End of Project Documentation

Autonomous Book Solution (A.B.S)

CPE 191/EEE 193B

Team 5
Team Name: A.B.S
Team Members: Khalid Shah, Emmanuel Silva, Farooq Alaulddin, Sajanpreet Malhi
Instructors: Professor Russ Tatro, Professor Neal Levine
Date: May 3, 2021

Table of Content

Table of Figures

Table of Tables

EXECUTIVE SUMMARY

A. Elevator Pitch

Our machine and interactive website allow the users to find and check out books from the library, all while maintaining social distancing and avoiding face-to-face interaction.

B. Executive summary

As we are all familiar with the current situation of the COVID-19 pandemic and how it has hit the world with a swift blow. Medical folks are out there risking their lives to help patients and others working day and night to find a vaccine. After given the senior project opportunity, our team decided to put our portion on decreasing face-to-face exposure and solve an issue that needs an urgent solution.

In March, when schools and universities started to move online, many issues surfaced because of it. One of the issues that many students suffered from was the library shutdown. Library is a resource for many students, not just for studying also for checking out books. As students, our team directly faced this problem, we had trouble checking out textbooks, books for our writing-intensive classes, and books that we needed for research. Many students rely on library textbooks as a primary source as they cannot afford to purchase a book, and other students prefer reading a hardcover book and have a hard time reading soft copies.

In May and June when we were discussing our individual problem statement, Farooq brought up the idea of creating a machine where students can check out books easily. This reminded us of the problems we faced and immediately we were all interested in solving this problem. We researched and found out this is currently one of the issues faced by libraries across the world. Based on our research we decided we need an online website for ordering the book and a physical machine to check it out.

To finalize our design for the checkout machine we researched and discussed ideas such as a warehouse belt system, a robotic arm to move a book from the counter to the user window, and an alternative way of sanitizing books using a specialized spray. All these methods had major faults such as high budget, implementation will take longer than a year, and could cause damage to the books. We decided to create a machine that scans the QR Code from the user, gets the book from a shelf cell, brings the book via tray and sliders to the user window.

We took apart our design idea piece by piece and research all the features for the machine and website. Next, as a team, we went through the process of elimination and eliminated features that were too expensive, would take longer than four weeks to implement, or would have been extremely complex. With narrowed down features, we created a detailed work breakdown structure. Completion of the work breakdown structure allowed us to see which features needed to be done first and which features could create blocks in the progress of the project. This enabled us to estimate the possible risks attached to our project and how they can affect our project progress and timeline.

After significant progress into our project, we revisited our problem statement to ensure the validity of our problem or any changes needed to meet the solution. We then formulated a device test plan to test our end product, and at the end of each feature completion we ran individual tests to ensure the functionality of the feature meets the requirements. Lastly, we tested after the integration of all features of the project to ensure the full functionality and completion of the feature. After testing was completed, we researched the market value of our project to see if we will like to deploy the full product in the market how well will it perform compared to other competitors.

At our end product website, the user will be able to create an account on the website, find books, get recommendations for books, search for the books, order books and get a QR Code e-mailed and saved on the website. Users can then at the selected time slot visit the checkout machine and scan the QR Code, which will send the tray via sliders to the cell where the book is located. Next, the book will be dropped on the tray and taken to the user window. After which the user will get the book and an email will be sent to the user.

We have completed both the website and checkout machine within our given timeframe, completing all the features within our budget. This report will look more thoroughly into the topics discussed above in prospect of our project and the end product along with other details of the background of the project.

***Abstract*—** There are thousands of different disasters happening and have happened around the world, but the disaster of the COVID-19 pandemic has left the whole world devastated! As most of the world is on stop and almost everyone is on a halt; however, work and education must go on and accommodations must be made to make the life of students and workers easy. This document will address one of such issues, an issue that has left millions of people without the opportunity to learn. This paper will shed light on the issue of libraries not being able to check out books to people along with this it will also provide detailed engineering solutions and design implementation. It will also cover the different features of this engineering solution and how making use of a website and physical machine can solve the issue. This paper is a product of research, personal experience, and discussion of the team (Khalid, Farooq, Sajanpreet, and Emmanuel). It includes data sets taken from various organizations and records for calculating the impact of the problem on people in the United States. This document also assumes readers are fully aware of the COVID-19 and understand the impact it has made across the world; however, the document will briefly go over the COVID-19 pandemic as background information. Lastly, this document will thoroughly go through the progress of the team from the problem statement to the end product.

***Index Terms*—** Books, library, automated library services, pandemic, accessible, social distancing.

I. Introduction

THIS document will go thoroughly into addressing the issue of libraries not being able to check out books as much as before the COVID-19 pandemic. In fact, some libraries might not even have checked out a single book since the pandemic started because of the lack of resources to adopt. Other libraries might have adopted some form of delivery methods, for example, drive-thru library, home delivery of books, and online e-books. As tempting as these solutions are, they are not the solution, they are more like a bandage on a broken bone, this document will provide a practical and achievable engineering solution. It will explain how this issue has affected the lives of students, professors, and others. Specifically, it will address a total of nine main sections, societal problem, design idea, funding, project milestones, work breakdown structure, risk assessment, design philosophy, deployable prototype status, and marketability forecast.

In the Societal Problem section, the document will dive into the background of the problem, details of the problem, probable solutions, and Team 5's solution to this problem. The background will talk in-depth about how this problem started, how team 5 found out about it, what other issues the team discussed, and why were they not selected. The background subsection will also include the personal experience of the team members dealing with this issue as students. Moreover, this subsection will include details on how overall, all libraries are affected by this issue especially in the light of COVID-19. In the details of the problem subsection, the document will dive more into the facts, data, and derive results from the data provided. This subsection will be a key subsection to address the issue, different aspects of how it is affecting the general public. It will also provide a deep analysis of the student's need for checking out a book. In the third subsection, the analysis of different solutions and different design problems will be conducted. It will also give a few data sets to show the reasons why physical books are still the need of today. It will also address a different aspect of how our project could have been implemented differently. In the fourth subsection, the document covers a brief description of our proposed solution. It includes the basic aspects of how it will be implemented and what are some key features along with the benefits.

The Design Idea section will be divided into three main components, the background of the design, website design, and physical machine design. This section is where the action of design and implementation details will be shared feature by feature. The background will talk in detail about how this solution was selected and how it is better compared to other solutions we considered. The website design subsection is where all features of the website will be discussed along with the implementation of each feature. For example, how a user account feature will be implemented and what will be some sub-features like the process of signing up, signing in, recommendations of books, etc. It also elaborates on software, programming languages,

libraries, and frameworks we made use of to implement all the features of the website. In the third subsection, the physical machine will be discussed and how each feature of this machine works. It will dive into details of what software, hardware, and libraries were utilized to create the checkout machine.

The Funding section will include details of the costs of the project parts, vendors, and how the team has paid or planned on paying the costs of the project. There will be a detailed table that lists each part along with its price, quantity, source purchased from, and detailed name of the product. Lastly, we discuss our estimated budget and the total cost of this project.

The Project Milestones section will be made up of main events that will be taking place in the timeline of our project. It will have a timeline of when a certain part of the project will be completed. The events and their assignment were organized using a Gantt chart and a PERT diagram.

The Work Breakdown Structure section is where all the data of the number of hours worked, tasks, features completed, team members who worked on each feature, and probable features that need completion is listed. It will provide a detailed overview of the flow of our project and how it went through. It is a vital section for looking at the overall time spent by the team and how can the structure of work could be improved.

In the Risk Assessment section, the team will discuss the probable risks to the project, users, team members and halt in completion of this project. We will also discuss the events that occurred during the project progress the slowed our project down. Along with an analysis of how the current situation of the COVID-19 will affect the completion of this project, university closure, lack of tools, and its effects on our project. The document will also provide possible mitigations to the probable risks.

The Design Philosophy section is where we discuss the inspiration behind this design for this project. How did we improve on our original idea? This section will dive into the brief details of the design and its importance of it. The section will be fairly short but on-point explaining the overall view of our project, design, and implementation.

From here our transition paper will transition into explaining the status and testing in The Deployable Prototype Status section. Here we lay out a detailed testing plan and provide testing results for each feature and project overall. This section was similar to a self-check assessment for the team to keep track of how was our project doing in our development, completion, and deploying phases.

The Marketability Forecast section holds the detail of the current market outlook for our project and how can we improve it. It will also provide a detailed analysis of the customer or client base and what we can expect if were to launch our project into the market. We will also provide the market value of our project, market share competition, and why our product is better compared to other already existing competitions in the market. This will be our last section of the project where we will be presenting unique details to the reader regarding our project.

After this section, our document includes a detailed conclusion of the report, references, glossary, and appendices with the user manual, hardware, software mechanical details of the project. It also includes the vendor contacts and project timeline and our team's resumes.

Overall, this document’s objective is to make a clear point of the vitality of the library, the need of users to check out physical copies of the book, and how our team’s project can enable the libraries to continue to checkout books whether there is a pandemic or not.

II. Societal Problem

A. Background

The outbreak of the novel coronavirus that originated in Wuhan, Hubei Province, China has made a permanent impact on millions of people in all nations throughout the world. The rapid spread of this virus has led to a pandemic outbreak that affected the daily lives of individuals who depend on their close contact necessities. In particular, jobs and education have taken a big turn for the worse as resources to which future students, employees, staff, and other individuals can access have diminished. One of these resources is public libraries which are one of the leading contributing factors for students and staff to educate themselves.

In March 2020, when California State University, Sacramento decided to move online, it was a sigh of relief for many students because of the obvious reasons of safety over everything. Especially, for those who are living or taking care of someone elderly or young. As a lot of good came from this decision, some issues surfaced with it too. From these issues, one of the most prominent issues is the library closing its doors to students and professors. Now, this issue might seem small, and some might consider there are alternatives available for it; however, that is not true! As students, our team has experienced first-hand the struggle of trying to check out a textbook for our class from the library, which ended up in rejection by the library due to the COVID-19 library policy. This led us to use alternative routes like purchasing the textbook or renting one, which causes a big dent in already struggling student bank accounts. In May, after forming our team we went over ideas that possibly we might consider and one of our teammates (Farooq) brought up this problem. Since we all went through this issue in the Spring 2020 semester all of us were very eager to find potential solutions to this problem. However, we were not sure of how big the problem is, what solutions can we implement, is it really a problem, will it get solved by itself after a while or is it going to stay for a longer duration. In the beginning, we were unsure if it is considered a problem or not so we decided to investigate other potential issues that we can possibly solve.

We went through three major issues Police officers getting in close physical contact with citizens, sanitization of gym machines, and a rising number of COVID-19. To solve the Police officer getting close in physical contact with a citizen, one of our teammates (Khalid) proposed the idea of making a specialized drone equipped with object detection that can detect the violation of traffic lights, seat belt violations, weapon detection, license plate detection to reduce the number of police officers. This idea was very good, but after researching potential implementation we came to realize that the implementation of this project requires dealing with the skill sets that the majority of the team have no experience in. It would require a lot of time for us to learn and implement this project, so we decided to not proceed with this idea. For sanitization of the gym machines, one of our teammates (Sajanpreet) presented the idea of a robot that can potentially clean and sanitize the machines automatically with different types of chemicals for different types of surfaces. Another idea that was presented by one of the team members (Emmanuel) was a robot that can potentially check the temperatures of people and enforce social distancing in public places. After going through all the potential issues, we can solve, we decided to proceed with solving the library book issue as it is more narrowed down and affected us directly.

Libraries have been affected due to this Covid-19 pandemic. According to the American Library Association, 98% of the libraries in the U.S. had to close due to this pandemic [1]. Only a small number of libraries stayed open with very limited services. According to the International Federation of Library Associations, public libraries in Albania, Argentina, Azerbaijan, Bangladesh, Bolivia, Brazil, Chile, Colombia, Costa Rica, Côte d'Ivoire, Cuba, the Dominican Republic, Ecuador, Ethiopia, Guatemala, Guinea-Bissau, Jamaica, Kazakhstan, Kenya, Kyrgyzstan, Madagascar, Malawi, the Maldives, Mauritius, Mexico, Myanmar, Nepal, Panama, Paraguay, Peru, the Philippines, the Republic of Korea, Saudi Arabia, South Africa, Ukraine, the United States of America, and Uruguay have been fully closed amid this pandemic [2]. This list includes some of the developed and developing countries, where keeping the libraries open could raise safety issues for the public and it would have the same effect on the library staff. Both sides have concerns about their safety and protection. Libraries have been known to provide resources and services to the communities, but due to this pandemic, most

of these services had to be stopped. Many libraries have continued their online services or even worked on expanding them. American Library Association also mentioned that about 74% of the libraries have tried to expand their services online [3]. Online services seem to be a quick solution in the current situation, but a lot of the resources and services like issuing physical books are lost. Converting physical books into digital books takes time and recourses that most libraries lack. So, what potential solutions can be there? To get potential solutions, we need to understand the problem and who it affects the most! Libraries were closed to keep everyone safe and slow the spread of the coronaviruses, but the effects of stopping the services can be seen in the public, especially students.

B. Details of the Problem

To understand or even consider something is a problem we need evidence to back it up. Now given the library situation, especially college and university libraries, there are almost no public records or data available for us to see how many students used the library for just the purpose of checking out physical copies of the books. There are a couple of individual estimates available on the web, but their credibility is unknown. So, to tackle this problem we will look at the numbers for public libraries in the United States. According to the Pew Research Center's survey conducted in 2016, two-thirds, 64%, of all the library visitors, ages 16 and older, visited libraries for borrowing printed books [4]. If we consider this number, there is a huge crowd of people that are currently restricted from checking out a physical copy of the book. According to the Institute of Museum and Library Services report from 2015, about 311 million Americans lived within a public library service area in 2015, and that same year 1.39 billion visits were recorded to the public libraries [5]. If we use the 64% of library visitors checking out a physical book rule on 1.39 billion visits, which will give us 889.6 million people checking out a book.

If we apply a similar percentage to the University library, assuming only 50% of California State University, Sacramento students, 15,578, out of 31,156 students visit the library and out of 15,578, 64% used the library to borrow books, we will have around 9969 students borrowing a book at least once a semester. If we include Professors and faculty into the equation and assumption that the majority of faculty use the library as a source for creating material for the class. Considering California State University Sacramento's total full-time instructional faculty, 830, if 70% of the faculty, 581, used the library as a source to prepare material for classes and 64% (372) of 581 faculty borrowed a book(s) from the library. In total, we will have 10,342 students and faculty checking out at least one book in a semester. If we apply this method on all 22 California State University campuses, we will have 159,818 students and full-time instructional faculty borrowing at least one book from the library as seen in Figure 1 [6] [7]. This number is based on an assumption, if real data was retrieved, the number of students borrowing a book could be way higher than assumed because of the fact that almost every university provides students with the ease of borrowing reserved textbooks for a short time. These numbers are just for California State Universities, if all universities, colleges, and public libraries across the United States are combined, the number of people borrowing books can easily cross 20 million. Given this scenario, there are currently millions of students, professors, teachers, and people of other professions who are seeking knowledge, but unfortunately are unable to gain it.

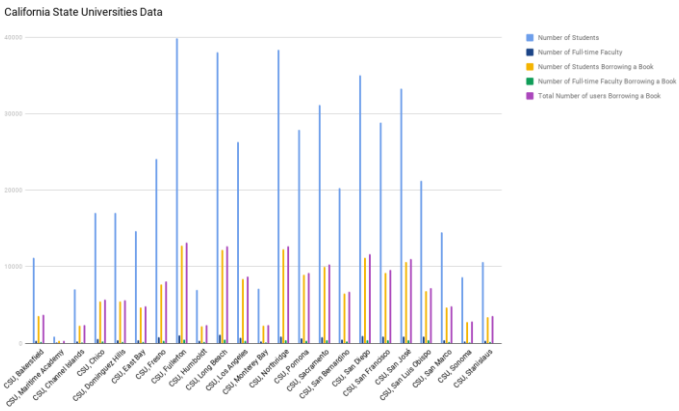


Figure 1: California State University Data [8]

Obtaining valid and credible information is a key factor to any project. Whether it was a school or a professional project, the data and the supporting arguments used for clarifications and further readings need to have authentic sources. I believe most scholars will agree on that. While we were brainstorming for our project, we asked ourselves how important is the library to students or professionals when collecting data compared to

alternatives such as online sources? We wanted to know that to clearly understand the problem we are trying to solve - which again is to automate the library's check-out systems. According to our research, libraries were and still are one of the first to-go places that scholars used to get or to confirm information. We concluded that libraries are still an essential location to obtain data despite the growing online research technology. However, libraries need to do better to keep up with the new technologies. One study done by Cynthia H. Kumah at the University of Nebraska, titled "A Comparative Study of use of the Library and the Internet as Sources of Information by Graduate Students in the University Of Ghana" showed that scholars do not evade libraries and rely only on online sources [9]. The participants in the study indicated that they would have fewer chances of running into misinformation compared to online. The study also indicated, "Few graduate students in the study mentioned influences such as difficulty in locating information or the need for convenience and speed when using the Internet." The study argues that it is true that the Internet provided fast access to massive collections of data. However, the fast access to such massive data could act as a negative factor and lead to misinformation due to the lack of fact-checking mechanism on some websites of the internet. Therefore, libraries are used to confirm these findings. This makes the libraries an essential part of any community.

Now, there is no doubt that libraries were negatively impacted by COIVID19 closures. As we saw earlier how important libraries are to scholars, we can observe the potential long-term problems that might arise. We are already seeing how communities are affected. According to Higher Ed, "Americans remain steady in their beliefs that libraries are important to their community, their family, and themselves. Two-thirds (65%) of all of those 16 and older say that closing their local public library would have a major impact on their community" [10]. This shutdown will impact students and their ability to use the library.

Another statistic found on Pew Research Center indicates, "37% of those living in homes with annual incomes of $30,000 or less say the possibility of closing their local public library would have a major impact" [11]. Low-income families rely more on libraries than the internet - or use libraries to access the internet. This is another factor that adds to the problem; students are losing easy access to the supplies they need.

Things are taking longer than expected to recover and go back to normal. According to the survey conducted by American Library Association, only 37% of the libraries in the US expect re-opening to begin to occur in June or July. Whereas 47% of the libraries do not know when their buildings would start to re-open to the public [3]. As the numbers are suggesting, re-opening is still far from becoming reality. The new mutations of the coronaviruses and the slow rollout of the vaccine are even delaying the safe re-opening of the libraries. This delay is also pushing libraries to look for alternative ways for their services like issuing books. According to the International Federation of Library Associations, a number of libraries are trying to continue their book issuing services by book deliveries and/or curbside pickup [12]. While these alternatives are quick brute force solutions, but these services are often very slow and still pose a risk of catching covid-19. An example could be found in new book issuing services provided by the Sacramento State library. According to the CSUS library website, the new book pickup requests are processed within 3-5 days and the delivery requests are processed within 7-10 days [13]. These book issuing services go through a long process before the books are available for students and faculty. The CSUS library has some lockers where they can leave the books. The students and faculty who request a pickup can go to these lockers and receive their requested books. This approach poses some risk of spreading covid-19 since the user needs to enter a password on a keypad and then open the locker. The user must touch multiple surfaces to get to their requested books. This raises many concerns and there must be staff available there to clean the lockers after their use.

Since the unexpected outbreak of the COVID-19, many institutions and businesses have begun to act towards a flexible work schedule. The number of companies that have switched to online platforms has spiked drastically. Although many have lost the ability to work remotely from home, most of the employees are using various platforms to perform their day-to-day work duties. Zoom, a video meeting software, saw a 40% jump from February 1 to March 18, and other platforms like Microsoft teams, Google Classroom, and Hangouts have more than 44 million daily users [14]. These are prime examples of how

the work industry and educational institutions have taken their steps to proceed to a better work and learning environment suitable for the communities. In a survey done in Ghana with 11 public universities, 58.3% of administrators did not utilize virtual means to perform their daily work activities. Out of these universities, only 31.3% of administrators had in-office and work virtual tasks and about 39.6% used virtual platforms to actively improve their performing roles [15]. The fickle situation has made a slow transition for a flexible work schedule so why not extend this to libraries with more flexible services. Adding a 24/7 checkout service autonomous book machine will allow the de-escalation of constrained library accessibility for all institutions and businesses. Some institutions have the technology ready to help improve the teaching experience using a remote/robotic microscope. These remote/robotic microscopes by Mikroscan Technologies, allow users to access slides and scan them while remotely viewing them [16]. Likewise, giving libraries a 24-hour service available to users will change the era of COVID-19 and become part of the new norm in a social distancing environment.

This research helped us understand deeply the roots of the problem so we can apply the right tools to provide a suitable solution.

C. Probable Solutions

A most casual solution for delivering books to the homes of those who need them is via e-book. This type of reading method has been around for the past decade but has not been fully implemented for all subjects in the education system. In general, it is typical for users to pick a physical book rather than an electronic one as a preference. In a study done by Pew Research Center [17], young adults and older adults share the same 6-7% likelihood of using digital-only books. This means that while the transition is there and is available for users to obtain digital copies of books, it is not the preferred source. Another reason why e-books could potentially be used over traditional books is their upgraded form of medium. However, unlike a traditional book, an e-book requires some type of technology to view it that may not be accessible to every user. The idea that one form of medium, whether it be a traditional book or e-book, can replace the other remains very controversial. This is because of the situational context in which each individual has towards reading a certain book. In situations like sharing a book with other people and reading a book to an infant, traditional books were highly preferable with 70-80% as compared to e-books. However, in another situation when people need books readily available on the go and easily accessible with a device, 76-85% of users preferred the e-book, according to Zhang and Kudva, in a sample of 2,986 individuals [18]. In this sense, users who still use traditional books are at a disadvantage if the transition to the e-book is the only option available. Out of the e-book users, only 11.7% completely stopped the use of traditional books while the rest of the 88.3% of e-book users kept on using print books [18].

Before we finalized this design approach, we went through a couple of other designs that we considered valuable and were key factors in helping us reach this final design. After finalizing this problem as our project, the first approach we took of solving this problem was using the warehouse conveyor system to get books to users. This idea came from the New York Times article, “That Mighty Sorting Machine Is Certainly One for the Books”, the article talked about how a New York Public library solved their problem of lack of book sorters by bringing a warehouse conveyor system to the table [19]. Inspired by this idea we wanted to implement something similar but instead have the belt deliver the book to the user standing outside. After some research and discussion, we came to realize that we will still need at least one person working at all times to put the book on the belt to be delivered outside to the user. This led to another idea of making use of a robotic arm that can pick up and place the book once the user outside requests it. After researching at the surface level to get to know how much it might cost us to build a conveyor belt and robotic arm, we realized that it will be out of our budget and will require longer than a year to implement. That is when we took Farooq’s idea of on-demand temporary storage of books and an easy delivery method without the use of a robotic arm and conveyor belt system.

D. Team 5’s Solution

The core issue of our Team Societal Problem is the closures that COVID-19 had caused which had affected library operations greatly. The engineering solution we are proposing would have a huge impact on libraries worldwide - it would allow libraries to

open in a way by utilizing newer technologies. Our solution involves a website that will provide a user-friendly dashboard that is customized based on a student's interests. The website will generate a QR Code that can be used at a certain time frame. This is to limit human interaction and keep people within a safe distance from each other. The website will be connected to a database that will contain the library books and will handle requests as they come in. Once a book is reserved, the user can go in the right time slot and get the book. The self-checkout system will only need the generated QR Code to read and pull the request and give the book to the user. The system will operate on itself with no need for library staff to be present at all times. This way we can make sure that everyone is safe, and the library could help students safely borrow books. Our design provides an interface where users are required to touch minimal surfaces. The user will be able to scan the QR Code sent to their email and available on the website, without touching any surface or camera. The only time they must touch something is the book itself when it lands on the checkout window. As many businesses and institutions are providing flexible schedules and it is becoming part of the new normal, our design also provides flexible 24/7 services depending on the user’s schedule. Our design will not only support the library services in this pandemic but also help them to adjust to the new normal. This will enforce the idea of social distancing at a new level, as users have a 10-minute window to check out their book. If their window expires or is scanned too early, they will be prompted to wait until their scheduled time window or if expired will be required to choose a new reservation through the website.

III. Design Idea

A. Background of the Design Idea

Now that the problem is well laid out and a brief picture of the solution is given, we can safely and easily go over our design idea. To give you an overview of this project, our project is divided into two different parts. One part is the physical book machine and the second one is the website. In the Fall 2020 semester, we will look at the physical machine or hardware portion of the solution. The physical machine has key features like QR Code scanning, which decreases the human to machine contact. Once the user receives the QR code via email by signing up and selecting their specific book, they will be prompted to scan the QR code at the machine at their selected time frame. The QR code will be decoded, and the machine will activate and find the location of the book on the shelf. Once the book location is confirmed, the vertical and horizontal sliders will activate to reach the cell location with that specific book. Once at the location, the program will give the signal that the destination has been reached. Then a servo will rotate the belts that will move the book forward and drop it onto the tray. IR break beam sensors will be used to confirm if the book has landed and activate the sliders to move back to the checkout window. Once at the checkout window, a servo will open the door at the tray and slide the book onto the checkout window for the user to pick up. The door will close by the servo mounted on the tray. In order to reset the program, another set of IR break beam sensors are used at the checkout window to send a signal to the program and reset it, making it ready for the next person to scan their QR Code.

Another part of this project is the website, which was implemented in the Spring 2021 semester. The website's main goal is to provide a platform for users to find books and order the book. When a user land on this website they will land on our homepage or dashboard page, which will provide the list of books. However, if the user decides to create an account or sign in to one, now the dashboard will be more personalized. It now presents a list of books that are more relevant to the user. The recommendations will be provided based on the user data, either provided by the user or collected by our website. For example, user data such as book checked out, books liked, authors they have read, subjects of the books, etc. When a user decides to create an account, they will be given an option to also enter information about themselves, in the future in case needed they will be able to edit and update this information.

Once the user decides to order a book, they will be given the option to add it to the Book Checkout cart. Once they add the book to the cart, it will now be reserved for them for the next 24 hours. The reason why this feature is implemented is because if it's the last copy of the book and when the user is ordering it disappears. To avoid a situation like this we decided to add this feature. Once the user safely starts the ordering process, the user will be prompted to choose a time and date to pick up the book. Once the checkout process is complete, the user will be provided a QR code both on-screen and in the email along with the pickup time they chose.

After the user arrives and picks up the book they ordered, the automated machine will communicate with the database to update the status of the book to picked up, along with sending an email to the user for pickup confirmation and due date. This whole process will be done without any face-to-face interaction of users or the library staff. To avoid the user-to-user interaction we have set times for each user to come and pick up their book. This design and implementation we have is probably the most effective way of solving this problem.

Before we finalized this approach, we went through at least two other different paths. Our first approach was to use a warehouse belt system to get the books outside of the window so the user can pick them up. That solution was fast pace and efficient; however, the major hiccup was that there needs to be at least one person that should be available to operate it. That person would place the book on the belt and start the belt to deliver the book outside to the user. Another flaw of this design was the cost, it would have cost us over 4,000 dollars to purchase the material to build the belt. Along with other costs of delivery and tax, it would have easily gone to 5,000 dollars. Another idea that we consider was a robotic arm that could either be used alone or integrated with the warehouse belt to eliminate human contact. However, we avoided the idea because of the time length, it would have taken us more than a year to finish the project. Given the short time frame of our class, we decided to choose an idea that is doable within the time frame and is cost-efficient. It is a unique way of combining hardware and website to provide a solution to this problem.

(Note: times below don't include the time of meetings, documentation, discussions, travel, etc… it is only time of physically working on the specific feature.)

Table 1: Feature Punch List [20]

Feature	**Estimated time to complete (Hours)**	**Assigned to**
Website: Homepage (Dashboard)	40 – 50 hours	Farooq
Website: Book Profile	30+ hours	Farooq and Sajanpreet
Website: User Account	50 – 60 hours	Farooq and Khalid
Website: Search Bar	10 – 20 hours	Khalid
Website: Book Recommendations	20 – 30 hours	Khalid And Emmanuel
Website: Book Checkout Cart	25 – 30 hours	Khalid and Emmanuel
Website: QR Code Generator	4 – 5 hours	Sajanpreet
Physical Product: QR Code Scanning	20 – 30 hours	Sajanpreet and Emmanuel
Physical Product: Vertical and Horizontal Sliders	90+ hours	Emmanuel And Khalid
Physical Product: Tray for Carrying the Book	35 – 40 hours	Sajanpreet and Emmanuel
Physical Product: Servo Motor Belts to Move the Book	30+ hours	Khalid
Physical Product: Book Landing Confirmation	40+ hours	Sajanpreet and Emmanuel
Physical Product: Physical Shelf	5 - 10 hours	Khalid

B. Website

As mentioned, our design consists of two main parts. The first part is the checkout machine that will be located at the library. The second main part is the online website. The website will allow the user to interact with the checkout machine. Users will be able to order and obtain the necessary directions that will be used in the checkout machine. It also provides many other features to give the user the best experience possible.

There are multiple approaches to consider the structure of our website. This was a critical decision to make because the technical implementation of our software features will vary. After research, we agreed on a base structure that will define the technologies that we were going to use for our future components. This base structure is based on the MVC model using Java Springboot as a backend framework with React as a frontend.

The MVC (Model–view–controller) is a software design pattern that organizes the code into three separate components. Each of those components will have a specific role [21]. The purpose of MVC is to separate the logic from the interface. This will add an additional layer of security since the logic is not in the same place as the interface. The MVC model consists of three main parts. The Controller handles the URL and or API requests coming in from the client-side [22]. Based on what was requested, the controller processes the requests and loads the data to the model upon success. The controller does not interact with the database nor the model interacts with the status of the request. If the request is corrupted, the controller will eliminate the request and it will never reach the model. The model receives the data from the controller and interacts with the database to perform CRUD operations. The model will return data to the controller once it's done with its processes. The controller calls for the view and provides it with the necessary information to be displayed. The controller acts as a middleman

between the two models. The crucial point to remember about this architecture is that the model and the view only communicate through the controller. This approach ensures that the presentation of data as well as the system logic is entirely separate, making it much simpler to construct complicated applications.

MVC design pattern offers many advantages. MVC provides support for fast and parallel development. MVC allows developers to work in parallel on different sections of the application. One developer can work on the view while the other can work on the controller. Rhetorically, it makes the process of development three times faster compared to regular programming. This is because there are multiple components being developed and tested simultaneously. This is ideal for our needs because each member of our design team can work on separate models with the ability to test independently. Also, debugging and identifying errors in MVC is much easier since every component is separate and moving from one model to another requires the success of a model. MVC makes appending and changing code much easier since it divides the applications into components. A good example of that will be using the React framework alongside MVC to act as the View model. The traditional View model in the MVC design patterns is written in plain HTML5 and JavaScript. However, the view in our case will be written using React JSX [23]. Developing the view model in HTML5 and JavaScript is doable, but it can get complex and hard to maintain quickly. Therefore, we wanted to approach the implementation of the View model differently [21].

React is an open-source, frontend, JavaScript library for building single-web applications. It provides a structured approach to design fast and reusable components. This library will push our applications' maintainability even further. Once we have the structure of the view ready, it would be easy to create and work on small segments of our application independently. Moreover, using the React library as our data presentation will improve the overall performance of the MVC model. The View component within the MVC model contains the HTML and JavaScript of different pages without any structure. This can cause a problem with mid to large applications. This makes it hard to update or apply changes. React provides a clear structure to our View model and forced our team members to follow it.

By identifying these structures early on, we were able to create a clear implementation plan for our software features. Since we were planning on describing all of our features, we needed to be crystal clear about our software design approach. These software features are described in the later sections of this document.

Programming in MVC design patterns is possible in many languages. However, we wanted to select a programming language that has large community support, clear documentation, and stability. These are important points that directly affected our progress. Therefore, the core of our backend development will be implemented using Springboot Java. Springboot Java satisfies all of the points above make it our best option.

1) Homepage (Dashboard)

One essential feature that our website has is a homepage. A simple dashboard design welcomes our users and allows them to explore the site in more depth. This page in practice had to be well designed because it gives the first impression to the first-time users. It contains many navigational components as well as data components. Navigational components mean buttons to direct users to various parts of the website such as sign-up/sign-in page, profile page, orders page, checkout cart, recommendations, etc... It will also contain data components to display lists of books available, the number of copies, number of likes and comments on the books, etc. This page will be the center of our website. Therefore, it must be well thought out. This page will be built piece by piece and put together by the small components such as search bars and the previously mentioned components. It might be the first landing page for our users. This page will let users access one of our next key features of the website, learning more about the book.

2) Book Profile

The Book profile is another great feature we implemented for our users. Book profile stores likes, comments, description of the book, the title, and subtitle of the book, ISBN, author, and a cover picture. The book profile contains a picture representing the book along with a descriptive paragraph. There will be other components to this page to help the users engage with each other. These

components are the Like button and comments. Having the Like button helps us track and identify highly active books as well as provide detailed usage data on all activity on each book page. This data is used in other components such as our search bar to provide targeted search results. Another great engagement feature comments, it is a great way to keep users engaged and active with the website. It also allows the users to see other reader's thoughts about the book. These techniques are used on many social media websites and it is proved to be highly effective. However, according to our research, these techniques are only useful if the users are willing to use them. It must be smooth, elegant, responsive, and most importantly user-friendly to be effective. Therefore, the implementation of these components took some time. We created an object for each book and update the database on any change in likes and comments. This required the use of Life cycle hooks. This means that we can update the database change insistently without loading the entire page. This way we ensured smooth transitions and updates of components when a change occurs. Also, we triggered an update on data change in the database so all users can see the same changes. All these operations are completed almost instantly to give the user the highest quality possible. Which made this feature more complicated, yet the most user-friendly.

3) User Account

To fully customize the user experience, we implemented a user profile page. This will allow us to collect any kind of data about the user. The data we are looking to have are but not limited to, Emails, Names, and Phone numbers. In addition to that, we will collect usage data. These data could be anything from what the user clicks, likes, and engages with. Linking such data to user profiles will allow us to build better predictions and suggestions models and provide a better experience.

There are many ways when designing a login system. It depends on the usage and contexts of the website. In our case, we used the JSON Web Token technique (JWT). JWT is a way for two parties to communicate securely over the internet. JWT enables the server to identify users based on encoded by secret key token that gets generated on the server-side and sent to the user to be saved as a cookie when at the login period. Once the user's browser saves the token, the browser could use it at every request to the server by attaching the token in the authorization header. This means that at every request the user makes to the server, we need to decode the information using the secret key. Any alteration to the token at the user side would result in an unmatched state in the server. Therefore, we can be assured that the token was not altered, and it is in fact the original. The token contains information like user identifier, end of a lifetime, email, first and last name, etc.

In order to implement a user profile page to collect the required data, we needed a complete and secure login and registration system. The implementation of such a system varies from one language to another. However, the design idea of it is fairly similar. Users will need to create an account and then log-in using their credentials. This feature will be necessary to checkout a book. However, it will provide many benefits as well.

One of the benefits of having a user profile page is that it would allow the system to send emails to the user. One of those emails is Reservation Acknowledgment. This email will be sent to the registered email address upon ordering a book. It will contain all the data needed at the physical machine to pick up the book such as the date and time at which the user is supposed to pick up the book, the QR Code needed to activate the arm of the machine, and the due date of the book. Another email is the Due Date Reminder. This email would be sent automatically by the system when the due date of a book is approaching. There is the Missed Reservation and rescheduling reminder email, and this email will be sent if the user failed to pick up the book. We implemented this feature utilizing the Java Mail Server This is great for our design needs since our website backend is designed using SprinfBoot Java framework which provides great functionality, and stability.

4) Search bar

To help our users find the books easily and efficiently we implemented a search bar. Even though the search bar can be considered a small portion of the whole project, but it is an important one. Without a search bar, users cannot really find books, especially if they are looking for a specific book that they need. The search bar has a feature of giving live suggestions as to the user search for the book so the user can find the book easily and in less time. To implement this feature of the search bar we utilized our database and React JS for the frontend.

5) Book Recommendations

Everyone likes to see what they like, need, or prefer. In the world of books, it is the same idea, users love to see more books based on the last books they checked out or based on the subject. Now there are more than a million ways you can recommend things to people, but they will only take action on the ones they feel connected to. For example, if I check out and finish a book, it is most likely if I see a book of the same genre, book written by the same author, or book that is a continuation of the last book I read, I will most likely take the book I see. On the other hand, if I see a random book that has nothing to do with me or what I like, it will be most probably ignored. In a world where data is the most important jewel in the crown of business, we should also think about applying it to the public sector to improve the sides that are lacking attention. On our website, we add a feature that will do just that, giving users the recommendation of books that they feel more connected to. We collect data as user checks out books throughout their time as our website user. Based on those books and their metadata such as subject, genre, authors, etc… we recommend books. When users will sign-in to their user account, they will land on their homepage, then they can navigate to the recommendation feature using the tab buttons on the left side of the website. This feature will provide an opportunity for users to get to know books that they could have never heard about, it will let them explore their interests at the same time provide the library a more user-preferred environment.

6) Book Checkout Cart

This feature has two main functionalities, one to let the user order the books and another to reserve pickup time to get the book from the checkout machine. Once the user adds a book to the Checkout Cart, it will automatically reserve the book for them for 24 hours so they will not lose the copy they need. However, this reservation is only temporary and will expire in 24 hours if they do not check out the book. When the user initiates the checkout process, it will prompt the user with open time slots when the machine is available to use. After choosing the pickup time and date for the book and completing the order, the user will be sent an email containing the information regarding the book, pick-up time and date, and the QR Code.

To make the website more user-friendly we implemented a sub-feature for previous orders. This sub-feature contains three main sections where previous orders of the user can be accessed, it contains all orders tab where all the previous orders are listed with each their status. The status of the book will be either, ready to be picked up, picked up, or returned. In other sections, one will contain the checked-out books and the second one will have returned books. prompted with a QR code to use at the machine. In this subfeature, there is information about every order completed, it also contains the QR Code for the orders. This allows users to access QR codes on the website as well just in case the email is lost or never received for any reason. This feature will make the ease of machine used by only one person at a time a lot easier and at the same time will help with social distancing because there will be never two people at one time at this machine.

7) QR Code Generator:

To implement this feature we had a couple of different options to go about, but given our framework and the programming language, we used the library that fits best with our design. We utilized the ZXing (“Zebra Crossing”) library, which allowed us to encode the order id for the book into the QR Code being generated. The QR Code is generation is triggered once the user completes the order. This QR Code contains the order id that is decoded by the machine. This order id allows the machine to pull up the data of the order and user from the database. It then confirms if this is the correct date and time of the user and when it is it will start the next process of giving the book.

C. Physical Product

1) QR Code Scanning

Once the user has received the QR code for their reservation in an email. They will be able to go to the location in the designated time frame where Autonomous Book Solution is located to check out their reserved book. The checking-out process will be initiated by holding the QR code in front of the camera. The camera will read and decode the order id number from the QR code and confirm it use it to get the information regarding the reservation time and date, user, and book name from the database. We are using a Logitech C270 webcam because of its reliable functionality and affordable cost. This webcam is connected to Raspberry Pi 4 through a USB cable [24]. We used python 3.8.5 to program this webcam to scan the QR codes. We made use of

the OpenCV library in python which will allow us to decode information from the QR code and match it with the database [25]. This is the first feature user interacts with and requires a couple of seconds to decode, connect to the database, and retrieve information to send to the next feature horizontal and vertical sliders. However, this feature is one the most vital feature to filter out bad QR Codes, expired time slots, before time, after time, etc… If it is in any other condition above it will send the user a problem-specific email to let them know why their order was not delivered to them. This prevents users from coming before or after their specific time slot or scanning a false QR Code. Once it meets all the conditions it then moves the program forward to the sliders.

2) Vertical and Horizontal Sliders

The idea behind the vertical and horizontal sliders is to transport the book from the cell to the user and back to its default station. Previous ideas, like constructing a mechanic arm that will grab the book and carry it over, were simply more work than intended for this project. Not only will it consist of more time and experience to build, but also consume a lot more coding and budget expenses. Since the sliders will be used as communicators between each cell, the range at which each axis will transfer the book will be divided into the x and y components, as shown in figure 2.

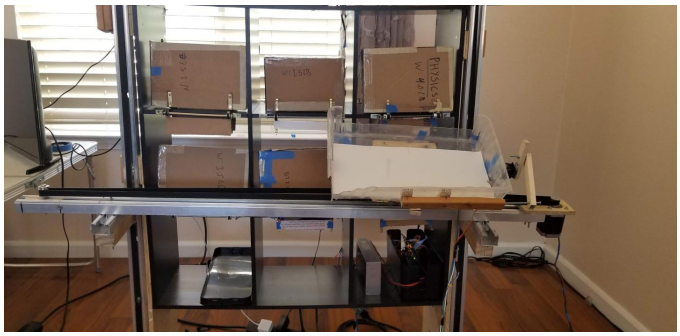


Figure 2: Rail system with stepper motors and tray [26]

Horizontal component: It is necessary for the slider in the x component to obtain the most critical calibrations as it will be the means by which the book will travel from cell to cell carrying the special tray. For the tray to move, we decided to construct a rail system that will extend from each end of the shelve. The rail was originally planned to be made of two tubes that will transport the tray by gliding using bearings, with an 86-90 cm length and a much smaller diameter than that of the vertical tubes. The horizontal rails were to be attached to a piece of wood that will connect to the vertical rails with bearings. However, the amount of calibration for each bearing to slide with the tubes would become inconsistent, not to mention the cost of this idea was already more than we could afford in our budget. Instead, we decided to look for other ideas. Sajanpreet found an alternative to replace our previous idea. Since our project needed a railing system, we thought of buying something that would reduce the amount of physical labor and cost for this aspect of the project. That is when we decided to use commercial-grade pocket/sliding door hardware. As shown in the figure below, this hardware, which is intended for sliding doors, was used to move the tray in the horizontal direction. The tray will be mounted at the top of the rail and connected with a belt that will rotate with a pulley at the opposite end. In order for the tray to move left or right, a stepper motor will be located at the far-right side of the rail that will rotate a small belt to turn the tray.

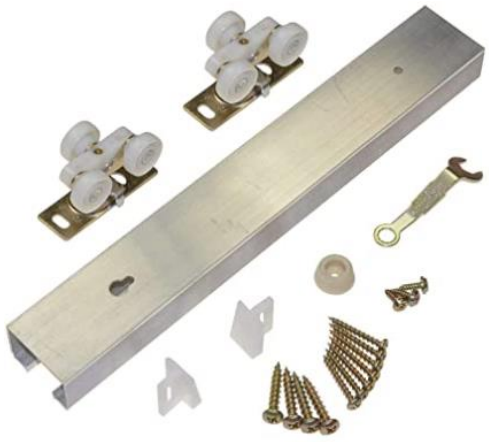


Figure 3: Commercial Grade Pocket/Sliding Door Hardware [27]

Vertical component: Much like the horizontal slider, the vertical tubes are also replaced with the sliding door hardware for which the horizontal slider will go up and down. However, unlike the horizontal slider, the vertical part of the tray will not be as critical because the books will fall onto the tray as opposed to the alignment of the horizontal part of the tray. Using a similar structure to the horizontal component, a stepper motor will be added at the top of the far-right side of the shelf and a pulley at the

bottom to rotate and move the platform on which the horizontal stepper is mounted on. This way the belt will move the horizontal components at the desired height for each cell.

A pulley will be attached to the far left of the horizontal railing which will allow the belt to be tied with one end to the tray and loop around the horizontal stepper motor and back to the tray. We tried using only one side of the shelf, the right side, where the stepper motor is located to carry the full weight of the horizontal slider. However, as we tested the actual weight of the horizontal components, it was clear that we needed to support the left side of the slider. Using a steel rod to connect the vertical stepper motor at the right top of the self to the left side of the shelf, we used another pulley at the top and bottom to help ease the weight and rotate at the same rate. This allowed the horizontal slider to move up and down without tilting to one side while the vertical stepper motor rotated the belts. In addition to this, we also added pulleys on the side of the shelf to help tighten the belt and keep the tension strong enough so the teeth of the pulley would not skip the belt. Figure 4 shows the front view of the shelf with the sliders with the mounted tray and stepper motors.

Figure 4: Front view of sliders with tray [26]

For stepper motors, we used the STEPPERONLINE Nema 23 Stepper Motor which will draw an approximate of 3.36V with a 3A per phase, consisting of 2 phases [28]. Since these stepper motors draw more current than a small 2A max H bridge can handle, a TB6600 4A 9-42V Stepper Motor Driver will be utilized to control these stepper motors and be used as current limiters [29]. We will be using PWM to control the rotation and speed of each stepper motor to match each designated cell location. This process will most likely be trial and error as the motors will have slightly different rotations. Using the Arduino platform, we used the SunFounder Mega 2560 control board and programmed it with several functions. Each function was tested to move a certain distance for the horizontal and vertical stepper motors. Since there are only four functions, up/down/left/right, we created these functions separately and set a rotational speed (delay in microseconds) for the stepper motors. From there, these functions are called in a switch statement, each indicating a number of rotations that the stepper motor will do. For each case in the switch statement, we used ASCII characters that will communicate with the Raspberry Pi via USB. This will allow the Arduino to know when to activate the sliders and to which cell when the Raspberry Pi calls for a specific cell on the shelf.

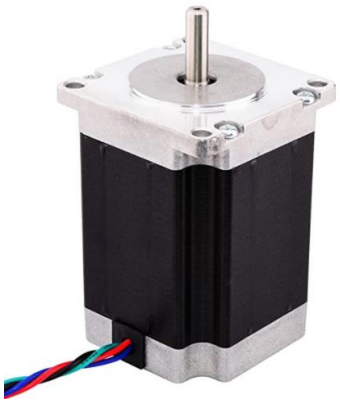


Figure 5:STEPPERONLINE Nema 23 Stepper Motor [28]

As mentioned earlier, the purpose of the TB6600 4A 9-42V Stepper Motor is to control the stepper motors like an H-bridge component. The only difference is that this H-bridge will have a much higher tolerance for Amps and serve as a current limiter. This will be wired to the PWM pin and the ends will control the stepper motor as desired. An L298N H-bridge could be a possible replacement for this since it will not have continuous rotations for the purpose of this design. Instead, it will rotate for a short period and reverse its direction to return to the default state (of the tray). However, in this case, it would be wise to avoid overheating the L298N and replace it with the TB6600 4A 9-42V Stepper Motor.

Figure 6: TB6600 4A 9-42V Stepper Motor [29]

Consumption of Power with most of the main components

As of now the total power that must be supplied to the components is around 227W to 240W. The main components that will consume the power are the servo motors, as there will be more of them than any other part. By adding up and multiplying the amount of voltage and current that will be drawn from each component, a power supply with a minimum of 350W must be used in order to keep the power consumption of 80% or less to keep the power supply in good working conditions. With this in mind, we searched for a power supply that would fit these needs and found the MENZO 360W power supply that will transform an input voltage of 110V AC from a power outlet and supply a 12V and up to 30A according to the max range as shown by the seller [30]. However, not all components that will need power will need the 12V supply. In order to get a lower voltage, we will be using a DC-DC step-down converter. This has not been selected yet as there may be components that require more than 2A and most of the commercial regulator's peak at about 2 to 3 Amps max until they begin dissipating excessive amounts of heat. In the end, we ended up using the power supply to give power to the two stepper motor divers and used the wall outlet for the remaining Raspberry Pi as they required 2.5A to 3A to operate at the recommended PSU current capacity.

Figure 7: MENZO 360W power supply [29]

A keynote to notice is the pins that will be used for PWM for each stepper motor and servo motor. In our case, we decided to use the Raspberry Pi interface to control the servo motors but the Raspberry Pi only contains 1 PWM GPIO pin output. The stepper motors will be handled by TB6600 but will not require a PWM pin output on the Arduino. Since we are using the TB6600, we can simply control the direction of the stepper motors using common ground and activating the direction of the motor using the digital I/O pins coming from the Mega 2560 control board.

3) Tray for Carrying the Book

The tray is where the book lands after being pushed from the cell. The tray is slightly tilted so the book slides right onto it and has walls around, so the books stay on the tray. The wall that is on the lower side of the tray will open acting as a door, controlled by the servo motor. It opens by tilting its top away from the tray and laying at the angle aligning with the checkout window. A servo motor is attached to the right side of the tray and has a metal latch connected to the lower wall of the tray. The servo motor will rotate clockwise and counterclockwise to open and close the door. Here is the diagram depicting two states of the lower wall.

Figure 8: Opening the tray with servo motor [31]

When the door is open the book slides out of the tray and onto the checkout window. Our first tray design consisted of being made from plywood, but after taking the weight into the consideration we decided to switch to plastic material. We have a servo motor connected to the right side of the tray, a metal hinge will be connected to the wood and wood the door. In this feature, we made use of the Continuous Rotation Servo Motor, attached to the right side of the tray. This motor is capable of 360 degrees clockwise and counterclockwise rotation with high torque of 11-kilogram per centimeter at 6 volts with pricing at 17 dollars and 95 cents [32]. Moreover, we utilized a plastic container that has a length of 12 inches, a width of 18 inches, a wall height of 4.3 inches, we modified it to convert one side of the container to a door [33]. We also added in the IR Beam break sensors. The tray feature communicates with the cells and the horizontal and vertical sliders for the confirmation of alignment and opening and closing of the door. Along with that it also communicated with the sensors for the status of the book.

4) Servo Motor Belts to Move the Book

After receiving the confirmation signal of tray alignment, this feature is activated. The servo will start rotating to move the book forward to slide onto the tray. To hold and move the books we made use of belts and book holders. The belts will be attached to the servo motor by a pulley. When the servo motor rotates, it moves the belts, hence pushing the book towards the tray. The servo is placed at the interior end of the cell on a shelf and the belts will be attached to it and the book holders are on the belt to keep the book standing. Each cell of the shelf is 11.25 inches high, 11.25inches wide, and 11.75 inches in depth which enables us to store a variety of different sizes of books [34]. There are two belts (6mm in width) attached to the motor and will run in parallel with help of pulleys, we chose to implement two belts so the books with different heights can be easily placed with the right balance. However, some books have a height higher than 9 inches causing imbalance and potential fall, to solve that problem, the book is placed horizontally. The results of moving the book toward the tray are achieved in both ways without any issue.

This approach allows us to attach the servo motor at the back of the shelf which leaves the whole 11.75 inches of space to place books. To implement this feature, we made use of the Continuous Rotation Servo Motor, one per cell. This motor is capable of 360 degrees clockwise and counterclockwise rotation with high torque of 33.8-kilogram per centimeter at 7.2 volts at each pricing at 25 dollars and 99 cents [35]. To operate and control this servo motor, we will utilize Raspberry pi 4 with Python's CircuitPython libraries like Adafuit CircuitPython PCA9685 and Adafruit CircuitPython ServoKit [36]. For belts, we will be using a GT2 timing belt and use a metal holder bought from a local store to hold the books when they are standing on the belts[37]. This feature communicates with the sliders and sensors located on the tray that sends book landing confirmation. If the book is not landed on the tray, the servo rotates again to ensure the drop of the book onto the tray.

5) Book Landing Confirmation

There are two instances where we need to confirm if the book has landed properly and the program can continue to the next part. First is when the book leaves the cell and lands on the tray. We need to confirm that the book has landed on the tray before proceeding to the checkout window. To confirm if the book has landed on the tray, we use IR Break Beam sensors that are attached to the left and right wall of the tray and will detect if there is any object in between them [38]. Once it detects that there is a book in between, the sensor, will send singles to the sliders to move the tray to the checkout window. However, if the book is not landed on the tray, it let the cell servos know to rotate once more time. We also need to confirm if the book has landed in the checkout window. The purpose of confirming if the book has landed in the checkout window is to trigger the program to finish the reservation and email receipt and return day to the user.

6) Physical Shelf

The shelf is a critical and most important part of the whole project! First, we were planning on creating our own shelf so we can have the exact dimensions we need including the height and width of each cell. However, after calculating the total price we came to realize that creating our own shelf will be far more expensive than purchasing a premade shelf. Like any other object with premade material, there are limitations we had to adjust to. For example, the height and width of each cell are a little bit off in numbers compared to what we were aiming

to build for ourselves. For example, our dimensions were to create each cell around 10 inches high and 7 inches wide, we were planning to keep the cell depth at 15 inches. On the other hand, the premade shelf's cell dimensions are 11.25 inches in height, 11.25 in width, and 11.75 in depth [34]. The height and width of the premade shelf's cell are a little longer which will require us to make some adjustments so the books will not slide off to the sides when moving forward in belt rotation. To fix the problem we have added pieces of cardboard that are attached to the two belts rotating to push the book forward while maintaining them upright. It also provides more support to work with books that are thicker in size and are harder to move. This premade shelf's total price is 41 dollars and 9 cents with a total height of 35.86 inches and a width of 35.86 inches. We also made some adjustments by adding in extra wood on the bottom, which raised our shelf by one foot and provided the vertical slider to access the lower cells easily.

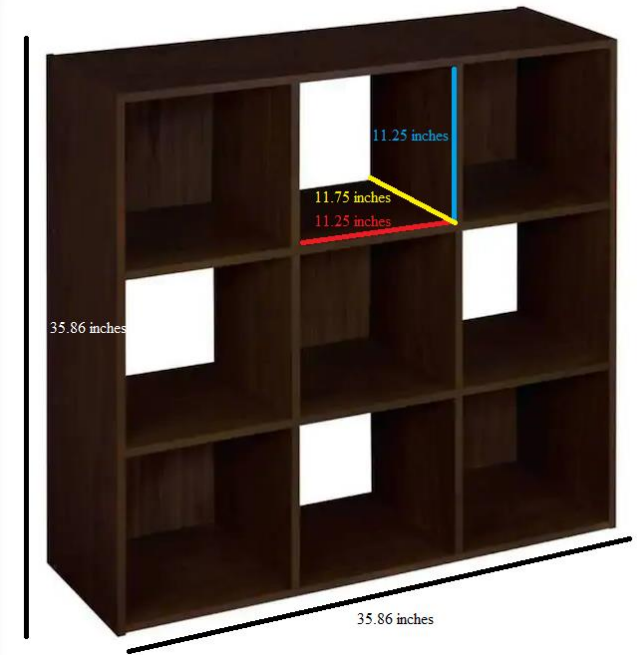


Figure 9: Physical Shelf [34]

IV. Funding

For this project, our team will pay all costs from our pockets. We will divide the cost of the project into four and pay evenly. Our assumption for the total cost of this project was between 1,000 to 1,200 dollars. However, this is estimate was done around the beginning of the fall semester and after further experimentations and material purchasing, it was just under what we had expected. In the table below are the parts utilized to implement the features of this project. From purchasing the shelf to buying miscellaneous material to construct the book machine, we used a big portion of the budget in the trial-and-error experimentations for the cells and the material used for the horizontal and vertical sliders. Some parts of the project had costs that were unavoidable, like all the servo motors and stepper motors as they were critical in this project. While there might have been some other servos and stepper motors out there to satisfy our design needs, we chose the ones shown in table 2. Please note tools used for programming and testing are not listed. Only the tools and parts that will become final features are listed.

Since this project was built by us, we had a ton of woodworking and several ideas that were brought up, attempted, and discarded. This exhausted both monetary and time value. Some additional parts were bought in order to implement some of these ideas but ended up not being used. For example, using a third stepper motor with its driver to help ease the weight of the vertical slider as it is the component that moves the tray and the horizontal railing up and down. This expense was later reduced when we came up with the idea to connect both sides of the vertical railings using a steel rod from the shaft of the vertical stepper motor to the opposite side of the shelf, onto a pulley.

Among many other materials bought to construct the horizontal and vertical sliders, the cells themselves required many utilities and testing for each book to successfully move up to the edge of the cell and drop onto the tray. With 6 cells in use, the amount of money spent per cell was more than anticipated, especially since the servo motors needed to be strong enough to push the books using 2 belts and pulleys.

Table 2: Funding [39]

Item	Source	Quantity	Price Per Unit ($)
Logitech C270	Logitech Web Store	1	39.99
Railing Tracks	Amazon	2	73.38
BEMONOC 5Meters 3M Open Ended Timing Belt Width 15mm Polyurethane Belt	Amazon	2	49.89
STEPPERONLINE Nema 23 Stepper Motor	Amazon	2	15.99
Nema 23 Stepper Motor Steel Mounting Bracket	Amazon	1	13.04
BIQU GT2 20Teeth 5mm Bore Aluminum Timing Belt Idler Pulley	Amazon	6	8.45
TB6600 4A 9-42V Stepper Motor Driver	Amazon	2	14.89
MENZO 360W Power Supply	Amazon	1	27.97
Raspberry Pi 4 Power Supply USB-C 5.1V 3A	Amazon	1	13.53
3x3 Wooden Shelf	Lowe’s	1	41.09
Servo Motors	Amazon	6	28.26
Raspberry Pi 4	Cana Kit	2	35.00
6mm Belts	Amazon	3	9.78

IR Break Beam Sensors	Amazon	2	10.00
18 Gauge Wire - 6 Colors of Tinned Copper Wires Silicone Rubber Insulated16ft / 5m	Amazon	2	15.21
Qibaok 250 PCS Heat Shrink Wire Connectors Electrical Terminals Kit	Amazon	1	17.99
15A 250V Rocker Switch	Amazon	1	8.71
120pc Dupont Wires	Amazon	2	13.61
Tools & various materials	Amazon/ various stores	-	414.78
Total Cost		**$1,268.3**	

V. Project Milestones

The project milestones are points throughout our work progress that represents a step in achieving the next big thing for the overall project. The following points summarize the achievement markers that provided a cornerstone for the basis of our project.

- Book testing for individual cell
- Slider calibration for alignment with tray and cell
- QR code integration to communicate cell and slider
- Website Functionality
- Website/Hardware integration

The importance of achieving these points can be best explained through the set of build features. Since the integration of hardware and software are big chunks of this project, we kept these major points as ideal achievements that require both time and effort for the full implementation of the overall project. From the beginning, we agreed to split the idea of the website and the physical prototype building of the project separately. However, as we had to keep the need to be combined in mind and not lose sight of it so it was only appropriate to consider it as a milestone of the entire project. Like any other project, the assignments were also of significance to include in our milestones. Section A lists the project's main features that will be included in our building process for the physical prototype and web interface. Section B lists the assignments that we have completed alongside our features.

A. *Project's M*ain Features

The first points will be the physical prototype milestones as follows:

- Picked, assembled, and modified the shelf
- Created a system for two belts to move the books in each cell
- Created/modified the railing system for the Vertical and Horizontal sliders
- Created a tray mechanism that allows the book to land from cell to slider.
- Book Landing Confirmation
- Checkout Window
- QR Code Scanning

These next few points list out the ideal features that will be implemented to design the web interface and project integration. They will be our main website features as follows:

- Website Structure
- Homepage (Dashboard)
- Book Profile
- User Accounts
- Search Bar
- Book Recommendations
- Book Checkout Cart
- QR Code Generator

Like most projects, we began by picking our societal problem. This was somewhat of a frustrating task as many of us had a very limited budget and lacked some skills needed to complete some ideas that came up. After thorough brainstorming and researching current issues in society, we decided to stick to the "libraries in a pandemic environment" and created a project that would alleviate the issues brought by this pandemic to library checkout of books.

As stated in the bullet points, the first milestone was to create a book checkout machine that will dispense the books to the user. Since we lacked the tools and the skills required to build a more professional machine, we decided to create something that would require parts that are relatively easy to obtain and disposable to us. This is when we created the two-belt system to move the books in each cell. While we tried other methods to approach this feature, the two-belt, one servo per cell system was one of the fastest and efficient ways we came up with to push the books out to the tray.

After that came the railing system for the vertical and horizontal sliders. This was one of the biggest obstacles to overcome as it required a lot of time to build the railings. After discarding ideas of using pipes and bearings to make this mechanism, we decided to build it using sliding door railing tracks. In which eventually we completed this task towards the end of the fall semester for the laboratory prototype evaluation. To keep this mechanism to sync with the cells, we utilized a tray mechanism using IR break beam sensors. This became another great milestone as we now had a functioning cell-slider-tray mechanism for book transportation.

Finally came the Checkout window with the book landing confirmation. These milestones provided the user with a front window in which they would pick up the book and make the program reset. Once this system was placed, the QR code scanning would reset and be ready for the next user.

The second portion of this project was the creation of our website, we initially thought of this phase to be faster and easier; however, we were wrong in this assumption. It took us a while to put together the structure of the whole website and get used to the new frameworks we chose to implement the website. Nevertheless, once the website structure was up, we were able to complete our website in no time. Some key factors that helped with that were setting up of XAMPP server, database, and Git repositories. It helped us improve the speed of our website creation. We started on the dashboard, user accounts, book profile, and checkout cart. We first started working on the backend to complete and integrate these features in the backend, but after some discussion, we decided to also start on the frontend as well to avoid any problem that might arise later. We did face many significant errors causing a slow down in our project, yet those errors could have been worst if the frontend was implemented later. Overall, our main features mentioned above were completed for the longest time and after that portion was completed rest of the website was faster to finished.

Creating these mechanisms took quite some time to create, calibrate, test, and modify. They provided a great deal of significant development as each assignment in the course would polish the design idea and improve each aspect of every feature to deliver the final deployable prototype evaluation at the end of the spring semester.

B. *Senior Design course assignments*

This project involved a series of assignments and documentations that were listed in the syllabus, thus marking importance in the completion of this project's milestone as follows:

- Individual Problem Statement
- Team Societal Problem
- Design Idea Contract
- Work Breakdown Structure
- Project Timeline
- Risk Assessment
- Project Technical Evaluation
- Laboratory Prototype presentation
- Revised Problem Statement
- Device Test Plan
- Market Review Report and Presentation
- Individual Feature Report and Presentation
- Testing Results Report
- Deployable Prototype Evaluation
- End of Project Documentation
- Public Deployable Prototype Presentation
- Weekly Activity Reports

VI. Work Breakdown Structure

A. Physical Product:

1) Shelf

Getting the self-done was the first task that needed to be completed since everything was going to be assembled on the shelf. After researching, we came across different types of shelves and bought the one we estimated to have the appropriate size and cells according to our project description. After assembling our shelf, we tested it for durability by putting the books on it and checking if it will move once we attach motors and other heavy things like rail tracks to it. The shelf we bought passed the durability test and was ready to have other features assembled on it.

2) Servo Motor Rotating Belts to Move the Book

This has been one of the most time-consuming tasks for our team so far. We had to consider a lot of variables while deciding the coils. First, we wanted to buy premade coils as they would be made with good precision. However, buying coils became impossible as most industrial coils are made in China and delivery would take over 3 months. Then we decided to make coils by ourselves. Purchasing the material was another major task for us. We had to look through different types of wires to see which would be able to hold books properly. After successfully purchasing the right material for the coils, we had to make the coils now. We used the pole of an old basketball hoop. Before testing the coils we created, we had to buy motors to test them. We researched two types of motors for the coils, servo, and stepper motors. After careful comparisons between the two, we decided to buy servo motors as they are easier to control with microcontrollers. However, since each coil that we created came in at different sizes and rotations, it made the book moving transition almost impossible to function properly. After that we decided to switch to a belt rotating system with cardboard alignments and purchasing the motors, now it was time to research for the microcontroller to control the motor to rotate the coils. We used Raspberry Pi to control the motors. Our feature is completed and tested, it meets our feature set requirements by holding different sizes and weights of books and moving them when triggered.

3) Vertical and Horizontal Sliders

Aside from the discharge mechanism with the servo motors rotating the belts, the second main component of the physical part of this project was the vertical and horizontal slider. The original idea was to build a bearing system that would slide between two poles on the horizontal and vertical axis. This idea would have required a lot more trial and error since it was essentially built from scratch and with our little experience in woodworking, it would be very time-consuming. The other deal with this is the number of bearings that would have been required to construct this method and would have cost a lot more.

In order to get over this problem, we decided to continue researching while this idea of the bearing system is still under consideration. After some possible considerations of other railing systems, we came across a rail and wheel system that is used for closet doors. This idea would not only help us in avoiding the construction of the bearings but also speed up the process of installing the steppers. With the new slider system in place, it was clear that some modifications had to be made for the wheel component in order to avoid slanted sliding of the horizontal slider. The stepper motors are connected with a module that regulates its power and feeds it a signal from the Arduino PWM pins. The driver is adjusted to fit the number of micro-steps per revolution before any programming is done. The physical placement of the steppers are at the top of the shelf to operate the vertical slider and another stepper is attached to the horizontal slider. The selection of belts was critical as the slider with the horizontal stepper is holding most of the tension when it is operational. A pulley is attached to the end of each slider to rotate the belt and move the horizontal slider and the tray on the horizontal slider.

4) Tray for Carrying the Book

To create the tray we went over different materials like plexiglass, wood, and prebuilt plastic tray. We confirmed that creating our own tray would be better as we need to have one of the walls open on the tray. While creating the shelf, we had some plywood left over. To stay within the budget, we used that plywood to create the tray. However, later we had to switch to a plastic container to create a second version to decrease the amount of weight on sliders.

5) Book Landing Confirmation

This feature was implemented to confirm if the book has landed on the tray or checkout window. We researched and confirmed that breakout sensors were the best to use in this case. The program confirms that the book has landed and has the system proceed to the next step.

6) Checkout Window

Like the book landing confirmation feature, this feature was used to confirm that the book has been picked up from the tray. The breakout sensors communicate to the Raspberry PI and send an email indicating that the book was checked out at a designated time and when it must be returned.

7) QR Code Scanning

This feature is responsible for scanning the QR codes and encode information to help locate the requested book. We had to research and find the best camera. We used Logitech C270 camera that provides a 720p live stream. After we configured the hardware, we started working to create a program that can scan and extract information from the QR code. The program is used the extracted information and matches it to our database. Soon after that, we assembled the camera above the checkout window. To test the camera and program, we generated different QR codes and scan them in different environments.

B. Website

1) Website Structure

Before we started implementing our features, we needed to design and configure our coding platform environment, server, and database. We needed to have a structured and global approach when working on features to maximize efficiency and minimize risks. For this part, we first began by researching and installing a web server application. We have some familiarity working with Apache distribution. XAMPP is an easy to install Apache distribution containing MariaDB, Java, and Perl. However, we will be using it with React and MysqlDB instead of Perl and MariaDB. After the installation is complete, we had the complete package of Java, MySQL, and a working server. We also built a small CRUD application that can test the server and identify any problems.

This app included the four basic requests of any application which are select, insert, update, and delete. It is very important to have this part done without any problems. Once this is done, we installed our frontend libraries and dependencies. This included React.js, node.js, and create-react-app. This gave us a stranded React web application. After that, we integrated our UI library to provide a smooth user experience. These kinds of libraries gave us front-end components ready to be used without the hassle of CSS designs. At the same time, we started working on the backend as well. We started building an API using Spring Java that is capable of handling all kinds of requests from the front-end. API implementation is a crucial part and will be time-consuming. Also, we needed to design a database. We started this section by designing an ER diagram. This allowed us to have a complete picture of our database. Once we had the final version of the ER diagram, we started creating the database tables and connecting them to our applications.

2) Dashboard

This is one of the main features of our website where users will have their first interaction. We started by research possible UI components and then work on the backend. Once we found and understood the UI components, we started working on the logical layer of the backend. Once we had frontend and backend done and tested, we integrated them.

3) Book Profile

This feature will allow the user to get a sneak peek into the book and will allow the administrator to keep track of the number of copies they have for each book. Having sub-features like a “Like” button helped us track and identify highly active books as well as provide detailed usage data on all activity on each book page. Once we finished all the sub-features logic in our backend, we moved forward with testing. The testing process ensured that the user has a smooth experience. When we were confident in our feature, we worked on the frontend and integrate both.

4) User Profile and Account Services

To fully customize the user experience, we needed to implement a user profile page. This allowed us to collect any kind of data about the user. The data we are looking to have are but not limited to, Emails, Names, and Phone numbers. The profile page is the center of the user activity. It contains order status, previous orders, and notifications. In addition to that, we collect usage data. These data could be anything from what the user clicks, likes, and engages with. Linking such data to user profiles will allow us to

build better predictions and suggestions models and provide a better experience.

5) Search Bar

To help our users find the books easily and efficiently, we implemented the search bar. Even though the search bar can be considered a small portion of the whole project, it is an important one. Without a search bar, users cannot really find books, especially if they are looking for the exact book that they need. The search bar has a feature of giving live suggestions as to the user searching for the book so the user can find the book easily and in less time. To implement this feature of the search bar we made use of the database and our React.

6) Book Recommendation Technique

To better improve user experience and to utilize library material we use book recommendations based on user data collected or entered by the user. This part will improve user experience because it will help users find books they are interested in or like. This feature works hand in hand with the search bar as well. It will be implemented alongside the side search bar feature.

7) Book Checkout Cart

This feature has two main functionalities, one to let the user check out books and another to reserve pickup time to get the book from the machine. This is a critical section that will be directly connected to our hardware. The users will need to be able to reserve the book and get the necessary information to check out the book.

Table 3: Work Breakdown Structure

Feature	**Tasks**	**Sub-Tasks**	**Estimated time to complete (Hours)**	**Assigned to**
Website				
Website Structure	A. Install & configure apache XAMPP	A1. Research	10 - 20 hours	Farooq
		A2. Install and configure XAMPP	10 - 20 hours	Farooq
		A3. Basics - PHP MySQL CRUD Application, and Springboot	10 - 20 hours	Farooq
	B. Install & configure React.js (frontend)	B1. Install React and its dependencies	5 - 10 hours	Farooq and Khalid
		B2. Configure basic React App	10 - 30 hours	Khalid
	C. Integrating frontend UI library with React (frontend)	C1. Research the library	5 - 10 hours	Khalid
		C2. Install the library and apply it to the frontend	10 - 30 hours	Khalid

	D. Building API (backend)	D1. Research possible implementations approaches	5 - 20 hours	Sajanpreet and Farooq
		D2. Building & connecting backend with frontend	10 - 30 hours	Sajanpreet
	E. Database	E1. Install and configure MySql	10 - 20 hours	Emmanuel and Sajanpreet
		E2. Database schema	15 - 20 hours	Emmanuel
Homepage (Dashboard)	A. Research possible UI components	A1. Research	5 hours - 10 hours	Farooq
	B. Implementation.	B1. work on frontend	10 - 15 hours	Farooq
		B2. work on backend	10 - 25 hours	Farooq
		B3. Integrating frontend and backend	5 - 10 hours	Farooq
Book Profile	A. Research possible UI components	A1. Research	5 - 10 hours	Khalid

	B. Implementation.	B1. work on frontend	10 - 15 hours	Khalid
		B2. work on the backend	10 - 20 hours	Khalid
		B3. Integrating frontend and backend	10 - 20 hours	Khalid
User Account	A. Research possible UI components	A1. Research	5 hours - 10 hours	Sajanpreet
	B.Implementation.	B1. work on frontend	10 - 20 hours	Sajanpreet
		B2. work on backend	20 - 30 hours	Sajanpreet
		B3. Integrating frontend and backend	10 - 20 hours	Sajanpreet
Search Bar	A. Research possible UI components	A1. Research	5 hours - 10 hours	Emmanuel

	B.Implementation.	B1. work on frontend	10 - 20 hours	Emmanuel
		B2. work on backend	20 - 30 hours	Emmanuel
		B3. Integrating frontend and backend	10 - 20 hours	Emmanuel
Book Recommendations	A. Research possible UI components	A1. Research	5 hours - 10 hours	Khalid
	B.Implementation.	B1. work on frontend	10 - 20 hours	Khalid
		B2. work on the backend	30 - 40 hours	Emmanuel
		B3. Integrating frontend and backend	10 - 20 hours	Emmanuel
Book Checkout Cart	A. Research possible UI components	A1. Research	5 hours - 8 hours	Sajanpreet

	B.Implementation.	B1. work on frontend	10 - 15 hours	Sajanpreet
		B2. work on the backend	15 - 25 hours	Farooq
		B3. Integrating frontend and backend	5 - 10 hours	Farooq
QR Code Generator	A. Research		4 – 5 hours	Sajanpreet
	B. Programming	B1. Program to retrieve Reservation information		
		B2. Program to generate and save QR Code in database		
	C. Test			
Physical Product:				
QR Code Scanning	A. Research and purchase camera		20 – 30 hours	Sajanpreet and Emmanuel

	B. Program to read QR codes	B1. Read QR code and extract information		
		B2. Match information with the database		
	C. Connect camera to a microcontroller	C1. Purchase required wires and cables		
	D. Test			
	E. Assemble camera above checkout window	E1. Research and purchase required material		
Vertical and Horizontal Sliders	A. Research	A1. Determine what type of railing system to use	100+ hours	Emmanuel And Khalid
	B. Assemble and measure sliders	B1. Determine belt/pulley placement		
	C. Stepper Motors	C1. Wire steppers to motor drivers/PSU		
		C2. Adjust stepper's physical placement on slider/shelf		

		C3. Code and test stepper motor's Vertical and Horizontal distance to each cell		
Tray for Carrying the Book	A. Research material for tray	A1. Discuss the different way to create a tray using different material	35 – 40 hours	Sajanpreet and Emmanuel
	B. Design and build tray	B1. Determine how the tray will be attached to the Horizontal slider		
	C. Add opening mechanism to tray	C1. Test if opening mechanism works		
Servo Motor Rotating Belts to Move the Book	A. Research	A1. Belt adjustment and alignment to cardboard	30+ hours	Khalid
		A2. Servos vs steppers, which one to use?		
		A3. Controlling servos, Raspberry Pi or Arduino		

	B. Purchasing material for testing	B1. Different material to test for making belt rotation with book load.		
		B2. Servos		
	C. Testing	C1. Belt testing: creation, functionality, errors in design, and alternatives if needed.		
		C2. Servos testing, how much can it push and pull forward.		
	D. Repurchase and test again if needed			
Book Landing Confirmation	A. Research for proper breakout sensors		40+ hours	Sajanpreet and Emmanuel
	B. Program sensors to detect if the book has been placed in the tray and checkout window	B1. Python program that process the signals from the sensor and confirms if book has been placed, ten send signal to proceed to next step		
	C. Connect sensors to microcontroller	C1. Research/purchase wires and cables		

	D. Test			
	E. Assemble sensors on tray and checkout window	E1. Research/purchase required material		
Checkout Window	A. User Email confirmation	A1. an email will be sent after the book has been picked up from the tray		
	B. Breakout sensors	B1. These sensors will communicate to RPI to send an email		
Physical Shelf	A. Research	A1. building shelf	5 - 10 hours	Khalid
		A2. purchasing prebuild		
	B. Purchasing and Assembling the Shelf	B1. Purchasing Shelf		
		B2. Assembling Shelf		
	C. Testing	C1. Testing for weight, movement, etc...		
Assignments:				

Individual Problem Statement	A. Research	A1. Research a societal problem and probable solution.		All
	B. Writing	B1. Writing a proposal of your Idea and probable design		
	C. Proposing	C1. Propose to Team and instructor in Team Instructor meeting		
Team Societal Problem	A. Choose & Research	A1. Choose a societal problem proposed by one of the team member or a variation of the societal problem proposed by a team member		All
		A2. Research		
	B. Writing	B1. Writing a proposal for TeamIdea and a probable solution.		
	C. Proposing	C1. Propose to Instructor in meeting		
Design Idea Contract	A. Research	A1. Probable solutions by each team member		All
		A2. Research the implementation of each solution, pros, and cons		

	B. Writing	B1. Writing a proposal of Team Design Idea and exact implementation.		
	C. Proposing	C1. Proposing to the whole class in-class meeting.		
Work Breakdown Structure	A. Assign tasks to each individual	A1. Determine individual strengths and weaknesses		All
	B. Determine flow of each task	B1. What must be done first, ie. physical building before the website		
	C.Chart	C1. Create a chart that describes all features and sub-features with each individual tasks		
	D. Narrative	D1.Describe how each task will be incorporated for each feature		
Senior Design Project Safety Assessment Form	A. Go through all the material, tools, machines that will be utilized and assess the probable hazards and PPE.	A1. Fill out the form		All
Project Timeline	A.Create a timeline of the whole project and when each feature will be completed.			All

Risk Assessment	A. List of possible safety risks	A1. Each task will have a list of risk regarding safety		All
	B. List of possible building risks	A1. Each task will contain parts, tools, and other hardware delay or malfunction		
Project Technical Evaluation	A. Working prototype	A1. Video recording and explanation		All
	B. Final Project report	B1. Specifics on the final version of the previous reports and ideas		
	C. Parts that worked and parts that need improvement			
Laboratory Prototype Presentation	A. Present to instructors and class, most likely via video, the functioning of the first-semester prototype.			All
	B. Video recording			
	C. Presentation of what's achieved and what needs to be done.			

VII. Risk Assessment

A. COVID-19

COVID-19 had significantly affected the progress of our project in so many ways. It was and continues to be a major risk to our project. In every single decision or alteration, we make to our design, we must account for COVID. At the start, we did not anticipate the possible issues. We failed to see the challenges we faced in advance. These challenges were the deciding factor in many of our discussions. For example, when we wanted to use our coils as our book-pushing mechanism, we did not think how hard it would be to create coils from scratch. And when we decide to buy a ready version, all the stores will be closed or out-of-business because of COVID-19. This is just one simple example of the routine issues we are still running into because of it. Stores around the city had decreased their hours of operations due to the low demand and strict government guidelines. Thus, we failed to find our desired hardware and had to settle with whatever we found. This led to more alterations to our original design due to the limited options.

Online stores are not any better. Most of the online shipping gets delayed. The estimated delivery time has doubled since September. We had to minimize that risk by investing in Amazon Prime Account to take advantage of its delivery time. However, this helped to an extent because most of our needed hardware parts are manufactured and shipped from Asia. Again, due to COVID-19, we always must account for delays because of the sudden change in international laws.

Also, during our meetings in the summer, we thought about how we can use the university resources to help us with the needed tools. Due to COVID-19, utilizing university resources was not an option. This was a huge setback to our progress because of the usefulness of such resources.

The inability to utilize the university rooms and labs as a working area affected our progress as well. We had to commit to a one-hour drive to Stockton multiple days a week to be able to have a large working area. This was a big risk for us since we were wasting almost two hours every day just commuting. This was a high risk since many problems such as traffic, accidents, and car issues could occur during our commute. And if any of this occurred, it will affect our arrival time. Therefore, it will affect our progress. Later we decided to move the working area to Sacramento. We decided one person driving such a distance is less risky than three people. However, during the move, we had to disassemble some of our hardware so it can fit in a normal-sized car. This risk we were willing to take to minimize the potential risks that could arise in the future.

COVID-19 with no doubt had a huge impact on our project. Some issues would have been easily solved if we were working in our school. The reality is different. We need to spend hours driving to open stores to find a small piece of hardware.

Despite all that, we were able to foresee any potential issues and solve them before they become real issues. For example, we learned to order in advance different pieces of hardware for different implementations and return the unused ones. Since the return is always free with Amazon Prime, we were able to avoid some obstacles like late delivery, faulty hardware, and missing parts within a delivery by ordering more than what we need.

Table 4: Risk Assessment Matrix [40]

5			Railing Tracks		
4			Integration		Web Interface
3	Moving Books				
2					
1			Other Mechanical Features		
	1	2	3	4	5

B. Railing Tracks

The railing tracks, or sliders, were originally planned to be made out of two pipes that would extend from the bottom of the shelve and across the shelve in order to provide the “rails” for the system. The problem with this idea was that it would involve more cost and claim more time than what we had planned. For one, it had to contain at least eight

bearings for the horizontal slider and another eight bearings for the vertical slider. This idea seemed like a good plan but we later found out that this would involve a lot more wood crafting and better design trial-error time in order to maximize the speed and reduce the friction caused by the horizontal-to-vertical joint attachment. After the start of our project, we noticed these red flags and began to plan out a different method to come up with a new railing system that would allow us to move horizontal and vertical components individually. Sajanpreet browsed ideas and came up with a new way to implement this new system: a closet door railing. This new method allowed a big lean in time consumption at the cost of minor adjustments to the rail itself.

In attempts to make the sliders work, we ran into an issue where the distance of the horizontal rail was obstructing the path of the tray alignment. Since we did not have the tools to properly cut out materials like wood and aluminum, it delayed the overall progress of our project. The main concern was to put together the tray and sliders to get coding and calibration testing on the way. A way to get over this issue was to buy handheld tools, like a saw and hacksaw, that would allow us to manually cut the desired parts. Although this is still very time-consuming, it was the only way to continue this project. The lack of power tools really impacted this part of the project and many other sections as well.

One last issue arose when the belt picked to make the rails horizontally and vertically move was too thin. The delivery of items and parts from vendors online to our homes also impacted the time frame in which we would be allowed to work on the project. With such hindering progress on the physical aspects, we were forced to stall in the time for the parts to arrive. It is estimated that the time spent stalling for the railing system alone was about one week and a half. It can also be said that the current pandemic caused the mailing delivery system to be overcrowded with demands, causing a major delay in our parts.

C. Web Interface

The risks associated with designing the website lays largely in our chosen design methodology. Since we selected the Model-View-Controller (MVC) as our design structure, there are several risks that we identify as possible issues during the implementation phase. One of those possible issues is how we integrate our backend and frontend code. Since the MVC allows us to work independently on different pieces of the code at the same time, we must research and clearly define the output-input for both sides. Test-case scripts must be coded for both teams to use so we know for sure what are the parameters for every end.

Also, the MVC framework best works with experienced developers. There are strong advantages to using such an approach. It makes life easy. Projects are simpler to plan and build. However, a lot of the code that we are going to write could be hard to label as a view or a model, a controller layer. For some of it, we could place it in the controller since it is the brain of code. However, doing so will defeat the purpose of having different components. This where experienced designers could make a difference. Anyhow, we are determined to make the best of this aspect.

Another risk is using external libraries for designing our UI. This indeed could be a plus. However, using such libraries could add the pressure of learning the syntax of a new library. According to our research, the time curve for learning new libraries differ based on the complexity of the component itself. Some components are greatly documented and thoroughly simplified by demonstrating life examples. This is not the case for all our libraries. Some of the libraries selected leak adequate documentations. Instead, they have good community support. Not like documentation but it should help with most of our debugging issues.

D. Integration

In the integration of all individual features, there will be many issues rising, some issues we can already see coming up are communication problems between different devices, the flow of the process, and errors.

1) Communication Problems Between Different Devices:

We are using different servo motors, sensors, stepper motors, cameras, raspberry pie, and Arduino, getting them all to communicate and work for hand in hand will be a bit tricky. As we made most of the features individually we will have to integrate them with each other and that will require both software and hardware integration and communication. This process requires a long time.

2) The flow of Process:

For this machine to correctly work each feature and each part needs to work in the correct flow. Meaning once the QR code is scanned and decoded, it will relay the message to find the location (cell number) of the book. Once the cell number is found we will have the vertical and horizontal sliders to move the tray to the right cell location and send confirmation of tray alignment. Next, the servo motor will activate after receiving confirmation of tray alignment and drop a book onto a tray. A sensor on the tray will send confirmation of the book landing on the tray and sliders will take the tray to the checkout window. In the checkout window, once tray alignment is confirmed, the door will open, and the book will slide toward the user. Once the book is picked up it will send a confirmation. All this requires the right flow, one thing is off, and nothing will work. We will for sure have a lot of problems occurring and we think that will take at least 4 to 5 days to put everything in place.

3) Errors:

As we merge codes and different inputs and outputs, we will for sure face big chunks of errors. To resolve those errors especially in the programming part will take a long time and repetition of testing. After thorough testing and debugging, we came to stabilize the common errors that would arise after running multiple merged codes and integrating them together.

E. Mechanism to Move Books

This feature is one of the milestones for us. While we were planning our feature sets, we carefully analyzed each feature and determined its importance in terms of other features relying on it. Moving the books out of the cell and into the tray is a crucial feature for our project. Once we have the mechanism to move the book out of the cells into the tray, only then our project will progress forward. If this feature is not working then, it will be very difficult to mark our project as completed.

Once the tray reaches the cell and is ready to take the book. It will send signals to that cell and the cell will move the book into the tray. Then our project will continue delivering the book to the user through the checkout box. If there is no book moved out from the tray then everything else starts to become less important. This feature needs to be finished to have most of the functionalities of this project working. If this feature is not completed, then our project kind of loses its purpose. So not having this feature is not an option for us. The work required to complete this feature is a little less than the other features, therefore we gave it a low ranking in the risk assessment matrix.

When we were discussing this feature and its implementation, we knew that we are not the best at mechanical work and therefore thought of different approaches to complete this feature. We also created a backup plan and set boundaries for ourselves, so we know when to move on to the next approach. We first started with our easiest approach which was to use coils. It required only two parts, motor, and coils. However, we had many obstacles with this approach. First, we could not find any coils to buy online. They were either out of stock or the delivery date was too late due to this pandemic. So, we decided to make our coils by ourselves. Since we were not good at mechanical work, this approach did not turn out to be good. We set a limit for us that if it failed 40% of our tests then we would stop that approach and start gearing towards the next approach. Once we realized that creating coils by ourselves is failing, we decided to work towards the next approach which is to use belts and motors to move the book out of the cell. After realizing the stopping point for our first approach, we were able to execute our backup plan and quickly start on the next approach. We scheduled an in-person meeting to discuss the next approach and its implementation. Once everyone was on the same page, we decided to order the parts and started to work on them. As we are progressing with this approach, we are gaining confidence that this approach will work. However, if something does come up, we have already discussed other possible approaches and will execute our backup plan again if needed.

F. Other Mechanical & Electrical Features

1) Tray:

The tray can have few potential risks involved with it that might delay or interrupt the delivery of the book to the user. The two major risks are malfunctioning the door and arm that controls the door and the book not sliding. These risks are crucial, but we believe we have solutions to them.

a) Door and Arm:

The door in front of the tray is the key to delivering the book to the user, opening, and closing at the right moments is the key. To open and close this door we have a mechanism of the arm that is moved by a servo motor. When the servo motor is at an angle at 0 degrees (based on unit Circle) the door will be closed, when the servo motor is activated, it will move the angle to 180 degrees (pi on the Unit circle) it will open the door. This process is pretty good and strong; however, the concern is that if the book is heavy and it falls onto the tray it might push the door. So far, we have not tested for how much weight can cause this instance, but we believe this can occur. One fail-safe we are looking into implementing is the use of a magnet to close the door tightly so when the book suddenly hits the door the arm has some extra support.

2) Book Sliding:

Another issue that can occur with the tray is the book not sliding downward to move to the checkout window or box. Since our tray's base is made of wood, the book will slide on it, but it has some friction. We have tried to smooth it with sandpaper, but the friction is still there. There are a few ways we are hoping to solve this issue, one way is to use plexiglass on the tray and door. Next, one is to use thin aluminum or stainless-steel sheets on the door and tray base. In both cases, we must add something onto the tray to ensure a smooth slide from tray to user checkout box.

3) Checkout box:

Checkout box or window has two main issues, one is a problem with QR code not getting scanned, and the second issue is the book getting stuck when sliding to the user.

4) QR Code Scanning:

When our camera is scanning a QR code, there might be some problems that might occur. For Example, there is too much light, and the QR code or part of QR code is not visible, it's too dark and QR code is not visible clearly, QR Code is held too far or too close, and scanning different sizes of QR codes.

5) Book getting Stuck:

After the book slides from the tray toward the user, the book can turn into an angle and can get stuck. To solve this, we will have to make the spacing wider so even if the book turns into a sloped angle, it can easily pass over to the user.

6) Electrical Components and Wiring:

The parts ordered and tested were only done so in a manner that was adequate for their individual section of the project. Once each part was received through the mail, it was tested for defaults and other technical qualities such as integration capabilities. In this sense, it would be clear what would need returning and what would need modification. One issue with this is the fact that components ordered online may not match specific product information once received. This was the issue with the Nema 23 stepper motors. On the product card that comes with the motors, it was specified with cable labeling that would indicate which coil would correspond to what wires. This problem caused a significant time frame to solve as it was not giving the desired results when tested. Another issue with the stepper motors was the driver's current capacity. Since the motors require more current than specified (recommended), we had to order a new digital stepper driver to supply both motors with enough current to rotate.

The integration of multiple sensors, motors, camera, and microcontrollers into a single power supply can become a hazard if the wires are not properly connected or the power supply exceeds the desired temperature specified by the manufacturer. Since some components in our project will be moving parts, the wiring will play a big role in determining how each part/component will affect the rest of the components. In the event of a short circuit accident, the component(s) may take permanent damage and potential harm may be caused by the improper use of wiring connections. One way to avoid this was to emphasize the protection and use of proper wire terminals at each component, especially the power supply.

VIII. Design Philosophy

As we began thinking of societal problems of living during a long-duration natural event that impacts at least 20 million people, we came to a major consensus that it would be related to this pandemic. Much of the public places like restaurants, movie theaters, shopping centers, etc. closed or were limited, we decided to focus on the aspect of library status. Since libraries are public spaces for individuals to join and learn by reading books or using them as a study place, we wanted to address the idea of checking out physical books. The aim is to create a design that will keep people from interacting with librarians and maintaining social distancing guidelines by creating a book machine that will have online website features to order books and checkout scan them at the machine. This section explores how we came up with the ideas and implementations that will be used to approach our solution.

A. Brainstorming Societal Aspects

The idea of designing a solution to the societal problem was more than just hammering a nail. We needed to learn how and where the solution to a pandemic problem would be influential to millions of individuals. At the beginning of our brainstorming, we needed to explore ideas and concepts that would enable users to have some type of access to books. Originally, we were thinking of using an autonomous robot to bring books from shelves to the user, but this idea was soon discarded as it will require a good amount of image processing and robotics fundamentals to build. As we kept researching for probable solutions, we used a vending machine as an inspiration. This idea was ideal for our design as a vending machine does not require an active individual, like a librarian (except for restocking). With this in mind, we looked at already existing vending machines and decided to use some of the features that preexisted in those machines and implemented our own.

B. Improvement of Design Aspect

Once we decided to use the vending machine concept, we needed to make modifications to meet the societal problem needs. For the physical machine, the idea of dropping the books from the cell to the user pick-up place seemed like the books were mistreated. So, to change this aspect, we decided to use a rail system that will enable the book to be safely transported from each cell to the user window without damaging the book. A vertical and horizontal rail, which we originally thought of making it out of aluminum tubes but changed to sliding door hardware tracks to transport a tray using pulleys, stepper motors, and belts. These will be the main component for the transportation of the books. The books will be placed inside each cell of a shelve in which two servo motors will spin two belts to push the book outwards towards the tray. Since the start of this pandemic, the problem of sanitation in public places has been of great importance. This brought the idea of implementing a website that will allow users to order their book online and use a QR scanning code to check out the book through the machine.

1) Book Cells

Before the fall semester began, we decided to try and use coils that will rotate using servos. This would allow the books to sit in between each coil winding and as the coil spun, it would move the book forward. However, as we began experimenting with different coil materials and making the coils, it turns out that this idea was more chaotic than we had initially anticipated. For one, the coils that were created by us had uneven and not strong enough to hold multiple books at once. Two, they did not rotate at the same speed because the wires had different windings, leading the servo motors to struggle to push the books and breaking them. After multiple attempts at using different AWG sizes, we deemed this method to be a failure and moved on to brainstorm another idea. This is when we thought of using a two-belt system that would move the books forward with separators attached to these belts. This would also only require one servo per cell, as the two belts were connected via a steel rod or wood rod to rotate both belts at once and move the book forward towards the drop-off area.

2) Design of Sliders

After further inspection, our initial idea of creating a system of pipes and bearings for the vertical and horizontal sliders, we decided to change the materials to be used for this feature. Since this project is a prototype, it seemed very unnecessary to create this system from scratch. If time had been permitted, we could have used 3D printed parts and use them to our advantage. However, this option was not on the table as we had to work from home during this pandemic. Instead, we used sliding door rail tracks and mounted them on each side of the shelf for the vertical motion,

and mounted another rail across these two, creating the vertical and horizontal slider. This put less work and less time into this specific feature and made the calibrations for the positioning of the tray much more feasible for our convenience. To attach each railing, we utilized pieces of wood and drilled on onto the aluminum tracks.

3) Checkout Window

For the checkout window, we used a wood piece to hold the carved-out area of the polycarbonate hallow sheet with the checkout box in order for the book to slide at a slope for the user to pick up the book after the tray has successfully dropped it. It also reinforced the checkout box to stay at that angle, as sometimes the book would make the slope decrease and prevent further books from sliding all the way. Since the system required an indication of when the book had been dropped off at the checkout window, we used IR break beam sensors to alert it. This ensured that when a user had picked up the book, the beam would break and thus, automatically allow the program to restart for another user to scan their code.

4) Using another Raspberry Pi

After testing the IR break beam sensors, we needed to solve an issue that was making the program have a delay in reading the sensor and reporting back as a 1 or a 0. After further inspection, we noticed that the raspberry pi could not handle both the sensors at the tray and checkout window without delays. These delays would cause the book to drop on the tray but never activate the Arduino to move the sliders back to the checkout window. How did we solve this? Using another raspberry pie connected to the main raspberry pi via TCP communication. This ensured that the GPIO pins on the main raspberry pi would not be occupied reading the sensor while executing other parts of the project, like activating the Arduino for the sliders, with ease.

5) Creating a tray

The creation of the tray had to have three things: weight, size, and sturdiness. We constructed a tray made of plywood, but it was too heavy for the vertical stepper motor to bring up and down that it would make the belt skip teeth. After examining other materials, we used a plastic container that would be mounted on top of sloped plywood that was attached to the horizontal slider. The tray was cut so that it would have the least friction and allow the book to slide off. We needed something to keep the book in the angled tray while it moved from cell to checkout window. By attaching a servo motor to the side of the tray and using pieces of wood to make an “arm” that would extend to the door of the tray, we created a system that would allow the book to safely stay in position until drop off was needed. The servo would rotate and open the tray and close after a period of time with a reverse rotation.

All of these can be referred to appendix D for visual aids of these mechanism.

IX. Deployable Prototype Status

A. Software Testing:

This section explores our testing millstone plan regarding the software aspect of the project. To ensure the best performance from our software application, we created the following milestones as our guide to determine whether a feature is ready for the production stage or not.

1) Compatibility Testing

The objective of the Compatibility Testing is to assess just how much the software performs on a specific browser, operating systems, devices. We need to make sure that our application is compatible with the most popular Browsers, Operating Systems, and devices such as Smartphones, tablets, or Personal computers. We need to know exactly under which circumstances our application will fail. These testing cases should be exact version numbers for browsers, devices, types of browsers, or Operating systems.

2) Usability Testing:

This section will assist each individual component (Homepage, Book Profile, User Account, Search Bar, Book Recommendations, Book Checkout Cart, QR Code Generator) for its Easy-User or simply as the web developer call it User-Friendly Components. Basically, system navigation is checked in Usability testing. Also, performing these kinds of tests will help us save time because we will be measuring our code using such tests before moving away from it. Some of these tests are as follows: grammar-free/spelling-free contents, texts/components are aligned, tooltips to help navigate, buttons should be in a standard format and size, Confirmation messages upon action, HTTP requests should not take long, Reasonable loading time, 404 error pages for incorrect URLs.

3) Functional Testing*:*

These kinds of tests are established to make sure that each individual component (Homepage, Book Profile, User Account, Search Bar, Book Recommendations, Book Checkout Cart, QR Code Generator) will perform as expected. For example, when a user completes the checkout process, we need to make sure that our backend would respond accordingly. Also, the Homepage should display all the books that are currently available. The same with the rest of the components.

4) Validation Testing:

These tests are to ensure the user inputs all the required fields and to control the user input. For example, the amount values should display in currency format, all input fields for special characters, Sorting, error handlers, exceptions, loss of connections, recovery options.

5) Database Testing:

This is essential in every application. We need to make sure that the data that is sent by the user is being handled properly and being inserted into our database. This will require a waiting-signaling between the frontend and backend to make sure the user that the process is completed and his or her actions are maintained. The tester should understand the functional requirements, business logic, application flow, and database design thoroughly. Verify the Tables, columns, column types, and defaults are alleged with our established components, encryption of the data, Security, implementation of flags, and exceptions.

6) Integration testing:

This is the process in which all the different pieces of code or units are merged and checked as a whole. This determines whether the code is compiled in compliance with the appropriate requirements.

To successfully test our software application, we will be using the following Criteria to establish a fail/pass component. Each criterion must be checked to call it a pass:

Table 5: Fail/Pass Criteria [41]

Criteria	**Check**
Browser Compatibility - PC	X
Browser Compatibility - Tablet	X
Browser Compatibility - Smartphone	X
OS - windows	X
OS - Linux	X
OS - Mac	X
Functional/Usability - Homepage	X
Functional/Usability - Book Profile	X

Functional/Usability - User Account	X
Functional/Usability - Search Bar	X
Functional/Usability - Book Recommendations	X
Functional/Usability - Book Checkout Cart	X
Functional/Usability - QR Code Gen & Email Notifications	X
Functional/Reasonable loading time & Request time	X
Functional/Validation - Registration Form	X
Functional/Validation - Login Form	X
Functional/Validation - Comments	X
Validation - Checkout Cart	X
Validation - ALL errors are handled throw exception (no strnage error)	X
Functional/Validation/Correctness Database	X
ALL components must work together to complete a user order	X

B. Hardware Testing:

This section will explore our testing millstone plan regarding the hardware aspect of the project.

1) Book testing for individual cell:

To test if the cell is fully completed and is working properly based on our expectation we will be testing for the following criteria. Book size, book weight, speed of belt rotation when the different weight of books is applied, and finally stability of the books. Based on our cell size of 11.25 inches high and 11.25 inches wide, we can assume that the maximum size of our book can be 11.20 inches X 11.20 inches; however, the minimum size of the book depends on the distance between the two belts. To test different weights of the book we will test from 2 – 6 pounds of each book, we will also test the total weight that can be pulled by placing multiple books, our estimate is a total of 35 pounds. To test how fast the belt will be rotating, we will have to test out the different speeds of rotation with different total weights of each cell and come up with the average speeds that can be used across. When we talk about speed and weight, we also must take stability into account by testing if a certain weight and height of a book can cause when speed is increased.

Table 6: Individual Cell Book Testing

Test	**Trials require**	**Trials required to pass**	**Criteria for pass/fail**
Book Size	10	8	- 5 different sizes between the Min and Max sizes - Minimum Size 5.6 inches X 5.6 inches - Maximum 11.20 inches X 11.20 inches - Books must not tip over, must not get stuck, and land perfectly on the tray
Book Weight	10	8	- 5 different weight sizes of books, books must be at most 6 pounds. - The total weight of one should not be more than 35 pounds - Books must not tip over, must not get stuck, must land perfectly on the tray, servo motor and belt must not break.

2) Slider calibration for alignment with tray and cell:

One of the most important factors to the hardware part of this project is the sliders and the calibration it needs for each cell. Since the stepper motors are controlled by the motor drivers hooked to the Arduino, a test must be set in order to ensure that the specified rotations are met for each cell. One option is to check to see if each cell has about the same distance from each other and mark the number of rotations the steppers must rotate. This may pose a threat later on as the machine will be operational for as long as orders keep coming, thus causing alignment miscalibration. If this is the case, further steps will be required to eliminate this process of miscalibration. The Arduino is programmed to move in 6 ways, in different directions. The number of rotations currently set for horizontal movement is 13,3000 steps for a single left or right. 14,500 steps for a single up or down and lastly 3,300 steps for the initial default up/down position of the tray. The motor drivers are set to 16th steps per revolution. A miscalibration can occur when a certain number of steps is lost during multiple uses of the machine or the slider itself skipped or slid slower than usual. This can and will cause the tray to slowly misalign with the cell during operation. If this is the case, once tested, an additional set of sensors will be used to measure the distance between the floor (for vertical alignment) and the horizontal distance for each cell. After many trials, the end result is to have a consistent mechanism that will default to its original position and operate as intended for each cell.

Table 7: Slider Calibration Testing

Test	**Trials**	**Trials required to pass**	**Criteria for pass/fail**
Slider default position	15	15	- After issuing a book, sliders must return to the original starting point - Avoids skips, leaps, or slows that can miscalibration
			the return position
Tray Alignment	15	15	- Tray must align with each cell when the book is rotating forward on to the tray - Tray must default back to a desirable height for the next book

3) QR code integration to communicate cell and slider:

There will be different sub-tests that will compose the testing of this milestone. We first have to test if the camera can scan QR codes in different environments. Since we are making this prototype as close to real-world use. The camera will be used under different conditions like in very bright light in the afternoon or in very low light at night. To test this, we will have to scan the QR code in different light conditions. The second test will be to check if the program could efficiently decode the QR code and pass the information to the next program. For this test, we will have to continuously scan the QR codes and let the program decode them. If the program gets noticeably slow over time and is taking a lot of time to decode and process the QR code information, then we can consider the test failed and will have to make fixes to cut down the decode and processing time. Our goal is to quickly decode the QR code and pass the information to the next program so the sliders can navigate to the target cell.

Table 8: QR code Testing

Test	**Trials**	**Trials to pass**	**Criteria for pass/fail**
Scanning QR Codes	8	8	Scan QR code in very low-light conditions Scan QR code in very bright-light conditions

			Scan with low screen brightness Scan with high screen brightness Each test will be conducted twice
Decode and send QR code information	100	90	Scan 100 different QR codes continuously At least 90 scans should be decoded and information should be passed to the next program Each scan should not take longer than one minute to decode and send information

4) Website/Hardware integration:

Once the physical device or hardware is all set up and the website is set up then comes to the hardest job of integrating both. Once integrated we will be testing many variants and problems, but the most important test that we must run for communication between physical device and website. Communication is done on a couple of occasions between both devices and on every occasion, it is critical for communication to succeed. First, we must test to make sure both the physical device and website have access to the same data, so when the website generates the QR code with an embedded book title machine can decode and deliver the book. Second, we must relay back to the website that the book is picked up. To test these conditions, we must run the final integration process multiple times, with conditions such as, user ordering book and picking it up right away, user picking up the book next day, user order the book but never goes to pick it up, and lastly student scan QR code to pick up the book but for some reason leaves the book. We must test to make sure that when such scenarios happen our machine can respond appropriately.

Table 9: Website and Hardware Integration Testing

Test	**Trials**	**Trials to pass**	**Criteria for pass/fail**
Same data access	20	15	Must be able to access and update the status of the book (available, checked out, etc..)
Stable communication	20	20	Check out the book(s) online and pick it up at the machine at a given time Marked unavailable on the website after the user checks it out, once picked up marked picked up and status is updated. Should test for a book not picked up Should test for the book left at the pickup tray after scanning QR code

Table 10: Prototype Testing Timeline

Test	**Assigned**	**Result**	**Expected Output**

Book testing for individual cell	Khalid Shah	To be tested	The book is successfully moved forward with a 360 degrees servo motor and belt and landing on the tray.
Slider calibration for alignment with tray and cell	Emmanuel Silva	To be tested	Horizontal and Vertical alignment with each cell is met. The tray is expected to default back to its default position to await the next order after repetitive books have been picked up.
QR code integration to communicate with cell and slider	Sajanpreet Malhi	To be tested	The camera successfully scans and decodes the QR code information, passes it to the next program and the sliders navigate the try to the appropriate cell
Website Functionality	Farooq Alauld din	To be tested	All software features are functioning as expected and according to the following minimum requirements: Integration, Database, Usability, Compatibility, Function, Validation. Please refer to the Software Testing for more details.
Website/Hardware integration	Team	To be tested	The website will successfully generate a QR code, the user will scan the QR code, the machine will decode and deliver the book to the user, and update the database for the website and machine.

C. Testing Results

The testing results all passed the fail/pass criteria. After integration, we tested for multiple overall system runs and found a few errors and issues that were solved on the spot. In terms of Hardware testing, there were no problems passing the test criteria mentioned before with a few adjustments made along the way. Below are the testing results we obtained from our testing.

Table 11:Individual cell book testing results

Test	**Trails**	**Passing Rate**	**Criteria Tested**
Book Size	5	100% (5/5)	• 7.5 x 7.5 in, 7 x 10 in, 10 x 8 in, 6 x 9 in , 8.5 x 11 in • Tested 5 different sizes mentioned above. None of the sizes tipped over or got stuck, and all the sizes landed on the tray

Book Weight	5	100% (5/5)	• 1 lb, 1.5 lb, 2 lb, 3.5 lb, 5 lb • Tested these 5 different weights. None of the books tipped over or got stuck, and all books landed perfectly on the tray. • Servo motors and belts were able to handle all these different weights.

Table 12: Slider calibration testing results

Test	**Trails**	**Passing Rate**	**Criteria Tested**
Slider default position	15	93.33% (14/15)	• After issuing a book, sliders must return to the original starting point. • Avoids skips, leaps, or slows that can miscalibrate the return position. • Out of the 15 runs, 14 resulted back in the original default position and one came short due to skipping
Tray Alignment	15	86.67% (13/15)	• The tray must align with each cell when the book is rotating forward onto the tray. • The tray must default back to a desirable height for the next book. • Avoids skips, leaps, or slows that can miscalibrate the return position. • Since slider default position was misaligned, tray alignment was also affected with misalignment of $\pm$ 1cm

Table 13: QR code testing results

Test	Trails	Passing Rate	Criteria Tested
Scanning QR Codes	8	100% (8/8)	• Scanned 2 QR codes in low-light condition. • Scanned 2 QR codes in very bright-light conditions. • Scanned 2QR codes with low screen brightness • Scanned 2 QR codes with high screen brightness
Decode and send QR code information	100	97% (97/100)	• Out of 100 scans, all QR codes were decodes successfully and information was passed to the next program • There were only 3 QR code scans where it took slightly longer than a minute (1:02, 1:08, 1:13) to decode and send the information to the next program.

X. MARKETABILITY FORECAST

1) Current Market Outlook

Automating customer service is on the rise. Many companies are developing automated robots to provide excellent service to their customers. As automated customer service was growing with improvements in technology, the Covid-19 pandemic has fueled this growth. Every sector is now moving forward towards automated customer service and is planning to make even bigger investments towards it. Due to this pandemic, having automated services sound more like a necessity than a luxury. According to an article from Forbes, it is predicted that the use of automated customer service will increase by 143% in the coming years [42]. Many private and public sectors are predicted to move towards automation. However, libraries seem to lack behind in this advancement. If there have been any steps taken towards automation by the libraries. It was only due to this pandemic and still, they are very far behind in advancing their system, especially their book checkout services that got hit hard during the pandemic.

As we stated in our societal problem statement, libraries across the world had to close due to the covid-19 pandemic which heavily impacted the libraries and their customers especially with the end to their book checkout system. Library services were amongst the many that were hit hard and part of the reason was that they had no investments or solution towards automation. When the stay-at-home orders were given almost every human-provided service closed. Those who had investments in automation were able to continue back their services fairly quickly. Since libraries did not make much investments in automation in their system, they were not able to continue their services. This project will help the libraries to move towards automation, especially with their book checkout systems. Our machine and interactive website will allow users to find and check out books from the library, all while maintaining social distancing and avoiding face-to-face interaction.

Our Autonomous Book Solution will allow libraries to provide autonomous 24/7 book check-out services. This will help libraries to take a step towards automation and improve their services. Investments in automation could potentially open many doors for libraries. As we are seeing in this pandemic, there is a new normal rising which is focused more on flexible schedules. Our solution will not only help libraries safely resume their book issuing service but also help them to adjust to the new normal by providing 24/7 services. In conclusion, the overall market is moving towards automation services and this could be due to pandemic and/or improving their customer services. This is a good opportunity for libraries to start investing in automation services that have the potential to improve and excel in their services.

2) Customer or Client Analysis

A customer or client analysis is an important part of the marketing plan, as described in our class. This section will identify our potential clients and describe their needs for our product.

Our target customers are mainly libraries. This includes colleges, universities, schools, and public libraries. Basically, an entity where their main purpose is to lend, sell or rent books to individuals. Our vision to target this type of environment was clear from the start of the first semester. Our Problem Statement indicated that libraries suffered from global closures, making library access almost impossible for individuals like students. Therefore, our product was introduced to solve the sudden problem of COVID 19.

Since then, libraries have had to adjust their lifestyle to match the new normal. Our product gives the opportunity to such customers to return to business with minimal risk of infection due to our product's features that can operate without any supervision or human commands. This would allow our customers to comply with the local and federal laws that were introduced to fight COVID 19. Our product offers full automation and requires little to no maintenance. It also offers a free-of-touch experience when checking out a book due to its camera and QR code features. Also, our product requires a unique time and date to pick up a book for every user. This would clear the environment for one user at a time to enforce social distance roles and regulation and minimize possible legal issues for that business.

3) Market Value

As previously mentioned in our design idea for the societal problem, our autonomous book machine will provide great financial value for libraries in general. As there have been new ways and implementations for libraries to continue to provide services, they lack

the time availability and costs; something our machine can eliminate.

As shown in our funding, an estimate of $1,400 was spent on this machine which includes all the required materials used to create it. The idea is to create an autonomous book machine capable of delivering the required and much-needed services that libraries are unable to give and provide a relatively inexpensive cost by doing so. Since our design idea was created based on our understanding of the societal problem, there can and will be ways to improve the durability of the machine with different materials, more sensors, or a rearrangement of the cell-to-checkout window mechanism. However, for the sole purpose of this project, we have used materials that will carry out the services while keeping costs at a minimum.

4) Market Share Competition

When it comes to the market of library and remote delivery books there are two main competitors to our project, Smart Lockers, and home delivery. Out of both of these methods, Smart Remote Lockers are the best option because they can provide 24 hours service without human-to-human contact. However, even with smart lockers, it takes the library 3 to 5 business days to process the request and have the book available for pickup in lockers. For home delivery of the books, the time frame is even worse with 7 to 10 business days. However, these are only viable options currently available and are being actively used by many libraries across the United States including our own library at California State University, Sacramento (Sac State).

Smart Lockers or Smart Parcel Lockers have been around for a while and have been used by either delivery services or the stores, and less used by places like libraries. However, that changed when the COVID-19 joined the game, now many libraries across the United States own these lockers for delivery of their books. Unfortunately, there isn't enough creditable data available to show how many libraries have adopted this feature and the costs of having Smart Lockers. Hence why we turned to look at just the overall market for Smart Lockers. According to the Verified Market Research report, "Smart Parcel Locker Market was valued at USD 601.1 Million in 2020 and is projected to reach USD 1505.4 Million by 2028" [43]. Given these numbers, we can assume that portion of this market belongs to the library market, but there isn't enough data to prove how much. To understand these products better we looked at a couple of different companies and their prices. Based on quotes provided to Cleveland Public Library, three custom wrapped towers Smart Lockers with use of library card for access, sold by Bibliotheca, LLC would cost $42,181, this does not include warranties, shipping, installations, and maintenance costs [44]. Another quote provided was from D-Tech International USA, they quoted $27,985 for 3 towers of Smart Lockers, this does not include warranties, shipping, installations, and maintenance costs [44]. Lastly, the most affordable quote was provided by the Smiota, $9520 for 3 towers of smart lockers, this does not include warranties, shipping, installations, and maintenance costs [45]. Based on the cost of these Smart Lockers it is very much obvious that not all libraries can afford them and there is an urgent need for another method of safely delivering the books to the users that is affordable and involves less touching of the public used surfaces.

5) Why our product over the competition?

As we looked at the functionality of roughly a $20,000 smart locker, the user has to enter the pin and then the locker opens where their book resides. Once the book gets picked up, there is no additional feature that gives the user their receipt or reminds them of the due date. On the other hand, with only the price of $1,500, our product emails the user their receipts once they have picked up the book.

When we look at the smart locker solution and our product, they have very distinctive differences. For example, our one cell can hold at least 6 books in one cell and similarly at least 36 books in 6 cells. A smart locker can only hold one book in one locker and there are around 42 total lockers in 3 towers, depending on the size of each locker. Our cost of the machine is around $1,400; however, this cost includes the cost of experiments we did with different features to get the correct parts. If we remove all the experimental costs, our material cost is at $700+-$20 and if we include $300 of labor cost our total cost is at $1,000. We can market our product at $1,500. We are providing 36 books space for $1,500, whereas the other companies are providing 42 spaces for prices of $9,000+. In that too if the locker is a little wider in size the number of lockers decreases. Our product also elongates the user having to touch the keypad, instead, with QR code scanning users can easily pick up a book without touching any public surfaces. Our

product not just provides a safe way to deliver books but also provides a more economically viable choice to libraries especially smaller public libraries that are already suffering financially.

XI. Conclusion

In times of highly contagious disease forcing humanity's hand to protect themselves by keeping a distance from each other, there are a lot of problems that surfaced. Among these problems is the problem of people not being able to check out books from the library. Now given a normal situation, it probably would not seem like a huge issue, but since almost every library across the nation has closed its doors problem is now on another level of scale. If this was an issue just with public libraries, it would have been ignored, but this problem has swift every single book borrower of the library across the nation into trouble. The ones who suffer the most losses from this problem are students and professors, given the circumstances, if a student is not able to borrow a certain book from a library, they could possibly fail an assignment or worst fail a class. Professors on another hand can run into trouble of not being able to design courses without the help of library books. However, like everything else keeping libraries open is not an option as it puts thousands if not millions of people directly in a path of being exposed to COVID-19. So how can this issue be solved? To make this problem into an opportunity for the library to take a step forward and allow new technology integration with physical books. Our team is proposing an idea that not just solves the issue of COVID-19 but also helps the library move toward the direction of modernization. We are proposing an engineering solution to help libraries or similar facilities serve customers on autopilot mode, meaning a self-operated check-out system for the books. This solution would replace the basics tasks of checking out books, it will make the process of tracking books easier and faster along with other considerable benefits.

To consider this a problem, our team went through intensive research to ensure that this problem is not just affecting a couple of folks, but is affecting more than 2 million people across the United States. Now give that this problem is affecting more than 2 million people across the United States, we can safely assume that this is not a problem of one or two libraries but the problem of libraries at large. Part of the problem is that libraries have not paid much attention to modernizing their services as other business and educational institutions have. When we were researching and assessing this problem, we went through many different solutions that can help the libraries solve this problem. However, we wanted to choose a solution that is not just helping libraries solve this problem but also provide them an opportunity to improve their services after the pandemic is over. A couple of solutions we went through included a warehouse conveyor belt system, robotic arm to move the projects, etc… All of these solutions had one thing in common we wanted it to not just meet our senior project timeframe and requirements, also provide cost-efficient and effective solutions to the libraries. One of the reasons why we focused on cost efficiency is because the majority of the libraries have lower budgets and given the current situation of the COVID-19 pandemic the budget cuts will be harming the sector even more. So we kept that in mind when discussing and researching the solutions.

In September, our team accepted this problem and drew out the solution. The solution was very simple at first hand because of our inexperience we added features that we thought were easier; however, as time progressed and we went through implementation and testing we realized some of the features could have been further simplified. Moreover, our solution when we started consisted of two major parts, a website, and a physical checkout machine. Our vision was clear, but we were extremely inexperienced in working on a full-scale project from the beginning. Which led to some complications when we were deciding and researching for probable features. We thought that the features we had will be less and we can complete them in less than 6 months, which soon hit us in the face. The key features we decided on for the website were, Dashboard, book profile, user accounts, search bar, book recommendations, book checkout cart, and QR Code generator. We assumed that these features could be finished within less than 2 months if all the teammates worked on them simultaneously. In reality, when we started working on the website, it took us over a month to just put together the overall structure of the website.

In December and January, we spent time researching how to effectively implement the website, we experimented with different frameworks and ended up choosing the Springboot Framework with Java programming language for our backend and React JS for the frontend. We spent some time individually learning and experimenting with the

frameworks to get some experience in debugging and developing styles. Next, we set up our git repositories so we can easily work on our project at the same time. Our first goal was to get the backend structure of the website up, set up XAMPP Server, MySQL database, and connect the basic structure of the website to them. This allowed our team the flexibility to work on our features and task simultaneously and speeded up the project by a lot.

The key features we tried to tackle in the website phase were the dashboard, book profile, user accounts, and checkout cart. These four features were the biggest and most important features of the whole website portions, getting them done meant completing the website. Also, these features provided either data or structure that was used by the rest of the features. Dashboard, where all the books are listed for the user, required the book's profile feature to be completed. Our strategy was to complete the backend for all the features before proceeding to the frontend. We completed the book profile and dashboard, so we started implementing the user accounts and decided to start the front-end as well. Which turned out a good decision because there were some issues in the backend that would have taken us a long time to figure out and change had we not started the frontend early. Moreover, after the backend and basic frontend for the main features were completed, we shifted to working on the less prominent features like search bar, recommendations, and QR Code generator. Also, we decided to add a sub-feature to the checkout cart where we can let the user access the list of previous orders and their information. Among these order information was the status of the book, ready for pick up, picked up, or returned. It also allowed the user to access the QR Code for the orders here. This made our website more user-friendly. Among other small features, we also implemented the email feature, which notifies the user of the book checkout time and date.

Overall, our website's main user cycle is as follows. When new users come to the website, they land on the dashboard or the homepage, they can navigate to look for specific books, and go through the book profile and take a look at the comments and likes. However, if they want to check out a book they have to signup with a unique email, first and last names, date of birth, and password. If the email is used already website will not allow the user to sign up. Once the user is signed up, they can like books, comment on them, get recommendations of books, and most importantly order books. Once they choose a book they can click on get this book and it will add it to the cart and from there they can click order this book and choose the date and open time slot, completing the order. Once the order is completed a QR Code will be sent to the user in email and also will be available on the website in the orders section. Users can also see their previous orders, picked up, ready to be picked up, or returned. Once the QR Code is obtained then the user will have to visit our physical checkout machine to get the book.

When it comes to physically designing and building, a physical checkout machine took a long time and effort. Since this year's senior project was not in person, we lacked tools, instructions, directions, and help from professors and peers. Which cost us both in terms of money and time. Our original features for the physical design included shelf, checkout window, scanning QR Code, tray, horizontal and vertical sliders, book landing confirmation sensors, and coils or belts to drop the book onto the tray. Our assumption again was to complete this within less than 2 months and again we were completely wrong about our total estimate. However, we did leave grace times between the completion of each feature, which allowed us to bounce back when we were behind on any feature. We started by thinking about building our shelf, but sooner when estimated the prices of building our own and buying premade, we decided to go with premade to keep the cost of the overall project lower. Our next step was to get started on the sliders and coil or belts. We divided into two sub-teams to work on both features to increase productivity. However, after implementing the coils feature in three cells, we realized that it can potentially cause harm to books. After testing heavier books, we saw the coils bend, which was a big drawback for us. It was one of the first failures we faced during this project; however, we had a backup feature to alternate the coils. We decided to use two sets of belts rotated by a servo via pulleys. Which was a success and allowed us to move forward.

Our next big feature was sliders, which the other half of the team was already working on, our first approach was the use of pipes to form stable horizontal and vertical sliders. After some research and basic experiment, we decided to change our

approach to a more practical one with sliders that are Commercial Grade Pocket/Sliding Door Hardware. We had to make a lot of adjustments in terms of size, movement, attachment to the shelf, and other sliders. We also had to formulate it so we can add it to the tray without causing any issues. These sliders are controlled by the Stepper Motors through belts and pulleys, which requires calibration and adjustment frequently. Once this feature was completed, our team decided to start working on the smaller features like the tray, checkout window, and landing confirmations. These features did not take very long as they were very simple; however, integrating these features with the rest of the project took a long time due to communication issues we had between microcontrollers. Our checkout window is made up of glass-like plastic that prevents the user from getting access to the machine and a user window where the book arrives. The tray feature is made from a plastic container that has walls to prevent the book from falling off, it also has a window that is controlled by the servo attached to its right side. Both the checkout window and the tray have sensors on them to confirm the landing of the book in the tray and checkout window. Once all these features were done, we implemented our last but the most important feature of scanning and decoding the QR Code. It decodes the QR Code and accesses the information regarding the order (time, date of pickup, user, book title, etc). We used the Logitech camera and the OpenCV library to achieve that. With its completion our last hardship was integrating all the features, at that point we have integrated the features in pairs, but the final integration took us a while because of bugs and errors arising. The biggest issue we had was communication between microcontrollers keeps breaking down. However, we were able to resolve the issues and complete our prototype.

Overall, our physical machine portion of the project flows as follows. After the user has obtained the QR Code from the website they can come to the physical checkout machine at their selected date and time slot. Once they are here, they can scan their QR Code, which is then decoded by the QR Code Scanner feature. If for any reason user come in with a corrupted QR Code, or if the order's time has expired or if they are here ahead of the time then they will receive a problem specific email letting them know they are not able to check out the book and they need to take a certain action. However, if their QR Code is valid and they are in their selected time and date, QR Code will be decoded, and information will be grabbed regarding the location of the book from the database. It will then pass the location to the sliders; sliders will move the tray to the cell where the book is located and trigger the servos to rotate and drop the book. Once the book is dropped, book landing confirmation sensors will send a signal to the sliders to take the tray to the checkout window, but if the book is not landed on the tray sensors will trigger the servo again to rotate till a book is dropped on the tray. Next, when the tray has reached the checkout window, it will open its door using the servo, and the book slides onto the checkout window. On the checkout window, sensors will be triggered, and the ending program cycle is started. An email is sent to the user, the book is marked picked up and the program will be reset for the next user to use. This was the overall flow of our physical machine.

Through our project, we have also worked on the marketability of our project, its market share. We have also investigated the current competition that it might possess and how is our project better compared to them. It also gave us a perspective on things that we can improve if we ever want our end product to be in the market. One thing that we did achieve was keeping our product at a lower price compared to other similar solutions out in the market, which is beneficial for us because our targeted clients are in the lower budgets. We also realized that after completing this project we can now think of ways t improve it, for example, we can 3D print our tray to make it more efficient or purchase tools ahead of time to prevent extra costs. All this can reduce the overall costs of our project.

At last, our team has achieved the project by providing the solution to the problem of libraries not being able to check out books to the users. We have successfully created a website that lets the user find and order the book, then using the physical checkout machine user can check out the book. It is all done while maintaining social distancing and avoiding face-to-face interactions with other users and library staff. We have met the feature set of the website and physical checkout machine, also the features were live demoed to the instructors and the students.

REFERENCES

[1] Morales, M. and Hines, S., 2020. Public Libraries Launch, Expand Services During COVID-19 Pandemic. [online] American Library Association (ALA). Available at: <http://www.ala.org/news/press-releases/2020/04/public-libraries-launch-expand-services-during-covid-19-pandemic-0> [Accessed 22 September 2020].

[2] Ifla.org. 2020. *IFLA -- COVID-19 And The Global Library Field*. [online] Available at: <https://www.ifla.org/covid-19-and-libraries> [Accessed 25 September 2020].

[3] American Library Association, "Libraries Respond: Covid-19 Survey," May 12-18, 2020. [Online]. Available: <http://www.ilovelibraries.org/sites/default/files/PLA-MAY-2020-COVID-19-Survey-Results_PDF-Summary-web.pdf/> [Accessed 30 January 2021].

[4] Horrigan, J., 2016. *Library Usage And Engagement By Americans*. [online] Pew Research Center: Internet, Science & Tech. Available at: <https://www.pewresearch.org/internet/2016/09/09/library-usage-and-engagement/> [Accessed 26 September 2020].

[5] Holtan, E., 2018. *People Visited Public Libraries More Than A Billion Times In 2015*. [online] Institute of Museum and Library Services. Available at: <https://www.imls.gov/news/people-visited-public-libraries-more-billion-times-2015#:~:text=In%202015%2C%20there%20were%201.39,attendees%20than%20the%20previous%20year> [Accessed 25 September 2020].

[6] CSU Faculty Data. 2020. *Head Count Of Full-Time Faculty By Rank And Campus*. [online] Available at: <https://www2.calstate.edu/csu-system/faculty-staff/employee-profile/csu-faculty/Pages/headcount-for-full-time-faculty-by-rank-and-campus.aspx> [Accessed 24 September 2020].

[7] Student Enrollment. 2020. *Student Enrollment Fall 2019*. [online] Available at: <https://www2.calstate.edu/csu-system/about-the-csu/facts-about-the-csu/enrollment> [Accessed 21 September 2020]

[8] K. Shah, "California State University Data", 2020

[9] Kumah, C., 2015. *A Comparative Study Of Use Of The Library And The Internet As Sources Of Information By Graduate Students In The University Of Ghana*. [online] Digitalcommons.unl.edu. Available at: <https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=3473&context=libphilprac> [Accessed 23 September 2020].

[10] Cox, C., 2020. *Academic Libraries Will Change In Significant Ways As A Result Of The Pandemic (Opinion)*. [online] Insidehighered.com. Available at: <https://www.insidehighered.com/views/2020/06/05/academic-libraries-will-change-significant-ways-result-pandemic-opinion> [Accessed 26 September 2020].

[11] Perrin, A., 2016. *Book Reading 2016*. [online] Pew Research Center: Internet, Science & Tech. Available at: <https://www.pewresearch.org/internet/2016/09/01/book-reading-2016/> [Accessed 23 September 2020].

[12] International Federation of Library Associations and Institutions, "Covid-19 and the Global Library Field," October 13, 2020. [Online serial]. Available: <https://www.ifla.org/covid-19-and-libraries#social/> [Accessed 30 January 2021].

[13] Sac State Library, "Book Request," [Online]. Available: <library.csus.edu/spotlight-and-events/book-request/> [Accessed 30 January 2021].

[14] SEMUELS, A. (2020). DOES REMOTE WORK ACTUALLY WORK? COVID-19 is forcing companies to find new ways for employees to do their jobs and stave off social isolation. Time International (Atlantic Edition), 195(12/13), 36–41.

[15] Anane, G. K., Addo, P. K., & Adusei, A. (2020). Impact of the COVID-19 Pandemic on the Work of University Administrators in Ghana. International Studies in Educational Administration (Commonwealth Council for Educational Administration & Management (CCEAM)), 48(1), 93–102.

[16]Mukhopadhyay, S., Booth, A. L., Calkins, S. M., Doxtader, E. E., Fine, S. W., Gardner, J. M., Gonzalez, R. S., Mirza, K. M., & Jiang, X. S. (2020). Leveraging Technology for Remote Learning in the Era of COVID-19 and Social Distancing: Tips and Resources for Pathology Educators and Trainees. Archives of Pathology & Laboratory Medicine, 144(9), 1027–1036. https://doi-org.proxy.lib.csus.edu/10.5858/arpa.2020-0201-ED
[17]Horrigan, J., 2015. *Chapter 1: Who Uses Libraries And What They Do At Their Libraries*. [online] Pew Research Center: Internet, Science & Tech. Available at: <https://www.pewresearch.org/internet/2015/09/15/who-uses-libraries-and-what-they-do-at-their-libraries/> [Accessed 28 September 2020].
[18]Zhang, Y. and Kudva, S., 2014. E-books versus print books: Readers' choices and preferences across contexts. *Journal of the Association for Information Science and Technology*, 65(8), pp.1695-1706.
[19]Taylor, K., 2010. *That Mighty Sorting Machine Is Certainly One For The Books*. [online] Nytimes.com. Available at: <https://www.nytimes.com/2010/04/22/books/22library.html> [Accessed 22 September 2020].
[20]F. Alaulddin, “Feature Punch List”, 2021
[21]Leff, A. and Rayfield, J. T, "Web-application development using the Model/View/Controller design pattern."
[22]Altar, A. and Paul, D, “Designing an MVC Model for Rapid Web Application Development.”
[23]Banks A. and Porcello, E, “Learning React, Functional Web Development with React”
[24]“C270 HD Webcam documentation,” Logitech official website.
[25]“Python Software Foundation, Open AI documentation,” Python official website.
[26]E. Silva, “Rail system with stepper motors and tray”, 2021
[27]“100PD Commercial Grade Pocket/Sliding Door Hardware”, Amazon official website.
[28]“STEPPERONLINE Nema”, Amazon official website.
[29]“TB6600 4A 9-42V Stepper Motor,” Amazon official website.
[30]“MENZO”, Amazon official website.
[31]S. Malhi, “Opening the tray with Servo Motor”, 2020
[32]“Continuous Rotation Servo,” Amazon official website.
[33]“Under Bed Clear Storage Container”, The Home Depot official website.
[34]“Storage Cubes,” Product from Lowe’s.
[35]“Readytosky Digital 30KG Servo”, Amazon official website.
[36]“Adafruit CircuitPython ServoKit” and “Adafruit CircuitPython PCA9685” GitHub Repository.
[37]“AiWAN LEZHI 5m GT2 Timing Belt 6mm”, Amazon official website.
[38]“IR Break Beam Sensor” Adafruit official website.
[39]E. Silva, “Funding”, 2021
[40]K. Shah, “Risk Assessment Matrix”, 2021
[41]F. Alaulddin, “Fail/Pass Criteria”, 2021
[42]B. Birnbaum, “The Rise of Human Agents: AI-Powered Customer Service Automation,” Forbes, 19-Jun-2019. [Online]. Available: https://www.forbes.com/sites/bradbirnbaum/2019/06/19/the-rise-of-human-agents-ai-powered-customer-service-automation/?sh=10504b8393f2. [Accessed: 01-Mar-2021].
[43]“Global Smart Parcel Locker Market Size By Product Type, By ...” [Online]. Available: https://www.verifiedmarketresearch.com/product/global-smart-parcel-locker-market/. [Accessed: 01-Mar-2021].
[44]“EXHIBIT 18 CLEVELAND PUBLIC LIBRARY RESOLUTION AUTHORIZING ...” [Online]. Available: https://cpl.org/wp-content/uploads/board/20200220-ex18-materials-hold-locker-w-attach.pdf. [Accessed: 01-Mar-2021].
[45]“EXHIBIT 18 CLEVELAND PUBLIC LIBRARY RESOLUTION AUTHORIZING ...” [Online]. Available: https://cpl.org/wp-content/uploads/board/20200220-ex18-materials-hold-locker-w-attach.pdf. [Accessed: 01-Mar-2021].
[46]S. Malhi, “User Manual Pictures”, 2021

[47]F. Alaulddin., “A.B.S Website Screenshots”, 2021
[48]E. Silva, “Large Scale Machine Hardware Block Diagram”, 2021
[49]“Raspberry Pi 4 Model B”, Amazon official website.
[50]“SunFounder Mega 2560 R3 ATmega2560”, Amazon official website.
[51]F. Alaulddin., “GitHub Repository Snapshot”, 2021
[52]E. Silva, “Book machine pictures and Components”, 2021
[53]S. Malhi, “Project Timeline and PERT Diagram”, 2021

GLOSSARY

API - Application Programming Interface, a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other services.

Check out Books - Physically borrowing or lending books from the library under a contract to return the book by a certain due date.

COVID-19 - Coronavirus (COVID-19) is an illness caused by a virus that can spread from person to person. The virus that causes COVID-19 is a new coronavirus that has spread throughout the world. COVID-19 symptoms can range from mild (or no symptoms) to severe illness.

CRUD - Create, Read, Update, and Delete, which are the four basic operations of a database.

CSS - Cascading Style Sheet, used for defining styles for web pages.

GPIO pin - general-purpose input/output pin.

HTML5 - Hypertext Markup Language revision 5, is a markup language for the structure and presentation of World Wide Web contents.

I2C – Inter-Integrated Circuit, is an asynchronous, multi-master, multi-slave, packet-switched, single-ended, serial communication bus.

JavaScript - an object-oriented computer programming language commonly used to create interactive effects within web browsers.

MVC (Model–view–controller) - is a software design pattern commonly used for developing user interfaces.

OpenCV - is a library of programming functions mainly aimed at real-time computer vision.

Pandemic - occurring over a wide geographic area (such as multiple countries or continents) and typically affecting a significant proportion of the population.

PHP - Hypertext Preprocessor, is a general-purpose scripting language especially suited to web development.

PHPMailer - is a code library to send (transport) emails safely and easily via PHP code from a web server.

Public Library - a nonprofit library maintained for public use and usually supported in whole or in part by local taxation.

PWM (Pulse Width Modulation) - is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts.

Python - is an interpreted, high-level, and general-purpose programming language.

QR Code - a machine-readable code consisting of an array of black and white squares, typically used for storing URLs or other information for reading by the camera.

Raspberry pi - The Raspberry Pi is a low-cost, credit-card-sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python.

ReactJS - is an open-source, front-end, JavaScript library for building user interfaces or UI components.

Servo Motor - is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration.

SMTPLIB - The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

Stepper Motor - is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.

URL: Uniform Resource Locator, the address of a World Wide Web page.

Springboot Java: Backend framework that allows us to build API endpoints – written in Java programming language.

APPENDIX A. USER MANUAL

The Autonomous Book Solution provides users with an online reservation system and an autonomous checkout station. To fully use these two systems the user and technician must consider the steps below:

A. On-site Setup (for technicians)

This setup requires 6ft x 6ft space and 7ft of ceiling clearance. The current prototype state requires the autonomous checkout station to be close to the power outlet and have three power sockets available.

1. The first step is to move the project to a desired place and connect the checkout window with the rest of the system. Find the three wires coming from the checkout window (blue, yellow, and white) and connect them with the wires coming from under the shelf. Connect by matching the same-colored wires. Now place the checkout window 12 inches away from the shelf (from the center of the shelf).

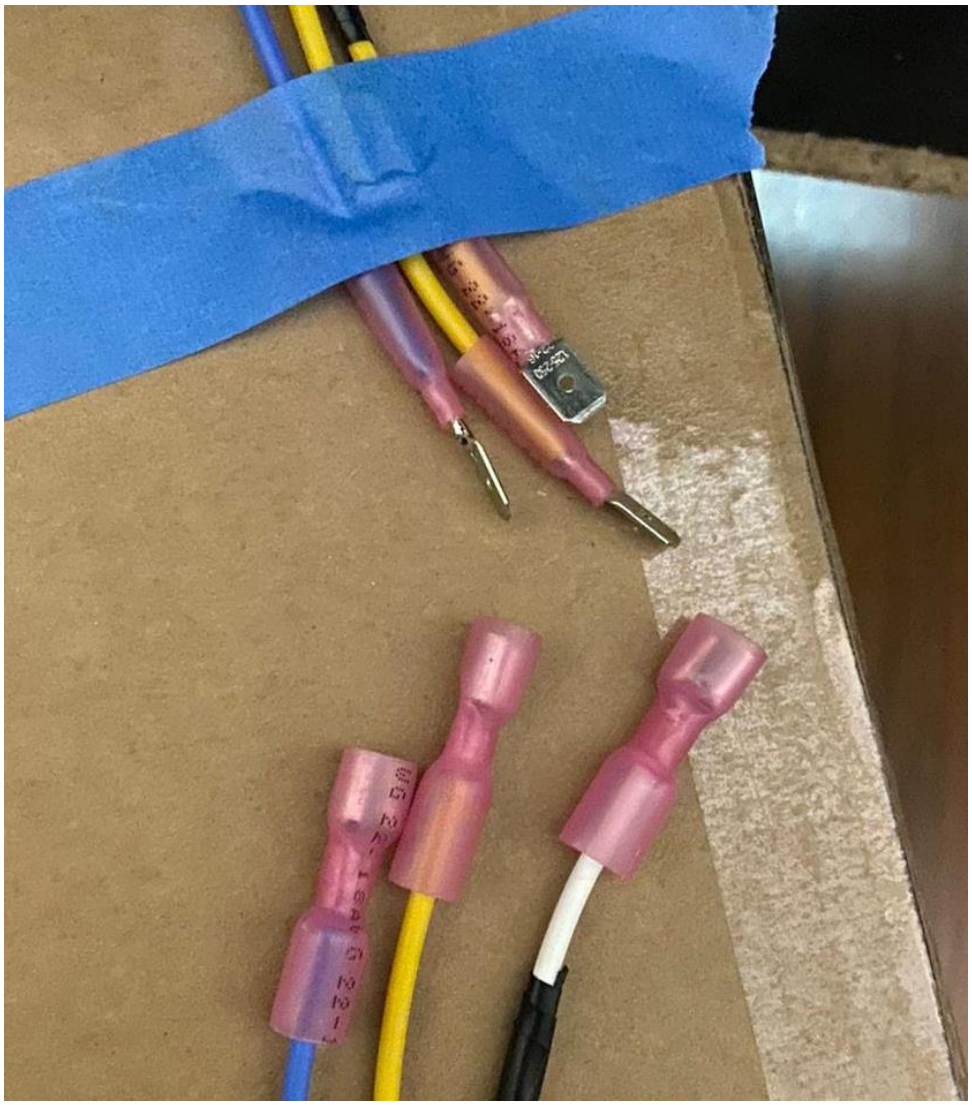


Figure A 1: Checkout Window Connecting Wires *[46]*

2. Plugin the power adaptors to both Raspberry pi microprocessors placed in the black box at the top of the shelf.
3. As part of the initial setup, attach the keyboard/mouse and display monitor to the microprocessors.

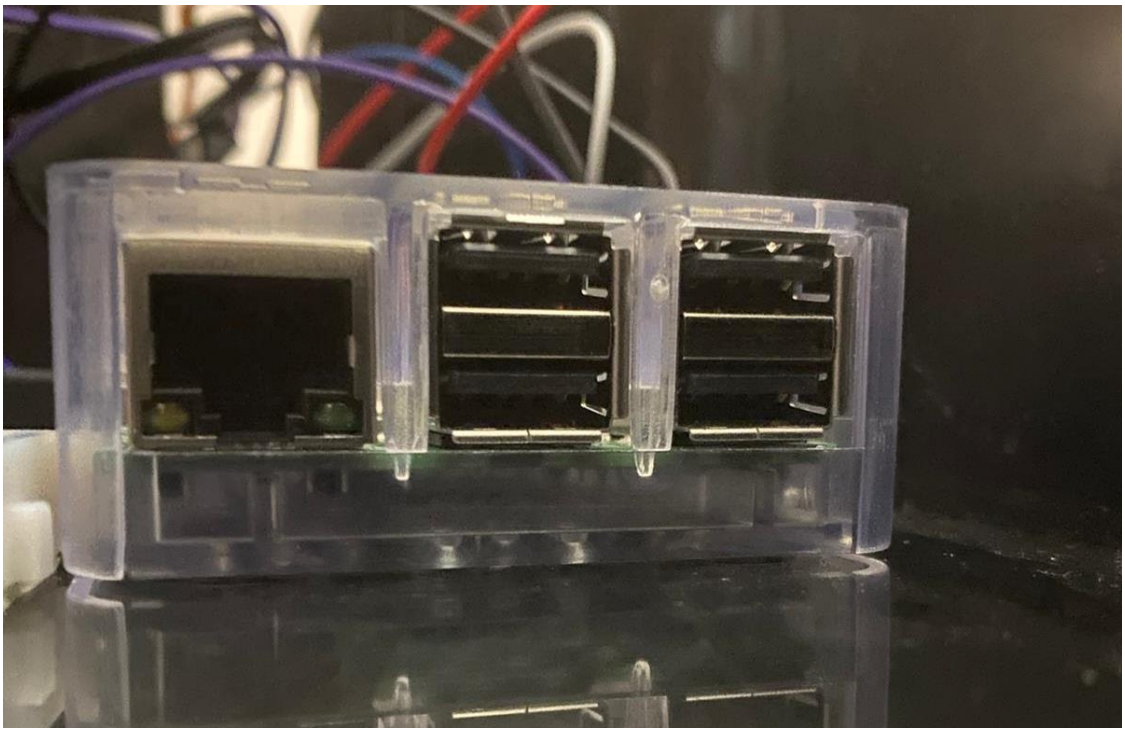


Figure A 2: Main Microprocessor I/O Ports *[46]*

4. Plugin the power supply cord.
5. In the main microprocessor, navigate to the NSP folder. Open and Run abs2.py
6. In the client microprocessor, navigate to the SP folder. Open and run client.py
7. Once both main pi and client pi have started to run, you can unplug the monitors, keyboards, and mouse. Your system is now capable of running autonomously.

B. Using the System (for users)

The User Diagram below gives an overall idea of how this system works for library users.

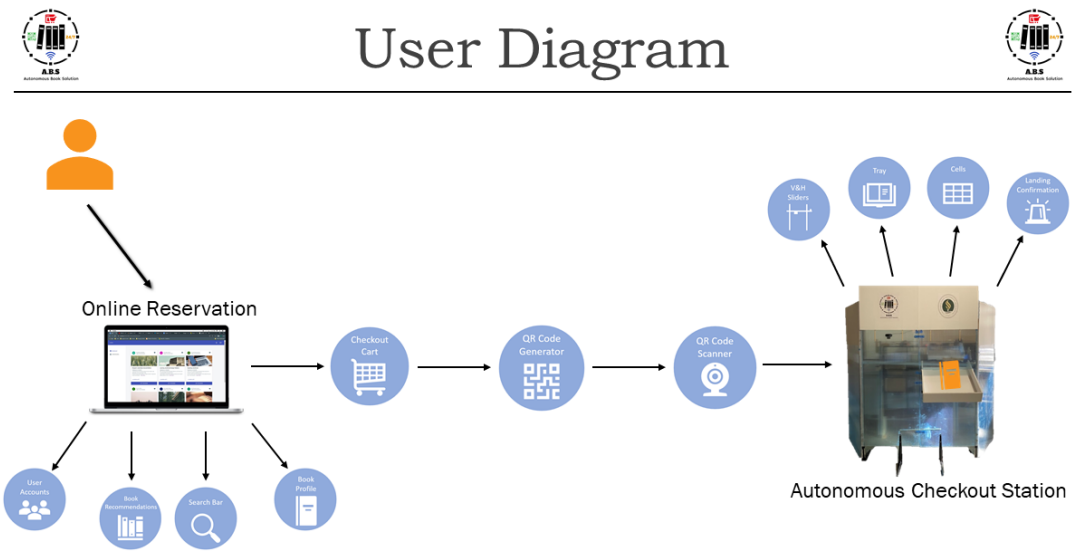


Figure A 3: User Diagram [46]

1. The user first needs to log in/sign up to the website to be able to check out a book.

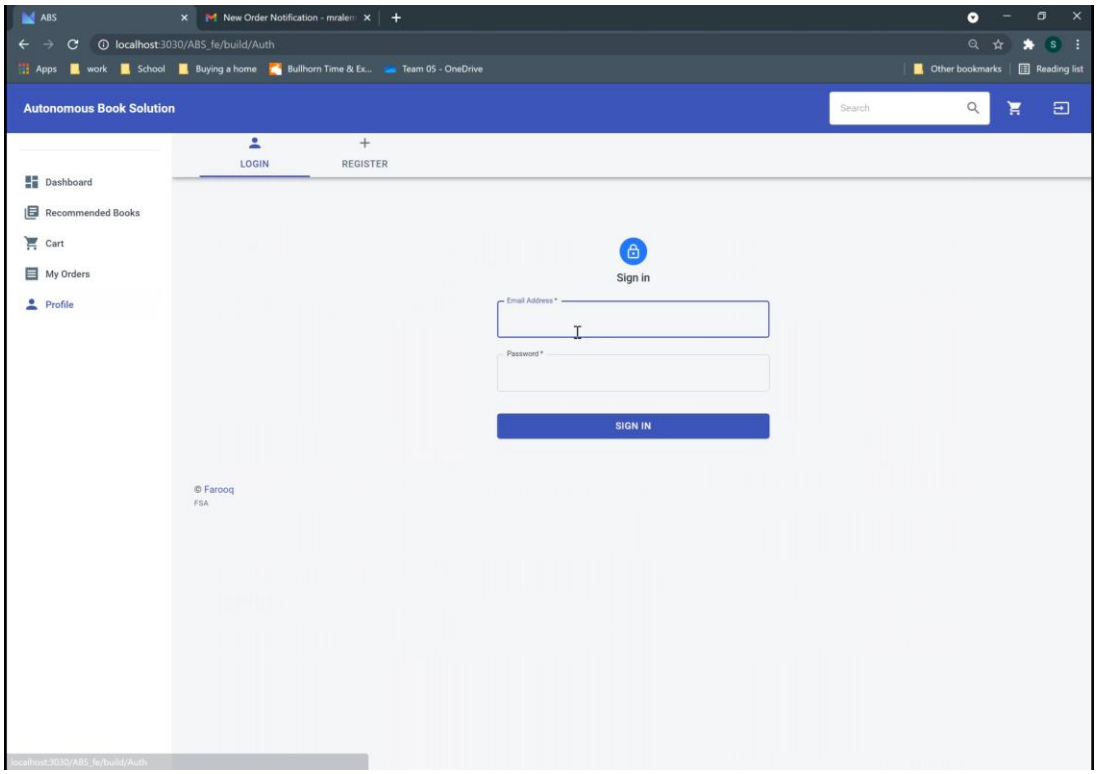


Figure A 4: Log In Page [47]

2. Once the user has logged in, they will see a dashboard with various books on it. Now the user has three options to find. They can either find the book in the dashboard, in the recommended books section or find the book through the search bar.

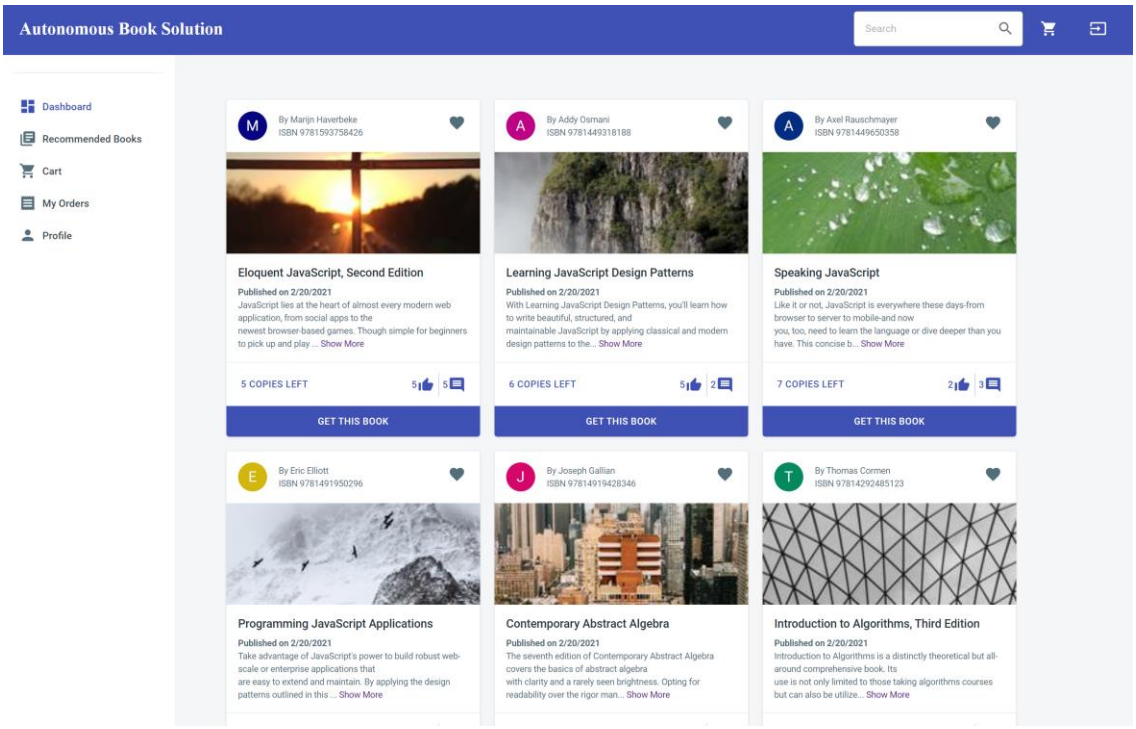


Figure A 5: Dashboard [47]

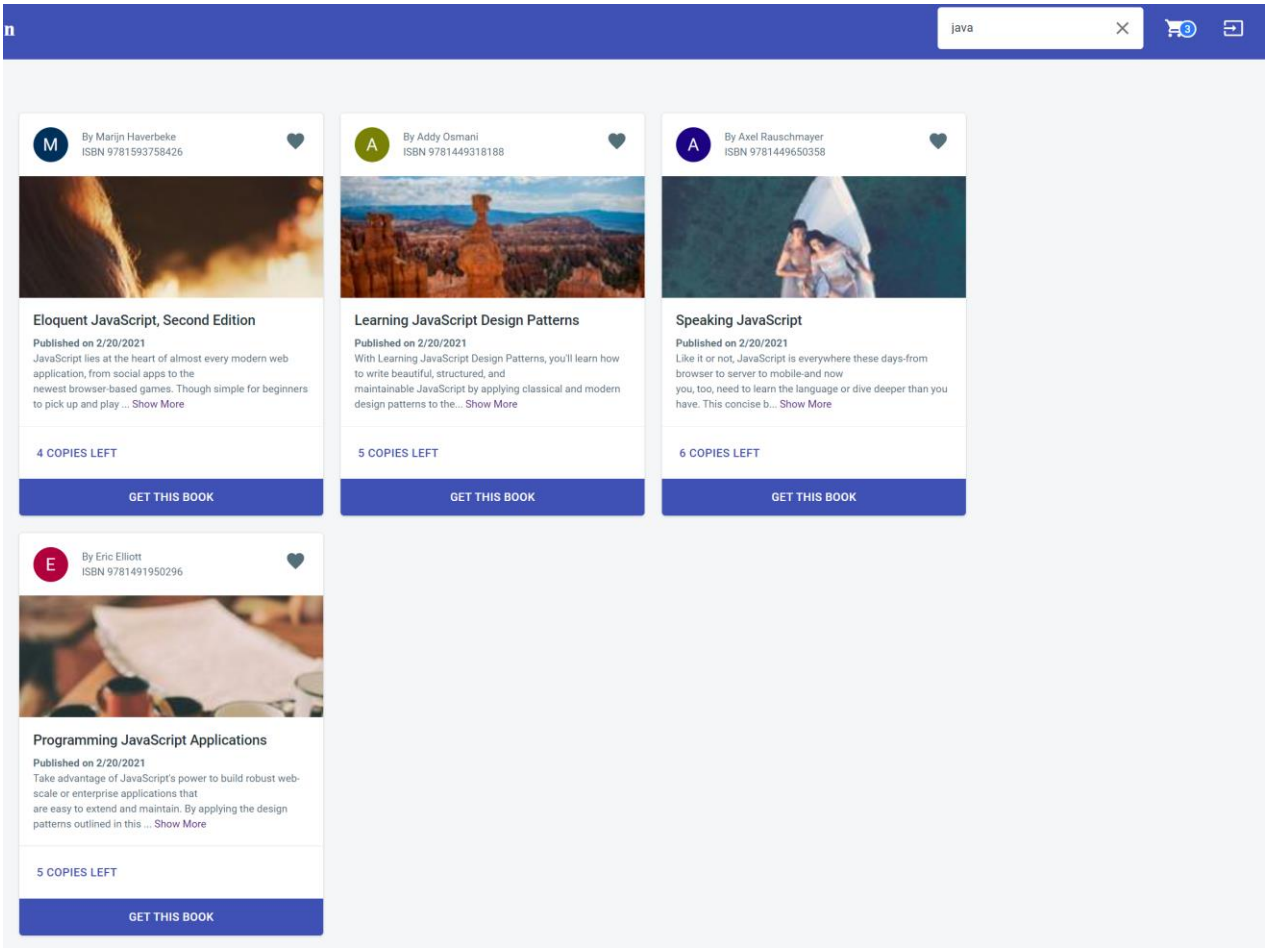


Figure A 6: Any-keyword Search [47]

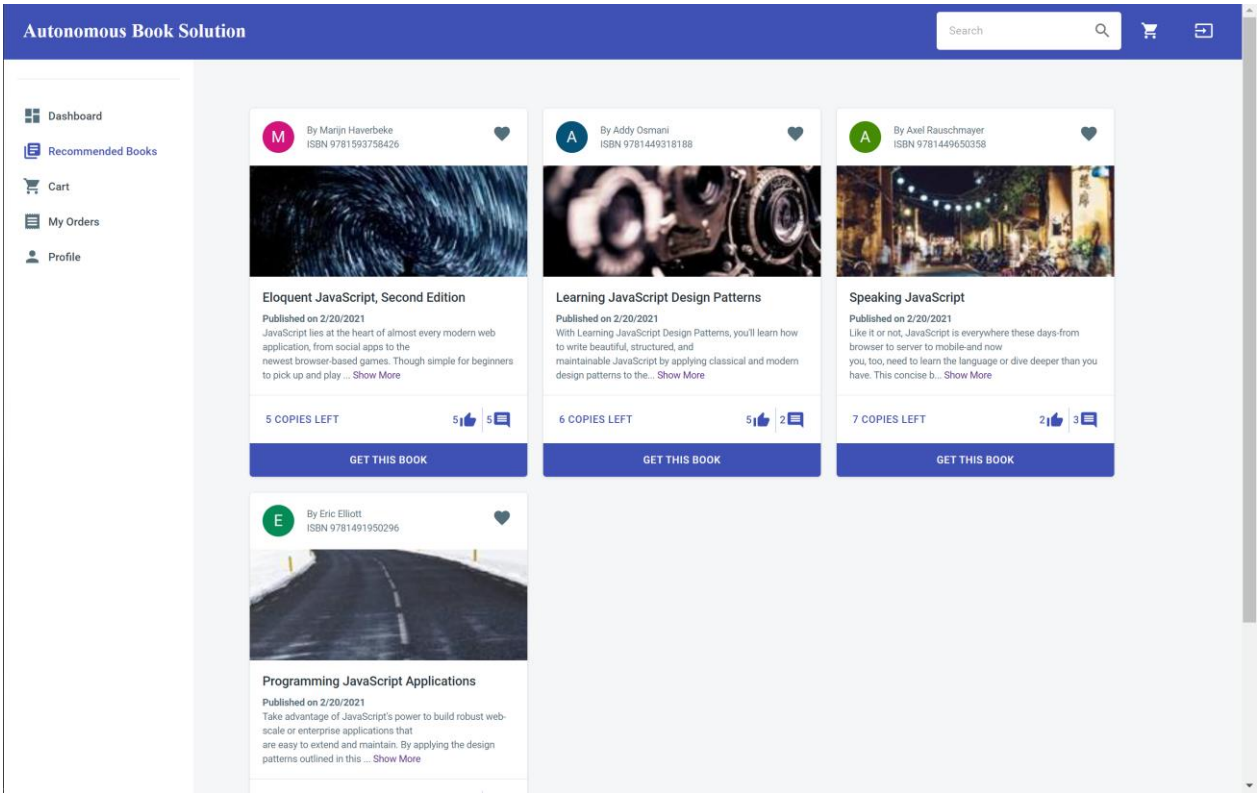


Figure A 7: Book Recommendation [47]

3. Once the user has selected the book they were looking for. They will be able to see the book profile and add the book to the cart.

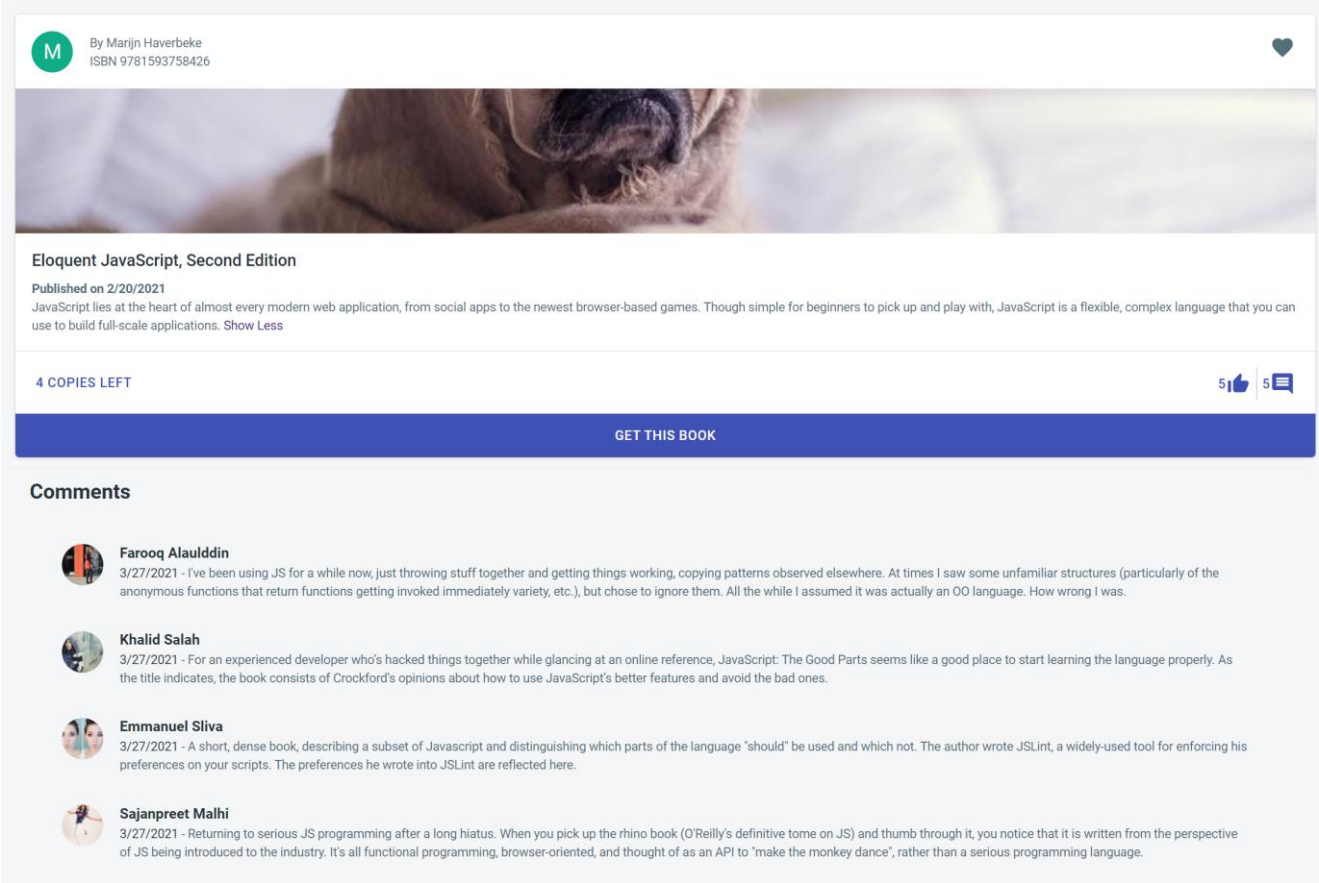


Figure A 8: Book Profile [47]

4. Once the book has been added to the cart. The user will have to get a chance to pick their appointment time.

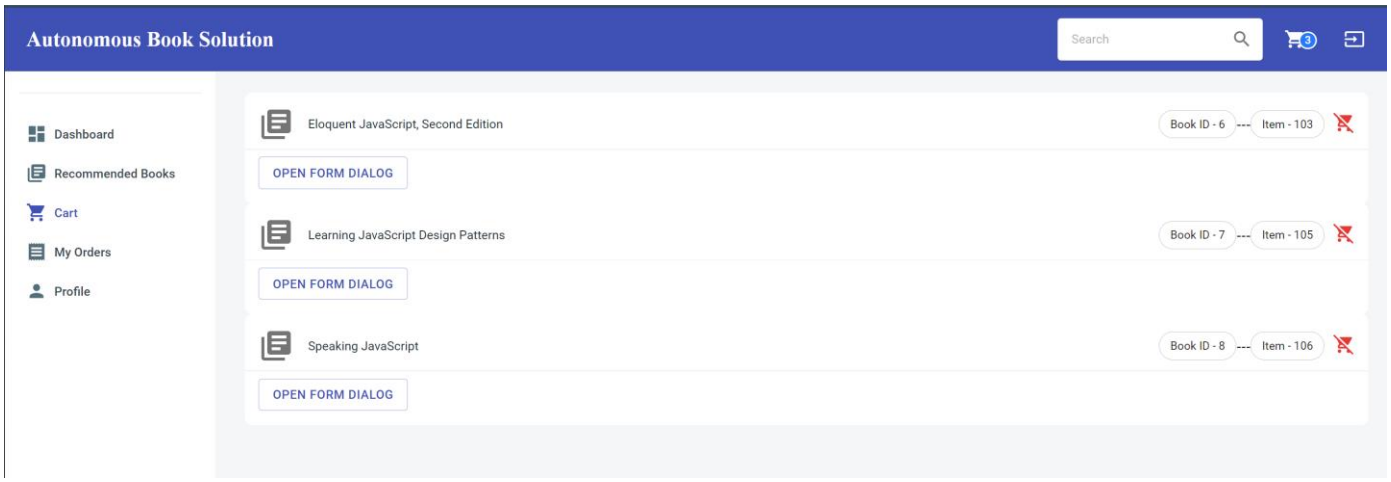


Figure A 9: Book Checkout Cart 1 [47]

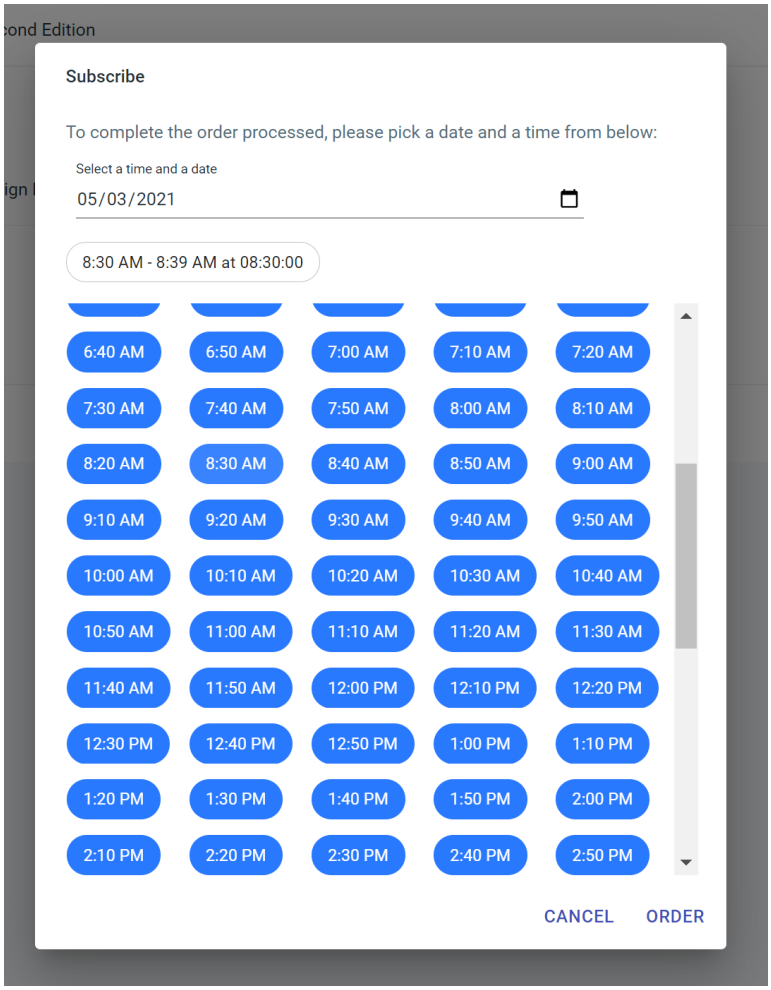


Figure A 10: Book Checkout Cart 2 [47]

5. Once the user has completed their appointment. Their orders will move to the My Orders section and will display their status.

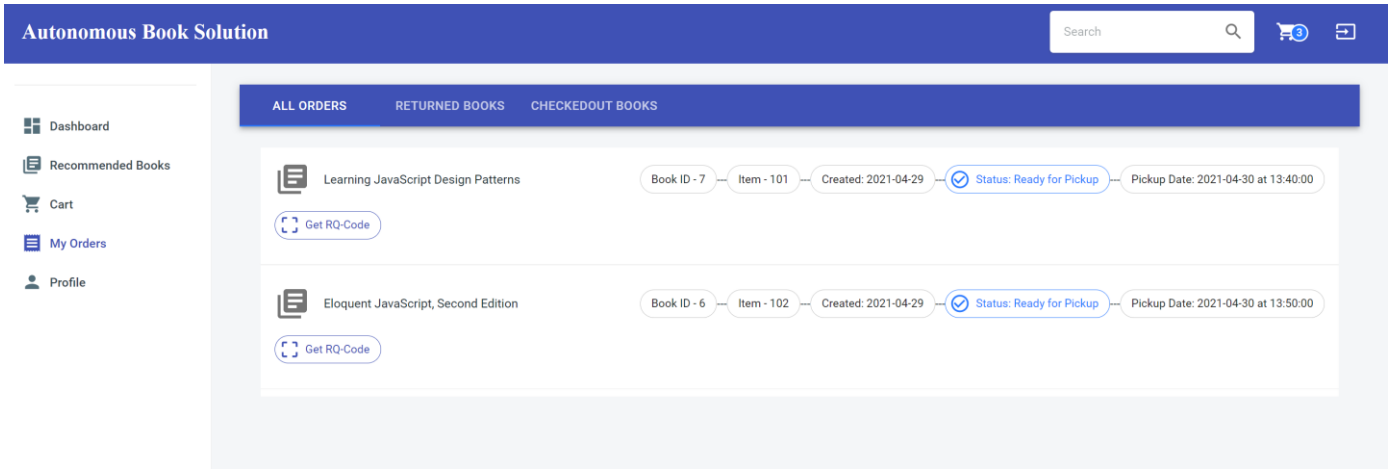


Figure A 11: User Profile and Account Services 1 [47]

6. When the order moves to the My Order section, an email is sent to the user regarding their appointment details and a QR code.

Figure A 12: Reservation Email Notification [47]

7. If the user wants to view their account, click on Profile from the side menu and the account information and history will be displayed.

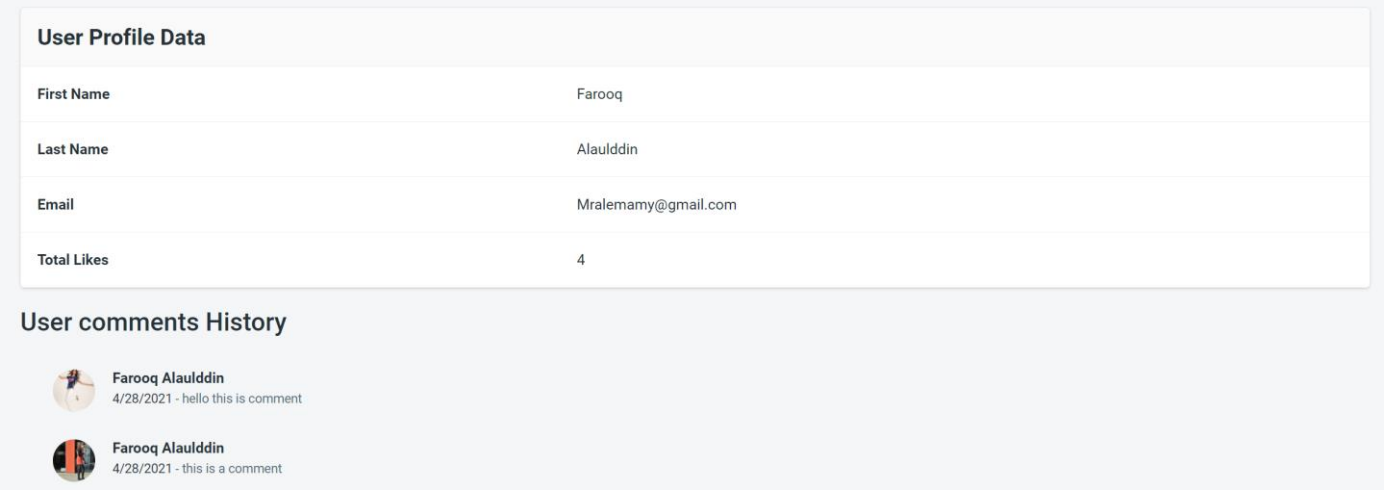


Figure A 13: User Profile and Account Services 2 [47]

APPENDIX B. HARDWARE

The machine of our project involved several components used to dispense the book from the shelf to the user. We used the Raspberry pi as the main component to execute and communicate between systems of software and hardware. However, it only contained the coding for the physical machine and not the website. We also used an Arduino Mega 2560 to execute the functions needed to move the stepper motors for the horizontal and vertical sliders. Since there are two-dimensional components to the sliders, we used two Nema 23 Stepper motors with two TB6600 Stepper motor drivers to move in the vertical and horizontal directions. These were supplied with power using a MENZO 360W 12V 30A DC universal switching power supply. In order to move the books from the cell to the tray, we used Readytosky Digital 30kg Servo in each cell to move the books by rotating a belt. Lastly, we used IR break beam sensors to communicate with the Arduino and Raspberry pi to confirm if the book had landed on the tray and to proceed onto the next step in the system. We used the Logitech C270 webcam to scan the QR code at the front of the machine.

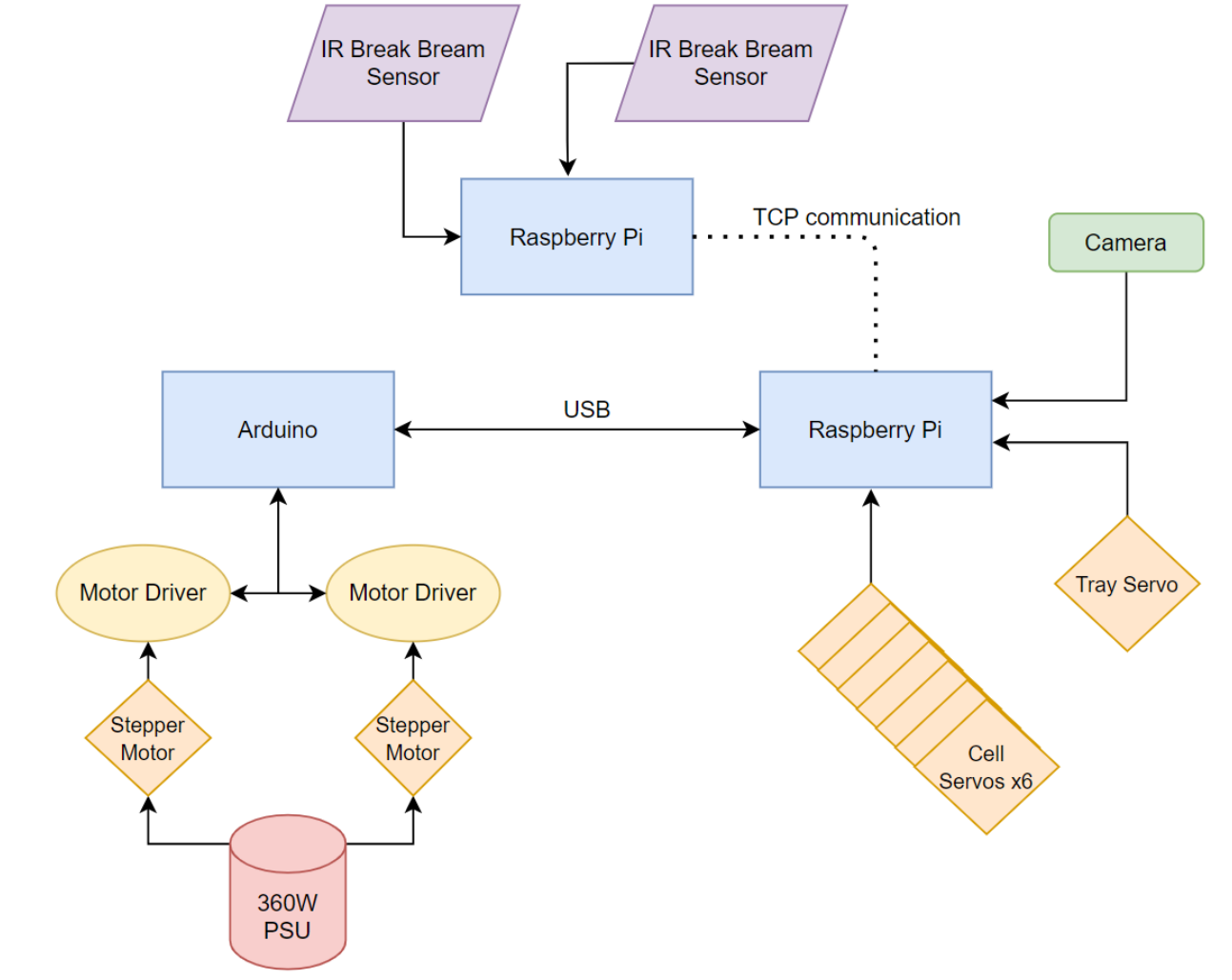


Figure B 1: Large-Scale Machine Hardware Block Diagram

Figure B 2: Raspberry Pi [49]

Figure B 3: SunFounder Mega 2560 [50]

Figure B 4: MENZO 360W Power Supply [30]

Figure B 5: TB6600 Stepper motor driver [29]

Figure B 6: Nema 23 Stepper motor [28]

Figure B 7: Readytosky Digital 30KG 360 Servo [35]

Figure B 8: IR Break Beam Sensor [38]

Figure B 9: Logitech HD Webcam C270 [24]

APPENDIX C. SOFTWARE

There are three software main components in our project Website frontend, Website backend, and Machine script. The Website frontend is built using popular frameworks and libraries such as React, Google Material UI. Our backend serves the essential CRUD operations with security and database support functionality. Lastly, we have the scripting for the physical machine itself. This software component deals with the Raspberry PI, Arduino, motors movements, and all the hardware that needed software control. The entire source code of our applications can be found on Autonomous Book Solution.

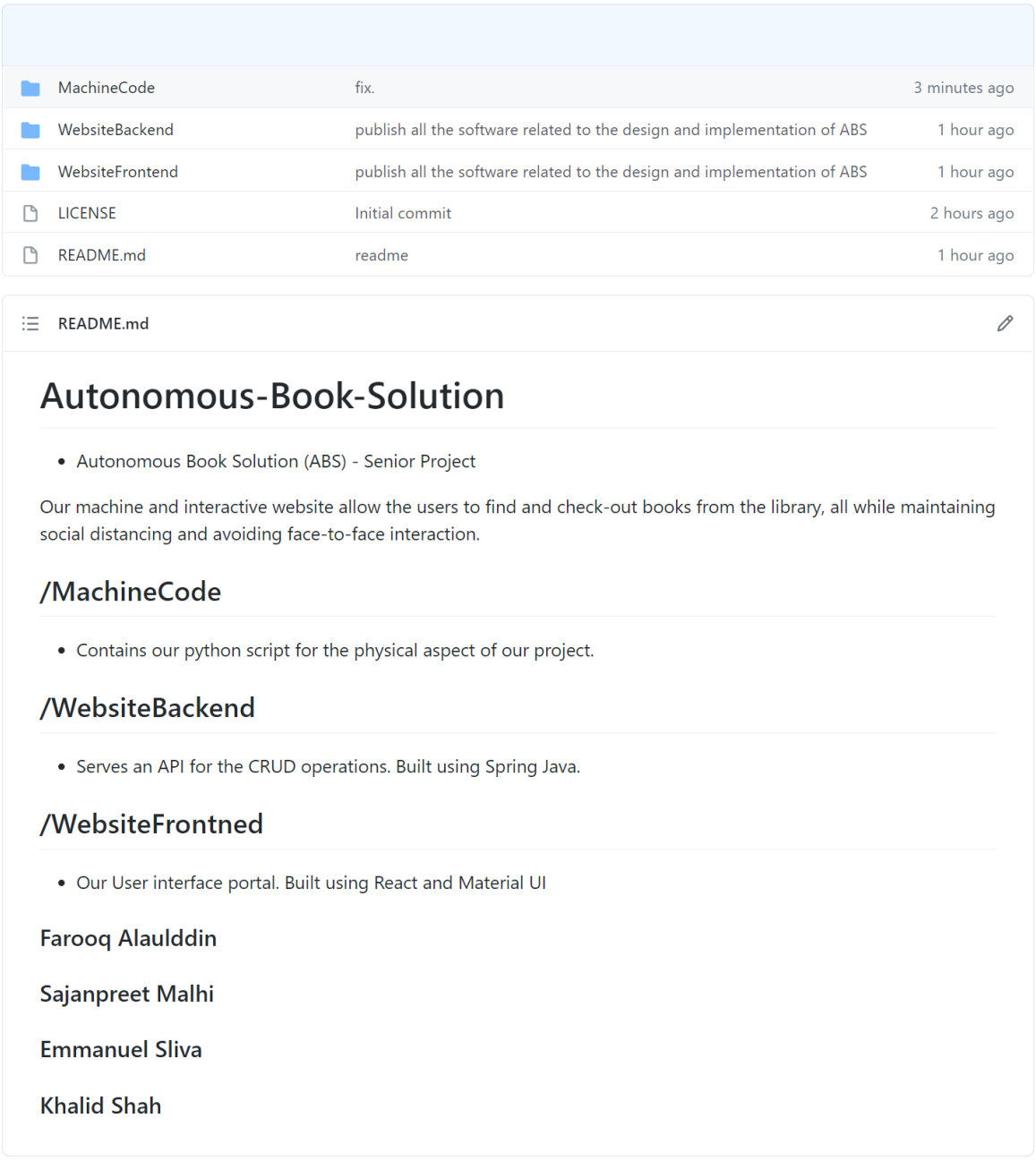


Figure C 1: GitHub Repository Snapshot [52]

APPENDIX D. MECHANICAL ASPECTS

The mechanical aspects of this project involved a lot of woodworking and mechanical building for the book machine. Since this project was built by us, we utilized materials like wood, commercial-grade pocket/sliding door hardware, belts, pulleys, plastic container, a shelf, polycarbonate hollow sheets, and stainless-steel rods.

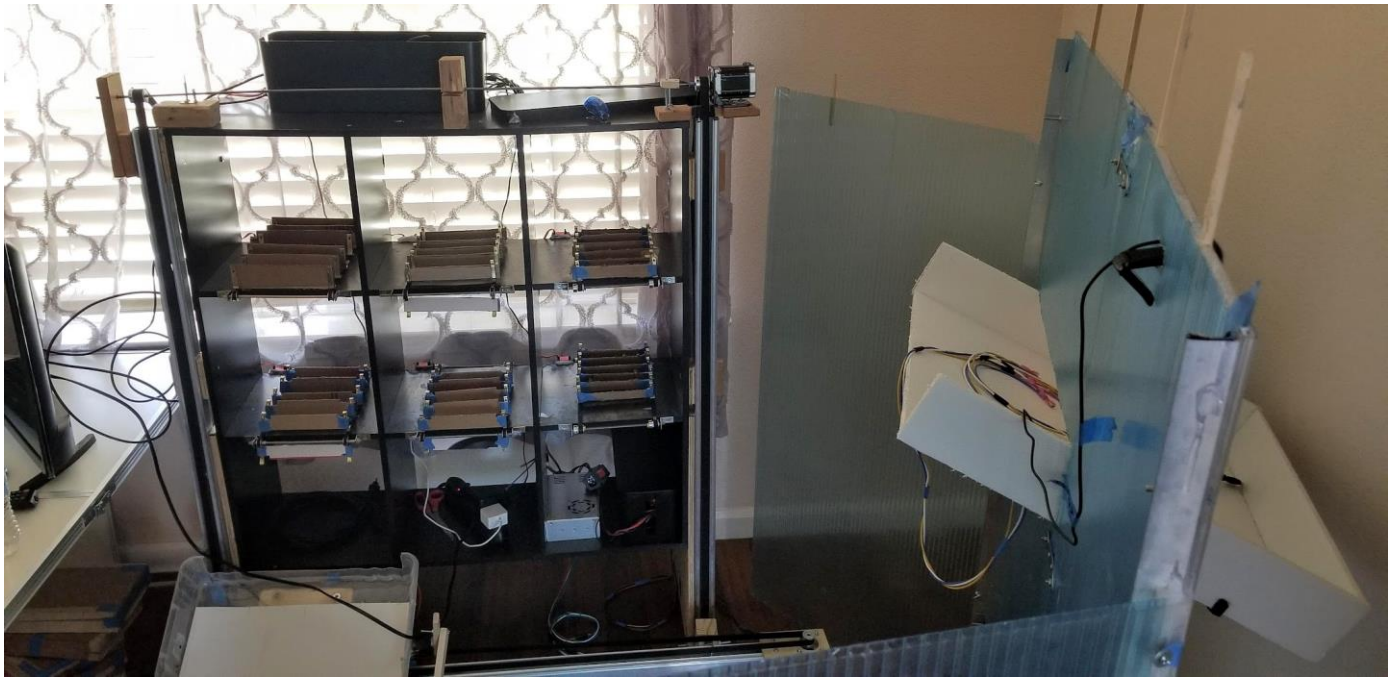


Figure D 1: Large-Scale Machine with all material (in-between view) [52]

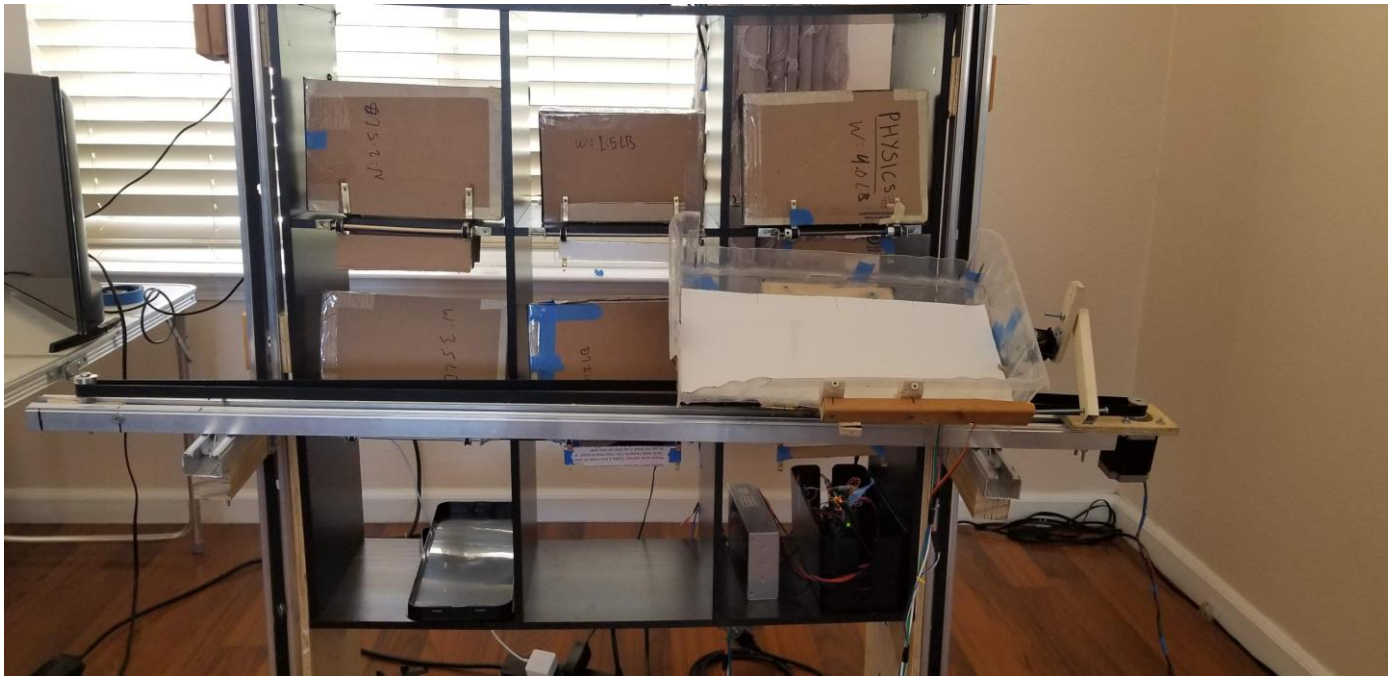


Figure D 2: Front View of Machine (behind checkout window) [52]

Figure D 3: Cells containing testing books on Shelf [52]

Figure D 4: Tray Mounted on Railing Tracks (sliders) [52]

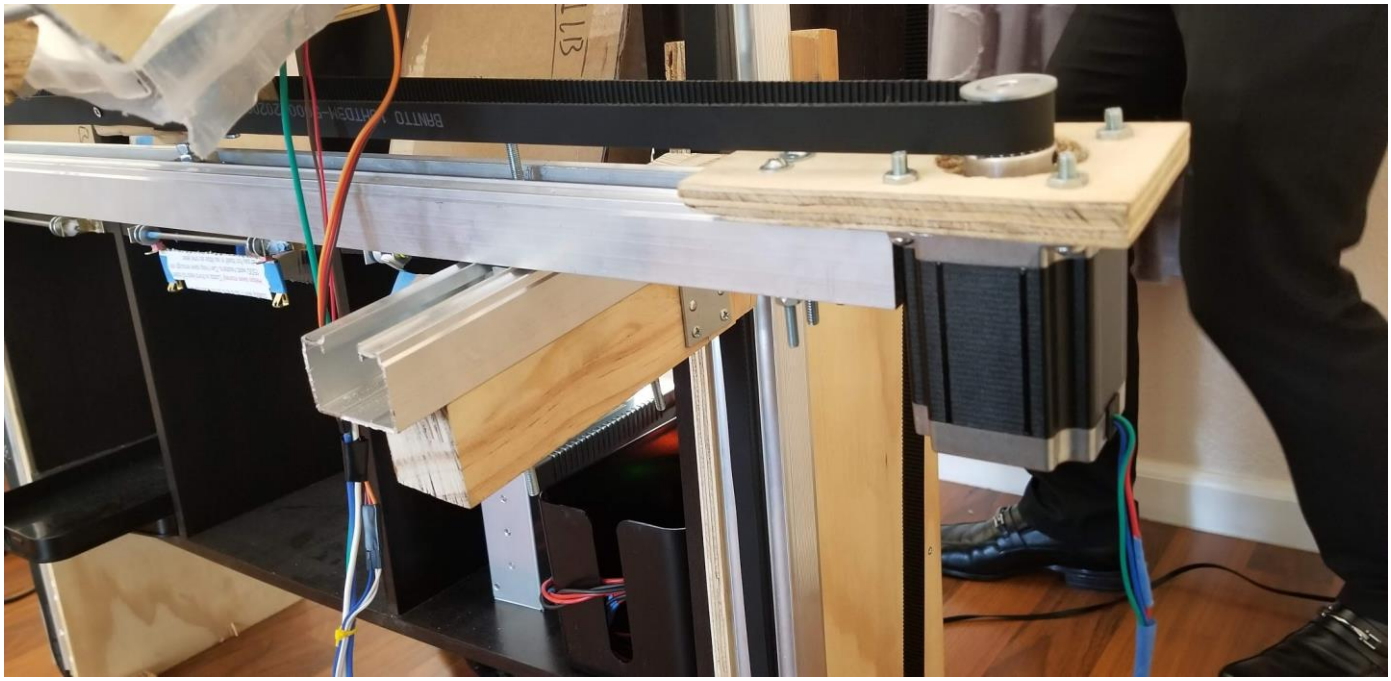


Figure D 5: Horizontal Portion of the Sliders [52]

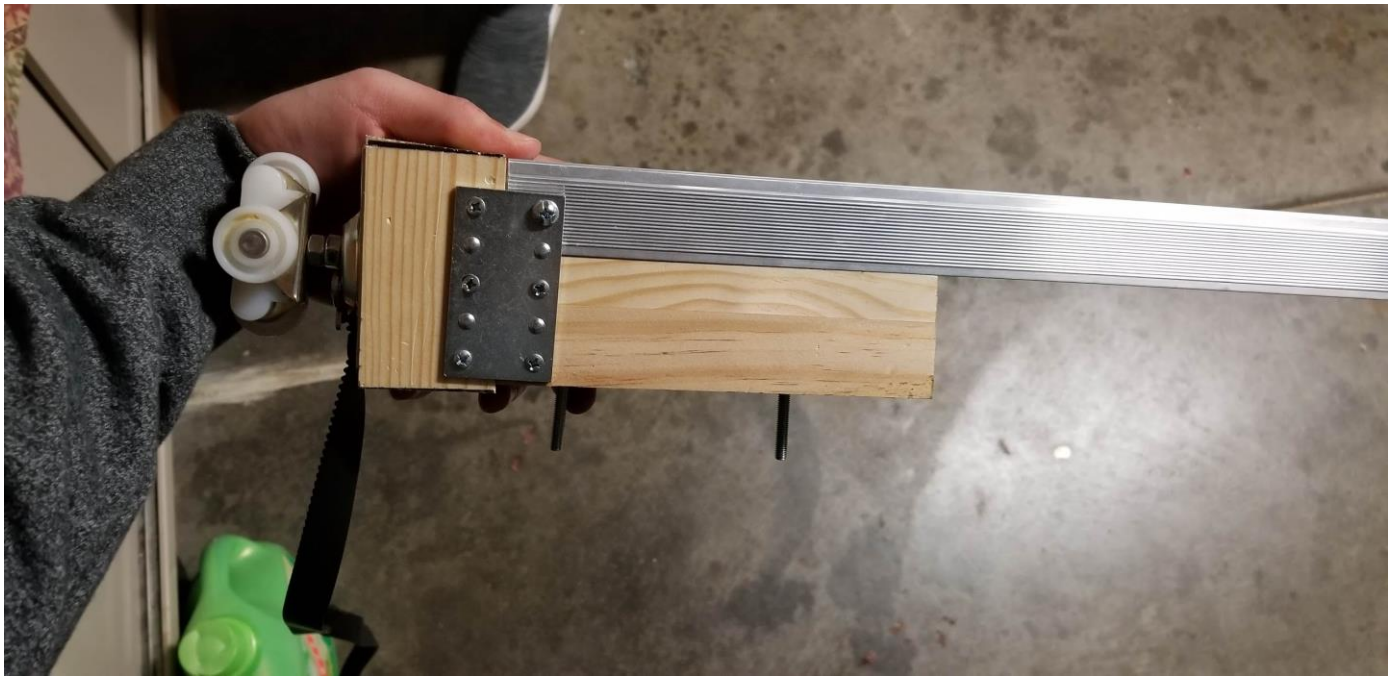


Figure D 6: Mount used to hold Horizontal slider with Vertical slider [52]

Figure D 7: Commercial Grade Pocket/Sliding Door Hardware [52]

Figure D 8: Checkout Window (in-between view) [52]

Figure D 9: Checkout Window (front view) [52]

Figure D 10: Front view for user [52]

APPENDIX E. VENDOR CONTACTS

On behalf of the ABS Team, we would like to thank Professor Neal Levine, from the Electrical Engineering Department at CSU Sacramento, for providing technical and professional assistance to our team in completing this project. His valuable insights into planning and implementing projects helped us maneuver through completing this project in unfortunate times of pandemic.

APPENDIX F. PROJECT TIMELINE

Autonomous Book Solution

Senior Design

Milestone Description	Start	End Date	Days
Website			
Dashboard	2/5/2021	3/1/2021	24
Book Profile	3/1/2021	3/14/2021	13
User Account	2/10/2021	3/7/2021	25
Search Bar	2/25/2021	3/15/2021	18
Book Recommendations	2/22/2021	4/5/2021	42
Checkout Cart	3/8/2021	4/10/2021	33
QR Code Generator	3/20/2021	4/4/2021	15
			0
Hardware			0
QR Code Scanning	10/1/2020	10/16/2020	15
Vertical and Horizontal	10/12/2020	11/8/2020	27
Book Tray	10/29/2020	11/12/2020	14
Cells	9/28/2020	11/1/2020	34
Landing Confirmation	11/2/2020	11/15/2020	13
Checkout Window	11/2/2020	11/12/2020	10
Shelf	10/5/2020	10/7/2020	2
Testing and Integration			
Hardware Feature Testing and Integration	11/15/2020	11/22/2020	7
	12/28/2020	1/14/2021	17
Software Feature Testing	4/1/2021	4/16/2021	15
Full Project Testing and Integrtion	4/16/2021	4/24/2021	8

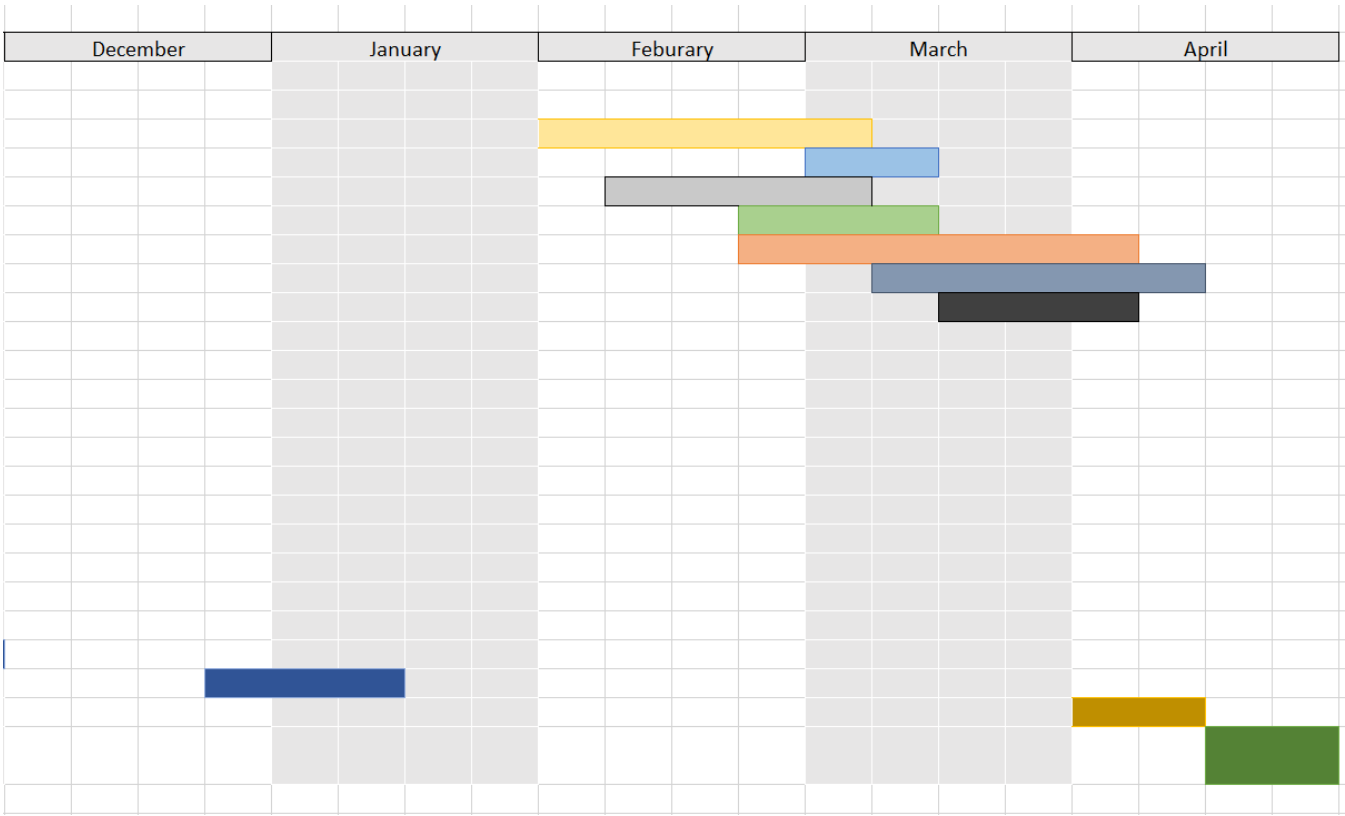
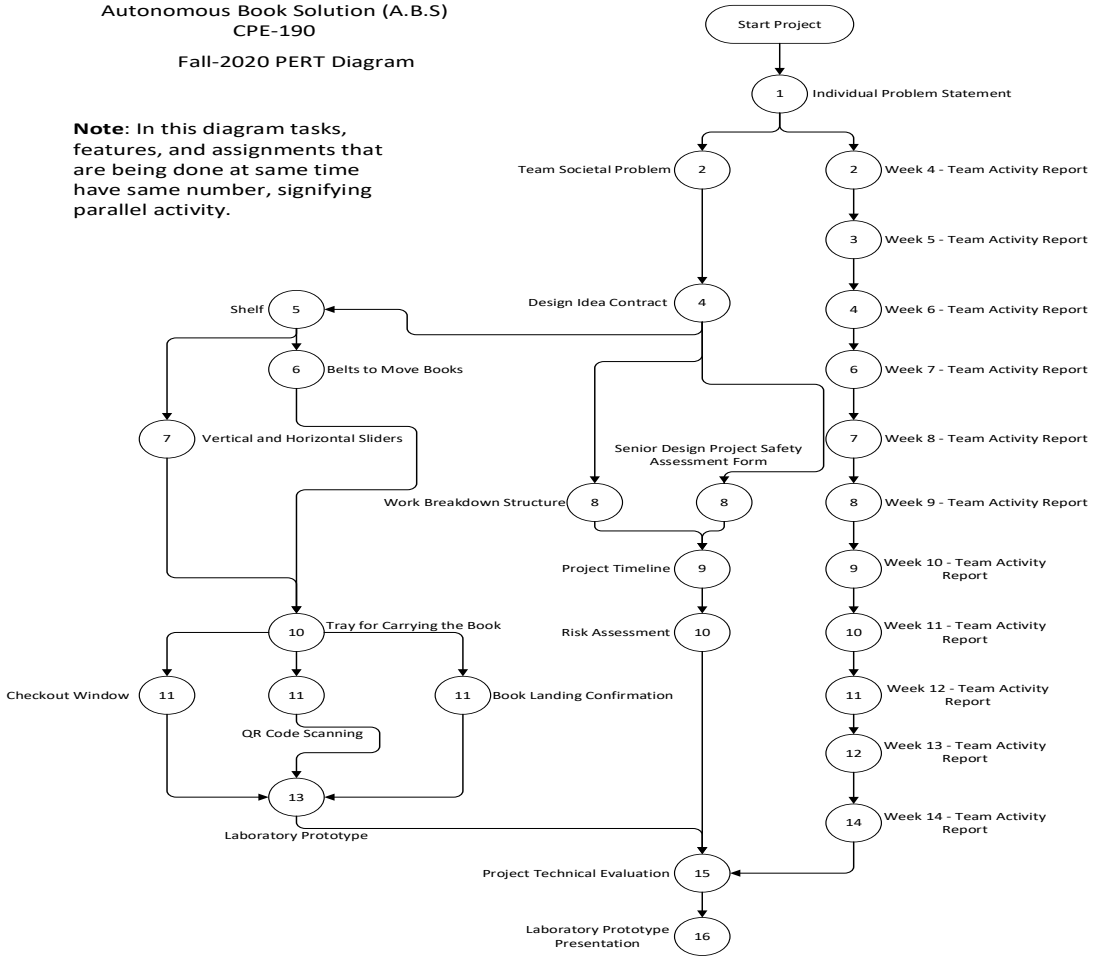


Figure F 1: Project Timeline [53]

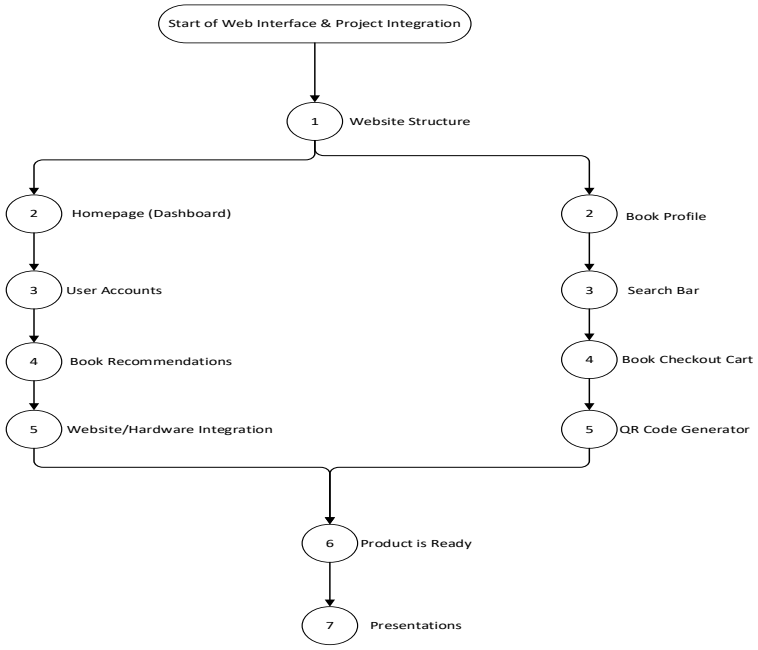
Autonomous Book Solution (A.B.S)
CPE-190

Fall-2020 PERT Diagram

Note: In this diagram tasks, features, and assignments that are being done at same time have same number, signifying parallel activity.



Spring-2021 PERT Diagram



Emmanuel Silva Avina

Objective

Actively seeking an internship in the areas of Hardware, Firmware or Software Engineering.

Education

BACHELOR OF SCIENCE | CALIFORNIA STATE UNIVERSITY, SACRAMENTO CA

Electrical & Electronic Engineering Expected: May 2021

- Electronics 1 & 2: Op-Amps, MOSFTETS, BJTs
- Modern Communication Systems
- Robotics: Principles of robotics and design of robots
- Feedback Systems
- Microprocessors: Computer Interfacing
- Applied Electromagnetics

Skills-Languages, Tools, Platforms

- C and Python Programming, MATLAB, OrCAD PSpice, Attollic TrueStudio, STM32CUBEMC, VMWare, Oscilloscope, Signal Generator, FPGA, x86 Assembly, PCB Design, Analog Circuits, Arduino IDE, Multisim, MS-DOS, Raspberry Pi, Arduino, Analog Discovery, Microsoft Office

Project Experience

SENIOR DESIGN PROJECT (IN PROGRESS)

- *Autonomous Book Solution (A.B.S)*: Collaborating with 3 other Computer Engineering students to design and build an autonomous book machine that will have a website, user-friendly, to order, scan and retrieve the book without physical interactions. This will allow the current pandemic status for libraries to continue dispensing books. Directly working with the electrical and mechanical components of the machine which will dispense these books by creating the schematic blueprint for the power and component connection to the microcontroller interface.

ROBOTICS PROJECT (IN PROGRESS)

- Constructing a RC micro-controller-controlled robot that will follow a predetermined path, and follow a path using QTI sensor through Arduino IDE programming.

HARDWARE DESIGN

- Designing a Common Emitter BJT: Designing a CE BJT with specifications to meet a Voltage gain, Input Resistance, Minimum swing capability and a desired power supply voltage range with OrCAD PSpice

COMPUTER INTERFACING PROJECT

- *El Weather Station*: A system of sensors that will detect that ambient readings such as temperature, pressure, humidity, Visibility, Infrared, and Ultraviolet using the Raspberry Pi as its platform. Using Python to code to program a switch button to activate the sensors and send and send email, text message, and screen display of such readings with date and time.

Awards/Clubs

- Dean's Honor List
- Dedicated to Educating, Graduating, and Retaining Educational Equity Students Project (DEGREES Project)

Sajanpreet Malhi

EDUCATION

Bachelor of Science, Computer Engineering
California State University, Sacramento, California Expected May 2021

Relevant Coursework: Advanced Logic Design, Computer Interfacing, Data Structure & Algorithm Analysis, Program Concepts & Methods II, Computer Hardware Design, Circuit Analysis, Computer Network & Internet, Adv Computer Organization, Signals and Systems, CMOS & VLSI (IP), Operating System Principles (IP)

Skills: Languages: Java, C, UNIX, HTML & CSS, Verilog, VHDL, Python; Pytest Framework, Git, Gerrit, SCRUM and Agile process, CI/CD; Microsoft Office: Excel, SharePoint; Fluent in Punjabi, Hindi, and English

WORK EXPERIENCE

Aruba, HPE, Roseville, California ***System Validation Engineer Intern, May 2020 – August 2020***

- Designed and executed level one and two system validation automated tests to validate a cloud-based network management platform used to manage wireless and wired devices.
- Participated in projects using the Agile software development framework.
- Performed software test development using pytests for scripting and Gerrit for code reviews to improve time efficiency.
- Ensured tests passed in the Continuous Integration pipeline to improve testing efficiency.
- Delivered all responsibilities in a fully remote work environment to meet deliverable goals on time

CSU Sacramento, Sacramento, California ***ECS IT Consulting/Lab Assistant, January 2020 – Present***

- Work with the College of Engineering & Computer Science (ECS) Computing Services to provide technical assistance to staff, faculty, and students.
- Setup, deploy, and troubleshoot a wide range of hardware and software issues including terminal servers, web development, faculty/staff computer systems, lab computer systems, classroom presentation systems, printers, scanners, computers, etc.

SMUD, Sacramento, California ***SAP Security, Student Assistant, May 2019 – January 2020***

- Created User IDs and roles in SAP (Systems, Applications and Products) and AMI (Advanced Metering Infrastructure) applications that helped the 90 percent of SMUD users to collect data in a timely manner and allowed for efficiency in SMUD transactions by 60 percent
- Assigned/unassigned access in SAP and AMI environments to secure data in different environments
- Managed User IDs in SAP and AMI environments to assure proper functionalities of the IDs
- Transported data across systems and maintained daily datasheets of user maintenance
- Collaborated with functional analyst to discuss requirements and support security testing

PROJECTS

Aruba CX Switches Neighbor Information

- Designed and programmed system validation automated test using REST and Python to validate the proper functionality of link layer discovery protocol to display neighbor devices on Aruba CX Switches interface and Aruba Central, a cloud-based network management platform.

SOD (Segregation of Duties) Project

- Collaborated with the functional analyst team to discuss the required access for different departments. In this project, I created roles for multiple departments with segregated transaction codes to avoid overlapping access. I also created test IDs to test these roles in the SAP development system. This project was completed in August 2019 and resulted in new and modified SAP access for every department in SMUD.

Wi-Fi Controlled Car

- As a team manager, directed a team in programming and developing a wi-fi controlled car using raspberry pi, infrared sensors, motors, and cameras. Programmed and set up TCP communication between the car and the computer to allow the transfer of data and information between the client and the host. Also assisted team members in installing sensors, motors controls and wiring on the car.

Gradebook

- Designed an interactive GUI Java program to maintain a binary file of student records for a class. A student record in the file should include ID number, name, gender, class level, age, and lab scores. The user should be able to create a new class, load students from a file, add a new student, view/delete a student, enter lab scores and backup students from a file.

AWARDS/AFFILIATIONS

- Member- (Mesa Engineering Program) MEP, IEEE and Sikh Student Association, Treasurer- Heeray Punjab De (Punjabi Folk Dance Club at CSUS), College of ECS Dean's Honor Roll (Every Semester), Golden Key International Honor Society (CSUS Fall 2018), Seal of Biliteracy (2017), Certificate of Recognition from California State Senate (2016), Aruba Networking Essentials Certificate (2020)

Farooq Alaulddin

ɔUCATION

Bachelor of Science: Computer Engineering, California State University, Sacramento (GPA: 3.5)
Associations: Dean's Honors List (2019, 2020), Tau Beta Pi, Engineer Honor Society.

ECHNICAL SKILLS

Programming: HTML, CSS, JavaScript, MySQL, PHP, Python, Java, Bash/Shell, C/C++, C#, Assembly

Engineering: Verilog, VHDL, Pspice, Cadence, Quartus, FPGA, MatLab, Microcontrollers.

Libraries/Frameworks: React, ReactNative, Angular, Laravel, Spring, JQuery, Bootstrap, Material & Semantic UI, Git & GitHub.

Academic: Data Structure, Adv Algorithms**,** Computer Network & Interfacing, Ad Logic Design, OS Pragmatic, Adv Electronics & Power, Adv Signals

XPERIENCE

CALTRANS (UEI). Sacramento, CA.
Software Engineer/Student Assistant *August 2019 – Present*

- Assist in developing, configuring, and customizing IT solutions to help the Division of Construction transition from doing business in a paper driven system to an electronic document management system or data gathering system.
- Applications I helped design are being used by many districts throughout California and expected to increase work-efficiency by 50% in respect to old methods.
- **Projects I worked on:**

Project Management System *(2020) - Caltrans*	Written based on LAMP Stack & MVC. This app helps engineers track the progress of their assigned projects. Involves CRUD operations, a real-time connection to live databases, multi-level user functionalities, and JWT authentication. Built using HTML/CSS, JavaScript, Jquery, bootstrap, PHP, Python, MySQL, MVC.
Admin Portal *(2020) - Caltrans*	Written using NodeJS/ReactJS framework, JavaScript & TypeScript, Bootstrap & Material UI, REST APIs, Python, Oracle db. This platform is used alongside the Management system to provide more administration features to managers.
Fast Contract Estimator *(2019) - Caltrans*	Built using HTML, CSS, Bootstrap, JQuery, and JavaScript. A multi-platform web application that estimates project costs based on given inputs combined with standard Caltrans rates.

GoSecure. Sacramento, CA.
Intern/network Support *May 2019 – August 2019*

- Daily tasks are identifying root causes of operational issues/process inefficiencies, using diagnostic testing software and equipment, and providing feedback on project related activities using Jira.
- Tasks range from network engineering and troubleshooting, to data cabling and systems administration.

HER PROJECTS

Project Slate OS *Academic - (2020)*	Written in C & Assembly. A multi-process OS built on top of the Intel x86 CPU architecture using SPEDE development environment. Functions: Kernel, and the supporting drivers, interfaces, and system services that make the Kernel. User processes to exercise the Kernel, drivers, interfaces, and system calls.
32 bit- 5-Stage Pipelined Processor *Academic - (2019)*	Written and simulated using Quartus & Verilog. This 5-stage pipelined processor can handle instructions: Arithmetic operations (add, subtract, add immediate), Data transfer instructions, Logical operations, Branch instructions.

COMPLISHMENTS

Leader, Autonomous Book Solution, Senior Project Team, Sacramento, Ca, 2020-2021

Khalid M. Shah

OBJECTIVE:
To obtain Student Assistant or intern position and use my knowledge and experience to assist staff and faculty along with gaining more professional skills and experience myself.

EDUCATION: Bachelor of Science in Computer Engineering, California State University, Sacramento

SKILLS: **Languages:** **Speak**: Urdu, English, Hindko/Punjabi

Read & Write: Urdu and English

Computer: Adobe Photoshop/Illustrator, Java, Python, PHP, Basic SQL, html5, css, X86 Assembly, C, Junits, Circuitry, MS Office, Spiceworks, SharePoint, Agile Central, Service Now, RAD, GIT, and social media (Facebook, Instagram etc…)

CLASSES:

- CSC 139: Operating System Principles
- CPE 190: Senior Design Project
- CPE 151: Cmos And Vlsi
- ENGR 120: Probability+Random Signal
- EEE 180: Signals and Systems
- EEE 108: Electronics 1
- CPE 142: Advance Computer Organization
- EEE 117: Network Analysis
- CSC 134 Database Mgmt Systems
- CPE 138: Computer Network + Internet
- CPE 166: Advanced Logic Design
- CPE 186: Computer Hardware Design
- CPE 185 Computer Interfacing
- CSC 130: Data Structure + Algorithm Analysis: Java
- CPE 64: Introduction to Logic Design
- CSC 60: Introduction System Program Unix/C
- CSC 35: Introduction Computer Architecture
- CSC 28: Discrete Structures
- CSC 20: Program Concept + Method II: Java
- CSC 15: Program Concept + Method I: Java
- ENGR 96A: Interdisciplinary Topics Engineering

EXPERIENCE:

- **DCSS TSD Student Assistant-Applications Development** (June 2019 – June 2020)
  - Track and modify change enhancement records in tools such as Service Now, Agile Central, and ClearQuest, code and resolve issues with existing JUnits, assist in monitoring and ensuring successful passing of JUnit suites, provide support to Application Code Specialist in organizing JUnit cleanup effort;, update development related procedural documentation, such as Onboarding, Code Review, Assessments, etc., track code review metrics and provide report to management, participate in coding system enhancements, and provide general assistance to Application Code Specialist, Section Manager, and App Dev supervisors in section related meetings and activities.
- **ECS Computing Services Student IT Consultant (Lab Assistant)** (January 2018 – May 2019)
  - From programmatic support & office duties (scanning, filing, operating basic office equipment, answering phones, checking voicemails), Create, process, rest, or renew student, classroom or project accounts. Troubleshoot problems/bugs documented/reported in Spiceworks, Deploy and undeployed printers, scanners, computers, and other office tech equipment
- **Full Circle Project (FCP) Scholarship Coordinator** (January 2017 – February 2018)
  - Finding Scholarship opportunities for students, Coordinate scholarships FCP, Nakatani & A Scholarship, maintain scholarship website & electronic inventory, and assist in hosting events.
- **TAP OP (Taking All Possible Opportunities Provided) Founder & Admin** (July 2016 – Present)
  - Posting opportunities on social media on regular basis & Giving feedback on scholarship essays
  - Maintaining Website (tapop.org), Facebook Page, Group, Twitter, Instagram page for TAP OP