SMART BACKPACK / BOOKSHELF

Team 15 - Anusha Kandula (kandula2), Tae Kyung Lee (tlee82), Gautam Putcha (gputcha2)

ECE 445 Design Document – Spring 2020

TA - Jonathan Hoff

Table of Contents

1. Introduction

1.1 Problem Proposition

We are building a smart bookshelf that serves all types of students. Our customer has a problem, and it is carrying a heavy and overstuffed bag on a daily basis because of the weight of several heavy books and in many cases, even the books not required for a particular day. Our product solves the customer's problem with a simple app and bookshelf system that can be used to tell the student what books he/she needs according to their schedule everyday.

1.2 Objective

A problem that students have been facing for a long time has been overstuffed backpacks that are too heavy. Carrying heavy backpacks can not only cause temporary issues, but also chronic back and neck problems later on in a student's life [1]. A study conducted by Inland Empire Spine Center in Riverside, CA showed that 64% of students carrying backpacks reported having back pain at some time, out of which 90% described their pain as "bad" or "very bad" [2]. Although the causes of these problems are clear, students continue to carry overstuffed backpacks to school and put themselves at the risk of developing chronic spine or back issues.

Currently, monitoring the weight of a student's backpack involves being in the presence of a scale in order to measure its weight. Oftentimes, a scale is not readily available which makes collecting measurements tough. Maintaining historical weight logs must also be done manually and can be a pain. To make matters worse, a parent or guardian is not always available to provide feedback and monitoring assistance in order to help a student break bad backpack habits. A smarter backpack system is badly needed to make it easier for students to develop and maintain healthy backpack habits.

1.3 Background

Any “smart” bookshelf designs on the market today have been designed to support a library system and aid in book positions and organization. An example for this is the 3M Smart Bookshelf system using RFID [3]. The smart bookshelf solutions on the market today address different problems but not our specific problem.

The original solution approaches the problem by simply warning the user of their backpack being overweight instead of actually solving the issue at hand. The original solution proposes weighing the bag before use and getting feedback on the weight limits on an IOS application that is bluetooth enabled. It then serves as a fitness application and becomes heavily software based. We tackle the backpack issue by looking at the problem from a completely different angle. Very often, this problem can be solved by just packing the necessary books for a day of classes as this reduces the overall weight of the backpack. Our solution proposes packing “smartly” based on the schedule and avoiding the issue of an overweight backpack. It uses completely different sensors, Wifi instead of Bluetooth, and an Android application which is not the main focus of the project unlike the original solution.

Our design proposes a solution to have a lighter backpack based on a student’s schedule. Repacking a backpack on a daily basis is a task that most students avoid and prefer to suffer the effects of a heavy backpack. Our solution uses the schedule data provided by the user to indicate directly to the user which books to pack on a daily basis to avoid packing any unnecessary books.

1.4 Visual Aid

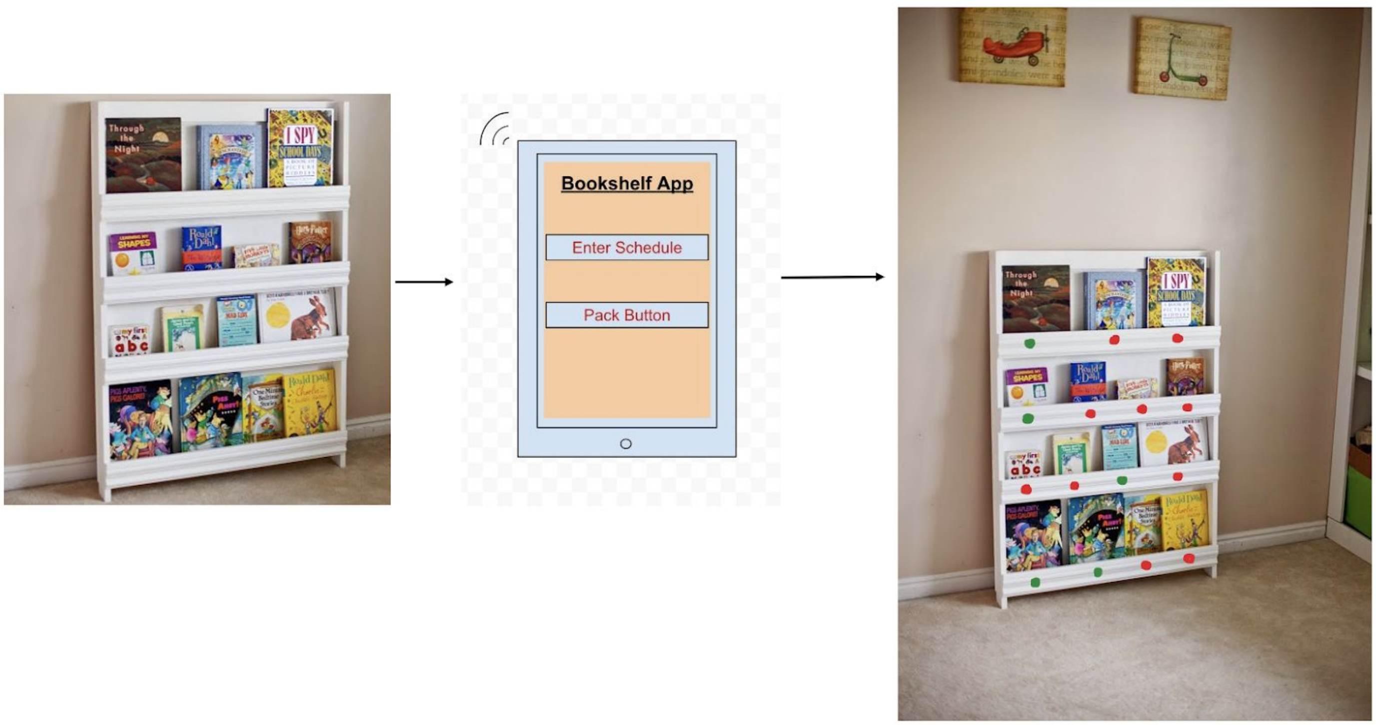

Figure 1: Visual Aid of how the bookshelf will look

1.5 High-level requirements

1) The user needs to enter their schedule and a picture of their textbook on the application and only then the “pack” button is functional and lets the vision system and the LEDs work to match the schedule.
2) The device must light up all 16 LEDs either red or green within 30 seconds after the pack button has been pressed and the LEDs will light up with an accuracy of at least 99% according to the subject data it gets.
3) The application will play an alarm-like sound within 5 seconds if the user takes out the “wrong” book (a book which has a red LED under it).

2. Design

2.1 Block Diagram & Flowchart

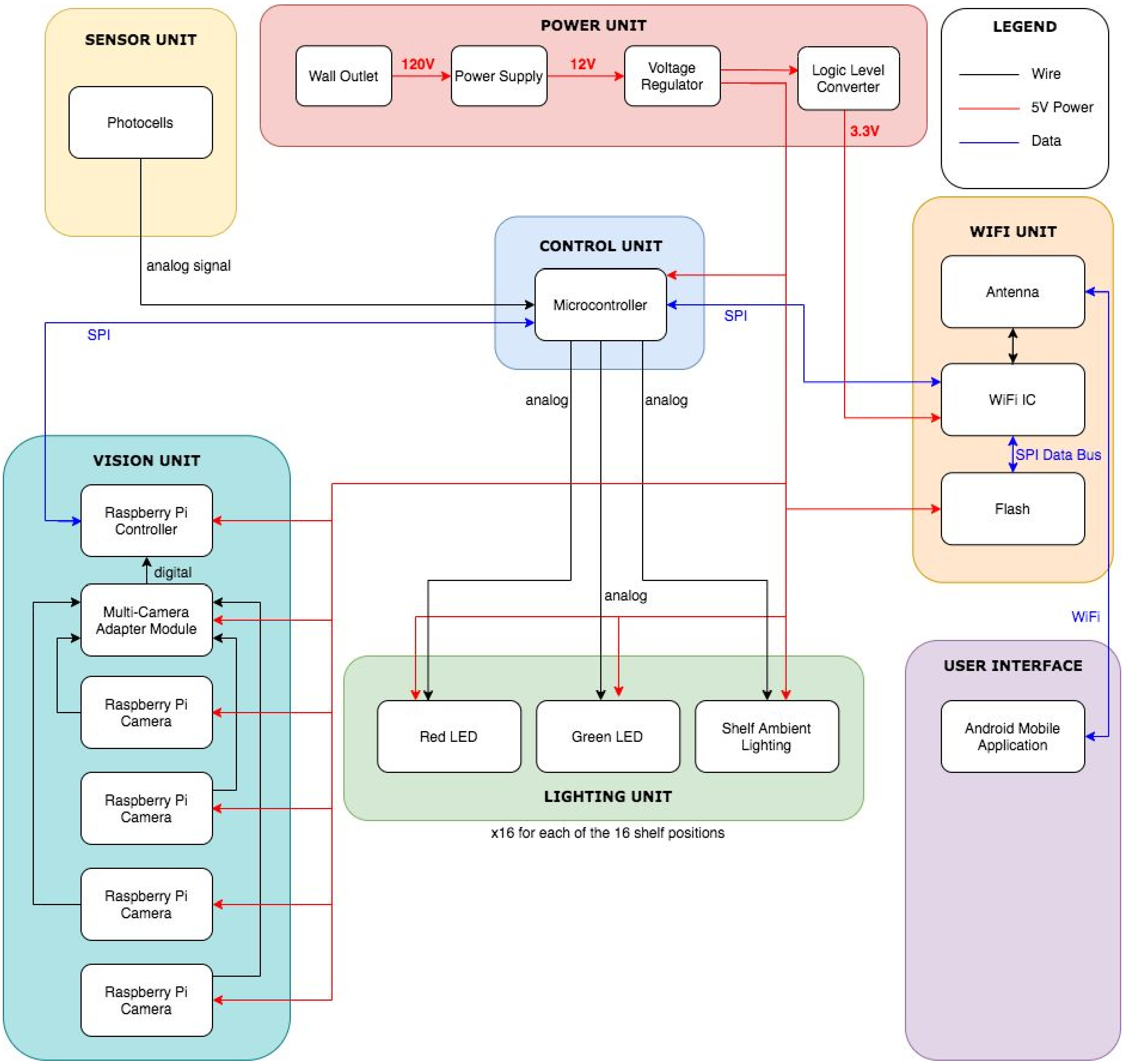

Figure 2: Block Diagram for our proposed design of the smart bookshelf

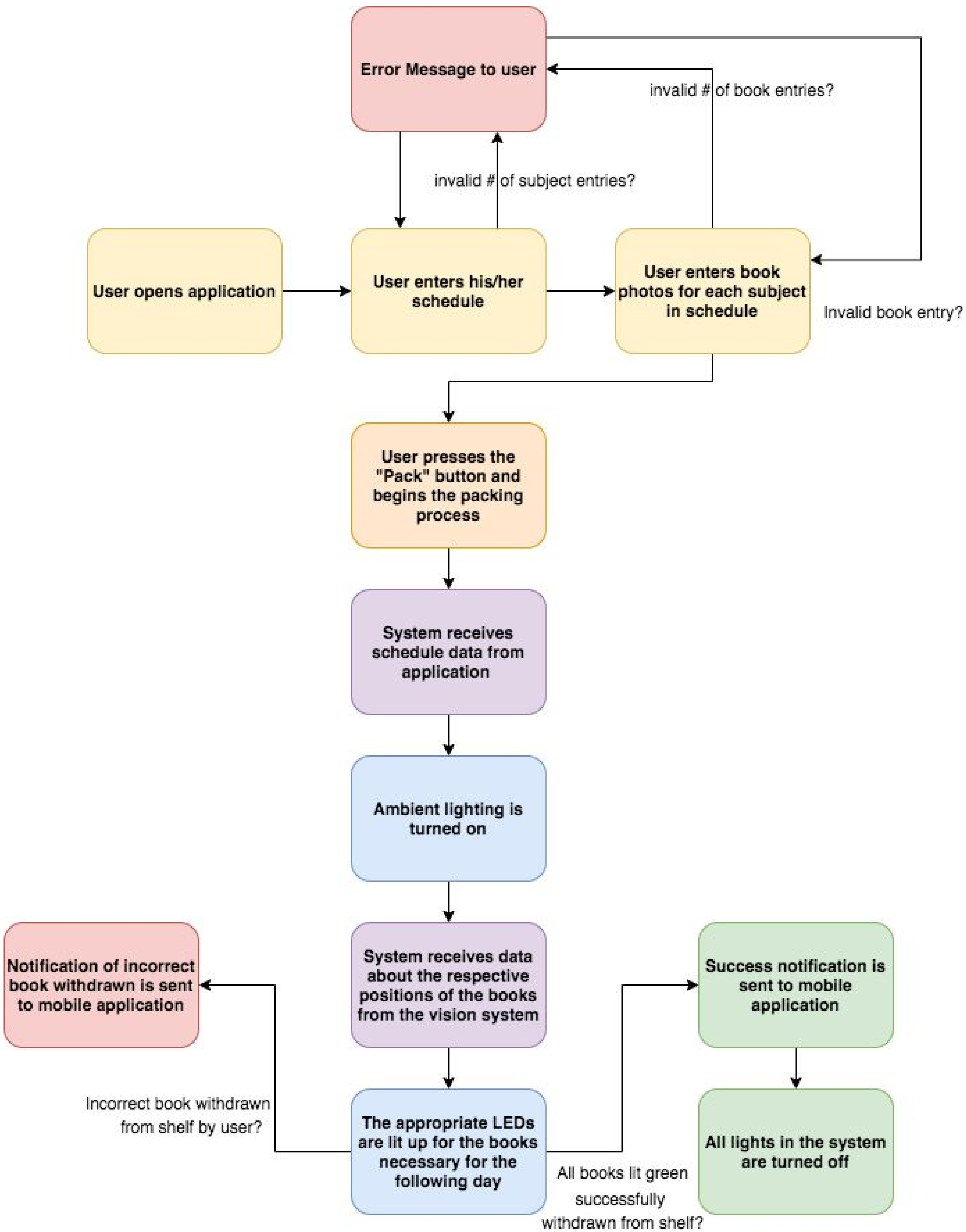

Figure 2: Flow Chart of series of actions

2.2 Physical Design

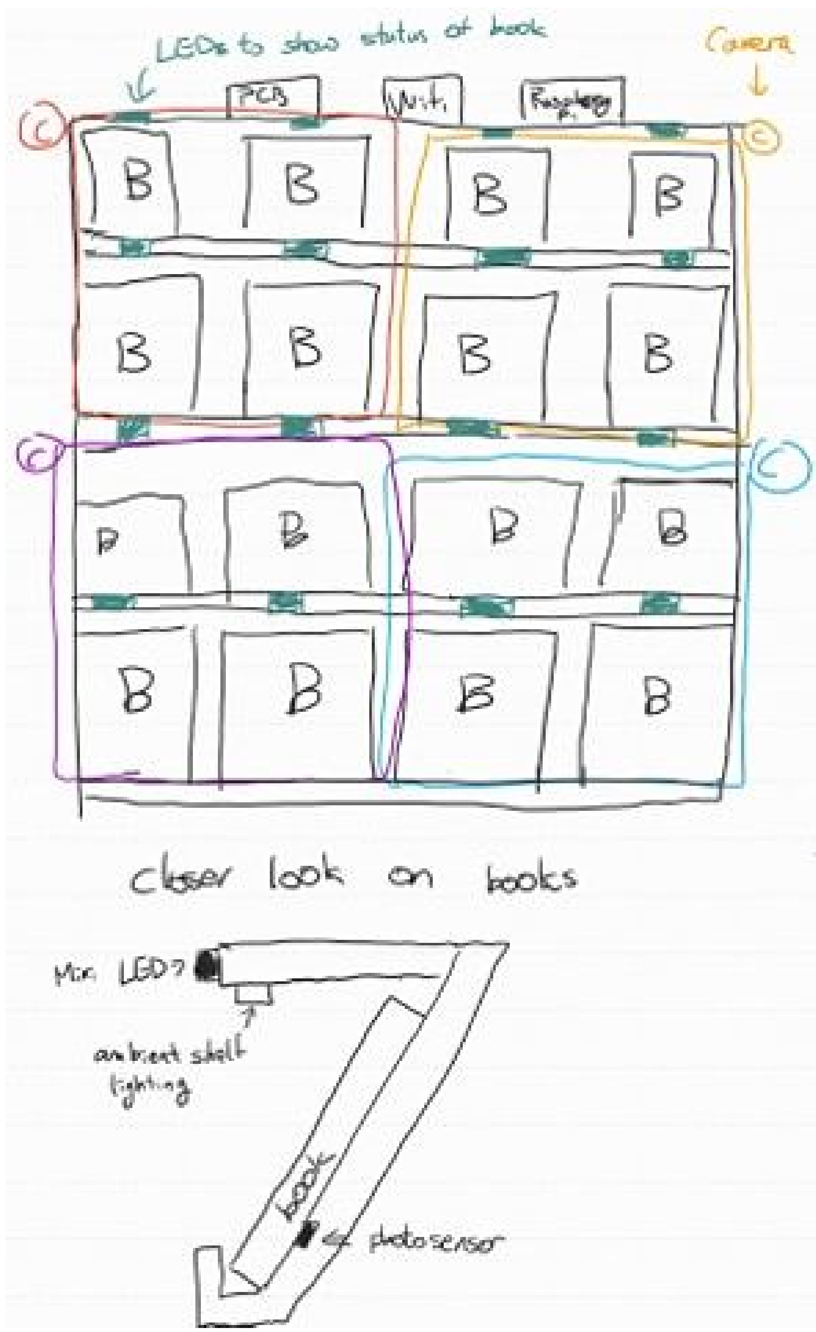

Figure 3: Physical design for our proposed design of the smart bookshelf

Figure 4: example of the book display shelf that we are going to use [4]

2.3 Functional Overview & Requirements / Verifications

2.3.1 Control Unit

The control unit will process the user input received from the user interface, and use the data to illuminate the necessary LEDs. It is responsible for carrying out the entire sequence of events by:

1. Receiving the day's schedule from the android application.
2. Receiving the image data from the cameras and data from the photocells and determining the respective positions of the books on the shelf. (This step is necessary for the system to know what book has been placed where)
3. Based on the schedule data and book positions, illuminating the LEDs for the necessary books with green, and illuminating the LEDs for the books not required red.
4. If an incorrect book is withdrawn from the shelf as indicated by the photocells, sending a notification to the android application.
5. If all of the correct books (labelled with green LEDs) have been withdrawn from the shelf, sending a notification of success to the mobile application.
6. Switching off all lights in the system when the process is complete.

2.3.1.1 Microprocessor

We will use the ATMEGA328P microprocessor. This microprocessor will be a part of our PCB and will be responsible for conducting all of the essential functions of the system. It will also be critical that all of the functions are executed in a sequential manner in the right order. Our microprocessor will also be responsible for communicating with the WiFi module using SPI.

Name	Requirement	Verification
ATMEGA328P microprocessor	1. The microprocessor can process the computations within 1 second. 2. The microprocessor must be able to communicate accurately with the photocells within 1 second. 3. The microprocessor must be able to communicate accurately with the cameras within 1 second.	1. Send dummy data with the same line of codes and calculate them within 1 second. 2. Test inputs from the photocell and verify that the change is reflected in the microprocessor by using a print statement and check if the signal is received within a second. 3. Test inputs from the camera and verify that the change is reflected in the microprocessor by using a print statement and check if the signal is received within a second.

2.3.2 Power Unit

This power unit will provide all of the necessary power requirements for the components of the system such as the Wifi IC, Flash, microcontroller, LEDs, raspberry pi controller & cameras, and the multi-camera adapter module.

Name	Requirement	Verification
120V - 12V AC/DC Converter (Power supply)	1. Power supply provides 12V ± 0.2V when on and provides 0V when off	1. Probe the output of the power adapter with a multimeter. a. When the device is off, the multimeter should read 0V. b. When the device is on, the multimeter should read a value between 11.8 V and 12.2 V.
Linear	The voltage regulator must be able to	1. Provide varying DC currents (8V,

Voltage Regulator	output 5V +/- 0.2V for any input	12V, 16V) using a function generator and rectifier as an input to the regulator. 2. Use a multimeter to verify that the output current remains between 4.8V and 5.2V.
Logic level converter	The logic level converter must convert an input 5V signal into a 3.3 +/- 0.05V signal for use in the WiFi module.	1. Provide a 5V source from a voltage regulator to the converter. 2. Measure the output of the converter using a multimeter and verify that it is 3.25V and 3.35V every time.

2.3.2.1 Power Supply

We are going to use PLT 55-3075-99 step-down transformer. This will step down the standard 120V wall power to 12V that will be connected to the Linear Voltage Regulator for use in our design.

2.3.2.2 5V and 12V Linear Voltage Regulators

5V and 12V LM338T linear regulators will regulate 12V from the power supply, to 5V and 12V respectively which will be used by the microcontroller and input to logic-level converter for 3.3V which will be used by the sensor unit.

2.3.2.3 Logic Level Converter

This converter will be used to step down a 5V signal to a 3.3V signal for the WiFi module. The ESP8266 THING WiFi module we have chosen to use does not have inbuilt 5-3 logic shifting and requires a logic level converter, BSS138, to step down the voltage to the necessary 3.3V.

2.3.3 Sensor Unit

Name	Requirement	Verification
Photocell Sensor	1. The photoresistor must be able to measure within +/-5 lm of the actual light intensity in the bookshelf. 2. The photoresistor should not have a latency of 10 ms or more.	1. Photoresistor is connected to an Arduino with a program that reads the light intensity of multiple light sources of varying luminosity (50lm, 60lm, 70lm, 80lm). These are tested and the output value is compared to the actual light intensity of the source. 2. Photoresistor is connected to an Arduino with a program that checks the latency and we can verify if it is 10 ms or less. [5]

2.3.3.1 Photocell

The photocells are going to be placed inside the shelf facing the books so that we can know the state and position of the books on the shelf. This information of the state of books will go to the processor as analog data but will be converted to digital data in the microprocessor and will be coordinated with computer vision to help identify which books are needed for the day and change the status in the Raspberry Pi and the LEDs. Since there is ambient light, it is guaranteed that the removal of a book would trigger the photocell to inform the processor about the new state of the shelf position.

The photocells are going to be tested with arduino using the following circuitry

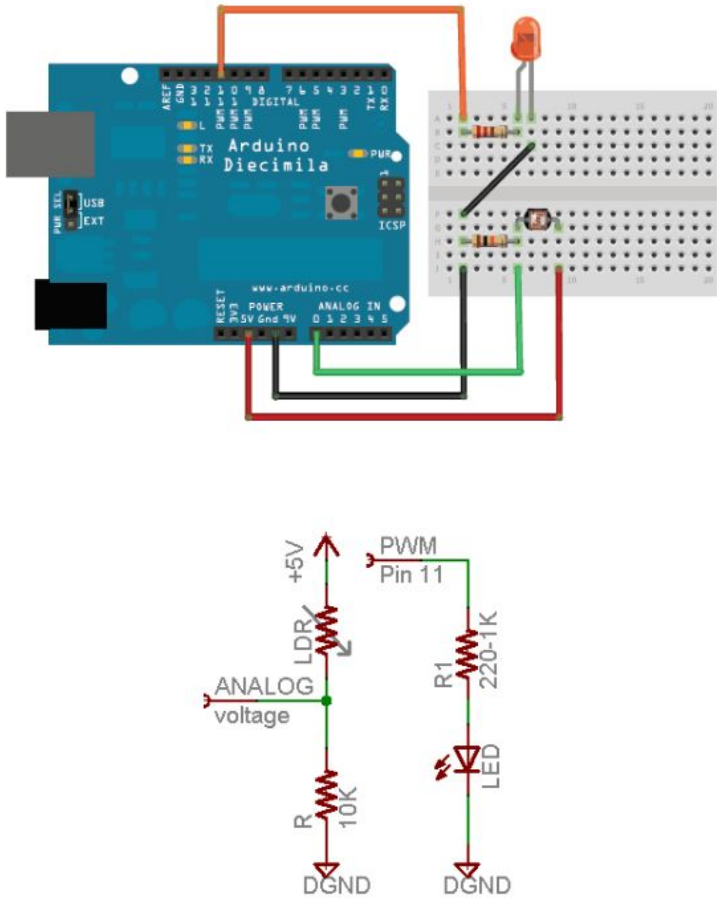

Figure 5: example circuitry of Photocell

Before we install any photocell into the bookshelf, we will build a circuit with our own led and the photocell. This way, we can get data of how bright a single LED is and how effective we can use the photocell with dim lighting. We will test this in a dark room showing LED, an ambient stand light, with or without an ece120 textbook in between the lightsource.

2.3.4 Vision Unit

Name	Requirement	Verification
Raspberry Pi camera system	1. The image data must be processed and the LED outputs must turn on in < 1 minute and 40 seconds. 2. The matching of books to the user's pictures will rely on image recognition and will have an accuracy around 60- 70%.	1. Send dummy image data to raspberry pi and have a program that turns on all LEDs green if the image has been processed. Verify that all the LEDs turn green within 1 minute and 40 seconds every time. 2. Test run raspberry pi and camera with a program that captures the 2 images of the same book and compares them to see if the recognition algorithm accurately identifies that they are the same book. Repeat this 10 times to verify accuracy.

2.3.4.1 Raspberry pi

Raspberry pi will be connected to the ATMEGA328P microprocessor, raspberry pi cameras. This will communicate with ATMEGA328P through SPI. Raspberry pi will be in charge of algorithm for the computer vision and matching the book placed on the shelf with the correct subject by reading the cover of the book.

For this, we are going to use OpenCV, an open source of python, to compare the image data of the textbook that is initially provided by the user using the Structural Similarity Index Measure model.

4 Raspberry Pi cameras (5 MP, 1080p resolution) and a Multi Camera Adapter will be the eyes of Raspberry pi so that we can check which books belong to which subjects. Each of the cameras will be attached to the 4 upper corners of each quarter of the shelf. This way we will get the same viewpoint of the books (upper view) for more accurate comparison.

2.3.5 Lighting Unit

Name	Requirement	Verification
LED	1. All 16 LEDs must light up either red or green within 30 seconds after the "pack" button has been pressed. 2. The LEDs must all go back to turning off (no light) 10 seconds after the user receives a notification that confirms that all the correct books have been taken.	1. Connect the LEDs to an Arduino that runs a program ten times to test if all 16 LEDs light up either red or green each time within 30 seconds. 2. Connect the LEDs to an Arduino that runs a program to test if all 16 LEDs go black after a notification signal is given through the mobile application.
Ambient Lighting	1. The ambient lighting must turn off within 3 seconds after the user receives a notification that confirms that all the correct books have been taken. 2. The ambient light must provide sufficient light for the camera unit to effectively recognize the books placed on the shelf.	1. Connect the ambient lighting to an Arduino that runs a program to test if it goes black after a notification signal is given through the mobile application in 3 seconds.[5] 2. Photo resistor connected to an arduino verifies if the light provided must be at least 300 lux or 30 footcandles.

2.3.4.1 LEDs

We are going to use an RGB Diffused common cathode to determine the status of the books on the shelf. Signals to these will be provided by our microprocessor to indicate which shelf has which books and need to be taken out. If the book on the corresponding shelf needs to be taken out, it will flash green. If the incorrect book is taken out from the shelf, it will flash red. This signal will be determined according to the schedule of the day derived from the application. The lights will turn on when the "pack" signal is received from the user and turn off when packing is complete as determined by the processing unit.

2.3.4.2 Ambient Shelf Lighting

We are going to use simple LED ambient lighting above the books when the “Pack” signal is issued by the user. This will allow the Raspberry Pi cameras to be able to identify the covers of the books. When the packing is determined to be done by the processing unit, the lights will turn off.

2.3.5 Wifi Unit

Data must be transferred between the user and the control module via SPI to be accessed on a WiFi network. Wifi SOC (System-on-a-Chip) operates based on an SPI flash program memory and uses an antenna to communicate with the user.

Requirement	Verification
1. Must be connected with 5Mbps access at 20m without obstruction / 5m with obstruction between the router and the module. 2. Must successfully be able to relay information such as number of cups of water/rice and start signal from the application to the microprocessor.	1. Check if ESP-12E Wi-fi module that will be connected to the Arduino with its PCB antenna can be connected through open space, with walls, and with multiple devices that use wifi in range. 2. Send dummy data input to the WiFi module to store into Flash. Then check if data write has occurred by checking Flash.

2.3.5.1 Antenna

Molex 1462200200 PCB trace antenna will be attached to the Wifi IC to gain maximum range. We will aim for 5Mbps access at 20m. This is within the specifications of the product and will allow the user to connect with the product within the boundaries of a household.

2.3.5.2 WiFi IC

ESP8266 THING Wifi Module is chosen for our project because it is very cost-effective and efficient for our communication. Our product does not require speedy transactions between the user so a cheaper model is used. It can operate at 160MHz when overclocked and has integrated power management units and Wifi transceivers. This module will communicate with the microprocessor via SPI.

2.3.5.3 Flash

ESP8266 THING Wifi module comes with a limited RAM memory of 12kb for user programs and variables/data. A Flash IC will be used, if necessary, to hold the program memory for the WiFi IC. This must operate at 80MHz for the WiFi microcontroller to operate at full speed. Currently, we are not certain of our program size for the microcontroller. We will prototype the size of 1Mb Flash IC and downsize for cost measures.

2.3.6 User Interface

Requirement	Verification
1. The application must have a “schedule” button for their user to input their schedule and take photos of their books. 2. A limit of 16 books must be set on the maximum amount of pictures allowed by the user. 3. A limit of 2 subjects per book should be set as the maximum number of subjects for one book. 4. The application must send a notification 10 seconds after all the correct books have been taken out. 5. The application must send a notification within 10 seconds if an incorrect book has been withdrawn from the shelf.	1. Check if the application opens and loads within 10 seconds. Ensure that choosing the schedule option allows you to enter a schedule along with the associated book photos for each subject. 2. Enter an input that is over the limit for the amount of books and the number of subjects per book to verify that the application throws an error. 3. Press the pack button and take out the correct books to verify if the application takes 10 seconds to send out a notification. 4. Withdraw a book labelled with a red LED from the shelf and check that a notification is sent to the application within 10 seconds.

2.3.6.1 Android Mobile Application

We will use an Android application to get the user’s input the schedule of the week. The application will have a “Pack” button to begin the process. The application will also have a section to enter the schedule of the user to facilitate the packing process. When the correct textbooks have not been taken out of the shelf, the system will send an alarm to the mobile application providing a warning to check your backpack, and return the unnecessary weight to the shelf. This application will be sending data to the wifi module which will in turn relay the information to the processing unit.

2.4 Tolerance Analysis

The most critical feature of our project is being able to correctly compare the image of the textbook taken from raspberry pi and the image provided by the user. We are going to use the OpenCV library of python. Using this, we are going to convert our original image from the BGR color space to grayscale using the code COLOR_BGR2GRAY. Then there are 2 models that we can compare the contrast of 2 pictures: Mean Squared Error and Structural Similarity Index Measure.

We will first look at the Mean Squared Error equation.

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

Equation 1: Mean Squared Error

Using the values of grayscale in every pixel of the pictures, we are going to add all differences in the pixels. In Mean Squared Error, 0 means that the following images are exactly the same. Then we are going to associate the picture of the lowest MSE score to be the most similar picture. [6]

Structural Similarity Index Measure is calculated as follows.

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Equation 2: Structural Similarity Index

Structural Similarity Index Measure does not compare the difference in error but it attempts to model the perceived change in the structural information of the image. Instead of comparing the grayscale values of the pixels, the SSIM takes into account of the (x,y) location of the N x N window in each image, the mean of the pixel intensities in the x and y direction, the variance of intensities in the x and y direction and covariance. [6]

So this blog conducted a test on 3 photos. The main photo, a slightly increased contrast version of the photo and a photoshopped testcase.[6] After using two equations to see which of the equation shows the increased contrast photo as more similar, it was determined that SSIM is slower than the MSE in runtime but a lot more accurate. As a result, we are going to use the Structural Similarity Index Measure model for our case.

Another factor that is important in our project is the photocells. The functionality of the photocells is to check whether a book is located in a particular space in the bookshelf, so that we have an indicator if the book is still placed in the bookshelf or removed. This is based on the logic that a textbook will create a shadow if it is located behind the textbook. The basic mechanism of the photocell is as follows: the greater amount of lux the photocell is exposed to, the lower the resistance of the photocell decreases to.

However there are variances that can occur due to the wavelength of the light. Sensitivity of a photocell varies with the light wavelength. If the wavelength is outside a certain range, its effect on the resistance of the device changes greatly. In some ranges such as infrared, heat can build up in the photocell that can alter the functionality of a photocell altogether. [7]

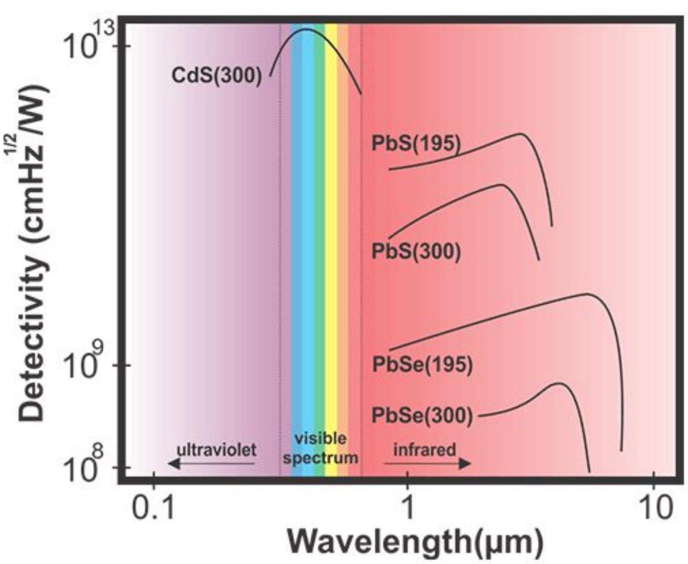

Figure 6: Photocell's detectability according to light's wavelength

In our case, we are going to use the wavelength of the visible spectrum but we would have to be careful that this bookshelf is not exposed to sunlight's infrared lights by being too closely located to the window. [7]

3. Project Differences

3.1 Overview

The original solution approaches the problem by simply warning the user of their backpack being overweight instead of actually solving the issue at hand. The original solution proposes weighing the bag before use and getting feedback on the weight limits on an IOS application that is bluetooth enabled. This design used strain gauges on the handle of the bag to allow the user to measure the backpack by picking it up. While this solution does indicate that the bag is too heavy, it does not provide the user with the tools to remove the unnecessary contents of the bag and repack effectively. We tackle the backpack issue by looking at the problem from a completely different angle. Our solution proposes packing “smartly” based on the schedule and avoiding the issue of an overweight backpack using a smart packing strategy.

Our design proposes a solution to have a lighter backpack based on a student’s schedule. Repacking a backpack on a daily basis is a task that most students avoid and prefer to suffer the effects of a heavy backpack. Our solution uses the schedule data provided by the user to indicate directly to the user which books to pack on a daily basis to avoid packing any unnecessary books.

While there are some engineering tradeoffs involved, our design provides a more effective and sensible solution to the problem at hand. More specifically, our design is a larger design and takes up more space, but is a cheaper solution and after repacking time is taken into account, is a more efficient design as well. It is often the case that in spite of knowing that a bag is too heavy, students are too lazy or choose not to repack their bags and continue to carry very heavy bags [8]. The previous solution does not take this issue into account while our solution does.

3.2 Analysis

The original Spring 2015 solution to the heavy backpack problem weighs the backpack in order to gauge if the weight is too much for a student's back through an IOS application. Our smart bookshelf provides a way to take the minimal books needed through an Android application so that the weight of the backpack is not harmful to the students. Our design has operational improvements over the original design in that it can function at a higher speed as well as achieve correctness in packing the right books. The original solution could tell if the bag is overweight but could not help the student take out the right books in order to repack his/her bag.

In this analysis, we will quantify and calculate the operational time of both projects as well as the correctness and accuracy in each for one student.

Original Project: This project's operations take the following steps:

1) Pack the bag (~10 minutes)
2) Weigh the bag and check IOS application (~0.1 seconds)
3) Repacks the bag (~10 minutes)
4) Weigh the bag and check IOS application again (~0.1 seconds)

<u>Accuracy</u>

The accuracy of packing in this system is not as high as our system. While it does provide the user with the notification that the bag is too heavy, as we have seen, several students choose not to repack. Even after repacking, the accuracy cannot be guaranteed and is based simply on what the user chooses to put in his/her bag. The accuracy of our system is much better since the correct books are chosen based on the LEDs and packed.

Time Taken

Now, the average packing time for a student is varied so let us assume for this one student it takes 10 minutes to pack their bag. The average time to wait for the data to process and check for a notification on your phone for this project is not specified in their final document so assume that it takes 0.1 - 1 seconds. The student would take less than a minute to check their phone. So, the minimum total time spent here is around **20 minutes**.

Our Project:

This project's operations take the following steps:

1) Pack the bag (~10 minutes)
2) Check Android application for the notification that it is complete (~ 1 second)

Accuracy

Our solution also proves to be more accurate since we are able to take only the books that are necessary using our LED labelling system. With our system, the user can pack all of the correct books based on their schedule. Our system proves to be more accurate because the old project checks only for the weight of the backpack and the student can re-pack the wrong books right after. The original device has no way of checking if the student packed accurately or not. So, **our project not only solves the problem of heaviness but it solves the problem of accurate packing as well.**

Time Taken

The average packing time for this student would be 10 minutes as well as it is the same student so we use the same assumption. The time to wait for the data to process and check for a notification on your phone for this project according to our requirements for our photocell and processor is 1 second. The student would take less than a minute to check their phone. So, the total time spent here is at most **11 minutes**. Hence, our project is almost twice as fast as the original project.

Cost

Finally, another aspect is cost and our project is a cheaper option for a more accurate and faster solution. Our total cost is $12,882.73 but their final cost comes out to be $31,652.49 which is more than double the cost of ours.

4. Schedule & Cost

4.1 Schedule

Week	Anusha	TK	Gautam
1	Work on Circuit Schematic for PCB Design	Work on footprints and making the actual PCB	Ordering Initial Parts
2	Talk to the Machine Shop again and begin application development	Test the photocell with small and large light changes, help in algorithm implementation, set up power unit	Work on setting up vision system, connecting cameras and begin image algorithm implementation
3	Integrate and test	Integrate photocell	Set up LED and

	vision system (machine shop)	onto the project (machine shop)	ambient lighting system with shelf
4	Start unit testing the control unit/photocell	Start unit testing the control unit/vision unit - cameras	Start unit testing the control unit/vision unit and lighting system
5	Make 2nd PCB Design if necessary + Order new/extra parts	Unit testing the control unit/photocell and wifi unit, continue work on mobile application	Integrate wifi unit and control unit, continue mobile application development
6	Work on integrating software unit with hardware unit	Debugging necessary for vision unit	Work on integrating software unit with hardware unit
7	Work on integrating software unit with hardware unit	Debugging	Debugging
8	Final Report	Final Report	Final Report

4.2 Cost

	Model	Quantity	Price/Unit	Total Price
Photocell	CDS PHOTORESISTOR	20	$ 0.95	$ 19.00
Camera	Raspberry Pi Camera Module v2	4	$ 25.00	$ 100.00
120V - 12V AC/DC Converter (Power supply)	PLT 55-3075-99	1	$ 15.95	$ 15.95
Linear Voltage Regulator (5V)	LM338T	2	$ 2.10	$ 4.20

Linear Voltage Regulator (12V)	LM338T	2	$ 2.10	$ 4.20
Logic level converter	BSS138	2	$ 2.95	$ 5.90
Camera Adapter	Arducam Multi Camera Adapter Module	1	$ 49.99	$ 49.99
Microprocessor	ATMEGA328P-PU	3	$2.08	$ 6.24
LED	COM 11120	16	$ 1.05	$ 16.80
Ambient Lighting	LED Super Bright White	25	$ 0.34	$ 8.50
WiFi IC	ESP8266 THING	1	$ 16.95	$ 16.95
Computer Vision Control Unit	Raspberry Pi	1	$ 35.00	$ 35.00
Resistors	N/A	10	Free (445 Inventory)	Free (445 Inventory)
Transistors	N/A	5	Free (445 Inventory)	Free (445 Inventory)
Capacitors	N/A	5	Free (445 Inventory)	Free (445 Inventory)

Labor Costs:

Labor = 3 members * 8 weeks * 15 hrs/wk * $35.00/hr = $12,600

Sum of Costs:

Cost of Parts + Cost of Labor: 282.73 + 15,750 = $12,882.73

5. Ethics & Safety

5.1 Ethics

We would like to build a system that is accurate in finding the assignment of books to LED. Even though we would like the vision system to get the exact book assignments perfectly, it is one of our high-level requirements that we aim for an accuracy of ninety eight percent or above. We did this to be realistic and not lie about the efficiency of our product. We are abiding by the [9] IEEE Code of Ethics #3 by doing this. To alarm the users of which books to use, we use colored LED indicators: Green for correct behavior and red for incorrect user behavior. According to IEC 60601-1-8 standards [9], in the collateral standards, test and guidance for alarm systems in equipment is necessary using colors of indicator lights. Red indicates that immediate user intervention is required or used in dangerous situations and green indicates normal situations and equipment is ready to be used. We will be following the IEC 60601-1-8 collateral alarm standard and implementing the alarm system into the smart bookshelf.

5.2 Safety

We plan to address the safety concerns with a few precautions so that our users are not afraid to use our product. We would want the alarm system on the phone to not be louder than 70 decibels as that is the limit to safe listening according to the World Health Organization [10].

Another safety issue is that in domestic situations that this bookshelf could be damaged by the spilling of water or other liquids and that could cause a short circuit and ruin the components [11]. This is a risk and adding a rubber casing for some of the electrical components of the device such as the pcb, wifi, and the raspberry pi will make it safer to use for our scenario. These are the only important components that need to be covered because they are the only ones which are in the open.

Additionally, we will be using a converter to change 120 V to 5V for some of our devices which would avoid any potential electrical hazards. Since we will be working with wall power, we have to be extra careful with high voltage outlets. We will first test our project in a safe lab environment where we have a guaranteed 5V source, then we will test our voltage converter to see if it does indeed provide a 5V source while using the one hand rule. This way, we will prevent potential damage to the sensors and microprocessors as well as ourselves. All these precautions comply with the lab safety guidelines.

References

[1] Iosrjournals.org, ‘Back Problems Due To Heavy Backpacks in School Children’, 2013. [Online]. Available: https://www.iosrjournals.org/iosr-jhss/papers/Vol10-issue6/D1062226.pdf?id=6253. [Accessed: 30 - Mar - 2020].

[2] Webmd.com, ‘Heavy Backpacks Can Hurt Students' Backs’, 2004. [Online]. Available: https://www.webmd.com/back-pain/news/20040813/heavy-backpacks-can-hurt-student-backs. [Accessed: 30 - Mar - 2020].

[3] 3m.com, ‘Smart Bookshelf System’, 2012. [Online]. Available: http://multimedia.3m.com/mws/media/685394O/smart-bookself.pdf. [Accessed: 1 - Apr - 2020].

[4] Manditremayne.blogspot.com, “Switching teams”, 2011 [Online]. Available: http://manditremayne.blogspot.com/2011/12/switching-teams.html

[5] GitHub, “arduino-latency-test”, 2019 [Online]. Available: https://github.com/sqwk/arduino-latency-test

[6] pyimagesearch.com, “How-To: Python Compare Two Images”, September 15, 2014 [Online]. Available: https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/ [Accessed: 31- Mar - 2020].

[7] resistorguide.com, “Photo-resistor”, [Online]. Available: http://www.resistorguide.com/photoresistor/ [Accessed: 31- Mar - 2020].

[8] tdn.com, “IEEE Code of Ethics”, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 30 - Mar - 2020].

[9] Ieee.org, “Study Haul — Student today carry dangerously heavy backpack and bags, doctors say”, 2005. [Online]. Available:

https://tdn.com/news/study-haul-student-today-carry-dangerously-heavy-backpack-and-bags/article_63922279-1f55-53f5-a54c-adfc8e0b806d.html [Accessed: 15 - Apr - 2020].

[10] World Health Organization, ‘Make Listening Safe’, 2020. [Online]. Available: ’https://www.who.int/pbd/deafness/activities/MLS_Brochure_English_lowres_for_web.pdf. [Accessed: 30 - Mar - 2020].

[11] Acm.org, “ACM Code of Ethics and Professional Conduct”, 2020. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 30 - Mar - 2020].