Universidade Federal do ABC

UNIVERSIDADE FEDERAL DO ABC
CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS APLICADAS

Topic: Design of an Embedded Statistical-use Taekwondo Body Protector

COORDINATED BY
Prf Dr. Marcelo Bender Perotoni

Douglas da Silva Mangini
Guilherme Paranhos de Oliveira

SANTO ANDRE, SAO PAULO

BRAZIL, 2015

Topic: Design of an Embedded Statistical-use Taekwondo Body Protector

This work has been delivered to the Universidade Federal do ABC as evaluative method for the unit Trabalho de Graduação I em Engenharia de Informação (EN1603)
COORDINATED BY
Prf Dr. Marcelo Bender Perotoni

Douglas da Silva Mangini
Guilherme Paranhos de Oliveira

KEYWORDS: ELECTRONIC TAEKWONDO BODY PROTECTOR, ARDUINO, STRAIN GAUGE, STATISTICAL APPROACH FOR SPORTS.

SANTO ANDRE, SAO PAULO
BRAZIL, 2015

ABSTRACT

Statistical research is of paramount importance for improved performance of professional competitors in a wide range of sports. This approach can be used in such way that may make the difference in competitions which require athletes who perfectly know their own strengths and weaknesses. This study suggests a new electronic approach for statistical analysis in Taekwondo by the use of electronic body protectors. The study starts from the historical arising of Taekwondo as a Korean resistance support against the Japanese invasions back in the 18th century. After that, the study focuses in the spreading of taekwondo worldwide as a sport and its ascension until being part of the Olympic Games at Sydney in 2000. Following these events, the international Taekwondo confederation sought technological solutions to push for a larger audience interaction by creating the electronic body protector. Moreover, starting from this idea, this study suggests protector adaptations that enable it to acquire data from the regions where the points where scored and measure their respective impact force. To perform this action, the strength measurement gotten from Strain Gauge sensors connected to an Arduino board are serially sent in real time to be later analyzed in a computer. The final aim of this project is to demonstrate the practical results of the statistical approach for taekwondo. To perform this achievement, this study is going to perform tests in practical experiments with real athletes and verify their opinions and suggestions to improve the data processing and display methods.

ACKNOWLEDGEMENTS

This study is going to be used as Final Undergrad Project for the Information Engineering Bachelor in the Universidade Federal do ABC. We consider that it is fundamental for professionals with academic degrees to be able to apply and transfer the theoretical background learnt in College to develop technologies and solve practical issues. Because of this, we are glad to show our Engineering skills to perform this task aiming the future of the sport which we both practice.

Never forget that, at the most, the teacher can give you fifty percent of the art. The rest you have to get for yourself through practice and hard work. I can show you the path but I cannot walk it for you.

Master Rudolf Kang –
International Taekwondo Federation

DEDICATED TO:

We would like to dedicate this work to our families and professors who supported us in this difficult step of our development as students to become qualified professionals and part of the technological core in our society.

LIST OF CONTENTS

INTRODUCTION

LIST OF FIGURES

LIST OF TABLES

1.0 – Project background

1.1 – Taekwondo history and evolution

Taekwondo is a millenary martial art primary based on kicking. It was originated in Asia, more precisely at the Korean region, and strongly spread all over the world during the last decades, due to its ascension as an Olympic sport on the early 2000s. Although it has become popular and many scholars are dedicated to Taekwondo, there still a huge field for historical, medical and technological research.

According to historical accounts presented by the World Taekwondo Federation (WTF), the highest entity of this martial art responsible for maintaining Taekwondo history and regulating its practice worldwide, Taekwondo originated two thousand years ago, descending from the earlier martial arts of Subak and Taekkyon. During that time, the late Choson Dynasty, martial arts in Korea had a higher emphasis on "literary art" over "martial art", which drove them to a decline. This had been intensified by the Japanese occupation, when all martial arts, including Taekkyon were forbidden by authorities and therefore, it was usually practiced in secret. Subak and Taekkyon decline lasted until Korea’s liberation, when it has re-stablished under a different name: Taekwondo.

During the 60s, the martial art passed through many transformations that led to its dissemination, first in Korea and then the world. In 1967, the Korean Taekwondo Association changes the directions that had been given to this martial art: from discipline and self-defense to a competitive sport. This was a strategy used by other martial arts, such as Karate and Judo, to become more popular.

In fact, the western dissemination also contributed with this transition from martial art to a competitive sport. Once Taekwondo was in touch with western lifestyle, it was surrounded by many influences, such as city violence, sportive training, marketing, media, profit, and so on [1].

In spite of its many changes, Taekwondo has not been turned completely to a sport, something that would imply in traditional Taekwondo extinction. Instead, it is noticeable that its main core (Martial Arts Philosophy) is still preserved and mixed with modern sports components (competition, physical performance, records, training rationalization and scientifically-based). This led Taekwondo to a seemingly evolution.

1.2 – Competitions and Olympic Games

In 1971, the South Korean President claimed Taekwondo as a national sport, and in 1973, the first Taekwondo World Championship took place in Seoul by the Korean Taekwondo Association. The first Taekwondo appearance in the Olympic Games was in 1988, “coincidently” hosted by Seoul, the South Korean Capital. It raised the modern martial art status to an Olympic sport, being so part of a rather restrict group and within a strong commercial framework [1].

Although in the Olympic Games of Seoul, Taekwondo was not classified as a competition style, but as a demonstration. Therefore, no medals were given to its competitors. This happened again on the next edition (The Olympic Games of Barcelona 1992 – Spain) and during the Olympic Games of Atlanta 1994 – USA, the sport was removed, once all the demonstration styles were taken off from the games. Only in 2000, at the Olympic Games of Sydney – Australia, Taekwondo had its status of a full-medal sport, and it has so been in every Olympic Games since then.

1.3 – Technology use in worldwide sports

Over the last two decades, the application of computing power to enhance sports has been intensified by the ascension of embedded technologies. Accelerometers, gyroscopes, microphones, and cameras has been used for performance enhancement, rehabilitation and injury prevention.

One of the sports that more intensely use technology weapons is fencing. The first innovation was the creation of a metal vest that, when touched by the fencing sword, connected to an electrical source, transmits a signal indicating it has been precisely touched (indicating a score point to its competitors).

In tennis, using a completely different technology approach, cameras are strategically positioned all over the court to keep track of the ball motion. The cameras, positioned in different angles, capture images in two dimensions which are sent to a central computer. There, the 2D pictures are processed by triangulation and turned into a 3D video, reproducing the ball trajectory with a precision that enables referees to analyze where the ball precisely touched the ground, validating a score point or not. This system has brought fairness and precision to the sport, enhancing tennis credibility and value [2].

Soccer is also enhancing its performance by the use the technology on players, supporters and everyone involved on its chain. Lately, the most used advance is a communication relay system that allows different referees to work with a better synergy. Furthermore, the international soccer federation (FIFA) is studying the full implementation of a system capable of telling the main referee when the boll crosses the goal line, indicating a correct goal. Some technologies are on the waiting line, ready to be picked on: one of them is a camera system similar to the one applied in tennis matches, and the second one is a sensor embedded inside the match ball. Both technologies basically solve the same problem, however FIFA is more inclined to adopt the second one, because it can indicate a goal to the referee in real time [2].

1.4 – Statistical study in worldwide sports

In some sports, coaches and athletes use biomechanics and statistics for improving performance. There is a German software tool, which is capable of comparing athletes data, obtained by periodic fitness evaluations. There is also another one called Dual Beam Absorptiometry, that computes the body composition (fat, muscles, water, bone density, etc). This kind of analysis is essential for preventing injuries [3].

In swimming, the biomechanical analysis is made by submerse cameras disposed all along the competition pool. Therefore, coaches can evaluate the minimum details of diving, arms and legs movements. The best athletes all over the world use this kind of technology for improving their results [3].

Basketball also uses statistical analysis for helping coaches. The Dart Fish software tool captures images of individual players, team formation and movements during a game, allowing a precise posterior analysis. This software allows the team personnel to design a better strategy for each adversary based on previous games. It also helps to evaluate the team performance and obedience to the previously designed strategy in such a way that the coach can fix eventual game mistakes and errors in real time and also analyze the full game statistics for addressing main identified issues to the next match. This software provides a full information repository that supports the coach [3].

The same trend towards the use of technology is followed in martial arts. Judo is widely using a software called Frami, which is developed for watching a fight, creating and saving movements and scores in a database, allowing coaches to evaluate a fight and decide which tactics should be used and also how to improve the weakest movements of a wrestler [3].

All in all, every sport can benefit from the existing technology. From popular sports such as golf, baseball, and basketball to extreme sports such as snowboarding, motocross racing and rock climbing, all athletes can gain from a better understanding of their muscle movements, orientation and heart rate response.

Achieving better performances with technology support requires the choice of the most appropriate sensors and their respective correct use. Being appropriate means enhancing players performances so that they want to use sensors in their training (i.e. comfortable experience), enabling unobtrusive instrumentation so that coaches can analyze the available data and cooperating with judges so that they make good, socially acceptable calls on the field [2].

1.5 – Technology development for Taekwondo use

The first technological innovation in taekwondo was a computational system for calculating and registering score points and warnings accumulated during a fight. This technology has a very simple concept and it was implemented only for replacing the traditional paper based scoring system. It was based on three joysticks disposed on the corners of the dojang (competition field) containing two buttons (red and blue ones). Once a referee sees a precise hit, he presses the button according to the color of its competitor. If two of the three referees press the button, the point is scored and shown on the main computer screen. Although it was an advance at the time, the human factor evaluating the hit accuracy and consequently a point was still remaining.

For preventing human inaccuracies interfering on the results of a fight, the World Taekwondo Federation invested on an automated decision making technology that would immediately recognize a precise hit during a fight and compute the score for its competitors. This technology is called “electronic protectors” and was used for the first time during the Olympic Games of London – 2012. It received this name because the electronic devices are embedded on the anatomic protectors currently used by fighters during official competitions. There are four of them, namely helmet, chest protector, hand and feet gloves. Anytime one of the gloves precisely hits the chest protector (punching and kicking) or a head protector (just kicking) a point is scored to the respective competitor [4].

This technology enhanced the fight result~~s~~ accuracy and has been implemented worldwide. Also, because of taekwondo speed and complexity, much research is being held for further improving its scoring system precision and reliability.

1.6 – Statistical study for Taekwondo use

As shown above, taekwondo is using technology mainly for improving its accuracy related to competition results. It is noticed by the huge investment sports articles companies and taekwondo federations all over the world are making on the research for more precise and faster point validation systems. Even though, there is still a huge gap left for technological enhancements on the practice itself.

Just like any human being movement, Taekwondo moves have some special characteristics that are essential for a better understanding of its mechanics. However, some factors, such as speed, can make an analysis hard to be done, once the levels of precision should be high. Therefore, it is necessary an optimized evaluation of athlete biomechanics during training periods so that its performance is enhanced and injuries are prevented [5].

The main objective of this project is to develop a training system that can accurately register training fight information and provide statistics of an athlete performance such as a movement map, total and average number of hits, total and average precise hits during a fight and so on.

Figure 1 depicts a block diagram of the main idea, from the viewpoint of the data analysis.

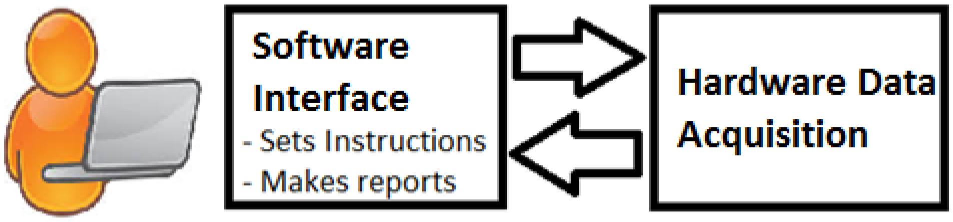


Figure 1. General Design Block Diagram.

2.0 – Hardware and Data Acquisition

2.1 – Arduino development and evolution

The Arduino concept has been developed in 2005 in the Interactive Design Institute of Ivrea, Italy, for academic purposes. The main idea was to build a low cost microcontroller board that allowed even people with no familiarity to microcontrollers create complex projects by linking embedded programming with a simple electronic connective interface [6]. This interface allows the board, for instance, to get data from a sensor, or to show the acquired data in a display [7]. Other important characteristic is the facility to program in its software interface, since Arduino language comes mainly from a C++ basis, common in most part of the Engineering Courses worldwide [8].

The idea of the Arduino project came from the software architect Massimo Banzi that as professor was disappointed with the quality and prices of similar products available in market. Moreover, the two aspects from those attributed to Banzi and his associates that utterly impacted the difference between his and other projects were the open-source characteristics of both software and hardware [9]. The first concept had intention to share the existing knowledge to create an attractive and increasingly complex network of projects, since a project that is shared may give tools to others to perform part of its functions in advanced design. In this way, this concept allows new users the design of impressive projects by reusing existing projects [9]. The second requirement was imposed by low budget circumstances and came from the idea that not only the software should be made open source, but also the hardware. He justifies this claiming that hardware is part of culture that must be shared with other people. These two concepts in action among the low price created a student friendly environment [9]. The first product unities were distributed for students from his University and their friends and then students from other universities soon get interested in the product and Arduino became famous.

2.2 – Current and future uses for Arduino

Ten years has been already passed since Arduino was first manufactured, so its use can be claimed as already mature and diversified. The future trend is to become even more popular and applied widely as the time goes by. When a new Arduino user needs to create new functions to read and send information to the peripheral electronic components for instance, part of the existing know-how made public in the Internet helps the design process. In view of this, current Arduino projects allow more time exploiting different and innovative characteristics than just creating basic concepts of operation from the scratch - a trend likely to increase in future applications. This prediction can be confirmed by the increasing amount of published articles with technological electronic programming embedded in Arduino. Because of its convenience of use and the progressive continuous tendency in current researches, the Arduino was selected among similar embedded programing boards.

2.3 – Strain Gauge (SG) Technology and use

The first, and probably more critical aspect about the use of embedded electronics for the measurement of fight impacts relates to the choice of the sensor. This remark considers that both fighters – attacker and defendant – do not get injured when the blow is applied. For the specific case of taekwondo, the critical factor just extends to the fighter who is blowing, since the other fighter is sheltered by the body protector. For this purpose, the sensor must be malleable so that both fighter integrity and the equipment operation are taken care of.

Given this requirement the Strain Gauge (SG) Technology was selected. The measured resistance across the SG terminals directly depends on its physical area. External strain forces applied to the sensor result in physical deformation, which lead to small variations on its resistance [10]. From this perspective, the Strain Gauge Sensor is a thin electrical load resistor manufactured in such way that reacts with resistance changes according to the strain force applied on it. An example of Strain Gauge Sensor can be seen in the figure 2.

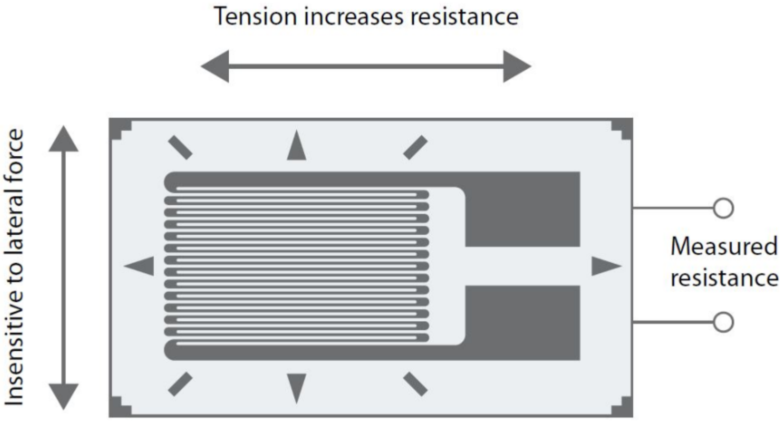


Figure 2. Example of Strain Gauge Sensor, obtained in: ueidaq.files.wordpress.com/2013/08/strain-gauge-figure1.jpg

2.4 – Wheatstone Bridge Applied in Strain Gauge Circuit

As commented in the previous item, strain forces applied to the sensor may cause a change in the resistance measured by an electrical circuit. However, this change has a small variation across the nominal SG resistance, so that a conditioning circuit is needed for further amplification.

A common approach for amplification of these weak signals is by the use of a Wheatstone Bridge. It follows the principle that a circuit with two parallel branches –each one with two resistances – as shown in the figure 3, will result in a zero voltage in the central element as long as the bridge is balanced, i.e. R1*R2 = R3*R4.

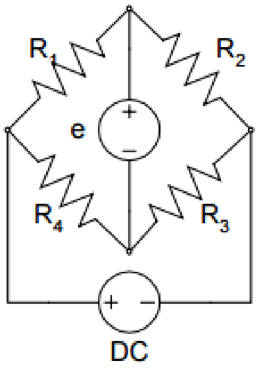


Figure 3. Example of Wheatstone Bridge

In conclusion to this analysis, considering R1 as the Strain Gauge Sensor, R2 and R3 as being equal and with large resistance values and R4 as being a value equal to the R1 resistance without any external strain applied, this simple circuit is able to amplify the smooth voltages caused by changes in the Strain Gauge resistance.

2.5 – Analog-to-Digital Converter

The analog voltage difference relative to the applied strain, from the SG connected in the Wheatstone Bridge output, needs to be converted to a digital string in order to be wirelessly transmitted. To perform this function the HX711 Analog-to-Digital circuit will be used to convert the resultant signal in a 24-bits string [11]. The circuit is designed specially to operate with bridge element sensors applying the input signal in a low noise gain amplifier.

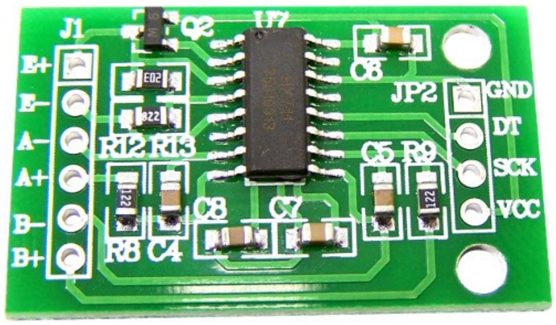


Figure 4. HX711 Analog-to-Digital circuit for Arduino obtained in: http://goo.gl/s680yf.

2.6 – Wireless Transmission Circuit

The last electronic block in this project is a RF communication circuit (figure 5). It will allow the reception of the data in digital form from the body protector, and send those strings to another Arduino connected in a remote computer. The chosen transmission and receptor circuit operate~~s~~ in 433MHz and are commonly used for Arduino communication functions.

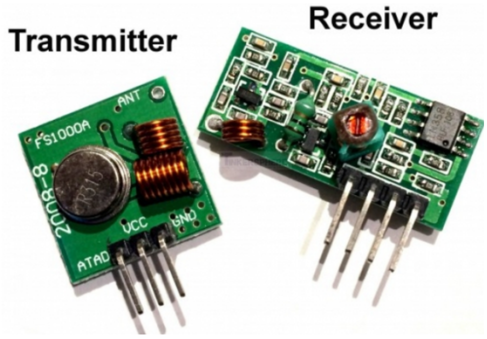


Figure 5. Transmitter and receiver 433MHz for Arduino obtained in http://goo.gl/ZwsNU5.

2.7 – Block Diagram for the Electronic Components

Figure 6 shows the proposed hardware data acquisition block diagram combining all items previously described.

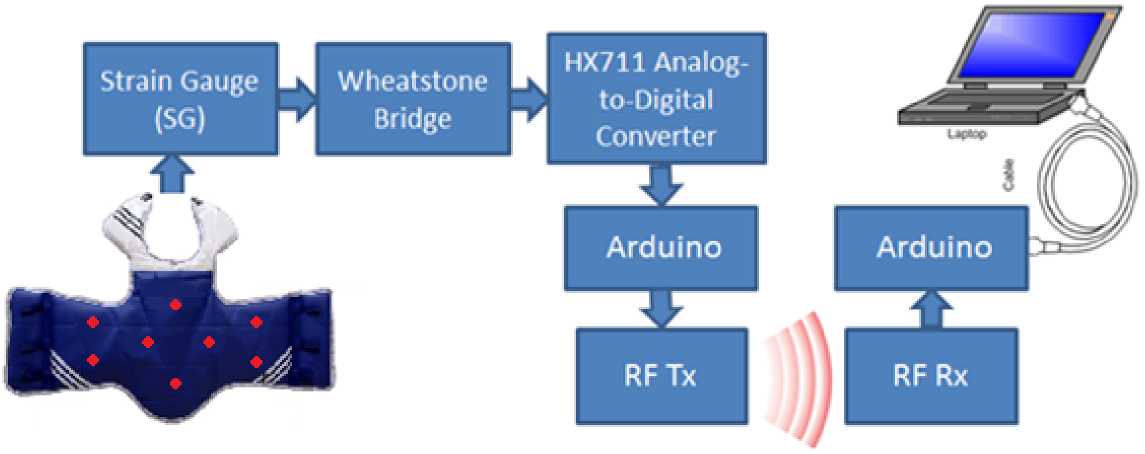


Figure 6. Block diagram for the hardware data acquisition.

2.8 – Strain Gauge Practical Design

The first practical decision taken was the choice of how the SG sensors should be physically attached to the structure. Preliminary ideas consisted in fixing it directly on the body protector surface; however it was turned down because the many wires which could not be hidden without damage the body protector. On the other hand, if the wires are not hidden the design would be fragile in a fight sports environment.

In view of these complications, the chosen alternative consists in the use of a large shirt, which does not harm to the athlete's movement, with the SG sensors inside of it, underneath.

The second practical decision was the choice of how the SG sensors should be attached to the shirt. The main issue in this question was whether the sensor is directly fixed on the tissue; this structure would be fragile as well. Other problem about it is the malleability of the soft tissue – even if an impact occurred in another part of the body protector, or in case of a bump, those could potentially lead to a miss hit measurement. From this perspective, the SG sensors are fixed in an independent structure which is stitched to the tissue, following the previous determined spatial divisions of the body protector.

To produce this design with both low budget and efficiently, PET's bottle caps top were chosen as the mechanical support where the SG are to be attached to.

There are several reasons for this decision, including that its malleability would not harm the athlete in the kick movement moment and would deform/recover in a smooth manner, since its structure has a homogeneous size and it is relatively flat.

In conclusion, the SG sensors are fixed using super glue in the PET's bottle caps top, welding its terminals in both connection wires. The last step drills five holes in this design, one larger in the SG sensor's bottom in order to pass the wire and the others four these to stitch the structure to the tissue. It is also important to glue the wires in both sides of the structure to refuse their movement and consequently make it more robust.

Figure 7 shows an example of implemented SG sensor in the described structure in both sides, in order to display their fixing method and how they are attached to the shirt. Furthermore, the last figure shows their group design in the shirt over the body protector.

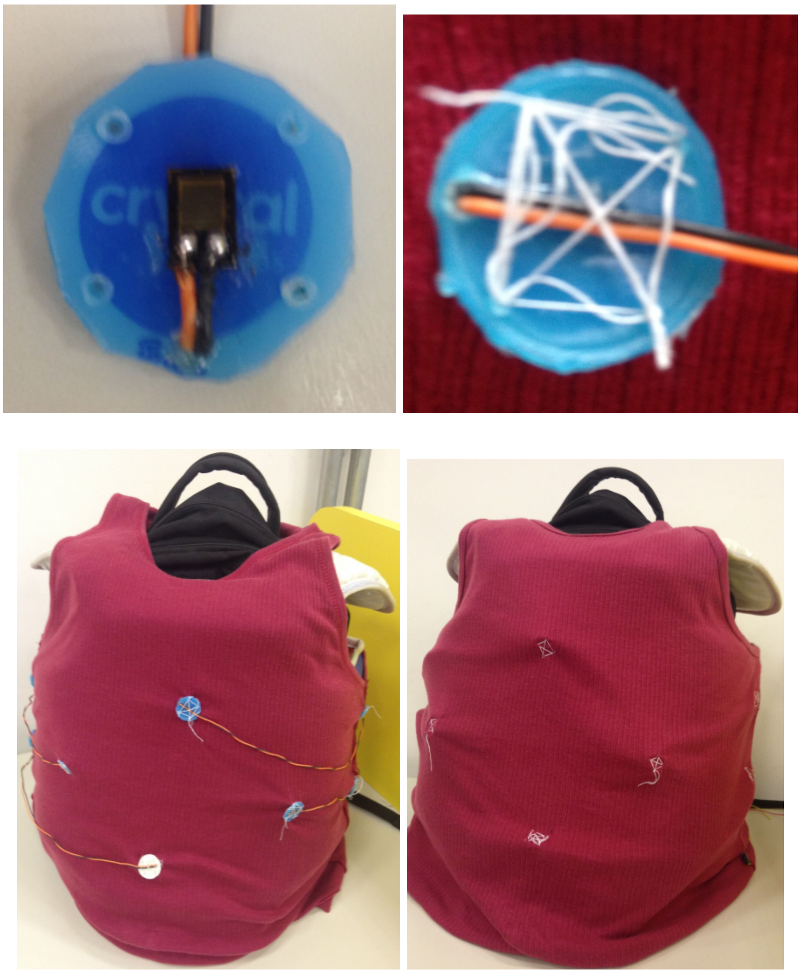


Figure 7. a) Back SG sensor application, b) Front SG sensor application

c) Back group GS sensors design d) Front group GS sensors design.

2.9 – Wheatstone Bridge and Analog-to-Digital Converter

As previously mentioned, the direct acquisition of the SG electrical data from the mechanical blow cannot be implemented directly, due to its small amplitude. A Wheatstone Bridge is then used, to increase the sensitivity of the transducer.

The SG sensor resistance without deformation (no mechanical stress), has a resistance R1 of around 350 Ohms. From this measurement, a Resistor with about the same resistance was placed as R4. To define which value would better suit the R2 and R3 resistances several high order resistances were assembled in the circuit (related resistances presented in Figure 3) in the design shown in figure 8.

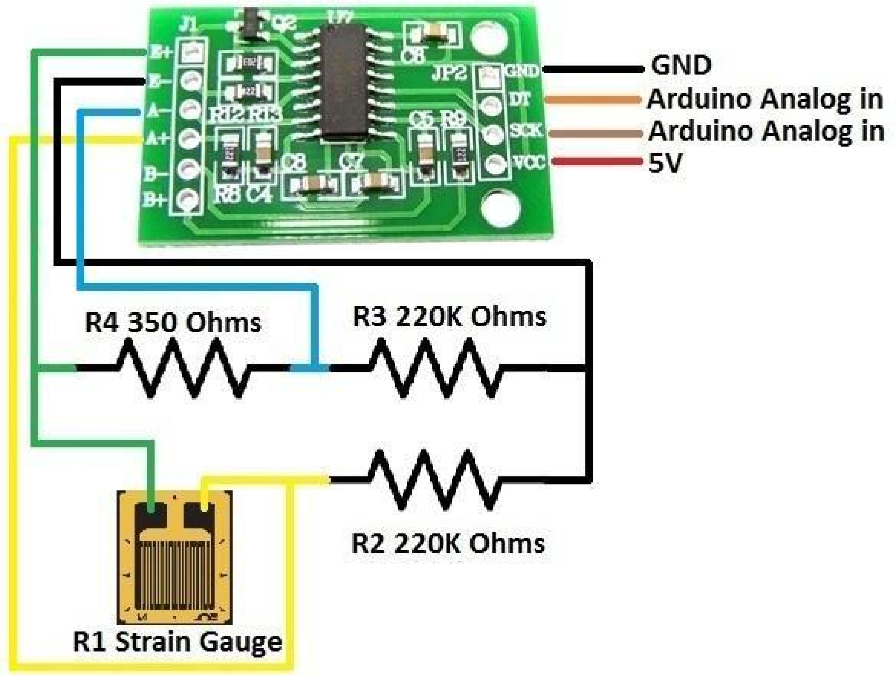
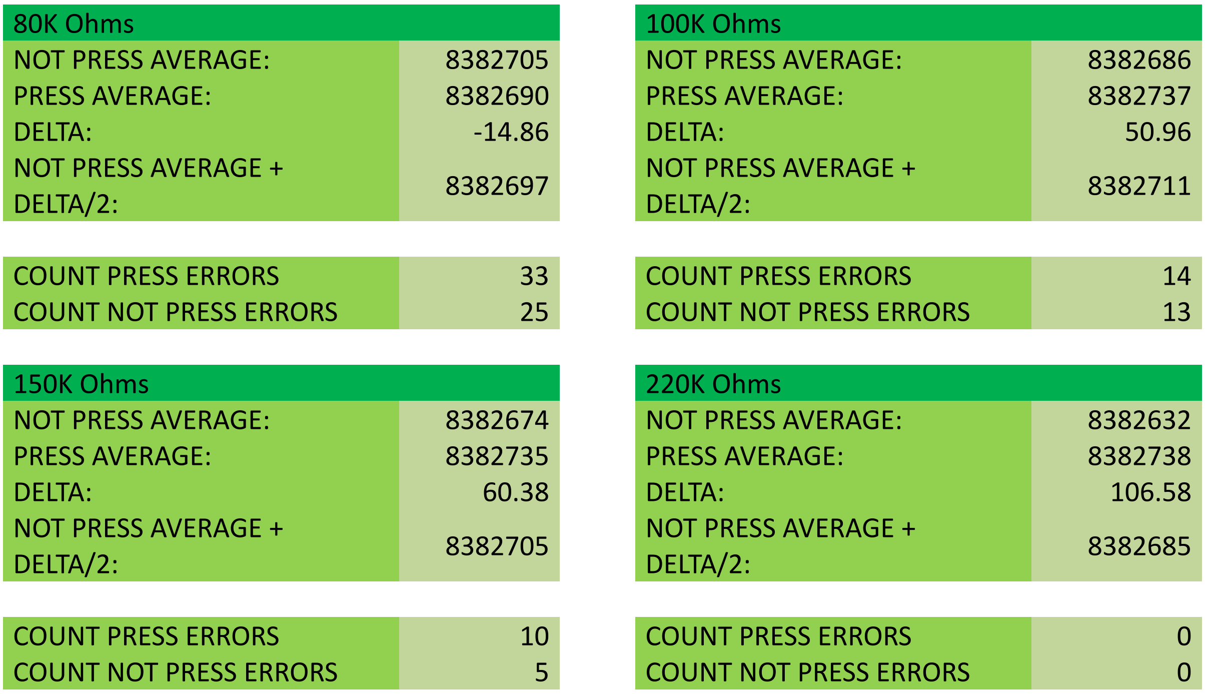


Figure 8. Wheatstone Bridge and Analog-to-Digital Converter Schematic

In order to define this optimal resistance value a test was proposed: the circuit with the tested values as R2 and R3 had 50 measurements taken by a test in which both SG sensors with and without pressure conditions were evaluated. For that, both averages were obtained and their values compared with a simple subtraction resulting in a delta constant. Taking into consideration that the pressed sensor had in general higher measured values than the not pressed ones, the two possible groups could be divided by their individual data measurement following this condition: if the measured value was lower than the not pressed measured average plus half of delta if would be considered not pressed, and if the measured value was higher than this value it would be considered pressed.

After the set up procedure, the next test consisted in checking all acquired values to see if they respected the condition of being within their respective zone range limit. Annex 5.2 shows a 15 elements sample for the result measured for 200K Ohms as R2 and R3, in this report the count of miss values consists in check whether the measured value is in the wrong range zone. The table 1 a), b), c) and d) shows the final results obtained for the tests using 80K Ohms, 100K Ohms, 150K Ohms and 220K Ohms.

Table 1 – a), b), c) and d) Wheatstone Bridge Tests Result Respectively for 80KOhms, 100K Ohms, 150K Ohms and 220K Ohms.



80K Ohms	
NOT PRESS AVERAGE:	8382705
PRESS AVERAGE:	8382690
DELTA:	-14.86
NOT PRESS AVERAGE + DELTA/2:	8382697

COUNT PRESS ERRORS	33
COUNT NOT PRESS ERRORS	25

100K Ohms	
NOT PRESS AVERAGE:	8382686
PRESS AVERAGE:	8382737
DELTA:	50.96
NOT PRESS AVERAGE + DELTA/2:	8382711

COUNT PRESS ERRORS	14
COUNT NOT PRESS ERRORS	13

150K Ohms	
NOT PRESS AVERAGE:	8382674
PRESS AVERAGE:	8382735
DELTA:	60.38
NOT PRESS AVERAGE + DELTA/2:	8382705

COUNT PRESS ERRORS	10
COUNT NOT PRESS ERRORS	5

220K Ohms	
NOT PRESS AVERAGE:	8382632
PRESS AVERAGE:	8382738
DELTA:	106.58
NOT PRESS AVERAGE + DELTA/2:	8382685

COUNT PRESS ERRORS	0
COUNT NOT PRESS ERRORS	0

From these tables it is possible to assure that according with the increase of the resistance value order in R2 and R3, the Delta value calculated increases and it is directly related with the decrease in the count of the Press Errors and Not Press Errors. This outcome is obtained comparing all individual measurements previously obtained for both groups with their respective numeric limit zone and counting all elements which are out of range.

Because of the positive results using 220K Ohms as R2 and R3 resistance, this value was chosen to be implemented in the final practical design.

2.10 – Arduino Wireless Communication

Two Arduino boards are used in the design to perform communication among the Arduino in the athletes' body protector and the software with which the training information is acquired and processed. For this function an RF Tx/Rx link use was chosen because practical factors. Several low budget antennas are available for this application due to the Arduino easy programming environment and the high number of users.

A preliminary test using 433MHz Tx/Rx antennas provided good results for both received information quality and operation distance. These two tests were the main concerns for this operation since it is necessary to have a satisfactory real time transmission in a dynamic environment where the athletes are in constant movement and sudden movements may do the difference in a fight score. For this test the design shown in figure 9 was used.

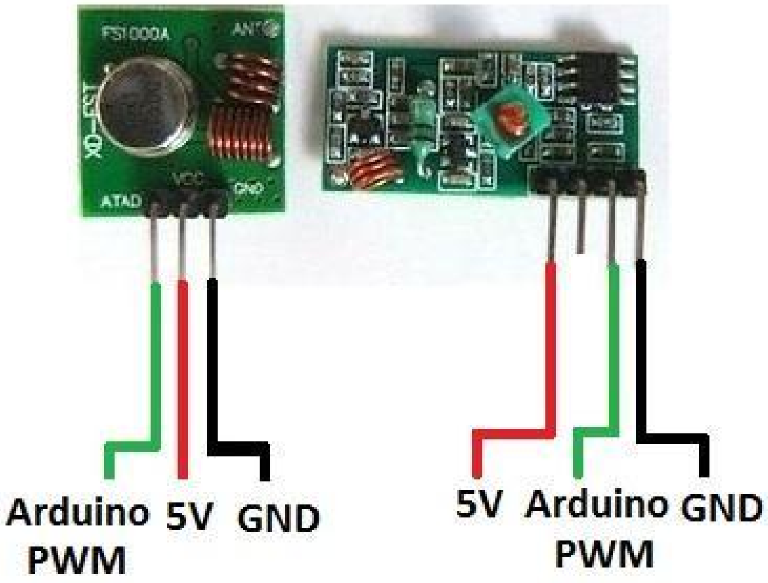


Figure 9. Wireless Transmission Circuit Schematic

2.11 – Arduino Shield Design

A critical concern about the Arduino design was the connection of all subsystems in a robust and sturdy way. This issue is even more critical for this project because the high amount of connections and needed electronic parts. To overcome this situation an Arduino Shield model was designed in a universal board, in order to make all connections with the Arduino and perform the electronic tasks in a solid platform.

Figure 10 shows this Arduino Shield designed from both sides with all electronic parts divided by its application in use.

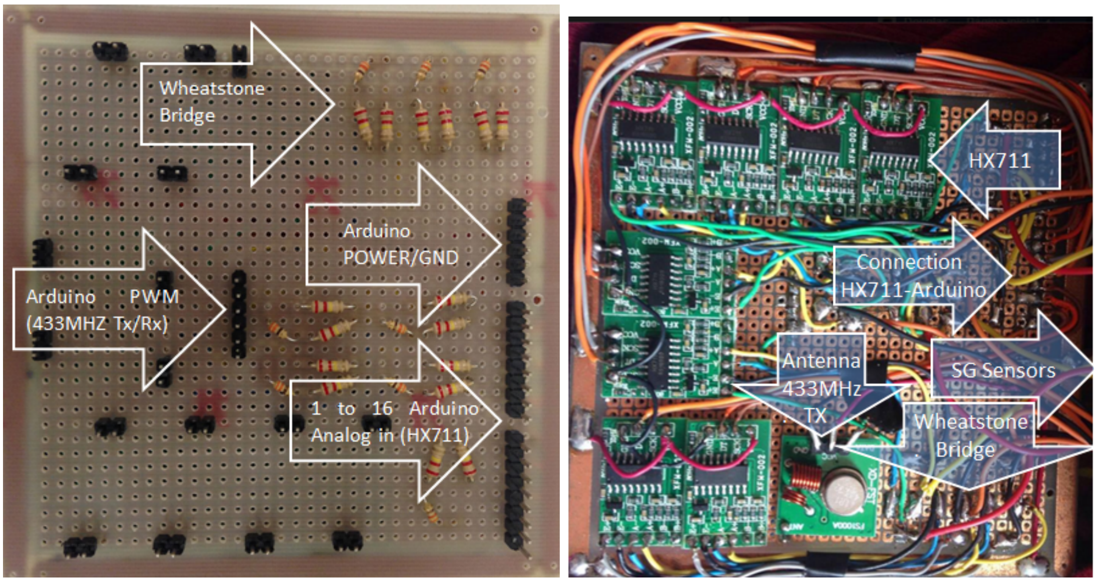


Figure 10. a) Bottom view of Arduino Shield b)Top view of Arduino Shield with electronics divided by application.

The last concern about the hardware design was the mechanical connection to the shirt, for this application a pocket with the Arduino size connected with the Arduino Shield was stitched onto the shirt's rear.

Figure 11 shows both structures allocated in the shirt's pocket rear.

Figure 11. Arduino and Arduino Shield allocated in the shirt's pocket rear.

2.12 – Arduino Shield Test Evaluation

For this analysis all data is already acquired in the Arduino allocated in the athlete's body protector and sent to the second Arduino connected in the computer via antenna streaming - Tx/Rx.

All SG sensors after connected to the rest of the circuit have different acquired response in time; the first individual analysis is to determine their average value and their respective variation. For that, it was observed that although the 8 SG sensors were implemented following a similar procedure, the fluctuation over the average value was significantly different for each case. It is utterly important for the design because the main idea in the blow measurement was to determine a line free of fluctuations in case of no impact and all values above this "threshold" line would be considered as an impact.

In order to identify a blow, a test against noise fluctuations is needed. It was considered a test of sets where the maximum level of noise amplitude was recorded, alongside with the level that actual blows produce. It was then set a threshold value, to which univocally represents the blow event, therefore counted as a valid event. Any level above this threshold value is deemed as a blow, any level below might be noise or a non-recorded strike. An alternative method proposed for cases which present high fluctuation levels considers fixing the only blows in a straightly limited amplitude range, therefore the measured blows will be within this zone. However, in real world tests the number of false positives found for large range analysis made this procedure unfeasible.

Four sets of tests were proposed to validate the firstly aforementioned implementation. The first set consists on the average measurement, with a large number of repetitions, in order to determine the superior limit of the fluctuation which is chosen manually a value higher than all measurements. The second is to evaluate this superior limit with a second blow free measurement which should be also free of false positives. This set of tests is very important to evaluate the robustness of the design against false results. The two final tests were to prove that the system is able to measure one and two blows over a large time acquiring data. It is important to detect sequential strikes whether or not the time between them is large or short.

The practical result for the last round is show in figure 12 where it is possible to evaluate the average value, the superior limit of fluctuation and the exact moment were occurred the two measured blows.

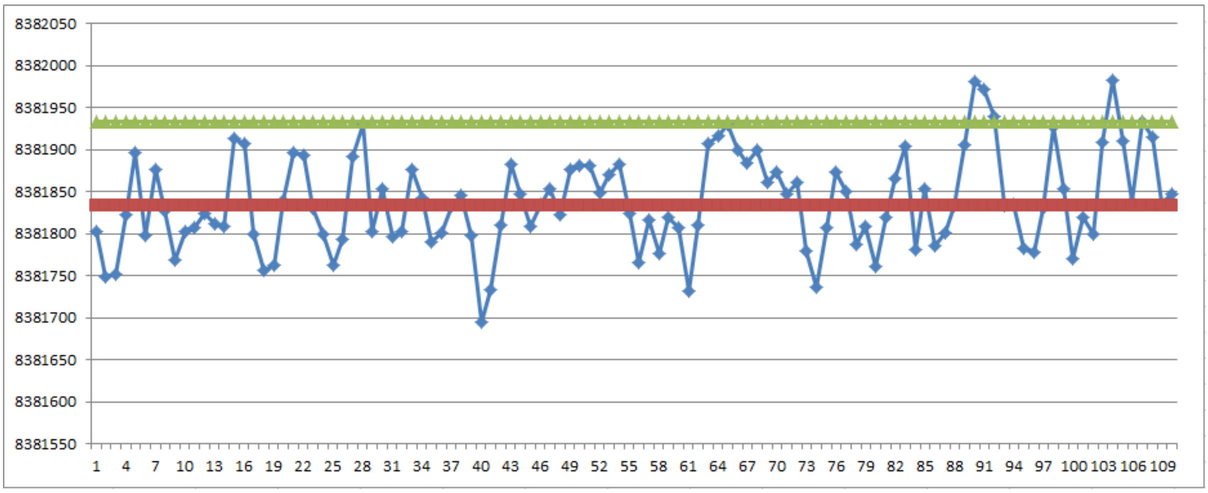


Figure 12 – Practical relation between average in red, determined maximum limit of fluctuation in green and acquired signal with two measured blows in blue.

2.13– Arduino Serial Communication

An important function that is largely facilitated using Arduinos' interface is the Serial Communication which is performed using the Serial Monitor Tool in its IDE. This function is essential for both tests and data acquisition. The test evaluation could be directly performed through data quality in real time processing. Data acquisitions function is performed extracting these values (from the Arduino IDE) to more robust programming languages that allow graphical interfaces and interconnection to other software applications.

3 – Software Interface and Data Processing

3.1 – User Interface

In addition to the hardware, a software application was developed in Java™ Language as to make the registering and record of athletes and their respective fight performance user friendly and comprehensive.

The data analysis is performed by fight mode, considering that the athlete can be studied by attack or defense skills. From this perspective, in the data acquiring interface the user has an option to choose the correct fight mode and all information of the session are going to be related to the corresponding method. Also, the date of acquiring session and number of session are available for the user when the data is sent to the databases server.

In addition to the data acquisition, the analysis is also dependent on the fight mode which it is necessary to be selected before to perform the next steps. As the mode is selected, the software selects by date the athlete activities and set the analysis type. It is discriminated rather in single day analysis, two day analysis or period analysis. In the single day analysis the user can select a single session in the chosen day by its number or the all-day session analysis. In two day analysis the user can choose two of the available days to dispose the all-day session analysis. In the period analysis all the available days available between the two selected dates are disposed in the all-day session analysis.

The result of these analysis is disposed in a new window which gives some summary of the operation, a protector image which has all the sensors discriminated by their spatial position and the processed statistical information. The statistical information is available in total number of hits, acquired by hits measured in all sensors with an impact higher than a parameter value, total number of effective hits, acquired in the same form as the previous but with a higher parameter, both indicators with time dependence, and the percentage relation between them.

After processing, all analysis information can be exported Excel sheet for future use.

Figure 13 shows the proposed software interface block diagram combining all functions previously described.

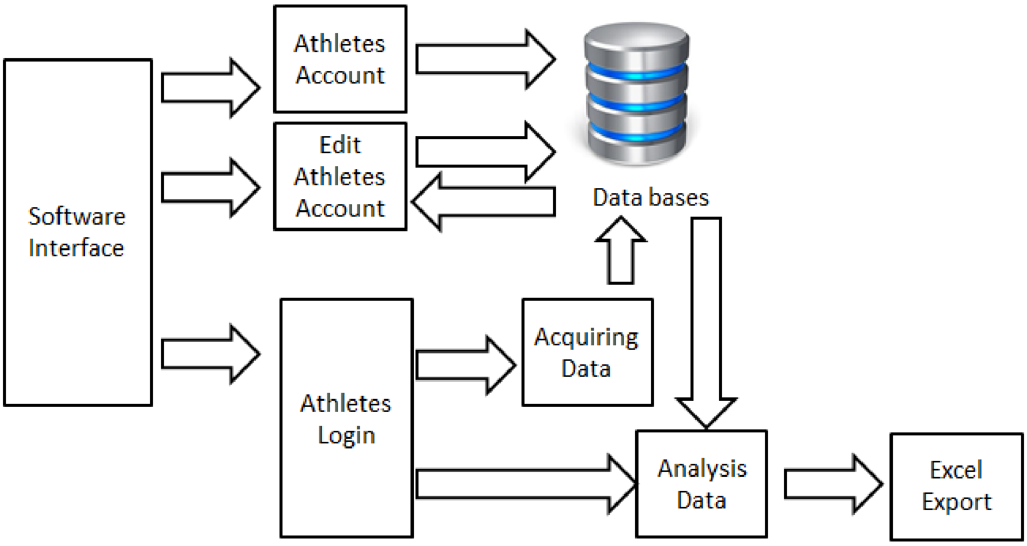


Figure 13. Block diagram for the software interface.

3.2 – Register Module

The first interface to be accessed is the main menu screen (fig. 14). There, a new athlete is able to create an account, by selecting the first option and inserting his personal information through a form, edit his previous created account when needed, by choosing the second button or simply login to reach the acquisition and analysis modules.

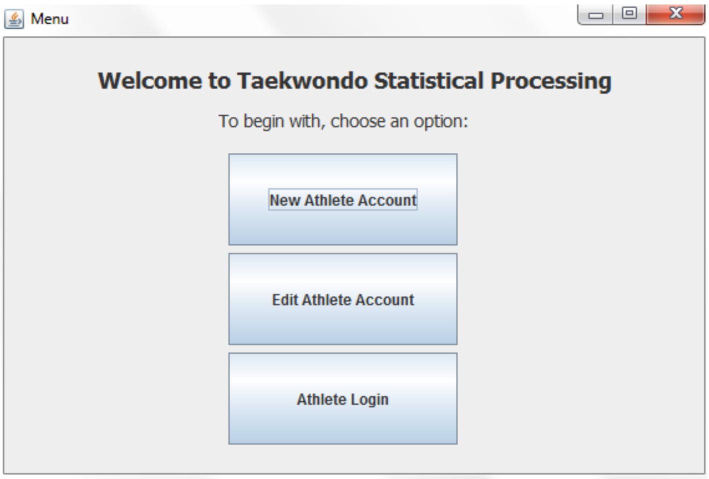
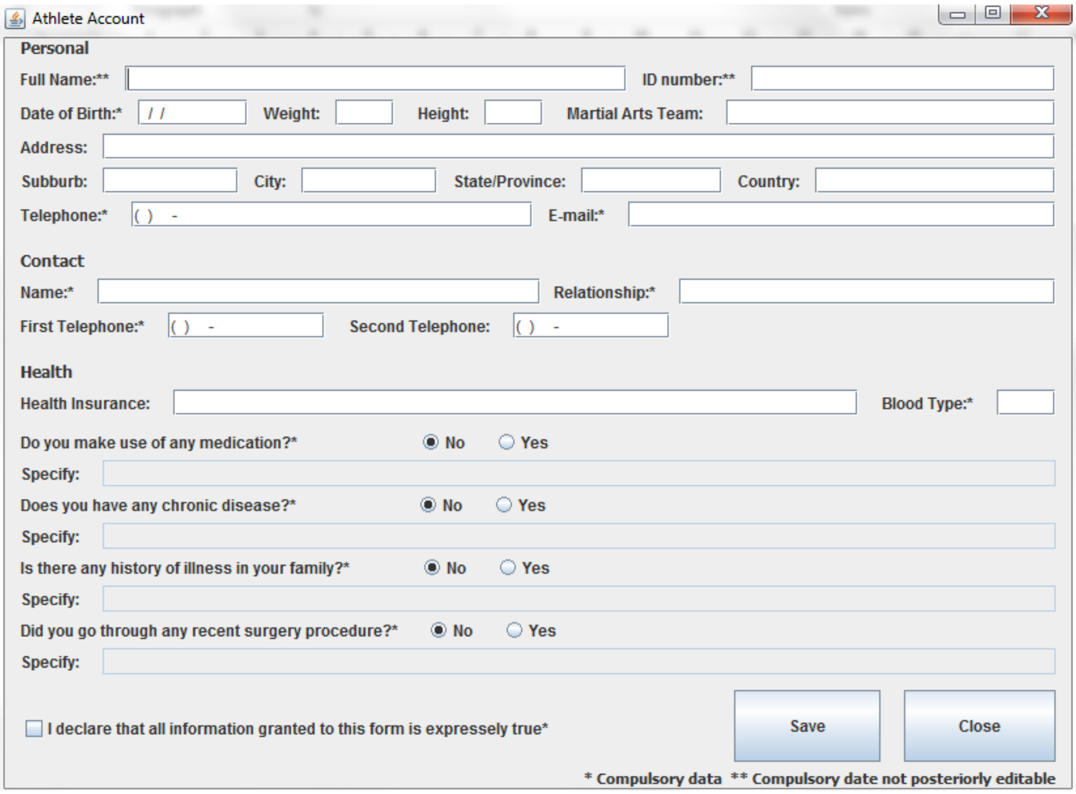

Menu
Welcome to Taekwondo Statistical Processing
To begin with, choose an option:
New Athlete Account
Edit Athlete Account
Athlete Login

Figure 14. Main menu screen

Before any other software operation, every user must create a new athlete account by filling his personal and health information in a form, indicating everything which is important for registering the athlete and also providing him a more accurate result analysis. This account is recorded in a database and can be accessed everytime the user wants to start a new acquisition, edit personal information or create a new analysis.

The form (fig. 15) indicates what kind of data is expected for each gap interval and also returns error messages when a user forgets to insert any compulsory data. This allows the application to keep a structured database in order to store, proccess and display analysis faster and precisely.



Athlete Account

Personal

Full Name:** ID number:**

Date of Birth:* / / Weight: Height: Martial Arts Team:

Address:

Subburb: City: State/Province: Country:

Telephone:* ( ) - E-mail:*

Contact

Name:* Relationship:*

First Telephone:* ( ) - Second Telephone: ( ) -

Health

Health Insurance: Blood Type:*

Do you make use of any medication?* ● No ○ Yes

Specify:

Does you have any chronic disease?* ● No ○ Yes

Specify:

Is there any history of illness in your family?* ● No ○ Yes

Specify:

Did you go through any recent surgery procedure?* ● No ○ Yes

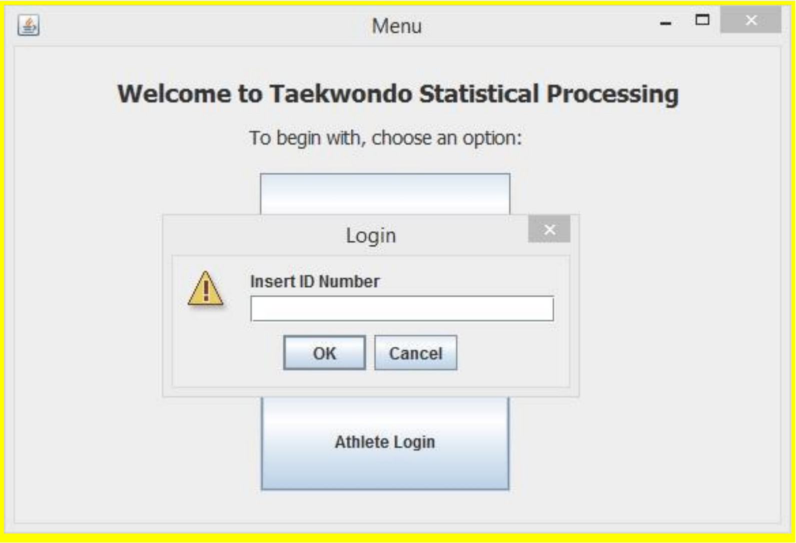
Specify:

□ I declare that all information granted to this form is expressely true*

* Compulsory data ** Compulsory date not posteriorly editable

Figure 15. Athlete account form

When an account is created, the user will own an ID number and be able to login (fig. 16), unlocking the main software functionalities, such as acquisition and analysis. Through all navigation steps, the athlete will be guided by error messages (fig. 17) and warning pop ups, indicating the correct application usage.



Menu

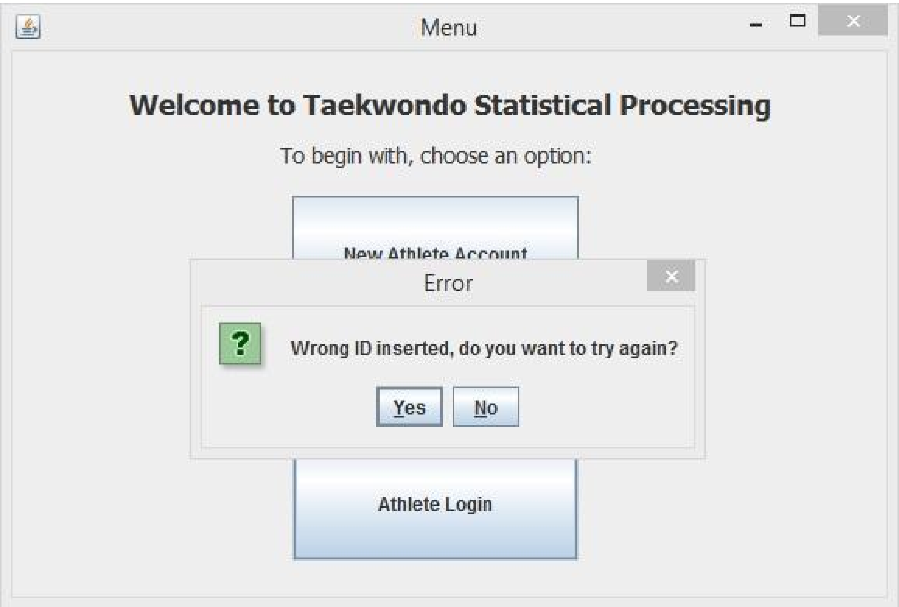
Welcome to Taekwondo Statistical Processing

To begin with, choose an option:

Login

Insert ID Number

Figure 16. Login pop up


Welcome to Taekwondo Statistical Processing

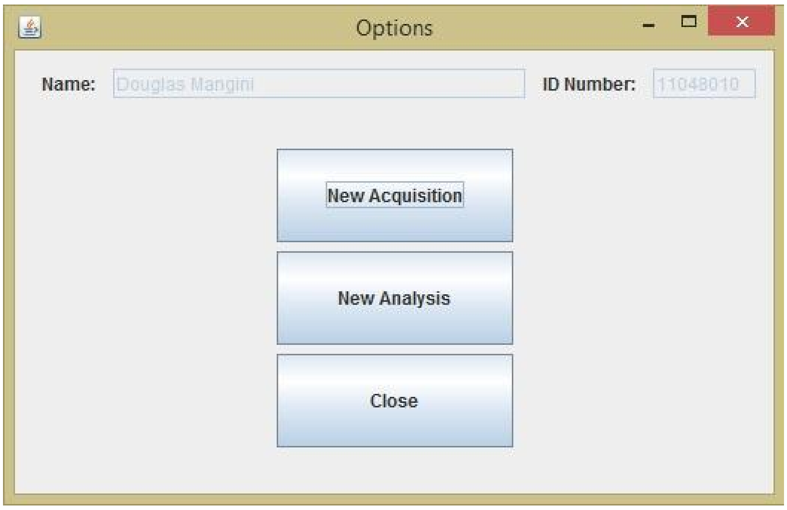
To begin with, choose an option:

Wrong ID inserted, do you want to try again?

Figure 17. Error message pop up.

3.3 – Options Module

If the respective ID number is found in the application database, the user will be redirected to the options screen (fig. 18). His/her name and ID number will be displayed on the top, to ensure it is actually the correct account. At this stage, the athlete is able to access the two main software functionalities: new acquisition and new analysis.


Name: Douglas Mangini

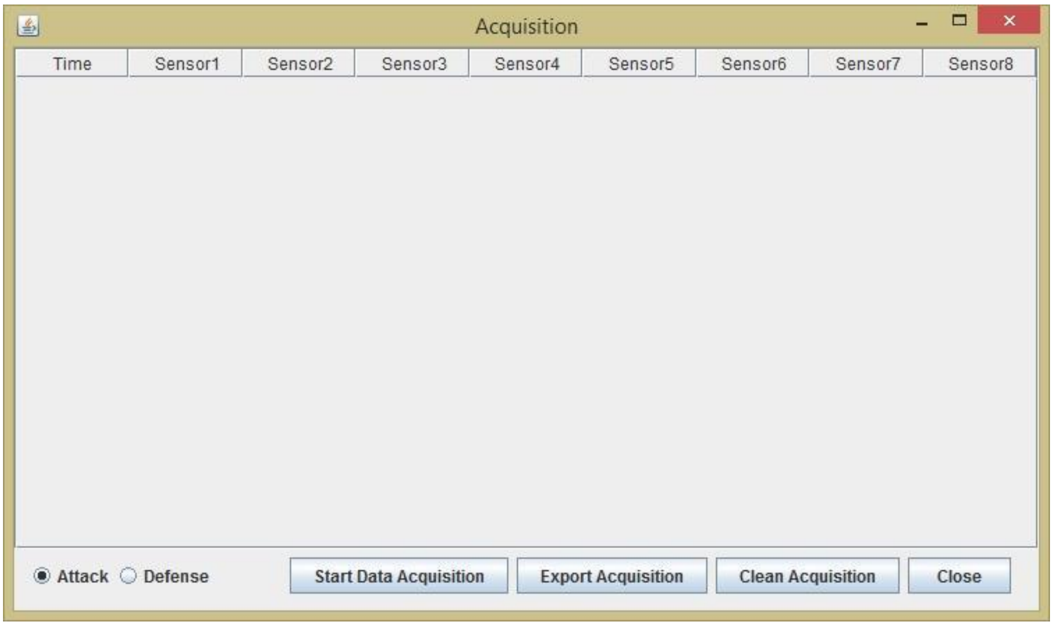
ID Number: 11048010

Figure 18. Options menu for a registered user

3.4 – Aquisition Module

The acquisition module (fig. 19) allows the user to gather information from the body protector during a training session and store raw data for further analysis. Basically it collects and displays time and all attached sensors information. Therefore it will be able to indicate which and when the sensor was hit.

Before each training fight session the athlete selects which mode will be practiced, whether attack or defense. Then he presses “Start Data Acquisition” and begins to fight.



Time	Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8

● Attack ○ Defense

Figure 19. Acquisition screen.

When the fight is over, he clicks on “Interrupt Data Acquisition” (fig. 20), which has the same position as the button mentioned above. Immediately, the software will stop receiving data from the body protector and the athlete will be able to store it in a database or simply delete this series, by pressing “Export Acquisition” or “Clean Acquisition” respectively.

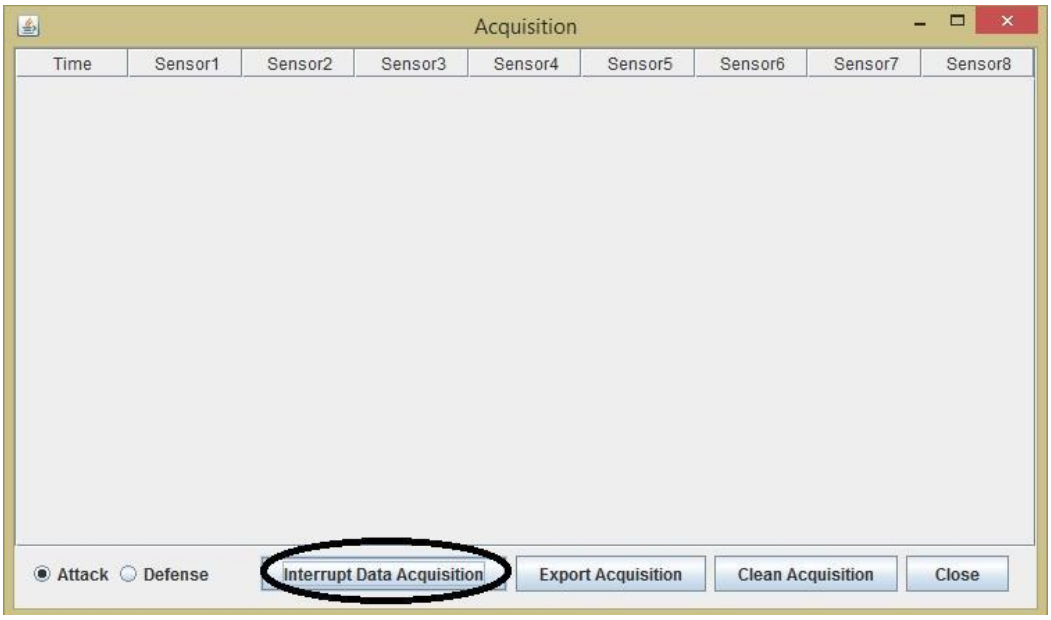
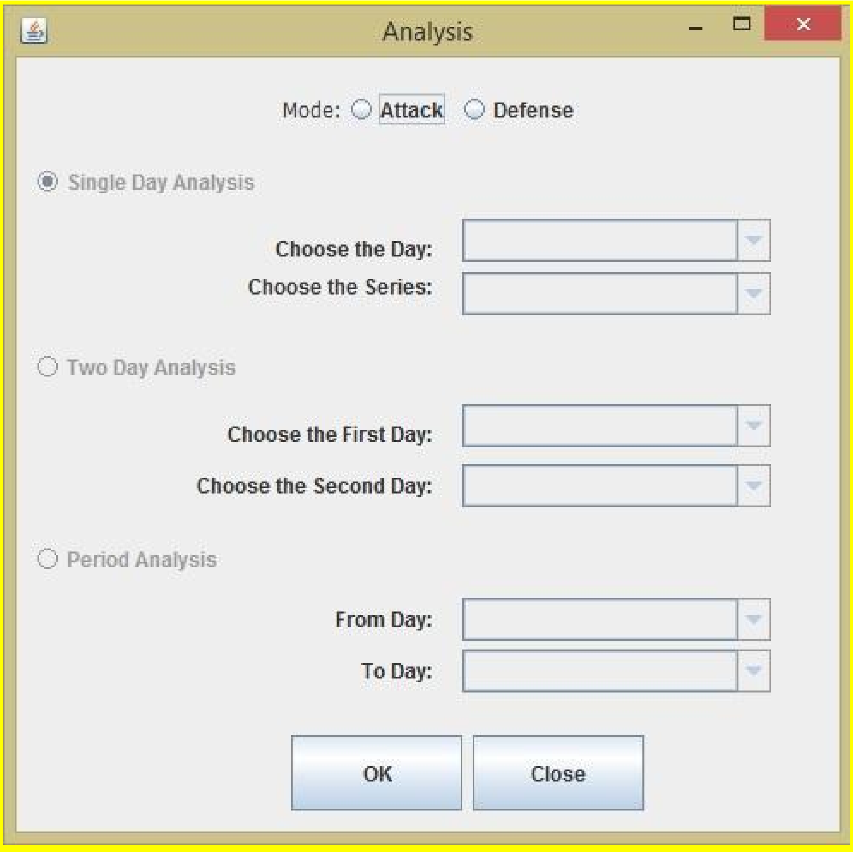


Figure 20. Interrupted acquisition.

3.5 – Analysis Module

Once the athlete has saved his first training series, he will be able to extract further analysis info from his training results. The Analysis Module (fig. 21), which is accessible from the options menu, is the tool for turning raw data into processed and relevant information.



Analysis

Mode: ○ Attack ○ Defense

● Single Day Analysis

Choose the Day:

Choose the Series:

○ Two Day Analysis

Choose the First Day:

Choose the Second Day:

○ Period Analysis

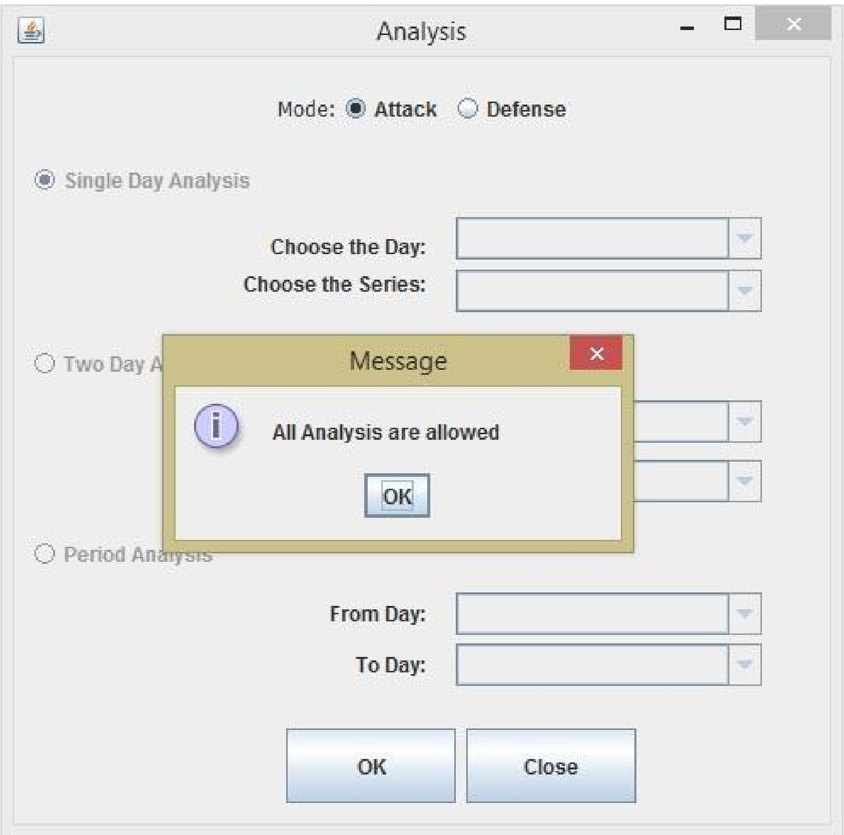
From Day:

To Day:

Figure 21. Analysis screen.

The athlete selects what type of mode he wants to analyze. When selected, the software pulls all correspondent series stored in the database and unlocks them for analysis. Then three kinds of report will be displayed to the user, based on the number of series he stored on the system.

For a single day analysis, the user chooses the day and the exact series he wants. This generates a single report about a training without any development comparison. Selecting a two day analysis, the athlete chooses two training days he wants to compare and it generates the desired report. The last option is selecting the period analysis, which allows the user to have a full comparison report related to a training period and check the day-by-day progress of his skills. Figs. 22 to 24 show the screenshots regarding the Analysis area.



Analysis

Mode: ● Attack ○ Defense

● Single Day Analysis

Choose the Day:

Choose the Series:

○ Two Day A

Message

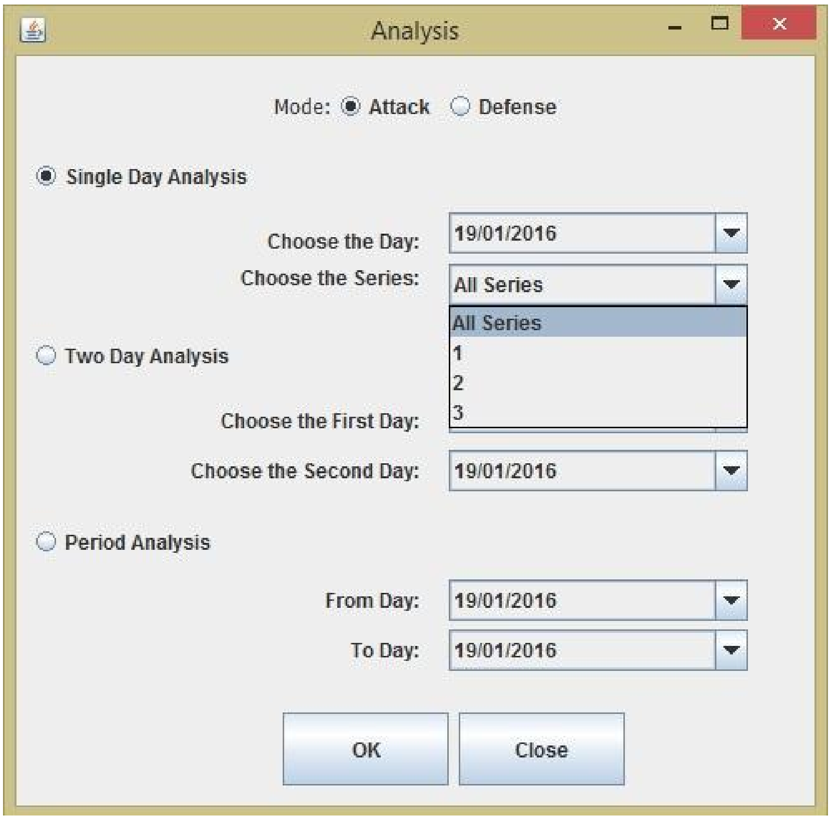
All Analysis are allowed

○ Period Analysis

From Day:

To Day:

Figure 22. Warning message indicating all analysis options are allowed for the selected mode.



Analysis

Mode: ● Attack ○ Defense

● Single Day Analysis

Choose the Day: 19/01/2016

Choose the Series: All Series

All Series
1
2
3

○ Two Day Analysis

Choose the First Day:

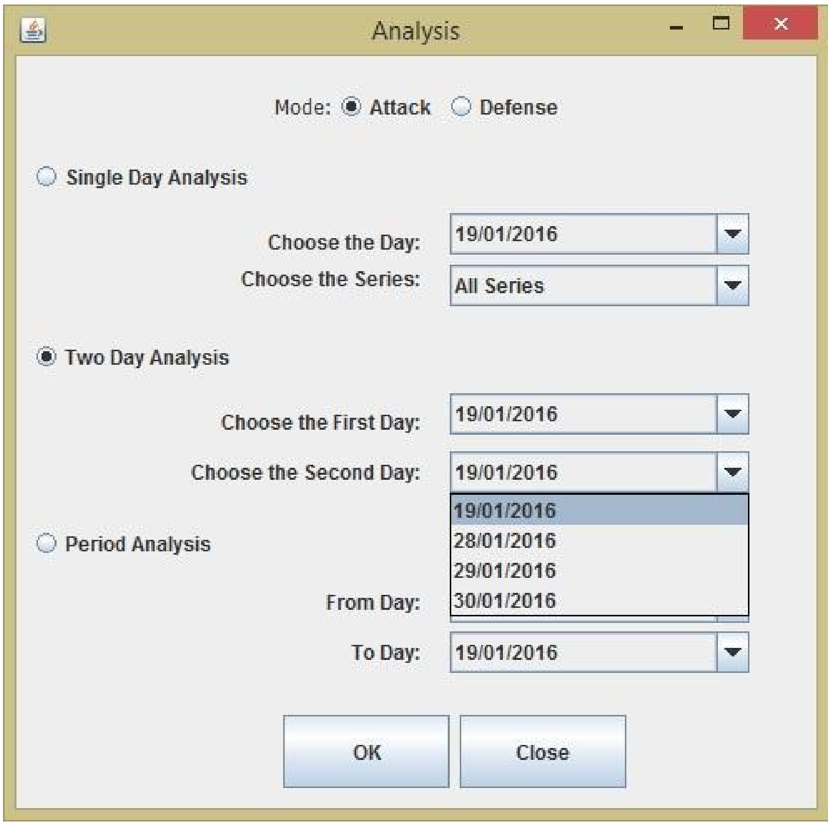
Choose the Second Day: 19/01/2016

○ Period Analysis

From Day: 19/01/2016

To Day: 19/01/2016

Figure 23. Single day analysis.



Mode: ● Attack ○ Defense

○ Single Day Analysis

Choose the Day: 19/01/2016

Choose the Series: All Series

● Two Day Analysis

Choose the First Day: 19/01/2016

Choose the Second Day: 19/01/2016

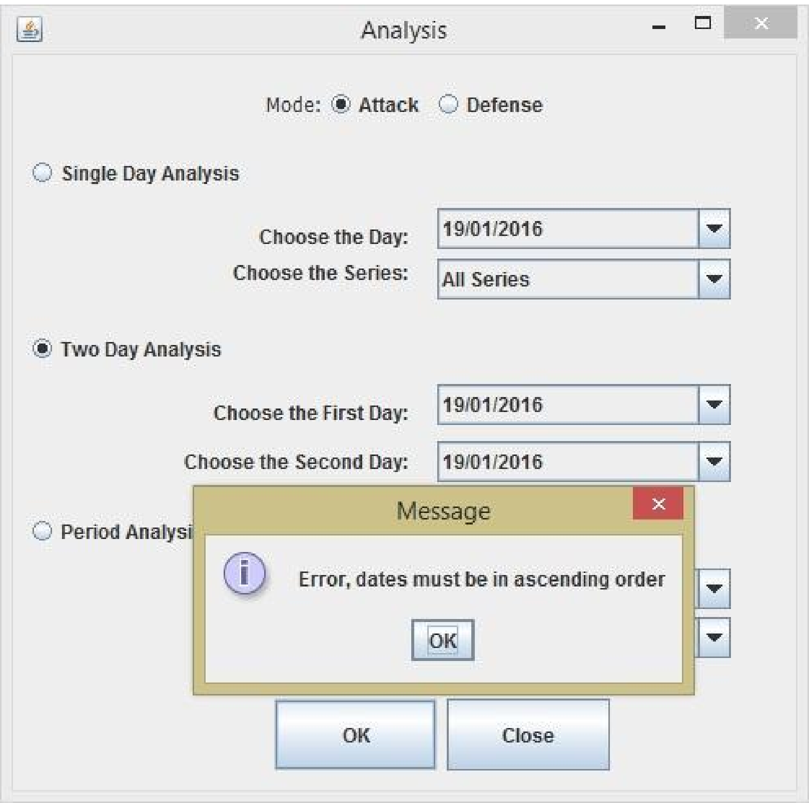
19/01/2016
28/01/2016
29/01/2016
30/01/2016

○ Period Analysis

From Day:

To Day: 19/01/2016

Figure 24. Two day analysis.



Analysis

Mode: ● Attack ○ Defense

○ Single Day Analysis

Choose the Day: 19/01/2016

Choose the Series: All Series

● Two Day Analysis

Choose the First Day: 19/01/2016

Choose the Second Day: 19/01/2016

○ Period Analysi

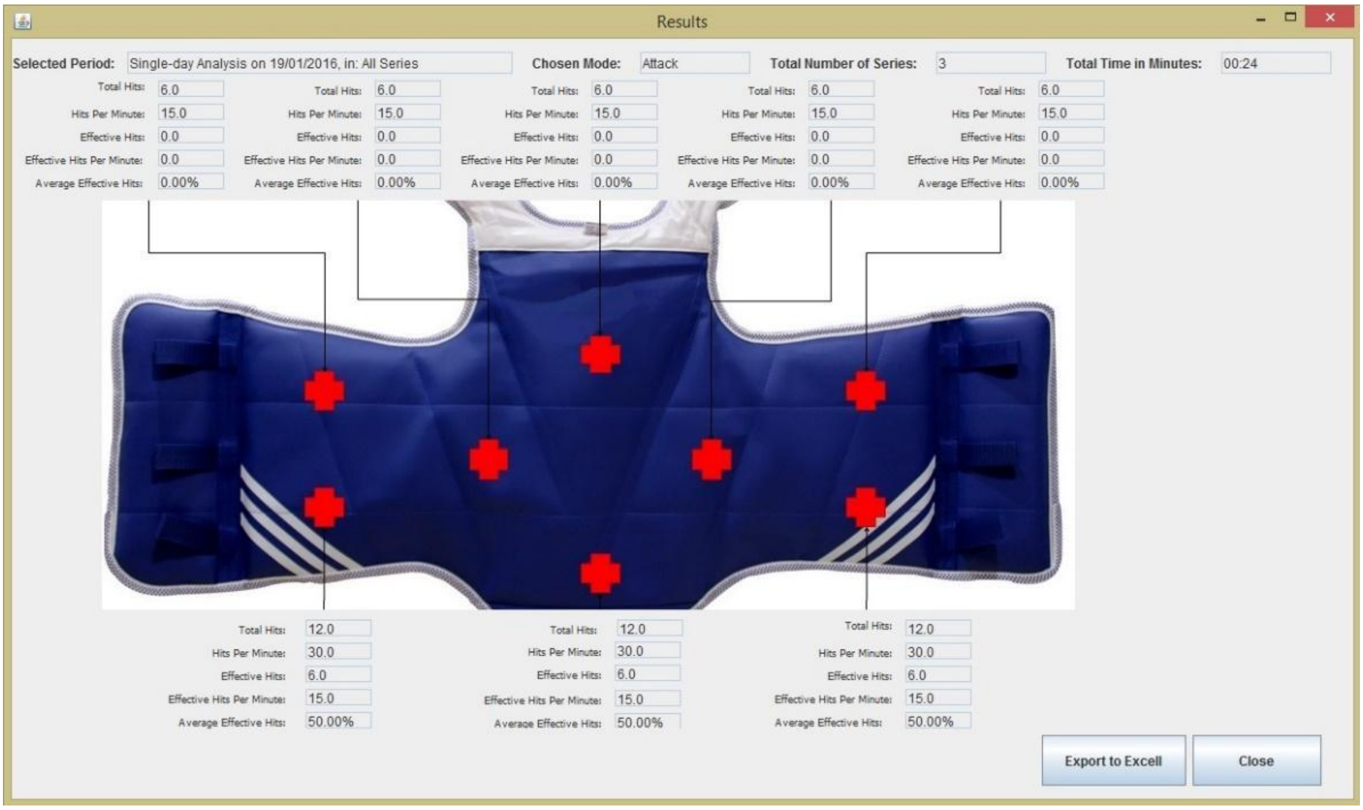
Message

Error, dates must be in ascending order

Figure 24. Error message for an invalid date selection.

3.6 – Results Module

After selecting the analysis mode, filling the required gaps and pressing “OK”, the application displays the information showed on Figure 25. This is a graphic report with all important information to a Taekwondo athlete attempting to improve his accuracy, speed and stamina. Information such as total hits, hits per minute, effective hits for each attached sensor and also training duration can be easily observed by the coach and team.



Selected Period: Single-day Analysis on 19/01/2016, in: All Series Chosen Mode: Attack Total Number of Series: 3 Total Time in Minutes: 00:24

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 12.0
Hits Per Minute: 30.0
Effective Hits: 6.0
Effective Hits Per Minute: 15.0
Average Effective Hits: 50.00%

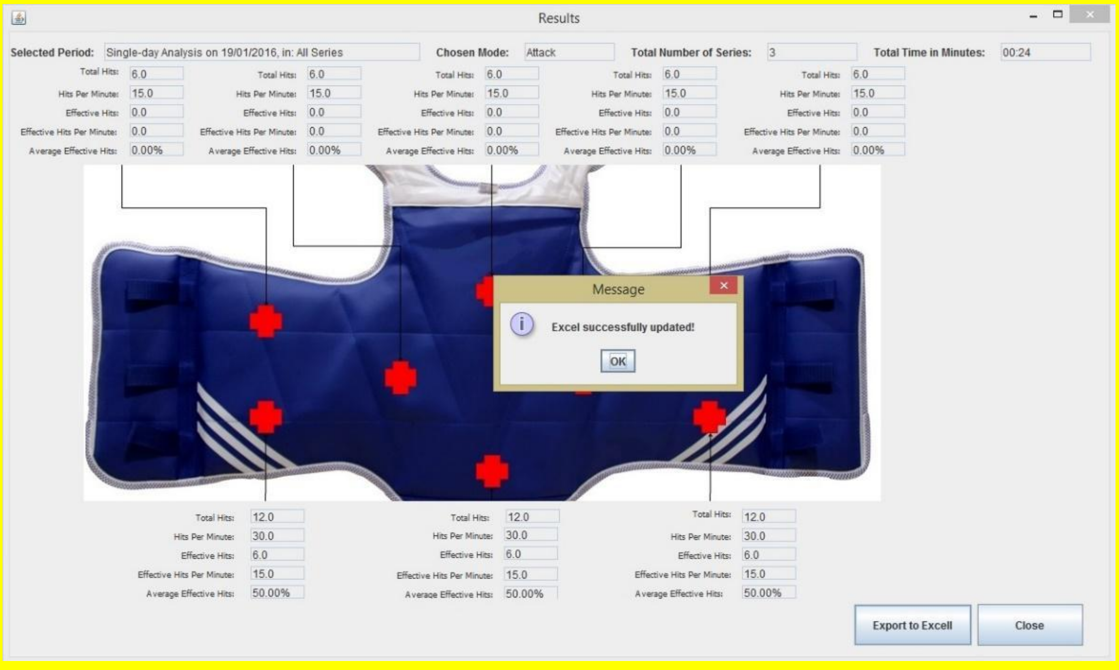
Total Hits: 12.0
Hits Per Minute: 30.0
Effective Hits: 6.0
Effective Hits Per Minute: 15.0
Average Effective Hits: 50.00%

Total Hits: 12.0
Hits Per Minute: 30.0
Effective Hits: 6.0
Effective Hits Per Minute: 15.0
Average Effective Hits: 50.00%

Figure 25. Results module screen.

3.7 – Excel Register Module

For further analysis purposes, any generated report can be exported to an excel file, as shown on Figure 26.



Selected Period: Single-day Analysis on 19/01/2016, in: All Series
Chosen Mode: Attack
Total Number of Series: 3
Total Time in Minutes: 00:24

Total Hits:	6.0	6.0	6.0	6.0	6.0
Hits Per Minute:	15.0	15.0	15.0	15.0	15.0
Effective Hits:	0.0	0.0	0.0	0.0	0.0
Effective Hits Per Minute:	0.0	0.0	0.0	0.0	0.0
Average Effective Hits:	0.00%	0.00%	0.00%	0.00%	0.00%

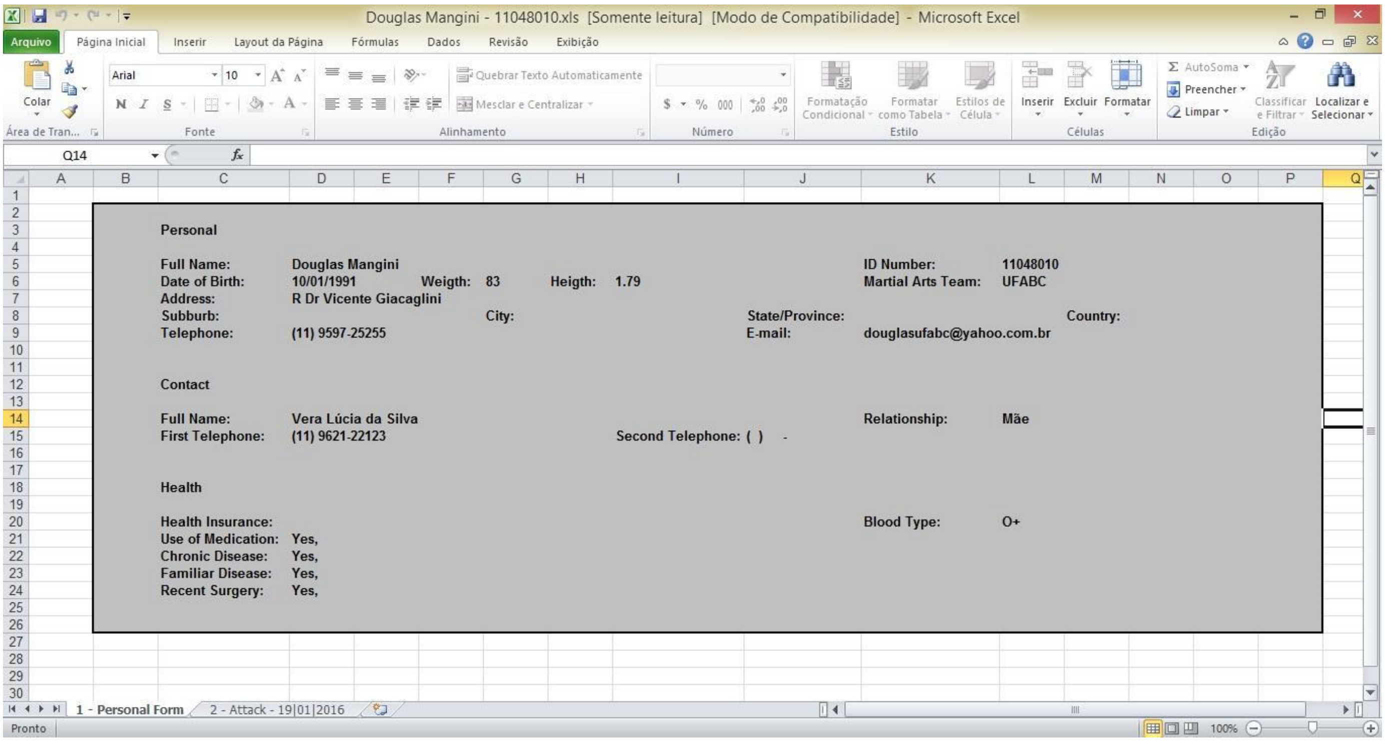
Message

Excel successfully updated!

Total Hits:	12.0	12.0	12.0
Hits Per Minute:	30.0	30.0	30.0
Effective Hits:	6.0	6.0	6.0
Effective Hits Per Minute:	15.0	15.0	15.0
Average Effective Hits:	50.00%	50.00%	50.00%

Figure 26. Results module being exported to an excel file.

The exported file (fig. 27 and 28) is automatically saved on the local computer, containing the athlete personal data and also the graphic report generated by his desired analysis. Each kind of information is displayed in a separate tab, for better organization.



Douglas Mangini - 11048010.xls [Somente leitura] [Modo de Compatibilidade] - Microsoft Excel

Personal

Full Name: Douglas Mangini
ID Number: 11048010
Date of Birth: 10/01/1991
Weigth: 83
Heigth: 1.79
Martial Arts Team: UFABC
Address: R Dr Vicente Giacaglini
Subburb:
City:
State/Province:
Country:
Telephone: (11) 9597-25255
E-mail: douglasufabc@yahoo.com.br

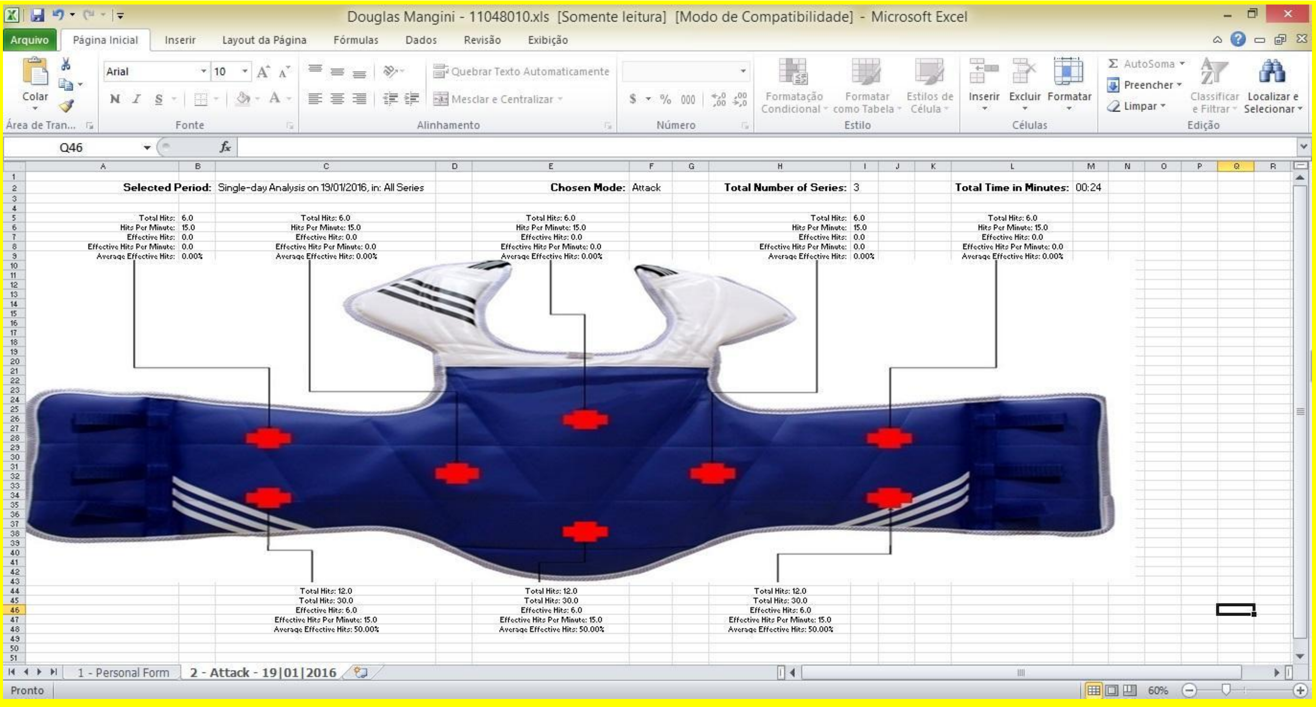
Contact

Full Name: Vera Lúcia da Silva
Relationship: Mãe
First Telephone: (11) 9621-22123
Second Telephone: ( ) -

Health

Health Insurance:
Blood Type: O+
Use of Medication: Yes,
Chronic Disease: Yes,
Familiar Disease: Yes,
Recent Surgery: Yes,

Figure 27. Exported excel – Personal Information tab.


Douglas Mangini - 11048010.xls [Somente leitura] [Modo de Compatibilidade] - Microsoft Excel

Selected Period: Single-day Analysis on 19/01/2016, in: All Series
Chosen Mode: Attack
Total Number of Series: 3
Total Time in Minutes: 00:24

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 6.0
Hits Per Minute: 15.0
Effective Hits: 0.0
Effective Hits Per Minute: 0.0
Average Effective Hits: 0.00%

Total Hits: 12.0
Total Hits: 30.0
Effective Hits: 6.0
Effective Hits Per Minute: 15.0
Average Effective Hits: 50.00%

Total Hits: 12.0
Total Hits: 30.0
Effective Hits: 6.0
Effective Hits Per Minute: 15.0
Average Effective Hits: 50.00%

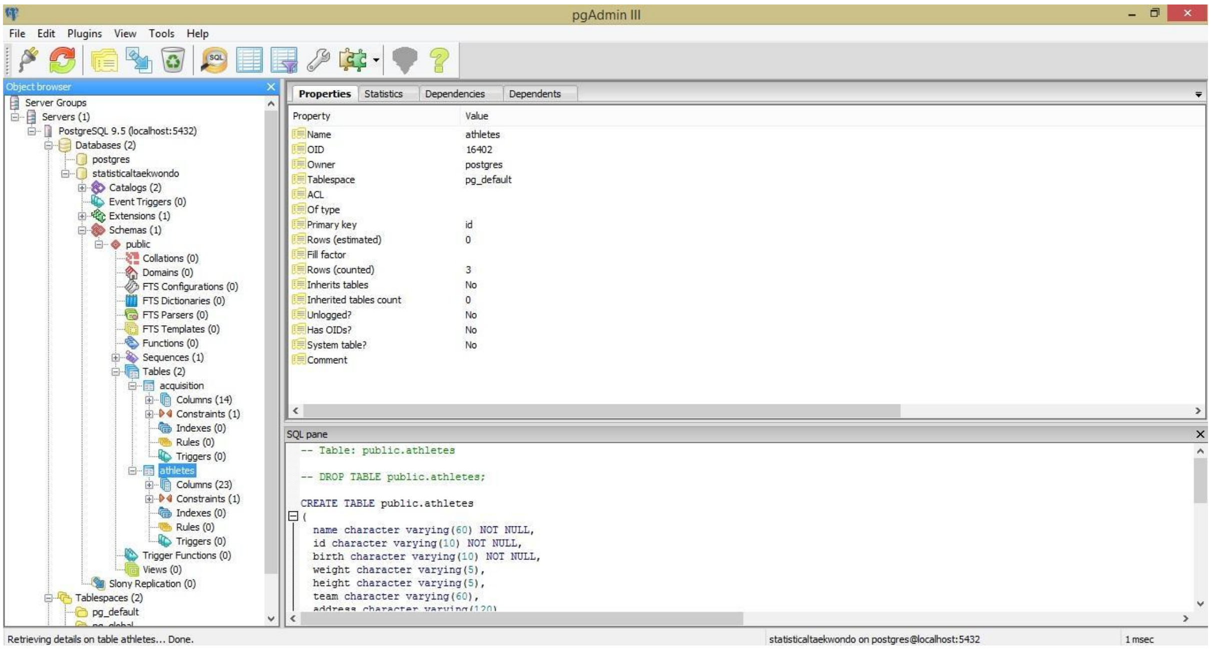
Total Hits: 12.0
Total Hits: 30.0
Effective Hits: 6.0
Effective Hits Per Minute: 15.0
Average Effective Hits: 50.00%

Figure 28. Exported excel – Analysis information tab.

3.8 – Database implementation

The full application was developed in Java by the use of Netbeans and knowledge of object oriented programming, which allowed the research team to implement an user friendly interface to Taekwondo athletes in the same way we could improve the application iteratively using an Agile software development method.

The database was implemented in SQL, using a software named PostGre (fig. 29 and 30). Its architecture was designed to meet the application modules and to address the demand of storing data from the user and the body protector, also retrieving previously stored data to the user interface layer in real time.



Property	Value
Name	athletes
OID	16402
Owner	postgres
Tablespace	pg_default
ACL	
Of type	
Primary key	id
Rows (estimated)	0
Fill factor	
Rows (counted)	3
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	

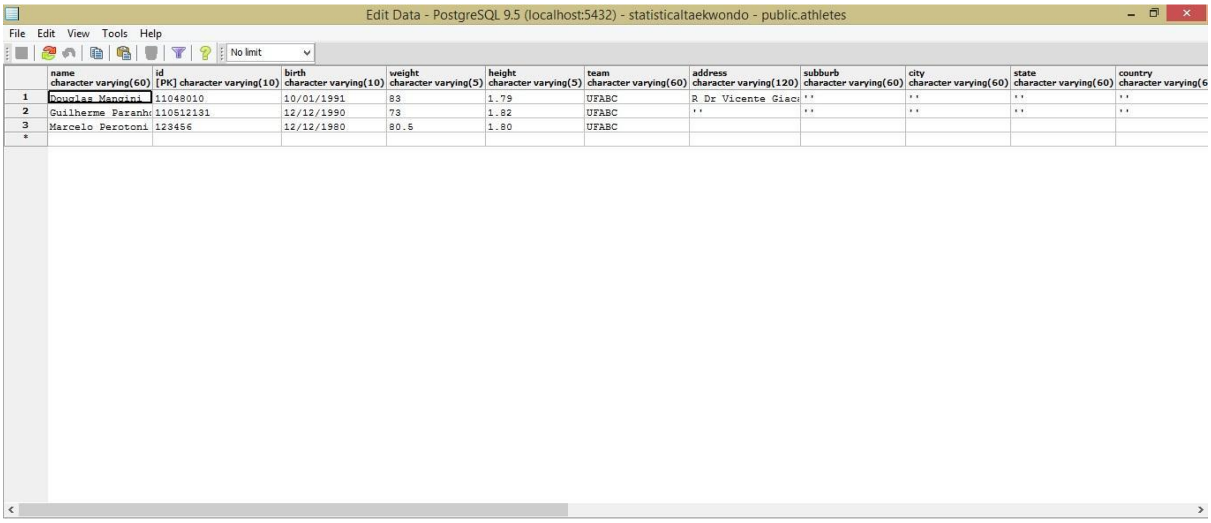
```
-- Table: public.athletes

-- DROP TABLE public.athletes;

CREATE TABLE public.athletes
(
  name character varying(60) NOT NULL,
  id character varying(10) NOT NULL,
  birth character varying(10) NOT NULL,
  weight character varying(5),
  height character varying(5),
  team character varying(60),
  address character varying(120)
```

Retrieving details on table athletes... Done.

statisticaltaekwondo on postgres@localhost:5432

1 msec

Figure 29. Database architecture


Edit Data - PostgreSQL 9.5 (localhost:5432) - statisticaltaekwondo - public.athletes
File
Edit
View
Tools
Help
No limit

	name character varying(60)	id [PK] character varying(10)	birth character varying(10)	weight character varying(5)	height character varying(5)	team character varying(60)	address character varying(120)	subburb character varying(60)	city character varying(60)	state character varying(60)	country character varying(6
1	Douglas Mangini	11048010	10/01/1991	83	1.79	UFABC	R Dr Vicente Giaca	''	''	''	''
2	Guilherme Paranh	110512131	12/12/1990	73	1.82	UFABC	''	''	''	''	''
3	Marcelo Perotoni	123456	12/12/1980	80.5	1.80	UFABC					
*											

Figure 30. Example of a database table

4.0 –Conclusion

In hardware concepts we can highlight the project reached excellent data acquisition for some of the implemented sensors. However, because of manufacturing and application characteristics some of them did not work as expected and we could not manage to solve the situation without replacing sensors. Therefore, we can determine that for market application, a deeper study should be done to identify causes of this issue, avoiding feedstock and time waste on tests. Another test we suggest concerns the protection of the applied sensor, which was frequently damaged during test events. About software interface, the project provided us a great set of programming skills, mainly in the connection among different software and platforms. The only improvement we suggest is to enhance software complexity to comport both athletes at the same training session and to be able of analyzing them both, in attack and defense at the same time.

5.0 —References

[1] Rios, G. B., O Processo de Esportivização do Taekwondo. Pensar a prática V.8 n.1 revisada.p65, September 16th, 2006.

[2] Chi, E. H., Borriello, G., Hunt, G., Davies, N., Pervasive Computing in Sports Technologies, *IEEE CS and IEEE ComSoc,* 1536-1268/05/, 2015.

[3] Filho, J. C. B., Schattenberg, L. D., Stollmeier, N., Tecnologias Esportivas Auxiliando no Esporte. Revista Eletrônica do Alto Vale do Itajaí, V. 2 No 2, December 2013.

[4] Leveaux, R., 2012 Olympic Games Decision Making Technologies for Taekwondo Competition, *Communications of the IBIMA, Vol. 2012 (2012), Article ID 834755, 2012.*

[5] Miziara, I. M. Proposta de um sistema para a avaliação biomecânica de atletas de taekwondo. Faculdade de Engenharia Elétrica, UFU. 2014.

[6] Cavalcante, M. A., Tavolaro, C. R. C., & Molisani, E., Physics with Arduino for Beginers, Revista Brasileira de Ensino de Física, v. 33, n.4, (2011).

[7] McRoberts, Arduino Básico, Cap.1, Editora Novatec, São Paulo, 2011.

[8] Souza, A. R., Paixão, A. C., Uzêda, D. D., Dias, M. A., Duarte, S., & Amorin, H. S., The Arduino Board: A Low Cost Option for Physics Experiments Assisted by PC, Revista Brasileira de Ensino de Física, v. 33, n.1, (2011).

[9] Arduino an Introduction, Available at: www.arduino.cc/en/guide/introduction. Acessed in 29 of August of 2015.

[10] Class notes of the unit AN3145 of the international University Of Sarajevo, department of Electrical and Electronic Engineering. Strain Gauge Circuits. Available at: http://ee.ius.edu.ba/sites/default/files/u747/strain-gages.pdf, in 30 of August of 2015.

[11] Operation manual of the HX711 circuit, AVIA SEMICONDUCTOR Available at: cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf, in 05 of September pf 2015.

6.0 – Annex

6.1 – Electronic Schematics

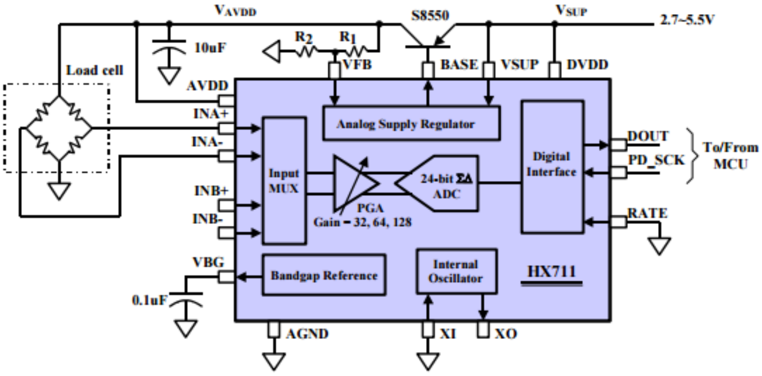


Figure 32. HX711 schematic block diagram obtained in its manual of use.

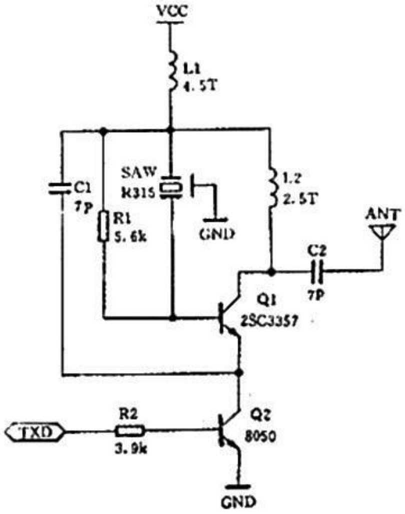


Figure 33. Transmitter MX-FS schematic block diagram obtained in its manual of use.

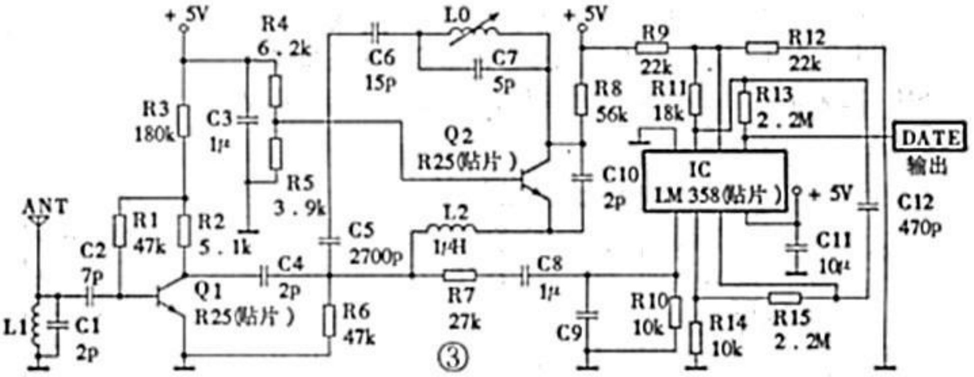


Figure 34. Receiver MX schematic block diagram obtained in its manual of use.

6.2 – Wheatstone Bridge Sample Test Using 200K Ohms

NOT PRESS	PRESS	MISS PRESS	MISS NOT PRESS
8382625	8382762	FALSE	FALSE
8382608	8382746	FALSE	FALSE
8382582	8382738	FALSE	FALSE
8382612	8382759	FALSE	FALSE
8382609	8382729	FALSE	FALSE
8382598	8382717	FALSE	FALSE
8382614	8382737	FALSE	FALSE
8382632	8382759	FALSE	FALSE
8382640	8382737	FALSE	FALSE
8382623	8382722	FALSE	FALSE
8382632	8382716	FALSE	FALSE
8382667	8382707	FALSE	FALSE
8382665	8382695	FALSE	FALSE
8382646	8382698	FALSE	FALSE

NOT PRESS AVERAGE:	8382632
PRESS AVERAGE:	8382738
DELTA:	106.58
NOT PRESS AVERAGE + DELTA/2:	8382685

COUNT PRESS ERRORS	0
COUNT NOT PRESS ERRORS	0

* DELTA = PRESS AVERAGE – NOT PRESS AVERAGE

6.3 – Expenses Management

Product	Quantit	Value (R$)
Arduino Mega	2	120
Hx711	8	120
GS Sensors	8	60

Resistors	24	2.4
433MHz Tx/Rx antennas	1	13
Electronic Universal Board	1	10
Plug/Support for Battery	1	7
Battery	6	5
Taekwondo Body Protector	1	90
Tank Top	1	20
Wires	-	10
Solder tin	-	2
Multimeter	1	28
	SUM:	487.4

6.4 – Arduino Transmitter Code

```
#include <VirtualWire.h>
#include "hx711.h"
float Sensor[8] = {0, 0, 0, 0, 0, 0, 0, 0};
static char sprintfbuffer[15];

void setup()
{
   vw_set_ptt_inverted(true);  // Required by the RF module
   vw_setup(2000);           // bps connection speed
   vw_set_tx_pin(3);        // Arduino pin to connect the receiver data pin
}

void loop()
{
  //Message to send:
```

```
const char *msg = "New";
vw_send((uint8_t *)msg, strlen(msg));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A1, A0);
Sensor[0] = scale.read;           // Receive Sensor 1 Signal
dtostrf(Sensor[0], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A3, A2);
Sensor[1] = scale.read;            // Receive Sensor 2 Signal
dtostrf(Sensor[1], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A5, A4);
Sensor[2] = scale.read;            // Receive Sensor 3 Signal
dtostrf(Sensor[2], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);       // We wait to send the next message

Hx711 scale(A7, A6);
Sensor[3] = scale.read;            // Receive Sensor 4 Signal
dtostrf(Sensor[3], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A9, A8);
Sensor[4] = scale.read;            // Receive Sensor 5 Signal
dtostrf(Sensor[4], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A11, A10);
Sensor[5] = scale.read;            // Receive Sensor 6 Signal
dtostrf(Sensor[5], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A13, A12);
Sensor[6] = scale.read;            // Receive Sensor 7 Signal
dtostrf(Sensor[6], 6, 4, sprintfbuffer);
vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
vw_wait_tx();       // We wait to finish sending the message
delay(50);        // We wait to send the next message

Hx711 scale(A15, A14);
Sensor[7] = scale.read;           // Receive Sensor 8 Signal
dtostrf(Sensor[7], 6, 4, sprintfbuffer);
```

```
  vw_send((uint8_t *)sprintfbuffer, strlen(sprintfbuffer));
  vw_wait_tx();       // We wait to finish sending the message
  delay(50);        // We wait to send the next message

  delay(500);            // wait for half second to send next transmission
}
```

6.5 – Arduino Receiver Code

```
#include <VirtualWire.h>
boolean Key = false; //Allow Serial Transmission
boolean New = false; //Waits Until First Element
int message = 1; //Serial Transmission Mode
String Buffer;

void setup()
{
  Serial.begin(9600);         // Configure the serial connection to the computer
  vw_set_ptt_inverted(true);  // Required by the RF module
  vw_setup(2000);           // bps connection speed
  vw_set_rx_pin(3);        // Arduino pin to connect the receiver data pin
  vw_rx_start();          // Start the receiver
}

void loop()
{
  if(Serial.available()>0){
  message = Serial.read();

  if (message == '1'){
   Key = true;
   New = false;
  }
  else
  if (message == '2'){
   Key = false;
  }
 }

 if (Key == true && New == false) {
  Buffer = "";
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if (vw_get_message(buf, &buflen))      // We check if we have received data
  {
   int i;
   // Message with proper check
   for (i = 0; i < buflen; i++)
   {
   Buffer = buf[i];
   }
   if (Buffer.equals("119"))
   {
    New = true;
   }
  }
 }
 if (Key == true && New == true) {
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if (vw_get_message(buf, &buflen))      // We check if we have received data
  {
    int i;
    // Message with proper check
    for (i = 0; i < buflen; i++)
    {
      Serial.write(buf[i]);  // The received data is stored in the buffer
                        // and sent through the serial port to the computer
    }
    Serial.println();
  }
 }
}
```

6.6 – Interaction between Arduino and Java Code

```
package Connection;

import java.sql.*;
import javax.swing.JOptionPane;

public class ConnectSQL {

    public Statement Statem;
    public ResultSet Result;
    public Connection Connect;
    private final String Driver = "org.postgresql.Driver";
    private final String Path = "jdbc:postgresql://localhost:5432/statisticaltaekwondo";
    private final String User = "postgres";
    private final String Pass = "5432";

    public void Conenction(){
        try {
            System.setProperty("jdbc.Drivers", Driver);
            Connect = DriverManager.getConnection(Path, User, Pass);
            // JOptionPane.showMessageDialog(null, "Server Successfully Connected");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
            Conenction();
        }
    }

    public void Desconnection() {
        try {
            Connect.close();
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error in Stop Connection! \n Error: " +ex.getMessage());
        }
    }
 }
```

6.7 – Java Login Code

```
package statisticaltaekwondo;

import Connection.ConnectSQL;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

    String ID = null;
    String IDSQL = null;
    String NameSQL = null;
    boolean seek = false;
    boolean edit = false;
    int answer;

    public Login() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton3 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Menu");

        jButton1.setText("New Athlete Account");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Athlete Login");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel1.setText("Welcome to Taekwondo Statistical Processing");

        jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel2.setText("To begin with, choose an option:");

        jButton3.setText("Edit Athlete Account");
    jButton3.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
      }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
        .addContainerGap(180, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 184,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 184,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 184,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(198, 198, 198))
      .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGap(75, 75, 75)
            .addComponent(jLabel1))
          .addGroup(layout.createSequentialGroup()
            .addGap(172, 172, 172)
            .addComponent(jLabel2)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(23, 23, 23)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel2)
        .addGap(18, 18, 18)
        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(23, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    edit = false;
    Register Window = new Register(edit, ID);
    Window.setVisible(true);
  }
```

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        edit = false;
        Log();
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        edit = true;
        Log();
    }


    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {
            JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
        }

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Login().setVisible(true);
            }
        });
    }

    public void Log() {
        ConnectSQL ConnectSQL = new ConnectSQL();
        ConnectSQL.Conenction();
        seek = false;

        ID = JOptionPane.showInputDialog(null, "Insert ID Number", "Login", JOptionPane.OK_CANCEL_OPTION );
        if (ID != null){
            try {
                String strSQL = "SELECT id, name FROM athletes";
                PreparedStatement pst = ConnectSQL.Connect.prepareStatement(strSQL);
                ResultSet result = pst.executeQuery();
                while (result.next() && !seek) {
                    if (ID.equals(result.getString(1))) {
                        IDSQL = result.getString(1);
                        NameSQL = result.getString(2);
                        seek = true;
                        if (!edit) {
                            Options Janela = new Options(IDSQL, NameSQL);
                            Janela.setVisible(true);
                        }
                        else  {
                            Register Window = new Register(edit, IDSQL);
                            Window.setVisible(true);
                        }
                    }
                }
            if (!seek) {
               answer = JOptionPane.showConfirmDialog(null, "Wrong ID inserted, do you want to try again?",
"Error", JOptionPane.YES_NO_OPTION);
               if (answer == JOptionPane.YES_OPTION) {
                Log();
                }
            }
         }
         catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
         }
      }
   }

   // Variables declaration - do not modify
   private javax.swing.JButton jButton1;
   private javax.swing.JButton jButton2;
   private javax.swing.JButton jButton3;
   private javax.swing.JLabel jLabel1;
   private javax.swing.JLabel jLabel2;
   // End of variables declaration
}
```

6.8 – Java Register Code

```
package statisticaltaekwondo;

import Connection.ConnectSQL;
import java.io.File;
import java.io.IOException;
import javax.swing.JOptionPane;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Register extends javax.swing.JFrame {

   ConnectSQL ConnectSQL = new ConnectSQL();

   String Name = null;
   String ID = null;
   String Birth = null;
   String Weight = null;
   String Height = null;
   String Team = null;
   String Address = null;
   String Subburb = null;
   String City = null;
   String State = null;
   String Country = null;
   String Telephone1 = null;
   String Email = null;
   String Contact = null;
   String Relationship = null;
   String Telephone2 = null;
   String Telephone3 = null;
   String Insurance = null;
   String Blood = null;
   String Medication = null;
   String Disease = null;
   String Familiar = null;
   String Surgery = null;
   boolean Confirm = false;
   boolean Case1 = false;
   boolean Case2 = false;
   boolean Case3 = false;
   boolean Case4 = false;
   boolean seek = false;
   boolean edit = false;
   String Path = "C:\\Users\\Douglas\\Desktop\\StatisticalTaekwondo\\";

   public Register(boolean editStr, String IDStr) {
       initComponents();
       ConnectSQL.Conenction();
       edit = editStr;
       if (edit == true) {
           ID = IDStr;
           EditRegister();
       }
       else  {
       jTextField1.setText(null);
       jTextField2.setText(null);
       jTextField3.setText(null);
       jTextField4.setText(null);
       jTextField5.setText(null);
       jTextField6.setText(null);
       jTextField7.setText(null);
       jTextField8.setText(null);
       jTextField9.setText(null);
       jTextField10.setText(null);
       jTextField11.setText(null);
       jTextField12.setText(null);
       jTextField13.setText(null);
       jTextField14.setText(null);
       jTextField15.setText(null);
       jTextField16.setText(null);
       jTextField17.setText(null);
       jTextField18.setText(null);
       jTextField19.setText(null);
       jTextField20.setText(null);
       jTextField21.setText(null);
       jTextField22.setText(null);
       jTextField23.setText(null);
       }
   }

   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {

       MedicationGroup = new javax.swing.ButtonGroup();
       DiseaseGroup = new javax.swing.ButtonGroup();
       FamiliarGroup = new javax.swing.ButtonGroup();
       OperationGroup = new javax.swing.ButtonGroup();
       jLabel1 = new javax.swing.JLabel();
```

```
        jLabel2 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        try{
           javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("*************************************************************");
           jTextField1 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel3 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        try{
           javax.swing.text.MaskFormatter mask = new javax.swing.text.MaskFormatter("######");
           jTextField2 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jTextField4 = new javax.swing.JTextField();
        try{
           javax.swing.text.MaskFormatter mask = new javax.swing.text.MaskFormatter("##***");
           jTextField4 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel6 = new javax.swing.JLabel();
        jTextField5 = new javax.swing.JTextField();
        try{
           javax.swing.text.MaskFormatter mask = new javax.swing.text.MaskFormatter("#****");
           jTextField5 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel7 = new javax.swing.JLabel();
        jTextField6 = new javax.swing.JTextField();
        jLabel8 = new javax.swing.JLabel();
        jTextField7 = new javax.swing.JTextField();
        try{
           javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("****************************************************************
**********************************************************");
           jTextField7 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel9 = new javax.swing.JLabel();
        jTextField8 = new javax.swing.JTextField();
        try{
           javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("*************************************************************");
           jTextField8 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel10 = new javax.swing.JLabel();
        jTextField9 = new javax.swing.JTextField();
        jLabel11 = new javax.swing.JLabel();
        jTextField10 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("************************************************************");
            jTextField10 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel12 = new javax.swing.JLabel();
        jTextField11 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("************************************************************");
            jTextField11 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel13 = new javax.swing.JLabel();
        jTextField12 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter telephone = new javax.swing.text.MaskFormatter("(##) ####-####*");
            jTextField12 = new javax.swing.JFormattedTextField(telephone);
        }
        catch (Exception e){
        }
        jLabel14 = new javax.swing.JLabel();
        jTextField13 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("************************************************************");
            jTextField13 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel15 = new javax.swing.JLabel();
        jLabel16 = new javax.swing.JLabel();
        jTextField14 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("************************************************************");
            jTextField14 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel17 = new javax.swing.JLabel();
        jTextField15 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("************************************************************");
            jTextField15 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel18 = new javax.swing.JLabel();
        jTextField16 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter telephone = new javax.swing.text.MaskFormatter("(##) ####-####*");
            jTextField16 = new javax.swing.JFormattedTextField(telephone);
```

```
    }
    catch (Exception e){
    }
    jLabel19 = new javax.swing.JLabel();
    jTextField17 = new javax.swing.JTextField();
    try{
      javax.swing.text.MaskFormatter telephone = new javax.swing.text.MaskFormatter("(##) ####-####*");
      jTextField17 = new javax.swing.JFormattedTextField(telephone);
    }
    catch (Exception e){
    }
    jLabel20 = new javax.swing.JLabel();
    jLabel21 = new javax.swing.JLabel();
    jTextField18 = new javax.swing.JTextField();
    try{
      javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("************************************************************");
      jTextField18 = new javax.swing.JFormattedTextField(mask);
    }
    catch (Exception e){
    }
    jLabel22 = new javax.swing.JLabel();
    jTextField19 = new javax.swing.JTextField();
    try{
      javax.swing.text.MaskFormatter mask = new javax.swing.text.MaskFormatter("*****");
      jTextField19 = new javax.swing.JFormattedTextField(mask);
    }
    catch (Exception e){
    }
    jLabel23 = new javax.swing.JLabel();
    jRadioButton1 = new javax.swing.JRadioButton();
    jRadioButton2 = new javax.swing.JRadioButton();
    jLabel24 = new javax.swing.JLabel();
    jTextField20 = new javax.swing.JTextField();
    try{
      javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("***************************************************************
*********************************************************");
      jTextField20 = new javax.swing.JFormattedTextField(mask);
    }
    catch (Exception e){
    }
    jLabel25 = new javax.swing.JLabel();
    jRadioButton3 = new javax.swing.JRadioButton();
    jRadioButton4 = new javax.swing.JRadioButton();
    jLabel26 = new javax.swing.JLabel();
    jTextField21 = new javax.swing.JTextField();
    try{
      javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("***************************************************************
*********************************************************");
      jTextField21 = new javax.swing.JFormattedTextField(mask);
    }
    catch (Exception e){
    }
    jLabel27 = new javax.swing.JLabel();
    jRadioButton5 = new javax.swing.JRadioButton();
    jRadioButton6 = new javax.swing.JRadioButton();
```

```
        jLabel28 = new javax.swing.JLabel();
        jTextField22 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("***************************************************************
*********************************************************");
            jTextField22 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel29 = new javax.swing.JLabel();
        jRadioButton7 = new javax.swing.JRadioButton();
        jRadioButton8 = new javax.swing.JRadioButton();
        jLabel30 = new javax.swing.JLabel();
        jTextField23 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("***************************************************************
*********************************************************");
            jTextField23 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }
        jLabel31 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jCheckBox1 = new javax.swing.JCheckBox();
        jTextField3 = new javax.swing.JTextField();
        try{
            javax.swing.text.MaskFormatter birth = new javax.swing.text.MaskFormatter("##/##/####");
            jTextField3 = new javax.swing.JFormattedTextField(birth);
        }
        catch (Exception e){
        }
        jLabel32 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Athlete Account");

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel1.setText("Personal");

        jLabel2.setText("Full Name:**");

        jLabel3.setText("ID number:**");

        jTextField2.setToolTipText("Compulsory use of 6 numbers");

        jLabel4.setText("Date of Birth:*");

        jLabel5.setText("Weight:");

        jTextField4.setToolTipText("Value in Kg");

        jLabel6.setText("Height:");

        jTextField5.setToolTipText("Value in meters");
        jLabel7.setText("Martial Arts Team:");

        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("***************************************************************");
            jTextField6 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }

        jLabel8.setText("Address:");

        jTextField7.setToolTipText("Street, Number, Complement");

        jLabel9.setText("Subburb:");

        jLabel10.setText("City:");

        try{
            javax.swing.text.MaskFormatter mask = new
javax.swing.text.MaskFormatter("***************************************************************");
            jTextField9 = new javax.swing.JFormattedTextField(mask);
        }
        catch (Exception e){
        }

        jLabel11.setText("State/Province:");

        jLabel12.setText("Country:");

        jLabel13.setText("Telephone:*");

        jTextField12.setToolTipText("(xx) xxxx-xxxxx");

        jLabel14.setText("E-mail:*");

        jLabel15.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel15.setText("Contact");

        jLabel16.setText("Name:*");

        jLabel17.setText("Relationship:*");

        jTextField15.setToolTipText("Family, friends, couple");

        jLabel18.setText("First Telephone:*");

        jTextField16.setToolTipText("(xx) xxxx-xxxxx");

        jLabel19.setText("Second Telephone:");

        jTextField17.setToolTipText("(xx) xxxx-xxxxx");

        jLabel20.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel20.setText("Health");

        jLabel21.setText("Health Insurance:");
```

```
jLabel22.setText("Blood Type:*");

jLabel23.setText("Do you make use of any medication?* ");

MedicationGroup.add(jRadioButton1);
jRadioButton1.setSelected(true);
jRadioButton1.setText("No");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jRadioButton1ActionPerformed(evt);
   }
});

MedicationGroup.add(jRadioButton2);
jRadioButton2.setText("Yes");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jRadioButton2ActionPerformed(evt);
   }
});

jLabel24.setText("Specify:");

jTextField20.setEditable(false);
jTextField20.setToolTipText("Please, provide more information");

jLabel25.setText("Does you have any chronic disease?* ");

DiseaseGroup.add(jRadioButton3);
jRadioButton3.setSelected(true);
jRadioButton3.setText("No");
jRadioButton3.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jRadioButton3ActionPerformed(evt);
   }
});

DiseaseGroup.add(jRadioButton4);
jRadioButton4.setText("Yes");
jRadioButton4.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jRadioButton4ActionPerformed(evt);
   }
});

jLabel26.setText("Specify:");

jTextField21.setEditable(false);
jTextField21.setToolTipText("Please, provide more information");

jLabel27.setText("Is there any history of illness in your family?*");

FamiliarGroup.add(jRadioButton5);
jRadioButton5.setSelected(true);
jRadioButton5.setText("No");
jRadioButton5.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jRadioButton5ActionPerformed(evt);
```

```
    }
});

FamiliarGroup.add(jRadioButton6);
jRadioButton6.setText("Yes");
jRadioButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton6ActionPerformed(evt);
    }
});

jLabel28.setText("Specify:");

jTextField22.setEditable(false);
jTextField22.setToolTipText("Please, provide more information");

jLabel29.setText("Did you go through any recent surgery procedure?* ");

OperationGroup.add(jRadioButton7);
jRadioButton7.setSelected(true);
jRadioButton7.setText("No");
jRadioButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton7ActionPerformed(evt);
    }
});

OperationGroup.add(jRadioButton8);
jRadioButton8.setText("Yes");
jRadioButton8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton8ActionPerformed(evt);
    }
});

jLabel30.setText("Specify:");

jTextField23.setEditable(false);
jTextField23.setToolTipText("Please, provide more information");

jLabel31.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel31.setText("* Compulsory data");

jButton1.setText("Save");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Close");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jCheckBox1.setText("I declare that all information granted to this form is expressely true*");
```

```
        jCheckBox1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jCheckBox1ActionPerformed(evt);
            }
        });

        jTextField3.setToolTipText("dd/mm/yyyy");

        jLabel32.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel32.setText("** Compulsory date not posteriorly editable");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                        .addComponent(jCheckBox1)
                        .addGap(18, 38, Short.MAX_VALUE)
                        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 112,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 116,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel2)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 384,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel3)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField2))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel5)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jLabel6)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jLabel7)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField6))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel8)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField7))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                        .addComponent(jLabel13)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField12)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel14)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField13, javax.swing.GroupLayout.PREFERRED_SIZE, 327,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel16)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField14, javax.swing.GroupLayout.PREFERRED_SIZE, 339,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(10, 10, 10)
                        .addComponent(jLabel17)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField15))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel21)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField18)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jLabel22)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField19, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel24)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField20))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel26)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField21))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel28)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField22))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel30)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField23))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(jLabel31)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel32))
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                          .addComponent(jLabel1)
                          .addComponent(jLabel15)
                          .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel18)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(jTextField16, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(jLabel19)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(jTextField17, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE))
                          .addComponent(jLabel20)
                          .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel23)
                            .addGap(89, 89, 89)
                            .addComponent(jRadioButton1)
                            .addGap(18, 18, 18)
                            .addComponent(jRadioButton2))
                          .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel25)
                            .addGap(88, 88, 88)
                            .addComponent(jRadioButton3)
                            .addGap(18, 18, 18)
                            .addComponent(jRadioButton4))
                          .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel27)
                            .addGap(53, 53, 53)
                            .addComponent(jRadioButton5)
                            .addGap(18, 18, 18)
                            .addComponent(jRadioButton6))
                          .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel29)
                            .addGap(18, 18, 18)
                            .addComponent(jRadioButton7)
                            .addGap(18, 18, 18)
                            .addComponent(jRadioButton8)))
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel9)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel10)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel11)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel12)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField11)))
                    .addContainerGap())
                );
                layout.setVerticalGroup(
                  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                  .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3)
                    .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(jLabel5)
                    .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel6)
                    .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel7)
                    .addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel8))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel9)
                    .addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel10)
                    .addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel11)
                    .addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel12)
                    .addComponent(jTextField11, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel13)
                    .addComponent(jTextField12, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel14)
                    .addComponent(jTextField13, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)
                    .addComponent(jLabel15)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel16)
                    .addComponent(jTextField14, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField15, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel17))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel18)
                        .addComponent(jTextField16, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel19)
                        .addComponent(jTextField17, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)
                    .addComponent(jLabel20)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel21)
                        .addComponent(jTextField18, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel22)
                        .addComponent(jTextField19, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel23)
                        .addComponent(jRadioButton1)
                        .addComponent(jRadioButton2))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel24)
                        .addComponent(jTextField20, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel25)
                        .addComponent(jRadioButton3)
                        .addComponent(jRadioButton4))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel26)
                        .addComponent(jTextField21, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel27)
                        .addComponent(jRadioButton5)
                        .addComponent(jRadioButton6))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel28)
                        .addComponent(jTextField22, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel29)
                        .addComponent(jRadioButton7)
                        .addComponent(jRadioButton8))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel30)
                        .addComponent(jTextField23, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 55, Short.MAX_VALUE)
                            .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGap(20, 20, 20))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jCheckBox1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel31)
                            .addComponent(jLabel32)))))
        );

        pack();
    }// </editor-fold>

    private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        if (jRadioButton1.isEnabled()){
            Case1 = false;
        }
        jTextField20.setEditable(false);
        jTextField20.setText("");
    }

    private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        Case1 = jRadioButton2.isEnabled();
        jTextField20.setEditable(true);
    }

    private void jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        if (jRadioButton3.isEnabled()){
            Case2 = false;
        }
        jTextField21.setEditable(false);
        jTextField21.setText("");
    }

    private void jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        Case2 = jRadioButton4.isEnabled();
        jTextField21.setEditable(true);
    }

    private void jRadioButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        if (jRadioButton5.isEnabled()){
            Case3 = false;
        }
        jTextField22.setEditable(false);
        jTextField22.setText("");
    }

    private void jRadioButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        Case3 = jRadioButton6.isEnabled();
        jTextField22.setEditable(true);
    }

    private void jRadioButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        if (jRadioButton7.isEnabled()){
            Case4 = false;
        }
        jTextField23.setEditable(false);
        jTextField23.setText("");
    }

    private void jRadioButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        Case4 = jRadioButton8.isEnabled();
        jTextField23.setEditable(true);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        Name = jTextField1.getText();
        ID = jTextField2.getText();
        Birth = jTextField3.getText();
        Weight = jTextField4.getText();
        Height = jTextField5.getText();
        Team = jTextField6.getText();
        Address = jTextField7.getText();
        Subburb = jTextField8.getText();
        City = jTextField9.getText();
        State = jTextField10.getText();
        Country = jTextField11.getText();
        Telephone1 = jTextField12.getText();
        Email = jTextField13.getText();
        Contact = jTextField14.getText();
        Relationship = jTextField15.getText();
        Telephone2 = jTextField16.getText();
        Telephone3 = jTextField17.getText();
        Insurance = jTextField18.getText();
        Blood = jTextField19.getText();
        Medication = jTextField20.getText();
        Disease = jTextField21.getText();
        Familiar = jTextField22.getText();
        Surgery = jTextField23.getText();

        if ((Name.equals("                                                  "))
                ||(ID.equals("      "))||(Birth.equals("  /  /    "))||(Telephone1.equals("(  )    -          "))
                ||(Email.equals("                                                  "))
                ||(Contact.equals("                                                  "))
                ||(Relationship.equals("                                                  "))
                ||(Telephone2.equals("(  )    -          "))||(Blood.equals("     "))
                ||((Case1 == true)&&(Medication.equals("
")))
                ||((Case2 == true)&&(Disease.equals("
")))
                ||((Case3 == true)&&(Familiar.equals("
")))
                ||((Case4 == true)&&(Surgery.equals("
")))
                ||(Confirm == false)){

            JOptionPane.showMessageDialog(null, "Please, check all the required fields", "Error",
JOptionPane.ERROR_MESSAGE);

        }
        else {
```

```
      try {
        PreparedStatement pst;
        if (edit) {
          pst = ConnectSQL.Connect.prepareStatement("update athletes set birth = '" +Birth+ "', weight = '"
+Weight+ ""
              + "', height = '" +Height+ "', team = '" +Team+ "', address = '" +Address+ "', subburb = '"
+Subburb+ ""
              + "', city = '" +City+ "', state = '" +State+ "', country ='" +Country+ "', telephone1 = '"
+Telephone1+ ""
              + "', email = '" +Email+ "', contact = '" +Contact+ "', relationship = '" +Relationship+ "',
telephone2 = '" +Telephone2+ ""
              + "', telephone3 = '" +Telephone3+ "', insurance = '" +Insurance+ "', blood = '" +Blood+ "',
medication = '" +Medication+ ""
              + "', disease = '" +Disease+ "', familiar = '" +Familiar+ "', surgery = '" +Surgery+ "' where id = '"
+ID+ "'");
        }
        else  {
         pst = ConnectSQL.Connect.prepareStatement("insert into athletes ("
            + "name, id, birth, weight, height, team, address, subburb, city, state,"
            + "country, telephone1, email, contact, relationship, telephone2, telephone3,"
            + "insurance, blood, medication, disease, familiar, surgery) values (?, ?, ?"
            + ", ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
          pst.setString(1, Name);
          pst.setString(2, ID);
          pst.setString(3, Birth);
          pst.setString(4, Weight);
          pst.setString(5, Height);
          pst.setString(6, Team);
          pst.setString(7, Address);
          pst.setString(8, Subburb);
          pst.setString(9, City);
          pst.setString(10, State);
          pst.setString(11, Country);
          pst.setString(12, Telephone1);
          pst.setString(13, Email);
          pst.setString(14, Contact);
          pst.setString(15, Relationship);
          pst.setString(16, Telephone2);
          pst.setString(17, Telephone3);
          pst.setString(18, Insurance);
          pst.setString(19, Blood);
          pst.setString(20, Medication);
          pst.setString(21, Disease);
          pst.setString(22, Familiar);
          pst.setString(23, Surgery);
        }
        pst.executeUpdate();
        JOptionPane.showMessageDialog(null, "Operation Successfully Saved!");
      } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
      }

      if (edit) {
        Path = Path+ "" +Name+ " - " +ID+  ".xls";
        if(new File(Path+ ".xls").exists()) {
          ExcelRegister ExcelReg = new ExcelRegister(ID, Path, edit);
          try {
            ExcelReg.Excel();
```

```
                } catch (IOException ex) {
                    JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
                }
            }
        }

        ConnectSQL.Desconnection();
        this.dispose();
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    ConnectSQL.Desconnection();
    this.dispose();
}

private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    Confirm = jCheckBox1.isSelected();
}

/**
 * @param args the command line arguments
 */
public void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Register(edit, ID).setVisible(true);
        }
    });
}

private void EditRegister() {
    seek = false;
    jTextField1.setEditable(false);
    jTextField2.setEditable(false);

    try {
      String strSQL = "SELECT id, name, birth, weight, height, team, address,"
          + "subburb, city, state, country, telephone1, email, contact,"
          + "relationship, telephone2, telephone3, insurance, blood,"
          + "medication, disease, familiar, surgery FROM athletes";
      PreparedStatement pst = ConnectSQL.Connect.prepareStatement(strSQL);
      ResultSet result = pst.executeQuery();
      while (result.next() && !seek) {
        if (ID.equals(result.getString(1))) {
          jTextField2.setText(ID);
          jTextField1.setText(result.getString(2));
          jTextField3.setText(result.getString(3));
          jTextField4.setText(result.getString(4));
          jTextField5.setText(result.getString(5));
          jTextField6.setText(result.getString(6));
          jTextField7.setText(result.getString(7));
          jTextField8.setText(result.getString(8));
          jTextField9.setText(result.getString(9));
          jTextField10.setText(result.getString(10));
          jTextField11.setText(result.getString(11));
          jTextField12.setText(result.getString(12));
          jTextField13.setText(result.getString(13));
          jTextField14.setText(result.getString(14));
          jTextField15.setText(result.getString(15));
          jTextField16.setText(result.getString(16));
          jTextField17.setText(result.getString(17));
          jTextField18.setText(result.getString(18));
          jTextField19.setText(result.getString(19));
          jTextField20.setText(result.getString(20));
          if (!result.getString(20).equals("
")) {
            jTextField20.setEditable(true);
            jRadioButton2.setSelected(true);
          }
          jTextField21.setText(result.getString(21));
          if (!result.getString(21).equals("
")) {
            jTextField21.setEditable(true);
            jRadioButton4.setSelected(true);
          }
          jTextField22.setText(result.getString(22));
          if (!result.getString(22).equals("
")) {
            jTextField22.setEditable(true);
            jRadioButton6.setSelected(true);
          }
          jTextField23.setText(result.getString(23));
          if (!result.getString(23).equals("
")) {
            jTextField23.setEditable(true);
            jRadioButton8.setSelected(true);
          }
          seek = true;
        }
      }
    }
```

```
  catch (SQLException ex) {
     JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
  }
}

// Variables declaration - do not modify
private javax.swing.ButtonGroup DiseaseGroup;
private javax.swing.ButtonGroup FamiliarGroup;
private javax.swing.ButtonGroup MedicationGroup;
private javax.swing.ButtonGroup OperationGroup;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JCheckBox jCheckBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JRadioButton jRadioButton1;
private javax.swing.JRadioButton jRadioButton2;
private javax.swing.JRadioButton jRadioButton3;
private javax.swing.JRadioButton jRadioButton4;
private javax.swing.JRadioButton jRadioButton5;
private javax.swing.JRadioButton jRadioButton6;
private javax.swing.JRadioButton jRadioButton7;
private javax.swing.JRadioButton jRadioButton8;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
```

```
    private javax.swing.JTextField jTextField14;
    private javax.swing.JTextField jTextField15;
    private javax.swing.JTextField jTextField16;
    private javax.swing.JTextField jTextField17;
    private javax.swing.JTextField jTextField18;
    private javax.swing.JTextField jTextField19;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField20;
    private javax.swing.JTextField jTextField21;
    private javax.swing.JTextField jTextField22;
    private javax.swing.JTextField jTextField23;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    // End of variables declaration

}
```

6.9 – Java Options Code

```
package statisticaltaekwondo;
public class Options extends javax.swing.JFrame {

    /**
     * Creates new form Options
     * @param IDSQL
     * @param NameSQL
     */
    public Options(String IDSQL, String NameSQL) {
        initComponents();
        String ID = IDSQL;
        String Name = NameSQL;
        jTextField1.setText(Name);
        jTextField2.setText(ID);
    }

    private Options() {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated
methods, choose Tools | Templates.
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Options");

        jLabel1.setText("Name:");

        jTextField1.setEditable(false);
        jTextField1.setEnabled(false);

        jLabel2.setText("ID Number:");

        jTextField2.setEditable(false);
        jTextField2.setEnabled(false);

        jButton1.setText("New Acquisition");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("New Analysis");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jButton3.setText("Close");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(176, 176, 176)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(175, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextField1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 69,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2)
                    .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(34, 34, 34)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(31, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        Acquisition Window = new Acquisition(jTextField2.getText());
        Window.setVisible(true);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        Analysis Window = new Analysis(jTextField2.getText(), jTextField1.getText());
        Window.setVisible(true);
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Options.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Options.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Options.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Options.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Options().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}
```

6.10 – Java Acquisition Code

```
package statisticaltaekwondo;

import com.panamahitek.PanamaHitek_Arduino;
import Connection.ConnectSQL;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Calendar;
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;

public class Acquisition extends javax.swing.JFrame {

PanamaHitek_Arduino arduino = new PanamaHitek_Arduino();
ConnectSQL ConnectSQL = new ConnectSQL();
int Slot = 1;
boolean State = false;
double[] Sensor = {0, 0, 0, 0, 0, 0, 0, 0};
```

```
Calendar calendar = Calendar.getInstance();
String Mode = "Attack";
String[] Export = {"", "", "", "", "", "", "", "", ""};
String ID = null;
int Series = 2;
String SeriesStr = "";
boolean seekID = false;
boolean seekMode = false;
boolean Wait  = false;
String[] Info = {"", "", ""};

   public Acquisition(String IDSQL) {
      initComponents();
      ID = IDSQL;
      Model = (DefaultTableModel) jTable1.getModel();

      try {
         arduino.arduinoRXTX(arduino.NameSerialPortAt(arduino.getPortsAvailable()), 9600, Event);
      }
      catch (Exception ex) {
         JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
      }

      ConnectSQL.Conenction();
   }


SerialPortEventListener Event = new SerialPortEventListener() {

   @Override
   public void serialEvent(SerialPortEvent spe) {
      while(Wait) {
         if (arduino.printMessage().equals("New")) {
            Wait = false;
            Slot = 1;
         }
      }
      switch (Slot) {
         case 1:
            Sensor[0] = Double.parseDouble(arduino.printMessage());
            Slot = 2;
            break;
         case 2:
            Sensor[1] = Double.parseDouble(arduino.printMessage());
            Slot = 3;
            break;
         case 3:
            Sensor[2] = Double.parseDouble(arduino.printMessage());
            Slot = 4;
            break;
         case 4:
            Sensor[3] = Double.parseDouble(arduino.printMessage());
            Slot = 5;
            break;
         case 5:
            Sensor[4] = Double.parseDouble(arduino.printMessage());
            Slot = 6;
            break;
```

```
        case 6:
          Sensor[5] = Double.parseDouble(arduino.printMessage());
          Slot = 7;
          break;
        case 7:
          Sensor[6] = Double.parseDouble(arduino.printMessage());
          Slot = 8;
          break;
        case 8:
          Sensor[7] = Double.parseDouble(arduino.printMessage());
          Slot = 9;
          break;
        }
      }
    };

    DefaultTableModel Model;

    public void Update(){

      String hour = calendar.get(Calendar.HOUR_OF_DAY) + "";
      String minute = calendar.get(Calendar.MINUTE) + "";
      String second = calendar.get(Calendar.SECOND) + "";
      if (Integer.parseInt(hour)<10) {
        hour = "0" + hour;
      }
      if (Integer.parseInt(minute)<10) {
        minute = "0" + minute;
      }
      if (Integer.parseInt(second)<10) {
        second = "0" + second;
      }
      String Output = hour+ ":" +minute+ ":" +second;
      calendar = Calendar.getInstance();

      Model.addRow(new Object[]{""+Output, Sensor[0], Sensor[1], Sensor[2], Sensor[3],
        Sensor[4], Sensor[5], Sensor[6], Sensor[7]});

    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

      Select = new javax.swing.ButtonGroup();
      jScrollPane1 = new javax.swing.JScrollPane();
      jTable1 = new javax.swing.JTable();
      jButton1 = new javax.swing.JButton();
      jButton2 = new javax.swing.JButton();
      jButton3 = new javax.swing.JButton();
      jRadioButton1 = new javax.swing.JRadioButton();
      jRadioButton2 = new javax.swing.JRadioButton();
      jButton4 = new javax.swing.JButton();

      setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLOSE);
      setTitle("Acquisition");

      jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Time", "Sensor1", "Sensor2", "Sensor3", "Sensor4", "Sensor5", "Sensor6", "Sensor7", "Sensor8"
    }
));
jTable1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
jScrollPane1.setViewportView(jTable1);

jButton1.setText("Start Data Acquisition");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Export Acquisition");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setText("Clean Acquisition");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

Select.add(jRadioButton1);
jRadioButton1.setSelected(true);
jRadioButton1.setText("Attack");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});

Select.add(jRadioButton2);
jRadioButton2.setText("Defense");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton2ActionPerformed(evt);
    }
});

jButton4.setText("Close");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addComponent(jRadioButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(20, 20, 20))
      .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 736, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 360, Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jRadioButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jRadioButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap())
    );

    jRadioButton1.getAccessibleContext().setAccessibleParent(jButton2);
    jRadioButton2.getAccessibleContext().setAccessibleParent(jButton1);

    pack();
  }// </editor-fold>

  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if (State == true) {
      jButton1.setText("Start Data Acquisition");
      State = false;
      try {
        arduino.sendData("2");
        Wait = true;
      } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
      }
    } else {
      jButton1.setText("Interrupt Data Acquisition");
```

```
        State = true;
        try {
        arduino.sendData("1");
        } catch (Exception ex) {
           JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
        }
     }
  }

  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     if (Model.getRowCount()== 0) {
        JOptionPane.showMessageDialog(null, "Error, no one acquisition has been made");
     }
     else {
        jButton1.setText("Start Data Acquisition");
        State = false;
        try {
           arduino.sendData("2");
           Wait = true;
        } catch (Exception ex) {
           JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
        }
        try {
           Thread.sleep(5000);               //5000 milliseconds
        } catch(InterruptedException ex) {
           JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
        }
        try {
           Save();
           JOptionPane.showMessageDialog(null, "Operation Performed with Sucess! \n Date: "
                 +Info[0]+ "\n Mode: " +Info[1]+ "\n Series Number: " +Info[2]);
        } catch (Exception ex) {
           JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
        }
     }
  }

  private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
     Mode = "Attack";
  }

  private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     Mode = "Defense";
  }

  private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
     jButton1.setText("Start Data Acquisition");
     State = false;
     try {
        arduino.sendData("2");
        Wait = true;
     } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
     }
     try {
        Thread.sleep(5000);               //5000 milliseconds
     } catch(InterruptedException ex) {
        Thread.currentThread().interrupt();
```

```
    }

    try {
        Clear();
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
    }
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        arduino.killArduinoConnection();
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
    }
    this.dispose();
}

private void Clear() {
    int total = Model.getRowCount();
    for (int i = total; i > 0; i--){
        Model.removeRow(0);
    }
}

 private void Save() {
    Series = 2;
    SeriesStr = String.valueOf(Series);
    seekID = false;
    seekMode = false;
    String year = calendar.get(Calendar.YEAR) + "";
    String month = calendar.get(Calendar.MONTH) + "";
    String day = calendar.get(Calendar.DAY_OF_MONTH) + "";
    int month2 = Integer.parseInt(month)+1;
    if (month2<10) {
        month = "0" + month2;
        }
    else {
        month = "" + month2;
    }
    if (Integer.parseInt(day)<10) {
        day = "0" + day;
    }
    String Date = day+ "/" +month+ "/" +year;

    try {
            String strSQL = "SELECT id, mode, series FROM acquisition";
            PreparedStatement pst = ConnectSQL.Connect.prepareStatement(strSQL);
            ResultSet result = pst.executeQuery();
            while (result.next() && !seekID) {
                if (ID.equals(result.getString(1))) {
                    seekID = true;
                }
            }
            if (seekID == true) {
                while (result.next() && !seekMode) {
                    if (Mode.equals(result.getString(2))) {
                        seekMode = true;
```

```
               }
            }
            if (seekMode == false) {
               Series = 1;
               SeriesStr = String.valueOf(Series);
            }
            else {
               while (result.next()) {
                  if (ID.equals(result.getString(1)) && Mode.equals(result.getString(2))) {
                     Series = Integer.parseInt(result.getString(3))+1;
                     SeriesStr = String.valueOf(Series);
                  }
               }
            }
         }
         else {
            Series = 1;
            SeriesStr = String.valueOf(Series);
         }
      }
      catch (SQLException ex) {
         JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
      }

   int total = Model.getRowCount();
   for (int i = total; i > 0; i--){
      for (int j = 0; j < 9; j++){
         Export[j] = Model.getValueAt(0, j).toString();
      }
   try {
         PreparedStatement pst = ConnectSQL.Connect.prepareStatement("insert into acquisition ("
               + "id, date, mode, series, time, sensor1, sensor2, sensor3, sensor4, sensor5,"
               + " sensor6, sensor7, sensor8) values (?, ?, ?, ?, ?, ?, ?, ?, ?,"
               + "?, ?, ?, ?)");
         pst.setString(1, ID);
         pst.setString(2, Date);
         pst.setString(3, Mode);
         pst.setString(4, SeriesStr);
         pst.setString(5, Export[0]);
         pst.setString(6, Export[1]);
         pst.setString(7, Export[2]);
         pst.setString(8, Export[3]);
         pst.setString(9, Export[4]);
         pst.setString(10, Export[5]);
         pst.setString(11, Export[6]);
         pst.setString(12, Export[7]);
         pst.setString(13, Export[8]);
         pst.executeUpdate();

      } catch (SQLException ex) {
         JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
      }
      Model.removeRow(0);
   }
   Info[0] = Date;
   Info[1] = Mode;
   Info[2] = SeriesStr;
}
```

```
    /**
     * @param args the command line arguments
     */
    public void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Acquisition.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Acquisition.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Acquisition.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Acquisition.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        }
        //</editor-fold>
        //</editor-fold>
        java.awt.EventQueue.invokeLater(() -> {
            new Acquisition(ID).setVisible(true);
        });
    }


    // Variables declaration - do not modify
    private javax.swing.ButtonGroup Select;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    // End of variables declaration

}
```

6.11 – Java Analysis Code

```
package statisticaltaekwondo;
```

```
import Connection.ConnectSQL;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;

/**
 *
 * @author Douglas
 */
public class Analysis extends javax.swing.JFrame {
String ID = null;
String Name = null;
String Choose = "";
boolean Test1 = false;
boolean Test2 = false;
boolean Test3 = false;
boolean Test4 = false;
String Date = "";
String Date1 = "";
String Date2 = "";
String Date3 = "";
String Period = "";
String Series = "";
boolean Case1 = true;
boolean Case2 = false;
boolean Case3 = false;
Date FirstDate;
Date SecondDate;
int CountSeries = 0;
int CountSeries1 = 0;
int CountSeries2 = 0;
boolean Unique = false;
    /**
     * Creates new form Analysis
     * @param IDSQL
     * @param NameSQL
     */
    public Analysis(String IDSQL, String NameSQL) {
        initComponents();
        ID = IDSQL;
        Name = NameSQL;
        jRadioButton1.setEnabled(false);
        jRadioButton2.setEnabled(false);
        jRadioButton3.setEnabled(false);
        jComboBox1.setEnabled(false);
        jComboBox2.setEnabled(false);
        jComboBox3.setEnabled(false);
        jComboBox4.setEnabled(false);
        jComboBox5.setEnabled(false);
        jComboBox6.setEnabled(false);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        Analysis = new javax.swing.ButtonGroup();
        Mode = new javax.swing.ButtonGroup();
        jButton1 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox<>();
        jComboBox2 = new javax.swing.JComboBox<>();
        jComboBox3 = new javax.swing.JComboBox<>();
        jComboBox4 = new javax.swing.JComboBox<>();
        jComboBox5 = new javax.swing.JComboBox<>();
        jButton2 = new javax.swing.JButton();
        jRadioButton1 = new javax.swing.JRadioButton();
        jRadioButton2 = new javax.swing.JRadioButton();
        jRadioButton3 = new javax.swing.JRadioButton();
        jLabel1 = new javax.swing.JLabel();
        jRadioButton4 = new javax.swing.JRadioButton();
        jRadioButton5 = new javax.swing.JRadioButton();
        jLabel4 = new javax.swing.JLabel();
        jComboBox6 = new javax.swing.JComboBox<>();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Analysis");

        jButton1.setText("Close");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel3.setText("Choose the Day:");

        jLabel6.setText("Choose the First Day:");

        jLabel7.setText("Choose the Second Day:");

        jLabel8.setText("From Day:");

        jLabel9.setText("To Day:");

        jComboBox1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox1ActionPerformed(evt);
            }
        });

        jButton2.setText("OK");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

Analysis.add(jRadioButton1);
jRadioButton1.setSelected(true);
jRadioButton1.setText("Single Day Analysis");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});

Analysis.add(jRadioButton2);
jRadioButton2.setText("Two Day Analysis");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton2ActionPerformed(evt);
    }
});

Analysis.add(jRadioButton3);
jRadioButton3.setText("Period Analysis");
jRadioButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton3ActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel1.setText("Mode:");

Mode.add(jRadioButton4);
jRadioButton4.setText("Attack");
jRadioButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton4ActionPerformed(evt);
    }
});

Mode.add(jRadioButton5);
jRadioButton5.setText("Defense");
jRadioButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton5ActionPerformed(evt);
    }
});

jLabel4.setText("Choose the Series:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                        .addGap(0, 152, Short.MAX_VALUE)
                        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(114, 114, 114))
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jRadioButton2)
                            .addGroup(layout.createSequentialGroup()
                                .addGap(97, 97, 97)
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                    .addComponent(jLabel7)
                                    .addComponent(jLabel6)
                                    .addComponent(jLabel3)
                                    .addComponent(jLabel9)
                                    .addComponent(jLabel4)
                                    .addComponent(jLabel8))
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(jComboBox3, 0, 180, Short.MAX_VALUE)
                                    .addComponent(jComboBox2, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                    .addComponent(jComboBox1, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                    .addComponent(jComboBox4, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                    .addComponent(jComboBox5, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                    .addComponent(jComboBox6, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
                            .addComponent(jRadioButton1))
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(155, 155, 155)
                        .addComponent(jLabel1)
                        .addGap(2, 2, 2)
                        .addComponent(jRadioButton4)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jRadioButton5))
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jRadioButton3)))
                .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(jRadioButton4)
                    .addComponent(jRadioButton5))
                .addGap(18, 18, 18)
        .addComponent(jRadioButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 10, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel4)
          .addComponent(jComboBox6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jRadioButton2)
        .addGap(10, 10, 10)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jComboBox2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jLabel6, javax.swing.GroupLayout.Alignment.TRAILING))
        .addGap(10, 10, 10)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jComboBox3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jLabel7))
        .addGap(18, 18, 18)
        .addComponent(jRadioButton3)
        .addGap(11, 11, 11)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel8)
          .addComponent(jComboBox4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(5, 5, 5)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jComboBox5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jLabel9))
        .addGap(25, 25, 25)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

    pack();
  }// </editor-fold>

  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
  }

  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    SimpleDateFormat DateForm = new SimpleDateFormat("dd/MM/yyyy");
    Unique = false;

    if (Test1 || Test2 || Test3) {
      if (Case1) {
```

```
      Date = jComboBox1.getSelectedItem().toString();
      if (jComboBox6.getSelectedItem().toString().equals("All Series")) {
         Period = "Single-day Analysis on " +Date+ ", in: "
            +jComboBox6.getSelectedItem().toString();
         CountSeries = jComboBox6.getItemCount() - 1;
      }
      else {
         Period = "Single-day Analysis on " +jComboBox1.getSelectedItem().toString()+ ", in the Serie: "
            +jComboBox6.getSelectedItem().toString();
         CountSeries = 1;
         Series = jComboBox6.getSelectedItem().toString();
         Unique = true;
      }
      Results Window = new Results(ID, Choose, Period, Date, CountSeries, Unique, Series);
      Window.setVisible(true);
   }
   if (Case2) {
      try {
         FirstDate = DateForm.parse(jComboBox2.getSelectedItem().toString());
         SecondDate = DateForm.parse(jComboBox3.getSelectedItem().toString());
      }
      catch (Exception ex) {
         JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
      }
      if (FirstDate.before(SecondDate)) {
         Test4 = false;
         CountSeries1 = 0;
         CountSeries2 = 0;
         try {
            ConnectSQL ConnectSQL = new ConnectSQL();
            ConnectSQL.Conenction();
            String strSQL = "SELECT id, date, mode, series FROM acquisition";
            PreparedStatement pst;
            pst = ConnectSQL.Connect.prepareStatement(strSQL);
            ResultSet result = pst.executeQuery();
            while (result.next() && !Test4) {
               if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
                     jComboBox2.getSelectedItem().toString().equals(result.getString(2))) {
                  Test4 = true;
                  Series = result.getString(4);
                  CountSeries1 = 1;
               }
            }
            while (result.next()) {
               if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
                    jComboBox2.getSelectedItem().toString().equals(result.getString(2)) &&
                    !(Series.equals(result.getString(4)))) {
                  Series = result.getString(4);
                  CountSeries1 = CountSeries1 + 1;
               }
            }
            Test4 = false;
            ResultSet result2 = pst.executeQuery();
            while (result2.next() && !Test4) {
               if (ID.equals(result2.getString(1)) && Choose.equals(result2.getString(3)) &&
                     jComboBox3.getSelectedItem().toString().equals(result2.getString(2))) {
                  Test4 = true;
                  Series = result2.getString(4);
               CountSeries2 = 1;
            }
         }
         while (result2.next()) {
            if (ID.equals(result2.getString(1)) && Choose.equals(result2.getString(3)) &&
                  jComboBox3.getSelectedItem().toString().equals(result2.getString(2)) &&
                  !(Series.equals(result2.getString(4)))) {
               Series = result2.getString(4);
               CountSeries2 = CountSeries2 + 1;
            }
         }
      }
      catch (SQLException ex) {
         JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
      }
      Date = jComboBox2.getSelectedItem().toString();
      Period = "Two-day Analysis on the first day: " +Date;
      Results Window1 = new Results(ID, Choose, Period, Date, CountSeries1, Unique, Series);
      Window1.setVisible(true);
      Date = jComboBox3.getSelectedItem().toString();
      Period = "Two-day Analysis on the second day: " +Date;
      Results Window2 = new Results(ID, Choose, Period, Date, CountSeries2, Unique, Series);
      Window2.setVisible(true);
   }
   else  {
      JOptionPane.showMessageDialog(null, "Error, dates must be in ascending order");
   }
}
if (Case3) {
   try {
      FirstDate = DateForm.parse(jComboBox4.getSelectedItem().toString());
      SecondDate = DateForm.parse(jComboBox5.getSelectedItem().toString());
   }
   catch (Exception ex) {
      JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
   }
   if (FirstDate.before(SecondDate)) {
      int First = jComboBox4.getSelectedIndex();
      int Count = jComboBox5.getSelectedIndex() - jComboBox4.getSelectedIndex();
      String[] Info = new String[Count+1];
      for (int i = 0; i < Count+1; i++) {
         Info[i] = jComboBox4.getItemAt(First + i);
      }
      try {
         ConnectSQL ConnectSQL = new ConnectSQL();
         ConnectSQL.Conenction();
         String strSQL = "SELECT id, date, mode, series FROM acquisition";
         PreparedStatement pst;
         pst = ConnectSQL.Connect.prepareStatement(strSQL);
         for (int i = 1; i < Count+2; i++) {
            Test4 = false;
            ResultSet result = pst.executeQuery();
            while (result.next() && !Test4) {
               if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
                        jComboBox4.getItemAt(First + i - 1).equals(result.getString(2))) {
                  Test4 = true;
                  Series = result.getString(4);
                  CountSeries = 1;
```

```
                     }
                  }
                  while (result.next()) {
                     if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
                           jComboBox4.getItemAt(First + i - 1).equals(result.getString(2)) &&
                           !(Series.equals(result.getString(4)))) {
                        Series = result.getString(4);
                        CountSeries = CountSeries + 1;
                     }
                  }
                  Date = Info[i-1];
                  Period = "Period Analysis on day " +i+ ": " +Date;
                  Results Window3 = new Results(ID, Choose, Period, Date, CountSeries, Unique, Series);
                  Window3.setVisible(true);
               }
            }
            catch (SQLException ex) {
               JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
            }
         }
         else  {
            JOptionPane.showMessageDialog(null, "Error, dates must be in ascending order");
         }
      }
   }
   else  {
      JOptionPane.showMessageDialog(null, "Error, you shall choose a Valid Mode");
   }
}

private void jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt) {
   Choose = "Attack";
   Verify();
}

private void jRadioButton5ActionPerformed(java.awt.event.ActionEvent evt) {
   Choose = "Defense";
   Verify();
}

private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
   Case1 = true;
   Case2 = false;
   Case3 = false;
}

private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   Case1 = false;
   Case2 = true;
   Case3 = false;
}

private void jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt) {
   Case1 = false;
   Case2 = false;
   Case3 = true;
}
```

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    jComboBox6.removeAllItems();
    jComboBox6.addItem("All Series");

    if (Test1 || Test2 || Test3) {
        Test4 = false;
            try {
                ConnectSQL ConnectSQL = new ConnectSQL();
                ConnectSQL.Conenction();
                String strSQL = "SELECT id, date, mode, series, time FROM acquisition";
                PreparedStatement pst;
                pst = ConnectSQL.Connect.prepareStatement(strSQL);
                ResultSet result = pst.executeQuery();
                while (result.next() && !Test4) {
                    if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
                                jComboBox1.getSelectedItem().toString().equals(result.getString(2))) {
                        Test4 = true;
                        Series = result.getString(4);
                        jComboBox6.addItem(Series);
                    }
                }
                while (result.next()) {
                    if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
                            jComboBox1.getSelectedItem().toString().equals(result.getString(2)) &&
                            !(Series.equals(result.getString(4)))) {
                        Series = result.getString(4);
                        jComboBox6.addItem(Series);
                    }
                }
            }
            catch (SQLException ex) {
                JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
            }
    }
}

/**
 * @param args the command line arguments
 */
public void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Analysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Analysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Analysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
      java.util.logging.Logger.getLogger(Analysis.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(() -> {
      new Analysis(ID, Name).setVisible(true);
    });
  }

  public void Verify() {
  Test1 = false;
  Test2 = false;
  Test3 = false;
  jRadioButton1.setSelected(true);
  jComboBox1.removeAllItems();
  jComboBox2.removeAllItems();
  jComboBox3.removeAllItems();
  jComboBox4.removeAllItems();
  jComboBox5.removeAllItems();
  jComboBox6.removeAllItems();

  try {
    ConnectSQL ConnectSQL = new ConnectSQL();
    ConnectSQL.Conenction();
    String strSQL = "SELECT id, date, mode FROM acquisition";
    PreparedStatement pst;
    pst = ConnectSQL.Connect.prepareStatement(strSQL);
    ResultSet result = pst.executeQuery();
    while (result.next() && !Test1) {
      if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3))) {
        Date1 = result.getString(2);
        Test1 = true;
      }
    }
    while (result.next() && !Test2) {
      if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
!(Date1.equals(result.getString(2)))) {
        Date2 = result.getString(2);
        Test2 = true;
      }
    }
    while (result.next() && !Test3) {
      if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
!(Date2.equals(result.getString(2)))) {
        Date3 = result.getString(2);
        Test3 = true;
      }
    }
    if (Test3) {
      JOptionPane.showMessageDialog(null, "All Analysis are allowed");
      jRadioButton1.setEnabled(true);
      jRadioButton2.setEnabled(true);
      jRadioButton3.setEnabled(true);
      jComboBox1.setEnabled(true);
      jComboBox2.setEnabled(true);
      jComboBox3.setEnabled(true);
```

```
            jComboBox4.setEnabled(true);
            jComboBox5.setEnabled(true);
            jComboBox6.setEnabled(true);
            jComboBox1.addItem(Date1);
            jComboBox2.addItem(Date1);
            jComboBox3.addItem(Date1);
            jComboBox4.addItem(Date1);
            jComboBox5.addItem(Date1);
            jComboBox1.addItem(Date2);
            jComboBox2.addItem(Date2);
            jComboBox3.addItem(Date2);
            jComboBox4.addItem(Date2);
            jComboBox5.addItem(Date2);
            jComboBox1.addItem(Date3);
            jComboBox2.addItem(Date3);
            jComboBox3.addItem(Date3);
            jComboBox4.addItem(Date3);
            jComboBox5.addItem(Date3);
            while (result.next()) {
                if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
!(Date3.equals(result.getString(2)))) {
                    Date3 = result.getString(2);
                    jComboBox1.addItem(Date3);
                    jComboBox2.addItem(Date3);
                    jComboBox3.addItem(Date3);
                    jComboBox4.addItem(Date3);
                    jComboBox5.addItem(Date3);
                }
            }
        }
        else if (Test2) {
            JOptionPane.showMessageDialog(null, "Single-day and Two-day Analysis are allowed");
            jRadioButton1.setEnabled(true);
            jRadioButton2.setEnabled(true);
            jRadioButton3.setEnabled(false);
            jComboBox1.setEnabled(true);
            jComboBox2.setEnabled(true);
            jComboBox3.setEnabled(true);
            jComboBox4.setEnabled(false);
            jComboBox5.setEnabled(false);
            jComboBox6.setEnabled(true);
            jComboBox1.addItem(Date1);
            jComboBox2.addItem(Date1);
            jComboBox3.addItem(Date1);
            jComboBox1.addItem(Date2);
            jComboBox2.addItem(Date2);
            jComboBox3.addItem(Date2);

        }
        else if (Test1) {
            JOptionPane.showMessageDialog(null, "Just Single-day Analysis is allowed");
            jRadioButton1.setEnabled(true);
            jRadioButton2.setEnabled(false);
            jRadioButton3.setEnabled(false);
            jComboBox1.setEnabled(true);
            jComboBox2.setEnabled(false);
            jComboBox3.setEnabled(false);
            jComboBox4.setEnabled(false);
            jComboBox5.setEnabled(false);
            jComboBox6.setEnabled(true);
            jComboBox1.addItem(Date1);
        }
        else {
            JOptionPane.showMessageDialog(null, "Sorry, no Analysis are allowed for this input");
            jRadioButton1.setEnabled(false);
            jRadioButton2.setEnabled(false);
            jRadioButton3.setEnabled(false);
            jComboBox1.setEnabled(false);
            jComboBox2.setEnabled(false);
            jComboBox3.setEnabled(false);
            jComboBox4.setEnabled(false);
            jComboBox5.setEnabled(false);
            jComboBox6.setEnabled(false);
        }
        }
        catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
        }
    }

    // Variables declaration - do not modify
    private javax.swing.ButtonGroup Analysis;
    private javax.swing.ButtonGroup Mode;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JComboBox<String> jComboBox3;
    private javax.swing.JComboBox<String> jComboBox4;
    private javax.swing.JComboBox<String> jComboBox5;
    private javax.swing.JComboBox<String> jComboBox6;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JRadioButton jRadioButton4;
    private javax.swing.JRadioButton jRadioButton5;
    // End of variables declaration
}
```

6.12 – Java Results Code

```
package statisticaltaekwondo;

import Connection.ConnectSQL;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.math.BigDecimal;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import org.apache.poi.ss.usermodel.Cell;
import java.io.FileInputStream;
import java.io.InputStream;
import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.ClientAnchor;
import org.apache.poi.ss.usermodel.CreationHelper;
import org.apache.poi.ss.usermodel.Drawing;
import org.apache.poi.ss.usermodel.Font;
import org.apache.poi.ss.usermodel.Picture;
import org.apache.poi.util.IOUtils;

public final class Results extends javax.swing.JFrame {
String ID;
String Choose;
String Period;
String Date;
String DateExcel;
String Name;
boolean Seek = false;
int CountSeries;
boolean Unique;
Date FirstTime;
Date SecondTime;
long TimeDifferenceSec;
String Seconds;
long TimeDifferenceMin;
String Minutes;
double Time;
boolean Test = false;
boolean Second = false;
String Series = "";
Integer Serie;
double[] CountSensor = {0, 0, 0, 0, 0, 0, 0, 0};
double[] CountSensorCrit = {0, 0, 0, 0, 0, 0, 0, 0};
double[] Percentage = {0, 0, 0, 0, 0, 0, 0, 0};
int Effective = 15;
int Hit = 5;
int answer;
int SheetCount;
String Path = "C:\\Users\\Douglas\\Desktop\\StatisticalTaekwondo\\";
String PathFile;
int width;
boolean edit = false;

    /**
     * Creates new form Results
     * @param IDStr
     * @param ChooseStr
```

```
  * @param PeriodStr
  * @param DateStr
  * @param Count
  * @param UniqueBL
  * @param SeriesStr
  */
 public Results(String IDStr, String ChooseStr, String PeriodStr, String DateStr, int Count, boolean UniqueBL,
String SeriesStr) {
    initComponents();
    ID = IDStr;
    Choose = ChooseStr;
    jTextField54.setText(Choose);
    Period = PeriodStr;
    jTextField1.setText(Period);
    Date = DateStr;
    CountSeries = Count;
    jTextField8.setText(String.valueOf(CountSeries));
    Unique = UniqueBL;
    Series = SeriesStr;
    GetName();
    Form();
    if (Integer.parseInt(String.valueOf(TimeDifferenceSec)) > 59) {
      TimeDifferenceMin = TimeDifferenceSec / 60;
      TimeDifferenceSec = TimeDifferenceSec - 60 * (TimeDifferenceSec / 60);
    }
    if (Integer.parseInt(String.valueOf(TimeDifferenceSec))<10) {
      Seconds = "0" + String.valueOf(TimeDifferenceSec);
    }
    else if (Integer.parseInt(String.valueOf(TimeDifferenceSec))<10) {
      Seconds = "0" + String.valueOf(TimeDifferenceSec);
    }
    else {
      Seconds = String.valueOf(TimeDifferenceSec);
    }
    if (Integer.parseInt(String.valueOf(TimeDifferenceMin))<10) {
      Minutes = "0" + String.valueOf(TimeDifferenceMin);
    }
    else {
      Minutes = String.valueOf(TimeDifferenceMin);
    }
    jTextField2.setText(Minutes+ ":" +Seconds);
    Time = Double.parseDouble(Minutes) + Double.parseDouble(Seconds)/60;;

    jTextField3.setText(String.valueOf(CountSensor[0]));
    BigDecimal round0 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[0]))/Time);
    round0 = round0.setScale(1, BigDecimal.ROUND_HALF_UP);
    jTextField4.setText(String.valueOf(round0));
    jTextField5.setText(String.valueOf(CountSensorCrit[0]));
    BigDecimal round1 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[0]))/Time);
    round1 = round1.setScale(1, BigDecimal.ROUND_HALF_UP);
    jTextField6.setText(String.valueOf(round1));
    BigDecimal round2 = new BigDecimal(Percentage[0]);
    round2 = round2.setScale(2, BigDecimal.ROUND_HALF_UP);
    jTextField7.setText(String.valueOf(round2) +"%");

    jTextField9.setText(String.valueOf(CountSensor[1]));
    BigDecimal round3 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[1]))/Time);
    round3 = round3.setScale(1, BigDecimal.ROUND_HALF_UP);
```

```
jTextField10.setText(String.valueOf(round3));
jTextField11.setText(String.valueOf(CountSensorCrit[1]));
BigDecimal round4 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[1]))/Time);
round4 = round4.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField12.setText(String.valueOf(round4));
BigDecimal round5 = new BigDecimal(Percentage[1]);
round5 = round5.setScale(2, BigDecimal.ROUND_HALF_UP);
jTextField13.setText(String.valueOf(round5) +"%");

jTextField14.setText(String.valueOf(CountSensor[2]));
BigDecimal round6 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[2]))/Time);
round6 = round6.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField15.setText(String.valueOf(round6));
jTextField16.setText(String.valueOf(CountSensorCrit[2]));
BigDecimal round7 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[2]))/Time);
round7 = round7.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField17.setText(String.valueOf(round7));
BigDecimal round8 = new BigDecimal(Percentage[2]);
round8 = round8.setScale(2, BigDecimal.ROUND_HALF_UP);
jTextField18.setText(String.valueOf(round8) +"%");

jTextField19.setText(String.valueOf(CountSensor[3]));
BigDecimal round9 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[3]))/Time);
round9 = round9.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField20.setText(String.valueOf(round9));
jTextField21.setText(String.valueOf(CountSensorCrit[3]));
BigDecimal round10 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[3]))/Time);
round10 = round10.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField22.setText(String.valueOf(round10));
BigDecimal round11 = new BigDecimal(Percentage[3]);
round11 = round11.setScale(2, BigDecimal.ROUND_HALF_UP);
jTextField23.setText(String.valueOf(round11) +"%");

jTextField24.setText(String.valueOf(CountSensor[4]));
BigDecimal round12 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[4]))/Time);
round12 = round12.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField26.setText(String.valueOf(round12));
jTextField25.setText(String.valueOf(CountSensorCrit[4]));
BigDecimal round13 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[4]))/Time);
round13 = round13.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField27.setText(String.valueOf(round13));
BigDecimal round14 = new BigDecimal(Percentage[4]);
round14 = round14.setScale(2, BigDecimal.ROUND_HALF_UP);
jTextField28.setText(String.valueOf(round14) +"%");

jTextField55.setText(String.valueOf(CountSensor[5]));
BigDecimal round15 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[5]))/Time);
round15 = round15.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField30.setText(String.valueOf(round15));
jTextField31.setText(String.valueOf(CountSensorCrit[5]));
BigDecimal round16 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[5]))/Time);
round16 = round16.setScale(1, BigDecimal.ROUND_HALF_UP);
jTextField32.setText(String.valueOf(round16));
BigDecimal round17 = new BigDecimal(Percentage[5]);
round17 = round17.setScale(2, BigDecimal.ROUND_HALF_UP);
jTextField33.setText(String.valueOf(round17) +"%");

jTextField34.setText(String.valueOf(CountSensor[6]));
```

```
    BigDecimal round18 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[6]))/Time);
    round18 = round18.setScale(1, BigDecimal.ROUND_HALF_UP);
    jTextField35.setText(String.valueOf(round18));
    jTextField36.setText(String.valueOf(CountSensorCrit[6]));
    BigDecimal round19 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[6]))/Time);
    round19 = round19.setScale(1, BigDecimal.ROUND_HALF_UP);
    jTextField37.setText(String.valueOf(round19));
    BigDecimal round20 = new BigDecimal(Percentage[6]);
    round20 = round20.setScale(2, BigDecimal.ROUND_HALF_UP);
    jTextField38.setText(String.valueOf(round20) +"%");

    jTextField39.setText(String.valueOf(CountSensor[7]));
    BigDecimal round21 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensor[7]))/Time);
    round21 = round21.setScale(1, BigDecimal.ROUND_HALF_UP);
    jTextField40.setText(String.valueOf(round21));
    jTextField41.setText(String.valueOf(CountSensorCrit[7]));
    BigDecimal round22 = new BigDecimal(Double.parseDouble(String.valueOf(CountSensorCrit[7]))/Time);
    round22 = round22.setScale(1, BigDecimal.ROUND_HALF_UP);
    jTextField42.setText(String.valueOf(round22));
    BigDecimal round23 = new BigDecimal(Percentage[7]);
    round23 = round23.setScale(2, BigDecimal.ROUND_HALF_UP);
    jTextField43.setText(String.valueOf(round23) +"%");

}


/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jToggleButton1 = new javax.swing.JToggleButton();
    jButton1 = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    jLabel13 = new javax.swing.JLabel();
    jTextField9 = new javax.swing.JTextField();
    jTextField10 = new javax.swing.JTextField();
    jTextField11 = new javax.swing.JTextField();
    jTextField12 = new javax.swing.JTextField();
    jTextField13 = new javax.swing.JTextField();
    jPanel3 = new javax.swing.JPanel();
    jLabel14 = new javax.swing.JLabel();
```

```
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jPanel4 = new javax.swing.JPanel();
jLabel19 = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
jLabel23 = new javax.swing.JLabel();
jTextField19 = new javax.swing.JTextField();
jTextField20 = new javax.swing.JTextField();
jTextField21 = new javax.swing.JTextField();
jTextField22 = new javax.swing.JTextField();
jTextField23 = new javax.swing.JTextField();
jPanel5 = new javax.swing.JPanel();
jLabel24 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
jLabel27 = new javax.swing.JLabel();
jLabel28 = new javax.swing.JLabel();
jTextField24 = new javax.swing.JTextField();
jTextField25 = new javax.swing.JTextField();
jTextField26 = new javax.swing.JTextField();
jTextField27 = new javax.swing.JTextField();
jTextField28 = new javax.swing.JTextField();
jTextField14 = new javax.swing.JTextField();
jTextField15 = new javax.swing.JTextField();
jTextField16 = new javax.swing.JTextField();
jTextField17 = new javax.swing.JTextField();
jTextField18 = new javax.swing.JTextField();
jTextField3 = new javax.swing.JTextField();
jTextField4 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();
jTextField6 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jPanel8 = new javax.swing.JPanel();
jLabel39 = new javax.swing.JLabel();
jLabel40 = new javax.swing.JLabel();
jLabel41 = new javax.swing.JLabel();
jLabel42 = new javax.swing.JLabel();
jLabel43 = new javax.swing.JLabel();
jPanel9 = new javax.swing.JPanel();
jLabel44 = new javax.swing.JLabel();
jLabel45 = new javax.swing.JLabel();
jLabel46 = new javax.swing.JLabel();
jLabel47 = new javax.swing.JLabel();
jLabel48 = new javax.swing.JLabel();
jTextField39 = new javax.swing.JTextField();
jTextField40 = new javax.swing.JTextField();
jTextField41 = new javax.swing.JTextField();
jTextField42 = new javax.swing.JTextField();
jTextField43 = new javax.swing.JTextField();
jTextField34 = new javax.swing.JTextField();
jTextField35 = new javax.swing.JTextField();
```

```
jTextField36 = new javax.swing.JTextField();
jTextField37 = new javax.swing.JTextField();
jTextField38 = new javax.swing.JTextField();
jLabel34 = new javax.swing.JLabel();
jTextField8 = new javax.swing.JTextField();
jLabel35 = new javax.swing.JLabel();
jTextField54 = new javax.swing.JTextField();
jPanel6 = new javax.swing.JPanel();
jLabel29 = new javax.swing.JLabel();
jLabel30 = new javax.swing.JLabel();
jLabel31 = new javax.swing.JLabel();
jLabel32 = new javax.swing.JLabel();
jLabel33 = new javax.swing.JLabel();
jTextField30 = new javax.swing.JTextField();
jTextField31 = new javax.swing.JTextField();
jTextField32 = new javax.swing.JTextField();
jTextField33 = new javax.swing.JTextField();
jTextField55 = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Results");

jLabel1.setText("Selected Period:");

jTextField1.setEditable(false);

jToggleButton1.setText("Close");
jToggleButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton1ActionPerformed(evt);
    }
});

jButton1.setText("Export to Excell");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel2.setText("Total Hits:");

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel4.setText("Hits Per Minute:");

jLabel5.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel5.setText("Effective Hits:");

jLabel7.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel7.setText("Effective Hits Per Minute:");

jLabel8.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel8.setText("Average Effective Hits:");

jLabel9.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel9.setText("Total Hits:");
```

```
        jLabel10.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel10.setText("Hits Per Minute:");

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel11.setText("Effective Hits:");

        jLabel12.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel12.setText("Effective Hits Per Minute:");

        jLabel13.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel13.setText("Average Effective Hits:");

        jTextField9.setEditable(false);
        jTextField9.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField10.setEditable(false);
        jTextField10.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField11.setEditable(false);
        jTextField11.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField12.setEditable(false);
        jTextField12.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField13.setEditable(false);
        jTextField13.setPreferredSize(new java.awt.Dimension(6, 15));

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel13)
                    .addComponent(jLabel12)
                    .addComponent(jLabel10)
                    .addComponent(jLabel9)
                    .addComponent(jLabel11))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jTextField11, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addComponent(jTextField12, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(jTextField13, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel9)
          .addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel10)
          .addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel11)
          .addComponent(jTextField11, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel12)
          .addComponent(jTextField12, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel13)
          .addComponent(jTextField13, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
    );

    jLabel14.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel14.setText("Total Hits:");

    jLabel15.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel15.setText("Hits Per Minute:");

    jLabel16.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel16.setText("Effective Hits:");

    jLabel17.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel17.setText("Effective Hits Per Minute:");

    jLabel18.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel18.setText("Average Effective Hits:");

    jLabel19.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel19.setText("Total Hits:");

    jLabel20.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel20.setText("Hits Per Minute:");

    jLabel21.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel21.setText("Effective Hits:");

    jLabel22.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel22.setText("Effective Hits Per Minute:");

    jLabel23.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel23.setText("Average Effective Hits:");

    jTextField19.setEditable(false);
    jTextField19.setPreferredSize(new java.awt.Dimension(6, 15));
    jTextField20.setEditable(false);
    jTextField20.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField21.setEditable(false);
    jTextField21.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField22.setEditable(false);
    jTextField22.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField23.setEditable(false);
    jTextField23.setPreferredSize(new java.awt.Dimension(6, 15));

    javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
    jPanel4.setLayout(jPanel4Layout);
    jPanel4Layout.setHorizontalGroup(
      jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jLabel23)
          .addComponent(jLabel22)
          .addComponent(jLabel20)
          .addComponent(jLabel19)
          .addComponent(jLabel21))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jTextField19, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jTextField20, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jTextField21, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField22, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))
          .addComponent(jTextField23, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );
    jPanel4Layout.setVerticalGroup(
      jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel4Layout.createSequentialGroup()
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel19)
          .addComponent(jTextField19, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel20)
          .addComponent(jTextField20, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel21)
          .addComponent(jTextField21, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel22)
          .addComponent(jTextField22, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel23)
          .addComponent(jTextField23, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
    );

    jLabel24.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel24.setText("Total Hits:");

    jLabel25.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel25.setText("Hits Per Minute:");

    jLabel26.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel26.setText("Effective Hits:");

    jLabel27.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel27.setText("Effective Hits Per Minute:");

    jLabel28.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
    jLabel28.setText("Average Effective Hits:");

    jTextField24.setEditable(false);
    jTextField24.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField25.setEditable(false);
    jTextField25.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField26.setEditable(false);
    jTextField26.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField27.setEditable(false);
    jTextField27.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField28.setEditable(false);
    jTextField28.setPreferredSize(new java.awt.Dimension(6, 15));

    javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
    jPanel5.setLayout(jPanel5Layout);
    jPanel5Layout.setHorizontalGroup(
      jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel5Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jLabel28)
          .addComponent(jLabel27)
          .addComponent(jLabel25)
          .addComponent(jLabel24)
          .addComponent(jLabel26))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jTextField28, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
          .addComponent(jTextField24, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField25, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField26, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField27, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(220, Short.MAX_VALUE))
    );
    jPanel5Layout.setVerticalGroup(
      jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel5Layout.createSequentialGroup()
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel24)
          .addComponent(jTextField24, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel25)
          .addComponent(jTextField26, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel26)
          .addComponent(jTextField25, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel27)
          .addComponent(jTextField27, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel28)
          .addComponent(jTextField28, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(3, 3, 3))
    );

    jTextField14.setEditable(false);
    jTextField14.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField15.setEditable(false);
    jTextField15.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField16.setEditable(false);
    jTextField16.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField17.setEditable(false);
    jTextField17.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField18.setEditable(false);
    jTextField18.setPreferredSize(new java.awt.Dimension(6, 15));

    javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);
    jPanel3Layout.setHorizontalGroup(
      jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel3Layout.createSequentialGroup()
        .addGap(0, 0, 0)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jLabel18)
          .addComponent(jLabel17)
          .addComponent(jLabel15)
          .addComponent(jLabel14)
          .addComponent(jLabel16))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jTextField18, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGroup(jPanel3Layout.createSequentialGroup()
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jTextField14, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jTextField15, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jTextField16, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jTextField17, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(24, Short.MAX_VALUE))
    );
    jPanel3Layout.setVerticalGroup(
      jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel3Layout.createSequentialGroup()
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel14)
          .addComponent(jTextField14, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel15)
          .addComponent(jTextField15, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel16)
          .addComponent(jTextField16, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel17)
          .addComponent(jTextField17, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel18)
          .addComponent(jTextField18, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
          .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        );

        jTextField3.setEditable(false);
        jTextField3.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField4.setEditable(false);
        jTextField4.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField5.setEditable(false);
        jTextField5.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField6.setEditable(false);
        jTextField6.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField7.setEditable(false);
        jTextField7.setPreferredSize(new java.awt.Dimension(6, 15));

        jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/statisticaltaekwondo/BodyProtector.jpg"))); // NOI18N

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
          jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                  .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(42, 42, 42)
                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                      .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                          .addComponent(jLabel4)
                          .addComponent(jLabel2))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                          .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
                          .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel5)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                  .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel7)
```

```
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addGap(9, 9, 9)
                                .addComponent(jLabel8)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(82, 82, 82)
                        .addComponent(jLabel3)))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel2)
                            .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel4)
                            .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel5))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel7)
                            .addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT_SIZE, 19, Short.MAX_VALUE)))
                    .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 372,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addGap(3, 3, 3))
);

jLabel6.setText("Total Time in Minutes:");

jTextField2.setEditable(false);

jLabel39.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel39.setText("Total Hits:");

jLabel40.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel40.setText("Hits Per Minute:");

jLabel41.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel41.setText("Effective Hits:");

jLabel42.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel42.setText("Effective Hits Per Minute:");

jLabel43.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel43.setText("Average Effective Hits:");

jLabel44.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel44.setText("Total Hits:");

jLabel45.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel45.setText("Hits Per Minute:");

jLabel46.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel46.setText("Effective Hits:");

jLabel47.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel47.setText("Effective Hits Per Minute:");

jLabel48.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
jLabel48.setText("Average Effective Hits:");

jTextField39.setEditable(false);
jTextField39.setPreferredSize(new java.awt.Dimension(6, 15));

jTextField40.setEditable(false);
jTextField40.setPreferredSize(new java.awt.Dimension(6, 15));

jTextField41.setEditable(false);
jTextField41.setPreferredSize(new java.awt.Dimension(6, 15));

jTextField42.setEditable(false);
jTextField42.setPreferredSize(new java.awt.Dimension(6, 15));

jTextField43.setEditable(false);
jTextField43.setPreferredSize(new java.awt.Dimension(6, 15));

javax.swing.GroupLayout jPanel9Layout = new javax.swing.GroupLayout(jPanel9);
jPanel9.setLayout(jPanel9Layout);
jPanel9Layout.setHorizontalGroup(
  jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
  .addGroup(jPanel9Layout.createSequentialGroup()
    .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel9Layout.createSequentialGroup()
              .addGap(52, 52, 52)
              .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel45)
                .addComponent(jLabel44)
                .addComponent(jLabel46)))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel9Layout.createSequentialGroup()
              .addContainerGap()
              .addComponent(jLabel47)))
          .addGroup(jPanel9Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel48)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jTextField39, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField40, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField41, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField42, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jTextField43, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(21, Short.MAX_VALUE))
    );
    jPanel9Layout.setVerticalGroup(
      jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel9Layout.createSequentialGroup()
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel44)
          .addComponent(jTextField39, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel45)
          .addComponent(jTextField40, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel46)
          .addComponent(jTextField41, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel47)
          .addComponent(jTextField42, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel48)
          .addComponent(jTextField43, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 11, Short.MAX_VALUE))
    );

    jTextField34.setEditable(false);
    jTextField34.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField35.setEditable(false);
    jTextField35.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField36.setEditable(false);
    jTextField36.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField37.setEditable(false);
    jTextField37.setPreferredSize(new java.awt.Dimension(6, 15));

    jTextField38.setEditable(false);
    jTextField38.setPreferredSize(new java.awt.Dimension(6, 15));

    javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);
    jPanel8.setLayout(jPanel8Layout);
    jPanel8Layout.setHorizontalGroup(
      jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel8Layout.createSequentialGroup()
        .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel8Layout.createSequentialGroup()
            .addGap(116, 116, 116)
            .addComponent(jLabel39)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jTextField34, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
          .addGroup(jPanel8Layout.createSequentialGroup()
            .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
              .addGroup(jPanel8Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(jLabel43)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextField38, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addGroup(jPanel8Layout.createSequentialGroup()
                .addContainerGap(56, Short.MAX_VALUE)
                .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                  .addGroup(jPanel8Layout.createSequentialGroup()
                    .addComponent(jLabel42)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextField37, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
                  .addGroup(jPanel8Layout.createSequentialGroup()
                    .addComponent(jLabel40)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextField35, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
                  .addGroup(jPanel8Layout.createSequentialGroup()
                    .addComponent(jLabel41)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextField36, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)))))
            .addGap(73, 73, 73)))
        .addComponent(jPanel9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(140, Short.MAX_VALUE))
    );
```

```
        jPanel8Layout.setVerticalGroup(
            jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel8Layout.createSequentialGroup()
                .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel8Layout.createSequentialGroup()
                        .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel39)
                            .addComponent(jTextField34, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(5, 5, 5)
                        .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel40)
                            .addComponent(jTextField35, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(6, 6, 6)
                        .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel41)
                            .addComponent(jTextField36, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel8Layout.createSequentialGroup()
                                .addGap(12, 12, 12)
                                .addComponent(jLabel42))
                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel8Layout.createSequentialGroup()
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(jTextField37, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel43)
                            .addComponent(jTextField38, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addComponent(jPanel9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(6, 6, 6))
        );

        jLabel34.setText("Total Number of Series:");

        jTextField8.setEditable(false);

        jLabel35.setText("Chosen Mode:");

        jTextField54.setEditable(false);

        jLabel29.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel29.setText("Total Hits:");

        jLabel30.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel30.setText("Hits Per Minute:");

        jLabel31.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel31.setText("Effective Hits:");

        jLabel32.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel32.setText("Effective Hits Per Minute:");
        jLabel33.setFont(new java.awt.Font("Tahoma", 0, 9)); // NOI18N
        jLabel33.setText("Average Effective Hits:");

        jTextField30.setEditable(false);
        jTextField30.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField31.setEditable(false);
        jTextField31.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField32.setEditable(false);
        jTextField32.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField33.setEditable(false);
        jTextField33.setPreferredSize(new java.awt.Dimension(6, 15));

        jTextField55.setEditable(false);
        jTextField55.setPreferredSize(new java.awt.Dimension(6, 15));

        javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
        jPanel6.setLayout(jPanel6Layout);
        jPanel6Layout.setHorizontalGroup(
            jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addContainerGap(142, Short.MAX_VALUE)
                .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel30, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel29, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel31, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel32, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel33, javax.swing.GroupLayout.Alignment.TRAILING))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jTextField55, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField30, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField31, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField32, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField33, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(46, 46, 46))
        );
        jPanel6Layout.setVerticalGroup(
            jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField55, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel29))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel30)
                    .addComponent(jTextField30, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jLabel31)
              .addComponent(jTextField31, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jLabel32)
              .addComponent(jTextField32, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jLabel33)
              .addComponent(jTextField33, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 131,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jToggleButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(29, 29, 29))
          .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 350,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel35)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jTextField54, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel34)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel6)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, 0)
                .addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(20, Short.MAX_VALUE))
          .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 1008,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jLabel1)
              .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel6)
              .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel34)
              .addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel35)
              .addComponent(jTextField54, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 46, Short.MAX_VALUE)
              .addComponent(jToggleButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        pack();
      }// </editor-fold>

      private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
      }

      private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

      PathFile = Path+ "" +Name+ " - " +ID+  ".xls";
      if(!new File(PathFile+ ".xls").exists()) {
        ExcelRegister ExcelReg = new ExcelRegister(ID, PathFile, edit);
        try {
          ExcelReg.Excel();
        } catch (IOException ex) {
          JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
        }
      }

      try {
        Excel(PathFile);
```

```
} catch (IOException ex) {
    JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
}
}

/**
 * @param args the command line arguments
 */
public void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Results(ID, Choose, Period, Date, CountSeries, Unique, Series).setVisible(true);
        }
    });
}

public void GetName() {
    try {
        ConnectSQL ConnectSQL = new ConnectSQL();
        ConnectSQL.Conenction();
        String strSQL = "SELECT id, name FROM athletes";
        PreparedStatement pst = ConnectSQL.Connect.prepareStatement(strSQL);
        ResultSet result = pst.executeQuery();
        while (result.next() && !Seek) {
            if (ID.equals(result.getString(1))) {
                Name = result.getString(2);
                Seek = true;
            }
        }
     }
    catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
    }
}
```

```
public void Form() {
  SimpleDateFormat DateForm = new SimpleDateFormat("HH:mm:ss");
  try {
    ConnectSQL ConnectSQL = new ConnectSQL();
    ConnectSQL.Conenction();
    String strSQL = "SELECT id, date, mode, series, time, sensor1, sensor2, sensor3, sensor4"
        + ", sensor5, sensor6, sensor7, sensor8 FROM acquisition";
    PreparedStatement pst;
    pst = ConnectSQL.Connect.prepareStatement(strSQL);
    ResultSet result = pst.executeQuery();
    if (Unique) {
      while (result.next()) {
        if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
            Date.equals(result.getString(2)) && Series.equals(result.getString(4))) {
          if (!Second) {
            try {
              FirstTime = DateForm.parse(result.getString(5));
            }
            catch (Exception ex) {
              JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
            }
          }
          Second = true;
          try {
            SecondTime = DateForm.parse(result.getString(5));
            }
          catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
          }

          for (int i = 0; i < 8; i++) {
            if (Double.parseDouble(String.valueOf(result.getString(i + 6))) > Effective) {
              CountSensorCrit[i] = CountSensorCrit[i] + 1;
            }
            if (Double.parseDouble(String.valueOf(result.getString(i + 6))) > Hit) {
              CountSensor[i] = CountSensor[i] + 1;
            }
          }
        }
      }
      TimeDifferenceSec = (SecondTime.getTime() - FirstTime.getTime()) / 1000 % 60;
      TimeDifferenceMin = (SecondTime.getTime() - FirstTime.getTime()) / (60 * 1000) % 60;
      }
      else {
        while (result.next()) {
          if (ID.equals(result.getString(1)) && Choose.equals(result.getString(3)) &&
              Date.equals(result.getString(2))) {
            if (!Test) {
              Series = result.getString(4);
              try {
                FirstTime = DateForm.parse(result.getString(5));
              }
              catch (Exception ex) {
                JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
              }
            }
            Test = true;
```

```
                  if (Series.equals(result.getString(4))) {
                     try {
                        SecondTime = DateForm.parse(result.getString(5));
                     }
                     catch (Exception ex) {
                        JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
                     }
                  }
                  for (int i = 0; i < 8; i++) {
                     if (Double.parseDouble(String.valueOf(result.getString(i + 6))) > Effective) {
                        CountSensorCrit[i] = CountSensorCrit[i] + 1;
                     }
                     if (Double.parseDouble(String.valueOf(result.getString(i + 6))) > Hit) {
                        CountSensor[i] = CountSensor[i] + 1;
                     }
                  }
               }
            }
            TimeDifferenceSec = (SecondTime.getTime() - FirstTime.getTime()) / 1000 % 60;
            TimeDifferenceMin = (SecondTime.getTime() - FirstTime.getTime()) / (60 * 1000) % 60;
            for (int i = 1; i < CountSeries; i++) {
               Second = false;
               Serie = Integer.parseInt(Series);
               Serie = Serie + 1;
               Series = String.valueOf(Serie);
               try {
                  ResultSet result2 = pst.executeQuery();
                  while (result2.next()) {
                     if (ID.equals(result2.getString(1)) && Choose.equals(result2.getString(3)) &&
                     Date.equals(result2.getString(2)) && Series.equals(result2.getString(4))) {
                           if (!Second) {
                              try {
                                 FirstTime = DateForm.parse(result2.getString(5));
                              }
                              catch (Exception ex) {
                                 JOptionPane.showMessageDialog(null, "Operation Error! \n Error: "
+ex.getMessage());
                              }
                           }
                           Second = true;
                           try {
                              SecondTime = DateForm.parse(result2.getString(5));
                           }
                        catch (Exception ex) {
                           JOptionPane.showMessageDialog(null, "Operation Error! \n Error: " +ex.getMessage());
                        }
                     }
                  }
               }
               catch (SQLException ex) {
                  JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
               }
               TimeDifferenceSec = TimeDifferenceSec + (SecondTime.getTime() - FirstTime.getTime()) / 1000 %
60;
               TimeDifferenceMin = TimeDifferenceMin + (SecondTime.getTime() - FirstTime.getTime()) / (60 *
1000) % 60;
            }
         }
```

```
      for (int i = 0; i < 8; i++) {
        Percentage[i] = (CountSensorCrit[i]/CountSensor[i]) * 100;
      }
    }
  catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
  }
}

  public void Excel(String input) throws FileNotFoundException, IOException {
    DateExcel = Date.replace( "/" , "|");
    FileInputStream workbookStream = new FileInputStream(PathFile+ ".xls");
    HSSFWorkbook workbook = new HSSFWorkbook(workbookStream);
    SheetCount = workbook.getNumberOfSheets();
    HSSFSheet worksheet = workbook.createSheet(SheetCount+1 + " - " +Choose+ " - " +DateExcel);
    worksheet.setZoom(68,100);   // 70 percent magnification

    Drawing patriarch = null;
    CreationHelper helper = workbook.getCreationHelper();
    ClientAnchor anchor = helper.createClientAnchor();
    InputStream inputStream = new FileInputStream(Path+ "BodyProtector.jpg");
    byte Image[] = IOUtils.toByteArray(inputStream);
    int indexImage = workbook.addPicture(Image, HSSFWorkbook.PICTURE_TYPE_PNG);
    inputStream.close();

    if (patriarch == null) {
      patriarch = worksheet.createDrawingPatriarch();
    }
    anchor.setAnchorType(2);
    int row = 9;
    int col = 0;
    anchor.setRow1(row);
    anchor.setCol1(col);
    anchor.setRow2(row + 1);
    anchor.setCol2(col + 1);
    Picture picture = patriarch.createPicture(anchor, indexImage);
    picture.resize();

    HSSFCellStyle style0 = workbook.createCellStyle();
    style0.setAlignment(HSSFCellStyle.ALIGN_CENTER);
    HSSFCellStyle style1 = workbook.createCellStyle();
    style1.setAlignment(HSSFCellStyle.ALIGN_RIGHT);

    HSSFCellStyle style2 = workbook.createCellStyle();
    Font font = workbook.createFont();
    font.setBoldweight(Font.BOLDWEIGHT_BOLD);
    font.setFontHeightInPoints((short) 13);
    style2.setFont(font);
    style2.setAlignment(HSSFCellStyle.ALIGN_RIGHT);

    HSSFCellStyle style3 = workbook.createCellStyle();
    Font font2 = workbook.createFont();
    font2.setFontHeightInPoints((short) 13);
    style3.setFont(font2);

    HSSFRow row1 = worksheet.createRow(1);
    Cell cell = row1.createCell(1);
    cell.setCellValue(jLabel1.getText());
cell.setCellStyle(style1);
cell.setCellStyle(style2);
cell = row1.createCell(2);
cell.setCellValue(jTextField1.getText());
cell.setCellStyle(style3);
width = (jTextField1.getText()+ "       ").length();
worksheet.setColumnWidth(2, width*250);
cell = row1.createCell(4);
cell.setCellValue(jLabel35.getText());
cell.setCellStyle(style1);
cell.setCellStyle(style2);
width = (jLabel35.getText()+ "  ").length();
worksheet.setColumnWidth(4, width*250);
cell = row1.createCell(5);
cell.setCellValue(jTextField54.getText());
cell.setCellStyle(style3);
width = (jTextField54.getText()+ "     ").length();
worksheet.setColumnWidth(5, width*250);
cell = row1.createCell(7);
cell.setCellValue(jLabel34.getText());
cell.setCellStyle(style1);
cell.setCellStyle(style2);
cell = row1.createCell(8);
cell.setCellValue(jTextField8.getText());
cell.setCellStyle(style3);
width = (jTextField2.getText()+ "  ").length();
worksheet.setColumnWidth(8, width*250);
cell = row1.createCell(11);
cell.setCellValue(jLabel6.getText());
cell.setCellStyle(style1);
cell.setCellStyle(style2);
cell = row1.createCell(12);
cell.setCellValue(jTextField2.getText());
cell.setCellStyle(style3);
width = (jTextField2.getText()+ "    ").length();
worksheet.setColumnWidth(12, width*250);

HSSFRow row2 = worksheet.createRow(4);
cell = row2.createCell(0);
cell.setCellValue(jLabel2.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(1);
cell.setCellValue(jTextField3.getText());
cell = row2.createCell(2);
cell.setCellValue(jLabel9.getText()+ " " +jTextField9.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel14.getText()+ " "+jTextField14.getText());
cell.setCellStyle(style0);
cell = row2.createCell(7);
cell.setCellValue(jLabel19.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(8);
cell.setCellValue(jTextField19.getText());
cell = row2.createCell(11);
cell.setCellValue(jLabel24.getText()+ " " +jTextField24.getText());
cell.setCellStyle(style0);
```

```
row2 = worksheet.createRow(5);
cell = row2.createCell(0);
cell.setCellValue(jLabel4.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(1);
cell.setCellValue(jTextField4.getText());
cell = row2.createCell(2);
cell.setCellValue(jLabel10.getText()+ " " +jTextField10.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel15.getText()+ " " +jTextField15.getText());
cell.setCellStyle(style0);
cell = row2.createCell(7);
cell.setCellValue(jLabel20.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(8);
cell.setCellValue(jTextField20.getText());
cell = row2.createCell(11);
cell.setCellValue(jLabel25.getText()+ " " +jTextField26.getText());
cell.setCellStyle(style0);

row2 = worksheet.createRow(6);
cell = row2.createCell(0);
cell.setCellValue(jLabel5.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(1);
cell.setCellValue(jTextField5.getText());
cell = row2.createCell(2);
cell.setCellValue(jLabel5.getText()+ " " +jTextField11.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel5.getText()+ " " +jTextField16.getText());
cell.setCellStyle(style0);
cell = row2.createCell(7);
cell.setCellValue(jLabel5.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(8);
cell.setCellValue(jTextField21.getText());
cell = row2.createCell(11);
cell.setCellValue(jLabel5.getText()+ " " +jTextField25.getText());
cell.setCellStyle(style0);

row2 = worksheet.createRow(7);
cell = row2.createCell(0);
cell.setCellValue(jLabel7.getText()+ " ");
width = (jLabel7.getText()+ "           ").length();
worksheet.setColumnWidth(0, width*250);
cell.setCellStyle(style1);
cell = row2.createCell(1);
cell.setCellValue(jTextField6.getText());
cell = row2.createCell(2);
cell.setCellValue(jLabel7.getText()+ " " +jTextField12.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel7.getText()+ " " +jTextField17.getText());
width = (jLabel7.getText()+ " " +jTextField17.getText()+ "         ").length();
worksheet.setColumnWidth(4, width*250);
cell.setCellStyle(style0);
```

```
cell = row2.createCell(7);
cell.setCellValue(jLabel7.getText()+ " ");
width = (jLabel7.getText()+ "         ").length();
worksheet.setColumnWidth(7, width*250);
cell.setCellStyle(style1);
cell = row2.createCell(8);
cell.setCellValue(jTextField22.getText());
cell = row2.createCell(11);
cell.setCellValue(jLabel7.getText()+ " " +jTextField27.getText());
width = (jLabel7.getText()+ " " +jTextField27.getText()+ " ").length();
worksheet.setColumnWidth(11, width*250);
cell.setCellStyle(style0);

row2 = worksheet.createRow(8);
cell = row2.createCell(0);
cell.setCellValue(jLabel8.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(1);
cell.setCellValue(jTextField7.getText());
cell = row2.createCell(2);
cell.setCellValue(jLabel8.getText()+ " " +jTextField13.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel8.getText()+ " " +jTextField18.getText());
cell.setCellStyle(style0);
cell = row2.createCell(7);
cell.setCellValue(jLabel8.getText()+ " ");
cell.setCellStyle(style1);
cell = row2.createCell(8);
cell.setCellValue(jTextField23.getText());
cell = row2.createCell(11);
cell.setCellValue(jLabel8.getText()+ " " +jTextField28.getText());
cell.setCellStyle(style0);

row2 = worksheet.createRow(43);
cell = row2.createCell(2);
cell.setCellValue(jLabel29.getText()+ " "+ jTextField55.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel29.getText()+ " " +jTextField34.getText());
cell.setCellStyle(style0);
cell = row2.createCell(7);
cell.setCellValue(jLabel29.getText()+ " " +jTextField39.getText());
cell.setCellStyle(style0);

row2 = worksheet.createRow(44);
cell = row2.createCell(2);
cell.setCellValue(jLabel39.getText()+ " " +jTextField30.getText());
cell.setCellStyle(style0);
cell = row2.createCell(4);
cell.setCellValue(jLabel39.getText()+ " " +jTextField35.getText());
cell.setCellStyle(style0);
cell = row2.createCell(7);
cell.setCellValue(jLabel39.getText()+ " " +jTextField40.getText());
cell.setCellStyle(style0);

row2 = worksheet.createRow(45);
cell = row2.createCell(2);
```

```
    cell.setCellValue(jLabel31.getText()+ " " +jTextField31.getText());
    cell.setCellStyle(style0);
    cell = row2.createCell(4);
    cell.setCellValue(jLabel31.getText()+ " " +jTextField36.getText());
    cell.setCellStyle(style0);
    cell = row2.createCell(7);
    cell.setCellValue(jLabel31.getText()+ " " +jTextField41.getText());
    cell.setCellStyle(style0);

    row2 = worksheet.createRow(46);
    cell = row2.createCell(2);
    cell.setCellValue(jLabel32.getText()+ " " +jTextField32.getText());
    cell.setCellStyle(style0);
    cell = row2.createCell(4);
    cell.setCellValue(jLabel32.getText()+ " " +jTextField37.getText());
    cell.setCellStyle(style0);
    cell = row2.createCell(7);
    cell.setCellValue(jLabel32.getText()+ " " +jTextField42.getText());
    cell.setCellStyle(style0);

    row2 = worksheet.createRow(47);
    cell = row2.createCell(2);
    cell.setCellValue(jLabel33.getText()+ " " +jTextField33.getText());
    cell.setCellStyle(style0);
    cell = row2.createCell(4);
    cell.setCellValue(jLabel33.getText()+ " " +jTextField38.getText());
    cell.setCellStyle(style0);
    cell = row2.createCell(7);
    cell.setCellValue(jLabel33.getText()+ " " +jTextField43.getText());
    cell.setCellStyle(style0);

    try (FileOutputStream FileOut = new FileOutputStream(PathFile+ ".xls")) {
        worksheet.protectSheet("5432");
        workbook.write(FileOut);
        FileOut.flush();
        JOptionPane.showMessageDialog(null, "Excel successfully updated!");
    }
    catch (FileNotFoundException ex) {
            JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
    } catch (IOException ex) {
            JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
    }

}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel39;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel40;
private javax.swing.JLabel jLabel41;
private javax.swing.JLabel jLabel42;
private javax.swing.JLabel jLabel43;
private javax.swing.JLabel jLabel44;
private javax.swing.JLabel jLabel45;
private javax.swing.JLabel jLabel46;
private javax.swing.JLabel jLabel47;
private javax.swing.JLabel jLabel48;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JPanel jPanel8;
private javax.swing.JPanel jPanel9;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;
private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField18;
private javax.swing.JTextField jTextField19;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField20;
private javax.swing.JTextField jTextField21;
private javax.swing.JTextField jTextField22;
private javax.swing.JTextField jTextField23;
```

```
    private javax.swing.JTextField jTextField24;
    private javax.swing.JTextField jTextField25;
    private javax.swing.JTextField jTextField26;
    private javax.swing.JTextField jTextField27;
    private javax.swing.JTextField jTextField28;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField30;
    private javax.swing.JTextField jTextField31;
    private javax.swing.JTextField jTextField32;
    private javax.swing.JTextField jTextField33;
    private javax.swing.JTextField jTextField34;
    private javax.swing.JTextField jTextField35;
    private javax.swing.JTextField jTextField36;
    private javax.swing.JTextField jTextField37;
    private javax.swing.JTextField jTextField38;
    private javax.swing.JTextField jTextField39;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField40;
    private javax.swing.JTextField jTextField41;
    private javax.swing.JTextField jTextField42;
    private javax.swing.JTextField jTextField43;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField54;
    private javax.swing.JTextField jTextField55;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    private javax.swing.JToggleButton jToggleButton1;
    // End of variables declaration

}
```

6.13 – Interaction between Java and Excel Code

```
package statisticaltaekwondo;

import Connection.ConnectSQL;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.io.FileInputStream;
import java.io.InputStream;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.util.HSSFColor;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellStyle;
import org.apache.poi.ss.usermodel.Font;
```

```
public class ExcelRegister {
  boolean Seek = false;
  String [] DataTable = {"", "", "", "", "", "", "", "", "", "", "", "", "", "",
  "", "", "", "", "", "", ""};
  int width;
  String ID;
  String Name;
  String Path;
  boolean edit = false;

  public ExcelRegister (String IDStr, String PathStr, boolean editBl) {
  ID = IDStr;
  Path = PathStr;
  edit = editBl;

  }

  public void Excel() throws IOException {
  try {
      ConnectSQL ConnectSQL = new ConnectSQL();
      ConnectSQL.Conenction();
      String strSQL = "SELECT id, name, birth, weight, height, team, address,"
          + "subburb, city, state, country, telephone1, email, contact,"
          + "relationship, telephone2, telephone3, insurance, blood,"
          + "medication, disease, familiar, surgery FROM athletes";
      PreparedStatement pst = ConnectSQL.Connect.prepareStatement(strSQL);
      ResultSet result = pst.executeQuery();
      while (result.next() && !Seek) {
        if (ID.equals(result.getString(1))) {
          Name = result.getString(2);
          DataTable[0] = result.getString(3);
          DataTable[1] = result.getString(4);
          DataTable[2] = result.getString(5);
          DataTable[3] = result.getString(6);
          DataTable[4] = result.getString(7);
          DataTable[5] = result.getString(8);
          DataTable[6] = result.getString(9);
          DataTable[7] = result.getString(10);
          DataTable[8] = result.getString(11);
          DataTable[9] = result.getString(12);
          DataTable[10] = result.getString(13);
          DataTable[11] = result.getString(14);
          DataTable[12] = result.getString(15);
          DataTable[13] = result.getString(16);
          DataTable[14] = result.getString(17);
          DataTable[15] = result.getString(18);
          DataTable[16] = result.getString(19);
          DataTable[17] = result.getString(20);
          DataTable[18] = result.getString(21);
          DataTable[19] = result.getString(22);
          DataTable[20] = result.getString(23);
          Seek = true;
        }
      }
     }
    catch (SQLException ex) {
      JOptionPane.showMessageDialog(null, "Error in Connect Server! \n Error: " +ex.getMessage());
    }
```

```
HSSFWorkbook workbook;
if (edit)  {
   InputStream excelFile = new FileInputStream(Path+ ".xls");
   workbook = new HSSFWorkbook(excelFile);
   workbook.removeSheetAt(0);
}
else  {
   workbook = new HSSFWorkbook();
}
HSSFSheet worksheet = workbook.createSheet("1 - Personal Form");
workbook.setSheetOrder("1 - Personal Form", 0);
int Last1 = 25;
int Last2 = 15;
HSSFRow row1 = worksheet.createRow(1);
HSSFRow row2 = worksheet.createRow(Last1);
HSSFCell cell1;
HSSFCell cell2;
HSSFCellStyle style0 = workbook.createCellStyle();
Font font = workbook.createFont();
font.setBoldweight(Font.BOLDWEIGHT_BOLD);
style0.setFont(font);
style0.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style0.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style1 = workbook.createCellStyle();
style1.setBorderTop(HSSFCellStyle.BORDER_MEDIUM);
style1.setTopBorderColor(HSSFColor.BLACK.index);
style1.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style1.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style2 = workbook.createCellStyle();
style2.setBorderRight(HSSFCellStyle.BORDER_MEDIUM);
style2.setRightBorderColor(HSSFColor.BLACK.index);
style2.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style2.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style3 = workbook.createCellStyle();
style3.setBorderLeft(HSSFCellStyle.BORDER_MEDIUM);
style3.setLeftBorderColor(HSSFColor.BLACK.index);
style3.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style3.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style4 = workbook.createCellStyle();
style4.setBorderBottom(HSSFCellStyle.BORDER_MEDIUM);
style4.setBottomBorderColor(HSSFColor.BLACK.index);
style4.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style4.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style5 = workbook.createCellStyle();
style5.setBorderTop(HSSFCellStyle.BORDER_MEDIUM);
style5.setTopBorderColor(HSSFColor.BLACK.index);
style5.setBorderLeft(HSSFCellStyle.BORDER_MEDIUM);
style5.setLeftBorderColor(HSSFColor.BLACK.index);
style5.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style5.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style6 = workbook.createCellStyle();
style6.setBorderTop(HSSFCellStyle.BORDER_MEDIUM);
style6.setTopBorderColor(HSSFColor.BLACK.index);
style6.setBorderRight(HSSFCellStyle.BORDER_MEDIUM);
style6.setRightBorderColor(HSSFColor.BLACK.index);
style6.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style6.setFillPattern(CellStyle.SOLID_FOREGROUND);
```

```
HSSFCellStyle style7 = workbook.createCellStyle();
style7.setBorderBottom(HSSFCellStyle.BORDER_MEDIUM);
style7.setBottomBorderColor(HSSFColor.BLACK.index);
style7.setBorderRight(HSSFCellStyle.BORDER_MEDIUM);
style7.setRightBorderColor(HSSFColor.BLACK.index);
style7.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style7.setFillPattern(CellStyle.SOLID_FOREGROUND);
HSSFCellStyle style8 = workbook.createCellStyle();
style8.setBorderBottom(HSSFCellStyle.BORDER_MEDIUM);
style8.setBottomBorderColor(HSSFColor.BLACK.index);
style8.setBorderLeft(HSSFCellStyle.BORDER_MEDIUM);
style8.setLeftBorderColor(HSSFColor.BLACK.index);
style8.setFillForegroundColor(HSSFColor.GREY_25_PERCENT.index);
style8.setFillPattern(CellStyle.SOLID_FOREGROUND);

cell1 = row1.createCell(1);
cell1.setCellStyle(style5);
cell1 = row1.createCell(Last2);
cell1.setCellStyle(style6);
cell2 = row2.createCell(1);
cell2.setCellStyle(style8);
cell2 = row2.createCell(Last2);
cell2.setCellStyle(style7);
for (int i = 2; i < Last2; i++) {
   cell1 = row1.createCell(i);
   cell1.setCellStyle(style1);
}
for (int i = 2; i < Last2; i++) {
   cell2 = row2.createCell(i);
   cell2.setCellStyle(style4);
}
HSSFRow row3;
HSSFRow row4;
HSSFRow row5;
HSSFCell cell3;
HSSFCell cell4;
HSSFCell cell5;
for (int i = 2; i < Last1; i++) {
   row3 = worksheet.createRow(i);
   row4 = worksheet.createRow(i);
   row5 = worksheet.createRow(i);
   cell3 = row3.createCell(1);
   cell3.setCellStyle(style3);
   cell4 = row4.createCell(Last2);
   cell4.setCellStyle(style2);
   for (int j = 2; j < Last2; j++) {
      cell5 = row5.createCell(j);
      cell5.setCellStyle(style0);
   }
}

Cell cell = worksheet.getRow(2).getCell(2);
cell.setCellValue("Personal");
cell = worksheet.getRow(4).getCell(2);
cell.setCellValue("Full Name:");
cell = worksheet.getRow(4).getCell(3);
cell.setCellValue(Name);
cell = worksheet.getRow(4).getCell(10);
```

```
cell.setCellValue("ID Number:");
cell = worksheet.getRow(4).getCell(11);
cell.setCellValue(ID);
cell = worksheet.getRow(5).getCell(2);
cell.setCellValue("Date of Birth:");
cell = worksheet.getRow(5).getCell(3);
cell.setCellValue(DataTable[0]);
cell = worksheet.getRow(5).getCell(5);
cell.setCellValue("Weigth:");
cell = worksheet.getRow(5).getCell(6);
cell.setCellValue(DataTable[1]);
cell = worksheet.getRow(5).getCell(7);
cell.setCellValue("Heigth:");
cell = worksheet.getRow(5).getCell(8);
cell.setCellValue(DataTable[2]);
cell = worksheet.getRow(5).getCell(10);
cell.setCellValue("Martial Arts Team:");
width = ("Martial Arts Team:  ").length();
worksheet.setColumnWidth(10, width*250);
cell = worksheet.getRow(5).getCell(11);
cell.setCellValue(DataTable[3]);
cell = worksheet.getRow(6).getCell(2);
cell.setCellValue("Address:");
cell = worksheet.getRow(6).getCell(3);
cell.setCellValue(DataTable[4]);
cell = worksheet.getRow(7).getCell(2);
cell.setCellValue("Subburb:");
cell = worksheet.getRow(7).getCell(3);
cell.setCellValue(DataTable[5]);
cell = worksheet.getRow(7).getCell(6);
cell.setCellValue("City:");
cell = worksheet.getRow(7).getCell(7);
cell.setCellValue(DataTable[6]);
cell = worksheet.getRow(7).getCell(9);
cell.setCellValue("State/Province:");
width = ("State/Province:  ").length();
worksheet.setColumnWidth(9, width*250);
cell = worksheet.getRow(7).getCell(10);
cell.setCellValue(DataTable[7]);
cell = worksheet.getRow(7).getCell(12);
cell.setCellValue("Country:");
cell = worksheet.getRow(7).getCell(13);
cell.setCellValue(DataTable[8]);
cell = worksheet.getRow(8).getCell(2);
cell.setCellValue("Telephone:");
cell = worksheet.getRow(8).getCell(3);
cell.setCellValue(DataTable[9]);
cell = worksheet.getRow(8).getCell(9);
cell.setCellValue("E-mail:");
cell = worksheet.getRow(8).getCell(10);
cell.setCellValue(DataTable[10]);

cell = worksheet.getRow(11).getCell(2);
cell.setCellValue("Contact");
cell = worksheet.getRow(13).getCell(2);
cell.setCellValue("Full Name:");
cell = worksheet.getRow(13).getCell(3);
cell.setCellValue(DataTable[11]);
```

```
cell = worksheet.getRow(13).getCell(10);
cell.setCellValue("Relationship:");
cell = worksheet.getRow(13).getCell(11);
cell.setCellValue(DataTable[12]);
cell = worksheet.getRow(14).getCell(2);
cell.setCellValue("First Telephone:");
cell = worksheet.getRow(14).getCell(3);
cell.setCellValue(DataTable[13]);
cell = worksheet.getRow(14).getCell(8);
cell.setCellValue("Second Telephone:");
width = ("Second Telephone:  ").length();
worksheet.setColumnWidth(8, width*250);
cell = worksheet.getRow(14).getCell(9);
cell.setCellValue(DataTable[14]);

cell = worksheet.getRow(17).getCell(2);
cell.setCellValue("Health");
cell = worksheet.getRow(19).getCell(2);
cell.setCellValue("Health Insurance:");
width = ("Health Insurance:  ").length();
worksheet.setColumnWidth(2, width*250);
cell = worksheet.getRow(19).getCell(3);
cell.setCellValue(DataTable[15]);
cell = worksheet.getRow(19).getCell(10);
cell.setCellValue("Blood Type:");
cell = worksheet.getRow(19).getCell(11);
cell.setCellValue(DataTable[16]);
cell = worksheet.getRow(20).getCell(2);
cell.setCellValue("Use of Medication:");
cell = worksheet.getRow(20).getCell(3);
if (DataTable[17].equals("                                                                  ")) {
   cell.setCellValue("No");
}
else {
   cell.setCellValue("Yes, " +DataTable[17]);
}
cell = worksheet.getRow(21).getCell(2);
cell.setCellValue("Chronic Disease:");
cell = worksheet.getRow(21).getCell(3);
if (DataTable[18].equals("                                                                  ")) {
   cell.setCellValue("No");
}
else {
   cell.setCellValue("Yes, " +DataTable[18]);
}
cell = worksheet.getRow(22).getCell(2);
cell.setCellValue("Familiar Disease:");
cell = worksheet.getRow(22).getCell(3);
if (DataTable[19].equals("                                                                  ")) {
   cell.setCellValue("No");
}
else {
   cell.setCellValue("Yes, " +DataTable[19]);
}
cell = worksheet.getRow(23).getCell(2);
cell.setCellValue("Recent Surgery:");
cell = worksheet.getRow(23).getCell(3);
if (DataTable[20].equals("                                                                  ")) {
```

```
            cell.setCellValue("No");
        }
        else {
            cell.setCellValue("Yes, " +DataTable[20]);
        }

        FileOutputStream fileOut = new FileOutputStream(new File(Path+ ".xls"));
        worksheet.protectSheet("5432");
        workbook.write(fileOut);
        fileOut.flush();

    }
}
```

6.14 – PostgreSQL Athletes Database Code

```
CREATE TABLE public.athletes
(
  name character varying(60) NOT NULL,
  id character varying(10) NOT NULL,
  birth character varying(10) NOT NULL,
  weight character varying(5),
  height character varying(5),
  team character varying(60),
  address character varying(120),
  subburb character varying(60),
  city character varying(60),
  state character varying(60),
  country character varying(60),
  telephone1 character varying(20) NOT NULL,
  email character varying(60) NOT NULL,
  contact character varying(60) NOT NULL,
  relationship character varying(60) NOT NULL,
  telephone2 character varying(20) NOT NULL,
  telephone3 character varying(20),
  insurance character varying(60),
  blood character varying(5) NOT NULL,
  medication character varying(120),
  disease character varying(120),
  familiar character varying(120),
  surgery character varying(120),
  CONSTRAINT athletes_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.athletes
  OWNER TO postgres;
```

6.15 – PostgreSQL Acquisition Database Code

```
CREATE TABLE public.acquisition
(
  "number" integer NOT NULL DEFAULT nextval('acquisition_number_seq'::regclass),
  id character varying(10) NOT NULL,
  date character varying(10),
  mode character varying(10),
  series character varying(10),
  "time" character varying(10),
  sensor1 character varying(10),
  sensor2 character varying(10),
  sensor3 character varying(10),
  sensor4 character varying(10),
  sensor5 character varying(10),
  sensor6 character varying(10),
  sensor7 character varying(10),
  sensor8 character varying(10),
  CONSTRAINT acquisition_pkey PRIMARY KEY (number)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.acquisition
  OWNER TO postgres;
```