Senior Design Project: Lapcycle

<u>End of Project Documentation</u>

Team 10:

J.R. Chadwick
Andrew Beltran
Lawrence Ly
Armando Fuentes

CpE 191 / EEE 193B
Professor Russell L. Tatro

May 5, 2016

Table of Contents

List of Figures

List of Tables:

Executive Summary:

Obesity has been a leading contributor to major health problems in nations such as the United States. Another notable trend that is occurring is the increasing use of major consumer electronics, such as smart phones, video game systems and laptops. With increasing usage of electronics adding their effects to modern day obesity, the crew behind Project Lapcycle attempts to tackle this issue by combining an indoors stationary bicycle together with a laptop. The display on the bike was replaced with specialized hardware which does two things: monitor the data incoming from the bike and to turn the bike from a mere mechanical machine into a semi-electronic machine. The Laptop connects to the hardware, and provides a display system showing off how the user exercises, and provides incentives to the electronically addicted crowd such that they want to ride the bike, gain exercise to lose weight, and still manage to work or play on their laptop. This “Lapcycle” is meant to provide a means for users to gain much needed exercise while maintaining their productivity or a sense of entertainment.

Abstract*: As citizens of First World nations become more dependent on computers for both their occupations and for their entertainment, the result is a decreasing amount of physical activity that has increased the rate of obesity (68% of American adults as of 2010* [27]) *and all the health consequences that it brings. The goal of this project was to create a device to help combat obesity by allowing people who lack time or motivation to leave their computers behind in order to exercise.*

By creating an exercise device that integrates the user's own personal laptop into an fitness machine, computer users of all ages will be more encouraged to exercise. While there exist products already that allow users to exercise while watching television, play music, or even operate a video game console, there is no devices that allows full use of their own personal computer and uses it to replace the performance readout standard on most moderately priced aerobic cardio machines. Allowing the user to continue operating their laptop computer while exercising is only part of the program that was written for the this prototype. The Lapcycle code also contains classes that are purposed to engage the user and make them wish to continue to use it for their exercise by saving their progress with a username and password and allowing them the ability to monitor their ongoing progress with achievements based on their performance metrics.

After two semesters, Team 10 has completed a deployable prototype of sufficient quality to demonstrate the principles they originally set out to accomplish. Starting from an uncomfortable repurposed mountain bike in Phase 1, the team has now built their prototype out of a much more comfortable and attractive recumbent spin bike with a mount to hold the user's laptop. The internal sensor is more accurate and provides better information than the original prototype by allowing the magnetic reed switch to receive two readings per revolution of the pedals.

Lapcycle is a five feature project consisting of three components. The base is a recumbent spin bike with a four point articulating laptop platform mounted to it. The controls system is a microcontroller connected to a sensor and a resistance servo. To make it all work, an application written in Java allows the user to monitor their workout in the same manner that they would on a standard spin bike. The prototype, at the conclusion of two semesters, is physically appealing, but with the team lacking the skills to produce costume computer peripherals with a hardware driver, the program is difficult to start and can not sample data at a high rate to improve accuracy. However, even with these drawbacks, the prototype proves the concept is sound and can be improved with further development.

Keyword Index: Exercise, Fitness, Java, Arduino, C, Obesity, Laptop, Spin Bike, Aerobic, Cardio

I. INTRODUCTION

Lapcycle was created in response to the growing problem of obesity caused by the increasingly sedentary lifestyle of Americans and other people of other first world nations. Team 10 decided to tackle the problem of obesity; specifically the inactive lifestyles and the overuse of computers in nearly every aspect of the modern First World citizen's life because of the many detrimental effects it has on those who are effected by it. The method of combating this problem derived by the team members involved creating a cardio device that one can operate while using their laptop or tablet computer.

Lapcyle is a portmanteau of the words laptop and bicycle and is used to illustrate the concept of combining the two activities together. The components of project Lapcycle are broken down into five features used to separate the requirements of the project into separate tasks. **Feature 1**: single person spin bike, describes the machine base as a single person spin bike and contains the bike base used to mount all of the other features and components. The device used is a Marcy Me 709 Recumbent Exercise Bike [19]. Rather than build this component, the existing bike was modified to use the components supplied by the other features and to use a motorized servo to adjust the resistance rather than a manual knob. It also had an articulating laptop mount affixed to it. **Feature 2**: measures user performance, describes the magnetic sensors used to count the peddle revolutions and the servo attached to the spin bike's resistance adjuster, both of which are connected to a microcontroller. **Feature 3**: real time exercise display, provides a display that calculates the user's exercise metrics based on the values received from Feature 2. This is presented by the **Feature 4**: attractive GUI, which also provides the ability to load and save profiles so users are able to monitor their workout through multiple sessions. Finally, **Feature 5**: Encourage fitness, is another part of the computer program which reminds the user to continue to exercise once the program has been started and presents them with achievements for reaching certain milestones in the Feature 4 metrics. Features 3, 4, 5 are combined together into a single Java program.

Development of the Lapcycle proceeded in two phases. The first commenced on 31 August 2015; during that time, the team formed the initial design concept, developed the aforementioned feature set, and created a prototype to demonstrate the features in a developmental state by the end of Phase 1 on 7 December 2015. Phase two commenced on 1 February 2016. The problem statement was revised to discuss the problems with motivating people with poor health management to change their habits. Based on this new information and feedback received during the demonstration of the laboratory prototype, Feature 5 was changed by replacing the WiFi lockout ability that was originally planned with the profile system mentioned in the previous paragraph. By the end of the development on 2 May 2016 the prototype is in a state of completion, but has some flaws due to the lack of resources and skills available to team 10. However, the concept has been well demonstrated and the remaining issues can be resolved with the development of a custom hardware interface complete with hardware drivers.

II. SOCIETAL PROBLEM

A. Introduction:

The target issue of Project Lapcycle is the growing rates of obesity in first world (economically developed) nations, such as the

United States. It is a complicated issue with many facets and causes. The device being designed is meant to combat one of those causes; the decrease of physical activity related to the increase in use of computers for work and entertainment. As more and more people are now working occupations requiring little to no physical activity and instead are relegated to performing computer related tasks. To top it off, there has been a stark increase in the use of media devices, particularly with the use of computers. It is the hope that project lapcycle will take advantage of this increase in computer use by integrating a user's computer usage into an exercise device.

B. *Obesity Prevalence*:

Obesity has been a growing problem in America and most other first world countries for since the mid the 20th century. The amount of overweight adults in America has increased from 46% in 1962 to 68% in 2010 (see Graph 1.) The cause of this is a complex mix of different factors, but it is primarily related to a decrease in physical activity combined with an imbalanced high calorie diet. Strength training and cardio vascular activity are vital to maintain a healthy weight and properly functioning organ systems [22]. While the increase emphases of electronic media have contributed largely to the decrease in physical activity, there has also been a shift away from heavy lifting and other physical occupations towards ones that are more computer related, further decreasing our physical activity. Although numerous private and public institutions have made many different programs to attempt to reverse this trend, especially in schools, the ratio of overweight Americans has still been increasing [27]. It is clear that our society is not going to shift away from the obsession with media, but encouraging increased physical activity is still a necessity.

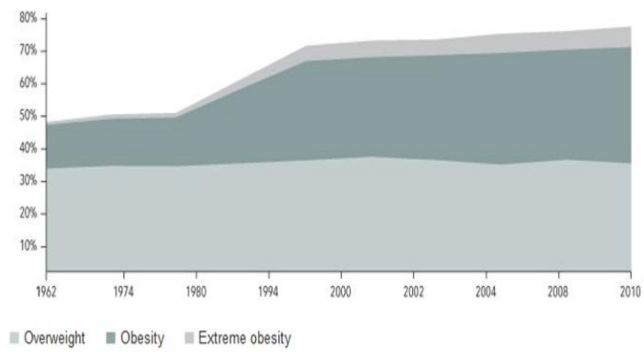


Figure 1: Proportion of overweight/obese persons from 1962-2010 [27].

Obesity and overweight are classified medical conditions that are pervasive in our society. They are caused by a variety of factors, but are primarily related to a metabolic energy imbalance [2]. In essence, an individual is consuming more calories than they can convert to energy in physical exertion and cellular respiration [2]. Some glucose is stored in the liver for the purpose of being released when blood sugar levels drop to a certain level, but if a person consumes too many extra hydrocarbons (in the form of either triglycerides of carbohydrates,) they will be stored as fat within adipose tissue in the human body. Some fat deposits are necessary for our survival [12]; fat insulates organs and protects internal cavities, but too much puts peoples at risk for reduced life expectancy due to various negative health effects [21].

C. *Obesity & Lack of Physical Activity*:

Humans evolved in an environment requiring almost constant physical activity. The development of long term settlements, plentiful food supplies, and labor saving devices developed very quickly when compared to an evolutionary time scale [4]. Hence, while we are biologically indistinguishable from our hunter-gatherer forbearers, we live drastically different lifestyles than what we evolved to be best suited.

The lack of cardiovascular exercise has numerous deleterious effects on the human body. As seen in Table 1, being cardiovascularly fit has numerous benefits, including decreased body fat, increasing muscle, and increasing metabolism [24]. Not only is someone who is fit burning calories while exercising, the increase in metabolism causes them to burn more calories for several hours after having performed the activity [17]. In addition, the increase in sarcomeres in muscle cells consumes more calories than someone who is not fit. Those lacking in physical activity are much more prone to developing chronic medical issues that can reduce life expectancy. Obesity is sometimes referred to as an epidemic due to its prevalence in America as well as the numerous deleterious health effects it causes. While cardiovascular disease and heart disease are often used interchangeably by the laymen, they have distinct meanings in the medical community. "Cardiovascular disease is caused by narrowed, blocked or stiffened blood vessels that prevent your heart, brain or other parts of your body from receiving enough blood" [4]. Cardiovascular disease is term used to describe the degradation of any and all blood vessels throughout the human body due to fat deposits clogging and hardening them [4]. Heart disease kills more people in America than any other disease; it can be caused by lifestyle and genetic factors, but the effects specifically linked to obesity are similar to the effects of cardiovascular disease. Atherosclerosis reduces, or even blocks blood flow in the blood vessels of the heart; when part of the heart can not get oxygen, it will start to die causing a heart attack [12]. Obesity is linked to heart disease as it causes an increase in blood pressure and atherosclerosis, both cause additional strain on the heart which leads to hypertrophic cardiomyopathy [12]. Obesity also can lead to Type 2 diabetes because an increase in fatty tissues leads to an increase in resistance to insulin [4]. Having diabetes can cause or dramatically worsen heart disease by obstructing blood vessels with a buildup of glucose sugar that occurs when the body becomes resistant to, or deficient in insulin. This buildup of glucose also causes damage to nerves, appendages, and eyes causing loss of sensation, necrosis, and loss of vision [4]. There are many other health risks associated with obesity and it is clear that considerable effort is needed to reverse this trend.

TABLE 1
BENEFITS OF RESISTANCE TRAINING [17]

Table 1. Benefits of aerobic exercise[25-36]

Variable	Benefit
Cardiovascular pathophysiology	
Maximal cardiac output	Increased
Peripheral oxygen extraction	Increased
Myocardial oxygen demands	Decreased
Fibrinolysis	Increased
Blood coagulability	Decreased
Endothelial function	Increased
Myocardial blood flow	Increased
Sympathetic hyperactivity	Decreased
Cardiovascular risk factors	
Resting blood pressure	Decreased
High density lipoprotein cholesterol	Increased
Triglycerides	Decreased
Body weight control	Increased
Insulin resistance	Decreased
Physical function	
Fitness/strength	Increased
Exercise capacity	Increased
Performance of activities of daily living	Increased
Return to work	Increased
Psychological wellbeing	
Depression	Decreased
Anxiety	Decreased
Quality of life	Increased

D. *Decrease in Physically Demanding Occupations*:

Over the past three decades, there has been a decrease in the physical activity required for the average occupation. The increase in computer use and automation has caused the development of more sedentary jobs. By measuring the METs (Metabolic Equivalents) of each occupation, researchers have been able to classify jobs that require moderate, sedentary, or light work. Moderate was illustrated as any value higher than 3.0 METs, while light work is between 2.0-2.9 METs, and sedentary is anything less than 2.0 METs [8]. This research demonstrates that there is an change from about 50% of the job market being moderate physical work decreased to 20% in the past few years [8] (see Graph 2.) Not only does this research show the increase in less physically demanding jobs, but also an increase in sedentary jobs.

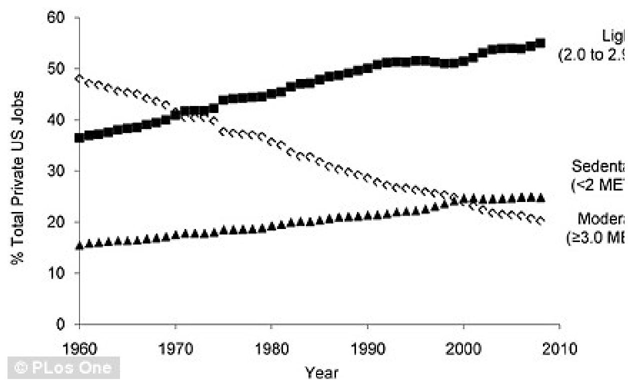


Figure 2: Occupations classified by physical activity [8].

Sedentary work requires lifting no more than 10 pounds at a time which includes objects such as papers, files, and small items [31]. A sedentary job usually involves sitting for the most part, although walking around and standing may be necessary at times to do the jobs tasks [31]. With less physical strain required at the workplace, it is not a surprise that some of them could be gaining weight. By sitting at their sedentary jobs, those workers are burning off less than 120 to 140 calories a day[10]. In the past 50 years, there has been a dramatic decrease in the amount of calories burned by performing one's occupational duties for both men and women (see Graph 3.) Burning off calories is an important way to decrease the problem in obesity, and losing out burning those calories during the work hours greatly adds to the difficulty.

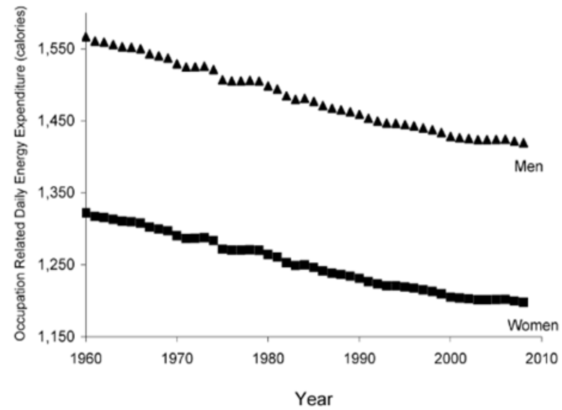


Figure 3: Caloric expenditure vs year [27].

As the dependence on computer guided technology increases, it is probably unlikely that the physically demanding jobs will increase as much as the sedentary ones. In 1960 a report showed that every one out of two Americans had a physically demanding job, now it is about one out of five [27]. The increase in sedentary jobs is most likely due to the increase reliance on the technology today such as the internet [27]. Though the decrease in physical activity in the workplace may be one of the factors in the increase in obesity rate, we need to also look at other causes of this problem, such as the increased time in watching television.

E. As Television Falls, Computer Entertainment Rises:

While the landscape of work has changed considerably from physically demanding to technical jobs, one major contributor of obesity has still been lingering: television. Like the shift from a physical to technical work methodology, the problem is that a more sedentary lifestyle has taken over. Over the past three decades, television has been one of the factors that have always been associated with the rise of obesity, but despite the changes in the viewer landscape and the format moving to the Internet, the watching of content still contributes to health risks.

According a survey performed by Liou and others, teenagers performed at least 450 minutes a day (6 hours and 30 minutes) on non-physical activity, with half of them consuming products that are either fried or containing sugar [20]. To reiterate on the negative effects at least two hours of television a day does to children, the most noticeable ones are degraded academic performance, improper eating habits attributed to food advertisements [eot 11, and lack of sleep [20], so no surprise comes from the fact that lack of exercise and a high-fat diet has a correlation to watching television. The directly proportional trend between high-fat foods and television can be explained by both snacking while watching television, and snacking between meals [21].

Despite the negative effects of television on children and the viewership in general, demographics and behind-the-scene realities of broadcasting television have considerably changed. According to Felix Richter and the Nielsen reports, the American 12 to 17 age demographic had dropped their viewing time by 25% since 2011, from 24 hours a week down to 18 [PS16)]. Rob Perogaro also points out average household monthly viewing time of live television has also dropped from 147 hours to 141, as seen in Graph 4 [17]. The other consideration to factor in is that as of December 2014, 2.6 million households are "broadband only," not having a cable or satellite service at all, and forty percent of viewers have one of the streaming services mentioned in a later paragraph, as opposed to only five percent from 2013 [11]. In terms of behind-the-scenes, broadcasters have been able to provide more video streams, otherwise known as the ability to "multicast," after the advent of digital television. Another note is that despite entertainment being fractured to as many competing franchises, news programming as of 2011 has been one of the long-standing foundations in the broadcast market in spite of their viewership drop from 50 million in 1980 to 22 million in 2009, even outdoing national news cable networks [11]. Some of the networks managed to take the decline by spreading their fixed costs in a multi-platform environment. The viewers, on the other hand, use either those new platforms or alternative methods to watch their content.

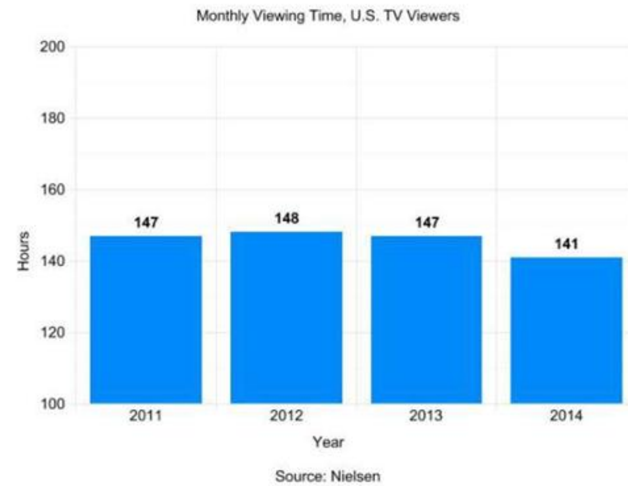


Figure 4: Nielsen Ratings: Viewing hours per month [22].

First off, one of the trends from switching from traditional "big box" television to a streaming model can be explained for a myriad of reasons. The first explanation is the phenomenon of "cord-cutting," or the act of cancelling a subscription from a cable or satellite service. Cord-cutting has been increasing in response to frustration over increasing payment rates, of which one increase happens by retransmission rates from Regional Sports Networks, as part as the high commanding rates major intercollegiate and professional leagues demand [11]. To illustrate an example of how much the retransmission cost of one team is, the Los Angeles Dodgers from Major League Baseball and Time Warner Cable made a 25-year, $7 billion contract to make the RSN *SportsNet Los Angeles*, and the cost of watching that team alone in only a handful of markets is close to four dollars per subscriber as of 2015, comparable to New England Sports Network and Comcast SportsNet Bay Area, both of which air games from at least two major league teams [4]. The inflating rates prompted some providers to either provide a viewing package without sports, or made viewers cancel their services to move on to the ones listed below [4].

Second off, for the sake of accessibility, multiple streaming services have been deployed for use on the internet, such as Amazon Prime, Hulu, and Netflix in the United States, and Freeview, the BBC iPlayer, and Sky Plus in the United Kingdom. James Bennett points out such services, with a combination of recording services like TiVo and Internet-Provided Television, provide a promise to the television audience the ability to become a self-scheduler [4]. The flexibility provided by the self-scheduling model extends to not just televisions and peripherals that support the service, but to the use of handheld devices and computers. The amount of viewable content on such systems has increased over the past decade, and consequently the amount of time spent on viewing such content, as seen in the next paragraph.

Eventually, the point to be driven from this discussion is that television, despite the decline of the traditional model, has found its way still in the negative health discussion by joining the services provided by the Internet. According to Phil Reed in 2015, Television and Film watching consisted of at least 67 percent of the time spent visiting websites as shown in Table 3 [4]. While traditional ratings have gone down, the realm of television (and everything that goes with it) has found a new niche in the realm of the internet. While cutting down on television usage is an ongoing high priority, cutting down or restricting internet usage time will provide to be the higher priority in a technical society.

TABLE 2
WEBSITE USAGE.

Table 1. Percentage of sample visiting websites of various forms, along with percentage male and females, and younger and older, participants visiting sites along with Phi coefficients.

	Sample	Female	Male	Phi	<30 years	30+ years	Phi
Social Networking	95.0	97.0	92.9	.094*	96.6	91.6	.105*
Shopping/Banking	87.1	90.6	83.2	.108**	84.6	92.9	.115**
Research	82.2	83.0	81.1	.023	84.6	76.8	.094*
Gambling	70.3	70.2	70.8	.007	70.8	69.9	.012
TV and film	67.3	64.2	70.8	.071	68.0	65.8	.022
Dating & sexual	65.0	59.2	73.3	.148***	66.0	65.8	.002
News	57.4	58.5	56.2	.023	52.9	67.7	.139***
Content sharing	46.5	44.5	48.8	.042	46.0	47.7	.016
Gambling	28.3	20.0	37.5	.194***	27.4	30.3	.030
Blogging	16.8	15.5	18.3	.038	14.3	22.6	.102*
Chat rooms	9.8	1.9	16.7	.259***	6.3	14.8	.138***

*$p < 0.05$
**$p < 0.01$
***$p < 0.001$.

doi:10.1371/journal.pone.0134538.t001

TV and Film is fifth highest on the list [4].

F. *Societal Problem in Relation to Design*:

As stated in the previous sections, people in first world countries are more attached to their technology. Many people spend the majority of their work week in front of a computer, only to go home and relax by using another computer! While many devices and programs designed to encourage fitness are meant to separate people from their engrossment with their computers, the concept behind this project instead works to integrate exercise into consumer's computer usage. Project Lapcycle will provide users with an affordable means to incorporate exercise easily into their work and entertainment while still providing an interactive program common to higher end cardio machines. Unfortunately, it seems that people who already have heart disease are still poorly motivated to maintain an exercises regiment [39]. Since the presence of healthier food and exercise equipment in the home tends to encourage their use [13], it makes sense to develop more devices that are economical, portable, and easy to use. Since there is also a relation to obesity and television usage [13] and there has been an increase in digital media usage [34], it is reasonable to assume that home exercise equipment that incorporated such devices and features would further encourage fitness. If everyone who spends one of the many hours in front of a laptop uses a Lapcycle for just one of those hours a day, they would exceed their minimum cardio activity recommendation. This additional physical activity that so many people are lacking would be a major step in increasing national cardiovascular health.

III. DESIGN IDEA AND FEATURE SET

A. *Addressing the Issue*:

Many ideas and programs have been presented in the United States and other leading nations to combat obesity, and their long term effectiveness leaves something to be desired. It is a multifaceted problem with no one-size-fits-all solution, with some recent attempts being the Play60 Initiative from the National Football League and the Let's Move initiative from First Lady Michelle Obama [43]. The following addresses one approach in the fight against obesity, namely, the addressing of the overuse of modern technology in work and fun [28]. While modern technology has their benefits in advances such as improved response to serious health issues, heart disease stays on top of the death rate list due to physical inactivity [25]. The best counter to heart disease is cardiovascular and aerobic exercise, and the design aims to allow for personal computing use while exercise is in effect. Incentives for exercise would be the allowance of use for the computer if the user proves to do enough exercise, while disallowance will reduce the amount of features available.

B. *The Elevator Pitch*:

"Design an exercise machine you can use while using your laptop or tablet computer."

C. *Constraints*:

The solution that Project Lapcycle aimed for may sound simple when spoken, but on paper, the project had to be grounded to Earth in some form. The first constraint was that since the Project is meant as an exercise machine aimed towards an indoors audience, ease of use must be emphasized on two fronts: ease of physical use and support of common operating systems on many end user computers. If the system is too bulky or difficult to operate, the user will be disinterested and one of the easier avenues of exercise will be neglected. The second primary constraint is since the point of the project is to serve both as an exercise program and a platform which allows either productivity or entertainment on a laptop, the complete body cannot be in exercise mode. If the whole body is exercising, then the laptop integration would be completely neglected as the laptop would not be in use at the time.

D. *Proposed Solution and Features*:

The "cycle" of Project Lapcycle is based off a stationary bicycle, with a computer-monitored resistance adjustment and reading system, a more comfortable seat for back support, and a monitoring system installed on an end user's laptop that uses resistance and work-force calculations to measure the exercising performance of the rider. The initial idea was to use a standard mountain or road bicycle converted to a stationary bicycle with a laptop attachment, however, the conversion was derided by the team for lacking elegance. However, a later revision of the project will use a pre-made stationary spin bicycle converted for laptop monitoring, allowing for a sleeker design.

Initially, three major components would be utilized to fulfill five features, the latter which is covered in Subsection D. The following components are:

1) Bicycle: Covers Feature 1.
2) Sensor Package: Covers Features 2 and 3.

3) Laptop Software Program: Covers Features 3 through 5.

E. Uniqueness of Design:

Unlike most exercise props a person would expect to see in a gym, Project Lapcycle uses a relatively inexpensive exercise and portable platform, but with incorporation of the user's tablet or laptop computer into the design. While other exercise machines (seen in the Figures below) do allow for the use of entertainment, their provided space is often limited, for room only for a miniature television or a book. A full tray can ensure users who are more attached to their work applications or games can continue their activities without interruption while getting their daily recommended cardio exercise.

Figure 5: An inexpensive machine, a step machine with strength training elastic bands [41].

Figure 6: An unpowered mechanical spin bike, the inspiration for the project [14].

Figure 7: A high-priced Elliptical, which has a digital readout panel [36].

F. Punch List:

The following list is a concise list of features that Project Lapcycle would be judged by, and the means of measuring their completion.

TABLE 3:
FEATURES AND METRICS OF COMPLETION

Features	Metrics
1. Single person spin bike.	A complete stable stationary hands free spin bike with adjustable resistance, supportive seat, and adjustable laptop mount.
2. Measure user performance.	Sensors communicating resistance setting and the rate at which the user is pedaling.
3. Real time exercise display.	Displays calorimeter, tachometer, speedometer, timer, and odometer. These displays update every second.
Features	Metrics
4. Attractive & easy to use GUI.	All metrics are displayed in distinguished graphics, extra prompts appear with simple messages and clear fields.
5. Use various attributes to encourage fitness.	The application displays always up front reminder prompts, usage reports, ~~ability to lockout internet usage,~~ and a profile* system that can save progress and grant achievements. *Added for Phase 2.

G. Measures of Success:

1) Feature 1: Single Person Spin Bike-

The main components of the spin bike came together far better than originally planned. The original concept involved building a mountain bike into a spin bike by reconstructing a mountain bike with a recumbent seat, a metal frame to hold it steady, a CRT monitor mount to hold the laptop, and a servo to adjust the derailleur. It was soon realized that this design was ungainly and unreliable. However, the mountain bike was used for the Phase 1 prototype (see Figure 8,) but the bike's gear system was not used and the resistance was controlled by a trainer mount that increased the level of friction to change the resistance. It was decided that using an already constructed recumbent spin bike would prove to be a far superior design. In addition, instead of trying to attach a CRT monitor wall mount to hold a laptop, Team 10 managed to find a four point articulating laptop desk mount and attach it to the spin bike handle. The result is comfortable to use and easy to operate while using a laptop (see Figure 9.)

Figure 8: Laboratory Prototype Dec 2015 [3].

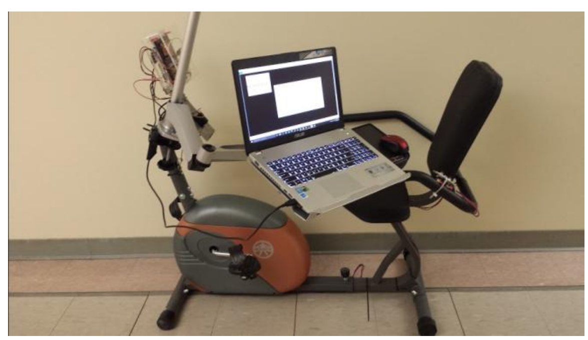


Figure 9: Deployable Prototype Apr 2016 [3].

2) Feature 2: Measure User Performance-

This feature works sufficiently for demonstrative purposes but could be made more accurate and detailed with further development. The optical movement sensor built into the interior of the bike used to detect motion was disconnected and in its place was a magnetic read switch. The bike was designed to sample movement in the millisecond range as well as measuring fractions of revolutions. In contrast, the laboratory prototype constructed in Phase 1 only took a reading once per rotation. A second magnet was added to the deployable prototype in Phase 2 which did make it more accurate, but it was still very rough. Attempting to add four magnets created such a strong magnetic charge on the metal plate that was behind the magnets that it became polarized and created a constant positive reading. With only two magnets, two of the metrics calculated by Feature 3 were inaccurate and the code for Features 2 & 3 had to be altered in order to make better use of the limited information gathered from the two readings per rotation sampling design. A suggested improvement, if this goes into further development would be to replace the Arduino microcontroller with a custom built interface device and use a sensor similar to the optical sensor that originally came with the spin bike. In place of the servo that was used to adjust the resistance level, if the spin bike was built from the ground up to have electronically controlled resistance, it would function more reliably and with a wider series of resistance levels. Additional sensors and metrics, such as a heart rate monitor would give more data regarding the user's performance and would help better measure caloric expenditures. Finally, Team 10 was hoping to use a new microcontroller package to improve communication between the two devices and allow for the addition of more features, such as preprogrammed workouts that auto adjust the resistance over a set time. Unfortunately, this new microcontroller did not meet the team's expectations and they were forced to resort to recycling the setup used for Phase 1.

3) *Feature 3: Real Time Exercise Display-*

This was another feature that had a lot of success. It presents readings for time, speed, calories, revolutions per minute, total revolutions, and simulated distance traveled (for more information, see Appendix C.) Most of its limitations were due to the lack of more sophisticated sensors as described in the previous sub-subsection. Originally, the Arduino sent two points of data to the serial line to be read by the Java program on the laptop. These were the number of revolutions since the start of the program and the current resistance level. Because the microcontroller only takes readings twice per revolution and the Java program updates its metrics once per second, the total revolutions can only increase by multiples of one half second resulting in RPM values that were always multiples of thirty and it caused a similarly inaccurate speed rating. In order to make better use of the limits of the tools at hand, the microcontroller code was rewritten to calculate the RPM and send that to the Java program instead of the total number of revolution. The methods handling the calculations and testing for zero movement were then modified to display the six metrics more accurately than in the original design. However, with the limited sampling rate, it still assumes a constant speed during the one second period. If the Arduino was replaced with a more powerful control system that did not rely on a text file read/write to transfer data, more readings could be taken per second creating a smoother rate of change graph.

4) *Feature 4: Attractive and Easy to Use GUI-*

Creating a visual interface was a necessary step in making the program easy and desirable to use. As discussed in the Software Development Guide in Appendix C, Java was used to make the program multiplatform and to facilitate creating graphics. The original display used only a JFrame with JLabels added to it. The deployable prototype used a custom background with the JLabels added to it and organized with a SpringLayout.

5) Feature 5: Encourage Fitness-
Several changes were made to this feature as the team learned what was possible with the available to them and after receiving feedback for the laboratory prototype. The basic idea of having prompts appearing when the user fails to move when the program is running remained, but the original idea of locking out internet access until the user fulfils preset requirements was replaced with a profile and achievement system to make it more enjoyable to use instead of forcing them to use it.

IV. FUNDING

Most of Project Lapycle was funded entirely out-of-pocket from the team, as the total cost provided in Tables 4 and 5 was low enough to not be considered in need of a grant or with external support. No sponsorship was sought.

The laboratory prototype was constructed out of an old mountain bike attached to a bicycle trainer. The bike was donated by Team 10 member Andrew Beltran and the trainer was purchased by member Armando Fuentes. Most of the other materials were purchased by J. R. Chadwick or were available at from his parents.

Constructing the deployable prototype required purchasing and recycling materials and resources once the design was finalized. The cost was relatively low and everything needed was purchased by the members of Team 10 or supplied by the parents of J. R. Chadwick. Several parts were also reused from the Phase 1 prototype as noted in Table 5.

TABLE 4
PHASE 1 LINE ITEM BUDGET

Item	Quantity	Cost (USD)	Paid By
*Arduino Uno series Microcontrollers	3	$30.99	Andrew Beltran (1), Armando Fuentes (1), J.R. Chadwick (1)
*Reed Switch	2	$6.55	J. R. Chadwick
Bike Support Frame Materials	1	Trivial	J. R. Chadwick
Bicycle Trainer	1	$75.59	Armando Fuentes
Standard Mountain Bike	1	(Donated)	Andrew Beltran
*Thermostat Guard	1	$19.99	J.R Chadwick
*Zip Ties (20)	1	$10.99	J.R. Chadwick
Total Estimated Cost:		$144.11	

*Item recycled for Phase 2

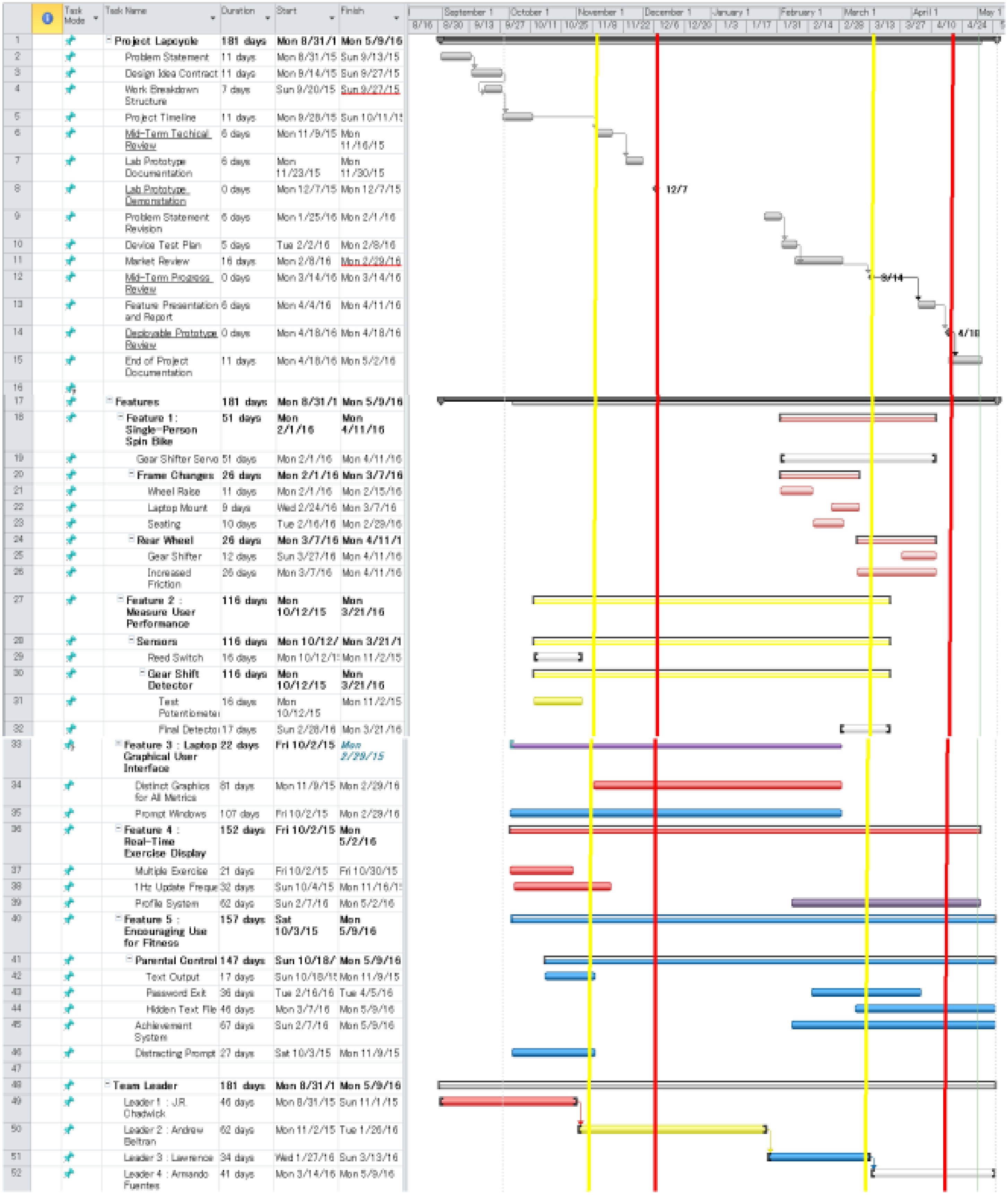
TABLE 5
PHASE 2 LINE ITEM BUDGET

Item	*Quantity*	*Cost*	*Provided By*
Microcontrollers	2	$30.99	Andrew Beltran (1) Team 10 (1)
Marcy Recumbent Exercise Bike: ME-709	1	$169.00	Team 10
Reed Switch	1	$6.55	J. R. Chadwick
NB Articulating Laptop Mount	1	$108.00	J. R. Chadwick
Miscellaneous supplies		$31.55	J. R. Chadwick & Armando Fuentes
Total Estimated Cost:	$346.09		

V. PROJECT SCHEDULE AND MILESTONES

A. *Work Timeline*:

The developers of Project Lapcycle were given one primary schedule restriction: complete implementation of all features discussed in Section II in a nine month span, from September 2015 to May 2016. Figure X2 provided the set schedule, providing who would work on what, and the tentative time span for those assignments. Each assignment in the Timeline had been color-coded, each raw color being allocated to one person (refer to the "Leaders" section to determine responsibilities). If any two team members were allocated to the same task, the color of the assignment is mixed. Most of the schedule was set generously to account for a "laissez-faire" management, with the milestones saying a responsibility was fulfilled when the team determines it was fulfilled. Figure 10 covers the timeline.



	Task Mode	Task Name	Duration	Start	Finish
1		Project Lapcycle	181 days	Mon 8/31/1	Mon 5/9/16
2		Problem Statement	11 days	Mon 8/31/15	Sun 9/13/15
3		Design Idea Contract	11 days	Mon 9/14/15	Sun 9/27/15
4		Work Breakdown Structure	7 days	Sun 9/20/15	Sun 9/27/15
5		Project Timeline	11 days	Mon 9/28/15	Sun 10/11/1
6		Mid-Term Techical Review	6 days	Mon 11/9/15	Mon 11/16/15
7		Lab Prototype Documentation	6 days	Mon 11/23/15	Mon 11/30/15
8		Lab Prototype Demonstation	0 days	Mon 12/7/15	Mon 12/7/15
9		Problem Statement Revision	6 days	Mon 1/25/16	Mon 2/1/16
10		Device Test Plan	5 days	Tue 2/2/16	Mon 2/8/16
11		Market Review	16 days	Mon 2/8/16	Mon 2/29/16
12		Mid-Term Progress Review	0 days	Mon 3/14/16	Mon 3/14/16
13		Feature Presentation and Report	6 days	Mon 4/4/16	Mon 4/11/16
14		Deployable Prototype Review	0 days	Mon 4/18/16	Mon 4/18/16
15		End of Project Documentation	11 days	Mon 4/18/16	Mon 5/2/16
16					
17		Features	181 days	Mon 8/31/1	Mon 5/9/16
18		Feature 1 : Single-Person Spin Bike	51 days	Mon 2/1/16	Mon 4/11/16
19		Gear Shifter Servo	51 days	Mon 2/1/16	Mon 4/11/16
20		Frame Changes	26 days	Mon 2/1/16	Mon 3/7/16
21		Wheel Raise	11 days	Mon 2/1/16	Mon 2/15/16
22		Laptop Mount	9 days	Wed 2/24/16	Mon 3/7/16
23		Seating	10 days	Tue 2/16/16	Mon 2/29/16
24		Rear Wheel	26 days	Mon 3/7/16	Mon 4/11/1
25		Gear Shifter	12 days	Sun 3/27/16	Mon 4/11/16
26		Increased Friction	26 days	Mon 3/7/16	Mon 4/11/16
27		Feature 2 : Measure User Performance	116 days	Mon 10/12/15	Mon 3/21/16
28		Sensors	116 days	Mon 10/12/	Mon 3/21/1
29		Reed Switch	16 days	Mon 10/12/1	Mon 11/2/15
30		Gear Shift Detector	116 days	Mon 10/12/15	Mon 3/21/16
31		Test Potentiometer	16 days	Mon 10/12/15	Mon 11/2/15
32		Final Detector	17 days	Sun 2/28/16	Mon 3/21/16
33		Feature 3 : Laptop Graphical User Interface	22 days	Fri 10/2/15	*Mon 2/29/15*
34		Distinct Graphics for All Metrics	81 days	Mon 11/9/15	Mon 2/29/16
35		Prompt Windows	107 days	Fri 10/2/15	Mon 2/29/16
36		Feature 4 : Real-Time Exercise Display	152 days	Fri 10/2/15	Mon 5/2/16
37		Multiple Exercise	21 days	Fri 10/2/15	Fri 10/30/15
38		1Hz Update Freque	32 days	Sun 10/4/15	Mon 11/16/1
39		Profile System	62 days	Sun 2/7/16	Mon 5/2/16
40		Feature 5 : Encouraging Use for Fitness	157 days	Sat 10/3/15	Mon 5/9/16
41		Parental Control	147 days	Sun 10/18/	Mon 5/9/16
42		Text Output	17 days	Sun 10/18/1	Mon 11/9/15
43		Password Exit	36 days	Tue 2/16/16	Tue 4/5/16
44		Hidden Text File	46 days	Mon 3/7/16	Mon 5/9/16
45		Achievement System	67 days	Sun 2/7/16	Mon 5/9/16
46		Distracting Prompt	27 days	Sat 10/3/15	Mon 11/9/15
47					
48		Team Leader	181 days	Mon 8/31/1	Mon 5/9/16
49		Leader 1 : J.R. Chadwick	46 days	Mon 8/31/15	Sun 11/1/15
50		Leader 2 : Andrew Beltran	62 days	Mon 11/2/15	Tue 1/26/16
51		Leader 3 : Lawrence	34 days	Wed 1/27/16	Sun 3/13/16
52		Leader 4 : Armando Fuentes	41 days	Mon 3/14/16	Mon 5/9/16

Figure 10: The Time Line.

VI. WORK BREAKDOWN

A. *Work Breakdown Structure*:

To organize how Project Lapcycle would be created as a collection of smaller components merged together, Figure X1 provides the structure of the project. This manages to contain the big picture, however breaks down the project into smaller, yet far more manageable pieces.

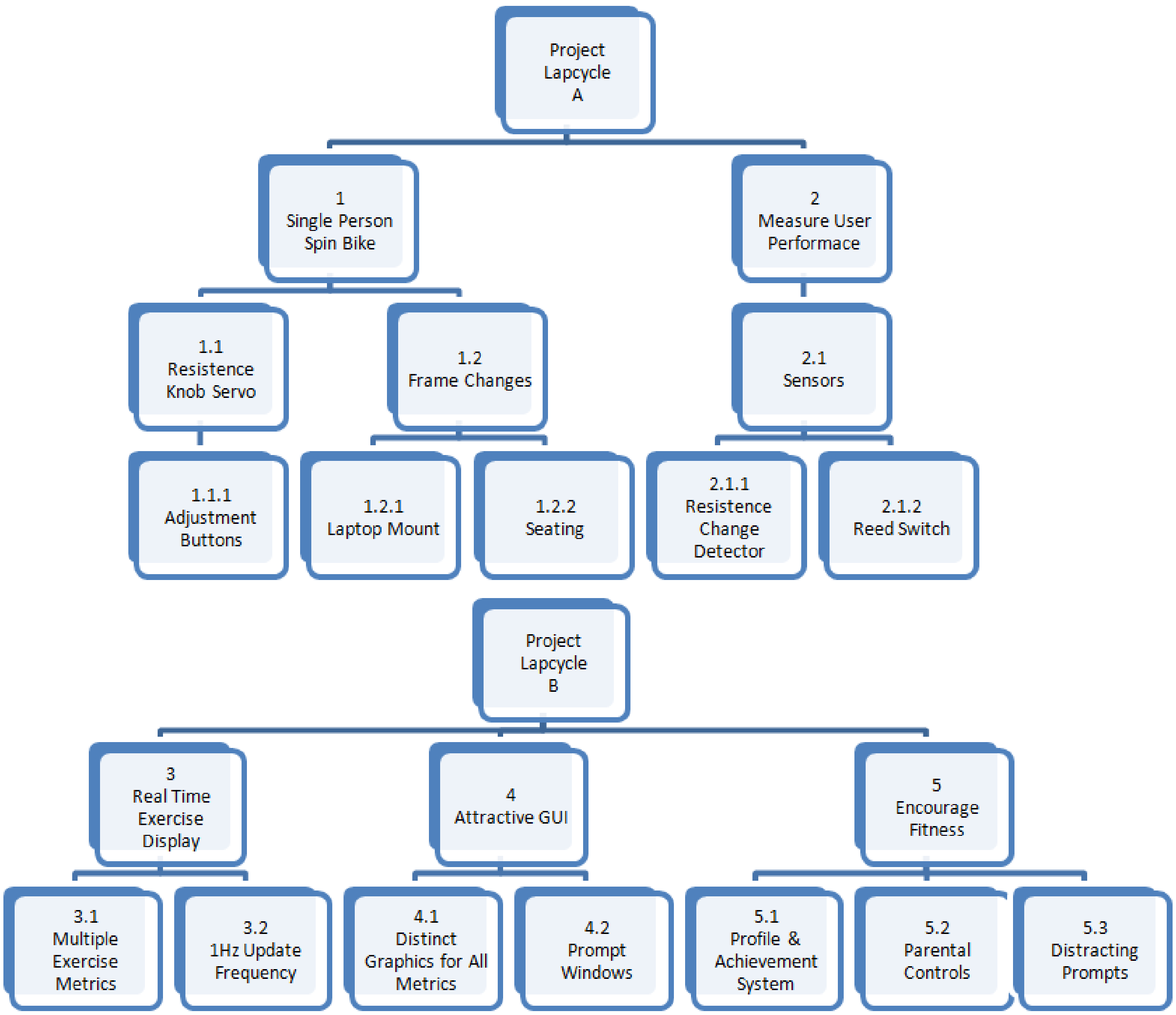


Figure 11: The Work Breakdown Diagram

The list of Figure 11 is separated into a listing form and their functional descriptions:

1: Single Person Spin Bike

- 1.1: Resistance Knob Servo
  - 1.1.: Adjustment Buttons
- 1.2.: Frame Changes
  - 1.2.1: Laptop Mount

1.2.2: Seating
2: Measuring User Performance
2.1: Sensors
2.1.1: Resistance Change Detection
2.1.2: Reed Switch
3: Real Time Exercise Display
3.1: Multiple Exercise Metrics
3.2: 1 Hertz (once per second) Update Frequency
4: Attractive Graphical User Interface
4.1: Distinct Graphics for All Metrics
4.2: Prompt Windows
5: Encouraging User Fitness
5.1: Profile & Achievement System
5.2: Parental Controls
5.3: Distracting Prompts

B. Work Allocation

To allow Project Lapcycle to be completed within a reasonable schedule, each team member was allocated a task. Most of Phase 1 dealt with secondary documentation, along with preliminary assembly and research into what physical components were needed to present a laboratory grade prototype. Phase 2 had even more secondary documentation, and dealt more with an optimization phase, including creating a more presentable product with the end user in mind in both hardware and software sections. Throughout nine months, almost a total of one thousand work hours had been allocated, most of them being allocated to the secondary documentation and the software development.

TABLE 6:
FALL SEMESTER 2015 TIME ALLOCATION

- Feature	**Task**	**Personnel**	**Hours**
Course Work			
	Problem Statement	Chadwick Beltran Ly Fuentes	13.5 8.5 2 2.5
	Design Idea Contract	Chadwick Beltran Ly Fuentes	21 5.5 4 5
	Work Breakdown	Chadwick Beltran Ly Fuentes	39 6 6.5 7
	Risk Assessment	Chadwick Beltran Ly Fuentes	15 9 4.5 7
	Timeline	Chadwick Ly	6 4
	End of Term Documentation	Chadwick Beltran Ly Fuentes	22 23 15 16
Feature 1			
	Trainer Resistance	Fuentes	7
	Laboratory Prototype Assembly	Chadwick Beltran Fuentes	6.25 2 7
Feature 2			
	Technical Research	Beltran Fuentes	9.5 4
	Reed Sensor	Beltran Fuentes	9 14.5
	Gear Detector	Fuentes	7
	Gear Shifter Servo	Fuentes	10
	Microcontroller Troubleshooting	Beltran	4
	Development: Microcontroller to Bridge	Chadwick Beltran	1 16.5
	Miscellaneous Testing	Beltran	1
Feature 3			
- Feature	**Task**	**Personnel**	**Hours**
	Main Class(With Timer)	Chadwick	58.5
Feature 4			
	Graphical Output Display	Chadwick	3
Feature 5			
	Technical Research	Ly	9
	Distracting Prompt	Chadwick Ly	1.5 13
	Stall Prompt	Chadwick Ly	1.5 8
	Parental Controls	Ly	9
Total		Chadwick Beltran Ly Fuentes	188.25 94 75 87

TABLE 7:
SPRING SEMESTER 2016 TIME ALLOCATION

Feature	**Task**	**Personnel**	**Hours**
Course Work			
	Problem Statement Revision	Chadwick Ly	7 2
	Change Request	Chadwick Beltran Ly Fuentes	7.5 11.5 5 3
	Timeline Revision	Chadwick Ly	1 2
	Design Test Plan	Chadwick Beltran Ly Fuentes	8.5 5 2 2
	Market Review	Chadwick Beltran Ly Fuentes	22.5 10 8 7
	Midterm Technical Review	Chadwick Beltran Ly Fuentes	4 12 10 3
	Feature Presentation	Chadwick Beltran Ly Fuentes	8.5 14 8 9
	Deployable Prototype Presentation	Chadwick Beltran Ly Fuentes	2 4 2 2
	End of Project Documentation	Chadwick Beltran Ly Fuentes	34 32 20 22
Feature 1			
	Gear Shifter Housing	Fuentes	4
	Laptop Mount Attachment	Chadwick	4
	Resistance Adjustment	Chadwick	7.5
	Button Attachment	Chadwick	2
Feature 2			
	AVR Programming Research	Beltran	17
	Atmega32u4 Tutorials/Programming	Fuentes	11
	Second Microcontroller Research/Bridge	Beltran	15
	Hall Effect Sensor	Beltran	4
	Gear Shifter	Beltran Fuentes	9 16
	Servo Mounting and Testing	Fuentes	12
Feature 3			
	Calculations Class	Chadwick	14
	Main Program	Chadwick	27
Feature 4			
	JFrame	Chadwick Ly	12.5 4
	JPanel	Chadwick	19
	User Profile	Chadwick Ly	21 12
Feature 5			
	Achievement System	Chadwick Ly	8 30
Total		Chadwick Beltran Ly Fuentes	210 133.5 101 91

VII. RISK ASSESSMENT & MITIGATION

A. *Introduction*:

Despite the fact that Project Lapcycle only had five features, various risks existed that would compromise the functionality of the project in some capacity, ranging from benign to catastrophic. For the sake of punctuality, only technical or project-related physical risks are covered in this section, while personnel-related risks (such as mental or financial problems among the team) are ignored. The assessment of these risks and their estimated costs (time and financial) and probability and weighed against each other are displayed in Table 8.

B. *Identified Risks*:

1) *Feature 1 Risks*-

Safety of the rider is the most pivotal aspect of exercising on a bicycle, and if the rider falls off the bicycle due to bad seating design, then the bicycle will be just unusable. Comfort of the bicycle is one considerable risk, as a standard bike seat does not suit the purpose of the project, and a new seat supporting the recumbent position would be more suited. The other risk imposed was with the rig used for the Fall Term, where modifying the actual bicycle to measure mechanical resistance for physical effort was deemed a mite too impractical. The Spring Term revisions hope to address the effort measuring issue.

2) *Feature 2 Risks*-

Most of the risks presented in this feature would affect the sensor packages installed on the bicycle, and if they fail from physical or internal damage, no useful information would arrive to the rider, giving him or her no idea on how well their efforts are doing. Without that measurement of effort, the design is essentially useless. Another consideration was that the program was initially written in the Java programming language, and that some of the hardware, depending on the interfacing connections, would run the risk of not cooperating.

3) *Feature 3 Risks*-

This feature deals more with a software perspective, after all the raw measurement numbers arrive from Feature 2. The greatest risks posed on this feature are if the calculations are somehow incorrect, the timing delay causes a ripple effect that shows inaccurate data, or the program crashes outright.

4) *Feature 4 Risks*-

One of the hallmarks of modern programming is making a Graphical User Interface, however, making one is actually a significant effort in and of itself. Two reasons explain why: one, the use of prefabricated GUI packages have their own complexities, and sometimes, the programming language used may have its own incompatibilities or convoluted intricacies to work with. Java has prefabricated packages known as Swing and AWT, however they have their own quirks and intricacies that can cause major annoyance.

5) *Feature 5 Risks*-

The core of Feature 5 is composed of three software components, namely the Wireless (WiFi) Lockout, Parental Controls, and Distracting Prompt. Like any software program, the precision required needs the system to both function and act in its intended manner. The major risks posed in this section involve the WiFi Lockout not functioning (or worse, crashing the user's laptop in a catastrophic fashion), the Parental Controls not outputting the correct monitoring data or having their purpose defeated with a simple bypass, and a "sticky" Distracting Prompt, meaning the prompt stays even if sustained bicycle activity is performed. The lockout failure does possess the worst consequence, providing the textbook definition of "sunk cost."

Creating the profile and achievement system used the existing Java File and JFrame libraries and as such had similar risks to Feature four. Allowing the profile system windows to display correctly requires proper use of the Java Layout Manager subclasses which are often counter intuitive and hard to fine tune. In addition, using reference to the JFrame class creates a separate thread and this can cause the main thread to continue before the profile thread has loaded the necessary file information, creating a race condition.

C. *Mitigation and Response Procedures Recapped*:

All of the above risks were capable of being reduced in some way to keep within the comfort zone of the team, and most of them had responses either by material replacement or by foresight. Table 5 provides the risk-to-cost assessment matrix. Features 1 and 2 consist mostly of physical hardware components, so the replacement of parts or relying on outside personnel with mechanical expertise were considered viable options within an immediate time frame. However, as Features 3 through 5 rely on mostly software operations, the only reliable response is to rely on practices normally reserved in software engineering: flowcharting approaches to simplify the issue, a higher than normal time allotment, side-stepping the issue with alternate paths, and expertise from external sources.

To support in the development of the project and mitigate the likelihood of risks the team would be unable to handle, two experts were available for consultation:

1. *Tod Chadwick*-

With decade decades of experience working with machines, cars, and tools, Mr. Chadwick's resources, knowledge, and work space are invaluable to the prototype construction and the completion of the prototypes and the construction of Feature 1.

2. *Cody Jackson, Graduate Student at CSUS*-

A Graduate Student in Computer Science at CSUS, Jackson has expertise in the programming languages of C, Java, and Python, and was one of the regular CSC tutors some of the Team had consulted with for several times prior to joining Project Lapcycle. His advice on programming had helped construct most of the software base that made this project possible.

TABLE 8
RISK IMPACT MATRIX

	Cost				
Probability			1) Bicycle Resistance	~~5) WiFi Lockout~~	
		1) Seating Stabilization		4) Java Swing	
	5) Distracting Prompts		2) Phase 2 Java Incompatibility	5) Parental Controls	3) Timing Desynchronization
			5) Profile System	2) Microcontroller Failure	3) Calculation Inaccuracies 3) Exercise Metrics
		2) Sensor Failure	4) Java Incompatibility		
	0.2	0.4	0.6	0.8	1.0

VIII. OTHER DESIGN DOCUMENTS

A. *Market Analysis*:

1) *Introduction*-

In order to make best use of the design potential, the members of Team 10 conducted a study of the size and variability of types of consumers for which the Lapcycle has viability as a marketable product. As this is a piece of exercise equipment, which is a market already flooded with devices, the first area analyzed was existing personal exercise equipment, including products that allow consumers to exercise while completing either tasks. It was discovered that the particular feature set of project Lapcycle is unique and nothing currently available offers the ability to use a personal laptop while exercising that also interfaces with the laptop. In addition to looking at the demand for exercise products, the team also examined the markets of people who do not necessarily exercise, but make frequent usage of computers for entertainment or for work. Again, this demonstrated to be rich source consumers who are active planners willing to spend a comparatively small amount of money in order to wok some positive life changes into their normal weekly activities.

2) *Similar Products*-

Exercising while you work is not a new concept. A common method used to encourage fitness into the lives of people who work long hours sitting at an indoor office station is to add a device to incorporate cardio exercise into their work hours without requiring them to take breaks. This concept is similar to the Lapcycle in that it allows multitasking; individuals who do not have the time or motivation to exercise, but spend many hours at a computer. There are several forms of these workstation additions. The simplest and least expensive option is placed on the floor in front of the user's office chair and operated by the user’s legs. These take the form of stepper or elliptical machines (see Figure 11) and are similar to their larger counterparts in function, but are far less expensive; often selling at prices under one hundred dollars. There are also more complicated devices that are designed to replace the user's entire work station. Bike Desks (see Figure 12) like these don't offer any other features than the under desk elliptical and cost over $1,700.00.

Although under desk devices are less expensive than Project Lapcycle, those devices are simplistic and not adjustable. Neither of these devices allow any interaction or metric recording that the Lapcycle offers. Since none of the devices described in this sub-subsection have digital readouts nor can they connect to a computer, there is no way to measure your progress and there is no added incentive to use them.

Similar to the devices described previously, devices such as stationary bikes, ellipticals, and treadmills (see Figure 13) are very good at burning calories and allow the user to distract themselves by placing them in the same room as their entertainment center. While some of these cost hundreds or even thousands of dollars, some of the spin bikes retail for only two or three hundred dollars. If so desired, an even cheaper route can be the models that forgo any sort of readout or electronics of any kind and operate only with a mechanical friction resistance adjuster.

Since the Lapcycle does not come installed with a digital readout and instead uses the computer the user already has, the microcontroller and sensors only add about forty dollars to the cost of the prototype making it only slightly more expensive than the standard unpowered models.

Critically, the Lapcycle is the only concept on the market which has the ability to fully integrate a computer into their workout while still measuring their performance and providing the standard features of a powered machine (adjustable resistance, incentive based encouragement, and workout metric readings.) The closest retail product found was the FitDesk (see Figure 14) which provides a platform for a laptop, but does not allow the user to store and monitor their continuing exercise habits, costs slightly more than the projected value of the Lapcycle, is not in the more comfortable recumbent position, and does not allow the use of an external mouse.

Figure 12: Under Desk Elliptical Device [15].

Figure 13: Desk Bike [1].

Figure 14: Treadmill [9].

Figure 15: FitDesk [42].

3) *Computer Usage at Home*-

Laptops and tablets are often used for entertainment as well as used for work. This is causing users to divert less time to physical activity. Worse yet, as described in the Societal Problem, people engaged in using a computer for long periods of time are more likely to consume excess calories. By using the Lapcycle the users will be able to multitask their entertainment/work with a good work out at the same time for both children and adults.

. Children and Teens born in the late 20th and early 21st centuries have grown up with a huge variety of electronic media to absorb their free time. Studies have shown that the number of sit-ups that a ten year old could do has declined by 27.1% between the years 1998 to 2008 [3]. Within a decade children, have become less active and less fit. In order to make up for lost physical activity, the Lapcycle will allow teens and children to have time to enjoy using their laptops or tablets for entertainment while getting proper exercise at the same time.

4) *PC Gaming*-

After the 1983 crash of the video game industry, it was thought that video games were a passing fad and there was no sense investing any further research into further development and marketing [6]. However, the release of the 1985 Nintendo Entertainment System, video games have made a roaring comeback [6] and are now a more profitable industry than movies [16], making over $67 billion dollars in 2013 based off of the sale of games, consoles, and peripherals. $21.5 billion dollars alone was spent on PC gaming in 2015 [38] and continues to drive the high end laptop and tower computer market even while the rest of the computer market has been gradually overtaken by the notebooks and tablet computers, "While overall PC sales might be down though, high-end desktops and laptops for gaming are doing better than ever." [23]. Martindale also states that PC game sales are increasing at a rate of 26% annularly. Tablet computers have been taking over a large portion of the market due to their compact nature, however they do not have the hard drive space, CPU frequency, memory, or graphics capability to run modern games and that is what is keeping the tower PC and full sized laptop market from being completely overrun. Laptop lines, such as the ASUS N56J and the Dell Alienware, that contain off board graphics cards with independent memory are common among gaming enthusiasts and video editors. As described in the problem statement and confirmed by the Team 10 survey results, this increase in computer entertainment usage has increased the sedentary lifestyle of many people who can no longer be sufficiently entertained by physical activity. Based on the analysis of this market, the PC gamer crowd is a viable source of consumers who are ready to adopt a new product designed to comfortably introduce a method of increasing their physical activity without interrupting their gaming time.

5) *Survey Results-*

To better understand the current market, Team 10 created a survey to present to people in their area to better understand the motivations and life styles that effect their willingness and ability to exercise. The method of data retrieval was with a paper survey, and the team gathered responses from 82 people. The team tried to gather data from as wide an age demographic as possible, but roughly 80% of the respondents ended up being in the 17-24 bracket (see Figure 4) since most of the people available to us were fellow college students. However, several strong trends in the gathered data revealed that the age did not seem to effect the responses. Since this survey had less than one-hundred participants and can only be classified as pseudo-scientific at best, the strong trends in the data fit well with the research discovered in the problem statement and support the purpose of the Lapcycle design. The following six figures display the compilation of the results of the most important questions asked on the survey. These results demonstrate a niche of people who have a strong desire to exercise more, make frequent use of a computer, and lack time and motivation to exercise more.

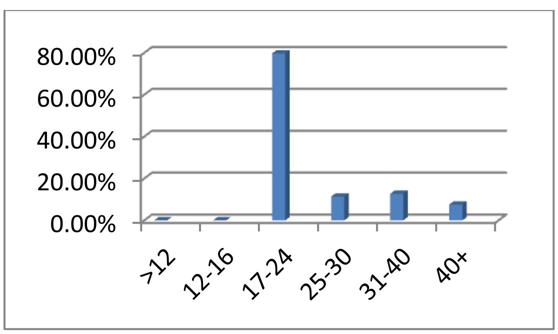


Figure 16: Age of survey participants in years [5].

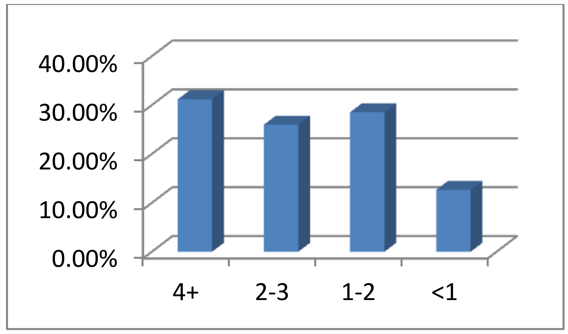


Figure 17: Hours spent using computers for entertainment [5].

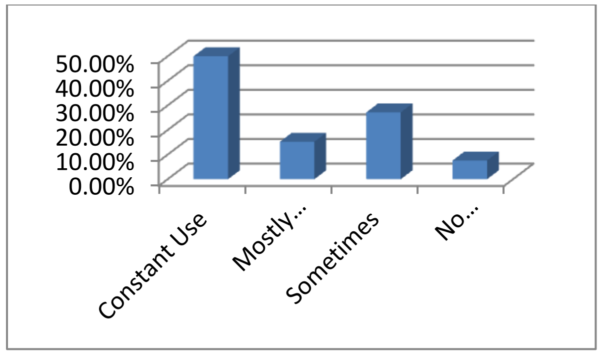


Figure 18: Computer usage for work/school [5].

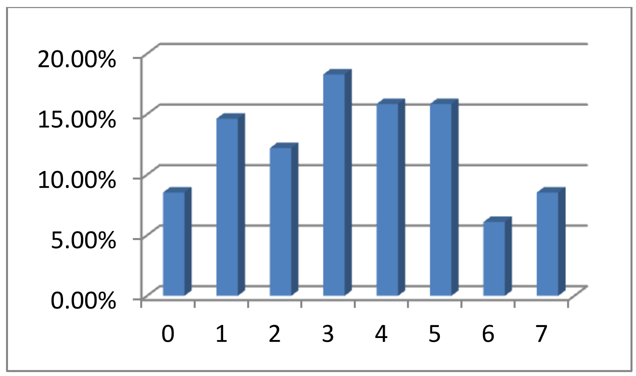


Figure 19: Days per week in which exercise occurs [5].

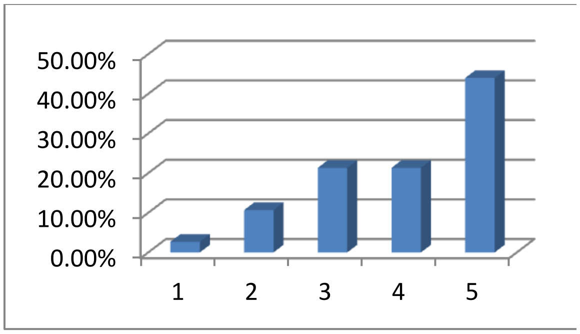


Figure 20: Perception of benefit from additional cardio exercise (1 being weak to 5 being strong) [5].

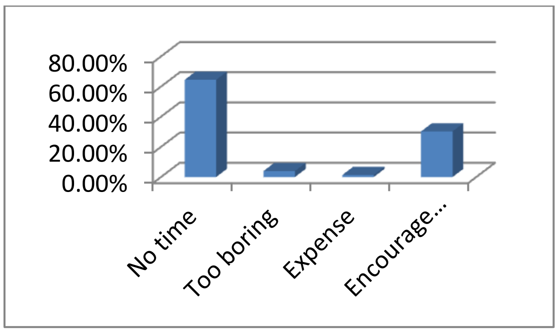


Figure 21: Reasons for not exercising.

6) *SWOT Overview-*

In order to find out how to effectively place the Lapcycle in its best market possible, the following factors were considered:

- **Strengths**: Low maintenance, relatively low cost, versatility, low production cost, large consumer base.
- **Weaknesses**: Incomplete product; requires consumer to already own a laptop or tablet.
- **Opportunities**: Large market of people desiring to have more time to exercise who also use computers frequently.
- **Threats**: Saturated market full of exercise equipment.

B. Testing Results:

1) *Introduction*:

Now that the Risk Assessment and Market Analysis sections have been covered, the possibility of risks runs high even during development phases. To diagnose whether these risks would affect the project in such an adverse manner, in-house testing was conceived to address both of these risks and prove the functionality of Project Lapcycle. The other point of these tests was to ensure that Most of these tests are arranged by Feature, and will provide a quick recap of each test. For further discussion on these tests, refer to Appendices B through D.

2) *Feature 1: Single Person Spin Bike-*

a) *Spin Bike Comfort Test*

Given that the "Cycle" of Project Lapcycle is actually a stationary bicycle, steps must be taken to make sure that no damage occurs to the rider or the bicycle during operation.

This test was done by letting a rider ride the bike as usual, and ensuring the rider did not suffer discomfort, and not letting their normal pedaling actions damage any of the cables or new components.

Results: The rider did not get injured, and that he did not hit the gear shifting wires from the pedal, nor did he unintentionally damage the gear system or Microcontroller case.

b) Rider Comfort Test

After checking that none of the components were unintentionally damaged during normal operation, this test would check if a variance of riders were capable of riding the bike. This was necessary as Project Lapcycle is meant to accommodate as the Stationary Bike frame would allow.

The testers consisted of each of the four Project developers, as they were of different body compositions. Their builds and results were:

- A tall person (6 feet, 180 pounds): No noticeable discomfort or loss of functionality.
- A short and slim person (5 feet 5 inches, 150 pounds): No noticeable discomfort or loss of functionality.
- An overweight person (5 feet 10 inches, 220 pounds): No noticeable discomfort or loss of functionality. Only needed to stretch the bike out using the bike's stretch function to fit.
- A slightly disabled person with muscular dystrophy: No discomfort or loss of functionality.

The overall result was barring anyone that the Marcy ME-709 manual said to not allow, this test proved that the bike would accommodate anyone in the normal weight range.

b) *Spin Bike Comfort Test*

Lapcycle is actually a stationary bicycle, steps must be taken to make sure that no damage occurs to the rider or the bicycle during operation.

This test was done by letting a rider ride the bike as usual, and ensuring the rider did not suffer discomfort, and not letting their normal pedaling actions damage any of the cables or new components.

Results: The rider did not get injured, and that he did not hit the gear shifting wires from the pedal, nor did he unintentionally damage the gear system or Microcontroller case.

3) *Feature 2: Measure User Performance*:

a) *Sensor Passing Test*

Feature 2 is meant to "Measure User Performance," and in order to allow for a reasonably accurate measurement of pedaling revolutions, this test had to verify that the new Reed Switch Array would have their magnet passes seen correctly.

Procedure: This was a test to check which arrangement of the Reed Switch Array would provide the best means for data recording. The way this was done was to place any number of magnets on the pedaling wheel, and perform bicycle pedaling as usual to check how well of a read was done: see if in each revolution, for two or more points, see if a consistent distance was attainable and a magnet pass was short enough. If double the revolutions or passes were read, the magnets would have to be realigned by virtue of taking too many points.

Results: The best arrangement was to allow two magnets on the wheel, aligned in such a way that they were equidistant (180 degree arc, intersecting at the pedal center) and aligned parallel to the Array's Receiver to allow the best passing possible.

b) *Baud Rate Test*

To allow for the best communication of data between the Microcontroller of Feature 2 and the Software Program of Feature 3, a Serial Connection over a USB port was established, and to allow for Feature 3's promise of providing a "Real Time Exercise Display," the connection speed would have to be fast enough to allow all the data to be received, and processed through preset mathematical equations on the Software Side.

Results: A baud rate of 9600 bits per second (0.1 ms per bit) was considered insufficient, as the displayed metrics on the Software Side came almost a full second after pedaling action and provided inaccurate "after the fact" results. The determined baud rate that let the entire system work was 115200 bits per second, allowing each bit to be sent in roughly 8 microseconds.

c) *RPM Cadence Test*

Being a higher quality test of Feature 2, this allowed for a baseline analysis to test actual revolutions per minute accuracy.

Procedure: A rider plays a Metronome at a certain Beats Per Minute (BPM) quantity and pedals according to the beat. If the RPM quantity at that time is accurate within a 10% tolerance depending on the rider's skill, the test would be called successful.

Results: Using Cadences of 40, 60, 70 and 100 BPM, the riders pedaled to the beats to the best of their ability, maintaining a range of 7 BPM over and under the designated cadence. This test proved that at least the Revolutions Per Minute timing was considered correct, and one ruled out input that would adversely affect the speedometer or distance calculations in Feature 3.

4) *Feature 3: Real Time Exercise Display & Other Software*:

The Lapcycle Java program displays six values in order for the user to monitor their workout. These were calculated based on the two values read in by Feature 2 (RPM & resistance level) and by several constants (the force required to move the pedals at all five resistance level & the diameter of the pedal wheel. The results were measured by allowing the program to run for a period of time and checking the values against the mathematical formula used (see Table 9.) The value of the calories had to be measured for each resistance setting the bike was set at during a workout session and totaled together. The last two features required binary test results to confirm that all the buttons, labels, and file read/writes did not fail.

TABLE 9
MATHEMATICAL TESTS CONDUCTED ON FEATURE 3.

Metric	**Formula**	**Procedure**	**Results**
Time (hh:mm:ss)	Incrementing second counter	Program run for over 1 hour. Confirmed seconds convert to hh:mm:ss format.	Pass
RPM $(\frac{Rev}{min})$	Read from microcontroller	See Sub-subsection C.3.	Pass
Revolutions	revs = revs + rpm/60	Truncated decimal value matches manual count.	Pass
Speed $(\frac{km}{h})$	$S = rpm * .001995km * 60s$	Pedaled at low and high speeds, checking for even change in reading.	Pass
Calories (c)	$c = c_g + F * revs * .001995km$	Confirmed changing gears does not reset caloric output.	Pass
Distance (km)	$D=D_0+rpm/60s*.00195$ km	Value displayed at end of session matches mathematically calculated value.	Pass

IX. CONCLUSION

Obesity, combined with poor health habits, is not an issue that can be solved quickly or with a single product. The massive increase of electronic media usage (particularly the use of computers) has impacted people of all ages in both work and entertainment. There exists many programs and products designed to encourage people to leave behind their computers while getting more exercise. But Team 10 designed Project Lapcycle with the idea in mind not to try fighting against the computer usage habits and requirements that have become so prevalent. Instead, the aim of the design was to work exercise into their normal computer usage. Lapcycle contains five features designed to maximize its effectiveness and consumer interest. **Feature 1**: Single Person Spin Bike is the starting point of the device; a recumbent cardio device that is capable of comfortably supporting the user without the requiring the use of their hands. It has been equipped with a four point articulating mount capable of supporting a wide range of laptops and tablet computers complete with a peripheral mouse. **Feature 2**: Measure User Performance uses sensors, a servo, and a microcontroller to allow data collection and control common to most cardio devices. **Feature 3**: Real Time Exercise Display uses a multiplatform Java program to process the data collected by Feature 2 into six exercise metrics as a display on the user's own laptop, allowing them to use their computer as normal while monitoring their workout. **Feature 4**: Attractive & Easy to Use GUI arranges the display into a clear looking image. **Feature 5**: Encourage Fitness uses the graphical displays of Feature 4 to encourage users to keep using the machine. After nine months or work, the efforts of Team 10 have constructed a workable prototype that demonstrates the potential and market viability, making it a worthwhile endeavor to pursue for further development. Building a final prototype from the ground up with dedicated hardware and a custom driver is a worthwhile endeavor for the team to take with them as they move out of undergraduate school and into industry.

X. Works Cited

[1]: 247erg. "LifeSpan C3-DT7 Bike Desk". 2016. [Online] Available: http://www.247ergo.com/product/lifespan-c3-dt7-bike-desk/?gclid=CO2i6-2G78oCFQusaQod6gYDzQ

[2]: Andreyeva, T., Ringel, J., Sturm, R. Rand Health. 2004. "The Shape of America's Future – Is There Really an Obesity Epidemic?" http://www.rand.org/health/feature/forty/obesity_trends.html. 2004.

[3]: Beltan, A. California State University, Sacramento. Computer Engineering 191. 2016.

[4]: J. Bennett. "Television Studies Goes Digital." *Cinema Journal*, Vol. 47, No. 3, pp. 158-165. Spring 2008.

[5]: Chadwick, J. California State University, Sacramento. Computer Engineering 191. 2016.

[6]: Chadwick, J. "Nintendo Entertainment System Dissection". YouTube. 2015. [Online] Available: https://www.youtube.com/watch?v=Vk8H9BOo6Ss

[7]: Chen-Stute, A., Kellmann, M., Kopczynski, S. "Attitudes Towards Physical Activity and Exercise Participation– a Comparison of Healthy-Weight and Obese Adolescents". Ruhr-University Bochum, Faculty of Sport Science. Bochum, Germany. 2014.

[8]: Church, T. et al. "Trends over 5 Decades in U.S. Occupation-Related Physical Activity and Their Associations with Obesity." [Online] Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0019657. 2011.

[9]: Cybex. "525T Treadmill. 2016. [Online] Available: http://www.cybexintl.com/treadmills-cybex-525t-treadmill.aspx

[10]: Gardner, Amanda. "Sedentary Jobs Helping to Drive Obesity Epidemic". [Online] Available: http://consumer.healthday.com/fitness-information-14/misc-health-news-265/sedentary-jobs-helping-to-drive-obesity-epidemic-653346.html. 2011.

[11]: C.B. Goldfarb. "How Changes in the Economics of Broadcast Television are Affecting News and Sports Programming and the Policy Goals of Localism, Diversity of Voices, and Competition." *J. of Current Issues in Media and Telecommun*, Vol. 3, No. 2, pp. 111-144. 2011.

[12]: Grazer, J. M., Dr. "Body Composition". http://www.drgrazer.com/bodycomposition.htm. 2005. http://www.drgrazer.com/. 2005.

[13]: Gorin, A., Phelan, S., Raynor, H., Wing, R. "Home Food and Exercise Environments of Normal-weight and Overweight Adults". American Journal of Health Behavior. 2011.

[14]: Gym Store. *Bladez Fusion GS Indoor Cycle*. 2011. [Online] Available: http://www.gymstore.com/bladez-fusion-gs-indoor-cycle-p/bladez-gs-fus.htm.

[15]: Inside Trainer. "Under Desk Elliptical". 2016. [Online] Available: https://www.theinsidetrainer.com/14153/under-desk-elliptical/

[16]: Kamenetz, A. "Why Video Games Succeed Where The Movie And Music Industries Fail". Fast Company. 2013. [Online] Available: http://www.fastcompany.com/3021008/why-video-games-succeed-where-the-movie-and-music-industries-fail

[17]: The Lancet, "Body-Mass Index and Cause-Specific Mortality in 900,000 Adults: Collaborative Analyses of 57 Prospective Studies". http://www.sciencedirect.com/science/article/pii/S0140673609603184. 2009.

[18]: "Jimb0". "RedBoard Hookup Guide." Sparkfun. 2016. [Online] Available: https://learn.sparkfun.com/tutorials/redboard-hookup-guide/meet-the-redboard

[19]:Marcy ME 709 Recumbent Exercise Bike. Amazon.com. 2015. [Online] Available: http://www.amazon.com/Marcy-709-Recumbent-Exercise-Bike/dp/B00RM2QYSG

[20]: Y.M Liou, et al. "Obesity among Adolescents: sedentary leisure time and sleeping as determinants." *J. of Advanced Nursing*, Vol. 66, No. 6, pp. 1246-1256. 2011.

[21]: R. Lowry, et al. "Television Viewing and its Associations with Overweight, Sedentary Lifestyle, and Insufficient Consumption of Fruits and Vegetables Among US High School Students: Differences by Race, Ethnicity, and Gender." *J. of School Health*, Vol. 72, No. 10, pp. 413-421. Dec. 2002.

[22]: V. Luckerson. Time: Fewer People Than Ever are Watching TV. http://time.com/3615387/tv-viewership-declining-nielsen. 2014.

[23]: Martidale, J. "PC Sales are Slumping, But PC Gaming is Booming". Digital Trends. [Online] Available: http://www.digitaltrends.com/computing/pc-gaming-expected-to-drive-pc-sales.

[24]: Mayo Clinic. "Heart Disease". [Online] Available: http://www.mayoclinic.org/diseases-conditions/heart-disease/basics/definition/con-20034056. 2014.

[25]: Mayo Clinic. *Heart Disease*. 2014. [Online] Available: http://www.mayoclinic.org/diseases-conditions/heart-disease/basics/definition/con-20034056.

[26]: Mayo Clinic. "Type 2 Diabetes". [Online]. Available: http://www.mayoclinic.org/diseases-conditions/type-2-diabetes/basics/definition/con-20031902. 2014.

[27]: National Geographic Society. "The Development of Agriculture". [Online] Available: https://genographic.nationalgeographic.com/development-of-agriculture. 2015.

[28]: National Institute on Aging. *Global Health and Aging.* 2011. [Online] Available: https://www.nia.nih.gov/research/publication/global-health-and-aging/living-longer.

[29]: Oracle. Javadoc Tool. 2004 [Online] Available : http://www.oracle.com/technetwork/articles/java/index-jsp-135444.html

[30]: OSEPP. *Plastic Gear Analog Servo - 360.* [Online]. Available: http://osepp.com/products/motors/plastic-gear-analog-servo-%E2%80%93-360/

[31]: Parker-Pope, Tara. "Less Active at Work, Americans Have Packed on Pounds**.**" [Online] Available: http://well.blogs.nytimes.com/2011/05/25/less-active-at-work-americans-have-packed-on-pounds/?_r=0. 2011.

[32]: RAD Cycle Products. *1103 Ultra Bike Trainer*. [Online]. Available: http://www.radcycleproducts.com/rad-products/trainers/5-1103-ultra-bike-trainer

[33]: Serial Terminal Basics - CoolTerm. Sparkfun. 2014. [Online] Available: https://learn.sparkfun.com/tutorials/terminal-basics/coolterm-windows-mac-linux

[34]: P. Reed, et al. "Problematic Internet Usage and Immune Function." *PLoS ONE,* Vol. 10, No. 8, pp. 1-16. Aug. 2015.

[35]: Polulu. *Tamiya 72001 Planetary Gearbox Kit*. [Online]. Available: https://www.pololu.com/product/70

[36]: Precor. *EFX® 885 Elliptical Fitness Crosstrainer.* 2015. [Online] Available: http://www.precor.com/en-us/commercial/products/cardio/ellipticals/efx-885-elliptical-fitness-crosstrainer.

[37]: Richter, F. American Teens are Turning Their Backs on TV. http://www.statista.com/chart/3613/tv-usage-by-american-teens. 2015.

[38]: Savov, V. "The PC industry is betting big on gamers". The Verge. 2015. [Online] Available: http://www.theverge.com/2015/9/7/9271463/lenovo-asus-acer-intel-pc-industry-ifa-2015.

[39]: Slovinec, M. et al. "The Roles of Self-Efficacy and Motivation in the Prediction of Short- and Long-Term Adherence to Exercise Among Patients With Coronary Heart Disease". University of Ottawa. 2014

[40]: Tao, D. "The Afterburn Effect: Keep Burning Calories After a Workout". Greatest. [Online] Available: http://greatist.com/fitness/afterburn-effect-keep-burning-calories-after-workout. 2012.

[41]: Target. "Stamina® InStride® Pro Electronic Stepper". InStride. 2014. [Online] Available: http://www.target.com/p/stamina-instride-pro-electronic-stepper/-/A-12153666?ci_src=17588969&ci_sku=12153666&ref=tgt_adv_XSG10001&AFID=google_pla_df&CPNG=pla_sports+shopping&adgroup=stepper_sc&LID=14pgs&KID=1382ce38-3e7b-4a4d-aa05-70dc4cd037ad&kpid=12153666&gclid=CLWo-ZXflcgCFUhufgod6lgCbg

[42]: Trend Hunter Tech. "FitDesk". [Online] Available: http://www.trendhunter.com/trends/fitdesk

[43]: United States Department of Health. *Let's Move*. 2015. [Online] Available: http://www.letsmove.gov.

[44]: Farnell element14. *Arduino Uno*. [Online]. Available: http://www.farnell.com/datasheets/1682209.pdf

X. Glossary

A

- Active Planner: An individual who has poor exercise and/or eating habits, but is planning to start making changes to improve their fitness level.
- Adipose Tissue: The tissues in tetrapods that store triglycerides.
- Atherosclerosis: The thickening of arterial walls caused by the build of fat, cholesterol, white blood cells, and various minerals.
- Application: A computer program designed to maintain system files, provide user interface, or accomplish a task.

B

- Body Composition: A measure of physical health that measures the percentages by weight of body fat, lean, and water.
- Body Mass Index (BMI:) A measure of physical health in units of kg/m^2.
- Boolean Logic: Determination of either true or false.

C

- C++: An update to the original C language providing new libraries and the ability to program in an object oriented style.
- Calorimeter: A device designed to measure work in units of calories.
- Camber angle: The lateral tilting angle on a bicycle, relative to the angular position where the bike handles are perfectly level.
- Carbohydrate: Molecules composed oxygen and hydrogen atoms of rings of six carbon atoms. This are found in various forms, in single rings or in chains of two or more.
- Cardiovascular Disease: The hardening and narrowing of the blood vessels that eventually can lead to atherosclerosis and infarctions (blood clots.)
- Cardiovascular Exercise: Repetitive full body activity designed to keep a person's heart rate elevated for an extended period of time.
- Cardiovascular Fitness Machines: Exercise machines designed to engage the user in low impact long term activities in order to burn calories and strengthen the cardio and respiratory systems.
- Crash: A condition in which a program experiences and irresolvable conflict and terminates.

D

- Derailleur: The components on bicycles with multiple gears used to move the chain across the different sized sprokets.
- Diabetes: A condition in which the body is unable to properly manage its blood sugar level.

E

- EEPROM: Electrically Erasable Program Read-Only Memory. A memory unit that only has its contents deleted from an electrical signal, and has the contents stay permanently over time, even when unpowered.
- Elliptical: A cardio machine that simulates walking, climbing, and running, in a low impact manner that moves the user’s feet in an elliptical motion.
- Epidemic: A disease which spreads to a large amount of a population in a short period of time.
- Exception: A condition the occurs when there is a conflict between a running program and memory in which the address range has been exceeded, the expected form or size of the data is different, or the data doesn't exist.

F

- First World: Non theocratic democracies with market based economies that are relatively wealthy and stable.
- Force: The interaction that causes change in motion of an object. The product of mass and acceleration.
- Force Gauge: A measurement tool that monitors forces or weights that push or pull on a given object.

G

- Glucose: A sugar composed of six carbon atoms, twelve hydrogen atoms, and six oxygen atoms.
- GUI: Graphical User Interface. A method of allowing interaction with a computer program using, images, menus, buttons, and other non-text features.

H

- Hall Effects Sensor: A magnetic pulse sensor used to detect motion by the presence of a magnetic source. See Reed Switch.
- Hydrocarbons: Molecules composed of carbon and hydrogen.
- Heart Disease: A condition that effects the structure and function of the heart due to genetic and/or lifestyle conditions.

I

- Insulin: A protein released into the blood stream from the pancreas which bonds to glucose allowing it to be received by cells.

J

- Java: High level multiplatform programming language.
- JFrame: A class that inherits from the Java Frame class and is responsible for the creation of display window that house the various graphical components.

L

- Liver: An internal organ found in most vertebrates which is vital for keeping the body clean from toxins and storing glucose.

M

- Metabolism: The process by which cells respirate by converting glucose into energy.
- Microcontroller: A packaged CPU with I/O pins and built in libraries designed for customizable tasks.
- Multithread: A CPU performing tasks simultaneously or pseudo-simultaneously.

O

- Obesity: An excess of body fat.
- Object: An instantiation of a class in an object oriented programming language.
- Object Oriented Programming: A method of programming allowing the creation of custom inherited classes.
- Overweight: Having an excess of weight including fat, muscle, bone, and water.

R

- Race Condition: In multithreaded programming, creating two or more threads that write/read to the same memory addresses might cause one of them to read outdated or invalid values.
- Reed Switch: A switch controlled by a non-contact magnetic field controlling ferrous (iron-based) material.

S

- Sarcomere: Protein structures within muscle cells that consume energy and react with potassium and sodium to cause movement.
- Sensor: Device designed to respond to external stimuli in a manner which can be processed by a computer.
- Serialization: The translation of a data structure or data object into a stream of bits in order to store information to memory, database data, or file form.
- Spin Bike: Stationary cardio device that simulates the physical strain of riding a bicycle.
- Stationary Bicycle: Similar to a spin bike, but it can be peddled backwards.
- String: A class in Java and C++ that handles a series of multiple characters.
- Sunk Cost: an economics term meaning a cost that has already been incurred which cannot be recovered.

T

- Treadmill: A cardio machine that simulates walking and running at a variable incline.
- Triglycerides: Commonly referred to as fats, they are composed of three fatty acids boded to a glycerol.

U

- UML: “Unified Modeling Language,” a visualization of a system design typically used in Software Engineering.

Appendix A: User Manual

I. Introduction

This is the operations guide for usage of the production version of Project Lapcycle. This guide exists for any persons who wish to setup and operate the prototype in its current state. For users interested in how the prototype was developed, what design choices were made and why, or for maintenance solutions, refer to Appendices B and C.

II. The Pre-Launch Checklist

As soon as you have received the Lapcycle, check that you possess all of the following materials for full operation:

- Marcy ME 709 Recumbent Exercise Bike
    - This bike should be modified in the manner described in Appendix B.
- 1 USB 2.0 Type-A to Type-B cable
- 1 Software Installation Disc
- 1 Personal Laptop or Tablet
- 4 AA 1.5V Batteries

Software System Requirements:

- Operating System: Windows Vista 7, 8 or 10, Mac OS X Leopard and above, any Linux distribution.
- Java JDK 4.7. Please note that depending on the changes that Java usually has as of this writing, this specific version *will be required* until further notice.
    - Make sure it matches the version of your operating system. If you use the 32b version when you have the 64b version, the Lapcycle program may not work.
- Java RTE 8.91 if you have the 32b version of an operating system pr 8.92 if you have the 64b version.

III. "Out of the Box Operation"

The Lapcycle comes out of the box with a complete recumbent Marcy ME-709 bicycle frame fully modified and assembled, with a black box housing an external data recording device. The left side armrest holds a pair of buttons that shift the gears for the spin bike into five different resistance settings. The black box has the capability to save the setting for the gear shifting mechanism between uses. While the control device is powered by the USB line connecting it to the user's laptop or tablet, the servo used to adjust the resistance will not function unless the battery compartment is equipped with four double A batteries. If the batteries are not present, the program will behave normally, but pressing the buttons will not actually change the resistance from the level it was left at the last time the system was correctly powered.

Keep in mind that when you power on the Lapcycle without a laptop connected, you can treat this as any other stationary bicycle and can use the desk rest for books or other forms of entertainment, but will be unable to record any data. The buttons and servo will work if the batteries are present as previously described and either the USB connector is attached to a 5V wall charger or there is a power adapter supplying 7-15V with a 5.5mm/2.1mm center-positive barrel connector inserted into the Redboard power jack [18]. Once this is done, the servo will adjust the resistance as normally when the buttons are pressed until it reaches the minimum or maximum setting.

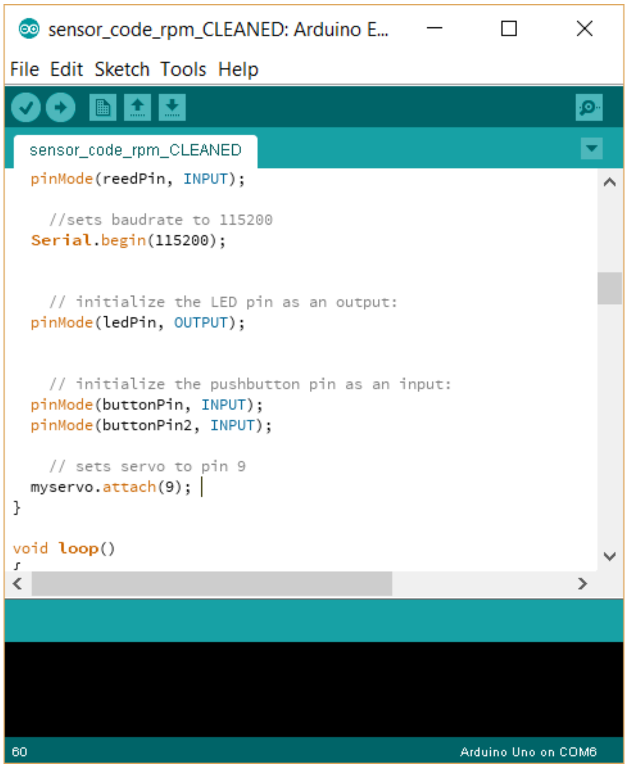
IV. First Time Run / Install Guide

To utilize the full features of the Lapcycle, you will need to connect your laptop to the Type-A connector on the USB line and connect the attached microcontroller port to the Type-B connector on the same line. If you have already installed the software, skip this section and proceed to Section V.

Procedure:

1. Copy all files from the disc into any directory that you wish.
2. *Run *arduino.exe*. (see Figure A-1.)
3. *Click on the **File** drop down menu and select **Open**.
4. *Navigate to the directory where you saved files from the CD.
5. *Select sensor_code_rpm_CLEANED.ino.
6. Run Coolterm.exe (see Figure A-2.)
7. Click on the **Connection** drop down menu and select **Options** (see Figure A-3.)
8. **Port** should only contain the serial port numbers that have serial lines connected to them.
9. **Baudrate** should be set to *115200*.
10. Click the **Connect** button on the tool bar.
11. Click on **Connection** drop down menu
12. Scroll down and highlight **Capture to Textfile**.
13. Scroll right to **Start**.
    a. Once this is complete, pedaling should cause two integers separated by a space two time per revolution.
14. Navigate to the directory where you saved the files from the CD.
15. In the **File name** text box type in out.txt and click **Save**.
16. Run **LapcycleBeta1.exe**.
17. Enter a profile name with the corresponding password (case sensitive) and click **OK**.
    a. If the program has never been run before or you desire to create a new profile, click on **New Profile**.
    b. Enter your username, password, and confirm the password.
    c. Click on **OK**.

*These steps only need to be completed the first time the Lapcycle is run or if the microcontroller reset button has been pressed.



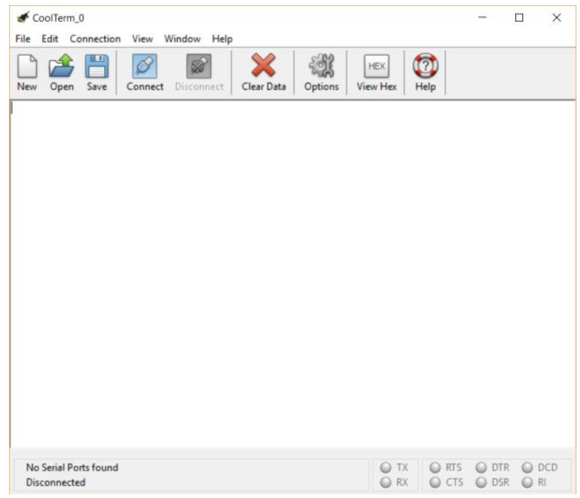


Figure A-1: Arduino.exe with the sensor code open [44]. Figure A-2: Coolterm.exe [33].

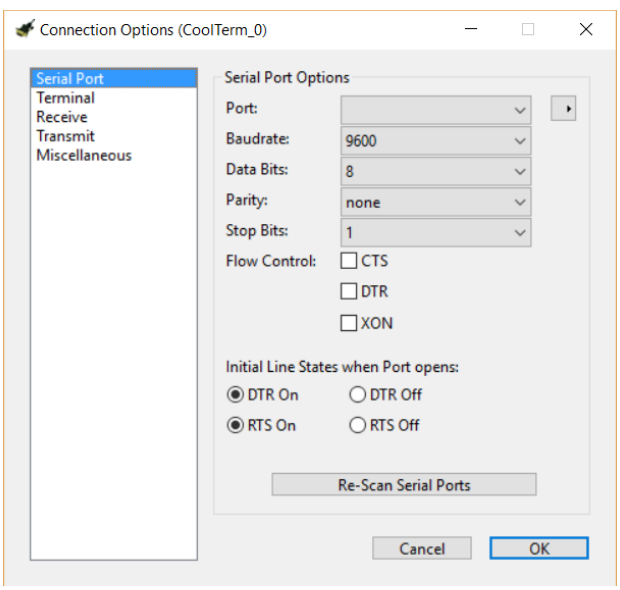


Figure A-3: Coolterm Options window [33].

V. STANDARD OPERATING PROCEDURE

This section handles everyday use of the Lapcycle.

When the Lapcycle Program is opened on the computer, the first thing you will have to do is to create a username and password, along with setting the difficulty level for use. The difficulty lowers or raises the requirements for achievements. After this is done, you can start pedaling straight away. Figure

A-5 provides what the login window should look like, and Figure A-4 provides the main window. You are capable of pedaling and the data recording can happen in the background. If either of the windows in Figure or Figure appear, click on the **Terminate** button. These windows will appear if you have not started peddling or have been stationary for five seconds.

Warning: if you are playing games while the Lapcycle program is running in the background, we strongly recommend that you play the game in windowed mode if possible. Pop-ups from the program can interfere with normal playing operation, leading to unintended consequences such as force closing the game or being detected from an anti-cheat system

Figure A-4: Main display window during session [5].

VI. THE PROFILER SYSTEM

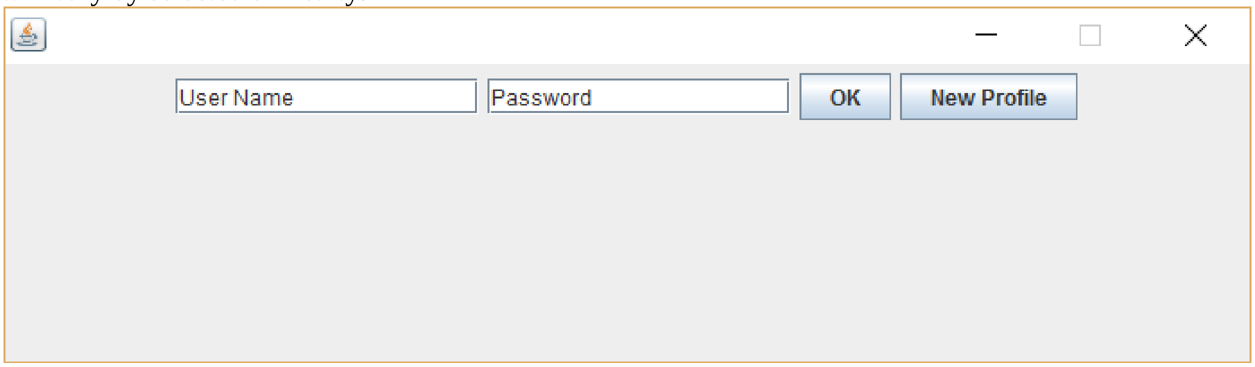
As mentioned in the previous section, the Lapcycle program does inquire for a username and password (see Figures A-4 & A-5). The information from the program is saved onto a text file so it can be reread into the program the next time the profile is loaded. The CalculationsBeta1 class also outputs the results of the last time the machine was used for review and the date and time it was accessed. This data is stored in Output File.txt.

The text file generated by the Profiler contains the following information:

- The User Name and Password
- Distance Traveled from start of profile
- Time Spent on Bicycle.
    - Without Activity (Idle Time)
- Calories put into the machine from start of profile
- Revolutions (360* pedal rotations) made since the start of profile

This profiler has two other components:

- The Prompting System, which will react depending on the amount of idle time on the bicycle, which will generate an image as shown in Figure A-3.
- The Progress Achievement Tracker, which will show achievements similar to those found in post-2000 video games as a means to reward users as a means to show progress. These achievements will vary by selected difficulty.



-
- Figure A-5: Profile Login window [5].

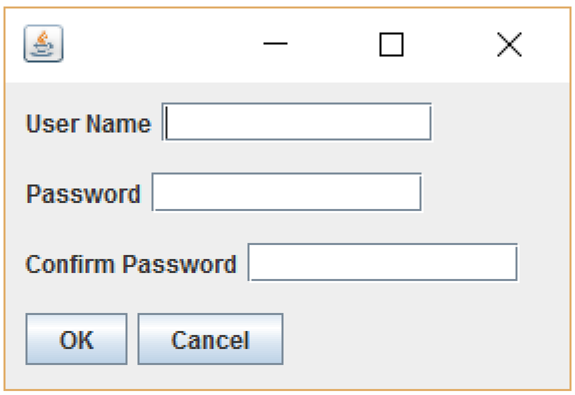


Figure A-6: New Profile creation screen [5].

VII. GENERAL TROUBLESHOOTING

A. *LapcylceBata1.Java exists with file reading exception*:

Confirm that Coolterm.exe is connected and has started saving to text file. If it already is saving to a text file, make sure the file is named out.txt.

B. *LapcycleBeta1.Java exists with array out of bounds exception*:

This may occur if the black box did not finish writing the entire line to the file. It is an extremely rare occurrence and only occurred during the entirety of development and testing a total of four times. Restart the program and see if it continues to occur. If it still happens, check subsection C.

C. *Coolterm.exe displays vertical lines when pedals are moved*:

The Baudrate has not been set to the correct value. Make sure it is set to 115200.

D. *Coolterm.exe does not show any options for the Port selection*:

Check USB cable integrity on both ends. If both ends are connected correctly, try connecting it to a different USB port on the laptop. If this fails to solve the problem, there may have been a hardware failure in either the Lapcycle or the laptop's USB controller. For the latter, check the USB drivers and test them with known good USB connected peripheral

Appendix B: Hardware Development Documentation

I. INTRODUCTION

Project Lapcycle's framework is a stationary bicycle integrated with a data recording system. Internal modifications have been made to the Marcy ME-709 Recumbent Bicycle, including the replacement of an internal sensor with a dual-magnet Reed Switch Array that provides a wheel passing signal, a mechanical servo controlling the magnetic wheel resistance and setting up a five-gear system, and the replacement of the bike's dashboard with an Arduino Uno series Microcontroller, which sends data over a Serial Port at a baud rate of 115200 bits per second. This sent data is later processed by a Software Program onboard a user's laptop, which uses data points of Gear Level and Revolutions per Minute to derive and calculate distance and speed.

II. BLOCK LEVEL VIEW

Figure B-1 is a block level diagram which shows off all the hardware components used.

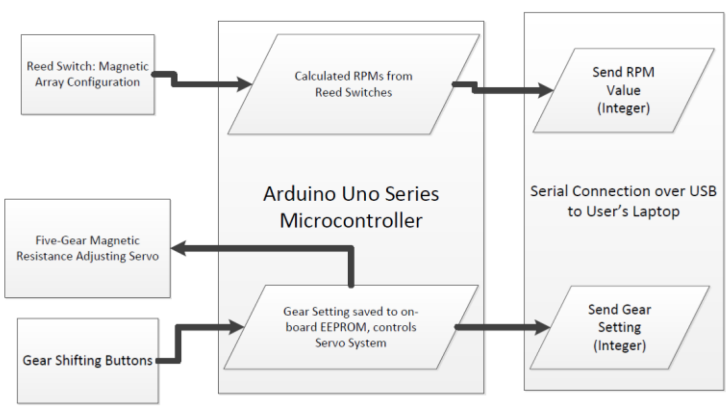


Figure B-1: The Block Level View

III. COMPONENT LEVEL DISCUSSION

A. Reed Switch Array

1) Technical Overview

The optical sensor that came with the bicycle was disconnected and replaced with three components: one Reed Switch Receiver similar to those mounted on home alarm systems, along with two corresponding magnet bars. All are each one inch in length. The purpose of the Reed Switch is to read a pass on the wheel, and when both magnets are in the same field, the circuit it is connected to is complete, sending a signal indicating a pass. Said signal is composed of a single five volt pulse sent to an Arduino Uno that converts an analog value to a digital high or low. In our case, any value above 3.3 volts is considered a "HIGH" and any value below that is a "LOW." We found that the average rise time of the reed switch is 1.6 milliseconds with a maximum value of 2 milliseconds and a minimum of 0.8 milliseconds. This average rise time of 1.6 milliseconds means that the default ADC sampling rate of 9615 Hz from the Arduino Uno is more than enough to accurately sample the system. Figures B-2 through B-5 provide relevant data regarding the use of the magnets through the oscilloscope function on an Analog Discovery, where the voltages, periods and rise times are measured on pedaling cadences of 40, 60, 70, and 100 beats per minute. Table B-1 provides a Rise Time Per Cadence Relationship.

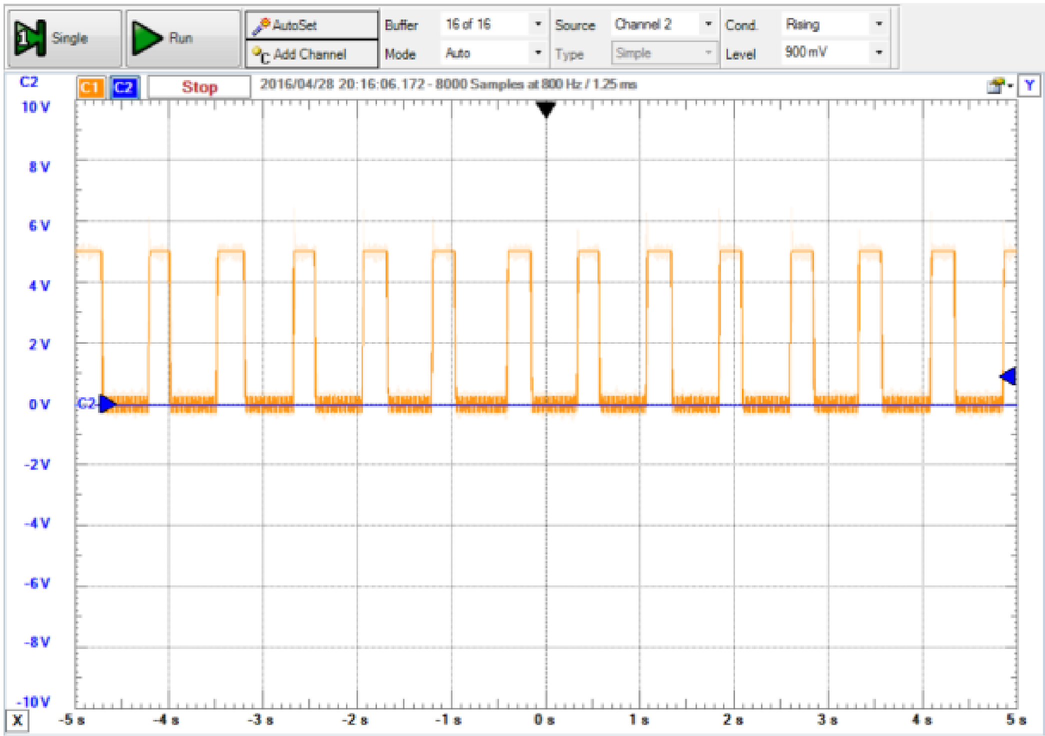


Figure B-2: The 40 BPM Waveform.

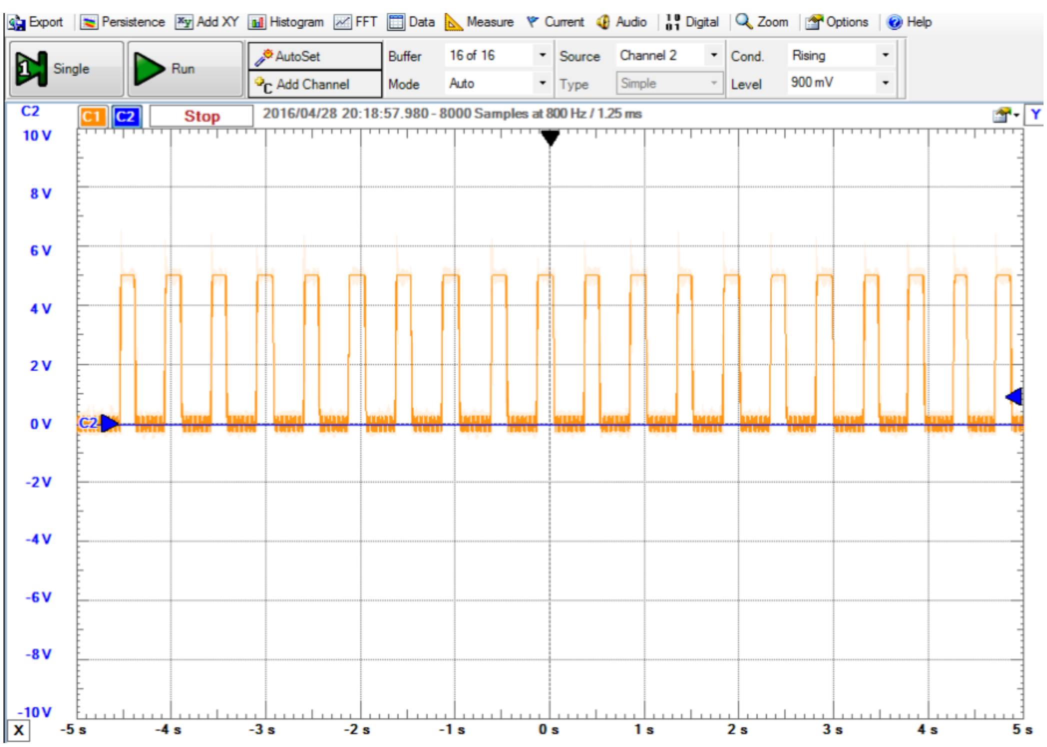


Figure B-3: The 60 BPM Waveform.

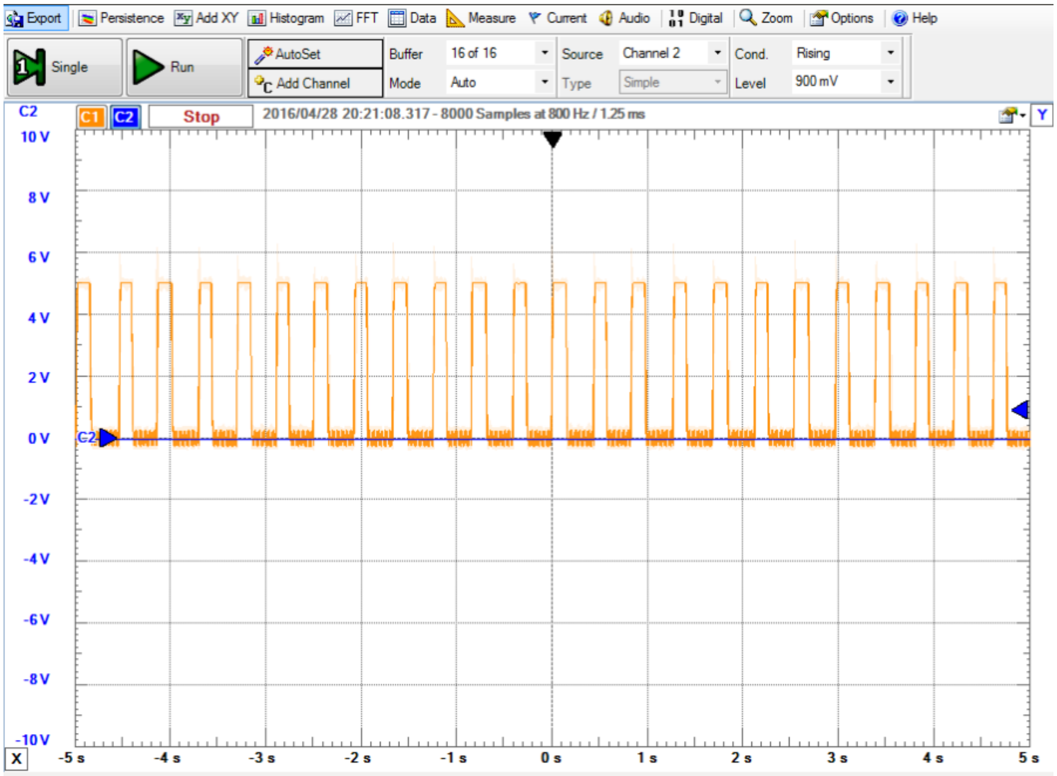


Figure B-4: The 70 BPM Waveform.

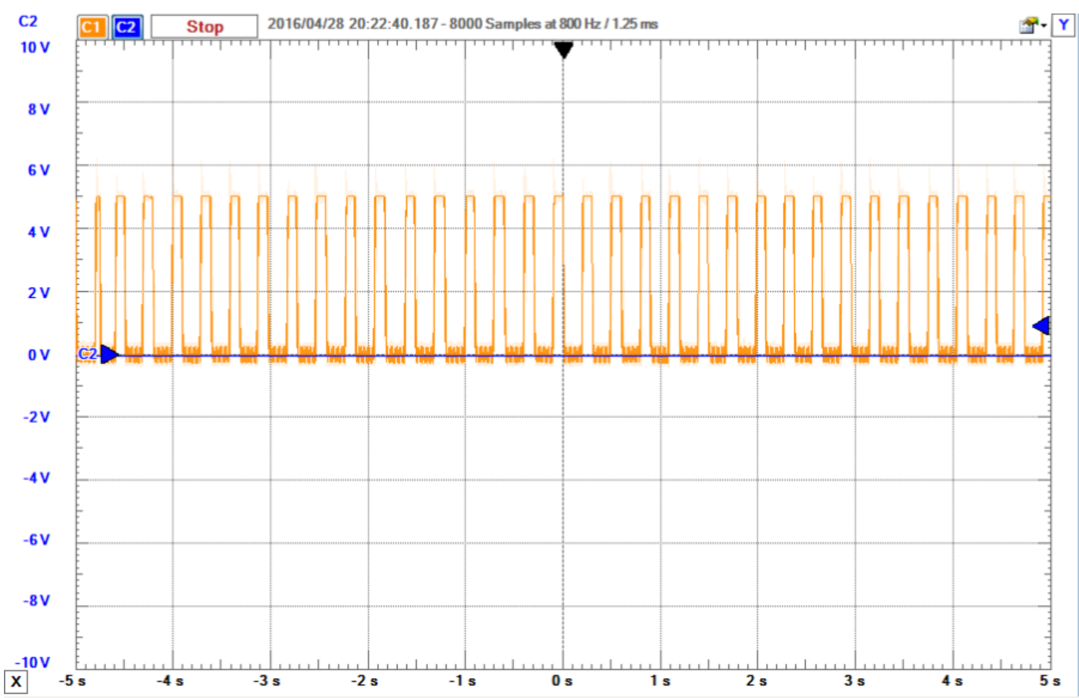


Figure B-5: The 100 BPM Waveform.

Table B-1:
Cadence RPM/Rise Time Relationship

RPM	Min Rise Time (ms)	Max Rise Time (ms)	Avg. Rise Time
< 5	N/A	2.2	2.2
> 130	0.8	N/A	0.8
30	2.01	1.02	1.58
60	2.11	1.02	1.61
70	2.04	1.04	1.67
100	2.1	1.02	1.65

2) Physical Installation Notes

At first, one bar was used, but the rate of return was considered too error prone to consider a full revolution calculation correctly, due to an easily abused method of shifting the pedals back and forth along the magnet. The consensus reached to read a full revolution more reliably was a two-bar configuration on the wheel. Attaching three bars or more affected the magnetic polarity of the metallic wheel to the point where the pass signal would be stuck high for three fourths of a revolution.

Figure B-6 provides a photograph of the array attachment. Both magnets were aligned in a 180 degree configuration and had one end facing the receiver to avoid a "double tap" situation, meaning, the receiver should not cross both ends of a magnet bar or result in more than one read per bar. An elaboration of "double tap" and polarity issues is in Section IV, Part A.

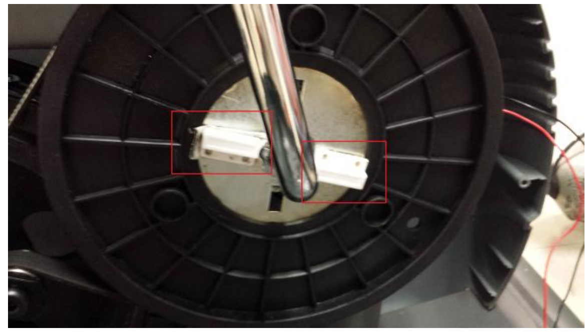


Figure B-6: The Finalized Reed Switch Array Configuration.

B. Gear Shifting System

1) Technical Overview

This component replaces the analog knob that controls the pedaling resistance of the bicycle. Replacing the knob is two standard push buttons located on one of the arm rests of the bike, providing a five-gear system. To measure the user effort from pedaling action, a manual force gauge has been attached to the pedal, and a reading was taken after one rotation. Table B-2 provides a gear and force relationship. These force values are used to help calculate an approximate calorie burning count.

Table B-2:
Gearbox Gear and Force Allocation

Gear Setting	Force Needed to Pedal (Newtons)
1 (measured)	7.78
2 (interpolated)	10.01
3 (interpolated)	12.23
4 (interpolated)	3.25
5 (measured)	16.68

2) Development and Installation

Development of the gear shifting system was not too difficult electronically. Said system consisted of the following components:

- One Servo
- Two Analogue Push Buttons
- One 25:1 Gearbox
- Six Volt Power Supply, used as a supply powered by 4 AA batteries
- The Arduino Uno (or a Microcontroller with EEPROM)

The Arduino is set to pick up a signal from either button such that it will spin the servo. The buttons send a five volt signal to the Arduino when pressed and signify an increase or decrease in pedaling resistance. The connected servo then spins a motor which pulls or releases a resistance cable attached to the spin bike (see Appendix D).

Figure B-7 shows the servo in action and the line that it is tugging. To see the other influenced components, the interior view of the bicycle is provided in Appendix D.

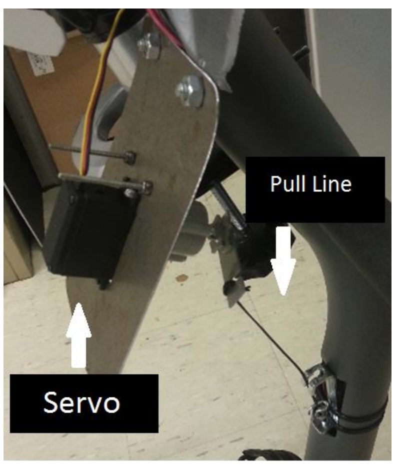


Figure B-7: The Servo and its connected line.

3) Memory Considerations

While the motor is spinning, the Arduino updates the gear level and saves the gear level to an on-board EEPROM unit, which is the Arduino's form of long term data storage. This is necessary since the servo does not have memory and cannot know the position the gear shifting mechanism is in, which could result in potentially pulling beyond the limits of the cable and snapping the line. Although it is not advisable to frequently write to the EEPROM due to the limit of 10,000 – 100,000 writes per sector before it deteriorates, we chose to implement writing to it since deterioration would not be reached during development of the project.

Should the positioning not be correct, the servo would need to be adjusted manually via the potentiometer screw located on the top of the servo. This means the servo will need to be made to turn to the lowest gear position. The calibration procedure is as follows:

- Detach the Gear Tugging Line.
- Press the Gearshift Down button until Gear State is at 1.
- Manually realign the servo until it is 90 degrees East from the North position.
- Reattach the Gear Tugging Line.

C. Arduino Uno Series Microcontroller

In order to connect and power the components displayed in Subsections A and B, a single Microcontroller was considered enough to do the job. The selected controller is the Arduino Uno, based off the ATmega128P processing chip. The Fritzing Diagram showing the connections is provided in Figure B-8A, and the actual system is showed in Figure B-8B.

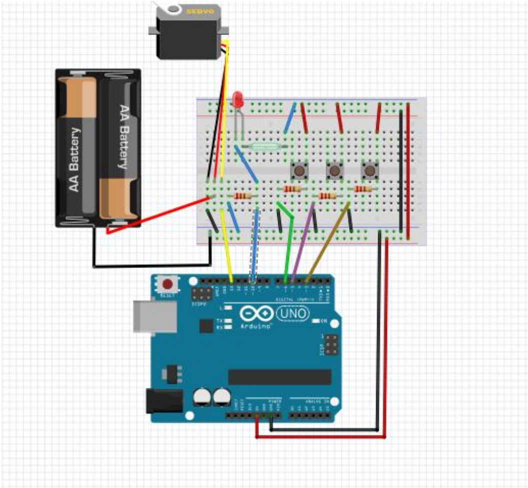


Figure B-8A: The Fritzing Diagram showing the Arduino Connections. Two of the buttons represent the Reed Switches, the third represents a reset state.

Figure B-8B: The Actual Connections.

1) *Selection Reasons:*

- All relevant inputs only required a 3.3 or 5.0 volt supply.
- Fewer than 10 digital pins were required.
- The 16 MHz clock was considered fast enough to get the input signals reliably.
- Wide platform support, and availability of prefabricated software packages.

2) *Proposed Alternate Solutions:*

The other candidate microcontroller considered for use was a Breakout Board variant of the Atmel ATmega32U4 processor as imaged in Figure B-5, however, development problems are explained within the next list:

- The first concerning issue was driver compatibility. Cross-platform development was considered unfeasible as the drivers to replace the bootloader-only status were originally packaged for one operating system.
- While not a deal breaking reason, the 32U4 could be programmed in the same vein as the Arduino Leonardo with the Arduino IDE, however, this was considered no different than actually using the Uno, only with the next problem.
- The last straw dealt more with the bootloader and the on-board programming tool. While attempting to upload new iterations of code, the reset state of the Breakout Board did not allow for a clean upload. What this means is that even though the reset line turned active allowing for a ten second window for the bootloader to take a new program, the COM Port taking the line would be considered “busy.” Eventually the workaround was to switch COM Ports between uploads, however, this was considered too great a hassle to manage.

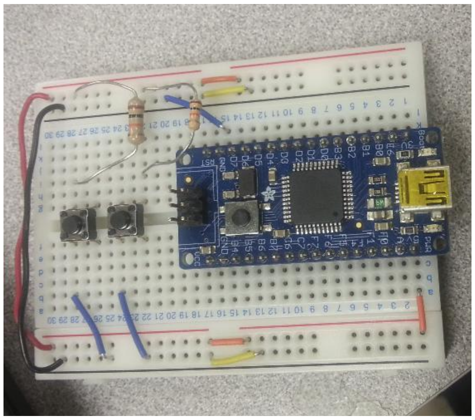


Figure B-9: A look at the Atmel ATmega32U4 Breakout Board.

3) Flow of Operations:

Figure B-10 provides the flowchart of operations on how the Arduino Uno handles the inputs. Considerations come on the list following the flowchart.

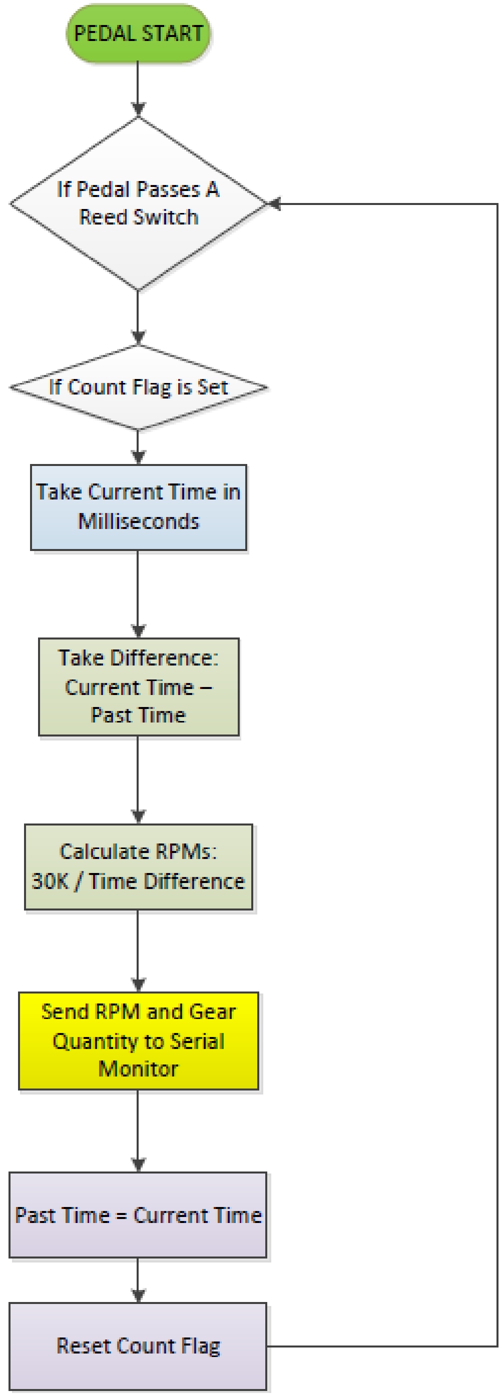


Figure B-10: Arduino Uno Flow of Operations.

- The first iteration of the program originally had a reed switch pass quantity and the gear quantity. However, due to the nature of the sensor reading and the delay constant being 1 second (1000 milliseconds), this turned out to provide severe miscalculations which would affect a derived RPM and result in an inaccurate speedometer.

- The current iteration sacrifices the reed switch pass counter for a revolutions per minute counter. In order to calculate the RPM the person is pedaling at, the Arduino keeps track of the time difference between passes. The Arduino uses its own millis() method to find the time at which a pass occurs, in milliseconds, and then subtracts it from the previous pass time. This give us the amount of time it takes the person to complete a half revolution, as highlighted in the next equation:

$$RPM = \frac{0.5\,Revs * 60K\,\frac{ms}{\min}}{\Delta ms\ between\ Passes} \quad (1)$$

D. *Serial Connection Methods*

1) *Proposed Methods*

The Arduino Uno has a Serial library such that information can be sent and formatted in a readable form, and has a USB Port, on which it can emulate a Serial Connection over a COM Port. In order to secure a bridge between the hardware components mentioned here and to the Software components in Appendix C, a connection monitor must be established. Since the Software Team decided to write their Recording Program in Java, the first solution sought was to use prefabricated Serial Connection code made in Java from the Arduino site, allowing for use of a Java RX/TX Library. However, despite the great advantages it presented, the Software Team deemed it too cumbersome to implement. The decided solution was to use a computer-side software program known as “CoolTerm” by Roger Meier, which would take the place of a Serial Monitor listening in on the COM Port, but has the capability of writing to a text file. Figure B-11 shows the display of CoolTerm.

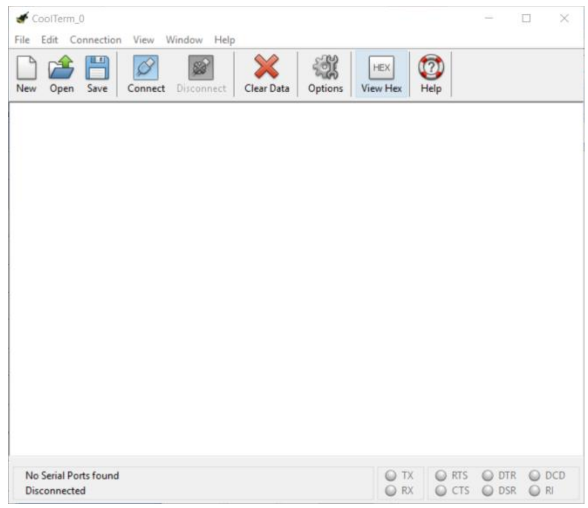


Figure B-11: The CoolTerm Window without input text.

2) *Disadvantages and Other Issues*

However, CoolTerm has one user-unfriendly disadvantage in context of the wider program: it requires an external stimulus such as an Auto Hotkey script to let it start writing to a text file.

In terms of recording issues that plagued the design regardless of implementation, the one outstanding one is the “zero read” case, as in, when the rider stops pedaling. The reason behind this is that when the rider stops, the instantaneous velocity should be zero, whether from speed or revolutions per minute. The deceleration read was difficult because if the rider stopped pedaling from a fast pace to just before passing another magnet, the last reported RPM value would be from the last pace and not at a zero value. The correction was to use a two second timing window such that the Software could know when to display a zero speed reading. However, this is not exactly foolproof because at sub-30 RPM cases, the zero speed reading can possibly apply. Adding more bar magnets to get more measuring points is a solution, however, the wheel that the magnets are mounted to must be made of a different material, for reasons defined in Section II, Part A-2.

IV. PERFORMED TESTS

These tests were of low to moderate quality, however they established a base line on how the hardware would operate. Multi-function tests that were tested on the Software Package are defined in Appendix C. The most relevant ones will be displayed in this section.

A. *Sensor Passing Test*

This test dealt more with the Reed Switch alignment and demonstrates why the Array Configuration in Section II, Part A has been arranged in its current matter.

As explained in that section, the original design was to use a single Reed Switch with its corresponding bar magnet in order to toggle a switch and increment a pass counter. However, chances for recording error became high, such as the back and forth abuse method mentioned before. The purpose of this test was to use multiple magnet bars to read a revolution in a more precise manner by checking more points on the wheel.

1) *Some considerations of the test:*

- As the bike being used had a metal wheel, the magnets would be directly attached without any need of adhesive tape. This raised the possibility of magnetizing a region of the wheel and completing the Reed Circuit too long to get an accurate pass.
- The alignment of the magnet bars relative to the Reed Switch Receiver had to allow for a “one pass, one read” rule. In some cases, a “double tap” was encountered as the Receiver would hit both the North and South Poles of the magnet bar. A suggested way to mitigate the polar issue was to count both taps per pass, but this required further time division and calibration for distinct sample points.

2) *Some Possible Improvements:*

- Remove the metallic plate the magnets are attached to. By removing the metallic plate, more magnets could be added to the device meaning more points of contact. As of this writing, only two magnets at a 180 degree configuration could be used before adverse polarization of the metal plate occurs. Adding more magnets would decrease the amount of time that the system does not know what the person is doing. Currently, the system only knows the speed of the user once the pedal passes one of the two sensor points. This means that there is a full 180 degrees where the system has no info on the user. Adding more points of contact will decrease this uncertainty drastically.

3) *The test results:*

- A four-bar intersecting configuration at 90 degrees separation resulted in a 270-degree arc of the wheel being read as a high reed sensor signal. Figure B-12 displays this configuration.

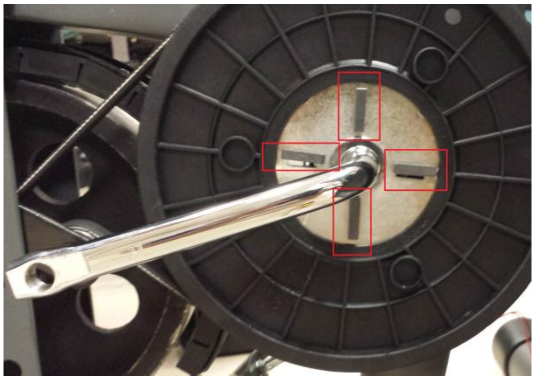


Figure B-12: The four-bar 90 degree setup.

- A two-bar configuration with the magnets aligned on the tangent of the wheel initially allowed for a single-pass method, however the signal eventually was stuck high late in development and required a 20 minute diagnostic. Figure B-13 displays this configuration.

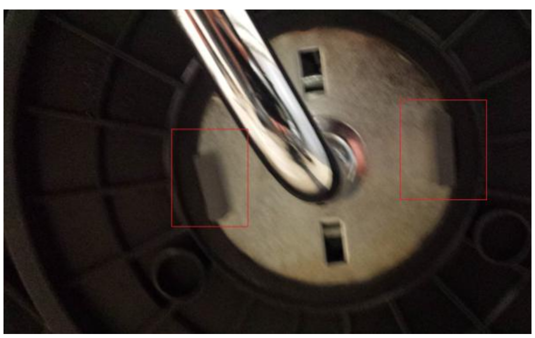


Figure B-13: The 2-bar parallel tangent setup.

- Eventually, the best possible result was the two-bar configuration intersecting at a 180 degree arc. This is shown back in Figure B-6.

B. Baud Rate Test

The second most notable test was due to the fact that the exercise display would need to be shown in "real time," or at least as fast as possible in a one second window, the idea was to find a Baud Rate quick enough to transmit all of the information. The standard Baud Rate used for typical serial transmissions was 9600 bits per second (104 microseconds per bit). For the amount of data to send, this was considered too slow. As USB 2.0 (which as of 2015 is still one of the most widely used peripheral connection methods) allowed for data transmission ranges in the 10^6 range, a bit of experimentation allowed a finding of an ideal rate.

1) *The Test Results:*

Repeated tests with the Serial transmission eventually yielded that 115200 bits per second (8.6 microseconds per bit) yielded the best compromise of data per time.

C. Revolutions Per Minute Cadence Test

Of all of these tests, the best defined was one that measured reliability of RPMs. This was done by using a Metronome application on a smart phone, pedaling according to the metronome, and checking if the recorded RPMs were timed correctly. Table B-3 provides an arrangement of this data. Upon recording the pedaling action on film, the RPM readings and pedaling were well within 10 percent of the metronome timing window.

1) *The Test Results:*

Table B-3:
Metronome/Cadence RPM Tests

Beats Per Minute (BPM)	Min. recorded value (RPM)	Max. recorded value (RPM)	Most repeated value (RPM)	% Diff.
40	37	41	39	2.5%
60	54	62	58	3.3%
70	71	77	73	4.2%
100	96	105	102	2.0%

The results of the Cadence Test can be seen on Table B-3 and it can be noticed that the calculated values can range by up to 7 RPM from the expected values of the metronome.

2) *Sources of Error:*

For this test, the ability of the rider affects the margin of error more than the accuracy of the reed switch system. This is because a trend of slowing down on the upswing of the pedal and a speed up with the down swing was very common during the testing. The rider would naturally slow down on the upswing of the rotation and then speed up on the down swing to try to keep on pace with the metronome. Since the sensor array is split to count two halves of a rotation, the recorded values came in pairs on a low reading and a high reading with the average being much closer to the desired values. This issue could be fixed with more experienced bike riders who can pedal with an even force at a desired pace.

APPENDIX C: SOFTWARE

I. *Block Level View of Software System*:

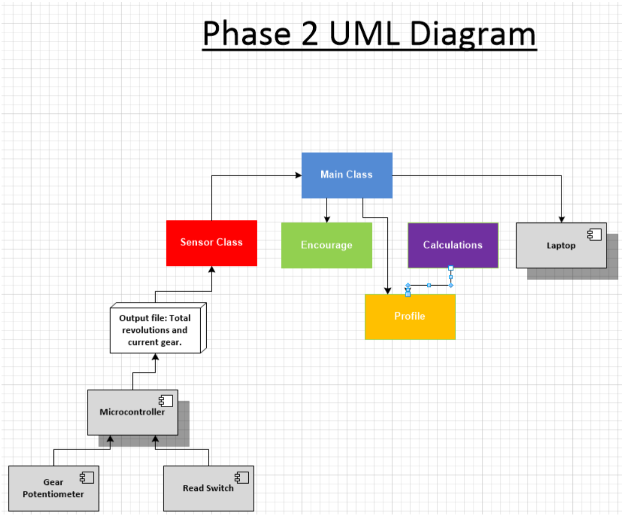


Figure C-1: Phase 2 UML Diagram [5].

II. *Subroutine Level Flowcharts*:

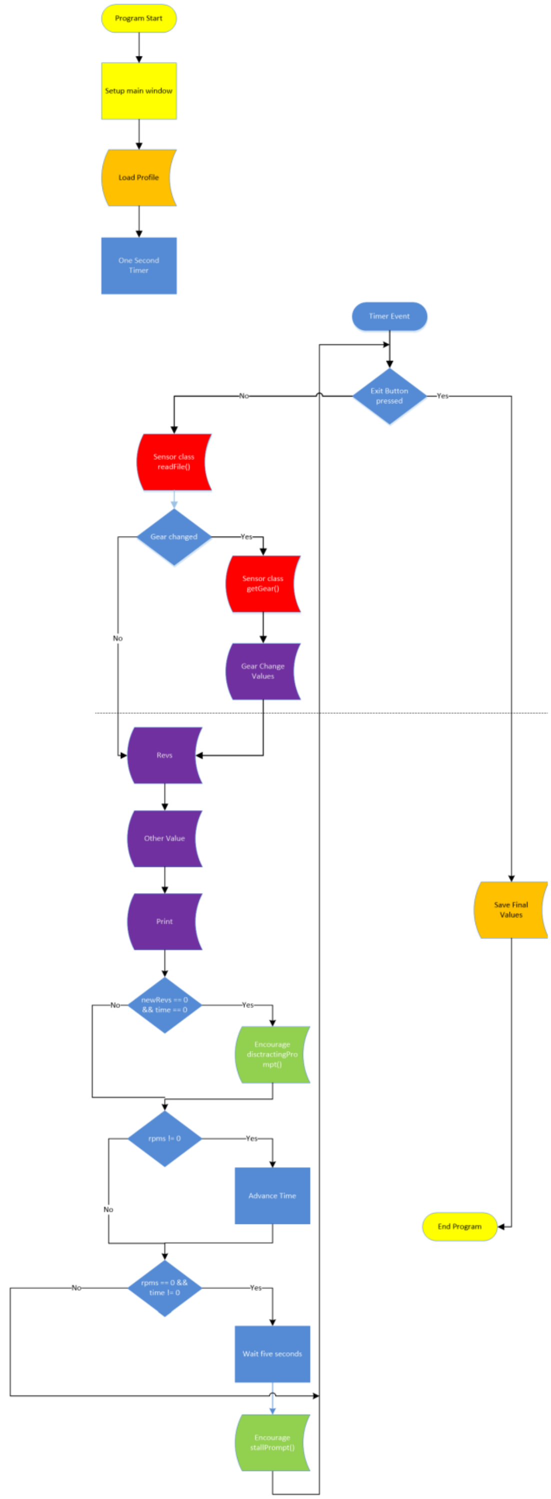


Figure C-2: LapcycleBeta1.Java flowchart with subclass [5].

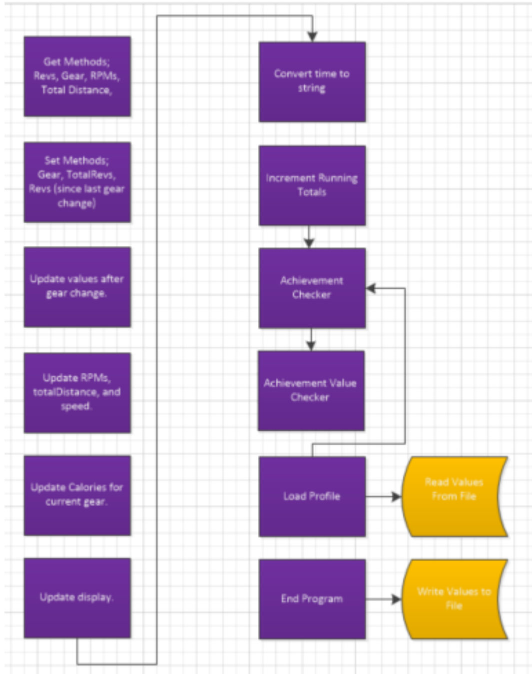


Figure C-3: CalculationsBeta1.Java Flowchart [5].

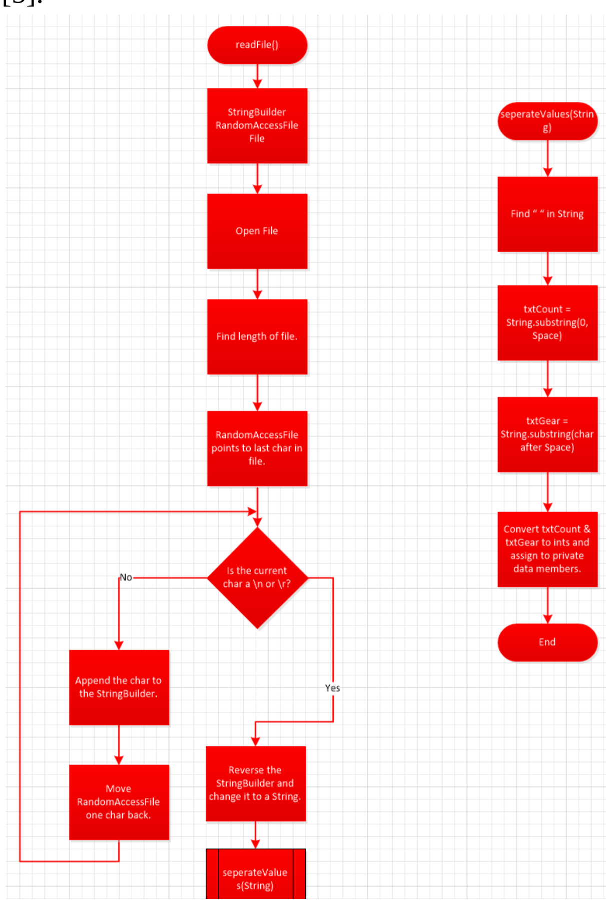


Figure C-4: SensorsBeta1.Java Flowchart [5].

III. *Development History*:

When designing the code for the Features 3,4 & 5, it was decided to use a modular approach with Java to produce the best possible product with the time and resources available. The main class is called LapcycleBeta1.Java. It is in its beta phase and is comprised of one main class and one imbedded class used to control the one second timer and call the Calculation, Sensors, profile, and Encouragement classes. It is comprised of a total of 160 lines of code and comments. The main class also creates the display window that is used to present the exercise metrics. It keeps time by inheriting the Java TimerTask class and scheduling the object of the embedded class to run every one -thousand milliseconds. SensorsBeta1.Java is 173 lines of code and comments and has gone through two revisions since the earliest alpha stage (the lowest of any class.) Originally, it read the number of revolutions that have been made during the session and the current gear setting from the microcontroller. This was later changed to read in the current RPM value instead of the number of revolutions. The mathematical calculations to extrapolate the six exercise metrics were originally managed by the main class, however as the burden for this program increased, it was decided to create a separate class, now called CalculationsBeta1.Java, dedicated to handling the metric calculations and updating the display. The entirety of the Calculations class is in its also in its beta phase. It has a size that is 470 lines long and has thirty-seven private data member members. EncouragementBeta1.Java is 115 lines of code with two methods (plus some support methods and a constructor) used to control the pop-up prompts to remind the person to keep moving (See Figures & .) AchievementBeta1.Java is the newest class and was born from the changes made in the design idea contract to replace the internet lockout encouragement feature. It is only 97 lines of code and uses a single method to display all of the achievements earned by completing preprogrammed milestones in time, distance, and caloric exertion. ProfileBeta1.Java also was created to replace the internet lockout feature. ProileBeta1.Java is 383 lines long and since it contains the code for the windows allowing the user to create and log into their profiles, it uses several embedded classes (up to three levels of inheritance) to control both windows, their buttons, other fields, and the corresponding actions to take when different information is entered. All these classes make up the project Lapcycle laptop program and are the result of nearly one-hundred lines of code.

Managing these metrics is more involved than simply incrementing them every time the program goes through a cycle. Measuring the calories put into the machine is achieved by using the formula for work (work = force * distance.) This is not an accurate reading of the amount of calories burned by the user because it does not take into account the user's age, level of fitness, heart rate, and other values. It is therefore only a measurement of how much energy the user is transferring into the machine and not how much they releasing into the environment from their metabolism. The force depends on the current resistance setting and the distance is calculated by measuring the total amount of revolutions made during the previous second and adding it to the total distance traveled so far. The There are extra private data members to record the number of calories burned, revolutions performed, and the distance traveled at the current resistance setting. The revolutions at the current resistance are calculated solely for the purpose of calculating the distance traveled at the current resistance. These are added to the private data members holding the total calories, distance, and revolutions every cycle and are reinitialized every time the resistance is changed. Thus, the total calories is the sum of all the calories burned at the different gear settings during the session. All of the six metrics (time, revolutions, RPM, calories, distance, & speed) are calculate and (if necessary) converted to different units by various methods before being added to the display window that was created in the main class before the JFrame was passed to the calculations class.

The laptop Java program also allows users to store and load profiles for the purpose of encouraging fitness by providing the ability to track their workout through multiple sessions and earn achievements for reaching certain milestones in the exercise metrics. The achievements are monitored by another class and is considered part of Feature 5: Encourage Fitness. The Profile class creates and controls a window to accept a username and password of a profile or to create a new profile (see Figures & .) It has sixteen private data members and a single public data member. Most of the private data members are objects from inherited JFrame subclasses used to make the opening window and the second window to create a new profile. The single public data member is boolean and prevents the main class from starting the main thread until a profile is successfully loaded. The profiles are saved in individual files. When a profile is created, a text file is created with the profile name in which all the running totals and achievements are stored. The Profile class is started by the main class, but the loading and storing of the running totals and achievements is handled by the Calculations class by the main class passing the Profile object to the Calculations class. The achievement class is called by the Calculations class when certain milestones are reached and by the profile class when they need to be stored or read. When a profile is read in at the start of the program, it checks if any of the achievements have already been earned since the program was last run and prevents them from being displayed again all at once when the program starts.

It is important to note that all of these separate classes exist solely to organize the code and make it simpler to read and manage. None of the classes are needed more than once at a time because there is always only one set of values. Hence, there is only one object made of each of these classes.

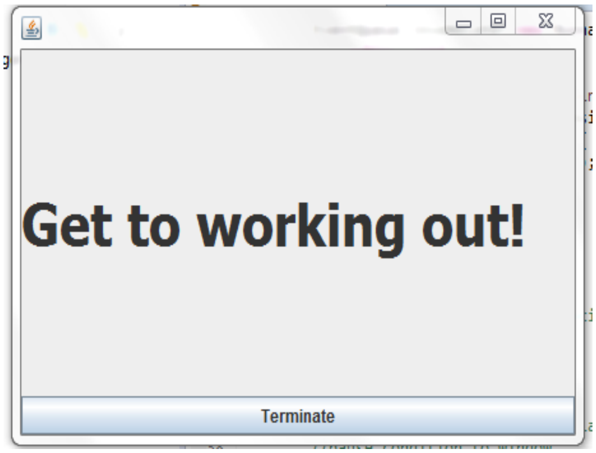


Figure C-5: Prompt displayed if the user has not started pedaling [5].

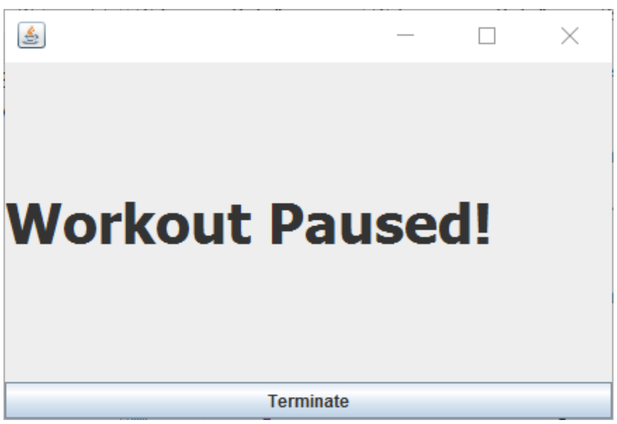


Figure C-6: Message displayed if the program reads zero movement for 6 seconds [5].

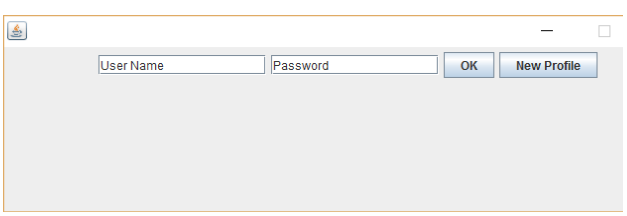


Figure C-7: Window from ProfileBeta1.Java allowing the user to login [5].

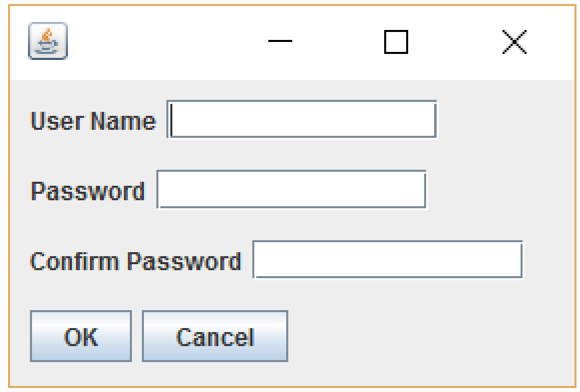


Figure C-8: Window displayed when the New Profile button is pressed [5].

IV. Development Divisions

The first task the team had to face when designing the higher level coding was which language to use. While there are many modern languages available today, the members of Team 10 only had significant experience in C++ and Java. C++ is a powerful language and is the primary language learned by all of the students who transfer to California State University, Sacramento from the Los Rios Community College District into the school of Engineering and Computer Science. C++ is an object oriented programming language, but it lacks the graphical libraries and error catching that was provided with the Java language. Fundamentally, anything that can be done in Java can be done in C++, because they are both based on C. However, Java already has built in classes that can handle the tasks required by this project and using C++ would require that the team spend quite a lot of resources developing their own classes that can fulfill the project feature set; a wise man once said, "you don't need to invent the wheel."

To keep time, initially the Thread.sleep method was used for 1,000 miliseconds and an integer was used to count the time in seconds before it was converted to minutes and hours in the Calculations class. Good programming practices are very important to Team 10 member J. R. Chadwick and while that would be sufficiently accurate for the purposes of this project which is rather resource light, it is not always reliable if the computer is being taxed for available short term scheduling because it tells time by counting processor cycles [29]. The Java class TimerTask uses actual milliseconds and is closer to being a true time keeper. While it also can be "overloaded" as well and not keep accurate time, since the graphical components are being handled by a separate thread, the demands on the imbedded class inheriting the TimerTask is very light in terms of resources and even an older midrange laptop has no trouble running it.

Another motivation for choosing Java was the throw/try/catch libraries that allow the code to handle errors that would cause a C++ program to crash (see Figure C-9) These exceptions can be used as conditions to provide responses to bad user input or to prevent the program from read/writing to memory improperly. If the user attempts to open a profile that has not been created, rather than crash with an error message, the user is instead given a window describing the problem and kicks them back to the previous prompt. Java also automatically multithreads when certain classes are used which allows the interface, timer, and calculations to run without interfering with each other. Finally, with Java's built in GUI swing library, creating a user interface with images in buttons is possible and greatly increases the available options for creating an interface. Java does have its drawbacks being that it is so complex; its nested classes can be less than intuitive when learning new features [29]. But attempting to accomplish these tasks in C++ or learning a new language on the fly would have made these tasks very difficult and without the Java Runtime Environment, the program would have only worked in sixty four bit Windows Operating Systems (a C version might also work in thirty two bit versions, but there would have been a chance of integer overflow.

Creating the interface required making GUI design and that limited the options available. It was considered creating a sort of pseudo GUI by way of using colored ASCII art if it was necessary to code the program in C++ instead of Java, however that would still require the use of a command line for the user to enter data when necessary. The lack of multiple threading windows would also have made this design cumbersome so it was decided to stick with the original plan of coding in Java. Finally, with the addition of a profile system, the lack of error catching in the C File library meant that there would be no practical method to handle the user attempting to open a profile that does not exist without the program crashing. The competed GUI is easy to read and provides the data organized in a manner that highlights the most important measurements by putting time in the upper left and calories and speed in the middle (see Figure C-11).

```
catch (IOException e) {
    System.out.print("Error: Unable to create file.");
    e.printStackTrace();
} //End catch
```

Figure C-9: An error generating code segment if the file create method fails.

```
/*
 * EMBEDDED CLASS
 * Purpose: Record the passage of time and update printed values once per second.
 * Version: 0.1.
 * Created:
 */
//static boolean timerReady = false;
static class OneSecTimer extends TimerTask {
    private Calculations2 calc1;

    /*
     *Purpose: Constructor to receive Calculations2 object.
     *
     *returns:
     */
    public OneSecTimer() {
        Dimension mainWindowSize = new Dimension(1000, 500);
        window.setMinimumSize(mainWindowSize);
        window.setMaximumSize(mainWindowSize);
        window.setVisible(true);
        window.setBounds(600, 300, 700, 400);
        window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        window.setResizable(false);
        JButton exitButton = new JButton("Exit Program"); //Creating Terminate Button
        window.getContentPane().add(exitButton, BorderLayout.SOUTH);
```

Figure C-10: Embedded class that controls the operation of the main code of the class.

Figure C-11: Beta version of main display window.

V. Performed Tests

When a profile has been loaded, the TimerTask will begin. It is scheduled to run every 1,000 milliseconds. Each time the scheduled task runs, it will read the output from the Feature 2: Measure User Performance microcontroller component. The program reads in two values that the microcontroller records; the current resistance setting and the current revolutions per minute. These values are read together as a single string. The first thing that needed to be tested was the Java program's ability to interpret the microcontroller data. If it failed to do so, the program would throw an exception because the string length was too short. It was discovered that the Java code needed to be modified to handle the case in which the program had been started, but the user had yet to begin moving resulting in the microcontroller sending an empty string. To account for this, the program checks for a blank input and defaults to zero RPM and resistance level one. The program also tended to crash approximately 10% of the times it attempted to read from the microcontroller output file during a cycle through the main thread. This turned out to be due to the Arduino's much slower processing speed compared to the laptops that were running the main program code; this caused the laptop to read from the output file when the Arduino had not written the entire line, causing the program to read in a string that was too short. Increasing the Arduino's processing rate to its maximum of 115,200Hz reduced the occurrence of this sort of read failure to less than 1% and this was deemed acceptable to the team.

VI. Measuring user performance is the definition of Feature 2 and refers to the sensors used to keep track of the force and movement of the user. However, the data is processed by the code for Feature 3. Testing these outputs mainly consisted in confirming that the displayed values accurately represented the predicted mathematical calculations. Time is a value that is incremented each time the main thread cycles, and the total number of seconds is converted into a standard hh:mm:ss output. Confirming its value was done by simply comparing the output to a clock with a second hand and running the program long enough for it to roll over into greater than zero hours. The calorimeter is not a true calorimeter in the sense that it does not take into account the size, age, gender, health, and other metrics into its calculation and is simply a measure of work put into a machine calculated by multiplying the number of revolutions made by the force required to move the pedals at the current resistance. Testing the accuracy of the RPM meter with the original code, as described in the previous sections, resulted in inaccurate RPM and Speed measurements. When only reading in the number of revolutions made since the last second with two passes per revolution, the increase in revolutions was always in multiples of .5 causing the RPM to always be in a multiple of thirty. Once the code was changed to read in the RPM and calculate the metrics based on that, a timing test with a metronome confirmed that the RPM reading was within a 90% accuracy value. This can be further improved by improving the processing ability of the interface device allowing for more than one reading per second.

Appendix D: Mechanical Development Documentation

I. INTRODUCTION

Projects Lapcycle's mechanical aspects includes a modified single person spin bike and it's magnetic resistance wheel. Since the Lapcycle promises the integration of cardiovascular exercise and casual laptop use, the group had to modify a spin bike to accommodate a laptop safely and comfortably. The original dash that the bike had was removed and in its place a laptop stand, with three points of articulation, was added so the person can pedal easily while using the laptop. The resistance mechanism the spin bike had previously was also modified to allow for easy electronic shifting. This resistance mechanism consists of a spinning metal wheel with a movable series of magnets that alters the amount of force needed to get the wheel spinning.

II. Component Level Discussion

A. Single Person Spin Bike

The main component of Feature 1 is the Single person spin bike. The spin bike is the Marcy Me 709 Recumbent Exercise Bike that has been modified in the following ways;

- Original dash and electronic components removed to make room for the Lapcycle's microcontroller.
- Reed switch attached to measure RPM
- Addition of laptop stand that allows for comfortable use of laptop during exercise. This is shown disassembled in Figure D-1 and attached to the Lapcycle in Figure D-2
- Original resistance adjustment mechanism removed and replaced for an electronic one.

Most of the assembly of Feature 1 was handled by Chadwick and his father. They assembled the laptop stand and attached it to the spin bike. They also helped greatly in the modification of the gear system. The Finished version of the Lapcycle can be seen in Figure D-2.

Figure D-1: Disassembled laptop stand

Figure D-2: Fully assembled Lapcycle with laptop stand and modified recumbent spin bike.

B. Resistance Mechanism

The resistance mechanism consists of primarily two components. A servo/microcontroller system and a spinning metal wheel with a movable series of magnets that alters the amount of force needed to get the wheel spinning. The servo/microcontroller combo serves to pull a cable that pulls the magnetic plate away from a spinning wheel. This servo system is described in more detail in appendix B section B. This Section will speak more about the resistance plates of the bicycle.

In order to change the force needed to pedal the recumbent spin bike, a series of magnets attached to an arc shaped plate must be either pulled away from or pushed closer to a metal wheel. The magnetic force provided by the magnetic arc causes the metal wheel to become more difficult to spin. Figure D-3 shows how the arc and the metal wheel are aligned. Figure D-3 also shows the magnetic arc at the highest resistance setting.

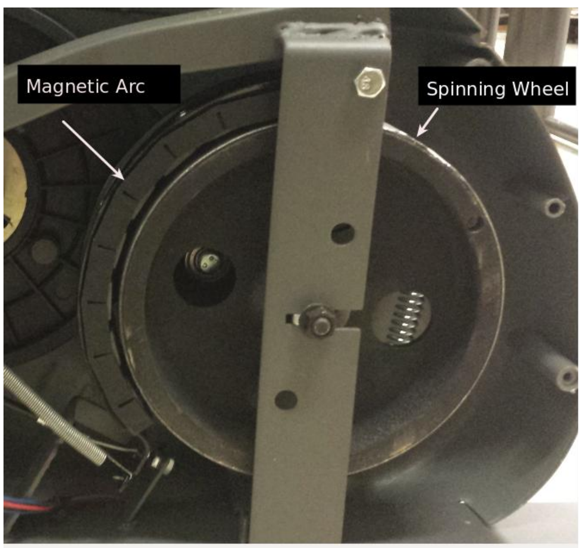


Figure D-3: Magnetic Arc and Spinning Metal wheel

As previously stated, change in resistance is achieved by pulling the magnetic arc away from the metallic wheel via cable attached to the arc. The maximum distance the arc can be pulled from the wheel is 7/8 of an inch and the minimum distance the arc can be is 1/8 of an inch. These distances from the metal wheel result in a pedal resistance of 7.78 Newtons and 16.68 Newtons respectively.

1) Problems and Possible Improvements

Some of the problems that were encountered with the magnetic resistance was that the magnetic arc was very difficult to pull away from the metal wheel while at the highest resistance. This was because the arc was touching the metal wheel and caused a very strong pull between the arc and the wheel. This caused a problem with the servo being able to pull the arc away. The problem was that the servo provides an even pulling force and the force required by the magnetic arc is not linear. The way to solve this issue is to change the way the resistance is created as. One idea to change that is to replace the magnets with rubber pressure plates and the pulling of the arc would result in less friction on the wheel. This idea was not implemented due to the limitations of the group as we have no means of creating this new resistance mechanism.

III. Performed Tests

The tests performed were to verify that the Lapcycle met the following criteria:

- The person can use the Laptop comfortably while exercising
- People of different sizes, weights and disabilities can use the Lapcycle.
- The project can withstand rapid pedaling.
- The gear system can successfully change the resistance of the bike

A. Spin Bike Comfort Test

The comfort test was created in order to see if a person can ride the bike and use the laptop with no wires getting in the way or any other issue. It was a very simple test in which the user was to use the spin bike normally and see if they experienced any discomfort or obstruction. The main concerns of this was that the Lapcycle was used with nothing camimg loose and the act of pedaling did not cause the user hit anything with their knees.

1) *Test Results*

The results of this test were pretty straightforward.

- Use of the bike did not result in any loss in functionality. RPM counter continued to work and so did the gear shift mechanism.
- The rider experienced no discomfort. This aspect of the test was more subjective and depended greatly on the user.

B. Person Diversity Test

This test was created because the Lapcycle needs to be able to be used by many different types of people. We used four different types of people and redid the rider comfort test.

- Tall person: 6ft, 180 lbs. Rider experienced no discomfort or loss functionality of the Lapcycle.
- Short person: 5ft 6i in, 165. Rider experienced no discomfort or loss functionality of the Lapcycle.
- Overweight person: 5ft 10in, 220 lbs. Rider experienced no discomfort or loss functionality of the Lapcycle.
- Person with slight disability. Person has slight muscular dystrophy. Rider experienced no discomfort or loss functionality of the Lapcycle.

C. Gear Pulling Test

This test was designed to confirm that the servo system could successfully pull the magnetic arc the full 7/8 of an inch away from the metal wheel and reduce the pedaling from 16.68 newtons to 7.78 newtons.

1) *Test results and Issues Encountered*

The servo system proved capable of pulling the magnetic arc the full 7/8 of an inch and also put it back to it's original position.

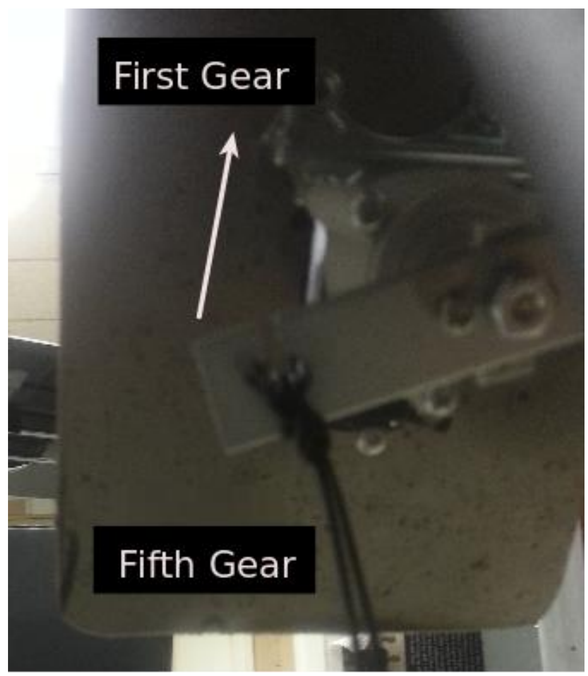


Figure D-4: Servo arm with corresponding gear positions

Figure D-4 shows the servo arm that pulls the magnetic arc away from the metal wheel. In order to pull the magnetic arc the full 7/8 of an inch the arm needs to travel an approximate 30 degrees. The 30 degree range of motion was split into 5 equal partitions and represents the five different resistance gears that Lapcycle has.

2) *Issues*

Two major issues were faced when first dealing with the resistance adjustment. The two issues were overcoming the strong pull of the magnets at a close distance and keeping the gears 2, 3 and 4 at repeatable positions. At first the positions of gears 2, 3 and 4 were not consistent and varied when the servo pulled. The servo was set to pull with a constant force even though the plate's attraction to the metal wheel was not linear this caused the gear levels to fluctuate distance depending on many factors. This was fixed with the addition of varying pull times depending on what gear is to be attained. An example of this is when going from gear 5 to 4 the servo pulls for a longer time because the magnetic arc is more difficult to move in that position.

APPENDIX E: VENDOR CONTACTS AND ACKNOWLEDGEMENTS

Dear Tod Chadwick:

Thank you for helping your son Joshua, and helping the rest of the team of Project Lapcycle in their endeavor to combat obesity. Your constructive and mechanical input into this bicycle had helped advance the Project to where it needed to go, and without it, the "cycle" of Lapcycle would not come to fruition.

We also wanted to apologize for one of the long nights we unfortunately subjected to you, and know that your efforts were not in vain. We realized that it was not fair to ask you to fix vital parts of the bike at an inopportune moment, and for that, we apologize.

Again, thank you for your contributions to the Project, and best of luck to your future endeavors.

Cheers and Regards,

J.R. Chadwick, Andrew Beltran, Lawrence Ly and Armando Fuentes

The Project Lapcycle Team

Dear Cody Jackson:

We wanted to thank you for your support with regards to Project Lapcycle. Your help in explaining how to kick start the software provided a massive boon in making the software that made the Project possible, and your constant guidance and advice with regards to software development helped establish a sense of good practice with us.

We hope to see you again, and wish you luck in your future endeavors. And from some of us, thank you for being our side tutor during the thick of our Computer Science classes.

Cheers and Regards,

J.R. Chadwick, Andrew Beltran, Lawrence Ly and Armando Fuentes

The Project Lapcycle Team

Appendix F: Personnel & Resumes

B. *Joshua Ryan "J. R." Chadwick*: (CpE)

1) *Introduction*-

J. R. Chadwick is the oldest member of the team and has the most diverse areas of experience. He was the person who came up the idea for the project and recruited the other team members. Chadwick has previously earned three Associate of Science Degrees and a Certificate. He has a variety of interests including video game technology, cooking, fitness, making YouTube videos, and science. After graduating in May of 2016, Chadwick plans to teach computer programming to teenagers at the University of California, Las Angels for the summer prior to teaching English for a year in southern China. When he returns, he will either find a job working in information technology support & network administration and product design & testing. He also may continue teaching at the high school level in the fields of computer science, biology, and/or mathematics.

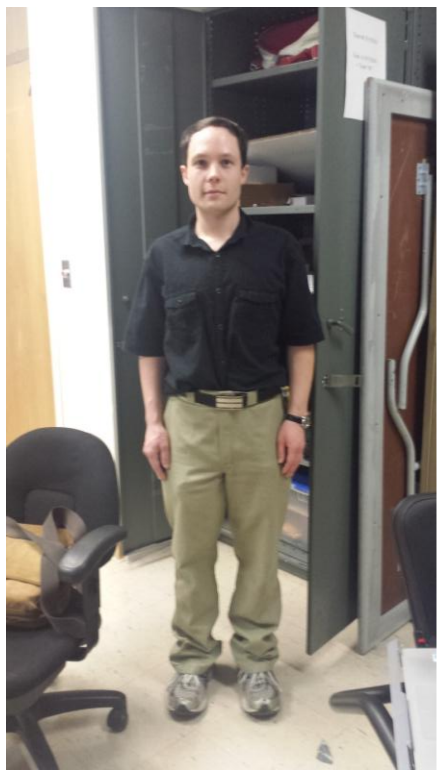


Figure F-1: J. R. Chadwick in the CpE / EEE-Electronics Senior Design Lab 25 April 2016 [3].

2) *Project Responsibilities*-

As it was Chadwick's idea to create a spin bike capable of being operated while using a personal laptop, he served as the initial team leader and helped organize the project design. Chadwick served two primary functions throughout the project; assembling the spin bike base for Feature 1 and writing the code for Feature 3& 4 as well as incorporating the code written for Feature 5. The efforts involved prove to be a valuable learning experience which greatly increased his understanding of the Java programming language which will be very useful in his upcoming teaching job. J. R. Chadwick's father, Tod, is an avid mechanic with a large selections of tools and supplies. To successfully assemble the prototype required more equipment and experience than the team had; Mr. Chadwick was able to assist J. R. in modifying the frame in any way that required powered tools such as attaching the laptop mount and creating the servo panel.

3) *Résumé*-

Objective: A position in computer technology support.

Education/Certification:

B.S. Computer Engineering; California State University, Sacramento • **GPA** • 2016

AS-T Mathematics; Sacramento City College • **Honors, GPA** 2012

Certificate of Programming; Sacramento City College • **Honors, GPA •** 2012

AS Computer Science; Sacramento City College • **Honors,•** 2012

AS Science; Sacramento City College • **Honors, GPA•** 2011

TEFL 60 Hour • Grade: A - 2016

Comptia A+ Certification - 2011

Skills & Knowledge:

Programming Languages: C, C++, Java, Assembly, SQL, Verilog, VHDL

Technical Skills: Software/hardware diagnostics, hardware/software upgrades, OS installation, BIOS updating, system optimization, soldering, surface mounting, DMM usage and care, virus/spyware removal, SOHO setup, AC/DC circuitry

Software: MS Office (Word, PowerPoint, Excel, Outlook, Access,) Microsoft Visual Studios, Microsoft Viso, Open Office, CoreFTP, Putty, Sony Vegas 9.0 Pro

Operating Systems: DOS, Windows 3.x/9x/2000/XP/Vista/7/8/10, Unix

Work Experience:

Java Teacher **ID Tech** **6/2016-8/2016**

Teaching Java to kids ages 12-18 at UCLA and manage activities while maintaining a safe environment.

EEE/CSc/CpE Tutor **ECS, CSUS** **2/2015-Present**

Tutoring students in Electronics / Computer Engineering & Computer Science.

Technician **Gamer World** **1/2015-4/2015**

Repair, buy, and sell computers and video games and related products.

Cofounder & Tech Support ***CADRE Docs*** **5/2013-9/2015**

Provide support and information on requirements for the development of financial instrument software project. Once the software is fully operational, will be personally responsible for customer support.

Math/Physics Tutor ***Los Rios Community College District, SCC*** **4/2008-5/2013**

Tutoring college students in beginning math, calculus, physics, and beginning chemistry in a drop-in tutoring lab at Sacramento City College.

IT Consult ***Altod Corporation*** **1/2007-5/2013**

Provide PC upgrades and maintenance.

PC Technician ***Computers 4 Kids*** **8/2007-12/2007**

Repairing/refurbishing tower computers and instructing elementary school children and their parents on the basic operation of the equipment.

Vender ***Circulation Marketing*** **9/2004-11/2004**

Sold subscriptions for the San José Mercury News Paper.

Awards & Activities:

Eagle Scout • Boy Scouts of America
World Conservation Award • Boy Scouts of America
Assistant Scoutmaster, Boy Scouts of America

C. Andrew Gozum Beltran: (CpE)

1) *Introduction*-

Originally, Andrew Beltran intended to take CpE 190 in the fall of 2016, but he was convinced by J. R. Chadwick to start it a year early so they could work together on a project and allowing Beltran to graduate one semester earlier. Beltran has an interest in Air Force aviation technology, due to his time invested in flight simulators.

Figure F-2: Andrew Beltran in the CpE / EEE-Electronics Senior Design Lab 25 April 2016 [3].

2) *Project Responsibilities*-

Primarily Beltran was responsible for managing the microcontroller used for Feature 2. Said responsibilities handled the assembly of the components needed for the controller to work, along with the discovery of the hardware-to-software bridge.

3) *Résumé-*

Objective: To attain a position leading to a career in the Computer Engineering field.

Education:
In Progress – **B.S. Degree, Computer Engineering,** CSU Sacramento – December 2016

Related Courses:

Introduction to Microprocessors	Advanced Logic Design	Advanced Comp. Organization
Data Structures/Algorithm Analysis	Computer Hardware Design	Physical Electronics *
Embedded Processor Sys. Design *	Signals and Systems	Operating Systems Principles
CMOS and VLSI Design *	Physics – General Mechanics	Physics– Electricity and Magnetism

** In Progress as of Spring 2016*

Knowledge and Skills:

Programming Languages:
Verilog • Java • C

Software:
Microsoft Office • NI MultiSim • Cadence (SPICE)

Operating Systems:
Windows XP • Windows 7 • Linux

Work Experience:

- **Inter-Library Loan Assistant** **CSU Sacramento Library** **1/15 – Present**
  Performing logistics duties regarding books and library resources. Said duties include locating, packaging for shipping, and scanning resources for electronic delivery.

Project Experience:

- **Micro-surveillance Detector and Disruptor** – Spring 2014
  Member of a four person team which developed an EMF detector circuit to find "spy" devices and to disable them with an EMP transmitter. Assigned role was to develop the arming sequence to identify EMF signals and fire the EMP on available demand on an Arduino Uno platform.

Accolades and Affiliations:

- CSUS Engineering Dean's Honor Roll: Springs of 2012 to 2014, Fall 2015
- Association for Computing Machinery: Member, Spring 2013 to present

D. *Lawrence H. Ly*: (CpE)

1) *Introduction-*

Similarly to Andrew Beltran, Lawrence Ly had originally planned to take CpE 190 in the following year, but was convinced by Chadwick to start it earlier which will allow him set his plans for graduation within the end of 2016. Ly has a wide variety of interests; he enjoys the simple things such as playing video games and building plastic model kits of robots. Ly uses his interests as inspiration to go into the computer engineering field. Skills that Ly provided to the project was his experience in programming Java code. As a student who will be continuing his education at the university he will use the project experience from this project to help him with life challenges to come.

Figure F-3:Lawrence Ly in the CpE / EEE-Electronics Senior Design Lab 25 April 2016 [3].

2) *Project Responsibilitie-s*

Ly's responsibility was creating the Java code for Feature 5 as well as Feature 4 GUI, and to assist Chadwick with integrating it into the main code. During phase one of the project the idea of Feature 5 was to create a User Encouragement system. The idea behind this was to motivate the user to want to use the machine while working on their laptop, this included distracting prompts, password protection, and Wifi Lockout. In Phase 1 encouraging prompts were completed and displayed. After the end of phase 1 the team went back to the drawing boards as the ideas in feature 5 weren't encouraging enough from the feedbacks. The team decided to rethink what it is the be encouraged and motivated to want to do something. What came out was the revised feature 5 which is now, encouraging prompts, user profile data, and an achievement system. The majority of phase 2 was to working on the achievement system and working with Chadwick to make the entire software portion complete.

3) *Résumé-*

OBJECTIVE: To expand my knowledge and to gain technical experience in a professional workplace

EDUCATION:

In process: **Bachelor of Science in Computer Engineering**
Expected date of graduation: May 2017
California State University, Sacramento

Related Courses:

Computer Hardware System Design	Signals & Systems
Intro Microeconomic Analysis	Network Analysis
Probability + Random Signal	Discrete Structures
Electronics I	Intro Computer Architecture
Computer Interfacing	Data Structures and Algorithm Analysis
Advanced Logic Design	Computer Networks and Internets

PROJECT EXPERIENCE:

Construction Challenge ***Destination Imagination*** May 2009

- Worked in a team of six to design and build working equipment that moves and transports materials to designated sites in order route large amounts of water
- Contributed to the design of the robot, as well as aiding in controlling and navigation of the robot to designated sites
- Project was chosen as one of the top three machines out of 50 total projects

Computer Interfacing Project ***CSUS EEE Department*** Dec 2014

- Worked in a team of four to create a working piano system that plays sounds and lights up
- Contributed to the project by coding the algorithms through Python that executes the commands
- Used materials included Raspberry Pi, and Ardiuno Uno Microcontroller

Senior Design Project(in Progress) ***CSUS EEE Department*** Sept 2015 - May 2016

- Working in a team of four to design a working stationary spin bike that would connect to a users laptop to display exercise data while the user pedals on the bike
- Contributing to the project by coding the display prompts through Java Swing class as well as creating an achievement system for the users to earn
- Hardware materials we are using are a Single Person Spin Bike base, Ardiuno Uno Microcontroller, Servo motors, Hall Effect Sensors, and any personal Laptop Computer

KNOWLEDGE AND SKILLS:

Programming Languages: C • Java • Python • Verilog • VHDL • x86 Assembly
Operating Systems: Windows XP • Windows 7 • Windows 8 • Mac OS • CentOS • Unix
Software: jGRASP • Eclipse • MS Office • MATLAB • Xilinx ISE • MultiSim
Tools: Oscilloscope • Function Generator • Logic Analyzer

WORK EXPERIENCE:

Volunteer Work ***Donut Time*** May 2014 – Present

Kings Concession Stand April 2016

ACCOMPLISHMENTS AND ACTIVITIES:

- Dean's Honor List Spring 2012 and Fall 2012
- Founder and President of the Plastic Model Club at Sacramento State University Spring 2015

E. Armando Fuentes (EEE-Electronics)

1) *Introduction-*

Unlike the previous two members, Fuentes did not join the group until the beginning of the senior design course. After the previous three members had formed the initial team the summer prior to beginning senior design, Fuentes lacking a group was invited to join to fill in as the team's EEE member. This did not pose to much of an issue as the group was very accepting and he had Electrical Engineering skills for which the group was looking. In his spare time, Armando can be found volunteering or participating in Greek organization related activities. Armando is National Board Representative for his fraternal organization Sigma Delta Alpha and Vice President for the Greek council of which ΣΔΑ is a part. He also enjoys lifting weights and experimenting with small electronics such as the Raspberry Pi and the Arduino. After graduation, Fuentes in physical design engineering at a corporation such as Intel.

Figure :F-4 Armando Fuentes in the CpE / EEE-Electronics Senior Design Lab 25 April 2016 [3].

2) *Project Responsibilities*-

As the team's only Electrical Engineering major, Fuentes's primary tasks were centered on testing the sensors and writing the code for the microcontroller used for Phase 1 and then adapted to Phase 2. This area dealt more with the hardware more than anything else. This included the gear system and reed sensor used to control and monitor the user interaction. With the gear system being very mechanically oriented, Fuentes relied on the help of Chadwick's help for the implementation of it. Fuentes also came up with some creative solutions for some of the groups problems, some of which made it in to the final design of the project, such as modifying the Arduino code to calculate the RPM instead of letting the Java code do it, greatly increasing the accuracy of the displayed exercise metrics.

3) *Résumé*-

Objective

A job in the Engineering Field

Education

in progress: **Bachelor of Science, Electronic Engineering,** CSU-Sacramento (Sac State), GPA 3.0 May 2016

Associate of Art with emphasis in Physics, Modesto Junior college, GPA 3.0, June 2012

Associate of Art with emphasis in Mathematics, Modesto Junior college, GPA 3.0, June 2012

Related Courses:

Advanced Logic Design	Circuit Analysis	Physics
Applied Electromagnetism	Programming in Matlab	Programming in Java
Network Analysis	Assembly programming for x86 processors	Public Speaking
Signals and Systems	Electromechanical conversions	

Skills:

Computer:

MySQL · Python · Multisim · Makefiles · GIT · Linux OS · Windows OS · MS Office · MS Outlook · Raspbien
Raspberry Pi · Java · Assembly · Matlab · Basic · Altera Max II · ADS · PLDs

Communication/Organization:

- Versed in Scrum Methodologies (Was even interim Scrum Master when needed)
- Excellent problem solving and analytical skills
- Strong technical writing and presentation skills
- Bilingual: English/Spanish

Projects:

System QA of New Automation Software.

Owned, maintained and transferred the System QA of automation software HEXA3 under Maestro. Created and executed test plans for the functional testing of the new automation software HEXA3. I also provided the ramp up for the System QA team who was to take over my responsibilities. Provided multiple tutorials and headed multiple meetings in order to facilitate ramp and debug known issues.

Video Game Design

Organized and promoted the development of video game projects for high school students. Assigned deadlines for the students to aid their progress and also provided assistance in troubleshooting along the way. Students worked with Dark-Basic to create 3-D video games.

Work Experience:

System Validation Engineering Intern **Intel** 1/14-1/15

I performed System QA for Maestro under two different automation programs, HEXA3 and TeamCity. Created three build configurations for the continuous integration of Maestro. Optimized existing build configuration by adding makefiles for building tests and wrote short cmd commands to dynamically grab latest version of Maestro (version name had to be hard coded in before this). Also debugged issues with data bases (using MySQL) before the above priorities became prioritized.

ATS Student Lab Assistant	**CSUS**	Summer 2013
Math Tutor	**Tutoring Club**	8/11-7/12
Physics Instructor	**Trio Upward Bound**	1/12-5/12
Math/Science Tutor	**Trio Upward Bound**	8/09-5/12
Computer Instructor	**Trio Upward Bound**	Summer 2011

Taught a computer programming to underprivileged high school students to increase interest in computers and science. Created daily lesson plans, weekly tests/quizzes, and provided an organized learning environment.

Accomplishments and Activities:

Member IEEE (Institute of Electrical and Electronic Engineers)

Dean's Honor Roll

Member SHPE (Society for Hispanic Professional Engineers)