

Software Design Document For iKarate: Improving Karate Kata

Bassel Emad, Omar Atef, Yehya Shams, Ahmed El-Kerdany
Supervised By: Dr. Ayman Ezzat, Dr. Ayman Nabil
and Eng. Nada Ayman

March 5, 2020

1 Introduction

1.1 Purpose

This Software Design Document purpose is to present fully detailed description of iKarate System Architecture, and to provide the purpose of implementing this application with full definition of functional requirements and showing the functionality of each component and their interaction together.

1.2 Scope

The scope of the system is to provide Karate coaches and players a system that will help them in the training and judging. For the coaches: the system helps them by providing the progress of their students in a report and the mistakes they made while performing. For the players: the system can be used to train without the coach by providing appropriate feedback that helps them master the movements.

1.3 Overview

The main goal of this project is to capture the moves of the performers in real time, analyse those moves and give them a feedback report to enhance their technique or alert them if they are performing a move or a stance incorrectly. One of the challenges we faced while comparing and analysing the captured motion, is that we should take into consideration that the activities might be performed with different speed, body proportions such as (Limbs length) and initial position of the students. Another challenge is the real-time feedback, giving the users a feedback and a report on their moves whether it was right or not in real-time is essentially important after the move is performed. The report includes tips on how to execute the move correctly the next time.

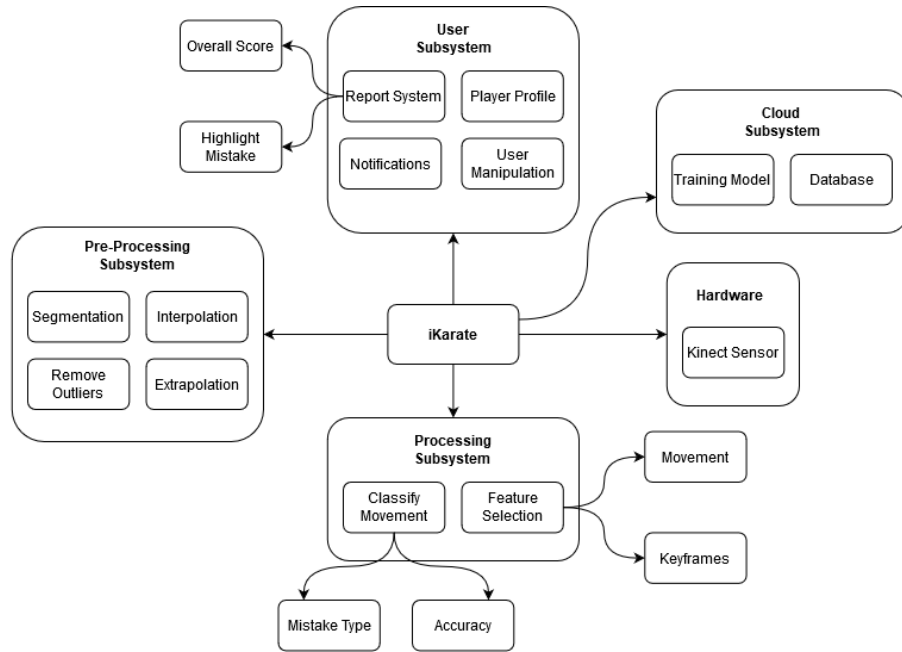


Figure 1: Block Diagram

2 System Overview

The system is composed of one or multiple Kinects. The Kinect(s) would be facing the user while he/she performs a sequence of moves. Then the frames and the skeleton are extracted from the Kinect, After that the pre-processing, enhancement, saving data on the cloud proceeds simultaneously with computing the key frames, feature extraction and finally the classification. After every move has been performed, the practitioner is presented with the move name and whether it was done correctly or not as shown in figure 2. The player is given a score to know how good their performance of the move after it has been analyzed. The score evaluation is based on the practitioners' motion while performing the move and their speed. Dynamic analysis of the movement gives real-time feedback and a report to the practitioner or the coach, making the application more interactive. The report contains the player name, age, weight, height, belt color, move name and duration, how well the player preformed the move, how to improve the user's performance and if any mistake were made it will be shown in the report.

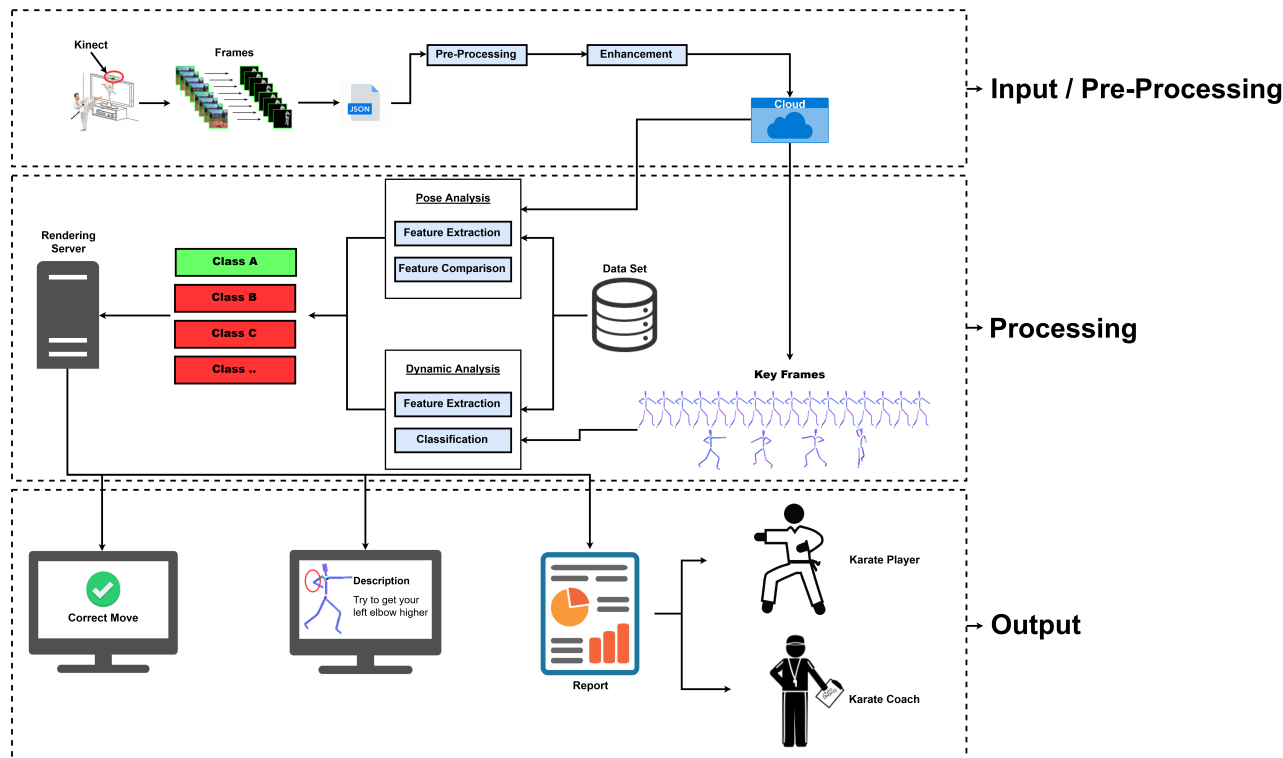


Figure 2: System Overview Diagram

3 System Architecture

3.1 Architectural Design

The system was designed to satisfy the MVC system architecture model based on the functional and non-functional requirements.

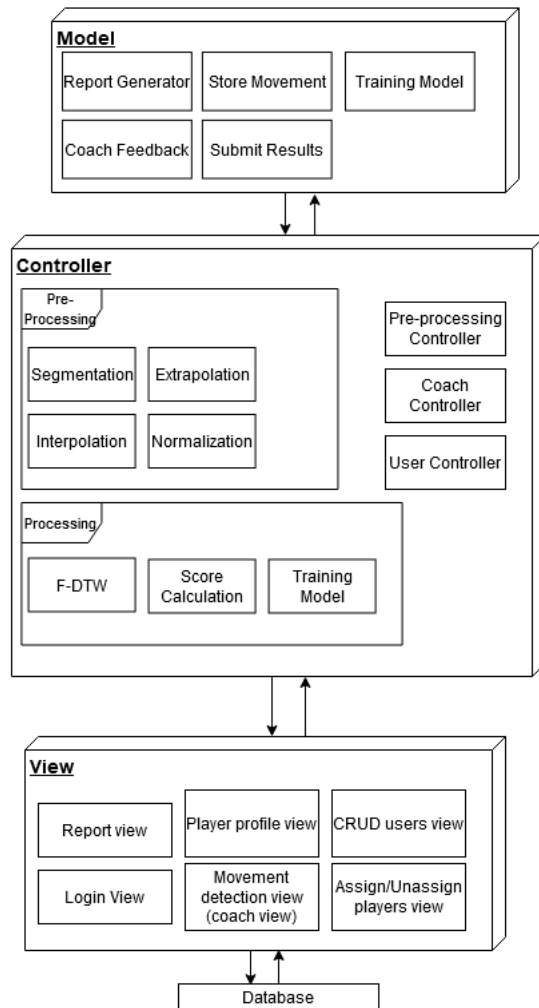


Figure 3: Architecture Diagram

3.1.1 Model

The model is the data used by the program. Model objects retrieve and store models state in the database. The model also contains the core of our application such as: classification, movement, report, pre-processing, player and coach.

3.1.2 View

The view is responsible for presenting the data in a User Interface. There are three different views, coach view is responsible for viewing the enrolled players and their progress, player view represents the functions and data available to

the player, and report view which is responsible for displaying the desired report from the report data generated by the system.

3.1.3 Controller

The Controller is responsible for rendering the appropriate view with the model data. The interactions and requests done in the view are handled by the database and the model and then data is sent back to the view to be shown to the user. The system overview contains three controllers, which are coach controller, player controller and report controller.

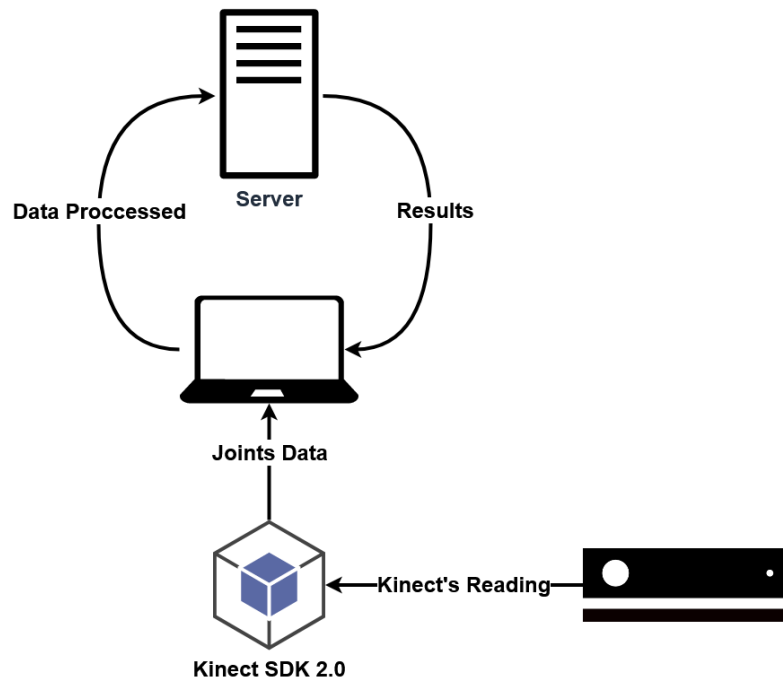


Figure 4: Hardware Architecture Diagram

3.2 Decomposition Description

3.2.1 Class Diagram

The whole system is based on the MVC design pattern which separates the application into three components, which are model, view and controller. The model corresponds to all the data-related logic. Model objects retrieve and store models state in a database. The View component is used for all the UI (User Interface) logic of the application. Controllers act as a bridge between the model and the view components to process all the business logic and incoming

requests, manipulate data using the model component and interact with the views to render the final output.

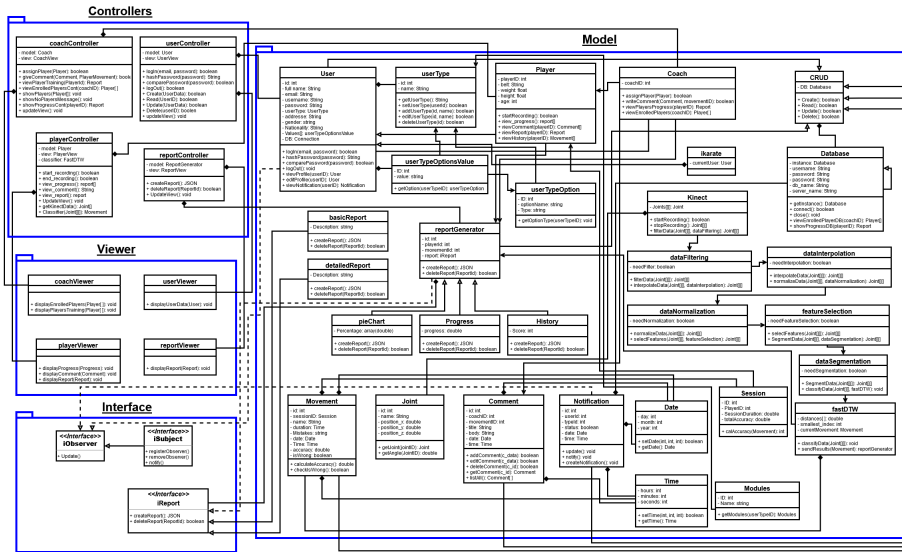


Figure 5: Class Diagram

3.2.2 Singleton Design Pattern

The single-tone design pattern is used to optimize the objects that is created in the system, specially database objects. The system will only need one database connection to avoid jamming the server. It is implemented by declaring the instance as a private static data member. Provide a public static member function that encapsulates all the initialization code and provides access to the instance.

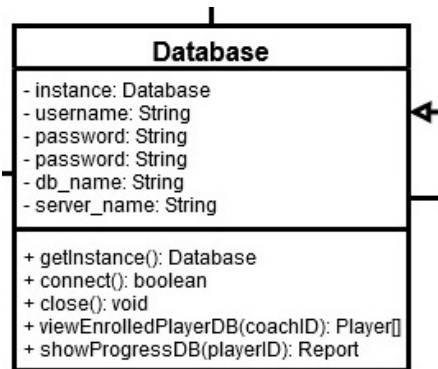


Figure 6: Singleton Design Pattern

3.2.3 Decorative Design Pattern

Decorator Design pattern acts as a wrapper to the existing class which allows the user to add new functionality and behavior to an existing object dynamically, without altering its structure. The system will use decorator design pattern in creating customized reports. So, each user can create a report with specific information that is needed.

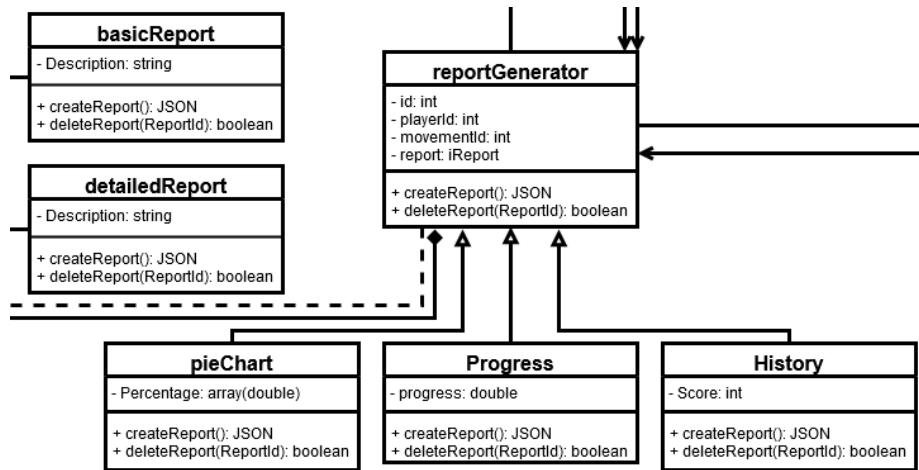


Figure 7: Decorative Design Pattern

3.2.4 Observer Design Pattern

Observer Design Pattern defines a one-to-many dependency between objects so that when one object changes its state, all its dependents are notified and updated automatically. It is used for the notification system, so that the users are always notified with the updates.

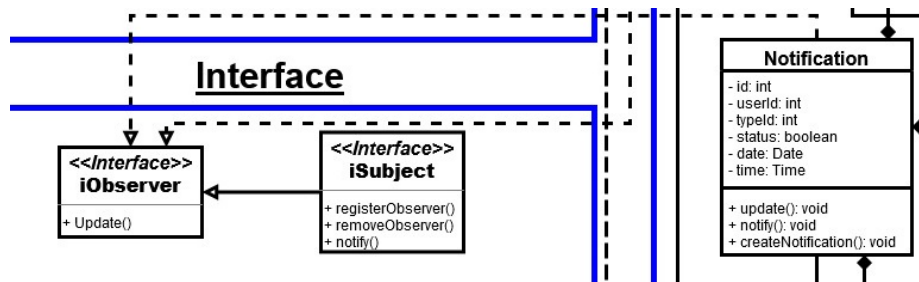


Figure 8: Observer Design Pattern

3.2.5 Activity Diagram

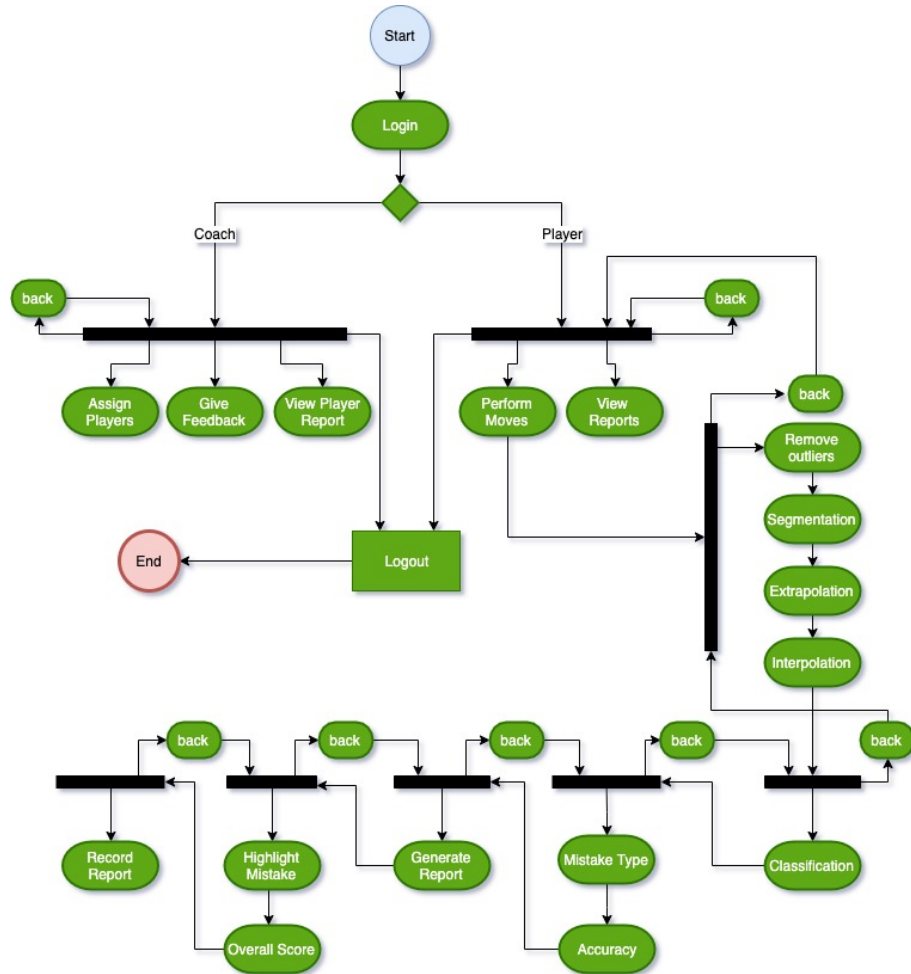


Figure 9: Activity Diagram

3.2.6 System Sequence Diagram

Recording, Pre-processing & Processing: As shown in figure 10, the player opens the application and start recording then when the player stops the recording, pre-processing and processing are done then the results are send to the player.

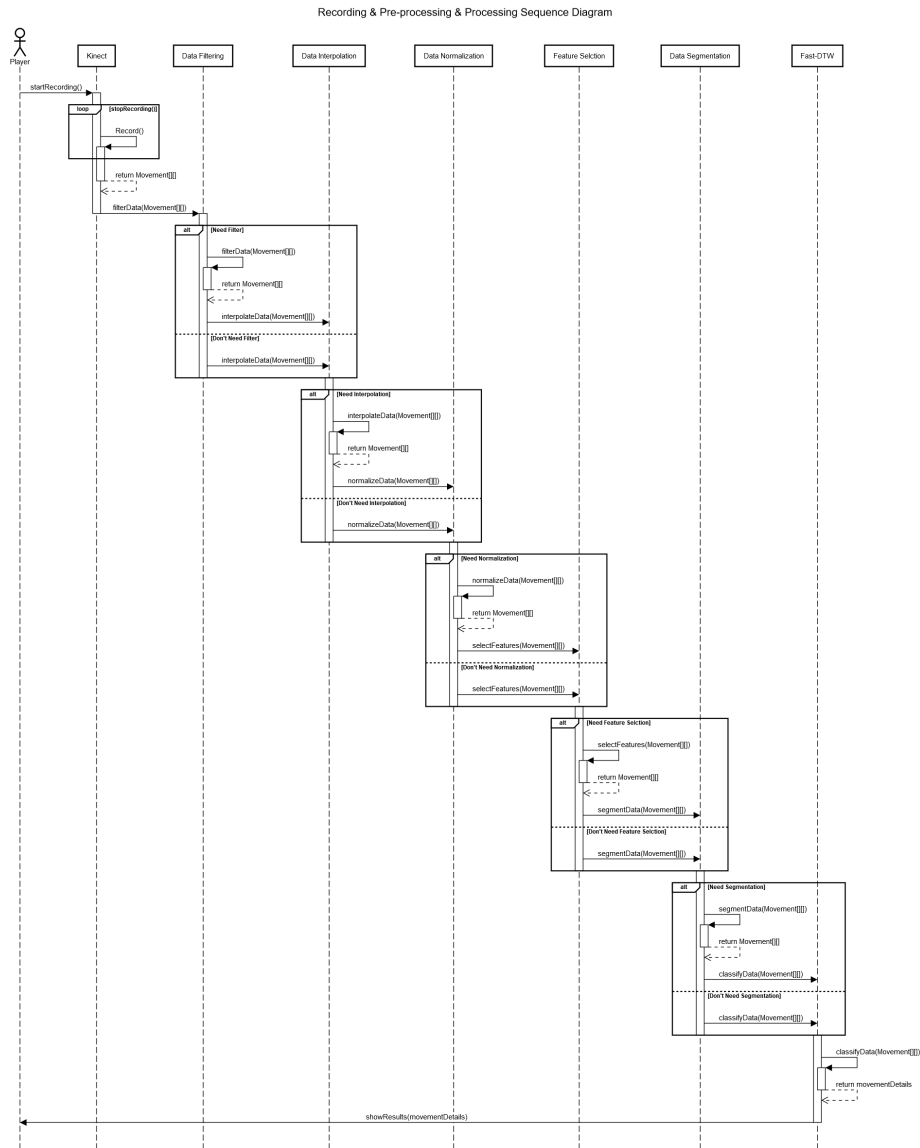


Figure 10: Pre-Processing & Processing Sequence Diagram

View History: As shown in figure 11, the user requests the history of a specific player which is retrieved from the database then the history is displayed on the screen.

View History Sequence Diagram

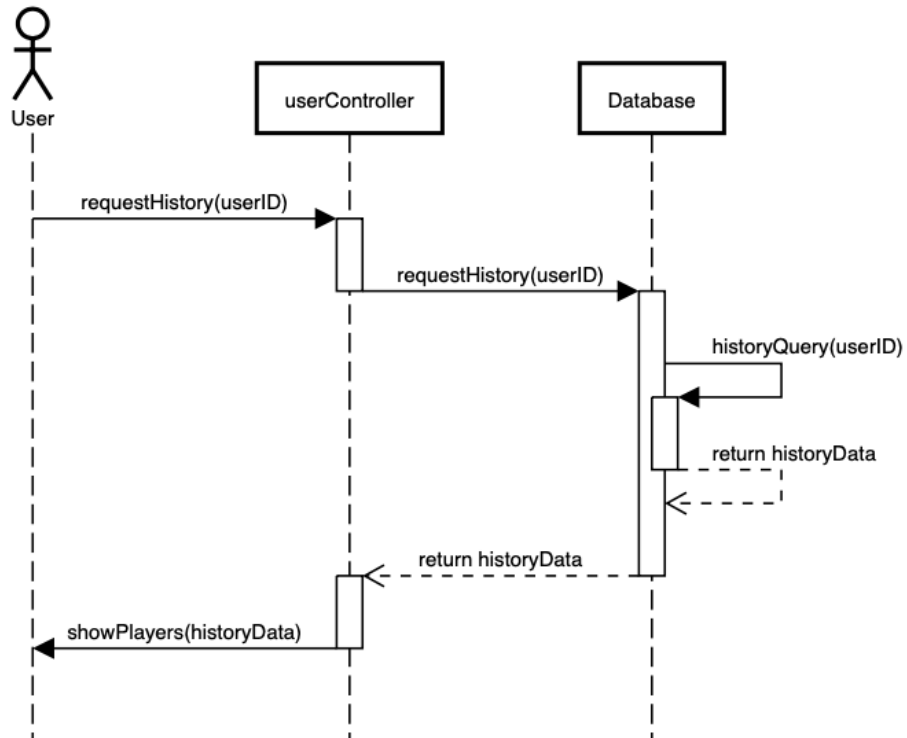


Figure 11: View History Sequence Diagram

Progress: As shown in figure 12, the coach request the progress of one of his students, the progress is retrieved from the database and displayed to the coach.

View Player's Progress Sequence Diagram

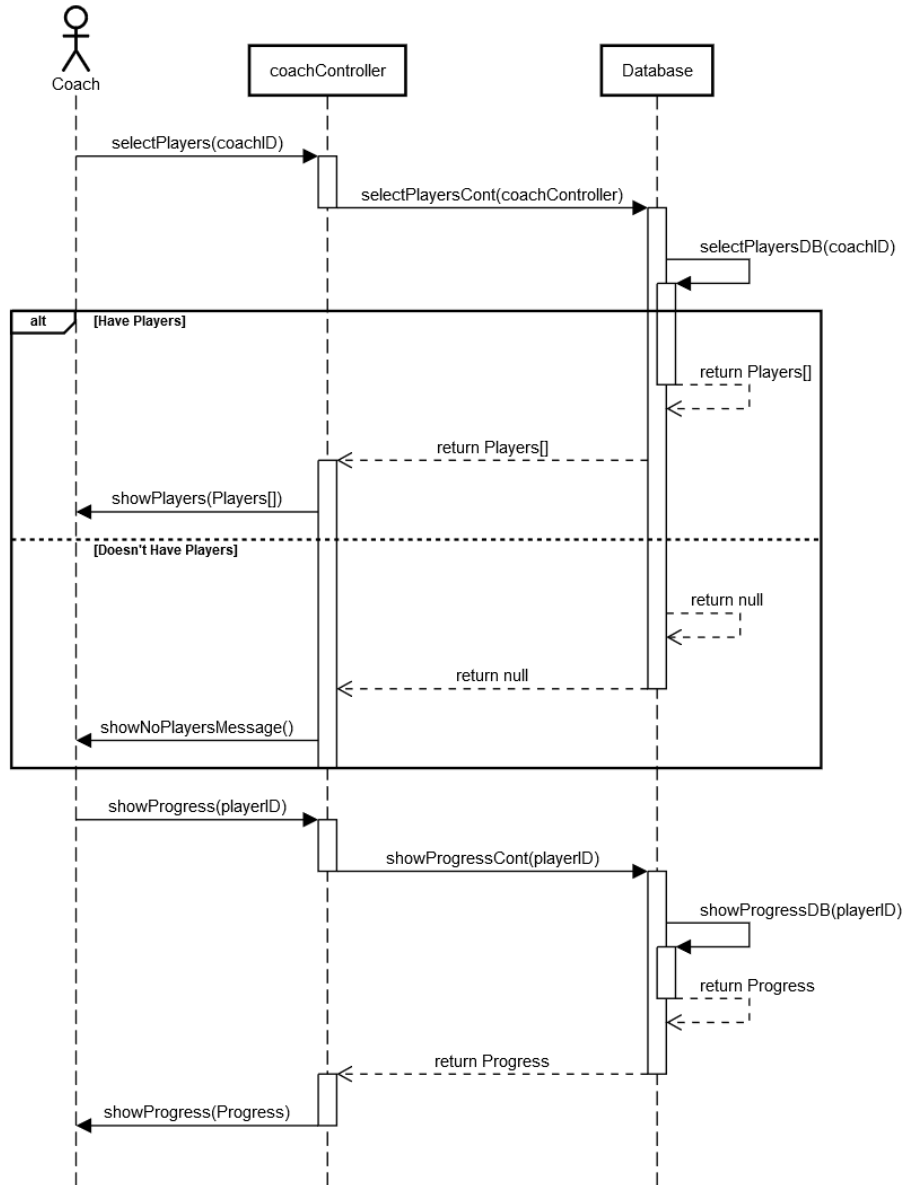


Figure 12: View Player's Progress Sequence Diagram

Assign Player: As shown in figure 13, the coach request a list of the un-assigned players which is retrieved from the database and then displayed to the coach, so that he chooses the players that he wants to assign.

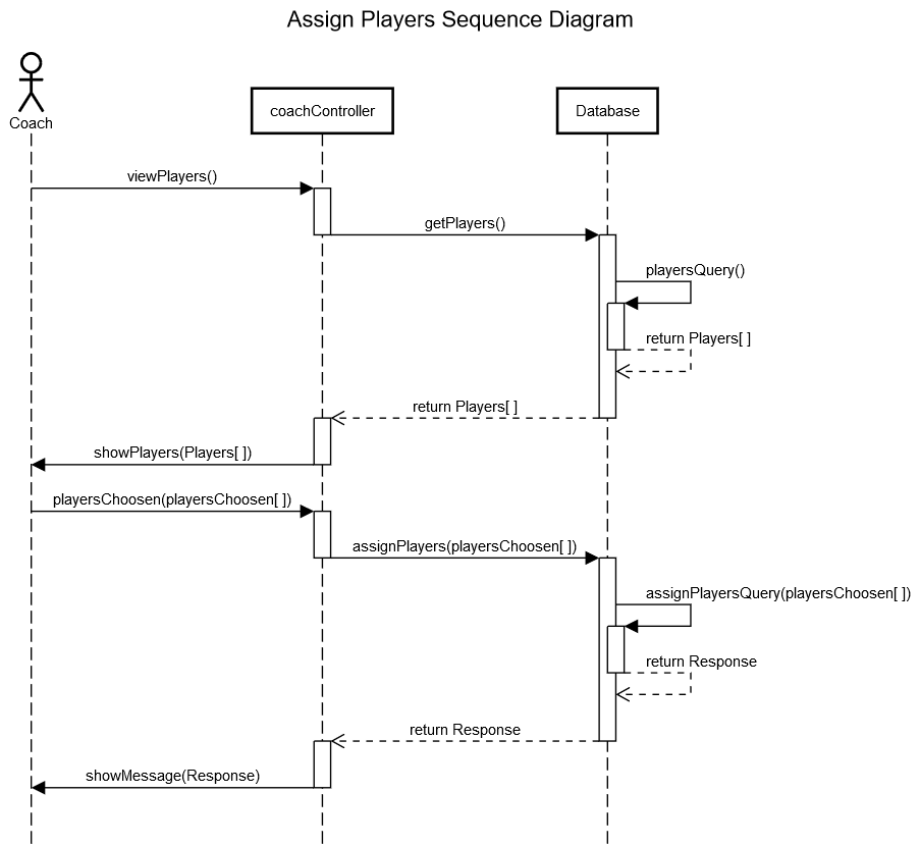


Figure 13: Assign Players Sequence Diagram

Login: As shown in figure 14, the user is represented with the login screen, so that the user is able to write the username and password, then the username and password are validated from the database, then if they are right and application will move to the next screen, but if they are wrong, an error message will be displayed.

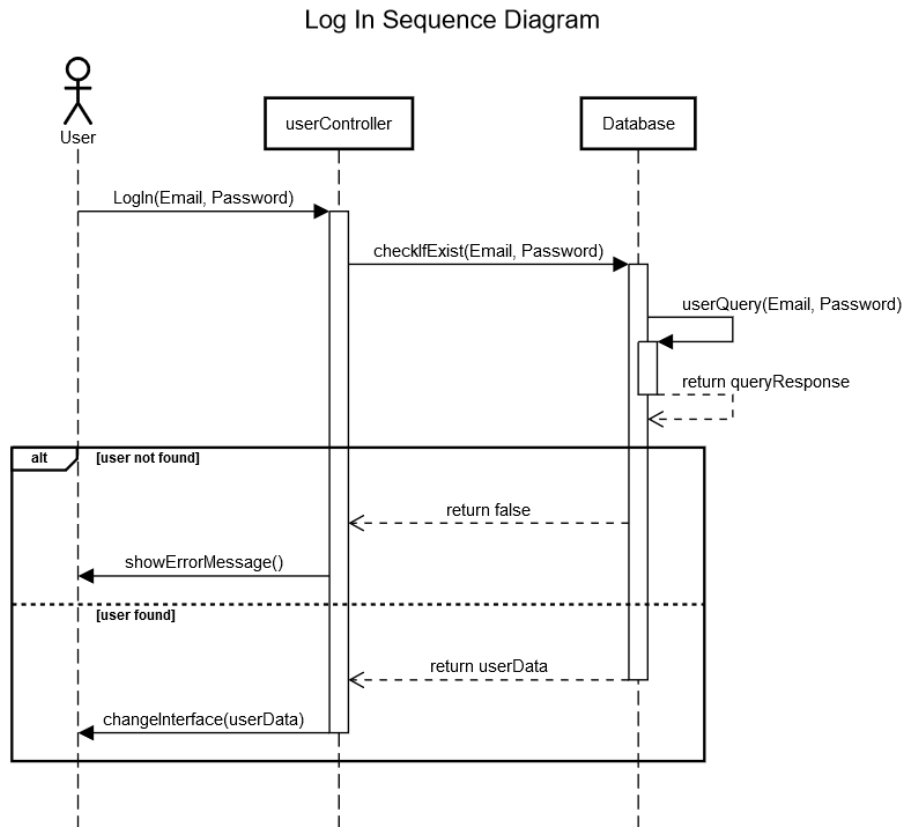


Figure 14: Log In Sequence Diagram

3.3 Design Rationale

The design of the system is based on the MVC as mentioned before to make it easier to create, modify, and optimize the functionality of the system. Firstly, in the hardware design we had multiple choices to choose from, the first option was the Kinect v1, the second option was the Kinect v2 and the third option was the accelerometers. We picked the Kinect v2 since it is the most reliable hardware. We could not choose the accelerometers since the practitioners are not allowed to wear any hardware during the performing of the move. Moreover, we needed to connect an accelerometer for each joint, which would be expensive and unreasonable. Also the Kinect v2 gets more joints and is better and more accurate than v1, that will help us achieve better results. Secondly, during the software design we had the choice between K-NN and Fast-DTW but we picked Fast-DTW since the K-NN is a basic algorithm that is used for proving the concept and is not intended for the deployment phases.

4 Data Design

4.1 Data Description

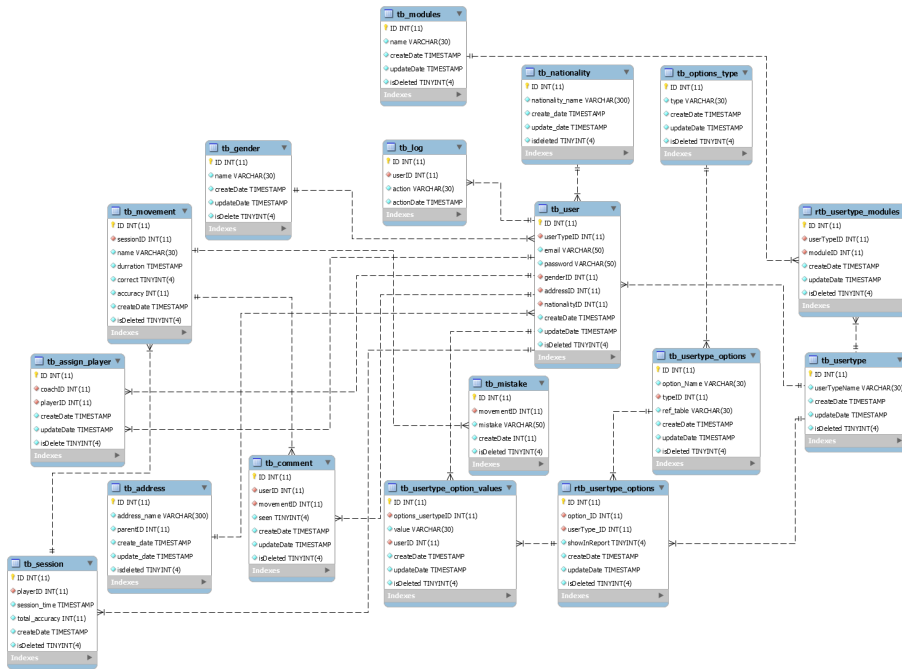


Figure 15: Database Diagram

4.2 Data Dictionary

The system database is derived from the main components in our application which are session, User and movement. The users tables are made with EAV (Entity Attribute Value) model, where we store the Entities and Attributes separately then collect their values in another table, to deliver highly dynamic system.

5 Component Design

5.1 Input

The input of the system is the coordinates of all the body joints in 3D space (X, Y, Z) captured from the Kinect. The Kinect's hardware is composed of an Infrared Emitter to track the body, displaying a basic skeleton and the body's joints using the Microsoft SDK for Kinect.

Furthermore, the Kinect is capable of providing 30 frames per second with a 640 x 480-pixel resolution using its video and depth sensor cameras. The Kinect works by starting the camera and capturing the RGB (red, green and blue) colors of the person to form its image. Then, the monochrome sensor and infrared projector start to receive the rays that were emitted to get the third dimension and form the 3D imagery of the skeleton of the person.

5.2 Pre-processing

Before processing on any of the data acquired by the Kinect, some pre-processing had to be done. The pre-processing in our system consists of series of operations needed to be done on the data for further operations such as classification. The pre-processing consists of data filtering, data interpolation, data normalization, feature selection and data segmentation. Normalization is the major pre-processing phase in the system, which will be used to overcome different body proportions (Height, Scale, etc.) or a dominant factor in the data. The algorithm proposed is "Z-score normalization".

$$X = \frac{Value - \mu}{\sigma} \quad (1)$$

Where "Value" is the data point, μ is the mean value and σ is the standard deviation of the data. If "X" is equals to the mean value of the feature, it will be normalized to zero. If it's below the mean, it will be normalized to a negative number, and if it's above the mean, it will be normalized to a positive number, The "X" value is calculated by the standard deviation. If the un-normalized data had a large standard deviation value, then the normalized values would be closer to zero.

5.3 Segmentation

The purpose of this phase is to segment each movement that the player had performed while the Kinect is capturing the data, this is essential to classify each move independently. By plotting the data we noticed that there is a small gap between each move, this gap can be used to segment each movement.

5.4 Classification

After pre-processing and segmenting the data, we chose Fast-DTW for the classification. Fast-DTW is used to manage the different speeds of the moves taken by the player using the Kinect and to provide the player a real time feedback as accurate as possible. It is an algorithm for measuring similarities between two signals, each signal may have a different speed from the other signals. Fast-DTW is also an alignment algorithm which is capable of classifying two different time signals. Fast-DTW could be used with many different distance equations but the "Euclidean Distance" is the one used to compute the distance between the classes.

$$d = \sum_{x_i, y_i}^n \sqrt{(x_i - y_i)^2} \quad (2)$$

Where "D" is the distance value, "X" represents the data-set joint position and "Y" represents the performer's joint position.

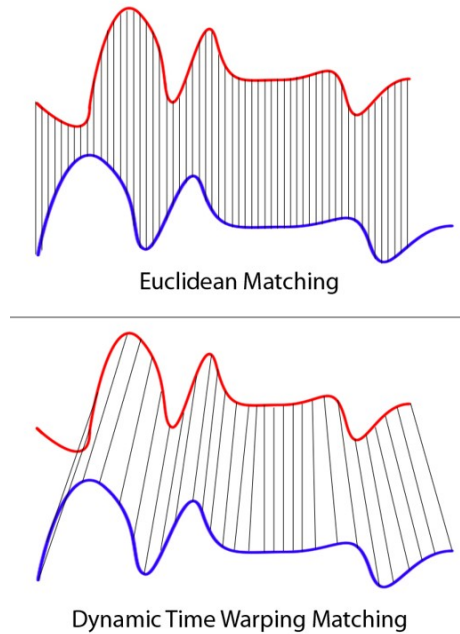


Figure 16: Fast-DTW

5.5 Output

The last part of the system is the output. Which will be categorized as follows:

Result Screen: This screen will tell the user if they performed the move correctly or not, with a percentage of how much the movement was performed correctly. If the percentage is acceptable and the move was done correctly, the screen will inform the user and display the percentage of the correctness. If the practitioner performed the move in a wrong way, the screen would display to them what they did incorrectly regarding the move and how to perform it correctly with a report.

Report: The second part of the output would be the report. The report will benefit both the student and the coach. Since this report will have a fully-detailed statistics of how accurate the practitioner performed the moves, mistakes and how to improve the performance.

6 Human Interface Design

6.1 Overview of User Interface

The system's user interface is simple and efficient that guarantees all users to find it easy. Firstly, the user will start the application by signing in with his account or creating a new one. After that, the system will move to the movement capture screen where he takes the position in front of the Kinect, assuming that it is already connected with the computer. The movement capture screen will have one button for start/stop recording that insures that the user will not have many complicated buttons so he will not get confused. Secondly, after recording the movement, a screen will show the overall accuracy and if the sequence of movements he/she performed is acceptable or not. Lastly, there will be a filter screen for the user to choose the preferences he wanted to be displayed in the detailed report.

6.2 Screen Images

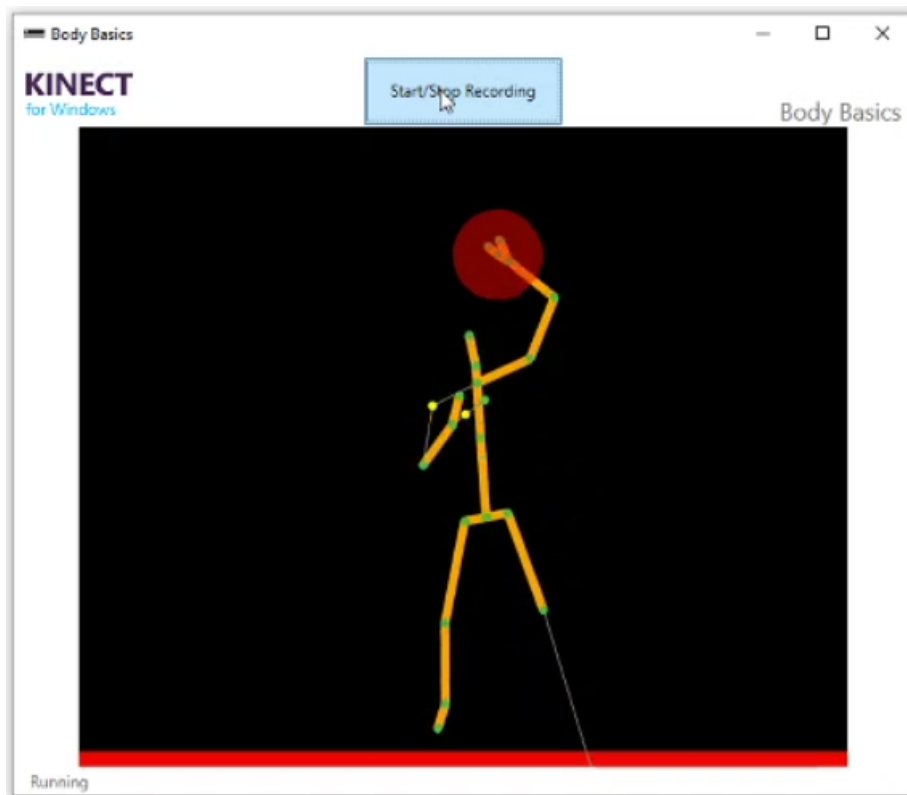


Figure 17: Movement Capture Interface

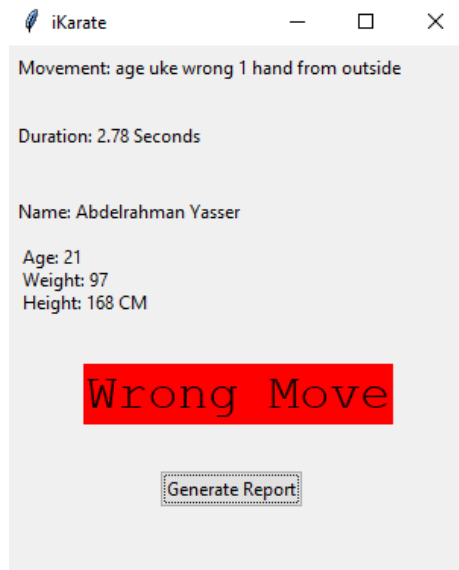


Figure 18: Results Interface

IKarate			
Name: Abdelrahman Yasser		Date: 03/03/2020 21:42	
Age: 21			
Weight: 97KG			
Height: 168CM			
Belt: Black			
Movement	State	Mistake	Accuracy %
Age-uke	Wrong	Hand curve was from outside	60
Age-uke	Wrong	Short leg angle	57
Gedan Brai	Right	None	100
Mae Geri	Wrong	Straight Leg	67
Mae Geri	Wrong	Lose hand	76
This table Shows all movement done in session with their accuracy and mistakes			

Figure 19: Report

6.3 Screen Objects and Actions

The system GUI is so simple since users might not have high knowledge with computers.

- Login interface consists of a simple form to get the users data and sign up form to register on the system. This data will be used to be displayed on the report.
- Movement capture interface which contains a button for the users to start/stop their session and some indicators to show if the Kinect detected the body or if he is close to an edge to adjust his/her positions as shown in FIG 17.
- Results interface is where the user will see the session results, if he/she made a wrong movement and what was the movement done with its details as shown in FIG 18.
- Report interface where the user starts to choose some filters for the report to be generated and displayed at the end of the session as a PDF.

7 Requirements Matrix

Code	Name	Type	Description	Testing Strategy	Status
F1	Check Pre-processing	Required	Analyzes data for pre-processing before being classified.	Data need to be captured and stored in file and checks if the data need any processing.	In Progress
F2	Interpolation	Required	Interpolates the movement data before being sent for classification.	Checks that the data is smaller than the previous file.	In Progress
F3	Extrapolation	Required	Extrapolates the movement data before being sent for classification.	Checks that the data is larger than the previous file.	In Progress
F4	Removing Outliers	Required	Used to remove any outliers from the data which could be generated from the Kinect.	Validate all the data in the files to ensure that there are no abnormal points.	In Progress
F5	Segmentation	Required	Used to segment the data read from the Kinect.	Validate that all the new arrays are originated from the same file.	In Progress
F6	Classify Movement	Required	Fast-DTW to compare the data to the model, After the player movement has been extracted and pre-processed.	Must give the movement name and the mistake type.	Completed
F7	Get Joints Data	Required	Capturing the skeleton joint coordinates using the Kinect and store them in an array.	Should return the body joints during the session in array.	Completed
F8	Store Data	Required	Fired after capturing the data from Kinect into array, to store our data into files for processing.	Should take the stored data and make pre-processing then store the final data in a file.	Completed
F9	File Upload	Required	Handles uploading files processes.	Checks that there is an internet connection and send the files to the cloud for processing.	In Progress
F10	View all Records	Required	Retrieves a specified player's records and display its details.	Returns a list of all the previous session records of a player.	In Progress
F11	Calculate Overall Score	Required	Calculates the overall score of the player.	Returns an accuracy or score for all the movement in the session.	In Progress
F12	View Report	Required	Used to view the report of the current training directly.	Validate that the session movements was classified and every move received a score, Also that the report was generated.	Completed

8 References

References

- [1] V. Kecman, “Support vector machines – an introduction,” *Support Vector Machines: Theory and Applications Studies in Fuzziness and Soft Computing*, p. 1–47, 2005.
- [2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” *2008 Eighth IEEE International Conference on Data Mining*, 2008.
- [3] K. Malik, H. Sadawarti, and K. G. S, “Comparative analysis of outlier detection techniques,” *International Journal of Computer Applications*, vol. 97, no. 8, p. 12–21, 2014.
- [4] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE Multimedia*, vol. 19, no. 2, p. 4–10, 2012.
- [5] T. Hachaj, M. R. Ogiela, and M. Piekarczyk, “Dependence of kinect sensors number and position on gestures recognition with gesture description language semantic classifier,” *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, 2013.
- [6] P. Alborn, N. D. Giorgis, A. Camurri, and E. Puppo, “Limbs synchronisation as a measure of movement quality in karate,” *Proceedings of the 4th International Conference on Movement Computing - MOCO 17*, 2017.
- [7] A. D. Calin, “Variation of pose and gesture recognition accuracy using two kinect versions,” *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2016.
- [8] Y. Choubik and A. Mahmoudi, “Machine learning for real time poses classification using kinect skeleton data,” *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, 2016.
- [9] Dharmayanti, M. Iqbal, A. Suhendra, and A. B. Mutiara, “Velocity and acceleration analysis from kinematics linear punch using optical motion capture,” *2017 Second International Conference on Informatics and Computing (ICIC)*, 2017.
- [10] E. Escobedo-Cardenas and G. Camara-Chavez, “A robust gesture recognition using hand local data and skeleton trajectory,” *2015 IEEE International Conference on Image Processing (ICIP)*, 2015.
- [11] T. Hachaj and M. R. Ogiela, “Qualitative evaluation of full body movements with gesture description language,” *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, 2014.

- [12] T. Hachaj, M. R. Ogiela, and K. Koptyra, "Human actions modelling and recognition in low-dimensional feature space," *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, 2015.
- [13] T. Hachaj, M. R. Ogiela, and K. Koptyra, "Effectiveness comparison of kinect and kinect 2 for recognition of oyama karate techniques," *2015 18th International Conference on Network-Based Information Systems*, 2015.
- [14] T. Hachaj, M. R. Ogiela, and K. Koptyra, "Learning from annotated video: An initial study based on oyama karate tournament recordings," *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2015.
- [15] T. Hachaj, M. R. Ogiela, M. Piekarczyk, and K. Koptyra, "Advanced human motion analysis and visualization: Comparison of mawashi-geri kick of two elite karate athletes," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.
- [16] T. Hachaj, M. R. Ogiela, and M. Piekarczyk, "The open online repository of karate motion capture data: A tool for scientists and sport educators," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.
- [17] T. Hachaj, M. Piekarczyk, and M. R. Ogiela, "How repetitive are karate kicks performed by skilled practitioners?," *Proceedings of the 2018 10th International Conference on Computer and Automation Engineering - IC-CAE 2018*, 2018.
- [18] V. M. S. Janaki, S. Babu, and S. S. Sreekanth, "Real time recognition of 3d gestures in mobile devices," *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2013.
- [19] J.-H. Kim, N. D. Thang, and T.-S. Kim, "3-d hand motion tracking and gesture recognition using a data glove," *2009 IEEE International Symposium on Industrial Electronics*, 2009.
- [20] K. Kolykhalova, A. Camurri, G. Volpe, M. Sanguineti, E. Puppo, and R. Niewiadomski, "A multimodal dataset for the analysis of movement qualities in karate martial art," *Proceedings of the 7th International Conference on Intelligent Technologies for Interactive Entertainment*, 2015.
- [21] K. Mitsuhashi, H. Hashimoto, and Y. Ohyama, "The curved surface visualization of the expert behavior for skill transfer using microsoft kinect," *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*, 2014.
- [22] K. Mitsuhashi, S. Yokota, H. Hashimoto, S.-G. Shin, and D. Chugo, "Educational system of physical motion based on 3d biomechanism evaluation," *2016 IEEE International Conference on Industrial Technology (ICIT)*, 2016.

- [23] R. Niewiadomski, K. Kolykhalova, S. Piana, P. Albornò, G. Volpe, and A. Camurri, "Analysis of movement quality in full-body physical activities," *ACM Transactions on Interactive Intelligent Systems*, vol. 9, p. 1–20, Nov 2019.
- [24] V. A. Prisacariu and I. Reid, "Robust 3d hand tracking for human computer interaction," *Face and Gesture 2011*, 2011.
- [25] K. Tanaka, "3d action reconstruction using virtual player to assist karate training," *2017 IEEE Virtual Reality (VR)*, 2017.
- [26] N. T. Thanh, N. D. Tuyen, L. Dung, and P. T. Cong, "Implementation of technical data analysis of skeleton extracted from camera kinect in grading movements of vietnamese martial arts," *2017 International Conference on Advanced Technologies for Communications (ATC)*, 2017.
- [27] K. S. Urbinati, E. Scheeren, and P. Nohama, "A new virtual instrument for estimating punch velocity in combat sports," *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013.
- [28] K. Wennrich, B. Tag, and K. Kunze, "Vrte do - the way of the virtual hand," *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology - VRST 18*, 2018.
- [29] T. Hachaj, K. Koptyra, and M. R. Ogiela, "Averaging of motion capture recordings for movements' templates generation," *Multimedia Tools and Applications*, vol. 77, no. 23, p. 30353–30380, 2018.
- [30] T. Hachaj and M. R. Ogiela, "Rule-based approach to recognizing human body poses and gestures in real time," *Multimedia Systems*, vol. 20, p. 81–99, Mar 2013.
- [31] T. Hachaj, M. Piekarczyk, and M. Ogiela, "Human actions analysis: Templates generation, matching and visualization applied to motion capture of highly-skilled karate athletes," *Sensors*, vol. 17, p. 2590, Oct 2017.
- [32] C. Vondrick, D. Ramanan, and D. Patterson, "Efficiently scaling up video annotation with crowdsourced marketplaces," *Computer Vision – ECCV 2010 Lecture Notes in Computer Science*, p. 610–623, 2010.
- [33] P. Paliyawan, K. Sookhanaphibarn, W. Choensawat, and R. Thawonmas, "Body motion design and analysis for fighting game interface," *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, 2015.
- [34] T. Hachaj and M. R. Ogiela, "Semantic description and recognition of human body poses and movement sequences with gesture description language," *Communications in Computer and Information Science Computer Applications for Bio-technology, Multimedia, and Ubiquitous City*, p. 1–8, 2012.

- [35] T. Hachaj, M. R. Ogiela, and M. Piekarczyk, “Real-time recognition of selected karate techniques using gdl approach,” *Advances in Intelligent Systems and Computing Image Processing and Communications Challenges 5*, p. 99–106, 2014.