SELF CHECKOUT CART

Team 13 — Rohan Khanna, Pooja Kankani and Cherian K Cherian

ECE 445 Design Document —Spring 2020

Prof: Joohyung Kim

TA: Johan Mufuta

Table of Contents

1 Introduction

1.1 Objective

Self checkout terminals have become an essential component of any major supermarket chain. They cut labor costs and make the checkout process efficient. However, these terminals are very inefficient when it comes to scanning items that do not have barcodes such as fruits and vegetables. For such items either the customer or the supermarket staff have to manually enter the information of the item. There are many categories and subcategories of fruits and vegetables which make this process very inefficient and time consuming. If the staff member forgets the lookup code of a fruit or vegetable then it can cause a large amount of delays for customers. Gizmido reports that one of the main reasons people hated self checkout carts was that a cashier or supervisor had to be called over to enter some arcane code into the system for every other item. [1]

We propose to solve this problem by designing a display box that will be placed next to each shelf with grocery items likely to be missing a tag. This box will contain a programmable LCD screen that shows the information for the items placed on the shelf. For example, if the shelf stores apples, the LCD screen will show the price per pound for the apples, the type of apples, and other useful information like when the apples were stocked or their specific kind. This can be changed by the store managers if the product on the shelf changes. The box will also have a weighing platform that the user can place the items on, and the LCD screen will show a barcode that stores the corresponding price for that weight of that particular item (e.g., if the user places 1 pound apples on the platform, the barcode will store the price for 1 pound of the apples using the preprogrammed price for that shelf). There will be a detachable scanner attached to each cart at the entrance. The person shopping can use the scanner that they have attached to their carts to scan the barcode. It will keep adding the price for all the items they have shopped. This way the scanner will keep the total bill of the person ready and they can pay using an app, without having to wait in any check out counter lines. This makes it extremely easy to self checkout. The handheld scanner can also be used to scan the barcode of packaged products like bread and will add the cost of those items to the customer bill as well. The display box will also include an option to print out a barcode sticker, whereby the customer can just put the sticker on the plastic wrapper holding the groceries and then proceed to checkout if the customer is uncomfortable doing the labor of self checkout.

1.2 Background

Currently, self checkout scanners expect the customer to manually search for the item that they are buying or enter its reference code. Most customers struggle with finding the exact reference code and end up needing assistance from a store employee. Since there is usually only one employee looking over multiple self checkout stations, there is great delay in receiving assistance and the whole point of the self checkout process is eliminated. Vox reports that a lot of people end up not buying certain items or just outright steal items because they get angry that an item won't scan and figure it's not their job to try that hard. [2] This is a huge loss for the supermarket. Moreover, given that there are multiple sub categories of items such as fruits, customers can enter the wrong information at the

checkout screen and end up paying less or more for their product. The English newspaper Guardian reports that since people can search for items
without a barcode on the self checkout machine, they try to sneak in more expensive items by keying them in as some other cheaper items.

1.3 Visual Aid

This picture represents the two different devices that this project consists of - the handheld scanner and the programmable shelf box. On each shelf, there will be a box
pre-programmed with the prices and will have a screen displaying information specifically for that shelf item. Once the user places the desired quantity of that item on the weighing plate of that shelf, a barcode will be generated for that item and displayed on the screen. The customer can now scan this barcode using the handheld scanner, which in turn will communicate to the mobile application over WiFi. On the phone, the user will be able to see the cart of selected items and their prices. The user can remove items if he/she wants, using the delete button on the app. He/She can also make their final payment through this application securely.

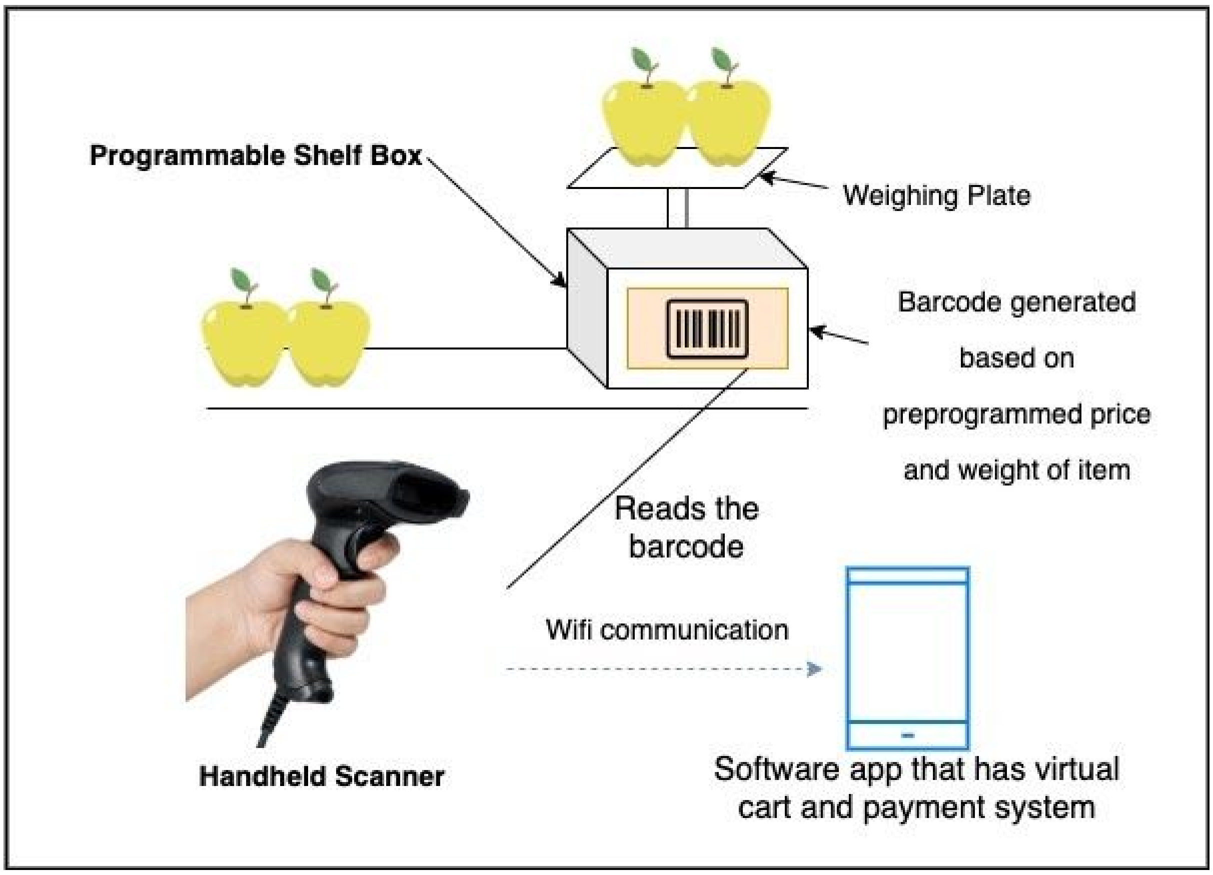

Programmable Shelf Box
Weighing Plate
Barcode generated based on preprogrammed price and weight of item
Reads the barcode
Wifi communication
Handheld Scanner
Software app that has virtual cart and payment system

Figure 1. Visual Aid of Project Design

1.4 High-Level Requirements

- The display box should automatically sync with an online server every 5 hours and update contents of its LCD display such as the time the product arrived, quantity left and other relevant information.

- When the user places an item on the scale, the LCD screen should show the price barcode for the selected weight for 15 seconds before going back to the price/pound screen. The weighing error should be within a tolerance of 10%.

- Once the item is scanned, there must be add/delete options for the item available on the software application cart within 1s and an option to make a payment through the app.

2. Design

2.1 Block Diagram

Our Block diagram is split up into 3 parts. The first is a programmable shelf box. This will be used for items such as fruits and vegetables for which it is difficult to weigh and scan items. Our solution involves using a load sensor to be able to weigh the item. Once we have weighed it, the Shelf box will calculate the cost and other information and display a barcode with the appropriate information. This can then be scanned by the handheld scanner to input the information for the quantity of the items you want to buy. The shelf is programmable so that we can change the item information depending on the item it is near.

The second part of our block diagram is the handheld scanner. This is a detachable scanner which can be used to scan the barcodes of any item. It contains a WiFi chip to connect to the server of the store to retrieve item information from it. Every time an object is scanned, it looks up the price, and updates the list of items and total cost stored in it. For items such as fruits whose price is dependent on weight, it scans the barcode from the programmable shelf box to receive information about the item. The handheld scanner also interacts with the Software app using the WiFi chip to send information about the items to the app.

The software app is used so that customers can view the items they have scanned and the total cost of all the items. It can also be used for payment so that checkout is done quicker.

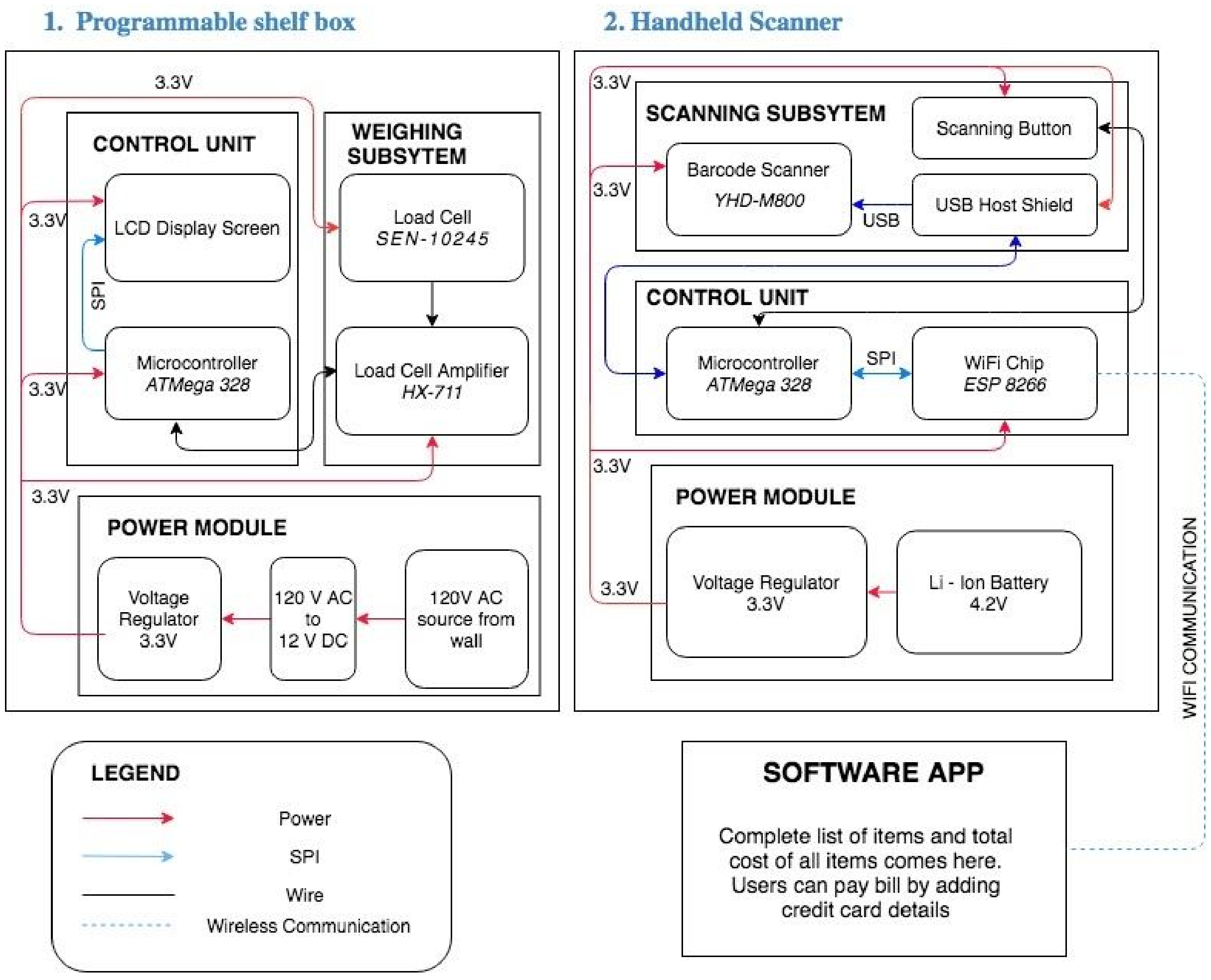


Figure 2. Block Diagram

2.2 Physical Diagram

We chose a scanner because it can easily be attached to the cart on a holder. Most people are familiar with how scanners work and hence the learning curve is minimal

The box is designed to be as small as possible for easy installment in the grocery area. The LED display gives much needed information about the product and has the potential to store a lot of information about the product.

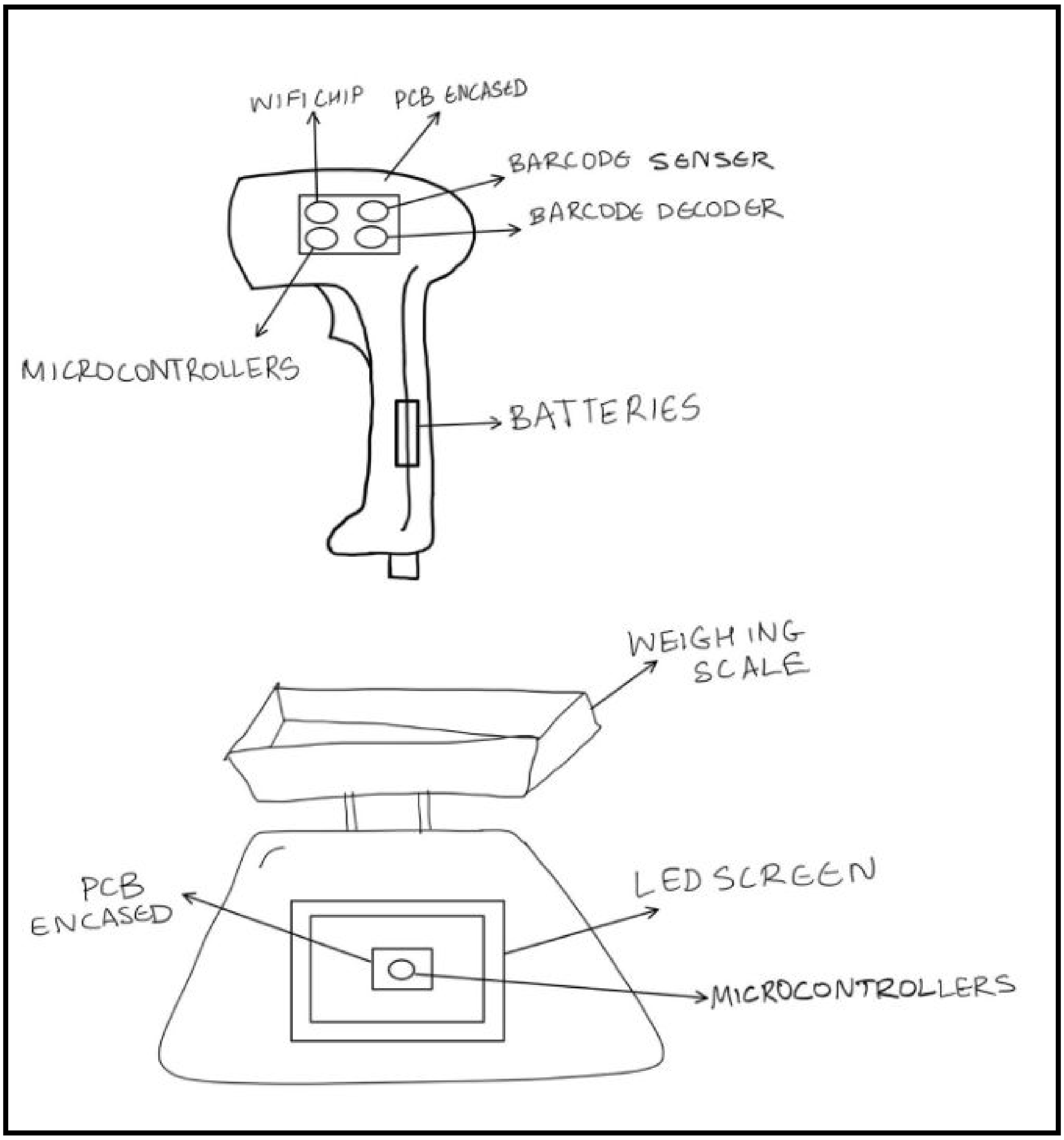


Figure 3. Physical Diagram of Wearable

2.3 Function Overview and Block Requirements

Programmable Shelf Box

This box will be placed on the shelf of grocery items and will consist of a weighing scale that the users can weigh grocery on. This weighing scale will communicate with the microcontroller to generate a barcode specific to that item and the weight the user selected. The barcode will show on the LCD screen and convey the total price for that item for that user.

2.3.1 Microcontroller

We are using an Atmega328P microcontroller for this because of its high performance and low power consumption. The microcontroller will be the main communication between the load sensor and the LCD display screen. Initially, it will be storing some information about the shelf product - like the price, best before date and details about the item itself. This will be sent to the LCD screen for display. Once the load sensor sends some information to the microcontroller about the weight of the item being purchased, the microcontroller will calculate the price of the item and send a generated barcode to the display screen. Each microcontroller can be re-programmed by the store representatives if they want to move it to a different shelf for another item.

Requirement	Verification
1. Should be able to receive load values from load cell amplifier	a) Connect the HX711 Load Cell Amplifier to the load cell using the color labeled pins b) Connect the load cell lifier to the microcontroller. The DAT pin is hooked up to an input pin in the microcontroller. c) Apply a small force on the load cell. Verify that the input pin on the microcontroller is receiving this value by printing to a terminal.
2. Should be able to send information to LED screen to be displayed over SPI protocol	a) Connect the microcontroller USB - SPI bridge. b) Use a terminal to send information from the microcontroller via SPI bridge. Verify that this is the same data getting echoed back.
3. Should be able to calculate price of item based on programmed information	a) Set preprogrammed price per kg value as 0.5$/kg. b) Send value of 10 to input pin on microcontroller. c) Calculation of price is Weight*price per kg. So, verify that the calculated price printed to the terminal is 5$. d) Repeat for different values of price per kg. The price per kg can be programmed depending on the item being measured.

2.3.2 LCD screen

The LCD display screen will act as a digital sticker on food items. We will be using an Adafruit 3627 display screen due to its low cost. It will initially display the shelf item information like the price and brand sent by the microcontroller. When the customer is trying to purchase a certain amount of an item, the microcontroller will generate a barcode representing the price of the purchase and send it to the screen for display.

Requirement	**Verification**
1. LCD screen should be able to display information sent to it from microcontroller	a) Connect LED display to microcontroller using an USB-SPI bridge. b) Send data to the LED display and verify that this shows up on the display. Eg. Send "Hello" and verify that this is being displayed.

2.3.3 Load Cell and Load Cell Amplifier

We are using a SEN 10245 load sensor and it can measure up to 110 pounds, which is enough for our use case. This sensor will be placed under a platform on which users can keep items they want to purchase, for example, 10 apples. The sensor will detect the weight of the products placed and send that information to the microcontroller for further calculation.

The load cell amplifier is used to amplify the signals received from the load cells. Since the voltage from the wheatstone bridge might be very small, we need an amplifier to be able to get measurable data out from the load cells. For our design, we are using the HX-711 from sparkfun.

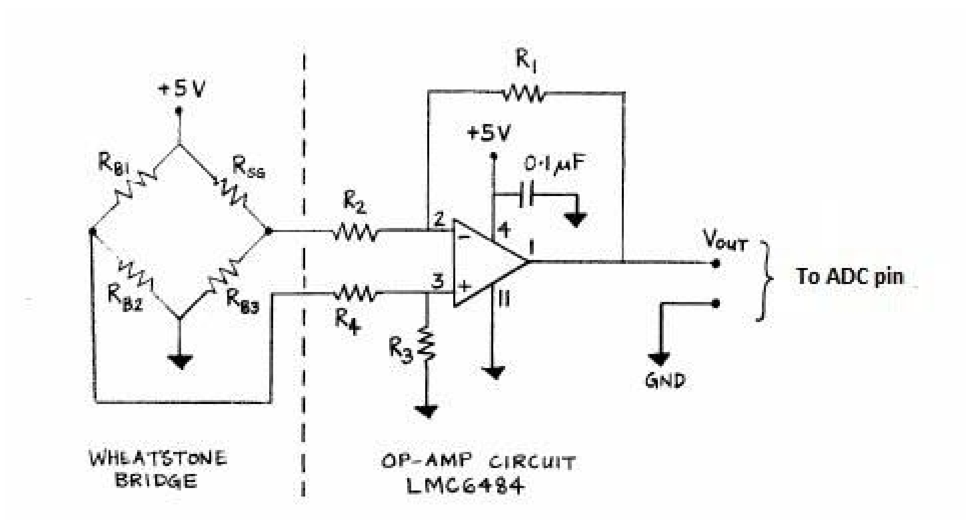


Figure 3. Testing amplifier circuit [3]

Requirement	Verification
1. Voltage change required for weight measurement should be detected in the wheatstone bridge network	a) Assemble the strain gauge sensors in the wheatstone bridge circuit shown in figure 3. b) Apply no weight. Measure the v output. It should be 0 c) Apply a 10kg weight on the strain gauge. d) There should be a voltage drop of a few mv across the wheatstone bridge network
2. The amplify the voltage difference generated in the wheatstone bridge.	a) Connect the wheatstone bridge from the first test to the op-amp circuit and set up the circuit as shown in figure 3 b) R1=R3 = 1Mohm R2=R4= 100kohm The gain should be Av = R1/R2 =10

Handheld Scanner

Each customer will have a handheld scanner that can be used to scan barcodes for items they wish to purchase. By placing the desired item on the programmable shelf box, they can scan the generated barcode on the LCD screen using this handheld scanner, which will send the price and information of the purchase to the user's mobile app through WiFi.

2.3.4 Microcontroller

We will be using the ATMega328P for the handheld scanner as well due to its low power consumption that will allow the scanner to last for the entire day on the batteries. Each microcontroller will be storing a unique user app id that can be changed per customer and this id links the hand scanner to the user's mobile app using WiFi. The microcontroller will receive a signal when the scan button is pressed by the user. It will then start the barcode scanner. Hence, the microcontroller is the main link between the button and the scanner. When the barcode scanner reads the total price from the LCD screen, it will send it to the microcontroller. The microcontroller will then signal to the WiFi chip to send that information to the software app for that given user.

Requirement	**Verification**
1. Should be able to send information regarding price to the WiFi chip.	a) Connect the microcontroller to the WiFi chip using the SPI protocol. b) Send dummy data to the WiFi chip through the MOSI port. Eg. "Apple 5$" c) Print input from MOSI port in WiFi chip to terminal and verify that it matches the data sent.
2. Should be able to receive data from the USB host shield which is connected to the barcode scanner module.	a) Connect the microcontroller to the USB host shield using the SPI protocol. b) Connect USB host shield to an input source, Eg. Computer. c) Send dummy data value from USB host shield to microcontroller through input source Eg. "12345". Print data received on microcontroller input to the terminal and verify that the printed value matches data sent.

3. Should be able to indicate to the barcode scanner to start scanning while the scanning button is pressed.	a) Connect button to microcontroller which is connected to the USB host shield. The USB host is connected to an output source such as a computer. b) Press the scanning button and send the signal received by the microcontroller(high signal) to the USB host shield. c) Print value received by host shield and verify that it is a high signal, indicating the scanner to start scanning.

2.3.5 Li-Ion Battery

We are using 4.2V Li-ion rechargeable batteries to power the handheld scanner. The reason we chose these rechargeable batteries is so that the store can recharge these scanners once the store closes and keep it ready for the next day. It will also last long enough to power the components throughout the day so that customers can use it without facing any issues.

Requirement	Verification
1. Should output at least 3.7V	a) Connect battery terminals to positive and negative terminals of multimeter b) Ensure multimeter output is at least 3.7 volts.
2. Should store 500mAh amount of charge	a) Connect positive terminal to a voltage source of 3.7V and negative terminal to ground b) Discharge battery at a rate of 100mAh for 4.5hours c) Use a multimeter to ensure battery voltage is still 3.7V. This means that it has at least 50mAh amount of charge since the battery still shows the correct amount of voltage.

2.3.6 Barcode Scanner and USB Host Shield

We will be using the YHD-M800 scanning module for scanning the barcode from the LCD display. We chose this module because this scanner can read both 2D and 1D barcodes printed on labels and displayed on device screens, which is ideal for our case. It has powerful decoding capacity and is capable of reading even fuzzy or incomplete barcodes, so it will work even when customers are in a hurry and not scanning the barcode properly.

The barcode scanner uses a USB interface, so we use a USB host shield to be able to send information to the microcontroller. The USB shield uses the SPI protocol to connect to the microcontroller.

Requirement	Verification
1. Should be able to scan a barcode and decipher the price of the item with no more than 4% error.	a) Generate a barcode for a given price (decimal number). b) Use the scanning module to scan the barcode. c) Connect the YHD-M800 to a computer using the USB port. d) Display the decoded price on the computer screen and calculate: Error% = (Original Price - Decoded price)/100 Ensure that the error percentage is less than 4%.
2. Should be able to communicate with the USB host shield and start scanning when input is sent.	a) Connect the barcode scanner to a computer using USB. b) Send a high signal to the barcode scanner and verify that the barcode scanner attempts to scan.
3. Should be able to communicate with the microcontroller about scanned information.	a) Connect the barcode scanner to a computer using USB. b) Scan a barcode for items whose data we know beforehand, Eg. “Apples” and send to the computer through USB. c) Print input onto the terminal and verify that it matches what we scanned.

2.3.7 Button

A push button will be used to allow the customer to start scanning when they see a barcode for an item. On press, the button will signal to the microcontroller that the scanning system should be started as the user wants to buy an item. The microcontroller will transfer control over to the scanning subsystem which will turn on the red laser beam for scanning. Once a button is pushed, and the scanning starts, further button pushes will be ignored until the entire scanning process for that barcode is done.

We are using the Momentary Pushbutton Switch - 12mm Square from sparkfun which is a cheap and reliable button and is pushdown meaning that it will send a signal only while pressed down which is ideal for our use case.

Requirement	Verification
1. Should be able to send signals to the microcontroller when pressed.	a) Connect button to the microcontroller. b) Press button, which sends a high signal(ie, 1) to the microcontroller. c) Print input received by button from microcontroller onto the terminal and verify that it is a high signal.

2.3.8 WiFi chip

We will be using the ESP8266EX WiFi chip and this will be used in communicating from the handheld scanner device to the customer's software application. Information such as each item purchased by the customer and its price will be sent to the software application for a virtual cart system and total bill display on the app. This WiFi chip has high range and accuracy and will work well even indoors, such as in a grocery store. It will establish connection with the mobile app only after the microcontroller sends it a signal to start along with all the information that needs to be delivered.

Requirement	Verification
1. Should be able to communicate with the microcontroller and receive price information of the scanned product from it.	a) Connect microcontroller and WiFi chip via the SPI protocol. b) Send dummy data from microcontroller to WiFi chip, Eg. ”Apples 5$” c) Print input received by WiFi chip onto terminal. Verify that it matches data sent.
2. Should be able to communicate with the software app and send item information to it.	a) Connect WiFi chip to dummy phone app. b) Send dummy data from WiFi chip to phone. c) Verify that the data received on the app is what is sent from the WiFi chip

Common Functionality

2.3.9 Voltage Regulator

We are using a voltage regulator to ensure that the entire circuit has access to a well regulated 3.3V power supply for both the handheld scanner and the programmable shelf box. This will be used to supply the microcontroller, WiFi chip and display screen, load sensor and the barcode system with power.

We will be using L4931 from ST for our purposes because it is a low dropout linear voltage regulator. The battery supplies 4.2-3.7V of power which needs to be regulated to the 3.3V needed for the chips. Since we can only allow for a 0.4V dropoff, we cannot use a normal linear regulator since the dropoff is at least 1V, which would make the output less than 3.3V. The low dropoff voltage regulator has a dropoff of 0.4V at 250mAh which is perfect for us since we can still get the clean 3.3V output needed for each part using the 3.7V power supply from the battery. Also, the peak current supply of 250 mA is more than enough for the use of our circuit.

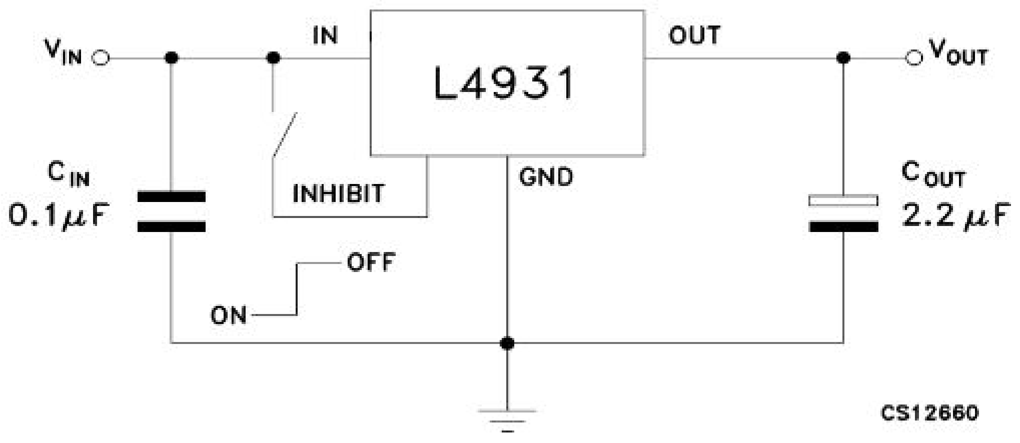


Figure 4. Voltage Regulator Application Circuit

Requirement	Verification
1. The 3.3V regulator should be able to output 3.3V +/- 2% when connected to the 3.7V battery.	a) Connect voltage regulator to a power supply of 3.7V. b) Connect output of voltage regulator to an oscilloscope to verify 3.3V output.
2. Maintains thermal stability below 60°C.	a) Connect battery to resistor. Voltage regulator is connected to GND and VDD across the battery. b) The IR Thermometer is used to

	ensure temperature is below 60°C.

Software Application

The software application will receive the information of all the grocery purchases made by the user from the handheld scanner through WiFi. This application can then be used as a personal shopping cart to keep track of items and quantities the user has picked up. The user can remove items and add items again as he/she pleases through the application and the scanner.
Once the user is done shopping, the application will allow the user to make payments for the items directly to the store through a secure online payment method. This way the user doesn't have to wait in line for making payments and can do so conveniently through their phone. It can also be used by the user to keep track of a buying list, to make the buying process easier for the customer.

Requirement	Verification
1. Allow the user to delete an item from the cart.	a) Make a cart on the app and add 3 hard coded items in it. b) Press the delete option for a particular item on the cart. c) Ensure that the deleted item does not show up in the user's cart anymore and the remaining items stay intact.
2. It should display the correct bill for the user based on all the items in the cart.	a) Add 10 items to the cart with an associated price for each. b) Press "Checkout" to see the total bill for the user. c) Check if the bill calculated for the accounted items is correct and only uses the items presently in the cart.
3. Should let the user make a payment using their credit card for the correct amount securely.	a) Create a hard coded cart for the user. b) Go to "Checkout" and then "Make a Payment". c) Make a payment using a credit card to a known bank account. d) Check in the bank account to see if the money has been deposited.

2.4 Flowchart of project workflow:

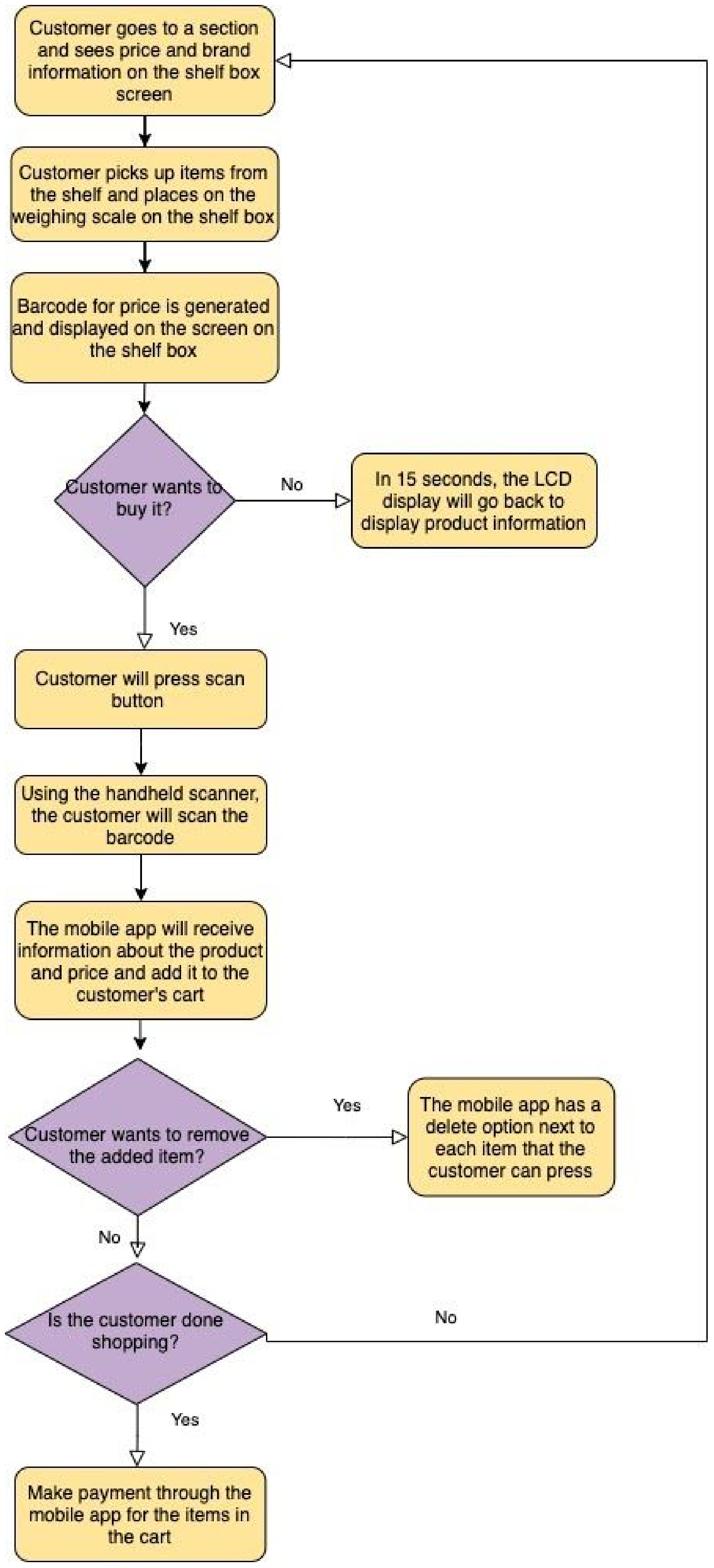

Customer goes to a section and sees price and brand information on the shelf box screen
Customer picks up items from the shelf and places on the weighing scale on the shelf box
Barcode for price is generated and displayed on the screen on the shelf box
Customer wants to buy it?
No
In 15 seconds, the LCD display will go back to display product information
Yes
Customer will press scan button
Using the handheld scanner, the customer will scan the barcode
The mobile app will receive information about the product and price and add it to the customer's cart
Customer wants to remove the added item?
Yes
The mobile app has a delete option next to each item that the customer can press
No
Is the customer done shopping?
No
Yes
Make payment through the mobile app for the items in the cart

Figure 5. Flowchart of project workflow

2.5 Risk Analysis

Firstly, sensitivity errors in the weighing component of the display box or errors in the communication channel could cause great issues in cost computation and lead to inaccurate costs being determined. Since this component needs to be smaller than a traditional weighing machine or one placed near self checkout carts, the size could affect accuracy and hence display incorrect costs for the user. To prevent this we could have a weight average mechanism that quickly computes three readings of the weight of the item and takes average to determine cost. In addition, we could try to use past data of fruits and vegetable weights to determine whether a measured weight is significantly inaccurate from the actual weight.

Finally, a critical component of our project is that the barcode generation, the scanning and the transfer of data to the mobile application must all happen very quickly within a max timeout of 5 seconds. All of these heavily rely on the wifi quality at the supermarket. If the Wifi signal is not strong there could be major lag in this process and hence cause inconvenience to the customer rather than improve the customer experience.

3 Tolerance analysis

According to our high level requirements when the user places an item on the scale, the LCD screen should show the price barcode for the selected weight for 15 seconds before going back to the price/pound screen. This is a critical component of our project as this is where the barcode generation occurs which is highly dependent on the weight of the items. In addition, the barcode generated should be on the screen for a suitable amount of time for the customer to scan the barcode using their handheld scanner.

The weighing module of our project is extremely important because even minor errors in weight can cause annoyance to the customer and delays in weight generation can lead to long wait times. Our weighing system should be tolerant to tolerate weights upto 10kg as bags of oranges or certain big fruits can weigh that much. In addition, weighing should be uniform across the scale (values between one end of the scale and the other should differ by atmost 1%).

Finally the error in cost should be less than 0.1% of the cost of the item.
Hence we are performing a tolerance analysis on the weighing subsystem of our project.

Current load cells have 5 major types of errors :

1. Non Linearity :   the maximum difference between load cell output readings for repeated loadings under identical loading conditions (that is, either increasing the load from zero or decreasing the load from the load cell's maximum rated capacity) and environmental conditions[5]
2. Hysteresis:The difference between load cell output readings for the same applied load, one reading obtained by increasing the load from minimum load and the other by decreasing the load from maximum load.[5]
3. Non-Repeatability:non-repeatability refers to the ability of a load cell to maintain a consistent output when an identical load is repeatedly applied.[6]
4. Creep: is the change in load cell output over time when a load remains on the cell for a long time[6]
5. Error due to fluctuating temperature.

These can be modelled by the diagram below:

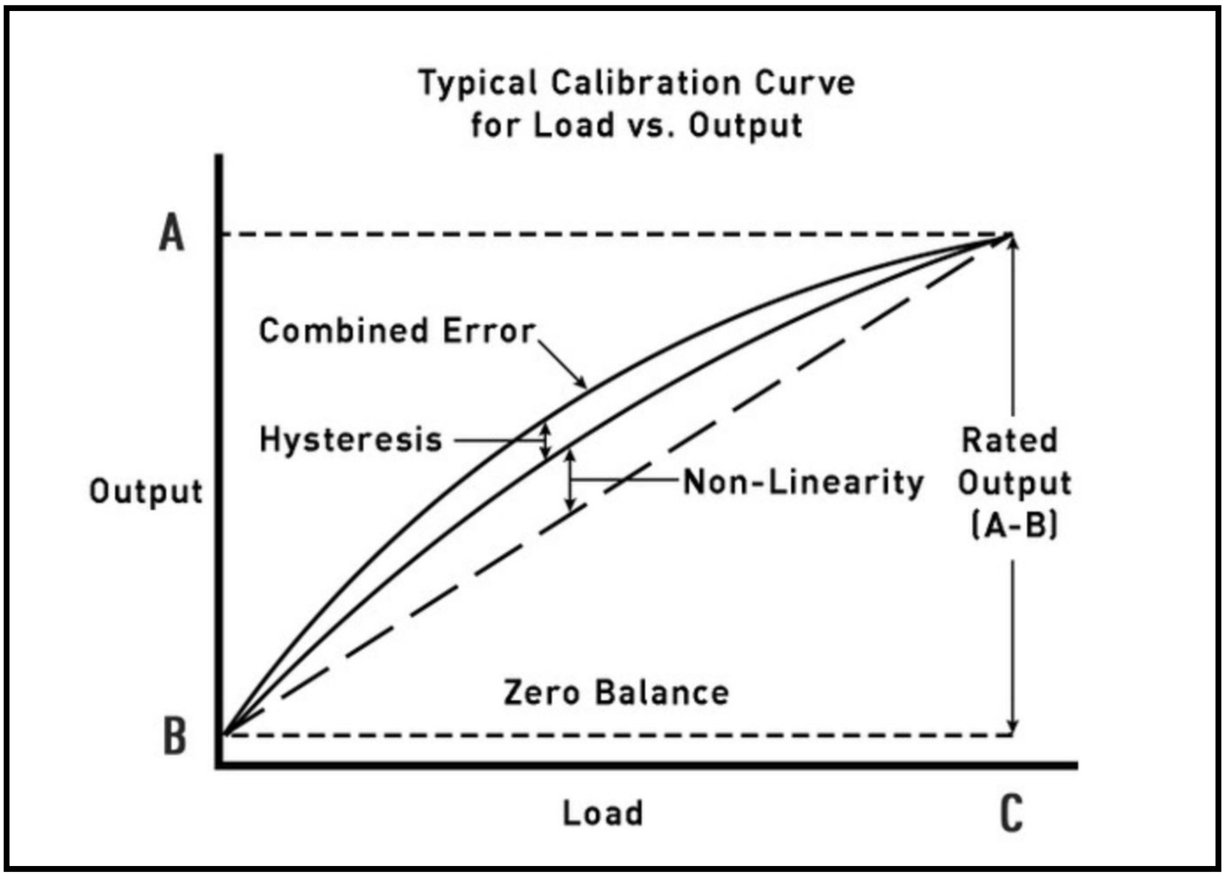


Figure 6 : Typical Calibration Curve for Load vs Output [6]

For our scenario of a grocery store Non-linearity, Hysteresis and Non-Repeatability and Creep are key parameters that our design choices need to account for.
In our project we are using 4 SEN-10245 load cells. According to the datasheet, the SEN-10245 has the following values for these three errors:

1. Non-linearity: 0.03%
2. Hysteresis: 0.03%
3. Non-Repeatability: 0.03%
4. Creep: 0.03%

The combined error in the load cell can be determined by the following formula .

$$\varepsilon > \sqrt{\varepsilon_L^2 + \varepsilon_H^2 + \varepsilon_R^2 + \left(\frac{\varepsilon_Z \times L \times N}{W_1} \times t\right)^2 + (\varepsilon_S \times t)^2}$$

ε	:	Measurement accuracy of the load cell (%)
ε_L	:	Nonlinearity (%)
ε_H	:	Hysteresis error (%)
ε_R	:	Repeatability (%)
ε_Z	:	Temperature effect on zero balance (%/°C)
ε_S	:	Temperature effect on span (%/°C)
L	:	Rated capacity of the load cell
N	:	Number of load cells to be used
W_1	:	Maximum load to be measured
t	:	Temperature variation range of the load cell (°C)

Figure 7 : Error formula for combined error of load cell [7]

We are using 4 load cells hence N =4. The rated capacity of the load cells is 50kg and the max weight we will measure is 10kg. Since the scales are in a controlled environment of a supermarket the temperature variation is not applicable.

Hence the total combined error we will face is : 0.05196%.
Our required error tolerance is 0.1% within the cost of the item which is greater than the 0.05196%. Hence our design of 4 load cells does not exceed our required accuracy level.

We also need to ensure that we measure a weight of 10kg. We use the SEN-10245 load cell that has a much greater rated capacity of 50kgs. The load cell uses resistance wires whose value goes up when a weight is applied as the wire is stretched. A strain gauge cannot measure extremely small changes in resistance. Thus we have performed an analysis on how our design choice ensures that these load cells fulfill the requirement of the project.

Now we need to measure weights upto 10kgs.

A 10kg weight on a strain gauge of 1.7 m by 2mm by 2mm causes the length to change by 0.04m
Strain = ($\Delta L/L$) = 0.04/1.7 = 0.0235 ε

The strain gauge has a gauge factor of 2 meaning the change in electrical resistance will only be 0.047 ohm.

This is a very small value and extremely hard to measure accurately. Hence we decided to assemble the strain gauges in a wheatstone bridge network and use an amplifier for the vout so that the output voltages can be used to determine the weight of the object being placed.

Our 4 strain gauge load cells are arranged in a wheatstone bridge network and are placed in the 4 corners of our scale. The wheatstone bridge network enables us to measure extremely small changes in resistance and converts them to appropriate voltage differences. The arrangement is as follows:

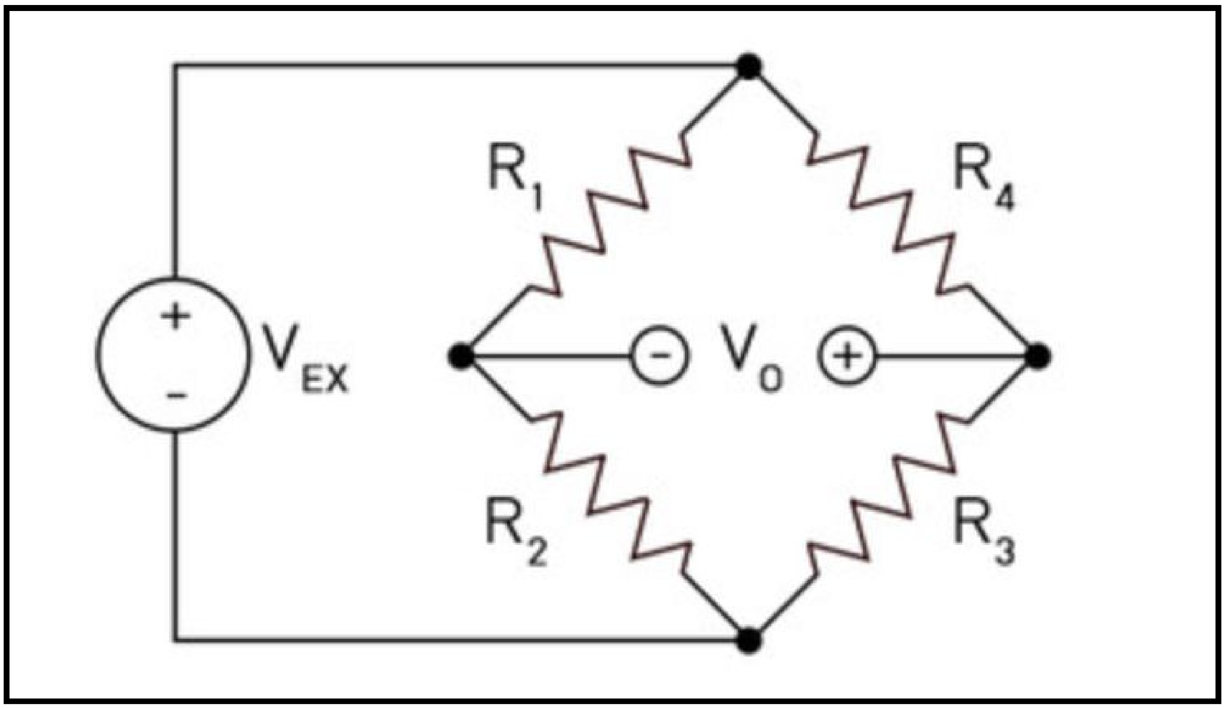


Figure 8: Wheatstone arrangement of the strain gauge resistors [6]

Voutput = $(R3 \div (R3 + R4) \ - \ (R2 \div R1 + R2) * V\,ex$

The wheatstone bridge will only produce a Vo >0 when there is even the smallest resistance change.

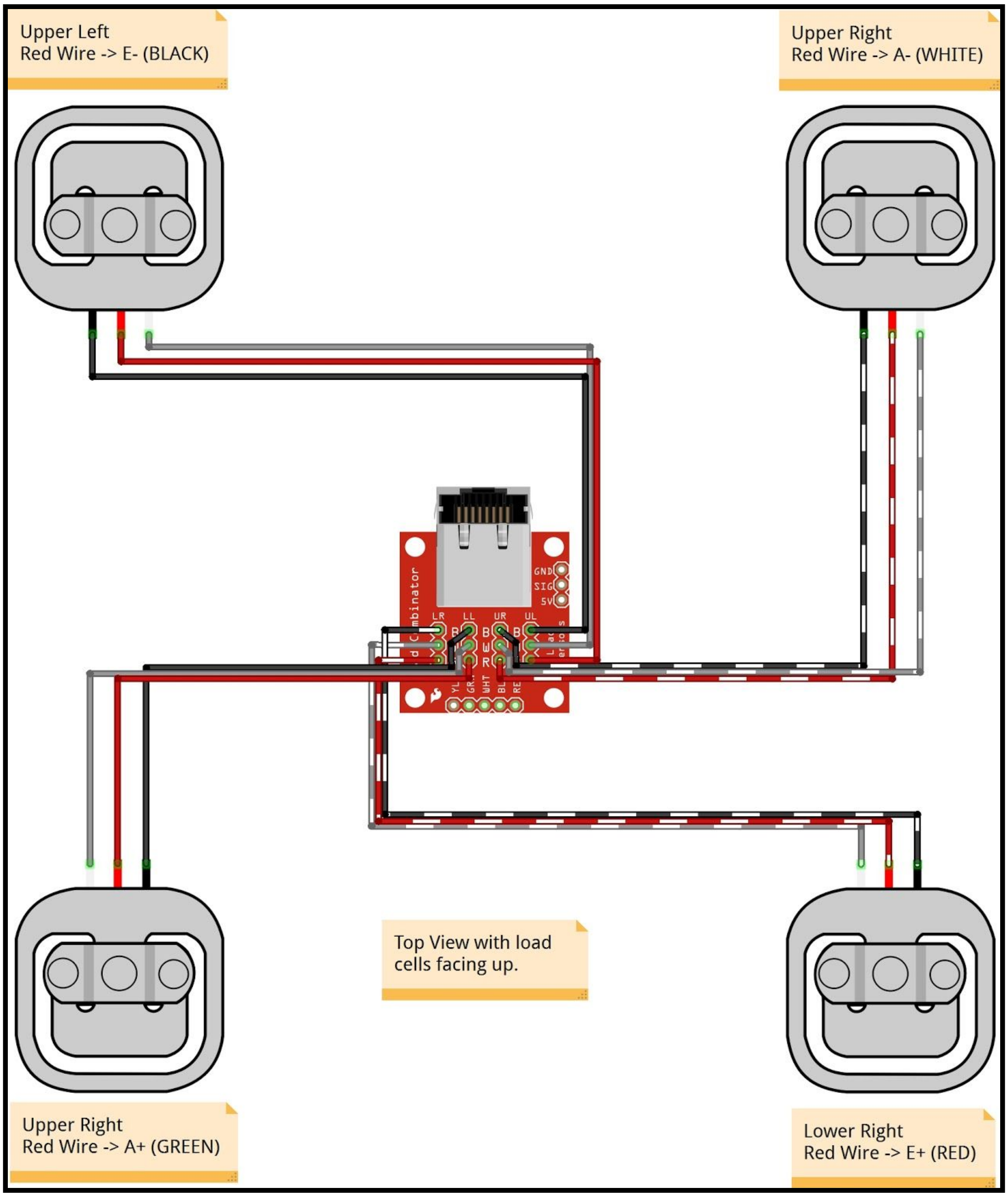


Figure 9: Amplifier network for the weighing scale. [8]

Thus the above circuit will ensure that we can measure weights of 10kg.

4 Project Differences

4.1 Overview

This project tries to solve the same problem that the FALL 2014 self checkout cart team was trying to solve. The solution proposed and developed by the FALL 2014 team involved the use of computer vision to automatically recognize fruits and vegetables. It uses a camera to take images of an item which will rotate on the scale and then tries to determine what the item is. The solution consists of a Panel block, Imaging block, User Interface block and a central processor block. The Panel block consists of the weight scale, the stepping motor and the panel to keep the item on. The camera block consists of the camera and an LED component which provides the necessary lighting for the fruit to take images. The components interface with each other through the processor block which conveys important information to the user interface block.
The solution has the following workflow:

1. The customer places the item on the panel. The Led then applies the appropriate lighting based on the surroundings and then the camera takes a picture of the item.
2. Once a picture is taken the processor instructs the motor to turn the item.
3. Four such pictures from different sides of the fruit are taken and sent to the processor.
4. The processor processes each image as it comes. It compares the image data received to that stored in its SD card and identifies the item

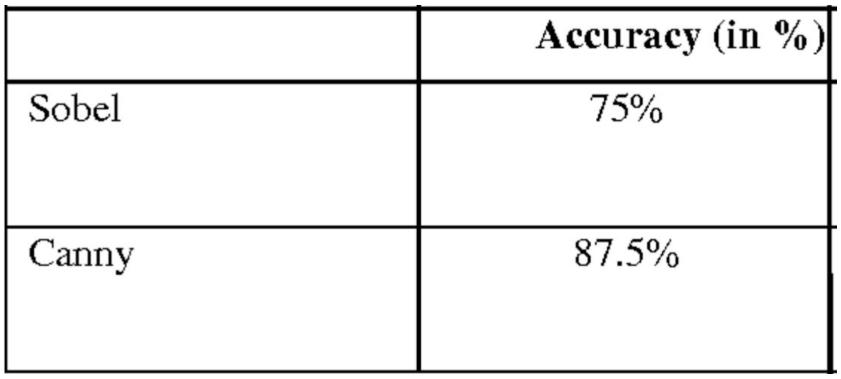
There are many issues with this approach. To start with about 40 percent of respondents of self checkout terminals cite technical glitches as the most annoying aspect of self service checkout. [10] Since computer vision has not reached 100% accuracy this solution is bound to raise technical glitches. These technical glitches in detecting items will definitely cause discomfort to customers. For example   It becomes extremely hard to apply image recognition to objects that are extremely similar like alphonso mangoes and Atualfo mangoes. Alphonso mangoes are way more expensive than Atualfo mangoes and hence this error can be extremely costly to the customer or the store owner. Next the solution required the customer to place each fruit one at a time so it can take 4 images. Customers tend to buy fruits in bulk and expecting them to place these fruits one at a time on the scale is unrealistic. Processing of each image takes 4 seconds. Thus for a bag of 10 apples the customer has to wait 40 seconds for the bag to self scan. In addition, this solution requires a lot of computing power and a significant amount of memory to operate correctly. Our solution eliminates the need for computer vision and hence does away with the inaccuracies and high cost associated with implementing a computer vision algorithm. Barcode scanners are reliable tools and have been used in grocery stores for decades. They ensure greater reliability and accuracy than the computer vision approach. In addition, our solution allows customers who are not fond of self checkout to just take their barcode scanner to the nearest cashier who can then read the id of the barcode scanner and bag all the items for the customer. Our solution hence has no computer vision component, no camera component taking images and is not located at the checkout counter. Instead, it is a handheld scanner that interacts with static lcd screens at different locations in the store which is a radically different approach from the first project proposal. In the next section, we will elaborate on the points mentioned in this overview and provide an in depth analysis on how our project improves upon the original design.

4.2 Analysis

The purpose of this section is to prove and provide further analysis to the points made in the overview section. Listed below are the issues with the solution proposed in the original project and how our solution is better.

1. Accuracy of computer vision

   ISSUE: The original solution aims to reduce time spent at checkout counters in entering barcodes for items such as fruits and vegetables. This is a valid problem as 40% of customers complain about having to call staff when using the self checkout for entering data for food[9]. The solution of the fall 2014 team uses the canny edge detection algorithm to analyze the edges of the pictures taken of the fruit. Another algorithm that the team proposes is the sobel edge detection algorithm. Dr S Vijayarani and and Mrs. M. Vinupriya Published a paper to semantic scholars where they show an in depth analysis on the performance of the two algorithms on detecting facial features of humans.   As shown below the sobel edge detection model only has a 75 % accuracy whereas a Canny detection algorithm has an 87.5% accuracy. These accuracy measures are for human faces which have very distinct features. The amazing variety of human faces – far greater than that of most other animals – is the result of evolutionary pressure to make each of us unique and easily recognizable, according to a new study by University of California, Berkeley, scientists. [10]. Fruits do not have such distinct features. Apples can easily be confused with oranges or pears. Moreover, There are items that have certain subtle differences that cannot be visually seen from the outside. For example, alphonso mangoes look exactly like Atualfo mangoes but alphonso tend to cost more. Having many false positives can hinder customer experience and cause long delays in checkout lines. Hence these algorithms will perform much worse on fruits and vegetables

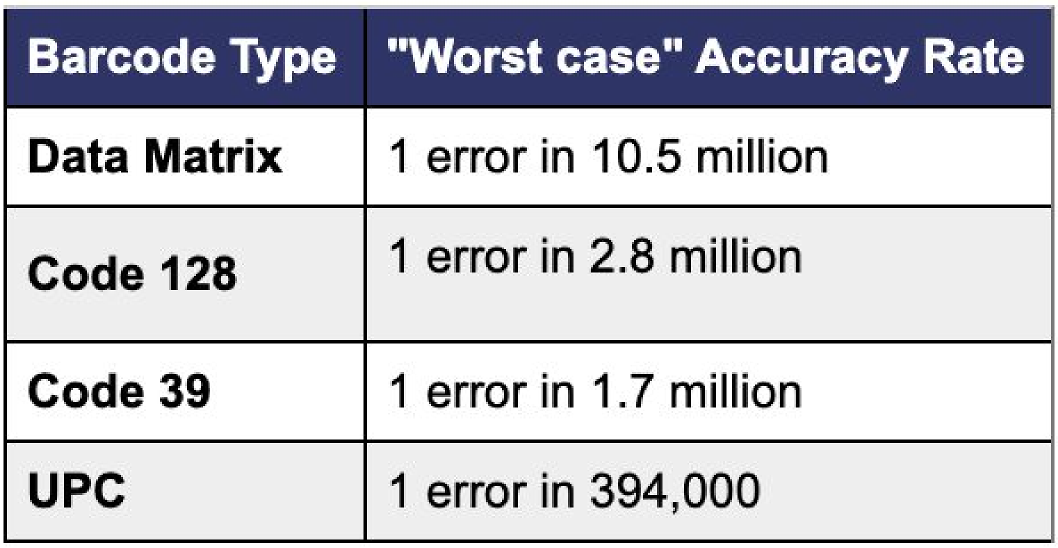


	Accuracy (in %)
Sobel	75%
Canny	87.5%

Table 1: Accuracy of edge detection models.

Assuming that an average customer tries to scan about 20 different fruits and vegetables. Based on the 87.5% accuracy the detection algorithm will fail for 2 out of 20 fruits.
Each time the fruit is detected incorrectly the customer has to wait for the cashier, effectively eliminating the whole point of creating this solution.

OUR APPROACH : Our solution involves a handheld scanner and an LCD screen with a tag at each fruit and vegetable station. When the scanner scans the tag it will know which item the customer is dealing with as the entire setup is with the information of the fruit. In addition, we have the microcontroller pre-programmed to detect if the average weight measured of the fruit is outside the range that this unit should be measuring. For example if a customer puts a watermelon on the weighing scale of an apple the LED box will alert the customer that they are at the wrong box. Finally we use barcode scanning technology which has unprecedented accuracy as compared to the edge detection models ensuring that a customer never has to face hassles or call the cashier while scanning their goods. In the table below is the error rate for various barcode scanning techniques.



Barcode Type	"Worst case" Accuracy Rate
Data Matrix	1 error in 10.5 million
Code 128	1 error in 2.8 million
Code 39	1 error in 1.7 million
UPC	1 error in 394,000

Table 2: Barcode Worst case accuracy rate. [11]

2. Processing power

ISSUE: Even if the computer vision part of the project can be done with a reasonable accuracy, it requires extreme processing power. The current approach of the group uses expensive raspberry pi's which alone account for 70 dollars in the total cost of the project. These raspberry pi's have a speed of 1.4 GHZ but our microcontroller only has a max speed of 28MHz. The difference between the two is about 1.37 GHZ This high processing power is needed for the image processing component of the original project. Since our project has relatively simple processes we only need the 28MHz microprocessor.

OUR APPROACH: Our solution has very simple components with no major processing power required. The microcontroller in the hand held scanner just has to be responsible for the scanning. The LED screen will consume minimal power as it just has to display basic information about the section.

3. Time sink

ISSUE: Rotating the fruit and continuously taking images will waste customer time and lead to long lines. According to the original project the aim was to make the entire computer vision project take about 4 seconds for each fruit. Since each fruit item has to be weighed separately to generate the cost, and every fruit has to be individually put on the scale and scanned, the time taken to checkout increases significantly. This solution thus trades off having no tags on fruits with a longer wait time. Considering long wait times is something that turns customers away from stores, this implementation would probably lose revenue for the store since customers are less likely to enter or return due to the longer checkout lines.

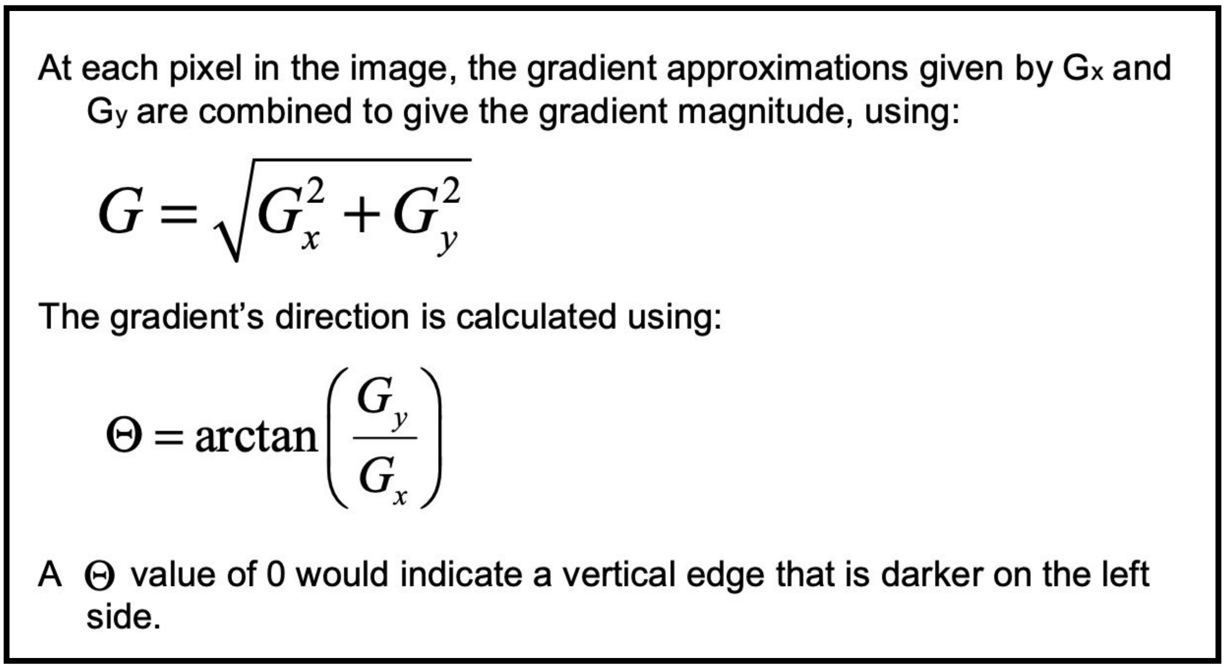
<u>OUR APPROACH</u>: Our approach does not have any component requiring the customer to wait and hence there is no question of a line. Secondly since the cost of the items is calculated on the handheld scanner, and the scanner is synced to an app on the phone. The customer can easily just pay on the phone and leave without having to stand in lines. Further, a lot of time spent walking around browsing for items can be spent scanning items. This significantly increases the speed of checkout since we can use the time walking around to scan items. Since there is a seperate weighing section for each fruit, customers are not made to wait a long time to checkout any item. Long checkout lines is one of the biggest factors that customers look for when avoiding a grocery store. By cutting these down significantly, we also add the possibility of more customers and thus more revenue for the store.

The edge detection algorithm used by the previous project is Sobel Edge detection algorithm. According to the analysis on accuracy above, the accuracy for Sobel algorithm is 75% based on a research conducted. We can assume that the accuracy for fruits is even lower since they are much harder to distinguish. Assuming an accuracy of 75% thus, the time taken on average to scan a fruit is 4*1.25seconds = 5 seconds.

On the other hand, barcode scanning takes around 2seconds per item, taking into account time taken to locate the bar code.[] Thus, the time taken to scan each item is 2.5 times faster with a scanner than with the computer vis

4. Data intensive

<u>ISSUE</u>: The original project uses the Sobel filter for edge detection, along with other algorithms for color and shape detection . At each pixel in the image, this algorithm calculates the gradient of the image intensity to identify edge orientation for both X and Y axes separately.
“It finds the direction of the largest increase from light to dark and the rate of change in that direction. The sobel filter uses two **3x3 kernels**. One for changes in the horizontal direction, and one for changes in the vertical direction.” [12]



At each pixel in the image, the gradient approximations given by G_x and G_y are combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2}$$

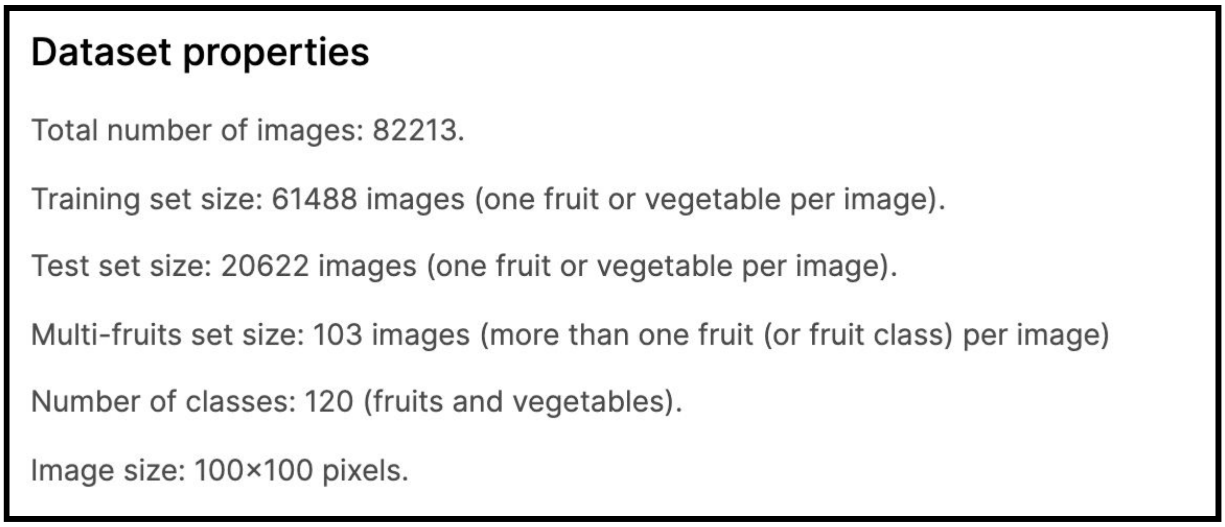
The gradient's direction is calculated using:

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

A Θ value of 0 would indicate a vertical edge that is darker on the left side.

Figure 10: Sobel filter calculation for edge detection. [12]

This entire operation for each item from four different angles to detect edges and color, and determine the final object is very data intensive and calculation heavy.



Dataset properties

Total number of images: 82213.

Training set size: 61488 images (one fruit or vegetable per image).

Test set size: 20622 images (one fruit or vegetable per image).

Multi-fruits set size: 103 images (more than one fruit (or fruit class) per image)

Number of classes: 120 (fruits and vegetables).

Image size: 100×100 pixels.

Figure 11: Example of large datasets required for detection [12]

It also requires multiple large datasets for training and testing purposes that should ideally contain examples for all varieties of different fruits and vegetables such as the one shown above. Once the Sobel filter returns an image with the edges highlighted, the project would have to match that to different items in the datasets to find the correct item and then predict a price.

OUR APPROACH: Our implementation does not involve any data intensive calculations. We are not using any machine learning algorithms or datasets for image detection. Our project relies on preprogrammed information on the shelf box by the employees who are knowledgeable about the different varieties and prices of the items. Once these prices are loaded, our shelf box uses the weighing scale to detect the weight of the item placed and multiplies the price and weight to display the accurate price for the selected item in the form of a barcode. As our project has no camera and image detection, the huge component of the datasets, the issues arising with their accuracy, any the computation of matrix multiplication in the multiple dimensions does not apply to us.

In conclusion, our solution has no computer vision component, no camera taking images and is not located at the checkout counter. Instead, it is a handheld scanner that interacts with static lcd screens at different locations in the store which is a radically different approach from the first project proposal.

5 Cost analysis

4.1 Labor

According to the official website of the Grainger college of engineering the average starting salary for Engineering graduates is $78,159. This equates to an hourly wage of 37.58 USD.

Hence the labor cost is given by:
37.58 USD x 2.5 x 70 = 6576.5 USD per person
Hence the total cost for 3 people = 19729.5 USD

4.2. Part analysis

4.2.1 Handheld Scanner

Component	Manufacturer	Product ID	Quantity	Cost
Barcode Scanner	YHDAA	YHD-M800	1	23.00 USD
USB Host Shield	SparkFun	MAX3421E	1	26.95 USD
Microcontroller	Microchip technology	ATMEGA-328P	1	2.08 USD

Wifi chip	Espressif Systems	ESP8266EX	1	6.95 USD
Lithium ion rechargeable battery	Adafruit Industries	ADAFRUIT 2750	1	6.95 USD
Voltage Regulator	Adafruit	L4931	1	0.95 USD
Button	SparkFun	Momentary Pushbutton Switch - 12mm Square	1	0.50 USD
TOTAL PRICE				67.38 USD

Table 3: Handheld Scanner cost

4.2.2 Programmable Shelf Box

Component	Manufacturer	Product ID	Quantity	Cost
Load Cell	SparkFun	Load Sensor - 50kg	4	4*10.95 = 43.8 USD
Load Cell Amplifier	SparkFun	HX711	1	9.95 USD
Microcontroller	Microchip technology	ATMEGA-328P	1	2.08 USD
LCD Display Screen	Nordic semiconductor	NRF51822	1	2.48 USD
Voltage Regulator	Adafruit	L4931	1	0.95 USD
TOTAL COST				59.26 USD

Table 4: Programmable Shelf cost

4.2.4 Grand Total

Labor : 19729.5 USD
Handheld Scanner cost : 67.38 USD
Programmable Shelf cost : 59.26 USD

Grand Total : 19,856.14 USD

6 Schedule

Week no. (days)	Pooja	Rohan	Cherian
1	Design Doc - requirements, visual aid, flowchart	Design Doc - Verifications, Tolerance analysis	Design Doc - subsystem description, block diagram
2	Work on making a presentation for design review	Work on making a presentation for design review	Start ordering the required parts for the project
3	Research more about load sensor an amplifier limitations and usage	Start making a PCB design	Start making a PCB design
4	Start setting up android application	Continue PCB design if required. Test connection between WiFi chip and mobile app	Start setting up android application
5	Test the barcode scanning module with different barcodes and its compatibility with the mobile app	Verify barcode generation shows accurate price for items selected by testing with different weights and items	Test The programmability of the shelf box and see that changes are being reflected on the LCD display screen
6	Integration, testing and future work if time permits.	Integration, testing and future work if time permits.	Integration, testing and future work if time permits.
7	Mock Demo/ Final Report/ Final Demo/ Final Presentation Preparation. Tie up any loose ends and test thoroughly	Mock Demo/ Final Report/ Final Demo/ Final Presentation Preparation. Tie up any loose ends and test thoroughly	Mock Demo/ Final Report/ Final Demo/ Final Presentation Preparation. Tie up any loose ends and test thoroughly

7 Safety and Ethics

Our project has quite a few safety or ethical issues based on our current project design and vision. We will completely abide by the IEEE Code of Ethics [13], throughout the development of our project and its usage. We will follow guideline #1 in the IEEE Code of Ethics, and our main goal is the wellbeing of our users and of the environment, while making the shopping experience easier and more efficient for all. We will be transparent about our design and limitations, so that the project cannot harm people. We will make ethical design choices and hold the user’s safety and well-being as the main priority while designing and implementing the project.

Through our scanner system, the grocery stores would get a lot of data about customer buying behavior that could be used by the store for different purposes. We will ensure that we do not store or allow misuse of any personal data of the users. We will make our system reliable to prevent data manipulation and information stealing through strong internal security measures and robust data transfer mechanisms. On the software application, we will allow users to pay online and we have to make sure that the payment data is securely stored and properly encrypted to prevent stealing. The user will also be given preference on whether they want to store that information or enter it everytime. The financial details of customers will not be shared or used in any way.

According to point #3 in the IEEE Code of Ethics, we will be honest and transparent with the user about any data we collect and how it will be used. Our project has a hand held device that allows users to scan barcodes. For user safety, we will make sure that the device is safe and does not, in any situation, cause any harm to the user. The current within the device will be cut off through a fuse if any circuit errors are detected. This will ensure that the user cannot get an electric shock due to exposed parts or other issues. Following guideline #7, we will always be willing to improve and correct any errors and criticism we receive, to improve the product and make it more useful.

The Li-ion batteries in the handheld scanner can pose risks that can be very harmful for human life, due to overheating and overcharging. This can cause the scanner to catch fire, while in the hand of customers. To ensure that this does not happen, the employees will be trained on how to charge the scanners and will be given instructions for proper storage. While working with the batteries, we will be careful and closely follow the instruction manual to ensure our safety and those of the people around us.

8 References

[1]https://gizmodo.com/why-self-checkout-is-and-has-always-been-the-worst-1833106695 [Accessed: 04/01/2020].

[2]https://www.vox.com/the-goods/2018/10/2/17923050/self-checkout-amazon-walmart-automation-jobs-surveillance [Accessed: 04/01/2020].

[3] Strogonovs, R. (n.d.). Strain Gauge based weight sensor (load cell) - MORF - Coding And Engineering. Retrieved April 17, 2020, from https://morf.lv/strain-gauge-based-weight-sensor-load-cell

[4] Voltage Regulator Datasheet: https://www.st.com/resource/en/datasheet/l4931.pdf [Accessed: 04/01/2020]

[5] Weighing System Accuracy, https://www.hardysolutions.com/tenants/hardy/documents/5FactorsA.pdf [Accessed: 04/01/2020]

[6] Strain Gauge Load Cell Basics, https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html [Accessed: 04/01/2020]

[7] How to use Load Cells, https://www.aandd.jp/products/weighing/loadcell/introduction/pdf/6-1.pdf [Accessed: 04/01/2020]

[8] Strogonovs, R. (n.d.). Strain Gauge based weight sensor (load cell) - MORF - Coding And Engineering. Retrieved April 17, 2020, from https://morf.lv/strain-gauge-based-weight-sensor-load-cell

[9] Study on Self Checkout Carts,
https://www.retailcustomerexperience.com/news/study-self-checkouts-causing-shopper-frustration-abandoned-carts/ [Accessed: 04/01/2020]

[10] Variances of human faces,
https://news.berkeley.edu/2014/09/16/human-faces-are-so-variable-because-we-evolved-to-look-unique/ [Accessed: 04/01/2020]

[11] LabCE MediaLab,
https://www.labce.com/spg650115_barcode_reading_and_accuracy.aspx [Accessed: 04/01/2020]

[12] Sobel Edge Detection Paper,
https://www.cs.auckland.ac.nz/compsci373s1c/PatricesLectures/Edge%20detection-Sobel_2up.pdf [Accessed: 04/01/2020]

[13] Institute of Electrical and Electronics Engineers, 'IEEE Code of Ethics', 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 04/01/2020].