Spring 2017

Smart Fitness Machine

Tyler M. Masters
tmm120@zips.uakron.edu

Recommended Citation

Masters, Tyler M., "Smart Fitness Machine" (2017). *Honors Research Projects*. 493.
http://ideaexchange.uakron.edu/honors_research_projects/493

Smart Fitness Machine

Tyler Masters

Department of Electrical Engineering

Honors Research Project

Submitted to

The Honors College

Approved:

Date 4/28/17
Honors Project Sponsor (signed)

Gregory A. Lewis
Honors Project Sponsor (printed)

Date 4/26/17
Reader (signed)

Shiva Sastry
Reader (printed)

Date 04/26/2017
Reader (signed)

Nghi Tran
Reader (printed)

Accepted:

Joan Carletta Date 4/27/17
Department Head (signed)

Joan Carletta
Department Head (printed)

Date 04/26/2017
Honors Faculty Advisor (signed)

Nghi Tran
Honors Faculty Advisor (printed)

Date ________
Dean, Honors College

The design project completed by my group was a smart fitness machine. This machine tracks a user's performance and stores it to be accessed by either a user or a trainer through the use of a website or mobile phone application.

In the early stages of the design process we collectively worked to narrow down topics and come up with our specific project through research and group communication. I specifically did research on existing patents that had similar concept in order to ensure that we did not legally infringe on any existing patents and also to gain a better understanding of what type of product the industry is lacking.

During the design phase of the project I was tasked with figuring out which sensors were necessary to provide us with data which we could then translate into user performance. Once it was determined what sensors were necessary, I then determined the model of sensor we would use based on the specs we determined to be necessary for our final product to operate smoothly. Along with these duties I assisted the group in developing software flow charts to plan out how the sensors would integrate with each other.

In the implementation phase I was in charge of writing the computer code for the microprocessor. The first step was to breadboard each of the sensors and gather data from each individually. Once every sensor was achieving smooth readable data, I began to implement certain functions that used sensor data to output data and trigger certain events. Eventually more functions were built from these smaller functions until a fully functioning state machine was built. Once the code was working the sensors were implemented onto the machine. After this, many code changes had to be made to account for the physics of the sensors being mounted on the machine as opposed to bread boarded on a work bench. Coding was the greatest challenge I faced due to a lack of experience. However, seeing my code work on the final product was the most rewarding part of the entire process.

Smart Fitness Machine

Midterm Design Report

Design Team 03

Tyler Masters

Jordan Pearce

Andrew Thornborough

Ben Waters

Faculty Advisor: Dr. Shivakumar Sastry
10/23/16

Table of Contents

Table of Figures

Table of Tables

Table of Equations

Abstract

The Smart Fitness Machine is an aftermarket product that is designed to automatically track a user's performance for a specific workout machine. The system offers a modular solution for any modern weight lifting machine that does not have a "smart" or internet connected feature. This product is inexpensive and offers an ideal solution for gyms to make their machines "smart" and upgrade a gym with modern specifications. This system eliminates the need for a user to manually record weight, sets, and repetitions during a workout. The system also allows for a personal trainer or physical therapist to monitor a client's progress as well as give a client specific instructions without being present. The data gathered is displayed in a useful manner via Android phone and online. The system consists of a microcontroller that is connected to multiple sensors on the machine. The microcontroller collects sensor data and calculates the amount of weight, reps, and sets a user has completed on the machine. This microcontroller will then send the data over the Internet through a Wi-Fi connection to a database to be stored and sorted. Then, a user can then view his or her workout information via Android phone and online. By providing an affordable and modular solution, which can be used on many weight machines, this product will allow users to focus on the workout while getting quality data.

1 Problem Statement

1.1 Need

In order to help users track their progress in exercise performance and achieve their objectives, it is important to design new technologies that can track performance from workout to workout. Having your workout logged is important when managing individual performances. It would be ideal for a gym user to have his or her performance tracked and compared to goals given by a trainer automatically during a workout without the hassle of manually logging information. Having these results tracked and displayed provides the user with the ability to conveniently track progress and set and reach goals. On the other hand, upgrading gym equipment is expensive and the amount of money that is needed to have this feature is not worth the cost of upgrading the machine, there needs to be a cheaper solution.

1.2 Objective

Understand the principles of systems design, implementation and verification. Create a wireless tracking system for workout equipment. This system would be able to input specific fitness goals for a user outside the gym setting while being able to recognize the user, track weight used, repetitions, and sets for a specific workout inside the gym. This information would be transmitted and displayed side by side via smartphone application. This system allows the user and trainer to track progress and set new goals in order to be most efficient in a user's fitness endeavors.

1.3 Background

In order to further develop the scope of our project it was necessary to conduct research on what product similar to ours currently exist on the market. First, applicable patents were researched to ensure the project will not infringe on any current designs. Next, a search of articles was done to gain an understanding of how exercise logging is currently being done and how it could be improved. The results of these searches are documented below.

1.3.1 Patent Search

Due to the widespread popularity of fitness in today's society, there are many applicable patents concerned with tracking a user's performance in a gym setting. Most of these, however, are more concentrated on tracking performance due to cardio or bodyweight exercises by using body sensors or GPS. Though still relevant, there are far less patents that deal with tacking the performance of weight training.

One existing design details a network of exercise machines in a gym that collect and store data from each user into a network system [4]. This design claims the ability to specify control of multiple machines for a particular user based on their past performance along with providing usage data to the gym itself [4]. This patent is worded to cover all types of exercise machines. However, due to the idea of controlling the machines, puts a focus on cardio based machines such as bicycles and treadmills where changing difficulty is easily accomplished through speed and resistance. The project described in this proposal focuses on weight lifting machines and, due to the variety of workouts performed by users, lets the user control the weight used. It then logs the data graphically in a convenient smart phone app providing convenient and easy usage.

Another design describes a more detailed system that will track the number of weights lifted and motion of the weights for a typical weight stack machine. This data will then be transmitted to a central storage and displayed on a screen [1]. However, this system seems to lack the ability to track the given data for a particular user and does not seem to transmit this data to anywhere other than a fixed screen located on the machine.

A third system is similar to the two systems described above. This product once again gathers information from a given type of exercise machine and sends it to a central computer. This data is then graphed on a host computer for the user to see [5]. Like the product detailed above this patent does not mention the ability to recognize one user versus the other.

A final design entails a system that using an identification device, recognizes a certain user and then prompts the user to fill in information based on what they performed during the exercise. This system automatically detects the user and has the certain exercise preprogrammed in, but cannot automatically retrieve the performance data [2]. This system is more or less just a convenient way to manually log a user's performance data.

There were more patents that described similar systems to the ones listed above. Through patent research it is evident that the idea of a performance tracking exercise machine is not a new concept. However, most of these systems could be modified or improved to make them either more user friendly or able to be used in a public gym setting.

1.3.2 Article Search

A group called Sports and Fitness Working Group is all about using Bluetooth v4.0. Bluetooth v4.0 is using low energy technology that allows any device, from a static data-collecting sensor to a laptop or tablet, the ability to connect. This connection accomplished by Bluetooth v4.0 allows sharing and distribution of information in real time. This Bluetooth sensor can be attached to a golf club and instantly send information about the users swing to his or her cell phone [6].

In a second article the focus was on the recognition of the sport activity. More specifically the qualitative assessment, user feedback, detection of mistakes, and the quality of the workout. The workout has been performed and tested by manufacturers before, and involves the manufacturers gathering information on the variables of each workout. These variables are grouped into different conditions, each of these different conditions are made up of mistakes or proper workout criteria. The sensors attached to the workout machine detect irregular workout patterns and send the information to the user. This allows physical trainers and the workout patients to consistently guarantee their workout is being done properly [7].

1.4 Marketing Requirements

1. The system should be power efficient
2. The system should be easy to operate
3. The system should detect the weight used
4. The system should detect the number of repetitions performed
5. The system should detect the number of sets performed
6. The data should be accurate
7. The data should be user specific
8. The data should display to the user in a convenient manner

1.5 Objective Tree

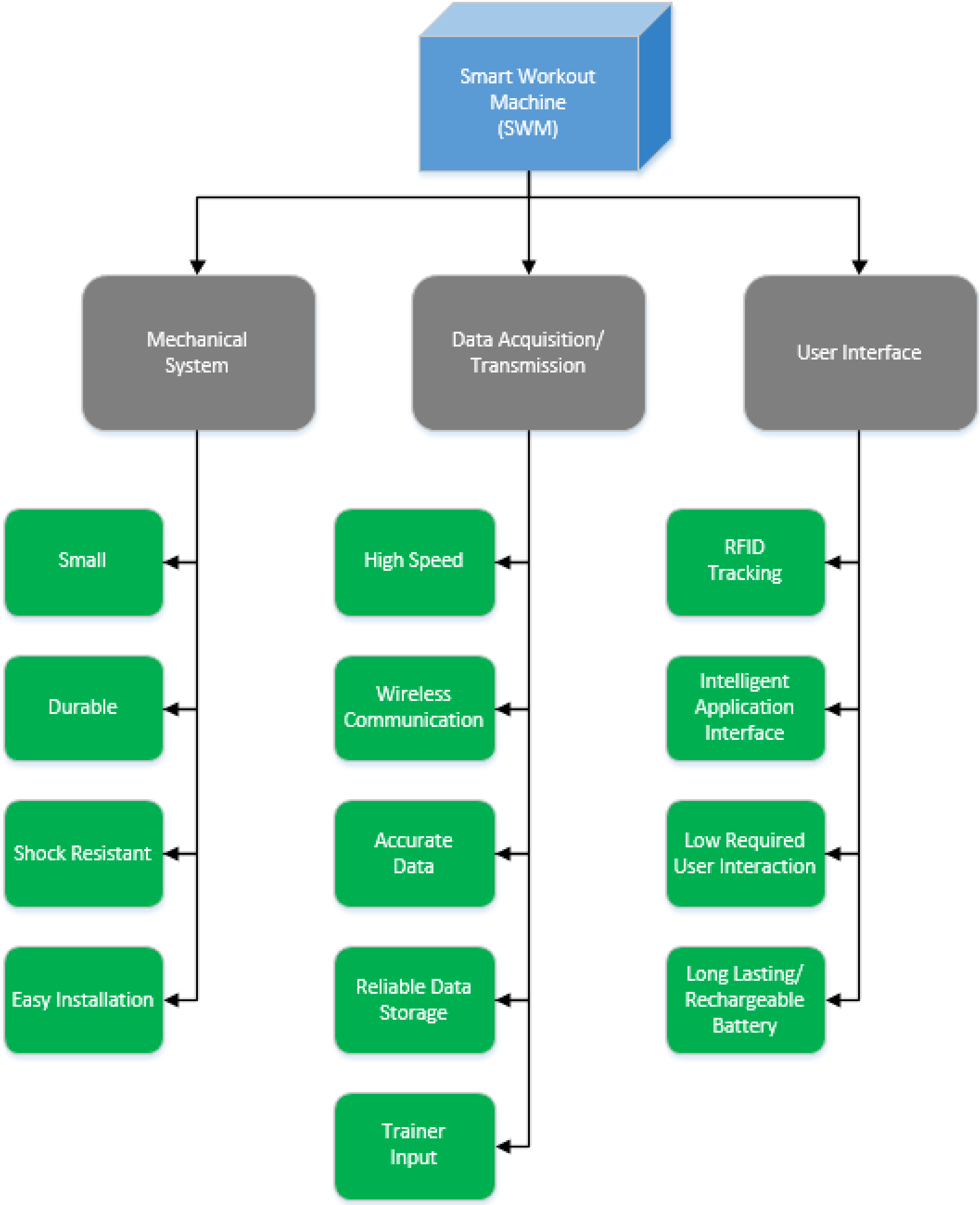


Figure 1: Objective Tree

1.6 Use Cases

We decided to follow a Use-Case driven design process. Each Use-case identifies a specific Stimulus-Response pattern that is triggered by an actor (user of the system). For each Use-case, we identified events of the system, the data that is transmitted throughout the system, and what controls must implemented. This type of thinking helped to analyze and understand a system, so that system requirements can be identified, clarified, and organized. Every Use-case shows the interactions between a user and the system related to a particular goal – the user's trigger is called the Stimulus and the action of the system is the Response. Our Use-cases include user interfacing events, data processing, data management, and control of the system. The Use-case also shows those functions that are internal to the system which process the data, and send controls to the hardware.

Based on this analysis, our team identified the following Use-cases for the system:

1. RFID Records Entry to the System

2. RFID Records Exit from the System

3. Data Management

4. Data Management and Display

Below, each Use-case is modeled and explained.

1.6.1 RFID Records Entry to the System

The first event in the system is when a user signs into the system. The LED will initially be lit red indicating the system is actively searching for a user. If the LED is currently green another user may still sign in triggering the previous user to be logged out and their data to be sent to the database. If the LED is lit yellow the database connection is lost and the user is to know that no data will be recorded until connection is restored and they are able to sign in. The user places their RFID band near the clearly marked RFID sensor. The microcontroller pulls data from the

RFID sensor at a rate of 50ms. The microcontroller takes that data and searches the database for the user account. If the user is authenticated a green LED is lit. If the user is not authenticated the red LED will flash three times. If no connection was made to the database a yellow LED will light and remain lit until communication is restored. The microcontroller will recheck for connection every 5 seconds until the connection is restored. At that point the system will reset and the red LED will once again light indicating the system is once again searching for log in. These steps are diagramed in Figure 2.

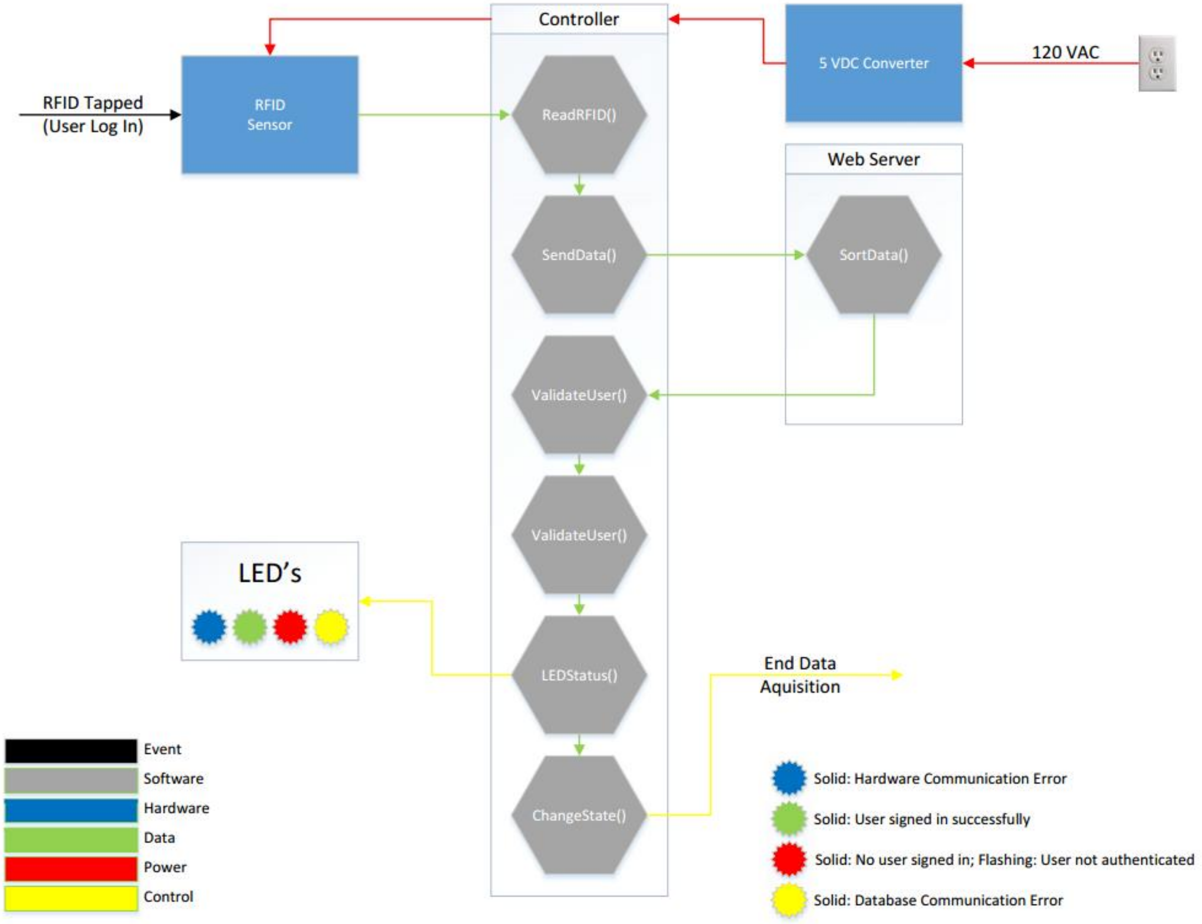


Figure 2: User Entry Use Case

1.6.2 RFID Records Exit from the System

Similar to use case number 1, the system also tracks when a user signs out of a machine. The LED is currently lit green indicating the user is successfully authenticated and their performance is being tracked. The user again places their RFID band near the clearly marked RFID sensor. This triggers the microcontroller to stop tracking data for the current user and sends any data that may not have been sent as a set. At this point the LED is again lit red indicating the system is looking for a new user. These steps are diagramed in Figure 3.

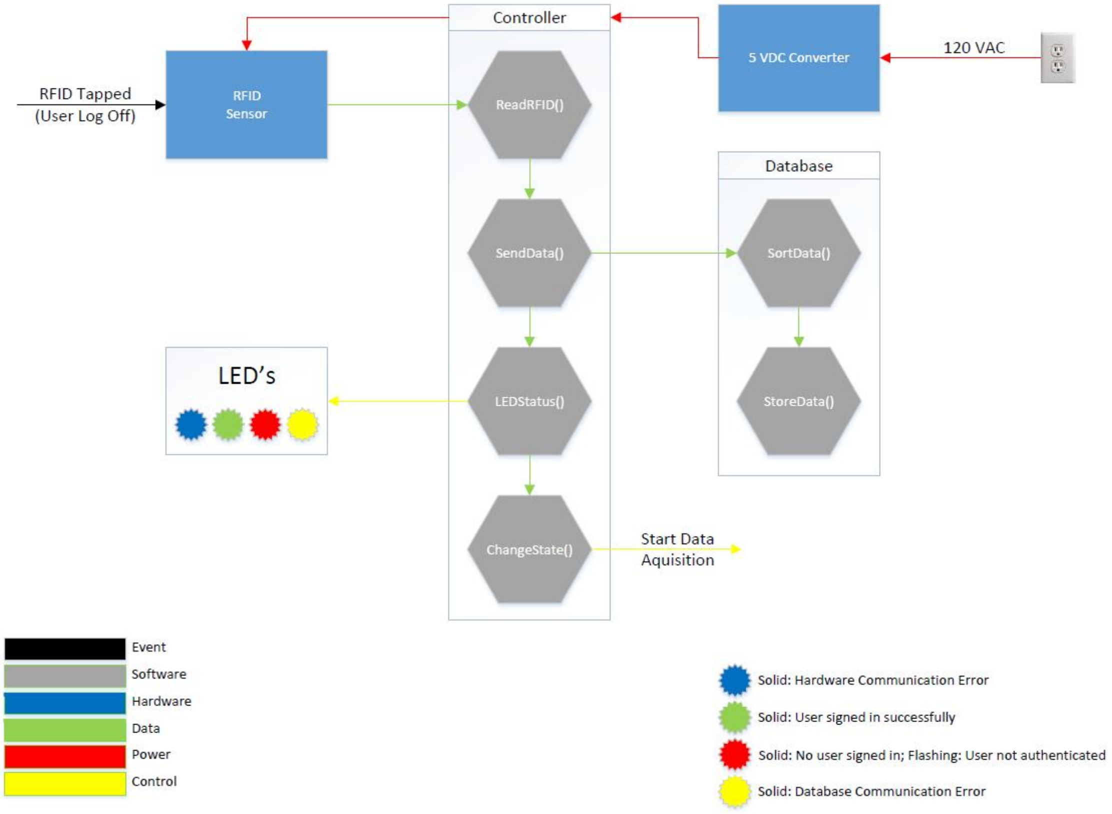


Figure 3: User Exit Use Case

1.6.3 Data Management

The main functionality use case is data management. Once the user is authenticated the microcontroller pulls data from the weight sensor at a rate of 100ms. The microcontroller is simultaneously pulling data from the proximity sensor at a rate of 100ms. Once the microcontroller receives the signal that the weight is no longer at home position it will store the last recorded weight value. Once the sensor has indicated the weight has left home position the microcontroller will begin to pull data from the infrared sensor and determine individual repetitions based on the distance the weight has traveled from home position. This process will repeat once the proximity sensor senses that the weight has returned to home position for 5 seconds. This signal will also trigger the microcontroller to group the previous repetitions into a set and send the set to the database. If the proximity sensor shows the weight is at home position for more than 2 minutes the system will log the user out of the system and once again send any remaining repetitions to the database as a set. If network connection is lost mid exercise the microcontroller will hold the data for the last set performed and the yellow LED will light indicating the system will not track any more sets. Once the connection is restored the microcontroller will send the stored data to the database and the system will reset to once again look for a user. These steps are diagram in Figure 4.

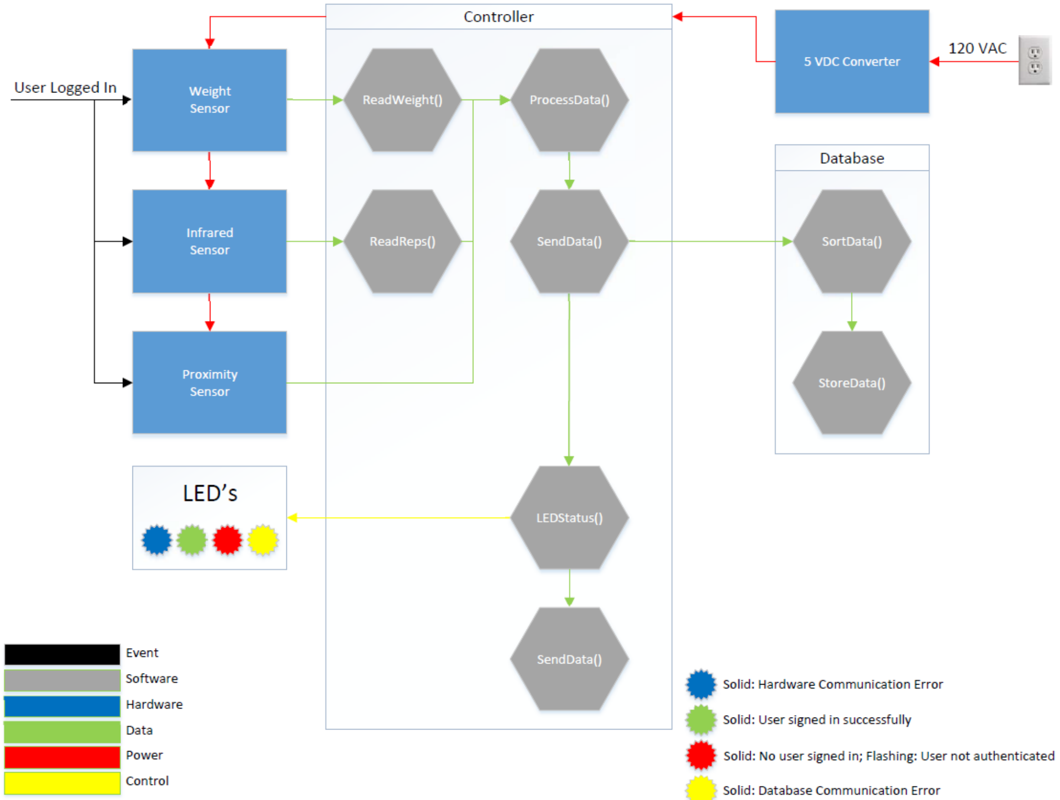


Figure 4: User Performance User Case

1.6.4 Data Management and Display

The final use case describes how a user or trainer communicates with the system. If so chosen a third party such as a personal trainer can access the database and implement suggestions for weight and repetitions for each set of the exercise for a particular user on that day. This data will be available to the user through the smartphone application any time after this initial entry. Once a user performs their work out their weight and repetition data for each set will display adjacent to the suggested set data from the trainer. Each set will be sent from the microcontroller to the database and pulled to the smart phone when either the weight is at home position for 5 seconds or a user is signed out and sets will be displayed in chronological order. The third party will then have access to this data to evaluate performance and make future suggestions for the user. These steps are diagram in Figure 5.

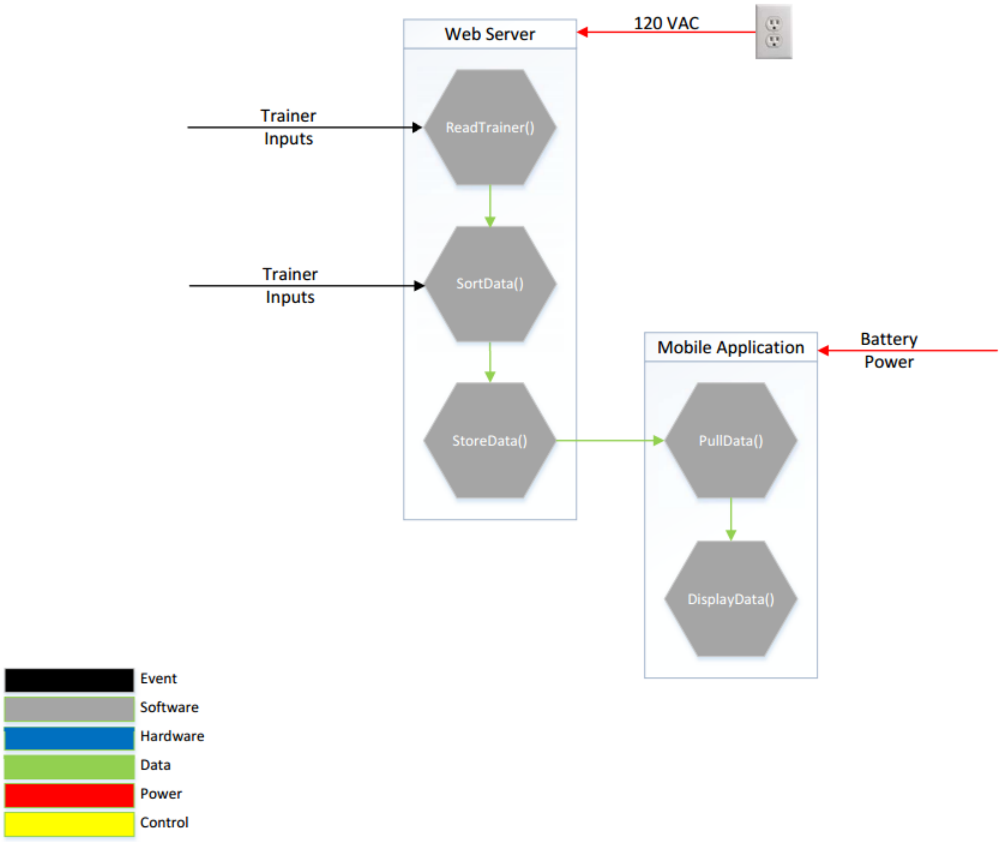


Figure 5: Data Management Use Case

2 Design Requirements Specification

Table 1: Design Requirements

Marketing Requirements	Engineering Requirements	Justification
1	The system will be able to operate for 4 hours in the case of line power failure.	In a normal gym setting machines are in use less than 50% of the time. This gives the system a minimum of 8 hour operating time which would allow the system to operate a sufficient amount of time until AC power can be restored.
2,6	The system will recognize when connection to the database is lost and notify the user. If the system loses connection in the middle of a set the performance data from that set only will be stored locally until connection is restored.	Due to the possibility of wireless connection being lost. There is a need for indication to the user that their performance is not being tracked.
2,6,7	The system will be able to identify a specific user, and indicate whether a user is recognized or not.	Many different users will be using the same machine so the system needs to have a way to identify to whom the performance data belongs. The indication lights make it clear to the user if their attempt to sign in was successful.
2,6,7	The system will be able to sign out a user at the conclusion of an exercise, and send any leftover set data for the specific user to the database.	Due to many different users using the same machine a user needs the ability to ensure that a separate user's data is not falsely recorded as their data due to the second user's failure to properly sign in. Also if a user is to sign out before five seconds has passed their data is still assured to be recorded for the previous set.
2,6,7	The system will be able to sign in a user on a machine that is currently active to another user and send the current users data to the database.	There could arise an occasion where a user would forget to sign out of the system. Because of this the system should always be tracking data for the last user to sign in.
2,4,6,8	The system will be able to detect when the weight is lifted more than one foot from a home position and log this as one repetition.	One foot should be a significant distance from home position to constitute a repetition being performed and not inadvertent

		movement.
2,5,6,7	The system will recognize when the machine has been inactive for 5 seconds and group the previous repetitions into a set and send the weight and repetition data for each set to the database.	When a current user rests during a set it will usually be less than 5 seconds. Any duration of inactivity longer than this would indicate the user is performing a separate set of repetitions.
2,4,5,6,7	The system will recognize when the machine has been inactive for more than 2 minutes and log the current user out sending their performance data to the database. A red indication light will be lit indicating the machine is free for another user.	A typical rest between sets lasts less than 2 minutes. Any duration longer than this would indicate that a user may have forgotten to sign out of the system. If a user does rest longer than 2 minutes the red light would indicate that they need to sign in again in order to record new data.
2,6,7,8	A third party will have access to the database through a web server and have the ability to suggest weight and repetition data for sets which will display adjacent to the actual data in table format.	Many people do not have a significant knowledge of workout strategies. This system allows a trainer to monitor many clients performance without spending hours in the gym with each.
2,8	The user will be able to access their performance data from an android application which will pull the data from a database.	In order for the system to be used by most common gym users the data must be easy to access from a convenient to use smart phone application.
Marketing Requirements		
1. The system should be power efficient 2. The system should be easy to operate 3. The system should detect the weight used 4. The system should detect the number of repetitions performed 5. The system should detect the number of sets performed 6. The data should be accurate 7. The data should be user specific 8. The data should display to the user in a convenient manner		

3 Accepted Technical Design

3.1 Hardware

The main controller, shown in Figure 6, takes the incoming data from the system, compiles the data, and sends the compiled data to an outside server database. This data comes from the RFID, weight, infrared, and proximity sensors inputs. The information from the database is then read from a web server or a mobile phone application. The controller runs on 5 volts DC, and also distributes a reference voltage to the sensors.

Figure 6: Hardware Block Diagram (Level 0)

Figure 7 shows a higher level breakdown of the system. The analog voltages of the RFID and infrared sensor are inputted, while the weight sensors voltage requires amplification. The power supply is a 120 volt AC to 5V DC converter which runs through a rechargeable battery which serves as a backup supply when unplugged. The software program within the controller takes the compiled data and sends it through the Wi-fi adapter to a computer with a SQL database. LED lights are also turned on and off to display machine status. A user can view their individual data from the smart phone application which ties to the database.

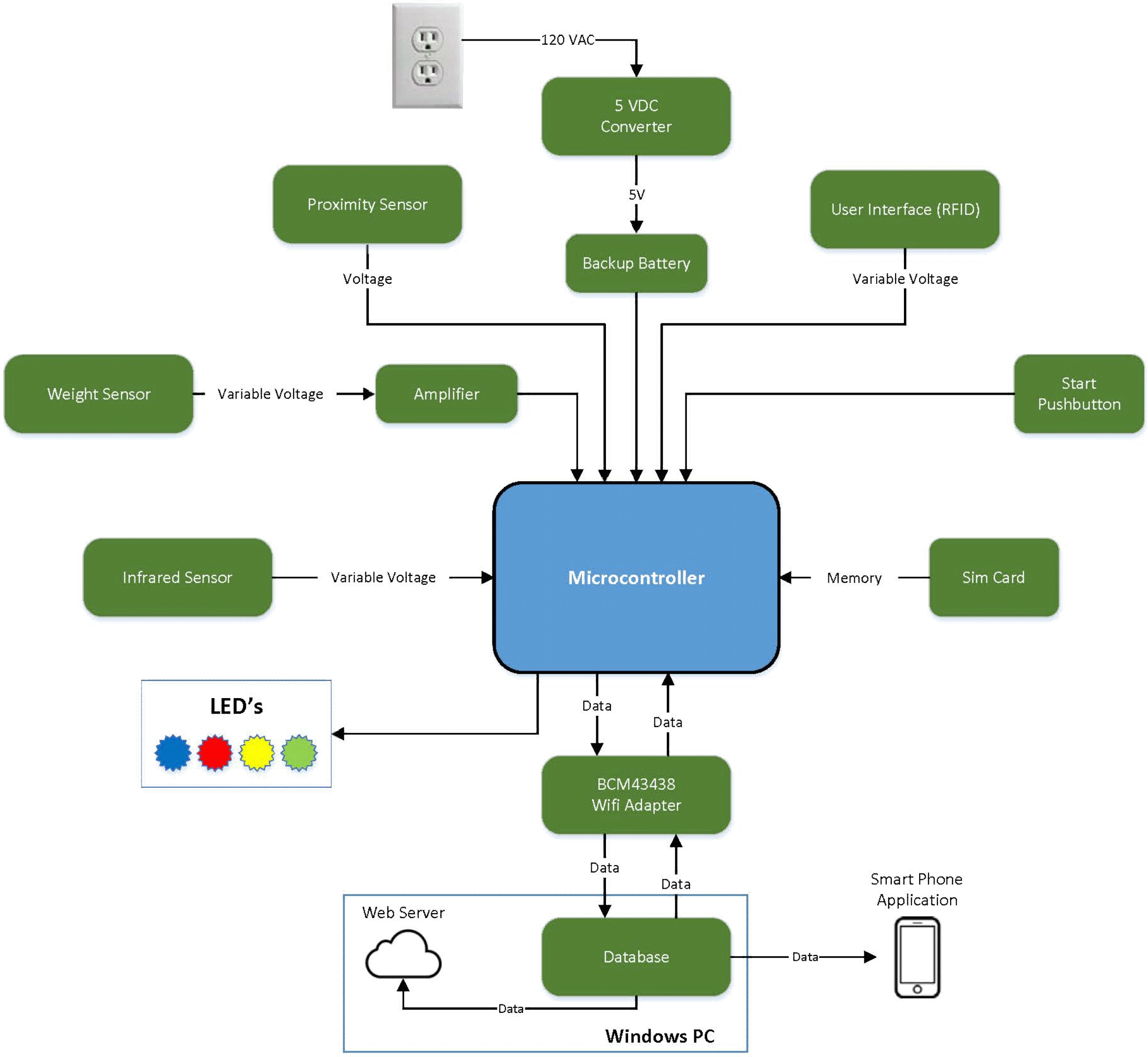


Figure 7: Hardware Block Diagram (Level 1)

3.1.1 RFID Sensor (By Jordan Pearce)

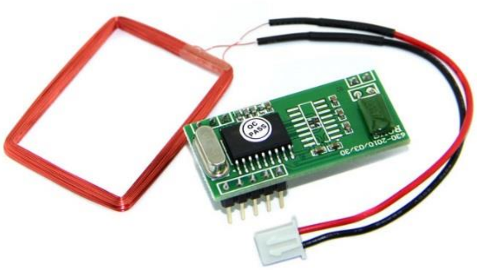


Figure 8: RFID RFR101A1M eio.com

We will be using the passive RFID shown in Figure 8. This system has a tag associated with the RFID and has a read zone determined by the location of the coil shown in the image. The tag will wait for a signal from the reader. The reader sends energy to the tag and converts it into an RF waveform. This generates a signal back to the RF system located on the Smart Fitness Machine. This is called back-scatter, it's a change in the electromagnetic wave. This is detected by the reader if the tag is in close enough proximity. Each read of a tag will identify different users. Reading the same tag twice will sign out that user. Once the RFID of the individual is tagged in it will begin to record/count the repetitions. After the user is finished, he or she will tag out. Then the data recorded is sent to that user. This will allow each users workout data to be sent to the correct person once the workout is finished. In summary, the RFID is verifying each user and telling the system to begin calculating, collecting, and organizing data. The RFID runs off of 5 VDC and distributes the signal to the microcontroller using a TX and RX connection. The connection for the RFID to the microcontroller is shown in Figure 9.

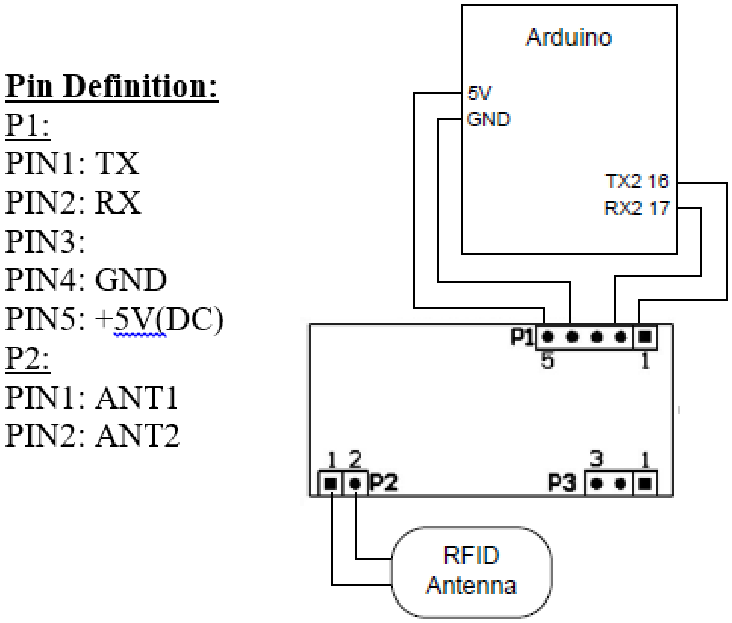


Figure 9: RFID to Arduino Connection

3.1.2 LED Lights

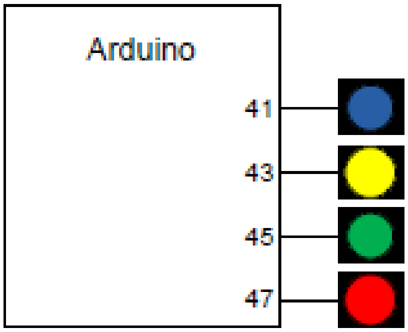
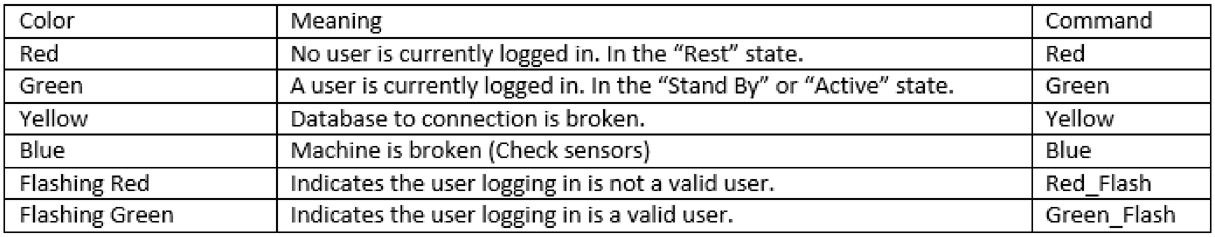


Figure 10: Arduino to LED Lights

The machine will have 4 LED lights on it with colors blue, yellow, green, and red. The lights will communicate to the user what state the machine is in. Each LED will represent a different state the machine is in. The LED's will have individual nameplates to identify which color will mean what. The pins used to turn on these LEDs are given in Figure 10. The various states and colors are shown in Table 2.

Table 2: LED Definition Table



Color	Meaning	Command
Red	No user is currently logged in. In the “Rest” state.	Red
Green	A user is currently logged in. In the “Stand By” or “Active” state.	Green
Yellow	Database to connection is broken.	Yellow
Blue	Machine is broken (Check sensors)	Blue
Flashing Red	Indicates the user logging in is not a valid user.	Red_Flash
Flashing Green	Indicates the user logging in is a valid user.	Green_Flash

3.1.3 Power Push Button

Figure 11:Ulincos Latching Pushbutton Switch U22A4 Amazon.com

There is going to be a push button capable of turning the system on and off. The push button will be located on the exterior of our box. There will be an LED light within the push button as shown in figure e XX??. Light on, will indicate the system is on. Light off, the system has powered down. The connection for this device is shown in Figure 12.

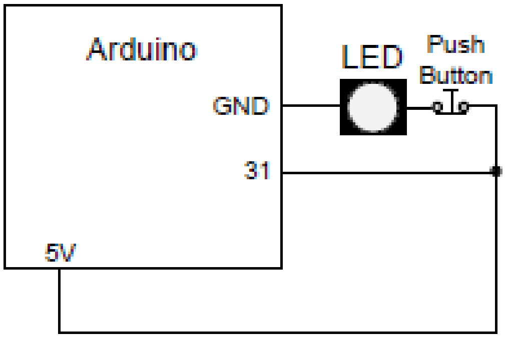


Figure 12: Push Button to Arduino Wiring

3.1.4 Junction Box

Figure X: AMP864NL - Allied Moulded JIC Size Junction Box wistexllc.com

A junction box will be mounted on the back of the machine. The box will house the microcontroller, backup battery, and all wiring terminal connections. This protects the microcontroller and allows easy connections for an implementer. The outside of the box will have the LED push button on the outside to turn the system on and off as discussed in the previous section. The box dimensions are shown in Figure 13.

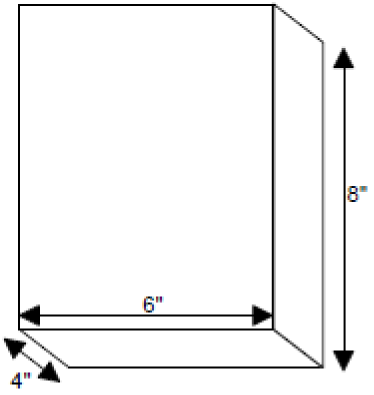


Figure 13: Junction Box Dimensions

3.1.5 Weight Sensor (By Tyler Masters)

Figure 14: TAS606 Load Cell from Sparkfun.com

The weight sensor we plan to use is a TAS606 button type load cell shown in Figure 14. This smaller profile will allow the load cell to be conveniently mounted on the stop of the machine which the weight rests. A 200kg load cell will still provide a small apparatus while giving us a more robust sensor that will be able to handle the force of weights possibly being lowered quickly. This sensor will be a combination of 4 load cells contained in a Wheatstone bridge configuration as shown below in Figure 15.

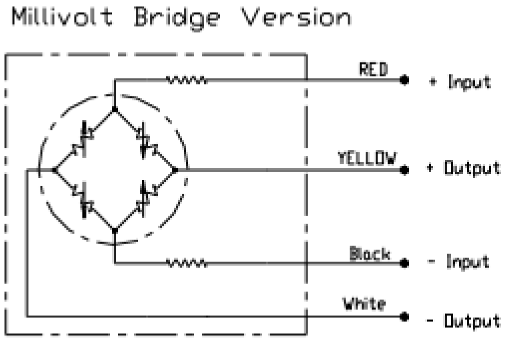


Figure 15: Wheatstone Bridge Schematic

This configuration will be able to sense the change in resistance from the load cells and translate that output to be an analog voltage of 1 to 2 mV per volt of input. The output of the bridge will be a small analog voltage value which will not be large enough to be accurately read by the Arduino and will need to be amplified. To do this we will connect the output to a load cell amplifier before connecting the device to the microcontroller. Due to the high precision and low noise necessary for the Arduino to convert the variable voltage to numeric values we will use a prepackaged amplifier breakout board designed for this specific purpose. The amplifier we have chosen is the HX711 as shown below in Figure 16.

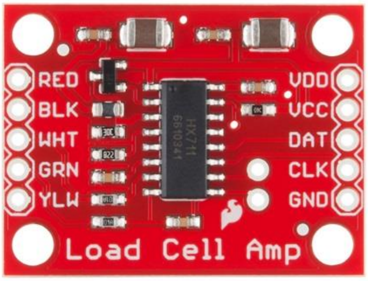


Figure 16: HX711 Load Cell Amplifier from Sparkfun.com

This will convert the analog voltages to digital values that we will read as weight values to the nearest pound. Due to the fact that our values of weight will be rounded to the nearest pound we are able to be more lenient on error factors such as hysteresis, creep, non-linearity, and zero balance. Due to this we should be able to deal with total combined errors of the sensor to be .5%. The connection for the Load Cell, Amplifier, and Microcontroller are shown below in Figure 17. The red, black, white, and green wires coming from the load cell will connect to their designated terminals on the HX711. The analog voltage to power the load cell (VCC) and the digital supply voltage (VDD) will both be connected to the 5V pin on the Arduino. The data and clock signals will be connected to two of the Arduino's digital pins.

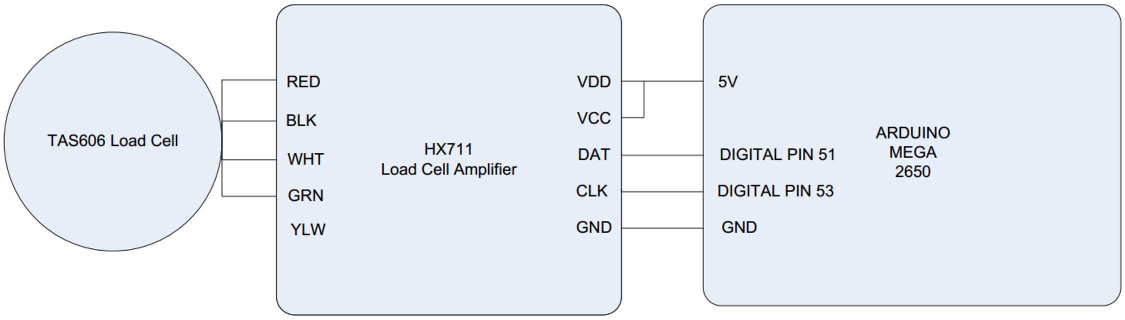


Figure 17: Load Cell Hookup to Arduino

3.1.6 Proximity Sensor (By Tyler Masters)

Figure 18: KMS-23 Magnetic Reed Switch from Sparkfun.com

The proximity sensor will tell the microcontroller when the weight is at a home or rest position. After research it was found that the best method to perform this task is to use a magnetic proximity sensor. The initial idea was to use an inductive type proximity sensor due to the fact that the slide to which the weight is applied is metal. However it was determined that due to the amount of surrounding metal on the apparatus it would be a more reliable option use a magnetic sensor and a permanent magnet. There are two types of magnetic proximity sensors that could be used for this application. A reed switch or a Hall Effect sensor. Due the fact that our output only needs to be a binary high or low voltage we will use the Reed Switch. A Hall Effect sensor could perform the same task however its output is in analog form which would be unnecessary for our use. The Reed switch operates as a normally open contact that will close when a magnetic field is applied. Therefore by placing a permanent magnet on the weight slide and the switch just below where the weight will come to rest, we will be able to receive a high voltage when the weight is at its home position and a 0 voltage when it is in motion. The KMS-23 sensor we plan to use is a permanent magnet and switch in a package as shown in Figure 18. The connection of the switch to the microcontroller is shown below in Figure 19 . The two leads from the sensor will connect to a digital pin and ground.

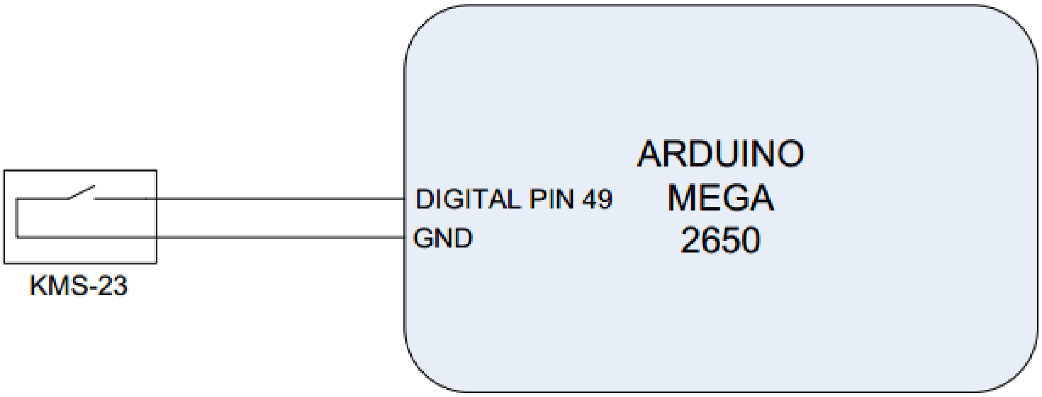


Figure 19: Proximity Sensor Hookup to Arduino

3.1.7 Infrared Sensor (By Tyler Masters)

Figure 20: GP2Y0A02YK0F Infrared Sensor from Sparkfun.com

Through research it was found that there are many different methods of distance detection. These many options were narrowed down to two separate methods which would work well for our specific project application. The first of which is sonar distance sensing which has greater distance range but also has a larger less precise target area. Due to the fact that we will be looking to sense the motion of a relatively small object on an apparatus that will have many different moving parts it was decided that infrared sensing would be best due to its more precise target area. The chosen sensor is shown in Figure 20. A block diagram of the sensor chosen is shown below in Figure 21.

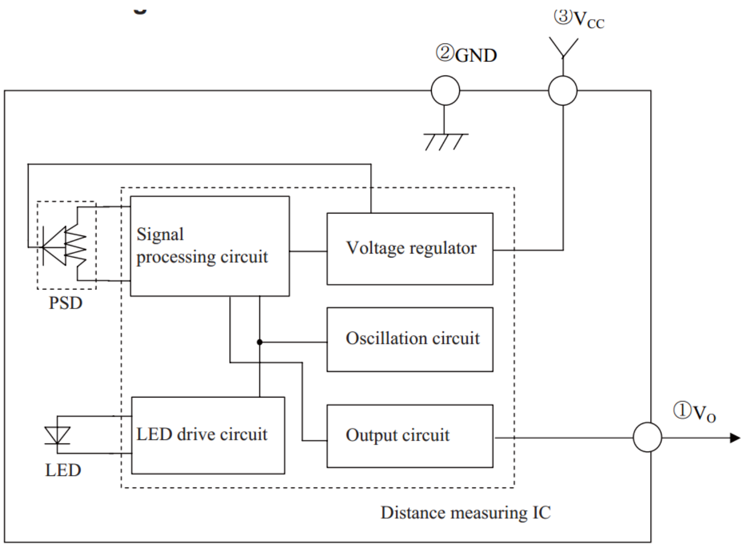


Figure 21: Infrared Sensor Block Diagram from GP2Y0A02YK0F datasheet

The range for this infrared sensor can go up to 150cm which will easily suffice for the motion of the weight. This sensor will output an analog voltage between 0 and 5V which will be calibrated within the Arduino using known values to correspond to distance which the weight has moved. A typical output curve is shown below in Figure 21. To calibrate we will use our own tested values, however this curve shows that from 20 to 80cm which will approximately be our range of detection the curve is steep enough for us to achieve relatively high accuracy in our measurements.

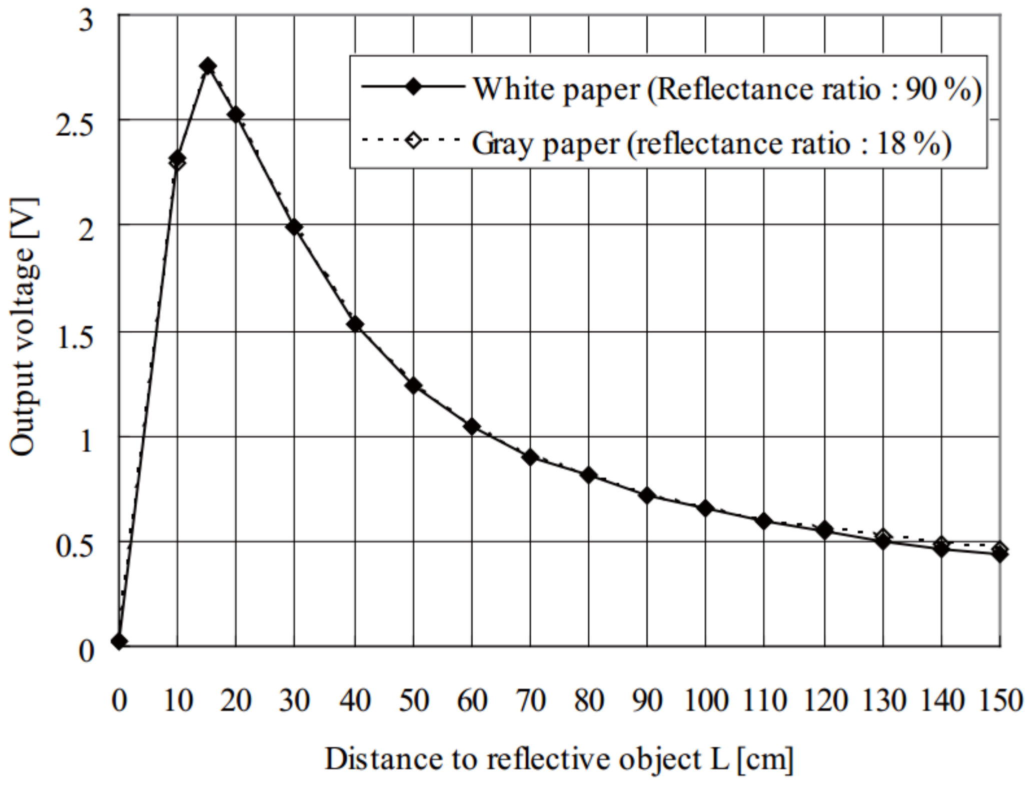


Figure 22: Graph of Output Voltage to Distance from GP2Y0A02YK0F datasheet

The connection of the sensor to the microcontroller is shown below in **Error! Reference source not found.**. The sensor will take in 5V for power and the Arduino will read the output through an analog pin.

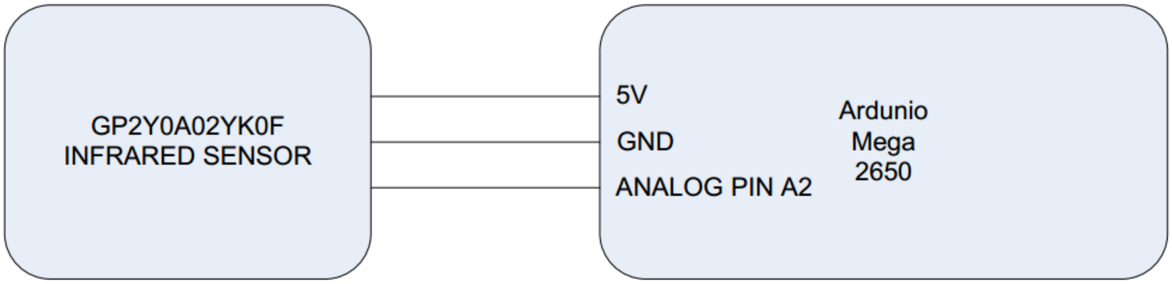


Figure 23: Infrared Sensor Hookup to Arduino

3.1.8 Wi-Fi Chip (By Tyler Masters)

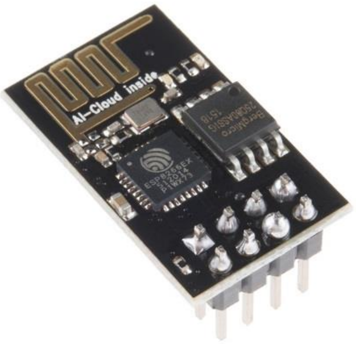


Figure 24: ESP8266 Wi-Fi Chip from Sparkfun.com

The Wi-Fi Chip which we chose is the ESP8266 shown in Figure 24. This chip is widely known to be very compatible with Arduino. This chip runs at 80 MHz, has 64KBytes of instruction RAM, 96Kbytes of data RAM, and 64Kbytes of boot ROM. The chip uses 2.4 GHz Wi-Fi and supports WPA/WPA2 and has an integrated TCP/IP protocol stack among many other specifications. The chip also has a very small profile (11.5mm x 11.5mm) which will serve well for keeping our design footprint smaller. On the power side the chip has a deep sleep mode that draws <10uA and a power leakage current <5uA. It has the capability to wake up and transmit in <2ms with a standby power consumption of < 1.0mW. This product has features that far exceed what is necessary for the small scope of our project and should be able to work seamlessly with our design.

The chip contains 8 pins GPIO_1, GPIO_2, Tx, Rx, 3.3V, Reset, Ground, and Chip Power down. For our purposes the GPIO and Reset pins will not be connected. The chip reset will be connected to the 3.3V supply. The connections are shown below in Figure 25.

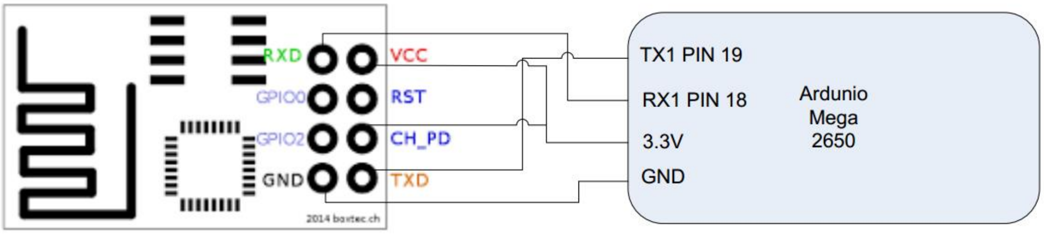


Figure 25: Wi-Fi Chip hookup to Arduino

3.1.9 Hardware Theory of Operation

The microcontroller will act as an intermediary between the user and the Smart Fitness Machine. The microcontroller will receive multiple analog and digital voltages from various sensors outfitted on the machine. A passive RFID sensor will send a signal to identify a specific user. A load cell will send an analog voltage which will be calibrated and translated into weight values. A magnetic reed switch will act as a proximity sensor and send a digital voltage representative of whether the weight is at a home position or not. An infrared sensor will send an analog voltage which will again be calibrated to show distance from its home position which the weight has traveled. The microcontroller will analyze the data from these and compile the weight used and repetitions performed for each set the specific user does as well as light specific LED's that will tell the current status of the system. The microcontroller will then send this data to a web database from which it will be accessible.

3.1.10 Hardware Design Modules

Module	**Microcontroller**
Designer	Tyler Masters, Ben Waters, Jordan Pearce, Andrew Thornborough
Inputs	• 5V: Input power to the microcontroller from the power supply • RFID Reader: Low Voltage (Dependent on RFID Wristband scanned) • Weight Sensor: Low Voltage (Dependent on Weight) • SIM card: Memory for the Microcontroller • Proximity Sensor: High or Low Signal • Infrared Sensor: Low Voltage (Dependent on Weight Slide Position) • BCM43438(Wi-Fi Chip): User Authentication from Database
Outputs	• Multicolor LED's: Display of current system state • BCM43438(Wi-Fi Chip): Weight and Repetition data to Database
Functionality	The microcontroller reads data from the RFID receiver and searches the database to authenticate the user. It then lights a certain LED indication light depending on whether or not the user was authenticated or whether connection to the database was unsuccessful. Once authenticated the microcontroller reads data from the weight sensor and proximity sensor simultaneously. Once the weight has been lifted the weight is recorded and the microcontroller records data from the infrared sensor to determine the number of sets performed. The microcontroller stores weight and repetition data until the data from the proximity sensor shows the weight has been at rest for 5 seconds or the RFID sensor detects sign out from the system. The microcontroller will then send the stored data to the database grouped as a set.

Module	**RFID Receiver**
Designer	Jordan Pearce
Inputs	• 5V: Input power from the microcontroller • RFID Band: User Tag
Outputs	• User Identification Signal
Functionality	The RFID Receiver detects a signal from an RFID tag located in a wristband on the user and transmits it to the microcontroller to be processed.

Module	**Weight Sensor**
Designer	Tyler Masters
Inputs	• 5V: Input power from the microcontroller
Outputs	• Low Analog Voltage Signal
Functionality	The Weight Sensor sends a low voltage signal to the microcontroller which will be processed to obtain a number for weight used.

Module	**Proximity Sensor**
Designer	Tyler Masters
Inputs	• 5V: Input power from the microcontroller
Outputs	• High or Low Signal
Functionality	The Proximity Sensor will send either a high or low signal indicating whether the weight is at home position or not.

Module	**Infrared Sensor**
Designer	Tyler Masters
Inputs	• 5V: Input power from the microcontroller
Outputs	• Low Analog Voltage Signal

Functionality	The Infrared Sensor detects position of the weight and sends a signal to the microcontroller to be processed in order to determine number of repetitions.

3.1.11 Weight Machine Hardware Layout

Figure 26: Machine Layout (Front)

Figure 27: Machine Layout (Side)

3.2 Software

The Software is split into three parts: The microcontroller software, the database and webserver software, and the software for the Android Application. As seen in Figure 26 the Level 0 software block diagram shows that the microcontroller software will get the data from the various inputs and format the data to be sent to the database, then the webserver and Android application can pull the up to date data from the database and present it to the user. Thus the microcontroller indirectly updates the webserver and Android application. The software will also perform additional functions here not visualized in the block diagram. One of the function will be setting the LED to the correct state to present the state of the machine to the user. Another function will be conserving power; this will be done based on the amount of activity that it has seen for a given time period which dictate how much power should be consumed.

Figure 28: Software Block Diagram (Level 0)

Figure 27 shows a more detailed breakdown and display of the main code functions in the system. From the left the inputs are coming into a function called check state. One of the inputs will be from the user which will provide the user ID number via the RFID band read by the RFID sensor. The other inputs are from the system itself. These inputs are the weight sensor, the inferred sensor and the proximity sensor. The weight sensor will need an amplifier so we can increase the gain so the software can tell the difference between weight using an Analog to Digital Convertor.

These inputs will go into the main function of the microcontroller called Check_State. This function is a big state machine that will update based on the conditions of the inputs. Thus it will switch states if the given input is correct for Example if one state is called State_Rest this state is the state where there is no login from a user. In this state the microcontroller will check the RFID signal every 50 ms while going to sleep while ignoring the rest of the sensors. The reason for this is it will help with battery life because it can ignore 3 inputs and be asleep 90% of the time. However, if the user logs in the signal will be sent to the microcontroller and switch to an active state to read the other sensors. This will continue indefinitely, switching states, going to low power mode, sending data, and receiving data. The data is then sent to the database which is a SQL database. This data will get normalized to the third form for data integrity. Then a PHP script will gather the data and send it to a webserver. Were the Android Application will pull the data from the server and then organize it for display on a smart phone.

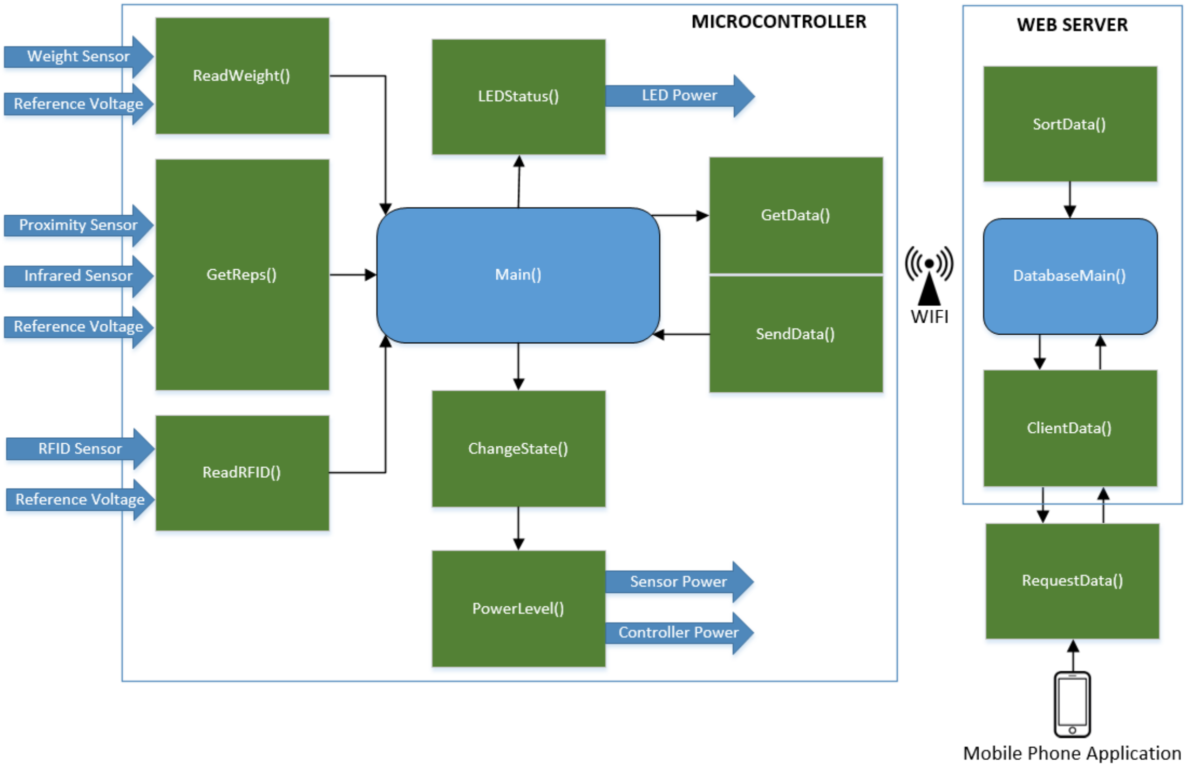


Figure 29: Software Block Diagram (Level 1)

In the level 2 Software diagram the main function was further broken down to specific functions it may perform internally. These functions are the ValidateUser(), ProcessData(), and AddCounter(). Also the Webserver functions were grouped into a main function due to the fact that these functions may be performed in several different orders depending on what is being asked for. A function for trainer input was also added to this function. The mobile phone application was further broken down into the two main functions it will be performing, PullData() and DisplayData().

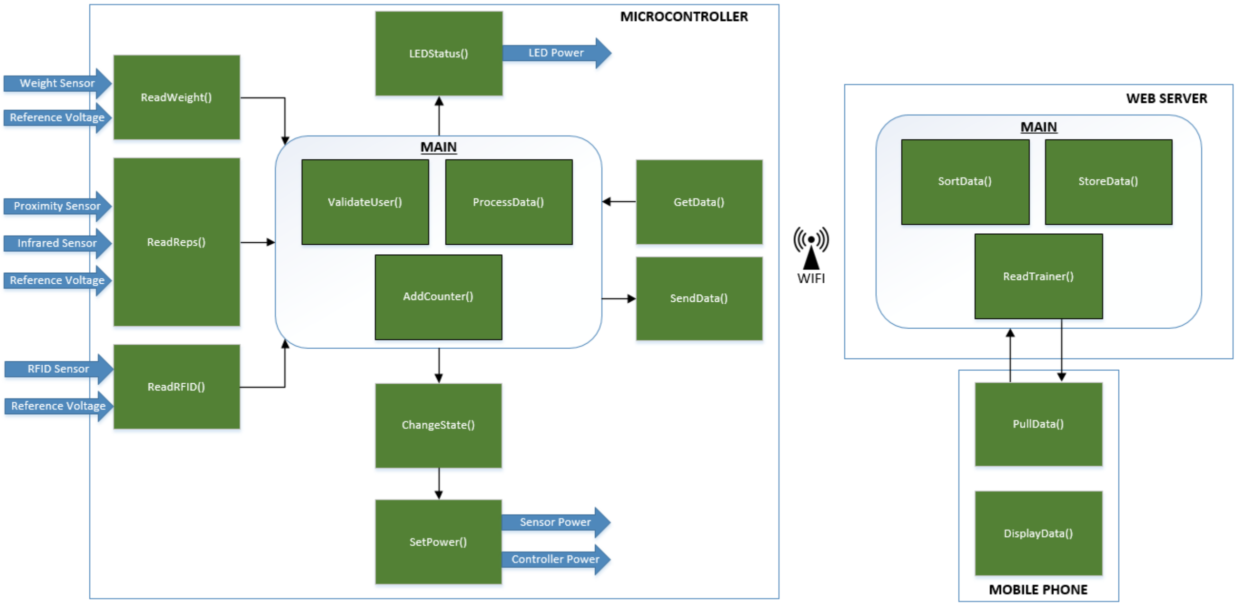


Figure 30:Software Block Diagram (Level 2)

3.2.1 Database (By Andrew Thornborough)

The database will be on windows PC using MySQL as our database management software. Each user will have a specific ID that will be the unique key for that profile. Each profile will hold a history up to 65,535 logins and each login is 20 Bytes. This in total will account 1.37 Megabytes per user and per machine. The data from the database will have the type of message it is sending to the PC, the ID of the machine and the user, the time and date of each session, the weight, rep, and set of the session, and finally if the user has passed the required amount of weight and reps based on an amount that was provided to the database. To access the database for the webserver and android application, MySQL will be called by PHP as a backend scripting language. This will format the data and give it to the web server or app so it can be presented in a neat and legible way. The database will use a relational model to handle the data which will be normalized by the 3rd form to avoid inaccurate data. The main tables for the normalization will be the user basic profile, the machine that is connected to each microcontroller, and the data from a session based on the date. The Embedded system will communicate to the PC via Wi-Fi and update the database with the user's data after a session.

3.2.2 Smartphone Application (By Ben Waters)

The smartphone application will be developed in Android Studio. This application consists of two or more screens for ease of use. This account is tied to the Smart Fitness Machine and will display the user's performance data in a convenient manner. A user can also view specific instructions from a trainer or physical therapist to ensure the highest quality workout. A fitness trainer or physical therapist can log in to the web server in order implement these instructions for a user.

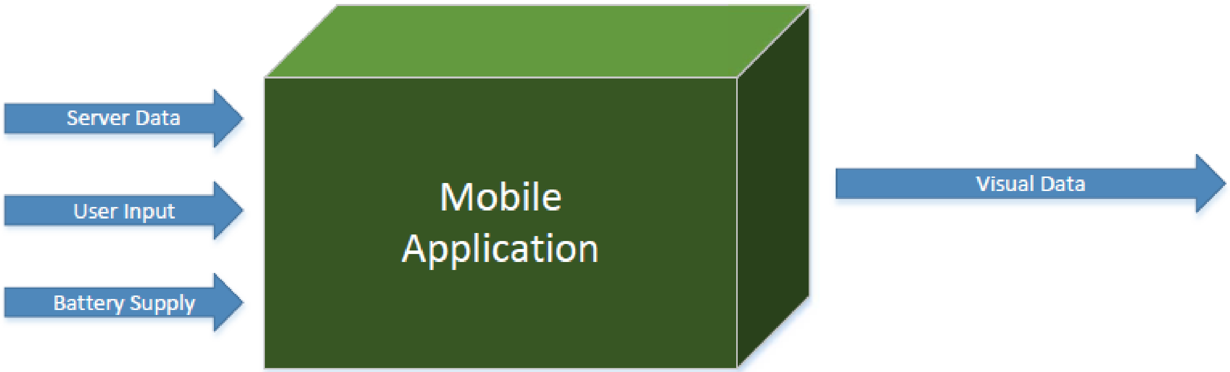


Figure 31: Mobile Application Block Diagram Level 0

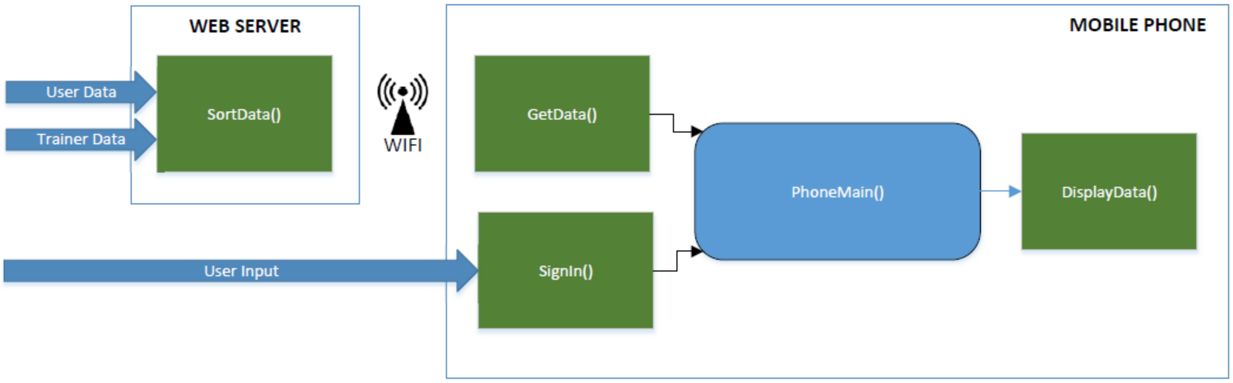


Figure 32: Mobile Application Block Diagram Level 1

3.2.3 Software Design Modules

Module	**Database**
Designer	Andrew Thornborough
Inputs	• Wi-Fi: Weight and Repetition data from Microcontroller • Wi-Fi: User Identification Signal from Microcontroller • Trainer Suggestions from Webserver
Outputs	• None
Functionality	The Database stores performance data for each particular user along with performance suggestions from a trainer inputted through a webserver. The microcontroller pulls user identification information from the database. The Smartphone app will pull user performance data from the database.

Module	**Smartphone App**
Designer	Ben Waters
Inputs	• Wi-Fi: Weight and Repetition data from Database
Outputs	• Display
Functionality	The Smartphone app pulls user performance data from the database along with trainer suggestions for that particular user and displays them side by side as soon as each set is completed.

3.3 Design Calculations

Due to the fact that we will be using a USB in line power pack to provide our system the capability to maintain operational in the case of power failure, it is necessary to calculate an estimate of the minimum size of this device in order to keep the device operational for the necessary time of 4 hours. Due to the nonlinear decay of batteries a simple calculation of rated battery capacity (mAh) divided by the discharge current (mA) will not return an accurate estimate of the size of battery needed. In order to be accurate Peukart's equation must be used as shown in Equation 1

Equation 1

$$C_p = I^k t \ (---)$$

The equation can be rewritten in order to solve for the time the battery takes to fully discharge as shown in Equation 2.

Equation 2

$$t = H\left(\frac{C}{IH}\right)^{k} (--)$$

In our scenario we are looking to solve for “C” representing the rated capacity of the battery so once again the equation is rearranged as in Equation 3.

Equation 3

$$C = (\frac{t}{H})^{1/k} * IH(---)$$

The variables in this equation are once again “C” representing the rated capacity of the battery at a given discharge rate given in milliamp hours (mAh), “t” representing the time for the battery to fully discharge, “H” representing the rated discharge time of the battery in hours, “I” representing the discharge current, and “k” representing Peukart’s constant. The values of “H” and “k” come from the data sheet of a specific battery so in order to obtain an estimate on the size of the battery which we need, we will take common values for these. Most batteries are tested at 20 hours therefore we will take the value of H to be 20. A common value for Peukart’s constant is 1.2. In order to give ourselves a safety buffer we assume that the battery will only discharge 60% in the minimum 4 hours. Therefore we will use a time value of 6.67 hours. Lastly, after research, it was found that a high amount of running current for an Arduino performing functions is 750mA. Our system is not expected to draw this high of a current but as a precaution this value was used for our calculation. The calculation was carried out as shown below in Equation 4

Equation 4

$$C = (\frac{6.67}{20})^{1/1.2} * (750)(20)(---)$$

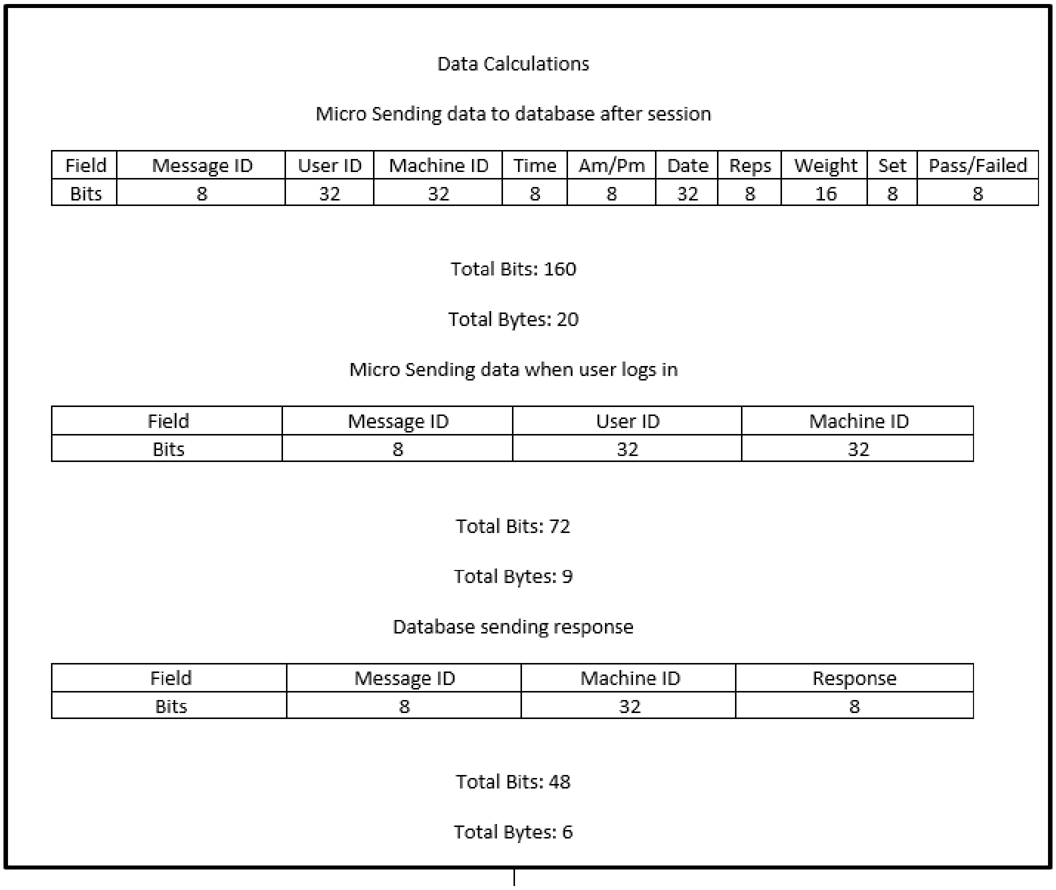
$$C \approx 6000mAh$$

In the engineering requirements specified for the project it was determined that the system should be able to operate on battery backup for a minimum of 4 hours. The simplest way to achieve this is to use an inline USB battery pack. It was calculated in section ______ of this report that the battery must have a minimum rating of 6000mAh to safely achieve this requirement. The Arduino board can accept power a number of different ways. Due to the fact that most USB power packs are for 5V applications and the voltage is regulated we will use the USB input of the Arduino which accepts 5V and delivers it directly to the 5V rail without going through the built in voltage regulator. The battery pack will be fed from an AC to DC converter which takes in 120V 60Hz AC voltage and delivers 5V DC to the pack. The pack also will feature built in fusing and thermal protection. The pack must have the capability of simultaneously charging and discharging so as to keep the battery fully charged until there is an outage of power.

3.3.1 Additional Software Calculations

Calculations were done to determine how much memory would be required for storage on the microcontroller. These calculations include amount of bits sent to the microcontroller after a user's session and when a user logs in to the system. They also include the amount of bits sent from the database as a response to the microcontroller.

Table 3: Memory Calculation



Data Calculations

Micro Sending data to database after session

Field	Message ID	User ID	Machine ID	Time	Am/Pm	Date	Reps	Weight	Set	Pass/Failed
Bits	8	32	32	8	8	32	8	16	8	8

Total Bits: 160

Total Bytes: 20

Micro Sending data when user logs in

Field	Message ID	User ID	Machine ID
Bits	8	32	32

Total Bits: 72

Total Bytes: 9

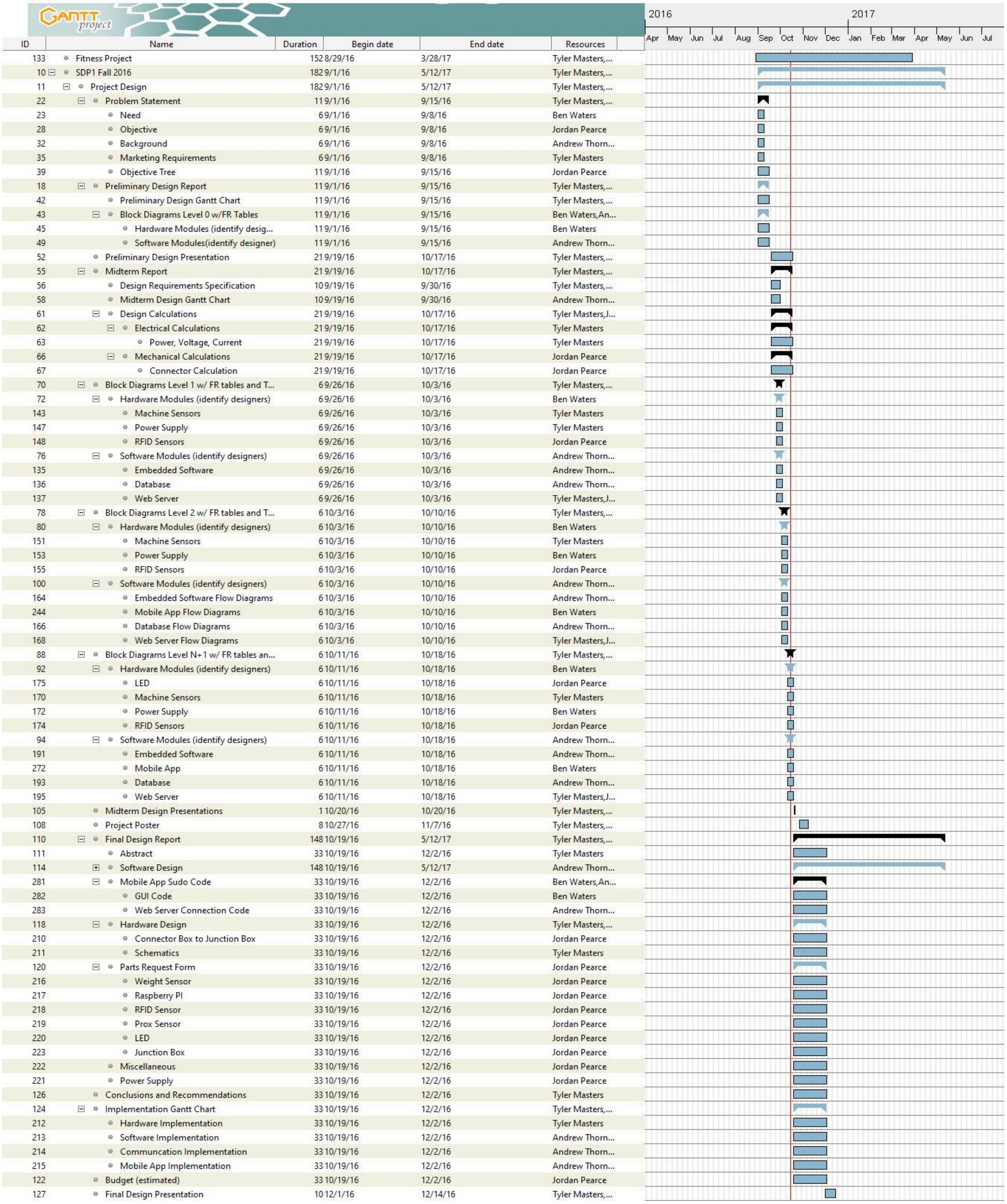
Database sending response

Field	Message ID	Machine ID	Response
Bits	8	32	8

Total Bits: 48

Total Bytes: 6

4 Final Design Gantt Chart



ID	Name	Duration	Begin date	End date	Resources
133	Fitness Project	152	8/29/16	3/28/17	Tyler Masters,...
10	SDP1 Fall 2016	182	9/1/16	5/12/17	Tyler Masters,...
11	Project Design	182	9/1/16	5/12/17	Tyler Masters,...
22	Problem Statement	11	9/1/16	9/15/16	Tyler Masters,...
23	Need	6	9/1/16	9/8/16	Ben Waters
28	Objective	6	9/1/16	9/8/16	Jordan Pearce
32	Background	6	9/1/16	9/8/16	Andrew Thorn...
35	Marketing Requirements	6	9/1/16	9/8/16	Tyler Masters
39	Objective Tree	11	9/1/16	9/15/16	Jordan Pearce
18	Preliminary Design Report	11	9/1/16	9/15/16	Tyler Masters,...
42	Preliminary Design Gantt Chart	11	9/1/16	9/15/16	Tyler Masters,...
43	Block Diagrams Level 0 w/FR Tables	11	9/1/16	9/15/16	Ben Waters,An...
45	Hardware Modules (identify desig...	11	9/1/16	9/15/16	Ben Waters
49	Software Modules(identify designer)	11	9/1/16	9/15/16	Andrew Thorn...
52	Preliminary Design Presentation	21	9/19/16	10/17/16	Tyler Masters,...
55	Midterm Report	21	9/19/16	10/17/16	Tyler Masters,...
56	Design Requirements Specification	10	9/19/16	9/30/16	Tyler Masters,...
58	Midterm Design Gantt Chart	10	9/19/16	9/30/16	Andrew Thorn...
61	Design Calculations	21	9/19/16	10/17/16	Tyler Masters,J...
62	Electrical Calculations	21	9/19/16	10/17/16	Tyler Masters
63	Power, Voltage, Current	21	9/19/16	10/17/16	Tyler Masters
66	Mechanical Calculations	21	9/19/16	10/17/16	Jordan Pearce
67	Connector Calculation	21	9/19/16	10/17/16	Jordan Pearce
70	Block Diagrams Level 1 w/ FR tables and T...	6	9/26/16	10/3/16	Tyler Masters,...
72	Hardware Modules (identify designers)	6	9/26/16	10/3/16	Ben Waters
143	Machine Sensors	6	9/26/16	10/3/16	Tyler Masters
147	Power Supply	6	9/26/16	10/3/16	Tyler Masters
148	RFID Sensors	6	9/26/16	10/3/16	Jordan Pearce
76	Software Modules (identify designers)	6	9/26/16	10/3/16	Andrew Thorn...
135	Embedded Software	6	9/26/16	10/3/16	Andrew Thorn...
136	Database	6	9/26/16	10/3/16	Andrew Thorn...
137	Web Server	6	9/26/16	10/3/16	Tyler Masters,J...
78	Block Diagrams Level 2 w/ FR tables and T...	6	10/3/16	10/10/16	Tyler Masters,...
80	Hardware Modules (identify designers)	6	10/3/16	10/10/16	Ben Waters
151	Machine Sensors	6	10/3/16	10/10/16	Tyler Masters
153	Power Supply	6	10/3/16	10/10/16	Ben Waters
155	RFID Sensors	6	10/3/16	10/10/16	Jordan Pearce
100	Software Modules (identify designers)	6	10/3/16	10/10/16	Andrew Thorn...
164	Embedded Software Flow Diagrams	6	10/3/16	10/10/16	Andrew Thorn...
244	Mobile App Flow Diagrams	6	10/3/16	10/10/16	Ben Waters
166	Database Flow Diagrams	6	10/3/16	10/10/16	Andrew Thorn...
168	Web Server Flow Diagrams	6	10/3/16	10/10/16	Tyler Masters,J...
88	Block Diagrams Level N+1 w/ FR tables an...	6	10/11/16	10/18/16	Tyler Masters,...
92	Hardware Modules (identify designers)	6	10/11/16	10/18/16	Ben Waters
175	LED	6	10/11/16	10/18/16	Jordan Pearce
170	Machine Sensors	6	10/11/16	10/18/16	Tyler Masters
172	Power Supply	6	10/11/16	10/18/16	Ben Waters
174	RFID Sensors	6	10/11/16	10/18/16	Jordan Pearce
94	Software Modules (identify designers)	6	10/11/16	10/18/16	Andrew Thorn...
191	Embedded Software	6	10/11/16	10/18/16	Andrew Thorn...
272	Mobile App	6	10/11/16	10/18/16	Ben Waters
193	Database	6	10/11/16	10/18/16	Andrew Thorn...
195	Web Server	6	10/11/16	10/18/16	Tyler Masters,J...
105	Midterm Design Presentations	1	10/20/16	10/20/16	Tyler Masters,...
108	Project Poster	8	10/27/16	11/7/16	Tyler Masters,...
110	Final Design Report	148	10/19/16	5/12/17	Tyler Masters,...
111	Abstract	33	10/19/16	12/2/16	Tyler Masters
114	Software Design	148	10/19/16	5/12/17	Andrew Thorn...
281	Mobile App Sudo Code	33	10/19/16	12/2/16	Ben Waters,An...
282	GUI Code	33	10/19/16	12/2/16	Ben Waters
283	Web Server Connection Code	33	10/19/16	12/2/16	Andrew Thorn...
118	Hardware Design	33	10/19/16	12/2/16	Tyler Masters,...
210	Connector Box to Junction Box	33	10/19/16	12/2/16	Jordan Pearce
211	Schematics	33	10/19/16	12/2/16	Tyler Masters
120	Parts Request Form	33	10/19/16	12/2/16	Jordan Pearce
216	Weight Sensor	33	10/19/16	12/2/16	Jordan Pearce
217	Raspberry PI	33	10/19/16	12/2/16	Jordan Pearce
218	RFID Sensor	33	10/19/16	12/2/16	Jordan Pearce
219	Prox Sensor	33	10/19/16	12/2/16	Jordan Pearce
220	LED	33	10/19/16	12/2/16	Jordan Pearce
223	Junction Box	33	10/19/16	12/2/16	Jordan Pearce
222	Miscellaneous	33	10/19/16	12/2/16	Jordan Pearce
221	Power Supply	33	10/19/16	12/2/16	Jordan Pearce
126	Conclusions and Recommendations	33	10/19/16	12/2/16	Tyler Masters
124	Implementation Gantt Chart	33	10/19/16	12/2/16	Tyler Masters,...
212	Hardware Implementation	33	10/19/16	12/2/16	Tyler Masters
213	Software Implementation	33	10/19/16	12/2/16	Andrew Thorn...
214	Communcation Implementation	33	10/19/16	12/2/16	Andrew Thorn...
215	Mobile App Implementation	33	10/19/16	12/2/16	Andrew Thorn...
122	Budget (estimated)	33	10/19/16	12/2/16	Jordan Pearce
127	Final Design Presentation	10	12/1/16	12/14/16	Tyler Masters,...

Figure 33: Final Gantt Chart

5 Design Team Information

Ben Waters, Electrical Engineering. Hardware Lead

Andrew Thornborough, Computer Engineering, Software Designer

Jordan Pearce, Electrical Engineering, Archivist

Tyler Masters, Electrical Engineering, Team Leader

6 Works Cited

[1] A.D. Greenburg, K.E. Camhi. “Fitness feedback system for weight stack machines.” U.S. Patent 5785632 A, Jul 28, 1998.

[2] B. Anderson, D. Nichols. “Exercise activity recording system.” U.S. Patent 7699752 B1, Apr 20, 2010.

[3] D. Salmon, D.R. Stubben. “System for monitoring fitness performance.” U.S. Patent 20140235409 A1, Aug 21, 2014.

[4] M.W. Brown, K.R. Lawrence, M.A. Paolini. “Exercise machine.” U.S. Patent 6702719 B1, Mar 9, 2004.

[5] R.D. Warner. “Computerized repetitive-motion exercise logger and guide system.” U.S. Patent 6669600 B2, Dec 30, 2003.

[6] Bluetooth Special Interest Group. “Bluetooth Technology To Change How We Play Sports, Exercise and More” Regional Business News. 2012

[7] E. Velloso, A Bulling, H. Gellerson, W. Ugulino and H. Fuks. “Qualitative Activity Recognition of Weight Lifting Exercises” 2013