



# Initial Project Document Divide and Conquer **Solar Picnic Table**

## **Group D:**

*Amos Luo - Computer Engineering*

*Azzan Al Busaidi - Electrical Engineering*

*Faisal Al Ghafri - Electrical Engineering*

*Thach Vo - Computer Engineering*

## **Reviewers:**

*Qifeng Li*

*Saleem Sahawneh*

*Aleksander Dimitrovski*

# Table of Contents

List of Tables	ix
List of Figures	x
1. Executive Summary	1
2. Project Narrative	3
2.1 Motivation	3
2.2 Objectives and Goals	3
2.3 Functionality	4
2.4 Project Constraints	4
2.4.1 Financial	4
2.4.2 Power Generation and Distribution	4
2.4.3 Power storage capacity	5
2.4.4 Weather	5
2.5 Specifications and Requirements	5
2.5.1 Power Distribution Table	6
2.5.2 House of Quality Diagram	7
2.6 Microcontroller/software implementations	8
2.6.1 Microcontroller	9
2.6.2 Software	9
2.7 Block Diagrams	9
2.7.1 Power Distribution	10
	10
<i>Figure 2 - Power Distribution Block Diagram</i>	10
2.7.2 Microcontroller Block Diagram	11
<i>Figure 3 - Microcontroller Block Diagram</i>	11
2.7.3 App Block Diagram	12
	12
2.8 Budget Estimation	13

3. Research and Part Selection	14
3.1 Solar panel	14
3.1.1 Rated Power	14
3.1.2 Efficiency	14
3.1.3 Material	14
3.1.4 Sun Hour	15
3.1.5 Climate	15
3.1.6 Solar Tracking	15
3.2 Types of Solar Panels	16
3.2.1 Monocrystalline Solar Panels	16
3.2.2 Polycrystalline Solar Panels	16
3.2.3 Passivated Emitter and Rear Cell Solar Panels	17
3.2.4 Thin-Film Solar Panels	17
3.2.5 Decision	18
3.3 Battery Storage System	18
3.3.1 DOD	19
3.3.2 Cycle life	19
3.3.3 Efficiency	19
3.3.4 Power density	19
3.3.5 Energy density	19
3.3.6 Charge time	19
3.3.7 Self discharge	19
3.4 Battery Comparison	20
3.4.1 DOD Comparison	20
3.4.2 Energy & Power Density	20
3.4.3 Self Discharge	20
3.4.4 Charge Time	20
3.4.5 Efficiency	21
3.4.6 Maintenance Requirements	21
3.4.7 Cycle Life & Lifetime	21
3.4.8 Weight	21
3.4.9 Cost	22
3.4.10 Decision (Lead Acid Battery)	22

3.5 Lead Acid Battery Types	23
3.6 Battery Parameters	23
3.7 Charge Controller	24
3.7.1 Functions of a charge controller	25
3.7.1.1 Overcharge protection	25
3.7.1.2 Over Discharge protection	25
3.7.1.3 Reverse current flow prevention	25
3.7.1.4 Overload protection	26
3.7.1.5 Charge Controller Operation	26
3.7.2 Charger Controller Types	27
3.7.2.1 Pulse Width Modulation (PWM) charge controller	27
3.7.2.2 Maximum Power Point Tracking (MPPT) Charge Controller	29
3.7.3 Decision (DEPVKO PWM)	30
3.8 Inverter	30
3.8.1 Different Inverter Technologies	30
3.8.1.1 Pure Sine Wave Inverter	30
3.8.1.2 Microinverter	31
3.8.1.3 Grid-tied Inverter	31
3.8.2 Inverter Comparison	31
3.8.2.1 Output waveform	31
3.8.2.2 Performance	31
3.8.2.3 Installation	31
3.8.2.4 Cost	32
3.8.2.5 Technology Decision	32
3.8.3 Inverter Parameters	32
3.9 Power outlet	34
3.9.1 Types of Power Outlets	34
3.9.1.1 AC Power Outlet	34
3.9.1.2 DC Power Outlet	34
3.9.1.3 USB Power Outlet	34
3.9.2 Decision (Type B)	35
3.10 LED	35
3.10.1 Efficiency	36

3.10.2 Flexibility	36
3.10.3 Programmability	36
3.10.4 Keeping it green	36
3.11 Programmable LED types	36
3.11.1 SMD5050	37
3.11.2 WS2811/12/12B	37
3.11.3 Decision (WS2812B)	37
3.12 Mobile App Connection	38
3.12.1 Mobile App Connection Comparison	38
3.12.1.1 Wi-Fi	38
3.12.1.2 Bluetooth	39
3.12.1.3 Decision (Bluetooth)	40
3.13 Developing the Mobile App	40
3.13.1 Platforms available	40
3.13.1.1 React Native/Flutter	40
3.13.1.2 Decision (React Native)	41
3.14 Interacting with the Microcontroller	42
3.15 Microcontroller options	42
3.15.1 ATmega2560	42
3.15.2 Atmega4809	43
3.15.3 ATSAMD21G18A-AU	43
3.15.4 MSP430FR6989	43
3.15.5 ESP32-WROOM-32E-N16	44
3.16 Microcontroller Comparisons	44
3.16.1 Pins	44
3.16.2 Memory	45
3.16.3 Power Consumption	46
3.16.4 Clock Speed	47
3.16.5 Price	47
3.16.6 Decision (ESP32-WROOM-32E-N16E-N16)	48
3.17 Motion Sensor Technology	49
3.17.1 Infrared	49
3.17.2 Passive Infrared	50

3.17.3 Microwave	51
3.17.4 Ultrasonic	51
3.17.5 Decision (PIR)	52
3.18 Temperature Monitoring	53
3.18.1 DHT22	53
3.18.2 TMP36	54
3.18.3 LM35	54
3.18.4 Decision (DHT22)	54
3.19 Battery Monitoring	55
3.19.1 Voltage sensor module	55
3.19.2 Manual Voltage Divider Circuit	55
3.19.3 Decision (Manual Voltage Divider Circuit)	55
3.20 PCB	56
3.21 Equipment Parameter Calculations	57
3.21.1 Total energy consumption	57
3.21.1.1 LED	57
3.21.1.2 Power Outlets	58
3.21.1.3 Temperature Sensor	58
3.21.1.4 Motion Sensor	58
3.21.1.5 Microcontroller	58
3.21.1.6 Inverter Rating	58
3.21.1.7 Battery Rating	59
3.21.1.8 Charger controller Rating	59
3.22 System Performance Simulation	59
3.22.1 Input	59
3.22.2 Output	60
3.23 Market product comparisons	61
3.23.1 Eshine Solar Coffee Table	62
3.23.2 Kay Park Solar Charge Station with Fiberglass Umbrella	63
3.23.3 Enerfusion Aurora Solar Picnic	64
3.23.4 Market product comparisons summary	65
4. Project Design	67
4.1 Hardware Design	67

4.1.1 Electrical System Wiring Design	67
4.1.2 PCB	70
4.1.2.1 Schematic Design	71
4.1.2.2 PCB Design	74
4.1.3 PIR Motion Sensor	77
4.1.4 Temperature and Humidity Sensor	78
4.1.5 Voltage Divider	78
4.2 Software Design	79
4.2.1 Software Development Tools	79
4.2.1.1 Arduino IDE 2.0	80
4.2.1.2 Mobile app	80
4.2.2 Motion Sensing	82
4.2.3 Temperature and Humidity Sensing	83
4.2.4 Battery Monitoring	83
4.2.5 Bluetooth Connectivity	84
4.3 Test Plan	86
4.3.1 PCB testing	86
4.3.2 Solar panel testing	87
4.3.2 Charge controller testing	87
4.3.3 Inverter testing	88
4.3.4 Battery storage system testing	89
4.3.5 Power Outlet testing	89
4.3.6 LED testing	90
4.3.7 Motion Sensor Testing	90
4.3.8 Temperature and Humidity Sensor Testing	91
4.3.9 Battery Monitor Testing	92
4.3.10 Bluetooth Connectivity Testing	92
4.3.11 Mobile App Testing	93
4.3.12 WaterProof Electronics Box	95
4.4 Prototype	96
5. Constraints and Standards	99
5.1 Constraints	99
5.1.1 Economic	99

5.1.2 Environmental	100
5.1.3 Social	101
5.1.4 Political	101
5.1.5 Ethical	101
5.1.6 Health and Safety	102
5.1.7 Manufacturability	103
5.1.8 Sustainability	104
5.2 Standards	104
5.2.1 UL 1703 Standard for Flat-Plate Photovoltaic Modules and Panels	104
5.2.2 IEC 60038 Standard Voltages	105
5.2.3 ISO/IEC 9899 Programming Languages C	105
5.2.4 Impact of ISO/IEC 9899 Programming Languages C	106
5.2.5 IEEE 1562-2017 Recommended Practice for installation of Lead Acid batteries in PV systems	107
5.2.6 National Electrical Code (NEC):	107
6. Administration	108
6.1 Milestone Discussion	109
6.1.1 Senior Design	109
6.2 Product Milestones	110
6.1.1 Table Decision Milestone	110
6.1.2 Milestone for Electrical Components	110
6.1.3 Milestone for Microcontroller and Sensors	111
6.1.4 Milestone for Mobile Application	112
6.1.5 End Product Milestone	112
6.3 Budget and Finance Discussion	112
6.4 Role Assignment	116
6.4.1 Project Manager	116
6.4.2 Electrical Designer	116
6.4.3 Software Developer	116
6.4.5 Hardware Designer (Microcontroller and Sensors)	117
6.4.6 Documentation	117
6.5 Task Distribution	117
7. Conclusion	120

8. Appendix	121
A.1 References	121

## **List of Tables**

Table 1 Measurements of Picnic Table.....	5
Table 2 Power Distribution.....	6
Table 3 Budget Estimation.....	13
Table 4 Comparison of PV Panel Technologies.....	18
Table 5 Comparison of Lithium-ion and Lead Acid Battery.....	22
Table 6 Comparison of Lead Acid Battery Types .....	23
Table 7 Battery Specification Parameters .....	24
Table 8 Comparison Between PWM and MPPT Controllers .....	30
Table 9 Specifications of MRI3011BU Inverter .....	33
Table 10 Specifications of B0774SCCND Inverter .....	33
Table 11 Comparison of LEDs .....	37
Table 12 Pros of Wifi and Bluetooth.....	39
Table 13 Cons of Wifi and Bluetooth .....	40
Table 14 Differences of Flutter and React Native.....	41
Table 15 Comparison of GPIO pins/current per pin on MCU .....	45
Table 16 Comparison of Flash Memory/RAM of MCU .....	46
Table 17 Comparison of Power Consumption between MCU .....	46
Table 18 Comparison of Clock Speed of MCU .....	47
Table 19 Price Comparison of MCU .....	48
Table 20 Motion Sensing Technology .....	52
Table 21 Temperature Sensor Comparison.....	54
Table 22 Comparison of Operating Voltage and Current.....	56
Table 23 Comparison of Different Solar Tables.....	66
Table 24 BOM of components for PCB.....	74
Table 25 Senior Design 1 Milestones .....	109
Table 26 Senior Design 2 Milestones .....	110
Table 27 BOM of all components .....	115
Table 28 Task Distribution .....	119

## **List of Figures**

Figure 1 <i>House of Quality Diagram</i> .....	8
Figure 2 <i>Power Distribution Block Diagram</i> .....	10
Figure 3 <i>Microcontroller Block Diagram</i> .....	11
Figure 4 <i>App Block Diagram</i> .....	12
Figure 5 <i>Demonstrates Solar Tracing comparison</i> .....	16
Figure 6 <i>The EXP1280 Model</i> .....	24
Figure 7 <i>Battery charging Stages</i> .....	26
Figure 8 <i>PWM pulses Demonstration</i> .....	28
Figure 9 <i>The Baomain Power Outlet</i> .....	35
Figure 10 <i>Depicts the Mechanism Behind IR Sensing</i> .....	49
Figure 11 <i>Depicts the Mechanisms Behind PIR Sensing</i> .....	50
Figure 12 <i>Depicts the mechanisms behind microwave sensing</i> .....	51
Figure 13 <i>Depicts the mechanics behind ultrasonic sensing</i> .....	52
Figure 14 <i>PV Energy Output</i> .....	60
Figure 15 <i>Battery performance</i> .....	60
Figure 16 <i>Charging State Frequency</i> .....	61
Figure 17 <i>Eshine Solar Coffee Table</i> .....	62
Figure 18 <i>Kay Park Solar Charge Station</i> .....	63
Figure 19 <i>Enerfusion Aurora Solar Picnic Table</i> .....	65
Figure 20 <i>Charge controller to Barrel jack port connector</i> .....	68
Figure 21 <i>Power Distribution</i> .....	69
Figure 22 <i>Voltage regulator design</i> .....	70
Figure 23 <i>12V to 5V regulator schematic</i> .....	71
Figure 24 <i>12V to 3V regulator schematic</i> .....	71
Figure 25 <i>Schematic Drawing</i> .....	73
Figure 26 <i>PCB Design Window</i> .....	75
Figure 27 <i>PCB final design</i> .....	76
Figure 28 <i>Schematic for PIR motion sensor</i> .....	77
Figure 29 <i>Schematic for the DHT22</i> .....	78
Figure 30 <i>Schematic of Microcontroller Connected to Voltage Divider</i> .....	79
Figure 31 <i>Flowchart of how the Emulators work</i> .....	81
Figure 32 <i>Flowchart of Motion Sensing and LED Interaction</i> .....	82
Figure 33 <i>Flowchart for BLE connection</i> .....	85
Figure 34 <i>BLE architecture</i> .....	85
Figure 35 <i>MCU connected to AM312</i> .....	90
Figure 36 <i>MCU connected to DHT22</i> .....	91
Figure 37 <i>Example of a Mobile App UI in Figma</i> .....	93
Figure 38 <i>How the application interacts with ESP32</i> .....	94

Figure 39 <i>App Development Process</i> .....	95
Figure 40 <i>Example of Electronics Box</i> .....	96
Figure 41 <i>ISO View of the Table</i> .....	97
Figure 42 <i>Top View of the Table</i> .....	97
Figure 43 <i>Bottom View of the Table</i> .....	97
Figure 44 <i>Outline and Labeled view of the Table</i> .....	98

# **1. Executive Summary**

For many people, going outside is an important aspect of living a healthy lifestyle. Another key aspect in almost everyone's life is technology. Everyday, we interact with all sorts of technology whether it be our phones, the computer, or our cars. However, with the advancements and increased usage of technology, one factor that we must keep in mind is how it will affect the environment. With this in mind, as a senior design group, we wanted to create a project that would incorporate technology and enhance people's outdoors experience, but at the same time, one that wouldn't pose a threat to the natural environment. This is when we came up with the idea of a solar powered picnic table.

For this picnic table, there are three major parts: the power management, the table functionalities, and the mobile app. As for power management, all power used in this project will be generated by a 100W monocrystalline solar panel. This panel will be linked to a PWM solar charge controller that will be responsible for distributing power to the battery and the loads, while providing protection against overcharging, over discharging, and overload. The excess power generated by the panel will be stored in our 8 Ah lead-acid battery, which will provide it back as needed at night time when the solar panel is not generating any. The table will come with many different functionalities. First, it will be equipped with LED lights, this way the table is usable during the day and night. Next, it will have various charging ports so that people are able to charge their electronic devices. It will also have the ability of motion sensing. This will be implemented using four PIR motion sensors. This will be used to automatically turn the LEDs on or off if someone is using the table at night. Finally, it will have temperature and battery monitoring. People will be able to gather information about the environment and how much power the table has available. All of the information gathered by the sensors will be sent to the ESP32 microcontroller and be outputted for users to see on the mobile app. In this mobile app, our goal is to provide the people sitting at the table an application to check everything they need while sitting at the table. The app will be available to all mobile phone platforms through the use of React Native which allows us to program a mobile app that is available on all platforms and connects to the ESP32 microcontroller through bluetooth LE. Once the user connects to the ESP32 microcontroller on the app, they will be able to see things such as battery life, temperature around them, and control LED lights, all in a clean and simple UI for anyone to pick up and understand.

In this report, we will be explaining the details of the research and design process of our solar picnic table. The paper will first discuss the motivation of the project and its goals. Next, it will go into depth about the research we conducted for different components such as microcontrollers and batteries. Furthermore, this section delves into the different part choices for each element of the picnic table and determines which part would best suit our needs and goals. From there the report discusses the various constraints that we must consider for our project and the standards that we must follow for a coherent design. Once this is done, the paper will go onto the design and testing portion of the project. In this section, it will explain the designs of both the

hardware and software elements of the solar picnic table. Along with this, there will be a detailed explanation of the various testing plans that will be used to ensure that the hardware and software are working as intended. Next, the prototype of the solar picnic table will be shown in figures and explained. Finally, the document will go over the budget of the project and milestones to achieve the finished product.

## **2. Project Narrative**

In this chapter, we will go over the initial design of our project. This includes the motivation, objective and goals, functionalities, constraints, and general requirements.

### **2.1 Motivation**

Life is advancing at an extraordinary pace, this evolution in life is accompanied by stressful jobs and lifestyles, and what way is better to relieve that stress than going out for a picnic in nature. However, because of the lack that we still have in our modern technologies like batteries running out and the absence of portability in other technologies, a lot of people bypass the idea of taking a picnic because they will be far from some essential amenities they enjoy at home. In this project, we are aiming to solve this problem, and since preserving earth is our duty, we are powering our project using renewable energy. Looking at the geographical aspects of our area, the optimum form of renewable energy to use will be solar energy. We will utilize the power generated by the solar panel to power our picnic table. This table will contain the most crucial amenities people are looking for when heading out for a picnic, like a power outlet paired with USB outlets, LED lights, a battery to store excess energy, and an app that will ease accessing these facilities through the user's smartphone.

### **2.2 Objectives and Goals**

For this project, we have two primary goals: creating a smart picnic table and making it environment friendly. To accomplish these two goals, we have set up different objectives. First, we will add different features to the table so that it can offer our customers a unique experience. Some of these features are bluetooth, lighting, and charging. On top of this, we plan to implement an app that allows users to control these functionalities. To make it environment friendly, all power consumed by the picnic table will be generated by solar panels.

In the beginning of the project, we have decided on a list of obtainable goals and then stretch goals. Obtainable goals are goals that the table must include and are goals that fit within the project time and budget restraints. Some of the obtainable goals we have set are: LED lights with motion sensors, temperature monitoring, mobile app to provide users with information about the table, solar panels gathering energy for powering all components of the table, and a battery to store energy. These goals are the main ones we are focusing on, however if we had more time and had an increase in our budget, we would have implemented our stretch goals to improve upon our project.

While planning out the project, we wanted to create the best experience for the user when it comes to using our solar picnic table. With the many ideas we had, we had to separate some of them to stretch goals, goals that we will not afford to achieve in our

initial model, but would like to add to our future models. Some of these stretch goals would be things like solar panel tracking to allow for the solar panel to tilt and rotate based on the sun to maximize energy production, bluetooth speakers integrated to the table and app, wireless charging, and custom made tables. These goals are separated due to either budget or time restrictions and may return to obtainable goals as the project continues.

## **2.3 Functionality**

Our goal is for the table to function during all day times, and to be safe to use under all weather conditions, whether rainy, windy, or cloudy, if the battery is charged with sufficient energy that is. Battery energy level can be checked through an app, in which it will be paired to. This app will allow the user to check for energy in the battery, manually turn LED lights on/off and check the temperature, all using bluetooth to connect through the cell phone. Another requirement of making this happen is protecting the battery and power distributing wires from water, dust, and insects, this will be done through a waterproof box that will keep the battery and wires protected at all times. The table will also contain an AC power outlet, and two USB ports, for charging and powering other devices.

## **2.4 Project Constraints**

For a project of this scale, there are many possible constraints and limitations that might have an effect on the final outcome of this project. It is very important that we thoroughly investigate, research, and plan ahead. These constraints could be either from the financial aspect, power generation and distribution, power storage capacity, or even software implementations. Below we have listed these possible issues and provided a detailed explanation on how they might affect the final product.

### **2.4.1 Financial**

The biggest constraint financially would be the table itself. If our group isn't able to use one of the tables provided by UCF, then we would have to build our own table. This could prove to be an obstacle financially as it can get expensive acquiring the materials and equipment for this task.

### **2.4.2 Power Generation and Distribution**

Since we are planning to power the whole table using solar energy, the main concern is having a solar panel that is able to generate enough energy to power all the electrical components in our system. Some of these components in our system would require AC current while others will require DC. Having one solar panel with a limited budget may provide challenges in providing the various types of currents at different voltages each without any power shortage.

## 2.4.3 Power storage capacity

The main constraint regarding the storage capacity is ensuring that the battery storage is able to store enough power that would last the whole night when sunlight is no longer available. Under ideal circumstances a fully charged battery storage should be able to last at least 6 hours (6 hours is the average “peak sun”) while powering all components at once.

## 2.4.4 Weather

While the weather in Orlando is a great place to utilize solar energy, there are some days of the week where it is going to be hard to generate electrical energy from solar. The table’s design will be able to withstand any rains without any damage, but won't be able to generate sufficient energy to last through the night.

## 2.5 Specifications and Requirements

Length	3ft - 5ft
Width	3ft - 5ft
Height	2ft - 4ft
Pole Height	4ft - 5ft
Material	Wood/Plastic/Steel
Shade	Enough to cover table area

*Table 1 - Approximate Measurements for the Picnic Table*

- Waterproof box
  - Waterproof box to enclose arduino and modules
  - Waterproof outlet box and cover to protect electronics
- Solar panels
- Solar charger controller
- DC to AC inverter 150W-250W capacity
- Solar Racking (optimum angle of solar panel 25-29 degrees)
- Power Storage
  - Lithium-ion battery, enough storage to hold excess power from solar panel
  - lead acid batteries
- Microcontroller

- microcontroller, main controller where a lot of the functionality/monitoring is centered around  
Bluetooth/Wifi module, allows for users to wirelessly connect, monitor, and control functions on the table
- Motion Sensor, turns on LED Lights when user is close to the table (feature only works at night using internal timer)
  - Some possible motion sensors would be PIR, IR, microwave, and ultrasonic
- Voltage & Current sensors, used to monitor output generated from solar panel and to monitor how much power is left in the battery
- Temperature monitor, monitors the temperature around the table to give the users accurate temperature readings
- Power outlets and USB ports
  - Standard type B power plug
  - Standard USB plug
- LED strips
- Temperature Sensor

## 2.5.1 Power Distribution Table

Item	Voltage (V)	Current (A)	Power (W)
<b>Solar Panel</b>	12	Up to 8.33	100
<b>Battery</b>	12	8Ah	100
<b>MCU</b>	3.3	0.130	0.429
<b>LED</b>	5	2.4	12
<b>Motion sensor</b>	3.3	0.00006	0.000198
<b>Temperature sensor</b>	3.3	0.0015	0.00495
<b>Power Outlet</b>	110 - 120 AC & 5 DC	Depends on device being plugged	Max 300

*Table 2 - Shows the Power Distribution of the Electrical Components*

## 2.5.2 House of Quality Diagram

The house of quality diagram represents the engineering tradeoffs involved in making this project. To construct our house of quality diagram, we had to specify the main pillars and requirements of our project, and link them with the qualities we expect our customers to demand from this project. Understanding both these aspects will play a crucial role in determining the features of our final project model.

Among these customer specifications is safety. Since our table will be used outdoors, we must ensure our customers that we took very safe measures on all aspects, that is why we are making sure that the components that will be vulnerable to the weather are at least IP68 water and dust proof, and the components that are not will be safely protected in a waterproof box. In addition to that, all components will be tested before the implementation of the final design.

Another aspect is the price, which is a determining factor for customers. And although customers want the price to be as low as possible, they also want to know that the product is durable and well worth its price. This also applies to us as a team, as we desire the cost of executing this product to be as low as possible. However, this is an engineering trade off because low cost of production means cheaper components, and that usually means both less efficiency and quality. What we are aiming for is to make an affordable product, while maintaining as much efficiency and quality as possible.

Since our product is intended for use throughout the day, it is important that it is equipped with a battery that could last when the solar array is not producing any power. To achieve that, we will need a battery that is efficient and that provides sufficient power to the loads so they can operate at full potential. This is a key aspect because it impacts how our customers will view our product. For example, if a customer is using the power outlet to charge a cell phone, if the power delivered to the power outlet is not sufficient, the charging process might be very slow, or in a worse case, the phone might not be charging at all. Having a battery with these specifications is directly related to our budget, so we are going for a battery with a reasonable price that has the efficiency and quality we desire.

One of the goals we set for our final model of this project is to make it as easy to use as possible. Having this feature on our project will broaden the audience targeted, paving the way for its success. To achieve this, we have incorporated an app that will group all important information and features about the picnic table. To begin with, the app will allow control of the LED lights, like brightness, light color desired, and other possible features. A section will also be available for displaying information obtained through the sensors, such as the voltage and current sensors, which will allow monitoring how much energy is left in the battery so that the customers can manage their power use accordingly and also to help avoid cases where power goes down

without any previous knowledge of that, and the temperature sensor, which will display the temperature of the area.

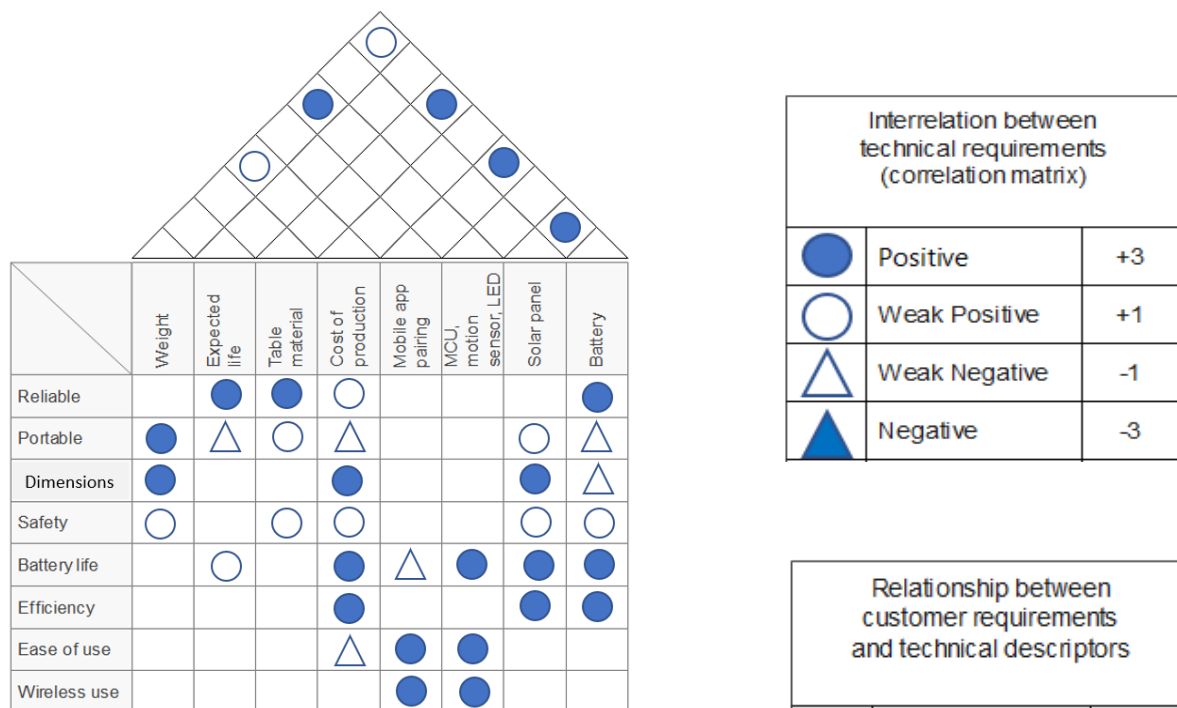


Figure 1 - House of Quality Diagram

The first row of this diagram represents the technical and material requirements of this project, and the leftmost column describes the customer specifications.

The technical requirements that are linked to a customer specification in any way have geometric shapes (square or triangle) in the box they share. There are two types of geometric shapes described in the “relationship between customer requirements and technical descriptors” legend, each corresponding to the strength or relation between the technical requirement and customer specification.

Then, we have the roof of the house of quality, which describes the interrelation between technical requirements. The technical requirements that share a relation have a geometric shape in the diamond they share. These geometric shapes are also described in the “interrelation between technical requirements” legend.

## 2.6 Microcontroller/software implementations

In this section, we will cover the methodology of how we will integrate the microcontroller and its functionalities into the solar picnic table along with the app. We will also go over the software we will be using for the mobile app and its connection.

## 2.6.1 Microcontroller

The way we are going to control features such as lighting, battery monitoring, temperature and motion sensing is by utilizing an Arduino microcontroller. To make sure the Arduino is capable of measuring these things, we need to buy and connect multiple modules to enable these capabilities. Voltage and current sensors are needed to read the outputting voltages/currents from both the solar panel and the battery. A temperature sensor is used to measure the temperature around the table. Lighting is controlled by plugging in the programmable LED lights into the Arduino and programming is based on the pins it's plugged into. For the motion sensor, we will use a PIR motion sensor that will detect whether someone is nearby and turn the LEDs on/off depending on the situation. All these functionalities will be programmed using C/C++ to create our own functions to read and display readings.

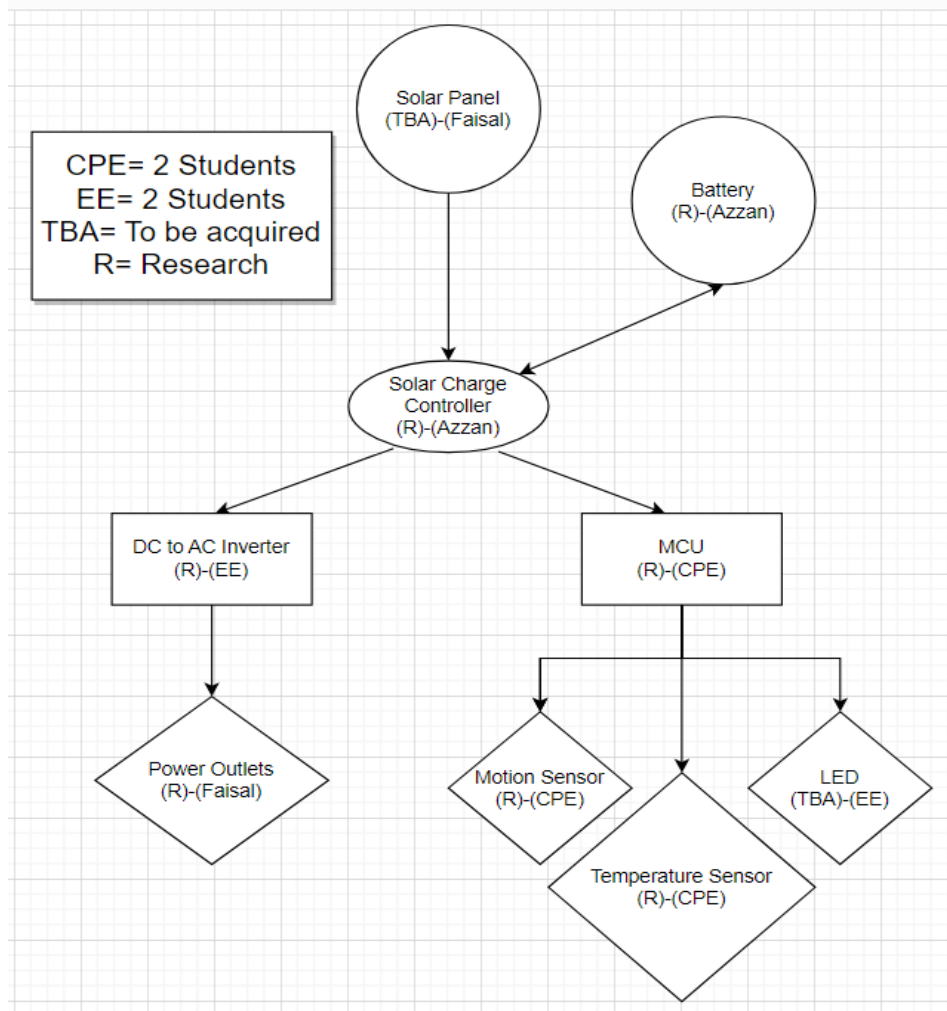
## 2.6.2 Software

To allow the user to read all the measurements and controls from the Arduino, we will implement a bluetooth module allowing connections from bluetooth devices such as phones. Our goal is to create an app that is available on any devices that allows the user to connect to the microcontroller through bluetooth. This app will provide the user with necessary information such as, displaying battery percentage, the temperature, controlling the LED lighting, and how much power the solar panel is generating. The main focus is to learn and research about both iOS and Android app development to create an application for any type of user. Both versions of the applications are going to be tested using built-in emulators in both Android Studio and XCode. Real world testing will be done using different phones provided by team members.

## 2.7 Block Diagrams

These block diagrams will demonstrate the general setup and implementation of each aspect of the solar picnic table. More specifically, it will describe the order of operations for the power distribution, microcontroller, and mobile app.

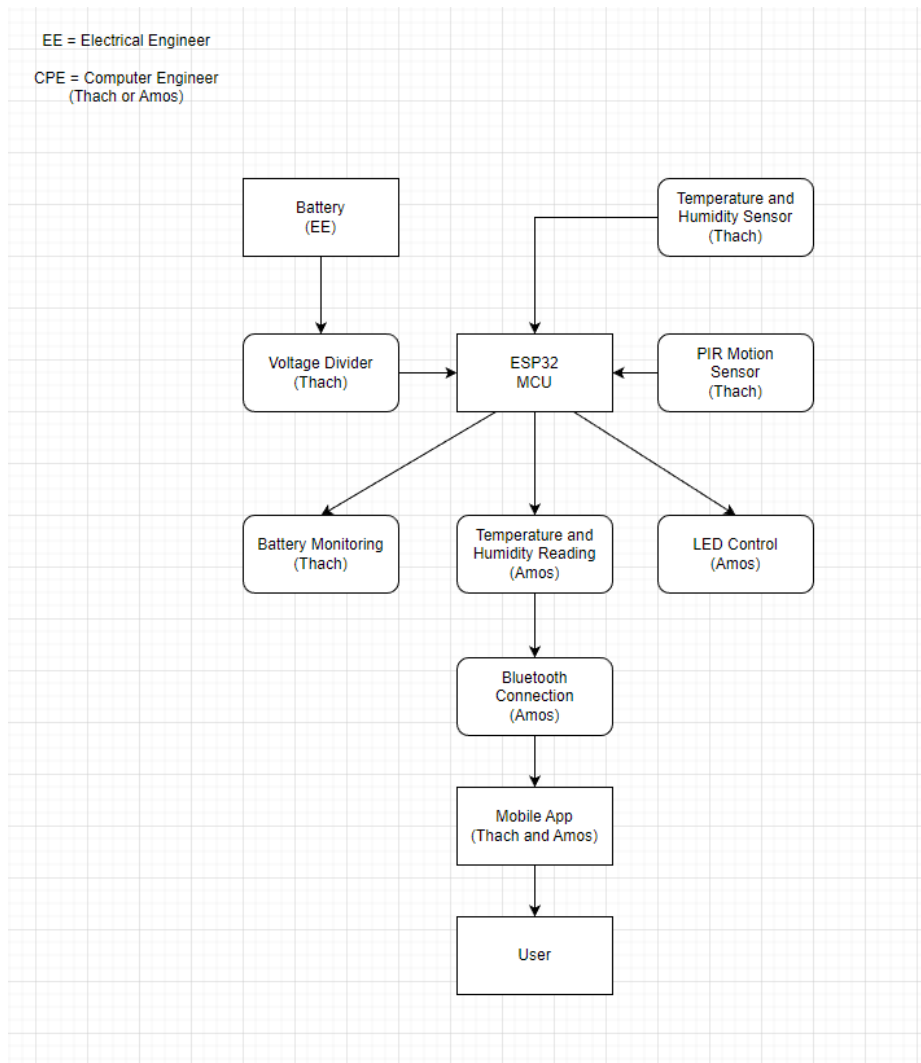
## 2.7.1 Power Distribution



*Figure 2 - Power Distribution Block Diagram*

Figure 1 is a block diagram that explains how the power generated from the PV panel is distributed. The solar panel generates electric DC current and sends it to the battery through the controller. The controller regulates the transmission of current between the two. Then, since the power outlets require AC input, the battery sends DC current produced by the PV panel to the DC to AC inverter. Here, the inverter converts DC to AC and steps up the voltage to the amount required by the outlets. On the right hand we have components that require DC input. These components are all connected to the MCU and are supplied with electric energy from the battery directly without an inverter.

## 2.7.2 Microcontroller Block Diagram

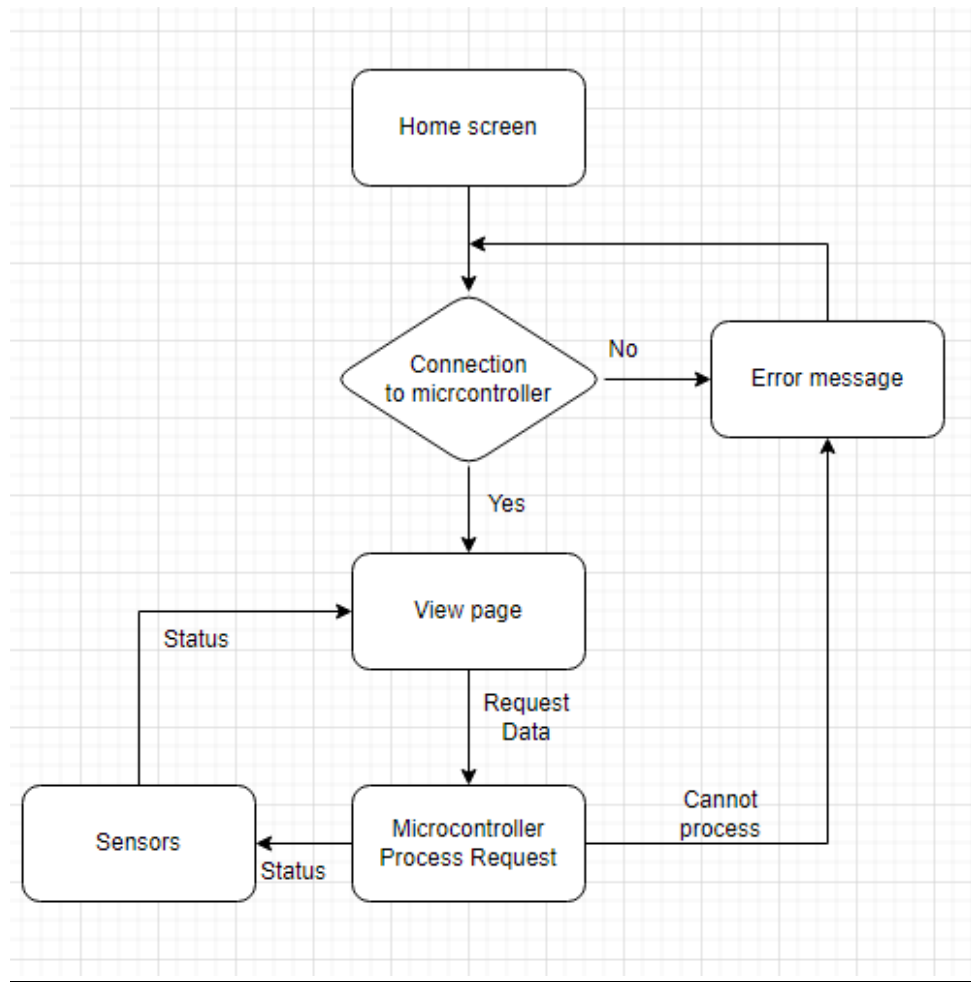


*Figure 3 - Microcontroller Block Diagram*

This block diagram shows a general overview of how the microcontroller will interact with the other components of the solar picnic table. The microcontroller will be responsible for implementing the LED light controls, motion sensing, battery monitoring, and temperature monitoring. For LED light controls, it will serve as a connection between the app and the LED lights so the user can alter the brightness at any given time. When it comes to the motion sensing, it will read in the data of the motion sensors to determine when to toggle the LEDs off and on. For battery monitoring, it will utilize a voltage divider to read in the voltage levels through an ADC pin. It will read in all of this information and output it to the mobile app for the user to read. This will be established through different types of bluetooth connection depending on the type of mobile device. The Computer Engineers will be responsible

for the research, design, and correct implementation of the microcontroller and its connected components.

### 2.7.3 App Block Diagram



*Figure 4 - App Block Diagram*

The mobile app will be compatible with both Android and iOS. The UI and API of the iOS and Android versions will be implemented using react native. The app functionalities will then be implemented using arduino IDE and the microcontroller will allow devices to connect to the app via bluetooth.

The application roughly follows the block diagram as shown in Figure 3. The user will launch the application and be presented with the home screen which will prompt the user to establish a connection with the microcontroller. If the connection is not established, an error message will appear guiding the user on what happened. If the connection is established, the user will be presented with the view page which will contain all the measurements such as temperature, battery life, etc. Once the connection is established, the app will send a request to the microcontroller to send

over all the sensor data. If the microcontroller cannot process the request for some reason the user will be prompted with another error message. If the request is processed, the microcontroller will read values and convert them to understandable values and send them to the view page for the user to see.

## 2.8 Budget Estimation

Item	Quantity	~ Price (\$)
ESP32-WROOM-32E-N16	1	1.8
LED light strips (1 meter)	4	20
PIR motion sensor	1	8.68
Lead Acid battery	1	25
Solar Panel	1	73
Solar charge controller	1	12
DC to AC Inverter	1	25
Power Outlet/USB Ports	1	13
Voltage/Current Sensors	2	Free
Waterproof box outlet box	1	15
Temperature Sensor	1	10
Estimated Total	-	203.48

*Table 3 - Depicts the Estimated Total Costs of All of the Components*

## **3. Research and Part Selection**

In this chapter, we will be going into detail about the research of different parts and picking which part is most optimal for our solar picnic table.

### **3.1 Solar panel**

Since using clean energy to power our project is one of our goals, and taking climate factors, and power needed for our loads in mind, we decided that solar energy would be the ideal form of clean energy to power our project. Solar energy is probably the most popular renewable energy source nowadays, therefore, solar panel technologies are always being researched and improved. To choose the right solar panel for our project, let us first look at some factors that shape how effective they are. The following are the most important factors that have to be taken in mind when choosing a solar panel:

#### **3.1.1 Rated Power**

The rated power of a solar panel is the maximum amount of energy the panel can produce in one hour in ideal conditions, also known as the panel's capacity. It is difficult to calculate the exact amount of energy a panel can produce, but the following equation gives is a rough idea of how much energy is produced per day:

$$P_{Rated} = ((s * 1000) * \eta * \text{number of sun hours per day}) / 1000$$

Where  $s \Rightarrow$  size of one solar panel in  $m^2$

$\eta \Rightarrow$  Efficiency of one solar panel in decimals

#### **3.1.2 Efficiency**

Efficiency stands for the rate at which solar panels can turn solar energy into electrical energy. This number is never 100% because the conditions are never perfect. In reality, the average efficiency of solar panels is around 20%. The higher the efficiency of a solar panel, the more expensive it is, because it needs less space to produce higher amounts of electrical energy.

#### **3.1.3 Material**

The material of which the solar panel is built plays a role in its efficiency. Like monocrystalline panels, which have high efficiency because they are made of a single

pure silicon crystal. Material also affects how much the efficiency of a panel drops as temperature rises.

### **3.1.4 Sun Hour**

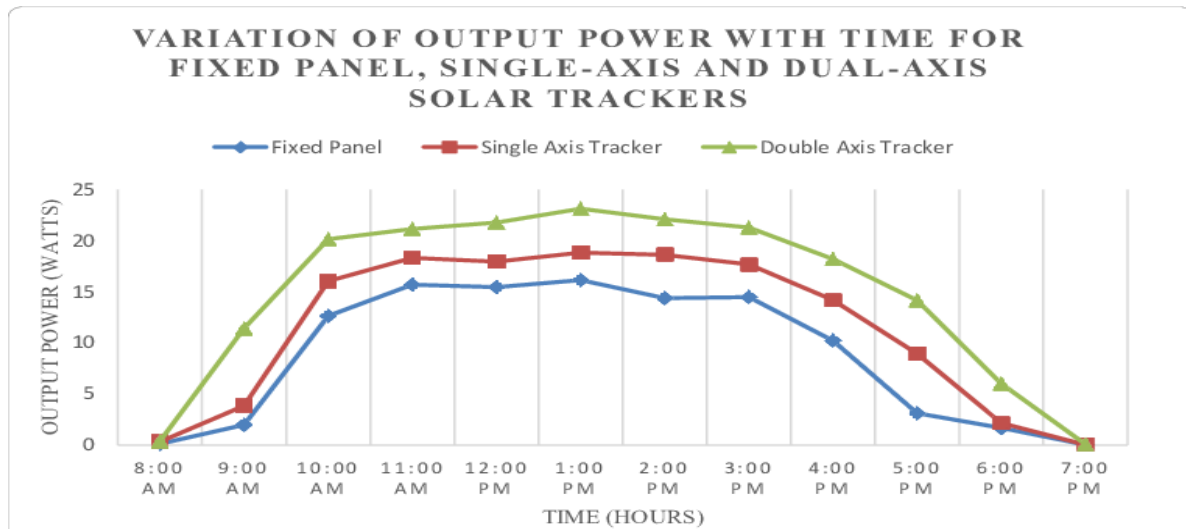
A sun hour is equivalent to 1000 W of sun radiation collected on a space of 1 square meter. Here in Orlando, for a fixed solar panel, the average amount of sun hours per day is approximately 5.4 hours. This is equivalent to collecting 1kW of solar energy per square meter for 5.4 hours.

### **3.1.5 Climate**

How often it rains in the operation area, cloudiness, dust and temperature are all important climate factors that influence the efficiency of solar panels.

### **3.1.6 Solar Tracking**

This feature enables the panels to track the sun throughout the day, maximizing the number of sun hours it can get. There are two types of solar tracking, single axis, and dual axis solar tracking. As their names suggest, single axis tracking allows the panel to track the sun in one axis direction, while dual axis tracking tracks it in all directions. To get a better idea of how tracking can improve the efficiency of panels, let's take Orlando again as an example. We said that the number of sun hours in Orlando for a fixed solar panel is about 5.4 hours, if we add single axis tracking, in an optimal case, we can add up to 1 sun hour daily, and 30% more power output annually. Adding dual axis sun tracking can add up to two sun hours per day, and increase the annual power output by 36%, adding another 6% of power output over the single axis tracker. The following figures show how both single-axis and dual-axis solar tracking works (figures 1 & 2), in addition to the difference in daily power output when applying each in comparison to a stable solar panel (figure 3):



*Figure 5 - Demonstrates a Comparison in Daily Power Output Between Fixed, Single Axis Tracking, and Dual Axis Tracking Solar Panels*

## 3.2 Types of Solar Panels

As for deciding which is the right panel for delivering enough power to our loads, we are choosing from four main types of solar panels, depending on the efficiency and material the panel is built with.

### 3.2.1 Monocrystalline Solar Panels

These panels are made from a single pure silicon crystal (which gives it its black color) that is then shaped into wafers, this process results in a lot of silicon being lost, sometimes even more than 50%, which makes the panel expensive. However, since this panel is made from pure silicon, it has a high efficiency of converting sunlight to usable power.

### 3.2.2 Polycrystalline Solar Panels

Second are the polycrystalline panels, which, unlike monocrystalline, consists of more than one silicon crystal. This significantly reduces the manufacturing losses, as there is almost no wasted silicon, thus, making it cheaper. However, this comes at the price of reducing the efficiency of energy conversion.

### 3.2.3 Passivated Emitter and Rear Cell Solar Panels

Third type is the Passivated Emitter and Rear Cell panels, or PERC panels. These are an enhanced version of the monocrystalline solar panels. In PERC panels, a layer called the “passivation layer” is added under the solar cell, which increases the efficiency of the panel. This layer reflects absorbable light to the solar cell, increasing the amount of light absorbed. In addition, Silicon in solar panels can absorb light with certain wavelengths, usually between 400nm and 1100nm. Light with higher wavelengths passes through the cells and heats up the rear sheet decreasing the efficiency of the panel. The passivation layer reflects light with higher wavelengths, preventing it from heating up the panel, thus, improving its efficiency.

### 3.2.4 Thin-Film Solar Panels

Fourth, **Thin-film** solar panels. Which are known for their flexibility, both in as a panel itself, and in size. These panels are made from different materials like Cadmium telluride (CdTe), Amorphous silicon (a-Si), and Copper indium gallium selenide (CIGS), Unlike the other types of panels which are strictly made from silicon, however, this has its impact in this type’s efficiency, as a tradeoff for flexibility. Moreover, thin-film panels do not require backing them up with a frame, which makes them easy to install on various surfaces.

The table below shows a comparison between the main solar panel types:

	Monocrystalline	Polycrystalline	Monocrystalline PERC	Thin film
Cost (per Watt)	\$1 to \$1.5	\$0.9 to \$1	Around \$1.74	\$1 to \$1.5
Material	Single pure silicon crystal	Multiple melded silicon fragments	Single pure silicon crystal + Passivation layer	Cadmium telluride (CdTe) or Amorphous silicon (a-Si) or Copper indium gallium selenide (CIGS)
Appearance	Black hue	Blue hue	Black hue	Black/Blue

Efficiency	15%-24%	13%-16%	22%-25%	7%-18%
Lifespan	Up to 40 years	Up to 25 years	Up to 40 years	10-20 years
Flexibility	Not flexible	Not flexible	Not flexible	Flexible
Temperature coefficient	-0.39%/degC	-0.38%/degC	-0.39%/degC	-0.2%/degC

*Table 4 - Comparison between the different PV panel technologies*

### 3.2.5 Decision

Due to the time and money constraint of our project model, we are going to buy a solar panel that satisfies the minimum power generation requirement. This panel has to have the capacity to power all loads needed in the day time while operating without the need of a battery. However, for our customers, we would provide a panel that has the capability not to only power all loads without battery usage, but also produces more power output that is then stored in the battery, so that the customer can enjoy the benefits provided by the table at day time, and also store power at night time usage.

Taking all factors in mind, as well as the power need that we have to satisfy for our loads, we have decided to go with a **100W monocrystalline** solar panel from **NEWPOWA**. This PV unit utilizes the 9BB solar cells technology, which aims to improve the overall efficiency of the module by lowering the lengths of currents and the output power lost. This module has a maximum efficiency of 21.3% and outputs a current of 5.26A at 19.06V in perfect conditions, which is more than enough to power all the loads connected to the table. The panel also has lower internal resistance, which prolongs its life.

The dimensions of this panel are 28.54 in x 27.76 in x 1.18, giving it a squared shape, which will make it easier to install and balance. Hardware wise, the frame is made of corrosion resistant aluminum and the panel connector is IP68 waterproof, this makes the panel perfect for outdoor use as our project is intended for that.

### 3.3 Battery Storage System

For an off-grid PV solar system, it is necessary to have a good battery storage system that fulfills the needs of our application. There are multiple technology systems in the market that we can use for our project. In this section, we will look into the characteristics of the two main batteries which are the Lithium-ion battery and the Lead acid battery technologies. We will weigh their pros and cons to finalize

a decision on what we will use. Then after the decision has been finalized, we will choose the battery storage of the specifications the system needs based on our calculations.

Here are some terms that we'll use in describing and comparing the battery:

### **3.3.1 DOD**

Depth of Discharge is the percentage of the battery's capacity that can be discharged without affecting its lifetime. It is the actual battery's capacity that can be used. For example, a battery with a 70% DOD means that we multiple 0.7 \* AmpHours to get the maximum recommended discharge rate.

### **3.3.2 Cycle life**

The number of discharge-charge cycles that the battery goes through before it starts to decrease in efficiency.

### **3.3.3 Efficiency**

It is the percentage output/input of energy through the battery. Any transfer of energy will lead to some losses. This value allows for more accurate calculations.

### **3.3.4 Power density**

The maximum power available per unit volume. This measurement allows us to quantify the battery's power supply in relation to its size and/or mass.

### **3.3.5 Energy density**

The maximum energy available per unit volume or per unit mass. This measurement allows us to quantify how much energy the battery can supply in relation to its size/mass

### **3.3.6 Charge time**

The time it takes for the battery to fully charge.

### **3.3.7 Self discharge**

It is when the battery's chemistry causes it to discharge over time even without it being connected to any load.

## **3.4 Battery Comparison**

Ideally, we would like to have a battery storage system that is safe, cheap, high efficiency, high DOD, long lifetime and cycle life, etc. But, to make the best decision on what type of battery technology we will go for in our project, we'll first need to decide what criterias are more important to us, we need to prioritize some over the others since no battery technology has it all.

### **3.4.1 DOD Comparison**

Lithium-ion batteries usually have a much higher DOD than Lead acid. With Lithium-ion, the DOD is around 80-90%, whereas with Lead acid it is only around 20-40%. Specifically for our application which will require the battery to be able to supply power to the system throughout the whole night it must be able to either a high DOD or a low DOD with a higher capacity storage. This is the case unless we are okay with affecting the battery's lifespan.

### **3.4.2 Energy & Power Density**

Energy density and power density are not the same. A high energy density storage system means that it can do work for longer time periods, whereas a system with a high power density will be able to generate more power but at a shorter time period. Both variables are a ratio of volume/mass.

In our application, we prioritize energy density over power density. The system will require a constant power supply in moderate amounts rather than in large bursts.

Lithium-ion batteries have an energy density of 200-500 Wh/L and a power density of 1.5k-10k W/L. On the other hand Lead Acid batteries have an energy density of 50-90 Wh/L and a power density of 10-400 W/L. We can see that Lithium-ion is better in both metrics.

### **3.4.3 Self Discharge**

Self discharge varies with temperature. At higher temperatures the rate increases. But at normal conditions, both batteries have a similar low average self discharge rate, which is about 0.1-5%.

### **3.4.4 Charge Time**

The charging time of Lead acid batteries is significantly more than that of lithium-ion. Lead acid batteries can take up to 10 hours to fully charge, whereas lithium-ion batteries would only need 4 hours to charge the same amount. It is obvious that lithium-ion is the better choice in regards to charging time. But since our project does

not require the battery storage system to have fast charging capabilities due to the exposure to sunlight for long hours, this variable is not a priority.

### **3.4.5 Efficiency**

It is very important to know the efficiency percentage of the battery before deciding which option to implement in our design. The efficiency value is an important part of the calculations section. It resembles the effective battery capacity. Lithium-ion batteries have higher efficiencies than Lead acid. Lithium acid batteries efficiency typically range from 90-99% whereas Lead Acid batteries only go up to 85%. Efficiency is not as important a factor as it may seem in our decision of battery technology. If we go for a battery with less efficiency then we will get one with a higher capacity. That is after we weigh the cost of both cases and go with the cheaper option.

### **3.4.6 Maintenance Requirements**

It is important that the battery technology of our choice does not require regular maintenance. Our customers shouldn't have to always check on the battery's condition. Lithium-ion batteries are designed to operate in such a way that it would require very bare minimum or no maintenance over a long period of time. Lead acid batteries on the other hand are very high maintenance, and they also require a special maintenance room as they emit toxic fumes while charging. We are not expecting our customers to be able to regularly maintain the battery as it is costly and tedious. This is why it is a priority for us to have a low maintenance battery storage.

### **3.4.7 Cycle Life & Lifetime**

The Lifetime of a battery is directly related to its Cycle life. The greater the cycle life the more lifetime of the battery in years is. On average, the cycle life of Lithium-ion batteries is 2-4 times that of the Lead acid. Not only that, Lead acid batteries require regular maintenance to last it's expected cycle life. The battery storage system of our project is expected to be in constant state of charging and discharging for long periods of time (days). So, it is very important that we have a battery that can withstand constant use without dying out.

### **3.4.8 Weight**

The weight and size of the battery are important factors to make the picnic table as convenient as possible. The weight of Lithium-ion is typically about 50-60% lighter than Lead acid batteries. These metrics are important but not a top priority since the battery will be held by a waterproof box.

### 3.4.9 Cost

Lead acid batteries are generally cheaper than lithium-ion. Since the battery is one of the most expensive parts of our project, the cost is going to be a huge factor and one of the priorities in our decision making. Our aim is to choose the best technology for a reasonable price.

	<b>Lithium-Ion</b>	<b>Lead acid</b>
<b>DOD</b>	More	Less
<b>Power Density (W/L)</b>	More 1.5k-10k	Less 10-400
<b>Energy Density (Wh/L)</b>	High 200-500	Low 50-90
<b>Charge time</b>	Fast 1-4 hours	Slow 5-10 hours
<b>Efficiency</b>	90-99%	80-85%
<b>Cost</b>	Expensive	Cheaper
<b>Self Discharge</b>	Low	Low
<b>Safety</b>	Safer	Less safe
<b>Weight (Same capacity)</b>	Lighter	Heavy
<b>Maintenance requirements</b>	Low	High
<b>Cycle life</b>	2-4x	1x
<b>Lifetime</b>	10-15 years	3-12 years

*Table 5 - Comparison Between Lithium-ion and Lead Acid Battery*

### 3.4.10 Decision (Lead Acid Battery)

From our comparison above, it is very clear that Lithium-ion batteries are generally better than Lead acid in all metrics except for cost. For our application, the best battery technology option is Lithium-ion. Our solar picnic table should have a battery storage system that has high energy density, is efficient, safe, and requires little to no maintenance over the years. But, due to the budget constraints and our project being only a model for our customers to view, we have decided to use the Lead acid battery instead for our project since it is cheaper. The Lead acid battery will only be used for

modeling purposes but a Lithium-ion battery will be used for the customer’s final product.

### 3.5 Lead Acid Battery Types

There are different types of Lead Acid batteries and the following table summarizes the comparison between them. The Lead Acid battery type will be decided based on this comparison.

Type	Maintenance Requirement	Lifespan	Weight	Cycle Life	Cost
Flooded (Liquid)	Regular	3-5 years	Heavy	Moderate	Cheapest
Sealed (Gel or AGM)	Not required	5-7 years	Light	Moderate	Cheap
Deep Cycle (Liquid)	Ocasional	4-8 years	Heavy	High	Expensive

*Table 6 - Comparison Between Lead Acid Battery Types*

For our application, the choice that makes the most sense is the Sealed Lead Acid battery for numerous reasons. It requires little to no maintenance, it is light, and has a high lifespan, and a decent cycle life. Although it is not the cheapest it is still considerably cheap and the tradeoff between the features listed with the cost is worth it.

### 3.6 Battery Parameters

First of all, we need to identify our minimum requirements for battery storage. Due to the budget constraints, we have decided to model our project so that the battery storage is able to provide enough power to all of the loads in the system simultaneously for at least an hour to showcase the final product. But this is only the case for our model project. For our customers, we would design it where the storage system can provide enough power to the system throughout the whole night when the PV panel is no longer in use. In this document, we will prove all the information and specifications of the customer’s version of our solar table.

Based on our calculations in the calculations section, we came to the conclusion that we will use a 12v battery with 8Ah. After some research online we have decided to go with the Lead acid battery **ExpertPower EXP1280**.



*Figure 6 - The EXP1280 Model*

This table provides the battery parameters from its official datasheet:

<b>Model</b>	EXP1280
<b>Type</b>	Sealed Lead Acid
<b>Voltage</b>	12V
<b>AmpHours</b>	8Ah
<b>Cycle life at 100% DOD</b>	260
<b>Weight</b>	2.4Kg $\pm$ 5%
<b>Dimensions LxWxH (mm)</b>	151x65x94
<b>Self Discharge at 25° per month</b>	3%
<b>Optimum Temperature</b>	25°
<b>Cost</b>	~\$25

*Table 7 - Battery Specification Parameters*

### 3.7 Charge Controller

A charge controller is a device that is linked to a battery to control the amount of charge the battery receives. Since all batteries have a charge capacity limit, charge controllers are needed to ensure that that limit is not crossed, nor that the battery is over-drained.

To do that, two voltage levels are determined: the first is a High Voltage Disconnect point (HVD), at which the battery is full, and the charge controller is signaled to stop the current supplied by the source from reaching the battery. The second one is a Low Voltage Disconnect point (LVD), at which the battery needs to be charged. Here the charge controller reconnects the charging circuit.

## **3.7.1 Functions of a charge controller**

In this section, we are going to discuss the main functions of a charge controller, types of charge controllers incorporated with solar systems, and then compare and choose the charge controller that would fit our project.

### **3.7.1.1 Overcharge protection**

Simply put, if the charge contained in a battery is beyond its capacity, the battery is overcharged. Overcharging a battery can lead to several issues. Firstly, overcharging can cause the electrolyte in the battery to split and release oxygen and hydrogen gasses, which can lead to an explosion and failure of the battery. Secondly, as the voltage of the battery increases more than it is supposed to, the inverter incorporated with the solar system might shut down or get damaged, as well as the loads supplied by this battery. Thirdly, when the solar panel connected to the battery is producing low power or none at all, or the loads connected to the system are heavily used, the battery voltage drops down, the charge controller sets a maximum charge to be supplied in this case, protecting the battery from being overcharged.

### **3.7.1.2 Over Discharge protection**

The lifespan of a battery relies upon its depth of discharge (DoD). Whenever a battery is drained below a certain voltage level, it is over discharged, and if left in that state, the battery's life expectancy is lowered and it might even get damaged. One of the functions of a charge controller is to prevent that from happening by disconnecting the battery from the loads of the system if the battery gets to this state, and then reconnecting them when the battery has sufficient charge to supply the loads.

### **3.7.1.3 Reverse current flow prevention**

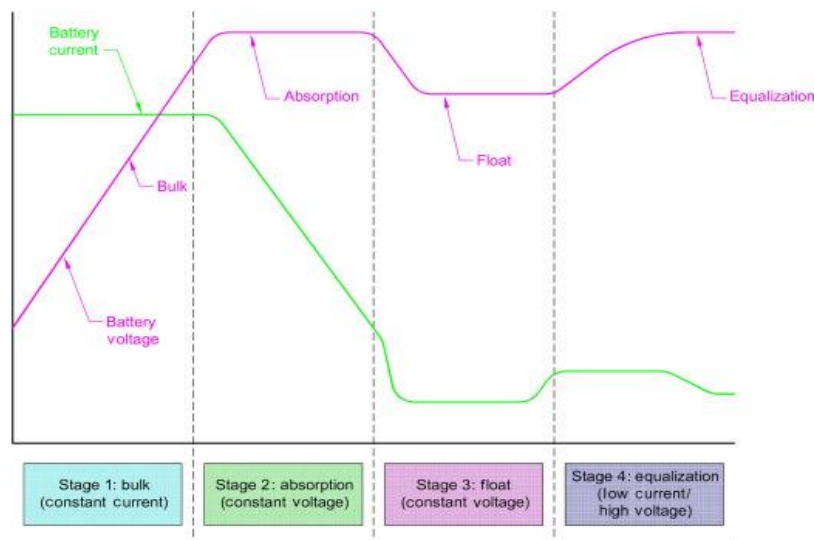
Once night time begins, solar panels stop producing power and their voltage drops to very low levels, while batteries maintain theirs according to their charge level. This causes the battery to get drained by the solar panel, as the current flows in the opposite direction. Charge controllers prevent this reverse current flow, as they allow the current to flow in one direction only using diodes.

### 3.7.1.4 Overload protection

Circuit overload occurs when a current that is too high is flowing in the circuit. This can be caused by either a short circuit in one of the wires, or an error in one of the devices receiving the current. Overload can cause the circuit components to overheat or fry. Charge controllers protect solar systems from being overloaded with current, as they contain a circuit breaker or a fuse that prevents that from happening.

### 3.7.1.5 Charge Controller Operation

To have a better understanding of how a charge controller operates, this section will demonstrate the stages that it goes through as it operates on a battery receiving charge from a solar panel. And since we are using a lead-acid battery in our initial model, let us look at that specifically.



*Figure 7 - Stages of a charge controlled lead-acid battery charging process*

The figure above shows the stages that a charge controller paired lead-acid battery goes through when being charged: Bulk charging, absorption charging, float charging, and equalization. In the following, we discuss the functions of the charge controller in each stage.

#### Stage 1: Constant Current (Bulk) Charging

At this stage, our lead-acid battery has a low charge level. The charge controller connects the charging circuit and lets the battery receive a continuous current at a high voltage. The battery keeps receiving this constant current until it gets about 80%-90% of its capacity filled.

#### Stage 2: Absorption Charging

Since the battery just left the constant current charging stage, its voltage is at its highest, however, to reduce the current received by the battery, the charge controller increases its internal resistance.

### Stage 3: Float Charging

Here the lead-acid battery is almost fully charged. If the battery reaches this state and it is not being used by the loads, there is a possibility that a small amount of charge leaks out of the battery, in this case, the charge controller allows small pulses of charge to maintain the charge level of the battery by using a feeding voltage just above the battery's voltage.

### Stage 4: Equalizing the battery

In lead acid batteries, there are chemical effects that occur as the battery is used, such as sulfation. Equalizing the battery is a process that aims to reverse these chemical effects by intentionally overcharging the battery to maintain its health and prolong its lifespan. Battery sulfation for example, is when sulfate crystals are built up in the battery, which causes a reduction in the ability of the battery holding and controlling its charge. This is treated by desulfation, which is the process of overcharging the battery so that the current runs through the built up sulfate crystals and causes them to dissolve in the battery acid. Although this stage is called a charging stage, the battery does not go through equalization in every charge cycle, several times per year are enough to maintain its efficiency and lifespan.

## **3.7.2 Charger Controller Types**

In this section, we will go over the different types of charge controllers and discuss their different operation mechanisms. There are two types of charge controllers used in solar systems: The Pulse Width Modulation (PWM) charge controller, and the Maximum power Point Tracking charge controller (MPPT). We will explain how each one works and compare their differences, and then, decide which best fits our project.

### **3.7.2.1 Pulse Width Modulation (PWM) charge controller**

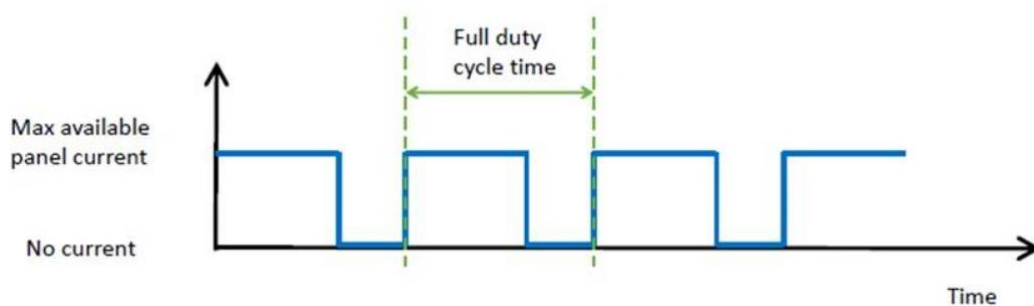
The PWM charge controller has to be directly connected with both solar panel and battery to operate. Like series charge controllers, it is connected in series with both of them. However, the PWM charge controller uses a transistor instead of a relay, This allows for the very high frequency of connecting/disconnecting the charging circuit. When using a PWM regulator, the voltage of the solar panel must be compatible with the battery's voltage, therefore, this type of controller is typically used in small PV systems, as it is very challenging to satisfy this condition in large systems. The "width" in its name stands for the length of the pulse of charge allowed to be delivered to the

battery, so when the width is at 100%, this means that the battery's voltage is relatively low compared to the PV voltage and it needs to be charged. On the contrary, when the width is at 0%, this means that the battery is at full charge. The PWM charge controller does not have the ability to lower the amount of current going to the battery from the PV, that means that the battery either receives the full charge or none, however, since a transistor is used, the PWM regulator has the ability to rapidly send "pulses" of charge to the battery, which makes it almost control the current flowing to the battery. Utilizing that feature, PWM regulators go through two stages of regulation:

The first stage is when the battery's voltage is low, so the PWM uses the high voltage difference between the battery and PV to supply it with a steady current to charge it to a certain charge level.

The second stage happens after the battery is fully charged, where the PWM will use the low voltage difference between the battery and PV to send very fast and frequent pulses of charge to maintain the charge level of the battery and resupply the battery with any charge that could have been lost while not using the loads.

The following figure shows current flowing to a battery through a PWM controller with a 70% on, and 30% off duty cycle, which means the battery is being supplied with current for 70% of the duty cycle duration, and then the charging circuit is disconnected for 30% of the duty cycle duration.



*Figure 8 - Demonstration of a 70% On, 30% Off Current "Pulses" Sent From a PWM Regulator to a Battery*

In the figure above, we can see the pulses sent via the PWM to the battery. The frequency of these pulses changes according to the charge level of the battery, or the demands of the loads, and this is a feature in the PWM charge controller, as its frequency can get very low, where it operates like a series charge controller. This feature is important when supplying loads that are affected with the noise caused by the frequency of the PWM.

Using a transistor as the switching element in the PWM charge controller has its disadvantages. One of these disadvantages is heating problems. Transistors can easily get overheated, and so are PWMs since they use them, therefore, PWM charge

controllers must be incorporated with a method for heat dissipation, such as ventilation, or heatsinking.

### 3.7.2.2 Maximum Power Point Tracking (MPPT) Charge Controller

Efficiency wise, the MPPT charge controller outperforms the PWM charge controller. Due to several factors, like temperature and amount of solar radiance being received by the panel, the maximum power point on the current-voltage curve is constantly changing. This type of regulator closely tracks the change in voltage at maximum power point, then, the DC voltage and current received from the solar panel are converted to an AC voltage and current that has a high frequency, then, both voltage and current are reconverted to DC. The goal of going through this operation is to convert the voltage received from the PV module to the exact voltage of the battery and vary the current to keep the power level constant. Let's take an example to comprehend how a MPPT works. Assume that you have PV that is capable of producing 18V at 3 Amps, this PV module is connected to a battery with a voltage of 12 V. Assuming perfect conditions, the PV is producing maximum power. The MPPT will convert this 18V 3A output from the panel to a 12V 4.5A input to the battery, stepping down the voltage and increasing the current being delivered to that battery. Notice that this type of regulator allows the solar panel voltage to be different from the battery voltage, which makes it a good choice for large PV systems where it is hard to make the PV voltage compatible with the battery voltage. On the other hand, MPPT charge controllers are solid state, which means they produce heat that has to be somehow dissipated, which is an issue, especially for large systems.

The following table compares the PWM charge controller to the MPPT charge controller based on their efficiency, connection relative to the solar panel and battery, ground pole connection and other factors:

	PWM	MPPT
Efficiency	75%-80%	93%-97%
Price	\$10-\$60	\$30-\$200
Connection type	Series	Series
Battery-PV voltage	PV nominal voltage must be same as battery voltage	PV voltage can differ from battery voltage
Heat dissipation requirement	Required	Required

Ground polarity requirement	Positive or Negative	Negative
System size	Small systems preferred	Large or small systems
Critical voltage points	Predetermined	Not predetermined

*Table 8 - Comparison Between PWM and MPPT Controllers*

### 3.7.3 Decision (DEPVKO PWM)

Due to our budget constraints, as well as our project being a model for our final product design, we are choosing the PWM charge controller. This regulator provides enough efficiency for our model, and since the system we are building is small, it is easy to get a battery that is compatible with a 12V solar panel. That being said, we are going to go with **DEPVKO**'s 30A PWM charge controller. This charge controller is compatible with both 12V and 24V solar panels and batteries and comes with a built-in microcontroller that memorizes the parameters set by the user.

## 3.8 Inverter

In order to supply alternating current (AC) to the power outlet, we must convert the direct current (DC) coming from our source into AC then step up the voltage to 120 volts. That is the required input by our power outlet.

In the past, we would have needed to use both an inverter and a transformer separately in our design. But since the requirement of having higher AC voltage levels increased, they started to implement transformers in most inverter designs.

We are now going to look into only some of the different inverter technologies available in the market since there are many and then compare which one fits our needs the most. The list below shows the three most common inverters used for solar PV applications.

### 3.8.1 Different Inverter Technologies

- 1 - Pure Sine Wave Inverter
- 2 - Microinverter
- 3 - Grid-tied Inverter

#### 3.8.1.1 Pure Sine Wave Inverter

The pure sine wave inverter outputs the smoothest and cleanest sine wave alternating current. It outputs a very high quality power, compatible with a wide range of

applications, and has high efficiency. This inverter is common for applications that supply the AC current to sensitive electronics like smartphones and laptops which is what we intend to do.

### **3.8.1.2 Microinverter**

The microinverter is an inverter that is specifically designed to be installed directly onto a single PV panel. This allows for a better configuration to be tailored to the specific PV panel and allows for easier installation.

### **3.8.1.3 Grid-tied Inverter**

The Grid-tied inverter is used for grid tied solar PV systems. It synchronizes the output of the AC current from the grid with the one produced by the PV panels. One of the main advantages of this inverter is that it allows for feeding of excess power into the grid.

## **3.8.2 Inverter Comparison**

Since we are designing a stand-alone off-grid solar PV system, we are only going to compare the Pure Sine Wave inverter and the Microinverter.

### **3.8.2.1 Output waveform**

Pure sine wave inverters produce a better quality AC current than of the microinverter. Its waveform is similar to that of the grid's AC current. Having a high AC's waveform quality is important it allows the electric system to avoid or minimize the effect Harmonic Distortion, which is an issue where there is a mismatch in electrical frequencies which can result in many issue like overheating, power loss, miscommunication, etc.

### **3.8.2.2 Performance**

Generally, Pure sine wave inverters are more efficient than microinverters. They are also more reliable and have a higher lifespan. Microinverters can handle larger power output but require more maintenance as they are much more complex in design and are tailored to one particular panel.

### **3.8.2.3 Installation**

Pure sine wave inverters can be installed and connected to multiple PV panels at the same time, while microinverters can only be connected to one. But this means that the pure sine wave inverters have to be installed close to all panels, whereas with microinverters it can be attached to the panel. Typically pure sine wave inverters are

easier to install since only one is required for multiple PV panels, but since we have only one PV panel it is going to be easier for us to install the microinverter.

### 3.8.2.4 Cost

Typically microinverters are more expensive since they can only be installed to one PV panel. In our case that is not a factor. For larger solar systems having a pure sine wave inverter would require an optimizer and combiner to go with it, but again this is not our case. In our case, the cost will depend on the other parameters like rating, efficiency, and the quality of the brand irrespective of the chosen technology.

### 3.8.2.5 Technology Decision

After analyzing the differences between the two, we have come to the conclusion that the pure sine wave inverter is a better choice for our project. Normally the drawbacks of the pure sine wave inverter would be that it cannot be modified for a specific panel like the microinverter, but since we only have one panel then we do not have to worry about that. Specifically for our application, having a pure sine wave inverter makes more sense especially because of its quality output and the devices it will supply power to.

### 3.8.3 Inverter Parameters

According to our calculations, the minimum power rating of the inverter must be 91W and of 12V. After researching some of the available pure sine wave inverters products we have come across two choices. The first one being the **BESTEK MRI3011BU**. Which is a high quality DC to AC inverter that has many great features such as AC output port and a built in DC charging output port. It also has ventilation and a cooling system.

The specification parameter of this inverter are listed in the following table:

<b>Model</b>	MRI3011BU
<b>Type</b>	Pure Sine Wave
<b>Rated Total Power Output</b>	300W
<b>Rated DC Input</b>	12-15 VDC
<b>Rated AC Output</b>	110-120 VAC
<b>Rated DC Output</b>	5 VDC
<b>Efficiency</b>	≥83%
<b>Dimensions LxWxH (mm)</b>	119x78x46

<b>Weight</b>	480g
<b>Optimum Temperature</b>	5-40°
<b>Cost</b>	~\$25

*Table 9 - Specification Parameters of the MRI3011BU Model*

The other inverter is the **UEIUA B0774SCCND** power inverter. The one is cheaper than the previous one but has less features. The power ratings of this inverter still meet our minimum requirements. The specifications for this model are listed below.

<b>Model</b>	B0774SCCND
<b>Type</b>	Pure Sine Wave
<b>Rated Total Power Output</b>	150W
<b>Rated DC Input</b>	12-15 VDC
<b>Rated AC Output</b>	110-120 VAC
<b>Rated DC Output</b>	5 VDC
<b>Efficiency</b>	85%
<b>Dimensions LxWxH (mm)</b>	80x65x40
<b>Weight</b>	240g
<b>Cost</b>	~\$15

*Table 10 - Specification Parameters of the B0774SCCND Model*

After comparing both products, we came to the conclusion that since the second product (B0774SCCND) meets our minimum requirements then it is logically the option we will go for. Yes we will be missing out on some of the features that the MRI3011BU model provides but we don't think that these features are worth the extra money. So the model of choice is the UEIUA B0774SCCND inverter.

Since both of these inverter models have an output port as a cigarette lighter port then this means that we would require a connector to then input the power outlet using two-terminal wires. More information about the wiring design will be in the design section of the report.

## **3.9 Power outlet**

One of the main features that our solar picnic table has is that it is equipped with power outlets that will supply power to devices connected. This feature is one of the most important features of the table. To make sure that we provide the best power supply, we need to look at the different types of power outlets then decide which one to implement into our design. We are also going to look at the power delivery needs of the power outlet. What is the power rating of it, the voltage, and the current drawing.

### **3.9.1 Types of Power Outlets**

- 1 - AC Outlet
- 2 - DC Outlet
- 3 - USB Outlet

#### **3.9.1.1 AC Power Outlet**

AC outlets are the most common type of power outlets. They are used in almost all households. These outlets supply an alternating current with a voltage of 120-240 volts depending on the country you are in. Today, most of our devices require DC supply. So the chargers that come with our devices contain an adapter that will convert the AC supply into DC and step down the voltage to the device's requirement.

AC outlets have many different shapes. Depending on the number of prongs and their placement on the socket. These shapes are named by the alphabet letters A, B, C, and so on. Different countries use different types of AC outlets. Here in North America we use Type A and B. Type A has two prongs while type B has three.

#### **3.9.1.2 DC Power Outlet**

DC power outlets provide direct current. Devices that require DC supply can directly connect to the outlet without the need of adapters. Unlike the AC power outlets, DC outlets are not very common in our day to day usage. These outlets are mainly used in vehicles such as the cigarette lighter outlets. The adapters for these outlets are mostly in a round cylindrical shape.

#### **3.9.1.3 USB Power Outlet**

Universal Serial Bus power outlets commonality has increased in the past years as most smart phones today have USB charging wires. USB outlets provide direct current DC directly to our devices through the USB adapter. It is a standardized protocol that USB outlets provide 5 volts for power supply. The maximum current supply varies from outlet to outlet.

## 3.9.2 Decision (Type B)

Our aim is to provide the most convenient features to our solar table as possible. Due to the fact that most people today use AC adapters and/or USB adapters to charge and power their devices, we have decided to provide power outlets for both. We are not going to use a DC outlet in our design.

The AC outlet will supply an AC current at 120V which is the standard in the US. The USB outlet will supply a DC current at 5V which is the standard voltage for USB charging purposes.

For convenience, we have found and decided to use a single outlet that will combine both an AC outlet port and a USB one. This is possible because there are two separate circuits within the outlet which allows the supply of the two types of current (direct and alternating) without interference. One circuit converts the AC input into a lower voltage DC then outputs that through the usb port, the other circuit outputs the high voltage AC through the AC port. The AC outlet is Type B which is the most convenient one since type A adapters can be plugged in too. The part seen in the figure below is a **Baomain USB Charger Outlet**. It contains two ports of Type B and USB. The AC port is **120VAC** and can supply up to **20 amps**. The USB port is **5VDC** and can supply up to **3.1 amps**.



*Figure 9 - The Baomain Power Outlet*

## 3.10 LED

For our table to be used at night, lighting the area covered by the table will be an essential task. To get that task done, we have decided to use LED lights. We are specifically choosing LED strip lights for our project, that is because they are flexible, can be programmable with microcontrollers, and give a modern look to the final model of the project. Light-emitting diode (LED) lights are defined with characteristics that go hand in hand with our project goals, we list some of them as follows:

### **3.10.1 Efficiency**

LED lights are the most efficient lighting method known today, and since the table is powered by a battery at night, this will help conserve as much power as possible. The average LED strip light consumes about 4W/ft, taking in mind that the least LED strip light density is 9 LEDs/ft and can go up to 36 LEDs/ft. Compared to incandescent light bulbs, which consume about 60W of power on average, LED lights have a far better efficiency.

### **3.10.2 Flexibility**

Flexibility is an important feature in modern designs, as it gives them a modern look and a wider range of use. For our project, this will help us manipulate the lighting of the table with ease, by sticking the lights on top of the table to light up the area of the table and its surroundings. Also, flexibility allows us to accumulate more light on areas that lack it as needed.

### **3.10.3 Programmability**

Since we are controlling the lights on our table with a microcontroller, it is crucial that our lights are programmable and easy to use by our customers. LED lights provide a wide range of programmable features ranging from turning the lights on/off, to controlling each LED unit from our microcontroller code, depending on the LED strip model. This is also a key feature for linking our lights with the app we are linking our microcontroller with.

### **3.10.4 Keeping it green**

One of our key goals to achieve is providing our customers with their needs when going out for a picnic while using as many sustainable components as possible, and the LED lights only bring us one step closer to achieving that goal. LED lights have an average lifespan of 4-6 years, compared to 1000 hours average life for incandescent light bulbs, which is nowhere near the lifespan of LEDs. longer lifespans means less waste, and therefore, more sustainability. Furthermore, LED lights are free of some dangerous gasses that some other types of lights are made of, like Mercury, which poses a threat not only to our nature, but also to our customers.

## **3.11 Programmable LED types**

In order for our LED lights to be microcontroller programmable, a data wire has to be connected from the LED strip to the microcontroller. There are two types of LEDs to compare that come with this feature. Our choice then will be based on the functionality, price, and voltage requirements. These two types are:

### 3.11.1 SMD5050

Usually, the SMD5050 LED strip lights come with a controller to control them, however in our case, we are using a MCU to do that as they come with a data wire that can be paired with one of the MCU pins. These LEDs require a 12V source to be connected to. And since the voltage they take is high compared to our microcontroller voltage, a MOSFET and some resistors will be required in between the wire connection between the lights and the MCU. This type of LED is good for ambient lighting, as individual LED housing unit control is not possible. As for their price, SMD5050 lights cost about 1\$/m, or 0.3\$/ft.

### 3.11.2 WS2811/12/12B

This type of LEDs lights are also referred to as Neopixels, as they enable individual LED housing unit control. These LEDs require a 5V power supply, which means they can be powered by the microcontroller itself, however, to eliminate chances that could get our MCU chip fried , if we go with this option we are going to power it using our battery. Like the SMD5050 lights, the WS2812B LEDs also come with a data wire that is MCU pairable to control them. Since LEDs can be individually controlled, this type has a higher price, they go for about 4\$/m or 1.22\$/ft.

The following table sums up the differences between the two LED strip light types mentioned above:

Model	SMD5050	WS2812B
Rated DC input	12VDC	5VDC
Compatibility	MCU programmable	MCU programmable
Price(\$)/ft	0.3\$/ft	1.22\$/ft
Individual LED control	Not available	Available

*Table 11 - Comparison between SMD5050 and WS2812B LEDs*

### 3.11.3 Decision (WS2812B)

We decided to go with the WS2812B LED strip lights for multiple reasons. The main reason is because it requires less rated input voltage, which makes it safer for MCU programming. In addition to that, the PCB we are designing to deliver power to this component will have a voltage regulator that supplies both 3V and 5V since all other components supplied by the PCB require these two different voltages. That said, we decided to get the **ALITOVE WS2812B** LED lights to incorporate in our project.

## **3.12 Mobile App Connection**

Our goal for the mobile app connection to the solar picnic table is to allow any person with a mobile device to connect to their current table and monitor necessary information such as how much battery percent is left on the table. Developing a mobile app also helps diagnose the table if anything is malfunctioning by showing readings from the microcontroller such as how much the solar panel is generating without being next to the table and reading the values from the solar panel device.

When developing a way to connect to the microcontroller, we have two main options which are utilizing bluetooth or wifi. With these two methods, both of them have their own limitations that determine the usefulness of each method.

### **3.12.1 Mobile App Connection Comparison**

We will be comparing the two forms of connection: Wi-Fi, and Bluetooth. We will delve into each of their strengths and weaknesses. From there we will compare the two to see which one is the most suitable for our project.

#### **3.12.1.1 Wi-Fi**

When it comes to wifi, the microcontroller needs to either connect to an existing network to share information transferred onto an full stack application or create its own local network where the user can connect to it on their mobile device and access a local application. A serious drawback of using wifi is being in range of an internet connection. This problem comes into play if we implement this solar picnic table into somewhere where that could be an issue such as a park. A solution to this could be adding cellular data to this but increase the budget and maintenance of the table. Another issue that comes with wifi is energy management. If the table is to connect to another network, it would have to constantly send signals to the application where it is hosted to get the most up to date information. Internet connection would also take a considerable amount more of power compared to bluetooth, especially if we use bluetooth in low energy mode. We can delay each request of information but that would still take a considerable amount of power when compared to bluetooth.

If we implement wifi into the table with ideal conditions, wifi has a few benefits. Having wifi allows for longer distance connections compared to bluetooth and allows in the future for a network of tables to be managed remotely. Long distance communication allows for easy access to the users. For example, if a student is about to walk to a table on campus from class, they could check the app which would show any table connected to the network. With this connection, while the student is walking towards the table, they can check if it has the power it needs to charge their laptop while they are outside. If not they can go to another table or go inside. Wifi also has a more considerable speed difference compared to bluetooth. However in our application, speed is not as important since the data sent between each device should not be that

large. Wifi also provides more secure options to make sure no unwanted connections are made to the microcontroller.

### 3.12.1.2 Bluetooth

When it comes to utilizing bluetooth, some of the limitations are solved. When using bluetooth, the table does not need to be in range of any networks which allows the table to be placed anywhere. All that is needed for connection is two bluetooth devices which would be the table and the user's mobile device. This simplifies the connection process compared to the wifi router. A downside to bluetooth when compared to wifi is that bluetooth has a considerably slow speed compared to wifi. However since we are only transferring a small amount of data (voltage/current and power left in the battery), the low transfer speeds do not affect us. Bluetooth also has another drawback with the number of connections it can make at one time, however there is not a high chance that more than 4+ people are connected to a table at the same time. There is a slight security risk when it comes to using bluetooth however we can add limitations to the microcontroller to only have the application control a certain section such as controlling the LED lights. The security risk is not as detrimental to us however since there is not valuable data someone would want to steal from a solar picnic table.

Bluetooth has a lot of benefits that will be a great value to our project such as the low power consumption. The low power consumption, especially in bluetooth low energy is very beneficial to us since every part of the table that draws in power needs to be considered to have the most optimal power management to save as much energy for the user. Bluetooth also has a low chance of being interfered with other connections around the table. In a highly populated area with a lot of network connections, bluetooth will provide us with stable connections to quickly check the battery health of the table.

WiFi	Bluetooth
Longer range	Lower Power consumption
Higher speed	Low Cost
Better security	Ease of use (connection wise)
More connections	

*Table 12 - Benefits of WiFi & Bluetooth*

WiFi	Bluetooth
More Expensive	Lower range

High power consumption	Less connections
Needs existing connection	Lower speeds

*Table 13 - Downsides of WiFi & Bluetooth*

### 3.12.1.3 Decision (Bluetooth)

After comparing both methods of connecting to the microcontroller, we have decided on using bluetooth. Even though wifi has very compelling benefits such as longer range, more connections for monitoring, higher speeds, and better security, the benefits do not outweigh the cons that are very crucial to how well our solar picnic table performs. The bluetooth's lower power consumption and low cost were the most important part when deciding what method we would choose. When it comes to solar power, power efficiency is one of the main aspects we look for when looking for components to build this table. Low cost also is a big benefit for us so we can allocate more funds towards more efficient solar panels which is the biggest component in this project.

## 3.13 Developing the Mobile App

In this section, we will be highlighting the different methods to develop our mobile app. We will discuss the different platforms that are available to us and compare them with each other. With the comparison, we will be able to decide which platform suits our needs the best.

### 3.13.1 Platforms available

When it comes to developing the mobile app, we had a few different platforms to choose from to develop the application. For IOS devices, we could utilize Apple's programming platform called Xcode which is developed for only IOS devices. For Android devices, we had the option to use Android studio. Since we wanted everyone to have access to the application, we wanted to develop an app to support all types of devices. We had two different options such as React Native and Flutter which allows developers to create applications that can be done on both Android and IOS devices. If we went with Xcode and Android studio, we would have to create the same application twice which would take away from development time of the actual table. In this scenario, allocating more time developing the table would make more sense then taking the time to learn two different programming languages to create the same app.

#### 3.13.1.1 React Native/Flutter

Two platforms we can utilize for cross-platform development are React Native and Flutter. React Native is created by Meta and relies on JavaScript mainly to create

React components such as the layout and UI of the application. These React components are then sent to a C++ engine that renders those components onto the screen. Flutter is created by Google and utilizes their Dart language. Unlike React Native, Flutter has everything written in Dart such as the rendering engine and code execution engine.

Flutter provides the user with many different options when it comes to UI creation. It has a large community that helps improve and offer support for Flutter and comes with great documentation for all components in Flutter. When compared to React Native, Flutter applications are quicker. A few downsides to Flutter are that they do not offer as many tools to help developers program.

React Native offers a lot of options for reusable code since React Native and React code, JavaScript platform for website development, are almost interchangeable similar to how C/C++ works. With this in mind, we could technically transfer our mobile app onto a website with the addition of finding a hosting company. React Native also has a very active community but not as big as Flutter. React Native also helps quicken some of the repetitive aspects of programming leading to quicker.

<b>Flutter</b>	<b>React Native</b>
Many options for UI creation	Large community support
Really good documentation	Reusable code for website
UI between different platforms are more similar	Larger app files
Larger community support	Built-in assistance
Uses Dart	Uses JavaScript
Quicker performance	Support for 3D components

*Table 14 - Difference between Flutter and React Native*

### **3.13.1.2 Decision (React Native)**

After researching both platforms, they both seem to provide what we needed in performance. Both of them seem like they have a very active community and both offer support when building mobile applications in both Android and IOS devices. As for performance, Flutter's slight performance gain over React does not seem too desirable considering this mobile application is not complex enough to notice the performance difference. When finalizing our choices, it all came down to which platform was going to be the easiest to implement and which one would optimize our time the best. Since the people working on programming the mobile application have

experience with JavaScript from classes offered at UCF, we have decided that moving forward with React Native is the best option. Experience with JavaScript does not mean instant transition to React Native, but it will help in the learning experience as compared to Flutter, learning Dart from scratch might pose a challenge during this whole development cycle.

## 3.14 Interacting with the Microcontroller

When it comes to interacting with the microcontroller (esp32), we have decided that utilizing bluetooth would be the best option since it is energy efficient and low cost since bluetooth low energy is built into the microcontroller. To interact with the ESP32, we can utilize the Arduino IDE which uses C/C++. We have to use the Arduino IDE to set up bluetooth connections and develop an application using React Native that can utilize bluetooth connection and connect to the microcontroller. From that, we can program the application to run certain functions such as turning on LEDs or relay information that the microcontroller is reading into a UI element on the mobile application.

## 3.15 Microcontroller options

A microcontroller unit (MCU) is an integrated circuit semiconductor that consists of one or more processor core(s), memory, and programmable I/O peripherals. MCUs have a wide range of applications that deal with embedded systems such as robotics, house appliances, and vehicles. For our project, we will equip the picnic table with an MCU and use it to implement various functionalities. The key features that we are looking at are large flash memory, large RAM, can support at least 24 GPIO pins, 18 if the microcontroller has built in Wifi and Bluetooth capabilities, and enough current output from the pins to support the modules, somewhat fast clock speed, low power options, and low price. After extensive research, we were able to narrow down our choices to five microcontrollers: the ATmega2560, ATmega4809, ATSAM21G18A-AU, MSP430FR6989, and the ESP32-WROOM-32E-N16. In this section, we will delve into the key features of these microcontrollers and see what they have to offer.

### 3.15.1 ATmega2560

The ATmega2560 is an 8-bit AVR microcontroller developed by Microchip Technology Inc. This microcontroller comes with 256 KB of flash memory, 8 KB of SRAM, 86 GPIO pins with a max current output of 40 milliamps per pin, and a CPU speed of 16 MHz. There are six sleep modes that the microcontroller can use to deactivate any unused modules to minimize power usage. It also has two modes for power consumption: Active mode and power-down mode. In active mode, given these conditions of 1 MHz and 1.8 volts, the microcontroller draws .5 milliamps. Under the same conditions, under power-down mode, the microcontroller draws 0.1 microamps. On the microchip website, the ATmega2560 comes out to be \$20.18 per unit. On paper, the ATmega2560 seems to meet all the requirements needed for our project. It comes

with a large amount of flash memory which should be enough to store our code. The amount of SRAM should be sufficient and there are more than enough GPIO pins to support all of the external modules.

### **3.15.2 Atmega4809**

The ATmega4809, like the previous microcontroller, is an 8-bit AVR microcontroller developed by Microchip Technology Inc. It has a flash memory of 48 KB, 6 KB SRAM, 41 GPIO pins with a maximum current output of 20 milliamps per pin, and a CPU speed of 20 MHz. Like the mega2560, it comes with sleep modes, however, instead of six it is three. However it is still capable of active mode and power-down mode. Under an operating voltage of 1.8 volts, 25 degrees celsius temperature, and clock speed of 1 MHz the microcontroller is able to draw 0.2 milliamps during active mode. The price per unit comes out to be \$1.85. Overall, this microcontroller was able to satisfy the needs for the project. One feature that stands out is the price, despite having less memory and GPIO pins as the first microcontroller, the price was about eleven times cheaper.

### **3.15.3 ATSAM21G18A-AU**

The next microcontroller is a 32-bit ARM microcontroller that was also developed by Microchip Technology Inc. It comes with 256 KB of flash memory, 32 KB of SRAM, 38 GPIO pins with a maximum current output of 7 milliamps per pin, and a clock speed of up to 48 MHz. It is capable of low power mode making it suitable for low power consumption. Running in active mode, at an operating voltage of 1.8V and clock speed of 1 MHz, the microcontroller draws a current of .06 milliamps. These microcontrollers are around \$4.41 per unit. This microcontroller is tied for the most flash memory, has the most SRAM, and is capable of having the highest clock speed of all of the microcontrollers. However, the drawback of this microcontroller is that it draws a large amount of current.

### **3.15.4 MSP430FR6989**

The MSP430FR6989 is a microcontroller that was developed by Texas Instruments. Along with this, it is a microcontroller that is used in many classes at UCF so we have experience working with this. It has 128 KB of flash memory, 2KB of RAM, 83 GPIO pins with a maximum current output of 6 milliamps per pin, and a clock speed of 16 MHz. It is also equipped with ultra low power modes. In active mode, the microcontroller draws approximately 0.103 milliamps at 1.8 volts and 1 MHz clock speed. The price per unit is \$4.99. This microcontroller was considered due to the familiarity with the device and its features. It has a large amount of GPIO pins to ensure that we can install all of the external modules. The amount of memory is fairly large and should suffice for our code. The clock speed is similar to the ATmega4809 and ATmega2560 but surpassed by the SAMD21. The pricing is fairly cheap when compared to what this microcontroller offers.

### **3.15.5 ESP32-WROOM-32E-N16**

This microcontroller was developed and manufactured by Espressif Systems. It is part of the ESP32 family and comes with a variety of features. The microcontroller is fitted with a 32-bit dual microprocessor with clock speed of up to 240 MHz, 4 MB of flash memory, 520 KB SRAM, 32 GPIO pins with a maximum current output of 12 milliamps per pin, 2.4 GHz Wifi and Bluetooth. It also has Bluetooth low energy (LE) which helps with power consumption. It also contains five different low power modes. The ESP32 has two different radio frequency modes: Wifi and Bluetooth. When both radios are active and with a supply voltage of 1.8 volts and clock speed of 1 MHz, the current drawn is .014 milliamps. However, for this project, using both radios is unnecessary so this number can be reduced further using the various low power modes. The price comes out at \$1.80 per unit making it fairly cheap. Taking into account all of the details, this microcontroller is incredibly cost efficient. It has the highest flash memory and SRAM. Although it doesn't have the most GPIO pins, it still has enough to accommodate the external modules. It also has built in Wifi and Bluetooth capabilities which can save us money on buying a separate bluetooth module.

## **3.16 Microcontroller Comparisons**

For our project, the microcontroller is one of the most important features as it will allow the implementation of many of the functionalities such as motion sensing, light control, temperature and battery monitoring, bluetooth, and mobile app connection. To ensure that all of these features are implemented correctly, we must choose the most fitting microcontroller based on a set of criteria. In the next sections, we will go into detail about the criteria and compare each microcontroller to determine which one would be the most optimal for our project.

### **3.16.1 Pins**

The microcontrollers are fitted with a certain amount of pins. These pins serve a purpose and send electrical signals from the microcontroller to various devices. These signals are important to ensure that they work properly. There are different types of pins that microcontrollers have. First they have power pins. These pins are used to power devices that are connected to the microcontroller such as an LED light. These pins usually supply 3.3V or 5V. The next type of pin is the ground pin. These pins output no voltage, however it is important for a circuit to work properly. Another type of pin is the general purpose input/output (GPIO) pins. There are many of these on the microcontroller. They serve no specific purpose meaning they are configurable. Their purpose is decided upon by the user. In terms of the solar picnic table, it is important that we choose a microcontroller that contains all of these pins. More specifically, it will need at least 24 GPIO pins to accommodate all of the external modules and have a sufficient current output per pin.

Microcontroller	# of GPIO Pins	Maximum current output per pin (mA)
ATmega2560	86	40
ATmega4809	41	20
ATSAMD21G18A-AU	53	7
MSP430FR6989	83	6
ESP32-WROOM-32E-N16	32	12

*Table 15 - Comparison of # of GPIO Pins/Current Output per Pin on Microcontrollers*

Looking at the table above, if we go purely by amount of pins and maximum current, then the ATmega2560 has the most. Having 86 pins and 40 mA current output is more than enough to enable our external modules to work properly. A close second in terms of pins is the MSP430FR6989, however, it is lacking in maximum current. The next highest current is the ATmega4809 with 20 mA. Despite being second, the ATmega2560 has double the amount of output. The other microcontrollers don't stand out in pins or output so they can be discarded. With this, we can determine that the ATmega2560 is the best choice when it comes to pins and current output.

### 3.16.2 Memory

When it comes to memory, microcontrollers have access to three kinds. The first type of memory is the flash memory, this is a non-volatile memory meaning that the stored data and information persists even after the device is powered off. The more flash memory there is, the more code that can be stored. It also gives the user more options when it comes to coding. This is a very important element when picking out our microcontroller since there will be a large amount of code implemented for all of the functionalities of the picnic table. The next type of memory is RAM or random access memory. RAM is a volatile type of memory so the data and information stored is only temporary. This is useful for any sort of calculation. Similarly, the more RAM the MCU has the more options we have when it comes to coding. The last type of memory is electrically erasable programmable read-only memory or EEPROM. EEPROM works similarly to flash memory however, it is primarily used for small data storage as opposed to large data. This type of memory isn't quite important for this project so it won't have a large impact on our decision for our choice. Overall, we are looking for a microcontroller with a large amount of flash memory and RAM.

Microcontroller	Flash Memory Size	RAM
ATmega2560	256 KB	8 KB
ATmega4809	48 KB	6 KB
ATSAMD21G18A-AU	256 KB	32 KB
MSP430FR6989	128 KB	2 KB
ESP32-WROOM-32E-N16	4 MB	520 KB

*Table 16 - Comparison of Flash Memory/RAM Size*

For memory, there is a clear winner. The ESP32-WROOM-32E-N16 has the most flash memory and RAM. Although it has the most amount, it could add onto the price of this microcontroller. However, despite it being a huge amount, having this much memory allows us to have more options when it comes to creating our code. We won't be restricted by our memory space and can create more complex code. This alone could outweigh the cons.

### 3.16.3 Power Consumption

Low power consumption is an important feature that we are looking for. When it comes to calculating the power consumption, there are many factors that can affect it such as power modes, operating voltage, and clock speed. For this comparison, we will be comparing the microcontrollers while they are in active mode and at a clock speed of 1 MHz. To calculate the power consumption we will multiply the operating voltage and the current per MHz.

Microcontroller	Operating Voltage (V)	Current per MHz (mA)	Power Consumption (mW)
ATmega2560	1.8	0.5	0.9
ATmega4809	1.8	0.2	0.36
ATSAMD21G18A-AU	1.8	0.06	0.108
MSP430FR6989	1.8	0.103	0.185
ESP32-WROOM-32E-N16	1.8	0.014	0.0252

*Table 17 - Comparison of Power Consumption Between Microcontrollers*

Since the solar panels won't be able to draw power for the picnic table during night time, we will be using a lithium battery to power the table throughout the night. This means that we want to use a low amount of power so that the battery can last the picnic table until the sun comes out. In the testing area, sunrise is usually around 7 AM where the sunset is around 7 PM. This means that the microcontroller has to be power efficient so it can last the whole 12 hours. Going off of only power consumption, the best choice would be the ESP32-WROOM-32E-N16 as it has the lowest power supply need of 0.0252 mW.

### 3.16.4 Clock Speed

The clock speed of a microcontroller deals with the speed at which instructions are fetched, interpreted, and then executed. Since some functionalities of the table require a constant measurement of the surrounding area such as temperature monitoring, we need the microcontroller to be able to keep up with this demand. However, the calculations for these numbers won't be too complex so a high clock speed isn't necessary. This means that the clock speed won't be a huge focus and we will allocate our budget more towards the other requirements.

Microcontroller	Clock speed (MHz)
ATmega2560	16
ATmega4809	20
ATSAMD21G18A-AU	48
MSP430FR6989	16
ESP32-WROOM-32E-N16	240

*Table 18 - Comparison of Clock Speed Between Microcontrollers*

Judging from the table above, we can see a clear winner. The ESP32-WROOM-32E-N16 has the best clock speed by a large margin at 240 MHz. This will allow our microcontroller to perform tasks fairly quickly.

### 3.16.5 Price

One of our goals for this project is to create a cost effective solar picnic table. This means that we are looking for a low cost microcontroller. However, we aren't just looking at the price, we must also take into account the accompanying features that come with the microcontroller such as large memory or pin count. So, we will take into account the other features and determine which microcontroller is the most cost effective for our project.

Microcontroller	Price per Unit (USD)
ATmega2560	20.18
ATmega4809	1.85
ATSAMD21G18A-AU	4.41
MSP430FR6989	4.99
ESP32-WROOM-32E-N16E-N16	3.6

*Table 19 - Comparison of Prices Between Microcontrollers*

Looking at the different prices, we can see that the ATmega2560 is clearly the worst option coming at 20.18 dollars per unit. This is almost five times more expensive than the second worst option. The best microcontroller when it comes to price would be the ATmega4809 at 1.80 dollars with the ESP32 coming right behind it but being almost twice more expensive. From these numbers, we have decided that the ATmega4809 is the best choice when it comes to price.

### 3.16.6 Decision (ESP32-WROOM-32E-N16E-N16)

From the comparison results, we are able to rule out a couple of microcontrollers. First, the ATmega2560, the biggest turn off was the huge price. Despite fulfilling all of the other requirements that we need for a microcontroller, the price was way too steep for us to consider it. Along with its price, it offered one of the lowest clock speeds and it consumed the most power. It excelled in the number of GPIO pins and current output per pin but this amount was overkill. The project called for at least 16 pins and a current output of at most 6 milliamps. Having this excess would just add more onto the cost of the microcontroller which is something we want to minimize as much as possible. Although it had a fairly large flash memory and RAM, it was still surpassed by the EP32-WROOM-32. Overall, this microcontroller is not what we are looking for.

With the ATmega2560 out of the pool of options, it leaves us with four more options. These four microcontrollers, like the ATmega2560, were able to satisfy all of the requirements but one was more qualified than the others. However, like the ATmega2560, three of them were either overkill in one category or they were simply outclassed by other microcontrollers. In the comparisons, the ESP32-WROOM-32E-N16 microcontroller was chosen as the best choice in almost all of the categories. The only sections that it was outranked in were the pins and price. Although it had the lowest amount of pins, it was still plenty to fit all of the external modules. On top of this

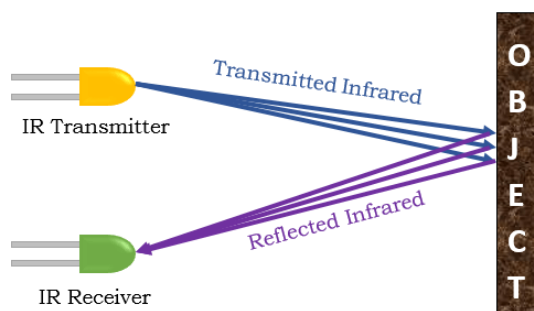
it had a sufficient amount of current output to supply these modules. It had the fastest clock speed, the most memory, the lowest power consumption, and the second lowest price per unit. It also comes with built-in Wifi and Bluetooth so it decreases the required amount of GPIO pins and saves us money as we don't have to buy a separate bluetooth/wifi module. The ESP32-WROOM-32E-N16E-N16 is also compatible with Arduino meaning there will be some familiarity working with this microcontroller. The ESP32-WROOM-32E-N16E-N16 outclasses all of the other microcontrollers and came with even more features. With that said, we have decided to use the ESP32-WROOM-32E-N16E-N16 for our project.

## 3.17 Motion Sensor Technology

One of the functionalities of the solar picnic table is the ability to detect if a person is nearby or not. Depending on whether there is a person present, the table will be able to turn the LED lights off or on. To accomplish this, there are different types of motion sensing technology that must be discussed: infrared (IR), passive infrared (PIR), microwave, and ultrasonic. These types of sensors use different methods when it comes to motion sensing. Each is better in their own way as they have specific strengths. We will go into depth with the mechanics of these methods and choose a sensor that specializes in each type of motion sensing. From there, we will weigh the strengths and weaknesses of each sensor and determine which one will be the best for our project.

### 3.17.1 Infrared

The IR sensor is a device that measures and detects the infrared radiation within its environment. The device is equipped with two different LEDs: the IR transmitter and the IR receiver. The function of the IR transmitter is to continuously send out IR lights. The lights that are sent out are constantly checked by the IR receiver. If the IR light hits an object, it will bounce back and be taken in by the receiver.



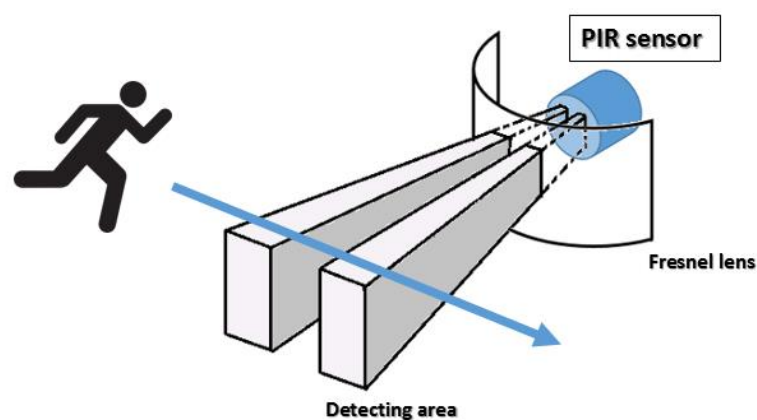
*Figure 10 - Depicts the Mechanism Behind IR Sensing*

*Adapted from [6]*

The infrared sensor has many strengths. It has more secure communication due to its line of sight methods. Another strength of the IR sensor is that it is unaffected by corrosion or oxidation. Since the picnic table will be seated outside, it is important to note that each component must be able to withstand the environment. On top of this, they are generally low power and cheap which is a key feature we look for when picking parts. Despite their strengths, they have some weak points. The IR sensors aren't able to work through hard objects so if something were to block it, it won't be able to respond. Since it uses line of sight communication, it has a small range.

### 3.17.2 Passive Infrared

Everything emits some sort of radiation. The PIR sensor is able to take in these levels of radiation and detect motion within the surrounding environment. This ability is thanks to the sensor's makeup. A PIR sensor is made up of pyroelectric elements that are able to capture and measure the radiation level of different objects. PIR sensors are usually made up of two to four pyroelectric materials. Each of these materials view different areas at the same time. When an object goes into the view of one area, the pyroelectric material will pick up the temperature change. This temperature change is what signals the sensor that there is motion.



*Figure 11 - Depicts the Mechanisms Behind PIR Sensing*

*Adapted from [7]*

Unlike the IR sensor, the PIR sensor specializes in covering a wide area. They are perfect for use when it comes to small or compact areas. This is a good feature as we don't want the LED lights to activate if someone just happens to walk past. This in the long run could wear out the lights and consume unnecessary power. Another advantage of the PIR sensor is that it is cheap and has low power consumption like the IR sensor. Since it uses temperature to detect motion, if the environment is

extremely hot, it could have trouble picking up any sort of movement. It also has lower coverage and less sensitivity when compared to the other sensors.

### 3.17.3 Microwave

The method of motion sensing when it comes to microwaves is very similar to that of a PIR sensor. However, the microwave sensor has an active approach. This means that it is actively sending out microwave pulses and receiving them back. If an object were to walk in the affected area, then the transmitted microwaves would hit the object and bounce back to the sensor. The sensor would then detect the change in the microwave and sense the object.



*Figure 12 - Depicts the mechanisms behind microwave sensing*

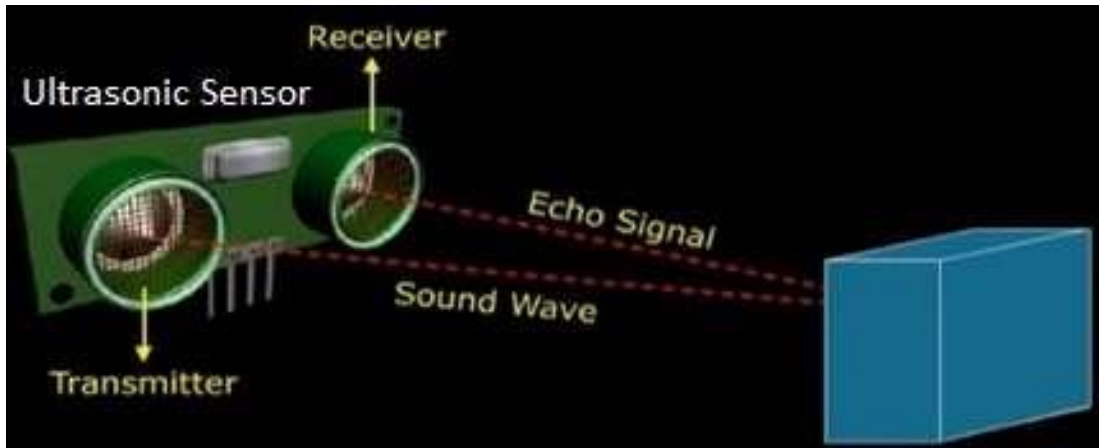
*Adapted from [8]*

Similarly to the PIR sensor, the microwave sensors specialize in covering an area. However, one strength that it has over the PIR sensor is that this area is much larger. These sensors are much more sensitive so they can pick up movement easily. Due to their larger range and sensing capabilities, they tend to be more expensive than PIR sensors. Due to their high range and sensitivity, they can often pick up unnecessary movement such as animals, or random objects moving in the wind.

### 3.17.4 Ultrasonic

To understand how ultrasonic sensors work, we must first understand what an ultrasonic wave is. Ultrasonic waves are a type of sound wave. Sound waves are vibrations that are transmitted through a medium like metal or air. Ultrasonic waves are any inaudible sound waves of high frequency. This frequency is usually above 20 KHz.

An ultrasonic sensor is a device that utilizes these sound waves to measure the distance of objects. It uses a transducer to send and receive ultrasonic waves to gain information about the object. This receiving sound wave is called the echo. The process of motion detecting works like this, the transducer sends out an ultrasonic wave. These waves will then hit an object and will be returned as an echo. The transducer will pick up this echo and calculate the distance of the object by measuring the time it takes between sending and receiving the sound wave.



*Figure 13 - Depicts the mechanics behind ultrasonic sensing*

*Adapted from [9]*

The ultrasonic has many strengths, it is able to sense any sort of material. It is also unaffected by environmental factors such as rain making it versatile in any sort of environmental conditions. Compared to the other sensors it has a higher sensing distance. Some drawbacks of an ultrasonic sensor is that it is sensitive to any sort of temperature variation and it also has trouble reading echos from soft, thin, or curved objects.

Technology	Pros	Cons
IR	Secure communication, resistant to environment, low power and cheap	Short distance, unable to penetrate hard objects, low area of effect
PIR	Excels in covering small or compact areas, low power consumption, cheap, reliable	Prone to hot temperature, low sensitivity
Microwave	Covers large area, high sensitivity	Can pick up unwanted movement, consumes more power, expensive
Ultrasonic	Senses any material, unaffected by environment like rain, high sensing distance	Sensitive to temperature changes, trouble reading echos from soft, thin, or curved objects

*Table 20 - Showcases the pros and cons of each motion sensing technology*

### 3.17.5 Decision (PIR)

When deciding the motion sensing technology that we need, we had to consider what we needed for the table. First, it must be able to cover 360 degrees so that it can sense motion around the whole table. Second, it had to be functional during the night time as that is when we would utilize the motion sensing. Finally, it had to be cheap and

have low power consumption. Since the motion sensor will be used during the night, it has to be power efficient so it doesn't drain the battery. After considering all of this, we decided that the passive infrared technology would be the best approach. This is because it is able to cover a small area, and it was fairly cheap and power efficient. The microwave sensors also fulfilled our requirements however, the area it would cover would be too large. We don't want to trigger the motion sensor just from someone walking by. This could lead to power waste and that is a big issue. Because of this, the small area coverage is perfect for our table. On top of this, since the motion sensor will operate at night, its vulnerability to hot temperature wouldn't be an issue since the environment will cool down during the night. For the actual PIR sensor, we picked the **AM312**, it has a detection range of around 3-5 meters and a detection angle of 100 degrees. This means we will have to use four of these sensors for a 360 degree coverage. On Amazon, it comes out at 2.17 dollars per unit. The sensor itself has low power consumption and it is fairly cheap so using more won't be an issue.

## **3.18 Temperature Monitoring**

One functionality that will come with our solar picnic table is temperature monitoring. This is a quality of life feature that will give the user information of the temperatures outside. Furthermore, this feature will be implemented through the microcontroller and the data will be displayed through our mobile application. When looking for a suitable temperature sensor, there are a few factors that we must take into account. The most important factor is the temperature reading accuracy. We want the user to be given the most accurate reading of the current temperature so they can be informed of their environment. Another factor is the temperature range. We want the sensor to be able to read a wide variety of ranges so it can be suitable for any environment. Finally, we want the sensor to be cheap and have low power consumption. This is important as we want to minimize costs when building our solar picnic table and we want to use as little power as possible to make it cost efficient. From these factors, we were able to pick possible parts: the DHT22, TMP36, and LM35.

### **3.18.1 DHT22**

This 4 pin temperature sensor comes with many features, it has an operation voltage ranging from 3 to 5 volts meaning that it is capable of being powered by the microcontroller without any sort of modifications. It is around 10 dollars on Amazon and has a low power consumption. The current consumption is around 2.5 milliamps while it is functioning. It can also read temperatures from -40 to 80 degrees celsius with a +- 0.5 degree accuracy. It also gives humidity readings from 0-100 percent with an accuracy of 2-5 percent. These features of the DHT22 makes it very enticing.

### 3.18.2 TMP36

The TMP36 comes with 3 pins and has temperature monitoring capability. It has an operating voltage of around 2.7 to 5.5 volts. Like the DHT22, this sensor can be powered by the microcontroller without the help of any modifications. It has a temperature reading range of -40 to 150 degrees celsius with an accuracy of +-2 degrees. On top of this, it has a low power consumption as it runs below 50 microamps. It has a price of 3.89 dollars on Amazon

### 3.18.3 LM35

Like the TMP36, the LM35 is a temperature sensor with three pins. It operates from 4 to 30 volts. This means that we would have to make some modifications if we wanted to run this sensor. It has a temperature reading range of -55 to 150 celsius with a +-0.5 degree accuracy. It is fairly low powered as it has a current consumption of less than 60 milliamps. This temperature sensor is at 13 dollars for a 5 pack on Amazon.

Temperature Sensor	Operating Voltage (V)	Range (°C)	Accuracy (°C)	Price per unit (usd)
DHT22	3 to 5	-40 to 80	0.5	10
TMP36	2.7 to 5.5	-40 to 150	2	3.89
LM35	4 to 30	-55 to 150	0.5	2.6

*Table 21 - Comparison of the different temperature sensors*

### 3.18.4 Decision (DHT22)

For the temperature sensor, we have decided that the one most suitable for our project is the DHT22. For the reading range, although the TMP36 and the LM35 had the best ranges, they were too excessive. Realistically, we would be in an environment of around 20 to 38 degrees Celsius or 70 to 100 degree Fahrenheit. The main reason we ruled out the TMP36 was because of the reading accuracy. It was by far the lowest out of the three and since we want the user to have the most accurate data possible, the TMP36 wouldn't fit our needs. The LM35 was ruled out due to its operating voltage. Since our microcontroller has a max output of 3.3 volts, it would over complicate the build to try and get the two to be compatible. This leaves the DHT22, it has an acceptable operating voltage range making it work with our ESP32 microcontroller. The range is also suitable for our project and the accuracy is tied for the best among the three. The only downside is the price, however, we are also paying for a humidity monitor which can prove to be useful as it provides more information about the area.

## **3.19 Battery Monitoring**

Through the mobile app, the user will be able to read statistics about various elements of the solar panel and lithium battery. For example, the user will be able to see the voltage output being sent through by the solar panel. For the lithium battery, the user will be able to monitor the charge level of the battery to see how much power is left throughout the night. This function is important as it will allow the user to determine how much they could charge their electronic devices or how long they can use the solar picnic table for. To implement this, there are a couple of factors that we must look into. First, we would have to connect both the solar panel and battery to the microcontroller's analog pin. These analog pins can be used to read the voltage of the two power sources. However, the max amount of voltage the ESP32 can withstand is 3.3 volts. Anything over this amount could cause serious damage to the microcontroller. To combat this, we would need to use a voltage divider circuit to cut down the amount of voltage being inputted. From here, we can use the ESP32's analog-to-digital converter (ADC) to convert the analog voltage values to digital. Once digital, they can be processed by the microcontroller and sent to our mobile app. The component that we will be looking at is the voltage divider. There are two methods we can approach for this.

### **3.19.1 Voltage sensor module**

First, we can use a voltage sensor module. These are a low cost external module that can be used to connect the power sources to the microcontroller. It has a built in voltage divider and is capable of reducing the output voltage to be 5 times smaller than the original value. It has a limited input voltage of 0 to 25. It would also take up less space on the breadboard. Overall, it would make dividing the voltage much easier as the voltage division is calculated for us. Therefore, we wouldn't have to worry about any miscalculations and potentially damaging the microcontroller.

### **3.19.2 Manual Voltage Divider Circuit**

The other method would be to create our own voltage divider. This method requires us to take the output voltage of the solar panel and battery then plugging them into Ohm's law. For the voltage drop, we will choose a value lower than the maximum output of the ESP32 which is 3.3 volts. The resistor values should be equal to the ratio at which the voltage drops across them. By doing this, we can figure out what resistor values we need to drop the voltage to our desired amount.

### **3.19.3 Decision (Manual Voltage Divider Circuit)**

After some consideration, our group decided for the manual voltage divider circuit. If we were to use the voltage divider module, it would add onto the total cost which is something that we want to minimize. For the resistors, UCF provides them for free within their labs so there would be no cost. The calculations themselves shouldn't be

an issue as they're pretty simple. On top of this, connecting the power sources to the microcontroller through the voltage divider circuit won't be difficult as we have experience from previous classes.

## 3.20 PCB

Since the MCU, Temperature sensor, Motion sensor, and the LED all have various operating conditions, we have decided to design a voltage regulator that will be in a Printed Circuit Board (PCB). The goal of this PCB design is to provide the right voltage to each of the components connected to it.

Looking at the figure (21) we can see that the charger controller is going to supply the electrical current to our devices and to the inverter. Our inverter operates at 12V so that voltage is going to be the output voltage of the charger controller. To accommodate the other devices that are connected we need to look at their operating voltages.

Device	Nominal Operating Voltage	Maximum Operating Current
ESP32 (MCU)	3V	250mA
ALITOVE WS2812B (LED)	5V	2.4A
AM312 (Motion Sensor)	3V	20 $\mu$ A
DHT22 (Temperature sensor)	5V	1.5mA

*Table 22 - Comparison of Operating Voltage and Current*

For this table we can see the different operating voltages and maximum operating current of our four devices. The reason we are using the maximum operating current is because we would like our voltage regulator circuits to be able to handle cases of maximum current usage.

Two of our devices require 5 volts DC and two require 3 volts DC. This means that we will make two separate voltage regulator circuits. Each will output its respective voltage

and the approximate sum of the maximum operating current of both devices it supplies to.

For our 5V voltage regulator we will design it so that it converts 12VDC to 5VDC. The current output would likely be 2.4A + 1.4mA. For assurance we will design it to be able to output up to 3 Amps. This regulator will supply both the led and the temperature sensor.

The other voltage regulator will convert the same 12VDC input into 3VDC output. The current output will be approximately up to 500mA. This regulator circuit will supply the microcontroller and the motion sensor.

Both of these circuits will be designed in the same PCB. It will have one input port which will be supplied from the charger controller 12VDC, and will have two different output ports 5VDC and 3VDC.

## 3.21 Equipment Parameter Calculations

In order for us to figure out the required parameters we are going to use for the PV panel, battery storage system, inverter, and charger controller we first need to calculate the total energy consumption by all the loads. Then we will work our way backwards till we reach the PV panel which is the source.

The first step to calculate the total energy consumption which is the **Watts\*number of hours**. As discussed earlier, this project is a model for our customers. So for showcasing and testing purposes we are going to use **1** for the number of hours for all loads. This means that the calculations are based on if all loads are on for an hour.

### 3.21.1 Total energy consumption

Here we will go over the calculations for the total energy consumption of every electrical component in the solar picnic table. We will be able to gauge the amount of power that we will need.

#### 3.21.1.1 LED

One LED strip is about one meter and consumes about **12W** power. If multiple strips are needed then connecting them in series would draw the same current but will divide the voltage between them, overall this will not change the total power consumption, but will divide the total power between them.

$12W * 1 \text{ hour} = 12Wh$

## 3.21.1.2 Power Outlets

The energy consumption from power outlets is totally dependent on the devices connected to it. The following calculations are estimations. Typically a smartphone would consume about **10W** charging in a power outlet. Either the adapters step down the 120VAC to its required voltage (about 5VDC) or from the 5DC ucb to its required voltage. Laptops on the other hand consume way more. A typical Dell laptop consumes about 40-90W. We are going to assume that it consumes **60W**.

$(10W * 1 \text{ hour}) + (60W * 1 \text{ hour}) = 70Wh$

## 3.21.1.3 Temperature Sensor

The temperature sensor that we will be using is the DHT22. This sensor has a low power consumption so it won't impact the total power consumption too much. With a current consumption of 0.0015 amps and a voltage of 3.3 v, the power consumption will come out as **0.00495W**

## 3.21.1.4 Motion Sensor

Generally, the power consumption on such devices is very minimal. At an operation voltage of 3.3V and a current consumption of 15 microamps. Based on the chosen sensor, the actual power consumption is 0.0000495W per sensor. Since we need 4 of these sensors for a complete 360 degree coverage, the total power consumption is **.000198W**.

## 3.21.1.5 Microcontroller

The power consumption of the chosen microcontroller "ESP320-WROOM-32E" is specified to be 0.0252mW, however, this is under the conditions of 1.8 volts and 1 MHz clock speed. For this power calculation, we will use an operating voltage of 3.3 volts, a clock speed of 80 MHz, and active mode. Under these conditions, the current consumption would be around 130 mA. Multiplying the current and voltage would give a power consumption of **0.429W**.

For the sake of having simpler calculations, we are going to round the consumption of the temperature and motion sensor to **1W**. The consumption of the microcontroller will be rounded to **1W** too.

Total Energy consumption =  $12 + 70 + 1 + 1 + 1 = 85Wh$ .

## 3.21.1.6 Inverter Rating

Due to the power losses within the inverter, when choosing the rating of one, it should have a rating of 120-130% that of the loads connected to it. In our case the inverter

will be connected to the power outlet which was estimated to consume about 70W. So our inverter should be able to handle at least 91 watts  $70 * 1.3 = 91W$ . The chosen **BESTEK MRI3011BU** inverter is rated at **300W** which is more than enough. It has an efficiency of  $\geq 83\%$ .

### 3.21.1.7 Battery Rating

In order to calculate the required minimum battery rating in Watts and AmpHours in our system, we need to have the following information: energy consumed by the loads, inverter losses, voltage of the battery, efficiency of the battery, and Depth of Discharge (DOD).

First of all to calculate the energy rating we will use this formula:

Energy of the battery in Wh denoted by "Y"

Energy consumed by loads in Wh denoted by "L"

$Y = L + ((L - \text{loads not connected to inverter}) * \text{loss from the inverter})$

$Y = 85 + ((85-14) * (1-0.83)) = 97.07$  approximated to **97Wh**

So the **rated power of the battery must be at least 97W**

Now to calculate the required AmpHours (Ah) we will use the formula provided below:

$Ah = (Y * \text{Time period in days}) / (\text{Voltage of the battery} * \text{DOD})$

$Ah = (97 * 0.25) / (12 * 0.5) = 4Ah$

So for 6 hours the battery needs to have a minimum current supply of **4 Amps**.

Since the **EXP1280** battery we have decided on is 100W 8AH this means that it will be more than enough.

### 3.21.1.8 Charger controller Rating

For safety measures, the charge controller should be able to handle at least the power rating of the solar panel divided by the battery voltage. In our case, the solar panel we are using has a power rating of 100W, and our battery's voltage is 12V. So, the charge controller should be able to handle  $100W/12V = 8.33A$ . The charge controller we chose can handle a maximum of 30A, so it should be more than safe to use.

## 3.22 System Performance Simulation

Using the simulation tool provided on the website by the European Commission. We have been able to simulate some of the performance metrics of our stand-alone off-grid solar system if it was installed on the University of Central Florida campus.

### 3.22.1 Input

Latitude/Longitude: 28.602,-81.191

PV installed: 100 Wp

Battery capacity: 100 Wh

Cutoff limit: 50 %  
 Consumption per day: 80 Wh  
 Slope angle: 35 °

### 3.22.2 Output

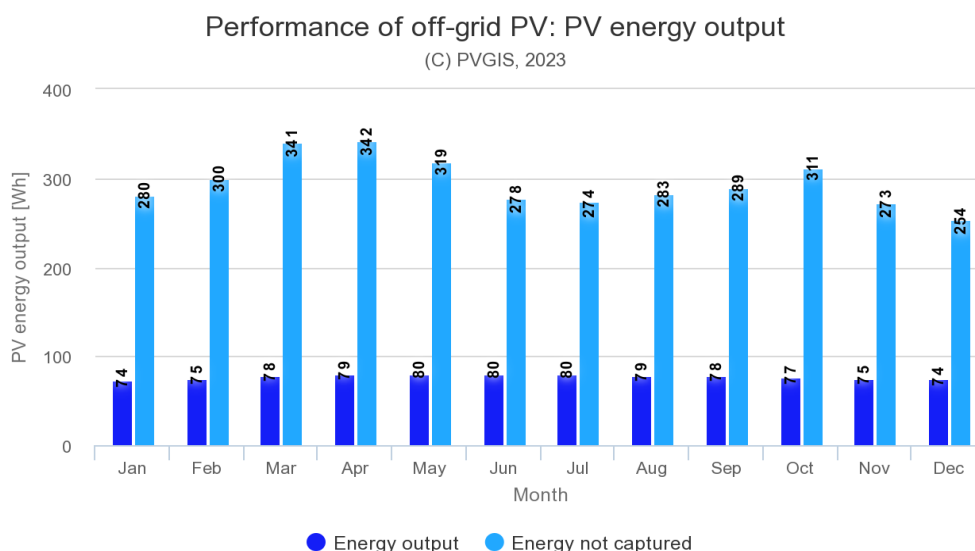


Figure 14 - PV Energy Output

The figure 14 above shows the simulated result of the energy output of our PV system. First of all we see that all variables (Energy captured and Energy not captured) are somewhat consistent. The energy not captured peaks around the summer time but the increase is gradual throughout the months. The value of the Energy output is capped at 80Wh because that is our inputted consumption Wh per day. We see that there is much more energy that could be captured by our system if we had higher load demands. This shows that our PV panel of 100W is capable of supplying energy to any extra demands.

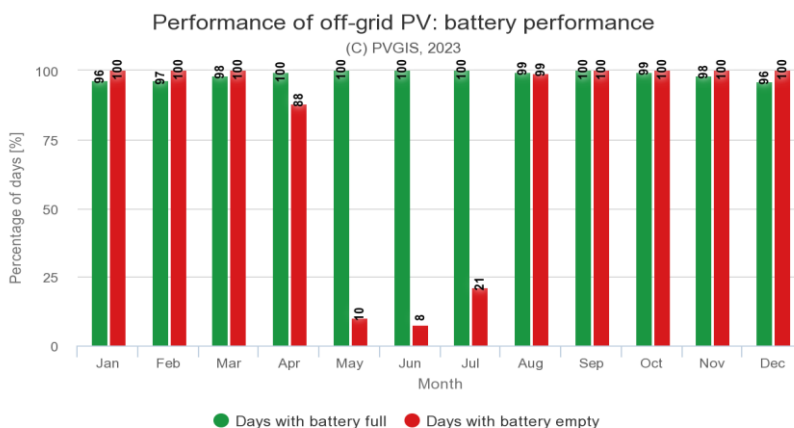
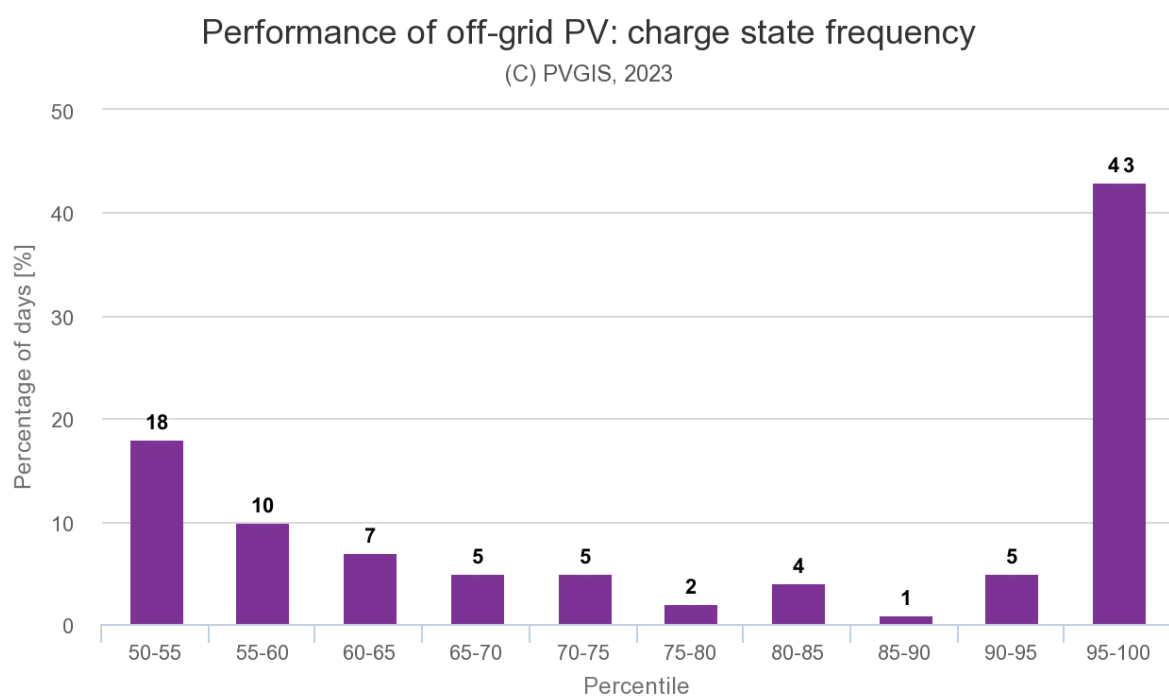


Figure 15 - Battery performance

Figure 15 represents the percentage of days that the battery will reach its maximum capacity and the percentage of days where it will be emptied (50% DOD) for each month. Throughout the year we can see that the battery will be full almost all days in each month. We can also see that the battery will be emptied fully everyday each month except for May, June, and July. In these three months when the sun is usually the strongest we can see that the batteries will be empty less often. This is logical for multiple reasons. In the summer we have more daylight time than other seasons of the year, so the system will rely on the PV panel more. Less night time hours lead to less usage of the battery so will likely less to be emptied. Also, the irradiance is at its peak during summer time which leads to higher energy supply.



*Figure 16 - Charging State Frequency*

Figure 16 here shows the charge state of the battery storage system in relation to the days. This information allows us to understand how well our storage system is doing and how well the PV panel provides energy to it. We can see that in about 48% of the days the battery will be charged to 90-100%. On other days the data shows that the percentage of the charging state is spread from 50% which is our input depth of discharge to 85% fairly evenly with a slight edge to 50-60% charging state. What we conclude from this graph is that our storage system is reliable because it is not fully discharged for almost 82% of the days.

### 3.23 Market product comparisons

When dropping our product in the market, there are going to be products that compete with ours. Customers are always overwhelmed with choosing the best product that meets their needs. This is where market product comparisons come in handy.

Observing our competition in the market will help us better understand what customers go for when buying a solar picnic table, through investigating the pros and cons in other products. In this section, we are going to list some products that have a similar idea behind them, but with different designs and features.

### 3.23.1 Eshine Solar Coffee Table

What is most interesting about this solar table is that the PV module is embedded into the table itself. The main idea behind this product's design is for it to be a leisure coffee table where customers can have their coffee while enjoying some features provided by the table, like wireless charging and wired standard USB and type-C charging, bluetooth speakers, and LED lights in some models. This solar coffee table is priced at \$222 per piece.

The PV module embedded in the table is a 40W monocrystalline solar panel with an efficiency of 17%. This panel is linked to a Lithium-ion battery that stores excess energy. Once the battery is fully charged, the table can be used for 3 - 5 days. Battery power information is provided to the customer via an LCD display that shows how much energy is left in the battery through printing a battery icon that is divided into five rectangular spaces corresponding to the battery charge level. The wired charging ports supply a power output of 2A 5V, which is enough to fast charge cell phones.

The hardware of the table is lightweight and its size is relatively small, as its dimensions are 23.62 in x 23.62 in x 27.77 in and it weights 40.35 lbs, this adds portability to its features, something our project lacks. As for its durability, the table is built with Aluminum alloy and has an IP65 waterproof rating.



*Figure 17 - Eshine Solar Coffee Table*

One key advantage this product has over ours is portability, it has a lightweight and a small size, which makes it easy to move it around. Another advantage is how long it can be used for once its battery is fully charged. This, apart from its Lithium-ion battery

efficiency, is due to the low number of loads that consume power, as the battery only needs to power a small LCD display, a wireless charger and two wired charging ports, and a speaker.

Our product has some advantages over this one. Like the Kay Park solar charge table, this product does not include an MCU, which we use in our product to make the customers' experience easier and gather all the information they need in one place. In addition to that, although our product is not portable, its solar panel's efficiency is higher and can get to 21.3% efficiency in perfect conditions. Also, the PV module that is embedded in Eshine's solar coffee table has some drawbacks, one of which is the solar arrays getting blocked from reaching the PV module, or part of it by any object that is placed on top of the table, while in our product, the panel is placed on top of a shade, which insures the panel is getting as much solar energy as possible. However, for its price, this solar coffee table is worth it.

### **3.2.2 Kay Park Solar Charge Station with Fiberglass Umbrella**

This product is similar to ours in some components, as they use a solar panel rated at the same power and a lead acid battery. Kay Park solar charge station is mainly designed to allow its customers to charge their devices in places where it is hard to access power. The only loads connected as outputs are two Apple and two Android standard charging cords, and four standard USB ports, all having a 2.1 Amp fast charge. This table is backed with a 7 amp hour sealed lead-acid battery that provides the loads with power when the PV module is off. This product is priced at \$3,119.



*Figure 18 - Kay Park Solar Charge Station*

The solar panel in this product is attached to a fiberglass umbrella that can be placed on any table that has a 1 ½" umbrella pole, which is an interesting idea that could make this product portable if it was not for its heavy weight of 221 lbs. Though it is

heavy, the fiberglass umbrella provides excellent protection against the sun, and the table hardware can withstand normal weather conditions.

Compared to our table, this product is more simple and focused on one goal, which is providing power for the customer's devices. Its advantages over our table are absorbing more devices to charge, and the fiberglass material that makes it more solid.

On the other hand, our product provides more features that our customers can benefit from. We provide lighting for the area around the table via the LED lights attached to it, which is very useful when using the table at night time. Furthermore, our product includes a microcontroller that we programmed the LED lights and the sensors on, and this microcontroller is linked to a cell phone application via bluetooth, that makes using the features and reading the information very easy and accessible, even with no network around.

### **3.23.3 Enerfusion Aurora Solar Picnic**

What is special about this solar picnic table is that it steps sustainability a level up, as this product is made using recycled material. The Aurora Solar Picnic table is made to be used on campuses and in any open space where sunlight can reach its solar panels. The main service it offers is providing power for the customers to charge their devices using different charging methods.

This table is powered by six 60W solar panels totalling 360W of power generation. All solar panels are placed on top of its shade structure. The generated power is then delivered to a solar charge controller that has digital readouts and can be connected to by an IOS or Android phone via bluetooth to monitor the system's information. The charge controller then distributes power to the loads, which are nine wired charging ports providing different USB type outputs, and four Qi enabled 3 coil fast wireless charging locations on a mini table, in addition to LED lights that are included, these lights will automatically turn on at dusk. The excess power generated by the panels is stored in a 40Ah gel cell battery bank.

The table is made of Aluminum to prevent any rusting, and the components are secured with stainless steel to protect them from weather conditions. This product's pricing starts at \$11,350, depending on the add-ons and coatings chosen.

Compared to our solar picnic table, this product has a higher battery capacity, and the power generated by its solar panels is 3.6 times the power generated by ours.



*Figure 19 - Enerfusion Aurora Solar Picnic Table*

Also, since sustainability is a goal of ours too, an additional advantage that this product has over ours is that its hardware is made of recycled material. This product has some features not provided in the two previously mentioned products, these are ambient LED lighting, and power information display, containing most of the key features we provide in our design, but with higher values in capacity and size. This can be viewed as an advantage and a disadvantage. It's true that its battery lasts longer, and its panel produces more power, however, this spikes its price, which makes it over the budget of a high number of customers.

### **3.23.4 Market product comparisons summary**

To sum up the market product comparisons, we compared three different solar tables available in the market today. Each one of these tables had a special feature that made it stand out over the others.

First we looked at the Eshine Solar Coffee Table, which costs \$222. The most desirable features in this product were its low price, and time of use once its battery is fully charged. This table is safe against normal weather conditions, and most importantly, it is portable because of its light weight and small size. This product provides two USB charging ports, and a wireless charger, in addition to a small LCD display that shows the power left in the battery.

Then, we looked at the Kay Park Solar Charge Station, which provides its customers with a solid and durable design because of the fiberglass material that the shade is built with. This product is designed to fit any table with a 1½", while carrying with it different fast charging ports so that the customers are provided with power in places where it is hard to find some. This product has a price tag of \$3,119.

Finally, after these two products, we moved to a higher price product, the Enerfusion Aurora Solar Picnic table, which has a price starting at \$11,350. The designers of this table had sustainability as a main goal to achieve, and they definitely did, as all tables are constructed with recycled material and all power generated is solar power. This product combined fast wireless and wired charging for all different devices, providing nine wired charging ports, and four wireless charging ports placed on a mini table. It also featured a high capacity 40 Ah gel cell battery, and a bluetooth connectable solar charge controller.

	Eshine Solar Coffee Table	Kay Park Solar Charge Station	Enerfusion Aurora Solar Picnic
Dimensions (in)	23.62 x 23.62 x 27.77	N/A	96 x 70.5 x 100
Price	\$222	\$3,119	>= \$11,350
Solar Panel Efficiency	17%	N/A	N/A
Solar Panel Rating	40W	100W	6*60W = 360W
Battery Type	Li-ion	Lead-acid	Gel Cell
Battery Capacity	N/A	7 Ah	40 Ah
Lighting	Not provided	Not provided	Ambient LED lighting
Power Information Display	0% - 20% - 40% - 60% - 80% - 100%	N/A	Number display

*Table 23 - Comparison of the different Solar Tables*

## **4. Project Design**

In this chapter, we will be going over the project's design. More specifically, this chapter will be split into two main sections: the hardware design and the software design.

### **4.1 Hardware Design**

In this section of the project design, we will focus on the hardware design of the project. Ranging from the electrical wiring between the components, to the PCB design and connecting the small pins between the sensors. Detailed information will be provided on the wiring procedure, specific part names, and design schematics.

#### **4.1.1 Electrical System Wiring Design**

Highlighting and explaining how the system wiring connections are designed is very important. The right connections in the right terminals must be made between all the devices. The wiring system will showcase the devices, terminals, and wiring connection of the system.

Starting with the solar panel, it will output around 12 volts to the charger controller connected to it. The connection between the two will be through two copper wires, positive and negative terminals. The input port of the charger controller has screw-in terminals. This means that there is no need for any connectors or adapters between the panel and the charger controller.

Then, at the central unit of this electrical system, is the charger controller. The charger controller will connect the PV solar panel, battery system, and AC & DC loads. The charger controller has 3 input/output ports, and each has two screw-in terminal ports. One is an input port that connects the PV panel, the second one can be either an input or an output depending on its interaction with the battery, and the last one is the DC load output supply.

As we have just mentioned, we don't need any connectors for the input port from the PV panel. For the port connecting the battery, we won't need it either because the Lead Acid battery that we are using uses two wires each connected at one terminal and that can be directly connected to the charger controllers port. As for the port connecting the charger controller to the loads, this will depend on the device connected to the other side. A connector or an adaptor will be used for any need of connecting the two wires coming off the charger controller into any other type of port. 12 volts DC will be the output voltage of the charger controller.

The battery storage system in our project will only be in a charging state when the energy generated by the solar panel is greater than the energy consumed by the loads. When that happens, the charger controller will supply the energy generated by the PV

panel into the battery outputting 12VDC. When the load demand is higher than the energy being generated, or in most cases at night time, the battery is going to be in a discharging state and the charger controller will draw current from the battery and use it to supply the loads being used. This interchange between charging and discharging will be controlled by the charger controller.

In the rare case of when the load demand being equal to the generation then the charger controller will directly supply the loads without charging or discharging the battery.

The charger controller output will supply two load types, DC and AC. At the AC load, the charger controller will supply the power outlet through the inverter. The inverter operates at 12 volts which is the output voltage of the charger controller. To connect the two terminal wires coming from the charger controller we would need a connector in between to accommodate the input port of the inverter. The connector should be able to connect the two-terminal wires to the cigarette lighter input port of the inverter.

Then at the other end of the inverter, it will supply AC current to the power outlet. The power outlet uses a three wiring system. It has a hot, neutral, and ground wire. The hot wire will carry the electric current from the inverter to the outlet. It represents the positive terminal. The neutral wire completes the circuit providing a returning path for the current. The ground wire is a safety wire in case of a short circuit or a fault.

Now looking at the DC loads, the charger controller will supply current to the microcontroller, LED, and the sensors. Since the output voltage of the charger controller is 12 volts and the input voltage of the other devices are different, there will be one common input port at the PCB and it will take care of all the operating voltage differences. All the DC devices will be supplied current through the PCB.



Figure (20) - Charge controller to Barrel Jack port connector

After looking at the PCB design section and seeing that the input port is a female barrel jack, we know that to connect the two copper wires coming from the PWM charge controller we require a connector that will connect them to the barrel jack port of the PCB. We have found the suitable part for this in Digi-Key. The part is from the manufacturer **Adafruit Industries LLC** with the Digi-Key part number **1528-1387-ND**. Using this connector the two-terminal wires of 12 volts will be connected into the PCB.

The LED, temperature sensor, and the motion sensor all have three pins. VDD (positive terminal), VSS (negative terminal), and the communication pin. Two of them will be connected to the PCB for power supply. The VDD and VSS pins will be connected to the PCB output ports through male-to-female jumper wires. The communication pins will be connected to the MCU.

So for these load devices, they will get their power supply through the PCB output and the data communication pin will be connected to the microcontroller.

The output ports of the PCB will be discussed in the PCB Design section below.

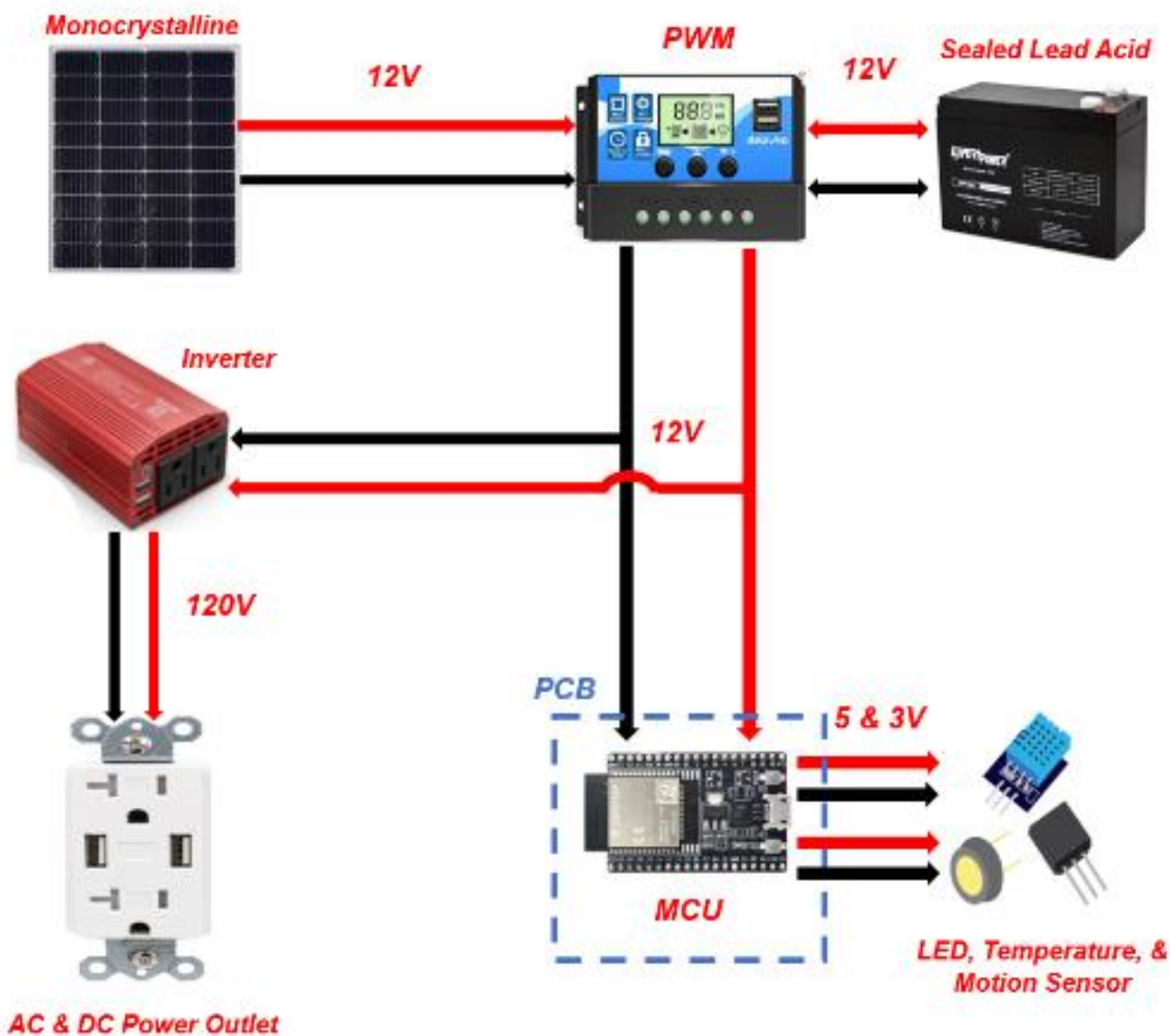


Figure 21 - Power Distribution

## 4.1.2 PCB

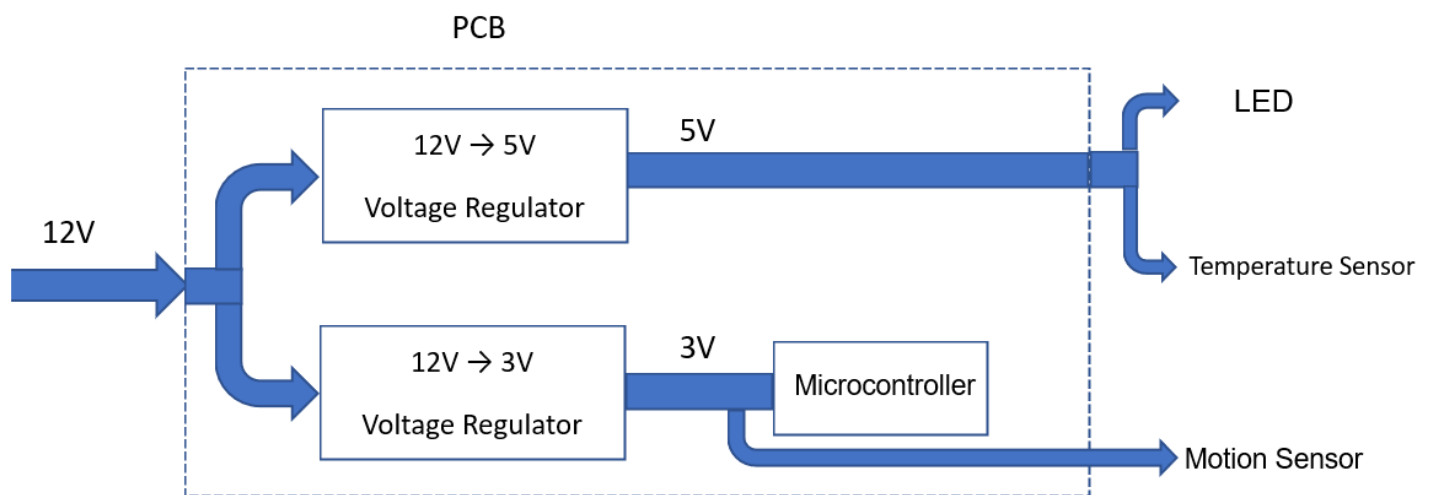
As we have discussed in previous sections, our PCB design will provide the suitable voltage and current magnitudes to the DC load devices. The figure (22) below illustrates how the PCB will function. It will receive 12 volts DC from the charger controller as shown in figure (previous figure). Then, the input 12 volts will supply power to the two parallel-connected voltage regulator circuits, one will convert the 12VDC to 5VDC and the other will convert the 12VDC to 3VDC. The PCB board will have one common input port and two separate output ports.

The **ESP32** microcontroller will be connected within the PCB and it will receive power supply of 3 volts and 250mA as per its optimum requirements.

The 5V voltage regulator circuit will supply the LED and the temperature sensor (**DHT22**) with 2.4A and 1.5mA respectively.

The 3V voltage regulator circuit will supply the microcontroller which is within the PCB and the motion sensors (**AM312**) with 250mA and 20 $\mu$ A (each) respectively.

Since the Temperature sensor has an input voltage range of 3-5 volts, it can be connected to either the 5V output port or the 3V port. For maximum efficiency of the voltage regulators, it will be connected to the 5V circuit since it has a higher maximum current output of 3A. Figure 22 below illustrates the design.



*Figure 22 - Voltage regulator design*

## 4.1.2.1 Schematic Design

The next step in this procedure is to design each of the voltage regulator circuits. For our design, we have used the software **Autodesk Eagle** for the circuit schematic and the final PCB design.

To design the circuit, we have used the WEBENCH Power Designer tool from Texas Instruments to generate circuit designs that satisfy our inputted circuit requirements and constraints. Then, from the resultant circuits, we choose the most optimum one that satisfies our design requirements. This process was done twice for each voltage regulator circuit. The design attributes that had an effect on the circuit selection were the efficiency, BOM (Bill of Materials) costs, and the BOM count. The footprint size was not a factor because our project does not require us to have a smaller PCB. We have generated two separate designs, one for each voltage regulator circuit. The first generated circuit outputting 5v and 3A maximum is shown below.

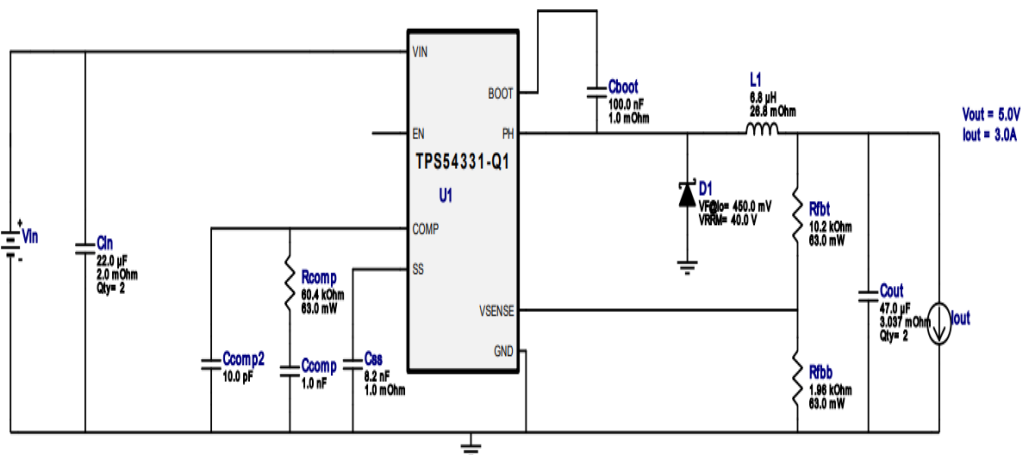


Figure 23 - 12V to 5V regulator schematic

The second generated voltage regulator circuit outputting 3V and 0.5A maximum is shown below.

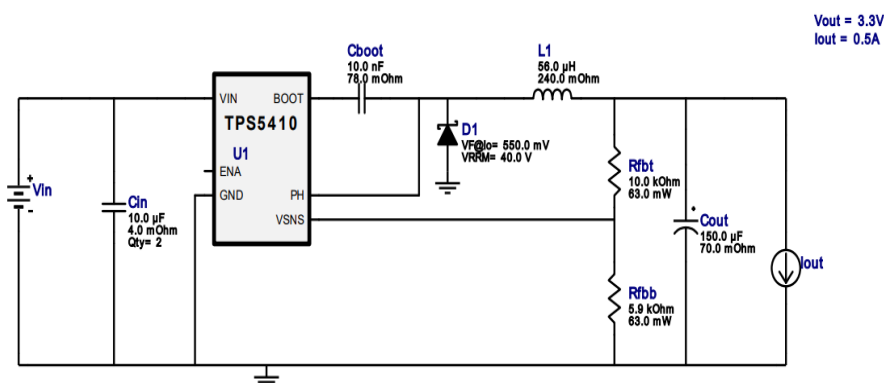


Figure 24 - 12V to 3.3V regulator schematic

Using these two generated designs, we have connected them together in Eagle with one common input port, while having two separate output ports. The microcontroller was added in too.

We have decided to connect the 12V coming from the charger controller with the PCB using the two-terminal wire to male barrel jack connector (**1528-1387-ND**). Then the input port of the pcb must be a female barrel jack connector. The female barrel jack connector we are using in our schematic is PJ-102AH from digi-key. This connector can handle 24 VDC and up to 5 Amps which is higher than our rating requirement. Through this connector, DC current will be supplied to both voltage regulator circuits.

Now, for the output ports the decision of its type depends on the input ports of its corresponding load device. Looking at the datasheet of the sensors and the LED we can see that they have different types of input ports. For convenience, we have decided to use the two terminal screw-in connectors. Using these connectors allows for more flexibility because it is very common. We can use jumper wires or other types of connectors to input the loads. These connectors can be seen on the outputs of the circuits on our schematic in figure (27). They can handle up to 10 Amps which is way over the loads current ratings. The connectors (**EBWA-02-A**) will be purchased from digi-key.

Using the parts mentioned above and connecting the two voltage regulator circuits this wraps up the schematic of our PCB. Its final schematic design is shown in the screenshot on figure 25.

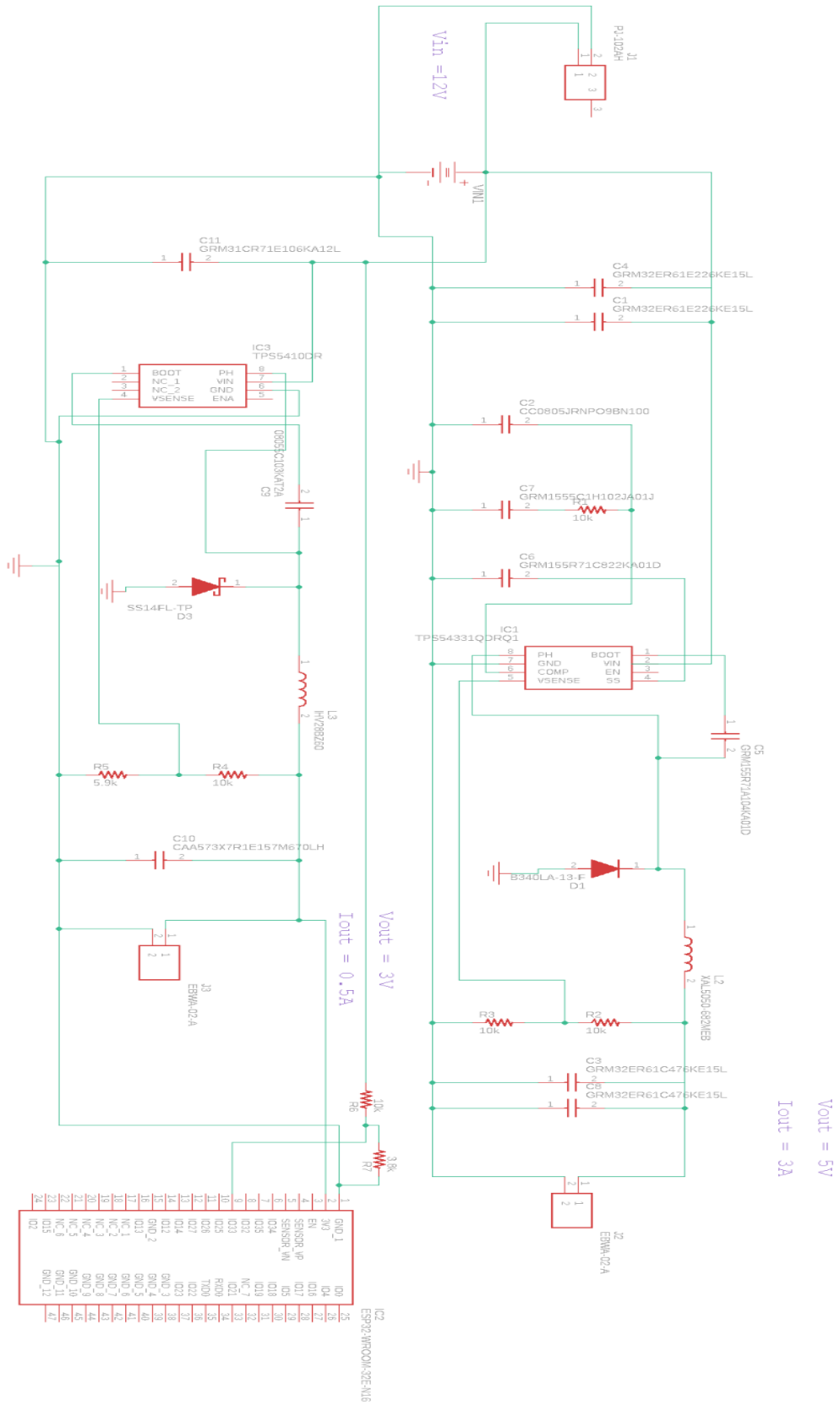


Figure 25 - Schematic Drawing

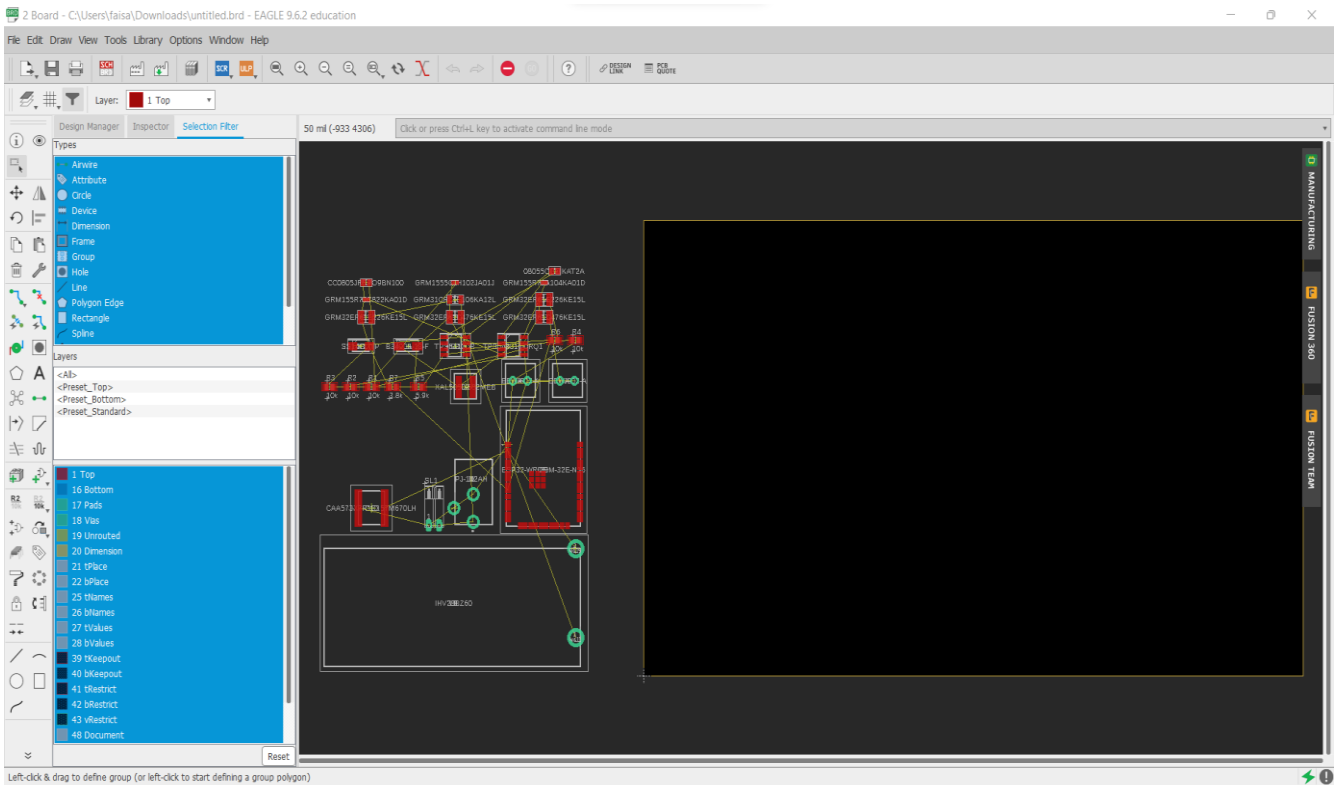
After designing the schematic of the two circuits and before we convert it into a PCB, the next step is going to be making sure that all the parts used in the circuits are available. We cannot proceed into later steps until we know that the BOM parts are available online for purchase and at a reasonable price. Here below is a table that provides all the information needed about all the parts that will be used in the PCB design.

Quantity	Value	Package	Vendor	Availability	Unit Price (\$)
1	Power Connector	PJ-102AH	Digi-Key	91,590	0.82
2	Power connector	EBWA-02-A	Digi-Key	1,159	0.57
1	Voltage Regulator	TPS5410DR	Digi-Key	55	2.99
1	Power Connector	1528-1387-ND	Digi-Key	100	2.00
1	Voltage Regulator	TPS54331DR1	Digi-Key	2,493	3.68
1	Diode	B340LA-13-F	Digi-Key	26,727	0.43
1	Diode	SS14FL-TP	Digi-Key	55,918	0.38
1	Inductor 6.8uH	XGL5050-682MEC	Digi-Key	1,073	3.43
1	Inductor 56uH	744771156	Digi-Key	8,293	2.26
<b>TOTAL</b>					17.13

*Table 24 - BOM of components*

## 4.1.2.2 PCB Design

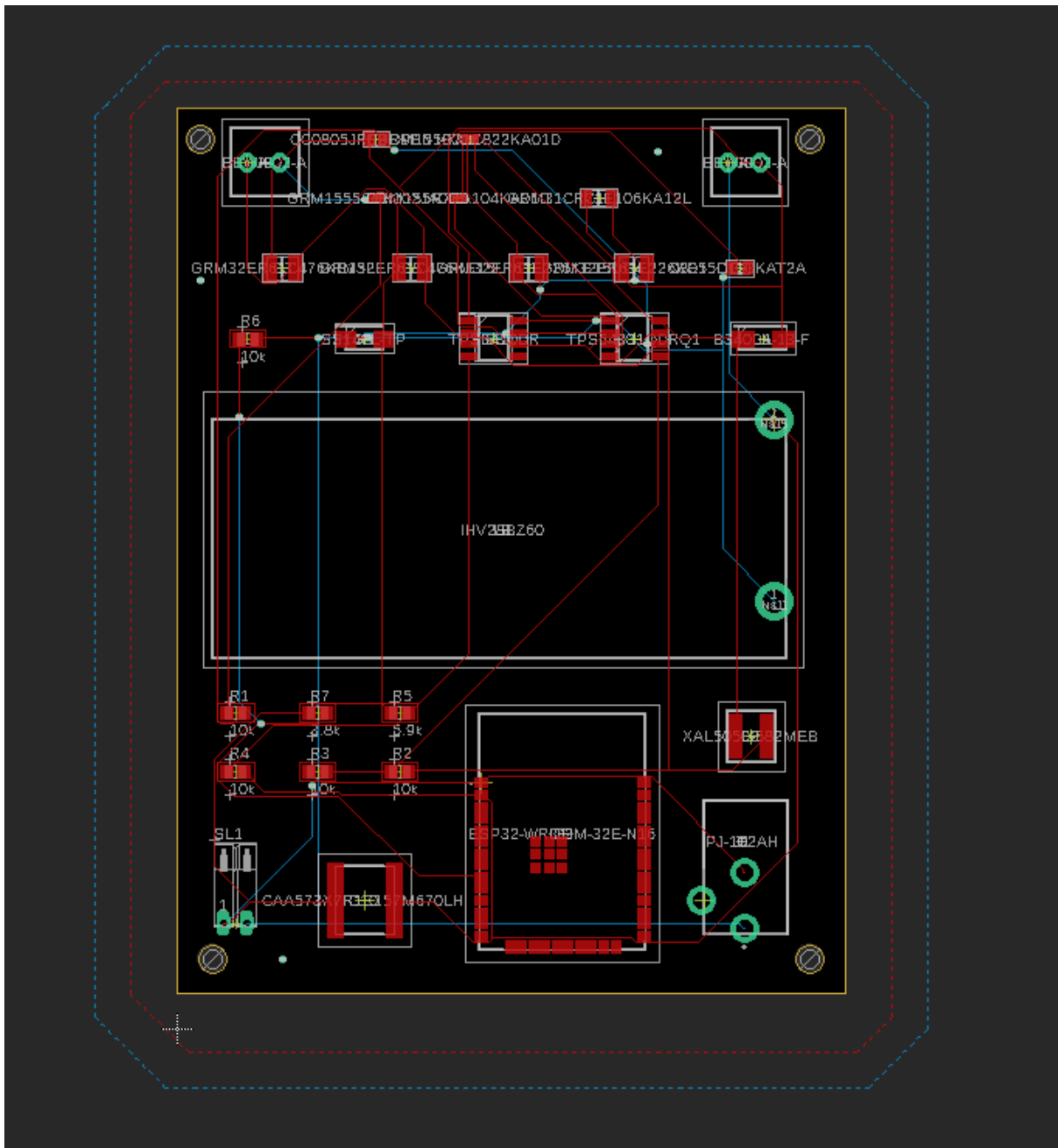
This step is where we convert the schematic we designed to a PCB. The first step towards achieving that is by switching to the BRD screen on **Autodesk Eagle**, which allows us to design a board using the footprints of the components we used in the schematic.



*Figure 26 - PCB Design Window*

The figure above shows the window that pops up when the BRD icon is clicked. We can see the footprints of all the components in the schematic. The yellow square on the right of the footprints resembles the bounds of the board. Then, on the left side of the window, we have a panel that is filled with tools to help us build our board.

For our PCB's final look design, we had some goals that we had in mind all along the designing process. First, we wanted our PCB to be as compressed and small as possible, while giving the components enough space to function properly and to avoid overlapping. This will reduce the PCBs price, as bigger PCBs cost more. Then, we wanted our power, input, and two output ports to be separated, and preferably, place each one of them on a separate corner, so that they are easily differentiable in the final design, and so that we avoid any concerns regarding the size of the wires we are connecting them to. Lastly, we wanted to avoid wiring the components at a 90° angle, so that we keep the electromagnetic interface as small as possible, as big electromagnetic interfaces can affect the functioning of some components.

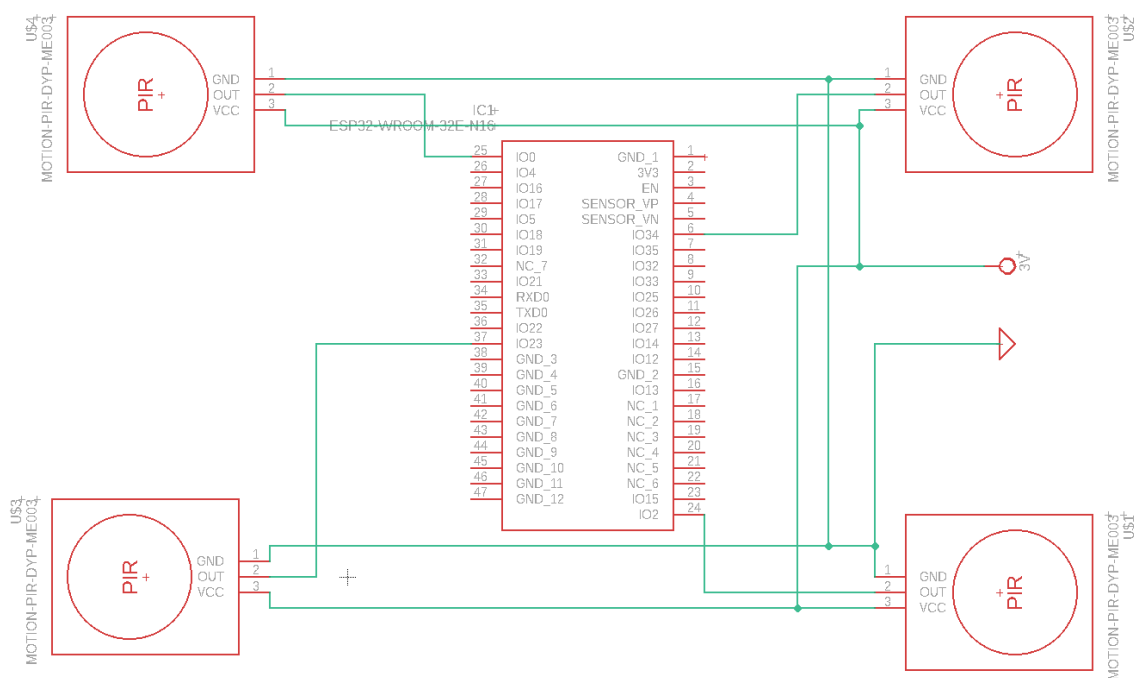


*Figure 27 - PCB final design*

This figure shows our final PCB design. We have achieved all our PCB design goals. From the figure, we can see that the components have sufficient spacing between them, yet the design is compressed and with no unnecessary space. Then, we have separated the power, input, and two output ports from each other, they are distributed as follows: The component on the bottom-left of the design resembles the power input. We chose a two port power input, a port for V+, and a port for ground (GND). The component on the bottom-right of the design is our voltage regulator circuit input port.

The two voltage regulator outputs are on the top-right and top-left of the design, the top-left components being the 5V 3A output, and the top-right is the 3V 0.5A output. Finally, as can be seen from the figure above, all wiring was done without a 90° turn, keeping our electromagnetic interface as small as possible.

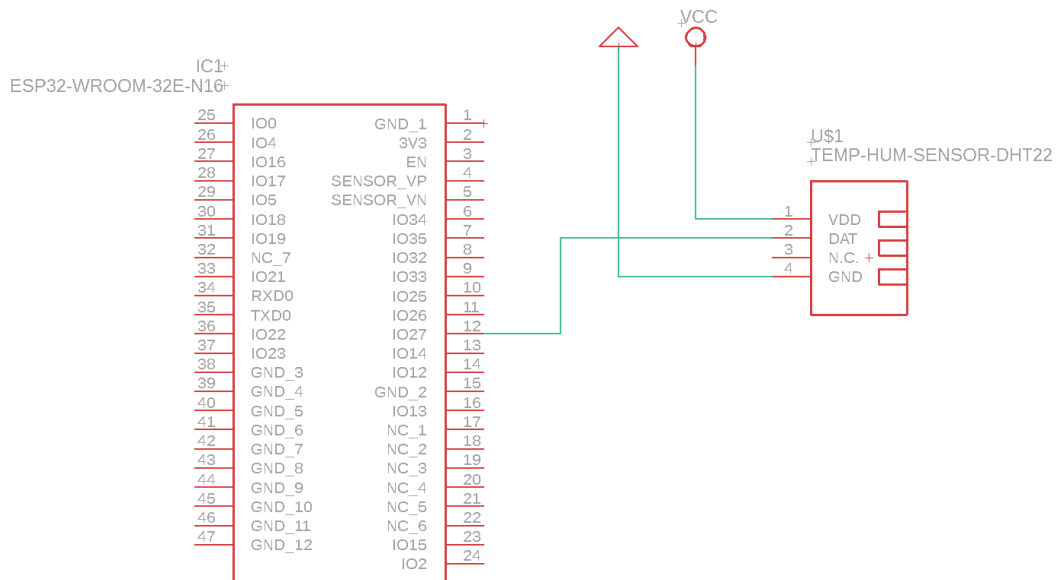
### 4.1.3 PIR Motion Sensor



*Figure 28 - Schematic for PIR motion sensor*

For the hardware design of the motion sensor, it will be externally connected to the PCB. The AM312 motion sensor comes with three pins: VCC, DATA, and GND. Since the detection range of the AM312 is 100 degrees, we will be connecting four external modules to the MCU to allow for full detection coverage. To power the motion sensors, we will connect the VCC pin to the PCB's 3 volt pin. Furthermore, the DATA pins will be connected to the GPIO pins of the MCU. For this, which pins they are assigned to is important. This is because we want to have the largest detection range possible. The four DATA pins of the motion sensors will be placed at each corner of the microcontroller. We have determined that this is the most optimal placement of the pins. These will be the following GPIO pins that will be used: 34, 2, 23, and 0. Finally, for the ground pins, they will be connected to the PCB's ground pin.

## 4.1.4 Temperature and Humidity Sensor



+

*Figure 29 - Schematic for the DHT22*

For this section, we will be detailing the overall hardware design for the DHT22. The figure above serves as a visual representation of this design. The design for the DHT22 is fairly simple. The module that we chose comes with three pins: VCC, DATA, and GND. In order for the microcontroller to be able to communicate with the sensor, we will connect the DATA pin of the DHT22 to the GPIO27 pin. With this, the microcontroller will be able to take in the temperature and humidity readings outputted by the DHT22. For the mcu and DHT22 sensor to work properly, it is important that we follow this design.

## 4.1.5 Voltage Divider

The main functionality of the voltage divider is so that the microcontroller can be powered by the battery. On top of this, it will allow us to correctly read in the battery's output voltage which is an important element to the battery monitoring feature. Since the battery is outputting 12 volts, if we were to connect it to the microcontroller, it can possibly damage the microcontroller since the maximum input voltage is 3v3. To prevent this, we will need to find the correct two resistor values to cut down the voltage from 12 to 3.3. We will use the voltage divider formula to calculate these two resistor

values. From the formula, we have determined that the first resistor (R1) will have the value of 10K ohms and the second resistor (R2) will have the value of 3.8K ohms

From here, we can begin connecting the voltage divider to the microcontroller for battery monitoring. We will begin by connecting the positive terminal of the battery to the R1 resistor. Next, we will connect the battery's negative terminal to the PCBs ground and R2. This means that the negative terminal and R2 will be grounded. With the two resistors in series, we will finally connect the two resistors to an analog to digital converter pin on the microcontroller. We have found that the GPIO33 pin is the most optimal. If designed correctly, the microcontroller should be able to read in the analog values of these pins so they can be outputted for the user to read. Below is what the final design of the voltage divider should look like.

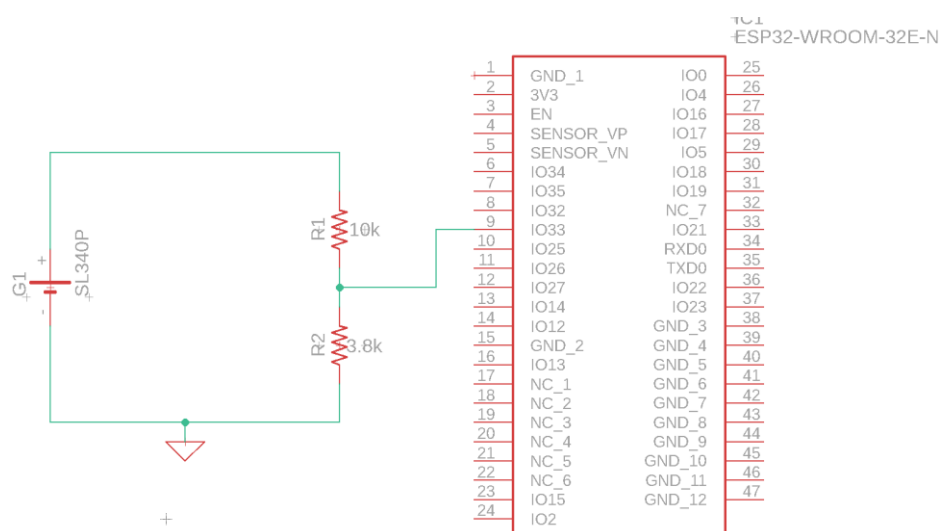


Figure 30 - Schematic of Microcontroller Connected to Voltage Divider

## 4.2 Software Design

The different functionalities of the solar picnic table will be implemented using the microcontroller. However, first, it is important to understand the software that enables the microcontroller to interact with the various hardware. In this section, we will be going over the different aspects of the software design.

### 4.2.1 Software Development Tools

In this section, we will be detailing the different types of software that we used when going through with the design phase. As stated previously, all software functionalities

will be implemented through the ESP32 microcontroller. To do this, we will be utilizing the Arduino Integrated Development Environment (IDE) and the C programming language. When designing the app for both platforms, Android and IOS, we will be using React Native for the majority of the frontend and user interface. When it comes to simulating and testing the app, we will be utilizing Xcode for IOS simulations, and Android Studio for Android simulations. All of the software development tools that we will be going over are essential to our design phase so it is important to understand their role in our project.

### **4.2.1.1 Arduino IDE 2.0**

For the IDE of this project, we decided to use the Arduino IDE 2.0. This IDE is a direct upgrade from the Arduino IDE 1.0 as it comes with much more improvements such as a built-in debugger and increased performance that will be helpful for developing our code. The IDE is equipped with a text editor and console to allow for easy and streamline coding.

To create any sort of code, our group will be utilizing the sketch feature of this IDE. A sketch is any sort of code that is written in the text editor. Using these sketches, we will be able to write the various programs that will allow for the microcontroller to operate correctly. Along with this, the IDE comes with built-in functions that can be used with the sketches. These sketches will be written in the Arduino language which is based off of the C/C++ programming language. Since we have past experience on using this language, it will help us tremendously when developing our software programs.

Another key feature of the Arduino IDE is that it supports third-party hardware. Since it was developed by Arduino, the IDE has built-in support for Arduino based microcontrollers. But, thanks to this feature, it can also support the Espressif microcontrollers. More specifically, the microcontroller that we will be using for our project: the ESP32-WROOM-32E-N16. On top of this, the IDE is cross-platform so it can be used across all operating systems.

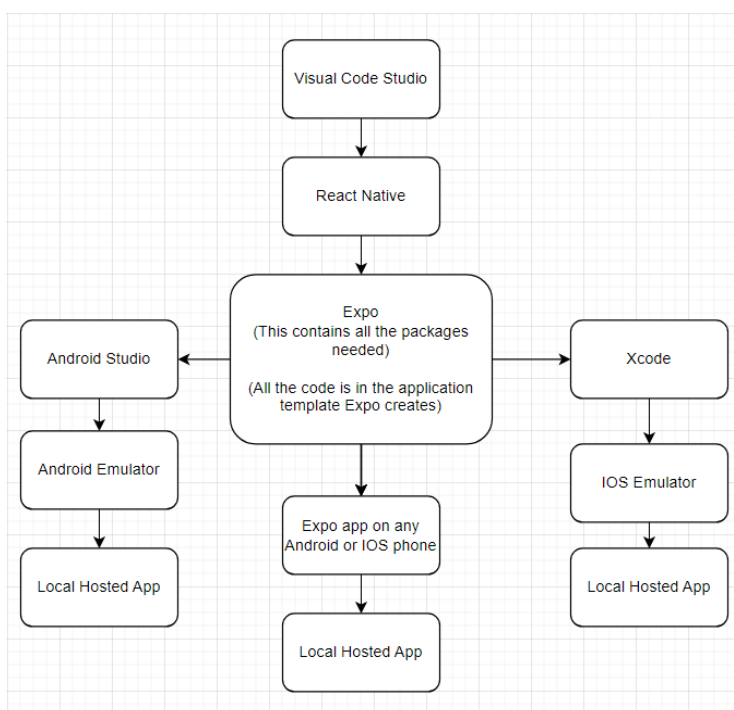
All of the code that is created will be uploaded into the microcontroller using the Arduino IDE. Overall, with its built-in functions and extreme versatility, it will be a great fit for our project during the software development process.

### **4.2.1.2 Mobile app**

For the mobile app of this project, we will be using React Native to program the application. React Native provides us the flexibility to program for both IOS and Android platforms without having to switch between two different platforms when creating the same application. To start coding using React Native, we need a code editor. We will be utilizing Visual Studio Code to edit our code because of familiarity, and the plentiful number of extensions that help developers code faster.

To begin the project, there are a few things that need to be installed to start the developing process of the mobile app. To provide an easy group of tools and libraries needed to create a mobile app for both platforms, we will be using Expo which helps us download all the necessary packages for development. Expo also provides us tools to simulate the mobile app on their desired platform without having to download the application onto our phones for every single change. Downloading applications without proper simulation and testing can lead to failures onto the phone itself such as if a memory leak is present in the application and it is downloaded onto the phone, it can be present forever until the phone is completely wiped. Expo gives us commands to run multiple emulators in Android, IOS, local hosted web applications, and can provide QR codes that allows users to download Expo's app and scan it to run a hosted version of the application.

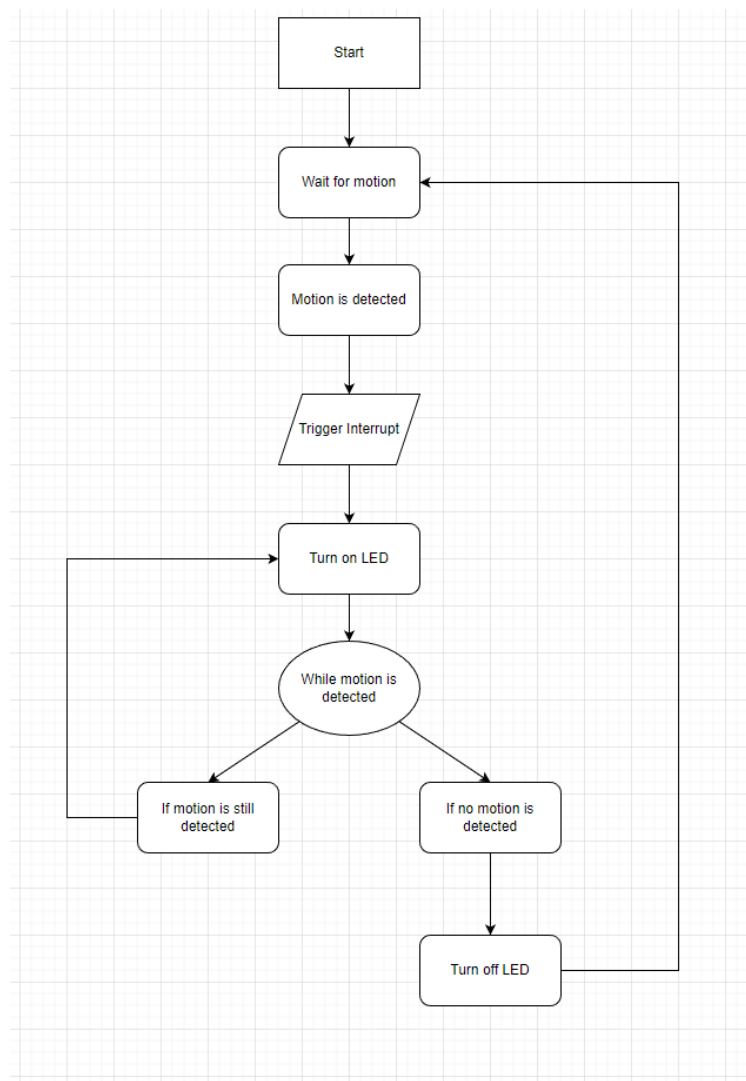
For Android emulators, Android Studio is needed on the computer for Expo to interact with it. Expo opens Android Studio which has a local copy of the application installed on the Android emulator which allows us to simulate using the application on any Android phone which is very important since none of us own an Android device. For IOS emulators, Xcode is needed which is only available on MacOS devices. Expo opens Xcode's IOS emulator which has the local copy of the application and then we can interact with the app as if we had the physical device. The Expo app is available on both Android and IOS devices. Once downloaded, we can open the app on our own phones, scan the generated QR code Expo creates on our computer, and then a local hosted version of the app is available on our phones.



*Figure 31 - Flowchart of how the Emulators Work*

Another application we will be using when developing the mobile app is Figma. Figma is a collaborative web application used for interface designing. Figma allows us to create a prototype of the UI without having to program anything. This allows for changes in the UI quickly without having to rewrite any code. Once the prototype of the UI is done and the general design language, color pallet, etc is complete, we can move onto actually programming the UI. Figma also allows for anyone to view the prototype and make changes they want without needing any prior knowledge of programming allowing everyone in the group to participate in changing up how the UI will look in the end.

## 4.2.2 Motion Sensing



*Figure 32 - Flowchart of Motion Sensing and LED Interaction*

To implement the logic of the flow chart above, there are some factors that we must take for it to function properly. First, to signal whether to wait for motion or if it was

detected, we must utilize a flag variable. This is important as it will indicate to the microcontroller when motion is detected or not. The main function of the flag variable will be to indicate when the microcontroller should set off the interrupt. Another important factor is to know when to turn the LED off and on. To account for this, we will use a variable to store the time of the last detected motion and another variable to determine the amount of time after which the LED should turn off if no motion is detected. These two variables are essential to the functionality of the motion sensor as it prevents the LED from being turned off or on at inappropriate times.

To fully implement the software design of the motion sensor, we must take into account both the logic and the factors. With these combined the code should operate in this manner. First, we will define the respective pins that the LED and PIR sensor is attached to. Then, we will initialize three important variables: motionDetected which will be our flag variable, lastMotionTime which keeps track of the amount of time the LED should stay on after motion is detected, and the motionTimeout which specifies the amount of time that the LED should stay on after motion detection. Once done, we will attach an interrupt to the motion sensor pin and set it to trigger whenever the pin's state changes. Now, if motion is detected, it will be sensed and the pin's state will change. Once the state is changed, the interrupt will be triggered causing the flag variable to be toggled and updates the lastMotionTime. Within the loop, it will check whether the flag variable is true or if the time since the last motion detection is less than the motion timeout. If either of these conditions are satisfied, then the LED will turn on, if not then it will be turned off. If all of these are implemented correctly then the motion sensor and LED light interactions should function properly.

### **4.2.3 Temperature and Humidity Sensing**

The logic of the temperature and humidity sensing is fairly straightforward. The only thing to keep in mind is that the DHT22 uses a specific library. This library contains built-in functions needed to correctly read in these values.

To implement this software design, we first need three variables to store the temperature in Fahrenheit and Celsius, and the humidity. Once done, we will use the built-in function of the DHT library to store these values. The temperature reading is originally outputted in Celsius so we will convert it into Fahrenheit. Once all of the values are inputted and the conversion is completed, they will be outputted onto the mobile app for the user to access.

### **4.2.4 Battery Monitoring**

The battery monitoring design has an important software element to it. In the hardware aspect, the voltage divider will do most of the physical work by cutting down the input voltage to a manageable amount for the ESP32 microcontroller. The software aspect will do the voltage conversion so that it is a usable value.

To begin, the code will first read in the analog value of the pin. These values can range

from 0 to 4095 which is a representation of the voltage level. From there, we can take this value and convert it into a percentage using the `map()` function. This value can be sent over from the microcontroller into the mobile app for the user to keep track of while they are using the solar picnic table.

## 4.2.5 Bluetooth Connectivity

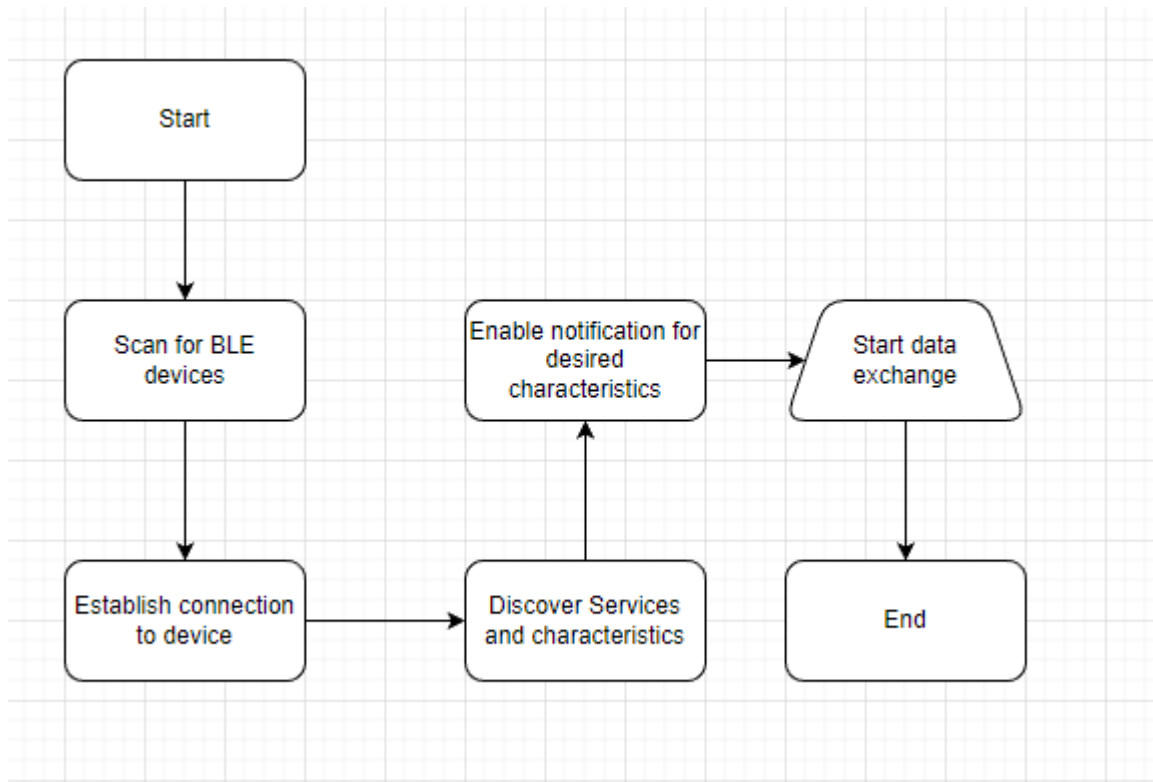
One key component for the mobile app to function properly is to secure a stable bluetooth connection. For this project, we will be using Bluetooth Low Energy (BLE). Since iOS isn't compatible with bluetooth classic, BLE will be the most optimal route to ensure compatibility. BLE works similarly to Bluetooth classic but it is optimized for a lower energy consumption which is perfect for our project.

For iOS we will be using a different software design as opposed to Android. In the Arduino IDE, we will be using the BLE unique libraries to implement this functionality. To start, we will be using two different UUIDs. UUIDs are unique identifiers that are used to identify the BLE services and characteristics that will be used for data transfer. We will build a BLE server class that contains two methods. Each of these methods will be used when a client connects or disconnects from the server.

Moving onto the setup, we can create the BLE device which will show up on the connecting devices bluetooth interface. Once done, we will be able to create the BLE server. The reasoning for this is that the server is responsible for providing the BLE service to the clients meaning that it is important for any sort of data transfer. With the server created, the next step is to create the BLE services. The service(s) can be assigned a service UUID which can then be used by the connected device to find that service and its characteristics. Furthermore, these services are assigned a characteristic UUID which identifies the specific characteristic that service has. In this case, we will use the NOTIFY characteristic as it sends notifications to clients as different values change.

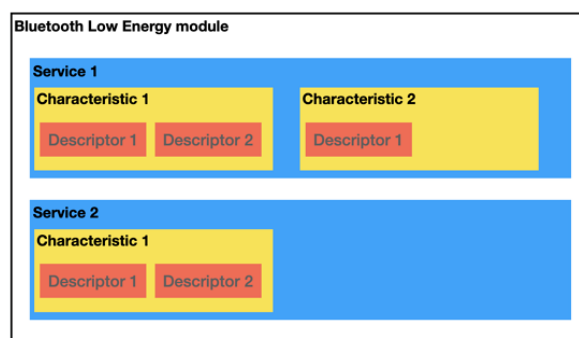
Once these values are properly assigned, the ESP32 microcontroller is ready to be connected to. To do this, we simply start the service and any device that is BLE compatible and within range will be able to connect to the microcontroller. The figure 33 shows a representation of the steps to BLE connection and data transfer.

When connecting to Bluetooth LE, we will be using the GATT communication profile which allows our application to see the ESP32 as a group of services similar to how APIs have their own separate routes. Each service has its own characteristics and then each characteristic has their own descriptors. Each service in BLE provides some feature which in our case would be the sensors connected to ESP32, so we would have services that would read voltages, temperatures, etc. Each service has characteristics which act as the properties of that service so it would describe things such as manufacturer or what the temperature is. Each characteristic has a descriptor which goes into detail what that characteristic provides such as the unit or standard the sensor is measuring at.



*Figure 33 - Flowchart for BLE connection*

Each sensor will need to be tested using this methodology. For example, our temperature sensor will have to fit this format. Referencing the figure below, our temperature sensor will be acting as “Service 2.” The main value we focus on in this “Service 2” is the characteristic which we will read to get the temperature value the sensor is detecting. The descriptor will describe this value we are reading by specifying what unit we are measuring. Once we have these values, we can send it off to the mobile application where it will be displayed for the user to read.



*Figure 34 - BLE architecture*

Bluetooth LE sends data asynchronously so transferred data is not instantaneous. Every time we would need information from a service, we would need to ask the ESP32 for the data, and then once it receives that request, it will then send data back from the sensor. This is how BLE functions to reduce the amount of power it will need.

How the phone will handle this asynchronous data is explained later in the testing mobile app section.

## 4.3 Test Plan

Before executing any project, it is a crucial step to test all the components that the project consists of. Since we have not yet started executing our project, we will take each individual component and design a test plan to make sure that all components yield the expected output and are compatible with all other components as well as our project's power constraints.

### 4.3.1 PCB testing

Testing the PCB for proper functionality is a very crucial step before implementing it in the final design. There are going to be several steps conducted to fully test the PCB.

The first test to be conducted on the PCB will be the main functionality test. We will test if it can regulate the output voltage within the specific range required when we supply it with an input voltage. This is done by connecting the output to a multimeter and measuring the output voltage. The input can be connected to a variable DC supply. Then measuring the output voltage while changing the input slowly. If the output is still within the specified range, then it passes this test.

Another important testing method is load testing. We need to make sure that the PCB will still maintain the specified output voltage when attached to different loads. This can easily be done by fixing the input voltage at 12 volts, then after ensuring that the output voltage at the specified value, we connect different loads with different values while using a multimeter reading the output voltage. If the output voltage reading in the multimeter remains steady through the load changes then the PCB passes the load test.

Then, the next test is to measure the ripple of the output voltage. It is important because the output voltage should be as steady as possible with little to no ripple. The test can be conducted by connecting an oscilloscope to the output and reading the voltage output graph displayed. If the oscilloscope display shows that there are no ripples in the voltage output then we know that the PCB functions correctly.

The last PCB test we will conduct is the temperature test. This test is important because although the PCB will be enclosed in a waterproof box our project will be outdoors and this makes it subject to temperature changes. We want to make sure that changes in the temperature do not affect the output voltage. This test can be done by subjecting the PCB to different temperatures and using a multimeter to measure the output voltage.

Once the PCB passes these four tests then we will feel confident in implementing it to our final design. In the case that it fails any of these tests then we will redesign it

whether the circuit needs changing, or the components placements, or the size of it. Based on what was the issue, changes will be made and then the retests will be conducted.

## 4.3.2 Solar panel testing

Our solar panel provides all our systems with power and without it functioning properly, our whole product won't work. It produces 100W at 12V in perfect conditions. To test the solar panel, we are going to need a multimeter to measure the voltages and currents it produces in these different cases:

The first case will be to test the solar panel's open circuit voltage. The open circuit voltage is the voltage the panel produces when it is not connected to anything. This voltage reading should be close to the panel's output rated voltage. From the manual, this value should be equal or close to  $V_{oc} = 21.82V$  for our panel.

The second step is to measure the panel's short circuit current. The short circuit current is the current that is produced when the positive power terminal of the panel is connected to the negative terminal. According to the manual, our short circuit current should be equal or close to  $I_{sc} = 5.55A$ .

The next step would be to measure both the voltage and current produced by the panel at the maximum power point using the multimeter. This means the voltage and current produced when the panel is producing its maximum output power. For our solar panel, the voltage at maximum power point should equal or be close to  $V_{mp} = 19.06V$ , and the current at Pmax should equal or be close to  $I_{mp} = 5.26A$ .

The last step of testing the solar panel will be testing it under a load. To perform this testing step we are going to connect our solar panel to our lead-acid battery. Then, we will measure the voltage and current produced by the panel and ensure that it is operating under the range specified in its manual.

By following the steps above, we are going to make sure that our solar panel is operating as it is supposed to and that it is ready to be installed to power our design.

## 4.3.2 Charge controller testing

The charge controller is a key component in our design. It will be serving as a connection bridge between the solar panel, the battery, and the loads. To test our PWM charge controller and ensure it is behaving as described in its data sheet we are going to need some components, such as a power source, a multimeter, a battery, a load, and an oscilloscope. Then, we are going to perform the following testing steps:

First off, we start by measuring the voltage of the power source, in our case, the power source will be our solar panel since the charge controller we are buying does not accept a DC power supply as power input. Once the voltage of the solar panel is

measured, we have to check that it is the range that the charge controller can accept, which in our controller is specified in the manual to be 12V for up to 390W, or 24V for up to 780W.

The next step is to test the output voltage of the charge controller. We do that by connecting the battery or another load to it and then measuring the output voltage of the controller. The value we get should be compatible with the values provided in the charge controller's manual.

Since we are connecting a PWM charge controller to our battery, it is important to test the pulses of current that the controller will send to charge the battery. To do that, we are going to use an oscilloscope and connect it to the charge controller. We should observe a regular and consistent pattern of pulses in the oscilloscope.

After that, we should be sure that our charge controller is properly charging the battery. We are going to measure the battery's voltage before charging it using the multimeter, and then re-measure its voltage after charging it using the charge controller. The battery's voltage should increase after charging it.

Now all that is left is to test the charge controller's load output. We are going to connect a load like a light bulb or any other load to the charge controller, then, we will measure the voltage and current delivered to the load. The readings we get should be within the range provided in the charge controller's manual.

By implementing the testing steps above, we will ensure that our PWM charge controller is functioning correctly and is ready to be implemented in the final design.

### **4.3.3 Inverter testing**

Before connecting the inverter to the power outlet where our devices will be charged we must first make sure that it functions as it is described in its datasheet. We will perform a couple of tests in order to do that.

The first test is going to be the main functionality test. We will connect an oscilloscope to the inverter's output and display the output waveform. Using the oscilloscope we will be able to determine if its output voltage and frequency match our requirements, also we will be able to make sure that it is indeed a pure sine wave inverter.

Then the second test will be the load test. We will again connect the output of the inverter to the oscilloscope then connect a load while checking on the displayed waveform on the oscilloscope. The inverter would pass the test if no significant changes happen to the waveform when connecting the load.

These two are the main tests which we will depend on before implementing the inverter to our design. But we will still do other tests to make sure that everything in the inverter functions properly. These small tests include making sure that the cooling system (fan)

inside the inverter is powered in when the inverter is on. Another test would be to test the led lights on the inverter design which are important because they tell us about the inverter's condition.

Once the inverter passes these tests we will implement it in the final design.

### **4.3.4 Battery storage system testing**

Testing the battery before using it in our system is one of the most important steps of making this project.

Testing it is so important to ensure, sufficient storage capacity, sufficient power output, and safety.

Before testing it the first step is to make sure that the battery is fully charged. If not then we will charge it following the instruction in the manual provided with the purchase. Then when the battery is fully charged we will use a multimeter to measure the open-circuit voltage. It should read a value about 12 volts.

To measure the battery's capacity we will perform a load test. First, we connect a load to the battery that will approximately use 8/3 Amps of the rated 8Ah of the battery. Then we will measure the voltage using a multimeter and wait for a voltage drop. Once the voltage drops from 12 to a value below 11 volts then we would multiply the load current by the time it took for the voltage drop. The result should be a value close to 8Ah.

These two tests are the main tests and other than this we will make sure that the temperature of the battery is at normal levels while charging and discharging. If the measured voltage and the calculated current match the respective battery ratings this will pass the battery and we will be able to use it in our final design.

### **4.3.5 Power Outlet testing**

In the process of testing the power outlet it is important for us to follow the safety measures thoroughly. Working with high voltages (120v in this case) can be dangerous. Keeping that in mind the steps to test the outlet is as follows:

First, we make sure that the outlet is not being supplied with any current from the inverter. Then, we need to remove the outlet cover. Using a volt-meter we will place its probes on the terminals of the receptacle. A reading of about 110 - 120 volts should be seen. Then if that happens, we move on to check the ground. Using the same volt-meter we measure the ground voltage. If it reads zero volts then the power outlet is good to be used.

It is important that we conduct these tests to ensure correct electrical wiring from our part. We cannot risk the chance of connecting a load device to the outlet without testing it beforehand as this poses serious hazardous risks and could lead to damage of the

devices connected. If in the case that the outlet fails any of these tests we will move on to checking off every possibility of why that might have happened.

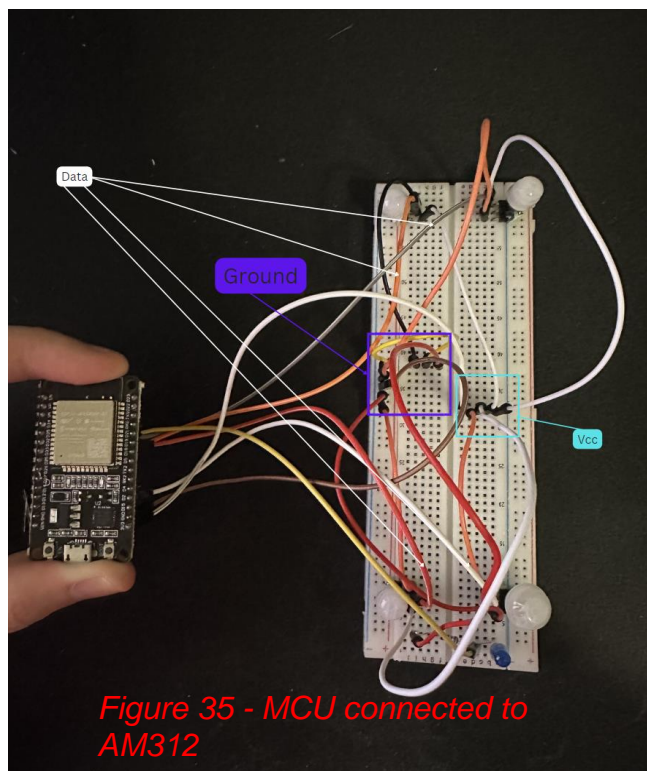
### 4.3.6 LED testing

The main areas in which we are testing our LEDs are their power consumption, connection with the MCU, and their programmability. To test the power consumption and MCU compatibility aspects of the LEDs, we are going to use a breadboard to connect the light strip with a 5VDC power supply. In the same breadboard, we are going to connect the MCU, and pair the LED data wire with the data pin of the MCU. To do this, we are going to need one resistor that separates the data wire from the MCU pin and that eliminates any interference that could be mixed with the data received by the MCU, and one capacitor to ensure that the power the LED is receiving is consistent, this capacitor will be connected to the GND and 5V pins of the breadboard. As for the LED strip programmability, we are going to test that via writing a code that executes some functions on the LED, like turning it on/off, and controlling the individual LED units.

### 4.3.7 Motion Sensor Testing

The motion sensor and LED functionality are a key feature of our picnic table. We want the table to be operable at all times of the day so it can be more accessible to anyone. To ensure this, the testing of this component is essential.

To carry out the testing procedure, we will be utilizing the ESP32 microcontroller, a breadboard, four AM312 PIR motion sensors, and a LED bulb. The testing will be conducted only on the four motion sensors to see if they will work with the LED bulb correctly. Since the maximum range of a single PIR motion sensor is up to 100 degrees, it is necessary to install four as it will allow the table for full coverage. For the motion sensors to be passed, they must first fulfill certain requirements. First, they should all be able to be powered by the ESP32. Second, they have to be able to properly turn the LED off and on. This means that the LED should remain on during the time someone is sitting on the table and turn off when no one is nearby or as the person is leaving. Finally, the four sensors together should be able to detect motion at all angles to ensure that there is a 360 degree coverage. The figure 35 shows the microcontroller and motion sensor pin connections.

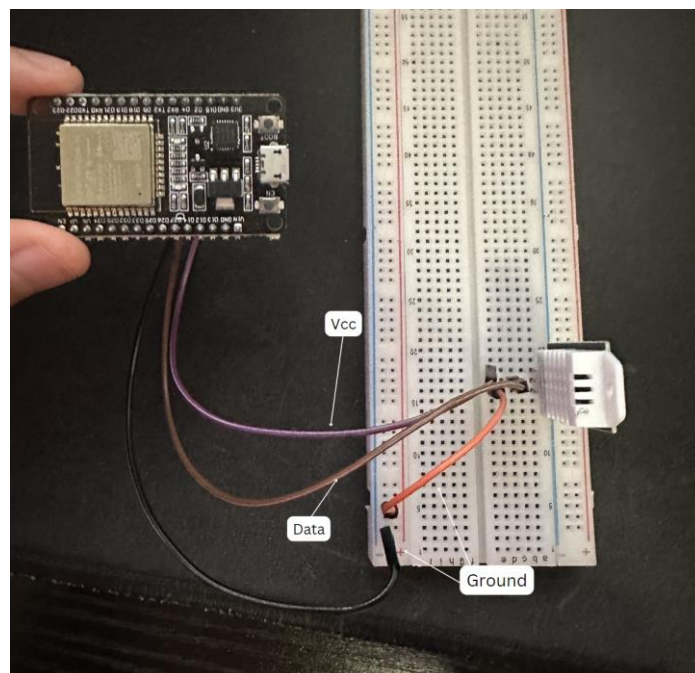


*Figure 35 - MCU connected to AM312*

## 4.3.8 Temperature and Humidity Sensor Testing

The temperature and humidity sensor is an important feature to our mobile app. When using the mobile app, the user should be able to see the climate of their surroundings. The DHT22 will allow us to provide this information to the user. However, to ensure that there are no problems and that everything is working correctly, we must first create a testing plan.

To carry out this plan, we will need a breadboard, the ESP32, and the DHT22. The microcontroller will be powered using a USB-C cable connected to a computer. Once all of the correct pins are seated correctly, we will upload the code using the Arduino IDE. The temperature and humidity readings will appear inside the serial monitor for us to check. Once everything is running correctly, we will simulate different types of environments to further test the DHT22. Figure 36 portrays the setup between the microcontroller and DHT22 through a breadboard.



*Figure 36 - MCU connected to DHT22*

For the environments, we will simulate four types: a hot environment, a cold environment, a dry environment, and a humid environment. For the hot environment, we will shine a lamp over the DHT22 to heat up the surrounding area. For the cold environment, we plan to place bags of ice nearby to lower the temperature. For a dry environment, we will place the DHT22 inside a container along with silica gel packets. These packets will absorb the surrounding moisture and keep the area dry. Finally, to

simulate a humid environment, we will use a humidifier to add moisture to the surrounding area.

The purpose of using these different environments is to ensure that the DHT22 can properly function in various types of environments. If it is able to provide accurate readings for all of the environments, then it can be used for our solar picnic table.

### **4.3.9 Battery Monitor Testing**

One important feature of the mobile app is the battery monitor. Users should be able to see the amount of power left inside of the battery. This will give the user an idea of whether or not to use the table for charging their devices. So, testing the monitor for accuracy is important for the reliability of the table.

To begin testing, we will first set up the voltage divider on a breadboard. Once this is done, we can connect a male to male jumper wire to R1 and another male to male jumper wire to the ground. Next, connect the R1 and R2 resistors to a ADC GPIO pin on the microcontroller. Once this step is completed, the setup is finished and we can start testing. To read in the voltage levels, connect the jumper wire connected to R1 to the positive terminal of the battery then connect the other jumper wire connected to ground to the negative terminal. The microcontroller should be able to read in these values and output them on the serial monitor. We will read the voltage levels using a digital multimeter and compare the two readings to see if they're the same. If both the readings from the microcontroller are similar to that of the multimeter, then the battery monitor is working as intended.

### **4.3.10 Bluetooth Connectivity Testing**

To test the bluetooth connectivity, we will need to test the connectivity with Android and iOS devices. Android is capable of connecting via Bluetooth Classic whereas iOS is unable to. With this in mind, we will be using Bluetooth LE for both platforms to help testing and keeping everything on the same platform.

When conducting the testing plan for the iOS connectivity, there are a few steps we must take. First, within the Arduino IDE, we have to use the BLE libraries to set up the connection. Next, since BLE devices aren't displayed in the default bluetooth interface for iOS devices, we will need a BLE connection app. This is strictly for testing purposes as we are only looking for a connection. The app that we will use for testing is NRF Connect. If the code is correct and uploaded, the ESP32 should appear in the menu and we should be able to test it. To further our testing, we want to make sure that data can be sent to the iOS device. To test this, we will use the DHT22 to read in temperature and humidity values. We will then send this data to the device where it will be outputted on the app.

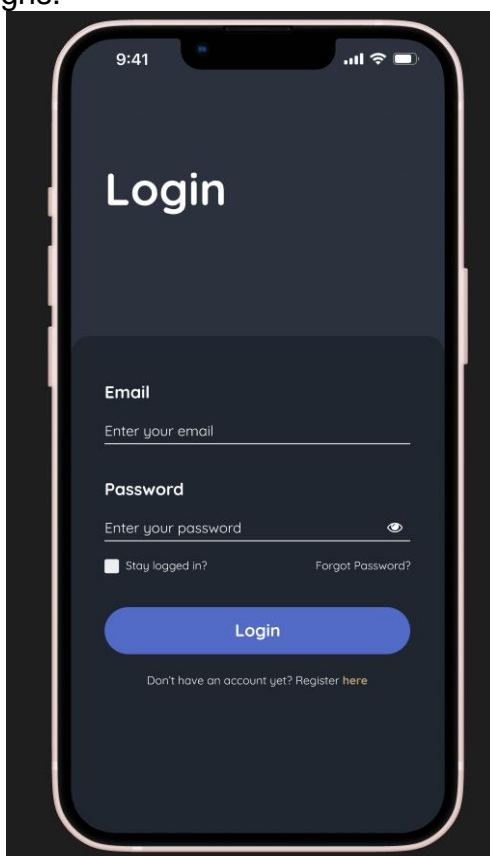
When creating a testing plan using the Android app, it should be fairly similar to the IOS version. Since we cannot directly connect an Android device to the ESP32 and

receive information, we need to go through a few steps first. First we will have to add a library for BLE on the Arduino IDE, once downloaded we add these files to the library file in the Arduino IDE software. Once implemented, we have to download an app that allows for bluetooth terminal access on our Android device.

### 4.3.11 Mobile App Testing

When creating the mobile app, testing is very important before deploying onto an actual phone. Without proper testing, mobile apps can cause serious issues that might lead to someone having to hard reset their phone. With the use of emulators, we can make sure the mobile app is fine tuned and works properly before deploying.

The first step in the mobile app testing is creating a UI prototype that is simple to use and displays the information nicely. We will be following basic UI design choices such as making sure everything is visible and contains contrasting colors. The UI has to follow accessibility rules to allow for anyone to use the website. This leads to simple but very effective UI designs.



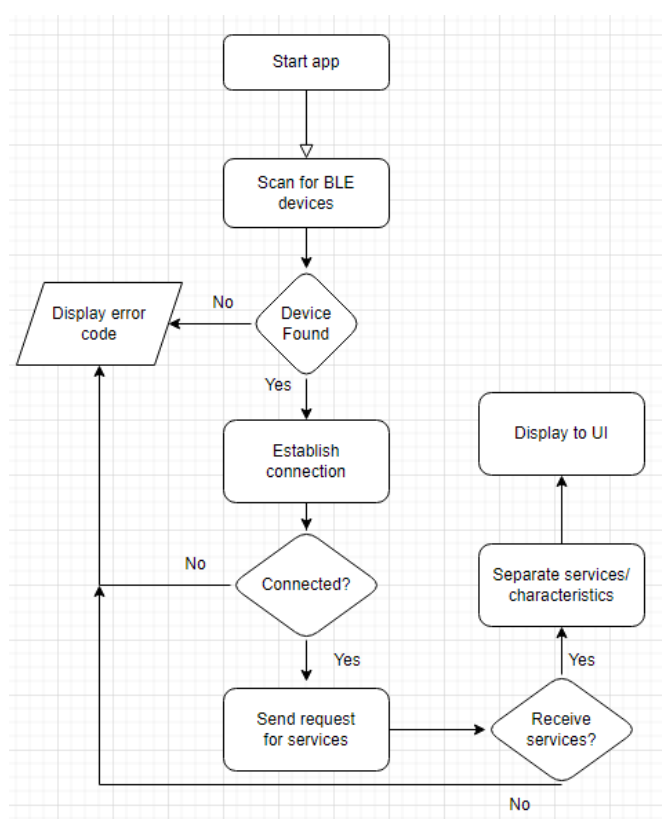
*Figure 37 - Example of a Mobile App UI in Figma*

After creating a UI prototype in Figma, we can transition into actually coding the UI through React Native. Once a simple UI is created, we can create simple functions that can act like the ESP32 sending information over to see if it is properly displaying. We will be using Expo, Android Studio, and Xcode to create Android or iPhone

emulators that are running the application and execute functions to send data over. Once the UI is functioning correctly we move onto connecting the React Native application to the ESP32 through bluetooth.

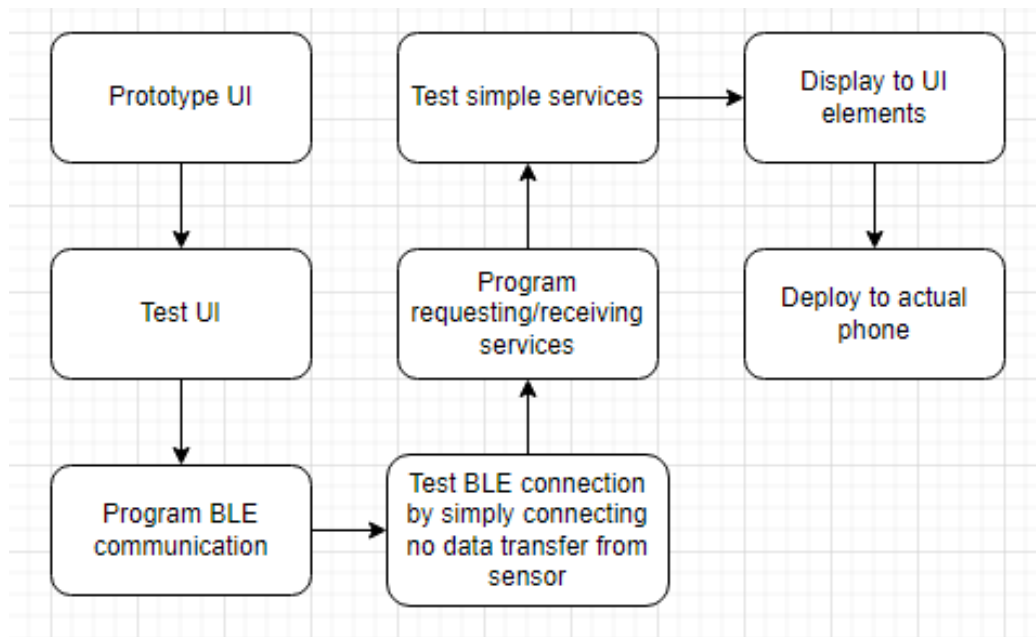
When testing for bluetooth connections on the React Native application, we will be making a test app that simply shows the information for connecting bluetooth devices. This test application will display what bluetooth devices are around our phone and allow us to establish a connection to other bluetooth devices such as our ESP32. Once this is done, we can test transmitting data through the COM terminal in the Arduino IDE software to see if that data is received on the app.

Testing to see if data is transferred between the bluetooth connection and the mobile app requires the application to discover the services and characteristics of the Bluetooth LE module. Since BLE sends data asynchronously, we will need to program the app to send requests for data from a sensor, and wait for the sensor to send the data back. Once that request system is done, we will have to write functions that read the service that is sent and separate the characteristics from it. Once that process of separating the characteristics from the service is done, we can display the characteristic value through the UI which are the main values the user cares about.



*Figure 38 - How the Application Interacts with ESP32*

Testing this data transfer is the hardest part of creating this application so we will be testing this with the sensor with the least amount of depends. In this case, we will be using the temperature sensor since we only need the sensor and the ESP32 unlike trying to test the battery indicator which uses many different sensors to work. Once the code is working for the temperature sensor, creating different functions for other sensors should be pretty similar.



*Figure 39 - App Development Process*

### 4.3.12 WaterProof Electronics Box

As indicated in Figure 44, the plan is to attach an electronics box to the underside of the table. The purpose of this box is to house several electronic components, including the battery, charge controller, inverter, and PCB with a microcontroller. This strategic placement ensures that the electronics remain protected from any external elements, particularly rain and sun.

To ensure that the electronics are not compromised, the box will be designed with a minimal number of openings. There will only be spaces for the wires and a ventilation window. These openings will be strategically placed at the side ends of the box, thus minimizing the risk of water entering the box. The ventilation window is a crucial component, as it will prevent the electronics from overheating.

Within the box, there will be separate compartments for each part, ensuring that they are held securely in place and preventing any tangling of the wires. The entire box will be 3-D printed, with approximate dimensions of 40x25x20 cm<sup>3</sup>. It will be designed to

have a similar appearance to that depicted in Figure 42, with the added bonus of being larger in size and containing separate compartments on the inside. The ventilation window can also be seen in Figure 42, highlighting its importance to the overall design.

In conclusion, the electronics box is an essential element of the table's design, ensuring that the electronic components remain protected from adverse weather conditions while also allowing them to function optimally.



*Figure 40 - Example of Electronics Box*

## 4.4 Prototype

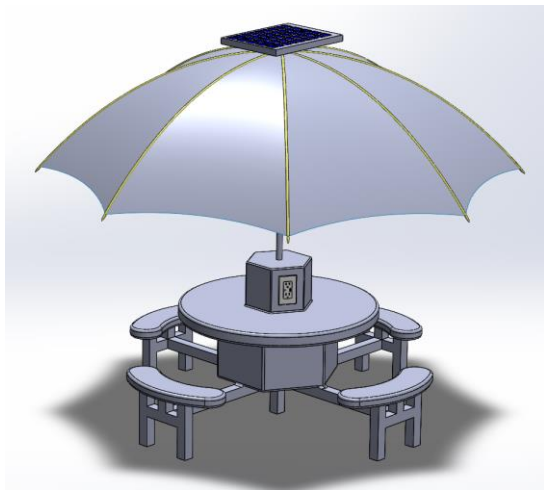
When creating the prototype of the solar picnic table, we wanted to use an existing table on campus as reference so we can utilize existing tables to prototype. The dimensions are based on the ones on campus.

The prototype of the solar picnic table contains three main components: the table itself, an electronics box, and the umbrella. The table has a seating capacity of four people but could allow for more people to squeeze in with the seats being more of a bench style seat instead of individual chairs which would limit seating capacity and increasing cost.

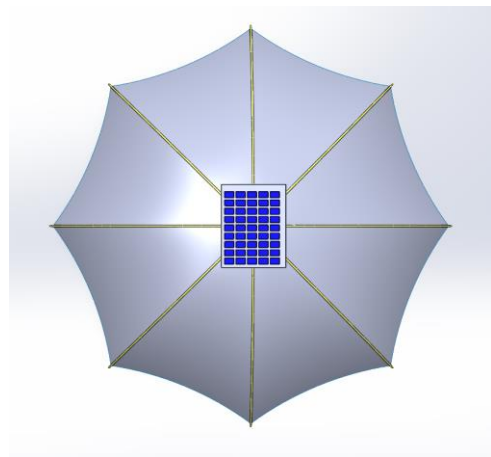
In the middle of this table is a small components box that will comprise of things the user will interface with such as the power outlet and simple instructions on how to obtain the app and safety instructions. Underneath the table will house the electronics

box which will contain all necessary parts of this project such as the batteries and controllers. Our goal is to incorporate the electronics box to maximize space for the users at the table and seal it off to prevent tampering and the elements. For the umbrella, it is a standard umbrella for shade and protection from rain.

On top of the umbrella, solar panels will be placed on different areas after further testing to see the most optimal placement for the most sun exposure. In the drawing below, we have only modeled one solar panel on top but that is only temporary to show the general idea and will be determined with further testing. On the underside of the umbrella, there will be LED lights. All the wiring is going to be run through the center tube of the umbrella and fed straight to the electronics box. The figures below are the 3D model of the general idea of the project.



*Figure 41 - ISO View of the Table*



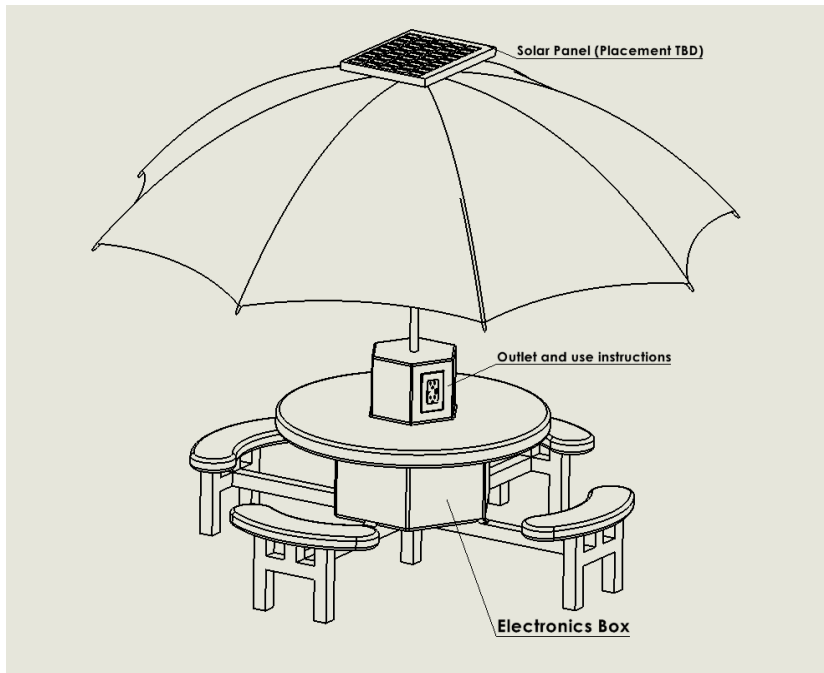
*Figure 42 - Top View of the Table*

The way the umbrella is shaped allows us plenty of room and flexibility for implementing the solar panel placement. With multiple places to put the panels and the ability to shift the angles of the panel allows for the maximum efficiency of our panels.



The underneath of the table may need more support with the additional weight of the electronics box which will contain heavy items such as the battery. The table has pegs as supports that may be a concern when placing the table elsewhere such as a park in the grass. These pegs may sink into the dirt and cause issues such as stability concerns.

*Figure 43- Bottom View of the Table*



*Figure 44 - Outline and Labeled view of the Table*

## **5. Constraints and Standards**

With every project, there are certain constraints that the groups must face. Along with this, there are standards that must be followed. It is important to acknowledge these to create a successful final product.

### **5.1 Constraints**

We will go over the various constraints that the team will have to face. We will also cover possible solutions to remedy these constraints.

#### **5.1.1 Economic**

When designing and creating this project, the cost was a huge restraint for us. We wanted to construct a solar picnic table that was functional but also affordable. However, since our project isn't sponsored, we would have to use our own money to fund this project. This would prove to be a huge economic restraint for the team. If we had a sponsor, we could purchase the best components to fulfill each of the table functionalities to the fullest potential. However, since this wasn't possible, we had to conduct extensive research to maintain a balance between cost and performance.

To combat this economic restraint, we have done research on multiple parts to see which one would suit our needs the best with the budget we have. Everywhere possible we are trying to cut costs and find parts that do not include features that we will not utilize. For example, when researching microcontrollers, we were initially looking at an Arduino that would need a bluetooth module to allow for connectivity with our mobile application. This would raise the cost of electronics however we found another microcontroller that has built-in bluetooth that was less than just the base Arduino.

Another way we can combat the funds restraint is to build some parts on our own. Instead of finding a waterproof box to house all our electronics, we are going to build our own. Since we are building our own waterproof box, we have to do extensive testing to ensure that our product does not fail. Buying a waterproof box would be a simple solution since it's been tested by the company who made it, however by creating our own, we can save funds but in exchange for that, we have to spend more time testing and prototyping. In the end, building our own components if possible will ensure our parts all fit together without having to modify pre bought parts.

Another solution to the economic restraint could be buying used parts. Buying used parts can pose a lot of risk such as parts malfunctioning or not being able to function as long as a part that is bought new. Since all our parts are going to be in the elements, buying new and reliable parts might be the only option to keep the table working for as long as possible and as reliable as possible.

Another economic restraint is the maintenance cost. If this product were to be created and made available to the masses, it would be important to keep maintenance costs low. We can do so by constructing the solar picnic table out of durable components. The most expensive and essential parts such as the solar panels and battery should have a large amount of budget allocated to them. This is to ensure that these two power sources are up to date and long lasting. In the end, using more durable components would cut down on maintenance costs tremendously.

## 5.1.2 Environmental

There are many constraints that the environment can pose. One constraint is the weather. There are many weather conditions that could prove to be dangerous to our solar picnic table. One example would be a hurricane. When dealing with a hurricane, there is torrential rain, heavy wind, and lightning. The combination of these elements can pose a lot of possible issues. The heavy winds could carry debris and hurl it at our table which can cause tremendous damage depending on the size of the debris. The rain can damage the electrical components such as the microcontroller or the sensors. In the unlucky scenario where a lightning strike hits our table, the whole table could be compromised. There are many more weather conditions that are dangerous. However, these conditions can be combated with the use of higher quality materials. These materials could have more durability and more resilience to the elements. However, for this project, our budget did not allow us to cater to these needs.

Another environmental constraint would be temperature. Depending on the temperature, there can be a wide variety of effects on the components. If the solar picnic table is exposed to extreme heat for too long, it can cause damage to many of the parts. The solar panels could overheat, this would decrease the overall efficiency of the solar panel which can affect the table as a whole when it comes to power delivery. The battery could also overheat. If any of the parts overheat, it would negatively affect their lifespan. On the other hand, if the environment is too cold, it could also cause issues. In low temperatures, the solar panels can lose their efficiency. This is also the case with the battery. Again, this is a serious problem as their reduced efficiency affects the whole picnic table since they are the two main power sources. In the events of extremely low temperature, the solar panel and battery can freeze over which can cause permanent damage.

Moisture is another environmental constraint. If water was able to infiltrate the table and get onto the electrical devices such as the microcontroller, it could destroy all of the table's software functionalities. On top of this, moisture can lead to corrosion and rust on any sort of metal components.

For solar panels, sunlight exposure is the most essential factor when it comes to generating power. This is why sunlight exposure and shading is an important environmental constraint to consider. If we choose an area where there are many trees or large structures, the solar panels could be covered by shade. This is problematic since it has to power the electrical components and charge the battery during the day. If

the solar panels aren't generating as much power, then there could be an insufficient amount of battery charge to last throughout the night. When deciding where to place our solar picnic table, we must consider a location with the most sunlight exposure and as little shade as possible. That way the solar panel can perform at maximum efficiency and all of the components can operate at full functionality.

These constraints, with the exception of the sunlight exposure and shade, can be solved by using more durable and weather resistant components. However, this would mean that we would need a larger budget to fund these features.

### **5.1.3 Social**

A social constraint of the solar picnic table would be its accessibility to the public. For example, if the table cost is too high, it would only be used by people who are able to afford it. This would limit its availability to everyone and only a select few can benefit from the table. Another accessibility issue would be the physical space that the picnic table can provide. It should be able to accommodate large groups of people so everyone can enjoy it. Finally, it should be accessible to the disabled. The disabled should be able to access and use the table and its features with ease. These constraints can be solved with a bigger budget, the physical table could be made with some modifications to fit more people or to allow people with mobility devices to fit or sit at the table comfortably.

### **5.1.4 Political**

After some research, a political constraint that the team realized is the possible regulations that could be placed upon the solar picnic table. These regulations could be a variety of things such as environmental laws or safety standards.

### **5.1.5 Ethical**

When dealing with ethical constraints, the solar picnic table could have detrimental effects on animals and their habitats. One constraint would be the effects on their habitat. During the installation of a single or even multiple solar picnic tables, it could require clearing an area for space. Clearing space can mean deforesting an area or destroying the surrounding natural environment. This could lead to the alteration or even the destruction of some animal's homes. On top of this, with the introduction of human activity, it could disrupt the wildlife. With human activity, this can lead to waste and litter. If trash isn't handled properly, it can attract dangerous animals which would threaten safety. To avoid these constraints, careful planning would have to be made to ensure that no natural fauna or wildlife is affected by the solar picnic tables and trash cans should be made available for correct waste disposal.

## 5.1.6 Health and Safety

The safety and reliability of our project is our number one priority. We have taken all measures in the designing process to reduce all possible risks and hazards that the project may pose. Still, there are many outside factors that are out of our control and may lead to some possible health and safety concerns.

The first thing that comes to mind when we talk about safety in our project is the battery storage system. As we know the battery stores high amounts of energy in them, and devices of high energy always come with an amount of risk and possible hazardous scenarios. Although the battery in our project will be contained inside a waterproof box this does not totally eliminate all the possible dangers. The battery could overheat, especially that it is outdoors. Some battery systems require a cooling system but unfortunately this is not possible for ours. We could implement a small cooling system that runs a fan that circulates air in intervals especially during times of high usage, however due to budget constraints, implementing a high end cooling system is not a possibility. There is also a possibility of it having some manufacturing defects and this could contribute or lead to overheating. Although the battery technology is advanced today, they are still flammable.

Another constraint regarding the safety of batteries is that our battery technology of choice for the customer's final product is Lithium-ion battery. These batteries have multiple risk factors and it is often necessary to include extra safety features to it. Some of these risks include short circuits which is when the positive and negative terminals touch and produce sparks that cause overheating or even catching on fire. Other risks in these batteries are the possibility of overcharging which could potentially cause "Thermal runaway". Thermal runaway is when heat coupled with pressure due to overcharging cause the battery to chemically react and will eventually lead to an explosion. Due to the possibility of these problems, extra safety measures like having a battery management system. BMS is responsible for controlling the process of charging and discharging of the battery. This helps in significantly reducing the risks listed above. Unfortunately it is unrealistic to implement a BMS in all of our solar table products.

In our system we have other electrical components that pose safety concerns in our system. Components such as the solar panel, inverter, and the power outlet. Solar panels generate electric current when exposed to sunlight and this makes the electric system in risk of possible fires, if any faults occur in the circuitry of the panel. Also, when dealing with the DC to AC inverter and the power outlets we are dealing with high voltage devices. High voltages are always a risk and pose electric and fire hazards. Proper safety procedures must be followed in the installation process.

When designing a product for customers that include such risks and possible hazardous situations we must also include a safety instructions sheet on how to safely use the system. It should also include instructions on what to not do. Things like

opening the battery's case without disconnecting it first or inserting metal objects in the outlet should be mentioned.

## 5.1.7 Manufacturability

A major manufacturability constraint for this project will be the availability of parts. Depending on the demand on the market, the supply will be heavily affected. If there is not a huge demand for the components that we are looking for, the supply can be very low. This can arise as an issue when it comes to manufacturing the table as there could be a shortage. However, if the demand for the components is extremely high, then it could also be a problem. If the supply can't keep up with the demand, then it would drive up the prices and cause cost issues. This is a worst case scenario as one of the main objectives of our solar picnic table is to make it as cost effective as possible. There isn't much we can do to combat this issue as we aren't able to directly affect the supply and demand of the products.

Another manufacturability constraint is the material selection. When selecting materials, it is good practice to pick out the most optimal components. This means picking a durable material for the physical table to be made of, a solar panel that is capable of both solar power output and outstanding power efficiency, a battery that can hold enough charge to last throughout the night, and a microcontroller that is able to support all the functionalities of the table such as battery monitoring. On top of this, the table has to be strong enough to support the weight of all these devices and at the same time, these devices have to also be durable and weather resistant to ensure longevity. The material selection for all of these components should be carefully planned and selected to ensure a balance between durability, power, and cost.

Quality control could also be seen as a manufacturability constraint. To ensure that a solar picnic table is working correctly and made for public use, it is paramount that all of the devices are working as intended. Without any quality control, it could cause some serious issues and safety hazards. The solar panels could be faulty and not work properly which could lead to the whole picnic table not even working since it can draw any power. If the solar panels aren't installed and racked on correctly or securely, they could fall off and hit someone which is a huge safety issue. Another example could be the battery, if the battery is faulty, it could have a wide range of issues. It won't be able to hold any charge and can't power any of the devices during the night time or worst case scenario, it could blow up or cause a fire. This is why quality control is important to undergo so that these issues can be found and resolved. Even if it would take more time for the manufacturing of the product, safety should be a number one priority for any public product.

The assembly of the solar picnic table can be a manufacturability constraint. If the installation and assembly of the solar picnic table is made too complicated, it could deter manufacturers from wanting to work with this product. This is because it would cost more money and time. If we wanted our solar picnic table to be widely available to everyone, then it has to be easy to manufacture.

## 5.1.8 Sustainability

One sustainability constraint that we found was the energy production of the solar picnic table. The solar panels will be able to produce energy through the sun. It is important to realize how much energy will be needed to charge and power every component on the solar picnic table. This can range from the microcontroller to the electronic devices being charged by the outlets. It is important to calculate all of these factors beforehand so enough power can be stored and generated.

Another sustainability constraint would be the material durability. If the solar picnic table is made of cheap and low quality components, then the table wouldn't be able to last very long. It would be susceptible to outside forces such as weather conditions, or general wear and tear. This is why it is important to procure quality components to ensure the durability and longevity of the table and its components. This in the long run will help with creating a sustainable solar picnic table.

Upkeep is also another constraint. If the solar picnic tables were left alone without any sort of regular maintenance, they would become more damaged over time. This can pose a serious safety hazard. For example, if the solar panels become loose over time, they can fall off and hit someone. This is a scenario that could be easily avoided through regular maintenance.

## 5.2 Standards

Here we will cover the different design standards that can affect our project. These standards can help us achieve our project goals. For example, one of our project goals is to make the solar picnic table as safe as possible. With so many electric components, there are many cases where injury could happen from a faulty battery or the solar panels malfunctioning. Following these standards could prevent a situation like that from happening. Another project goal of ours is to maintain efficiency. Since power generation is a huge aspect to our project, finding ways to use it more efficiently would be a tremendous help. For example, some standards could assist us in our coding in making it optimal. This in turn would cause less power consumption by the microcontroller and ultimately lower the overall power consumption of the solar picnic table. As you can see, following these standards will help us fulfill our project goals. Not only this, but it will make our product more familiar to other companies that also follow these standards

### 5.2.1 UL 1703 Standard for Flat-Plate Photovoltaic Modules and Panels

This standard is established by the Underwriter Laboratories, which is a safety organization responsible for testing and certifying new products. This third-party organization has been around since 1894. As for their UL 1703 standard, it is specific

to solar panels. To get a solar panel up to this standard, it has to go through a fire testing procedure, both with flames directed at the panel, and with a burning object on top of it.

Since we are placing our solar panel on top of our table, and that this project is intended to be used in picnics, there is always a chance of it encountering flames or burning objects, not to mention that it will be placed outdoors and the chance of a lightning strike hitting the panel is always there. Therefore, getting a UL 1703 certified solar panel is important, considering that the solar panel is the core that provides energy to power all other components in the table. One of our project goals is to create a solar picnic that is not only cost effective but also safe. So, following these standards will ensure product safety and prevent any sort of injury.

## **5.2.2 IEC 60038 Standard Voltages**

The IEC, or International Electrotechnical Commission is an organization that is responsible for establishing international standards related to what is known as electrotechnology, which is a term that includes electrical devices, electronic devices, and any device or technology in relation to that. The IEC 60038 standard clarifies a set of voltages that are standard to high voltage, low voltage, and extra low voltage AC and DC systems. The IEC defines boundaries to classify these systems, and the risks involved in each of them.

Adhering to this international standard can have a positive impact because by following these standard voltages, especially in our power outlet, we are ensuring that we are building a safe project, as these standard voltages are tested and proven to be the best for both use, and device safety. Furthermore, using international standard voltages will make our project suitable for more people, increasing its chance of success.

## **5.2.3 ISO/IEC 9899 Programming Language C**

This standard was created by the ISO (International Organization for Standardization) and the IEC (International Electrotechnical Commission). It highlights the rules and guidelines when working with the programming language C. This standard is set to be international so it is widely used. It goes over the various aspects of the programming language C such as the representation of the C programs, the syntax and constraints, the semantic rules for interpreting the language, the representation of input data that is processed by C programs, the representation of output data that is produced by the C programs, and finally, the restrictions and limits of the implementation of C.

One important aspect of the ISO/IEC 9899 is that it goes over the various definitions of certain terms and elements of the C language. For example, the standard defines a wide variety of data types such as characters and integers. This is important as it informs the user of the different data types and what their purpose is along with their attributes like size.

Another important aspect that the ISO/IEC 9899 covers is the memory management and allocation. When it comes to coding, memory plays a huge role as it is used to store and manipulate data. With this guideline, it can guide programmers to properly use their memory and prevent any errors or confusion from happening.

Overall, this standard is important as it provides a set of rules and guidelines for a rather commonly used language. The C programming language is commonly used in many different applications. So, by following this standard, our code can be understood by many other places as they will also follow similar guidelines when it comes to C.

## **5.2.4 Impact of ISO/IEC 9899 Programming Languages C**

The ISO/IEC 9899 will have a major influence on how we approach our code. The ESP32 microcontroller is capable of supporting the C language. The microcontroller will be integral for any software functionalities that our solar picnic table has to offer, this includes the mobile app connection, motion sensing, temperature sensing, battery monitoring, and LED control. By using these standard's guidelines and rules, we are able to code the microcontroller properly and ensure that it works

Along with this, the standard also ensures that the code will be reliable and efficient. An important aspect of the ISO/IEC 9899 was the variable and function declaration. When it comes to coding, the variables and functions are arguably the most important part. For example, variable declaration allows programmers to create memory space for storing any sort of data or values. On top of this, it enables the manipulation of this data so we can perform complex calculations. Functions are equally as important since they can be called. This means that code can be used whenever needed. So, having rules and guidelines to properly implement these two elements is paramount in having reliable and efficient code.

Another way that this standard can be of use is in its error handling. This aspect of coding is also important for our solar picnic table. In the case of an error, we will have to find a way to handle it so it can be contained. If the error is not handled, it could compromise the whole system. This is why it is important to handle any error so this can be prevented. The standard can aid us since we can follow its procedures and we can achieve effective error handling.

Another way that this standard can be used is for handling legacy code. With a set standard, it guarantees that older versions of our code are supported as more updates to the language are applied. This is important because if any future updates happen in the C language, code that worked before should remain working without having to reprogram everything. Overall the general structure and components of C have remained the same. Newer versions of C have fixed existing issues and do not change

the structure of the language allowing us to run code written now for many years to come.

Overall, there are many benefits that we can gain for using the ISO/IEC 9899 standard. The different aspects that it covers and the rules and guidelines that it provides will be a tremendous help when we begin programming our microcontroller.

## **5.2.5 IEEE 1562-2017 Recommended Practice for installation of Lead Acid batteries in PV systems**

This is a standard approved by IEEE in 2017. The standard explains design considerations for installation, storing, location, and assembly of Lead Acid batteries in solar PV systems. Additionally, this document covers information about measuring the voltage, current, and state of charge of the battery system.

Following this standard is important because when handling battery storage systems like a Lead Acid battery it is important to be cautious. There are many safety risks involved in the process of the installation of these batteries, and by following the standard we are not only minimizing the risk of installation and assembly, we are also ensuring the maximum safety of all future users of our system.

Also, the information in the document about the measuring of the battery parameters is crucial. There are specific ways to measure these parameters and if they were not followed many possible hazardous situations could occur. For example, it is important to know the ways in which you could accidentally short circuit the battery or cause it to overheat while in the process of measuring the current. Following the information provided by the standard ensures that we do not make those mistakes.

## **5.2.6 National Electrical Code (NEC):**

The National Electrical Code is a standard published by the National Fire Protection Association (NFPA). It covers topics around battery installation protocols and that includes wiring, grounding, ventilation, and has information about battery labeling and Marking.

This standard was published to create safe guidelines for people using battery systems to ensure safe installations and wirings. It is followed and enforced by lots of local authorities having jurisdiction.

Properly following these guidelines will ensure safe installations from wiring, locating, ventilating, or even testing the battery system. Not following these guidelines puts the worker and everybody else around the system at risk because for example any issue of not having proper ventilation could result in overheating or fires.

Following these standards are also very crucial for us in the project. Having to work with lead acid batteries for the first time and having to install and assemble them is not an easy task. By following the guidelines provided we will be able to safely do it without the fear of something going wrong.

Our project will be outdoors so this means that our battery will be exposed to high temperatures and possibly rain. We need to look out for these specific installation guidelines to ensure that no short circuiting or over heating occurs. Proper ventilation of the water-proof box is important too.

## **6. Administration**

In this chapter, we will be going over important aspects for proper project progress. We will be detailing the different milestones that we want to achieve, our budget and finance, our role assignment, and member contribution.

## 6.1 Milestone Discussion

For any project, planning and setting certain goals to accomplish is essential. This enables us to split up the project into more manageable parts and gradually build upon these parts until we come up with the end product. In this section, we will go over the different milestones that we have setup to meet our end goal.

### 6.1.1 Senior Design

In order to complete and turn in the senior design I and II due dates correctly and on time, we must keep track of when they're due. By knowing this, we can plan ahead accordingly to complete each task properly and efficiently. Below are the two tables that we created to highlight the different tasks we must complete. Currently, we haven't taken senior design II so the due dates are unknown. However, we will still work on the project over the summer in preparation for fall.

Tasks	Start	End
<u>Senior Design 1</u>		
Project Selection	1/13	1/24
Divide & Conquer I	1/24	2/2
Group Meeting	2/6	2/8
60 page draft	2/8	3/24
Group meeting	3/29	3/29
120 page final	3/29	4/25

*Table 25 - Senior design 1 milestones*

Tasks	Start	End
<u>Senior Design 2</u>		
Design	TBD	TBD
Ordering parts	TBD	TBD
Prototype	TBD	TBD

Testing	TBD	TBD
Finalizing prototype	TBD	TBD
Present project	TBD	TBD

*Table 26 - Senior Design 2 Milestones*

## 6.2 Product Milestones

The senior design due dates are not the only milestones that we must take into account. In order for the completion of our solar picnic table to go smoothly, we must approach this project with project decomposition. Each major component of this project can be split up into sub projects that will help guide us into the right direction. Splitting up the project into more manageable chunks also helps improve our testing through this project and ensures that everything functions together. We can split our testing by doing unit testing and integration testing similar to how applications are tested. Unit testing is when we test a subsection of the project, for example, if we are doing a unit test on the ESP32, we would connect individual sensors and test them one by one. Integration testing is when we implement more parts into the testing so for the ESP32 example, we would connect all the sensors we are using to ensure everything is working correctly and not interfering with anything else.

### 6.1.1 Table Decision Milestone

The first milestone of this project before we reach any other milestones is first deciding on a general design of a table and how to work around that table. Since we do not have the budget to buy a table or buy the material to make our table, we will be using a table available to us on campus to do further testing. We have a few tables available to us, so we have to discuss which table offers the best platform for us to build upon it. Having a lot of space available under the table is essential since we need to fit a lot of components for our project underneath the table. Picking a table that is always exposed to the sun is also very important to maximize the amount of energy the table will produce. Once a table has been picked, we will be taking measurements to use as reference. Once that is all done, the first major milestone is complete and almost all the other components are reliant on this reference.

### 6.1.2 Milestone for Electrical Components

For our electrical components, the first objective was to figure out what components we are going to need to ensure the successful functioning of our product. We had to carefully analyze the specifications and requirements of our project and determine which components would be necessary to achieve our desired outcome. The next step was to consider how we would like to connect these components. This involved

examining various wiring and connection options and selecting the most suitable approach for our device.

After determining what components we needed and how they would be connected, we needed to calculate the minimum rating requirements of each component. This step was crucial to avoid overspending on components that had higher ratings than we needed. We had to carefully analyze the electrical characteristics of each component and perform calculations to determine the minimum ratings required for proper functioning.

Once we had figured out what components we needed, how they were going to be connected, and their minimum rating requirements, our next objective was to conduct research. This was a time-consuming but essential process as it involved comparing various components to determine the best options for our final product. We had to carefully examine the features, specifications, and cost of each component to ensure that we selected the most appropriate ones for our device.

Up to this point, we had already submitted a 60-page draft of the report. However, for the final draft, our goal was to figure out how the hardware design would look like, from the electrical wirings to the I/O ports to the PCB design. We had to create a detailed plan for the electrical layout of our device, including the placement and orientation of each component, to ensure efficient functioning.

Finally, after figuring out the hardware design, we made a bill of materials table listing all of the electrical components that needed to be purchased. This allowed us to keep track of all the necessary components and ensure that we had everything we needed before starting the actual construction of our device. Overall, the process of selecting and designing our electrical components was a meticulous and time-consuming process, but it was essential to ensure the successful functioning of our final product.

### **6.1.3 Milestone for Microcontroller and Sensors**

When it comes to milestones for the microcontroller, there are a few stages involved. The first step when using the ESP32 and all the sensors is to test them all individually and write code that allows us to utilize the sensors. Once the testing is complete and we ensure that all the sensors work, our first milestone is done. Doing individual testing gives us reassurance that they work and gives us knowledge with how each individual sensor works so creating an integration test for all the sensors would be easier. Since some of the sensors rely on battery and solar panel testing to be done first such as the battery monitoring system, we will test the sensors with similar technology as a temporary solution while we wait for more components to be tested. The second major milestone for the microcontroller would be getting everything connected and working. This means we have all sensors connected simultaneously and all reading values at the same time. Once that is done, the majority of the ESP32 is finished and the next milestone will be mounting and placing these components on the table.

## **6.1.4 Milestone for Mobile Application**

When it comes to the mobile app, there are a few milestones to complete before we can have a finished product. The first milestone is deciding the design for the mobile application, so this includes details such as design language, color scheme, UI design, etc. Since the mobile application is going to be responsible for reading the majority of the values received from the ESP32, we have to decide what values are going to be available to the public and how to convert those values from what the sensor reads to values the user can understand. Once the general design is figured out, creating a prototype UI using Figma is important to have a guide when coding the UI. Once a prototype is finalized one of the first major milestones are complete for the mobile app. The second milestone is creating a test app that does not have a complete UI but allows for the user to connect to the microcontroller and read values from it. This allows us to test connections and see if they are reliable before creating a finished UI for users. Once the second milestone is complete, the last and final milestone is going to be creating a finished UI that displays all the information we need to in a clean and readable manner. This finished UI will be deployed to the users after different tests are done to see if all values are being displayed in an understandable value.

## **6.1.5 End Product Milestone**

Once we have achieved all of the above milestones, the final goal that we will achieve is the end product. This end product would include the picnic table that is equipped with all of our components. For this milestone to be achieved, there are a few aspects that must work correctly. First, we must ensure that the power generation and distribution works properly. The solar panel must be properly secured onto the top section of the table, allowing for it to take in the solar energy. Along with this, this energy must be able to charge the battery. The battery must then be able to connect to the power outlets and the PCB. Once the power distribution and generation aspect is completed, we can move onto the second part.

For this part, we must ensure that the microcontroller is working properly with the other components and it is able to fit within the waterproof box. To do this, all of the 4 PIR motion sensors, the temperature sensor, and the LEDs must be connected to the microcontroller at the same time and placed within the box. From here, all of these components must be interacting with the microcontroller and each other properly. Next, the user should be able to connect to the mobile app using the bluetooth connection and read all of the information that is being outputted by the microcontroller. Along with this, the various power outlets should be fully functioning and be able to charge an electrical device. With all of these aspects working as intended, then the end vision of our solar picnic table would be achieved.

## **6.3 Budget and Finance Discussion**

When it comes to this project, there are a lot of components needed for this to function the way we envisioned it. With all these components comes a cost to each part. Throughout the planning and testing of this project, we have tried our best to minimize cost in as many places as we could. There are a few components that the budget leans towards the most and that would be the battery and solar panels. Since we want to have the best and efficient way to gather and store electricity, the majority of our budget will be spent on those components.

One of the biggest ways we cut our budget down is to source a table instead of building our own table. In the very beginning of part of planning this project, we were deciding on if we needed to build our own table or source the table through somewhere else. There are a few benefits of creating our own table which include creating a table that is tailored to our project and we can add any other details we want such as if we wanted individual seating or bench style seating. This will make our design process flow easier and faster since everything will be built upon each other. However after more research, we realized that creating a table will result in more issues down the line. If we went down the route of creating our own table, we would have to source out wood and hardware to build our own table. This would result in a cost double what we have estimated now. Even if we had more of a budget to buy the material, we would have to source out tools such as wood saws, drills, etc to actually build our table. In the end, we decided that finding a table solves a lot of issues but just requires us to plan ahead more for this project. Finding a table solves the budget issue which is one of the major components holding this project back from being able to use the best and most efficient components for this project. Going down this route also saves us a lot of time to focus elsewhere since the process of planning and building our own table from scratch will require a lot of time and prototyping if we want the table to be as nice as the ones available to us on campus. Sourcing a premade table might lead to us having to plan more accordingly but the benefits of saving on our budget outweighs that issue a lot more.

When it came to sourcing components for the microcontroller, it was fairly easy to decide. Since the microcontrollers we were looking at were not anything crazy in terms of performance, the majority of the choices were relatively cheap. After doing research on different microcontrollers, we decided on the ESP32. Sourcing the components for the ESP32 were relatively easy and we found very affordable options that give us multiple microcontrollers which allows us to have a backup plan if something were to happen to our microcontrollers. The sensors were also relatively cheap so all the microcontrollers and sensors were one of the least impactful to our budget. Majority of the testing tools needed for the microcontroller are offered to us on campus or we already have these tools from previous classes that involve prototyping with a breadboard.

When it comes to the mobile application, it does not affect our budget at all. All the platforms needed to create the app are free to the public. Since we decided to go the bluetooth route in terms of connecting our mobile app to the microcontroller, we do not need to depend on a hosting platform to host a web server for our microcontroller to connect to. If we decided to go to a hosting platform for our microcontroller to connect

to, it would not be that impactful for our budget. Most of the services available offer monthly payment plans so we would have to start that plan in the beginning of our testing phase. There will be no end to the hosting service since if we want to continue using this table in the future, it will be reliant on this service. All the testing tools required for the mobile app are free however there is a caveat when it comes to iOS emulators. Since we need a MacOS device to install an iOS emulator, we can always resort to a virtual machine or ask other people to borrow a MacOS device.

When dealing with the most expensive parts of our project, a lot of research is involved to make sure we purchase the best performing component for the budget we have. The components most impactful to our budget are the battery and solar panels. A lot of research went into picking the best components from reading technical specifications provided from the manufacturer to reading online reviews from people who have actually purchased the component we are looking at. Since the solar panel and battery are the most important components when it comes to this project, we have to make the best decision when selecting which component to buy. We are allocating the majority of our budget to these two components since if we get subpar components, the project will not function that well and everything else we have bought will be a waste. For example, if we get subpar solar panels and batteries, no matter how much we spend on a microcontroller, the table will still not output enough power for simple things such as charging a phone or powering LED lights.

Another way we are trying to conserve our budget is trying to design components by ourselves if possible. For example, the majority of the housing components for our electronics are going to be modeled and 3D printed ourselves to save budget from buying third party housings. When designing mounting components for lighter components such as LED lighting or the motion sensor

When it comes to funding this project, we are going to pay for all the components needed out of pocket. As the project comes together, we will look into finding sponsors willing to provide components we need or provide any funding we need to further improve our project. Since we are paying for components out of pocket, we want to make sure we obtain the best performing components for the money.

<b>Quantity</b>	<b>Item</b>	<b>Vendor</b>	<b>Model</b>	<b>Price \$</b>
1	Solar Panel	Amazon	NPA100M	75

1	Battery	ExpertPower	EXP1280	24
1	Inverter	Amazon	B0774SCCND	15
1	Charge Controller	Amazon	B08NFSCZ4V	10
1	LED	Amazon	WS2812B	16
1	Microcontroller	Digi-Key	ESP32-WROOM-32E-N16E-N16	3.60
1	PCB	Digi-Key	-	17
1	Receptacle Outlet	Amazon	UR2-15TR	13
2	Temperature Sensor	Amazon	DHT22	2*5
4	Motion Sensor	Amazon	AM312	4*1.78
<b>TOTAL</b>				191

*Table 27 - BOM Estimate*

## **6.4 Role Assignment**

For our project, we have two electrical engineers and two computer engineers, so we recognized the importance of distributing roles fairly among ourselves to ensure that each team member had an equal amount of responsibility and contribution to the project. To achieve this, our approach was to identify the tasks, categorize them, and then distribute them between us so that each one of us has an equal amount of responsibility and contribution in the project.

### **6.4.1 Project Manager**

As project manager, the purpose of this role is to oversee the direction of the project. This responsibility comes with many responsibilities. First, as project manager, one must delegate roles to the other group members. This will ensure that everyone has some sort of responsibility and everyone can contribute equally. Next, they must be able to plan for the project. This means that they can create a schedule for group meetings and set deadlines to achieve certain goals of the project. They must also ensure that the project is completed in time, within the budget, and on time. For this role, the group decided that Faisal Al Ghafri would be the best fit.

### **6.4.2 Electrical Designer**

This role included foreseeing the part research, comparison, calculation, and selection of the electrical components. The assigned member also was responsible for the design of the system, and that includes making the schematics, flowcharts, PCB design, electrical wiring design, and much more. Two of the group members were responsible for this role assignment. As the electrical engineers of the group Azzan Al Busaidi and Faisal Al Ghafri took on this role. Both members contributed to each type of task. Both contributed in the design, the research, selection, calculation of the electrical system.

Despite the complexity of the task, Azzan and Faisal were able to work together effectively to ensure that all aspects of the electrical component selection and system design were completed to a high standard.

### **6.4.3 Software Developer**

Another important role that is required for this project was the software developer. This role encompasses any element of the solar picnic table that required some sort of software implementation. This means that the member(s) assigned to this role will be responsible for the following: mobile app, the code for the sensors, bluetooth connection, the microcontroller, and LED control. The software developer(s) will be responsible for all of the following and ensure that they are working properly.

Considering the background of the project team, the two computer engineers, Thach Vo, and Amos Luo were given these roles as they had a background with software development.

## **6.4.5 Hardware Designer (Microcontroller and Sensors)**

For this role, the responsibilities included wiring the microcontroller to the different sensors. In the end product, the microcontroller will be built into the PCB so it is important that the person understands how to wire the sensors and microcontroller properly. The computer engineers, Thach Vo and Amos Luo had a background in this sort of subject due to previous classes so they were delegated with this role.

## **6.4.6 Documentation**

In this role, the primary responsibility is to document and write anything important that happens. This includes testing results, group meeting information, or reports. Since this role is very general in its tasks, everyone in the group decided that we should all take upon this role. That way, any vital information that is said or that occurs isn't missed and is properly recorded and documented for future use.

## **6.5 Task Distribution**

The task distribution was carried out with great care and attention to detail. We wanted to make sure that each team member was assigned tasks that aligned with their expertise and would allow them to contribute meaningfully to the project. Therefore, we divided the tasks into three categories based on their relevance to either electrical engineering, computer engineering, or non-technical work. The electrical engineering students took on tasks that were more relevant to their field of study, while the computer engineering students tackled tasks that were more aligned with their area of expertise. The remaining tasks that were neither electrical nor computer engineering-related were split equally among the four of us.

Once we had successfully distributed the tasks, we began the research phase. During this phase, we closely monitored our progress and assessed whether any changes needed to be made to Our team was able to make significant progress throughout the semester, thanks to our well-planned task distribution and effective communication. The tasks were completed on time, and everyone contributed their fair share towards the project.

One of the most significant advantages of having a diverse team with different skill sets and backgrounds was the unique perspective each team member brought to the table. This allowed us to approach the project from different angles, resulting in a more comprehensive and well-rounded final product.

As we progressed through the project, we encountered a few roadblocks that we needed to overcome. However, our team's flexibility and willingness to adapt to changing circumstances helped us navigate these challenges successfully. We learned to work collaboratively and support each other throughout the process, which allowed us to come up with creative solutions to the problems we faced.

The following table lists the tasks more specifically and shows who was responsible for them.

<b>Task</b>	<b>Name</b>
Solar Panel (Research)	Faisal
Battery Storage System (Research)	Azzan
Charge Controller (Research)	Faisal
Inverter (Research)	Azzan
Power Outlet (Research)	Azzan
PCB (Design)	Azzan & Faisal
LED	Faisal
Electrical Wiring (Design)	Azzan
Table Prototype	Amos
System Performance Simulation	Azzan
Market Comparison	Faisal
Equipment Parameters Calculations	Azzan
Testing Plan	Everyone
Constraints & Standards	Everyone
House of Quality	Faisal
Microcontroller (Research)	Thach
Motion Sensor (Research)	Thach
Battery Monitoring (Research)	Thach
Temperature Sensing (Research)	Thach

Mobile App for iOS	Thach
Bluetooth testing for iOS	Thach
Mobile App for Android	Amos
Bluetooth testing for Android	Amos
Mobile App testing	Amos

*Table 28 - Task Distribution*

## **7. Conclusion**

For our senior design project, we wanted to utilize renewable energy and improve the quality of life for the outdoors. To incorporate these two elements into our project, our team decided to create a solar picnic table.

For the initial start of the project, the team discussed the overall objectives and goals for the solar picnic table. We decided that we wanted the table to be eco-friendly and a sustainable solution for outdoor activities. Once we decided on this, we moved on to mapping out a general overview of what the solar picnic table needed and our budget. From here, the next step was picking what components were needed for the table.

When picking parts for this project, we had many criteria that the part had to fulfill in order to be a perfect fit. It had to have low power consumption, be power efficient, affordable, and perform its respective task(s) properly and efficiently. Our team conducted extensive research to ensure that the parts we picked for our solar picnic table met all of these requirements. With this done, the team is confident that the parts we picked will perform to expectations.

When it comes to designing, Each member was assigned various components to focus on and design how it was going to be implemented into the solar picnic table. Generally, the computer engineers of the group were responsible for the microcontroller, sensors, and the mobile app. Whereas the electrical engineers were responsible for the solar panel, battery, and power distribution. With this in mind, the team was able to design schematics and a PCB that would incorporate all of these components and allow them to interact with each other properly.

Once the designing phase was completed, we created testing plans to conduct on the different components. Each testing plan was developed to test the functionalities of the components. Within each testing plan, there were certain goals that each component had to achieve in order for it to be considered passing. For example, the DHT22 had to be capable of measuring temperature and humidity levels accurately in a wide variety of environments. If it was able to, then it would be considered passing and move onto being installed on the solar picnic table. With extensive testing, we can make sure that all of our components are functioning correctly.

With all of this done, we will begin prototyping the solar picnic table and start incorporating all of the components to work simultaneously together. With this said, our team is confident that we will be able to successfully build a solar picnic table that is power efficient, consumes low power, and is financially affordable.

## 8. Appendix

### A.1 References

- [1] “Off-grid system performance simulation website,” *JRC Photovoltaic Geographical Information System (PVGIS) - European Commission*, 11-Jan-2016. [Online]. Available: [https://re.jrc.ec.europa.eu/pvg\\_tools/en/tools.html](https://re.jrc.ec.europa.eu/pvg_tools/en/tools.html). [Accessed: 20-Mar-2023].
- [2] “DigiKey Part Figures,” *DigiKey*. [Online]. Available: <https://www.digikey.com/>. [Accessed: 10-Mar-2023].
- [3] “Seal Lead Acid Battery Figure,” *ExpertPower Direct*. [Online]. Available: [https://www.expertpower.us/collections/lifepo4-batteries?gclid=CjwKCAjw0ZiiBhBKEiwA4PT9z4PzAcs0FPxvkmsl1Z6Dw0HgD0RpxzHsSm1Tyw0eC1cdXms7UgRLahoCo2MQAvD\\_BwE](https://www.expertpower.us/collections/lifepo4-batteries?gclid=CjwKCAjw0ZiiBhBKEiwA4PT9z4PzAcs0FPxvkmsl1Z6Dw0HgD0RpxzHsSm1Tyw0eC1cdXms7UgRLahoCo2MQAvD_BwE). [Accessed: 25-Feb-2023].
- [4] “What's the difference between bluetooth and wi-fi?,” *Encyclopædia Britannica*. [Online]. Available: <https://www.britannica.com/story/whats-the-difference-between-bluetooth-and-wi-fi#:~:text=Bluetooth%20allows%20for%20short%20Drange,to%20connect%20to%20t he%20Internet>. [Accessed: 23-Mar-2023].
- [5] “Bluetooth technology overview,” *Bluetooth® Technology Website*. [Online]. Available: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>. [Accessed: 20-Mar-2023].
- [6] A. Shetty, “Infrared Sensor - how it works, types, applications, Advantage & Disadvantage,” *electricalfundablog.com*, 23-Oct-2018. [Online]. Available: <https://electricalfundablog.com/infrared-sensor/>. [Accessed: 23-Mar-2023].
- [7] C. Fair, “What are PIR sensors and why do I need them?,” *IES Edmonton*, 22-May-2018. [Online]. Available: <https://iesedmonton.org/news/2018-5-22/what-are-pir-sensors-and-why-do-i-need-them>. [Accessed: 23-Mar-2023].
- [8] “RF Wireless World,” *Advantages of Microwave sensor | disadvantages of Microwave sensor*. [Online]. Available: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Microwave-sensor.html>. [Accessed: 23-Mar-2023].
- [9] “RF Wireless World,” *Advantages and Disadvantages of Ultrasonic Sensors*. [Online]. Available: <https://www.rfwireless-world.com/Terminology/Advantages-and-disadvantages-of-Ultrasonic-Sensor.html>. [Accessed: 23-Mar-2023].

- [10] "Make your first IOT react native application with Bluetooth Low Energy," *RSS*. [Online]. Available: <https://www.bam.tech/article/make-your-first-iot-react-native-application-with-bluetooth-low-energy>. [Accessed: 13-Mar-2023].
- [11] "Charge controller," *Charge Controller - an overview | ScienceDirect Topics*. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/charge-controller>. [Accessed: 25-Apr-2023].
- [12] "Solar charge controller types," *SunWize*, 18-May-2018. [Online]. Available: <https://www.sunwize.com/tech-notes/solar-charge-controller-types/>. [Accessed: 25-Apr-2023].
- [13] "General 1," *EnerFusion Inc. - Power-Up GREEN!* [Online]. Available: <https://www.enerfusioninc.com/aurora-solar-picnic>. [Accessed: 25-Apr-2023].
- [14] "Solar charge station with fiberglass umbrella," *Kay Park & Recreation*, 22-Feb-2023. [Online]. Available: <https://kaypark.com/product/solar-charge-station-with-fiberglass-umbrella/>. [Accessed: 25-Apr-2023].
- [15] "Solar table leisure coffee table with wireless charging Bluetooth speaker for music playing villa garden," *Solar Table Leisure Coffee Table With Wireless Charging Bluetooth Speaker For Music Playing Villa Garden - Buy Solar Table, Coffee Table Solar, Coffee Solar Table Product on Alibaba.com*. [Online]. Available: [https://www.alibaba.com/product-detail/Solar-table-leisure-Coffee-Table-with\\_1600260568108.html?spm=a2700.pccps\\_detail.0.0.4e6113a0Ny2Eyt](https://www.alibaba.com/product-detail/Solar-table-leisure-Coffee-Table-with_1600260568108.html?spm=a2700.pccps_detail.0.0.4e6113a0Ny2Eyt). [Accessed: 25-Apr-2023].
- [16] "Amazon.com : [upgraded] 30A solar charge controller, Black Solar Panel ..." [Online]. Available: <https://www.amazon.com/Controller-Intelligent-Regulator-Paremeter-Adjustable/dp/B09799F9BQ>. [Accessed: 25-Apr-2023].
- [17] "Analysis of power-efficient high torque solar tracker through ... - strath." [Online]. Available: [https://strathprints.strath.ac.uk/73987/1/Ali\\_Shah\\_etal\\_ICEEE\\_2020\\_Analysis\\_of\\_power\\_efficient\\_high\\_torque\\_solar.pdf](https://strathprints.strath.ac.uk/73987/1/Ali_Shah_etal_ICEEE_2020_Analysis_of_power_efficient_high_torque_solar.pdf). [Accessed: 25-Apr-2023].