# Solar Power Tracking System

Team 9
Michael Sciancalepore
Ted Carey
Esteban Bravo
Collin Mulcunry
Technical Advisor: Professor Johanna Raphael
Instructor: Professor Ying Yu

May 8, 2022

**Section 1: Project Overview**

The objective of this project is to make a solar panel system that tracks the sun to the position where it receives the most light energy in order to maximize the power output of the system. The system will be easily portable and completely self-powered. It can be used to power small devices at the user's campsite. The system also comes with a Bluetooth app that tracks the voltage of the battery and the battery percentage.

The four team members were Michael, Esteban, Collin, and Ted. Michael is a computer engineering major and interested in electronics and the programming of microprocessors. His job on the project was to design, build, and code the sensors on the system, as well as assist with the Bluetooth connectivity. Esteban's major is computer engineering and his primary interest involves electronics and wireless systems. His job on the project was to design and build the circuitry that acted as a voltmeter for the app, as well as implement the Bluetooth connectivity of the system. Collin's major is computer engineering and his main interest includes software design. Collin's job for the project was to design and code the layout of the app, as well as incorporate the necessary features on the app. Ted's major is electro-mechanical engineering technology, and his main interest involves designing and building mechanical systems. His job on the project was to design and build the frame and base of the system for the solar panel and sensors.

The primary use-case of the product is campsite users. Campsite users can take this outside during their camping trip and position it so the tilt of the panel faces the east-west direction. The panel will adjust accordingly to the current position of the sun. When completed, the user can store the system on the RV for whenever they will use it next. When the system is outside, a button on the charge controller will turn it on and off. The user can plug the load they want into the charge controller to use the system's power. They can also connect their phone to the system using the Bluetooth app and track the battery voltage and battery percentage.

The social and economic impact that this project has is that solar panels can save people thousands of dollars long term on their electrical bill. Of course, this is a long-term investment, as solar panel systems are quite expensive to buy up front, but are almost always worth it in the end. People can also save a large sum of money by taking advantage of government tax deductions that are available for installing a renewable energy source [1]. Solar panels also have a great environmental and global impact on society. By using renewable energy from the sun, this reduces energy produced from fossil fuels, which in turn reduces carbon emissions into the atmosphere. It is estimated that about 30 billion metric tons of carbon dioxide was emitted globally in 2020 [2]. While climate change is a common concern in today's world, trying to reduce the amount of carbon emissions from fossil fuels will certainly have an impact. Our group is hoping that more people will consider using renewable energy by using more solar powered products.

One ethics topic that this project may be involved in includes safety. Solar panels can potentially be a health hazard. If people get continuous exposure to some of the electromagnetic radiation, it can be harmful. However, the amount of radiation generated is usually too small to affect a human's body [3].

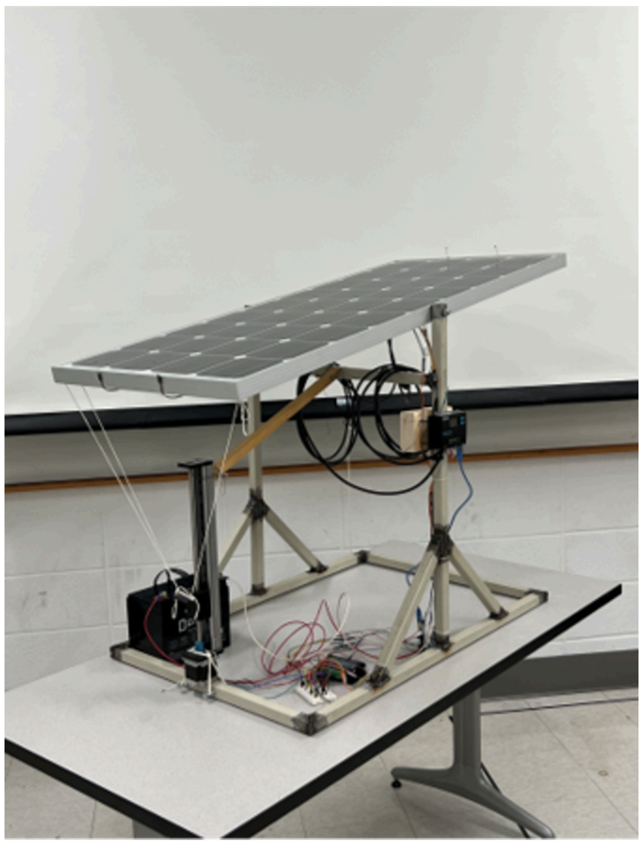

**Figure 1.1: Final system build, showing the solar panel, linear actuator, battery, and charge controller**

**Section 2: Detailed Use-Case**

The design goal of the project was to create a solar panel system which would effectively track the sun's movement in the sky across an entire day. While this system could be implemented on any solar panel with any use, it would be more realistic to prove the concept in a portable form rather than on a larger, more industrialized scale. The use case was selected based on the idea of a smaller scale system. Eventually, the intended use case of powering a campsite was selected. This would fit the desired system size and would also make sense as an actual product as many campers utilize solar panels to power the electronics on their sites. Implementing the tracking system would increase the efficiency of the panels already used as well as the rest of the system allowing for portability of the panels; something which is not always possible depending on the existing panel setup.

The main purpose of the panel in the campsite would be to power small electronics. It would store charge in a battery, which could be accessed to power devices like lights or charge phones. The system would be for small devices and would not be used for providing power on a large scale, such as to a cabin. The system would serve as a replacement for a generator, or as an add on to a battery which would allow for around-the-clock recharging. This would also mean the system would need to remain portable, as it would need to be easily moved in the event of the location of the campsite changing. The frame would also need to be adjusted once on site to match the east-to-west pathing of the sun.

Some design elements would be essential for the system to fit the campsite use case. Most importantly the system would need to be portable. This would be achieved by making the frame of the system out of lightweight material, choosing a smaller and lighter- but still powerful- solar panel and minimizing the weight of the battery. The battery specifically would be the most difficult element of the system to make portable, as batteries of the size needed for the system are often far heavier than the panels themselves. Next the system would need to be self-sustaining; if it were not, it would require an outside source of energy- such as a generator- to power the remainder of the system, which would negate the purpose of the panel. The system would also need to be weather resistant, due to it spending most of its time outdoors. Wind was the focus of the weatherproofing, as an imbalance in the panel could knock the frame over if a gust of wind hit it the wrong way, Finally, the system would need to be easy to assemble, disassemble and store for transportation, as the completed system would be too large to store in a regular car for transportation.

## Section 3: Constraints and Requirements

### *Section 3.1 Requirements*

To fit the campsite use case, there would be requirements for the system and its functionality. The focus was portability, as the system was intended to be relocated often. Because of this, the system would need to be lightweight. This could be achieved by using light materials to construct the frame, choosing fewer and lighter components, and making the system compact and tightly held together so components or wires were not hanging in the way of those trying to move it. The intention was to have the system easily moved by two people, or one person with a dolly. The system would also need to be self-sustaining, meaning the electronics powering the tracking would be powered by the system's battery, which would be charged by the system's panel. This would negate the need for an outside source to power the system's electronics, which would be difficult assuming the system is at a campsite with no other source of electricity. Weatherproofing and ease of transport would also be required, though these took lesser priority over the previous design elements.

Other requirements outside of the specific use case were considered. An app was to be created for ease-of-use and user compatibility with the system, connecting to the system via Bluetooth. While the functions of the app were not specific, they were intended to give the user information about the panel without having to be directly next to it while also being user friendly and easy to access. The system would also be required to track the sun in some manner, though photoresistors were quickly selected in order to complete this. Certain components were also required in order to fully complete a self-sustaining system of this kind; specifically, a battery, a charge controller to regulate battery voltage, an electric motor, and the panel itself.

### *Section 3.2 Constraints*

Based on the above requirements, constraints arose for individual components and the system. To achieve a lightweight and easy to transport system, component selection would be of utmost importance. A smaller sized solar panel would need to be selected, preferably still with 100W of power or more in order to charge the battery quickly. The battery, charge controller, motor and miscellaneous system pieces could not collectively be too heavy or too large to transport by hand. The frame of the system would also need to be made of lightweight but strong enough material to support the panel and any of its movements. Among other constraints included the microprocessor chosen for the motor code needing Bluetooth compatibility, as the app would need to be connected through Bluetooth. The motor would also need to be strong enough to move the panel while also doing it efficiently to minimize the amount of power being used to sustain the system's electronics.

## Section 4: Risk Analysis and Investigation

### *Section 4.1 Key Principles or Technologies and Risk Analysis*

The first key principle in our project is energy storage since we will be using a battery to store the collected energy as it mentions [4] "Solar energy can also be stored in electrochemical batteries. When solar energy is pumped into a battery, a chemical reaction among the battery components stores the energy.” Our second key principle is user interface because we will be making a user-friendly phone application for it. As it mentions, [5] “From recognizing interactive and static elements to making navigation intuitive, clarity is an essential part of a great UI design.”. For our third we have sensor to solar panel interface as it mentions [6], “Position sensors are useful for determining a device's physical position in the world's frame of reference.” Our fourth is circuitry since we will be connecting the sensors and motors together to a micro controller. It mentions [7] “The DC motors of the robot are connected to the controller using a motor driver IC”. For our fifth, we have mechanical mounting since we will be mounting a solar panel into a rotating base. It mentions, [8] "The standard residential system uses rails attached to the roof to support rows of solar panels”. Our sixth is environmental impact as it mentions, [9] “Solar energy systems/power plants do not produce air pollution or greenhouse gases.”. Our seventh is packaging as it mentions, [10] “Solar panels are typically either horizontally or vertically stacked in a box. Usually, separators are placed between each module, and extra protections are added to the four corners of each module stack.”. Finally, we have Programming since we will be doing some coding on the sensors, motors, and UI with a microcontroller. It mentions [11] “A microcontroller is a small, inexpensive computer, usually used for sensing input from the real world and controlling devices based on that input”.

Risk Assessment Table:

| | KPTs | Knowledge | Experience |
|---|---|---|---|
| 1 | Control mechanism | MS, EB | EB |
| 2 | Energy storage | MS, EB | EB |
| 3 | User interface | MS, CM | MS, CM |
| 4 | Sensor to solar panel interface | MS, EB, CM, TC | EB |
| 5 | Circuitry | MS, EB, CM, TC | MS, EB, CM,TC |
| 6 | Mechanical mounting | | |
| 7 | Environmental impact | CM, TC | |
| 8 | Packaging | MS, EB, TC | EB |
| 9 | Programming | MS, EB, CM | MS, EB, CM |

**Figure 4.1.1: Risk Assessment Table, showing the key principles and technologies with our group's initials showing who has knowledge and experience on each one.**

| | Know little about **‘0’** | Know about **‘1’** |
|---|---|---|
| Have done before **‘1’** | | - Control mechainsm<br>-Energy storage<br>-User interface<br>-Sensor to solar panel interface<br>-Circuitry<br>-Packaging<br>-Programming |
| Have never done **‘0’** | -Mechanical Mounting | -Environmetal impact |

**Figure 4.1.2: Summary Risk Map, showing a summary of the riskiest key principles or Tecnologies in our project.**

The riskiest key principles or technologies in our project are the mechanical mounting which Ted investigated. Programming which Michael Investigated. User Interface which Collin investigated. Sensor to solar panel interface which Esteban investigated

*Section 4.2 Investigations*

**Mechanical Mounting.**

Investigation 1: Literature Review

When it comes to solar panels, mechanical mounting is essential, the solar panel requires a base that is capable of holding twice its weight and allows the solar panel to rotate so it can follow the sun. In our research, we found that engineers developed special panel mounts for their solar panels. These mounts use special technology that changes the angle of the panel to coincide with the direction of the sun. They designed two types of mounts that are used depending on the amount of axis that the panel will function. One of them is the single axis mount, which simply tracks the sun east to west and does not consider seasonal changes. For example, the panel will not move according to the positional changes that the sun performs from the summertime to the wintertime. The other design type is the dual axis mount, which tracks both North and South and East and West. This type does take into consideration the change in position that the sun takes seasonally.

Investigation 2: Mockup

The solar panel has a weight of 7.5Kgs or 16.5lbs with dimensions of 1195 x 541 x 35mm or 47 x 21.3 x 1.4in. This means that our mounting base needs to hold about double the weight of the solar panel and be big enough to allow it to move at an angle. Below in Figure 4.2.1 we can see an illustration of our mockup design. The dimensions of the design can be seen below in Figure 4.2.2. The design is simple and effective. The linear actuator will be mounted to the left and the panel will be mounted on top and will be connected to the actuator in order to move the solar panel depending on the sun's position. The solar panel will be mounted using screws in a way, so they do not interfere with the solar panel's movements.

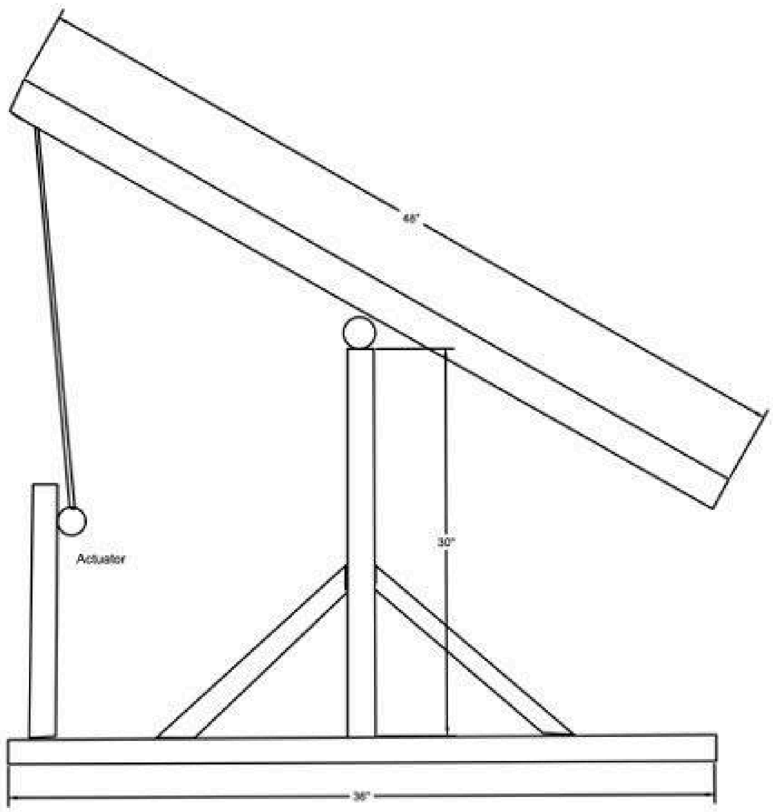


**Figure 4.2.1: Mockup design illustration, showing how our mounting base for the solar panel looks like.**

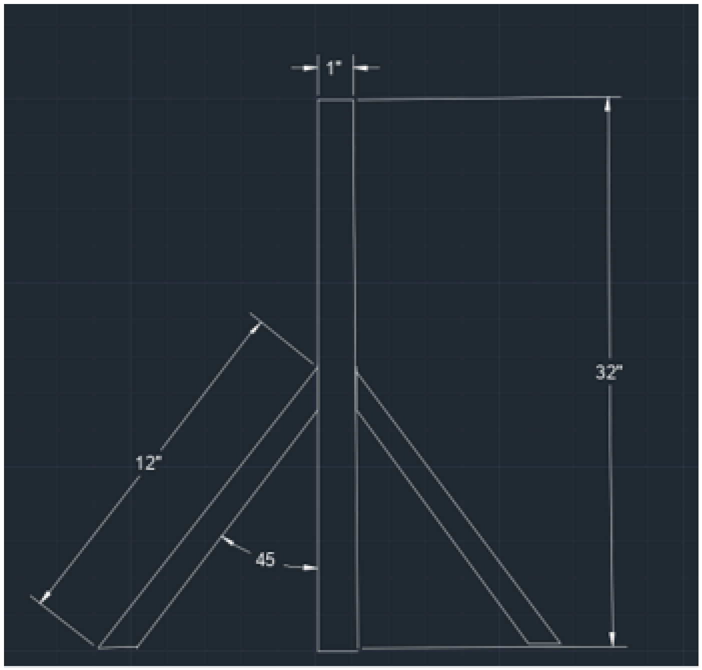


**Figure 4.2.2: Mockup design dimensions, showing the dimensions of the parts that make up our base.**

**Programming.**

Investigation 1: Literature Review.

The system will need some type of micro controller that allows the solar panel to move the motors depending on the sensor's readings. This micro controller would have to be programmed using a programming language. For the mobile application, we would need to design it, so it is compatible with the micro controller's code and be able to send data back and forward via Bluetooth. In our research, we found that engineers could use Arduino uno boards to interact with all types of sensors which include light sensors and Bluetooth chips. For our mobile application, we investigated that it would take an exceptionally long time to program an application using one of the most popular programming languages such as Java. To combat this, people with very tight due dates utilize MIT app inventor, which is a powerful tool for mobile application development. MIT app inventor uses its own style of programming which consists of connecting blocks to create lines of code.

Investigation 2: Prototype.

In our prototype, we decided to test if we could truly design an application in a short amount of time. We developed a prototype application so the user can turn an LED on and off by the press of a button. Using MIT app inventor, we developed this application which is code can be seen below on Figure 4.2.3. The application's interface can be seen below in Figure 4.2.4 as well.

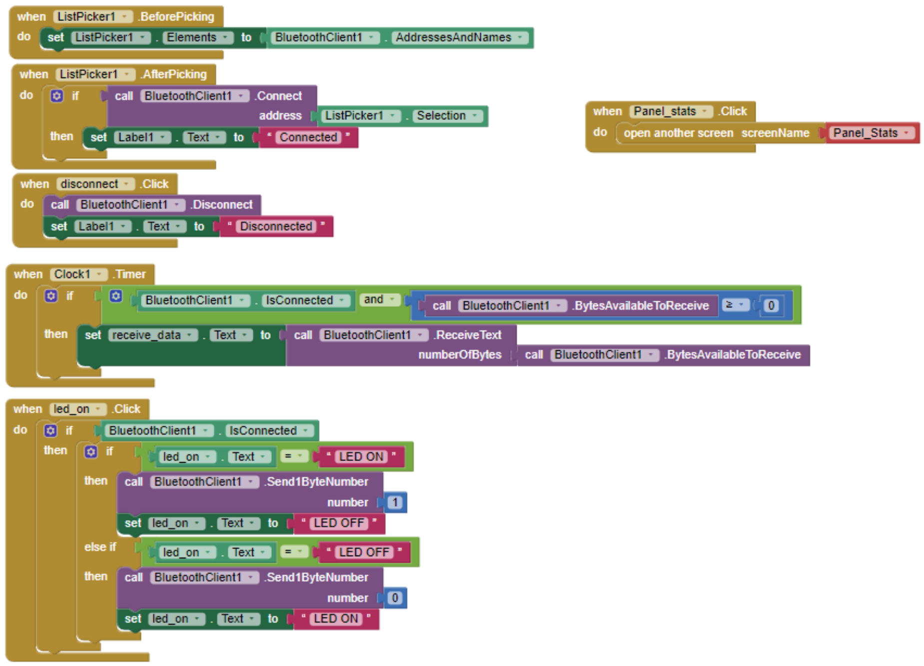


**Figure 4.2.3: MIT app inventor application code, showing the block code that allows to turn an LED on and off by the press of a button.**

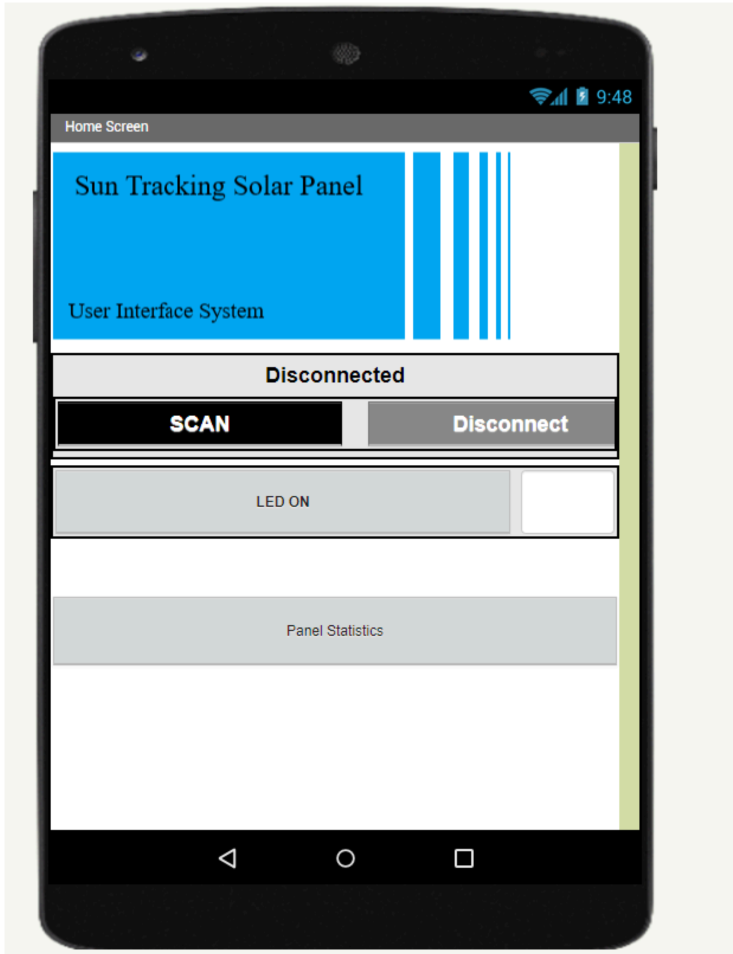


**Figure 4.2.4: Mobile application display, showing the turn on and off button for the LED.**

The prototype assisted with ensuring that we can develop a well-functioning mobile application in a short amount of time. It also ensured that the application can be used with Arduino code. When the user presses the “LED ON” button, the application will send the instruction to the Arduino via Bluetooth and make the LED turn on until the user presses the button again and will turn the LED off. This testing proved to be successful, and we will be using a similar style of development for our main system.

**Sensor to solar panel interface.**

Investigation 1: Literature Review.

The system needs a way to tell which position the sun is emitting the lightest in order to reach that ideal power efficiency that we can gather throughout the day. In our research, we found that engineers use light sensors in order to determine that ideal position. By placing light sensors on both sides of the solar panel, they can track that ideal position throughout the day and maximize the amount of power they can gather throughout the day. To make the solar panel move according to the light values read from the sun, micro controllers were programmed with motors that will move the solar panel whenever one side of the panel is reading a much higher amount of light than the other.

Investigation 2: Prototyping.

We designed a prototype that would test the light sensor's light readings. We gathered an Arduino Uno board and light sensors which were tested using a laptop. We coded a program that when the light sensors were exposed to light then an LED would turn on. In the contrary, when the light sensors are not resaving any light, then the LED will turn off. Once the light sensors were confirmed to work successfully, the prototype was modified in a way so instead of an LED turning on and off then a motor would turn on and off if the light sensors were exposed to any light or not. While these are not the exact components that will be used in the design, testing their functionality is important as Arduino will likely be the program used to code the final projects, and the final components we select will also need to be compatible with it. The code for the prototype is shown below:

```
int PIN1=1;
int PIN2=2;
int PINA=0;
int Value; //initializes the value at which the motor turns
int PIN11=11; //sets the variable PIN11 to pin 11 on the Uno board
void setup() {
     pinMode(PIN1, OUTPUT);
     pinMode(PIN2, OUTPUT);
     Serial.begin(9600); //opens serial port, sets data rate to 9600
bps
     pinMode(PIN11,OUTPUT); //sets pin 11 to an output pin
     }
void loop() {
     Value = analogRead(PINA); //reads the value from the LDR
     Serial.println(Value); //prints the value read from the LDR
     delay(100); //100 ms pause
      if(Value>100) {
          digitalWrite(motorPin, HIGH);
```

```
        }
        else {
              digitalWrite(motorPin, LOW);
        }
}
```

The prototype assisted with ensuring the system that was planned on being used would work, as it confirmed that the light sensor would work in an Arduino circuit. As the LEDs and electric motors were both confirmed to work in the system, it proves that individually the systems work. While they need to be tested together in order to confirm full compatibility, the systems working on their own remove a relatively large area of concern from the project and allows for other risk areas to be focused on, mainly the mechanical mounting.

**User Interface**

Investigation 1: Literature Review.

One of the features that our system will have in order to be unique is the mobile application that the user will have access to. When developing mobile applications, the user interface is one of the most important things to consider. App developers refer to the four main design principles to consider when developing a user interface. The first one consists of placing the users in control of the interface, meaning that a good user interface instills a sense of control in the users. If the user has control, then the user will become increasingly comfortable which will allow them to learn quickly and gain a fast sense of proficiency in the application. The second user interface principle is to make it as comfortable as possible to interact with the product. This leads to eliminating all elements that are not helping the user as well as protecting the user's work. The third user interface principle is to reduce cognitive load. This means that the user interface needs to be designed in a way, so the user uses as little amount of mental processing power as possible when interacting with the product. Finally, the fourth user interface principle is to make them consistent. The main idea of consistency is the idea of transferable knowledge throughout the application. Letting the users transfer their knowledge and skills from one part of the app's user interface to another is an effective way to make the user happy because they do not have to learn new knowledge or skills to interact with that other feature in the app.

Investigation 2: Prototyping.

When designing the mobile application prototype, we thought about making the application as user friendly as possible and require no technical knowledge to use. MIT App inventor was utilized to program and the design the user interface of the mobile application. By using MIT app inventor, we save a great amount of time on developing the application due to its powerful and simple programming style. The main page of the application prototype can be seen below in

Figure 4.2.3 as well as a confirmation message that will be prompted to the user whenever they execute an action. The prototyping consisted of creating the user interface of the mobile application as well as connecting the Arduino board with the application via Bluetooth. Once connected, the application will allow the user to turn an LED on and off by pressing the “Turn on” or “Turn off” button which will change depending on the status of the system.

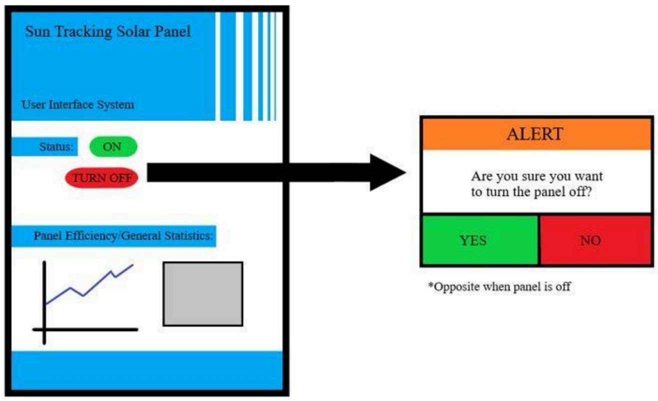


**Figure 4.2.5: Mobile application user interface, showing all features that will be available for the user as well as a confirmation message to prevent accidental actions.**

The prototype assisted with ensuring that MIT app inventor and Arduino were compatible and could be used to send any type of data between each other via Bluetooth. During the testing, we determined that the HC-08 Bluetooth chip was not compatible with Arduino, so we utilized an earlier version, the HC-06, which proved to work successfully. By pressing the “Turn on” or “Turn off” button on the mobile application, we had a value of 1 or 0 be sent to the Arduino code via Bluetooth. Once received, the Arduino will turn the LED on or off depending on the value sent which proved to be a complete success. This prototype ensured us that we would be able to manipulate the system using our mobile application however we want.

**Section 5: Project Break-Down and Schedule**

*Section 5.1: Break-Down into Areas*

1. Research
2. Ordering Materials
3. Preliminary Testing
4. Design Frame
5. Design Code
6. Design App

7. Test Subassemblies
8. Final Assembly
9. Prototype Testing

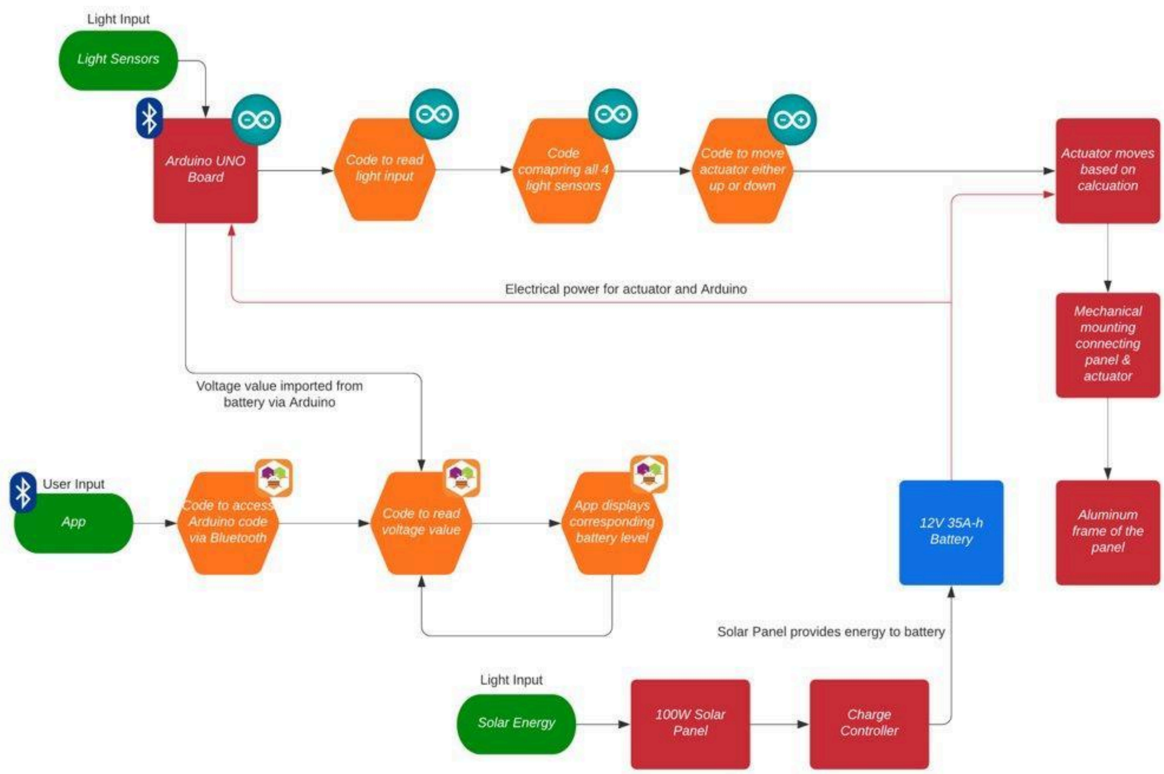


**Figure 5.1.1: Block Diagram**

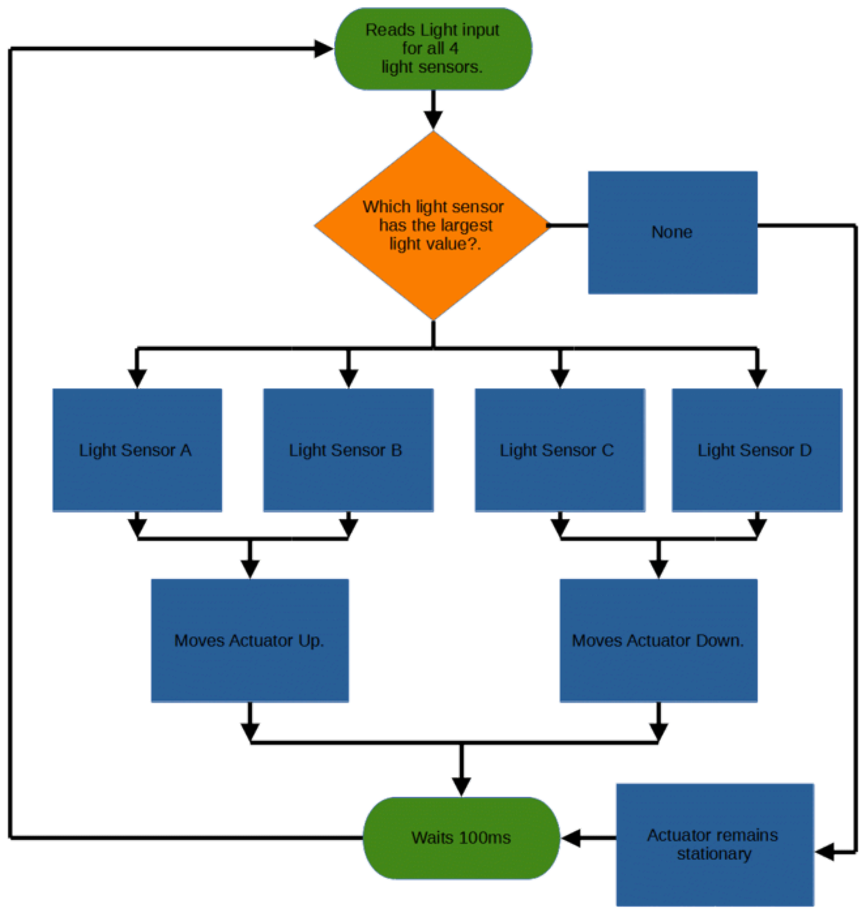


**Figure 5.1.2: System Flow Chart**

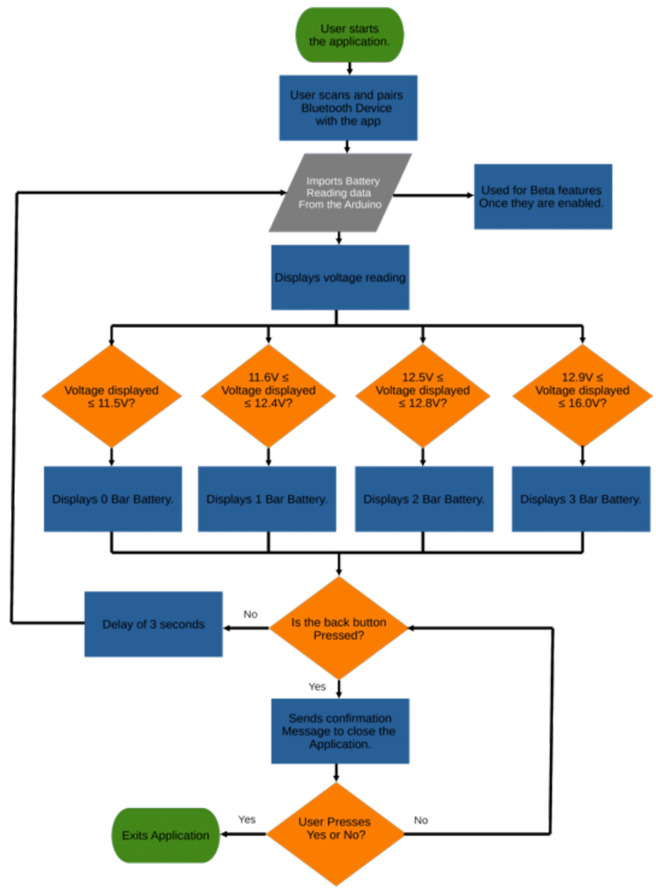


**Figure 5.1.3: App Flow Chart**

*Section 5.2: Design Evidence Groupings and Contributors*

Detailed Design Evidence Grouping can be found in section 6.1. Michael was the primary spokesperson for the group. He was the communicator for setting up meetings with our technical advisor and capstone professor. He also designed, built and coded the light sensors and actuator systems. He assembled the wiring system around the charge controller. Assisted with the Bluetooth development for the app. Planned our budget and parts list, and finally assisted with testing a troubleshooting the prototype. Ted was primarily responsible for the fabrication and assembly of the system. This means selecting materials, designing the framework, cutting, welding and testing the frame for strength. Mounting the panel onto the frame as well as the charge controller. Esteban Designed and tested the mobile application with working Bluetooth connectivity with the main system. Created dynamic QR code to allow all android devices to download and install the app. Designed, built, and coded voltmeter circuit for the battery reading of the main system with working Bluetooth connectivity with the app.  Performed cable management on the main system. Assisted with light sensors and actuator testing. Assisted with testing and troubleshooting of final product. Collin Created initial mobile app versions & code, performed initial Bluetooth setup. Designed app layout & visuals. Assisted with finalized Bluetooth connectivity circuit & battery reader circuit. Tested compatibility between actuator and Bluetooth circuit. Assisted with assembly, testing, and troubleshooting of the system. As a group we all monitored our own as well as the group's overall progress.

*Section 5.3: Schedule*

Capstone 1 class was spent primarily researching our project. Research was done to make sure the project was feasible with our timeframe and budget. To start this semester, the first objective planned was to have a detailed Gantt chart to keep the project on track. We began by putting together a parts list and ordering all material required, also gathering components borrowed from professors. While waiting for materials to be delivered we began designing the project. This includes the frame for the project and sample code to get a basic test for the photocells and linear actuator. This is the order that our Gantt chart was set up, however due to order delays our parts didn’t come when expected, so this created a time conflict with the project. There was also a delay in building the frame due to machine shop hours and class schedules. Once the steel material for the frame was gathered, over the course of a week the material was cut to all required lengths. I spent a week practicing welding before work began on the assembly of the frame. Once enough practice was performed, it took about two weeks to weld the frame together and test the strength of it.

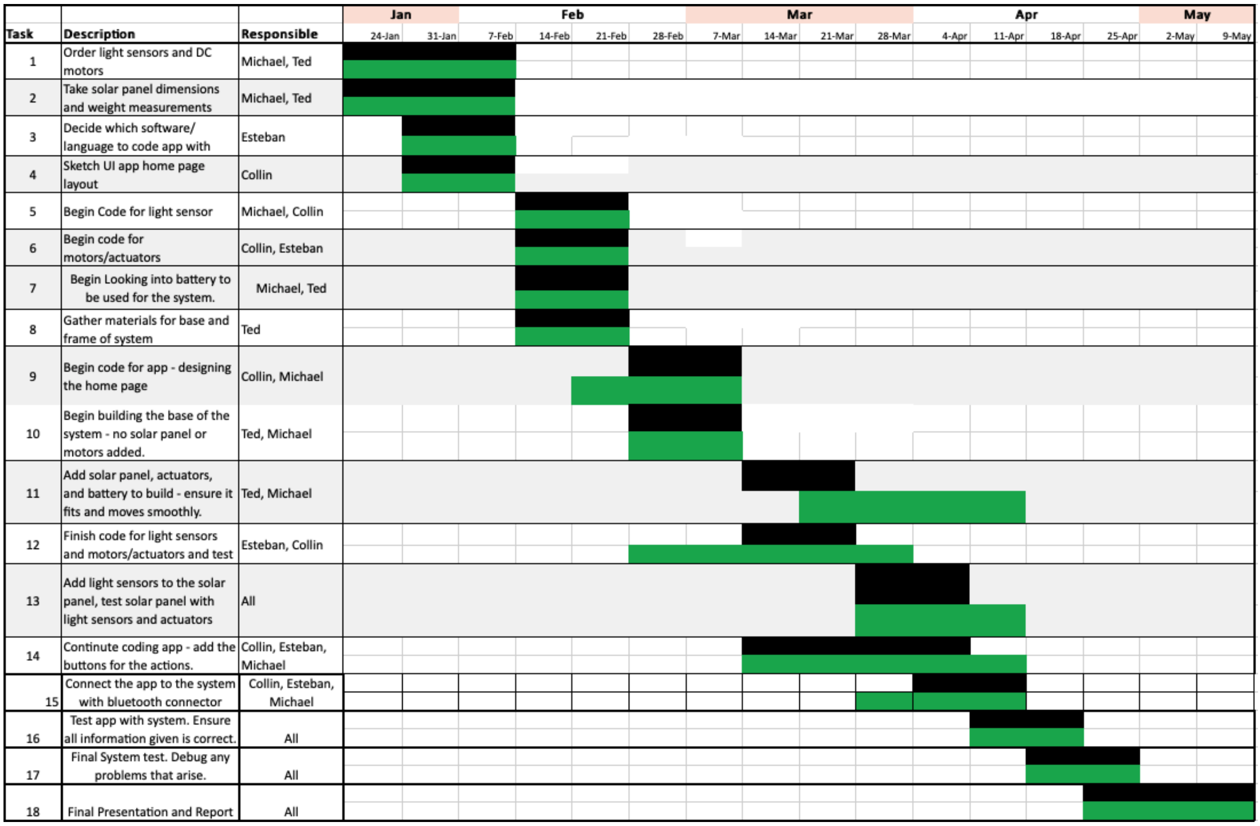


**Figure 5.3.1: Completed Gantt Chart**

## Section 6: Detailed Design

*Section 6.1: Design Evidence Groupings*

*Section 6.1.1: Design Introduction*

The design of the system is centered around the charge controller. The charge controller is the piece that sits at the center of the system and connects all the major parts of the project together. This includes the solar panel, the battery, and the load. The load that the system uses to run includes three main parts: the light sensors, the actuator, and Bluetooth. The light sensors are installed in voltage divider networks and act as the main input for the system. The actuator controls the tilt of the panel depending on the input of the light sensors. These components are controlled through the first Arduino controller. The second circuit on the system is a voltmeter that reads the voltage of the battery. This circuit is controlled by the second Arduino controller, which has an HC-06 Bluetooth module to display this information on the app.

One approach that was considered early in the design process was a 2-dimensional axis. This would cause the panel to tilt on two axes which would take into consideration seasonal changes of the sun's position. This would be a good feature to implement in the project because it would give more flexibility to the panel's position, taking in more energy from the sun, which is the main purpose of the project. The problem with this design is that it would be more expensive, larger, and add more weight to it. The portability factor of the project is important, which is why the team decided against it.

The team had to design the system based around our budget and the materials we were given. The key components that were provided to us by the university were the solar panel, the linear actuator, and the frame material. The Arduino controllers were already owned by members of the team, and the rest of the equipment was bought with the budget provided. This equipment includes the battery, charge controller, fuse, extra wires, extra photoresistors, and the Bluetooth module.

*Section 6.1.2: Photoresistor, Linear Actuator, Voltmeter, and Bluetooth Designs*

The design for the light sensor system was done in voltage divider networks. Each photoresistor is in series with a 1kΩ resistor and is powered by the microcontroller. Each photoresistor is connected to one of the input pins on the controller. There are two photoresistors on each side of the panel in case one of them malfunctions, the system will still work as intended. The actuator is positioned at one end of the solar panel and moves linearly due to the stepper motor and lead screw. The two pins on the actuator, the directional pin and the stepper pin, are connected to two of the output pins on the controller and move according to the input of the photoresistors. If one side of the light sensors is receiving more light than the other side, the actuator will tilt the panel up. If the other side of the light sensors is receiving more light, the actuator will tilt the panel down.

The design for the voltmeter circuit involved connecting the battery pins to the circuit and outputting the voltage to the serial monitor. The positive side of the circuit is connected to a 100kΩ resistor and one of the input pins on the controller. The negative side of the circuit is connected to a 10kΩ resistor and connects to the positive side of the circuit. The Bluetooth module is connected to the Rx and Tx pins on the controller and connects the circuit to the app. The app then displays the voltage reading on the serial monitor and then outputs a picture depending on the percentage of the battery. A separate Arduino controller was used for this to keep the programs independent from each other.

*Section 6.1.3: Arduino Code Design*

The design for the photoresistors and linear actuator code begins by initializing all the variables used, and to which pin they are assigned to. The photoresistors are all set to the input pins at A0, A3, A4, and A5. The stepper motor pins are assigned to pins 2 and 3 on the output, with the directional pin being assigned to pin 2 and the stepper pin being assigned to pin 3. In the setup function, the baud rate is set to 9600 bps, and the directional pin and stepper pin on the actuator are set to outputs. In the main loop, the move function is called. The move function starts by reading all four photoresistors, storing them in separate variables, and printing them to the serial monitor. There is then a delay of 100ms. Next, it compares all light sensor values to another. Light sensors A and C are on one side of the panel, while sensors B and D are on the other. If A or C are 80 points greater than B and D, the actuator will call the down function. If B or D are 80 points greater than A and C, the actuator will call the up function. Inside each if statement is a counter that increments to 10 before resetting and checking the next if statement. The up and down functions in the code are what move the stepper motor up or down. When the directional pin is high, the actuator moves up. When the directional pin is low, the actuator moves down. The main loop repeats every 100ms.

The design for the voltmeter circuit starts out by initializing all the variables. The input is set to pin A1, Vout and Vin are set to 0.00, R1 is set to 100k, R2 is set to 10k, and the initial variable is set to 0. The setup function initializes the input variable as an input and sets the baud rate to 9600 bps. In the main loop, the program reads the input, which is the voltage given to the circuit, and uses that to calculate the input and output voltage. From these calculations, it prints out Vin to the serial monitor as the main output. When the app connects to the circuit via Bluetooth, it automatically reads the value for its serial monitor.

*Section 6.1.4: Frame, Base, and App Design*

The design for the frame and base were simply based around the size and weight of the solar panel. The steel material was chosen because it was available to the team in the machine shop, and it would be strong enough to hold and move the solar panel. The design has one stand on each side that would hold the panel upright, and a bar going across each of these stands to stabilize the system better. The actuator is tied down at one end of the base and connected to the middle bar via a wooden strip. This is to stabilize the actuator. The actuator is then connected to the panel using a thread of yarn. The charge controller sits on one side of the stands where all the pieces of the project connect to, and the battery sits at the corner of the base.

The app is designed to be as user friendly as possible. It was designed using MIT App Inventor because of its ability to connect to Arduino systems and its easy use with Android devices. On the main screen, the user can press a button to scan for the proper Bluetooth module, and the app says whether or not it is connected. There is also a disconnect button to disconnect from the current Bluetooth module. The battery voltage will display automatically as long as it's connected. There is a button to display the charge picture depending on the battery reading, and another button to display information on what the percentage means. The last area of the app is the beta features, which are features to be implemented in the future. These include buttons to manually control the solar panel, and to see some other statistics about the system.

*Section 6.1.5: Conflicts, Components, Attributes, and Prototyping*

The two biggest problems that occurred during the implementation of the design were getting the actuator to move according to the light sensors inputs and getting the Bluetooth module to connect to the app correctly. The issue with the photoresistors and actuator was due to the controller not reading through the loop correctly. It didn't output the correct function based on the input because there was never a clear variable that said when to enter and exit certain statements. To address this, a counter variable was added that looped through each if statement and had the panel move correctly. The main issue with the Bluetooth connectivity was that the app was connecting, but not outputting the corresponding voltage to the serial monitor. The solution to this problem ended up being that an incorrect loop statement was inserted at the beginning of the function which wasn't necessary. This statement ended up causing the program to never enter the main loop to read the voltage. Once removed, the program worked correctly.

The components selected were based on what was given, and what the budget would allow. The solar panel, linear actuator, and frame material were provided, while the battery, charge controller, photoresistors, and Bluetooth module were bought using the budget. The solar panel used is 100W and measures 47 x 21.3 x 1.4 inches. The battery used is a Mighty Max 12V 35A-h battery. The reason this battery was chosen was due to the Watt-hours generated. The solar panel is 100W, and the average peak sun hours in New England is 4.2 hours. This would give an average of 420 W-h generated by the solar panel per day throughout the year. With the battery being 12V and 35A-h, that also gives 420W-h stored per day. This means the battery can store all the power generated by the solar panel. The charge controller was chosen because it was reasonably priced, was able to regulate the voltage of the battery, and was able to connect all the necessary parts together on the system. The HC-06 Bluetooth module was chosen because it was easily accessible and worked with Arduino and MIT App Inventor. Arduino was chosen as the microcontroller used because of the team's familiarity with the controller, as well as already having access to several units. The photoresistors that were chosen were used because of their easy accessibility and their compatibility with Arduino.

Significant attributes to the system include the sample code that was used to prove the linear actuator’s movement, a circuit that was used in the past by certain team members that used a photoresistor and LED, a referenced voltmeter circuit used on a breadboard, tutorial instructions on how to use the detailed features in MIT App Inventor, and reference code on how to connect the Arduino to MIT App Inventor using an HC-06 Bluetooth module. All of these attributes will be described in Section 8.1.

The prototyping done on the project includes testing all the different aspects of the system separately before putting them together. The prototyping for the photoresistors included building a small circuit using LEDs as outputs rather than the actuator. This would help test the logic and circuitry necessary on the final build. The linear actuator was prototyped using sample code that proved that the rotational motion of the stepper motor would be converted to linear motion by the lead screw [12]. The prototyping for the Bluetooth connection was done with a simple program that turned an LED on and off, and the prototyping for the voltmeter circuit was done using a DC power supply in the lab as the input [13]. All this prototyping helped break down the project to know which parts were working as specific times, which helped the team progress the project at a steady rate.

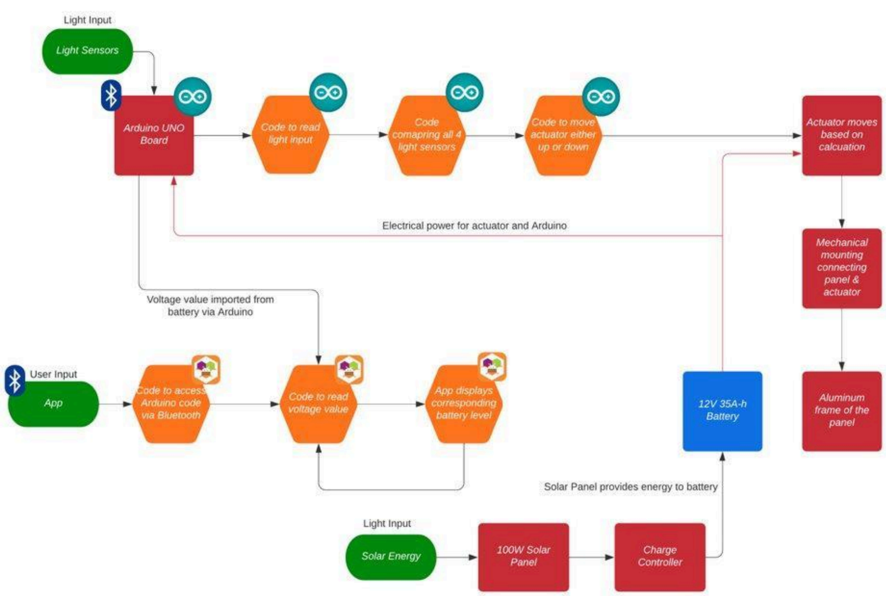


**Figure 6.1.1: Block Diagram**

**The overall diagram shows how all parts of the system are connected. The top section shows the steps of the photoresistor and actuator program, the middle section is the steps that the app takes while it’s running, and the bottom section shows how the solar energy is stored into the battery through the charge controller.**

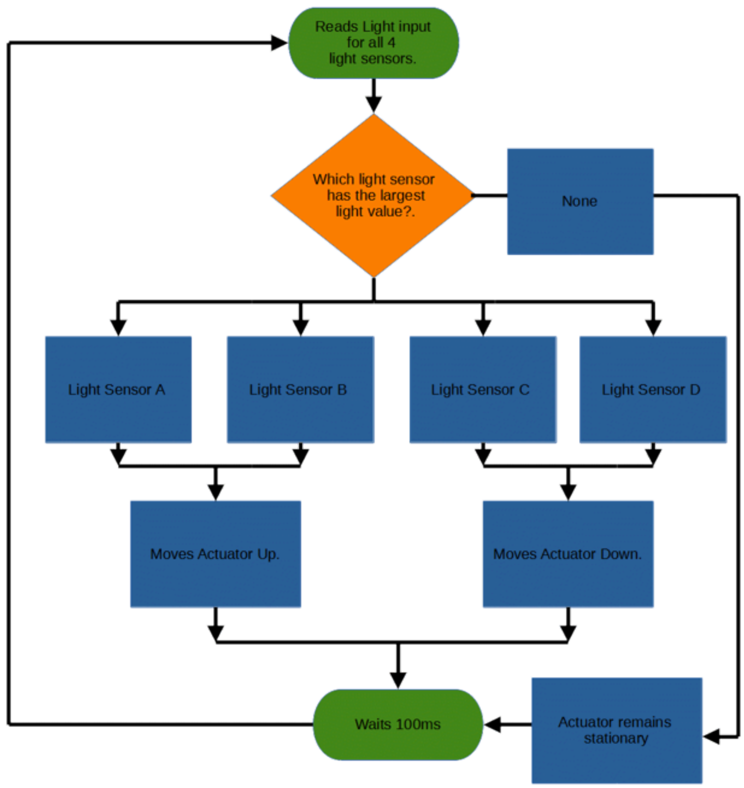


**Figure 6.1.2: System Flow Chart**

**The flow chart shows the logic of the system as it reads the light sensors input and outputs it to the actuator.**

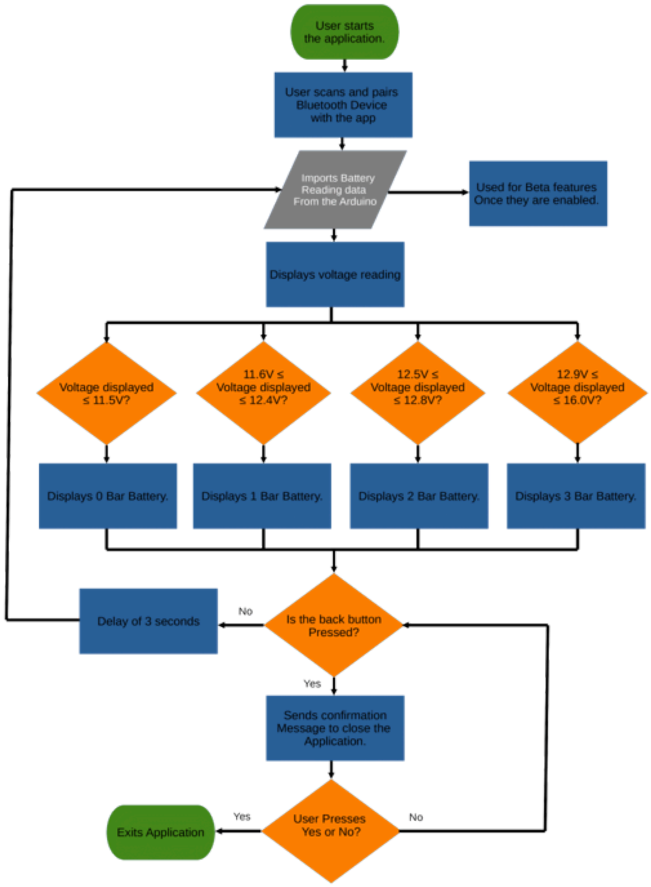


**Figure 6.1.3: Application Flow Chart**

**This chart shows the logic the app uses when it's opened, when each button is pressed, and the logic it uses to determine the percentage of the battery.**

*Section 6.2: Bill of Materials*

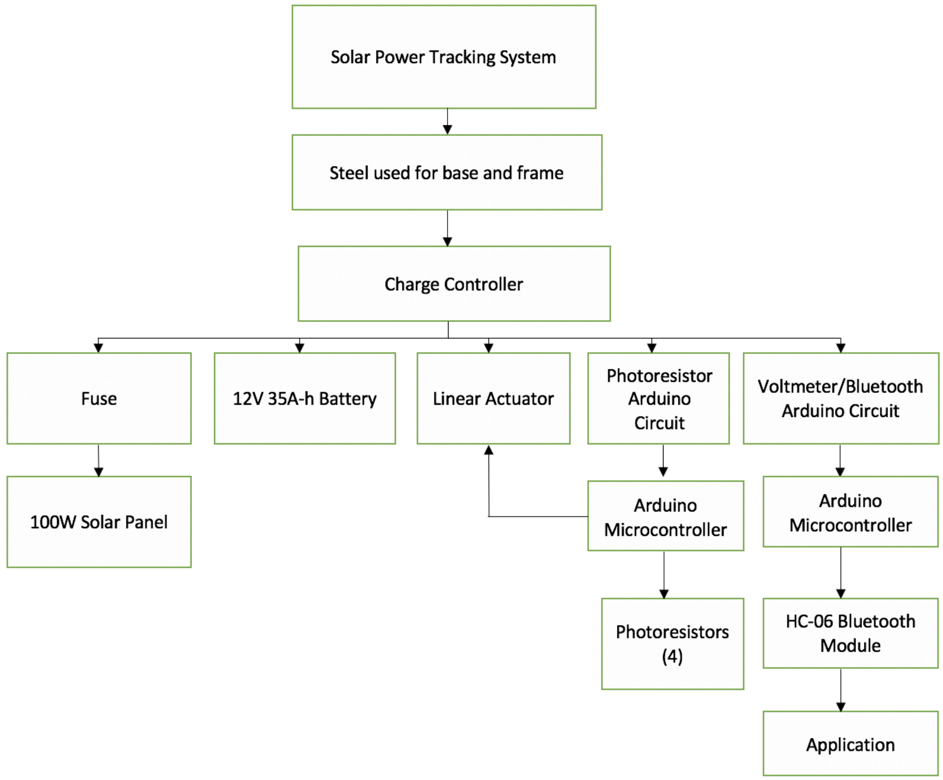


**Figure 6.2.1: BOM chart showing the hierarchy of the materials used**

**Table 6.2.1**

**Table showing level and quantity for each material used**

| Level | Description | Quantity | Given or bought |
|---|---|---|---|
| 1 | Steel used for base and frame | 13 | Given |
| 2 | Charge controller | 1 | Bought |
| 3 | Fuse | 1 | Bought |
| 3 | Mighty Max 12V 35A-h Battery | 1 | Bought |
| 3 | Linear actuator | 1 | Given |
| 3 | Arduino Microcontroller | 2 | Given |
| 4 | 100W solar panel | 1 | Given |
| 4 | Photoresistors | 4 | Bought |
| 4 | HC-06 Bluetooth Module | 1 | Bought |
| 5 | Application | 1 | Created |

## Section 7: Testing Methodology and Results

The final product testing consisted of seeing if the system would follow the ideal sun's position throughout the day. The system was set up outside on a regular sunny day, a few clouds were present, the testing environment was close to the ideal sunny day that would gather the most amount of power. The system was left to run for a couple of minutes at noon time when the sun's position was at its peak. This test was performed to see if the solar panel would adjust to the sun's position even though the starting angle was completely inaccurate. The system was also tested to see if the battery voltage was being read and displayed on the mobile application correctly.

**Figure 7.1.1: Solar panel starting position, showing the solar panel with the sun at its highest position ready to adjust the angle.**

**Figure 7.1.2: Solar panel adjusted position, showing the final position after letting the system run for a couple of minutes with the sun at the highest position.**

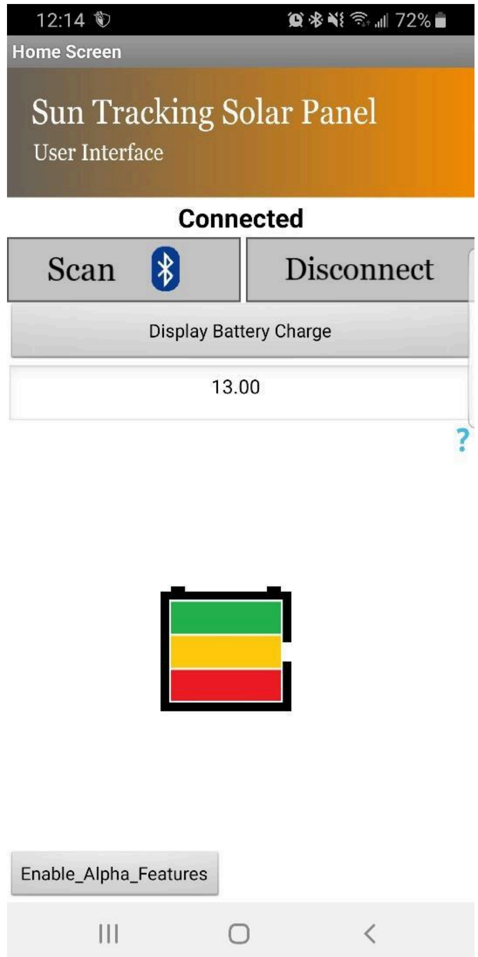


**Figure 7.1.3: Mobile application's display, showing the battery voltage reading being displayed according to the value read.**

| | |
|---|---|
| 3 Bars | Battery Voltage (12.9V- 16.0V) |
| 2 Bars | Battery Voltage (12.5-12.8V) |
| 1 Bar | Battery Voltage (11.6-12.4V) |
| No Bars | Battery Voltage (11.5V and below) |

**Figure 7.1.4: voltage reading table, showing the battery's voltage reading parameter display ranges.**

- The system was set up outside on a sunny day with very few clouds being present that would block the sun at times.
- A camara was placed to start to record as soon as the system started to run and would Capsure the entirety of the test until the system stops moving.
- The system was left to run for about 45 minutes.
- The Battery voltage was measured using a multi meter giving a reading of about 12.9v to 13v.
- The system was connected to the mobile application to test the battery voltage reading shown on the application which gave a reading of 13V and showed a full battery status display.
- The values from the multimeter and the applications were compared which ended up being extremely close to each other.

**Table 7.1.1: Table results, showing the data gathered.**

| Starting angle approximation | Finished angle approximation | Multimeter voltage reading. | App voltage reading |
|---|---|---|---|
| 210°(Facing West) | 180 °(Facing Up) | 12.9V | 13.0V |

Theoretically, since the system was started with a random angle around 30° facing West, with the sun's position at its peak, we should see the system adjust until the final angle is around 180° facing upwards and then stop. Analyzing Figures 7.1.1 and 7.1.2, who show the different timelines of the test, we can see that the solar panel did adjust to the sun's location which is above the system. Also, when displaying the battery's voltage readings on the application, we should see 0,1,2, or 3 bars being shown to the user depending on which range the value land on. For example, if we were to have a battery voltage reading of 13V then we should see a 3-bar display since the range of a 3 bar is between 12.9V to 16V indicating that the battery is well charged. Looking at Figures 7.1.3 and 7.1.4, showing the battery's voltage readings being displayed on the mobile application and the battery's voltage reading parameter ranges, we can conclude that the display of the voltage readings are accurate since we get a 13V reading and a display of 3-bars. These results mean that the system is functioning as expected and is ready for demoing. To finalize the project, there could be some small things we could improve upon such as the ascetics. The light sensors have small pieces of paper that are placed so that they can read at a better angle, instead of simple pieces of paper being taped, we could 3D print some small parts that would look more professional. This could also improve the angle readings since the parts would be designed with specific measurements in mind.

## Section 8: Attributions and References

*Section 8.1: Attributions*

The first attribution referenced for the project was the light sensor circuit that was used in a previous class by certain team members. In ECE 234, there was an assignment done that involved turning an LED on or off depending on the input of a photoresistor. This circuit was referenced and modified initial prototype of the photoresistors using LEDs as outputs.

The second attribution referenced was a demo with sample code that showed how the rotational movement of the stepper motor was converted to linear motion by the lead screw [12]. This tutorial showed where the pins on the driver connect to on the Arduino controller, and the sample code that causes the movement.

The third attribution referenced was a demo that showed how to create a voltmeter on a breadboard [13]. This tutorial showed how to set up the circuit as well as the code for the program. This demo was then used as the prototype for the voltmeter and Bluetooth portion of the project.

The fourth attribution referenced were several demo tutorials on how to use the functions in MIT App Inventor. Throughout the making of the app, several online tutorials were referenced to get comfortable with how the layout and coding are done, as well as what each block's function is in the code. These all helped what would eventually become the prototype for the app.

The fifth attrubtion referenced was a sample tutorial on how to setup and program an HC-06 Bluetooth module with Arduino and MIT App Inventor. The tutorial was used to connect the module to the proper pins, as well as program it to work with the app.

*Section 8.2: References*


1. Cooper, Santee. "Is Solar Energy Actually Cost Effective?" *Is Solar Energy Actually Cost Effective?*, 20 Oct. 2020, https://www.santeecooper.com/news/2020/102020-Is-Solar-Energy-Actually-Cost-Effective.aspx.
2. "Global Emissions." *Center for Climate and Energy Solutions*, 8 Sept. 2021, https://www.c2es.org/content/international-emissions/?gclid=Cj0KCQiAqbyNBhC2ARIsALDwAsDDqZ87DnrVVEDfrbSiMWla1KVY5SJP9Mx7MV29-0NGGbxDS6U6s0IaAhk2EALw_wcB.

3. McCloy, John, and Jeanne Yacoubou MS. "Do Solar Panels Cause Cancer? (an Analysis & Case Study)." *Green Coast*, 3 Dec. 2019, https://greencoast.org/do-solar-panels-cause-cancer-an-analysis-case-study/.
4. Ali McBride,Storing solar energy: everything you need to know, March 16, 2021, https://www.aurorasolar.com/blog/storing-solar-energy-everything-you-need-to-know/
5. Majam S. Chappal, The 6 key principles of UI design, May 25, 2021, https://maze.co/collections/ux-ui-design/ui-design-principles/
6. Android for developers, December 8, 2021, https://developer.android.com/guide/topics/sensors/sensors_position
7. Administrator, November 28, 2018, https://www.electronicshub.org/line-follower-robot-using-microcontroller/
8. Kelly Pickerel, February 13, 2017, https://www.solarpowerworldonline.com/2017/02/different-types-solar-mounting-systems-roofs/
9. U.S. Energy information administration, September 23, 2020, https://www.eia.gov/energyexplained/solar/solar-energy-and-the-environment.php
10. Niclas, May 19, 2014, https://sinovoltaics.com/solar-basics/basics-of-solar-panel-packaging/
11. Code, circuits and constructions, https://www.tigoe.com/pcomp/code/controllers/all-about-microcontrollers/
12. Benne, Author. "TB6600 Stepper Motor Driver with Arduino Tutorial (3 Examples)." *Makerguides.com*, 2 Mar. 2022, https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/.
13. "Voltmeter Using Arduino." *Arduino Project Hub*, https://create.arduino.cc/projecthub/next-tech-lab/voltmeter-using-arduino-00e7d1.

## Section 9: Reflections and Conclusion

### *Section 9.1: Reflection on Greater Impact*

Our project has many impacts on the environment. Solar energy continues to be a rapidly growing industry. Solar energy has proven to be a cleaner energy source than traditional power grids. For our use case, campers can collect the sun's energy rather than using a gasoline powered generator that pollutes the air. Using a generator also creates noise pollution in what are supposed to be calm and quiet surroundings. With the rising cost of fuel in the US, solar energy collection also avoids price hikes. This will also influence public health standards. If we commit to cleaner energy sources then the air will be more breathable, increasing life expectancy. The impact of leaving a smaller footprint on the environment.

*Section 9.2: Reflection of Knowledge Gained*

What Michael learned the most during this project was the overall project process as he did the most organizing, emailing, and electronic design. He also learned about the design process and programming microprocessors. Specifically, how to interface separate parts and program each as an input and output.

Esteban said that where he learned the most was battery reading, app development and programming microprocessors. He learned how to get two different software's to interact with each other, and more on how Bluetooth design of wireless connections work.

Collin also worked in the same areas as Esteban and said he learned the most about app development and working with Arduino and the Bluetooth chip. He learned more about app design, block coding, interfacing, and wireless connections

Ted had the most knowledge gained when learning how to weld. It was a week process in learning how to weld and another week to weld the frame together. He also learned more about how to design a mechanical system and how to combine it with an electrical system.

*Section 9.3: Conclusion*

Through the last semester of capstone, the group got to experience working on something from start to finish and see it come to life. As with every project in life, there were roadblocks, however the group analyzed and rethought the plan and found a solution each time. One implementation that was made to our frame design was an additional horizontal bar to connect the two vertical arms of the frame. This was not in the initial plans but was needed to provide additional support for the frame to hold the panel. The design process for code was writing out what the goal of the code was and the different ways to do it. The design process for the frame was to measure the full range of the actuator. Half the full range would be how much taller the vertical arms needed to be than the actuator. The base of the frame was designed to be stable on rugged terrain. Our final design had some differences than our initial design. Our final result was that our prototype worked with our theory, however we were not able to compare our results with the results if the panel were in a fixed position. A requirement for our project was that it needed to be portable, as a future improvement the frame can be redesigned to make it more collapsible. Another future improvement would be to implement further testing.

**Section 10: Appendices**

*Section 10.1: Project Costs*

**Table 10.1.1**

**Table showing the cost of all products bought**

| Part | Cost |
|---|---|
| Elegoo Dupont Wires | $8.49 |
| Photoresistors | $7.10 |
| Charge controller and battery | $95.69 |
| Waterproof in-line fuse holder | $13.25 |
| Solar panel to charge controller adapters | $20.97 |
| HC-06 Bluetooth module | $8.59 |

*Section 10.2: App Screenshots and Code*

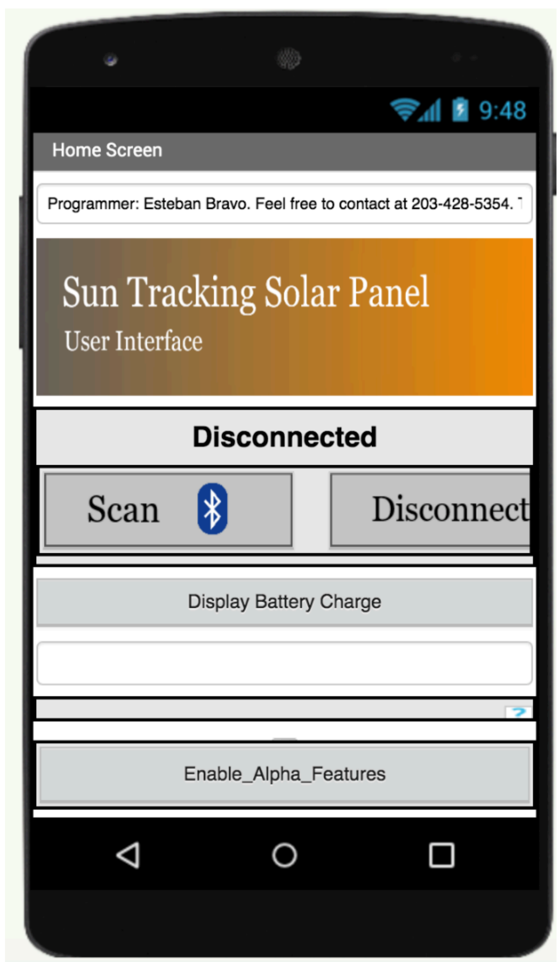

Figure 10.2.1: App home screen

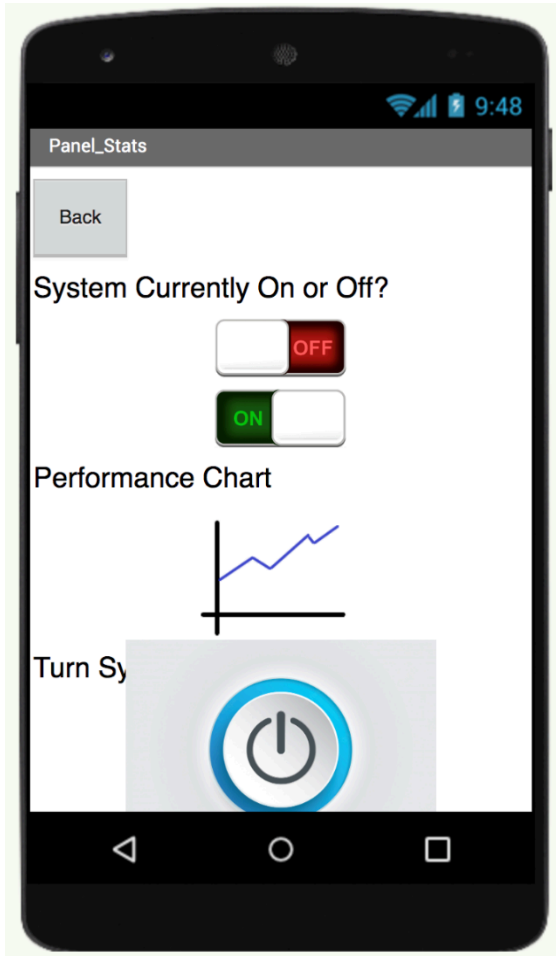


Figure 10.2.2: Panel Statistics Screen

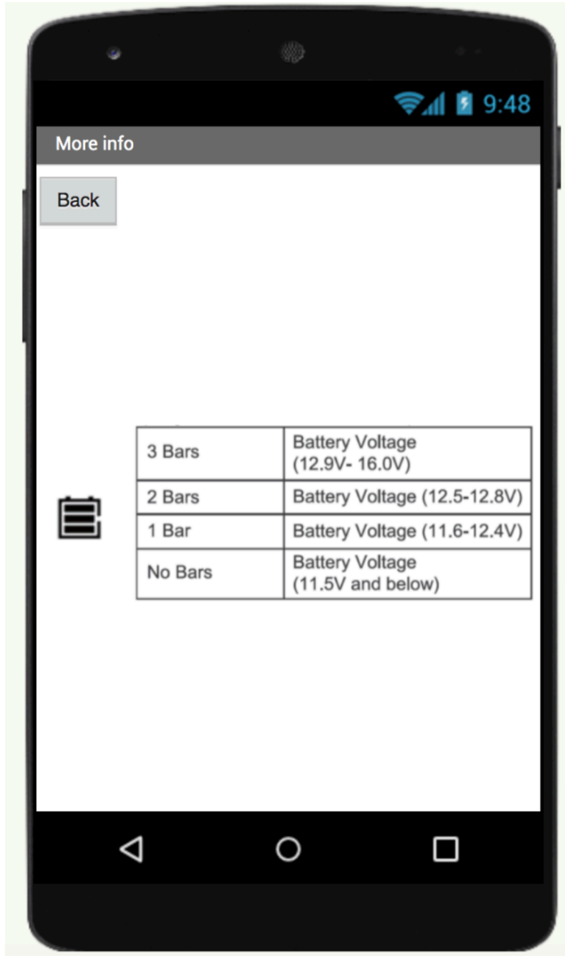


Figure 10.2.3: Battery Statistics on App

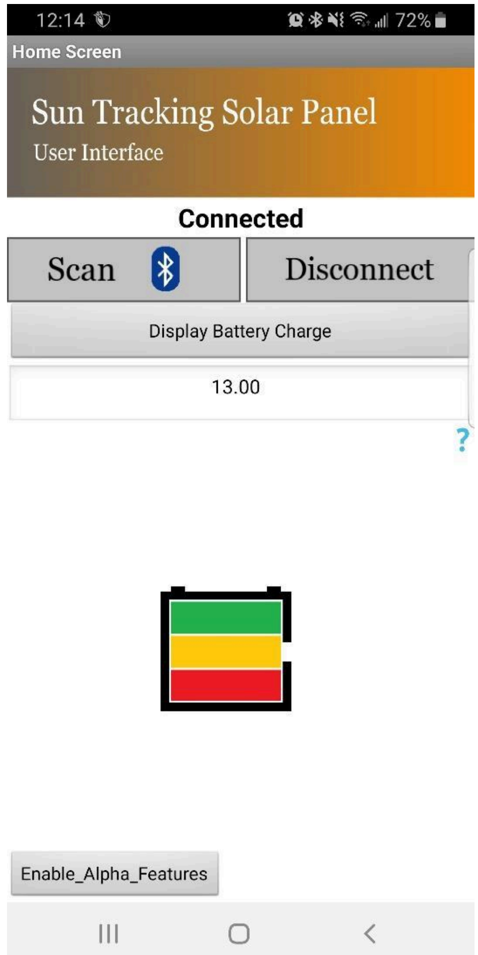


Figure 10.2.4: Demo of Application Running on a Smartphone

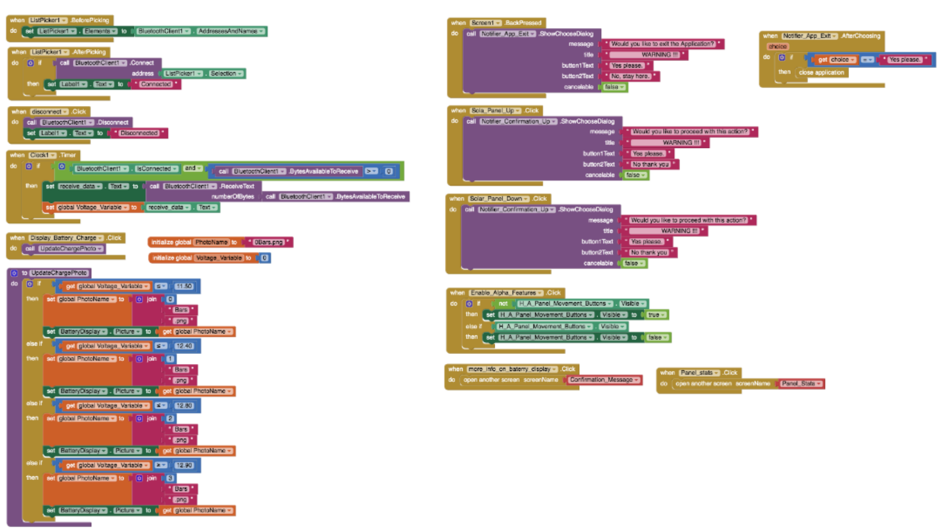

Figure 10.2.5: Block Code of App's Main Screen


Figure 10.2.6: Panel Stats Screen Code

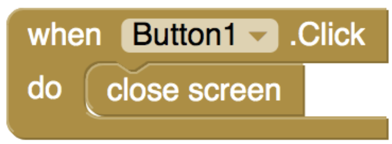


Figure 10.2.7: Confirmation Screen Code

*Section 10.3: Arduino Code*

Photoresistors and Linear Actuator Code:

```
int ldra = A0;

int ldrb = A5;

int ldrc = A3;

int ldrd = A4;

int ValueA;     //initializes the value at which the LED will turn on and off

int ValueB;

int ValueC;

int ValueD;

int dirPin = 2; //direction pin on actuator set to pin 2

int stepPin = 3; //step pin on actuator set to pin 3

int LED1 = 11; //sets the variable PIN11 to pin 11 on the Uno board
```

```
int LED2 = 12; //sets the variable PIN12 to pin 12 on the board
void Up();
void Down();
void Move();
int count = 0;
char value='1';

void setup()
{
  Serial.begin(9600);     //opens serial port, sets data rate to 9600
bps
  pinMode(dirPin, OUTPUT); //sets direction pin on actuator to output
  pinMode(stepPin, OUTPUT); //sets step pin on actuator to output
}

void loop()
{
  Move();
}

void Move(void)
{
  ValueA = analogRead(ldra);  //reads the value from the LDR
  ValueB = analogRead(ldrb);
  ValueC = analogRead(ldrc);
  ValueD = analogRead(ldrd);
  Serial.println("A");
  Serial.println(ValueA);     //prints the value read from the LDR
  Serial.println("B");
  Serial.println(ValueB);
  Serial.println("C");
```

```
  Serial.println(ValueC);
  Serial.println("D");
  Serial.println(ValueD);
  delay(100);
  if (ValueA > (ValueB + 80) && ValueA > (ValueD + 80))
  {
    while (count < 10)
    {
      Down();
      digitalWrite(LED1, HIGH);  //turns first LED on if LRD-A is
higher than B
      digitalWrite(LED2, LOW);
      count++;
      //Serial.println(count);
    }
    count = 0;
    //Serial.println(count);
  }
  else if (ValueB > (ValueA + 80) && ValueB > (ValueC + 80))
  {
    while (count < 10)
    {
      Up();
      digitalWrite(LED1, LOW);   //turns second LED on if LDR-B is
higher than A
      digitalWrite(LED2, HIGH);
      count++;
      //Serial.println(count);
    }
    count = 0;
    //Serial.println(count);
```

```
  }
  else if (ValueC > (ValueB + 80) && ValueC > (ValueD + 80))
  {
    while (count < 10)
    {
      Down();
      digitalWrite(LED1, LOW);   //turns second LED on if LDR-B is
higher than A
      digitalWrite(LED2, HIGH);
      count++;
      //Serial.println(count);
    }
    count = 0;
    //Serial.println(count);
  }
  else if (ValueD > (ValueA + 80) && ValueD > (ValueC + 80))
  {
    while (count < 10)
    {
      Up();
      digitalWrite(LED1, LOW);   //turns second LED on if LDR-B is
higher than A
      digitalWrite(LED2, HIGH);
      count++;
      //Serial.println(count);
    }
    count = 0;
    //Serial.println(count);
  }
  else
  {
```

```
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
  }
}
void Up(void)
{
  digitalWrite(dirPin, HIGH);

  digitalWrite(stepPin, HIGH);
  delayMicroseconds(500);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(500);
}

void Down(void)
{
  digitalWrite(dirPin, LOW);

  digitalWrite(stepPin, HIGH);
  delayMicroseconds(500);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(500);
}
```

Code for voltmeter circuit:

```
int analogInput = 1;
float Vout = 0.00;
float Vin = 0.00;
float R1 = 100000.00; // resistance of R1 (100K)
```

```
float R2 = 10000.00; // resistance of R2 (10K)

int val = 0;

void setup() {

pinMode(analogInput, INPUT); //assigning the input port

   Serial.begin(9600); //BaudRate


}


void loop() {

   val = analogRead(analogInput);//reads the analog input

   Vout = (val * 5.0) / 1024.0; // formula for calculating voltage
out i.e. V+, here 5.00

   Vin = Vout / (R2/(R1+R2)); // formula for calculating voltage in
i.e. GND

   if (Vin<0.09)//condition

   {

     Vin=0.0;//statement to quash undesired reading !

   }

    Serial.print(Vin);

    Serial.println("\n");

    delay(1000); //for maintaining the speed of the output in serial
moniter

}
```