Spring 2006

Development of a Race Simulation to Be Used with Treadmills and Exercise Bikes

J. Rodney Dolen
Regis University

Recommended Citation

Dolen, J. Rodney, "Development of a Race Simulation to Be Used with Treadmills and Exercise Bikes" (2006). *Regis University Student Publications (comprehensive collection)*. 144.
https://epublications.regis.edu/theses/144

REGIS UNIVERSITY

SCHOOL FOR PROFESSIONAL STUDIES

MASTER OF SCIENCE
IN
COMPUTER INFORMATION TECHNOLOGY

Development of a Race Simulation to be used with Treadmills and Exercise Bikes

PROFESSIONAL PROJECT PROPOSAL

J. Rodney Dolen

April 15, 2006

Executive Summary

ICON Health & Fitness, Inc. has become one of the largest manufacturers of fitness equipment. This company now includes many popular fitness equipment manufacturers like Weider, Weslo, HealthRider, and NordicTrack through various acquisitions and launches of new lines. In 2005, it is projected four million units of fitness equipment will be manufactured by the company (ICON).

With the popularity of treadmills and exercise bikes, ICON Health & Fitness added the capability to download exercise programs using its iFit® technology. This allows an audio cable to be plugged into the equipment from a PC and utilizes sound waves to control a treadmill's speed and inclination or an exercise bike's resistance.

By mapping a race course with a GPS device, the GPS device could download its readings in a text format to a PC to have the course's elevation and length emulated. With this simulation, a Java program could process it and allow a program to be ran on a Java capable device to control a treadmill or exercise bike utilizing the iFit® technology.

This capability would cater to the runners who cannot run a race due to family or work schedules. They could emulate an upcoming race on a treadmill and workout when their schedule permits. In 2005, the Springfield Road Runners of Springfield Illinois sanctioned 18 races for women and 17 races for men to be used for their points' series (Springfield). A member of the Springfield

Road Runners can now train for the upcoming race without the worry of running at night or during inconvenient times. With this one example from a running club in a community in Illinois having 18 possible races to download, we see the market possibility when the number of running clubs in the United States is factored into the equation.

This capability would also cater to the runner who wishes to run the Boston Marathon, ride in the Tour De France, or other popular races. Instead of qualifying and then joining the 20,000 runners in April for the Boston Marathon, the runner could load up the Java program and see how he would compare against other runners within the race (“Boston”).

Once a GPS race is captured, it can be processed to emulate the race and store the simulation in an encrypted format. It will then be available to be purchased and downloaded from a website. A Java program will be available to purchase which will process the downloaded encrypted file and emulate the race on a treadmill or exercise bike.

Table of Contents

Table of Figures

Chapter One: **Introduction**

1.1 Statement of Needs

People have a basic need for exercise. Exercise improves overall health such as reducing the risk of diseases like colon cancer and diabetes, reducing high blood pressure, and reduction in pain from arthritis (Center). In 2000, the Behavioral Risk Factor Surveillance System reported over 26% of adults did not report any physical activity during their leisure time (Center). This trend away from exercise in our society will lead to poorer health.

In today’s society, it is easily understandable why exercise is being overlooked. Over 23 years starting in 1973, the United States’ worker’s average hours increased by 19% (“Working”). This coupled with the demands of family has greatly reduced the ability of people being able to exercise. This reduction in the ability to exercise causes more calories to be consumed without being used by our bodies through physical activities such as running or biking.

Many communities across America have running and bicycling clubs promoting either running and/or biking. For instance, the Springfield Road Runners Club in Springfield Illinois boasts of 660 members (Road Runners. “Springfield”). It also sanctioned 18 races for women and 17 races for men during 2005 (Springfield). These clubs’ passion to promote exercise to their communities through various races can be greatly hampered by the inability of people to get free time to train for a race.

An option to promote health and races during the free time of individuals would be the utilization of treadmills and exercise bikes. In the convenience of people's homes or at their health clubs, they could train whenever they have available time. By simulating races on a treadmill or an exercise bike, an individual could set a speed and have the same simulated hills and distance of the race they are hoping to run. When the day of the race arrives and if they choose to participate, they would be prepared to run it without having to find available time during the daytime hours for training taking away time from their family or work.

1.2 Review of Existing Technologies

To create a race simulation, the course must be mapped out. This mapping would have to be done between the start and the finish points of the course. Using a GPS, or global positioning system, device, points along the course could be measured to determine the altitude and length between each successive point. Through a series of readings and then downloading them to a text file on a PC, the race could be mapped out utilizing the points along the way to determine the incline and length of the incline for the treadmills and resistance of exercise bikes.

ICON Health & Fitness produces much of the popular treadmills and exercise bikes available today. These include the brand names of Weider, Weslo, HealthRider, and NordicTrack. ICON has created the ability of their treadmills and exercise bikes to receive sound waves from a PC to control inclination and speed on a treadmill and resistance on an exercise bike. This ability utilizes their iFit® technology which uses the sound from a PC. By using the same technology, a program could produce sound waves at calculated intervals to run on devices with the ability to output sound to control either an exercise bike or treadmill to simulate a selected race.

To encompass a large array of devices, the Java programming language from Sun Microsystems was selected for this project. Its platform independence allows for more devices than the standard Microsoft platform found on most PC's. This will allow the user a broad range of devices to run a race simulation especially if they need the device to be more mobile than a PC would offer.

Java will also be used to allow the website to create the interfacing tools to allow secure commerce, the uploading of GPS information of races, the downloading of race simulations, and communications to the internal staff. Using Java on the website with PayPal® tools, the website will allow for secure transactions with acceptance of major credit cards.

1.3 Business Opportunities

This system facilitates the training of races that do not fit into the schedule of a working individual. This may include the 175,000 members of the over 700 clubs that make up the Road Runners Club of America (Road Runners. "Road"). With various sanctioned races, the runner could choose to train on a treadmill for an upcoming race even if the runner is traveling for business, unable to run during the day, or unable to run due to the weather.

This would also cater to the runner who wants to test himself on a course. For a race like the Boston Marathon, 20,000 runners must qualify to run in the

April race (“Boston”). This simulation would allow the runner to see how his time would measure against runners who competed in the race. The Chicago Marathon accepts the first 40,000 runners who apply and only tracks their time up to six hours (“Chicago”). This project would give the runner a chance to test himself before applying for the race.

Like the runners, bicycling enthusiasts can test themselves against simulations for their respective races. An example is the prestigious Tour de France. With this race being the premier race for biking, a bicyclist could test his skill on the different stages on the tour. This gives the individual who would never compete in this race the ability to test himself against the professionals of the world.

This project will also cater to the novice runners who are just starting to get involved in running or biking. An individual may have a goal to lose weight and run in a local race. Many times this individual does not know where to start or may feel that they will embarrass themselves by taking an extended amount of time. The simulations will give them an opportunity to run or bike the course without having to actually go to the course.

Chapter Two: **Research**

The research has shown this project will be divided into multiple areas. The areas are currently divided into the race simulation setup, race program, and the website.

The race simulation setup program is diagramed below:

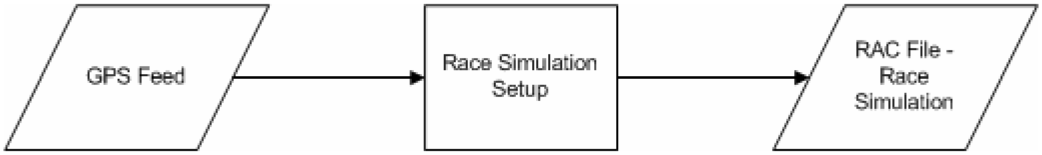


Figure 2: Diagram of Race Simulation Setup

The external GPS feed file will be used as input for the race simulation setup program. The program will then create the race simulation as a file with an RAC extension. The RAC file holds a description of the race, the factor for the number of feet in a mile, and the GPS measurements converted to feet to have a common unit of measurement. The RAC file is used by the race program for the simulation.

The race program is diagramed:

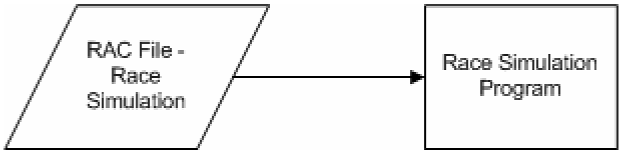


Figure 3: Diagram of Race Simulation Program

The race simulation, or RAC file, is used as input into the race simulation program. The program will then be able to control exercise equipment connected to the device running the program.

The final area to be diagramed is the web site:

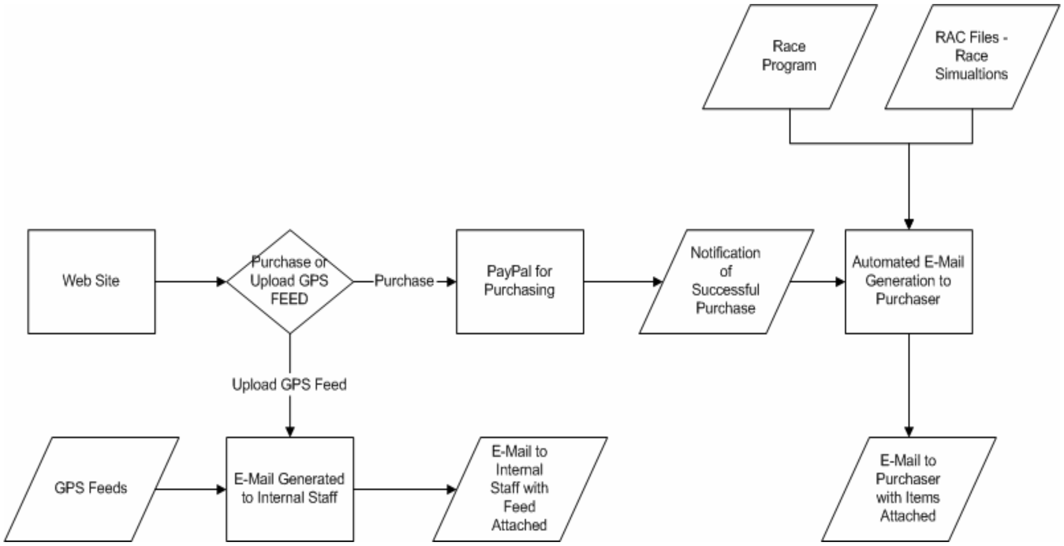

Race Program
RAC Files - Race Simualtions
Web Site
Purchase or Upload GPS FEED
Purchase
PayPal for Purchasing
Notification of Successful Purchase
Automated E-Mail Generation to Purchaser
Upload GPS Feed
GPS Feeds
E-Mail Generated to Internal Staff
E-Mail to Internal Staff with Feed Attached
E-Mail to Purchaser with Items Attached

Figure 4: Diagram of Web Site

The diagram shows how the web site is used for uploading GPS feeds for future simulations and selecting purchases which communicates to PayPal®'s website to process. PayPal® will send an automatic notification to an automated process on successful purchases. The automated process will then generate an e-mail to the purchaser with the purchased programs or files attached to the e-mail.

The following sections will step through the three different areas in more detail.

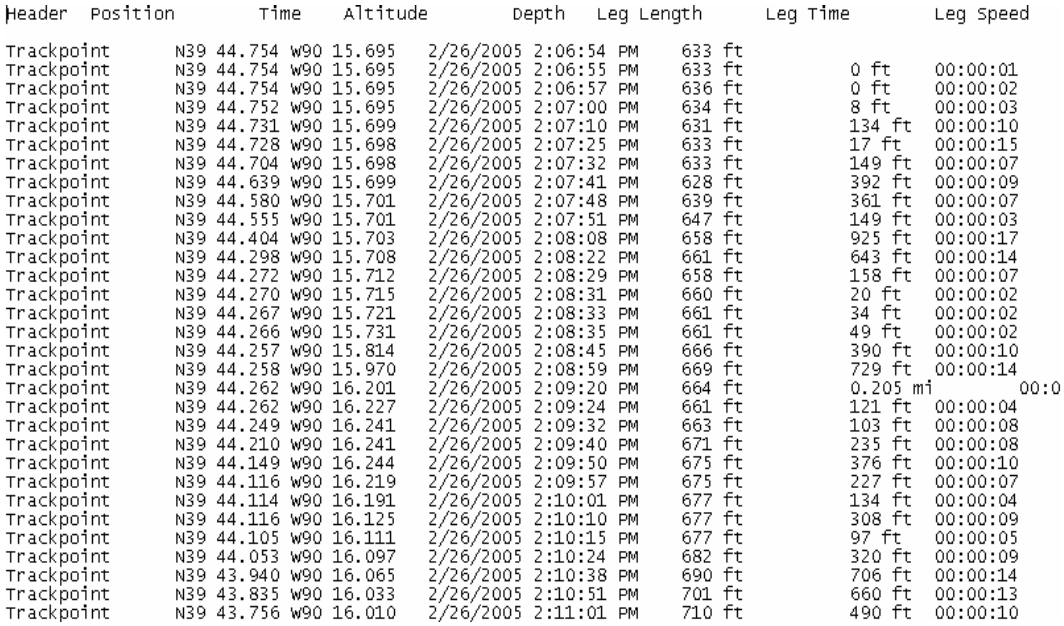
2.1 Race Simulation Setup

For creating race simulations, a standalone program will be created to produce them. The program will be used only for internal use to process text files produced from GPS devices. It will create the race simulation by doing all the calculations against the feed that was sent and will also encrypt the file for distribution.

By keeping the program internally, this gives the internal staff the ability to ensure the various formats of the text files are acceptable and can be processed. This will also ensure only approved races will be processed by the program which runs the race. The racing program will only accept encrypted files produced by the setup program.

2.1.1 GPS Feed

The feed from the GPS will come in various formats. The GPS device must be able to produce a text file, which then can be sent to the internal staff to process. The setup program will read the text file for the race and extract the needed information. An example of the GPS text file can be shown in the following image:



```
Header  Position        Time    Altitude        Depth   Leg Length      Leg Time        Leg Speed

Trackpoint      N39 44.754 W90 15.695   2/26/2005 2:06:54 PM    633 ft
Trackpoint      N39 44.754 W90 15.695   2/26/2005 2:06:55 PM    633 ft          0 ft    00:00:01
Trackpoint      N39 44.754 W90 15.695   2/26/2005 2:06:57 PM    636 ft          0 ft    00:00:02
Trackpoint      N39 44.752 W90 15.695   2/26/2005 2:07:00 PM    634 ft          8 ft    00:00:03
Trackpoint      N39 44.731 W90 15.699   2/26/2005 2:07:10 PM    631 ft          134 ft  00:00:10
Trackpoint      N39 44.728 W90 15.698   2/26/2005 2:07:25 PM    633 ft          17 ft   00:00:15
Trackpoint      N39 44.704 W90 15.698   2/26/2005 2:07:32 PM    633 ft          149 ft  00:00:07
Trackpoint      N39 44.639 W90 15.699   2/26/2005 2:07:41 PM    628 ft          392 ft  00:00:09
Trackpoint      N39 44.580 W90 15.701   2/26/2005 2:07:48 PM    639 ft          361 ft  00:00:07
Trackpoint      N39 44.555 W90 15.701   2/26/2005 2:07:51 PM    647 ft          149 ft  00:00:03
Trackpoint      N39 44.404 W90 15.703   2/26/2005 2:08:08 PM    658 ft          925 ft  00:00:17
Trackpoint      N39 44.298 W90 15.708   2/26/2005 2:08:22 PM    661 ft          643 ft  00:00:14
Trackpoint      N39 44.272 W90 15.712   2/26/2005 2:08:29 PM    658 ft          158 ft  00:00:07
Trackpoint      N39 44.270 W90 15.715   2/26/2005 2:08:31 PM    660 ft          20 ft   00:00:02
Trackpoint      N39 44.267 W90 15.721   2/26/2005 2:08:33 PM    661 ft          34 ft   00:00:02
Trackpoint      N39 44.266 W90 15.731   2/26/2005 2:08:35 PM    661 ft          49 ft   00:00:02
Trackpoint      N39 44.257 W90 15.814   2/26/2005 2:08:45 PM    666 ft          390 ft  00:00:10
Trackpoint      N39 44.258 W90 15.970   2/26/2005 2:08:59 PM    669 ft          729 ft  00:00:14
Trackpoint      N39 44.262 W90 16.201   2/26/2005 2:09:20 PM    664 ft          0.205 mi        00:0
Trackpoint      N39 44.262 W90 16.227   2/26/2005 2:09:24 PM    661 ft          121 ft  00:00:04
Trackpoint      N39 44.249 W90 16.241   2/26/2005 2:09:32 PM    663 ft          103 ft  00:00:08
Trackpoint      N39 44.210 W90 16.241   2/26/2005 2:09:40 PM    671 ft          235 ft  00:00:08
Trackpoint      N39 44.149 W90 16.244   2/26/2005 2:09:50 PM    675 ft          376 ft  00:00:10
Trackpoint      N39 44.116 W90 16.219   2/26/2005 2:09:57 PM    675 ft          227 ft  00:00:07
Trackpoint      N39 44.114 W90 16.191   2/26/2005 2:10:01 PM    677 ft          134 ft  00:00:04
Trackpoint      N39 44.116 W90 16.125   2/26/2005 2:10:10 PM    677 ft          308 ft  00:00:09
Trackpoint      N39 44.105 W90 16.111   2/26/2005 2:10:15 PM    677 ft          97 ft   00:00:05
Trackpoint      N39 44.053 W90 16.097   2/26/2005 2:10:24 PM    682 ft          320 ft  00:00:09
Trackpoint      N39 43.940 W90 16.065   2/26/2005 2:10:38 PM    690 ft          706 ft  00:00:14
Trackpoint      N39 43.835 W90 16.033   2/26/2005 2:10:51 PM    701 ft          660 ft  00:00:13
Trackpoint      N39 43.756 W90 16.010   2/26/2005 2:11:01 PM    710 ft          490 ft  00:00:10
```

Figure 5: Example of GPS File

In this example of a GPS text file, the data for each point can be found with a header on the front of the line of “Trackpoint”. The actual GPS text file is delimited with tabs. The altitude and leg length from the file will be utilized to simulate the race.

2.1.2 Settings

The setup program has eight settings to process this file. The settings are “Delimiting ASCII Value”, “Header for Data Line”, “Header Position”, “Altitude Position”, “Leg Length Position”, “Feet Abbreviation”, “Mile Abbreviation”, and “Feet to Mile Factor”. The detail of each setting will be described in the following sections.

2.1.2.1 Delimiting ASCII Value

The “Delimiting ASCII Value” is the setting to keep track of how the text file is delimited. For a tab-delimited file, the ASCII value would be 9. This value can be found on the ASCII chart. The value is an integer value and will be between 0 and 255 to represent the values within the ASCII chart. The default value for this field will be 9.

2.1.2.2 Header for Data Line and Header Position

The “Header for Data Line” and “Header Position” are used to determine the line in the text file, which will hold the data readings. The “Header for Data Line” will tell the value of the field that will determine the data is present. The

“Header Position” will be used to tell the position of the field that will hold the value to determine the data line. In the example, “Trackpoint” value in the zero field position will tell the setup program to find the data lines. Both the values from our example are the default values. The field positions are zero based, so the first field would be zero.

2.1.2.3 Altitude Position

The “Altitude Position” will tell the position of the reading for the altitude on the data line. For treadmills, these readings will control the inclination. For exercise bikes, these readings will control the resistance. Each line will be compared against the previous to determine these settings by the actual race program. For the setup program, it will save the actual value of the altitude field. From the example, the altitude will be found in field 3, which is the default value.

2.1.2.4 Leg Length Position

The “Leg Length Position” will be used to determine the length between the data readings. For treadmills and exercise bikes, this reading will be used to determine the length of time a runner will be running at their desired speed within the actual race program. The setup program will find the actual value of the leg length field. From the example, the leg length will be found in field 5, which is the default value. Between the altitude field and the leg length field, the depth field is actually blank within our example.

2.1.2.5 Feet Abbreviation, Mile Abbreviation, and Feet to Mile Factor

The “Feet Abbreviation”, “Mile Abbreviation”, and “Feet to Mile Factor” will be used for altitude and leg length. For the default settings, the “Feet Abbreviation” is ft, the “Mile Abbreviation” is mi, and the “Feet to Mile Factor” is 5280. This allows the program to find the leg length and altitude between readings and convert the readings to feet. From our example, one line has its leg length measured in miles and not feet. By using these settings, the mile measurements will be converted to the common measurement of feet.

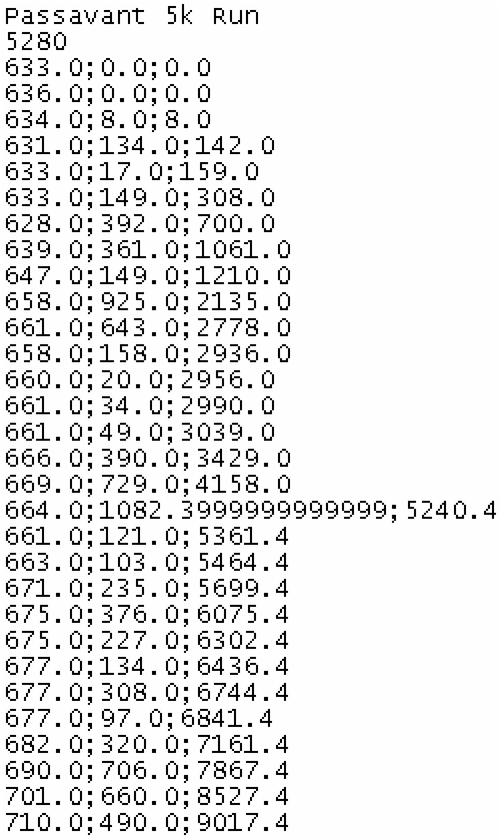
The “Feet Abbreviation”, “Mile Abbreviation”, and “Feet to Mile Factor” would also allow for readings to be done when the values are in metric. The abbreviations for feet could be changed to meter and for mile to kilometer. The factor would then be changed to 1000 to reflect one kilometer is made up of a thousand meters.

2.1.3 Creating the Simulation

When running the setup program, the program will use the previous discussed values from the GPS.INI file, or if the GPS.INI file is not present or the entries in the file are not valid, it will employ the default settings. This allows for the last settings to be saved. The GPS.INI file will be saved within the working directory for the program.

The file outputted by the program will have an extension of RAC. The setup program will prompt for a file name and a descriptive name. The file generated will have the descriptive name on the first line, the value found in the

"Feet to Mile Factor" on the second line, and then the remaining lines will be the data from the text file. The data will be altitude, then leg length, and then the cumulative leg lengths separated by semicolons. The following image shows the output created from our example of the GPS text file:



```
Passavant 5k Run
5280
633.0;0.0;0.0
636.0;0.0;0.0
634.0;8.0;8.0
631.0;134.0;142.0
633.0;17.0;159.0
633.0;149.0;308.0
628.0;392.0;700.0
639.0;361.0;1061.0
647.0;149.0;1210.0
658.0;925.0;2135.0
661.0;643.0;2778.0
658.0;158.0;2936.0
660.0;20.0;2956.0
661.0;34.0;2990.0
661.0;49.0;3039.0
666.0;390.0;3429.0
669.0;729.0;4158.0
664.0;1082.3999999999999;5240.4
661.0;121.0;5361.4
663.0;103.0;5464.4
671.0;235.0;5699.4
675.0;376.0;6075.4
675.0;227.0;6302.4
677.0;134.0;6436.4
677.0;308.0;6744.4
677.0;97.0;6841.4
682.0;320.0;7161.4
690.0;706.0;7867.4
701.0;660.0;8527.4
710.0;490.0;9017.4
```

Figure 6: Example of Processed GPS File

The race program will be able to read in the RAC file and process it. For treadmills, this information will determine the length of time and the inclination. For exercise bikes, this information will be for the length of time and the resistance.

2.1.3.1 Calculations

Within the setup program, calculations will be done to produce a proper output file. These calculations will be for converting measurements to a common reference and have a cumulative length for each measured line.

The converting of measurements must be done to ensure a common reference. From the example, one field has a length of .205 mi. Since the common measurements are done in feet, .205 will be multiplied by 5280 for the "Feet to Mile Factor" to get 1082.3999999999999 feet. This value will be rounded to 1082.4 feet by the race program.

The cumulative length will represent the total length of the race to the current point. This will be used within the race program for its calculation. It would also allow for the program to be able to know the exact distance of the race by reading the very last line for the cumulated value.

2.1.3.2 Encryption

The output file produced by the setup program will be encrypted, so the information will only be able to be read by the race program. The file will then be read in using the hashing routine to decrypt the file by the race program. This will ensure only the proper file will be used by the racing program. If the file is not in the proper format, the race program will raise an error and not process it.

2.2 Running the Race

The race program will utilize the race file produced from the setup program. The program will allow the selected race to be simulated on either a treadmill or exercise bike with the iFit® technology capability. The device running this program must have the ability to output sound to the exercise equipment. The exercise equipment will have an audio input port to receive the sound from the device running the program. The proper audio cable will need to be employed between the exercise equipment and the device running the program to ensure the proper transmission of the sound.

The audio outputted by the program will be in a wave format described in appendix A. Instead of generating a wave file on the device, the wave output will be generated and sent within the program to eliminate the need for disk access.

2.2.1 Settings

The race program will have six settings to execute the race properly. These settings are “Minimum Incline”, “Maximum Incline”, “Incline Increment”, “Minimum Speed”, “Maximum Speed”, and “Number of Times to Send Tone”. The detail for these settings is described within the following sections.

2.2.1.1 Minimum Incline

The “Minimum Incline” setting will be the lowest value the exercise equipment will incline. For an exercise bike, the lowest incline will be the lowest resistance the bike supports. For a treadmill, this will be the lowest incline of the equipment. The default value will be 1.5, which means the lowest inclination will be 1.5% for the exercise equipment. The program will use this to know the inclination generated by the race program using the simulated race cannot be lower than the minimum incline value. An example of this need is when the race actually goes downhill. Since exercise equipment does not support downhill running or biking, the program will set the race to be at the minimum inclination when the race goes downhill.

2.2.1.2 **Maximum Incline**

The "Maximum Incline" setting will be the highest value the exercise equipment will incline. The default value will be 10, which means the highest inclination will be 10% for the exercise equipment. For races with steep inclines, the program would raise the equipment to the highest inclination if it is above the "Maximum Incline" value.

2.2.1.3 **Incline Increment**

The "Incline Increment" setting will be the incrementing factor the exercise equipment will be able to rise. The default value will be .5, which means the equipment will be able to be raised in increments of .5%. Using the default values, the exercise equipment can be raised 1.5, 2, 2.5, 3, 3.5, and in the same pattern until 10 is reached. When the inclination is calculated, the calculated values will be rounded to the nearest increment.

2.2.1.4 Minimum Speed

The “Minimum Speed” setting is the minimum speed the exercise equipment will support. This will be the lowest speed supported by the exercise equipment. For the default value of 1, the entered value for the speed to run or bike the race will not be able to be below this value.

2.2.1.5 Maximum Speed

The “Maximum Speed” setting is the maximum speed the exercise equipment will support. For the default value of 10, the entered value for the speed to run or bike the race will not be able to be above this value.

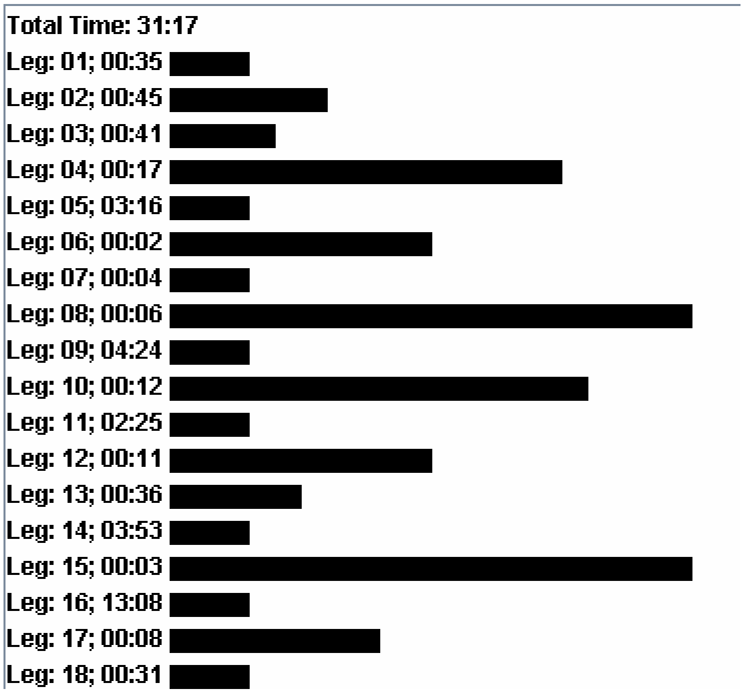
2.2.1.6 Number of Times to Send Tone

The “Number of Times to Send Tone” setting is the number of times to send the sound tones to the exercise equipment. The default value is 2. This means that the audio is sent twice for each time it is sent to change the equipment. Sending it more than once ensures the equipment receives the audio for the changes.

2.2.2 Selecting a Race

When selecting the race file, the program will bring up a file selection box to allow for files with the RAC extension to be selected. The selection process will include the ability to change directories in case the needed RAC file is not in the current working directory. If a file being selected is invalid, the selection process will display an error informing the user. A valid file must have an RAC extension and be able to be decrypted properly with the fields populated correctly within the file.

When a valid file is processed, the main screen will display a graph of the race. The following image shows an example of the graphic display:



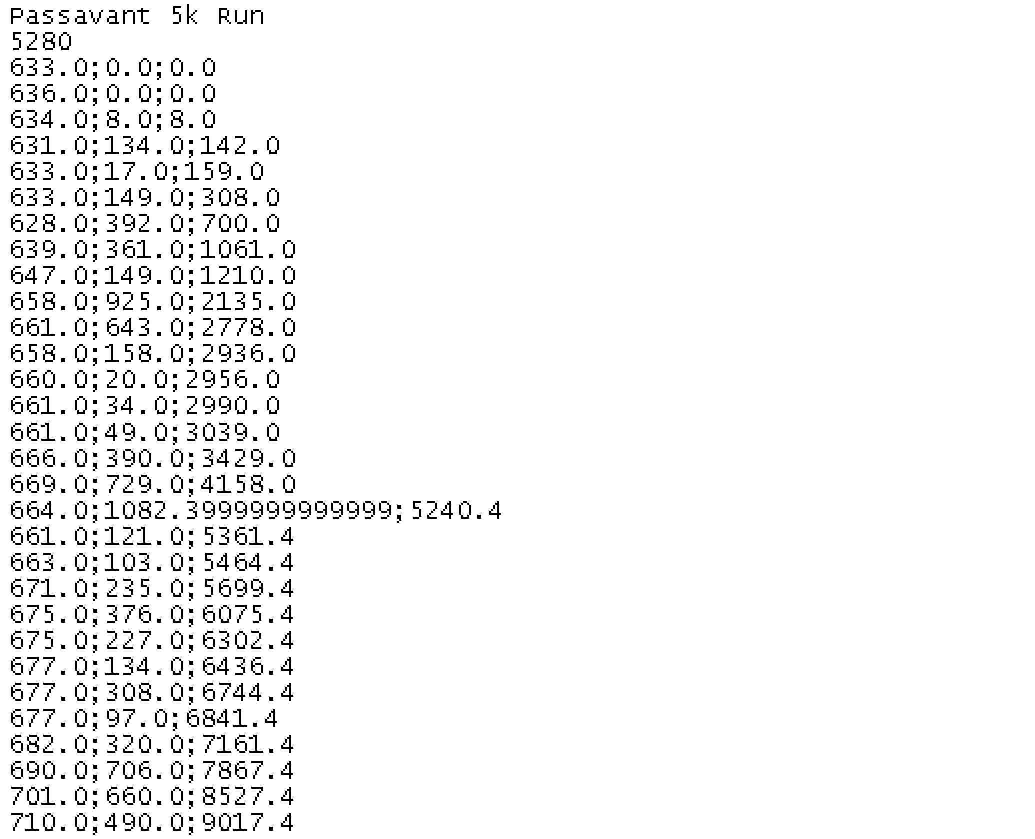
```
Total Time: 31:17
Leg: 01; 00:35
Leg: 02; 00:45
Leg: 03; 00:41
Leg: 04; 00:17
Leg: 05; 03:16
Leg: 06; 00:02
Leg: 07; 00:04
Leg: 08; 00:06
Leg: 09; 04:24
Leg: 10; 00:12
Leg: 11; 02:25
Leg: 12; 00:11
Leg: 13; 00:36
Leg: 14; 03:53
Leg: 15; 00:03
Leg: 16; 13:08
Leg: 17; 00:08
Leg: 18; 00:31
```

Figure 7: Display of Selected Race to Run

Bars will be used to show the height of the inclination with the bars running horizontally. To the left of the inclination bar, the length of time the program will be at this inclination will be shown. This will give the user a graphic representation of the race they selected.

How the race is processed and shown in the display is based upon the settings entered before the selection. The settings are stored in TREADMILL.INI file found in the working directory. If the TREADMILL.INI file is not present, the default settings will be used. If the values found in the TREADMILL.INI file are not correct, the default settings will also be used. Holding the values within an INI file will ensure the user can set the race program up for their equipment one time and pull the values on each subsequent run.

To calculate the inclination for the race, the current altitude and leg length must be known and the next altitude needs to be determined except for the first two lines, which will store the description and “Feet to Mile Factor” value. The following example from the decrypted race file will be shown to explain the calculation:



```
Passavant 5k Run
5280
633.0;0.0;0.0
636.0;0.0;0.0
634.0;8.0;8.0
631.0;134.0;142.0
633.0;17.0;159.0
633.0;149.0;308.0
628.0;392.0;700.0
639.0;361.0;1061.0
647.0;149.0;1210.0
658.0;925.0;2135.0
661.0;643.0;2778.0
658.0;158.0;2936.0
660.0;20.0;2956.0
661.0;34.0;2990.0
661.0;49.0;3039.0
666.0;390.0;3429.0
669.0;729.0;4158.0
664.0;1082.3999999999999;5240.4
661.0;121.0;5361.4
663.0;103.0;5464.4
671.0;235.0;5699.4
675.0;376.0;6075.4
675.0;227.0;6302.4
677.0;134.0;6436.4
677.0;308.0;6744.4
677.0;97.0;6841.4
682.0;320.0;7161.4
690.0;706.0;7867.4
701.0;660.0;8527.4
710.0;490.0;9017.4
```

Figure 8: Example of Processed GPS File

When processing each data line, each line is processed and compared to the following line. If the next line’s altitude is different than the current altitude, the next line’s altitude is subtracted from the current line’s altitude with the product being divided by the length and multiplied by 100 to the get the percentage of inclination. The percentage of inclination would then be compared against the “Incline Increment” value to round the inclination to the nearest valid increment for the exercise equipment. If the current line’s inclination and the next line’s inclination is the same, the lengths of the lines are combined.

For the values from our example using our default settings, the altitude of 628 with the next altitude being 639 is the first inclination that changes. For 308 feet, the inclination would be at 1.5%. For 392 feet, the inclination would change to 3% after it is rounded to the nearest .5%.

The first six lines of data’s inclinations fall below the minimum inclination. Since they fall below it, the minimum inclination is substituted for the actual inclination. If the inclinations are above the maximum inclination value, then the maximum inclination would be substituted for the inclination.

If the last line is being processed, the current line’s altitude is substituted in the equation for the next line, so the change in inclination will be zero. This would mean the treadmill is at the minimum inclination during the last leg.

To calculate the length of time of the inclination, feet per second must be calculated. The speed is multiplied against the “Feet to Mile Factor” value found on the second line. The value is then divided by 60 minutes in an hour and then by 60 seconds in a minute. For running a race at 6 MPH, it would mean the race would be 8.8 feet per second.

For the first inclination of 1.5%, the 308 feet would be divided by 8.8 feet per second to get 35 seconds. For the second inclination of 3%, the 392 feet would mean it would take the runner 45 seconds.

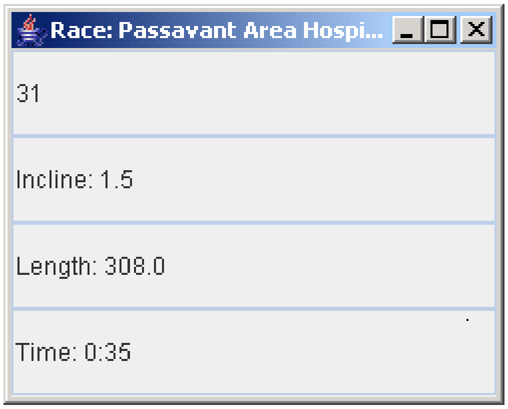
Along with the calculations for the display, the sound output is generated at this time and stored in memory. Appendix A explains the process of generating the sound output. The sound output is what will control the inclination and speed on treadmills or the resistance on exercise bikes. This will increase the speed of the program by having the overhead of generating the sound waves upon the selection of the file instead of during the actual running of the race. If the overhead was during the running, the length of time of running the race would not reflect true length of the time needed to run each leg.

2.2.3 Running

The actual running of the race is in three parts: start, processing, and stopping. “Start” and “Stop” options are part of the menu options. The time to run each leg is used to send the sound at the appropriate time.

2.2.3.1 Starting

When the start option is selected, a secondary window will appear. An example of the secondary window is:



31

Incline: 1.5

Length: 308.0

Time: 0:35

Figure 9: Timer Display

The secondary window will have the first leg's information and send the sound for the leg to the audio port. The top line of the window will have a countdown for the length of time it takes to run the leg. The second line will be the incline for the leg. The third line will be the distance of the leg. The bottom line will be the total amount of time it takes to run the leg.

2.2.3.2 Processing

Once the countdown is at zero, the next leg's sound is sent to the audio output port. The information for this leg is then displayed in the secondary window. Each leg is then processed in order until completed.

2.2.3.2 Stopping

Once all the legs are processed, a final sound wave will be sent to the exercise equipment to tell it to set the inclination back to the minimum and the treadmill to stop. The secondary window will close at this time.

If the "Stop" option is selected off the main window, the secondary window will close. The user can then stop the exercise equipment manually without worrying about commands being sent to the equipment.

2.3 Web Site

A web site will be used for communications and commerce for the program and its associated races. The communication piece will focus on users contacting the internal staff on technical issues plus if they wish to submit a text file of a GPS reading for a race they wish to add to the web site. The commerce piece will utilize PayPal® tools to allow for wide arrange of credit cards over secured transactions.

2.3.1 Communications

The communications' piece for the web site will focus on customers contacting for technical issues as well as runners or bikers uploading a race they would like simulated. The goal of this process will be to have an e-mail capability to communicate to the internal staff, which will allow for timely communications without a need for constant support by the staff. The staff will then be able to focus on other jobs while still maintaining a high standard of support.

A user with a technical issue will be able to click on a technical contact button, which then will prompt them for information. The prompted information will be “Subject”, “Message”, “Name”, and “E-mail”. This will allow the user to put in a subject of his question, the question or issue in the message box, their name, and their e-mail for return contact. The person on the staff in charge of technical issues will receive the message and be able to respond via e-mail without the user having to know the staff person’s e-mail address.

The uploading of race information will be handled in a similar way. The prompted information will be “Race”, “Description”, “Attachment”, “Name”, and “E-mail”. This will allow the user to enter the name of the race, a description of it, attach the GPS text file of the readings they took, their name, and their e-mail address. A brief explanation will be on the page explaining the optimal GPS readings needed for the best quality simulation of the race. The person on the staff in charge of creating the race simulations will be able to review the text file and process it through the simulation setup program after adjusting the needed settings. With the name and e-mail address of the individual submitting the race, the user can be notified when the race has been processed and added to the site.

2.3.2 Commerce

The commerce part of the website will be focused on selling the race program and simulations over the Internet. The program or simulations will be purchased through secured transactions utilizing PayPal® tools, and when successfully purchased, will then automatically contact with the purchaser's e-mail address and their selected purchases.

The race program and the race simulations will be available for users to purchase. The internal staff will modify the web site to add additional race simulations as they become available. The internal staff will also set the prices for the race program and the various simulations. For purchasing, PayPal® utilities will be used.

2.3.2.1 PayPal® Utilities

PayPal® provides a free shopping cart utility. This utility will allow for the multiple items to be purchased at one time. Once the user fills his shopping cart, the user can then purchase their items.

For purchasing, PayPal® gives the ability to use existing tools for secure credit card transactions. PayPal®'s Website Payments Standard allows for users to make purchases without set-up, monthly, and gateway fees (Paypal. "Compare Solutions"). The standard option will also accept major credit cards and bank transfers taking a lot of the e-commerce issues experienced when customizing a purchasing interface (PayPal. "Compare Solutions"). The fee for using this utility will vary based upon monthly sales volume but will range on a per transaction basis from 1.3% to 2.9% of the total price of the purchase with an additional $.30 charge (PayPal. "Compare Features"). The standard option will also allow for a free shopping cart utility to be added to the website to allow the user to purchase multiple items such as multiple simulations with or without the race program (PayPal. “Compare Features”). Using the PayPal® tools with the website, the secured credit card transaction requirement with acceptance of major credit cards without additional static fees will be met.

When a successful purchase is completed, PayPal® will notify the internal staff of what was purchased and the purchaser’s e-mail. The internal staff will have an automated process to monitor the notifications.

2.3.2.2 Automated Transfer of Purchased Items

The automated transfer of purchased items is a process developed to transfer purchased items to the users. Once a successful purchase occurs, the user will be sent to a screen displaying a message stating to expect their purchases in an e-mail message.

When PayPal® sends a successful purchase notification; the automated process will extract the e-mail address of the purchaser and their purchases. An e-mail message will then be generated and sent to the user with their purchased items attached to the e-mail. The e-mail will also give directions on how to utilize the purchased items whether it is the race simulation program or the simulated races.

For the race simulation program, a setup program will be used for Windows installs. This is done to install the Java runtime environment and the race simulation program along with a shortcut to be placed on the desktop.

On non-Windows environments, the user will install the Java runtime environment themselves with instructions on setting up the program. Detailed instructions will need to be generated for the user with links to the correct Java runtime environment. This will be changed in future releases to have a setup program supported in other environments.

Chapter Three: **Project Methodologies**

3.1 Life-Cycle Model

In order to accomplish this project, the project was divided into six phases. This practice was utilized in the system development life cycle, or SDLC. The phases are Analysis, Design, Construction, Testing, Implementation, and Maintenance. The phases were designed to allow for the team to step through the phases until the project could be released and arrive in the maintenance stage.

3.2 Phase I - Analysis Model

Phase I of the project was the Analysis. This phase determined what needed to be done to analyze the initial idea of producing the race simulation based upon GPS readings. The feasibility of the project was determined, plus what elements were needed and how they related to one another.

After the introduction, the analysis model was documented by being separated into sections. A technology assessment section described the programming language choice and tools to be used for this project. The system objectives section described the goals the system needed to achieve to be successful. The project phases' sections were the breakdown of the different phases and a brief description of them. The phases were created as an organized process to complete the project. A Project timeline gave a timetable to complete the phases of the project. The project budget showed the estimated costs of the project.

Once the analysis phase was completed, the design was done.

3.2.1 Introduction

The initiator of this project approached the project developer with the idea of taking readings from his GPS device and creating sound waves to control his treadmill. The initiator had previously used the iFit® technology with his treadmill and experimented with it to determine how sound waves were being manipulated. The initiator had even found free source code for an existing Microsoft Visual Basic program on the Internet that showed the sound patterns, described in appendix A, for iFit® technology, but the code did not accomplish what the initiator wanted. The developer had the extensive development experience the initiator needed to make his idea into a reality.

Both people involved had run extensively in the past but found their time limited due to work, family, and education schedules. The initiator's idea was to have the developer develop the system to allow them, and others in the same predicament, to train for races.

3.2.2 Technology Assessment

When determining the technology to utilize, Java seemed most feasible. The initial idea was to use Java due to its extensive capabilities to go on other platforms outside of Microsoft. The goal was to go beyond having desktop PC's or laptops to be able to run the race simulation.

The second concern was the developer's extensive use of Microsoft's products in his current job. Their employment contracts state that any idea or use of technology linked to their employer becomes the property of the employer. The developer's use of Java was done for course work at Regis University and not for his employer. The developer would use Java from Sun Microsystems, Inc., (offered for free on the Internet) which further eased employer concern over technological properties.

Another concern arose with the acceptance of payment. Creating a secure credit card system to accept payment would have taken an extremely long time and have been risky. It was determined to utilize PayPal® tools to be able to implement a secure credit card transaction system without a monthly payment. Since PayPal® was utilized, bank transactions are shown on their account statements to track sales.

3.2.3 System Objectives

The objectives for the project were determined after an assessment was completed. Seven objectives were identified and are listed below.

Create a race simulation to be used on numerous devices and not limited to Microsoft technologies. This would increase the possible market by including users utilizing Macintosh, Palm, Linux, or other operating systems.

The race simulation program should be easy to use. If it is hard to use and understand, users will not purchase or utilize the program.

To meet the users' demand for the program to run on Windows, the race simulation program must be easy to install on these users' systems. Even though other operating systems will be supported, the Windows operating system is the most popular in the world. Due to a previous project done by the developer, Wise Solutions' InstallMaster 8.1 for Windows environments is already available for a Windows setup program to be generated.

With such a high number of users on the Internet, the best and easiest way to distribute the application and race simulations is over the Internet. Internet users will need to be able to purchase the race program and available races, and after a successful purchase be able to receive them automatically.

With such a high liability from Internet commerce, secure credit card transactions need to protect from any liability found from unlawful access to credit card files. The easiest method for this is to utilize third party tools that have a proven history against unauthorized access. A third party tool will also already be able to support purchases from major credit cards.

Allow for users to send races to have a simulation created. This will increase the number of races to support by allowing others outside of the project to record races.

Ensure protection for the technology created within this program. The project team wanted to protect their hard work and ideas. This determined the need to have the race simulation setup program for internal use only, the need for encrypting the race file, and the ability to create the sound waves dynamically in memory without any saving to a device like a hard drive.

3.2.4 Project Phases

To do this, it was determined to break the project into the six phases of Analysis, Design, Construction, Testing, Implementation, and Maintenance.

Analysis determined the feasibility of the project and determined the needs including making choices to keep costs low. This was especially important with the limited financial resources of individuals doing the project.

Design was the design of the various elements of the project and their interaction. This stage was broken into pieces in the project with the pieces working together. Breaking the project into pieces sped up development by not having one large piece to do everything, but specialized elements focusing on one function and working with the other elements when the function was needed.

Construction created the programs needed within this project by using the design. This was the programming part of the project.

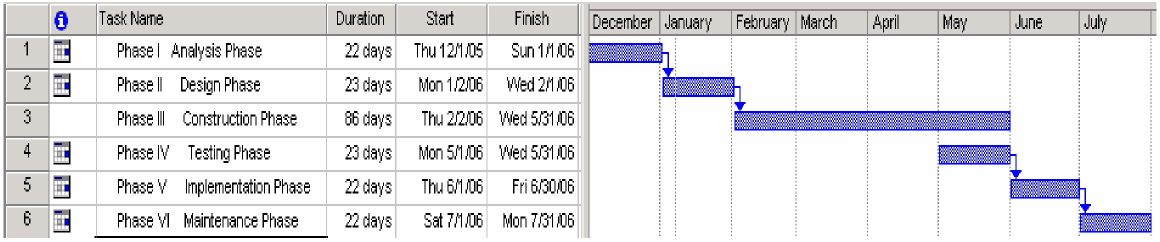
Testing overlapped with the last month of construction. This phase ensured everything operated properly.

Implementation was moving the system into production. This included working with the Internet provider and ensuring the secured credit card transaction processing was in place for production.

Maintenance was the documenting and reviewing of the maintenance processes that needed to be done on a timely basis. With two people initially doing all the work, this improved overlapping of their jobs to allow for vacations, business travel, or other issues that took one away from doing his job for a period of time.

3.2.5 Project Timeline

Due to the limited free time of the project team, it was determine that a very modest approach would be done on the project schedule. This gave them the ability to work on the project and still have time for their employment and personal obligations. The following chart shows the timetable for the project.



	ℹ	Task Name	Duration	Start	Finish
1	▦	Phase I Analysis Phase	22 days	Thu 12/1/05	Sun 1/1/06
2	▦	Phase II Design Phase	23 days	Mon 1/2/06	Wed 2/1/06
3		Phase III Construction Phase	86 days	Thu 2/2/06	Wed 5/31/06
4	▦	Phase IV Testing Phase	23 days	Mon 5/1/06	Wed 5/31/06
5	▦	Phase V Implementation Phase	22 days	Thu 6/1/06	Fri 6/30/06
6	▦	Phase VI Maintenance Phase	22 days	Sat 7/1/06	Mon 7/31/06

Figure 10: Project Timeline

From the chart, the month of December was dedicated to the analysis phase. The design phase was the focus of the month of January. Construction was given three months to complete with the final month coinciding with the testing phase. Implementation was done during June, and the final month of the project, July, was dedicated to maintenance.

As shown from this timeline, the hope was to have the system fully implemented by the middle of summer in 2006.

3.2.6 Project Budget

Since this project was being done independently of a company, the costs associated with it were kept to a minimum. All work was done by the project initiator and developer with their already purchased personal equipment. This included laptops, desktop PC's, a treadmill, and an exercise bike. The following breakdown shows the low development costs from the project.

Java is a free download from Sun Microsystems, and the developer had already utilized it for his classes. With that knowledge, all programming was determined to be able to be done by this free tool.

PayPal® Website Payments Standard is free to use due to its charges at the time of the purchase. This eliminated any costs to the project team. PayPal® subtracts their fee from the actual purchase.

Wise Solution's InstallMaster 8.1 had been purchased in the past for other projects and was available for use at no additional costs. InstallMaster can be used for Windows installations thus eliminating the need for an installation package to be purchased for the Windows users.

For the budgeted costs of this project, $10 per month was allocated for third party web hosting, and $35 per year was needed for registering the name of the website. To minimize the need for space and security, the web site has options to purchase the race simulation program and the races, but the programs and races reside on an existing PC available to the internal staff. With purchasing being provided through PayPal®, the web site is only used for communications and selecting items to purchase.

3.3 Phase II – Design Phase

The design phase, or phase II, focused on planning of the project by designing the differing elements needed to meet the requirements from the analysis phase and how the differing elements worked together.

Three elements were created for this project. The first element was the race simulation setup, which took the GPS feed and created the race simulation for the race program. The second element was the race program to run the simulation. The last element was the website to allow for commerce and communications. The website accepts GPS feeds of races, allows the user to purchase the race program and existing simulations, transmits the purchased items to the user on successful purchases, and allows communications for user support. These three elements work together to deliver the objectives for this project.

3.3.1 Meeting the System Objectives

The system objectives from the analysis phase are protecting the technology, Internet users purchase and receive items, secure credit card transactions, used on numerous devices, setup program for Windows, easy to use race simulation program, and allow for submission of GPS feeds. These objectives were determined during the analysis as what would be needed to have a successful project. The following describes the meeting of each objective.

3.3.1.1 Protecting the Technology

The requirement for protecting the technology is concerned with protecting the intellectual property of the project. The development team wanted the programs they created not to be pirated and also the programs to run correctly. They believed protecting the technology would bring a quality product to the market from their hard work and long hours of development. To meet the requirements, the program was divided into the three pieces described above.

With the many possibilities of GPS feeds, the race simulation setup piece is only used internally by the project team. GPS feeds are sent via the website to the project team who can then analyze them. The staff is then able to determine whether the feed was done properly and check to make sure it is not a duplicate race. If the feed is good and a new race, the staff can then set parameters within the race simulation setup program to process the GPS feed. The output file created is the race simulation. The race simulation is encrypted so the race simulation program only runs simulations created by the internal staff.

For added security, the race program and simulations are not stored on the website. The program and simulations are stored on a PC used at the project team's location. The PC has an automated process to e-mail the purchaser their purchased items.

The notification of successful purchases is done by PayPal®. It sends the purchaser's information and selected items to the automated process. Only successful purchases have notifications sent.

PayPal®'s tools kept the staff from having to create a secure credit card transaction system. The elimination of this process protected the project from having an additional piece to be concerned with. The focus was then on the race simulations without spending a large amount of the project's hours on an extremely important piece as electronic commerce.

3.3.1.2 Internet Users Purchase and Receive Items

Internet users are able to browse a website to purchase the race simulation program and race simulations. From the “Protecting the Technology” section, the PayPal® tools are used to purchase the items. The tools also support automatic notification. The notification goes to an automated process supported by the internal staff to send the purchased items to the e-mail address supplied within the notification.

By sending the purchased items via e-mail and not on the website, the users are able to save their received e-mails, the internal staff can review sent items, and the size of the space needed from the web hosting service are kept to a minimum. Reducing the space reduces the budget needed for services from the web hosting service. The e-mails give the added benefit of a history of what was sent to the users, plus on unsuccessful e-mails, the error returned can be reviewed by the internal staff for better quality service in the future.

3.3.1.3 Secure Credit Card Transactions

Creating a secure credit card transaction process would have become costly to the project. To eliminate the costs, existing tools were used. PayPal® was selected for its vast support of major credit cards, and its proven tools.

Using a third party tool also eliminated the liability that is incurred when a secure credit card transaction process is created internally. Third party vendors, like PayPal® specializes in electronic commerce, and the internal staff do not have the expertise they would offer.

3.3.1.4 Used on Numerous Devices

To meet the needs of the users, the race simulation program must be able to run on multiple devices. This included non-Windows devices such as Macintosh, Linux systems, or handheld PC’s, plus mobile devices to increase the ability for the user to take the simulation to health clubs where a PC could not be hooked directly to the treadmill or exercise bike.

Java allows for multi-platform support. Sun Microsystems allows for Java runtime environments to be downloaded to devices and run Java programs. The compiled Java race simulation program can then be called from Java runtime for the platform.

3.3.1.5 Setup Program for Windows

Even though the race simulation program is able to run on multi-platforms, Windows is the most popular platform in the world. To enhance the installation of the program on Windows, Wise Solution’s InstallMaster 8.1 created an installation process to install the Java runtime environment for Windows, the race simulation program, and a shortcut to execute the race simulation.

When purchasing, the user is able to select the Windows version of the race simulation software. After successful purchase and notification, the automated process to e-mail the purchased items send the setup program created by InstallMaster. Installation instructions are part of the e-mail message when it is sent.

3.3.1.6 Easy to Use Race Simulation Program

The race simulation program needed to be easy to use to increase user satisfaction and limit the need for training. The options within this program are minimized to achieve this requirement. The program needed to do the following.

Races are stored on the hard drive of the device running the program. The program had to be able to browse to a race and select it. Once selected, it gives a graphic display of the race.

Once the race is selected, the program allows the user to start running the simulation. The user needs an easy way to start the race. This gives the user time to review the graphic display of the race and select another race if it is not correct.

After the race is started, it then needs to allow the user to stop the race if needed, or if the race is ran to completion, the simulation needs to stop automatically. If the user stops the race simulation, the automatic stop is sent, but the user can then stop the exercise equipment manually. Automatic stop stops the race at the exact distance from the mapped course.

The program also stores settings for the exercise equipment. Once the settings are set, they are stored for future use. This eliminated the need for setting the exercise equipment settings within the program each time the program runs in the future.

Help menu needs to give instructions to the users as well as an “about selection” to give the version of the program. This improves the customer support by having the user be able to reference the version when contacting customer support. Customer support knows exactly what version of the program they have and whether a later version fixed their issue.

The final option allows the user to exit the program. When the user exits the program, the program exits gracefully and does not send any control sounds to the exercise equipment.

These options limit the complexity of the system. Future versions may increase the complexity of the program, but for the initial release, it was decided to minimize the complexity. Minimizing the complexity sped up the programming by focusing on the core needs.

3.3.1.7 Allow for Submission of GPS Feeds

To increase the number of races to support, submissions are allowed from other sources outside of the internal staff. An example is a marathon in Alaska can be mapped out and sent to have a simulation created for it even though the project team was not able to get there to measure it. By allowing the web piece to accept GPS feeds, this allowed others to submit races. The submission needs to describe the race and send a text file from the GPS device of the readings of the race.

Instructions are on the website to inform the submitter the best possible way to take GPS readings of a race. The readings need to be taken to reflect the starting point of the race, points within the race where elevation has changed, and the ending point. Accurate GPS readings give the best possible simulation of the races.

3.3.2 Interacting with the Different Pieces

Three pieces were needed to interact with each other to reach the objectives. The three pieces are race simulation setup, race program, and the website. The following sections diagram the process of determining the separate pieces and how these pieces interact with each other.

3.3.2.1 Use Case Diagram

The original diagram during the project design was the use case diagram. This illustrates the interaction between the users and the system. The following illustration shows the use case utilized for this project:

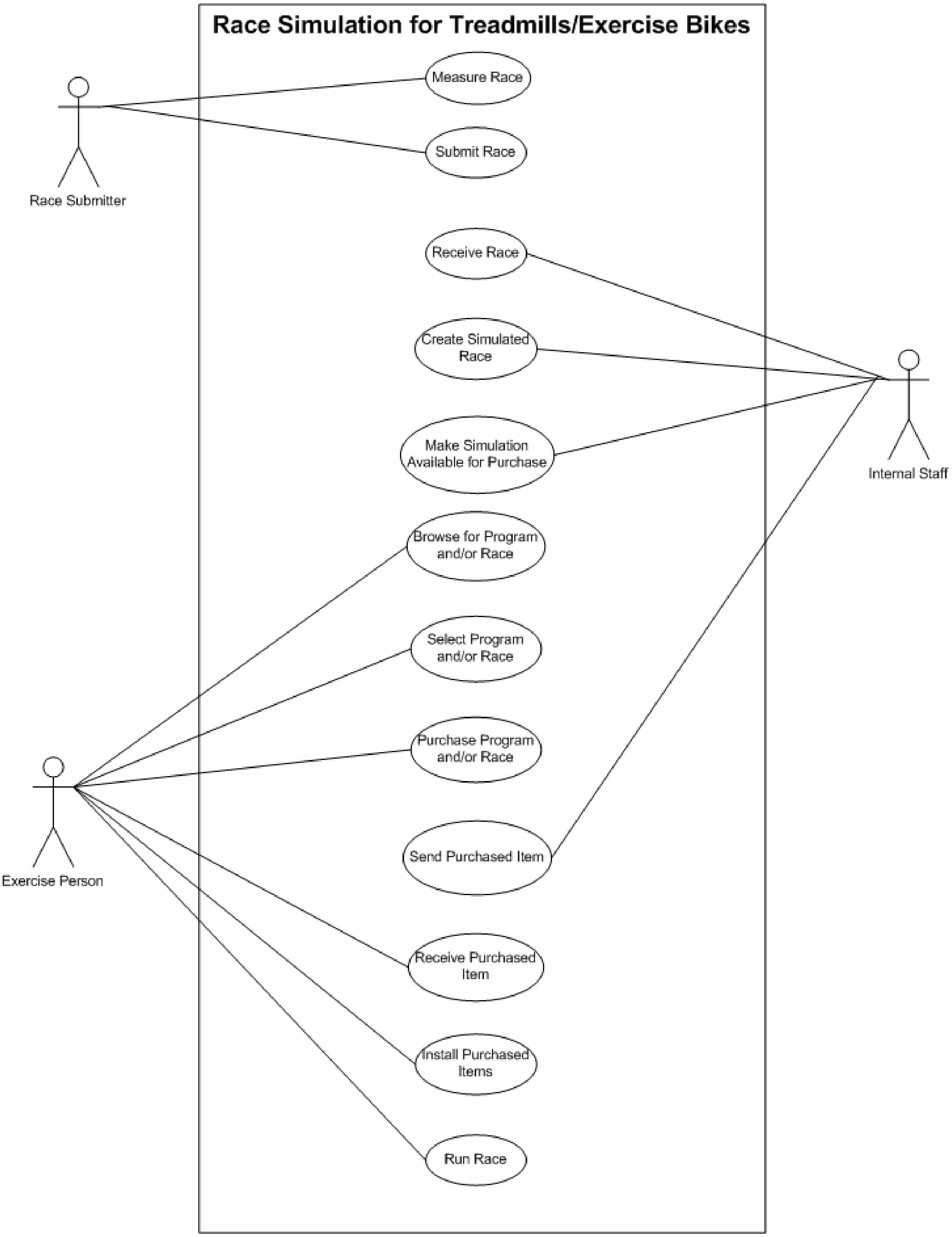


Figure 11: Use Case Diagram

Three actors are shown interacting with the system. They are “Race Submitter”, “Internal Staff”, and “Exercise Person”. These actors interact with twelve processes within the system.

The “Race Submitter” has the role of measuring a race and submitting it to the system. The “Race Submitter” utilizes their GPS device to record the race. The “Race Submitter” could be either a member of the internal staff or an exercise enthusiast wishing to simulate the race. Once the measuring is completed, they submit the race to the system to have the simulation created.

The “Internal Staff” receive the race from the system, which the “Race Submitter” submitted. They then create the race simulation. Once created, the “Internal Staff” makes the simulation available for purchase. When an item is purchased from the system, the internal staff sends the item to the purchaser.

The “Exercise Person” is able to browse for the race simulation program, simulated races, or both. The race simulation program runs the simulated races. Once they select the items to purchase, they are then able to purchase those items. When the items have been successfully purchased, the “Exercise Person” receives those items. They then install the purchased items and run the race.

3.3.2.1.1 Class Diagram for Use Case

The next diagram is the class diagram based upon the use case diagram:

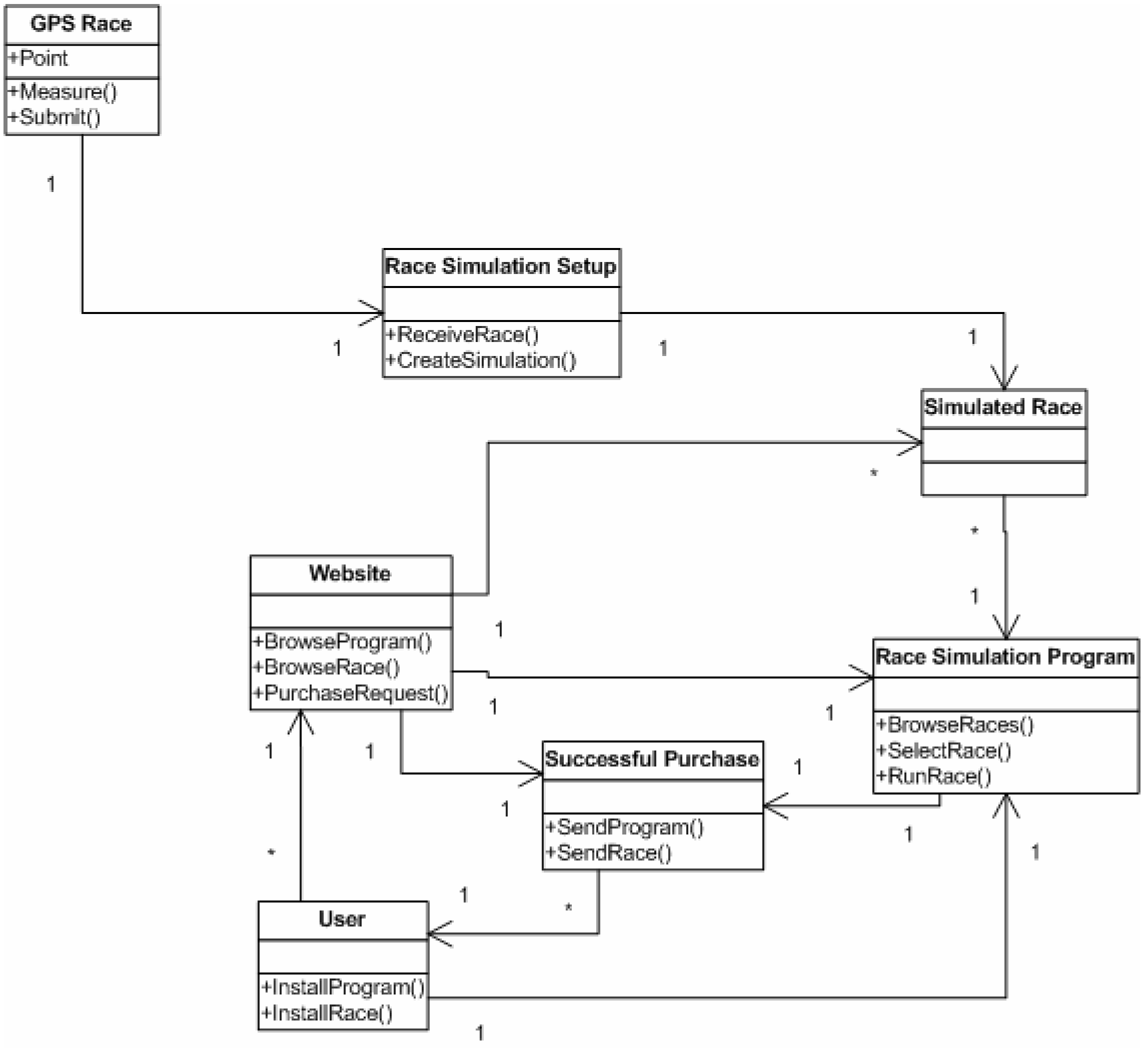


Figure 12: Class Diagram From Use Case

The GPS race class has measured points along a race. Once the race measuring is completed, the GPS race is then sent into the race simulation setup class where it will be received. The race simulation setup class creates the simulated race.

The race simulation setup class is a program for use by the internal staff. Its focus is solely on the creation of the simulated races received. This was determined to be an internal program due to the possible varieties of GPS feeds received. Since multiple manufacturers manufacture GPS devices, this program is flexible enough to process the differing feeds.

The website allows for the user to browse for the race simulation program and simulated races. If the user has decided to request any of these items to be purchased, the website supports secure electronic commerce. On successful purchases, the purchased items are sent to the user for installation. The user is then able to install the items on their system.

To run the simulated race, the simulated race program and simulated races have to be installed on the user’s systems. When running the simulated race program, the program allows the user to browse the installed simulated races, select a race, and then run it. This process separates the piece from the race simulation setup program and the website, since it must be installed on the user’s system.

Once the pieces were determined, the next step was to design each piece by diagramming them in more detail and explaining their interactions.

3.3.2.2 Race Simulation Setup

The race simulation setup program is shown in the following diagram:

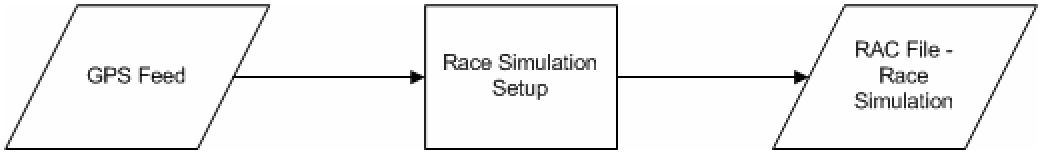


Figure 13: Diagram of Race Simulation Setup

The GPS feed are inputted into the race simulation setup program, which then make the necessary calculations to convert the information into a race simulation.

This piece was designed to allow for the GPS feeds to be prepared by the internal staff. With the numerous possibilities of possible GPS feeds, the internal staff is able to determine whether the GPS feed sent is viable.

The RAC file, or race simulation, is a file with an abbreviated name for the race as the file name and an RAC extension. The RAC file is the file used by the race simulation program to simulate the race and is created by the race simulation program after the GPS feed is converted to a usable race simulation.

3.3.2.2.1 GPS Feed Class Diagram

The following class diagram shows the pieces for creating and submitting a GPS feed to the race simulation setup piece:

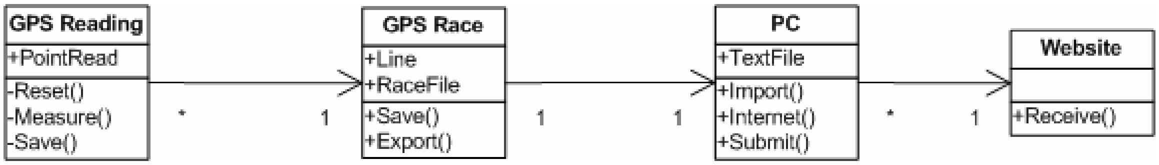


Figure 14: Class Diagram of GPS Feed

Even though this is prior to the race simulation setup, it is beneficial to show the steps needed for a proper GPS feed.

The GPS reading and race classes occur on the GPS device. The GPS device can be reset before taking readings, it measures multiple points along the race, and saves them. When the points are saved, the GPS device saves them together as a race. The points are saved as lines within the race file. The race file can then be exported from the GPS device.

A PC imports the race file as a text file. The PC can then use its Internet connection to submit the text file to the website. The website is reviewed later.

This shows the GPS feed utilized in the race simulation setup piece is measured in pieces. The first point measured after the device is reset will be the starting line. As each point is measured and saved, the race will be mapped out. The final point measured before saving the race file is the finishing line. As more points are measured in a race, the race increases in detail. The detail is the length from the starting point along with the elevation at that point.

3.3.2.2.2

Race Simulation Setup Class Diagram

The race simulation setup class diagram is used for programming this piece:

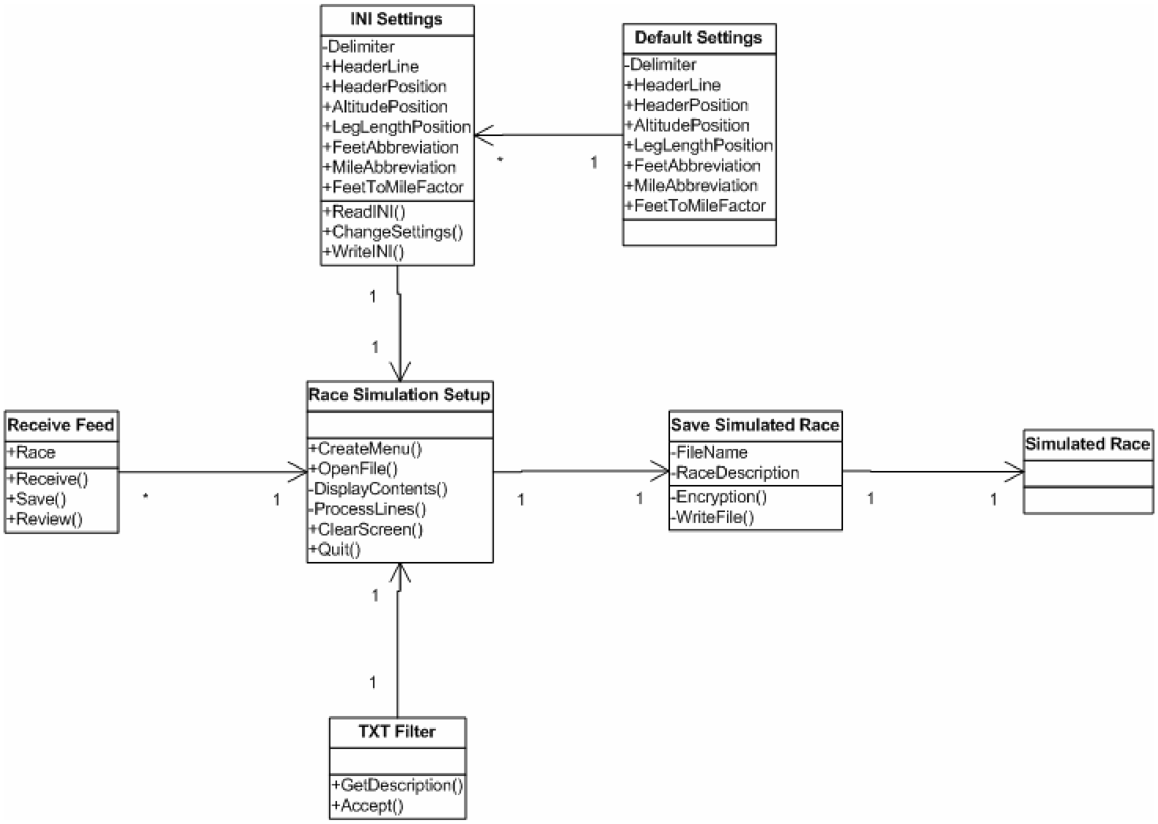


Figure 15: Class Diagram of Race Simulation Setup

The diagram shows the classes used with the race simulation setup piece to create the simulated race.

The receive feed class deals with the GPS race file transferred from the submitter’s PC to the website. The race file is received, saved to a file, and then reviewed by the internal staff. As stated in the GPS Feed Diagram section, it is crucial for the race to be measured properly for an accurate representation. This step ensures the accuracy as well as learning the settings needed to process the file within the race simulation setup.

The race simulation setup class is the main focus. It creates a menu for the interaction with the user. The settings stored within an INI file are able to be read, reviewed, and changed by the user. If the INI settings are not found, default settings are used. If the user changes the settings, the changes are written to the INI file. From the research, the settings are the field delimiter of the line, identification of the header for the data line, the field position for identification, the altitude field position, leg length field position, abbreviation for feet, abbreviation for miles, and the feet to mile factor. These settings can be changed by the user to process the receive race file.

Race simulation setup also utilizes the TXT Filter class to access only text files, which the feed of the race files are. The TXT Filter gives a description of the filter being used and only accepts files with a TXT extension. It also allows for the navigation of the directories to locate the file. Once opened, the settings are used to process the lines and display their contents to the screen. The displayed contents are the simulated race in an unencrypted format before it is saved.

If the user approves of the displayed contents, they then can save the simulated race. File name and race description are prompted for. The information for the simulated race is encrypted and then written to a file with an RAC extension. The written file is the simulated race used by the race simulation program.

Once completed, the user can quit from the race simulation setup program.

3.3.2.3 Race Simulation Program

The following diagram illustrates the race simulation program:

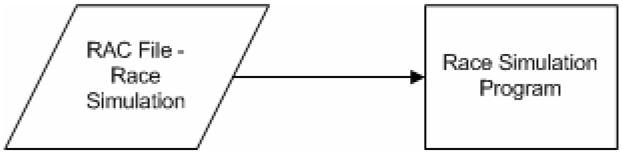


Figure 16: Diagram of Race Simulation Program

The RAC file generated by the race simulation setup program will be used as input into the race simulation program. The race simulation program will be able to use the simulation file to control the exercise equipment attached to the device running the program.

This piece is installed on the users' devices. The RAC file is encrypted to ensure only valid RAC files created from the race simulation program are used. This helps ensure the race simulation program runs correctly. Invalid files cause errors within the program at the time of execution.

3.3.2.3.1 Race Simulation Program Class Diagram

The development of the race simulation program will be based on the following class diagram:

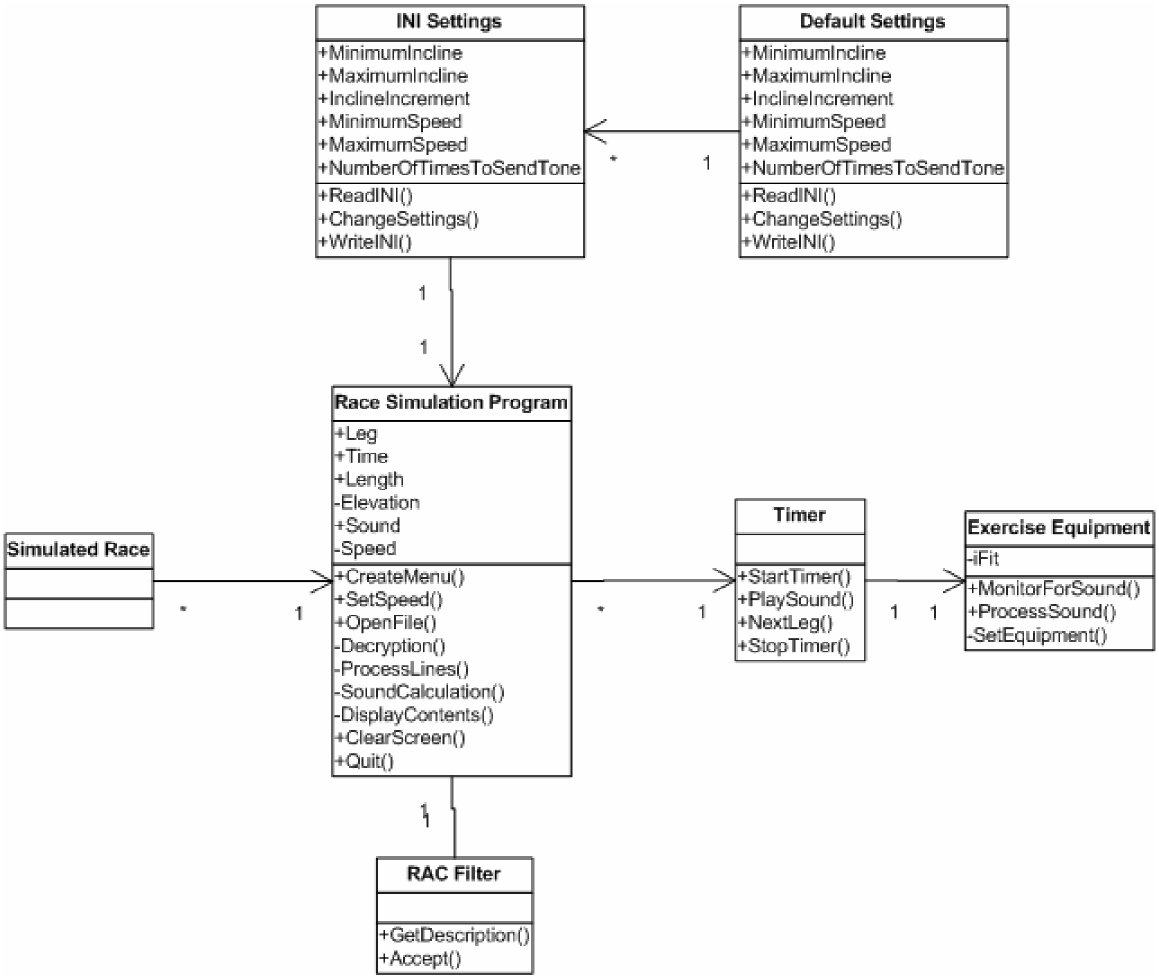


Figure 17: Class Diagram of Race Simulation Program

The class diagram utilizes the simulated races created during the race simulation setup. The race simulation program starts with the creation of a menu to allow interaction with the user and process the simulated races.

The race simulation program uses settings from an INI file to control the processing of the simulated race for the exercise equipment. The settings are minimum incline, maximum incline, incline increment, maximum speed, minimum speed, and number of times to send tone. If the INI file does not have these settings, default settings are used. The user has the ability to change these settings and write the changes to the INI file.

Once the settings are ready, the user is able to open the simulated races. By using the RAC filter, the user can browse directories and look for the simulated races. The RAC filter gives the description in the open file dialog and only accepts files with the RAC extension.

When opening the simulated race, the user sets the speed, and the file is decrypted. Each line, or leg of the race, is processed to find its elevation, length, and, based upon the speed, the time to run the leg. These values fall between the settings from the INI file. Examples are the elevation between minimum and maximum elevation and the speed within the acceptable limits. The sound to control the exercise equipment, described in appendix A, is generated during the processing of each leg. If the user reviews the displayed contents and does not choose to run or bike this race, they can also clear the screen and select a new race.

When the user is ready to start, the timer is started. The timer sends the sound to the exercise equipment at the start of each leg. Each leg is processed in order until completion. The stop timer is ran automatically after the last leg is completed or manually by the user.

The exercise equipment is not programmed by this project. The exercise equipment already has the iFit® technology. This technology monitors for sound, processes the sound, and then sets the equipment. The sound generated by this piece is compatible with the technology to create the simulation. The description of the sound expected by iFit® technology exercise equipment is described in appendix A.

The user is able to quit through the quit operation of the race simulation program.

3.3.2.4 Website

The website controls the commerce and communications for the project. The goal of the website is to reach the vast number of users on the Internet. This gives Internet users the ability to purchase and receive the race simulation and the race simulation program as well as a way to communicate with the internal staff about problems or suggestions. The following diagram shows this interaction:

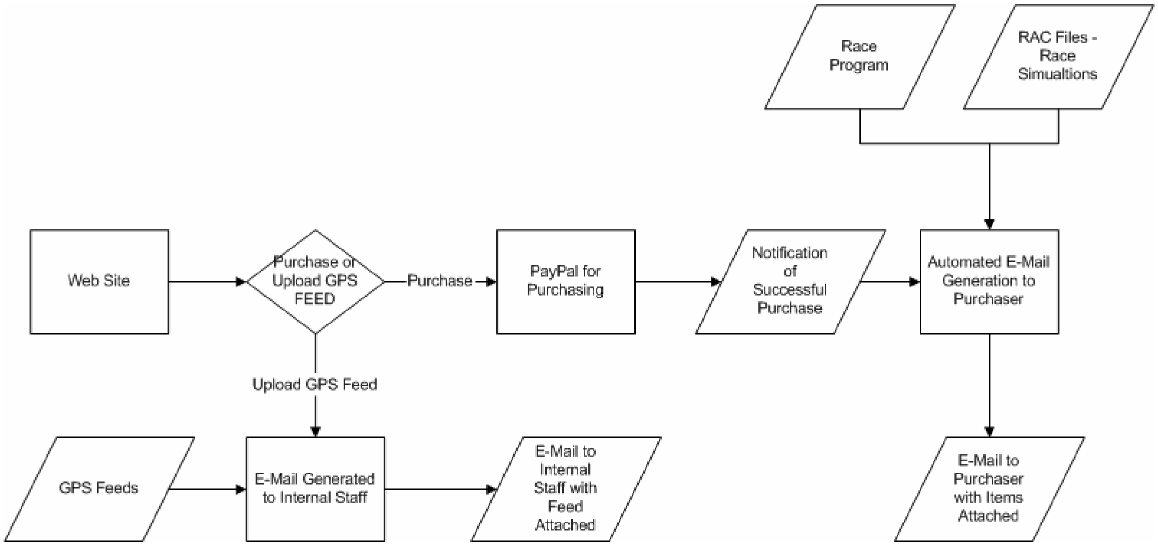

Race Program
RAC Files - Race Simualtions
Web Site
Purchase or Upload GPS FEED
Purchase
PayPal for Purchasing
Notification of Successful Purchase
Automated E-Mail Generation to Purchaser
Upload GPS Feed
GPS Feeds
E-Mail Generated to Internal Staff
E-Mail to Internal Staff with Feed Attached
E-Mail to Purchaser with Items Attached

Figure 18: Diagram of Web Site

The diagram shows a GPS feed sent through the website and e-mailed to the internal staff. Once the internal staff receives it, they are able to use it within the race simulation setup program. Once processed through the race simulation setup program, the output for the RAC file is able to be added to the website for purchase by the Internet users. The submitter can be notified though e-mail by the internal staff when the race has been added to the website.

The diagram also shows the process to purchase the race simulation program and the race simulations. Once the users receive the program and simulations from a successful purchase, they can then run the race simulation program on their devices.

The automated e-mail generation process monitoring successful purchases resides outside of the website. It runs on a PC by the internal staff and receives notifications via the Internet on successful purchases on PayPay®. By keeping it internal, the e-mail generation is closely monitored for quality, and space is not limited as it is when stored externally.

3.3.2.4.1

Website Class Diagram

To develop this piece, the following class diagram is being utilized:

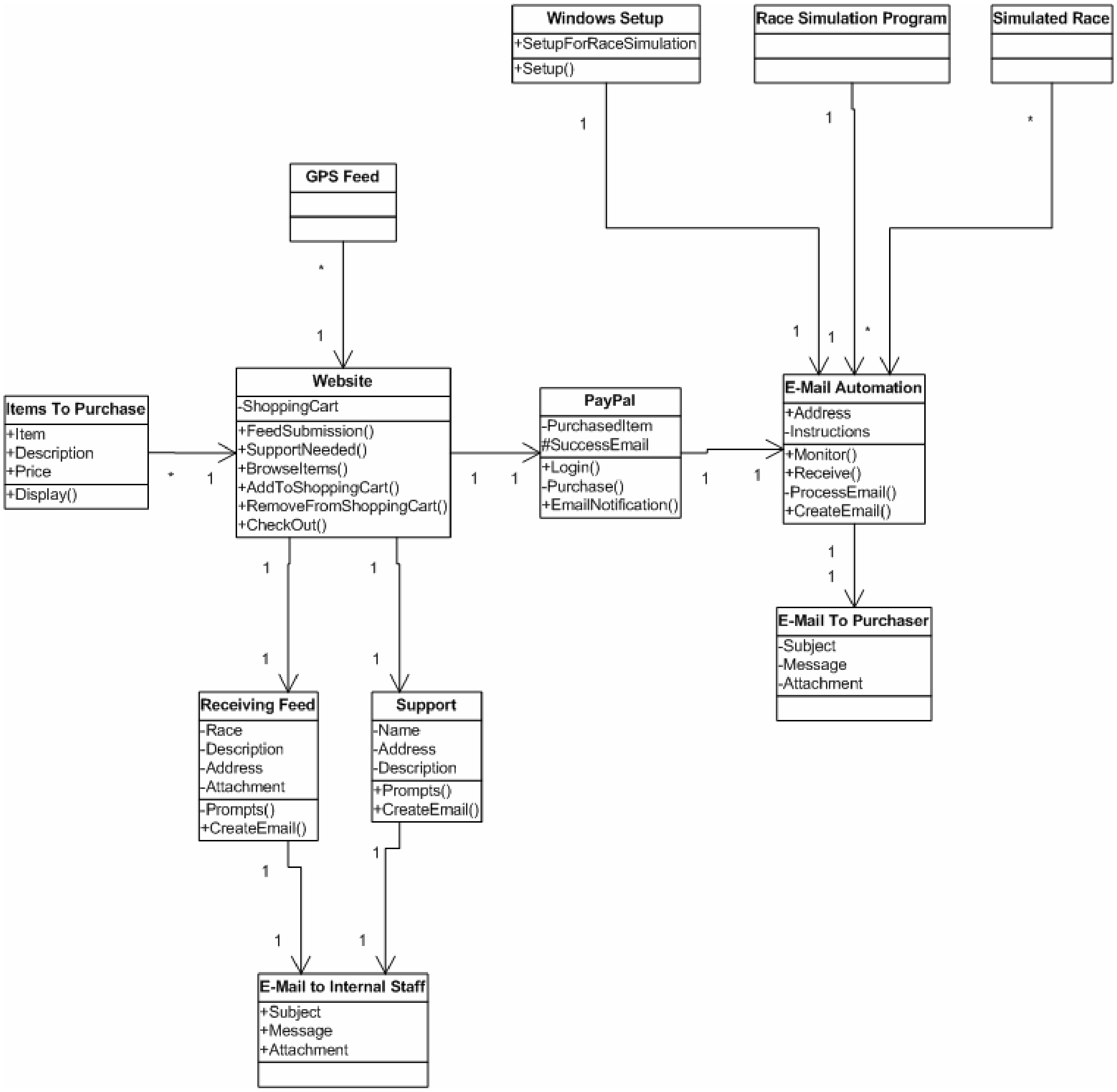


Figure 19: Class Diagram of Website

The website class is the key to this piece. Three pieces are occurring within this class for the users: submitting feeds, support, and electronic commerce.

When a user has a feed submission, the GPS feed comes into the website. The receiving feed class is used to prompt the user for the name of the race, a description, and their e-mail address. The GPS feed attaches to the e-mail when created and sent to the internal staff.

For support, the user identifies they are in need of help. Their name, e-mail address, and problem description is prompted for. The support class generates an e-mail to the internal staff of their issue.

For electronic commerce, the available items to purchase can be browsed. Each item has a description and price associated with it and displayed on the site. A shopping cart allows the user to add items to their cart for later purchase, remove items, and when done shopping, check out.

The checkout process utilizes PayPal® utilities. PayPal® is an external piece for secure credit card transactions. It allows the user to purchase items after they login. Once successfully purchased, an e-mail is generated to notify the internal staff.

E-mail automation is a class developed by the internal staff. It monitors for the successful e-mail, receive it, and process it. During the processing of the e-mail, the e-mail address of the purchaser and their purchased items are retrieved. When creating the e-mail, the e-mail to the purchaser has instructions on how to install the purchased items as well as the items as attachments. The items may possibly be simulated races, the race simulation program, or a Windows setup of the race simulation program. The installation process is to be done by the user when they receive the e-mail.

3.4 Phase III – Construction Phase

Phase III of the project focused on the construction. This was the actual building of the systems within the project. This included the programming of the setup program, race simulation program, and the website including the interfacing to the PayPal® technology and the program to monitor successful purchases. The design documented how these elements were to be done to ensure interaction between the differing phases.

The next section is a milestone report giving dates and statuses of the different pieces being constructed. It breaks the construction into the three systems. Each system is further divided to encompass important milestones, which they must produce.

By meeting the milestones at the scheduled intervals, the project was judged on timing to ensure it is not slipping behind schedule. If the milestones were not met, the project was reevaluated to set a new deadline.

3.4.1 Milestone Report

Prepared by: Rod Dolen **Date:** 04/15/2006

Milestone	Date	Status	Responsible	Issues/Comments
Race Simulation Setup				
Sample GPS Feed	01/20/2006	Completed	Initiator	GPS sample from the Passavant 5K Race
Program Setup Parameters for GPS Input	02/03/2006	Completed	Developer	All parameters including defaults added
Create Race Simulation File	02/17/2006	Completed	Developer	Encrypted race simulation file was created
Race Simulation Program				
Program Setup Parameters for Exercise Equipment	02/24/2006	Completed	Developer	All parameters including defaults added
Selecting & Reading Race Simulation File	03/03/2006	Completed	Developer	Race simulation file chosen and read in
Creating Sound	03/27/2006	Completed	Developer	Sound has been created per appendix A
Sound Output	04/13/2006	Completed	Developer	Sound sent to the audio port and tested on exercise equipment
Website				
Web Interface	04/27/2006		Initiator/ Developer	
GPS Feed Processing	05/01/2006		Developer	
PayPal® Interface	05/15/2006		Initiator/ Developer	
Automated Program Monitoring Purchases	05/31/2006		Initiator/ Developer	

Figure 20: Milestone Report

3.5 Phase IV – Testing Phase

Phase IV of the project is the testing. This encompassed testing the constructed programs and various elements related to them to ensure they worked together to meet the requirements.

The testing phase coincided with the last portion of the construction phase. This overlap allowed for planned programming changes found from errors during testing.

3.6 Phase V – Implementation Phase

The implementation phase is phase V. In this phase, the actual project was implemented including the setting up of the web site and possible transfers of the race program and the races currently available.

When implementing the website, the site name needed to be registered as well as a third party Internet service provider selected. The goal was to allow the user to enter in a web address for the software and race simulations that are easily identifiable and remembered. Another goal was to choose a reliable service provider which supported a Java website being proposed by this project.

3.7 Phase VI – Maintenance Phase

The phase VI is Maintenance. The maintenance phase was a post-implementation phase where the focus was on documenting and training each staff member on the necessary maintenance steps. With the current staff being two individuals, cross training of the individuals was needed to cover any issues with vacation or scheduling problems.

If any problems are found within the project, the problems were evaluated and were corrected by setting up a new project and going through the SDLC process.

Chapter Four: **Conclusion**

The main goal of this project is to fill a void in exercising. This void in training for races hampers the individual from fighting obesity. With busy lifestyles, potential running and bicycling enthusiasts feel they cannot compete in races due to their work schedules and family commitments interfering in their training regimen. Instead of training, the individual gives up in frustration on trying to fit it into their already hectic lives. This frustration leads to a sedentary lifestyle, which is giving rise to obesity.

Another goal shown from this project also deals with promoting exercise. This promotion is for the exercise clubs across America being able to have their sanctioned races promoted. A wide audience of exercise enthusiasts can access a group's simulated race from anywhere on the Internet. The enthusiast can then train for the exact course and run the course without seeing it beforehand. An example would be running a marathon in Alaska by an individual in Illinois. The individual can run the simulation to prepare for the changes in incline and distances. The exercise clubs will have a wider audience for their race by promoting to the Internet users searching for a challenging race.

Modern technology has allowed us to meet these goals using GPS devices, the Internet, and iFit® technology. Training can now be done for an exact race's inclines and length in an individual's home on a treadmill or exercise bike. Using Java as the programming language for the race simulation program, a Java compatible device only needs to have an audio port with a cable connecting from it to the iFit® compatible exercise equipment.

The project being described allows for potential growth for future releases. It has been kept to the fundamentals of processing GPS feeds, producing the simulation, running the simulation, and purchasing the simulations and race simulation program to achieve the basic needs of the project. In future versions, improved screen interactions such as a video of a computer generated race changing with the changes in the race, the ability to start at differing points along the race course especially for marathons, ability to pause the race, or change speeds during the race can be added to service the growing needs of the users.

The success of this project will rest upon the market accepting it. By keeping the costs extremely low, the return on investment will be met quickly after the site is launched. A quick return will lead to improvements such as the potential to buy installation wizards for other platforms besides Microsoft Windows, the staff traveling to additional race locations for GPS readings, and a reduction in purchasing costs to the end users. As the profits increase, the potential for success rises from the additional enhancements that are made which will meet the goals of promoting exercise for the individual and within the numerous running clubs across America.

Chapter Five: **Lessons Learned**

The lessons learned section gives a reflection of the successes and failures related to the project. The main idea is for future projects not to repeat the failures from a previous project. By reviewing this section, project managers will follow the successful lessons and try to avoid the failures. The following will identify these lessons.

5.1 Did the project meet scope, time, and cost goals?

The initiator of the project originally brought this idea to the project developer in March of 2005 with the hopes of having it released during the summer of 2005. The scope and the costs never changed on the project, but the time goal was not met. Problems arose from issues with the developer's free time as he finishes his educational goals at Regis University.

After taking Regis University's "Information Technology Project Management" course, the developer decided to use this project as his professional project for graduation and create a new deadline. By doing the project plan, the developer was able to get the project moving forward with the goal of releasing in the summer of 2006.

The project, after the plan was created, is being constructed with a major emphasis on eliminating scope creep, which occurred in the first attempt without a formal plan. This elimination will help meet the time and cost requirements. The scope is being limited to the basic needs of simulating races, and this document is being continuously reviewed to ensure the development follows the guidelines set in it. If anyone requests changes to the project, this document will be modified with both of them reviewing the changes to ensure it does not adversely affect the costs and time requirements. If it does adversely affect the requirements, the change may be eliminated.

The tools for this project are either free or already purchased. Java was selected for this point of being free for development plus meeting the objective of being widely supported on differing operating system besides Microsoft Windows. The developer had also taken “Java Programming” and “Advanced Java Programming” from Regis University through his coursework, so the free tools would not cause time delays from having a learning curve by the developer. Tools like InstallMaster for Microsoft Windows installs or the development PC’s were already purchased and available to be used by this project.

5.2 What are the success criteria within the project?

The project success is the creation of the simulation of the race. The project will need to be able to simulate the race while protecting the integrity of the technology. This protection includes the ability of users to send GPS feeds to the staff to have the staff ensure the feed meets the standards set for quality.

The project is also successful by choosing Java as the development technology. The widespread support of the Java programming language increases the devices to run the race simulation on.

The success criteria will be initially measured in the initiator and the developer’s respective environments. The initiator utilizes an iFit® compatible treadmill with access to a laptop and handheld PC. The developer utilizes an iFit® compatible exercise bike with access to a laptop. The initiator will only have the runtime version of the program, but the developer’s laptop will be used for modifying the code when controlling the exercise bike. Once the developer is successful, he will create a runtime version similar to what will be released and sent to the initiator for his testing.

5.3 Were the success criteria met?

With the project still in development, the main goal of the success criteria is being met. A GPS feed has been simulated and the sound created from the simulation has controlled both an iFit® compatible treadmill and exercise bike. The Java programming language is providing a strong platform for this development.

The original work before the project plan did not meet the success criteria. The initial failure occurred during the transmission of the sound properly through Java to the audio port of the device. This failure was mainly due to the project not being planned properly.

Without the plan, everything was occurring simultaneously. Ongoing research was being done, while the problems with the sound were being investigated. The plan broke down the project into a system development lifecycle, or SDLC. This placed the analysis completed first, then the design, and when construction occurred, the focus could be on the development instead of trying to do additional research that should have been done during the design phase.

5.4 What are the important lessons from managing the project?

When managing this project with such a small staff, it is beneficial to focus on roles. The project manager is also the main developer. It is important to focus on one role and try to minimize the other. An example is creating this project document without stopping to try and develop the programs being reviewed. By focusing on one role at a time, the project is better served.

This idea of focusing on roles is described in why the success criteria was not met with the original development previously discussed. By the developer designing and developing at the same time, failure occurred on the project. By using the knowledge from his project management courses, the developer was able to separate these roles and produce a program to control his exercise bike. The lack of success without the dividing of the roles proved the importance of proper project management especially with an SDLC.

5.5 An example of what went right on this project.

The research part of this project is an example of what went right. The research is thorough and includes the most important piece of how to generate the sound. By having this research available at the time of development, the developer proceeds without having to do major research during development.

An example of the importance of this research would be with apendix A. Appendix A shows the sound format used with iFit® technology. The original process without the project plan did not have this piece researched. The ideas for how to control the device happened to be present, but minor issues arose that stopped development.

These minor issues were overcome when the research showed the exact layout of the sound and what each variable meant during the coding. Once the research was completed, the developer reviewed the code to understand why the sound problem was occurring. Once he understood what the exercise equipment expected, he was able to make minor changes within the program using appendix A's information.

5.6 An example of what went wrong on this project.

The time constraints on the individuals involved has proved to be a major hurdle. Without the plan, the individuals were doing everything quickly without fully researching the project. By taking time to properly develop the project, the time constraints were eased. The individuals knew what needed to be done and by what milestone. Even though they still had limited time, the project plan improved the free time they had.

5.7 What will be done differently on the next project?

Project management must be done unto completion for any project to be successful. Individuals were willing to skip pieces to get started on the development. This proved harmful to the project.

The next project will have an understanding of the individuals involved to have all the project documentation completed prior to the project with needed updates during the remaining pieces. The updates include meeting milestones, enhancing the lessons learned portion, or updating changes to the project found to be either inadequately designed or no longer beneficial to the project.

Both the initiator and the developer are making a commitment to future project management. Since the initial inception of this project in March 2005, the initiator fulfilled the requirements to receive Project Management Institutes' designation of Project Management Professional. The developer has been approved to test for Project Management Institutes' designation of Certified Associate in Project Management and plans on taking his exam once his degree requirements are completed at Regis University.

This desire to improve on their project management skills can be directly linked to the lessons proven within this project. Both realized the improvement within the project when it was done properly. Their initial desire for a speedy product taught a valuable lesson of project management and its benefits.

Appendix A: **Sound Format For iFit® Technology**

This section describes the sound manipulation used to control exercise equipment using iFit® technology. The source for this code was a free downloadable program found by the project initiator when researching how sound was manipulated and sent to the equipment. The program was written in Microsoft Visual Basic and was tested to ensure it was able to control the exercise equipment properly. The project developer reviewed the downloaded program and was able to generate the same sound patterns through the Java programming language.

The sound transmitted between the Java program through the audio output of the device and the audio input of the exercise equipment was in a wave format. Wave data is divided into three pieces called RIFF, FORMAT, and DATA (Sapp).

For creating the sound data within the race program, most of the values are hard coded. Calculations are done during the DATA piece to set the speed and incline, which are the only changes needing to be done. The following sections describe each piece and the values that are used to create the sound data.

A.1 RIFF Piece

The RIFF piece is the first twelve bytes of the sound data. The following table displays the settings for the RIFF piece:

Length in Bytes	Description	Hex Values
4	Chunk ID identified by characters "RIFF"	52 49 46 46
4	Chunk size (size of data not including first 8 bytes)	76 47 00 00
4	Format identified by characters "WAVE"	57 41 56 45

Figure 21: RIFF Piece

A.2 FORMAT Piece

The next twenty-four bytes make up the FORMAT piece. The following table displays the settings used in creating the sound:

Length in Bytes	**Description**	**Hex Values**
4	Character values of “fmt “ for FORMAT piece	66 6D 74 20
4	Size of FORMAT piece (value of 16 for PCM)	10 00 00 00
2	Audio format for PCM (value of 1)	01 00
2	Channel numbers for Mono (value of 1)	01 00
4	Sample rate of 44100 Hz	44 AC 00 00
4	Byte rate	88 58 01 00
2	Block align	02 00
2	Bytes per sample for 8 bits	10 00

Figure 22: FORMAT Piece

A.3 DATA Piece

Most of the DATA piece is hard coded, but there are three calculations. The calculations are for speed, incline, and a check sum value. The check value is the combination of speed and incline. The following table displays the DATA piece:

Length in Bytes	Description	Hex Values
4	Character pieces of “data” for DATA piece	64 61 74 61
4	Size of DATA piece	52 46 00 00
2824	Repeating null values	00
1760	Speed calculations	
1760	Incline calculations	
1760	Check sum calculations	
2108	Repeating null values	00
1760	Speed calculations	
1760	Incline calculations	
1760	Check sum calculations	
2508	Repeating null values	00
2	Carriage return and line feed	0D 0A

Figure 23: DATA Piece

A.3.1 Calculations for iFit® Technology

The iFit® technology uses two different phases to produce the controlling sounds. The phases are then transmitted in sequences to control the speed and inclines. The first phase is 176 characters of nulls, or hex 00 values. The second phase also has 176 characters and is calculated.

A.3.1.1 Phase Calculation

The calculated second phase is programmed by the following steps:

1. Initialize phase to nothing
2. Set value1 to 32768
3. Set value2 to .28495
4. Initialize loop value to 1
5. Multiply value2 to the loop value and then take the sine of the returned value
6. Multiply the calculated value by value1.
7. If the calculated value is less than 0, add 65536 to it
8. Round the calculated value and save it to value2.
9. Value2 is converted to a hex format and is padded with leading zeroes until 4 characters long.
10. The last two characters are saved to the phase and then the first two characters.
11. Increment the loop value by one.
12. If the loop value is less than 89, got to step 5.
13. The calculated phase is now in a hex format.

This step is the actual calculation of the sine wave used for the sound. As the speed and incline are sent to the exercise equipment, patterns of this sine wave are used.

A.3.1.2 Pattern Calculation

The pattern for the wave transmission is based upon a sequence of zeroes and ones. The zeroes in the pattern transmit the null phase and the ones transmit the calculated phase. The pattern is generated by the following calculation routine:

1. An integer value is passed to the routine.
2. If the passed value is zero, the pattern is “0100000000“and skips to step six.
3. The pattern is initialized to “01”.
4. The passed value is divided by two with the remainder added to the end of the pattern.
5. If the new passed value (which was divided by two in step 4) is not equal to one, go to step four.
6. Add “1” to the end of the pattern.
7. If the pattern is greater than 10, the pattern is set to only the first 10 characters.
8. If the pattern is less than 10, add “0” to the end of the pattern.

These steps are used by the program to generate the proper controlling sound pattern for the exercise equipment.

A.3.2 Speed Calculations

The desired speed value is multiplied by 10. The speed is then passed into the pattern calculation. The pattern is then calculated and then added to the DATA piece in the "Speed calculations" area.

A.3.3 Incline Calculations

The incline calculations start with the incline being multiplied by 10. The incline is then passed into the pattern calculation. The pattern is then calculated and then added to the DATA piece in the "Incline calculations" area.

A.3.4 Check Sum Calculations

The check sum calculations start with the incline and speed both being multiplied by 10. The multiplied values are then added together. The check sum is then passed into the pattern calculation. The pattern is then calculated and then added to the DATA piece in the "Check sum calculations" area.

A.3.5 Stop Calculations

The last transmission sent by the program will be to stop the treadmill once the race has been completed. To stop the treadmill, the settings for speed and incline are set to 252. The patterns are then generated the normal way except for the values are not multiplied by 10 in the speed, incline, and check sum calculations.

References

"Boston Marathon." Wikipedia. 18 Mar. 2006. 19 Mar. 2006. <http://en.wikipedia.org/wiki/Boston_Marathon>

Centers for Disease Control and Prevention "Overweight and Obesity: Contributing Factors." Department of Health and Human Services. 2005. 13 Dec 2005 <http://www.cdc.gov/nccdphp/dnpa/obesity/contributing_factors.htm>

"Chicago Marathon." Wikipedia. 12 Mar. 2006. 19 Mar. 2006. <http://en.wikipedia.org/wiki/Chicago_Marathon>

ICON Health & Fitness. "About ICON." ICON Health & Fitness. 2004. 17 Nov 2005 <http://www.iconfitness.com/>

PayPal. "Compare Features." PayPal. 2005. 3 Jan. 2006 <https://www.paypal.com/cgi-bin/webscr?cmd=_feature-and-pricing-comparison>

PayPal. "Compare Solutions." PayPal. 2005. 3 Jan. 2006 <https://www.paypal.com/cgi-bin/webscr?cmd=_profile-comparison>

Running Clubs of America. "Road Runners Club of America." Running Clubs of America. 2005. 15 Dec 2005 <http://www.rrca.org/>

Running Clubs of America. "Springfield Road Runners Club." Running Clubs of America. 2005. 15 Dec 2005 <http://www.rrca.org/members/showclub.php?id=1666>

Sapp, Craig. "Wave PCM Soundfile Format." Stanford's Center for Computer Research in Music and Acoustics. 2005. 16 Jan 2006 <http://www-ccrma.stanford.edu/ccrma/courses/422/projects/waveformat/>

Springfield Road Runners. "2005 Race Totals." Springfield Road Runners. 2005. 17 Nov 2005 <http://www.srrc.net/default.asp?id=6&mnu=6>

"Working Time." Wikipedia. 2005. 15 Dec 2005 <http://en.wikipedia.org/wiki/Workweek>