Final Thesis Document for iKarate: Improving Karate Kata

by

Bassel Emad, Omar Atef, Yehya Shams, Ahmed El-Kerdany

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Bachelor of computer science

in

Department of Computer Science

in the

Faculty of Computer Science
of the
Misr International University, EGYPT

Thesis advisor:
Dr. Ayman Ezzat, Dr. Ayman Nabil,
and Eng. Nada Ayman

(July 2020)

Abstract

Karate is a complicated sport; practiced using hands and feet to deliver and block strikes. Karate moves must be done in a certain way and is difficult to master.

Because of the growing of interest in this sport over the past few years, attempts were made to improve the experience and the quality of training through different methods. There are multiple issues to consider for building a Karate smart coaching system, such as: hardware utilization, data stream analysis, and generating an accurate report based on the users' movements. In this document we propose a system that will capture the players' moves using a virtual recognition device equipped with an IR (Infrared) camera, perform analysis on the data and pre-process it for classification. The System will classify the output using F-DTW (Fast Dynamic Time Warping) and then present the feedback containing the analysis results. This application's performance and results were tested using groups of professional and unprofessional candidates in karate to evaluate how the system would behave with each person and verify the effectiveness of methods used. The system classifies each move and its most known mistakes with an accuracy of 91.07%.

Acknowledgments

We sincerely like to thank Dr. Ayman Ateya in the faculty of Computer Science at Modern Science and Arts University for his effort and playing a big role in the team, pledging to us his experience and knowledge that led us to move forward, through following up with every step and providing useful feedback that we needed and helped us reach our goal with the project to what it is now. He was always available, welcoming every question and answer them meticulously, and he has been with us with in every step in every phase of the project providing outstanding support and the aid to compete.

We are thankful to Dr. Ashraf Abdelraouf in the faculty of Computer Science at Misr International University for his incredible work in managing and tending to the whole faculty's graduation projects, utilizing his experience and knowledge. He is always quick to step up whenever an issue arises to cooperatively resolve it, executing well thought-out decisions.

We would like to show appreciation to Dr. Ayman Nabil for his undeniable impact within every phase by giving us useful constructive feedback and the means to improve our project. And providing us the financial support for publishing our work in multiple conferences and journals.

We can't deny the huge effort given by the dean of the faculty of Computer Science at Misr International University Professor. Ayman Bahaa in developing our skill-set and beneficial knowledge through the previous years, getting us prepared for the real world context. Only he adds more appraisal in our work and for his on going work in further improving the faculty's worth and essence.

Last but not least we thank Eng. Nada Shorim for her outstanding passion in assisting and supporting us through the journey of our project. Not only she gave us her knowledge and experience, but she aided us by following-up, leaving comment and recommendations to achieve the best version of our project.

Finally, Al-Ahly sporting club and its performers sponsored this work, we would like to thank all the performers who have helped in this approach.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Karate moves are combination of successive moves. Kids nowadays may find difficulties learning those moves at a young age, since the training may consist of a large number of players, the trainer himself may not be able to focus on every detail of every player's move, which may result in taking more time for learning the move or to be precious to master it or it may lead to learning that move in a wrong way from the beginning. The main goal of this project is to capture the moves of the players, analyse those moves and give the players a feedback to enhance their technique or alert them if they are playing in a wrong way.

1.2 Motivation

Despite the importance of the sports field and improving it, not many people focus on working in the Karate sport, so as a team we saw a good opportunity to continue researching in the Karate sport, by improving and adding some features that might make a difference. We also found that kids playing Karate at the beginning don't have the attention needed from the coach who should focus on the little mistakes made by the kids.

1.3 Problem Definitions

There is the problem of the real-time feedback, giving the player a feedback on his/her moves whether it was right or not in a couple of minutes is essential after he/she finishes the move, the feedback should have high-accuracy. The feedback contains tips on how to execute the move correctly. Moreover, each player has different body proportions than the other players.

1.4 Project Description

Karate training assisting system that detects Karate moves and analyze them to detect the mistakes and to present a feedback in an interface to the user, showing the recognized moves and explaining whether there was a mistake or not in a certain position.

1.4.1 Objective

The project aims to improve Karate coaching with the assistance of motion capture technologies and algorithms. And deliver a system that can be used to practice at home or at a sports club

1.4.2 Scope

The system is designed to cover multiple things:

1. Help Karate players who don't have Karate coaches.
2. Help Karate Players practice at home.
3. Assist Karate coaches.
4. Extract body position mistakes that could be hard to be noticed by a coach.
5. User data can be recorded and analyzed, thus improving the player's experience and training.

1.4.3 Project Overview

As shown in figure 1.1, the system consists of a Kinect connected to a computer that has an internet connection. The player starts by logging in to his account, then calibrates his positioning to the Kinect, so that the joints could be captured correctly. After everything is set-up and the performer is prepared to perform the moves, the performer will click on the start recording button so that the Kinect could start recording his movements by capturing the frames then extracting the skeleton data from these frames to be saved in a CSV file. After the performer finishes, the performer will click on the stop recording button. Then the segmentation and the pre-processing phases will occur on the CSV file so it can be forwarded to the processing stage. The processing will classify the movement using the stream analysis. The stream analysis is where the segmented movements that the player has performed is classified. After the processing is done, a class will be assigned to each movement, then the results are sent to be stored in the database. The results are also displayed briefly on the computer's screen. It also can be shown in detail in a report which contains all the performer's information and also all the details of the movements that the performer has done, such as the movement's name, movement's duration, movement's score, their performance so far. The detailed report could be personalized to show what the user really needs.

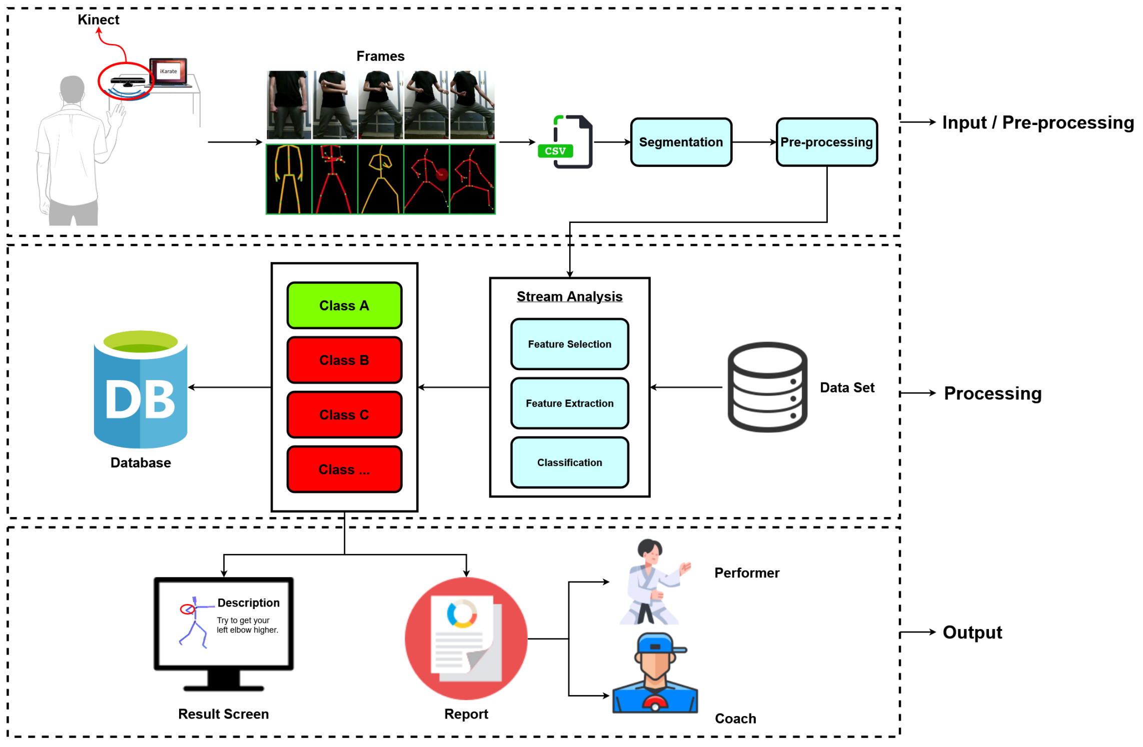


Figure 1.1: System Overview

Chapter 2

iKarate: Improving Karate Kata

2.1 Similar System Information

- T. Hachaj, et. el in [4] proposed statistical comparison between Kinect 1 and Kinect 2 recognition in some of the Karate movements. This research is done to evaluate effectiveness of Kinect 1 and Kinect 2 for Karate motion recognition. Their Motivation was different types of Kinect sensors, made them start this research to find the best suited sensor for Karate. They had proved in all cases that Kinect 2 is more reliable than Kinect 1 due to more accurate calculation of legs joints positions. It is important to us to make it clear that Kinect 2 will give us more accuracy and will improve our recognition rather than Kinect 1.

- T. Hachaj, et. el in [5] proposed evaluation and visualization technique for the advanced human motion analysis. This paper is to evaluate the method for comparison, analysis and visualization of similarities and differences between 3D trajectories of body joints in Karate movements. They were interested in investigating what are the differences in the movements, due to imperfect imitation problem. This paper is important since they touched an important problem that might face us, which is imperfect imitation problem.

- E. Escobedo-Cardenas and G. Camara-Chave in [6] proposed an approach for dynamic hand gesture recognition using Kinect. They wanted to Overcome the problems of hand gesture recognition using Sensor and Video based tools. They came up with better recognition compared to the methods that uses visual information only with 88.38% accuracy. Their approach can be used to improve our accuracy and get the exact hand gestures, which could benefit our system.

- T. Hachaj, et. el in [7] proposed actions descriptions with maximally three keyframes. Their aim was to make motion recognition in low-dimensional feature space and selection of proper features to model a set of multiple human actions. They were able to reach recognition rate of level of 88%. Their selection of proper features set to model human actions in low-dimensional space could help us prioritise our features selection for better recognition.

- P. Alborno, et. el in [8] proposed a method to compute the measurement of Karate movement quality. Analysis of human full body movements to evaluate movement qualities is an important problem. They reached a solution by studying how much the limbs are synchronized during relevant motion phases. They proved that there is a way to measure the quality of Karate movements, that we can use to deliver a complete system.

- Y. Choubik and A. Mahmoudi in [9] was able to make a real-time human poses classification technique. Poses recognition is important in many situations, and their problem was to apply machine learning algorithms to classify real-time poses. Finally, they were able to classify user's poses whatever his size and his position in the scene, which will help us to classify the pose, which is a part of the Karate movements.

- N. T. Thanh, et. el in [2] made a program that applies the depth data of images to human performance scoring system. Their aim was to make a standardized modeling system that can be used around the world for Vietnamese Traditional martial arts. They used Data analysis on the Kinect's motion capture, which will guide us.

- A. D. Calin, in [10] compared the efficiency of several classifiers, trained and tested on a data-set obtained from Kinect v1 and v2. Their aim was to Evaluate the accuracy of the two Kinect Sensors and Analyse the variation of the gesture recognition accuracy of several classifiers. He reached an accuracy of 99.0874% by Multilayer Perceptron and Kinect 2 data. This will help us in choosing our data-set and the classifiers wisely.

- T. Hachaj, et. el in [11] made a video annotation method that enables both numerical and categorical features calculation. Their aim was to create efficient system for learning Karate. Since most of the up-to-date videos data are amateur films which are not reliable for self-learning. This system is very close to our system, which we could benefit from knowing the methods and techniques they used.

- T. Hachaj, et. el in [12] proposed a calibration procedure of three Kinect sensors that integrates the data into one skeleton. Their problem was to find a way to increase Karate motion recognition accuracy and effectiveness of the fusion of the body joints gathered from different sensors. They reached a positioning of the three sensors to improve the non-classified techniques with 48%. In case of using GDL (Gesture Description Language) and more than one Kinect this paper will help us to increase our accuracy.

2.1.1 Similar System Description

- A system that was proposed by Mitsuhashi, el. in [1] presents a support system of body motion based on three-dimensional bio mechanism evaluation. The most important problem they face was imperfect imitation problem, that human cannot attain the perfect copy or imitation of the referenced motion because of skeleton size, each skeleton has different size from the other skeletons.

- The system made by Wennrich, el. in [3] is different because they didn't involve any Kinects in Video/Motion capture setups. The system they built is a game, were the player should learn and repeat different Karate movements to increase the progress and reach the next level. They also, used "HTC VIVE" to build this Virtual Reality game.

2.1.2 Comparison with Proposed Project

Our proposed project is a hybrid system, that will take Karate movements classification, measurements of quality and grading system to a better level. We will also propose another approaches and methods to enhance the movements classification. We will be combining all of these features to deliver a coaching and judgment system with all the requirements needed to reach high-quality performance in Karate (Kata style).

2.1.3 Screen Shots from previous systems

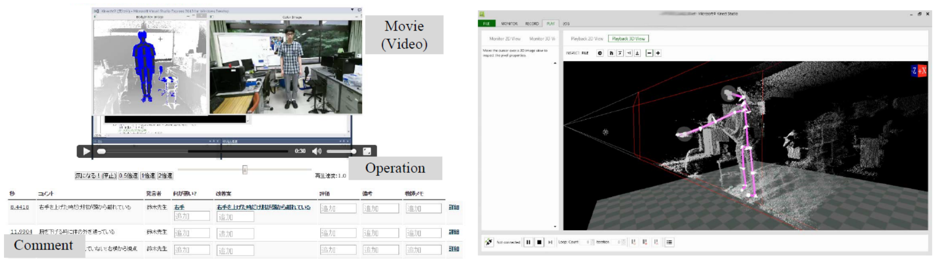
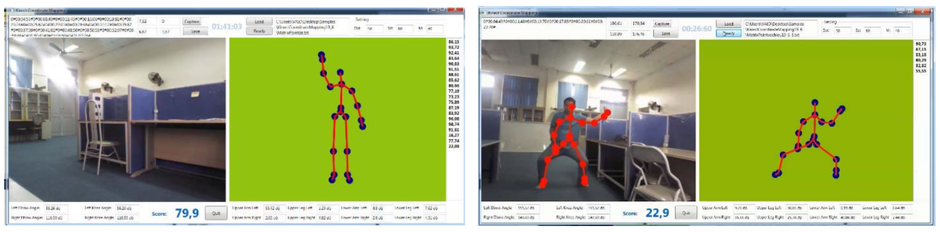


Figure 2.1: Similar System [1]



(a) First test on the system (b) Second test on the system

Figure 2.2: This System is taken from [2]

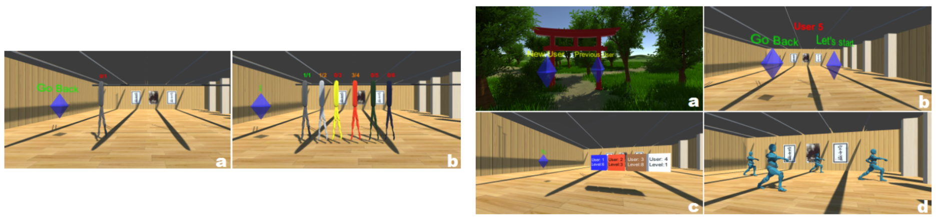


Figure 2.3: This system is taken from [3]

2.2 Project Management and Deliverables

2.2.1 Tasks and Time Plan

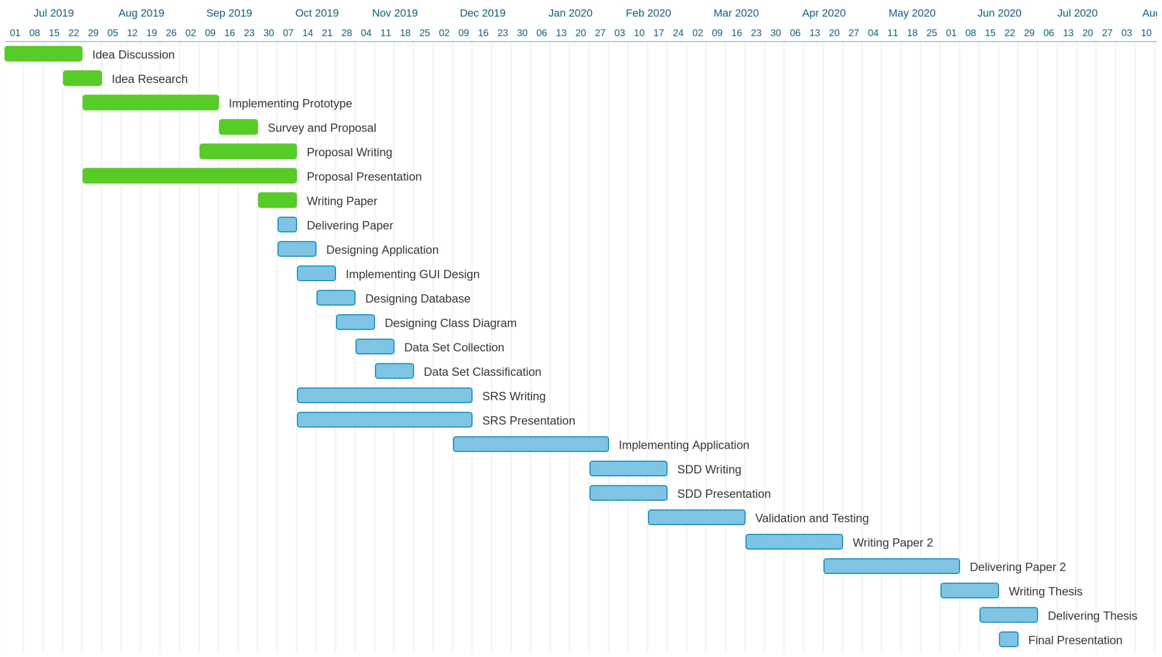
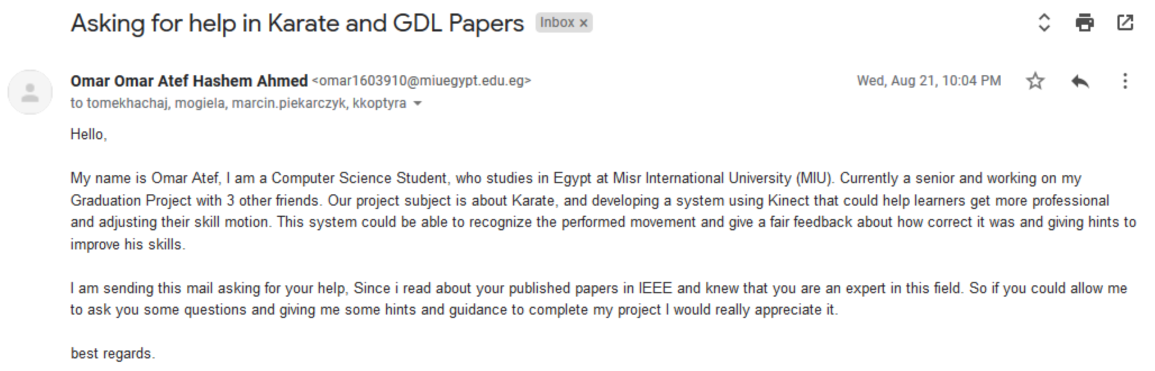


Figure 2.4: Tasks And Time Plan

2.2.2 Budget and Resource Costs

1. Microsoft Xbox One Kinect Sensor: 300$.
2. Microsoft Xbox One Kinect Sensor Adapter For Windows: 25$.
3. A Server To Process The Data: 160$/Month.
4. A Cloud To Store The Data: 20$/Year.

2.2.3 Supportive Documents


Asking for help in Karate and GDL Papers Inbox ×

Omar Omar Atef Hashem Ahmed <omar1603910@miuegypt.edu.eg>
to tomekhachaj, mogiela, marcin.piekarczyk, kkoptyra

Wed, Aug 21, 10:04 PM

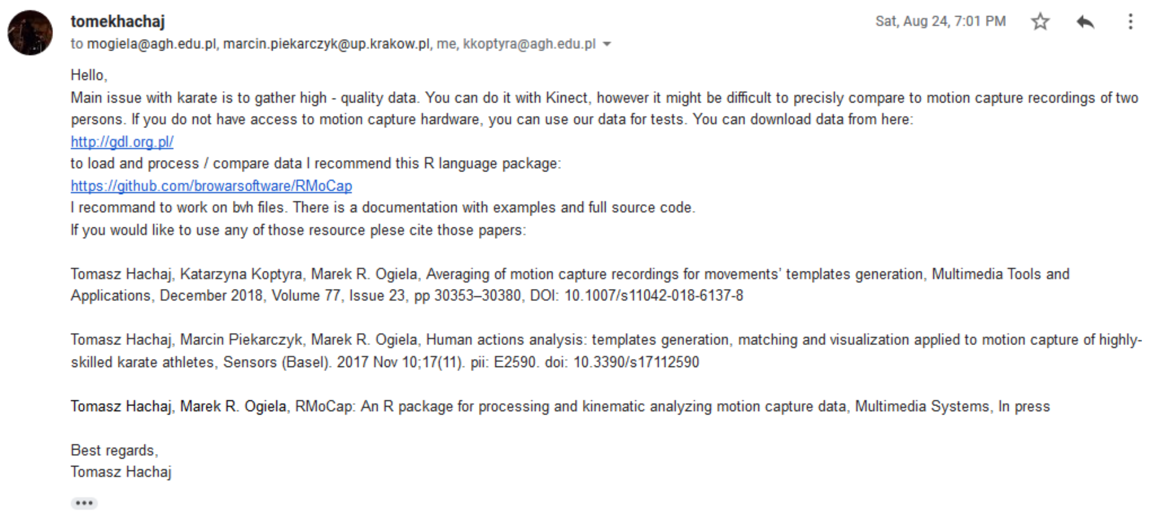
Hello,

My name is Omar Atef, I am a Computer Science Student, who studies in Egypt at Misr International University (MIU). Currently a senior and working on my Graduation Project with 3 other friends. Our project subject is about Karate, and developing a system using Kinect that could help learners get more professional and adjusting their skill motion. This system could be able to recognize the performed movement and give a fair feedback about how correct it was and giving hints to improve his skills.

I am sending this mail asking for your help, Since i read about your published papers in IEEE and knew that you are an expert in this field. So if you could allow me to ask you some questions and giving me some hints and guidance to complete my project I would really appreciate it.

best regards.

Figure 2.5: Data-Set 1/2


tomekhachaj
to mogiela@agh.edu.pl, marcin.piekarczyk@up.krakow.pl, me, kkoptyra@agh.edu.pl

Sat, Aug 24, 7:01 PM

Hello,
Main issue with karate is to gather high - quality data. You can do it with Kinect, however it might be difficult to precisly compare to motion capture recordings of two persons. If you do not have access to motion capture hardware, you can use our data for tests. You can download data from here:
http://gdl.org.pl/
to load and process / compare data I recommend this R language package:
https://github.com/browarsoftware/RMoCap
I recommand to work on bvh files. There is a documentation with examples and full source code.
If you would like to use any of those resource plese cite those papers:

Tomasz Hachaj, Katarzyna Koptyra, Marek R. Ogiela, Averaging of motion capture recordings for movements' templates generation, Multimedia Tools and Applications, December 2018, Volume 77, Issue 23, pp 30353–30380, DOI: 10.1007/s11042-018-6137-8

Tomasz Hachaj, Marcin Piekarczyk, Marek R. Ogiela, Human actions analysis: templates generation, matching and visualization applied to motion capture of highly-skilled karate athletes, Sensors (Basel). 2017 Nov 10;17(11). pii: E2590. doi: 10.3390/s17112590

Tomasz Hachaj, Marek R. Ogiela, RMoCap: An R package for processing and kinematic analyzing motion capture data, Multimedia Systems, In press

Best regards,
Tomasz Hachaj

Figure 2.6: Data-Set 2/2

Chapter 3

System Requirements Specifications

3.1 Introduction

3.1.1 Purpose

The purpose of this chapter is to present a detailed description of the system (iKarate). The system is designed for Karate coaches, Karate players and self-learning Karate beginners. It aims to improve Karate Kata training and coaching with the aid of motion recognition. Further, in this chapter, features, design and the goals of the system will be explained and how the system behaves based on the customer's requirements. This software requirements specification (SRS) chapter defines how our stakeholder, team and audience see the product and its functionality.

3.1.2 Scope

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

3.1.3 Overview

The system aims to allow coaches to have a better follow up with their students by creating a profile for each student and storing it in a database, help the coach while training the students by analyzing their movements while performing through Kinect in real-time and evaluate their performance. The system also can assist the Karate Kata player train without a coach by learning from their mistakes using the full feedback report provided after performing.

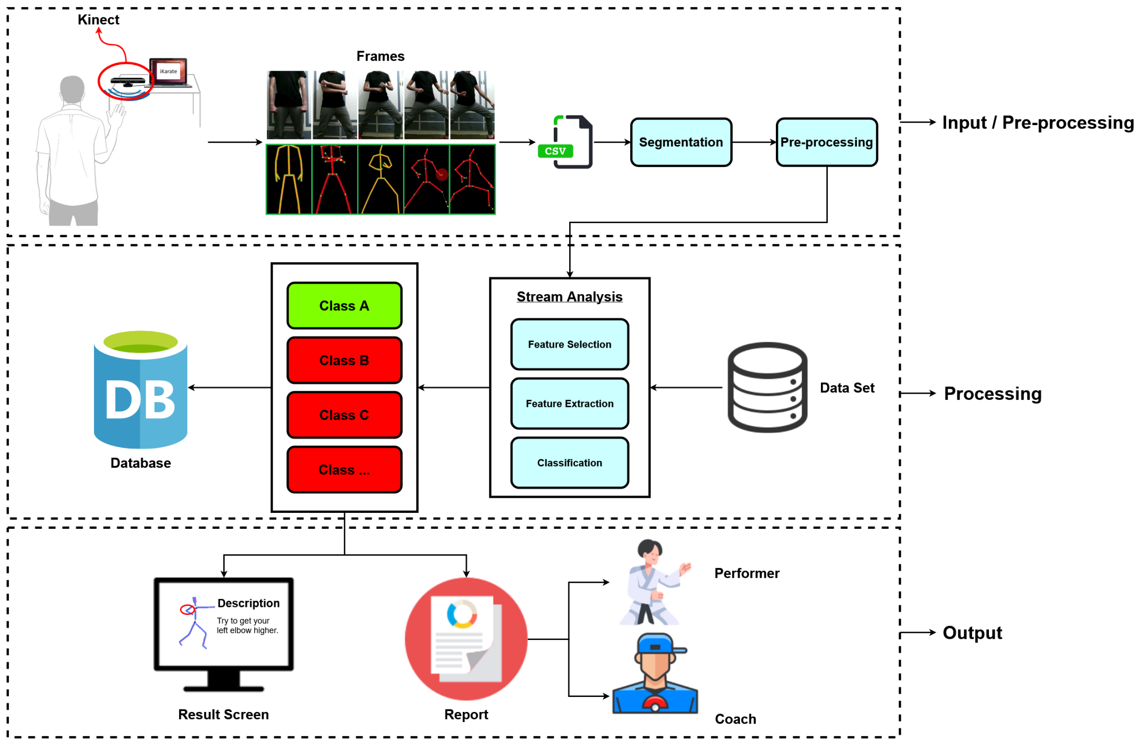


Figure 3.1: System Overview

3.1.4 Business Context

The sponsoring organization for this system is Al-Ahly Sporting Club by supporting us in information and database gathering, also in testing the software. The system will help trainees and the organization to be able to train Kata more efficiently and effectively, by increasing the ability to follow up with students using generated report. It will also help judges detect mistakes made by the performer and provide overall score.

3.2 General Description

3.2.1 Product Functions

1. Detecting Karate movements and displaying it in real-time.
2. Highlights the mistake made by the Karate player while performing and displays how to perform the move correctly.
3. Evaluates the overall performance of a Karate player and provide a score.
4. The system can generate a full report of the player movements.
5. Karate coach/player can register on the system and the coach could be linked with the students to track their progress.
6. The player's previous history can be viewed in the system.

3.2.2 Product Context

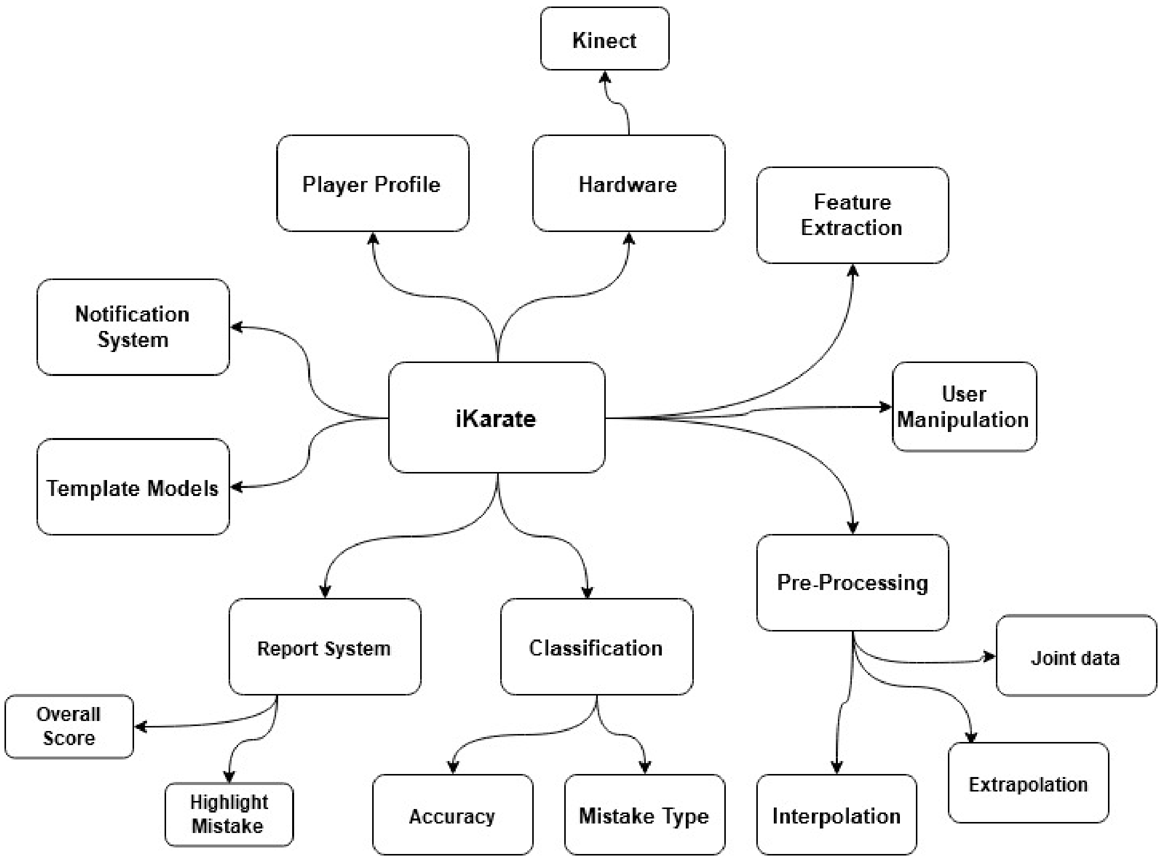


Figure 3.2: Context Diagram

3.2.3 Similar System Information

The system proposed by N. T. Thanh, el. in [2] presents a scoring software program in grading martial arts' movements. This system helps in self-training and evaluation of practitioner movements in Vietnamese traditional martial arts.

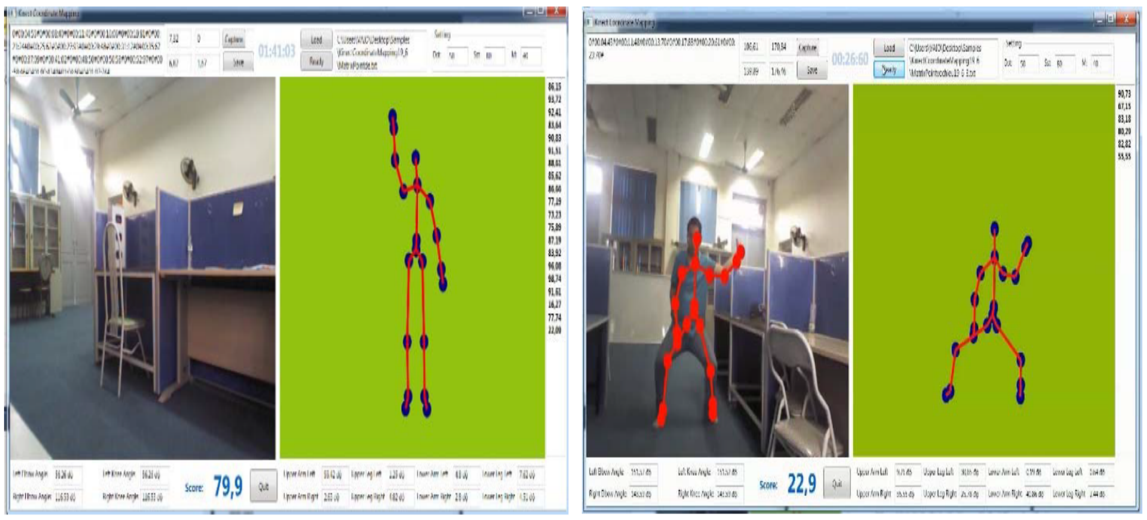


Figure 3.3: This System is taken from [2]

A system that was proposed by Mitsuhashi, el. in [1] presents a support system of body motion based on three-dimensional bio mechanism evaluation. The most important problem they face was imperfect imitation problem, that human can't attain the perfect copy or imitation of the referenced motion because of skeleton size, each skeleton has different size from the other skeletons.

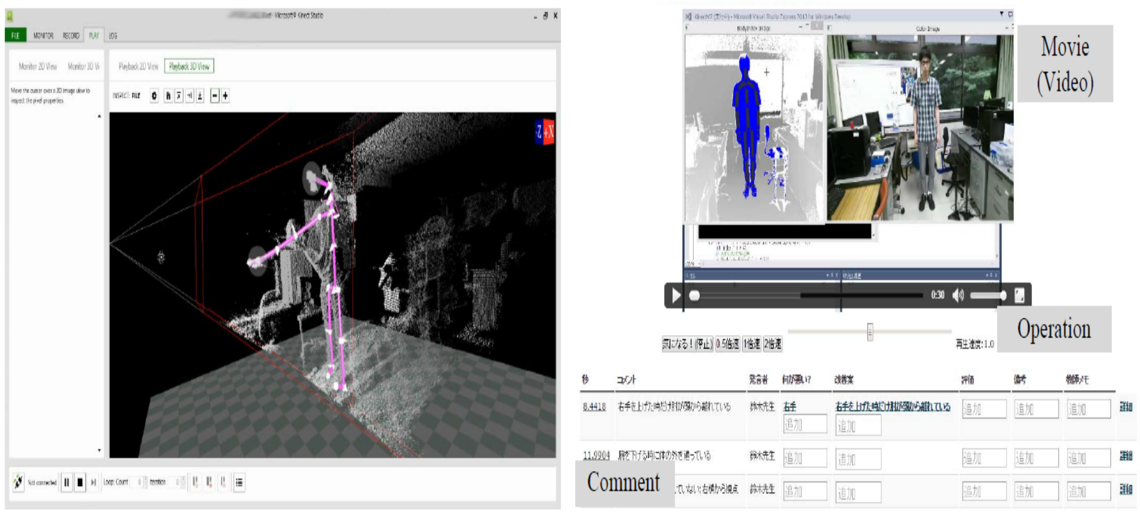


Figure 3.4: This System is taken from [1]

3.2.4 User Characteristics

1. Karate coach: the coach can register and get linked with his students, start motion observation and analyses and view reports. The coach must have basic knowledge on how to operate a computer system.
2. Karate player: the player can start motion observation, analyses and view reports for self-learning or to be viewed by the coach. The player must have basic knowledge on how to operate a computer system, or his parents could do the registration if he is below 10 years.

3.2.5 User Problem Statement

Karate players training require strong observation and analysis. The Kata players need to train on the movement to perform it efficiently. Moreover, new players need guidance all the time. The coaches view point could lead to miss judgment. Coaches also, can't always remember the mistakes of each performer. However, they need to remember the mistakes of everyone to improve it.

3.2.6 User Objectives

Karate coaches need support with training the players. Moreover, they need to have more detailed information about their players, so that they could work on the weak points of every player. On the other hand, the player will want to practice alone at home to improve his movements and to send the report to his coach to advise him. And the judges could use it to support their evaluation on the player's performance.

3.2.7 General Constraints

One of the major constrains of this project is the Kinect, weather the user could afford it (Depends on the price in the country) or to be positioned in a right way to over the full view of the player. Also, this system will need good internet connection for real-time feedback and the generation of the report.

3.3 Functional Requirements

3.3.1 Pre-processing

Table 3.1: Pre-Processing

Name	Check Pre-processing
Code	F1
Priority	Extreme
Critical	10/10
Description	This function analyzes data for pre-processing before being classified. Based on the analyses the data is sent to the appropriate functions, to remove outliers, extrapolate or interpolate the data.
Input	Array of joint coordinates
Output	Sends the JSON file to the appropriate functions(F2, F3, F4, F5) as an input
Pre-condition	Movement must be finished.
Post-condition	Pre-processing functions should be ready to receive the output data.
Dependency	Kinect data should be already stored in array from (F7)
Risk	The function may not detect outliers in the data

Table 3.2: Interpolation

Name	Interpolation
Code	F2
Priority	High
Critical	9/10
Description	This function interpolates the movement data before being sent for classification based on the movement templates length so they would be equal for better results.
Input	Captured data (JSON file)
Output	Interpolated data (JSON file)
Pre-condition	Data capturing must be finished (F8) and the data files are ready to be read.
Post-condition	Data uploading function (F9) should be ready to receive and send the output
Dependency	It should be called from (F1)
Risk	The function may remove important points in the data

Table 3.3: Extrapolation

Name	Extrapolation
Code	F3
Priority	High
Critical	9/10
Description	This function extrapolates the movement data before being sent for classification based on the movement templates length so they would be equal for better results.
Input	Captured data (JSON file)
Output	Extrapolated data (JSON file)
Pre-condition	Data capturing must be finished (F8) and the data files are ready to be read.
Post-condition	Data uploading function (F9) should be ready to receive and send the output
Dependency	It should be called from (F1)
Risk	The function may extrapolate unnecessary points in the data

Table 3.4: Removing Outliers

Name	Removing Outliers
Code	F4
Priority	High
Critical	9/10
Description	This function is used to remove any outliers from the data which could be generated from the Kinect due to error in detection.
Input	Captured data (JSON file)
Output	Filtered data (JSON file)
Pre-condition	Data capturing must be finished (F8) and the data files are ready to be read.
Post-condition	Data uploading function (F9) should be ready to receive and send the output
Dependency	It should be called from (F1)
Risk	The function may remove important points

Table 3.5: Segmentation

Name	Segmentation
Code	F5
Priority	Extreme
Critical	10/10
Description	This function will be used to segment the data read from the Kinect, based on key-frames to facilitate the processing, classification and uploading.
Input	Processed data (JSON file)
Output	Multiple JSON file
Pre-condition	Data capturing must be finished (F8) and the data files are ready to be read.
Post-condition	The data is segmented
Dependency	Data should be already processed (F1)
Risk	The function may include unnecessary frames and exclude important frames

3.3.2 Processing

Table 3.6: Classify Movement

Name	Classify Movement
Code	F6
Priority	Extreme
Critical	10/10
Description	This function uses F-DTW to compare the data to the model. After the player movement has been extracted and pre-processed, the server compares the player's movement to existing movement's template, and determine which movement was performed, the mistake type and the movement accuracy.
Input	Template Model, JSON file of player movement.
Output	Array of Movement objects (Name, Mistake type and Accuracy)
Pre-condition	Classification model must be ready on the server, Internet condition must exist.
Post-condition	Report functions(F18)/system receives the output.
Dependency	User must be logged in, F1, F8, F9
Risk	Classifier might misinterpret a move for another or give false accuracy.

Table 3.7: Get Joints Data

name	Get Joints Data
Code	F7
Priority	Extreme
Critical	10/10
Description	This function is fired after 5 seconds countdown from pressing the start recording button. The system will start capturing the skeleton joint coordinates using the Kinect and store them in an array and sent to the pre-processing function(s).
Input	Nothing
Output	Array of joints
Pre-condition	Kinect is working and capturing data, user should be ready to perform movement.
Post-condition	Array is ready to be stored in (JSON file).
Dependency	User should press start recording and ready to perform the movement.
Risk	The Kinect could miss some joints or couldn't work at all.

Table 3.8: Store Data

Name	Store Data
Code	F8
Priority	Extreme
Critical	10/10
Description	This function is fired after capturing the data from Kinect into array, to store our data into files for processing.
Input	Array of joints
Output	Movement data (JSON file)
Pre-condition	Capture Data is working and returning arrays.
Post-condition	files are stored in a file and ready to be processed.
Dependency	Kinect should already captured data in (F7)
Risk	file mustn't be opened or altered while the function works

Table 3.9: File Upload

Name	File Upload
Code	F9
Priority	Extreme
Critical	10/10
Description	This function handles uploading files processes. Files are uploaded to the cloud server to generate reports, process and classify the data.
Input	JSON file
Output	Boolean indicator for successful file upload.
Pre-condition	files must be ready to be read.
Post-condition	Data is received by the cloud
Dependency	F8
Risk	The function may fail to upload a file

3.3.3 Interface

Table 3.10: Check Notification

Name	Check Notification
Code	F10
Priority	Low
Critical	5/10
Description	This function is used to display notification for the coach and players. After the coach adds a comment, the player will be notified and when the player finishes a movement the coach will also be notified. This will add the notification to the database to be represented on the interface
Input	Message, Time, From UserID and To UserID
Output	Record in the database of the notification
Pre-condition	User must have an account to receive notifications
Post-condition	Notification is received
Dependency	Notification system must be ready to check for updates in the database
Risk	Function may fail to check notifications

Table 3.11: CRUD Users

Name	CRUD Users
Code	F11
Priority	high
Critical	7/10
Description	CRUD functions for the users. Users on the system are players and coaches. Their data on the database can be handled by using these functions.
Input	User data
Output	Query updating the database
Pre-condition	User must exist.
Post-condition	The database is updated
Dependency	UI must be ready to submit the data
Risk	Queries may not succeed

Table 3.12: Hashing User passwords

Name	Hashing User passwords
Code	F12
Priority	High
Critical	7/10
Description	This function hashes user passwords in the database and saves the hashes to be compared later, to add more security to the system in case of tampering.
Input	user ID.
Output	Query inserting the hashed password
Pre-condition	User must exist.
Post-condition	User password is hashed in the database
Dependency	None
Risk	Hashing is not powerful

Table 3.13: Compare Hash

Name	Compare Hash
Code	F13
Priority	High
Critical	7/10
Description	This function will compare the password hashes to insure there is no leak of users passwords
Input	Current Hashes/Past Hashes
Output	Query comparing the hashed input with the user password
Pre-condition	Database must be ready to be accessed
Post-condition	User gains access to login
Dependency	F12
Risk	Previous hashes doesn't exist

Table 3.14: Remove Player

Name	Remove Player
Code	F14
Priority	High
Critical	7/10
Description	This function is used by the coach to remove players assigned to him by removing his access to view player records from the database.
Input	Player ID/coach ID
Output	Query removing access record from the database.
Pre-condition	Player/coach ID must exist in the database.
Post-condition	Specified player no longer assigned to the coach
Dependency	Coach must be logged in
Risk	No players assigned which leads to the failure of the function

Table 3.15: CRUD Comments

Name	CRUD Comments
Code	F15
Priority	Medium
Critical	6/10
Description	These are CRUD function for the comments formed by the coach to players. These comments will be attached to the player record, for each player to be updated with his coach hints and opinion.
Input	Record ID/coach ID
Output	Comment data (Text, Coach ID, Record ID).
Pre-condition	Player, Coach and Record must exist in the database.
Post-condition	Comment is inserted in the database
Dependency	Coach must be logged in
Risk	Function may fail to insert the comment

Table 3.16: Select Player

Name	Select Player
Code	F16
Priority	High
Critical	7/10
Description	This function is used by the coach to select players to be under his supervision so he could view all his records in details.
Input	Player ID/coach ID
Output	Record in the database.
Pre-condition	Player/coach ID must exist in the users database.
Post-condition	UI must be ready to receive and display the output of this function
Dependency	Coach must be logged in to use this function
Risk	No players in the database to select from

3.3.4 Report

Table 3.17: View all Records

Name	View all Records
Code	F17
Priority	High
Critical	8/10
Description	This function retrieves a specified player's records and display its details to the user like the Name, accuracy, score, mistake, etc...
Input	Player ID
Output	Array of records which contains each record info
Pre-condition	Check if the player exists on the records database.
Post-condition	UI must be ready to receive and display the output of this function
Dependency	User must be logged in to use this function
Risk	If there is no records the function will fail

Table 3.18: Calculate Overall Score

Name	Calculate Overall Score
Code	F18
Priority	Extreme
Critical	8/10
Description	This function calculates the overall score of the player after performing the movement, based on the move performance accuracy retrieved from the database and the scoring criteria.
Input	Movement name, accuracy, mistake type
Output	Store the record score into the database
Pre-condition	‘Karate move’ accuracy/classification.
Post-condition	None
Dependency	F6
Risk	The movement accuracy must be calculated in a right and accurate way

Table 3.19: View Progress

Name	View Progress
Code	F19
Priority	High
Critical	7/10
Description	This function will be used after retrieving the records from the database for the user to see his history and his overall progress and this data should be represented in statistical graphs.
Input	Array of all user records
Output	Statistics on the screen
Pre-condition	User must have records
Post-condition	None
Dependency	F18
Risk	Not enough records which could lead to non-accurate statistics/Internet failure

Table 3.20: View Report

Name	View Report
Code	F20
Priority	High
Critical	7/10
Description	This function will be used to view the report of the current training directly after the player finishes the training.
Input	Array of joint's coordinates
Output	Movements names, rates, mistakes
Pre-condition	User must exist
Post-condition	None
Dependency	F6
Risk	The movement accuracy must be calculated in a right and accurate way

3.4 Interface Requirements

3.4.1 User Interfaces

This is system is a multi-platform application, which is mainly based on a desktop application to record the Kinect and have real-time feedback with reports and statistics displayed on a web application. So mainly the system will not have any command line interface but will have simplified, easy to use GUI.

3.4.1.1 GUI

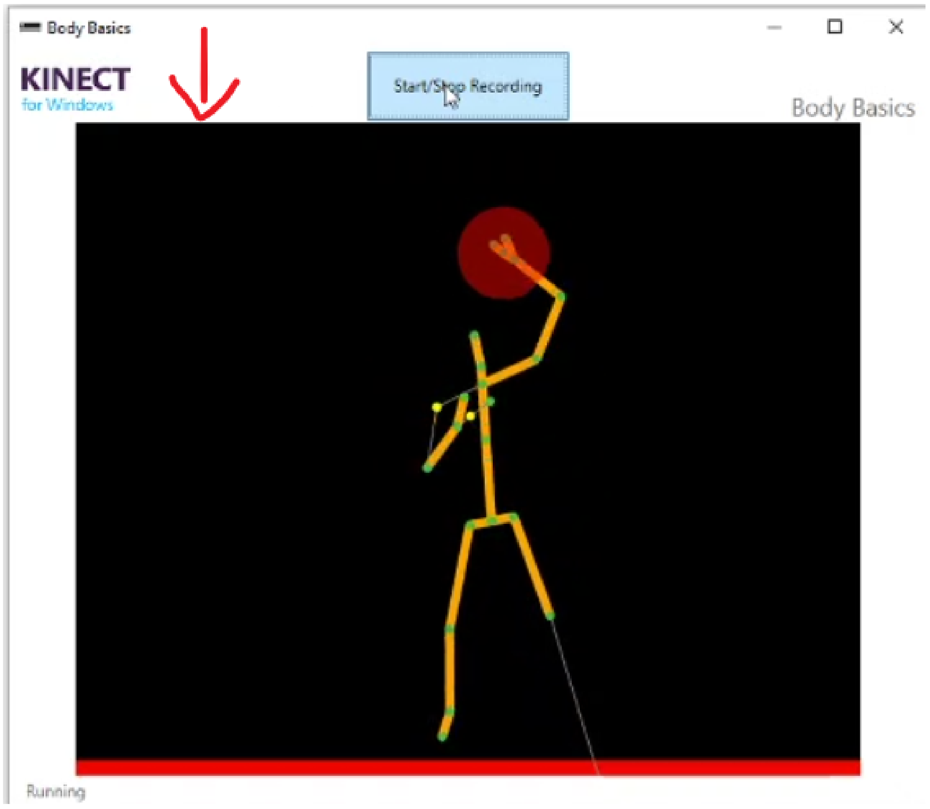


Figure 3.5: Motion Detection Window: Player's Skeleton

Fig: 3.5 Shows the Screen where the visualization of the practitioner movement is displayed.

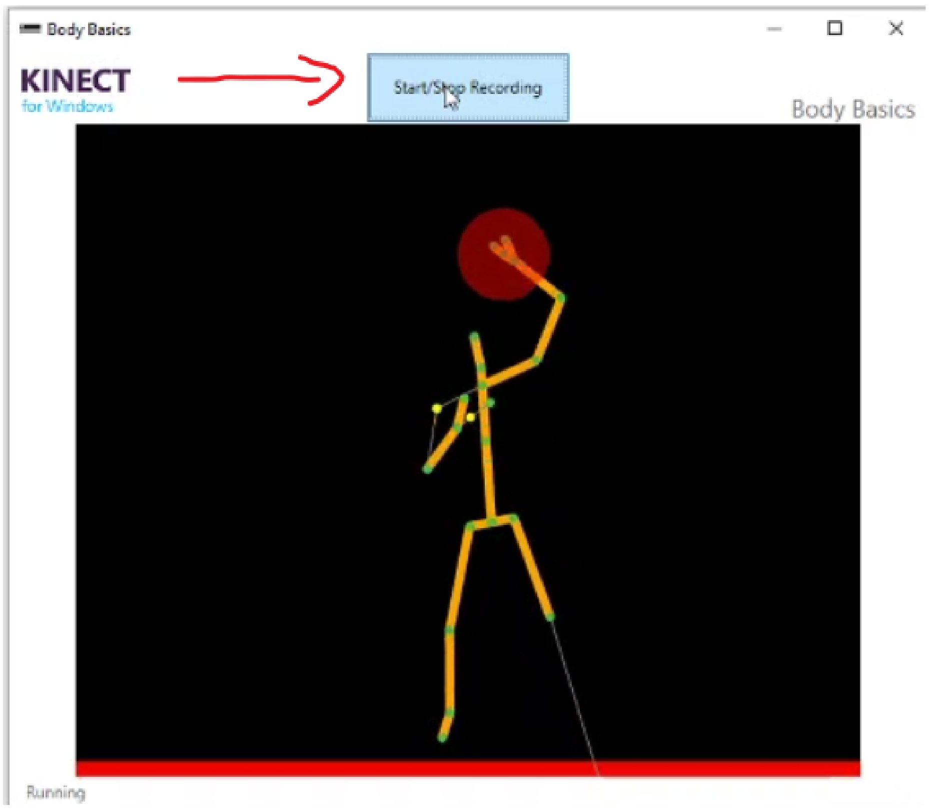
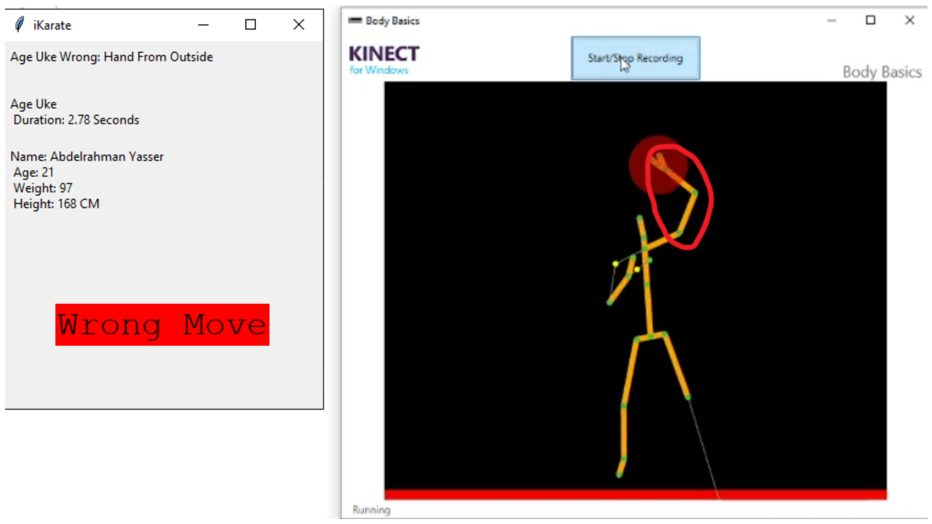


Figure 3.6: Motion Detection Window

Fig: 3.6 Shows the "Start/Stop Recording Button" this button will be used to start or stop capturing the data.


iKarate

Age Uke Wrong: Hand From Outside

Age Uke
Duration: 2.78 Seconds

Name: Abdelrahman Yasser
Age: 21
Weight: 97
Height: 168 CM

Wrong Move

Figure 3.7: Motion Detection Window: Start/Stop Recording Button

Fig: 3.7 Shows how will be the final output and the final screen of the system, where it displays the players data and informs him if the movement is wrong or correct.

3.4.2 Hardware Interfaces

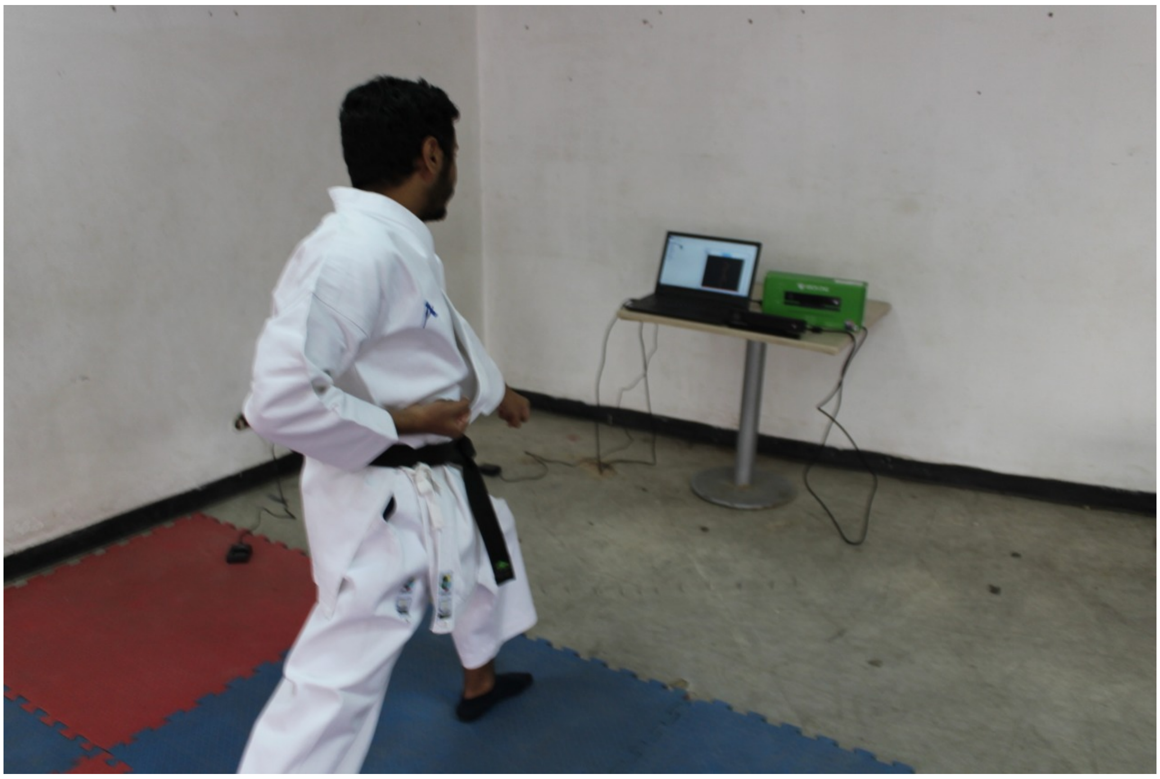


Figure 3.8: Motion Detection Window: System's Feedback

In Fig: 3.8 Shows the hardware setup. Where the Kinect should be connected to a computer and facing the Player to capture all his movements correctly. Also, the player shouldn't exceed 6 meters between the Kinect for better accuracy.

3.4.3 API

1. Kinet API
2. FastDTW
3. scipy
4. numpy
5. tkinter
6. sklearn
7. jinja

3.5 Performance Requirements

iKarate makes use of the Kinect motion capture capability. The system must be able to process multiple joints (25 if we take all joints), each joint contains three coordinates (X, Y and Z) per frame. Also, The system should be able to handle multiple large data-sets for modeling.

3.6 Design Constraints

3.6.1 Hardware Limitations

Virtual recognition device must be Kinect V2 to provide more joints, higher accuracy and capture further moving objects.

3.7 Other non-functional attributes

3.7.1 Security

Player's recordings saved on the system should only be accessed by the coach and the player. Personal user login data should be saved securely.

3.7.2 Reliability

The system must be reliable in its main classification and motion detection functions, as the operations and data reading must be accurate. When a wrong move is detected in the classification process, the system should identify the mistake correctly with no mistakes.

3.7.3 Maintainability

The system should have the ability to be improved by enhancing the accuracy and adding more moves. Movement data should be stored before being processed. Mistakes made by the system should be recorded to further improve the system.

3.7.4 Usability

The system main functions should be easy to use for the user with the least amount of steps required to perform a certain task.

3.7.5 Availability

The system servers should always be running when needed and the database should always be accessible.

3.7.6 Scalability

The system must be scalable. It can be upgraded to add more processes, data, storage and models to the system to expand its capacity and capabilities.

3.8 Preliminary Object-Oriented Domain Analysis

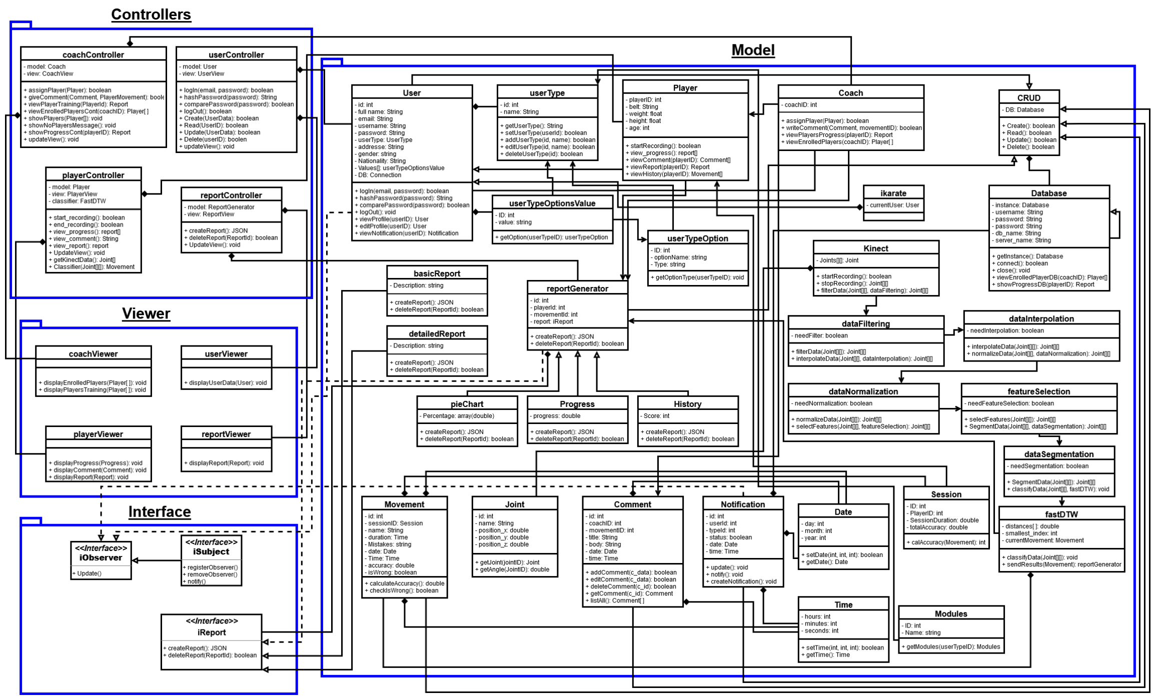


Figure 3.9: Class Diagram

3.8.1 Database Diagram

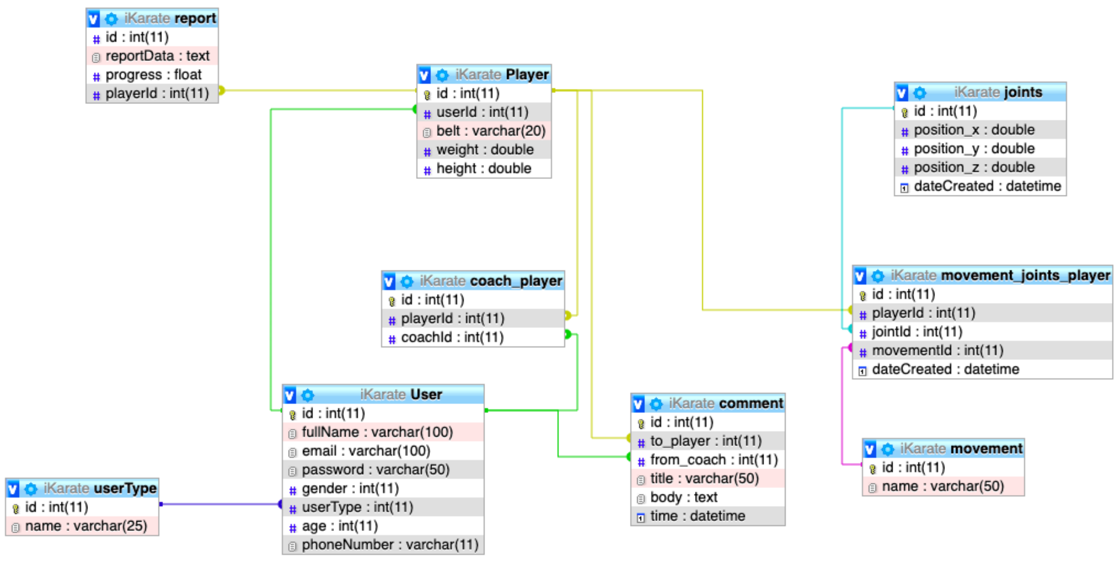


Figure 3.10: Database Diagram

3.9 Operational Scenarios

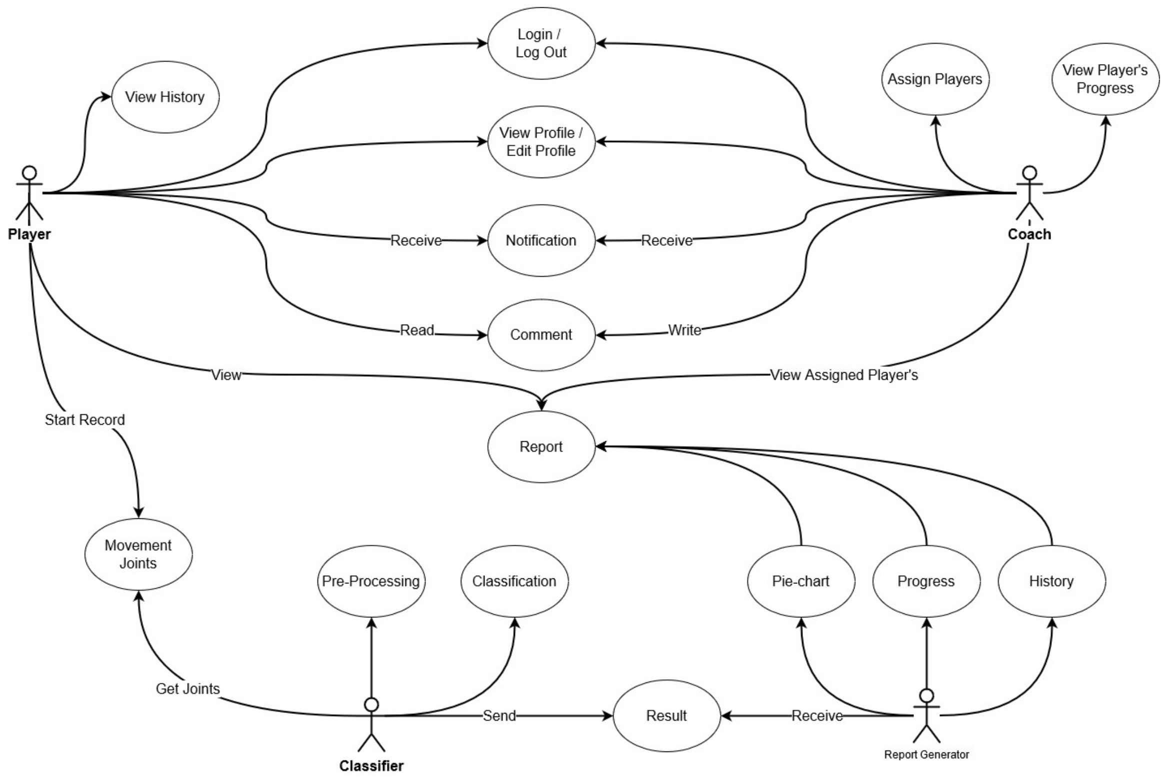


Figure 3.11: Use Case Diagram

Scenario 1: Karate performance The Karate player starts performing and the virtual recognition device capture his/her movement. The system receives the movement data and processes it.

Scenario 2: players accounts/profiles The coach handles his/her student's accounts/profiles. Or in case of the system being used by an independent player; he/she handles their own account/profile. Handlers control certain functions in the system: - Coaches add students/-players to the system

- Player's information can be edited.

- Coaches remove students/players from the system.

- View all players, search and filter them.

Scenario 3: movement detection and classification After the movement data is processed, it is analyzed by the system to be classified. This data is being processed and analyzed as the player is performing and the virtual device is capturing his/her movement. When the movement data reading shows a high probability of a certain movement is currently being performed, the system predicts the movement then classify if it was done correctly.

3.10 Preliminary Schedule Adjusted

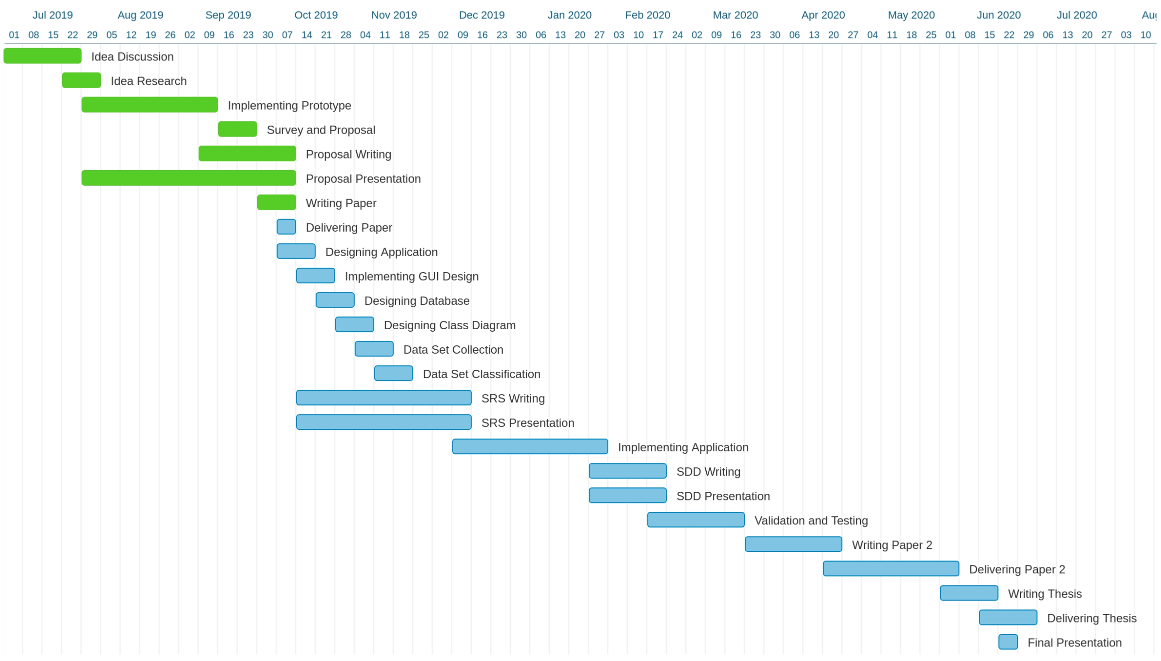


Figure 3.12: Tasks And Time Plan

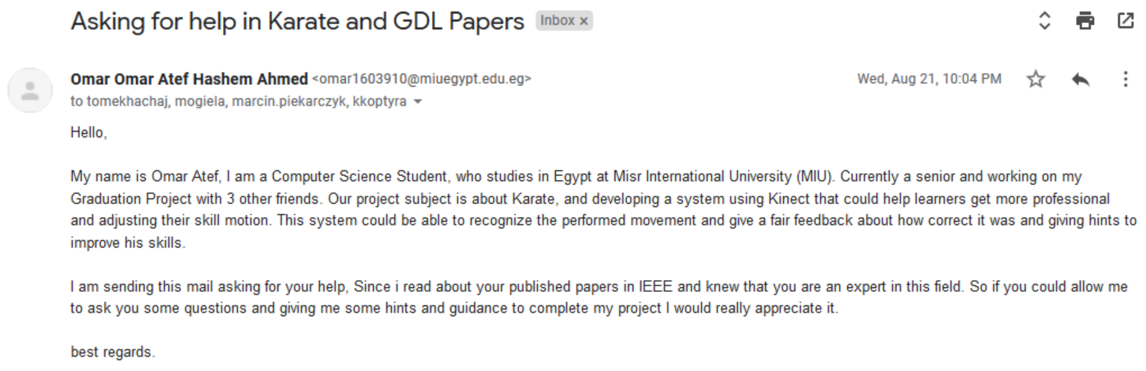
3.11 Preliminary Budget Adjusted

1. Kinect Sensor And Adapter Bundle: 280$.
2. Azure Storage: 0.058$/GB Per Month.
3. Azure SQL Database 4 VCORE: 1.01$/Hour.
4. Azure Machine Learning.

3.12 Appendices

3.12.1 Collected material

Those are the mails sent to the authors of [13] to get their data-set and ask for guidance.



Asking for help in Karate and GDL Papers Inbox ×

Omar Omar Atef Hashem Ahmed <omar1603910@miuegypt.edu.eg> Wed, Aug 21, 10:04 PM
to tomekhachaj, mogiela, marcin.piekarczyk, kkoptyra

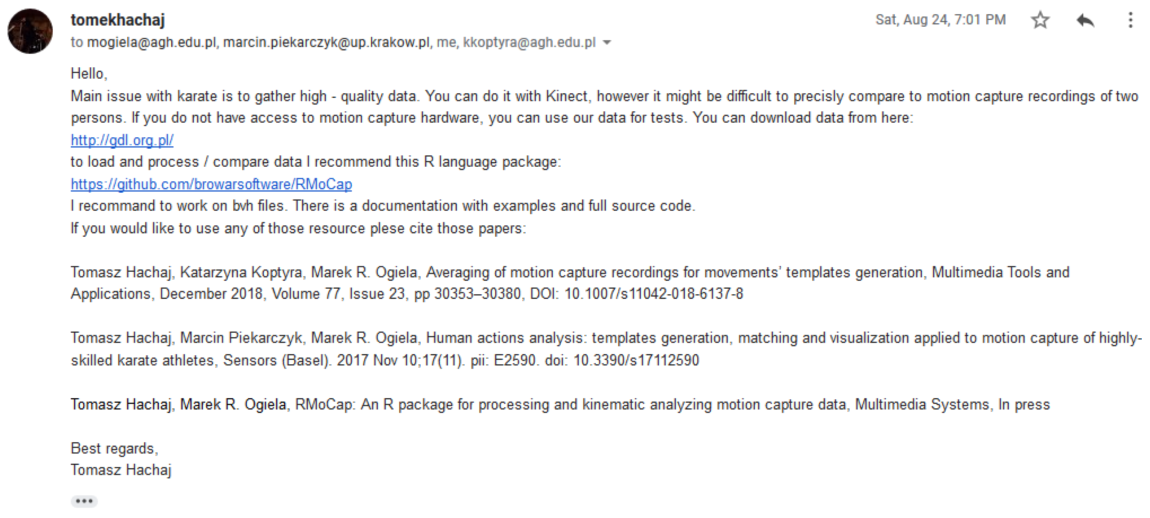
Hello,

My name is Omar Atef, I am a Computer Science Student, who studies in Egypt at Misr International University (MIU). Currently a senior and working on my Graduation Project with 3 other friends. Our project subject is about Karate, and developing a system using Kinect that could help learners get more professional and adjusting their skill motion. This system could be able to recognize the performed movement and give a fair feedback about how correct it was and giving hints to improve his skills.

I am sending this mail asking for your help, Since i read about your published papers in IEEE and knew that you are an expert in this field. So if you could allow me to ask you some questions and giving me some hints and guidance to complete my project I would really appreciate it.

best regards.

Figure 3.13: Data-Set 1/2



tomekhachaj Sat, Aug 24, 7:01 PM
to mogiela@agh.edu.pl, marcin.piekarczyk@up.krakow.pl, me, kkoptyra@agh.edu.pl

Hello,
Main issue with karate is to gather high - quality data. You can do it with Kinect, however it might be difficult to precisly compare to motion capture recordings of two persons. If you do not have access to motion capture hardware, you can use our data for tests. You can download data from here:
http://gdl.org.pl/
to load and process / compare data I recommend this R language package:
https://github.com/browarsoftware/RMoCap
I recommand to work on bvh files. There is a documentation with examples and full source code.
If you would like to use any of those resource plese cite those papers:

Tomasz Hachaj, Katarzyna Koptyra, Marek R. Ogiela, Averaging of motion capture recordings for movements' templates generation, Multimedia Tools and Applications, December 2018, Volume 77, Issue 23, pp 30353–30380, DOI: 10.1007/s11042-018-6137-8

Tomasz Hachaj, Marcin Piekarczyk, Marek R. Ogiela, Human actions analysis: templates generation, matching and visualization applied to motion capture of highly-skilled karate athletes, Sensors (Basel). 2017 Nov 10;17(11). pii: E2590. doi: 10.3390/s17112590

Tomasz Hachaj, Marek R. Ogiela, RMoCap: An R package for processing and kinematic analyzing motion capture data, Multimedia Systems, In press

Best regards,
Tomasz Hachaj

Figure 3.14: Data-Set 2/2

Chapter 4

Software Design Document

4.1 Introduction

4.1.1 Purpose

This chapter 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.

4.1.2 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.

4.1.3 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.

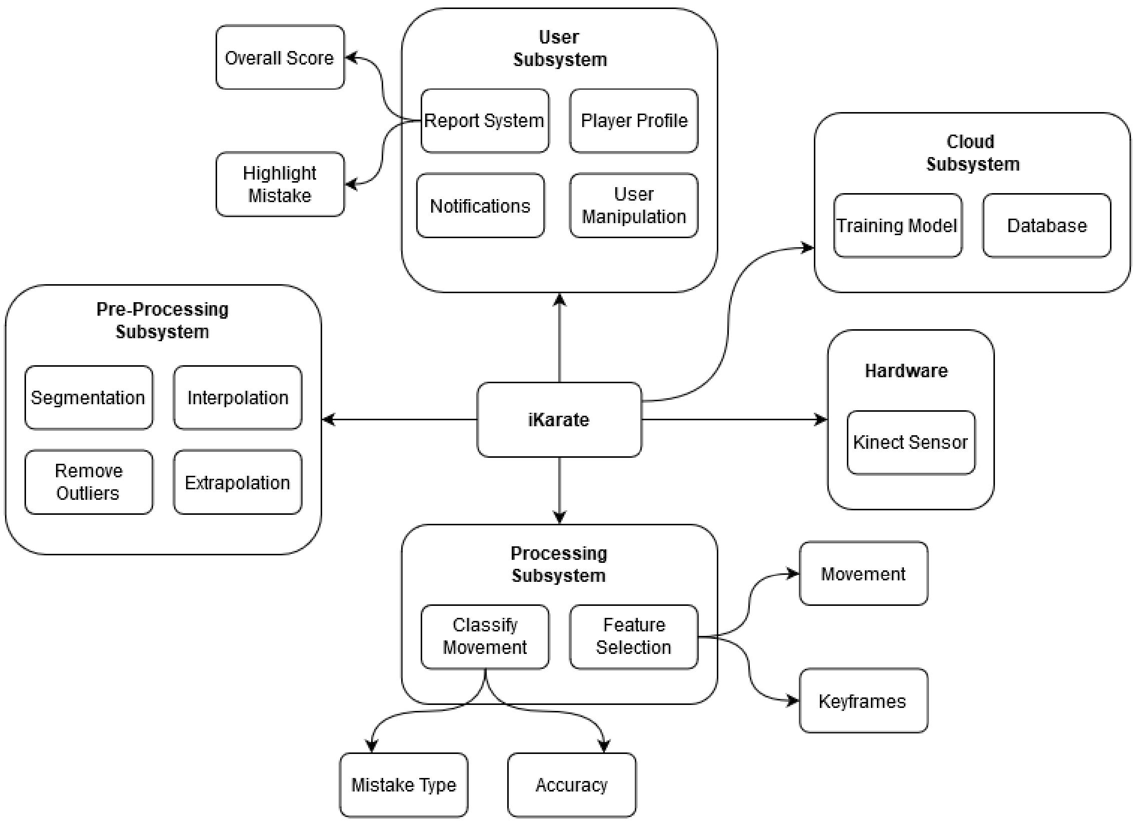


Figure 4.1: Block Diagram

4.2 System Architecture

4.2.1 Architectural Design

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

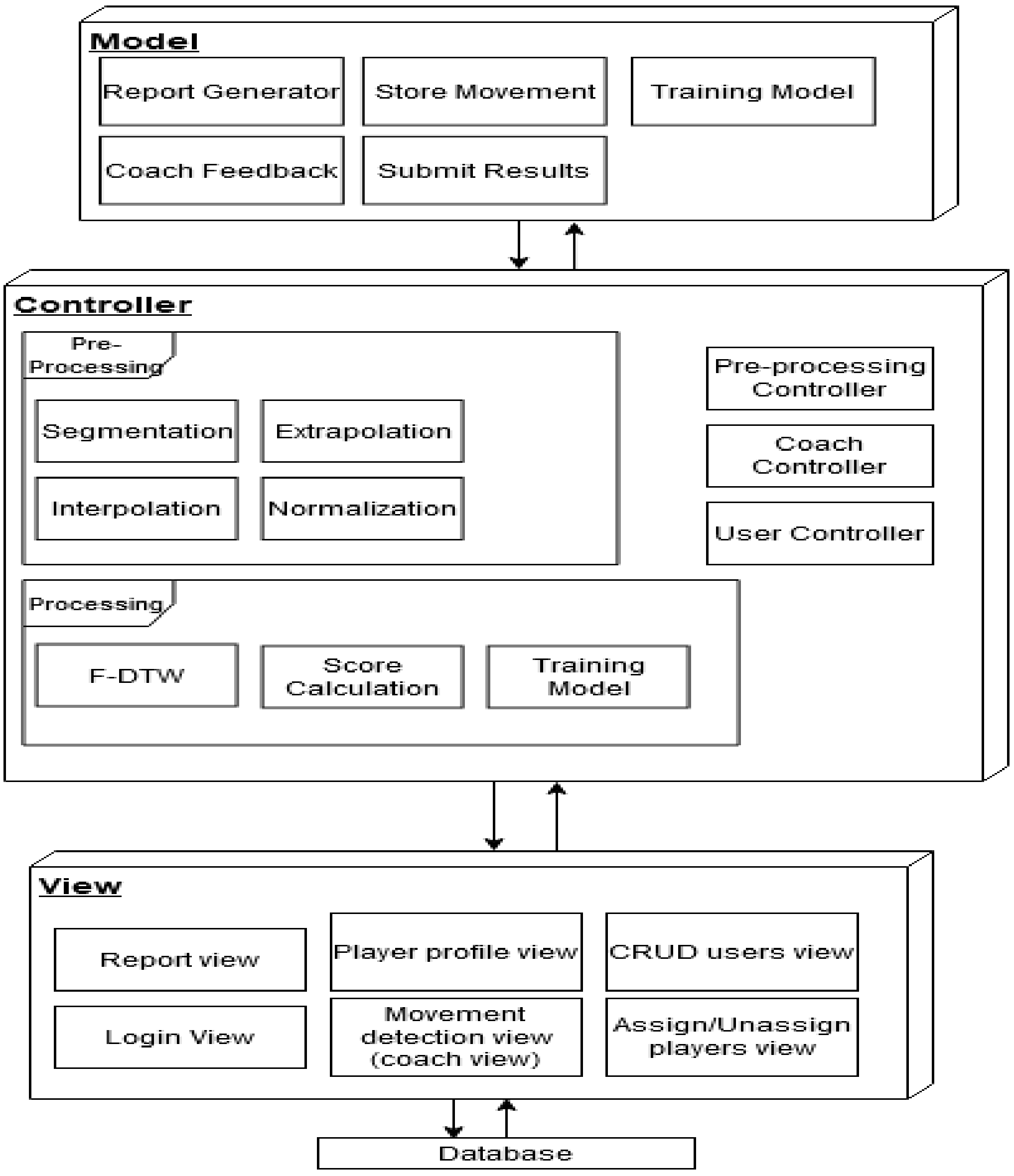


Figure 4.2: Architecture Diagram

4.2.1.1 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.

4.2.1.2 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.

4.2.1.3 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.

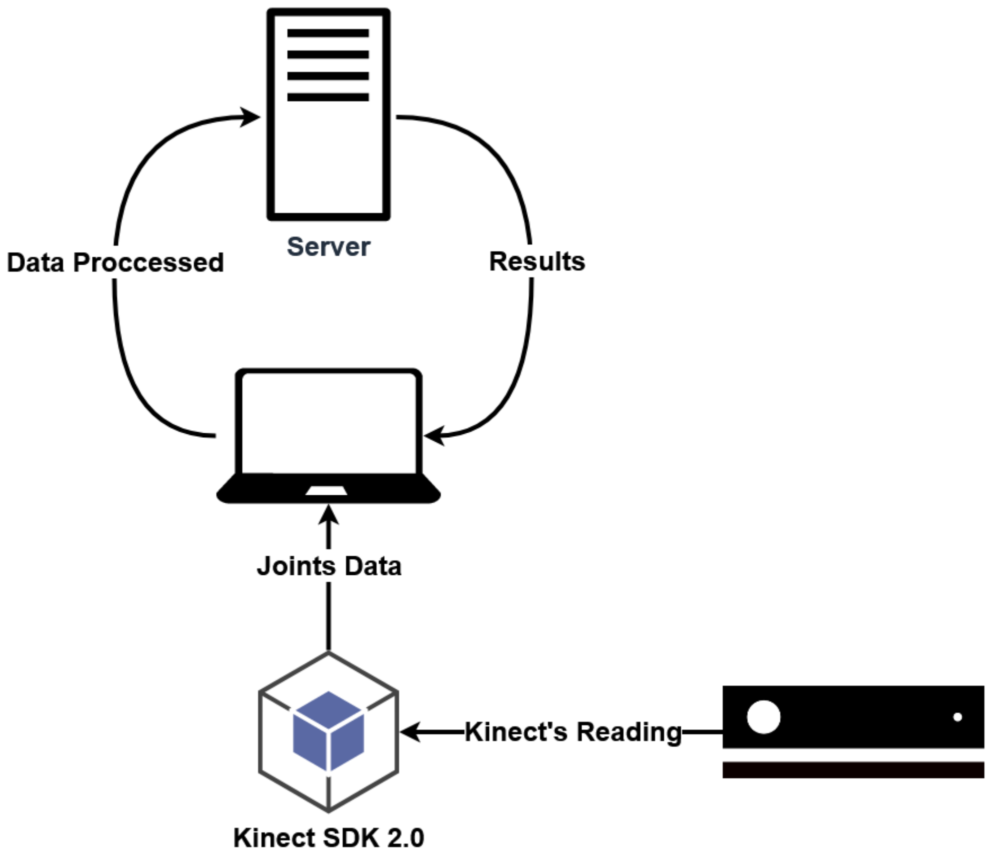


Figure 4.3: Hardware Architecture Diagram

4.2.2 Decomposition Description

4.2.2.1 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.

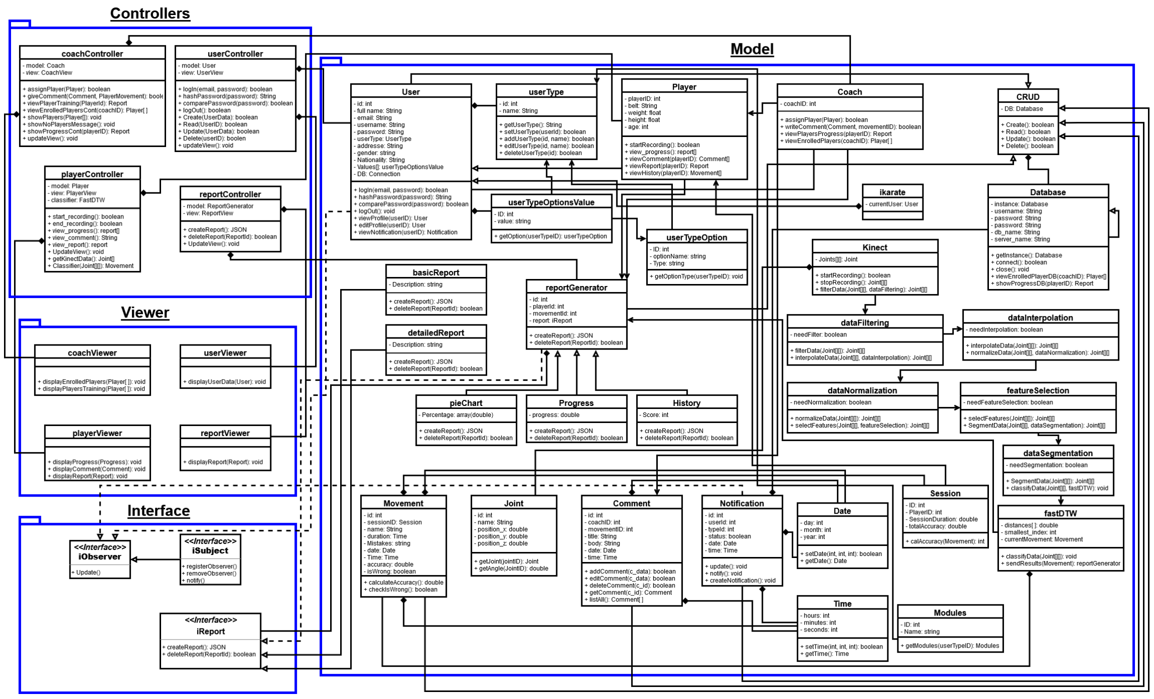


Figure 4.4: Class Diagram

4.2.2.2 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.

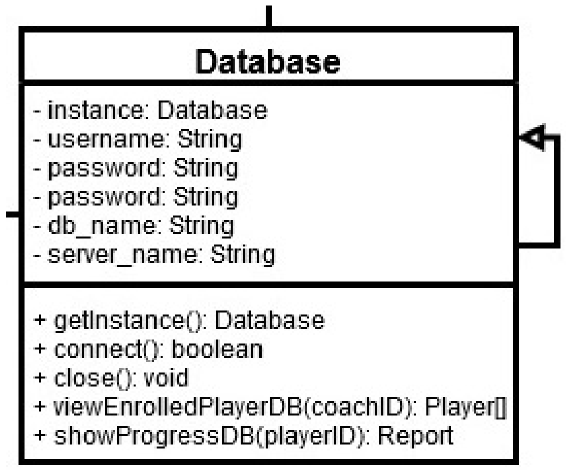


Figure 4.5: Singleton Design Pattern

4.2.2.3 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.

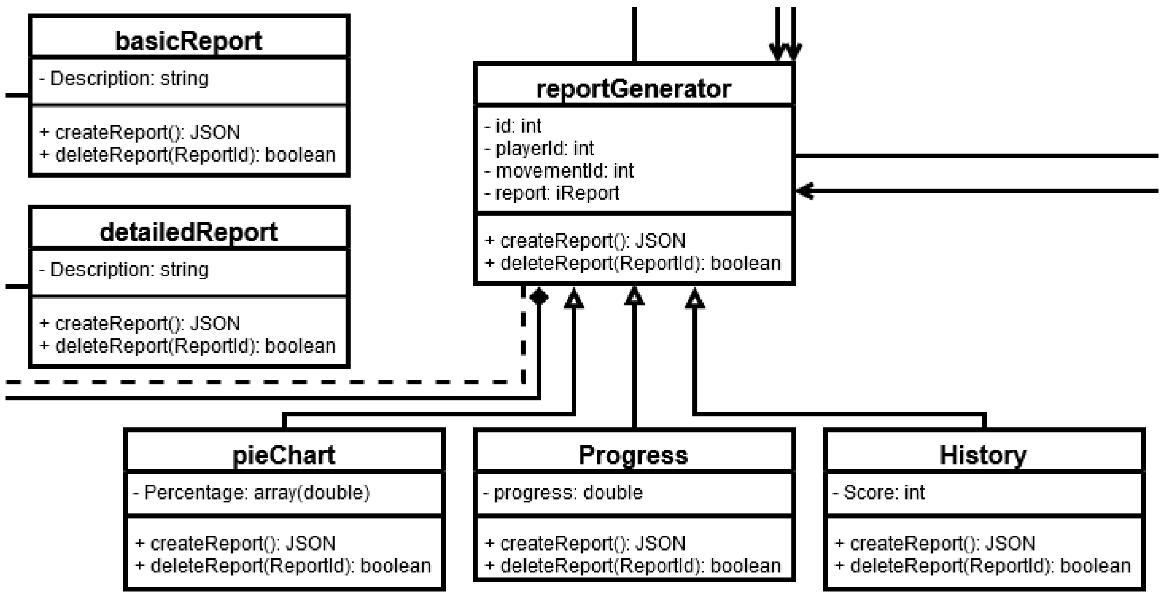


Figure 4.6: Decorative Design Pattern

4.2.2.4 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.

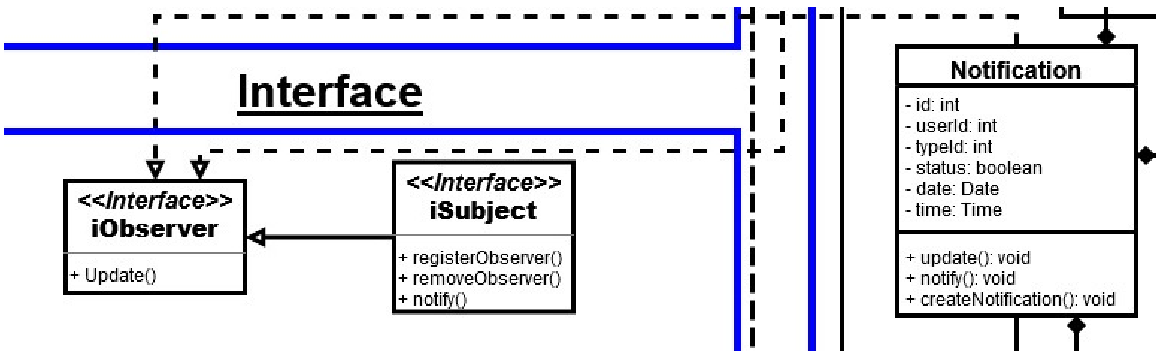


Figure 4.7: Observer Design Pattern

4.2.2.5 Activity Diagram

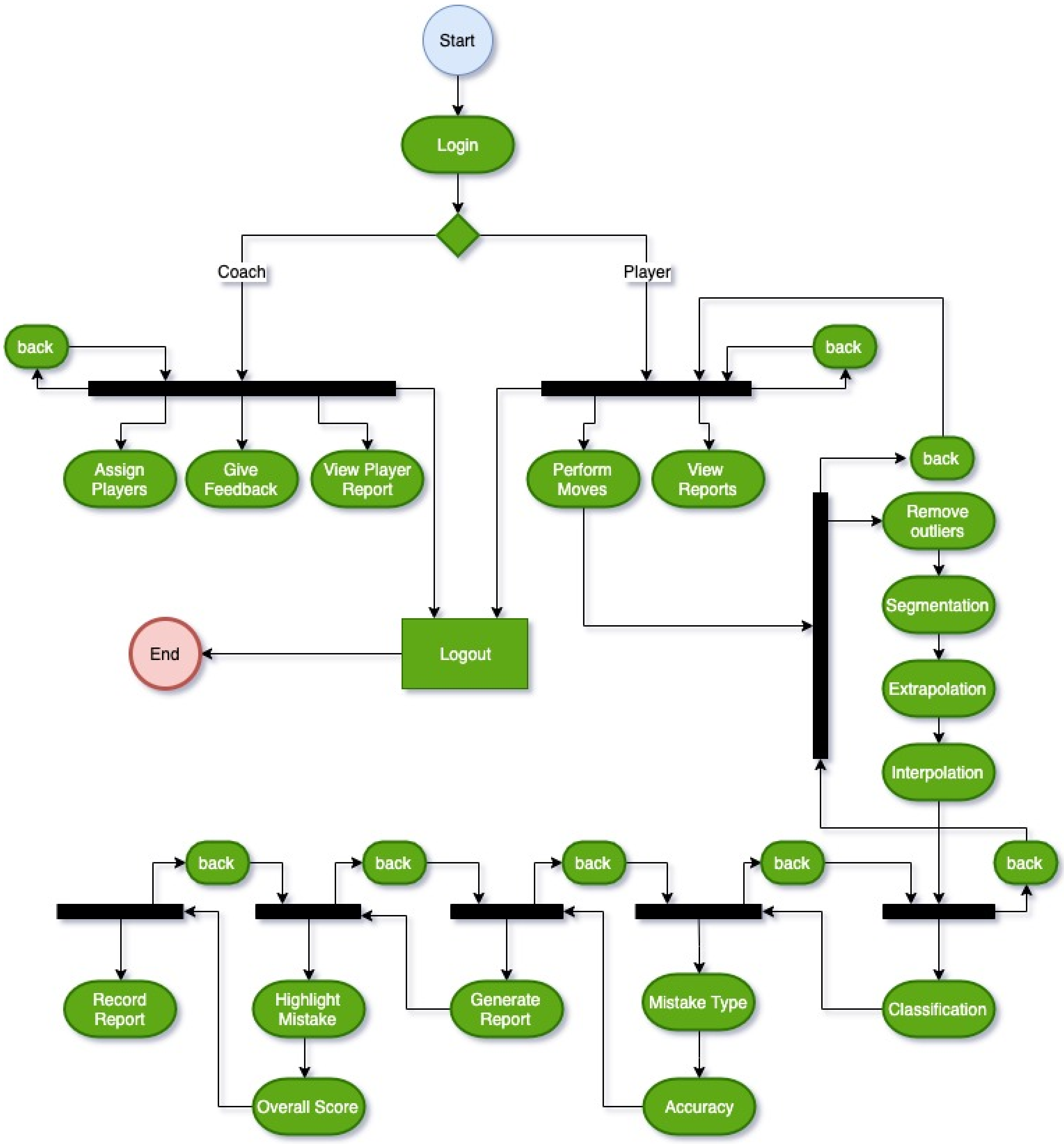


Figure 4.8: Activity Diagram

4.2.2.6 System Sequence Diagram

4.2.2.6.1 Recording, Pre-processing & Processing: As shown in figure 4.9, 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.

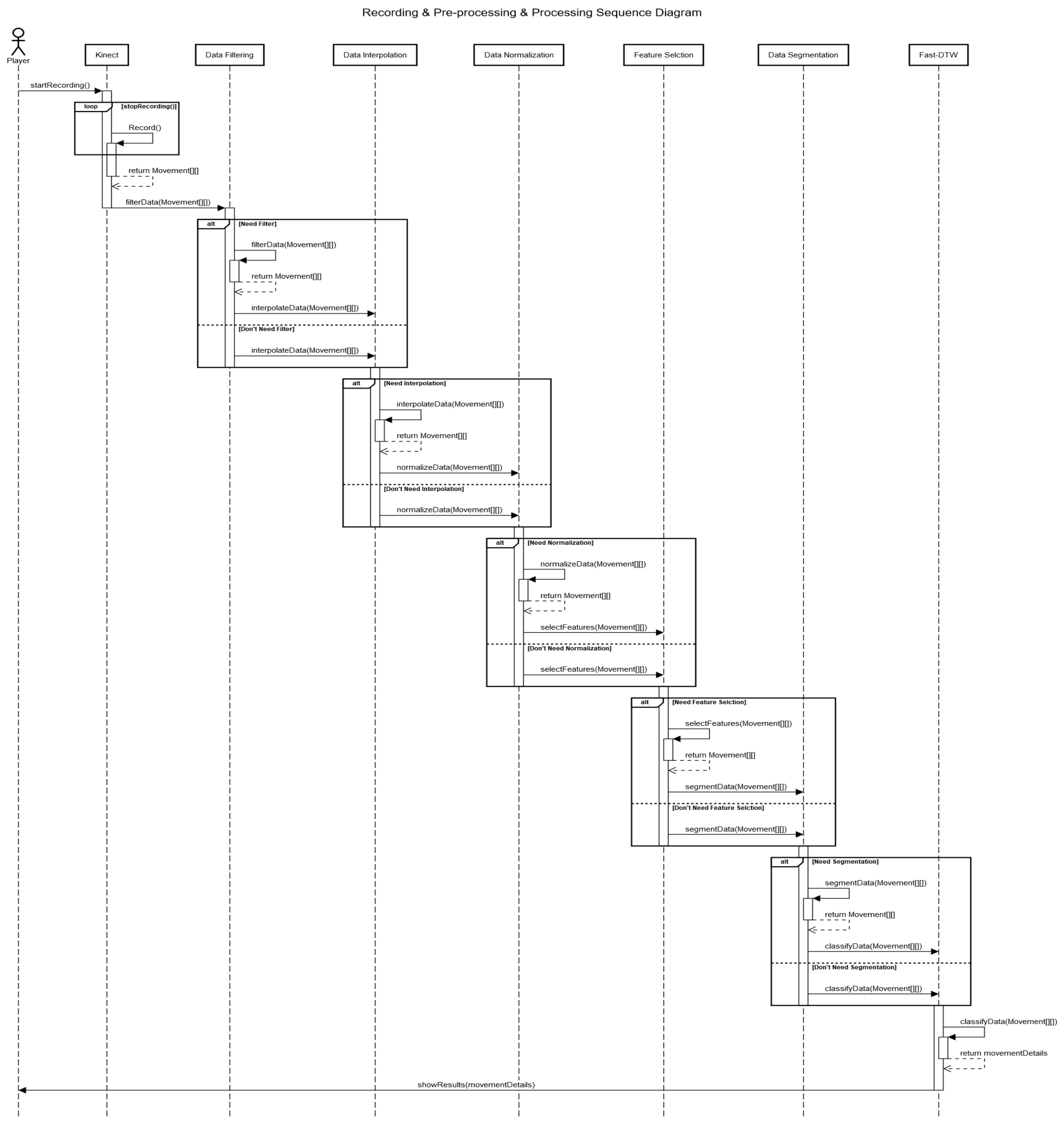

Recording & Pre-processing & Processing Sequence Diagram
Player
Kinect
Data Filtering
Data Interpolation
Data Normalization
Feature Selction
Data Segmentation
Fast-DTW
startRecording()
loop [stopRecording()]
Record()
return Movement[][]
filterData(Movement[][])
alt [Need Filter]
filterData(Movement[][])
return Movement[][]
interpolateData(Movement[][])
[Don't Need Filter]
interpolateData(Movement[][])
alt [Need Interpolation]
interpolateData(Movement[][])
return Movement[][]
normalizeData(Movement[][])
[Don't Need Interpolation]
normalizeData(Movement[][])
alt [Need Normalization]
normalizeData(Movement[][])
return Movement[][]
selectFeatures(Movement[][])
[Don't Need Normalization]
selectFeatures(Movement[][])
alt [Need Feature Selction]
selectFeatures(Movement[][])
return Movement[][]
segmentData(Movement[][])
[Don't Need Feature Selction]
segmentData(Movement[][])
alt [Need Segmentation]
segmentData(Movement[][])
return Movement[][]
classifyData(Movement[][])
[Don't Need Segmentation]
classifyData(Movement[][])
classifyData(Movement[][])
return movementDetails
showResults(movementDetails)

Figure 4.9: Pre-Processing & Processing Sequence Diagram

4.2.2.6.2 View History: As shown in figure 4.10, the user requests the history of a specific player which is retrieved from the database then the history is displayed on the screen.

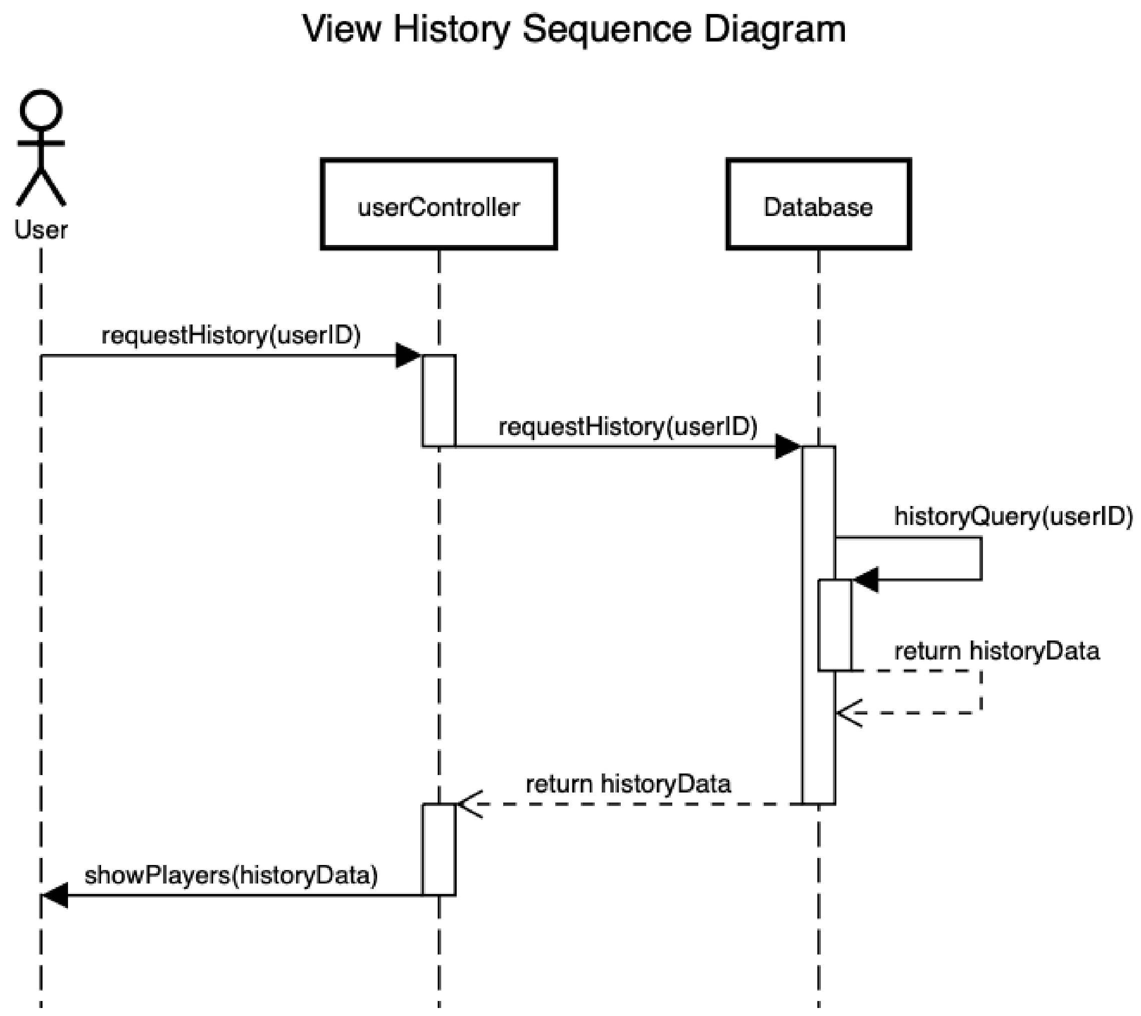


Figure 4.10: View History Sequence Diagram

4.2.2.6.3 Progress: As shown in figure 4.11, the coach request the progress of one of his students, the progress is retrieved from the database and displayed to the coach.

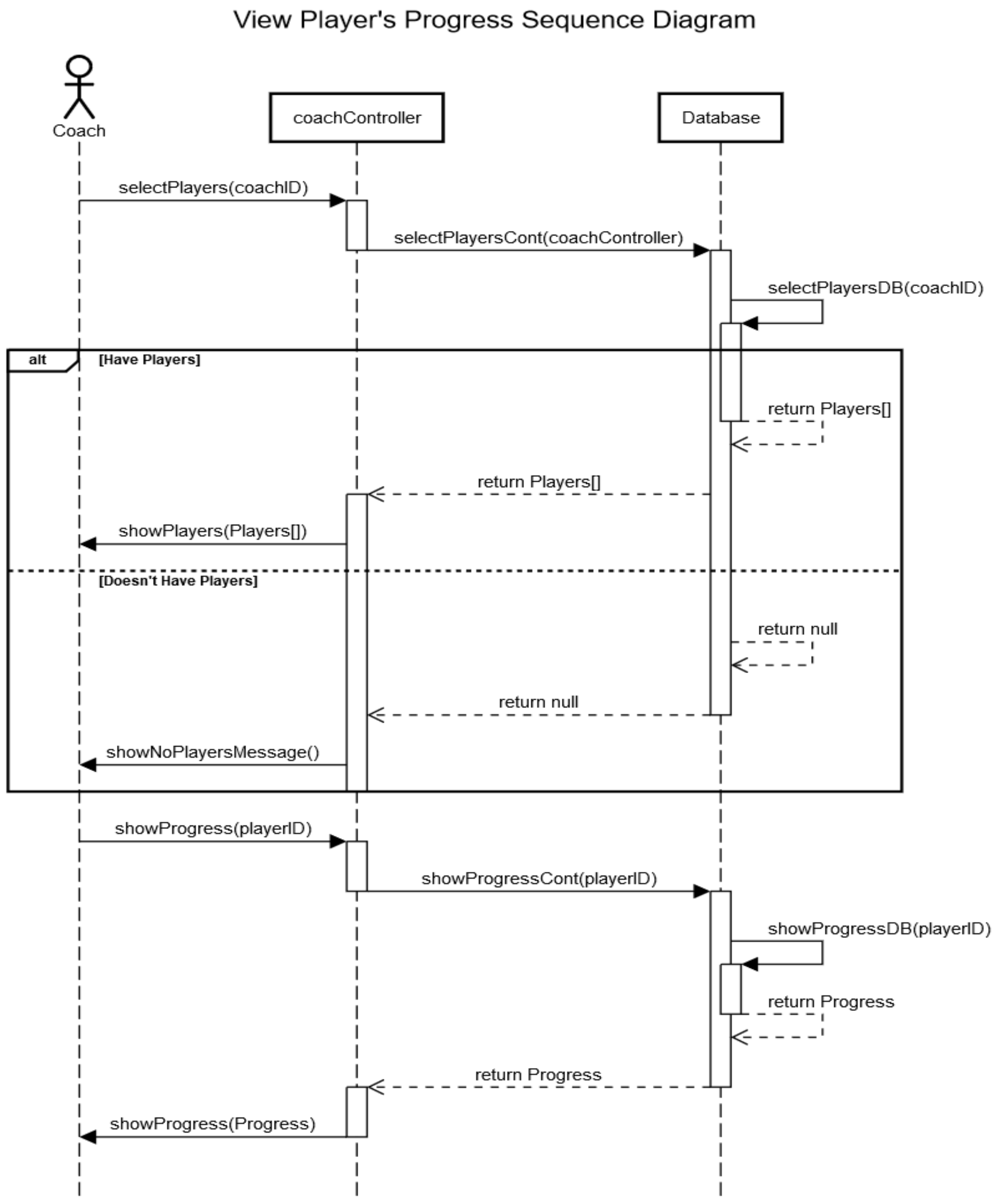


Figure 4.11: View Player's Progress Sequence Diagram

4.2.2.6.4 Assign Player: As shown in figure 4.12, 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.

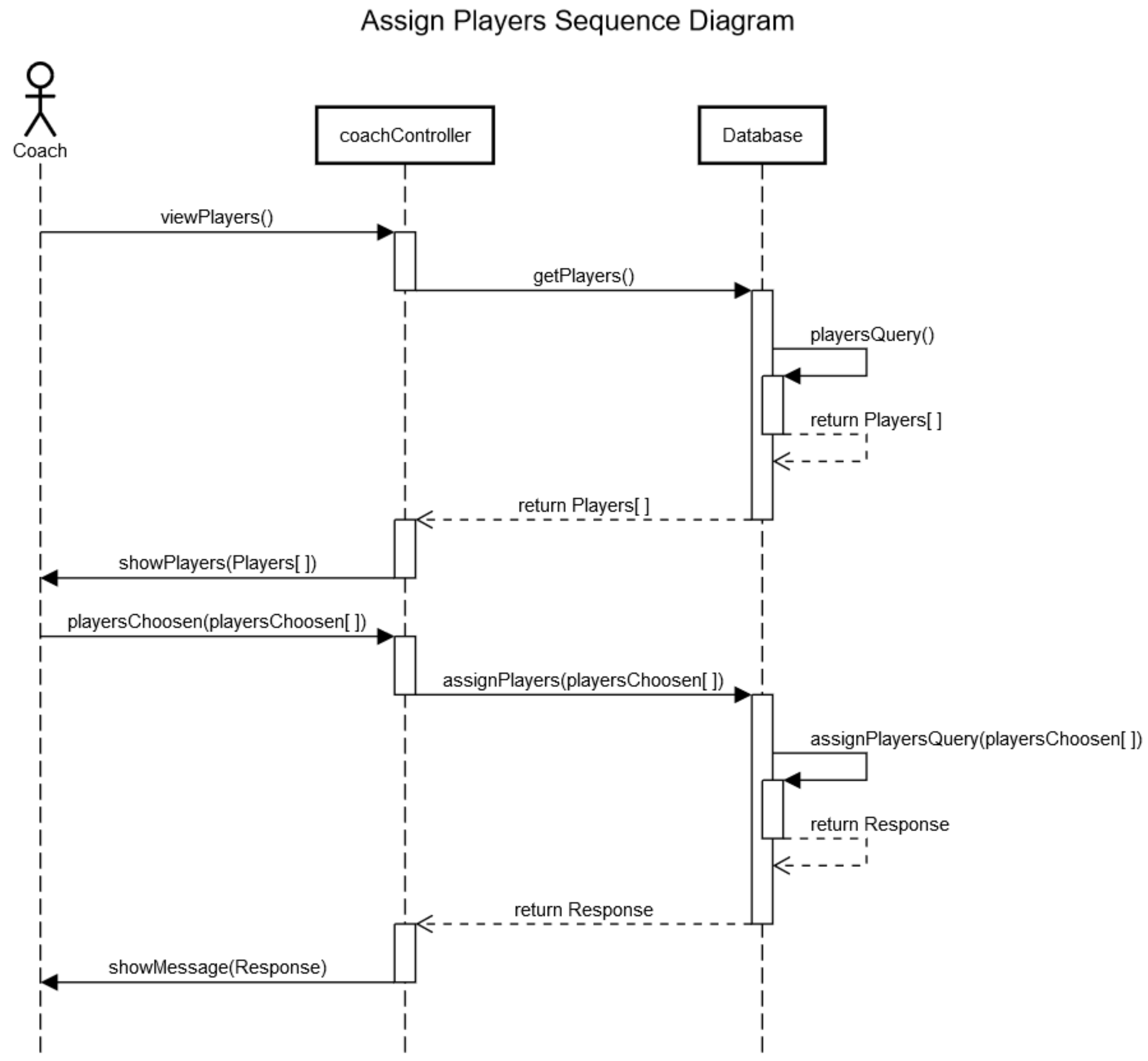


Figure 4.12: Assign Players Sequence Diagram

4.2.2.6.5 Login: As shown in figure 4.13, 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.

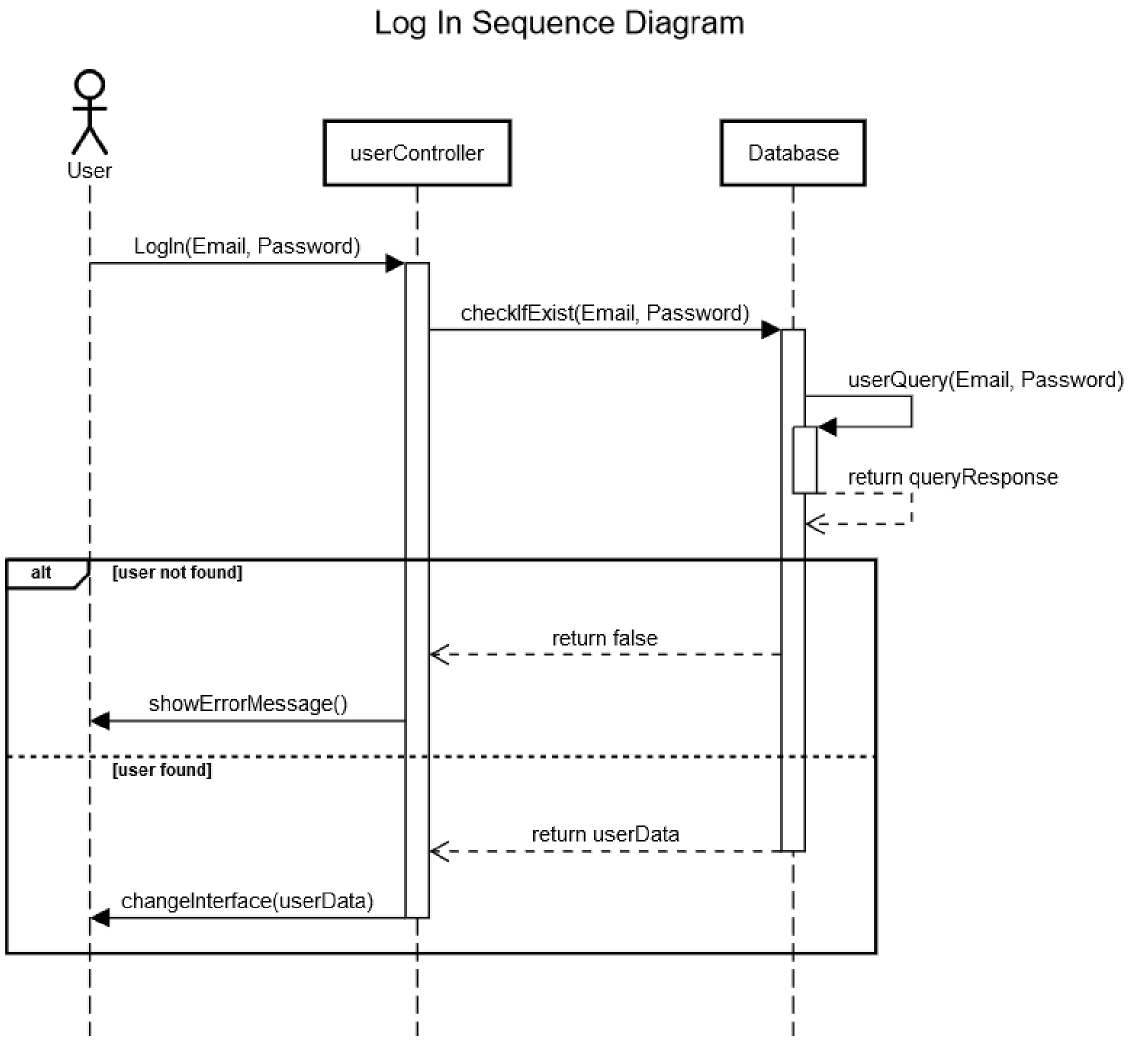


Figure 4.13: Log In Sequence Diagram

4.2.3 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.

4.3 Data Design

4.3.1 Data Description

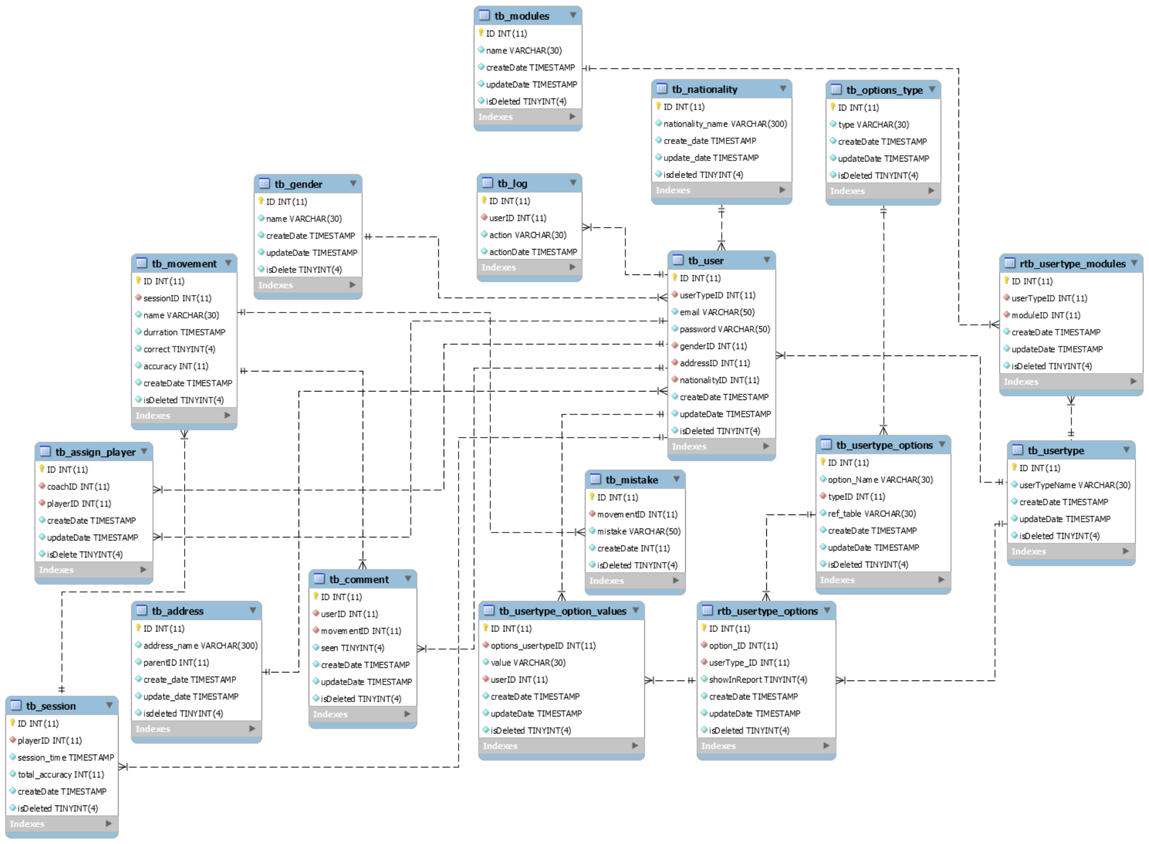


Figure 4.14: Database Diagram

4.3.2 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 collects their values in another table, to deliver highly dynamic system.

4.4 Component Design

4.4.1 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.

4.4.2 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".

$$X = \frac{Value - \mu}{\sigma} \tag{4.1}$$

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.

4.4.3 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.

4.4.4 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.

$$d = \sum_{x_i, y_i}^{n} \sqrt{(x_i - y_i)^2} \tag{4.2}$$

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

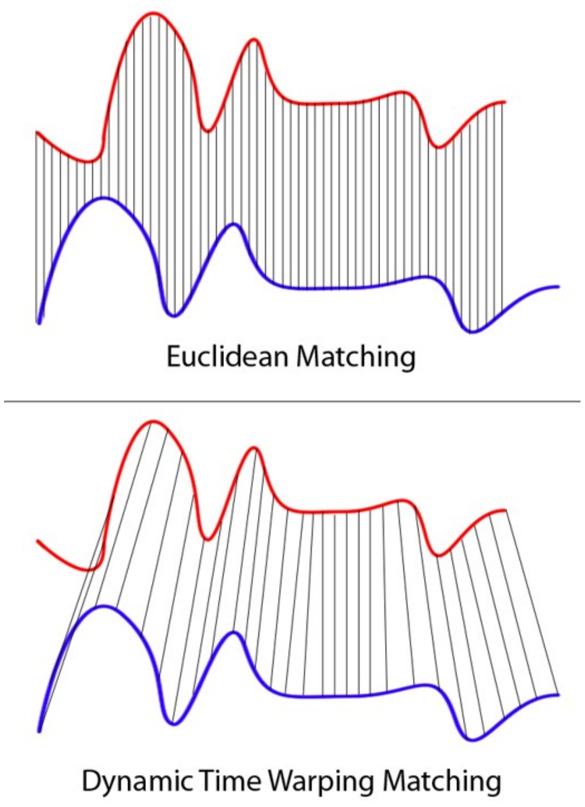


Figure 4.15: Fast-DTW

4.4.5 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.

4.5 Human Interface Design

4.5.1 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.

4.5.2 Screen Images

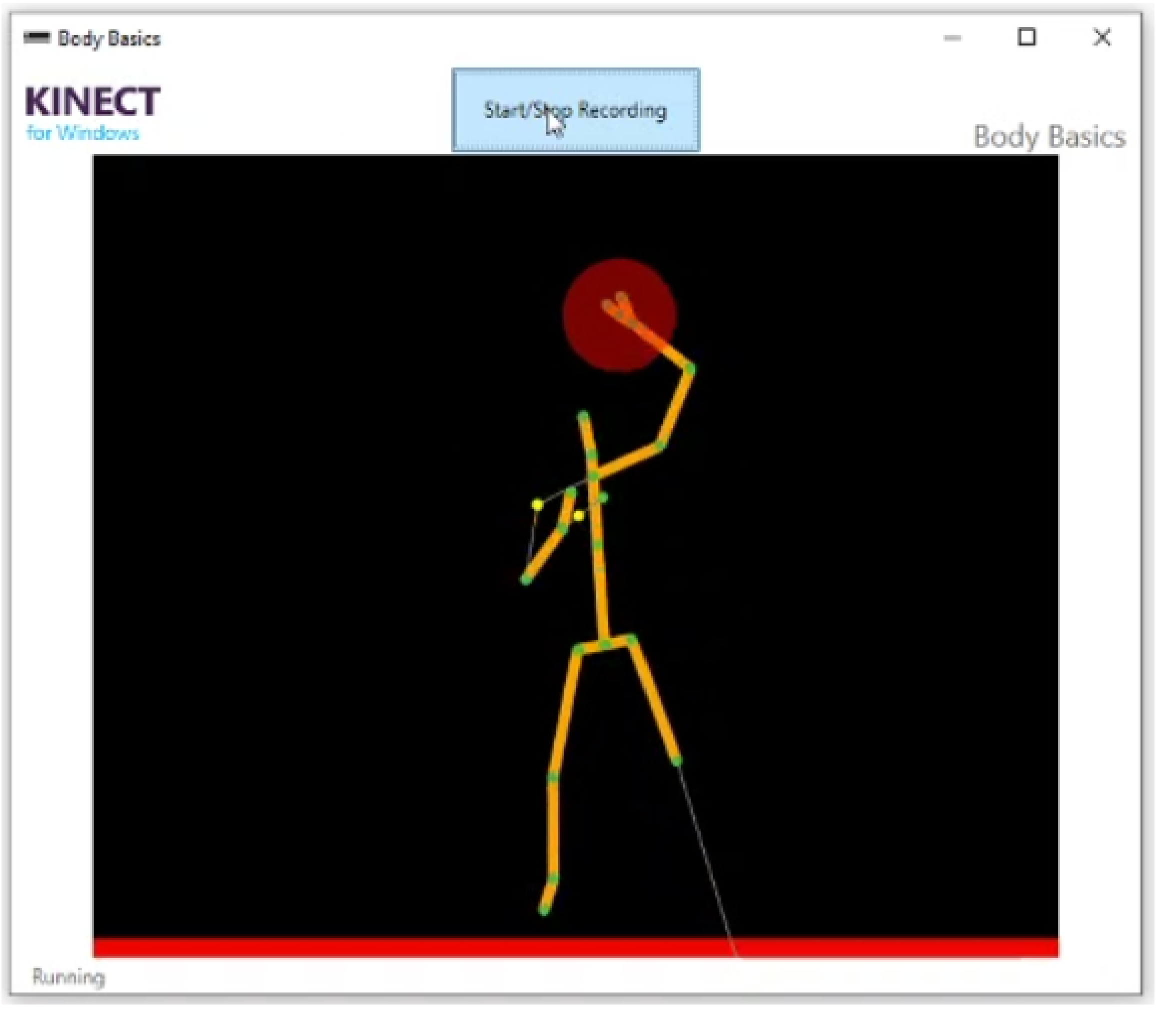
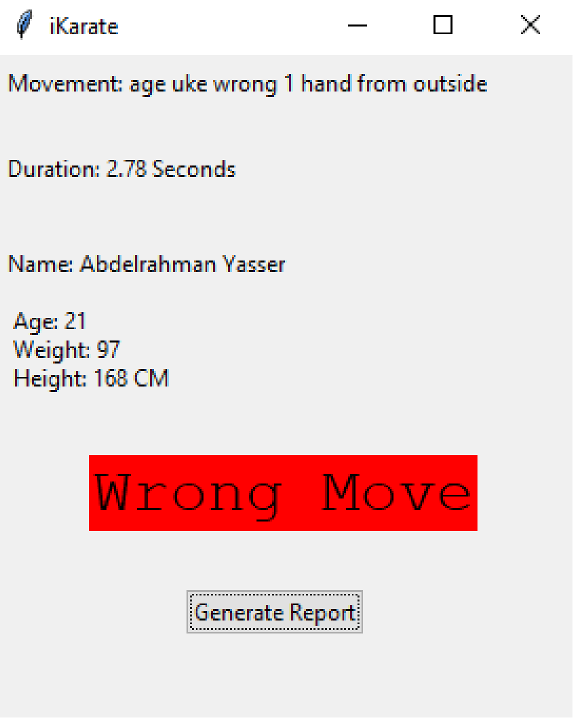


Figure 4.16: Movement Capture Interface



iKarate

Movement: age uke wrong 1 hand from outside

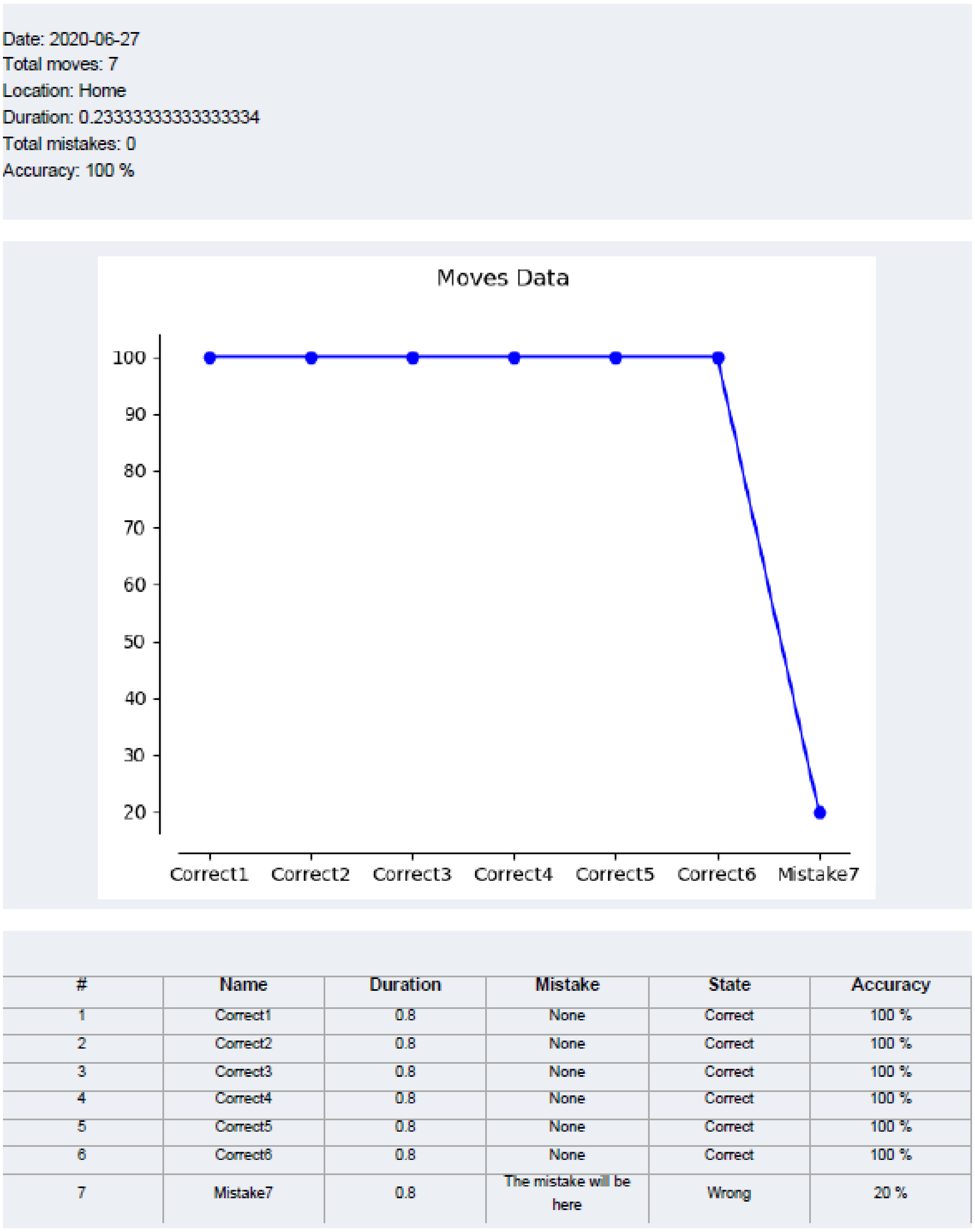
Duration: 2.78 Seconds

Name: Abdelrahman Yasser

Age: 21
Weight: 97
Height: 168 CM

Wrong Move

Figure 4.17: Results Interface



Date: 2020-06-27
Total moves: 7
Location: Home
Duration: 0.23333333333333334
Total mistakes: 0
Accuracy: 100 %

#	Name	Duration	Mistake	State	Accuracy
1	Correct1	0.8	None	Correct	100 %
2	Correct2	0.8	None	Correct	100 %
3	Correct3	0.8	None	Correct	100 %
4	Correct4	0.8	None	Correct	100 %
5	Correct5	0.8	None	Correct	100 %
6	Correct6	0.8	None	Correct	100 %
7	Mistake7	0.8	The mistake will be here	Wrong	20 %

Figure 4.18: Report

4.5.3 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 4.16.
- 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 4.17.
- 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.

4.6 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.	Completed
F2	Interpolation	Required	Interpolates the movement data before being sent for classification.	Checks that the data is smaller than the previous file.	Completed
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.	Completed
F5	Segmentation	Required	Used to segment the data read from the Kinect.	Validate that all the new arrays are originated from the same file.	Completed
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.	Completed
F11	Calculate Overall Score	Required	Calculates the overall score of the player.	Returns an accuracy or score for all the movement in the session.	Completed
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

Chapter 5

Evaluation

TABLE: 5.1 demonstrates the accuracy of each movement. The average accuracy calculated from the F-DTW as it was the best classifier is 91.07%. This accuracy has been discussed with a Karate coach to determine whether is it good enough or not. The coach confirms that this accuracy is very good implying the Karate Kata system.

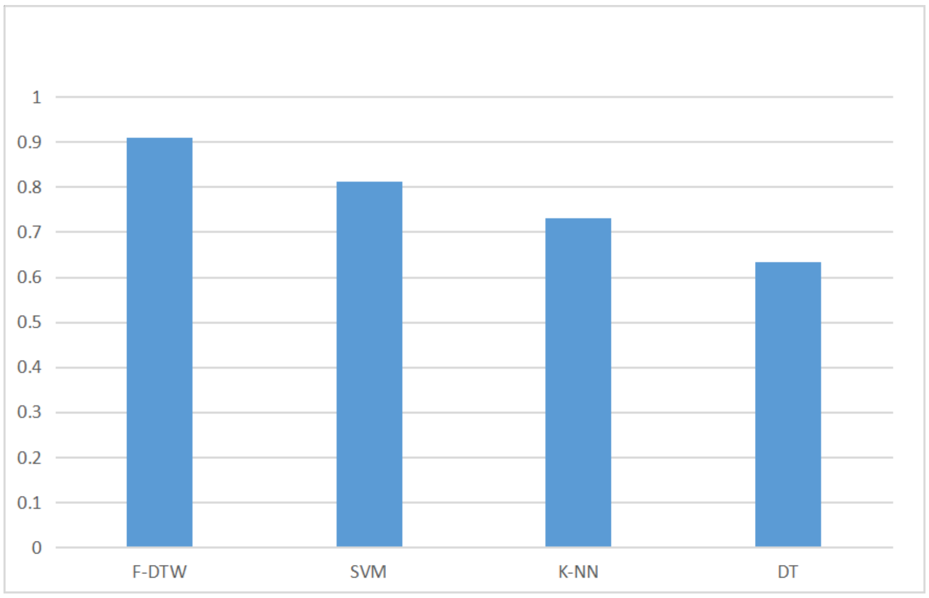


Figure 5.1: Overall Accuracies

Table 5.1: F-DTW Movements Results

Move	Trials	Correct	Wrong	Percentage
Hidari gedan-barai	16	15	1	93.75 %
Migi chudan oi-zuki	16	14	2	87.5 %
Migi gedan-barai	16	15	1	93.75 %
Migi tetsui-uchi	16	15	1	93.75 %
Hidari chudan oi-zuki	16	15	1	93.75 %
Hidari gedan-barai	16	15	1	93.75 %
Migi jodan age-uke	16	13	3	81.25 %

5.1 Experiments and results

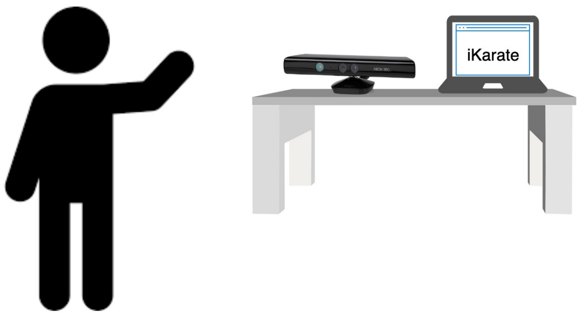


Figure 5.2: Setup

As shown in Fig: 5.2, the performers need to calibrate their bodies in-front of the Kinect in order to make sure that their whole skeleton is detected. Afterwards, they perform the moves and when they finish, the system will presents the results of the whole session.

Three players made 30 trials overall, each trial contains the 21 moves to test the system. Each trial was a combination of the correct moves and the incorrect moves, to see whether the system will classify them correctly.

We evaluated our system using dependant and in-dependant data, where we conducted an experiment by using professional and non-professional players to test the system, some players on each side tried using the system have some experience with using the system, while the other players were testing the system for the first time.

The tests indicate that among all classifiers F-DTW had the best average of accuracy followed by KNN, SVM, Decision Tree, $P, C-NN, and Multilayer Perceptron respectively. F-DTW had the highest accuracy due to its ability to analyze various time series to measure the association between two temporal sequences with similar behavior where time and speed vary. The $P had the lowest accuracy since it takes only 2D Data (X, Y), so it neglects an enormous part of the data. F-DTW bested all the other classifiers and was able to reach an accuracy of 91.07% as shown in Table: 5.2. It also needed the least pre-processing as it accepts data with different sizes.

Table 5.2: Different classification algorithm comparison.

#	F-DTW	K-NN	SVM	Multilayer Perceptron	C-NN	Decision Tree	$P
Performer 1	95.23 %	80.95 %	66.67 %	66.667 %	66.667 %	61.90 %	52.38 %
Performer 2	95.23 %	71.42 %	80.95 %	76.19 %	71.42 %	52.38 %	66.667 %
Performer 3	90.47 %	85.71 %	57.14 %	57.14 %	57.14 %	71.42 %	61.90 %
AVG	**93.65 %**	79.36 %	68.25 %	66.67 %	65.07 %	61.90 %	60.31 %

5.1.1 Usability Study

An experiment was conducted to evaluate how easy, accurate and satisfactory the system is. Six different users were introduced to the system and given the steps for each task, each user was observed while interacting with the system and they did not encounter any problems or experience confusion. They recorded their moves and the system generated a detailed report for every session. Two players used to play Karate in the past. At the end of the experiment, each user was given a questionnaire to obtain the most accurate results from the experiment. The results are shown in Table: 5.3.

5.2 Motion Capture Hardware

For capturing the movements' signals, an IR camera sensor was used to detect the human body. The Kinect camera was used for capturing the data, since it contains an IR sensor and is capable of detecting the skeleton of the human body. The Kinect is able to detect the 3D coordinates (X, Y, Z) of 25 joints of the performer and extract them using the Infrared Emitter. The software used as mentioned in [14] and [15] to utilize the Kinect is the Kinect SDK (Software Development Kit).

Furthermore, the Kinect is able to provide 30 FPS (Frames Per Second) with a resolution of 640 x 480-pixels as mentioned in Microsoft's book [15] with each frame containing the skeleton of the performer. The Kinect is also able to detect up to six performers.

5.3 Data Pre-processing

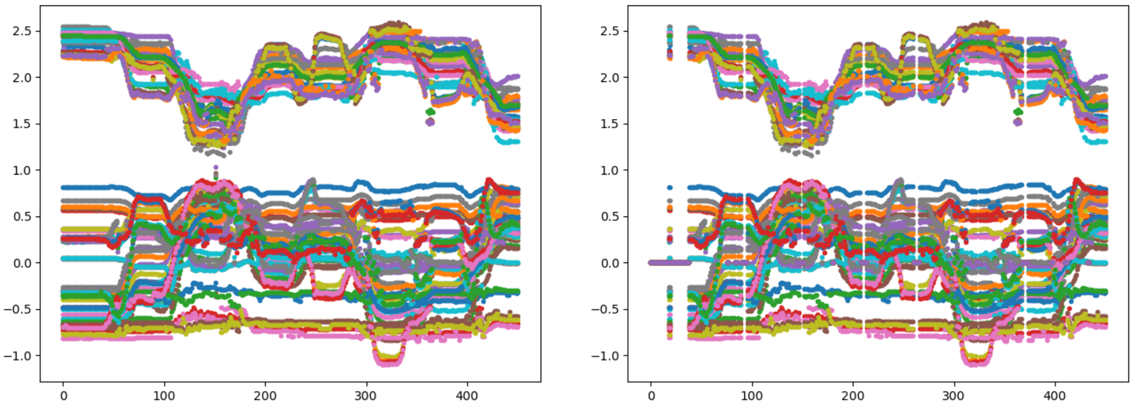
Before using the data in any of the classification algorithms, pre-processing was needed. The pre-processing contains two phases. The first phase is segmenting the sequence of movements and separating them, the second phase is interpolating the signal to be suitable for creating classification models. Further, to achieve better accuracy and results, the training data were clustered (A cluster for each move) since every Kata in karate must be performed in the same sequence.

5.3.1 Segmentation

Each move should be independently analyzed from the other movements to be processed and classified without interfering with the others. Therefore, a segmentation method was needed to split each move from the other moves and then send each move to the next phase.

$$Difference = |Mean(Window_1) - Mean(Window_2)| \tag{5.1}$$

In Karate Kata, there is a slight pause before each move, which was decided upon after performing several tests. Therefore, a method of segmentation was developed according to the collected information. The technique used to segment the movement stream was to construct a queue that will serve as a window filled with joint coordinates. A feature selection was used to detect the dominant joint. After that, the mean of each two consecutive windows is calculated and subtracted to use their absolute value as shown in Eq: 5.1. A threshold is calculated automatically based on the difference using an unsupervised clustering method (ISODATA algorithm). If the difference is below the threshold, then this indicates that there was no motion and the move will be segmented at this point. As shown in Fig: 5.3, the left SUB-Fig: 5.3a represent the data before segmentation, while SUB-Fig: 5.3b represents the data after segmentation and that the sequence was separated.



(a) before (b) after

Figure 5.3: before and after pre-processing

5.3.2 Linear Interpolation

The movement signal data needed to be interpolated for classification purposes. Multiple classifiers required the data to be of the exact size to be able to generate a model of those signals. Those classifiers took the whole signal data as features for better and accurate classification. So linear interpolation in Eq: 5.2 was used to simply unify the signals size, without losing the signal shape or properties.

$$B_2 = \frac{(A_2 - A_1)(B_3 - B_1)}{(A_3 - A_1)} + B_1 \tag{5.2}$$

A is the time space that represents the points location and B represent the signal data. (A_1, B_1), (A_3, B_3) are two known points that will be used to interpolate and create a new point (B_2) between them using (A_2).

5.4 Processing methods

This section will be representing the processing and classification phase with all the different algorithms used. This approach implemented different algorithms to deduce the suitable one for classifying and analyzing human motion data that was captured from IR sensors. Each algorithm will be explained along with the type of data entered.

5.4.1 Fast-Dynamic Time Warping

F-DTW is an algorithm for comparing differences between two waves [16]. Each movement is likely to be performed at different speeds which make every signal different from the other in size, so an algorithm to handle this problem was necessary. F-DTW is used to handle the player's various performing speeds captured by the Kinect and to present the player with an optimal real-time evaluation. It is also used in [5], [6], [7], [17], [10], [18], [19] and [20]. F-DTW can be extended to various data forms, such as images, audio and graphics. Therefore, any signal that can be transformed to a linear sequence could be evaluated using it.

After extracting the data from the Kinect and pre-processing it, as mentioned in SUBSECTION 5.2,

$$D(P) = \sum_{K=1}^{\kappa=K} D(P_{ki} - P_{kj}) \tag{5.3}$$

F-DTW computes the distance and path as shown in EQUATION: 5.3. Where **D(P)** is Euclidean Distance of the warp path **P**. Where the wrap path length is **K**, and **D(pki, pkj)** represents the distance of two data points (one from Training data and the other is from the Testing data) in the **kth** element of the warp path as mentioned by Stan Salvador et al. in [16]. F-DTW can compute the distance with multiple methods, but the **Euclidean Distance** is the one this approach is using.

5.4.2 Other Algorithms

In addition, other algorithms have been tested on the system to establish the highest accuracy. Unlike the F-DTW, these algorithms need more pre-processing. They need to be interpolated first, so that all the signals have the same size, because these algorithms don't accept data with different sizes.

- SVM: Is a machine learning algorithm used to find a hyperplane that distinguishes classes in a pattern, and it is widely used in multiple fields. The created hyperplane illustrates the maximum margin distance between data points [21], [22].

- DT: Decision tree is a visual representation of alternative options based on particular criteria for a decision. It is known as a decision tree since it starts with a single box (or root), which branches off into different solutions, similar to a tree [23].

- K-NN: K-nearest neighbors is a non-parametric pattern recognition algorithm used to classify objects by setting a weight to each object, and a set of neighbors vote to decide which class the object belongs to [24].

- $P: Point-cloud recognizer recognizes unordered 2-D strokes as point-clouds. Which allows it to recognize multi-strokes (Multiple joints in our case) with equality. Its predecessor is $1 algorithm, which is built upon Dynamic Time Warping. Our experiment included using multiple combination of two dimensions (Ex: X and Z) to determine the best outcome the recognizer could use to detect the performers' moves [25].

- C-NN: Convolutional neural networks represent a group of neural networks that have been surprisingly efficient in areas such as image recognition and classification. In addition to controlling vision in robots and self-driving cars, ConvNets has been effective in recognizing faces, objects and road signs [26].

- Multilayer Perceptron: Represents a class of neural networks composed of at least three nodes. In addition, each of the node of the multilayer perceptron, aside from the input node is a neuron that utilizes a non-linear activation function. The nodes of the multilayer perceptron are organized in layers [27].

Table 5.3: Usability Study

	Very Weak	Weak	Neutral	Agree	Strongly Agree
Was the interface pleasant ?	0%	17%	0%	66%	17%
Did iKarate analyse all your moves ?	0%	0%	17%	83%	0%
Did your style improve at the end of the session ?	0%	0%	17%	33%	50%
Did you find the brief message useful after playing ?	0%	17%	17%	17%	49%
Was the detailed report appealing ?	0%	0%	17%	33%	50%
How accurate did you find the scores acquired in the detailed report ?	0%	0%	17%	83%	0%
Would iKarate improve your Karate performance ?	0%	0%	17%	33%	50%
Would iKarate help you practice Karate at home ?	0%	0%	17%	33%	50%
Would you tell a friend about iKarate ?	0%	0%	17%	33%	50%
Would you use iKarate frequently ?	0%	0%	17%	83%	0%
Has your style improved after using iKarate ?	0%	0%	17%	33%	50%

Chapter 6

Conclusion

We present an approach that utilizes the IR camera sensor on karate performers to detect and classify the movements of the performers. After implementing multiple algorithms and tested different methods to detect movements and segment the data correctly, we reached acceptable results using the F-DTW algorithm. Our biggest challenge was to correctly segment each Karate move from the stream of data and classify them, and our best approach to segment the moves was to use windowing method and use ISO data algorithm to calculate the threshold. combining the F-DTW algorithm and the segmentation method, we achieved a total accuracy of 91.07%. With these results, we evaluated our system using dependant and in-dependant data, where we tested the system with professional and non-professional players, some players on each side tried using the system in a previous session, while the other players were testing the system for the first time. The result of this experiment satisfied the system's expectations and reached a great milestone. The 21 moves of kata 1 (Heain Shodan) were the primary movements that have been used in the system. The system results satisfy the field of guiding Karate trainees and assisting coaches, as it involves no risk for the users. In addition, the system generates a comprehensive report, so it can be reviewed by a human to determine whether the system failed to correctly classify movements done by the performer. This approach assists the performers in practicing and improving their performance by presenting them with the mistakes in their style of performing. The report system gives a score to the player determined by the mistakes performed and stores all the session's data to be reviewed later by the coach or the player, making the system very applicable for home or remote Karate practising.

6.1 Future directions

For future, we aim to present the result to the performer in real-time, and enhance the accuracy. More features could be implemented such as handling multiple players and classify each player's movement. We would like to consider more

Bibliography

[1] 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.

[2] 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.

[3] K. Wennrich, B. Tag, and K. Kunze, "Vrte do - theway of the virtual hand," *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology - VRST 18*, 2018.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] P. Alborno, 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] ——, “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.

[14] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE Multimedia*, vol. 19, no. 2, p. 4–10, 2012.

[15] R. S. Miles, *Start here!: learn the Kinect API.* Microsoft, 2012.

[16] S. Salvador and P. Chan, “Fastdtw: Toward accurate dynamic time warping in linear time and space,” 2004.

[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 - ICCAE 2018*, 2018.

[18] 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.

[19] 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.

[20] B. Emad, O. Atef, Y. Shams, A. El-Kerdany, N. Shorim, A. Nabil, and A. Atia, “ikarate: Improving karate kata,” *Procedia Computer Science*, vol. 170, p. 466–473, 2020.

[21] H. P. Yue Shihong, Li Ping, “Svm classification its contents and challenges,” 2003.

[22] P. P. P. S. N. Y. W. W. X. SHUJUN HUANG, NIANGUANG CAI, “Applications of support vector machine (svm) learning in cancer genomics,” *Cancer Genomics Proteomics*, 2017.

[23] H. Sharma and S. Kumar, “A survey on decision tree algorithms of classification in data mining,” *International Journal of Science and Research (IJSR)*, vol. 5, 04 2016.

[24] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, “knn algorithm with data-driven k value,” *Advanced Data Mining and Applications Lecture Notes in Computer Science*, p. 499–512, 2014.

[25] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock, “Gestures as point clouds,” *Proceedings of the 14th ACM international conference on Multimodal interaction - ICMI ’12*, 2012.

[26] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *ArXiv e-prints*, 11 2015.

[27] A. Rana, A. S. Rawat, A. Bijalwan, and H. Bahuguna, “Application of multi layer (perceptron) artificial neural network in the diagnosis system: A systematic review,” *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, 2018.

[28] “Coronavirus,” https://www.who.int/emergencies/diseases/novel-coronavirus-2019, last accessed on 15/04/20.

[29] “Shotokan karaté do,” https://shotokancrsa.com/heianshodan-eng.htm, last accessed on 16/04/20.

[30] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” *2008 Eighth IEEE International Conference on Data Mining*, 2008.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] R. Niewiadomski, K. Kolykhalova, S. Piana, P. Alborno, G. Volpe, and A. Camurri, "Analysis of movement quality in full-body physical activities," *ACM Transactions on Interactive Intelligent Systems*, vol. 9, no. 1, p. 1–20, Nov 2019.

[38] V. A. Prisacariu and I. Reid, "Robust 3d hand tracking for human computer interaction," *Face and Gesture 2011*, 2011.

[39] K. Tanaka, "3d action reconstruction using virtual player to assist karate training," *2017 IEEE Virtual Reality (VR)*, 2017.

[40] 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.

[41] 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.

[42] T. Hachaj and M. R. Ogiela, "Rule-based approach to recognizing human body poses and gestures in real time," *Multimedia Systems*, vol. 20, no. 1, p. 81–99, Mar 2013.

[43] 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, no. 11, p. 2590, Oct 2017.

[44] 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.

[45] 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.

[46] 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.

[47] S. Wada, M. Fukase, Y. Nakanishi, and L. Tatsuta, "In search of a usability of kinect in the training of traditional japanese "kata" — stylized gestures and movements," *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)*, 2013.

[48] K. Mitsuhashi, S. Yokota, H. Hashimoto, S.-G. Shin, and D. Chugo, "Education system of skill succession based on 3d evaluation and improvement in time series," *2015 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2015.

[49] ——, "Educational system of physical motion based on 3d biomechanism evaluation," *2016 IEEE International Conference on Industrial Technology (ICIT)*, 2016.

[50] Y.-M. Chou, H.-R. Chen, and Y.-T. Shih, "Design of motion sensing martial art learning system," *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, 2019.