the final robot build. Our philosophy behind the decision to modulate the work was that every member would become an expert in their own respective feature, then they would be able to quickly teach the other team members about their expertise during integration. In practice, this allowed us to greatly increase the amount of research and design we could do by decreasing the number of multiple people working on the same thing at one time.

IX. DEPLOYABLE PROTOTYPE STATUS

A. Capable of taking a (1kg max) object within the wire basket.

Object is defined to being capable of fitting within the physical confines of the wire basket without leaving any mass behind.

B. Capable of storing that basket and object within a designated storage space.

The success of storage is defined by having the basket and object remain inside the correct cubby after the robot has departed.

C. Capable of retrieving that basket and objected from that designated storage space and returning it to the starting position.

Success is defined by the presence of the selected wire basket and object being retrieved from the cubby and returned to the starting position.

With these three main aspects completed, the deployable prototype is considered operational.

X. MARKETABILITY FORECAST

The development of a new product requires succinct research into the intended market to ensure success. In Fall 2019, our team came up with a product that intended to fill a niche market. The product was developed to solve a societal problem involving elderly care. In this report we will present the projected market data for our product and analysis of this data.

Our team is currently developing the Elderly Assistance Storage and Retrieval robot. The EASR is an autonomous robot that uses wheels, digital sensors, and a graphical interface. This allows for users to give the robot items for it to store and retrieve. Our product intends to service those who have limited mobility or suffer from a cognitive disability. This is especially true in the elderly population.

In this report, we will discuss the current market for our product. We present research into the current industry outlook for our product, our intended target audience, as well as market share competitors. These statistics will be used to further support the need for our product in the market, from a business perspective.

We feel that the uniqueness of our robot and its ability to amply solve the niche of storing and retrieving personal items makes it a strong chose compared to other similar products we have researched. The target user group is diverse and growing rapidly, which shows a positive trend of marketability for the future of our product. Our pricing would also be much more reasonable versus hiring a caretaker or buying a trained service dog.

A. Current Industry Outlook

The EASR. is a device that is intended to service the elderly population, especially those with disabilities. When looking at the current industry outlook for Americans who currently or who will require assistance, there is clear growth.

The projected rise in average American age show that, in addition to understanding that approximately 41.7% of those aged 65 and over report to suffer from a disability shows a need for a product to service this population

B. Target Consumers

Our original report suggested that our target consumer would be adults over the age of 65. While that is our primary audience, we will not limit our marketing to solely that audience. The American Institute of Physics reports that 15% of adults over the age of 18 have difficulty functioning physically. [15] While the mobility-impairing disability rate in those aged 65 and older is around 40% according to t

C. Market Share Competition

1) Service Dogs

One way the societal problem is already being solved with is the use of mobility assistance dogs. These dogs are specially trained to perform various tasks. One of their main tasks is usually retrieving out-of-reach objects. Service dogs are well trained and excel at their jobs, but not everyone needs this level of assistance. Especially when a service dog has the upfront cost of 15, 30 and even sometimes 50 thousand dollars. Our robot will only compete with the specific task of storing and retrieving items for the users. The robot should be desirable for those who only need help with that specific task.

According to ShareAmerica, a website run by the U.S. Department of State, there are currently an approximate 500,000 service dogs helping people in the United States currently.[16] If our robot can serve as an alternative to a fraction of those cases, we will have a decent market for our product.

2) Professional Medical Care

The two main types of professional medical care are nursing homes and full-time in-home caregivers. Both have advantages over our robot since humans are much more diverse than our robot, able to assist the user in multiple ways, and able to assist in the event of an emergency. The main disadvantage of this type of care is price. The U.S. Administration for Community Living estimates a semi-private nursing home room to cost $82,128 per year, an assisted living facility to cost $43,536 per year, and an in-home health aid to cost $20.50 per hour, which full time adds up to $42,640 per year (8 hours a day, not including weekends)[E]. If our robot could reduce the time need for an in-home caregiver, it would result in money saved for the user. Another major disadvantage of professional medical care is the loss of independence and privacy. This is difficult to back with data, but we have spoken with elderly family members who have stated that they strive to remain independent for as long as they can.

According to a study from the National Center for Biotechnology information, up to 29% of adults over the age of 65 who receive Medicare benefits receive either in home or out of home assistance [6]. The breakdown is shown by the figure above. Clearly adults prefer in-home care rather than living in a nursing home. This helps with our audience, because nursing home residents and those who receive full-time in-home medical care would not have any need for our robot.

3) Crowdfunding Robots

The popularity of this concept is reflected in the large number of robot assistants, butlers, and home companions on crowdfunded and crowd equity platforms. Several large campaigns (Aido, BUDDY, Alpha 2), which have spread across multiple platforms and surpassed their public funding goals. Smaller robots with more limited, health-focused scopes (Pillo, Autom) have also found their audiences online. All of these demonstrate a growing demand for in-home, personal care.

Kinetics, somatic kinematic studies, etc. Mobility solutions do have some impact on the market for this device, though whether they are complementary, supplementary, or invisible to one another is less well studied and unknown to us.

XI. CONCLUSION

Team 1 began this two-semester project with limited experience in truly utilizing teamwork in a larger project like senior design. Over the last eight months, we slowly developed and perfected the way that we best work together. This project was helpful in cultivating our somewhat lacking communication and documentation skills that will be required from us when we graduate and begin the transition to our careers.

The societal problem we covered is the increasing number of elderly people in the world. The elderly populations face mobility problems and have a high rate of injuries due to falling. Taking such a broad societal problem and narrowing it down into a focused, specific solution was a major challenge for us, and was the stage of our project that required the most discussions, disagreements, and compromise.

With our societal problem locked in, we were able to move on to what we were all most excited for, beginning our design. Our Design Idea Contract stated exactly what our project would do, and how it would do it. The biggest problem we faced was finding the balance of what we should promise in our feature set. If we promised too little, our project would be too easy, and unimpressive. A grandiose feature set, or even general feature creep, could jeopardize the project's success, leaving the team unable to fulfill the promised design as intended, and far reaching consequences for the project and the team. With help from Professor Levine, the feature set was refined into a reasonable, challenging target for graduating seniors at CSU, Sacramento.

We then began our Work Breakdown Structure, which is a living document that will define when we complete our work, and how each team member's completed work will interact with each other. Because the flow of our work was dependent on other teammates, this document required us to trust one another, and manage our time well to ensure that we stay on track to keep our project moving along. We were able to modularize our project, and each team member was able to work on a portion of the project that they had an interest in. We were able to utilize both our electrical and computer engineers and give them tasks that they were more comfortable with based on their previous experience. This document broke down our entire project into individual tasks and subtasks that made it seem much more achievable and helped us with our time management that we were struggling with before this document was created.

Every project comes with a possibility of something disrupting the normal workflow, from something as small as a resistor burning out to as large as a global pandemic. We planned for these risks in our Risk Assessment Report. This report focused on everything that we believe could go wrong in our project. We mainly focused on parts breaking, as we believed those would be most likely in our project. We discussed and estimated each risk that was associated with something going wrong in our project, and how impactful that risk would be to us being able to complete our project on time. One of our Senior Project advisors, Professor Tatro, warned us about the Camp Fire and its impact on the senior design classes in fall of 2018 and encouraged us to include large natural disasters that would shut down school, and our ability to access anything we had stored on campus, on our report. Because of our Risk Assessment Report, we had a plan in place when the COVID-19 pandemic reached the United States, eventually restricting our access to campus and our ability to meet up and collaborate on the project. Our Risk Assessment Report ended up being more useful than we expected when writing it, and one of the most impactful documents of our project.

This document then discusses our design philosophy, which was heavily influenced by our target user. The elderly population are often limited with their technological skill, so our design was focused on ease of use and high visibility. The users will interact with our robot via its GUI. The GUI design was completely driven by our target user and resulted in a more minimalistic take on user interfaces. The interface was not text heavy, but instead relied on imagery and intuitive design to make learning the operation as simple as possible.

The deployable prototype has been completed, with each metric that we promised in our feature set. This was an important accomplishment for our team, given our forced transition to remote-only work in March.

As the team neared the completion of the EASR project, some attention was diverted to perform analysis on what the current and potential market for EASR was. We focused on our competition and believe there is a market for our robot with elderly people who do not require constant human care and just want to automate some of their day to day activities. We are not able to find a robot like ours that is proven to be effective and well-received.

In conclusion, by the metrics established in this report, the EASR project has succeeded in addressing the societal problem of increasing demand on care solutions (i.e, for elderly persons). The team members collaborated to make a project which will aid the elderly population, allowing older people to extend their independence for as long as possible.

REFERENCES

[1] "Fact Sheet: Aging in the United States – Population Reference Bureau", Prb.org, 2019. [Online]. Available: https://www.prb.org/aging-unitedstates-fact-sheet/. [Accessed: 21- Sep- 2019].

[2] U. Bureau, "Population Projections Data", The United States Census Bureau, 2020. [Online]. Available: https://www.census.gov/programs-surveys/popproj/data.html. [Accessed: 27-Apr- 2020].

[3] “2018 Profile of Older Americans,” acl.gov, Apr-2018. [Online]. Available: https://acl.gov/sites/default/files/Aging%20and%20Disability%20in%20America/2018OlderAmericansProfile.pdf[Accessed: 21-Sep-2019].

[4] “Poverty Among Americans Aged 65 and Older,” Congressional Research Service Report, Jul. 2019.

[5] G. Bergen, M. R. Stevens, and E. R. Burns, “Falls and Fall Injuries Among Adults Aged ≥65 Years - United States, 2014 | MMWR,” Centers for Disease Control and Prevention, 23-Sep-2016. [Online]. Available: https://www.cdc.gov/mmwr/volumes/65/wr/mm6537a2.htm.

[6] Caregiver Action Network. 2020. Caregiver Statistics. [online] Available at: <https://caregiveraction.org/resources/caregiver-statistics> [Accessed 27 April 2020].

[7] Evans, D., Pester, J., Vera, L., Jeanmonod, D. and Jeanmonod, R. (2015). Elderly fall patients triaged to the trauma bay: age, injury patterns, and mortality risk. [online] Www-sciencedirect-com.proxy.lib.csus.edu. Available at: https://www-sciencedirect-com.proxy.lib.csus.edu/science/article/pii/S0735675715006002 [Accessed 21 Sep. 2019].

[8] “HCUPnet,” HCUPnet. [Online]. Available: https://hcupnet.ahrq.gov/. [Accessed: 21-Sep-2019].

[9] C. S. Landefeld, “Goals of Care for Hip Fracture,” Archives of Internal Medicine, vol. 171, no. 20, Nov. 2011.

[10] “2019 Alzheimer's Disease Facts and Figures,” alz.org, 2019. [Online]. Available: https://www.alz.org/media/documents/alzheimers-facts-and-figures-2019-r.pdf. [Accessed: 21-Sep-2019].

[11] “Alzheimer's Disease Fact Sheet,” National Institute on Aging, 22-May-2019. [Online]. Available: https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet. [Accessed: 21-Sep-2019].

[12] “How Much Care Will You Need?,” longtermcare.acl.gov, 10-Oct-2017. [Online]. Available: https://longtermcare.acl.gov/the-basics/how-much-care-will-you-need.html. [Accessed: 21-Sep-2019].

[13] E. James, W. Gellad, and M. Hughes, “In This Next Phase Of Health Reform, We Cannot Overlook Long Term Care,” Health Affairs Blog, Mar. 2017.

[14] R. Schulz and J. Eden, Families caring for an aging America. Washington, DC: The National Academies Press, 2016.

[15] “Physical & Mobility Impairments: Information & News.” Aip.org, 1 Aug. 2016, www.aip.org/sites/default/files/history/teaching-guides/accessibility/Accessibility_Physical & Mobility Impairments Reading.pdf.

[16] M. Trainer, “Service Dogs Save Lives,” Share America, 30-Sep-2016. [Online]. Available: https://share.america.gov/service-dogs-save-lives/

[17] J. Shaw, J. Aucelluzzo, A. Smith-Evans, and/or A. Leones, 2019-2020.

GLOSSARY

A4988 - a single channel DrMOS which takes a PWM input to smartly sense and drive two stator coils, implicitly in a stepper motor configuration.

Arduino - a family of software-unified microcontrollers, all utilizing the Arduino IDE (itself a derivative of the Wiring IDE) and its attendant C variant (commonly called Wiring, after the original IDE). The EASR ultimately specified an Arduino Uno MEGA R3. Several other Arduinos, including devices not designed by Arduino AG, were examined or utilized in the development of this project.

DrMOS – short for Driver-MOSFET. A package which consists of a MOSFET and some logic. The purpose of the logic is to provide a feedback/control loop for the MOSFET, and to provide current sensing/limiting. Note, DrMOS is Intel terminology. Intel's DrMOS specification (version 1.0) was last updated on November 2004.

Encoder Motor - a motor with a hall-effect sensor (optical encoders also exist) measuring the rotational speed of the motor and sending the data back to the microcontroller.

GUI - Graphical User Interface, a method of allowing an user to interact with the values and variables of a machine in a visual manner.

LDO – Low DropOut regulator. A MOSFET used to reduce the input voltage to a lower, stable output voltage.

L298N - a dual channel MOSFET used to take logic level inputs (5V) and output drive power to 1 or 2 DC motors.

Linear Stage Actuator – a linear actuator, capable of being utilized at any stage (linear position within the travel range of the linear actuator). The systems used in this project translate rotational motion into linear motion, providing a significant mechanical advantage, relative to the power requirement. The severe mechanical advantage of the screw allows the linear actuator to passively maintain the stage of the linear actuator, with minimal electrical demand.

MOSFET - Metal Oxide Semiconductor Field Effect Transistor, a silicon device utilized to give a range of voltage outputs against a range of corresponding voltage inputs. In this case, generally used as a voltage booster and voltage regulator (LDO).

NEMA17 / NEMA23 - two common stepper motor specifications.

PID - Proportional, Integral, Derivative. A control schema which provides smooth outputs in response to sudden, staccato inputs.

Raspberry Pi – A small computer with various GPIO and other interfaces (Wi-Fi, Bluetooth, USB host, Ethernet, I2C, HDMI). The diminutive form factor (86mm x 53mm footprint) and low cost (35USD, without the necessary mini SD card and 5V [regulated] power source) make for an ideal higher-level controller. It does not require a separate computer to develop software for, unlike the Arduino, since it is a fully-fledged computer by itself. Runs a 32bit Linux distribution called Raspbian. Our project used a Raspberry Pi 3B+ running Raspbian Wheezy.

Appendix A. User Manual

User manual organization

- Parts included "in package"
- Safety precautions
- Requirements
- Setup (of room)
- Setup (or robot)
- Setup (of cubbies)
- How to use (demo 2 cubbies)
- Help

- Parts included "in package"

  Robot

  Wire baskets

  Cubbies (optional)

- Safety precautions

  Requires an adequate space to setup and run. Also utilizes an Li-ion battery. While sealed, it still poses a risk when mishandled.

  The robot is also low to the ground and can only see in the direction of its travel, so it may still pose a tripping hazard.

- Requirements

  120VAC outlet or 29.4VDC power source, to charge the battery.

  ~10' x 4' minimum operating space is required.

  Hard, flat surface floor. Preferably with high contrast to whatever material the guiding line is made of. When the robot is on and in its default state, the two green LEDs at the front will indicate any detected changes in contrast.

- Setup (of room)

  The room layout (10x4 space, user location, storage location) is dependent on having the user and cubbies on opposite sides of a robot traversal area. Within this traversal area is a series of guiding paths for the robot to follow.

  Guiding path on the ground is to be setup with stop markers are the ends of the path. Stop markers are perpendicularly placed path strips, on the plane of the floor surface.

- Setup (of robot)

Robot comes pre-setup.

- Setup (of cubbies)

The baskets are to be loaded into the cubbies. They will only fit in a single direction.

The cubbies are to be facing the guiding paths on the ground.

- How to use (demo 2 cubbies)

Steps and procedures on how to use the robot to store and retrieve items, using two different cubbies.

Storage:

Once you have the guiding path, storage cubbies and robot in the starting location you are ready to use the robot. The first step is to take the item you wish to store and place it inside of one of the baskets supplied with the robot. Next, hook the basket into the arm of robot. Once secured and ready to store, click on one of the four side-by-side buttons you see on the touchscreen interface to store in the corresponding cubby. The robot will then take a photograph of the item within the basket and begin traveling along the path to the destination storage cubby. After storage, the robot will return to the start point awaiting subsequent retrieval or an additional storage operation.

Retrieval:

Once at least one item has been stored, the GUI will then show images of all the stored items in place of the button corresponding to the occupied cubby. To retrieve an item, you can scroll through the photos of items, find the one you are looking for, and click on the image to retrieve that item. Make sure that there is no basket attached to the arm during this operation or the retrieval will fail, and the robot may get bumped off track. The robot will then travel to the storage location, pick up the item you wish to retrieve's basket, and return to the start location next to you with the item. The GUI will update, removing the image of the item, allowing you to store something new in the newly vacant cubby.

- Help

If the robot stops moving, check the Arduino's user LED. If lit, it means the Robot has returned to the initial state and is assuming a resting position.

Mobility System.

The system requires 24V, ~1.7A to operate all its functionality (traction motors, logic, sensors, and lifter arm). Operational, sustained power loads are likely to be lower, since our state machine forbids simultaneous operations of the lifter arm and the main drive motors. However, this only reflects average sustained loads, and does not measure instantaneous power spikes. Battery is a nominal 82.88Whr sealed unit, with its own charge regulation and basic power regulation facilities. It requires a 29.4V/0.95A charger, which is included in the Robot's weight measurements. The robot moves at a maximum pace of ~8cm/second, when laden to its maximum load of 1kg.

Lifter Arm.

The lifter arm is a subsystem of the chassis, utilizing a TB6600 microstepping controller. This controller uses 5V logic inputs and in this robot, is configured for 24V input, 24V output (to the stepper motor). The Stepper motor is a NEMA23, affixed to an aluminum extrusion and lead screw. Attached to this lead screw is a bogey with the lift arms. This bogey is guided by the aluminum extrusion, which also prevents the bogey from rotating under the torque of the stepper motor's lead screw. This allows the stepper motor to move the bogey in a linear direction, along the length of the aluminum extrusion.

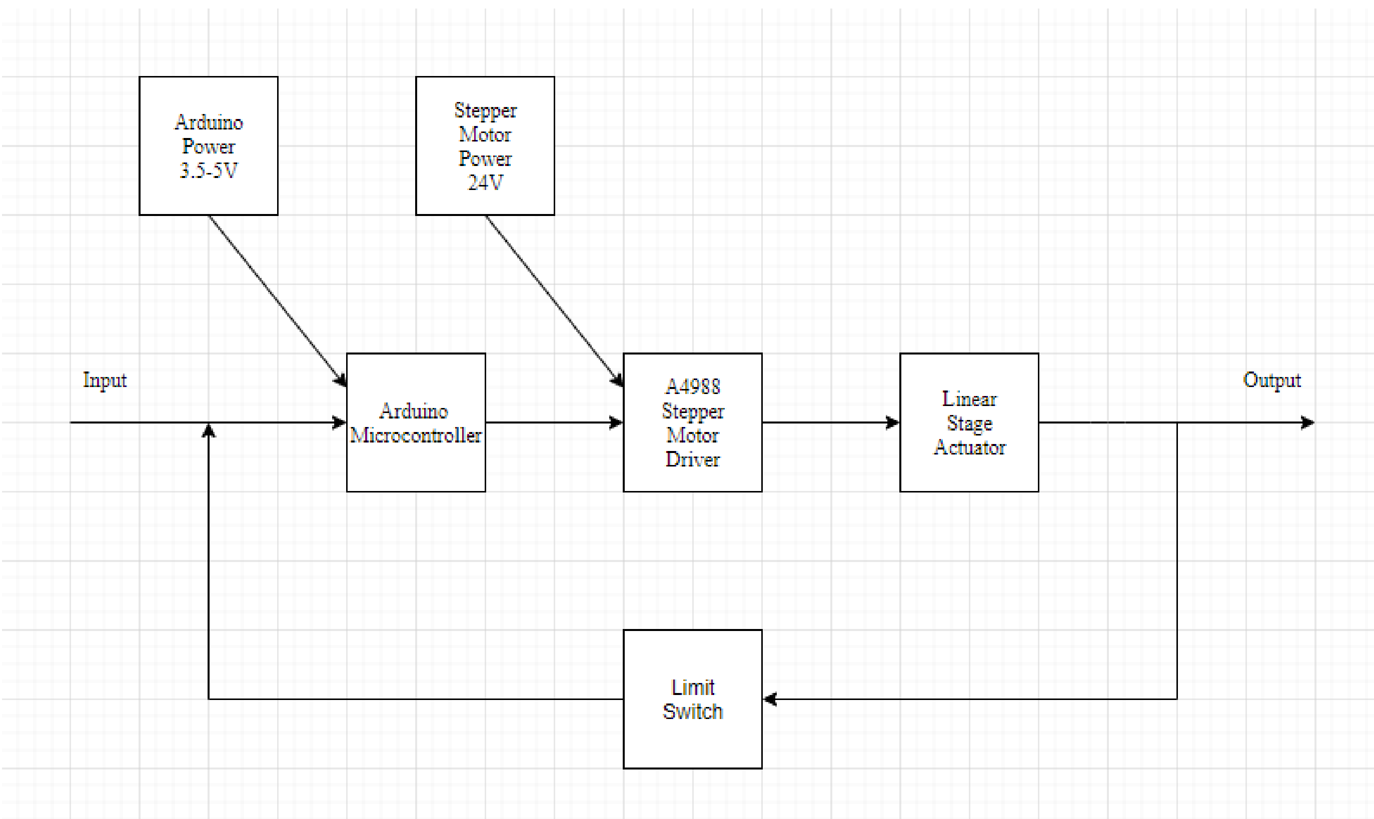

Figure B1. Lifting Arm Block Diagram with Feedback [17]

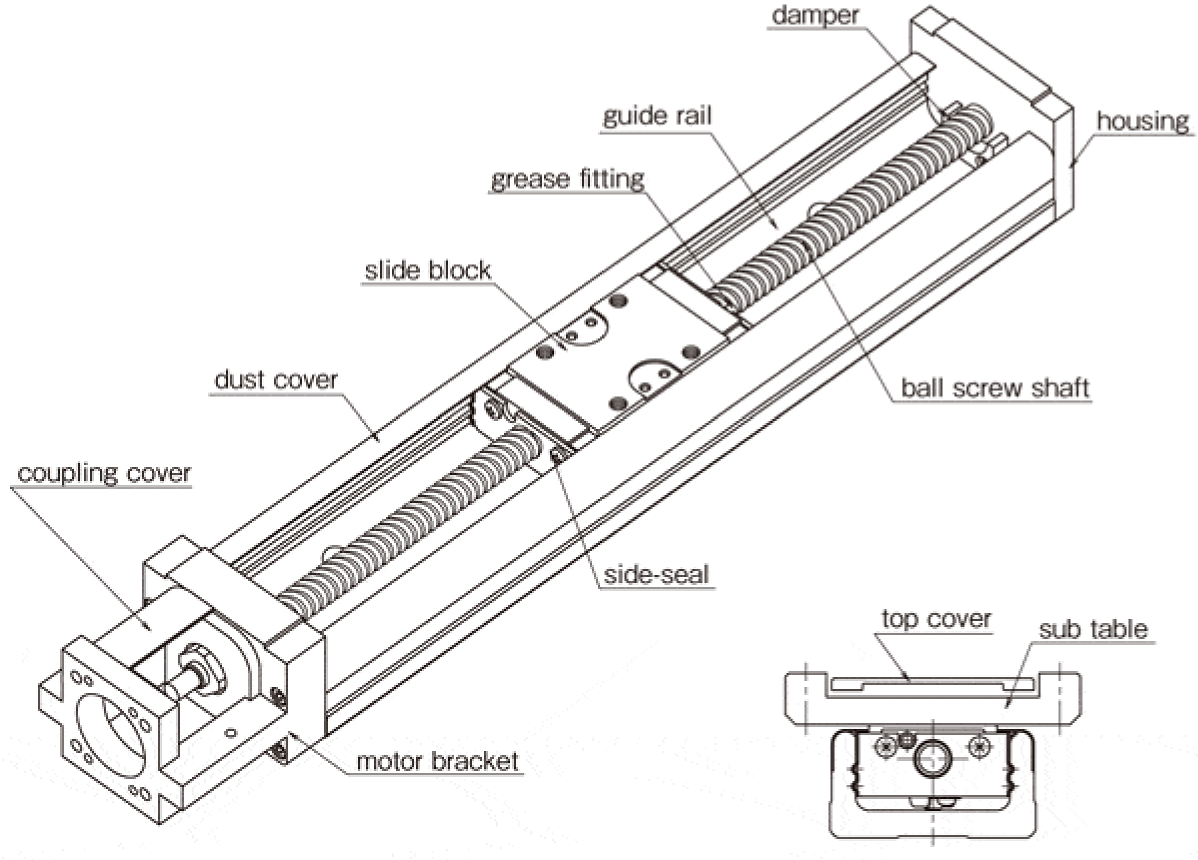

Figure B2. Linear Stage Actuator example [17]

Cubby and Baskets. Each cubby has to be around 9” by 9” with around 9” depth from the opening.

Room floor. The floor is preferably (required?) to be flat, hard, and of decent contrast to the line markers.

User location. The user is required to be at the other end of the line markers, from the cubbies.

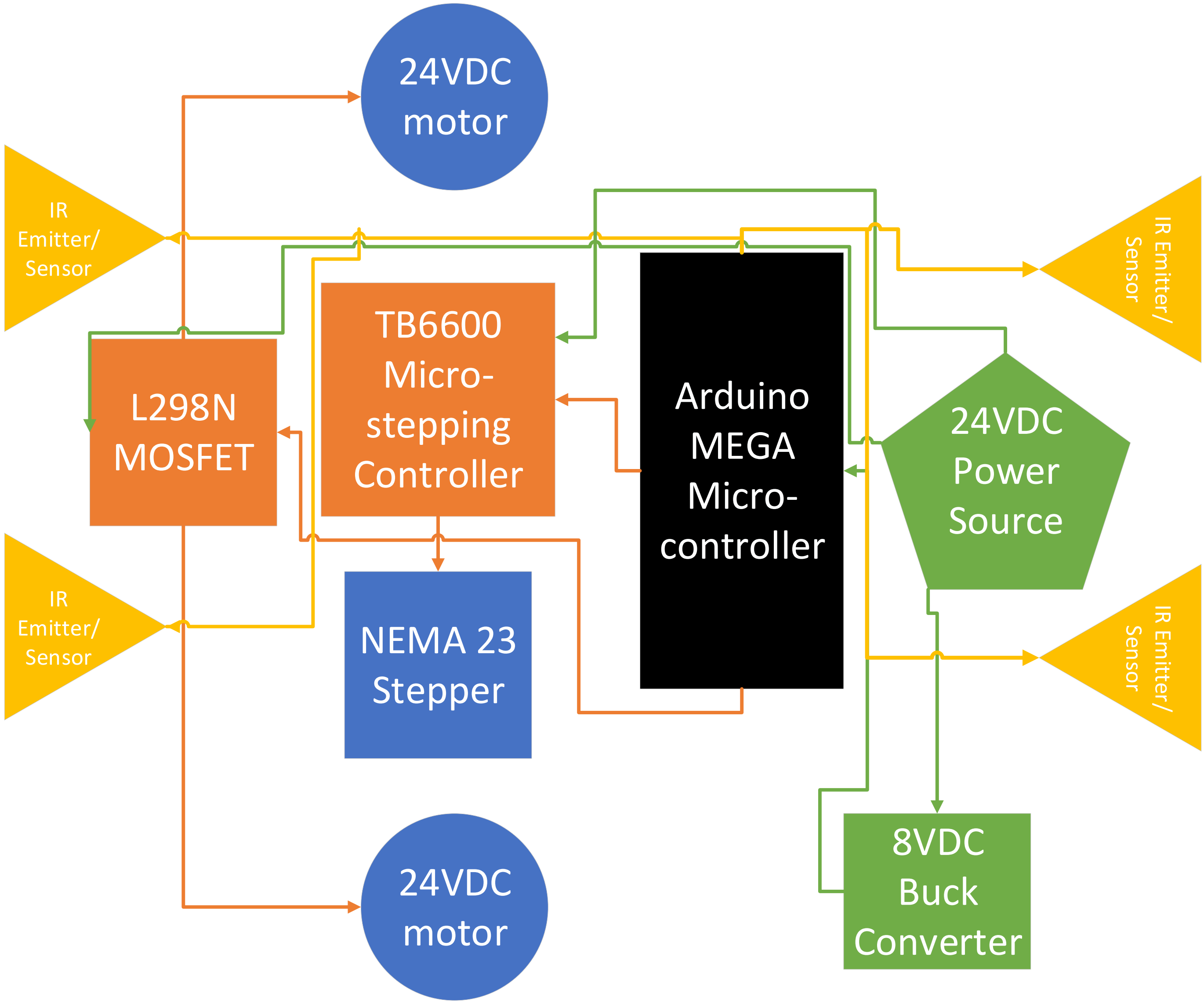

Figure B3. Chassis Electrical Schematic [17]

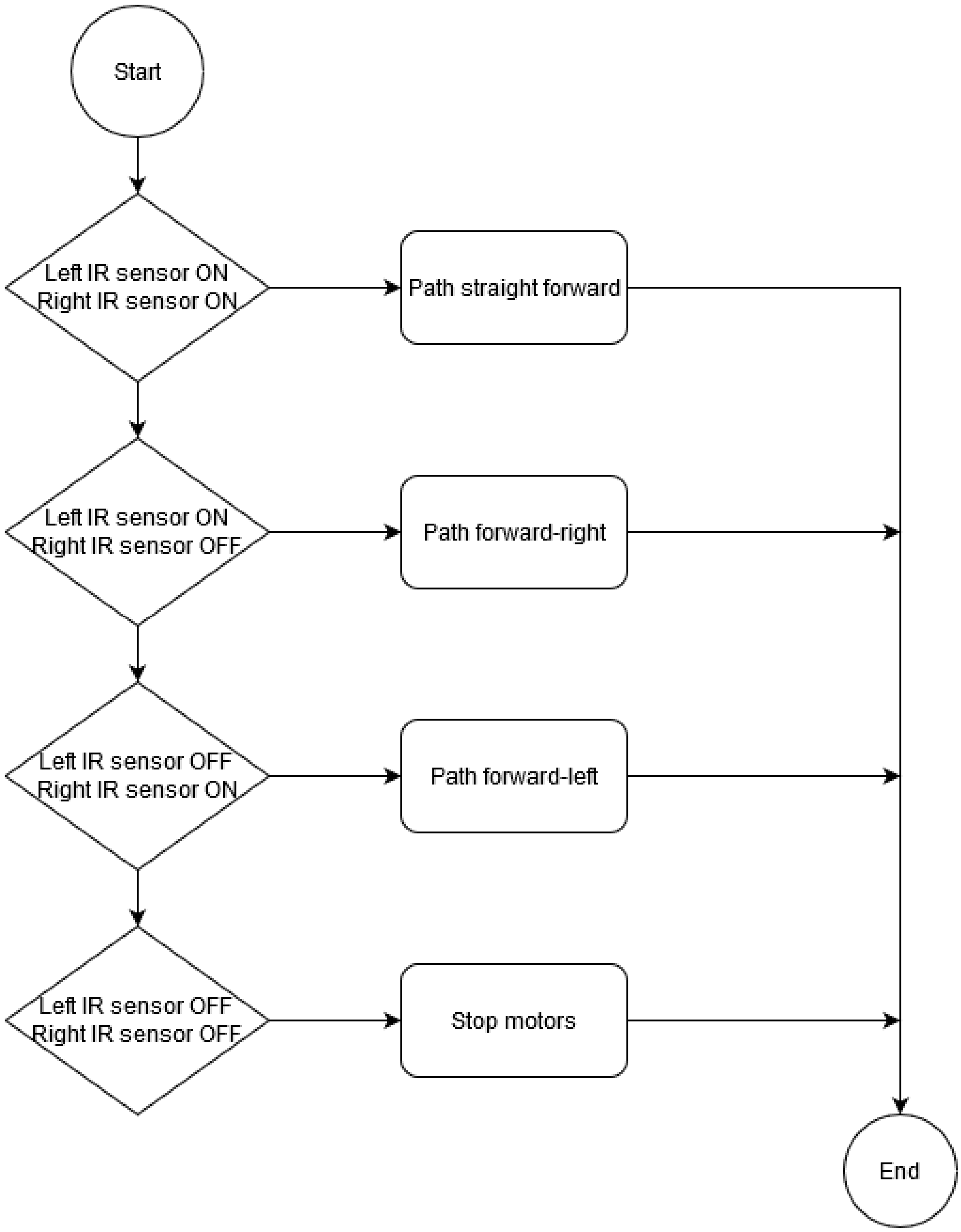

Figure C1. Infrared Sensor Line-following Subroutine [17]

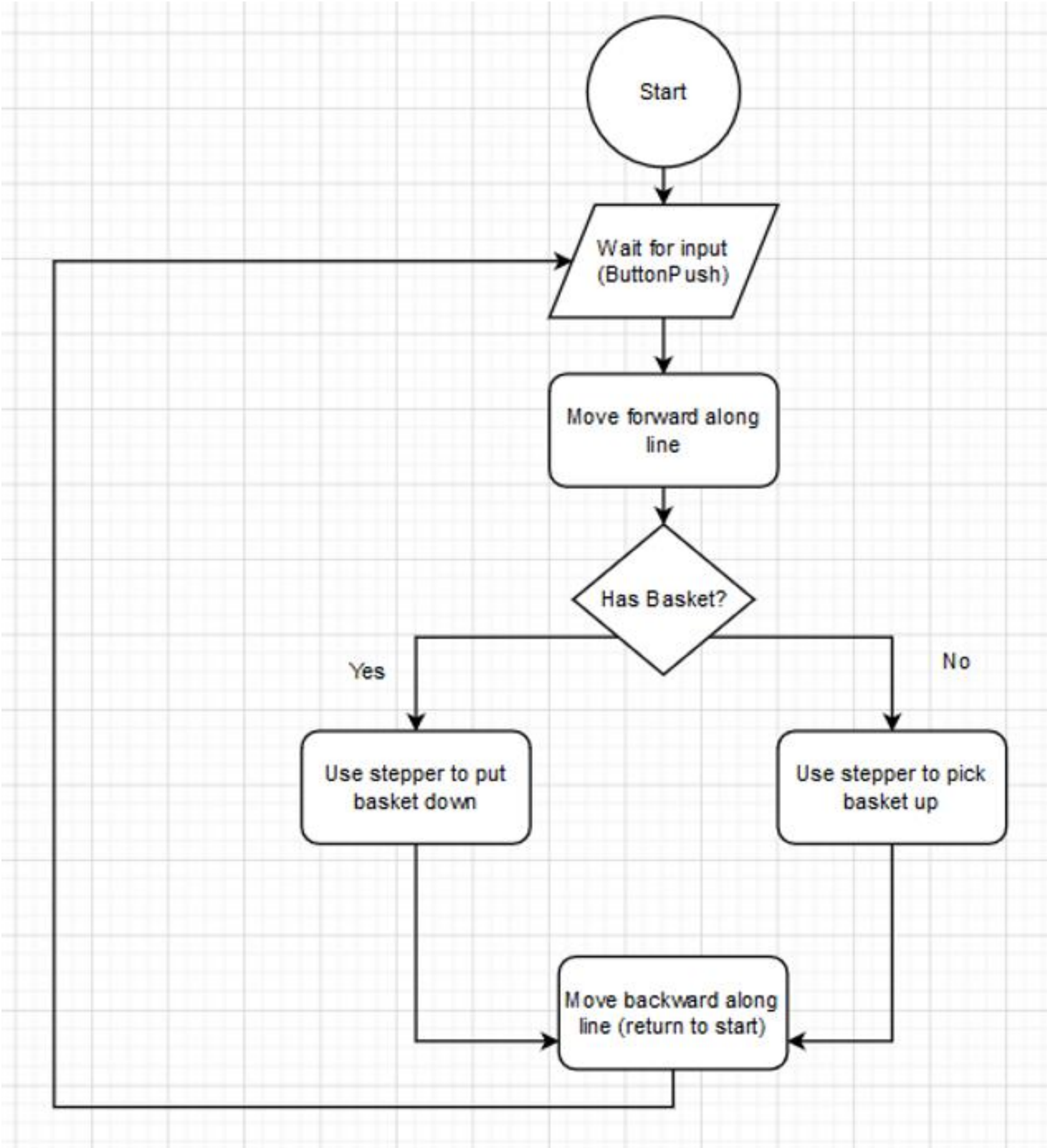

Figure C2. EASR Robot's Logic Algorithm Flowchart [17]

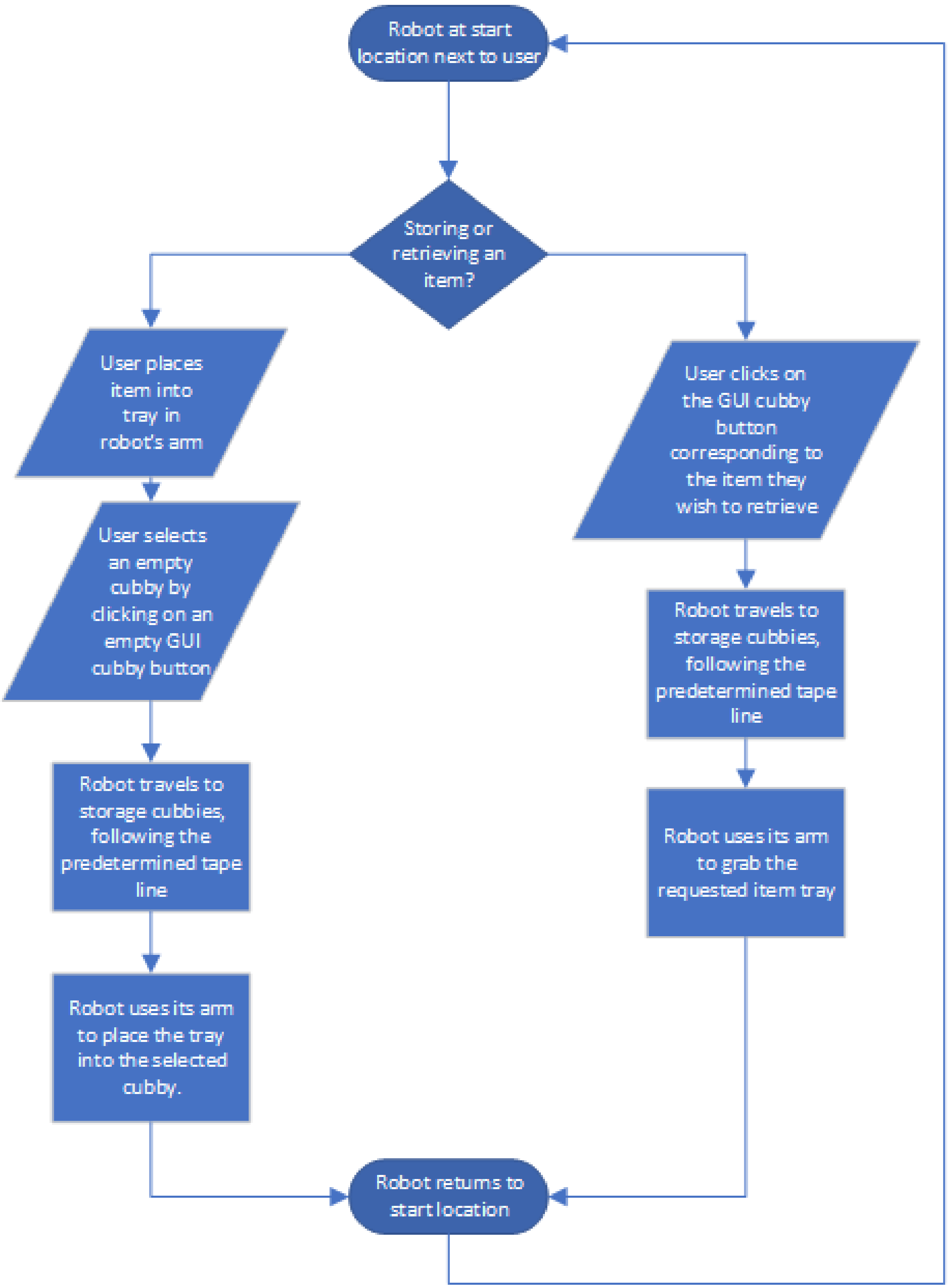

Figure C3. Logic workflow [17]

```
container = Frame(master)
canvas = Canvas(container, height=319, width= 780)
scrollbar = Scrollbar(container, orient="horizontal", command=canvas.xview, width = 64)
scrollable_frame = Frame(canvas)
scrollable_frame.bind(
    "<Configure>",
    lambda e: canvas.configure(
        scrollregion=canvas.bbox("all")
    )
)
```

Figure C4. Code snippet showing the creation of a scrollable frame using Python's Tkinter GUI building package. [17]

```
Onlabel1 = Label(scrollable_frame, text="Cubby Empty (system start)", font=25
Onlabel1.grid(row=0, column=0)
```

Figure C5. How to add a widget to the frame using grid. In this case, a label in the top left grid location. [17]

```
Cubby1StoreButton = Button(scrollable_frame, text="GPIO 21 Cubby 1",font= 25, command= partial(cubbyButton, 1, cubby1, Onlabel1)
Cubby1StoreButton.grid(row=2, column = 0, sticky=W+E+N+S )
```

Figure C6. Creating a button widget using Python's Tkinter GUI building package. [17]

Software Test Plan:

Most of the time the user will be interacting with the GUI to control the robot. The GUI must be tested for responsiveness, errors, and usability. To test the responsiveness of the GUI, a test needs to be set up in a way that measures how long the GUI takes to send and receive signals/data from the rest of the robot when a button is pressed. When testing for errors, the best thing to do is try as many different input combinations as possible in a controlled manner so we can reproduce and fix any error that we record. Usability is a focus of ours in the development of our GUI. Things to be tested for usability include the ease of physically accessing all commands, the clarity and audibility of audio queues and the visibility of all the controls on the 7” for people of all vision levels.

Our design will be tested to prevent errors from common environmental factors that could disrupt normal functions of the robot. The test area is only about 15x5 feet conservatively. The line-following tests will mostly be conducted on a well-lit white tile floor with black tape for the path. There is about one 2-inch difference between the widths of the cubby boxes and cubby spaces. The most common disruption we foresee happening are obstacles in the way of the normal path of the robot. We will be adding a collision detection system that will stop the robot’s movement before it collides with an object in its path. This will prevent damage to both the robot and the object that may be in its path. The objects we expect can be separated into two categories. The first are mobile objects. These can be pets, other people in the house, the user themselves, or other automated robots, such as a Roomba. In this case, we expect the robot to stop and give an audio cue to signal there is an object in the way. The collision detection system would continue to test to see if the object is still in its way. Once the object moves, the robot will continue its normal operation. The second category would be immobile objects, such as a bag of groceries or any other object places in the normal path of the robot that will not move on its own. In this case, the robot would stop to avoid the collision and give an audio cue to signal there is an object in the way. After a certain amount of time testing to see the if the object has moved, if the object is clearly not moving, the robot will send a different audio cue to signal that it is shutting down until the object is moved. This will prevent the robot from wasting power and sending out consistent audio cues that could be viewed as undesirable to the user. Additional status cues will be optional and not prioritized.

General Testing Steps:

1. Compile a list of all commands and inputs the GUI program will deal with.

2. Test all commands and inputs for expected behavior and make sure there are no freezes or unnecessary delays.

3. Record.

Table C1. Software Feature component testing

Software Feature Under Test	Pass/Fail	Expected Outcome
Storage buttons	Pass	When a button is pressed, an audio prompt is played, and the proper signal is sent to the microcontroller.
Audio Prompts	Pass	When a button is pressed, an audio prompt is played conveying the robot’s actions.
Pathing	Pass	When the store or retrieve signals are received, the robot paths to the proper locations.
Dynamic images	Pass	Photographs captured by the camera are displayed on the corresponding button.

Appendix D. Mechanical Aspects

Table D1. Hardware Specifications

Specification	**Value**
Maximum Rated Load	1 kg
Dimensions (2MAR20)	30 cm x 45 cm x 28 cm (W x D x H)
Maximum Power Consumption	24V / 1.7A (40.8W)
Robot Weight, unladen	3.3 kg
Robot Weight, maximum gross	4.3 kg

Appendix E. Vendor Contacts

In order of appearance:

Dr. Warren Smith, CSUS Professor (Comm. Theory & Bioengineering) – Presented us (as students in his class) with various senior design concepts, successful projects, and where those projects could have been improved upon.

Dr. Fethi Belkhouche, CSUS Professor (Robotics) – Provided some motivation and starting ground for our project.

Professor Levine, CSUS Professor (Senior Design) – Constant feedback and ideation. Thanks!

Professor Tatro, CSUS Professor (Senior Design) – Lead us down the fundamentals of system design and the holistic approach to future projects (at scale).

Professor Thomas, CSUS Professor (Senior Design) – Stepped in for Levine, thanks for the group meeting sessions.

Outside of the academic assistance of these CSUS professors, our team received no outside support for this project. All funding came from ourselves. We did not pursue vendor/professional support during this project and did not intend to.

Also, special thanks to Michael Khoo, a member of our initial senior design team and was involved in the ideation process during the Summer months before Senior Design formally started.

Jeremy Shaw

Education

California State University, Sacramento cumulative GPA 3.41 January 2017 - May 2020

B.S Computer Engineering

Honors and Activities: Dean's Honor List, Tau Beta Pi, IEEE, IPC, ACM

Experience

Caltrans - Engineering Intern Worked 20-40 hours per week July 2018 - Jan 2020

- Digitized in-class, professional development content and delivered it via an internal learning management system (Moodle) to Caltrans' licensed professional engineers and engineers in training.
- Created greater internal awareness of my office's services and mission by upgrading and updating our internal website (Caltrans Onramp - Professional Development).
- Enabled greater statewide collaborative efforts to update the manual with current procedures, standards, and financial methods by modernizing an ancient, central document (Workflow Task Manual).

Projects

JAWOS Time sharing OS for Intel x86 systems, developed in C and x86 assembly Fall 2019

https://github.com/jeremyshaw/JAWOS - cross disciplinary team

- Developed strong problem solving skills due to helping other teams; by continuously reevaluating my own coding practices and design decisions.

MicroGreenHouse Web monitored & controlled greenhouse, using rPi, Python, Arduino, & C Spring 2018

https://github.com/jeremyshaw/microgreenhouse - cross disciplinary team

- Learned the necessity of properly documenting my self-developed APIs and providing better code examples, as to prevent wasted work and alleviate time spent debugging the system.

EAR In-home robot for storing and retrieving items, using rPi, Python, Arduino, & C Fall 2019 - Spring 2020

https://github.com/JAJA-CSUS/EAR - work in progress; cross disciplinary team

- Learned documentation is to guide, but also protect the team from feature creep and outside expectations.
- Systems Design course covering product design, market evaluation, ethics, IEEE documentation, and design for testing.

CPU 5 stage Accumulator CPU in Verilog; UVM verification & testbenching Spring 2019

- In our courses, we cover a lot of how programs work, how high level languages are translated into machine code, and how that machine code controls the data flow (datapath) within a computer. In this project, I implemented the hardware side of the equation. Later, **JAWOS** explored the low level OS and software necessary for computer hardware operation.

Fan Duct FreeCAD modeled, 3D Printed using PLA Summer 2019

https://github.com/jeremyshaw/fan-duct

- Iteratively created a fan duct for my desktop computer, preventing hot air recirculation and lowering load temperatures by ~8C for the CPU (from ~85C in Cinebench R15 to ~77C; 33C ambient)

GPU powered VM Windows Guest, Linux Host with GPU passthrough using KernelVM March 2020

- Utilized KVM & IOMMU to enable high performance GPU-accelerated applications in Win10 guest OS

Relevant Courses

CMOS and VLSI (VLSI Design and Analog Effects)

Operating System Pragmatics (OS Architecture)

Advanced Computer Organization (Computer Architecture)

Intermediate Object Oriented Programming

Data Structures and Algorithms

Advanced Logic Design (Digital Logic Synthesis & RTL)

Computer Interfacing (Embedded Microcontrollers and Devices)

Computer Networking and Internet

Figure F1. Jeremy Shaw's Resume [17]

Alexander Leones

Objective

To obtain an entry-level or internship position which helps establish my professional engineering foundation.

Education and Activities

California State University, Sacramento ▪ B.S. Computer Engineering

People Skills Module of the Tau Beta Pi Engineering Futures Program

- Learned business communication techniques that encourage professional and effective communication between my coworkers and myself.

Tau Beta Pi ▪ Member

Technical Skills

- **Excellent written and verbal communication skills.**
- **Computer/Technical skills**:
  - **Programming Languages:** Python, C, Java, Verilog, x86 Assembly
  - **PSPICE/CAD software:** Cadence Virtuoso, Circuit Analysis CAD, Wireshark, Git.
  - **Microsoft Office** including Excel, PowerPoint, Word, Visio and Project.
  - **Currently learning:** Cybersecurity Methods, Data Science, and object-oriented Python Programming

Skills and Abilities

Teamwork/Approachability

- I possess the patience and professionalism to be trusted by my team whenever they need my assistance. You can always approach me and expect a speedy and professional response.

Problem Solver

- I think critically and creatively to complete whatever task I'm working on. Eager and willing to learn from others as well as independently search for the solution to whatever questions I have.

Projects

Elderly Assistant Robot ▪ Fall 2019 - Present

- Working with 3 teammates to develop an easy to use robot that can store, catalog and retrieve items for those with reduced mobility. Assisted in the research, documentation and technical development. Utilizing DC encoded motors and IR sensors for robot operation and pathing connected to Arduino microcontrollers and a Raspberry Pi running a Python/Tkinter GUI.

Unix-Like Operating System in C/Inline Assembly

- My teammate and I created an operating system for our OS Pragmatics class. The system includes system calls, inter-process communication, timesharing, virtual memory, device drivers, system signaling, mutex locking, bidirectional TTY and more. Developed in C and inline assembly using GDB for debugging.

Work Experience

Recreation Leader II ▪ Cosumnes CSD ▪ Summer 2016 – Present

- Acted as lead concessions cashier amongst a team of 2-5 employees. Assisted supervisors in training new employees, updating procedures, completing bank deposits, and managing stock levels.

Cyber Security Research SULI Intern ▪ Oak Ridge National Laboratory ▪ Postponed due to COVID-19

- Wrote four essays conveying my interest in research and qualifications as an applicant.

Figure F2. Alex Leones' Resume [17]

Angelica Smith-Evans

OBJECTIVE: An internship or full-time position involving computer science/engineering

EDUCATION:

B.S. in Computer Engineering — *Exp. Dec 2020*
California State University, Sacramento

KNOWLEDGE AND SKILLS:

- Programming Languages: Java, C, Python, HTML/CSS, JavaScript, MySQL, Verilog, VHDL, x86 Assembly
- Software: MS Office 2016, Adobe Acrobat, PhotoShop, SharePoint, Project Professional, Eclipse, NetBeans, Visual Studio, ModelSim, Multisim, Xilinx ISE, Matlab, Remote Desktop, all forms of Linux, GlobalProtect VPN,
- Hardware/Tools: Raspberry Pi, Arduino, Oscilloscope, Function Generator
- Foreign Language: Japanese (basic), Mandarin Chinese (basic)

WORK EXPERIENCE:

IT Student Assistant—ECS Computing Services — *August 2017—Current*

- Solve hardware and software problems for students, staff, and faculty.
- Complete prioritized long-term tickets for managing staff.
- Document work tasks to create easy-to-follow technical documents on software procedures.
- Research difficult tickets and communicate solution through email or phone to inquirer.

PROJECT EXPERIENCE:

Group Research—Cryptography on IoT Devices — *Jan 2020—May 2020*

- Research on the efficiency of lightweight cryptography algorithms on Internet of Things devices under Dr. Yuan Cheng
- Use established research to reference methodologies to implement in our own research

Senior Project Design—Elderly Assistance Robot — *Jun 2019—May 2020*

- Produce a ground-up working prototype that can deliver items to the disabled.
- Document in detail the process of the development into reports and compile these reports for instructor review.
- Perform a professional presentation of team's research and working prototype at showcases.

Applied Robotics Project Team—Sumo Bot — *Oct 2018—Dec 2018*

- Designed and implemented an autonomous robot for the "Sumo Bot Competition".
- Used a Parallax microcontroller, two continuous servos, and two infrared sensors.
- Wrote detailed technical report which closely described the development of the project.

Research Assistant—Efficient Traffic Controller Design — *Jun 2015—Aug 2015*

- Research assistant for a traffic controller efficiency design project for Dr. Jing Pang.
- Gathered, analyzed, and organized transportation and traffic system data.
- Presented the results of our research at the Cal Poly Pomona Creative Activities and Research Symposium (CARS) and was awarded the Fellowship and Participant Stipend.

ACCOMPLISHMENTS AND ACTIVITIES:

Hornet Leadership Program: SMUD Scholars — *Oct 2019—Current*
CSUS Badminton Club President — *Feb 2018—Aug 2019*
Cal Poly Student CARS Research Stipend — *Sept 2015*
Japanese National Honors Society Inductee — *May 2012*

Figure F3. Angel Smith-Evans' Resume [17]

Jesse Aucelluzzo

A hardworking team player with strong personal interactive capabilities looking for an entry-level position allowing career development as an Electrical Engineer. I am interested in a challenging and rewarding environment for professional growth in developing technologies. Willing to relocate and travel as necessary.

PROFESSIONAL EXPERIENCE

California Energy Commission Sacramento, CA
Engineering Internship August 2017 – Present

- Review customer applications to ensure New Solar Homes Partnership (NHSP) Solar Rebate Program requirements are fulfilled.
- Manage data entry into the NSHP program database.
- Create invoices and perform pre-audits to ensure compliance to California Energy Commission's Accounting Department processes.
- Respond to public telephone and email inquiries on renewable energy and corresponding government programs.

Golden Spikes Baseball Loomis, CA
Assistant Coach for Titans 10U July 2019 – Present

- Run bi-weekly practices and bi-monthly tournaments.
- Regularly communicate our team goals with parents and players.
- Report team and individual performances to our program's owner and work together with him to solve any issues within the team.
- Foster a culture of sportsmanship, cooperation and self-accountability within the team.

Code Automotive Tools and Restoration Supplies Thousand Oaks, CA
Store Associate January 2017 – July 2017

- Completed online orders from Amazon, eBay, and Codeautotools.com.
- Managed responses to customer inquiries and complaints.
- Responsible for maintaining warehouse and storefront inventory levels.
- Financial responsibility for deposit and cash box management and related banking interface.

EDUCATION

California State University, Sacramento CA

- Bachelor of Science, Electrical and Electronics Engineering, degree expected May 2020.
- Accomplished National Club Baseball Association (NCBA) intercollegiate baseball player.
- Balanced a full-time course schedule while working, competing and practicing with our campus' club baseball team, coaching a youth travel baseball team, and attending social activities.

Moorpark College Moorpark, CA

- Received Associate of Science (AS) Degrees in Mathematics and Physics while working part-time.
- Two-year collegiate baseball player.
- Two-year accolades as a scholar-athlete.
- 2016 Western State North Second Team All-Conference Award

ADDITIONAL SKILLS

Accomplished in various software and code development applications including: MATLAB, Allegro PSpice System Designer, C, C++, Verilog, Modelsim, AutoCAD, MS Excel, and MS Word. Accomplished athlete as a baseball player and coach.

Figure F4. Jesse Aucelluzzo's Resume [17]