

# ROBOT

*Red Owl Book Organization Tool*



University of Central Florida  
Department of Electrical Engineering and Computer Science  
Dr. Lei Wei

Initial Project Document and Group Identification  
Divide and Conquer

## Group 15

**Amanda Walenciak**

Electrical Engineering, Comprehensive

**Joshua Christie**

Electrical Engineering, Power Track

**Jostyde Lekadou**

Electrical Engineering, Power Track

**Montserrat Santillan-Rodriguez**

Electrical Engineering, Comprehensive

## Reviewers

**Zakhia Abichar**

PhD in Computer  
Engineering

**Sonali Das**

PhD in Electrical  
Engineering

**Ronald DeMara**

PhD in Electrical  
Engineering

# TABLE OF CONTENTS

<b>1 Executive Summary .....</b>	<b>1</b>
<b>2 Project Description.....</b>	<b>2</b>
2.1 Motivation.....	2
2.2 Objectives and Goals .....	2
2.3 Requirements, Specifications, & Constraints .....	3
2.3.1 Implementation: Using Computer Vision.....	3
2.3.2 Constraints for Implementation .....	4
2.4 House of Quality.....	5
2.5 Block Diagrams .....	6
2.5.1 Software Diagram .....	6
2.5.2 Hardware Diagram for Custom PCB .....	7
2.5.3 General Operation & Procedure.....	8
<b>3 Project and Part Selection Research .....</b>	<b>10</b>
3.1 Related Existing Projects .....	10
3.1.1 INTECH Smart Bookshelf .....	10
3.1.2 Smart AI Modular Bookshelf .....	11
3.1.3 SmartShelf Borrow .....	11
3.1.4 ARFiD0 Smart Bookshelf .....	12
3.2 Relevant Technology.....	12
3.2.1 Orange Pi 5 Dev Board .....	12
3.2.2 Raspberry Pi 4 Dev Board .....	13
3.2.3 Coral Dev Board.....	14
3.2.4 Jetson Nano Dev Board .....	15
3.3 Parts Research and Comparison .....	16
3.3.1 Power Design.....	16
3.3.2 Battery.....	17
3.3.3 Microcontroller .....	18
3.3.4 Cameras and Bookshelf.....	22
3.3.5 LEDs .....	25
3.3.6 PCB Vendors.....	27
3.3.7 Display Screen.....	29

3.3.8 RFID Sensors .....	31
3.3.9 Computer Vision.....	32
3.3.10 Application and Database .....	34
3.3.11 Serial Communication .....	37
3.3.12 IR Sensors .....	41
3.3.13 Wireless Communication .....	42
3.3.14 Operating Systems.....	44
3.4 Final Component Selection .....	45
3.4.1 PCB and Firmware.....	46
3.4.2 Hardware General.....	48
<b>4 Standards and Constraints.....</b>	<b>50</b>
4.1 Constraints .....	50
4.1.1 Economic Constraints .....	50
4.1.2 Environmental Constraints .....	51
4.1.3 Ethical Constraints .....	52
4.1.4 Health and Safety Constraints .....	53
4.1.5 Sustainability.....	54
4.1.6 Manufacturer Constraints.....	55
4.1.7 Social Constraints .....	56
4.1.8 Political Constraints.....	56
4.1.9 Time Constraints.....	57
4.1.10 Testing & Presentation.....	59
4.2 Standards.....	62
4.2.1 Kotlin and Android Standards .....	62
4.2.2 C Language Standard .....	65
4.2.3 Python Standards.....	65
4.2.4 PCB Standards .....	66
4.2.5 Computer Vision Standards .....	67
<b>5 ChatGPT and Similar Platforms .....</b>	<b>69</b>
5.1 Comparisons .....	69
5.2 Software Debugging.....	70
5.3 Data Analysis .....	70

5.4 Limitations .....	70
<b>6 Hardware Design .....</b>	<b>73</b>
6.1 Power, Voltage and Current Ratings .....	73
6.1.1 Jetson Nano & Raspberry Pi Camera .....	73
6.1.2 PCB Components .....	74
6.1.3 Display .....	75
6.2 Connections and Setup .....	76
6.2.1 Jetson Nano and cameras .....	76
6.3 Tablet with Raspberry Pi .....	79
6.3.1 USB and HDMI .....	79
6.3.2 Python.....	80
6.4 PCB Modules .....	81
6.4.1 PCB Individual Module Schematics .....	84
<b>7 Software Design .....</b>	<b>91</b>
7.1 Computer Vision.....	91
7.1.1 Computer Vision Flowchart .....	91
7.1.2 Computer Vision Use Case Diagram .....	92
7.1.3 Computer Vision Sequence Diagram .....	94
7.2 Computer Vision Software Overview .....	95
7.3 Coding Strategy for Application and Database.....	97
7.3.1 Application Code Implementation .....	97
7.3.2 Database Code Implementation.....	98
<b>8 System Construction .....</b>	<b>99</b>
8.1 Overall Schematic & Integration .....	99
8.2 PCB Vendor Assembly.....	100
8.2.1 PCB Specifications & Quotes.....	101
8.3 PCB Layout .....	103
<b>9 Prototype Test Plan.....</b>	<b>105</b>
9.1 Hardware Testing Environment.....	105
9.2 Hardware Specific Testing .....	106
9.2.1 LED Testing .....	106
9.2.2 Camera and Python Testing .....	107

9.2.3 Jetson Nano Testing .....	108
9.2.4 PCB Component Testing .....	108
9.3 Software Test Environment .....	109
9.3.1 Computer Vision Test Environment.....	109
9.4 Software Prototype Testing .....	110
9.4.1 Computer Vision Testing.....	110
<b>10 Administrative Information .....</b>	<b>116</b>
10.1 Milestone Discussion.....	116
10.2 Budget and Finance .....	119
<b>11 Conclusion .....</b>	<b>122</b>
11.1 Project Overview .....	122
11.1.1 Achievements .....	122
11.1.2 Challenges.....	123
<b>Appendix .....</b>	<b>124</b>
References.....	124

# 1 Executive Summary

The rising popularity of artificial intelligence and machine learning in everyday settings makes room for new and innovative solutions. The ROBOT project seeks to change traditional bookstores and libraries by adding computer vision to create a one-of-a-kind smart bookshelf. By using text and image recognition, our system will be able to detect and quickly scan entire bookshelves at the snap of a picture.

The principal characteristic of ROBOT lies in its computer vision and database capabilities. By using a camera and computer, the system can see, detect, and log each individual book it can identify. The computer directly updates an online database that users can access in real time. The uses of such a system are many, from effortlessly maintaining an organized book inventory to allowing users to locate specific titles through an application. Users can also quickly locate books rather than scanning entire shelves with their own eyes, due to the LEDs placed along the edges of the bookshelf. The combination of physical and digital tools allows users to spend less time searching for books and gives them more time for reading and learning.

This project uses a combination of hardware and software engineering. As the ROBOT system detects changes in its inventory, it communicates with a backend database, records additions, deletions, and ensures an accurate, up-to-date representation of the library. The project is not just innovative with its use of computer vision, it also acknowledges practical challenges, such as the varied nature of human and library interactions and setting changes. ROBOT aims to improve traditional bookkeeping by making it faster, more accessible, and more accurate for users. This project could well be extended to general inventory applications such as tracking a store's merchandise and stock.

ROBOT stands out due to its integration of computer vision, a feature that is absent in ordinary bookshelves. Many existing solutions lack the ability to independently recognize and catalog books and heavily rely on manual input. As a result, our smart bookshelf not only exceeds the limitations of everyday bookkeeping; it also contributes to the artificial intelligence era by making an everyday object become independent, responsive, and integrated into the digital world for many to access.

## **2 Project Description**

In this section we describe the motivation, objectives, and goals of the ROBOT project. We provide broad specifications for hardware, software, and structure from our initial draft idea of the project. Overall, this section describes our expectations and ideas of how the project will work and how the work will be carried out.

### **2.1 Motivation**

Even though many people enjoy reading physical books, public libraries and physical book retailers have been decreasing in popularity in recent years. A common point of frustration is that public libraries and bookstores do not take inventory in real time like electronic book retailers. Although many prospective readers enjoy being able to sample books, the tendency of readers to hold on to books for a long time before checking them out makes inventory management a challenge. Though many modern libraries and retailers keep websites that can inform customers whether the book they want has been checked out yet, they cannot keep track of the books that are neither on the shelf nor checked out. Many books that may register as “available” on these websites are either being read by another customer, are lost, or may even be stolen.

### **2.2 Objectives and Goals**

Our project aims to transform book inventory management by introducing a smart bookshelf equipped with Computer Vision (CV) technology. Using an LED tracking system, our innovation will deliver a more efficient way to track the availability and location of books within libraries and bookstores. Moreover, it will offer a better user experience by providing real-time access to book information through an Android application and database.

The first goal of the project will be the construction of a bookcase equipped with a wide-angle camera on each shelf. This camera will be connected to a computer containing a programmable book database. The computer will be programmed with CV, enabling the cameras to detect in real time when a book has been removed from or put back on to the bookshelf. This information will then be communicated to the app, informing prospective readers whether the book they want is currently shelved.

A main deliverable of this project will be to use CV to increase data efficiency, accuracy, and reliability. For this project, we chose CV as opposed to conventional barcode scanners. This is because a barcode implementation would require moving parts, as well as proximity to the scanner. CV can identify multiple objects simultaneously, can scan from a larger distance than barcodes can, and can update in real time.

In addition to an efficient scanning system, the project also aims to deliver a programmable app with an LED search feature. The physical bookshelf will be lined with a strip of LED lights that each correspond to the location of a book. The search function in the app will be paired to the LED lights. After searching for a book within the app, an LED corresponding to the location of the book will light up, enabling the easy identification of available books.

## 2.3 Requirements, Specifications, & Constraints

Currently, the team is considering implementing the smart bookcase by using cameras, computer vision, database, and a computer such as a Jetson Nano to process the camera images and quickly scan books on the shelf.

### 2.3.1 Implementation: Using Computer Vision

Hardware:

1. **Cameras:** These cameras will be small, wide-angled cameras that capture the horizontal view of the bookcase. Pictures taken by the camera will be sent to a computer for processing and scanning of books. The cameras could either be wireless or wired.
2. **LEDs:** These lights will be placed along the front, lower edges of the bookshelf to help quickly locate books by lighting up to give users the approximate location of the book being searched for.
3. **Power supply:** The power supply needs to provide power to the computer, LEDs, and PCB that will be used.
4. **Computer:** The computer can be a laptop, tablet, or a Jetson Nano, that will manage communication between the cameras, LEDs, PCB, and software. Additionally, the computer will be responsible for receiving and processing images from the camera and updating the database. This computer should be able to power the cameras.
5. **Printed circuit board:** The circuit board will help to control information between the LEDs and computer by receiving commands from computer on which LED to switch on and off. The circuit board should have a serial communication module, power distribution and switches for several LEDs, and a JTAG interface for programming and debugging the PCB.
6. **Display touch screen:** If we were to use a computer such as Raspberry Pi, we would possibly need to implement a small display screen. This display screen could provide further interaction for users by allowing them to search the bookshelf and trigger an LED light book find.

Software:

1. **Computer vision:** A program to help process and scan the pictures received from camera and update database.

2. **Book database:** The book database would receive and update information from the computer based on the information received from the images. The database would update the book statuses based on three different states: on shelf, checked out, or missing. The database could also keep track of dates and times books were last seen on the shelf.
3. **Application:** There will be integration of an Android application for phones that users can use for searching and checking out books. We will also make a web application that will be accessible on the touch screen alongside a Raspberry Pi.

Physical Structure:

1. **Bookcase:** The bookcase will hold the books and be modified to put the cameras in a suitable location. It will also be modified to potentially hold a display screen at the end of it. Wiring will be attached to it and hidden in the back. LED strips
2. **Camera pockets:** Basic 3D printed structures could be made to hold the camera and place it on bookshelf in a manner that is non-invasive and has minimal visibility for library users.

### 2.3.2 Constraints for Implementation

Possible limitations and problems that could arise from implementing the smart bookshelf with computer vision could be:

Books without the title and author on the side of spine could require special case detection by the system.

If a book is stolen and/or missing, the system cannot track down the location of the book.

If a book is missing but not stolen, it could be that someone is currently reading it in the library. However, the current system design would not be able to detect such a situation.

Because books are formatted in many ways, and though most have the title and author on the side of the spine, there are also some that do not fit these criteria. Some books could also be in poor condition, so much so that the color on the outside is fading away. Such book conditions could complicate our computer vision and database implementation. We are currently looking at methods of scanning that could serve as a “one size fits all.” This could be implementing stickers, perhaps like barcodes, that every book can have on its spine on a location where it is accessible to the camera and is easy to detect.

Our system will only keep track of inventory within the bookcase, so another limitation is that the system itself cannot help to track down precise location of

books if they are not within the view of its designated shelf. This provides little backup for books that can be marked stolen by our system.

Lastly, another situation that our system will not know how to account for is the fact that people at the library or a bookstore like to take books off the shelf and read them while hanging around. This is not a true “stolen book” situation, but if they are holding onto it for more than 4 hours, for example, our system might alert the book as missing and/or stolen, even if it is not. The system will not have knowledge on how to differentiate between the two situations.

## 2.4 House of Quality

To visualize the work and goals that need to be met, we created a House of Quality shown in Figure 1. This helped us to see the correlations between the engineering required and the functional expectations of the final product.

Figure 1 shows the legend map on the left. The correlation matrix symbols are used to see relationships between different engineering requirements. The relationship matrix symbols show the relationships between different engineering and customer requirements.

From the table we can see that book detection accuracy and effectively helping to quickly locate books are some of the most critical expectations for the project. Both things will be limited by money and power constraints.

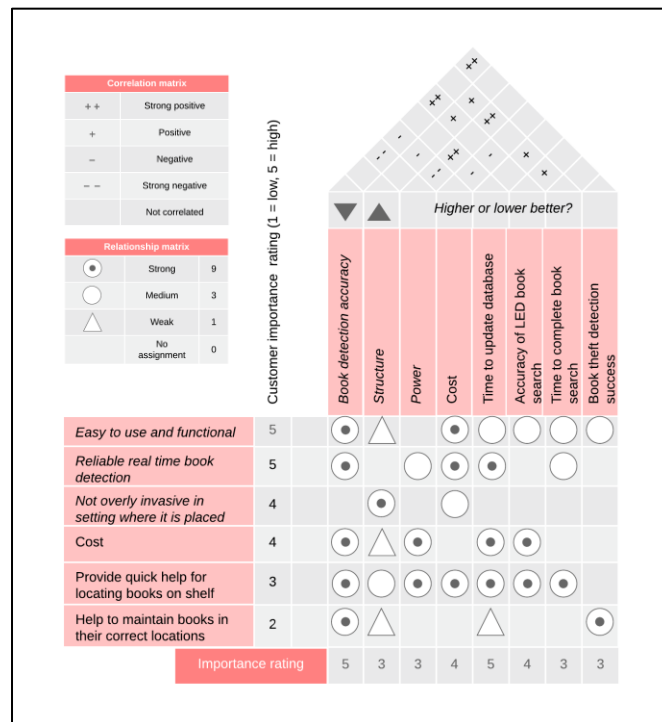


Figure 1: House of Quality

## **2.5 Block Diagrams**

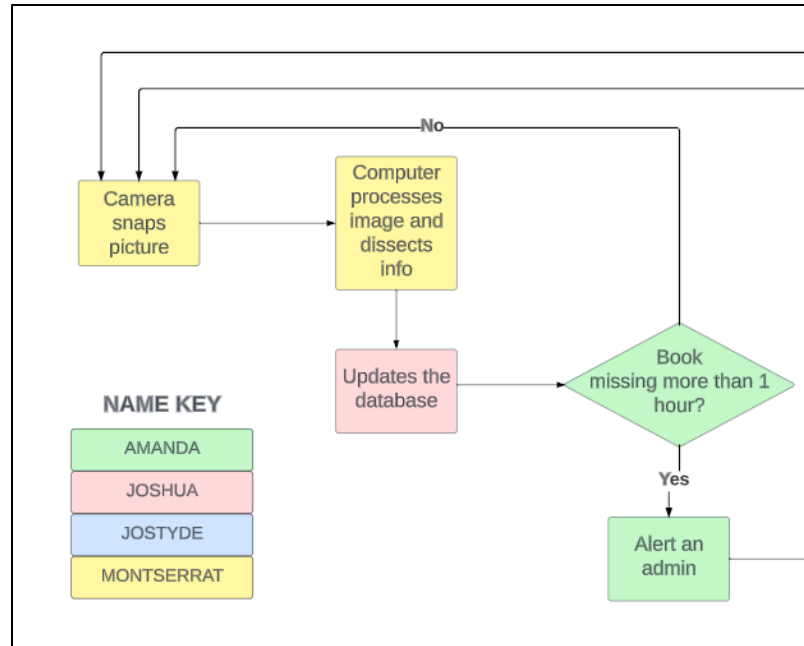
This section aims to brainstorm how we expect the software to work and what the hardware will consist of. We broke it down into three different diagrams. The first shows an example of the control process for how various sub systems will work with software including the cameras, database, and main computer. The second diagram shows what the expectations for the custom PCB to be built are. Lastly, the last diagram shows the general operation of the final product including how users interact with the system.

### **2.5.1 Software Diagram**

Our system is heavily reliant on many software concepts; each member in the group will participate in implementing the software side of the project. The project will consist of four major software related parts:

- Application
- Database
- Computer Vision
- Embedded firmware

The diagram in Figure 2 below shows how the cameras, computer, and database will interact with software. The software will be implemented in the main computer which will process images and update the dataset. The database will get updated based on messages received from the computer. If the computer detects a book as missing, it will send a command to the database to update its status field. Also, the PCB will receive commands from the computer and/or user application to initiate a book find which will trigger an LED to light up.



**Figure 2: Software Block Diagram**

Two people, Amanda and Jostyde, will take care of the embedded firmware. The embedded firmware consists of managing communications between the custom PCB, computer, and LED lights and alarms. The computer should be able to send and receive messages to the PCB and vice versa, either through direct connection or a wireless communication method like Bluetooth.

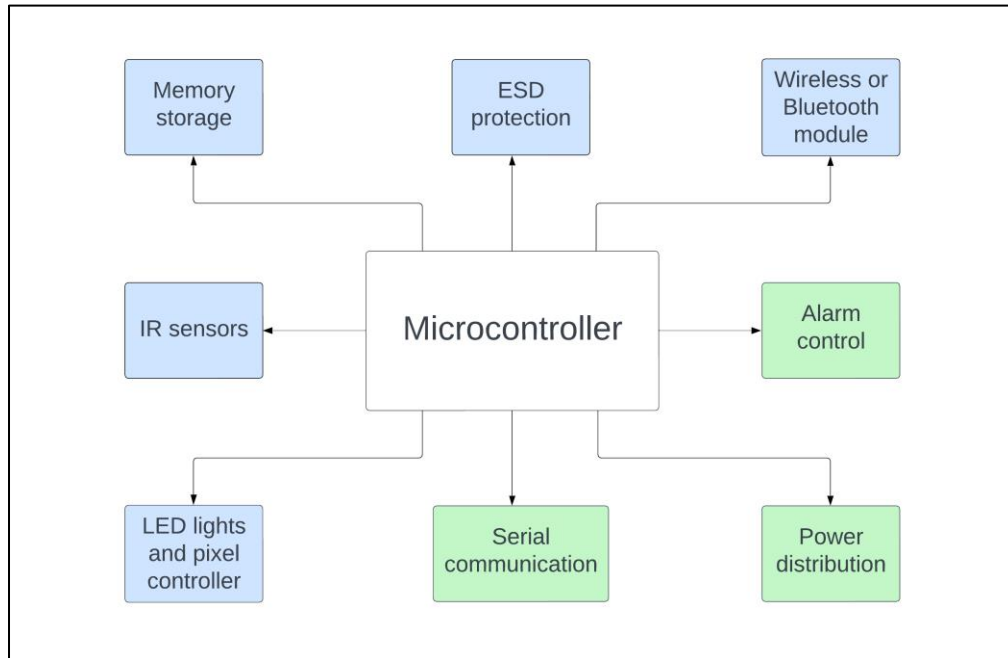
Two people, Joshua and Montserrat, will take care of implementing the application and database. The database should be easily accessible, modifiable, and readable by the computer processing images from the camera and by multiple users. Mobile and web applications should be able to also access the database. Implementation of the database will consist of enabling users to create accounts and store some personal information like book due dates through a mobile application.

Lastly, two people, Amanda and Montserrat, will take care of implementing computer vision. Using a computer like a Jetson Nano or Raspberry Pi, the computer should be able to receive images from the cameras and process them to detect and extract text. This will be done through open-source libraries and a programming language like MATLAB or Python.

## 2.5.2 Hardware Diagram for Custom PCB

Our custom PCB will take care of communication between several devices. A connection between the computer processing images and the PCB should be established so that they can receive and send commands to one another. The computer will send the PCB commands such as which LED light to trigger on. This communication is likely to be done through Bluetooth protocol. The PCB will have

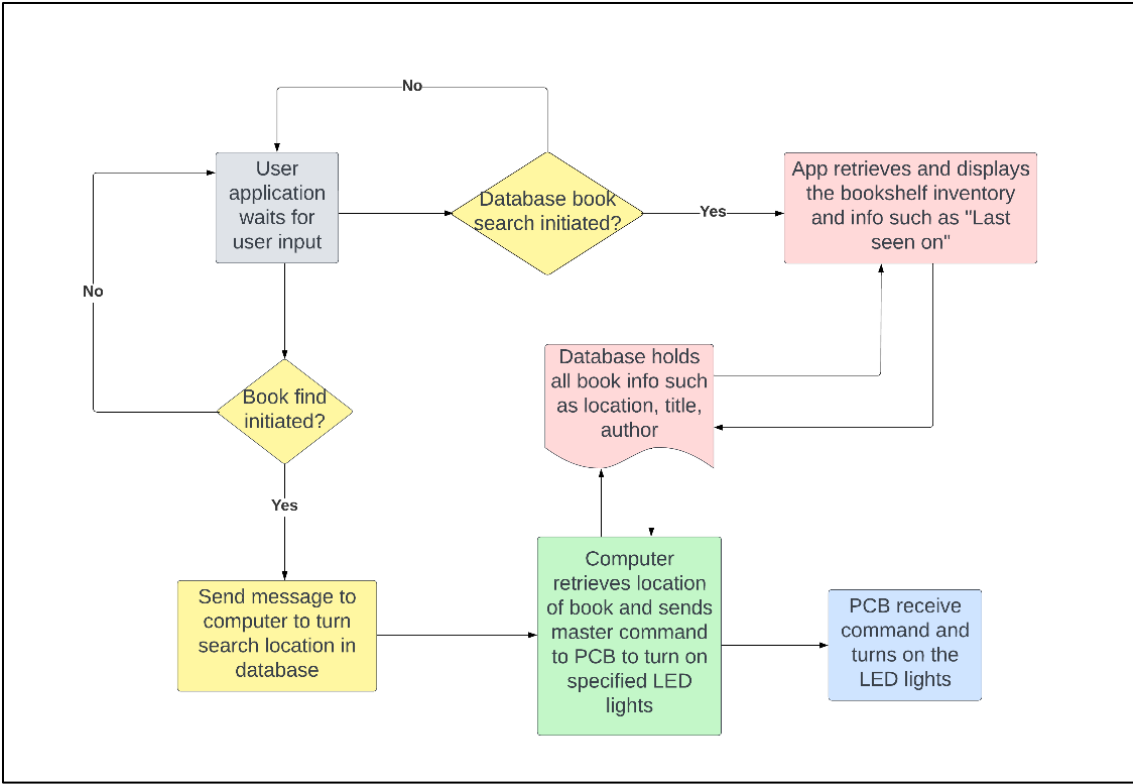
ESD protection to protect it from potential damage due to human touch. It will also contain memory storage so that it can store the firmware it will run throughout the project. The PCB will also distribute power the Bluetooth module and LED lights controller. A stretch goal is to implement an alarm system to ring when a book is identified as missing past “open hours.” An overview of this is shown in the diagram of Figure 3.



**Figure 3: Hardware Block Diagram**

### **2.5.3 General Operation & Procedure**

The diagram in Figure 4 shows a merged version from the previous two diagrams and aims to show how the overall system will work alongside sub systems and user interaction.



**Figure 4: General Procedure Block Diagram**

## **3 Project and Part Selection Research**

In this section, we explore potential parts and features that can be used in our final product and relate existing projects to our model. By doing this, we will get a clearer understanding of what to modify in our current plan and how we can innovate upon past projects. Comparing and analyzing specific component parts will help us to choose the best parts that will ensure our system is functional and economical.

### **3.1 Related Existing Projects**

There are currently several existing smart bookshelf products and projects underway that are similar to the idea of ROBOT. By analyzing the features, specifications, and dimensions of these bookshelves, we will gain valuable insights into how to improve upon them through our own project. By doing so, we aim to provide a new perspective with never-before-seen features in the world of bookkeeping.

#### **3.1.1 INTECH Smart Bookshelf**

The INTECH Smart Bookshelf is offered by a Chinese company that specializes in smart library and classroom technology [1]. The bookshelf has capabilities to detect equipment in real time, provide position and navigation of equipment, and generate equipment loss reports, all through RFID (Radio Frequency Identification) implemented within the bookshelf. The bookshelf can hold 350 books or less, has a frequency reach of 13.56 MHz, uses the Android operating system for its interactive tablets attached to the bookshelf ends, and consumes 40 W or less of power. More of the system's features include letting librarians know when a book is placed on the wrong shelf and ranking book popularity.

This product has more features than ROBOT aims to have, though there are some that overlap. Like the INTECH Smart Bookshelf, one of the goals of our project is to provide the approximate location of a specific book. While INTECH uses RFID and maps a route on the screen for the user, our project would look for the book through computer vision, and then light up an LED on the bookshelf at its appropriate location. This is easier to follow and faster since the user then just looks for a lit-up light on the bookshelf as opposed to still having to scan the bookshelf once you arrive at the approximate vertical location of the book. The implementation of the INTECH uses an RFID system which is known to have drawbacks such as expensive, risk of being compromised, and the exposure RFID tags have on items which thieves can easily remove [2]. Using computer vision to achieve similar results, we can make the project simpler, cheaper, less power consuming, and decrease the risk of interference.

### **3.1.2 Smart AI Modular Bookshelf**

The Smart AI Modular Bookshelf won the reddot award in 2018 [3]. The bookshelf provides location tracking of books through image recognition. It has an embedded camera and books can be registered to the system manually. Whenever the user initiates a book find, the bookshelf holding the book will light up. It also allows users to log their reading progress and publish book reviews. Thus far, there is little further information available about the project concept design.

The modular bookshelf has similar features we aim to provide through our project such as location tracking through image recognition. While this project would light up an entire bookshelf holding a particular book, the ROBOT system would light up a more specific light directly on the bookshelf.

### **3.1.3 SmartShelf Borrow**

SmartShelf Borrow by Bibliotheca is another comparable existing product. This product allows users to simply place returns on the shelf while it identifies the book, removes it from the user's account, and enables security. This allows the book to be immediately available for recirculation [4].

The main appeal of this product is its effortless self-service capabilities. According to the website, the use of this technology has enabled libraries to make more efficient use of staff time, cut costs, prevent wait times, and create hold slips that print at a convenient location for staff [4].

Our product is both similar to and different from the SmartShelf Borrow. Our product does not rely on RFID or a scanner to identify books. Instead, we will use a wide-angle camera equipped with computer vision.

Additionally, our product comes with a night mode intended to sound a security alarm if a book is removed during unauthorized hours. Though the SmartShelf Borrow mentions security, they do not seem to offer an alarm with their product.

Our product does not intend to offer hold slips. However, we do have a similar idea in the form of LED tracker lights that will help staff and customers identify the location of a book. These LEDs will be interfaced with the app. SmartShelf Borrow does not specify how a user should print a hold slip; it seems marketed as a feature for staff rather than helping customers directly [4].

Finally, our product will come with an app that both staff and customers will be able to use. Though both products will offer the ability to instantly check out or return a book, they are implemented in different ways.

### **3.1.4 ARFiD0 Smart Bookshelf**

Finally, one last project that can be compared against ROBOT is the ARFiD0 Smart Bookshelf. This is another RFID project that features partitions on the shelves where the books belong. Unlike other projects that use a scanner on the bottom of the shelf and unlike ROBOT's use of computer vision, the RFID scanner is located in the partitions themselves. This unique design enables it to accurately track the location of each book on the shelf, achieving a similar aim to our computer vision camera [5].

This Smart Bookshelf also features a control module with a touch screen LCD and a receipt printer [5]. ROBOT will feature an LCD screen much like this Smart Bookshelf. However, instead of a receipt printer, ROBOT will utilize an app to handle the book location and inventory process.

Another difference between this product and ROBOT is that books removed from ROBOT are not automatically considered checked out. They will show in the app as unavailable, however, they will not be considered checked out until they are checked out by a human. This is in recognition of the fact that sometimes people pick up books to view or read that they do not actually intend to check out. Additionally, our system acts as more of a failsafe against theft because a book that is not on the shelf and not checked out will be considered lost or stolen.

The Smart Bookshelf communicates with the library via the SIP2 protocol [5]. ROBOT will likely rely on a different method to communicate with the database of books we will be tracking.

## **3.2 Relevant Technology**

At the core of the ROBOT project is the computer that will process images with computer vision, interact with the database, and send and receive messages from the PCB controlling the LED lights. It is critical that the computer we choose has the appropriate performance and specifications to ensure our system is fast, accurate, and responsive. By analyzing and comparing 4 different boards, we will pick the cheapest and most efficient for the needs of our system.

### **3.2.1 Orange Pi 5 Dev Board**

The Orange pi 5 is another single board computer designed by a company of the same name. It is an open-source computer that contains very capable hardware. Specifically, the GPU, as it will be handling most of the computer vision load for recognizing the books on the ROBOT. shelf. In addition, the operating system of the Orange Pi is based on Android, which will make it simpler to integrate the data from the cameras into the mobile application. In addition, it also has great

integration with cloud computing and database applications, which will be great with our book database system [6].

The Orange pi 5 contains a Rockchip RK3588S octa-core (4 Cortex-A76 Cores + 4 Cortex-A55 Cores) CPU, running up to 2.4Ghz. With an ARM Mali-G610 GPU, with either 4, 8, or 16GB of DDR4 RAM [6]. Furthermore, the Orange Pi supports camera inputs as high as 8K at 60fps with an included codec. The unit does not come with on-board storage but does have support for NVMe SSD storage with an M.2 slot as well as a MicroSD card slot [6].

As far as connectivity, the Orange Pi boasts a full-sized ethernet port, capable of up to 1000Mbps transfer speeds. As well as a 3.5mm headphone jack, Full-sized HDMI port, USB 3.0 ports as well as two USB type C ports (One for power at 5v/4v, and the other for data at 3.0 speeds). Most important of all, there are three on-board camera ports that can be used for CCTV cameras and other applications. This is the reason there is no WiFi chip present on the board, it is done to avoid lag and stuttering in camera feeds and recordings, as wireless can be unstable and inconsistent, where a wired connection is more stable. The Orange Pi also supports UART, SPI, PWM, and I2C communication modes [6].

### **3.2.2 Raspberry Pi 4 Dev Board**

There are several reasons why Raspberry Pi was chosen as one of the technology options for this project. One of the reasons it got on our radar was due to its popularity. Because it is well-known, we decided it was something worth looking into and researching. Like many things, there are many advantages and disadvantages to using the Raspberry Pi and it all comes down to how you intend to use it as well as what you need.

The main advantage is that they are cheap, small, and easy to use [7]. Raspberry Pi tends to be the best for beginners and projects that are simple and do not require large power capacity. Raspberry Pi is affordable and can be used with a computer to execute simple tasks [7]. Even though it is affordable, the more recent a model is or the more memory it has, the more money it costs. For example, since the Raspberry Pi 4 is a newer model than the Raspberry Pi 3B+, it will be more expensive [7]. Price is a very important factor when it comes to picking a computer because this project is being funded by four low-income college students, therefore, to be able to afford good quality products is extremely important. Being that Raspberry Pi is well known in the technology community, it would be easy to find many resources when it comes to needing help with any problems that can occur.

Obtaining the Raspberry Pi is not enough on its own. Other components are needed as well such as the SD card, fan, HDMI cable, and power switch. All the components needed can add up, but it still comes out cheaper than buying a

complete computer. The great thing about the design of the Raspberry Pi is that it comes with many different hardware interfaces such as USB ports, HDMI, GPIO ports, etc. making it easier to connect different sensors or devices and make it unique.

Raspberry Pi was originally made for kids' education purposes which is why it is affordable, small, and easy to use [7]. The device is small, which is needed for this project, so it will not be hard to hide due to its size. On top of that, it is important that it is easy to use, especially if you are a beginner. Having a computer that is easy to use and understand is important for this project so that more time can be spent on other important aspects of the project. The Raspberry Pi is a great computer to use for beginners or small tasks.

Although the advantages of the Raspberry Pi may be good, the disadvantages might make us question if the Raspberry Pi is right for this specific project. A disadvantage is that the Raspberry Pi has limitations on what it can do, which is understandable considering it was originally made for kids, how small it is, and that it should only be used for small tasks. On top of doing relatively small tasks, it has slow processing speed being a single-board computer [7] which means that multiple tasks should not be done at once. Raspberry Pi is intended to be used for simple projects and education meaning it won't be able to do 3D graphics or gaming and has limited storage especially compared to other products.

Some important features that the Raspberry Pi model 4 has are the Quad-core Cortex-A72 (ARM v8) 64-bit System-on-Chip (SoC) at 1.8 GHz as well as 2.4 GHz and 5.0 GHz IEEE802.11ac wireless, and Bluetooth 5.0. Raspberry Pi includes 2 USB 3.0 ports, 2 USB 2.0 ports, 40 40-pin GPIO header, 2 micro-HDMI ports, and 5V DC power input through a USB-C connector or GPIO header. Additionally, the Raspberry Pi supports UART, I2C, SPI, Ethernet, camera interface, display interface, and network communication. These are all important for the functionality of ROBOT and play a big part in the decision-making [7].

The Raspberry Pi would be a good fit for our project if what we intend to do is simple. The more complicated our project becomes, the more we would have to consider other options meaning more expenses.

In the end, we decided to use the Raspberry Pi 4 board as well. It was a great choice for our project due to its balance of performance and versatility. While the Jetson Nano had its strengths, it didn't quite provide the power we needed to smoothly run our main applications. Fortunately, the Pi 4 stepped in to fill the gap, offering enough processing capability to handle our needs without a hitch. The combination of these two devices provided us with a robust setup that worked well for our project.

### **3.2.3 Coral Dev Board**

The Coral Dev Board is a development board to quickly prototype on-device machine learning (ML) products. It contains the Edge TPU hardware accelerator, which enables high power ML interfacing at low power cost [8]. The Coral Dev Board supports TensorFlow Lite, which allows developers to deploy machine learning models onto the board directly. This simplifies the development process for a wide range of AI applications, including image and video processing and voice recognition. Because ROBOT deals with computer vision, this could prove useful.

The Coral Dev Board includes a quad-core Cortex-A53 CPU, a Cortex-M4F, and an integrated GC7000 Lite Graphics GPU for high-performance computing and graphics rendering. This GPU contains a Google Edge TPU coprocessor, which offers 4 TOPS (int8) of AI processing power and energy-efficient performance. It provides options for 1 or 4 GB of LPDDR4 RAM and comes with 8 GB of eMMC flash memory, expandable via MicroSD [8].

Connectivity options include 2x2 MIMO Wi-Fi (802.11b/g/n/ac), Bluetooth 4.2, Type-C OTG, Type-C power, Type-A 3.0 host, Micro-B serial console, and Gigabit Ethernet. The board supports audio through a 3.5mm audio jack, digital PDM microphones, and a terminal for stereo speakers. Video output includes HDMI 2.0a and connectors for MIPI-DSI displays and MIPI-CSI2 cameras. It offers versatile GPIO capabilities with a 3.3V power rail and programmable impedance. The board is powered by a 5V DC supply through USB Type-C. Additionally, the board supports communication modes such as I2C, I2S, SPI, and UART [8]. This makes it compatible with many peripherals.

The Coral AI toolkit is another available tool. This toolkit has a Linux-based operating system, as well as many optimized libraries and APIs tailored specifically for AI and machine learning applications.

### **3.2.4 Jetson Nano Dev Board**

The Jetson Nano Dev Board is available from Nvidia and is among one of the most popular computers purchased for small scale AI projects. The board is capable of handling 4GB of random-access memory—vital for quick access to information. According to Nvidia [9], the dev board can run several neural networks in parallel to allow applications to run simultaneously. The company advertises its usefulness for applications related to AI and robotics. Boesch [10] states that the Nano is ideal for “processing data from high-resolution sensors” and “ideal for AI-based computer vision applications and to perform tasks like image classification, image segmentation, object detection, and more.” The board features a 128-core Maxwell GPU, 2 MIPI CSI-2 DPHY camera lanes, connectivity to Ethernet and M.2 Key E. Connection to display is compatible through HDMI or DisplayPort. It allows 4 USB 3.0 connections and 1 USB 2.0 Micro-B connection and provides general input and output pins. For power, a supply providing 5V and within 2.5 to 3A is required and

can be connected using micro-USB, DC barrel jack, or USB-C. Lastly, it supports I2C, I2S, SPI, and UART communication modes.

Supported software for the board includes Jetpack SDK which provides a Linux-based operating system, accelerated libraries, and APIs specifically for applications like deep learning and computer vision [11]. Furthermore, according to Nvidia, Jetpack SDK “provides computer vision and image processing algorithms implemented on multiple hardware accelerators” such as the encoder and programmable vision accelerator.

Because of its popularity, there are several resources—including YouTube tutorials and even entire courses [12]—specifically catered for using the Jetson Nano for computer vision applications. This would help to make the set up and troubleshooting of the board much easier and quicker, too.

In the end, we chose the Jetson Nano 4GB as the computer for our project. Setting it up with all our needed libraries and peripherals turned out to be a bit challenging. The Nano requires proper attention and setting up early on in implementing projects physically. We also had trouble when powering it through the micro-USB port as it kept shutting down when running demanding scripts. Powering it through the barrel jack solved this problem. With four electrical students, setting up the Jetson Nano, which runs a Linux Ubuntu operating system was a huge learning curve. Nonetheless, we were able to learn a lot about Python and setting up Linux environments, as well as being able to purchase a fully functional Nano for under \$80.

### **3.3 Parts Research and Comparison**

To accomplish the outlined goals of ROBOT, comparisons of similar parts, products, services, and languages must be made to determine what parts are the best for the specific needs of the project. These comparisons will be made with the help of a large amount of research and discussion amongst the team based on the direction of ROBOT. This research will highlight components such as: connectivity, microcontrollers, cameras, LEDs, PCB vendors, and software suites.

#### **3.3.1 Power Design**

Power is a fundamental need when it comes to technology, without it, nothing would function. Power has become a big part of our everyday lives, whether it is to power on your phone, the computer you use at work, or even the computer vision systems in your vehicle. Power is important, and for this project, it's no different.

Power is a fundamental component that gives each component energy and allows them to cooperate to accomplish their assigned responsibilities. The computer, camera, and LEDs are a few of these parts; each having a different power demand.

Our goal is to have a power source that reduces complexity and maximizes efficiency in our effort to optimize power management and guarantee ROBOT's seamless operation. The computer will function as the main processing unit, coordinating its movements and the process of making decisions. Due to its essential nature, this core component needs a steady supply of electricity to function properly. The power supply for the camera and LEDs will come from the computer since they may be connected to it. This makes the setup process simpler and guarantees a more ordered and effective power distribution system. Our team has resorted to rechargeable batteries in our search for a consistent and adaptable computer power supply. To further increase our power options, we have included a DC Jack connector that allows for a 5V DC power input. This connector simplifies operation and guarantees compatibility with a range of power sources. By providing redundancy, this dual-powered strategy increases ROBOTs resistance to power-related issues.

### **3.3.2 Battery**

The computer will feature a rechargeable battery in addition to a USB-C port for power because every device should have a backup power source. There should be a rechargeable battery in the device even though it will be stationary and plugged in all the time to serve as an uninterruptible power supply (UPS). Because we intend to make the computer fixed rather than portable, rechargeable batteries are important because, without them, it would still be susceptible to unplanned power outages. Rechargeable batteries make sure that work can go on without interruption and protect against data loss and interrupted processes. They have several benefits, such as serving as a UPS and being beneficial for data security and energy efficiency. Energy efficiency can be enhanced via rechargeable UPS systems. They can be configured to switch off the computer after standard business hours or after prolonged periods of inactivity. In addition to lowering electricity use, this also helps with cost and environmental responsibility. The environment also benefits from rechargeable batteries. The need for discarded batteries is decreased and electronic waste is kept to a minimum by giving computers a long-lasting power source.

#### **3.3.2.1 Pisugar S Pro and Geekworm**

The UPS for the Raspberry Pi will be provided by Pisugar S Pro. This is due to its compact size, low cost, and simplicity of installation as a power source. Pisugar S Pro is a lithium battery, and because it was designed for the Raspberry Pi, its voltage management is appropriate for the computer. When it comes to the Jetson Nano, we will be utilizing the Geekworm 5V UPS, which works with both the 2GB and 4GB versions. This is convenient for our group as we have the 4 GB model [13].

#### **3.3.2.2 USB**

The Universal Serial Bus (USB) protocol is user-friendly and has numerous uses. Almost all electronic devices, including printers, scanners, cameras, cellphones, and keyboards, use this protocol. Additionally, it is used for data logging operations between a PC and a measuring device. USB is a common form of computer port that allows data transfer between two devices and device charging. It replaces the slower and larger serial and parallel interfaces because of its size and speed. The advantages of USB are its ease of use, wide application coverage area, low interference, small size, affordable price, and support for multiple speeds. Its 5-meter USB connection cable standard is one of its drawbacks and to increase the length, you will need to use a USB hub. Moreover, it lacks a broadcasting feature [14].

There are many different types of USBs, the most common ones being USB-A and USB-C. Other USBs are USB-B, USB 2.0, USB 3.0, USB 3.1, USB 3.2, USB 4, USB Power Delivery (PD), and USB On-The-Go (OTG) [14].

### **3.3.2.3 USB-C**

USB-C is the current standard that combines display connectivity with data and power delivery. Most new technology, including laptops, earbud cases, smartphones, etc., have USB-C ports. Its reversible, small, and oblong form factor allows it to be connected in either way. Even with larger devices, the port's 100-watt connection makes it ideal for quick data transfers and charging. Compared to other USB types, USB-C is faster and more capable. USB-C's versatile multitasking capabilities make it a potential cable that could replace all others [14]. Given the USB-C's power, widespread availability, and compatibility, we concluded that it would serve as the primary power supply for the computer.

## **3.3.3 Microcontroller**

Several different microcontrollers are being considered for the purpose of our project. The ideal microcontroller is one that is easy to use, runs on relatively low power, is compatible with our goals and other technological components, and can be programmed using a language we are familiar with. To that end, we are considering the MSP430FR6989, the MSP430G2553, and the Arduino Uno.

### **3.3.3.1 The MSP430FR6989**

The MSP430FR6989 is a Texas Instruments MSP430 microcontroller. It is known for its power efficiency, meaning it will be a good choice for projects with power constraints. This microcontroller has a clock speed of 16 MHz and has 2 KB RAM, which is a decent amount of computing power that should be sufficient for our project. The device features 1.8-V to 3.6-V operation [15].

This microcontroller has several different modes meant for saving power, such as standby, hibernate, and real-time clock modes. This MSP430 is capable of

reaching as low as 100uA/MHz in active mode, 0.4 uA in standby mode, and uses an extremely low amount of power in hibernate mode [15]. Because of the low power consumption, it can even be operated on battery power.

The MSP430FR6989 uses Ferroelectric Random Access Memory (FRAM). FRAM is a non-volatile memory technology that combines the best aspects of both Flash and SRAM. FRAM provides fast read and write speeds and lower power consumption. This will make programming simpler and is known to be especially useful in projects involving data logging, configuration storage, and frequent data updates. Debugging and communication with the PC is made easy with the eZ-FET, which provides a "backchannel" UART-over-USB connection with the host [15].

The MSP430FR6989 Launchpad Development kit can be used with this microcontroller. This kit adds significant functionality to the MSP. The kit has an LCD display, meaning users will be able to see information displayed on it instantly. The MSP430FR6989 comes with a wide range of peripherals, including GPIO pins and five timers. These peripherals mean that it can be used for many applications, such as sensor interfacing, communication with other devices, and precise timing control. The MSP430FR6989 is compatible with all three major communication protocols: I2C, UART, and SPI. This means a wide variety of peripherals and devices are able to be used with this MSP430 easily. Additionally, it offers high-resolution analog-to-digital converters (ADCs) for applications that require accurate analog data acquisition [15]. This MSP430 is also compatible with a large number of BoosterPacks that can be developed ourselves or purchased through the Texas Instrument website.

Another benefit is that the group already has access to an MSP430FR6989 due to prior class requirements. This prevents us from having to get a new microcontroller, saving money. Furthermore, the group has access to a comprehensive book on MSP430 microcontroller basics from an earlier class. Because of our past experiences, multiple group members are comfortable with using and programming this device. We are also comfortable using Code Composer Studio, which is compatible with this microcontroller.

One downside is that the microcontroller uses relatively more power in active mode compared to alternatives. Additionally, its limited community support, when compared to certain other microcontrollers, can pose challenges in finding resources and troubleshooting.

### **3.3.3.2 MSP430G2553**

Another popular microcontroller in the Texas Instruments MSP430 family is the MSP430G2553. The MSP430 features a 16-bit RISC CPU known for bringing balance between performance and power consumption needs [16]. Compared to other microcontrollers, the MSP430G2553 is known for its energy efficient qualities

and low power consumption modes that make it suitable for maximizing battery life. This comes with a slight tradeoff as it has somewhat limited processing power.

The MSP430G2553 has limited flash memory (16KB) and RAM (512 bytes). This is sufficient for simpler projects but may also be a constraining factor when dealing with the complexity of this level of project [16]. Developers must constrain their codes to meet these parameters.

The microcontroller comes with a range of integrated peripherals, including timers, GPIO pins, ADC, and UART. Because these features are built in, it saves the need to purchase additional external hardware. However, the MSP430G2553 does not have built-in USB connectivity. Another thing to consider is that the analog-to-digital converter has a 10-bit resolution, which may not be sufficient for some precision measurement applications [16]. If we find ourselves needing higher ADC resolution, we might need to consider purchasing external ADCs.

Like the MSP430FR6989, another benefit of using this microcontroller is that the team already has access to this microcontroller, as well as experience programming it. We also have access to Code Composer Studio, which simplifies the process and eliminates the need for us to become familiar with additional software environments.

### **3.3.3.3 Arduino Uno**

As one of the most popular microcontroller boards for hobbyists, the Arduino Uno is an open source affordable and great gateway into tinkering with coding and electronics design. With an operating voltage of 5v, the Uno uses an ATmega328P as its processor. Running at up to 16MHz it contains 32 KB of flash, 2KB of SRAM and 1KB EEPROM memory [17]. The ATmega328P works in tandem with an ATmega16U2. An 8-bit microprocessor with 16KB of ISP Flash, 512B of EEPROM and 512B of SRAM memory. This second microcontroller is used as a USB to serial converter, which enables a computer (which uses USB) to interface with the 328P microprocessor (uses Serial communication). The 16U2 also contains an interface known as debugWIRE that is used for debugging and programming [17].

As for The Uno's peripheral connections it has six PWM channels, useful for possible LED projects. One SPI and I2C controller, a 16-bit timer, two 8-bit timers, and a separate watchdog timer on an oscillator. There are also 14 GPIO pins (10 digital, 4 analog), one 3.3v and 5v rail, two ground rails and two analog I2C pins (SDA, SDL) [17].

While the Arduino Uno has some capabilities that MSP microcontroller boards do not it has disadvantages. This being the team's unfamiliarity with the software and functions of the Arduino family of devices. This leads to more time spent researching and learning to accomplish certain tasks that are done easily on the MSP platform due to prior experience.

### 3.3.3.4 Final Microcontroller Selection

For this project, we have decided to use the MSP430FR6989 microcontroller for its energy-efficiency and ease of use. The MSP430FR6989 can operate at a higher clock speed and can handle more computational complexity than the MSP430G2553. Though this does come with the tradeoff of using slightly more power, the team has decided that the MSP430FR6989 is still energy-efficient enough to handle our needs. Both MSP430 microcontrollers are capable of running on battery power and the slight tradeoff was an acceptable compromise.

Additionally, with 128KB of FRAM, the MSP4306989 has significantly higher memory capacity than the MSP430G2553. Both MSP430 microcontrollers share many of the same peripherals, though the MSP430FR6989 includes additional features such as USB connectivity, a real-time clock, and an LCD controller. More details are summarized in Table 1.

Because of the team's experience with the MSP430 brand of microcontrollers, the Arduino Uno was less strong of a contender.

**Table 1: Comparing Microcontroller Options**

Specification	MSP430FR6989	MSP430G2553	Arduino Uno
Manufacturer	Texas Instruments	Texas Instruments	Arduino
Architecture	16-bit RISC	16-bit RISC	8-bit AVR
Operating Voltage	1.8 V to 3.6 V	1.8 V to 3.6 V	5 V
Flash Memory	Varies (typically 64 KB to 256 KB)	16 KB	32 KB
RAM	2 KB	512 bytes	2 KB (including 1 KB EEPROM)
ADC Resolution	Varies (up to 16-bit)	10-bit	10-bit
Built-in USB	Yes	No	Yes
Peripheral Support	- GPIO pins - Timers - ADC - UART - I2C	- GPIO pins - Timers - ADC - UART	- PWM - SPI - I2C - Timers
Community Support	Limited	Moderate	Extensive

Programming Software	Code Composer Studio	Code Composer Studio	Arduino IDE
Price Range	\$20-\$25	\$10-\$20	\$25-\$30
Active Mode Current	100 $\mu$ A/MHz	330 $\mu$ A/MHz	6 mA at 16 MHz
Analog Features	High-resolution ADC	10-bit ADC	10-bit ADC

### 3.3.4 Cameras and Bookshelf

It is important to consider the cameras and bookshelf together since the books will be scanned by a camera mounted along the bookshelf rims. We must ensure that we buy the appropriate number of cameras to cover a horizontal view of the books in a single row. Then, it is ideal to choose one single camera per row that is capable of efficiently capturing the entire row by itself. In this way, we will have to spend less money and time configuring several cameras.

We begin by first analyzing the cameras and then compare bookshelves based on those findings because it is likely the cameras will be more costly. Comparatively, there will be much more variety in options when picking a bookshelf because, in general, it is an item that we only need to purchase one of, and it is a common household item.

#### 3.3.4.1 Raspberry Pi Camera Module v2

The Raspberry Pi Camera Module v2 is a popular option within the hobbyist's community. There are many resources available for configuring and setting up environments for it, and it is compatible with our option of the Jetson Nano computer. Its size is roughly 25 x 24 x 9 mm, weights approximately 3g, has a still resolution of 8 megapixels, and most importantly, has an adjustable focus with a focal length of 3.04mm (about 0.12 in) that gives it a 62.2-degree horizontal view [18].

The camera supports 3 different video modes which are of little importance in relation to the project since we will only use the camera for still images. Further details are summarized in Table \_\_ at the end of Section 3.3.4.

#### 3.3.4.2 IMX219-160 8MP Camera

This waveshare camera is similar to the Raspberry Pi Camera Module v2. They both use the Sony IMX219 image sensor and have a still resolution of 8 megapixels. This model of camera is a larger module at 25 x 24 x ~14 mm. It has a diagonal field of view of 160 degrees and a smaller focal length of 1.88 mm (about 0.07 in); however, its focus is fixed [19]. Relating it to similar cameras with

diagonal fields of view [19], it is estimated this camera has a horizontal field of view of 104 degrees.

### 3.3.4.3 IMX219-77IR 8MP IR Night Vision Camera from Seeed Technology

The last camera is also from Seeed Technology, and in contrast to the previous two cameras, it supports night vision with the addition of two IR LED modules [20]. The specifications for it only list its diagonal field of view to be 77 degrees but fails to mention its horizontal view. From lenses with similar diagonal fields of view like the Intel RealSense cameras [20], the horizontal field of view is estimated at 70 degrees. Its focal length is 2.96 mm (about 0.12 in), and its focus is fixed. An advantage of this option is that it would help to reduce complications that may come from the bookshelf overhead shadow on the books. The same camera is available without night vision at a cheaper price.

### 3.3.4.4 Final Comparison of Cameras

Because our team currently has one IMX219-160 8MP ready to be used and due to the various resources available when paired when the Jetson Nano, we chose it as our final camera selection. We will purchase two additional cameras at the beginning of Senior Design II. All three cameras were similar in price and while we did choose the more expensive option, the other cameras were not significantly cheaper. Contrast and comparisons can be seen in Table 2. The IMX219-160 8MP is small and can easily be mounted onto the edge of the bookshelves without creating an obstacle for people browsing nearby.

**Table 2: Comparing 3 Camera Options**

Camera	Price (\$)	Horizontal FOV (degrees)	Focal Length (mm)	Focus Fixed or Adjustable	Still Resolution (Megapixels)
Raspberry Pi Camera Module v2	25	62.2	3.04	Adjustable	8
<b>IMX219-160 8MP Camera</b>	27	104	1.88	Fixed	8
<b>IMX219-77IR 8MP IR Night Vision Camera</b>	21	70	2.96	Fixed	8

### 3.3.4.5 IKEA Billy Bookshelf

This white IKEA bookshelf is 80 cm wide, 28 cm (about 11.02 in) deep, and 106 cm tall [21]. It has three rows and can be bought at our local IKEA store. Although

it is affordable, a problem could be that it is not deep enough since there should be sufficient space between the top rim of a row and a reasonably wide book so that the camera can capture all the books in the row. The package weights approximately 44 pounds.

### 3.3.4.6 ClosetMaid Bookshelf with 3 Shelf Tiers

This white ClosetMaid bookshelf is 76.2 cm wide, 34.49 cm deep, and about 112.47 cm tall [22]. Depth wise, this is a much better option than the previous. We would have more room to focus the camera and the angle would be less likely to cause problems. However, it is taller and we lose some of the width offered by the Billy bookshelf. Transporting it might be harder considering its height and weight of 38 pounds and it has to be ordered on Amazon.

### 3.3.4.7 Better Homes & Gardens 3-Cube Storage Organizer

The Better Homes & Gardens organizer comes in various colors, is 41.70 cm wide, 39 cm deep, and 100.11 cm tall [23]. A problem with this one is that we lose a lot of the width when compared to the previous two bookshelves discussed. However, we get significantly more depth which is vital to the implementation of our project. It is also important to consider the width, however, because it will allow us to place more books on each row and therefore better showcase the capabilities of our project. The package weighs 28 pounds and would have to be ordered from Walmart.

### 3.3.4.8 Final Comparison of Bookshelves

Because each option has its own advantage over another, it was a tough decision for the team to select one of the three. After careful comparing and assessment of potential problems that could arise from each of the options, we chose the Better Homes & Garden storage unit as it gives the most promising depth to ensure our camera captures a good angle of the books on a single row. When building our final physical project, we opted to using two of the cubes to fill with books and the middle, last cube was used to store all electronics required for our project. To cover up this electronics, we added an acrylic painting with our team logo on it. We had no issues with the shelf and were able to safely drill and mount the rods for the cameras.

**Table 3: Comparing 3 Bookshelf Options**

Bookshelf	Price (\$)	Width (cm)	Depth (cm)	Height (cm)	Package Weight (lbs)
<b>Better Homes &amp; Garden</b>	54	41.7	39	100.11	28
<b>Billy</b>	49	80	28	106	44

<b>ClosetMaid</b>	81	76.2	34.49	112.47	38
-------------------	----	------	-------	--------	----

### **3.3.5 LEDs**

The LEDs present on the ROBOT bookshelf are integral to the design, as they help guide the library patrons to the specific book that they are looking for on the shelf. To accomplish this, simple LEDs will not suffice, as in a simple strip all the lights are lit simultaneously. To achieve our specific function, individually addressable LEDs are required to indicate exact positions of the desired novel. For the LEDs to be addressable they must be able to communicate with at least one of the communication protocols, these being: SPI, I2C, DMX, or PWM.

#### **3.3.5.1 SMD LEDs**

One way of accomplishing this functionality is to procure multiple SMD addressable LEDs and fabricate a PCB to solder them to. This approach can be tested before fabrication with a breadboard and an LED controller. However, this method may introduce a considerable amount of error and troubleshooting to function. This is due to possible human error from soldering leads of the LEDs incorrectly as this can cause damage to the PCB and other LEDs requiring some components to be reordered. Furthermore, the cost of each single LED must be considered and taken into account as well as buying extras in case of failure, which leads to increased cost.

Another option is to use an already assembled strip of individually addressable LEDs where the chance of human error from soldering is eliminated. This is the more viable option, and the one that we will be choosing, however there are many individually addressable strips to choose from, all of them having their strengths and weaknesses. These choices being specifically: WS2811, WS2812B Eco, WS2813 and SK6812 [24].

#### **3.3.5.2 WS2811 LED Strip**

Starting with the WS2811 which is a 12- volt strip that is compatible with most libraries for control and monitoring. The main reason why this strip runs on 12v and not the standard of 5v is to reduce the effect of voltage drop across the LED strip at long distances. For instance, an LED strip with input voltage of 5v may have a voltage drop of up to 2.5v meaning the lights closer to the end of the strip are very color inaccurate and dim compared to the beginning of the strip [24]. While a 12v strip is one way to solve this issue, it introduces another component as a separate power supply separate from the microcontroller is needed to run the LEDs. Power injection in the 5v strip can also fix the inaccuracy problem; however, it may have to be done in multiple places on the strip for longer runs. The main disadvantage of the WS2811 is that the strip does not have truly individually addressable LEDs

as it is addressable in groups of three known as pixels. However, the other lighting options do have LEDs that can be changed one by one [24].

### **3.3.5.3 WS2812B Eco LED Strip**

Next are the WS2812B Eco strips, which run on 5v meaning they can run off of just the power from a microcontroller without supplemental power. It is not only the cheapest option, but it also has a low idle power consumption and active power draw, making it suitable for low power applications. The biggest deterrent for avoiding this strip is there is only one line for data present on the strip, and if one pixel is dead, every LED after that nonfunctioning one will not work.

Another important detail to note about this strip is that at their maximum capacity, all LEDs lit up white at full brightness requires 5V 20A power. The custom PCB's microcontroller is unlikely to provide such amperes, however, we can potentially either power the LED strips through an external power supply or build a power conversion with a voltage regulator that can take 20A at its input. This is likely to require a buck regulator that is able to convert 5V into 3.3V and above. Regular linear voltage regulators can typically only take and output smaller amounts of current.

### **3.3.5.4 WS2813 LED Strip**

This is where WS2813 LED strips come in. This product solves this design oversight by including two serial data lines for redundancy, so as long as two nonfunctional LEDs are not adjacent to each other the other parts of the strip will still function. The main con of this option though is its considerable increase in cost.

However, these strips are significantly more expensive and if an LED were to malfunction on the other strip options mentioned, the nonworking LED can easily be cut off and the other two ends re-soldered back into one.

### **3.3.5.5 SK6812 LED Strip**

Finally, the SK6812 Also runs on 5v like the previous two options, yet it is distinctly different. In a standard addressable LED, there are three color channels: Red, Green, and Blue. To create the color white, all three channels are used at their it's highest setting, causing a substantial increase in power draw. However, the SK6812 has these standard channels as well a specific one for white. This is done not only for color accuracy, (as White with only RGB channels is a blue-ish tint) but also for an increase in efficiency.

### **3.3.5.6 Final Decision on LED Strips**

Out of these four options, we are still discussing what LEDs strip to choose. Mostly because it relies heavily on the communication protocol that we end up choosing. The MSP430FR6989 does not have native support for most of the listed LEDs. To combat this the singular SMD LEDs may have to be soldered together in series to

create a compatible LED strip. If this approach is used there will have to be redundancies similar to the WS2813 LEDs. Furthermore, based on the operating voltage of the selected SMD LEDs power injection may have to be employed as well.

It is important to note that many users have taken the time to create libraries like the Arduino NeoPixel—which is made precisely for LED strips such as the WS2812B type. These libraries have the same functionality as the NeoPixel library and are made specifically for the MSP430 family of microcontrollers. If we were interested in selecting such a strip, we could reference other hobbyists and see how they were able to accomplish interconnecting the LEDs and the MSP430.

Table 4 below shows an overall summary of all the LED strip options.

**Table 4: Comparing 4 Led Options**

LED Strip	Operating Voltage (V)	Color Channel	Individually Addressable	Redundancy	Power Injection needed	Price (USD)
WS2811	12	RGB	No (3 LEDs)	No	No	<a href="#">\$16.99</a>
<b>WS2812B Eco</b>	<b>5</b>	<b>RGB</b>	<b>Yes</b>	<b>No</b>	<b>Yes</b>	<a href="#">\$15.99</a>
WS2813	5	RGB	Yes	Yes	Yes	<a href="#">\$30.99</a>
SK6812	5	RGBW	Yes	No	Yes	<a href="#">\$27.99</a>

### 3.3.6 PCB Vendors

For prototyping for ROBOT, a breadboard and wires will be used to test our specifications for the product. However, after this prototyping phase, a full featured PCB must be created to fulfill the functionality of the breadboard implementation. With the creation of the PCB comes the challenge of choosing a vendor to review and manufacture the given design. Choosing a vendor is based on a myriad of factors, such as age of the business, tolerances, capabilities, price, etc.... Based on the previously stated factors, our team has narrowed down the choice of vendor to three: JLC PCB, PCBway, and Würth Elektronik.

#### 3.3.6.1 JLC PCB

JLC PCB was founded in 2006 in Shenzhen, China and since their creation, has become one of the top five PCB manufacturers in the World. They can manufacture up to 20-layer PCBs in FR4, Aluminum, Copper, and PTFE Teflon. With a max dimension board size of 400x500mm and very tight tolerances to prevent variance between PCB batches. JLC PCB also has fast manufacturing and shipping for prototyping, however, most of these orders require a specific amount of a prototype to make an order known as Minimum order quantity or MOQ [25].

While JLCPCB does not have the most experience out of our potential options, it has a balance of both good customer service and support, and good adherence to given specifications. Furthermore, it is a great choice financially as their prototyping costs are low and reasonable.

### **3.3.6.2 PCBWay**

PCBWay is by far the most advertised and publicized PCB manufacturer on the Internet. While recently being founded (2014), they have over 200,000 clients and are used in 150 different countries for PCB manufacturing and prototyping. Additionally, PCBWay also provides CNC machining and 3D printing services [26]. Meaning if needed, the creation of a product or project can be prototyped, tested, and manufactured all by the same vendor, giving PCBWay a slight edge over its competitors. However, for Our project, the only service that will be judged is its PCB manufacturing division.

Compared to its competition, PCBWay has less satisfactory customer service, and is more likely to make mistakes in the specifications for orders [26]. This may be a problem as ROBOT may not function these specifications are not met. In addition, if an issue arises, it will be harder to pinpoint if it is a problem with the design or a mistake made by the vendor, leading to possible added troubleshooting that will halt the project's progress.

### **3.3.6.3 Würth Elektronik**

Compared to the other two manufacturers, Würth has by far the most experience in the PCB manufacturing space. Founded in 1971, They are the leading PCB manufacturer in Europe with a majority of their manufacturing plants located in Germany [27]. The main advantage of Würth compared to their competitors is the company's unrivaled customer service, as they have no MOQ requirements, Design support from industry experts, and fast delivery and free samples for fabrication and prototyping. In spite of these advantages, the main deterrent for small designers to work with Würth is the price of manufacturing, being higher than most competitors in the space.

While the pricing for prototyping and manufacturing is higher, Würth's immense amount of quality control will lead to next to no errors in the specifications, meaning if there is a problem it will lie in our design, which we can consult the manufacturer

and get a solution to design flaw. Making Würth a better choice due to their expertise and customer support.

### 3.3.6.4 Final Comparison of Manufacturers and Decision

Based on the current scope of ROBOT, all three vendors have the minimum specifications needed to manufacture the PCB, meaning this decision comes down to other factors shown in Table 5. After careful consideration, the vendor that we will be using JLCPCB. ROBOT will be using an FR-4, and out of the three vendors, JLCPCB has the most economical prototyping costs as well as a fast turnaround time. In addition, if any design issues are discovered, JLCPCB will be willing to aid in troubleshooting. While Würth Elektronik has better customer service and designers, the increase in cost, and lead time concerns being based in Europe (VAT taxes and regulations). As for PCBWay, it is remarkably like JLCPCB, but is laxer with specifications. They can be used as a backup manufacturer but mistakes with manufacturing must be accounted for if used.

**Table 5: Comparing 3 Vendor Options**

Vendor	Materials (Simple PCBs)	Layer Count	Max Dimension (mm)	Board Thickness (mm)
<b>JLCPCB</b>	FR-4	1, 2, 4, 6	400mm x 500mm	100.11
<b>PCBWay</b>	FR-4, Aluminium	1-10	500mm x 1100mm	106
<b>Würth Elektronik</b>	FR-4,	1-20	570 mm x 500 mm	112.47

### 3.3.7 Display Screen

Selecting the right display screen for ROBOT is a major decision that has a significant impact. First and foremost, the display screen would have to be compatible with the Raspberry Pi or the Jetson Nano. This is the primary requirement that the display screen must meet for ROBOT to function properly. We already own a Raspberry Pi or Jetson Nano, which is why the display screen needs to be compatible with them. Therefore, it's sensible to select a device that is compatible with something we already own to reduce the cost. We would also like to include the size as a requirement as well as the display screen to be a touch screen. From there, we can assess which would be most appropriate for ROBOT by comparing additional specs and budget.

The display screen's size is significant because it directly affects visibility. The price and screen size are typically closely correlated. The larger the screen, the higher the price usually is. Since this project has limited funding, it is essential that we find a display screen that is both affordable and small enough to meet all of the project's interactive and visual requirements. We determined that a 7-inch display screen would work best to accomplish that because the screen is intended to be used up close. The display screen doesn't need to be very large or complicated because it will primarily be used for book searches.

The display screen's ability to be a touch screen is the final and one of the most significant requirements. Given that the display screen will be fixed to a floor stand for a tablet or the bookshelf. Touch screen technology eliminates the need for a mouse or stylus, which are easily lost or stolen, making the device more user-friendly. With a touch screen device, users can interact with content directly through gestures like pinching, swiping, tapping, and zooming, which can make the experience engaging for them.

It can be challenging to locate 7-inch display screens that meet all three requirements because they are either too expensive, don't meet one of the requirements, or are located on a website we don't know much about and therefore don't trust. In the end, our group was able to compare three display screens: one from Walmart and two from Amazon

#### **3.3.7.1 FREENOVE**

One of the brands of display screens available on Amazon is FREENOVE. It is the least expensive option and satisfies all prerequisites. FREENOVE is a 7-inch display screen with a 5-point touch capacitive screen that is compatible with Raspberry Pi. One drawback to this is the 800x480 pixel display resolution and it cannot be mounted.

#### **3.3.7.2 HAMTYSAN**

The other display screen brand that is offered on Amazon is HAMTYSAN. This specific display screen has become a serious contender for our project because of how well it meets our three main requirements. Not only is this display screen suitable, but it is also an ideal fit for ROBOT. The Raspberry Pi 4 and the Jetson nano, in particular, are the cornerstones of our project, and this device is unbeatable in that regard because it is compatible with both of them.

#### **3.3.7.3 YLSHRF**

This was the only brand found outside of Amazon that met all the necessary standards and was a decent match for comparison. YLSHRF has a 7-inch display, a 5-point touch capacitive screen, and is compatible with the Raspberry Pi 4, just like the other display screens. It also has a better display resolution than HAMTYSAN.

### 3.3.7.4 Final Decision

There were plenty of other brands on Amazon that were suitable for ROBOT, but after a certain point, they all essentially functioned the same way; the only differences were in price and brand as shown in Table 6. After careful consideration, our group decided on HAMTYSAN primarily due to its cross-platform compatibility. Despite being among the priciest from the compared selection, it boasts excellent ratings for both product use and customer support, along with a respectable resolution. It is an ideal 7-inch size, touchscreen capabilities, and compatibility with our selected platforms make it the ideal choice for ROBOT.

It is marketed as a display screen for computers such as the Raspberry Pi, which the group does plan to use as a user interactive tablet at the edge of the bookshelf for searching and checking out books.

**Table 6: Comparing 3 Display Screen Options**

DEVICE	Price	Resolution / Pixels
FREENOVE	\$49.95	800 x 480 Pixels
<b>HAMTYSAN</b>	\$69.99	1920 x 1080 Pixels
YLSHRF	\$70.15	1024x600 Resolution

### 3.3.8 RFID Sensors

RFID or Radio Frequency Identification is a system that consists of a tag and a sensor/reader [28]. The reader sends out signals through either electromagnetic fields or electronically that contain data and are intercepted by the tag portion and relayed back to the receiver. The collected data can be used to track and provide the current state of the device.

The tag portion of RFID is made of a coiled wire (Either copper conductive ink) a transponder, and a substrate usually made of a flexible material or even a sticker. Then this tag is attached to an object so that it can be catalogued or tracked. There are two types of RFID sensing, these being passive and active. Passive RFID does not require a separate battery power which gives the tag a smaller footprint compared to active. For power they rely on the reader to supply the power for the data to be interpreted. However, the main disadvantage of passive solutions is their small range as the reader must be close to the tag to read it [29]. As for active RFID, they require an auxiliary battery to function taking up more space and, in some cases, must be maintained or replaced due to the battery. Despite this, active

systems have a significant increase in range compared to passive, meaning an object with an active tag can be tracked and contain other electronics such as temperature or a gyroscopic sensor.

### **3.3.8.1 Possible RFID Application**

For our needs, ROBOT may need to make use of RFID, specifically a passive system. This would be a fail-safe if computer vision is unable to identify the book on the shelf. As some books do not have designs or words on the spine, giving the computer vision software little to no information. The passive RFID tag would be placed inside the front cover of the selected book, and a reader would be placed on the back of the bookshelf keeping it in proximity for scanning. Using an active tag system for ROBOT is mostly out of the question, as the added bulk from the addition of a battery and different sensors makes it an undesirable choice being invasive to the reading experience for students.

### **3.3.9 Computer Vision**

Computer vision, a subset of artificial intelligence, makes it possible for computers to analyze and interpret information within images or videos. This includes the identification of faces, extraction of text, and analysis of colors, among other applications. By using artificial intelligence and machine learning, computer vision systems mimic the cognitive processes of the human brain to comprehend and make sense of visual data.

To implement computer vision, you mainly need a camera and a computer to process the information the camera sends. When the computer receives visual data from the camera, an algorithm is used to “convert[s] the image into a set of numerical data that can be processed by the computer” [30]. The algorithm then analyzes each pixel to identify patterns. While it is a powerful tool, there are also limitations to it including the “large amounts of data [required] to train and test algorithms,” power and time needed for processing, and the varying nature of images [30].

Python and MATLAB emerge as the predominant programming languages for computer vision applications. Both are easy to use alongside our chosen computer, the Jetson Nano, and there are many resources available for using either or. Throughout our undergraduate career, our group has worked with both in projects and classwork.

#### **3.3.9.1 Python**

Python is a high-level, general-purpose programming language that is highly acclaimed and accessible due to its “simple, easy to learn syntax” that makes it simple to read and modify over time [31]. Python is often praised for its readability and clarity, which makes it accessible for beginners and contributes to its

popularity. Python uses exceptions to handle errors, which provides a more structured way of dealing with unexpected situations than the segmentation faults common in languages like C. Python's approach to exceptions can contribute to easier debugging and improved productivity.

OpenCV (Open Source Computer Vision Library) is an open-source library compatible with Python, C++, Java, and MATLAB [32]. It includes over 2000 algorithms for machine learning and computer vision. It is useful for a variety of computer vision applications including detecting faces and identifying objects. OpenCV can be paired with EasyOCR (optical character recognition) to extract lines of text from images [33]. Another possible method of implementing text recognition using Python is using the Tesseract Python library (alongside OpenCV) that is also capable of extracting text from images.

In a study published in 2022 [34], it was found that EasyOCR outperformed Tesseract OCR by 5% accuracy when detecting license plate numbers real time. On the other hand, the research found that Tesseract is more successful in extracting letters and EasyOCR performs better with numerical characters.

### **3.3.9.2 MATLAB**

MATLAB is a programming language “based on matrix-language” used by engineers to design and analyze systems [35]. Through MATLAB and its software, users can analyze complex data, make algorithms, and create models [35]. It is advertised for its use in various academia applications “including deep learning and machine learning.” [35].

MATLAB offers a Computer Vision toolbox that has many algorithms and functions catered for computer vision, image and video processing, and object detection and tracking [36].

The methods of implementing text detection and extraction of images using the Computer Vision Toolbox seems to require deep knowledge of machine learning and related algorithms. MATLAB also supports reading barcodes from an image [37], which could be an implementation the project could consider if extracting texts from images fails. MATLAB provides several resources and guidance on how to localize and extract barcodes from an image using the readBarcode and defining regions of interest [38].

### **3.3.9.3 Final Decision on Implementing Computer Vision**

According to Munawar [39], Python’s online community and support is much larger compared to MATLAB’s, probably since MATLAB is an expensive, paid software while Python can be freely accessible. Python provides support and capabilities of deploying projects on online platforms like databases [39]. Additionally, for the Jetson Nano Developer kit, which we will use for the project, mainly provides support for Python and C++ [39]. For these reasons, the team has chosen to move

forward with implementing computer vision using Python. First, we will attempt to successfully extract text from images. If we are not able to achieve reasonable accuracy rates, we will switch to instead detecting barcodes. Either method will still use Python and computer vision through the OpenCV library.

### **3.3.10 Application and Database**

To have a system that allows users to have a GUI to check book availability, location, and check books in and out, we need to have both an application and database. The application will be responsible for allowing user interaction and visualization of book information. Meanwhile, the database will be responsible for storing all book information including authors, titles, status, and location. The application will send and receive information to and from the database while the database will interact with the Jetson Nano to initiate book location LED light search. Then, we must make sure the database, application method, and Jetson Nano communication are all compatible with one another. The application will be based on Android. We will compare potential languages to use to build the application, options for databases, and ease of connecting the application, database, and Jetson Nano.

The two main languages used to build Android applications are Java and Kotlin. For Android app databases, Firebase, SQLite, and PostgreSQL are popular options.

#### **3.3.10.1 Application for Android Using Java**

Apps are built for Android through their IDE, Android Studio. There are thousands of courses and video tutorials and articles for using Java and Android Studio to make applications. Java has a wide variety of uses including web development and mobile app development [40]. Unlike programming languages like C, Java automatically takes care of allocating and freeing memory, meaning there is no worrying about corrupted memory. This significantly improves productivity. Another advantage of the Java language is its portability; if you begin a program in Java, for example, it is easy to then switch to Kotlin [41]. Because Java has been around since 1995, meaning it has a lot of support and documentation, a large community, and mature libraries.

#### **3.3.10.2 Application for Android Using Kotlin**

The Kotlin programming language is also used within the Android Studio and was created around 2010. For this reason, it is considered a fairly new language with not as much documentation available as with Java. Nonetheless, it has quickly raised popularity due to its quick learning curve coming from Java [40]. Kotlin was made with the goal to be easily adaptable to or from Java while also being more “efficient, concise, expressive, and safer” than Java [42].

### 3.3.10.3 Comparing Java and Kotlin

To give insight into how Kotlin is made simpler and more readable for programmers, consider the following Hello World programs. To simply print “Hello World” in Java we need 4 lines with nested statements while in Kotlin, we only need 3 lines and no nested statements.

Another key feature of Kotlin over Java is its ability to support type inference. In languages like C or Java, to declare a line of text, we would have to initialize a variable to type ‘string’ or ‘char string[].’ However, in Kotlin, you can declare a variable without having to specify the data type; it will be determined automatically at compilation [40].

Due to its easier learning curve, syntax, and formatting, the project will be written in Kotlin using the Android Studio. The program is promising to improve productivity and have large amounts of resources and tutorials, though likely less than Java. Kotlin is usable with Android applications and most popular databases like SQLite and Firebase.

### 3.3.10.4 Firebase Database

Firebase is a platform that, although not technically a database, offers numerous database services including real time data synchronization, authentication, and traffic analytics [43]. According to Google, Firebase is easy to use alongside various phone operating systems including Android and iOS [44]. It provides you the ability to:

- Authenticate new users
- Read data in real time
- Write data in real time
- Subscribe users to notifications
- Save images to the cloud

Singh mentions that the Firebase Realtime Database is a large JSON object that can be modified in real time [45]. The services provided can well eliminate the need for servers in the app development cycle [45]. If a user goes offline, the database allocates local cache to temporarily store changes and once it is online again, it is automatically updated across all devices [45].

Since we are implementing Firebase, we will need to learn about JSON objects. Important operations to perform include writing, reading, and deleting. JSON stands for JavaScript Object Notation and is a format catered for interchanging data. Key concepts of JSON include collections of name and value pairs and sorted lists of values [46]. As an analogy, these two items can be compared to structs and arrays in C, respectively.

### 3.3.10.5 SQLite Database

SQLite is a relational database. A relational database structures information into tables, rows, and columns, much like an Excel file [47]. Columns are attributes that “specify a data type” and rows are records that “contains the value of that specific data type” [47].

Like Firebase, SQLite allows the elimination of a server. A single file stores all the information needed, and this is where all reading and writing operations take place [47]. This is a popular option for IoT and embedded applications, websites with less than 100k requests per day, internal testing, and for data analysis [47].

Advantages of a relational database like SQLite include its flexibility. It is straightforward to modify specific data without making drastic changes to the rest of the information. Relational databases possess ACID properties which means they are atomic, consistent, isolated, and durable. Additionally, various users can access the file at the same time, but they cannot update the database at the same time.

#### **3.3.10.6 PostgreSQL**

PostgreSQL is another type of relational database. One of its notable strengths is its extensibility, allowing users to define custom data types, operators, and functions. Some data types are much like those in programming languages like C and include smallint, integer, and decimal, to name a few [48]. The system ensures data integrity through features like constraints and foreign key support, and it adheres to the ACID principles to maintain transaction reliability [49]. PostgreSQL is cross-platform, compatible with Linux, Windows, and macOS. With an already large community, it continues to be a popular choice for web applications, enterprise systems, and projects requiring a reliable and feature-rich database management system.

From looking at tutorials and articles and considering that the team has little experience with database implementation, it seems PostgreSQL is more advanced and rigorous than are Firebase and SQLite. Additionally, there are not many sources specific for guidance on interconnecting an Android project to PostgreSQL.

#### **3.3.10.7 Final Decision on Application and Database Implementation**

Kotlin is fully endorsed by Google as a coding language for app development. This is due to its integration of the latest Android features and taking advantage of more powerful phone hardware as well as introducing extension functions. Extension functions are a way of extending a class's (like a struct in C) functionality without the need to inherit from another class (taking attributes from class to another). This is especially important when there is a class from a library that cannot be altered. Another advantage of Kotlin is that it is more beginner friendly, as its syntax is easier to write and follow compared to Java. Lastly, the main draw of using Kotlin

is its focus on combating a big issue in app coding development: null references. Null references are when the written code attempts to access a variable where its value is null (different from 0). This causes many errors and vulnerabilities hard to pinpoint and fix in Java, leading to Kotlin's creation.

Together with Kotlin, we will be using a Firebase database. Firebase is also encouraged by Google for Android applications and there is full support and guidance from Google and various other sources for implementing the database on Android projects. Firebase provides the perfect features we need to make our project successful. For example, since we want to periodically update and upload book inventory, Firebase's feature of syncing data in real time will be used to make sure all current users have the most recent data possible. The database can also be accessed by several devices meaning that all visitors inside bookstore can access the database simultaneously. In addition, our Jetson Nano will also be able to access and update the database without preventing other devices from accessing book inventory. Lastly, it allows user authentication, meaning that we will be able to create user accounts so that they can potentially view information made specifically for them such as their current checked out books, due dates, and to-read lists.

### **3.3.11 Serial Communication**

Several different aspects of the project require serial communication. The form of serial communication selected must be common to both the microcontroller and the dev board. The Jetson Nano supports I2C, I2S, SPI, and UART communication modes. The MSP430FR6989 is also compatible with these modes. Additionally, the Raspberry Pi is also capable of all four of these modes.

Modern technology relies heavily on serial communication, which enables efficient data and information flow between systems and devices. It enables communication between devices over large networks and over long distances. Data can be sent bit by bit, consecutively, via a single communication channel. The two wires used in serial communication are commonly referred to as "TX" for transmit and "RX" for receive [50]. This is different from parallel communication, which sends numerous bits simultaneously over many wires. There are three different types of modes in which serial communication can operate: simplex mode, half-duplex mode, and full-duplex mode.

#### **3.3.11.1 Simplex, Half-Duplex, and Full-Duplex Mode**

In simplex mode, data can only flow in one direction; sender to receiver. It is a method of one-way communication, meaning when a transmitter sends data, the receiver can only accept it. Simplex mode is unidirectional. Examples of simplex modes include Radio and Television transmission as well as keyboards and monitors [50].

In half-duplex mode, both the transmitter and sender can be active, just not at the same time. It is also acceptable to describe it as a two-way method of communication—one activity at a time. Half-duplex allows data to flow in both directions, but not simultaneously. The way it operates is that the receiver can only accept the data if it is being sent by the transmitter. However, if data is being sent by the receiver, the transmitter can only receive it. A simple example of half-duplex is walkie-talkies [50].

Lastly, full-duplex mode allows data to flow in both directions simultaneously. Therefore, unlike half-duplex, the transmitter and sender can send and receive data at the same time with no restrictions. Both half- and full- and full-duplex are bidirectional. A few examples of full duplex include phone conversations, browsing the internet, and video meetings [50].

### **3.3.11.2 Synchronous and Asynchronous Protocol**

There are two different types of serial communication regarding the different clock signals that the serial devices require. The two categories are synchronous protocol and asynchronous protocol or synchronous serial interface and asynchronous serial interface.

### **3.3.11.3 Synchronous Serial Interface**

When data is transferred using a synchronous serial interface, both the sender and the recipient share a clock signal, allowing for synchronized data transmission. It is a type of communication that relies on synchronized timing using a shared clock signal. Typically, the data is transmitted in clearly defined frames or packets. A start, data, parity, and stop bits make up these frames but there are no bits added to the transmitter side. These bits are used to assist the receiver in correctly identifying and capturing the data. Many current digital devices and systems favor this approach because of its effectiveness and quick data transfer. Protocols that use synchronous serial interfaces that we are looking into for ROBOT are SPI, I2C, I2S, and USB. But there are also CAN, Microwire, and JTAG that use a synchronous serial interface [50].

### **3.3.11.4 SPI**

Serial Peripheral Interface (SPI) is another versatile and very popular communication protocol. SPI is different from UART, but shares many similarities as well. While UART is asynchronous and is known for being simple and easy to use, SPI operates in a synchronous fashion and offers a different set of advantages, disadvantages, and applications.

SPI is known for its high-speed, full-duplex communication. When rapid data communication exchange is crucial, SPI is an ideal choice. Unlike UART, SPI uses a shared clock signal, often referred to as the clock (SCK) signal, which keeps both

the transmitting and receiving devices in sync. This synchronization enables SPI to achieve faster data rates compared to UART [51].

SPI communication typically involves multiple wires or lines, including the SCK for clock synchronization, MOSI (Master Out Slave In) for data transmission from the master to the slave device, MISO (Master In Slave Out) for data transmission from the slave to the master device, and one or more select (SS/CS) lines to enable communication with multiple slave devices [51].

Unlike UART, SPI can support for multiple master devices, allowing for more complex communication setups that may involve several peripherals [51]. SPI's ability to connect multiple devices could be particularly valuable for a project that requires communication between various sensors, displays, or memory modules, and each device needs to be addressed individually.

#### **3.3.11.5 I2C**

I2C stands for the inter-integrated circuit, which connects different ICs or modules via two wires. Serial Data Address (SDA) and Serial Clock (SCL) are the two lines. SDA is the data transmission and reception line for the master and slave while SCL is the line on which the clock signal is carried [52]. Because it uses a 7 or 10-bit addressing system to locate the requested device on the I2C bus, it can connect up to 1024 peripherals. I2C pins are mostly used by modern microcontrollers in order to communicate.

I2C has a few advantages over SPI, including being easier to set up, ideal for onboard communication, and requiring fewer wires. I2C also supports multiple masters as well as multiple slaves. Additionally, I2C features an ACK/NACK bit that verifies the successful transfer of each frame. The acknowledge/no-acknowledge bit, ACK/NACK, is always present after a message. An ACK bit is sent back to the sender by the receiving device if an address frame or data frame was successfully received [52].

The downside to I2C is that its 400 Kbps speed is slower than SPI, and because it employs a 7 or 10-bit addressing system, it has fewer connected devices. In addition, the data bit frame size is limited to 8 bits [52].

#### **3.3.11.6 I2S**

Inter-IC Sound (I2C) and I2S methods do share certain relationships, they both have the initials "I2" since they are meant for inter-IC communication, and they were first created by Philips Semiconductors (now NXP) [53]. The "S" in I2S stands for "sound," and it has been designed to carry audio data. Through the use of a standardized interface for digital data transmission between ADCs, DACs, digital filters, digital signal processors, and other types of ICs used in audio systems, I2S is designed to simplify the development of audio electronics. It was created for stereophonic sound, or stereo, therefore it is a two-channel interface [53]. I2S, in

contrast to I2C, is intended to transfer a particular kind of digital data effectively. Transfer speed is especially significant since real-time serial transmission of two-channel, high-resolution audio uses a lot more bandwidth than the typical I2C or UART interaction obligations. I2S resembles SPI more than I2C. It is essentially the same setup used by an SPI implementation meant for unidirectional data transmission: "one signal for the clock, one for the data, and a third for word-level synchronization" [53].

The three primary signals of the I2S interface are Bit Clock (BCK or SCK), Word Select (WS or LRCLK), and Data (SD) [54]. I2S can be characterized by four key features: synchronous data transmission, stereo audio support, high audio quality, and adaptability. I2S is well-known for its excellent quality and low data loss during audio data transmission. Because of this, it is a well-liked option for professional audio systems and equipment intended for audiophiles [54]. By ensuring synchronous audio data transmission, I2S helps to eliminate timing-related problems between the transmitter and receiver. I2S is especially useful for stereo music since it has a Word Select signal that separates the left and right channels.

### **3.3.11.7 USB**

USB-A is the most common type of USB and can be found in most modern computers, laptops, smart TVs, and other devices. USB Type-A connectors, also known as plugs, are commonly found on peripheral devices and accessories such as USB flash drives, mice, keyboards, external hard drives, digital cameras, webcams, game controllers, mobile devices, and many more. These plugs can be inserted into Type-A ports [14]. For ROBOT, USB-A is important to have if we were to choose to have a wired camera.

### **3.3.11.8 Asynchronous Serial Interface**

Asynchronous Serial Interface does not have an external clock signal. It operates best for stable communication and is primarily used in long-distance communication. The parameters that this interface depends on are error-checking, synchronization, data bit selection or framing, and baud rate. While the data is being transferred, noise may cause an error. The error control method involves examining the "parity" to produce a stable output. This is called error-checking. Finding the beginning and ending of the data bits is the process of synchronization. The data frame is identified by the receiver after the transmitter connects the start and stop bits, respectively. The number of data bits that the transmitter is going to send to the receiver is framing. The most popular format is 8 bits. Different applications do, however, require different data bits. Lastly, the Baud rate, which is expressed in bps (bits per second unit) units, specifies the speed at which data is transferred from the transmitter to the receiver. The 9600 baud rate is the most popular. The speed of data transfer increases with increasing baud rate. The baud

rate needs to be the same on both the transmitter and receiver sides. Protocols that use asynchronous serial interfaces include UART, SBUS, and MIDI [50].

### **3.3.11.9 UART**

Universal Asynchronous Receiver-Transmitter (UART) is a popular, practical serial communication protocol used to exchange data between microcontrollers, computers, and embedded systems. UART is both simple, efficient and widely used. It allows for the transfer of serial data between two devices. Typically, they are a transmitter and a receiver. UART is asynchronous, meaning that data is transmitted without a shared clock signal between the devices. Instead, both devices must agree on a specific baud rate to ensure proper data synchronization [55].

UART communication typically consists of a start bit, data bits, an optional parity bit for error checking, and one or more stop bits. For two devices to be able to communicate, they must agree on the same frame structure and baud rate. UART can operate in simplex, half-duplex, and full-duplex modes. In full-duplex mode, the data is transferred bidirectionally simultaneously. Half-duplex is not simultaneous but communicates both ways. The information flows only one way in simplex communication. Simplex would likely not work for our Jetson Nano since we need information sent in two directions [55].

The MSP430FR6989 microcontroller can communicate with a Jetson Nano development board using UART. This will allow it to exchange data such as sensor readings.

One benefit to UART is it is simple to use and popular. We also have previous experience handling UART. On the downside, multiple masters are not possible with UART. Because of this, I2C is seen as more useful for connecting multiple devices. The communication mode we select should reflect our unique project needs.

### **3.3.11.10 Serial Communication Decision**

It is crucial to remember that, despite having a choice between UART, SPI, and I2C for our serial communication, the main considerations here are our level of experience with the technology and whether an asynchronous or synchronous serial interface is more appropriate for our needs with ROBOT. To summarize, while UART is known for its simplicity and ease of use, SPI provides a high-speed, bidirectional, and multi-master communication method. SPI will be used because it is fast, bidirectional, and can support the complexity of our project.

### **3.3.12 IR Sensors**

Infrared (IR) sensors are devices that can detect and measure infrared radiation. They are designed to sense heat or thermal energy emitted by objects [56]. These

sensors operate based on the principle that all objects with a temperature above absolute zero emit infrared radiation. The intensity of this radiation is directly related to the temperature of the object.

IR sensors consist of an IR source and an IR detector. The source emits infrared radiation, and the detector measures the intensity of the radiation emitted by objects in the sensor's field of view. There are various types of IR sensors, including passive infrared (PIR) sensors, which are commonly used in motion detection applications, and active infrared sensors, where an external IR source is used to illuminate the target [56].

IR sensors have widespread applications such as in security systems, motion detectors, and consumer electronics, where they enable touchless interfaces in devices like smartphones for features like using one's face to unlock phone. Additionally, IR sensors are utilized in industrial automation for tasks like temperature sensing and object detection. Their ability to operate effectively in low-light conditions and without the need for visible light makes them valuable [56].

The team is considering adding an IR sensor feature to the system to prevent the camera from snapping pictures when it detects a human is nearby. The human could possibly interfere with the image to be taken, so it would be best to halt the camera from taking pictures until it no longer detects a human nearby. There would be IR sensors near each camera and if triggered, the camera would enter a pause-job mode.

If we do not implement this, we run the risk of wasting processing time and power on images that might be too dark due to human shadow or images that may be partially or entirely blocked off by humans browsing.

### **3.3.13 Wireless Communication**

Wireless communication is essential for ROBOT's design. The two main methods considered for achieving this are Bluetooth and Wi-Fi. Below, each possible implementation is compared. The wireless communication will be the method in which the host computer and our microcontroller will communicate.

#### **3.3.13.1 Bluetooth**

A possible implementation for communication between the Jetson Nano and the LEDs is wireless communication. This is something to consider since the wiring might get too complicated and interfere with the book system and human interaction. Additionally, the Jetson Nano only has two camera slots, but we will need at least three cameras to be connected. This also requires super long wires for direct connection. Then, we might overload the Jetson Nano with devices to power on and overboard the small Jetson Nano case with too many wires.

By integrating a Bluetooth module directly into the LED control system, we can establish communication between the Jetson Nano and the microcontroller driving the LED lights. This simplifies the setup by reducing components and wires within the vicinity of the Jetson Nano. Bluetooth's Serial Port Profile (SPP) mode facilitates straightforward serial communication that will allow the Jetson Nano to send commands to control the LED lights. SPP replaces serial communication interface and "is great for sending bursts of data between two devices" [57].

Moreover, the implementation of Bluetooth communication aligns with power-efficient considerations. Bluetooth typically has lower power requirements compared to Wi-Fi, making it suitable for scenarios where power consumption is a critical factor. This choice contributes to an energy-efficient design; it allows the LED lights to operate effectively without adding a significant burden to the power supply. The simplicity and power efficiency achieved by implementing Bluetooth communication make it an attractive wireless protocol.

To implement this, we would use the HC-05 Bluetooth module that is compatible with the MSP430 family of microcontrollers. The module's Rx and Tx lines have a 3.3V level that can easily be detected by the MCU without any shifting necessary.

There are 4 connections to be made: VCC, GND, TxD, and RxD. Since the sent and received commands should be relatively small, there is little concern for the capacity of the RAM that the buffer might encounter by implementing Bluetooth.

Table 7 is an overview of the HC-05 Bluetooth module and some of its most important features to the project such as size and power required.

**Table 7: HC-05 Bluetooth Module**

<b>Feature</b>	<b>Description</b>
Internal ROM	6 Mbits
Power	1.8V
Communication modes	USB, UART
Transceiver and baseband frequency	2.4Ghz
Size	27mm x 13mm
Bluetooth class supported	Class 2 and Class 1 with external power amplifier
Operating Voltage	4V to 6V
Operating Current	30 mA
Protocol	IEEE 802.15.1
Baud rates	9600, 19200, 38400, 57600, 115200, 230400, and 460800

Size	27mm x 13mm
Bluetooth class supported	Class 2 and Class 1 with external power amplifier

Class 2 Bluetooth has an operating range of about 10 meters (30 feet), while class 1's operating range is about 100 meters (328 feet) [58]. For the application of our project, class 2 Bluetooth is more than enough since all the components to be used will be within the proximity of the bookshelf.

### 3.3.13.2 Wi-Fi

The Raspberry Pi model to be used alongside the display screen supports Wi-Fi protocol IEEE 802.11ac. Wi-Fi, short for Wireless Fidelity, is a technology that enables devices to connect and communicate wirelessly using radio waves [59]. The Wi-Fi protocol, formally known as IEEE 802.11, defines the standards and specifications for wireless local area networking (WLAN) communication. The protocol operates in the 2.4 GHz and 5 GHz frequency bands and provides a versatile means of connecting devices such as smartphones, laptops, tablets, and smart home devices to the internet and local networks without the need for physical cables.

The Wi-Fi protocol encompasses several standards, each denoted by a letter and a number (e.g., 802.11ac or 802.11n). Each letter and number reflect advancements and improvements in the Wi-Fi technology. These standards define properties such as transfer rates, frequency bands, and security features. For example, the latest standards, like Wi-Fi 6 (802.11ax), introduce enhancements such as improved data transfer speeds, better performance in crowded environments, and increased efficiency in power usage [59].

Security is a critical feature of the Wi-Fi protocol. WPA (Wi-Fi Protected Access) and WPA2 are common security protocols implemented to protect wireless networks from unauthorized access [59]. Encryption methods like WPA3 are designed to enhance the security of Wi-Fi networks by using stronger encryption algorithms that protect sensitive information from malicious activity.

### 3.3.13.3 Final Selection

The group has selected Bluetooth as the method of communication between the host and the microcontroller that will be on our custom PCB. This protocol will be in charge of receiving commands from the Raspberry Pi and sending acknowledgement bytes back to it.

## 3.3.14 Operating Systems

Two main operating systems we will be learning and interacting with the ROBOT project are Android OS for phones as the mobile application will be based on it.

Additionally, we will be interacting with the Ubuntu OS, which is a derivation of the Linux OS, for the Jetson Nano. Since all we will do using the Jetson Nano is run a Python script that accesses the camera data, it is more important to elaborate on the Android mobile OS and why we chose it over other popular ones such as iOS.

#### **3.3.14.1 Android OS on Mobile Phones**

Android is a widely used mobile operating system developed by Google. It is built on a modified version of the Linux kernel and is part of the larger Android ecosystem that includes the operating system itself and a range of applications, services, and development tools [60]. Android's open-source nature has contributed to its popularity by allowing device manufacturers to customize the software to suit their hardware and user interface preferences [61].

One of the key features of the Android operating system is its extensive customization options [61]. As many mobile phone users might know, while Apple may have a beautiful user interface, it lacks the customization and interconnection ease when compared to Android. Manufacturers often apply their own user interfaces, known as skins or overlays, on top of the core Android OS [62]. This supports a diverse range of user experiences across devices and applications. Additionally, Android's open nature has fostered a large community of developers who contribute to the platform's growth.

Each new Android version introduces enhancements in terms of performance, security, and features. For example, Android 10 introduced a system-wide dark mode, enhanced privacy controls, and improved gesture navigation [63]. Subsequent versions have built upon these improvements, with Android 11 focusing on conversation management and device controls, and Android 12 introducing a revamped user interface known as Material You that provides an even more personalized experience [63].

Android dominates the global mobile operating system. Its widespread adoption is due to its flexibility, broad device compatibility, and the variety of options it offers to both manufacturers and users [61]. While Android faces challenges such as fragmentation (different devices running different versions of the OS) and security concerns due to its open nature, ongoing updates and improvements continue to address and improve these issues.

While Apple is also a highly popular mobile operating system, we decided to use Android for our application due to its adaptability and range of customization. It also has a large hobbyist and online discussion community. Additionally, a team member had an off-use Google Pixel that could be shared with the group for the duration of the project which made it easier to collaborate and take turns modifying the app on our own time.

### **3.4 Final Component Selection**

The final components for ROBOT were selected after careful consideration. Factors such as price, availability, and ease of use went into consideration. A variety of different types of PCBs, firmware configurations, and hardware options was considered.

### **3.4.1 PCB and Firmware**

Printed Circuit Boards (PCB) and firmware are two important and essential components when it comes to technology because they are necessary for the functionality of electronics. PCBs and firmware have different roles and significance, but working together, they enable electronic devices to function.

#### **3.4.1.1 PCB**

An essential part of every technological device is a printed circuit board (PCB), which provides the actual framework or structure on which other electronic components are mounted and connected. PCBs are composed of an insulating platform that is carved or printed with conducting routes. This layer usually consists of fiberglass cloth and epoxy resin [64]. Depending on the details of the devices, PCBs can have a single layer or multiple layers. They also come in different sizes and designs. Electronic components such as sensors, connectors, memory, and microprocessors can be connected to a PCB. The components are connected to the board circuitry by soldering them onto metal pads [64]. PCBs are in charge of directing electrical signals between components, making sure that the signals needed for power, data, and control are sent correctly and efficiently. To reduce disruption and signal damage, the design and structure are essential. In addition to providing electrical connectivity, PCBs provide components mechanical support, which keeps them safely installed and secured. This is especially crucial for portable devices that could be subjected to physical strain. Electronic devices can be designed to be smaller and more compact than PCBs. A factor in the device's smaller size is the PCB's capacity to accommodate components that are close to one another. They can also be made to be adjustable, which allows the addition of new components or features without significantly altering the form of the device.

##### **3.4.1.1.1 Types of PCB**

PCBs can be split into three different categories: rigid, flex, or metal core [64].

The great majority of boards that a designer will come across are rigid boards, in which the board layout is enclosed within a hard layer of material made by a high heat and pressure lamination procedure [64]. These boards typically consist of FR-4 (fiberglass-reinforced epoxy-laminated sheet), but this can be changed to enhance specific board features based on the specific requirements of the design.

Flexible boards have far higher deflection because they are made of a less stiff substance. The Firmware board is typically much thinner than a typical rigid board,

and the material has a feel similar to a film roll. Although flexible boards are already widely used, it is hoped that they will bring wearable technology to the next level and eliminate the structural limits that rigid board devices currently have.

Metal-core PCBs are essentially an offspring of rigid board designs with an improved capacity to disperse heat throughout the board to safeguard delicate electronics. To avoid overheating and failure, this type may be a choice for high-current systems.

Given that rigid PCBs are the most basic and widely used board to configure, we will be using them for ROBOT. In addition, we were instructed on how to make one during Junior Design.

### **3.4.1.2 Firmware**

Firmware is a type of software that is embedded in the device's hardware and controls its functionality. The program is written into a hardware device's non-volatile memory which is a type of static random-access memory (SRAM). Meaning, that when a hardware device is shut off or loses its external power source, its content is kept in the non-volatile memory [65]. Firmware is an essential software layer that serves as a link between higher-level applications and hardware, such as the PCB. Hardware components must be configured, controlled, and managed by firmware for them to carry out specified tasks such as processors, memory, and storage. During a device's boot-up, firmware plays a major role since it sets up important hardware components, executes self-tests, and loads the operating system. Additionally, firmware configures hardware parameters such as clock speeds, power management, and communication protocols to make sure that the device operates at its peak performance. Firmware is also necessary for enabling security features to protect devices against tampering and unauthorized access, such as access control, encryption, and security boot operations. Lastly, updating firmware can fix bugs, enhance performance, and add new features.

#### **3.4.1.2.1 Types of Firmware**

Low-level: ROM chips, which are non-volatile read-only memory chips, are commonly used to store low-level firmware. Low-level firmware is part of the hardware meaning it cannot be updated or overwritten. Therefore, a device with one-time programmable memory has low-level firmware [66].

High-level: One kind of firmware that permits code changes is called high-level firmware. This kind of firmware is commonly found on flash memory chips. Because the code instructions in high-level firmware are more complicated than those in low-level firmware, it is more comparable to software than hardware [66]. High-level firmware is great for user interface, data processing, or general functionality.

Subsystem: Subsystem firmware is typically found in embedded systems. It is similar to high-level firmware because it can be updated. Subsystem firmware is also great for real-time responses.

For ROBOT, a combination of High-level and subsystem firmware will be used. The subsystem can handle the low-level hardware interactions while the high-level firmware can be used for user interface and application logic.

### **3.4.2 Hardware General**

PCBs are necessary for modern technology because they provide the complex connectivity that makes electronics function. Although the PCB is a necessary component of hardware, its effectiveness is determined by the components that are installed on it. This section briefly explores the wide range of hardware components that can be found on our PCB, emphasizing their functions and importance. These components include resistors, transistors, inductors, capacitors, and more.

A basic component that prevents electrical electricity from flowing is a resistor. They are key in regulating the circuit's current and voltage levels, which guarantees that different components get the proper amount of power. Resistors are used in voltage division and current regulation.

Electronic switches known as transistors can regulate or enhance current flow in a circuit. They serve as the foundation for both analog and digital amplification. Transistors are essential in a variety of applications because they allow signals to be manipulated.

Coils of wire that store energy as a magnetic field are called inductors. They are necessary for energy storage, the creation of magnetic fields for a variety of applications, including inductance-based sensors, and the filtering of high-frequency noise.

An electrical field is the form that capacitors use to store and release electrical energy. They are necessary for a circuit's filtering, alignment, and voltage level stabilization. Together with other components, capacitors provide reliable performance.

The brain of the system is composed of integrated circuits (ICs) and microcontrollers. Although integrated circuits (ICs) carry out a variety of tasks like signal processing, amplification, and logic operations, microcontrollers are tiny computers that carry out program instructions.

Diodes are used as electrical check valves since they only permit current to flow in one direction. They are essential for rectifying, protecting against reverse voltage, and converting alternating current (AC) to direct current (DC).

Connectors are components that make it easier to attach other PCBs or external devices. They enable the development of electronic systems and the exchange of power and data.

Lastly, fans keep the PCB operating at peak efficiency. Despite being frequently overlooked, they are essential for controlling the temperature. Fans enable the PCB to function within acceptable temperature limits, minimizing overheating and component failure by dissipating heat produced by components.

## **4 Standards and Constraints**

ROBOT's ability to be completed successfully depends on how well standards and constraints are taken into account. Standards provide a foundation for reliability and consistency by acting as criteria to guarantee the compatibility, safety, and quality of a product. Meanwhile, constraints serve as the limits that our project has to work inside; these parameters include time, safety, and other significant factors. Maintaining standards while managing limitations involves strategic planning. Throughout the course of the project, standards and constraints encourages efficiency and effectiveness in addition to assuring the quality of the finished result.

### **4.1 Constraints**

When dealing with any engineering project, there are a multitude of constraining factors to take into consideration. A project that is inefficiently funded will not be profitable or marketable. An ethical engineer should not create a project that causes harm to the environment, society, or other people. Safety is paramount when dealing with products that are marketed to other people. By abiding by engineering constraints, we can better assure that our project is safe, profitable, and ethical.

#### **4.1.1 Economic Constraints**

ROBOT is not funded by any organization, professor, research group, or company. Because this project is entirely self-funded by college students, keeping the costs low and affordable is paramount. In order to ensure affordability, the team has taken multiple steps to cut costs.

To overcome the economic constraints and maintain affordability as a core principle, the team has taken several measures to make sure the cost of the project does not get out of hand. One thing the team has done is look into open-source software and tools as a strategy to save costs. Using open-source platforms not only reduces software licensing costs but also provides access to many community-developed resources. One such open-source resource is OpenCV (Open Source Computer Vision Library). By using open-source materials and software provided for free to students such as MATLAB, we can avoid costs associated with buying software.

Rather than opting for expensive, pre-built solutions, the ROBOT team has also chosen to adopt a do-it-yourself (DIY) approach. This includes the assembly of the bookshelf itself. By self-assembling a bookshelf kit as opposed to buying one premade, we can save on assembly costs. The team has been resourceful in sourcing materials and components, often finding affordable alternatives without

sacrificing quality. This approach allows for cost savings on essential hardware and components while not sacrificing important functionality.

Another method of keeping costs low utilized by the team is by using microcontrollers and similar equipment that has already been purchased for previous classes. By reusing old equipment, we can avoid the costs associated with purchasing new equipment. In addition to reusing old equipment bought by the team members, the professors who mentor us may be able to loan out microcontrollers.

Because ROBOT is intended to be a cost-saving project used by places such as public libraries, keeping the production costs low ensures that the product is not only marketable, but competitive. Since it has been established that other smart bookshelf concepts exist, ROBOT must be created efficiently and cheaply in order to be of use.

Another method of keeping costs low is to use places such as thrift stores in order to purchase second-hand technology. By using second-hand and gifted devices, we will avoid having to make large purchases. For instance, the screen most likely will not need to be purchased new, and it does not need to be state of the art technology. Any screen that displays the data cleanly will suffice. The team should take care not to make superfluous purchases that will increase our bill. For development purposes, we will be reusing an Android phone donated by a team member's mother.

Finally, one area in which we must save money is printing our PCB. Though this is an unavoidable expense to a certain extent, we can choose cost-efficient vendors that minimize unnecessary expenses. Additionally, we can save on costs by being careful with our equipment so we do not need to buy more.

## **4.1.2 Environmental Constraints**

It is essential to ensure that the environment is not harmed due to the smart bookshelf. To this end, our team must consider the safety of wildlife, aim to use renewable and low-power energy sources, and minimize waste.

Though our device is not intended to come into contact with wildlife, there are some scenarios during which this may occur. Mainly, wildlife may encounter the bookshelf after the bookshelf has served its purpose and is thrown away or recycled. It is essential that the bookshelf is made of material that is non-toxic, safe, and recyclable as much as possible. Wildlife may be poisoned or made sick by toxic paint. To prevent this from occurring, we should avoid painting the shelf with potentially toxic paint. Any paint or stain used to decorate the bookshelf should be thoroughly researched for environmental risks. Only those that are environmentally friendly and non-toxic should be used.

Additionally, the bookshelf may become dangerous to wildlife if made from dangerous materials. Materials such as wood are typically considered safe due to being found in nature and easily decomposable. Artificial materials may pose a variety of other risks, such as toxicity, inability to decompose, or tendency to physically injure wildlife. Choking hazards and microplastics are a concern specifically attributed to many plastics.

Small parts and electronics should be especially carefully monitored to make sure they are not disposed of in a way that could be detrimental to the environment. Many technology vendors offer options to safely dispose of or recycle technological parts. Safe disposal should be kept in mind during the testing phase when excess or broken parts may need to be recycled.

Looking into low-powered options and reusable energy will also help reduce our environmental footprint. To this end, one of our stretch goals will be to look into using a reusable power source to power the bookshelf. There are several possibilities that might make this possible. If there is sufficient time, we might investigate using a cheap solar panel for bookshelves that are placed near windows.

Another factor of environmental safety that must be considered is the hazard involved with packaging our parts and product. Because our bookshelf contains many parts that must be ordered and assembled, lots of packaging will be required. Additionally, if this product were to be sold, it would require many packaged parts for the same reasons. In order to prevent the packaging from becoming dangerous, it should be made out of compostable materials such as cardboard as opposed to plastic or Styrofoam. Packaging should also be broken down and compressed before disposal. This will prevent wildlife from being poisoned or strangled by our packaging.

### **4.1.3 Ethical Constraints**

Though the smart bookshelf has many useful applications, it is important to consider ethical constraints as well. One of the primary ethical concerns is that this technology might intrude on people's privacy. The wide-angle cameras on each shelf have the capability to capture images of people as they browse or interact with the books. There are many reasons why some people would wish to avoid this. Some people distrust technology in general and do not consent to being surveilled. Others may feel that being recorded is against their religion, sense of safety, or personal beliefs. To address this concern, it is important to make sure that users are notified and are aware of the presence of the cameras. This way, they may decide for themselves if they are comfortable using the product. Additionally, we can mitigate concerns by making sure we do not record more of people's bodies than is needed. There is less potential for misuse when only recording people's hands as opposed to their full bodies.

Another privacy-related concern is that users may not be comfortable with information about what books they interact with being stored in a database. Some people may want to view reading material that is controversial, potentially offensive, or sensitive. Because privacy is essential, we must ensure that the data we collect is not sold, distributed, or shared in any unauthorized way. Additionally, we should not collect any more personal information about customers than is strictly necessary.

Because this project implements computer vision, it is also important to ensure that it is implemented in a way that is fair and inclusive. A common point of frustration among many users of similar technologies is that poorly trained systems might discriminate against people. If certain groups of people are not included in training the model, it might not properly detect them. We can take steps to mitigate this by using a diverse population in the training of our model.

Lastly, it is essential to make sure that users are protected against inaccuracy. Accuracy in this case refers to the system's ability to correctly identify and track the presence or absence of books on the shelves. Inaccurate readings can result in users receiving incorrect information about book availability, which can lead to frustration and a loss of trust in the system. To address this, regular calibration and maintenance of the CV system is essential. Continuous testing, fine-tuning, and updates to the system should be carried out to minimize errors and improve accuracy. Because our product has anti-theft applications, these steps will help us avoid false positives.

#### **4.1.4 Health and Safety Constraints**

Though a smart bookshelf project may not seem to be dangerous, there are still several health and safety constraints that must be taken into account. Firstly, cable management is essential to consider. Loose wires in an area exposed to many people's hands may be dangerous for several reasons. A hand stuck in a wire might cause books or the entire bookshelf to fall, injuring the customer. Additionally, exposed electrical wires could be both a fire and a shock hazard. Because the project deals with books, flammable machinery is particularly dangerous. Lastly, small children might bite the wires, causing serious injury. To mitigate these concerns, we must minimize the use of exposed wires, and make sure wiring is securely fastened.

Additionally, because the bookshelf contains many parts, each must be routinely monitored and maintained to make sure it is functional and not dangerous. The wide-angle camera could easily become dislodged and fall on a customer. The glass from the lens might also shatter and hurt someone. The best way to mitigate this is to make sure that the camera is very securely fastened to the bookshelf.

Another possible concern is that since this project deals with LED lights and digital screens, the lights should not be bright enough to cause damage to people's eyes. Luckily, this can be easily avoided by following regulations and not using LEDs that are brighter than they need to be. Because some people with epilepsy might be sensitive to flashing lights, we can mitigate harm by placing a warning sign within the vicinity of the shelf.

Another possible health risk is that the bookshelf itself might be easily tipped over. A small child climbing the shelf, or a clumsy adult might find themselves hurt by the bookshelf. This product may not be ideal for children because of the amount of technology involved. A good way to avoid injury would be to market the bookshelf to older audiences and ensure the bookshelf is secure on the ground. This means the ground should be smooth and even. Additionally, books should not be placed in a top-heavy manner that would destabilize the bookshelf. Heavier books should be avoided for demonstration purposes. The bookshelf should be stocked with books that are a safe weight to carry and are well-cared for. This will prevent customers from damaging their wrists or getting infected paper cuts when using our product.

#### **4.1.5 Sustainability**

A smart bookshelf project contains several sustainability risks that must be mitigated in order to maximize the potential of the product. Our goal is to have a product that is sustainable and environmentally friendly. To this end, we pledge to take environmental concerns into consideration in every step of development.

First and foremost, the bookshelf itself should be sourced sustainably. For our demonstration we will seek an existing bookshelf that can be repurposed so we do not have to waste resources on another bookshelf. Sustainability was a factor in deciding which bookshelf to use, as we wish to avoid unnecessarily large bookshelves, bookshelves made out of cheap and unsustainable material, and toxic paint.

The addition of electrical components is one of the other largest concerns. In order to preserve the functionality of the prototype and keep it as sustainable as possible, the electrical components must not be subject to unnecessary wear and tear. Though the bookshelf contains a night mode feature for monitoring the inventory during closing hours, it might be more efficient to find times when security threats are low during which the devices can be powered entirely off.

In order to maximize sustainability, the team must strongly consider power consumption. Components should be selected with energy-efficiency in mind so as to minimize power usage. This should be done both during the design stage and the parts selection stage of the project.

Another thing we can do to aid in the sustainability of our project is to utilize low power modes whenever possible. By dimming the lights on the screen and configuring the microprocessors correctly, we can avoid unnecessarily wasting power where it is not needed. A combination of minding our hardware and software will be necessary for this.

Longevity is another factor to consider when dealing with sustainability. Some parts that see heavy wear and tear might eventually need to be replaced. In order to prevent parts from being treated as disposable, we should caution patrons to treat the bookshelf with care. Additionally, part selection plays an important role here. We must choose parts that are built to last, even if they are slightly more expensive than cheaper, flimsier parts. For instance, our choice of LED should not burn out easily, as having to constantly replace the LEDs would be wasteful.

Finally, we must take recycling and end-of-life responsibility into consideration. The disposal of smart bookshelf components and materials at the end of their lifecycle can be environmentally harmful. Sustainability constraints should promote the recyclability of components and materials, as well as establish clear responsibilities for the safe disposal and recycling of end-of-life products.

#### **4.1.6 Manufacturer Constraints**

As well as the bookshelf portion of the ROBOT project having specific constraints the design, prototyping and construction of the PCB and other electronic components may be limited to factors out of our control, which may solely be based on the manufacturer. One of the main aspects of this project is to create a project that can be replicated, and to achieve this, certain things must be considered.

One of the constraints that may pose the most effect on the project is component availability. If specific, non-standard components are chosen and are relied upon in the design, how will the design fare when that specific component is not available or becomes obsolete? It will lead to massive lead times for the product to be completed and may cause the project to be abandoned. To combat this, the components chosen in our design must have 2-3 alternatives that can be used as a substitute to account for this possible hinderance.

Another large constraint of manufacturing is meeting the explicit specification given by the customer for fabrication. Not following the design provided may lead to a substantial increase in lead time and more importantly cost. The mistake will require PCBs or other components to be reordered from the same or different manufacturer. In many cases, the manufacturer also requires a MOQ to be ordered leading to the cost increasing dramatically. Having this constraint constantly appear in development is a detriment to the process. This, however, can be prevented by closely communicating with the manufacturer on the progress of the

product and giving feedback on mistakes when they are discovered, saving the time and money of both the vendor and the consumer.

Intellectual Property regulations may also affect the manufacturing process. This is because if protected technologies are used in the design of a product or project, it is subject to be delayed until permission to use technology is granted or a license is gained. If production of the product proceeds without acknowledging this factor, it opens the manufacturer and the designers up to litigation and may end the project.

#### **4.1.7 Social Constraints**

The goal of ROBOT is to provide a product that is time-efficient, cost-efficient, and useful. That being said, there are several socially constraining factors that the team must consider when developing this project. One social constraint is that we must gauge the overall desire for a product like this to exist on the market due to real or perceived difficulty using the product. Adding technological parts to a simple task like locating a book is seen unfavorably by some people. This is because some people may have a difficult time with technological literacy, meaning they would not appreciate having to use an app to locate books. One way we can address this constraint is to make it clear that using the app is optional and users who want to locate books for themselves are still free to do so.

Another social constraint is related to user privacy. Due to the use of sensors, cameras, and databases, some users may feel their privacy related to their reading habits is being infringed upon. By angling the camera correctly and ensuring it is not capturing more than it needs to, we can ensure that people do not feel infringed upon then using our product.

Finally, another limiting factor related to social constraints is that book vendors might not want to invest in our product if it is unaesthetic, inconvenient, or looks too complex or cumbersome. When designing our product, we should take care to ensure that the technological components are unobstructive. We should also make sure that we do not visually clutter the bookshelf with parts or wires that would make our product look sloppy or unappealing.

#### **4.1.8 Political Constraints**

One prominent political constraint involves adherence to regulatory rules and laws established by government authorities. There may be regulations related to data privacy, surveillance, or intellectual property rights that must be carefully navigated. For instance, the implementation of a system that captures and processes images could be subject to privacy laws and require specific measures to ensure the ethical and legal use of collected data. Complying with these regulations is essential to avoid legal conflicts and maintain the trust of consumers.

Some governments have specific regulations that dictate where certain types of data, including sensitive or personal information, should be stored. This constraint can influence decisions regarding the design of the system and the selection of database and other local storage tools.

In the development of innovative technologies, intellectual property (IP) considerations become crucial. Political constraints may arise in the form of existing patent laws, copyright regulations, or trade secrets protection. It is important to navigate these legal aspects with attentiveness so that the project does not infringe on existing intellectual property rights. Additionally, political decisions related to IP laws and international agreements can impact the project's ability to collaborate with certain technologies and research.

It is important to understand and pay attention to these political constraints for the legal and successful implementation of ROBOT. By doing so, our project will be carried out under legal, ethical, and responsible development.

#### **4.1.9 Time Constraints**

It cannot be overstated that time management is a crucial component of the senior design process. Time is the ultimate judge of development and completeness. The senior design project journey is characterized by a sequence of deadlines, each with a distinct meaning. But none is more important than the upcoming event of Senior Design II's final presentation.

As students undertaking this senior design project, we face numerous limitations that are imposed by the harsh limitations of time. These limitations include a broad spectrum of factors that have a significant impact on the project's outcome in the end, from conception to execution, and all of the tedious planning in between. As soon as the project was assigned, the countdown began, establishing the tone for the future. From the first brainstorming meetings until the final product launch, these time limitations influence the project's path. Time is an unavoidable factor for this project, including testing, research and development, prototyping, and troubleshooting. These temporal limits are also extremely significant; they are not placed at random. The final presentation, which we have to do in front of our professors and colleagues, marks the end of our Senior Design journey. This presentation is the ultimate test of our creativity and knowledge.

Time constraints can be both a challenge and a source of motivation. Working within a defined timeframe sets an example of real-world engineering scenarios and teaches the importance of meeting deadlines. By having set deadlines, we learn essential project management skills and how to make educated decisions. These skill sets are extremely important in Senior Design as well as in the real world because time management can either make or break how successful a product is. It's important to recognize that one of the biggest time restrictions we

have is related to the different stages of PCB creation. This includes obtaining the necessary parts and creating the PCB design, as well as the complex procedures of actual assembly and testing.

Getting the necessary parts by itself can take a considerable amount of time. In the complicated world of electronics today, where parts differ in size, availability, and complexity, finding the proper parts requires time and careful attention to detail. We have to interact with suppliers, manage wait times, and frequently consider other options in the event of unanticipated delays. This phase creates the groundwork for the entire PCB design process, thus any setbacks or delays here could affect the schedule as a whole.

Designing the PCB is also very important in terms of timing. A thorough grasp of the project's specifications, component placement, routing, and consideration of power distribution are all necessary while designing a PCB. Software tools, prototyping, and interactive development are all part of this lengthy process. Our project's timeframe may be significantly impacted by the design phase alone due to the complexity and precision involved.

After the design is complete, we proceed to the physically taxing step of PCB construction. Drilling, soldering, and assembly are all part of the actual board manufacturing process. For the PCB to perform as planned, the physical construction process can be labor-intensive and has to be very accurate.

The total duration of these phases—from purchasing parts to conducting final testing—creates a sizable time constraint that our senior design team needs to carefully monitor. Any delay can compromise our ability to reach project milestones and, ultimately, the project's success. It emphasizes the necessity of careful planning, backup plans, and an organized project management strategy to deal with the punctuality, precision, and difficulty that comes with PCB development.

Our project has an additional difficult problem that adds a great deal to the overall complexity and time restrictions of the project, on top of the complexities of the PCB process. This additional element includes the coding and debugging stages, which are essential elements that are crucial to ROBOT's functionality. The coding component of our project is essential to bringing ROBOT to life and giving it the capability and intelligence, it needs to complete its assigned tasks. In this stage, code, algorithms, and scripts are created to control how ROBOT responds to its surroundings, gathers information, and makes judgments. Determining standards and actions that ROBOT will follow, the coding procedure is like writing the software's DNA. Integration becomes important and requires careful attention to detail when hardware and software interact, which takes time.

The work doesn't stop at writing code. Eventually, the procedure takes us into the difficult world of debugging. The skill of finding and fixing mistakes, conflicts, and unexpected behaviors in code is known as debugging. It's a laborious process that

calls for attention to detail because seemingly insignificant coding errors can have a significant influence on ROBOT's functionality. In the interactive process of debugging, we test the code, find problems, fix them, and then run it again to make sure it works as intended.

Throughout the entire design process, we face a distinct set of obstacles that complicate our teamwork in addition to technical and project-related time constraints. Our team's location is the source of these difficulties. Although half of our team lives near campus, the other half faces daily commutes of 30 min or more to reach our designated workspaces. Team members who live near campus may find it easier to schedule face-to-face meetings because of their close proximity. Longer commuters, however, will find that this method requires more planning and dedication. We use techniques like scheduling meetings in advance or using online communication platforms to make the most of our few in-person encounters. The problem of managing time extends to face-to-face meeting time as well. These encounters happen less often because of the commutes, thus the time allotted for them needs to be used wisely. Since we have limited time together, we need to make the most of it by concentrating on important project tasks, decision-making, and teamwork.

Lastly, we still have other college classes to consider. Consequently, time spent on senior design may be limited by arranging time for assignments, studying for other classes, and maintaining a personal life. Our team is aware of the importance of this time limitation as we continue and are working to use it as motivation to complete our project.

#### **4.1.10 Testing & Presentation**

Concerning senior design, testing is an important task. It prepares us for a presentation that performs well. The CV system must be programmed to develop ROBOT. This involves training the CV to effectively capture pictures and send them to the computer, which will then run several algorithms and analyses. The ultimate aim is to be able to identify books by their unique visual features as well as by their titles and authors. The key to accomplishing this result is to successfully encode the PCB. The PCB functions as the technological conductor that coordinates the CV system's and the computer's smooth interaction and cooperation. The PCB will be able to support an elaborate and dynamic flow of data and instructions through coding and programming, allowing the CV system to complete its book recognition responsibility accurately and effectively. Essentially, hardware and software are working together seamlessly, and the PCB code serves as the link between them. The core of ROBOT's capabilities is embodied in this development process, where creativity and technology come together to turn book recognition from a mere idea into a reality.

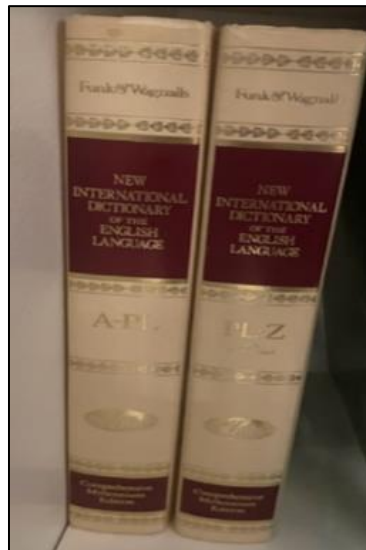
One thing to keep in mind during the ROBOT testing phase is figuring out the best angle to place the camera. This choice affects the system's capacity to recognize books from photos clearly and accurately. It's also important to keep in mind that there's a slim chance that, in the course of testing, it will become clear that the first bookcase selected isn't appropriate for the system requirements. In that case, the project might involve a different bookcase that better suits ROBOT's operational requirements. To guarantee the final success and effectiveness of the book recognition system, flexibility and adaptability throughout the testing phase are crucial.

Another problem that could occur concerns the application of the LED tracking system. When creating the underlying code, it is important to ensure that books are fully aligned with their corresponding locations on the LED grid. Several tests must be run to fully evaluate the system's functionality. An example of an organized evaluation would be choosing a book from the shelf and keeping a close eye to verify that its location is correctly noted in the mobile app that goes with it. The book would then be taken out and put back on the shelf after a period of thirty seconds to make sure the system had properly recorded the return of the book. Ten iterations would be required to develop consistency and dependability in this approach. This set of tests would then be run again at different intervals to evaluate the endurance of the system. The time intervals should include one-minute, five-minute, thirty-minute, one-hour, and a final assessment following a full day. This extensive testing schedule is necessary to see whether the system can operate without issues for longer periods than just one day.

These evaluations would need to be carried out on several books with various sizes, shapes, and characteristics to take into consideration the wide variety of volumes found in the library. This would guarantee that a large variety of library materials, each with its distinct qualities, are recognized and tracked by the system with ease. Additionally, it is critical to evaluate the correctness of the system when handling both the removal and replacement of several books on various shelves. This scenario resembles a more realistic library setting where users could interact with multiple books at once. It must be able to distinguish between different book movements and precisely capture them in real-time. Testing would also need to include figuring out whether the system can recognize when a book is positioned incorrectly in relation to an LED. Since it guarantees that volumes are returned to their proper places, this is essential for preserving the structure and quality of the library's collection. To sum up, extensive testing that takes into account a wide range of time intervals and library items is necessary to ensure that the LED tracking system is accurate and reliable, which makes it a vital tool for efficient library management.

One other testing category that requires careful consideration consists of analyzing books that are extremely similar to ones that have already been tested. This portion

of the review process is for publications that are identical in terms of design, with the primary feature that sets them apart is the order of their alphabetical content. Consider the following example: There are two books, "Funk & Wagnalls New International Dictionary of the English Language," that are identical to one another in terms of look. The content organization of these two publications, however, is what really sets them apart. Entries from A to PL are included in Book 1, while entries from PL to Z are included in Book 2. We may look to Figure 5, which illustrates the resemblance in their physical characteristics, to help paint a clearer image. Here, the difficulty is making sure that the LED tracking and CV system can distinguish between these volumes that appear to be the same just by looking at the alphabetical data in the books' sequence.



**Figure 5: Sample Project Image**

This particular kind of testing becomes necessary in library settings where books with similar covers are common. Maintaining the integrity of the library's collection and enabling effective retrieval depends on the system's ability to appropriately identify and differentiate between such volumes. Additionally, it minimizes the possibility of user misunderstanding when looking for certain materials, improving both the general user experience and the effectiveness of library management. This type of testing therefore emphasizes even more how careful inspection and optimization of the LED tracking system are required to ensure its reliability and effectiveness when managing different books including those that share many similarities but differ greatly in terms of content arrangement.

Furthermore, another component of our thorough testing procedure to consider is a careful analysis of the LED tracking system when it is operating in Night Mode in case we reach this stretch goal. There are unique constraints and obstacles specific to this testing phase. For the system to function properly at night, a carefully thought-out code needs to be put into place. The first step in our Night

Mode testing process is choosing a book from the library's collection. Here, determining if the alarm system operates as planned is the main goal. This is a continuous operation that is carried out ten times for every book on the library's shelves. Moreover, the selection of books is done at random to further reduce the chance that the system may pick up on a predictable trend, which complicates our testing process. Overall, our testing procedure continues into the Night Mode and includes factors like logistical concerns, detailed information logging, and random book selections. Together, these actions strengthen our resolve to thoroughly assess and improve the LED tracking system in order to guarantee its best performance in a variety of practical circumstances.

There could be a lot of possible difficulties and logistical obstacles on the day of the presentation. A primary concern relates to the effective transportation of our project to campus. This may be difficult because of having to transport a bookshelf that has LED systems and fragile cameras, in addition to the numerous books that are necessary for our presentation. Luckily several of the group members have large cars that are able to fit the entire bookcase and additional bags for transporting our project for live demonstrations.

## **4.2 Standards**

Within the field of engineering, standardization is necessary to direct a project's progress from concept to completion. Engineering projects are evaluated for their quality, safety, and performance using standards. Following standards is more than a simple requirement it is an essential component that ensures the delivery of products and services that either meet or exceed established requirements, thus promoting longevity, innovation, and reliability.

### **4.2.1 Kotlin and Android Standards**

Standards for Android and Kotlin are designed to protect the interests of the user. Additionally, they ensure that uniformity is applied, creating a standard that all developers can abide by. By abiding by these standards, we can reinforce the functionality of our design.

#### **4.2.1.1 Kotlin Standards**

Kotlin provides null safety as a fundamental principle. It addresses well-known null pointer exceptions by separating nullable types from non-nullable types. Developers are encouraged to explicitly declare when a variable or property can be null using the nullable type modifier [67]. The compiler checks to ensure that nullable variables are handled properly, reducing the likelihood of null-related runtime errors [67].

Kotlin encourages the use of immutable data structures through the **val** keyword to declare read-only properties. Immutability helps to prevent unintended variable

changes. By default, developers are advised to use **val** for declaring variables whenever possible, only resorting to **var** when mutability is clearly required [68]. This practice aligns with functional programming principles and preserves a safe coding style.

Kotlin supports extension functions. Extension functions enable developers to add new functions to existing classes without modifying source code. This feature fosters code reusability and allows for the creation of Domain-Specific Languages (DSLs) [69]. Developers are encouraged to leverage extension functions and DSLs to enhance code readability.

Kotlin's type system allows for smart casts that reduce the need for explicit type checks and casting in code [69]. When the compiler detects that a certain type check has already been performed, it automatically casts the variable to the expected type. Developers are encouraged to leverage smart casts to write cleaner code. It is important to remember that smart casting only works “when the compiler can guarantee that the variable [will not] change between the check and its usage” [70].

Kotlin introduces coroutines, a powerful feature for asynchronous programming. Instead of relying on callbacks or complex threading models, developers can use coroutines to write asynchronous code in a more sequential manner. The standard library “`kotlinx.coroutines`” provides utilities for asynchronous programming, including `async` and `await` constructs, which enable developers to write efficient and scalable asynchronous code [71].

In Kotlin projects, adopting thoughtful package and class naming conventions is fundamental to creating a well-organized and readable codebase. Packages serve as a means of structuring code hierarchically, providing a namespace for classes and ensuring a logical separation of concerns. Following the reverse domain format for package names, such as **com.example.myapplication**, contributes to clarity and avoids naming conflicts. Package names should be lowercase and have no underscores [71]. When naming classes, adhere to the CamelCase convention, starting with a capital letter and capitalizing the first letter of each subsequent word [71]. Employ descriptive and meaningful names for classes that convey their purpose, promoting code readability. Additionally, consider incorporating type information into class names to enhance understanding. For instance, prefix classes representing interfaces with “`I`” or suffix classes denoting data entities with “`Data`” for clearer identification. By consistently applying these package and class naming standards, developers create a structured and comprehensible codebase that facilitates collaboration and maintenance throughout the software development lifecycle.

In Kotlin, consistent and well-formatted code is not just a matter of style but a crucial element for readability and maintainability. The Kotlin coding conventions

emphasize a standard indentation style of four spaces per level, promoting a clean and visually coherent structure [71]. This indentation choice enhances code readability by clearly delineating code blocks, such as loops, conditionals, and function bodies. Additionally, it ensures that nested structures are easily discernible, facilitating the understanding of code flow.

Spaces should be present around binary operators apart from the range to operator [71]. There should be no spaces around unary operators, nor “before an opening parenthesis in a primary constructor declaration, method declaration or method call” [71]. Spaces should be present between control flow words such as if and while [71]. No spaces allowed after opening parenthesis or bracket and neither after their closing counterparts. All comments are marked by ‘//’ and require a space after to begin comment. It is also important to remember to never put a space after the ‘?’ which marks a nullable type [71].

Adhering to these common Kotlin standards contributes to code consistency, readability, and maintainability. By following these practices, developers can harness the full potential of Kotlin's expressive syntax and modern features to create robust and efficient applications.

#### **4.2.1.2 Android Standards**

When creating an Android app, there are certain principles that should be upheld to create a functional and stable application. Of these principles, the architecture is the backbone of software and can make or break its responsiveness. The first of these principles being Separation of Concerns. Meaning a computer program or application should be split into sections rather than all in one section. Following this can help when debugging as it is easier to troubleshoot a problem in a specific section than one large activity (or main). Another important principle is Single Source of Truth or (SSOT). Where a new type of data is defined, an owner should be defined too. The owner of the data is the only one that can adjust it leading to all the data to be centralized and identifiable for future errors [72].

Unidirectional Data Flow is used alongside SSOT to create a robust application. UDF is where the state of data flows in one direction, where the cases that modify that data flow in an opposite direction. This is done to increase data consistency and remove the chance of data being mixed and misinterpreted [72]. With these above-mentioned principles, there are also recommendations for how to structure an Android Project. Where there are up to three distinct layers of architecture.

##### **4.2.1.2.1 UI Layer**

The function of the UI layer is to render data (UI elements) created through toolkits such as Jetpack Compose. This data is altered when there is an action or user interaction, and when altered should reflect these changes [72].

##### **4.2.1.2.2 Data Layer**

This layer of the app architecture is extremely important, especially for the project. This is because the camera on the ROBOT shelf will constantly be sending data to the app through a database. Then the application will have to store and react based on that information showing that the data layer and UI layer work together. This makes it even more important for the UI layer and data layer to be separated, as it opens the door to use different data storage solutions.

#### **4.2.1.2.3 Domain Layer**

While this is an optional layer, it may be mandatory for some applications. The domain layer is between the data and UI layers that perform business functions. One example of a business function is the validation of data that is collected, checking if the data is in a specific prescribed range and format (i.e units of measurement) [72].

### **4.2.2 C Language Standard**

ROBOT relies on multiple programming languages, one of which is the C programming language. Therefore, it is essential that we thoroughly understand and make sure we abide by the C language standards that apply to our project. The international standard that governs the C programming language is known as ISO/IEC 9899 [73].

These standards are a result of the collaboration between the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), as well as multiple member countries and other international organizations.

The four major parts of the C language standards pertain to preliminary elements, the characteristics of environments that translate and execute C programs, the language syntax, constraints, and semantics, and the library facilities. The scope of the C language standards includes the representation of C programs, the syntax and constraints of the C language, the semantic rules for interpreting C programs, the representation of input data to be processed by C programs, the representation of output data produced by C programs, and the restrictions and limits imposed by a conforming implementation of C [73].

By recognizing and abiding by these standards, we can ensure the smooth operation of the C language related portions of our project. Failure to do so might mean we encounter errors ranging from compiling problems to difficulty with reproducing our results and ideas. Therefore, we should closely monitor C programming standards.

### **4.2.3 Python Standards**

The ROBOT project involves multiple programming languages spanning multiple devices to achieve its goals. Among these languages, Python plays a vital role in enabling our combination of computer vision, database management, and user interaction. It is essential that the project team thoroughly understands and adheres to the Python programming standards that are relevant to our project.

Python is known for its readability, simplicity, and user friendliness. Adhering to established standards will further simplify the ROBOT coding process. The relevant standard for Python is PEP 8, which stands for Python Enhancement Proposal 8. PEP 8 is a comprehensive style guide for writing clean, readily readable Python code [74].

PEP 8 provides a set of guidelines that enhance code collaboration. With multiple team members contributing to the ROBOT project, adhering to these standards ensures that code is written in a consistent style. This consistency facilitates collaboration, makes code easier to read, and ensures that the project remains maintainable in the long run.

Clear and readable code is very important for the ROBOT project because of its complexity and the important role played by Python. PEP 8 emphasizes the importance of readable code, which makes debugging and maintenance easier. PEP 8 also includes best practices and conventions for code organization, naming, and documentation [74]. By following PEP 8, the project team ensures that any issues that arise can be resolved quickly.

Because Python is known for having an extremely large user base with many community resources, following PEP 8 will ensure that we can maximize our access to community resources as well. Following established guidelines will make it much easier for both collaboration within our team and with the larger developer community.

In summary, PEP 8 plays a critical role in the development of the ROBOT project by ensuring code consistency, readability, and maintainability. Adhering to these standards is essential to avoid errors, promote collaboration among team members, and promote compatibility with the broader Python ecosystem. By closely monitoring and following Python programming standards, we can make our project progress much more smoothly.

#### **4.2.4 PCB Standards**

PCBs are the invisible heroes of the modern world; they enable our linked lifestyles and secretly power our electronic devices. In the development of the ROBOT project, the design and implementation of a customized Printed Circuit Board (PCB) is a vital part of our project, comprising the bulk of the work. The PCB can be called the “soul” of the project. The complex boards conceal a vast network of standards that guarantee the dependability, security, and connectivity of

electronic devices. Given how important this part of the project is, we must make sure we adhere to the PCB standards that apply to us. PCB standards are created by the Association Connecting Electronics Industries, formerly known as the Institute for Printed Circuits. IPC has established, comprehensive standards to make sure PCBs are uniform and reliable. Following these standards will help to make sure that the PCB portion of our project is free of avoidable error.

IPC standards in fact have a large group of specifications. These are known as IPC-2220-FAM, which encompasses a set of standards for board design. Namely, these refer to IPC-2221, IPC-2222, and IPC-2223 [75]. These standards outline specific requirements for designing rigid organic printed boards, flexible printed boards, and organic multichip modules, taking into account thermal, electrical, and mechanical considerations. Because we intend to use a rigid board, these are the standards that apply closely to us.

#### **4.2.4.1 IPC-2222**

IPC-2221 is a standard mainly focused on rigid PCB whether they are single, double, or multisided [76]. Some of the key concepts outlined in this standard are: Dielectric spacing, Interconnects, Lead-free laminate, and Non-functional lands. Dielectric spacing is based on how far the traces are from each other to minimize interference and unintentional bridging. The interconnect portion is based on VIA points that connect different layers together. In the past, lead was an industry norm in solder and epoxy laminate, this concept is based on the phasing out of lead out of products due to its harmful effects on human health and the environment. Finally non-functional lands are based on the through-holes on the PCB that are not electrically connected, mostly used for mounting.

#### **4.2.4.2 IPC-2223**

While the previous standard focused on rigid PCBs IPC-2223 is based on the design of flexible PCBs with key concepts such as: Material Selection and Impedance & Capacitance Control [77]. The material selection section is based on choosing a substrate/cover layer, conductor, and adhesives to fabricate the flexible PCB based on the given application such as polyimide or PTFE. Impedance and capacitance control focuses on regulating the possible increase in resistance and signal reflections due to the substrate material and other factors that may lead to malfunctions of components [77].

### **4.2.5 Computer Vision Standards**

For ROBOT, we plan to implement computer vision to allow us to process the books placed in the shelf. There are multiple standards that apply to our plan for implementing computer vision. Abiding by these standards will help us ensure our product is functional, operational, and ethical.

The IEEE Computer Society is one organization that is highly relevant to computer vision standards. One of its notable contributions is IEEE 1857, which focuses on image and video processing standards. These standards will help us with matters related to image compression, processing, and encoding. By making sure we follow IEEE 1857, we can ensure that our computer vision implementation operates efficiently. This will also help us ensure our implementation is compatible across many platforms [78].

Another important group of standards come from ISO and IEC. ISO and IEC collaborate on various standards that impact computer vision, particularly in the realm of image and video compression. Standards such as those within the MPEG family contain methods for encoding and decoding multimedia content. Additionally, image file format standards, like JPEG and PNG, ensure consistency in how images are encoded and decoded [78]. This allows for widespread use and exchange of visual data that we will need for our project.

When integrating cameras and sensors into computer vision systems, we must make sure we follow camera interface standards. MIPI CSI-2 and USB3 Vision provide clear guidelines for hardware communication. MIPI CSI-2 offers a standardized approach for mobile and embedded devices, ensuring proper connections between cameras and processors. USB3 Vision, on the other hand, is useful for USB-based cameras, offering guidelines for video streaming and control [78].

## 5 ChatGPT and Similar Platforms

In the emerging world of technology, AI has begun taking hold in society as the future. With Google and Microsoft at the forefront of the and consumer platforms such as ChatGPT are exposing everyday people to the amazing strengths of this technology. However, what implications does AI have for the future of academics, is AI a great tool for development and research, or are they a detriment?

### 5.1 Comparisons

ChatGPT has many advantages, for instance, the platform can generate general comparisons between certain products which is helpful for our project, as there are multiple components of the project being compared. The aspect of this that is the most impressive is of the AI platform can decipher conversations of the products or materials desired and use it to give an educated answer [ChatGPT used]. Furthermore, when the comparison based on the given prompt is answer, it will learn the specifications and become more accurate each time it is asked for a similar comparison. For example, here is ChatGPT's response to the prompt "What are the differences between the MSP430FR6989 and MSP430G2553 microcontrollers?":

Architecture:

MSP430FR6989: This microcontroller is part of the MSP430FRx FRAM series, where FRAM stands for Ferroelectric Random Access Memory. FRAM is a non-volatile memory technology that combines the benefits of both Flash and RAM.  
MSP430G2553: This microcontroller is part of the MSP430Gx series, which uses traditional Flash memory for program storage.

Memory:

MSP430FR6989: It typically comes with more memory, including more FRAM for both program and data storage.

MSP430G2553: It has less program memory (Flash) and RAM compared to the MSP430FR6989.

FRAM vs. Flash:

MSP430FR6989: Uses FRAM technology for both program memory and data storage. FRAM has advantages such as faster write times, lower power consumption, and higher endurance compared to traditional Flash memory.

MSP430G2553: Uses Flash memory for program storage. Flash memory is more common in traditional microcontrollers but may have different characteristics compared to FRAM.

Peripherals:

Both microcontrollers come with a variety of peripherals, including GPIO (General Purpose Input/Output), timers, UART (Universal Asynchronous Receiver-Transmitter), I2C (Inter-Integrated Circuit), and SPI (Serial Peripheral Interface). However, the specific peripherals and their configurations may vary between the two models [79].

As shown above, ChatGPT gives a few different attributes in a bulleted format. However, ChatGPT repeats some of the comparison points about the available RAM and storage multiple times. While this information is useful for research purposes, ChatGPT fails to highlight physical differences between the two microcontrollers such as the number of available pins and well as board layout. This can add even more time to research as it fails to give details that are very important for the testing and development phase of ROBOT.

## **5.2 Software Debugging**

There are many instances when coding and encountering a bug, syntax mistake, or runtime error that it not easily pinpointed. For this, ChatGPT can take the code, parse through each line, and in most cases, diagnose the specific line and give a description of what went wrong. This not only can speed up the development of a project, but also serve as a learning tool for novice programmers as they are still learning the rules and syntax of their chosen language. Moreover, if code is given in a specific language ChatGPT can also attempt to recreate in a completely different language, however, this generated code will have to be reviewed as not all functions of one language directly translate to another.

## **5.3 Data Analysis**

The next great strength that platforms like ChatGPT have is its data analysis skills. When given a set of data and set parameters, the chatbot will take the data and run calculations or provide code that when pasted in an IDE will produce the desired output. This is incredibly useful as it also explains how the code works when prompted. Another feature of the AI is that it can take a set of messy data with given classifications or categories and organize it with the prescribed prompt. In addition to this, ChatGPT can give insight into how to clean data that has yet to be processed, such as removing duplicates, outliers, and accounting for missing data points. However, if the data is not formatted correctly before providing it to ChatGPT it may not be able to parse through the data, and in some cases, can even tell the user that it is unable to decipher the data [79].

## **5.4 Limitations**

The biggest limitation of ChatGPT is the reliability of the information required, specifically specifications including large amounts of data. In some cases, ChatGPT can give the wrong information under the guise that it is correct. This is why facts and numbers given by ChatGPT must be fact checked for validity leading to even more time researching that could be spent finding the correct information efficiently. Another flaw of ChatGPT is it has the possibility to be biased on certain topics and blind spots in others. This is due to Open AI pulling from many sources, and if those sources are skewed more in the favor of one stance over another the answer the AI will give is likely to be skewed as well. As for its blind spots, if there is a topic that does not have enough public sources ChatGPT will either answer that there is not enough information to generate, or it will generate a response and have a higher likelihood to be incorrect or outdated. Below is the response ChatGPT gives when asked about CGIC (or Current Generalized Immittance) filters, which are important for sectioning out noise from signals:

“It appears that "CGIC" is not a standard term in filter design, so I'll explain the differences between Butterworth and bandpass filters without specific reference to "CGIC" [79].

The AI platform will only give a generalized summary of these filters and omit one of the more important distinguishing aspects of it. In this situation OpenAI platforms cannot be substituted for using academic sources that will give the desired information with even more context and credibility.

Furthermore, the lack of citing sources when ChatGPT generates a response. While what the AI is writing is based on sources on the internet, it fails to give credit or give links to the sources for citation. Another blind spot of ChatGPT is its inability to create graphics within the program. It can provide code for a preferred language to create the graphic, but a good majority of the time the code must be adjusted to give the desired graph or figure. ChatGPT as a learning tool is also questionable, sometimes when giving a prompt to solve a math problem no steps will be given on how the AI achieved this value. Instead, only the final value will be displayed, meaning it is very difficult to double-check ChatGPT without doing all the calculations yourself. Which defeats the purpose of using the text-based AI platform. Here is an excerpt of ChatGPT's response to the prompt “what are some of the limitations of ChatGPT”:

Overuse of Certain Phrases:

- The model may be prone to repetitive or verbose answers and may overuse certain phrases. It can also exhibit a preference for safe, generic responses.

#### Lack of Memory:

- ChatGPT doesn't have a memory of past requests within a conversation. Each input is treated as a separate query, and it doesn't retain information from prior interactions in the conversation.

#### Vulnerability to Biases:

- The model may inadvertently exhibit biases present in the training data. It can generate responses that reflect societal biases or prejudices.

#### Inability to Perform Actions:

- While ChatGPT can provide information and answer questions, it doesn't have the ability to perform actions in the real world. It's a text-based model and can't execute commands or interact with external systems.

#### Difficulty Handling Ambiguity:

- The model may struggle with ambiguous queries and might guess the user's intention rather than asking for clarification.

#### Potential for Inappropriate Content:

- Despite efforts to filter inappropriate requests during fine-tuning, ChatGPT may sometimes respond to harmful instructions or exhibit biased behavior.

#### Resource Intensive:

- Generating responses can be resource-intensive, especially for long conversations, which may affect response times [79].

## 6 Hardware Design

The hardware design section will touch on the overall implementation and role of each hardware subsystem such as the Jetson Nano, custom printed circuit board, and the MSP430FR6989.

### 6.1 Power, Voltage and Current Ratings

ROBOT contains many physical subsystems that come together to create a functional design. With these subsystems, comes a need to account for required voltages and power ratings for each component. If this aspect is not addressed, it will cause massive setbacks in the implementation and demonstration portion. The subsystems highlighted in this section are: Jetson Nano, Raspberry Pi Camera, PCB Components and Display Technology.

#### 6.1.1 Jetson Nano & Raspberry Pi Camera

The Jetson Nano (Used for the CV Portion) has multiple voltage and current specifications. To begin, the Jetson Nano runs on DC and has a minimum operating voltage of 4.75V and a usual accepted voltage of 5V. However, its maximum stated voltage is 5.25V, while it will function at this voltage, it is not recommended to operate at this voltage for an extended period as it may cause long-term damage. As previously stated, the Jetson Nano has a Maximum acceptable voltage of 5.25V, yet the Nano also has Absolute Maximum ratings that are higher than that but may lead to malfunctions during operation. The absolute max voltage rating for DC input is 5.5V and an absolute minimum of  $-0.5V$ . Furthermore, the Nano also has an optional back up battery that is connected to a specific input (PMIC\_BBAT) which can either be a lithium-ion battery, standard capacitors, or super capacitors [69]. This back up will manage the real-time clock voltage when the input DC voltage is not supplied. The back up battery must be rated for a range of 2.5V to 3.5V and have a constant current between 50 and 800 $\mu A$ . This input also has recommended operating conditions as well as absolute maximum conditions. The recommended conditions being a DC voltage min of 1.65V and a max of 5.5V. While the absolute maximum limits being  $-0.3V$  min and 6V max. However, these ratings only highlight power inputs on the Jetson Nano, and do not consider the voltages required for logic functions which are as important as input voltages. These logic portions being CMOS pins and Open drain pins.

##### 6.1.1.1 CMOS Pins

The CMOS pins are responsible for keeping track of data for system configuration as well as settings for hardware and booting including the pins:  $V_{IL}$ ,  $V_{IH}$ ,  $V_{OL}$ , and  $V_{OH}$ . Representing Input Low Voltage, Input High Voltage, Output Low Voltage and Output High Voltage respectively. Figure 6 shows all the aforementioned pins.

Symbol	Description	Minimum	Maximum	Units
$V_{IL}$	Input Low Voltage	-0.5	$0.25 \times V_{DD}$	V
$V_{IH}$	Input High Voltage	$0.75 \times V_{DD}$	$0.5 + V_{DD}$	V
$V_{OL}$	Output Low Voltage ( $I_{OL} = 1\text{mA}$ )	---	$0.15 \times V_{DD}$	V
$V_{OH}$	Output High Voltage ( $I_{OH} = -1\text{mA}$ )	$0.85 \times V_{DD}$	---	V

**Figure 6: CMOS Pin Characteristics**

For the Low and High Input Voltages, some of the minimum and maximum requirements are dependent on the VDD voltages. For example, let say VDD is 5V, to calculate the maximum low input voltage we multiply 0.25 by VDD: ( $0.25 \times 5V = 1.2V$ ). This is similar to the input high except it is multiplied by a factor of .75 for the minimum and adding 0.5 to VDD for the maximum. Yet, there are some output voltages where there are no minimums and maximums. Where the currents for output low and high voltage are 1mA and -1mA.

### 6.1.1.2 Open Drain Pins

The Jetson Nano also contains several open drain pins that function as a method of pulling down the voltage of the pin (usually to GND) with the use of a transistor. The pulling down of the pin voltage is when the transistor is on, when it is off the pins become open or 'floating'. These pins have similar specifications to the CMOS pins with the main difference being the addition of a I2C pin sharing the same VOL symbol and the lack of a minimum voltage rating.

### 6.1.1.3 Raspberry Pi Camera

To power the Raspberry Pi camera, a supply voltage of 1.65V is required. Conversely, the lowest voltage provided on most single-board computers is around 3.3V. To combat this issue, resistors must be used to lower the voltage before it is used by the camera module. Both resistors must have a value of 10kΩ each which will then be connected to the 3.3V head on the PCB and GND [70]. This supplies the camera with the rated voltage as well as creating an impedance from the source of 5kΩ.

## 6.1.2 PCB Components

This section will include details about two important components to be placed on our custom printed circuit board. The Bluetooth module is important for communication between our devices, while the ultrasonic sensor will help us detect obstacles and save power and time.

### 6.1.2.2 Bluetooth Module (HC-05)

The HC-05 is a SMD Bluetooth module that will be integrated into the ROBOT PCB design. It contains its own integrated power supply. This supply runs at a voltage

of 3.3 V with a linear regulator that runs between 3.15 V to 3.3 V. In addition, the HC-05 has a low power mode that lowers the operating voltage from 3.3 V to 1.8 V. In addition, there is an EN pin that must also be supplied with a voltage of 3.3V after VCC is supplied.

### 6.1.2.3 Ultrasonic Sensor (HC-SR04)

The Ultrasonic Sensor present on the ROBOT PCB design is a pre-made part based on a through hole design. Because of this, through holes that interface with the sensor are required. The sensor runs a DC voltage of 5V with an operation current of 15mA leading to a power draw of 0.075W. Even at the sensor’s higher operating voltage, the module is very efficient. Meaning if the sensor needs to be running for long periods of time (constant polling) it will not noticeably increase the amount of power draw of the unit. The table below contains all the main PCB Components, and separate components as well as their power ratings. Below is a table of the containing all the Main PCB Components as well as power hungry Components such as the addressable LED strip:

**Table 8: Main PCB Components and Power Characteristics**

Component	Recommended Voltage Supply		Recommended Power Supply		Max current draw estimate
	Min Vin (V)	Max Vin (V)	Min Power (W)	Max Power (W)	
MSP430FR6989	1.8	3.6	varies by usage		varies by usage
144 LED strip	5V		varies	35 W	20 A
Bluetooth module	3.6	6	0.060 W		10 mA
Ultrasonic sensor	5V		0.075 W		15 mA

### 6.1.3 Display

ROBOT uses a touch screen display for the user to search for the availability of a specific book as well as its location. The power draw of this screen will also have to be considered as unless it enters a low-power mode it will take a considerable amount of power to drive. Based on the given specifications provided by Amazon the ‘HAMTYSAN’ runs at on 5 volts DC which may vary based on the required workload.

## **6.2 Connections and Setup**

This section will cover the hardware and software module connections for the Jetson Nano, Raspberry Pi cameras, Python scripts, and Python libraries.

### **6.2.1 Jetson Nano and cameras**

The combination of hardware and software is an important aspect of the production of ROBOT. An excellent example of this collaboration is the combination of the Jetson Nano, a powerful single-board computer, with advanced camera technology for image documentation and book scanning. This dynamic integration represents a significant advancement in the field of technology applications as it not only makes it easier to digitize books but also opens up new possibilities for advanced image processing and analysis. The combination of the Jetson Nano with high-definition cameras takes book digitalization to a whole new level of productivity. The Jetson Nano offers an optimal platform for managing complex image processing tasks and implementing machine learning algorithms.

#### **6.2.1.1 Python**

Taking up book scanning and documentation project is made possible by the adaptability of Python programming and the combination of high-resolution cameras and single-board computers. The merging of hardware and software of devices like the Jetson Nano and the Raspberry Pi Camera Module V2 become essential elements in the goal of effective and accurate photo scanning and documentation.

The configuration between the Raspberry Pi Camera Module V2 and the Jetson Nano are explored in this section, with a focus on Python-based picture scanning and documentation. Technical details aside, Python acts as the link of a seamless integration of the union of the Raspberry Pi Camera Module V2 and the Jetson Nano. Because of its versatility and widespread use, Python is a great programming language for navigating the complexity of this connection and achieving the full potential of both software and hardware components. In this dynamic configuration, the Jetson Nano stands out as a key component because of its significant computing capacity, which is designed for demanding image processing demands. This foundation is essential for the successful implementation of documentation and book scanning when it comes to ROBOT.

##### **6.2.1.1.1 Picamera**

The Raspberry Pi Camera Module V2 stands out as an amazing piece of technology. It has an 8-megapixel sensor that works in perfect harmony with Python's adaptability, thanks to the powerful picamera module. This dynamic integration opens the door to real-time photo analysis by allowing users to control the camera's settings while making it easier to produce clear, high-resolution

images and films. The Raspberry Pi Camera Module V2 and Python's picamera package work together in a way that displays a relationship between software and hardware and makes the process of documenting visual data simple and easy. The library plays a crucial role in creating a relationship between the sensor abilities and Python programming skills by allowing users to adjust camera parameters with ease. This complex connection between hardware and software is further complicated by the Python code's dependency on the picamera package. By using it, we are able to precisely set camera settings, which allows us to modify the photography process to the unique requirements of this project. This library acts as a translator, turning user directions into actual actions and creating the foundation for a hardware and software system to work together successfully.

#### **6.2.1.1.2 Python and OpenCV**

Python's ability to be a diverse programming language is shown by the integration of OpenCV into the workflow. This allows Python to reach further into the field of advanced image processing by utilizing external packages that enhance its abilities. Python packages like OpenCV, which is used for this project, and Tesseract can be used to develop techniques like image enhancement, edge detection, and OCR (optical character recognition). Python's abilities can be seen when it is used for tasks beyond simple image processing, such as text extraction from scanned documents. This ability makes it possible to build intelligent systems that can convert visual data into organized, searchable, and useful data in addition to capturing it. Python's versatility blends in nicely with the powerful GPU capabilities of the Jetson Nano, which can extract text for documentation needs as well as capture clear pictures of book spines.

#### **6.2.1.1.3 Libraries**

Python's versatility keeps showing itself as we explore book scanning and documentation; this is another example of how well it can integrate scanned photographs with powerful documentation tools. The integration of Python's capabilities with frameworks and libraries like Flask, SQLAlchemy, and Django represents development in the effectiveness of digital content organization. Combining Python with these frameworks changes the digital environment by providing functions beyond image processing. These tools improve classification by guaranteeing that digital books are carefully arranged according to genres, categories, and other relevant criteria, rather than just being stored. Python holds the hardware and software components together between the camera module and the Jetson Nano. This combination produces an arrangement that provides a solid basis for the creation of efficient, flexible, and user-friendly book scanning and documentation systems. Python can be used for more than only image processing; it can also be used for software integration, allowing for the creation of solutions that address the many demands of storage and digitalization.

Because of its many libraries, community support, and ease of use, Python is a great option for creating applications that go beyond simple documentation, machine learning, image analysis, and feature extraction. This combination of the processing power of the Jetson Nano and Python's capabilities turns the standard into an imaginative environment full of opportunities. Python encourages creativity and innovation, pushing the bounds of what is possible when it comes to digitizing textual information. In addition to enabling communication between the Raspberry Pi Camera Module V2 and the Jetson Nano, the wide variety of Python libraries allows us to explore different approaches in image processing, resulting in a compatible hardware-software system.

### **6.2.1.2 CSI-to-CSI connection**

In order to achieve this smooth integration, a hardware configuration is needed to enable communication between the Raspberry Pi Camera Module V2 and the Jetson Nano. A part of this physical connection is the CSI-to-CSI wire, a channel that connects the camera module and the Jetson Nano. The significance of this cable lies in its ability to enable high-speed data connectivity, which guarantees the quick and effective flow of visual information, in addition to creating a physical link. As the connector in this dynamic, the CSI-to-CSI connection makes sure that there is as little resistance to the communication between our camera module and the Jetson Nano. It moves the system as a whole into an environment where fast data transfer becomes essential, ensuring that the huge amount of visual data recorded by the camera module can be easily handled by the Jetson Nano's processing capability.

The combination of the camera module, the Jetson Nano, and the Python programming language—as well as the picamera library and the CSI-to-CSI cable—makes it clear that this technology integration goes beyond simple picture-taking. Every part is essential to the planning of a complex dynamic between software and hardware, guaranteeing not only efficiency but also a smooth connection that maximizes the potential of this dynamic duo.

### **6.2.1.3 Overall**

These applications guide the GPU-driven performance of the Jetson Nano to quickly and precisely gather, process, and analyze visual input. Python's ability to efficiently handle the scanning process becomes an important aspect, guaranteeing that the entire process is not only quick but also customized to meet the particular requirements of documentation responsibilities.

The entire process represents the uses of Python in modern technology, from taking excellent photos with the Raspberry Pi Camera Module V2 and processing them quickly with the Jetson Nano to integrating them into detailed documentation systems using Python platforms.

## **6.3 Tablet with Raspberry Pi**

Combining a touchscreen tablet with the Raspberry Pi 4 board is an extremely great partnership that not only makes use of the Raspberry Pi 4's powerful computing capabilities but also changes how we interact with digital content. The Raspberry Pi 4, a multipurpose single-board computer known for its advanced processing capabilities, a number of USB ports, and many connectivity choices, is at the center of this collaboration. The Raspberry Pi 4 is a particularly flexible platform, which makes it a great fit for this project. The Raspberry Pi 4's adaptability is made better by its touchscreen display abilities, which adds an extra level of user interaction. The Raspberry Pi 4's capabilities are enhanced by this combination of hardware, providing a more user-friendly and engaging interface. The Raspberry Pi 4's touchscreen functionality increases its application reach and its versatility, making it an effective tool for a variety of projects and uses where interactive displays are needed. As a result, this integration offers users a modern interface for experiencing the wide range of possibilities of the Raspberry Pi 4.

The Raspberry Pi 4, which is widely recognized for its durable processing capabilities and customizable design, is at the center of this dynamic. Its usefulness increases through integration with a touchscreen tablet, allowing for the smooth union of hardware and interactive display technologies. This combination creates a more natural interaction between the user and the computer by allowing people to engage with digital content through touch. This integration adds a hands-on element that greatly increases accessibility and user engagement. The touchscreen tablet's interactive display allows direct, hands-on interaction, making the experience more engaging. By removing barriers and allowing users with different levels of technological ability, this component makes computing more accessible to a wider audience.

### **6.3.1 USB and HDMI**

The responsive software makes the user experience more fluid by making sure that every interaction is met with immediate and smooth feedback. This leads to increased customer satisfaction overall. Using a touchscreen tablet with Raspberry Pi 4 requires an elaborate interface made possible by several connectors, particularly USB and HDMI, which are necessary for a smooth computing experience. This network of connections creates a strong channel of communication, enabling the Raspberry Pi 4 and the touchscreen tablet to work together. The HDMI connection is an important link that allows the Raspberry Pi 4 to send high-definition images to the touchscreen tablet. The HDMI link creates the foundation for an engaging display that improves the user experience in its entirety.

### **6.3.1.1 USB**

Apart from the display, the USB connections are the main components to this integration. Touch inputs from the tablet are sent to the Raspberry Pi 4 through a communication link using USB, therefore turning the tablet into an active and responsive input device. With the use of simple touch motions on the tablet, users may navigate, control, and engage with digital files on the Raspberry Pi 4 thanks to this dynamic interaction. This combination is more versatile than only touch-based communication. The Raspberry Pi 4 allows users to create programs that take advantage of the touchscreen's capabilities for a variety of tasks. The touchscreen tablet can be used as a centralized control center in this project, providing a user-friendly interface for finding information.

### **6.3.2 Python**

Python is the programming language we are using when it comes to the integrated touchscreen setup, especially because of its strong compatibility with the Raspberry Pi. Python is a recommended language for creating programs that take advantage of the touchscreen tablet's interactive capacity because of its compatibility and adaptability with the Raspberry Pi 4. Python's strength is mostly due to the wide variety of libraries that are available. By allowing us to take advantage of pre-existing functionalities, these libraries offer an efficient development process and can potentially save time and effort.

The data from the Jetson Nano, which will contain the status of a particular book, will be successfully transferred and displayed on the Raspberry Pi 4 in the structure of this integration. The use of Python ensures seamless data flow and functionality, allowing the Raspberry Pi 4 to work as a central processing and display center for information. Python, the touchscreen tablet, and the Raspberry Pi 4 work together to create applications that are responsive and easy to use, which not only speeds up the development process but also improves user experience overall.

Because of its flexibility, Python is a great choice for developing interactive interfaces that combine the tablet's touchscreen functionality with visual data from the Jetson Nano and Raspberry Pi Camera Module V2. The Jetson Nano's processing power can be used by real-time object recognition applications to analyze visual data taken by the Camera Module V2 and provide users with up-to-date information. Python also makes it easier to create interactive displays that display visual data and react to touch inputs with ease, providing users with an easy-to-use interface.

Python makes it possible to display the data that the Jetson Nano and Camera Module V2 have collected in a variety of ways on the touchscreen tablet. For example, it is possible to display the current state of books in real-time, giving readers access to the most recent data. This improves the system's functionality

and demonstrates Python's flexibility in providing users with up-to-date, accurate information.

## **6.4 PCB Modules**

Using a PCB is one of the main hardware for the successful completion of this project. When integrating many components, including power buttons, LEDs, Bluetooth modules, and microcontrollers like the MSP430FR69, the complexity of connectivity and configurations increases substantially. Using a PCB that has been carefully constructed not only improves the interactions between various components but also ensures the system's overall performance and reliability.

Starting with the integration of the MSP430FR69 microprocessor, LEDs, Bluetooth modules, ESD safety circuits, and a power button, the first steps entail careful PCB design and layout. The function that each component's placement plays in optimizing signal paths, reducing potential disturbances, and guaranteeing effective power distribution throughout the system is its strategic positioning. Achieving an effective layout is for both establishing a small yet dependable PCB design and improving overall performance. The precise placement of components is essential in the first phases of the design process, as the integration's success mostly depends on their carefully planned placement.

### **6.4.1 LEDs**

The incorporation of LEDs into the PCB is a design component in ROBOT. with the goal of providing additional visual indicator features. This integration introduces a range of factors, three of which stand out as connection concerns: the location of the LED arrays, the use of resistors that limit current flow, and the installation of a stable voltage source. To ensure connectivity and fully utilize the visual aspects, the coordination of these elements requires attention to detail. The best location for the LED arrays on the PCB is a process that takes into account factors other than appearance. Basic factors like heat transmission and light dispersion is necessary since they have an immediate effect on how well and how long the LED components work. To create an environment where LEDs operate properly and provide not just visual appeal but also constant and long-lasting lighting performance, the PCB architecture must intertwine all of these factors together.

Beyond the obvious aesthetic concerns, the PCB design is the long-term efficiency and durability of the integrated LED components. This is where effective power management becomes necessary. It involves not just providing a steady and dependable power source but also putting control and protection methods in place to prevent and minimize voltage swings. This approach to power management is the way to protect the LED components from potential damage, contributing greatly to the dependability of the entire visual signal system over the project's lifecycle.

## **6.4.2 Bluetooth**

Creating wireless communication links is a major step in PCB integration. This requires a careful approach in order to connect the Bluetooth module with the microcontroller (in this case, the MSP430FR69) in a smooth manner. The combination of these elements is a component that requires careful consideration, especially in the layout phase of the PCB design. Signal quality and data transfer speed become factors when it comes to connectivity. Component placement and trace routing are carefully managed to ensure that signals travel throughout the PCB with the least amount of damage and that data transfers at the fastest possible rate.

When considering Bluetooth communication, it is needed to preserve maximum isolation and minimize interference. The PCB was to be designed to reduce electromagnetic interference and noise as the design develops in order to protect the quality of the wireless communication lines. A stable and continuous wireless connection is provided by the routing of traces, the placement of components, and the use of ground planes to create an environment in which Bluetooth signals can travel with minimum disruption.

## **6.4.3 Electrostatic Discharge**

A device's lifespan and reliability can be threatened by electrostatic discharge (ESD), which can damage electronic components. Considering this, it becomes clear that adding ESD protection circuits to the PCB is needed to provide security, especially for delicate parts like the MSP430FR69. These safety precautions improve the overall durability of the electronic system and play a role in reducing potential injury from electrical charges. The placement of ESD protection circuits serves as firewalls against harmful ESD events and can consist of diodes or transient voltage suppressors. The connections involving these ESD protection devices were placed on the PCB carefully in order to strengthen the protection.

Component placement is difficult and involves more than just space constraints. To prevent and diffuse electrostatic discharges before they reach sensitive components requires a careful arrangement of protective devices and an in-depth knowledge of the possible routes of ESD incidents. Thus, strategically placed ESD protection circuits act as a barrier against electrostatic hazards, preventing damage and maintaining the functionality of the system. Adding ESD protection circuits to a PCB is a proactive approach to strengthen electronic components against the risks of electrostatic discharge, rather than just a precaution. The design provides a strong protection system by incorporating these protective parts, which maintains the functioning of delicate components throughout the device's use.

#### **6.4.4 MSP430FR6989**

The low power consumption and adaptability of the MSP430FR6989 microcontroller make it a great choice for systems integration, although doing so requires proper planning. Unlocking the full functions of this microcontroller needs proper setup and connection procedure. Factors include signal line, power supply, and communication interface configurations, all of which are there to maximize the MSP430FR69's overall performance in the system. A unique approach is needed to make sure the microcontroller operates consistently and the way it was envisioned to work. Implementing appropriate grounding techniques and reducing the possibility of noise and interference are all necessary when it comes to the completion of ROBOT.

The MSP430FR6989 works nicely when power, ground, and signal lines are properly managed thus creating a stable electrical environment. Furthermore, having programming links available is a component of the system as it provides access to firmware updates and troubleshooting methods. This includes planning the PCB layout to provide for programming interfaces, which enables firmware updates and effective troubleshooting of possible problems that might occur throughout the microcontroller's use. Essentially, the MSP430FR69 application requires more than just physical connection; it also calls for a setup that takes into consideration its unique characteristics. The microcontroller performs at its best when signal lines, power sources, and communication interfaces are optimized. Other factors that help include efficient grounding techniques, programming connection regulation, and more. This broad method not only optimizes the MSP430FR69's performance but also creates a structure for an electrical system that is reliable.

#### **6.4.5 Power Button**

The power button design contains a series of connections that are there to control the device's power state. The presence of these connections is to enable power-on and power-off operations. To provide smooth transitions between the power levels, the power supply network's structure requires much thought. An approach to technical issues is required while arranging a PCB that includes components including LEDs, Bluetooth modules, ESD safety circuits, the MSP430FR69 microcontroller, and a power button. It requires an organized setup of complex technological demands. A detailed approach to PCB design and a deep understanding of the properties and needs of each component is needed for the integration of these components. Planning serves as the basis, for guiding the PCB layout and specifying how each component will work with the others in the system. Carefully arranging components is considered to maximize the integrated system's reliability and performance. A focus in this connection is signal quality. To reduce signal damage and provide the best possible data transfer, the paths that signals

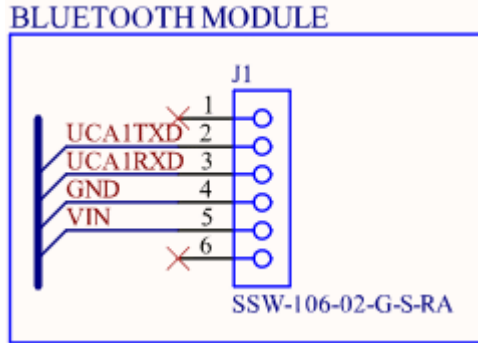
take on the PCB need to be carefully selected. A functional PCB assembly is the result of the combined impact of these factors. Essentially, an extensive plan that goes beyond the physical layout is required for the design and integration of a power button and other components. An electrical system that functions smoothly over a range of power levels is the result of careful planning, deliberate component placement, and a strong focus on signal quality.

### **6.4.1 PCB Individual Module Schematics**

This section will outline the various subsections found on the first version of our printed circuit board. The board was connected and laid out using the EAGLE program by Autodesk. We retrieved several footprints and symbols using the Ultra Librarian tool. An order has been placed for this version with the intention of being mostly for testing out the physical structure and angle of our initial design. Upon arrival, we will quickly finalize the design and place an order for the final PCB with slight modifications including adding the power for the LEDs and for the microcontroller on the same PCB. As of now, they will only share a ground and a data in pin for sending commands to the LEDs. Due to the high demand of the LEDs at their maximum capacity, they require 20A current. However, our group only needs 5 LEDs to light up at any given time so it is highly likely that both the LEDs and the board can be powered through the same signals. Additionally, since our design will include using an ultrasonic sensor, we will decide if it is a good idea to add this part to the PCB or if we should use a separate board for it.

#### **6.4.1.1 Bluetooth Module**

The Bluetooth module will be implemented using the HC-05 sensor. It will be wired up to the MSP430FR6989 and other traces on the printed circuit board as shown in Figure 7 and Table 9. The sensor and MSP430 must have a receiving and transmitting line connected to one another in opposite form. The standard baud rate they will both be set at will be 9600 baud rate which should be sufficient for the requirements of our goal of sending simple commands and acknowledgements. A problem to consider once we receive the initial prototype is to ensure that the Bluetooth module is not being interfered with by other signals on the board. If so, it is important to consider changing the board layout and possibly spacing out the components more.



**Figure 7: Bluetooth Schematic**

**Table 9: Bluetooth Sensor Connections to PCB**

HC-05 Bluetooth Sensor		connects to	Printed Circuit Board
Pin number	Pin functionality		Signal / Trace
2	VCC	→	VIN
3	GND	→	GND
4	TX	→	UCA0RXD
5	RX	→	UCA0TXD

The sensor is connected to the VIN trace that delivers 5V of power since the operating voltage of it is between 3.3 to 6V. The sensor receiving line connects to the transmitting line of the MSP430, since when the MSP430 wants to ‘talk’, the sensor will receive (‘listen’). Likewise, the transmitting line of the sensor connects to the receiving line of the MSP430. These two lines support the serial communication between the two devices. Both devices must share the same ground.

#### **6.4.1.2 Individually Addressable LEDs WS2812B**

The individually addressable LEDs, WS2812B, will connect to the PCB through the ground and UCB0SIMO signals. This will allow the MSP430 to control which lights on the strips turn on. Three of these strips will be connected to an external power source of 5V 15A for testing. For the final PCB, we aim to have the same 5V source power the LEDs and PCB by adding a voltage regulator circuit that can regulate 5V to 3.3V and can handle 20A of current. As previously discussed briefly, it will be a more difficult voltage conversion circuit than the one we currently have as the current demands for the LEDs is much higher than typical linear voltage regulators. Nonetheless, through tools such as the TI Power Bench Designer, the team hopes to accomplish this design by the beginning of Senior Design II. It is possible that the new conversion circuit will also be able to power the Jetson Nano and cameras.

Figure 8 and Table 10 shows the connections from the custom PCB to the LED strips.

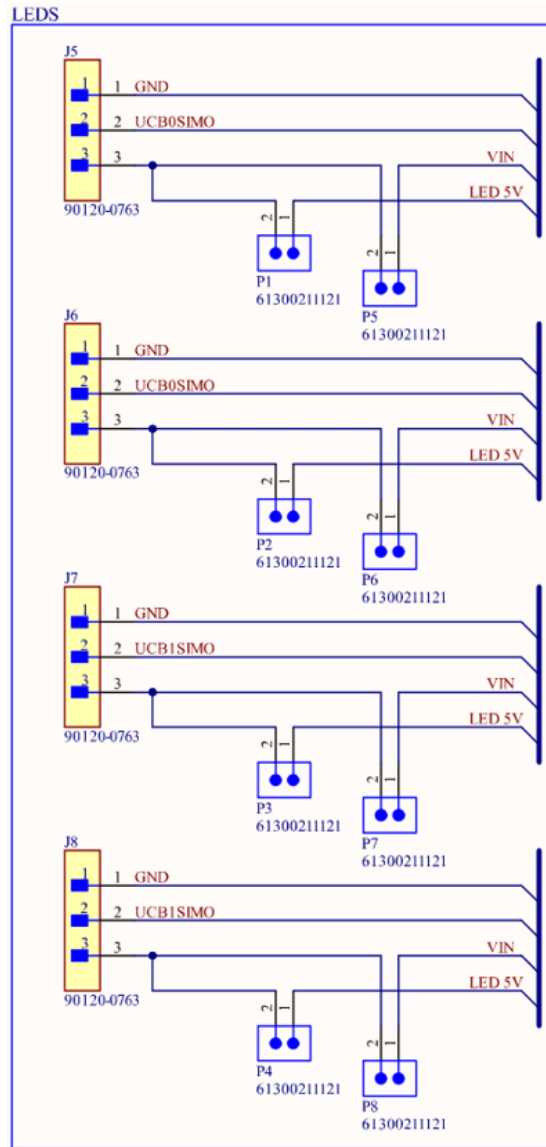


Figure 8: Addressable LED Schematic

The jumper pins on each of the LED headers is present so either the main DC jack (powers the microcontroller and bluetooth module) or the alternate DC jack

(solely for the LED strips) can be used. This is due to the amount of current required to run both LED strips at the same time.

**Table 10: Led Strip and PCB Connections**

WS2812B LEDs		connects to	Printed Circuit Board
Pin number	Pin functionality		Signal / Trace
1	VCC	→	External 5V 15A
2	GND	→	GND
3	DIN	→	UCB0SIMO

#### 6.4.1.3 JTAG and Debugging Header

The MSP430 launchpads used throughout our undergraduate courses have a debugging and flashing module attached that can be used on external MSP430 MCUs after proper disconnections and connections have been made. The module is called the ez-FET emulator and is shown in Figure 9. To complete these connections, we used the official MSP430FR6989 launchpad datasheet that has a specific section that gives instructions on how to use the launchpad debug attachment on external devices. This greatly simplified the complexity and cost of our custom PCB since we could eliminate our bootloader and flashing circuit module. The module now consists of a female pin header that we will plug our PCB MCU pins to the debugger module on the launchpad as shown below in the table.

It is important to note that from reading the datasheet, Texas Instruments only ensures this is a valid procedure for any microcontroller in their MSP430 series that has the pins listed in the table below with the same functionalities. The microcontroller we are using fits the criteria. Additionally, the RXD and TXD pins were optional, but the team decided it is a good idea to add them in case we want to later implement wired serial communication for any reason. It can also help to test faulty Bluetooth implementation and/or sensor. The complete wiring is shown in Figure 10 and Table 11.

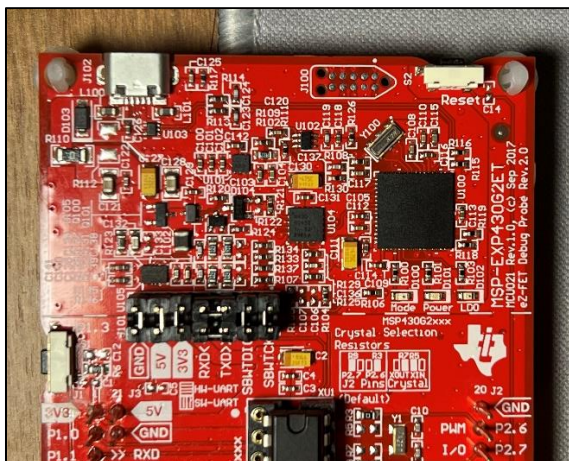


Figure 9: ez-FET

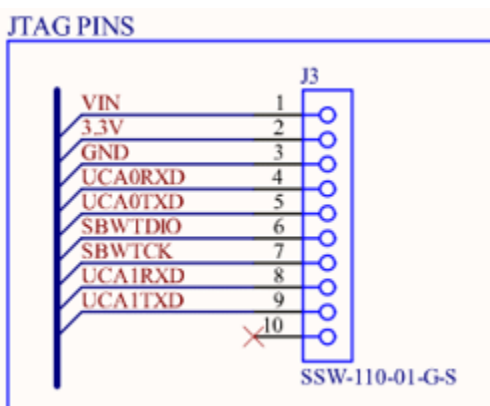


Figure 10: JTAG Schematic

Table 11: JTAG Connections to PCB

Printed Circuit Board		connects to	External Debugging Module
Signal / Trace			Signal / Trace
VIN		→	5V
3.3V		→	3V3
GND		→	GND
UCA0RXD	→		RXD <<
UCA0TXD		→	TXD >>
SBWTDIO		→	SBW RST
SBWTCK		→	SBW TST

UCA1RXD	→	RXD <<
UCA1TXD	→	TXD >>

#### 6.4.1.4 Debugging LEDs

Four LEDs were added to debug the five most important signals on the board. These signals include VIN, 3.3V, and Bluetooth RX/TX. When the signals are active, the LEDs light up. This will be helpful in case of transmission gone wrong, we can easily see if our PCB is connected correctly and differentiate between a software versus hardware configuration gone wrong. The schematic showing these connections is shown in Figure 11.

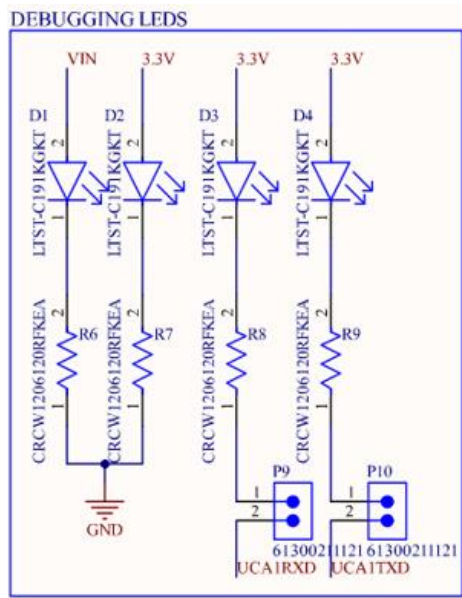


Figure 11: Debug LED Schematic

#### 6.4.1.5 Power Supply and Conversion

The power supply and conversion module connect a 5V power supply source through a barrel jack and converts it to 3.3V for the MSP430 chip. The 5V signal is also used to power the Bluetooth sensor. The 3.3V signal powers the MCU and the debugging LEDs. Many decoupling capacitors were added to ensure clear power lines that can filter out voltage spikes as well as an inductor. Figures 12 and 13 show the power conversion section of the schematic and how the MSP430 receives power.

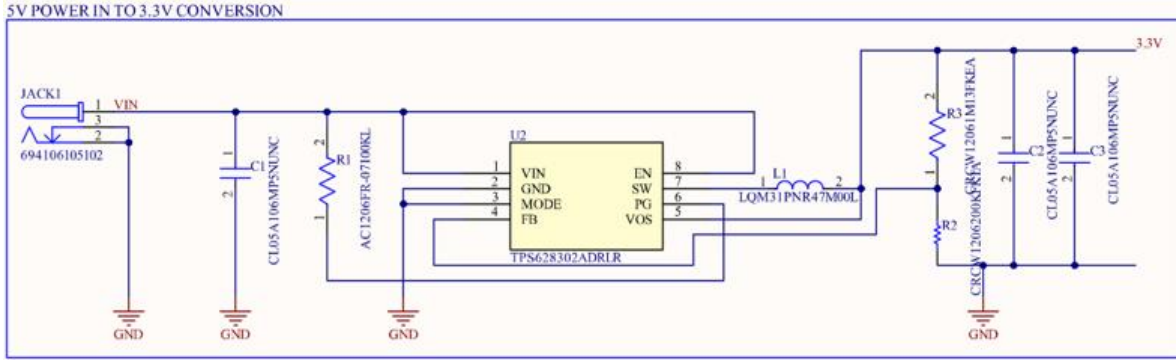


Figure 12: 5V to 3.3V Power Supply Conversion Schematic

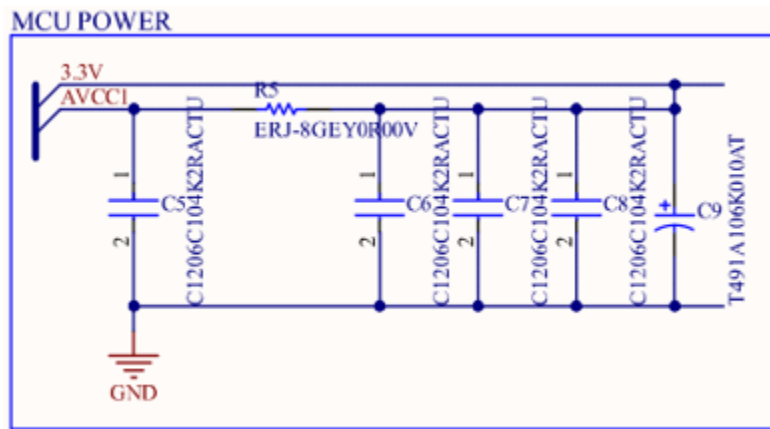


Figure 13: MSP430FR6989 Power Connections to PCB

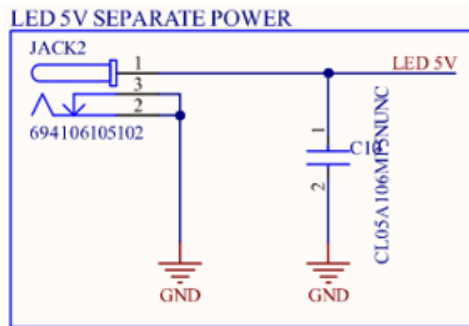


Figure 14: LED strip power jack

# 7 Software Design

ROBOT uses a complex variety of software integrated with our hardware in order to achieve its goals. The first programming task completed was to use Computer Vision to detect the text written on books on our bookshelf. To this end, libraries such as OpenCV and EasyOCR were implemented, modified, and enhanced in a python program. The efficacy of this program was tested by uploading sample images of a bookshelf into the program and testing the success of the results.

## 7.1 Computer Vision

ROBOT utilizes computer vision, implemented in Python, to detect the text of the books located on the bookshelves. To this end, multiple libraries and algorithms are needed. Computer vision involves two steps, optical character recognition (OCR) and image feature extraction. Optical character recognition is performed using EasyOCR, a library compatible with Python. Image feature extraction is performed using a combination of original programming logic, k-means clustering, OpenCV algorithms and image editing.

### 7.1.1 Computer Vision Flowchart

The programming logic for the computer vision component of our project outlines the sequence and functionality of our processing workflow. The objective of the script is to automate the organization of books on shelves using camera images, enhanced with OCR technology for text extraction. The flowchart is designed to offer a clear and concise representation of this process.

Starting with the initialization, the flowchart begins with the setup of EasyOCR and OpenCV, the libraries that provide the script's capabilities for text recognition and image manipulation. Images are captured using cameras alternately, and this sequential capture is depicted as a loop in the flowchart.

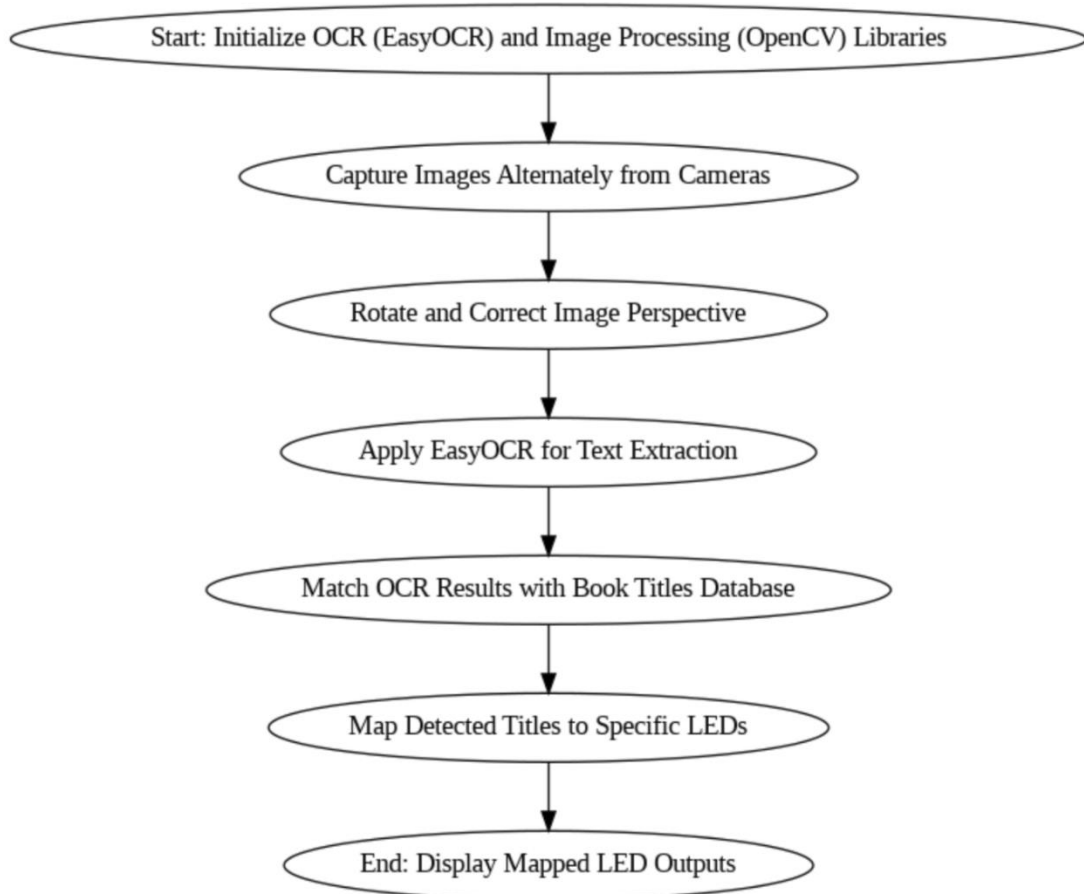
Following image capture, each image undergoes rotation and perspective correction to ensure the text orientation aligns with the OCR's requirements. This is important for preparing the images for effective text recognition and is displayed clearly in the flowchart.

The next step involves applying OCR using the EasyOCR library, which directly extracts readable text from the processed images. This library was chosen for its high accuracy and efficiency in real-time text recognition scenarios, such as detecting license plate numbers.

After the text extraction, the script matches the OCR results with a pre-existing database of book titles using fuzzy string matching. This step makes sure that each

identified text snippet is accurately linked to the corresponding book title in our system, despite small mismatches.

The final step of the script's operation is mapping the detected titles to specific LEDs on the bookshelf, which guide the user visually to the book's location. This involves calculating the appropriate LED based on the text's position in the image. The full process is detailed by Figure 14.

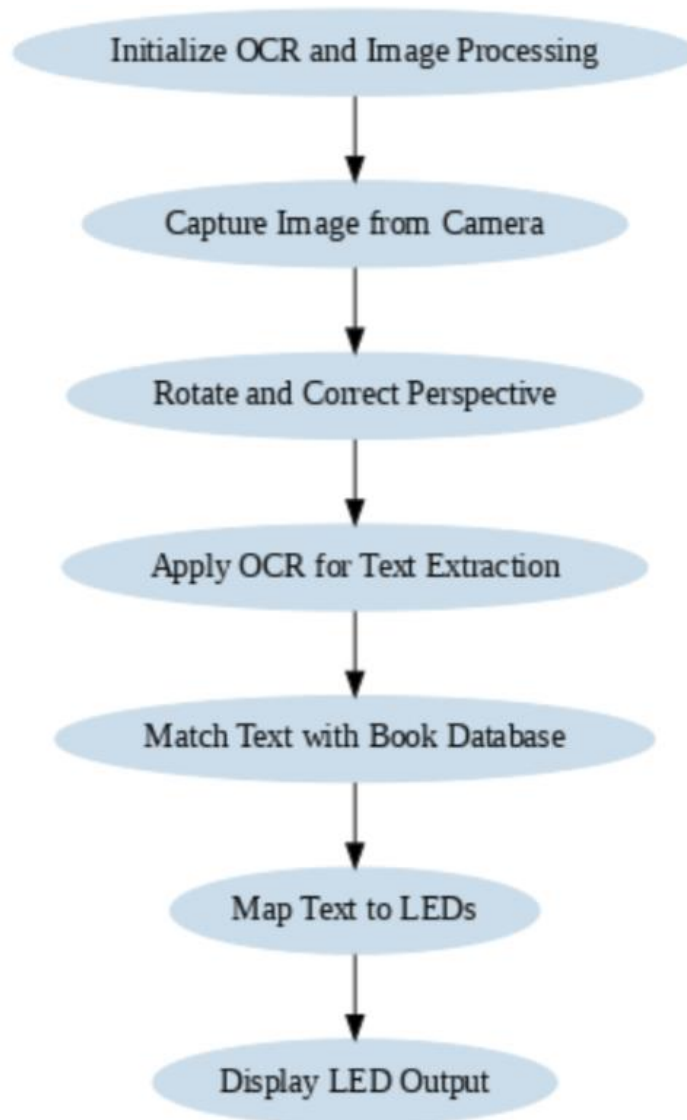


**Figure 15: Computer Vision Flowchart**

### **7.1.2 Computer Vision Use Case Diagram**

The use case diagram outlines the ways in which users interact with the software. During the testing phase, the user manually uploads the image. During the final integration, the user will be replaced by a camera periodically feeding images into the code.

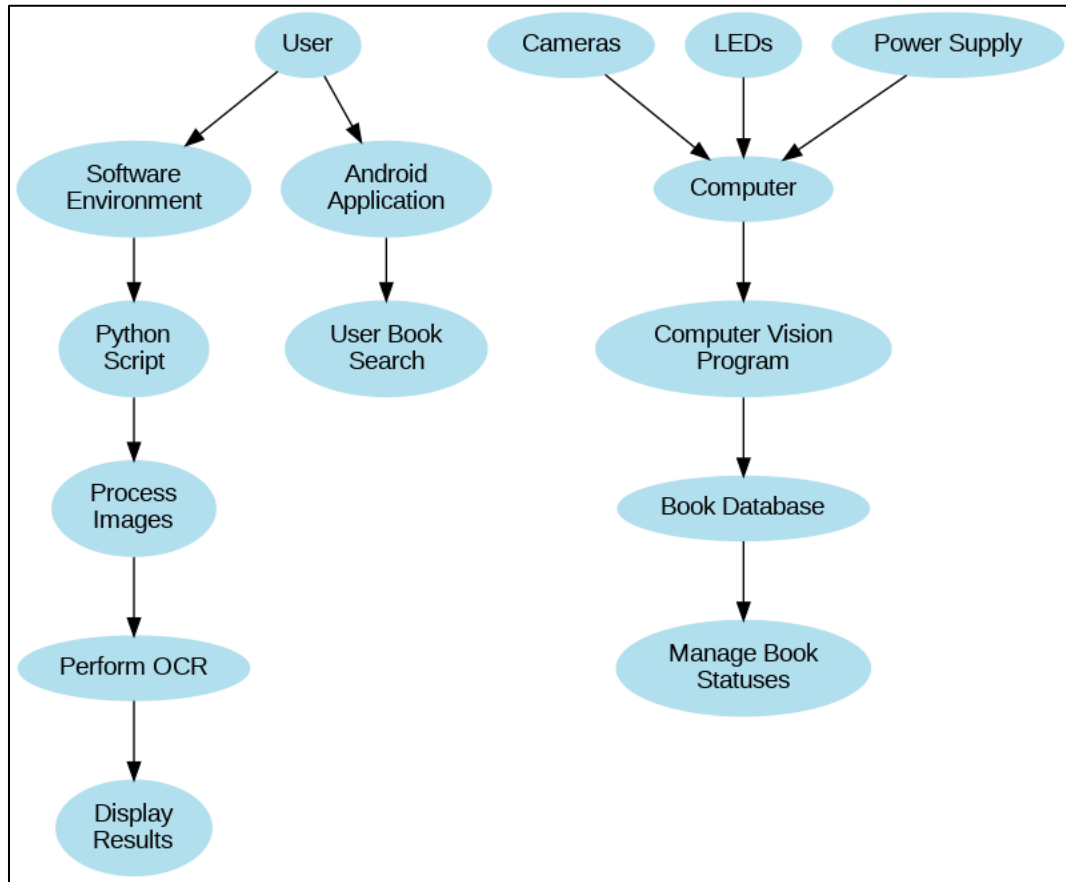
The user must first upload the image. After that, it is processed, rotated, and OCR is performed to extract the text. The resulting text is displayed as an output, as is described in figure 15.



**Figure 16: Computer Vision Use Case Diagram**

Figure 16 outlines the use case diagram of the computer vision logic integrated holistically. The diagram illustrates the workflow operationally. The user interacts with the system through two primary pathways: one is via a software environment where image processing tasks are conducted, and the other through an Android application for book searches. This works slightly differently in the prototyping, testing, and completed versions of the project. For testing, the user uploads images, but in the final product, the cameras automatically upload them. The system's central processor integrates inputs from cameras, LEDs for book location, and a power supply, functioning as the hardware control center. It executes a computer vision program to process the images of book spines, which are then analyzed by a Python script to extract text using OCR. The extracted titles

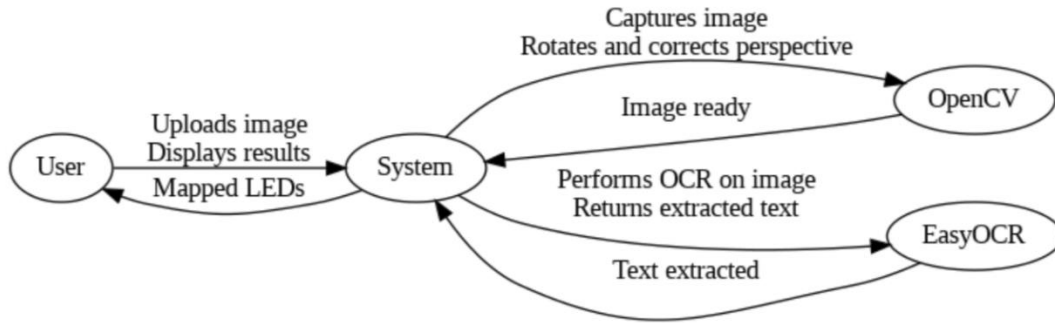
are displayed to the user, and concurrently, book statuses are updated in the database, maintaining an up-to-date, real-time record of the library inventory.



**Figure 17: Integrated Use Case Diagram**

### 7.1.3 Computer Vision Sequence Diagram

The sequence diagram outlines the steps of how users, the core system, and OCR tools interact to handle the text recognition from book images. When a user uploads an image, it kicks off the sequence of events. The system, acting as a central hub, manages the flow, initially sending the image to OpenCV. OpenCV's task is to adjust the image's orientation and fix perspective issues to get it ready for text detection, then sends it back to the system. Following this, the system hands the image over to EasyOCR, which is tasked with identifying and extracting text. Once EasyOCR has done its job, it sends the text it has found back to the system. The system then gathers this data and displays the outcomes to the user. This flow is visualized in Figure 17, making the steps and interactions clear.



**Figure 18: Computer Vision Sequence Diagram**

## 7.2 Computer Vision Software Overview

The script starts by setting up the necessary environment, which includes installing and configuring OpenCV for image manipulation and preparing to use the EasyOCR library for character recognition. It uses Python libraries such as NumPy for handling numerical operations.

Real-time image capture happens using cameras. After capture, each image undergoes a rotation and perspective correction to align the text for accurate OCR being read horizontally. The pre-processing ensures that text orientation is consistent across all images and optimized.

EasyOCR is used to perform text extraction on the processed images. Once the text is extracted, the script matches the recognized titles with an existing database of book titles to verify and finalize the detection process.

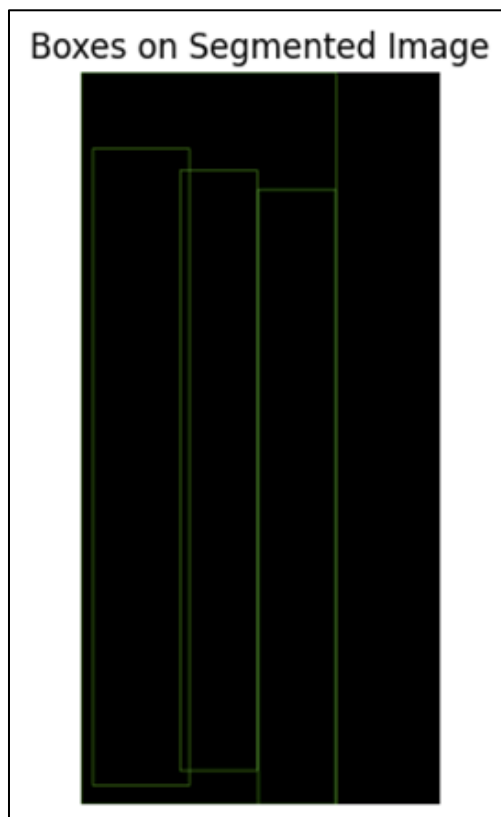
Following text recognition, the program maps the identified book titles to their corresponding positions on the bookshelf, using a series of LEDs as indicators.

The code then outputs the text extraction results, which are now linked to specific LEDs for easy identification on the bookshelf.

Previously, k-means clustering was used to segment the books on the shelves, demonstrated by figures 18 and 19.



**Figure 19: K-means Clustering**



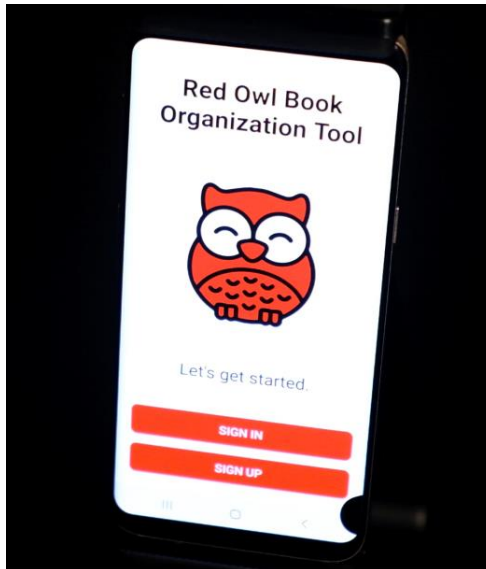
**Figure 20: Final Segmentation**

## 7.3 Coding Strategy for Application and Database

The application and database will work together to allow users to check out and search for books. The application must be able to retrieve information from the database such as book title, authors, and last seen on shelf date and time. The database should also be readable and writeable from the Jetson Nano running Python script and the application.

### 7.3.1 Application Code Implementation

The application code will be written in Kotlin using the Android Studio. As of December, the application code is still in progress as the group has taken this semester to learn Kotlin and Java, as well as completing tutorials and sample applications for Android mobile applications. A prototype of the login page colors and design is shown in Figure 20 and was completed in Figma. Figma is a popular application for brainstorming and building the user interface of applications.



**Figure 21: Sample Design of UI For App**

The mobile and web apps will have the following functionalities:

- Search for books
- View lists of all books
- Show title, author, genre and when the book was last seen on the shelf
- Show location of books
- Initiate an LED book find
- Check out a book
- Return a book
- Write and read to database

## 7.3.2 Database Code Implementation

The ROBOT bookshelf not only is able to identify the books but also catalog the availability and position of specific books on the shelf. As stated earlier, we will be integrating Firebase into the Android application. To begin this integration process, we must first create a project on firebase, then register the Android application with the newly created firebase project. However, the Android application does not know how the firebase is configured, so a configuration or '.json' file must be added to the application. This file is usually named 'google-services.json' and must be moved into the root level of the application to function. Once this .json is added a plug in known as Google services Gradle, which makes the configuration file accessible to Firebase for editing. After this, SDKs for Firebase can be added to the application to specify the products from the database system that are needed.

Now that Firebase is integrated into the application, code can be written in the firebase project for collecting the information from the book spines. First, a class system needs to be defined for taking the extracted string from CV. In addition, books can also be added without the use of CV creating a new variable that uses the same class as CV. Once CV extracts the titles from the books spines, adds them to the database as well as adding predetermined books code can be written to read the books in the database and display them in the Android application.

The Firebase database can also be used by the application to initiate a book search, check out and check back in a book. Searching for a book can be done by creating a user entered variable 'search'. When the book Title is given by the user, it will be entered in added to a query command that will search the titles in the database for a match. Because of this, capitalization is important as a given title may be available but will not show up due to grammar mistakes. If a book is found, it will notify the user and will illuminate the corresponding LEDs with the approximate location as well as the shelf location. After with the display will prompt the user whether they would like to check out the searched book. If the user is not sure what book they would like to check out, there will be a list of the current inventory with a brief description of the book as well as its genre and author. To accomplish the inventory feature, a variable can be added to the database project named 'checkedout' where if it is marked true, the book is checked out. Soon after the book is checked out, the database will record the date and time the book left the shelf. This is to provide the library patrons and application admins on how long the book has not been present on the shelf. When the book is checked back in via the display screen the value of 'checkedout' will be set to false to indicate the book is back on the shelf.

# 8 System Construction

This section will briefly summarize and touch on previous sections to show the goal of the end product and how each part will be used in the project.

## 8.1 Overall Schematic & Integration

The overall schematic is shown in Figure 21. The initial version of our custom printed circuit board will contain 4 main modules previously discussed in Section 6. At the beginning, we chose the 100-pin version of the MSP430FR6989 chip in case that we wanted to add additional features and have the ability to easily test on it. However, in our final design, if the 64-pin version allows for 2 UART connections and 2 clocks, we will change to that version to cut down the budget and make the PCB smaller in size. This will only be done if no additional and current features are affected.

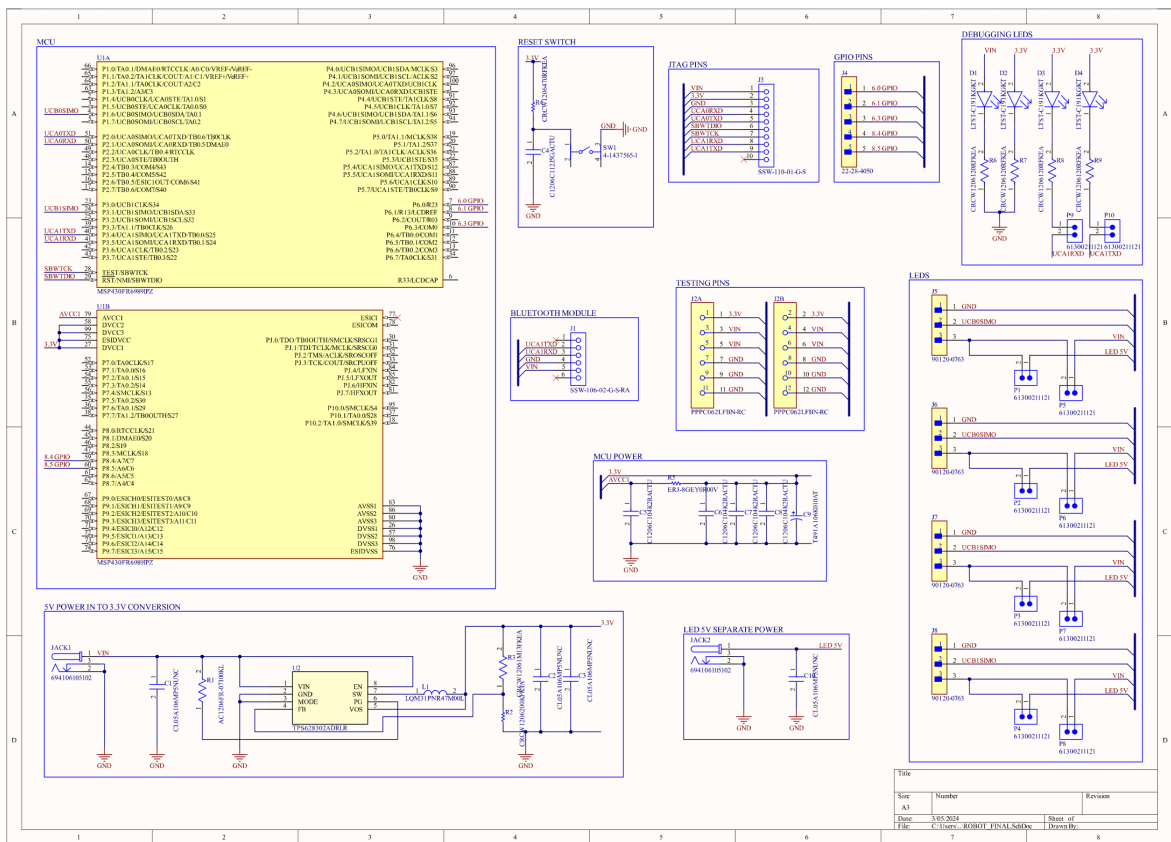
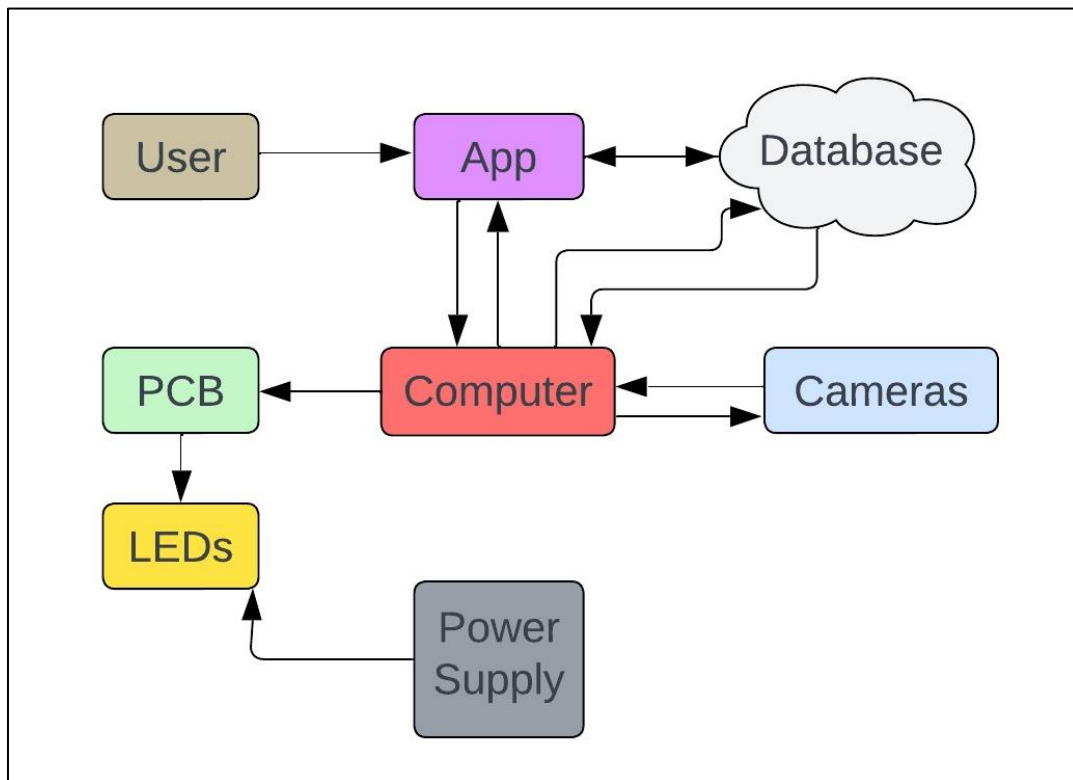


Figure 22: Complete PCB Electrical Schematic V2

Figure 22 shows how each subsystem will interact with one another.

1. The LEDs will show the user an approximate location of desired book when carrying out a lookup request.
2. The PCB will receive commands from the Python script running on the Jetson Nano when an LED book find is triggered.
3. The cameras will connect to the Jetson Nano and send it pictures of books on the shelf.
4. The database will maintain statuses of books such as last seen on, on shelf, currently checked out, etc.
5. The computer will process images and scans pictures sent by the camera and uses a Python script with computer vision capabilities to detect books and update database in real time.
6. The application will handle user requests and will be able to access the database, either through the Jetson Nano, or directly.



**Figure 23: Subsystem Interactions and Connections**

## 8.2 PCB Vendor Assembly

After creating the schematic and PCB design for ROBOT, we can finally determine the true pricing and lead time required for fabrication of the first version of the PCB. To accomplish this, multiple gerber files are created and exported from the CAD modeling software of our choice (EAGLE) into the online configurator of our vendor of choice (JLC PCB). As stated in a previous section, PCBway is a possible second

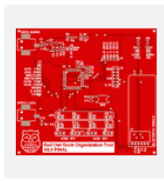
option for manufacturer so both vendors will be looked at in this section for their offerings.

## **8.2.1 PCB Specifications & Quotes**

Providing the ROBOT gerber files to JLC and PCBway in each online configurator, many options for PCB fabrication are presented, as well as some specifications that are detected and auto filled. One example of this autofill effect is in the dimensions and layer of the PCB, as these values are set in the CAD software. Other variables such as color, surface finish, and thickness can all be adjusted to what is desired. Changing the thickness of the PCB can negatively affect the durability and connections created. Outside factors may cause the PCB to flex and contort into positions that make sever connection and create shorts between routes. Due to this, 1.6mm will be used for the PCB as that is the standard with most designs. Changing color of the circuit board does not affect the connections but can affect the lead time for manufacturing as well as pricing. If the vendor does not have the chosen color option (Red) green will be chosen.

For the surface finish, the ROBOT PCB will be using HASL (Hot air solder leveling) with lead. Although there is a non-leaded option, leaded was chosen due to the decreased cost in manufacturing as compared to unleaded. For the VIAs present on the PCB, untent will be used. Untenting leaves the VIAs uncovered exposing the bare copper underneath, giving the same finish as surface pads. This is quite different from tented pads, which are covered by solder mask only exposing a small area of copper. Untented is superior as if the VIAs need to be soldered too, there is a decreased risk of damaging the pads if desoldered incorrectly.

One of the most vital factors for the PCB fabrication is the assembly of the components on the circuit board. Without assembly the cost is highly affordable, however, more work must be done attaching all the SMD components by hand. Given the number of pads required for some components, this will pose quite a challenge and can lead problems such as shorts and damaged pads, rendering the PCB unusable. To combat this possible pain point, we will be having each vendor assemble the PCB with the components. This choice does impact the overall cost of the order but saves time in the long run that can be used in other facets of ROBOT. Lastly, the Minimum Order Quantity (MOQ) will be set to the lowest amount of 5 for cost reasons. In addition, if there are unforeseen problems with the PCB design, there will not be a large amount of waste if a correction is needed. Below are the pricing quotes for JLC and PCBway. Each quote contains all the specifications addressed in this section, as well the pick and place components:



### PCB Prototype

Order #:Y1-7455815A

Build Time:3-4 days

5pcs \$10.37

[Product Details](#)

ROBOT\_GERBER\_FILES\_Y1

Production Completed

[Quality Complaint](#)

Merchandise Total:\$24.55

Shipping Charge \$41.62

Sales Tax:\$4.30

Order Total:\$70.47



### SMT Stencil

Order #:SO12403076085-7455:

Build Time:1 day

2pcs \$14.00

[Product Details](#)

ROBOT\_GERBER\_FILES\_Y1

Production file [↓](#)

Production Completed

[Quality Complaint](#)

**Figure 24: (JLPCB)**

Build Time	Qty	Total
<input checked="" type="radio"/> 3-4 days	5	\$ 5.00
<input type="radio"/> Express 48hours ?	5	\$ 38.96
<input type="radio"/> Express 24hours ?	5	\$ 63.56
<b>Assembly Service Price</b>		
Per Piece	Qty	Total
<input checked="" type="radio"/> 6.0/pcs	5	\$ 30
<input type="radio"/> 1.5/pcs	20	\$ 30
<input type="radio"/> 1.8/pcs	50	\$ 88
<input type="radio"/> 1.2/pcs	200	\$ 243.43
🔔 Final price is subject to our review.		
<b>Shipping Cost:</b>		US \$23.92
🇺🇸 UNITED STATES OF AMERICA		
DHL 2-4 business days, wt:0.57kg ?		
Shipment Date	Delivery Date	
2023/12/9 AM	2023/12/12	
PCB Cost: ?		US \$ 5.00
Assembly Service Cost:		US \$ 30.00 <del>91.29</del>
Shipping:		US \$ 23.92
Order discount:		US \$ -23.92
<b>Total:</b>		<b>us \$ 35.00</b>

**Figure 25: (PCBway)**

After factoring in the pick and place components for both vendors, the pricing is quite similar, only being a few dollars apart from each other. However, it is possible that one or both price quotes may change as the availability of components fluctuates a large amount throughout the year. If a component that we have chosen for the PCB becomes out of stock (such as a specific resistor or capacitor footprint) swift changes will be made to find a replacement with the same value and comparable footprint to the original. For this reason, we have compiled a BOM

separate from the one provided by EAGLE so the average prices of components can be recorded if a specific component must be ordered from a different vendor:

Column1	Column2	Column3	Column4	Column7	Column8
Part	Value	Device	Package	MANUFACTURER	PRICE
C1		08055C104JAT2A	0805_AVX		\$ 0.02
C2		08055C104JAT2A	0805_AVX		\$ 0.02
C3		08055C104JAT2A	0805_AVX		\$ 0.02
C4		08055C104JAT2A	0805_AVX		\$ 0.02
C5		T491A106K010AT	CAPMP3.4X1.6_1.8N_KEM	KEMET	\$ 0.14
C6		C0805C106K8PACTU	CAP_8PACTU	KEMET	\$ 0.05
C7		C0805C106K8PACTU	CAP_8PACTU	KEMET	\$ 0.05
C8		C0805C106K8PACTU	CAP_8PACTU	KEMET	\$ 0.05
J1		PJ-037A	CUI_PJ-037A	Cui Devices	\$ 0.59
J2		5103310-1	CONN10_2X5_S4UR_TEC	TE CONNECTIVITY AMP CONNECTORS	\$ 1.13
J3		61300211121	CONN_61300211121_WRE	Würth Elektronik	\$ 0.13
J4		PPPC041LGBN-RC	CONN_PPPC041LGBN-RC_SUL	Sullins Connector Solutions	\$ 0.28
L1		SPM6530T-R47M170	IND_SPM6530_TDK	TDK Corporation	\$ 1.17
LED1		LTST-C191KGKT	LED_LTST-C191KGKT_LTO	Lite-On	\$ 0.02
LED2		LTST-C191KGKT	LED_LTST-C191KGKT_LTO	Lite-On	\$ 0.02
LED3		LTST-C191KRKT	LED_LTST-C191KRKT_LTO	Lite-On	\$ 0.02
LED4		LTST-C191KRKT	LED_LTST-C191KRKT_LTO	Lite-On	\$ 0.02
R1		ERJ-3GEY0R00V	RC0603N_PAN	Panasonic Electronic Components	\$ 0.10
R2		CRG0805F470R	RES0805_CRG_TYCO_TYC	Te Connectivity	\$ 0.10
R3		CRG0805F470R	RES0805_CRG_TYCO_TYC	Te Connectivity	\$ 0.10
R4		CRCW0603100KFKEA	RES_CRCW_0603_VIS	Vishay Intertech	\$ 0.01
R5		CRCW04021M13FKED	RC0402N_VIS	Vishay Intertech	\$ 0.10
R6		CRCW0402200KFED	RES_CRCW_0402	Vishay Intertech	\$ 0.01
R7		CRG0805F470R	RES0805_CRG_TYCO_TYC	TE Connectivity	\$ 0.10
R8		CRG0805F470R	RES0805_CRG_TYCO_TYC	TE Connectivity	\$ 0.10
U1		MSP430FR69891IPZ	PZ0100A_N	Texas Instruments	\$ 10.81
U2		TP5628302ADRLR	DRL0008A_TP5628302ADRLR_TEX	Texas Instruments	\$ 1.07
					\$ 16.22

Figure 26: Edited BOM

### 8.3 PCB Layout

The final board layout of our first design after completing routing is shown in Figure 26. The top and bottom layers are both designated as ground signals and the decoupling capacitors were placed close to their corresponding pin to protect. The barrel jack connector was placed along the edge of the board with the hole facing away from the PCB. This ensures we can power the PCB without any awkward bending that might create noise and distortion. Additionally, the Bluetooth pins were placed on the edge spaced out evenly away from the MCU, pin headers, and power module to make sure there is not much signal interference and noise to distort data being transmitted and received. We also made sure not to run lines parallel in respect to the top and bottom as this can have unpredictable behavior and couple as a capacitor. Additionally, we made sure to not leave small strips of unconnected signals as this can interfere with other signals by behaving like an antenna. Lastly, the pins for the LEDs were placed on the left hand side of the PCB, away from everything else to make sure no awkward bending takes place at any of its three pins. We also did not want its cables to have to go over the entire PCB as this might make putting three of them in a daisy chain difficult. The LED strips must not be bent in sharp angles. If any of the physical features are limiting or a risk to the project, upon receiving the prototype, we will revise the design and board placement accordingly.



## 9 Prototype Test Plan

For the prototyping phase of the project, both the hardware and software were tested. Many goals and expectations of both were tested individually such as connecting an MSP430 chip to the Bluetooth module and controlling the LEDs were each their own separate software and hardware tests.

For the software portion, the first phase of the prototype was to test the computer vision capabilities. We also tested portions of the firmware side. For the hardware, many of the tests were self-integrated along the software tests.

### 9.1 Hardware Testing Environment

The Raspberry Pi Camera Module V2, Jetson Nano, and Raspberry Pi 4, the PCB, and LEDs are the main hardware elements used in this project; each has a specific function in the inventory system. When looking for a specific book, the Raspberry Pi 4 displays visual data that is processed by the Jetson Nano, processed by the camera module, and then passed to the PCB, which turns on an LED. Setting up the components physically is the first step in creating a hardware test environment. The Jetson Nano is positioned nearby for data processing, the Raspberry Pi 4 is linked to the touchscreen tablet to visualize the inventory data, and the Raspberry Pi Camera Module V2 is positioned on the bookshelf to take pictures.

It is important that all of the components communicate with each other correctly. The Raspberry Pi 4 is connected to the Jetson Nano for data processing, and the Raspberry Pi Camera Module V2 is attached to the Jetson Nano for initial data collection. The PCB is also connected to the Raspberry Pi 4, causing the LEDs to turn on when a book is being searched for. It is necessary to carefully configure every connection point in order to have uninterrupted data transfer.

Each component needs to have a proper power supply. To prevent unexpected system failures, the Raspberry Pi Camera Module V2, Jetson Nano, Raspberry Pi 4, PCB, and LEDs have power needs that must be satisfied. For reliable and consistent operation, there must be a stable power supply. One component of the hardware test environment is the data flow from the camera module to the Jetson Nano and ultimately to the Raspberry Pi 4. Careful evaluation of data transfer methods, signal, and device compatibility is necessary for achieving accurate and efficient data transfer.

Precise configuration is needed for the Raspberry Pi 4 to interact with the touchscreen tablet. Testing and improving touch sensitivity and responsiveness are key elements that ensure a smooth user experience when viewing the inventory data on the touchscreen. Comprehensive testing is necessary after the hardware test environment is set up. To verify the system's accuracy, responsiveness, and functionality, real-world scenarios are to be tested. Finding

and fixing any possible problems with data coordination, device compatibility, or user interface interactions requires continuous testing and improvement.

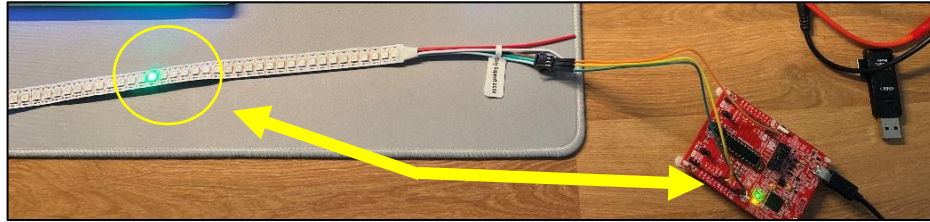
A hardware test environment for this project that integrates the Raspberry Pi Camera Module V2, Jetson Nano, Raspberry Pi 4, PCB, and LEDs in R.O.B.O.T is needed. We can establish an efficient and dependable system by making sure that the physical setup, interconnection, power management, data flow, and touchscreen integration are all done correctly. Proper testing and ongoing improvement are important to optimizing the hardware environment and laying the foundation for a smooth and effective system.

## **9.2 Hardware Specific Testing**

For the hardware, we tested individually addressing each LED, connecting a board similar to the MSP430 to UART for serial communication and sending and receiving commands through a Python script. We also tested several OpenCV algorithms with a camera.

### **9.2.1 LED Testing**

The addressable LEDs used in ROBOT are vital to achieve the goals set about in the project proposal. Because of this, it is even more important that the LEDs are functioning as intended. One of the tests that were performed was a color and brightness test. By cycling through both, we can check the color accuracy as well as the strip's dimming capabilities. If the color chosen does not match up if the color output, it may be a dying or malfunctioning diode. While if the diodes are too dim, more voltage may be required as the full 5 V is not being supplied across the entire strip of LEDs. Furthermore, checking the data received by the microcontroller can ensure that each of the individual LEDs are able to be addressed. This specific test is vital as if the data line is broken, portions of the strip may be unable to be detected, as there is a single data line with no redundancy due to the LEDs chosen. This data transmission, color and brightness can be tested by connecting the LED strip to a launchpad (MSP430G2GET). This connection combined with code written in Code Composer Studio resulted in the LEDs functioning as intended. This includes increasing and decreasing their brightness and addressing one, more, or all of them at the same time, each with its own color. An output for one of the tests is shown in Figure 27. Similar to our PCB, the data pin of the LED strip connected to the P1.6 pin on the launchpad that had the functionality of SIMO.



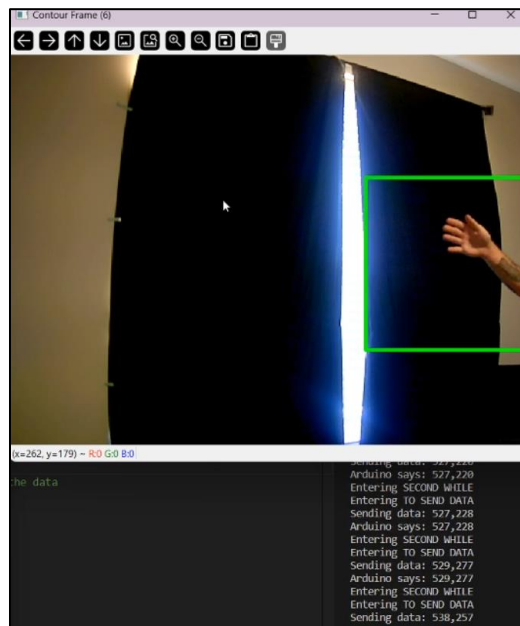
**Figure 28: Individually Addressable LED Testing**

For the live demo, there are a few tests that can be done to show the function of the LEDs in ROBOT. For instance, after using the display to search for a book, the LEDs can be used to determine the approximate location of the desired book. This will be shown with a range of 3 to 5 of the addressable LEDs illuminating in the area the book is present. Furthermore, when in alarm mode, the LEDs flash red and blue to indicate a theft has taken place.

### 9.2.2 Camera and Python Testing

The IMX219 camera in ROBOT is used for capturing the images for the CV portion of the project. For testing, we connected the IMX219 camera to a Raspberry Pi and wrote a Python script to open the serial communication port for the Pi board and sent data to and from the Python script and vice versa. The sample algorithms we ran included a motion detector that draws a green rectangle around the pixels with the most changes in real time as seen in Figure 28.

The script also opens the serial port. To make this work, the script was also coded through the VSCode IDE with lines that polled the receiving line. The Python script also polled its sending line for 0.5ms, then if it did not receive anything, it would assume the Pi did not send any new information.



**Figure 29: Motion Detection Results**

We got the following motion detection samples from our Python script. Where it detected the most motion, it sent the x, y coordinates to the Pi and the Pi received and then acknowledged the successful transmission by sending Python back what it received. Below are sample images of the algorithm working to find the center of maximum motion and the data transmitted between the Pi and script on PC.

This was done while the board and camera were powered and connected through wired serial communication to a portable laptop. Nonetheless, similar methods can be used to send information through Bluetooth from the Jetson Nano to our custom printed circuit board with the MSP430 microcontroller.

Additionally, this highly related to many small steps we must take in our project next semester. Now we are much more knowledgeable in opening and accessing serial com ports in Python and how to poll Rx and Tx lines effectively to ensure we do not lose important data at any given second. The camera testing also gave us an idea of the capabilities, frame borders, and angles of the IMX 219 camera. Then, a similar connection will be made with the cameras and the Jetson Nano running a Python script with access to Rx and Tx to the custom PCB. Like our test script, the Nano will also be implementing several algorithms and functions in the OpenCV library to detect book edges and extract the text from these images.

### **9.2.3 Jetson Nano Testing**

For processing the CV portion of ROBOT, we must ensure that the Jetson Nano is tested thoroughly during our live demo. The most vital part of the Jetson Nano is the Graphics Processing Unit (or GPU). It will be doing most of the heavy lifting will all of the graphic portions of ROBOT. For testing the GPU functionality, a suite of benchmarks will be used to rigorously test the Nano to ensure it can function under load for long periods of time. This is because ROBOT will be running for a large amount of the day, if not all day for both library patrons and security applications during the night. Deep learning models can also be used to stress-test the Nano as well as helping it distinguish different book from one another and improve its title recognition capabilities. For the database portion, it is important that the network functions of the are also working according to specification. Simply connecting the Nano via an ethernet cable to make sure it receives an IP address for data transmission. As for the camera's functionality with the Jetson Nano, it is critical that the serial communication works without issues. To test this, I2C and GPIO devices can be connected, and code can be written to ensure the information received is the anticipated value with no errors in communication.

### **9.2.4 PCB Component Testing**

This section covers the testing we completed on two important PCB components including the ultrasonic sensor and the Bluetooth sensor. We connected both components to the MSP430G2553 launchpad. Each test was done separately and only one component was connected to the launchpad.

#### **9.2.4.1 Ultrasonic Sensor Testing**

The ROBOT system is designed to be combined with a shelf to catalogue books in a library setting. Because of this, it is almost inevitable that ROBOT will come into contact with library patrons on a daily basis. With this interaction comes a possibility for damage and an opportunity to increase efficiency. To begin, in a live demo the ultrasonic sensor can be used to determine the distance between a patron's hand and ROBOT. If the hand of the patron comes too close to the product (distance pre-determined by the sensor) it can give a warning by flashing the LEDs. To increase power savings, ROBOT may be capable of entering a low-power state and waking up after detecting an object (such as a hand) is close to the sensor. This can also be deployed in the alarm mode where any change in distance from an object to the sensor will activate the alarm, which activates other hardware devices present in ROBOT.

We built a small, simple circuit to test the easiness of use and functionality of the ultrasonic sensor HC-SR04 with an MSP430 launchpad and received successful results. We made similar connections in our electrical connections for our custom PCB.

#### **9.2.4.2 Bluetooth Module Testing**

ROBOT works in tandem with an Android application to provide the user with a interface for the book system. This Bluetooth module is important in this aspect, as it gives a way for the library staff to monitor the activity of the shelf. For example, when a phone or Bluetooth enabled device is connected, it can give the admin as well as images taken from the Raspberry Pi camera. Furthermore, ROBOT can send push notifications to the connected device if it detects if something is within a certain range of the ultrasonic sensor (i.e. a patrons hand). ROBOT can also send push notifications when a book is missing for a specified amount of time when in alarm mode.

We built another small and simple circuit to test our HC-05 Bluetooth module and its ability to be connected functionally to the MSP430 family of microcontrollers. Our computer and launchpad were able to successfully send and receive messages to one another.

### **9.3 Software Test Environment**

The software prototype of ROBOT requires a testing environment. Each specific aspect of the software was tested in its respective testing environment. For the computer vision portion, since Python is used, Google Collab was an acceptable way to develop and test early stages of the prototype. [Talk about other parts here]

#### **9.3.1 Computer Vision Test Environment**

The test environment of the computer vision portion of the project involved multiple aspects. First, Google Collab was used to ensure the Python program worked correctly. Google Collab is an online free cloud service tool hosted by Google that allows users to write and execute Python code through their browser. It is very

useful for machine learning applications, making it a good choice for our project. Because Collab notebooks can be shared, they are also very useful for team projects such as ROBOT.

In order to ensure its usefulness, the text extraction software was tested on a wide variety of books, lighting conditions, and angles. Because we aim for ROBOT to be practical, the software must be robust enough to handle variations in the environment. To this end, a battery of tests was employed, and the software was refined as the prototype became more developed.

## **9.4 Software Prototype Testing**

ROBOT's software underwent a series of tests as part of the prototyping process. The computer vision portion involved taking, uploading, and processing a series of images to test the effectiveness of the Python text reader program.

### **9.4.1 Computer Vision Testing**

In order to test the computer vision portion of ROBOT's software, a plan was developed to make sure the software was sufficiently robust to handle the variety of environments, text fonts, and book conditions that practical applications might involve. When errors were discovered, the parameters of the code were refined to account for the conditions.

The plan to test the computer vision Python code is as follows:

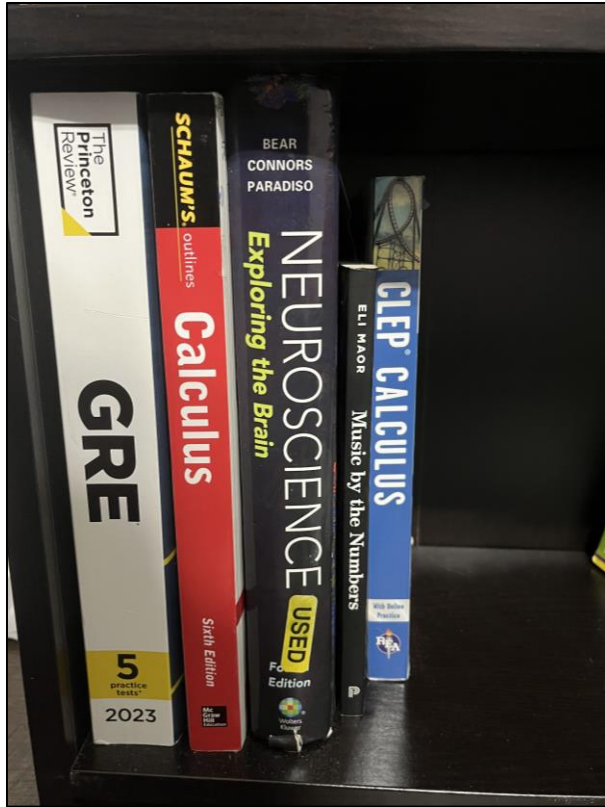
- Test EasyOCR before image feature extraction
- Test the first method of image feature extraction
- Test the second method of image feature extraction
- Compare the unprocessed results to both methods

#### **9.4.1.1 OCR Testing**

EasyOCR is an open-source optical character recognition engine that can be used to convert images of text into machine-readable text data. It's widely used for applications like document scanning, automated data entry, and indexing for search engines. It works well with clean, clear text, but can struggle with more complex layouts or poor-quality images without proper preprocessing.

The first phase of testing ROBOT's computer vision capabilities was to test how well OCR works on its own without any preprocessing. The purpose of this was twofold: the first reason being that it was necessary to see how well the OCR performs, and the second reason being that discovering its limitations would help establish what needed to be improved.

In order to test the OCR's capabilities, a handful of images of books on a bookshelf were taken, uploaded, and analyzed. The first image, figure 29, contains five books. The second image, figure 30, is the analysis.



**Figure 30: First Tested Bookshelf**

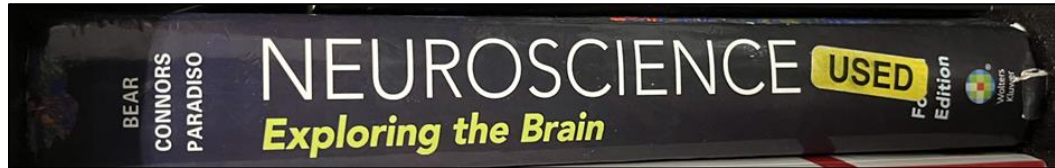
é1]	JBPAN y
by the Numbers	)
Music	V4
ELI MAOR	=
an	\$
-	uo!zPS
- Ss	a
LL	Exploring the Brain
¢ )	dvd
e	ZCO?
Se	=S3S3}
A	83})0e1d
O	ton
O-	os
=	U0
LI	ofEs
a	aC)
	-0aZ

**Figure 31: Results of Test 1**

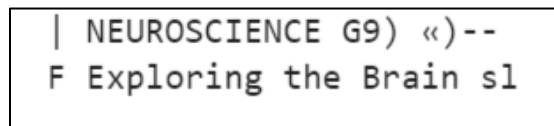
Although a handful of relevant phrases are visible in the extracted text, they do not represent complete titles. Additionally, there is a substantial number of extra

characters that do not seem to correspond to any title at all. Overall, using OCR without preprocessing the images in some way would not be a good approach.

In order to test the capabilities of OCR on isolated, horizontal book spines, I used image processing software to manually crop and rotate the image. Figure 31 contains one tested book image, and the output is visible in figure 32.



**Figure 32: Edited Book Image**



**Figure 33: Output of Edited Book Image**

By simply rotating and cropping the image, a significantly higher proportion of the text becomes discernable. However, this step was taken in an image processing software manually and was not performed by code. It was done to test OCR only. In order to accomplish this in Python, two methods of machine learning were implemented and compared.

#### **9.4.1.2 Greyscale Method Testing**

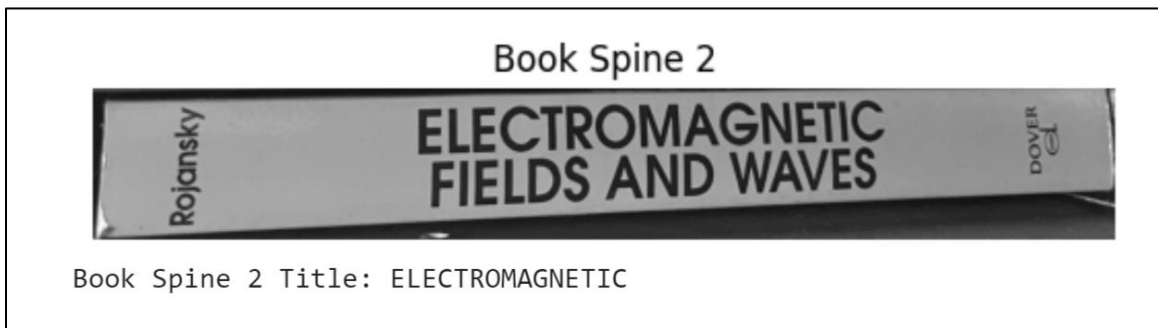
The first test underwent by the prototype in solved several steps. First, the uploaded image undergoes a series of transformations. These changes facilitate text segmentation and feature identification. Initially, the image is stripped of its color and converted into a grayscale version. This enhances contrast and simplifies subsequent processing steps. Following this, the image is rotated 90 degrees in a clockwise direction to align text regions horizontally for better recognition since most books on bookshelves stand vertically.

The next step involves thresholding, a technique applied to create a binary image where the text regions are segmented, distinguishing them sharply from the background. This step is important for the final phase, where the algorithm employs contour detection methods. By analyzing the shapes and structures within the image, the algorithm identifies the spines of books through filtering contours based on their area, which isolates them from other elements in the image. This intricate process results in a clear delineation of book spines so that individual books can be identified for the app. Figures 33 and 34 denote the image and the output result for this method.



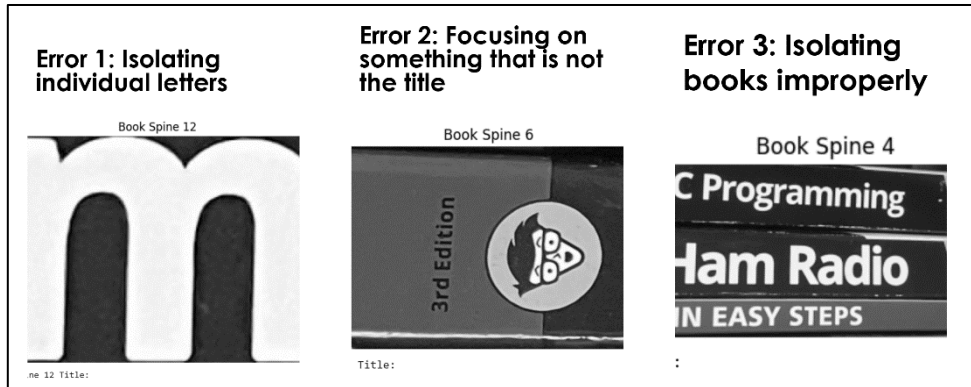
**Figure 34: Greyscale Method Testing Image**

Using this method, the books in the above image were processed. There was a highly variable success rate. While some of the trials using this method yielded more acceptable results, there were several serious flaws that made it clear that the prototype required further revision.



**Figure 35: Greyscale Method Results**

The “Electromagnetic Fields and Waves” book was more consistently read correctly. However, three types of errors generally occurred using this method, shown in figure 35.



**Figure 36: Errors**

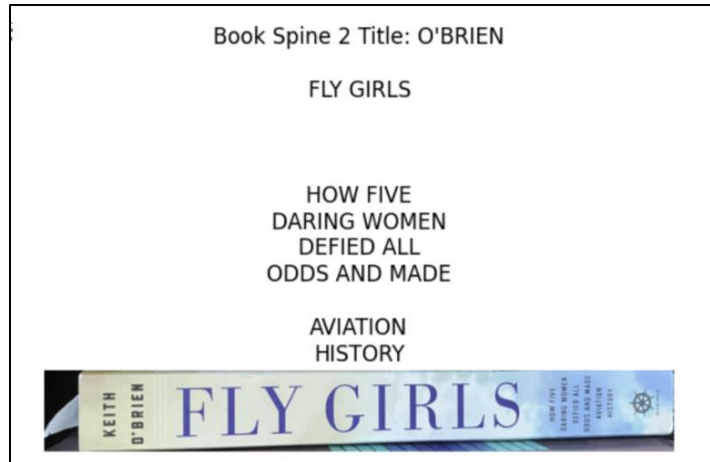
The three errors most often faced were that the program focused on individual letters, focused on something other than the title, and improperly isolated the books. In order to combat this issue, a different method of book spine isolation was implemented before the greyscale went into effect.

#### **9.4.1.3 K-Means Clustering Method Testing**

The second series of tests underwent by the prototype used K-means clustering to isolate individual book spines. The image is first processed to identify distinct color regions. Each color represents a possible book spine. This is achieved through k-means clustering. In k-means clustering, the algorithm groups pixels of similar colors together. The "k" in k-means represents the number of clusters the algorithm will create. For this program, it is the number of book spines.

After segmenting the image by color, the script analyzes the shapes within each color cluster to locate the book spines. It calculates the area of these shapes. It assumes that larger areas within a cluster likely correspond to the spine of a book.

Once the spines are identified, the script extracts these regions for text recognition. The goal here is to read the titles, which is where OCR comes into play. It converts the images of text into actual strings of text. A successful result is shown in figure 36.



**Figure 37: Successful Result**

Here, it can be seen that this method is highly effective. From a picture of four books, the program was able to isolate, rotate, and process this image so that Tesseract could extract its text. Though minor revisions may be made before the final presentation, the prototype of the computer vision portion of this project is largely successful.

# 10 Administrative Information

During Senior Design I, the team successfully completed many tasks set at the beginning of the semester. We made several financial changes and purchases to begin testing and implementing our design.

## 10.1 Milestone Discussion

So far, we have completed all the goals we set forth in our Divide and Conquer report. As a team, we have had outstanding collaborative meetings, and we make sure to hold one another accountable if we feel someone is not doing their part. Meeting attendance has always been timely with 4/4 of the team members showing up with eagerness to help and work on our project. While we have disagreed on several occasions, team members have remained respectful and are willing to hear each other's perspective with consideration.

The team began Senior Design I with four main ideas, one given by each member, and slowly, after meetings with advisors and instructor, selected one idea that we felt could reflect many of the group's strengths and weaknesses. After this, we divided up main portions of what we thought the project would consist of and assigned two people per section. However, as we began to carry out testing and small scale prototypes, we realized we needed to reassign and redivide sections. These sections and assigned person(s) can be viewed in Table 12.

**Table 12: Project Sections and Work Distribution**

<b>Project Section</b>	<b>Primary</b>	<b>Secondary</b>
<b>PCB &amp; Firmware:</b> Consists of choosing components, proper connections, and wiring it accordingly. Upon receiving custom PCB, it will be coded to fit the project's needs	Jostyde	Amanda
<b>Computer Vision:</b> Consists of implementing Python script to detect books and text accurately, ensuring Jetson Nano and PCB can communicate	Amanda	Montserrat
<b>App &amp; Database:</b> Consists of creating Android app through Kotlin and ensuring the Nano and app can read and write to the database	Montserrat	Joshua
<b>Hardware Design:</b> Consists of routing PCB and ordering. Will also ensure mechanical structures are fitting to the goals of the project	Joshua	Jostyde

Thus far, the group has successfully been able to:

- Test Python script communicating with a microcontroller through serial communication
- Learn OpenCV and EasyOCR library functions and uses
- Set up the Jetson Nano with the required operating system
- Individually control LED lights on a strip through an MSP430 MCU
- Collaboratively learn how to use Android Studio
- Collaboratively learn and complete small projects using Kotlin and Java
- Complete version 1 of the electrical schematic and board layout for our custom PCB
- Successfully flash and debug an external MSP430 MCU through an external launchpad with the detachable ez-FET module
- Detect book spines in images of bookshelves and individually crop each book into its own image
- Extract text from images with 60 percent accuracy

The group's progress so far can be attributed to the weekly meetings we held to check on individual progress and determine weekly goals as seen fit. Our main mode of communication to coordinate meetings were text messages and Discord, though most meetings were held in person, either at the UCF library or at teammates' homes.

Table 12 below outlines the goals set at the beginning of Senior Design I and the current statuses for each.

**Table 13: Team Schedule and Goals for Senior Design I**

<b>SENIOR DESIGN I</b>				
<b>Task</b>	<b>Start Date</b>	<b>End Date</b>	<b>Description</b>	<b>Status</b>
Brainstorming	8/21/23	9/6/23	Discuss project ideas with team and meet with faculty members to get their opinion. Get three topics that group agrees with and discuss advantages and disadvantages of each one.	Complete
Project selection	9/1/23	9/6/23	Review and analyze previous senior design projects and how they were able to successfully complete their projects. Choose one final idea out of the three and consult with professors again to get their opinions.	Complete
Divide and Conquer Draft	9/6/23	9/15/23	Complete Divide and Conquer draft that will help to outline the hardware and software needed for the project. Assign roles to each person. Form project committee.	Complete
1 <sup>st</sup> Meeting	9/20/23	9/20/23	Complete meeting with Senior Design I instructor and present Divide and Conquer draft.	Complete
Divide and Conquer	9/15/23	10/6/23	Fix draft based on feedback from Senior Design I instructor and project committee.	Complete
60 Page Draft	10/6/23	11/3/23	Begin working on 60 page draft document and expand on Divide and Conquer paper. Research PCB components based on required specifications.	Complete
2 <sup>nd</sup> Meeting	11/6/23	11/8/23	Discuss and present 60 page document to project committee and Senior Design I instructor.	Complete
60 Page Report	11/8/23	11/17/23	Fix final 60 page report based on feedback from project committee and Senior Design I instructor. Begin ordering parts for testing.	Complete
120 Page Report	11/17/23	12/5/23	Work together to finish 120 page report by December 5. Finalize design and order more parts for testing. Try to mock parts of project for testing and evaluation over the break.	Complete

Table 13 outlines the goals the group has set for Senior Design II, but have not decided on most of the target start and end dates yet. Next semester, we hope to

communicate more often with our advisors and collaborate with them more. In between the end of December and beginning of January, we will continue to meet once a week and set these target dates.

**Table 14: Team Schedule and Goals for Senior Design II**

SENIOR DESIGN II				
Task	Start Date	End Date	Description	Status
PCB Assembly	1/29/24	1/30/24	Finalize design for PCB and test on breadboards. Order a batch (5 or so) for testing. Test entire design with breadboard until PCBs arrive.	Completed
Testing and Redesign	2/3/24	3/6/24	Test design more and adjust as needed. Get input from project committee and Senior Design I instructor.	Completed
PCB Assembly 2.0	3/19/24	3/22/24	In case first batch of PCBs fail, ensure to analyze PCB schematic and routing for errors. Fix and reorder. Ask for help if needed.	Completed
Testing	3/22/24	4/13/24	Finalize physical implementation of project and test! Make sure project runs smoothly and is of good quality. Review website and documentation. Prepare for presentation. Ensure all group members are well versed in the jobs they were responsible for.	Completed
Final Presentation	4/18/24	4/19/24	Present project and prepare for evaluation.	Completed

Having a structured schedule as such helped to guide the team to continuously make progress throughout the semester and ensured we did not neglect the project.

## 10.2 Budget and Finance

This semester the team has purchased a few parts needed such as the Jetson Nano, single WS2812B strip, Bluetooth module, and IR sensors for testing. Additionally, the custom printed circuit board was ordered. The total price of items bought thus far is outlined in Table 14.

**Table 15: Items and Total Cost for Senior Design I**

<b>Device</b>	<b>Cost (\$)</b>
PCB	32.29
HC-05 Bluetooth Module	9.99
Jetson Nano	60
IR sensors	5.79
WS2812B strip	11.62
MSP430FR6989 100 pin	18.46
<b>TOTAL COST</b>	<b>105.86</b>

The Jetson Nano was purchased second-hand from a colleague; the Nano was tested and verified to be working prior to agreeing to purchase. The MSP430 chip was bought from Digikey. All other items were purchased on Amazon. For the upcoming semester, the group will make sure to purchase items over the holidays and in first few days of Senior Design II.

Our initial budget plan and estimation is as shown in Table 15.

**Table 16: Initial Budget Estimate**

<b>Member</b>	<b>Amount to Contribute</b>
Amanda	\$112.52
Joshua	\$112.52
Jostyde	\$112.52
Montserrat	\$112.52
<b>TOTAL: \$450.00</b>	

The group has only spent approximately \$100 this semester; we are definitely within budget, and we expect no complications as we previously overestimated to be within safe margins.

**Table 16: Final Budget Calculation Senior Design II**

Member	Amount to Contribute
Amanda	\$224.09
Joshua	\$224.09
Jostyde	\$224.09
Montserrat	\$224.09
<b>TOTAL: \$896.37</b>	

Expenses ...		
Aa Name	# Amount	Source
First PCB draft	\$59.52	JLPCB
IR sensors	\$5.79	Amazon
Microprocessor	\$18.46	Digikey
1 LED strip	\$11.62	Amazon
Prototyping boards	\$11.99	Amazon
Jetson Nano	\$60.00	Facebook Marketplace
Bluetooth module	\$9.99	Amazon
Voltage regulators	\$14.36	Texas Instruments
Solder flux	\$8.99	Amazon
Solder paste	\$9.99	Amazon
V2 PCB and Stencils	\$70.48	JLPCB
DIGIKEY FINAL COMPONENTS	\$119.64	DIGIKEY
Pin Header Sockets	\$9.99	Amazon
Pin Headers	\$5.73	Amazon
Jumper Caps	\$6.99	Amazon
Mouser Voltage Regulators & Resistors	\$24.19	Mouser Electronics
2 LED Strips	\$27.98	Amazon
Wire Jumpers	\$9.29	Amazon
HAMTYSAN Pi Touchscreen	\$69.49	Amazon
Jetson Nano Power Adapter	\$13.68	Amazon

barrel to usb	\$7.44	Amazon
Pin connectors	\$24.07	Digikey
Bookcase	\$57.51	Walmart
Thrift Books	\$15.00	OMG Thrift
Cable Management Covers	\$16.00	Amazon
Nails for Bookshelf Backi	\$15.00	
Double Sided Tape	\$7.00	
Extension Cord	\$10.00	
White Paint	\$4.00	
Acrylic sheet	\$18.09	
Two cameras	\$55.24	
Pi camera cable short	\$5.85	
Pi camera cable 6.56ft	\$10.64	
Bluetooth module	\$10.64	
Jetson nano case	\$18.95	
Raspberry pi case and fan	\$12.77	
sarah commission for painting	\$40.00	

+ New	Calculate	SUM \$896.37
-------	-----------	--------------

**Figure 39: Budget Breakdown**

# 11 Conclusion

The main goal of the ROBOT project is the development of a library system that will alter how users locate and organize books. The basis of this project is the integration of multiple hardware elements, which work together to simplify user interfaces and improve the entire library experience. This project, which goal is to have a user-friendly library environment, is done with the use of the Jetson Nano for data collection and processing, the Raspberry Pi 4 connected to a touchscreen tablet for the user interface, computer vision for book scanning, and the Raspberry Pi Camera.

## 11.1 Project Overview

The process begins with the Raspberry Pi Camera using computer vision to begin book scanning and collect data that will be processed by the Jetson Nano later on. This processed data is received by the Raspberry Pi 4, which serves as the center, and it is used to show relevant data on the touchscreen tablet that is attached. Its user-friendly design makes a big change from traditional library systems by enabling users to find and choose books with ease. From the tablet a signal is immediately transmitted to the LEDs through the PCB when a user picks a book. This sets off the LEDs, which then intentionally light up the exact spot where the selected book is located in the library. In addition to increasing user engagement, this real-time feedback makes navigating the library area visually simple. Our team's approach to creativity, problem-solving, and the goal of creating a more effective and user-centered library experience is shown by ROBOT.

### 11.1.1 Achievements

ROBOT is a great example of the dedication and creativity we had to have to overcome many obstacles to accomplish. The project was put through extensive testing during the development phase in order to verify the performance and functioning of each individual component and, most importantly, to make sure the system functions. During these testing stages, the team was able to improve and optimize the several components that make up the ROBOT library system in addition to using them as progress milestones.

Of all the difficulties faced, understanding LED control turned out to be a huge accomplishment. Our team demonstrated our abilities in handling delicate hardware connections by successfully setting up a command system via the PCB, which activated the lighting of LEDs based on user selections in a specific location. The requirements of the PCB schematic design were also successfully

navigated, demonstrating careful attention to detail and the ability to convert ideas into functioning systems.

Overcoming challenges when incorporating OpenCV into the system was an important event for the project. This achievement improved the ROBOT's general function.

A factor adding to the project's success was the achievement of real-time updates for book data. The system's dynamic, which is made possible by information that is updated instantly and continuously, gives customers access to the most recent and correct library data. The project's efficiency is based on its real-time capacity, which assures that users can easily grasp the most recent information on books that are accessible.

The team's combined efforts have produced a solution that not only meets the main objective of enhancing book search and administration but also pushes the limits of what is possible in the field of library systems.

### **11.1.2 Challenges**

Our team overcame several obstacles during the development of our project. A lot of coding was present and became the glue that held our project together as we worked our way through it. Although this connecting element created a sense of unity, it also presented a big obstacle. A successful integration of different technologies required great attention to detail, pushing the limits of our group's experience and understanding. The difficulties faced during the project were not a single incident but rather a repeated pattern, challenges that required creative approaches to problem-solving. Our aim, the installation of the library system, is progressing well, thanks to the strong foundation that was built during the planning process.

Another challenge came about when transferring our CV script to the Jetson Nano. With little familiarity with Linux or Ubuntu, it took our group a long time setting up the required libraries and Python environment.

Overall, our project was more than just a class project—it was a path of growth and development. Not only was the integration of technologies into the PCB an achievement, but it also demonstrated the problem-solving abilities our group possesses. With the completion of Senior Design I, we have not only created individual working parts for the system but also gained knowledge that will influence our future careers.

# Appendix

## References

- [1] "INTECH Smart Bookshelf - Efficient Way to Manage & Access Books," [www.cnintech.com](http://www.cnintech.com), <https://www.cnintech.com/smart-bookshelf.html>
- [2] S. Rani, "Implementing RFID In Library -Methodologies, Advantages and Disadvantages," International Journal of Library and Information Studies, vol. 7, no. 4, 2017, <https://www.ijlis.org/articles/implementing-rfid-in-library--methodologies-advantages-and-disadvantages.pdf>
- [3] Red Dot, "Red Dot Design Award: Smart AI Modular Bookshelf," Red-dot.org, Nov. 16, 2023, <https://www.red-dot.org/project/smart-ai-modular-bookshelf-26628>
- [4] "Intelligent Return Shelving | Bibliotheca Returns Solutions." [www.bibliotheca.com](http://www.bibliotheca.com), [www.bibliotheca.com/solutions/intelligent-return-shelving/](http://www.bibliotheca.com/solutions/intelligent-return-shelving/).
- [5] "SOLUTIONS RFID Borrowing RFID Returns Organization of Collections Protection of Collections Visit Statistics Organization of Collections." *ARFiD0*, [www.arfido.com/organization-of-collections,84.html](http://www.arfido.com/organization-of-collections,84.html).
- [6] "Orange Pi - Orangepi." [www.orangepi.org](http://www.orangepi.org), [www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html](http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html).
- [7] P. Fromaget, "Raspberry Pi Pros And Cons," Raspberry tips. <https://raspberrypitips.com/raspberry-pi-pros-and-cons/>
- [8] "Dev Board." *Coral*, [coral.ai/products/dev-board/](http://coral.ai/products/dev-board/).
- [9] NVIDIA, "Jetson Nano Developer Kit," *NVIDIA Developer*, Mar. 06, 2019. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- [10] G. Boesch, "What You Need To Know About NVIDIA Jetson in 2022," *viso.ai*, Mar. 21, 2022, <https://viso.ai/edge-ai/nvidia-jetson/>
- [11] Walter, "NVIDIA® Jetson™ L4T and JetPack Support," *Stereolabs*, May 21, 2022. <https://www.stereolabs.com/blog/nvidia-jetson-l4t-and-jetpack-support/>
- [12] "Jetson Nano Computer Vision Course - TensorRT & DeepStream SDK," [www.augmentedstartups.com](http://www.augmentedstartups.com). <https://www.augmentedstartups.com/jetson-computer-vision-course-sp>
- [13] "PiSugar2 - Portable Power Platform for Pi0." [www.pisugar.com](http://www.pisugar.com), [www.pisugar.com/](http://www.pisugar.com/).
- [14] "What Is a USB Port and How Can You Use It?" *Lifewire*, [www.lifewire.com/what-is-a-usb-port-818166#:~:text=USB%20stands%20for%20Universal%20Serial](http://www.lifewire.com/what-is-a-usb-port-818166#:~:text=USB%20stands%20for%20Universal%20Serial).
- [15] "MSP430FR6989." *Texas Instruments*, [www.ti.com/product/MSP430FR6989?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=epd-msp-null-](http://www.ti.com/product/MSP430FR6989?utm_source=google&utm_medium=cpc&utm_campaign=epd-msp-null-)

44700045336317329\_prodfolderdynamic-cpc-pf-google-  
wwe\_int&utm\_content=prodfolddynamic&ds\_k=DYNAMIC+SEARCH+AD  
S&DCM=yes&gad\_source=1&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8RI  
RFZQBjfqLJSSRP\_31fCI3K9\_r6\_Hsxf5bB9sFJuzWI5s82M-M6laAs-  
uEALw\_wcB&gclsrc=aw.ds.

- [16] "MSP430G2553." *Texas Instruments*,  
[www.ti.com/product/MSP430G2553?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=epd-null-null-GPN\\_EN-cpc-pf-google-wwe&utm\\_content=MSP430G2553&ds\\_k=%7B\\_dssearchterm%7D&DCM=yes&gad\\_source=1&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8Q6ltyUhVKOzBqVzh9v5fauy1TCDCIjVNHb2pKxN0z05n4GqhYRRjlaAtY5EALw\\_wcB&gclsrc=aw.ds](http://www.ti.com/product/MSP430G2553?utm_source=google&utm_medium=cpc&utm_campaign=epd-null-null-GPN_EN-cpc-pf-google-wwe&utm_content=MSP430G2553&ds_k=%7B_dssearchterm%7D&DCM=yes&gad_source=1&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8Q6ltyUhVKOzBqVzh9v5fauy1TCDCIjVNHb2pKxN0z05n4GqhYRRjlaAtY5EALw_wcB&gclsrc=aw.ds).
- [17] Arduino. "UNO R3 | Arduino Documentation." *Docs.arduino.cc*, 2022,  
[docs.arduino.cc/hardware/uno-rev3](https://docs.arduino.cc/hardware/uno-rev3).
- [18] "Raspberry Pi Documentation - Camera," [www.raspberrypi.com](http://www.raspberrypi.com).  
<https://www.raspberrypi.com/documentation/accessories/camera.html>
- [19] "IMX219-130 8MP Camera with 130° FOV - Compatible with NVIDIA Jetson Nano/ Xavier NX," [www.seeedstudio.com](http://www.seeedstudio.com), Nov. 23, 2021.  
<https://www.seeedstudio.com/IMX219-130-Camera-130-FOV-Applicable-for-Jetson-Nano-p-4606.html>
- [20] "IMX219-77IR 8MP IR Night Vision Camera with 77° FOV - Compatible with NVIDIA Jetson Nano/ Xavier NX," [www.seeedstudio.com](http://www.seeedstudio.com), Dec. 28, 2021.  
<https://www.seeedstudio.com/IMX219-77IR-Camera-77-FOV-Infrared-Applicable-for-Jetson-Nano-p-4607.html>
- [21] "BILLY Bookcase, white, 31 1/2x11x41 3/4"," *IKEA*.  
<https://www.ikea.com/us/en/p/billy-bookcase-white-30263844/>
- [22] "Amazon.com: ClosetMaid Bookshelf with 3 Shelf Tiers, Adjustable Shelves Tall Bookcase, Sturdy Wood with Closed Back Panel, White Finish," [www.amazon.com](http://www.amazon.com).  
[https://www.amazon.com/ClosetMaid-Bookshelf-Adjustable-Shelves-Bookcase/dp/B01N9LTFP7/ref=sr\\_1\\_5?crid=2GQNCZMJKRZ36&keywords=15%2Binch%2Bdeep%2Bbookcase&qid=1698423038&s=home-garden&prefix=15%2Binch%2Bdeep%2B%2Cgarden%2C84&sr=1-5&th=1](https://www.amazon.com/ClosetMaid-Bookshelf-Adjustable-Shelves-Bookcase/dp/B01N9LTFP7/ref=sr_1_5?crid=2GQNCZMJKRZ36&keywords=15%2Binch%2Bdeep%2Bbookcase&qid=1698423038&s=home-garden&prefix=15%2Binch%2Bdeep%2B%2Cgarden%2C84&sr=1-5&th=1)
- [23] "Better Homes & Gardens 3-Cube Storage Organizer, White Texture," *Walmart.com*, 2023.  
<https://www.walmart.com/ip/Better-Homes-Gardens-3-Cube-Storage-Organizer-White-Texture/812931457?from=/search>
- [24] "LED Strips, what's the difference? WS2811, WS2812B, 2812Eco, WS2813, WS2815, SK6812, SK9822.," [www.youtube.com](http://www.youtube.com), Jul. 31, 2019.  
<https://www.youtube.com/watch?v=QnvincC22hU>
- [25] "PCB Prototype & PCB Fabrication Manufacturer - JLCPCB." *Jlpcb.com*,  
[jlpcb.com/?from=VGE\\_PCBsupplier&gad\\_source=1&gclid=Cj0KCQiA67](http://jlpcb.com/?from=VGE_PCBsupplier&gad_source=1&gclid=Cj0KCQiA67)

- CrBhC1ARIsACKAa8TYBexWkHUQVGoc\_jiw7uxiykVj80HjFItCtjrf2w\_ue9  
QGM93hIrMaAoM3EALw\_wcB. Accessed 3 Dec. 2023.
- [26] "PCBWay." *PCBWay*,  
[www.pcbway.com/?adwgc=666&campaignid=172480651&adgroupid=8787904531&feeditemid=&targetid=kwd-297443275619&loc\\_physical\\_ms=9011779&matchtype=p&network=g&device=c&devicemodel=&creative=347469560617&keyword=pcbway&placement=&target=&adposition=&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8RV1XQyjW\\_5pfWdKFOSwyoJZP0RJ9CqX2Xo06ixy0-fP5mfrQy7GQ4aAgnMEALw\\_wcB](http://www.pcbway.com/?adwgc=666&campaignid=172480651&adgroupid=8787904531&feeditemid=&targetid=kwd-297443275619&loc_physical_ms=9011779&matchtype=p&network=g&device=c&devicemodel=&creative=347469560617&keyword=pcbway&placement=&target=&adposition=&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8RV1XQyjW_5pfWdKFOSwyoJZP0RJ9CqX2Xo06ixy0-fP5mfrQy7GQ4aAgnMEALw_wcB).
- [27] "Würth Elektronik CBT's Technology and Service Portfolio." *Www.we-Online.com*,  
[www.we-online.com/en/products/printed-circuit-boards/technology-and-service-portfolio](http://www.we-online.com/en/products/printed-circuit-boards/technology-and-service-portfolio). Accessed 3 Dec. 2023.
- [28] Center for Devices and Radiological Health, "Radio Frequency Identification RFID," *U.S. Food and Drug Administration*, Sep. 17, 2018.  
<https://www.fda.gov/radiation-emitting-products/electromagnetic-compatibility-emc/radio-frequency-identification-rfid>
- [29] F. Costa, S. Genovesi, M. Borgese, A. Michel, F. A. Dicandia, and G. Manara, "A Review of RFID Sensors, the New Frontier of Internet of Things," *Sensors*, vol. 21, no. 9, p. 3138, Apr. 2021, doi:  
<https://doi.org/10.3390/s21093138>.
- [30] "What Is Computer Vision? An Introduction," *University of San Diego Online Degrees*, May 11, 2023. <https://onlinedegrees.sandiego.edu/introduction-to-computer-vision/>
- [31] Python Software Foundation, "What is Python? Executive Summary," *Python*, 2019. <https://www.python.org/doc/essays/blurb/>
- [32] OpenCV, "About OpenCV," *OpenCV*, 2018. <https://opencv.org/about/>
- [33] Dharmaraj, "Text Recognition and Extraction In Images," *Medium*, Oct. 17, 2021. <https://medium.com/@draj0718/text-recognition-and-extraction-in-images-93d71a337fc8>
- [34] D. R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun, "Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm," *IEEE Xplore*, Dec. 01, 2022.  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10009215>
- [35] MathWorks, "What is MATLAB?," *Mathworks.com*, 2019.  
<https://www.mathworks.com/discovery/what-is-matlab.html>
- [36] "Computer Vision Toolbox," [www.mathworks.com](http://www.mathworks.com).  
<https://www.mathworks.com/products/computer-vision.html>
- [37] "Detect and decode 1-D or 2-D barcode in image - MATLAB readBarcode," [www.mathworks.com](http://www.mathworks.com).  
<https://www.mathworks.com/help/vision/ref/readbarcode.html>

- [38] “Localize and Read Multiple Barcodes in Image - MATLAB & Simulink,” [www.mathworks.com](https://www.mathworks.com/help/vision/ug/localize-and-read-multiple-barcodes-in-image.html). <https://www.mathworks.com/help/vision/ug/localize-and-read-multiple-barcodes-in-image.html>
- [39] M. R. Munawar, “Why Do Computer Vision Engineers prefer Python over Matlab?,” *Medium*, Nov. 11, 2023. <https://medium.com/augmented-startups/why-do-computer-vision-engineers-prefer-python-over-matlab-23484a6aa530>
- [40] Israel deh.39, “Kotlin VS Java – What’s the Difference?,” *freeCodeCamp.org*, Mar. 31, 2023. <https://www.freecodecamp.org/news/kotlin-vs-java-whats-the-difference/>
- [41] E. Jhonson, “Kotlin vs Java: Which is the Best Choice for Android App Development?,” *Javarevisited*, Oct. 26, 2021. <https://medium.com/javarevisited/kotlin-vs-java-which-is-the-best-choice-for-android-app-development-7c9fc782d2c9>
- [42] R. Payne, “Kotlin versus Java,” *Developer.com*, Mar. 24, 2023. <https://www.developer.com/java/kotlin-vs-java/>
- [43] “Create Mobile Apps with Firebase,” *freeCodeCamp.org*, Apr. 12, 2023. <https://www.freecodecamp.org/news/firebase-course-html-css-javascript/#:~:text=Firebase%20is%20an%20app%20development>
- [44] Firebase, “Firebase Realtime Database | Firebase Realtime Database | Firebase,” *Firebase*, 2019. <https://firebase.google.com/docs/database>
- [45] V. Singh, “Firebase For Web: Firebase Realtime Database,” *CodinGurukul*, Mar. 23, 2021. <https://medium.com/codingurukul/firebase-for-web-firebase-realtime-database-9280a52ced83>
- [46] JSON.org, “JSON,” [www.json.org](http://www.json.org). <https://www.json.org/json-en.html>
- [47] “What is SQLite?,” *Codecademy*. <https://www.codecademy.com/article/what-is-sqlite>
- [48] “PostgreSQL - Data Type - Tutorialspoint,” [www.tutorialspoint.com](http://www.tutorialspoint.com). [https://www.tutorialspoint.com/postgresql/postgresql\\_data\\_types.htm](https://www.tutorialspoint.com/postgresql/postgresql_data_types.htm)
- [49] PostgreSQL, “PostgreSQL: About,” *Postgresql.org*, 2019. <https://www.postgresql.org/about/>
- [50] Sudrajat, “Serial Communication: How It Works, Types, & Pros Cons of Each Model,” *FullyInstrumented*, Jul. 16, 2021. <https://www.fullyinstrumented.com/serial-communication/>
- [51] Grusin, Mike. “Serial Peripheral Interface (SPI) - Learn.sparkfun.com.” *Sparkfun.com*, 2019, [learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all](http://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all).
- [52] “I2C - Learn.sparkfun.com.” *Sparkfun.com*, 2018, [learn.sparkfun.com/tutorials/i2c/all](http://learn.sparkfun.com/tutorials/i2c/all).
- [53] “Introduction to the I2S Interface - Technical Articles,” [www.allaboutcircuits.com](http://www.allaboutcircuits.com). <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2s-interface/>

- [54] B. Miller, "The I2S Protocol and Why Digital Audio is Everywhere," *Keysight Technologies*. <https://www.keysight.com/blogs/tech/bench/2022/04/29/the-i2s-protocol-and-why-digital-audio-is-everywhere#:~:text=I2S%20is%20a%20serial%20bus>
- [55] Electronics | 69, Circuit Basics | DIY. "Basics of UART Communication." *Circuit Basics*, 13 Feb. 2016, [www.circuitbasics.com/basics-uart-communication/](http://www.circuitbasics.com/basics-uart-communication/).
- [56] "Infrared (IR) Sensors - STMicroelectronics," *STMicroelectronics*, 2023. <https://www.st.com/en/mems-and-sensors/infrared-ir-sensors.html>
- [57] "Bluetooth Basics - SparkFun Learn," *learn.sparkfun.com*. <https://learn.sparkfun.com/tutorials/bluetooth-basics/bluetooth-profiles>
- [58] Akash Peshin, "What Is The Range Of Bluetooth And How Can It Be Extended?," *Science ABC*, Jan. 09, 2018. <https://www.scienceabc.com/innovation/what-is-the-range-of-bluetooth-and-how-can-it-be-extended.html>
- [59] Marshall Brain & Talon Homer. "How WiFi Works." *HowStuffWorks*, 17 Aug. 2021, [computer.howstuffworks.com/wireless-network.htm](http://computer.howstuffworks.com/wireless-network.htm).
- [60] "Platform Architecture," *Android Developers*. <https://developer.android.com/guide/platform>
- [61] E. Cervantes, "What is Android? Here's everything you need to know," *Android Authority*, Dec. 03, 2019. <https://www.androidauthority.com/what-is-android-328076/>
- [62] "Android Overlays: How Do They Compare?," *PCWorld*. [https://www.pcworld.com/article/500163/android\\_overlays\\_guide.html](https://www.pcworld.com/article/500163/android_overlays_guide.html)
- [63] J. R. Raphael, "Android versions: A living history from 1.0 to 11," *Computerworld*, Sep. 11, 2020. <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html?page=2>
- [64] "Printed Circuit Board Basics: From Design to Final Artwork," *resources.pcb.cadence.com*, Jul. 20, 2022. <https://resources.pcb.cadence.com/blog/2023-an-introduction-to-printed-circuit-boards>
- [65] B. Lutkevich, "What is Firmware? Definition, Types and Examples," *WhatIs.com*. <https://www.techtarget.com/whatis/definition/firmware>
- [66] "What Is Firmware? (Definition, Types, vs. Software) | Built In," *builtin.com*. <https://builtin.com/hardware/firmware>
- [67] "Kotlin best practises," *kotlin-guide*. <https://unity-technologies.github.io/kotlin-guide/>
- [68] "Type-safe builders | Kotlin," *Kotlin Help*. <https://kotlinlang.org/docs/type-safe-builders.html#full-definition-of-the-com-example-html-package>
- [69] "Type checks and casts | Kotlin," *Kotlin Help*. <https://kotlinlang.org/docs/typecasts.html#smart-casts>
- [70] "Coroutines guide | Kotlin," *Kotlin Help*. <https://kotlinlang.org/docs/coroutines-guide.html>

- [71] “Coding conventions | Kotlin,” *Kotlin Help*. <https://kotlinlang.org/docs/coding-conventions.html#formatting>
- [72] “Guide to app architecture,” *Android Developers*. <https://developer.android.com/topic/architecture>
- [73] “C Programming Language Standard.” *GeeksforGeeks*, 10 Jan. 2014, [www.geeksforgeeks.org/c-programming-language-standard/](http://www.geeksforgeeks.org/c-programming-language-standard/).
- [74] “The Python Standard Library — Python 3.8.1 Documentation.” *Python.org*, 2020, [docs.python.org/3/library/index.html](https://docs.python.org/3/library/index.html).
- [75] “Generic Standard on Printed Board Design.” Available: <https://www.ipc.org/TOC/IPC-2221A.pdf>
- [76] “IPC-2222 - Rigid Board Design,” [www.saturnflex.com](http://www.saturnflex.com). [https://www.saturnflex.com/technology\\_hub/design-ipc-2222.php](https://www.saturnflex.com/technology_hub/design-ipc-2222.php)
- [77] A. Bahl, “7 Must-Knows for Your First Flex PCB Design,” *Sierra Circuits*, Jan. 25, 2023. <https://www.protoexpress.com/blog/5-must-knows-for-your-first-flex-design/>
- [78] “IEEE Standards Association.” *IEEE Standards Association*, [standards.ieee.org/ieee/3110/11253/](https://standards.ieee.org/ieee/3110/11253/).
- [79] “ChatGPT.” *Chat.openai.com*, OpenAI, 2023, [chat.openai.com/](https://chat.openai.com/).