iKarate: Improving Karate Kata

1 Refined Project Description

1.1 The problem addressed specifically in this project and its importance

Karate moves are combination of physical moves. Its methods are qualitative not quantitative which makes it hard for students to perform the perfect version of the motion and harder to judge it. Kids nowadays may find difficulties learning those moves at a young age and since the training may consist of a large number of students so the trainer himself may not be able to focus on every detail of every student's move, which could result in taking more time to learn and master the move or it may lead to learning incorrectly from the beginning.

1.2 Project scope and expected outcome

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. Taken into consideration that the activities might be performed with different speed, body proportions such as (Limbs length) and initial position of the students. Also, 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.

2 Refined Project Plan

2.1 Detailed schedule and milestones

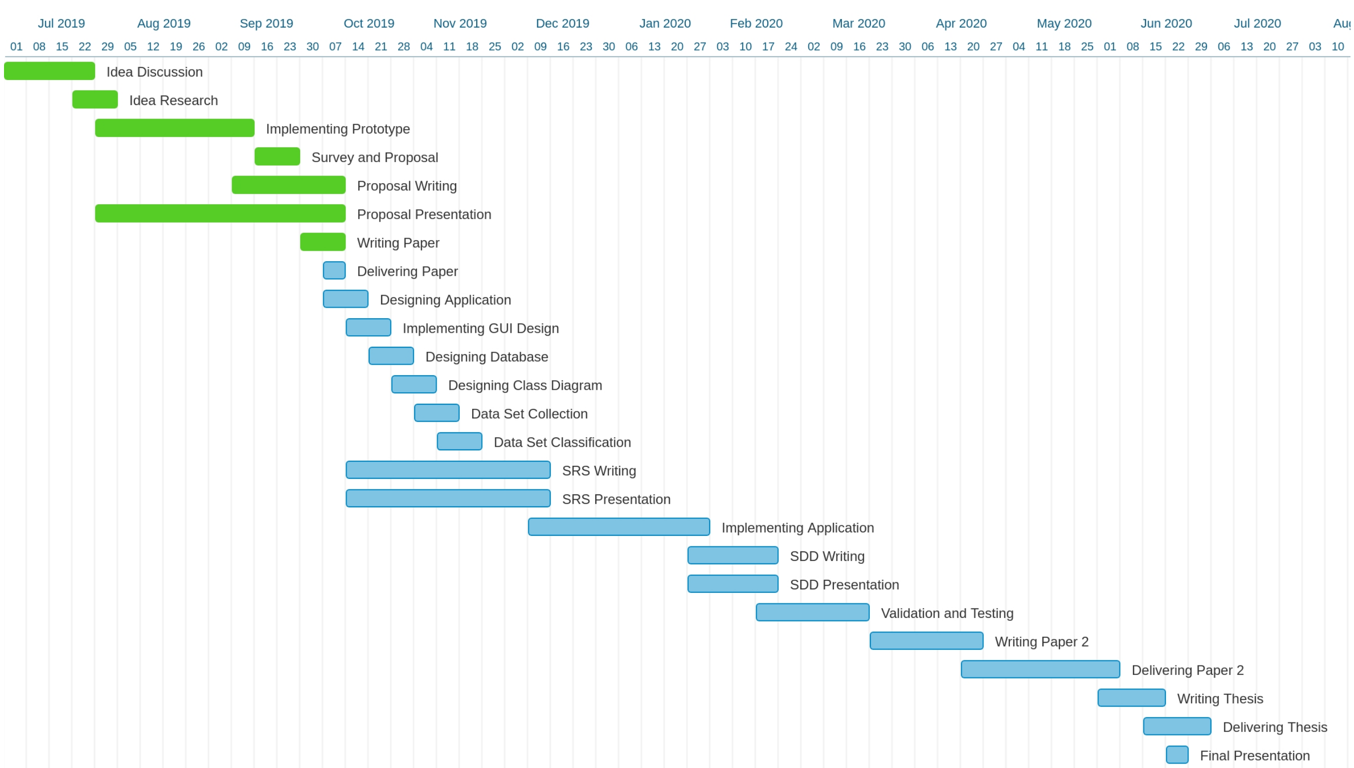

Figure 1: Tasks And Time Plan

2.2 Team structure and detailed roles/responsibilities of each member

Team Member 1 (Full-Stack Developer): Is the team leader. His role is to manage the project tasks and deliveries. Also, he is responsible for extracting the data from the Kinect then send the data to be enhanced.

Team Member 2 (Back-End Developer): Is the team Co-Leader. His role is to receive the data that has been extracted from the Kinect and make some enhancements on the data then classify the data and send it to be stored.

Team Member 3 (Front-End Developer): His role is to make a simple and friendly application interface that uses Nilson Heuristics for User Interface Design, so that any user can use the application without having any troubles.

Team Member 4 (Database Developer): His role is to organize the data that has been received from the classifier and store the data in the database for later use. He is also responsible for moving the old data to a new database so that the performance of the database doesn't decrease.

2.3 Contingency and risk mitigation plan

2.3.1 Risk

1. Kinect may not be suitable in some countries due to its price and availability.
2. Internet connection may not be good enough for the data to be transferred to the server.

2.3.2 Contingency

1. Kinect can be replaced with Accelerometers due to its cheap price.
2. Data can be classified locally but it will take more time.

3 System Requirements

3.1 Requirements Elicitation Process

3.1.1 Description of the processes that were actually used for: requirements elicitation, analysis, prioritization, and change management

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. The performers 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.

3.1.2 List and categorization of system stakeholders, users, and clients

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.

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.

3.1.3 Challenges encountered [and lessons learned] during the requirements gathering, analysis and prioritization phases

The main challenge was in the data gathering. In order to collect the data-set we needed multiple karate player to perform the movements while we record them with the Kinect, we also need karate coached so that they can tell us how can the move be performed correctly and what are the common mistakes of each move. Moreover, we needed the coaches to tell us whether our system is classifying the move correctly or not. We also needed some users (Specially non-karate and amateur users) to test the system.

3.2 System Requirements List

3.2.1 Functional requirements

Name	Interpolation
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 and the data files are ready to be read.

Name	Segmentation
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 and the data files are ready to be read.

Name	Classify Movement
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.

name	Get Joints Data
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.

Name	Calculate Overall Score
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.

Name	View Progress
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

Name	View Report
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

3.2.2 Non-functional requirements

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.

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.

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.

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.

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

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.

4 System Design

4.1 High level system architecture, data flows, etc.

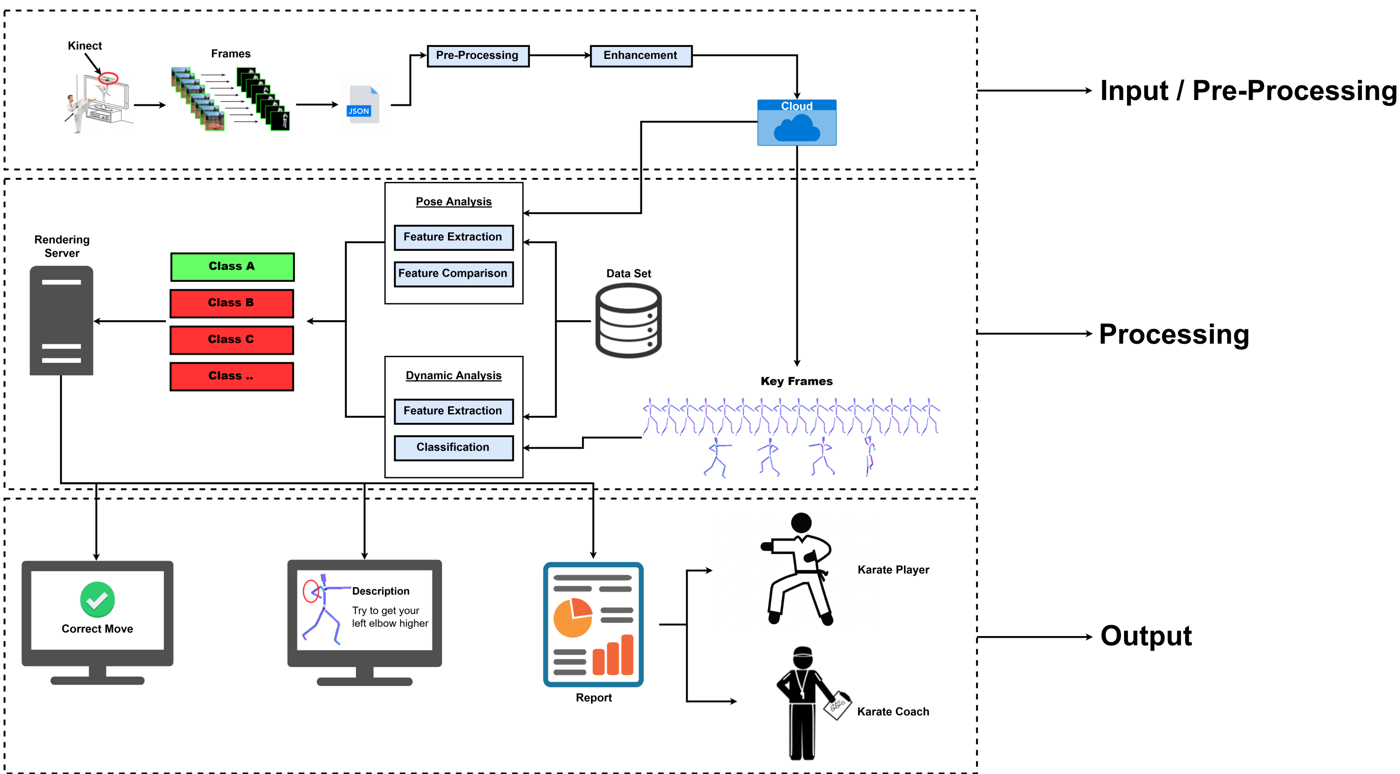

Figure 2: System Overview

4.2 User interfaces, if any

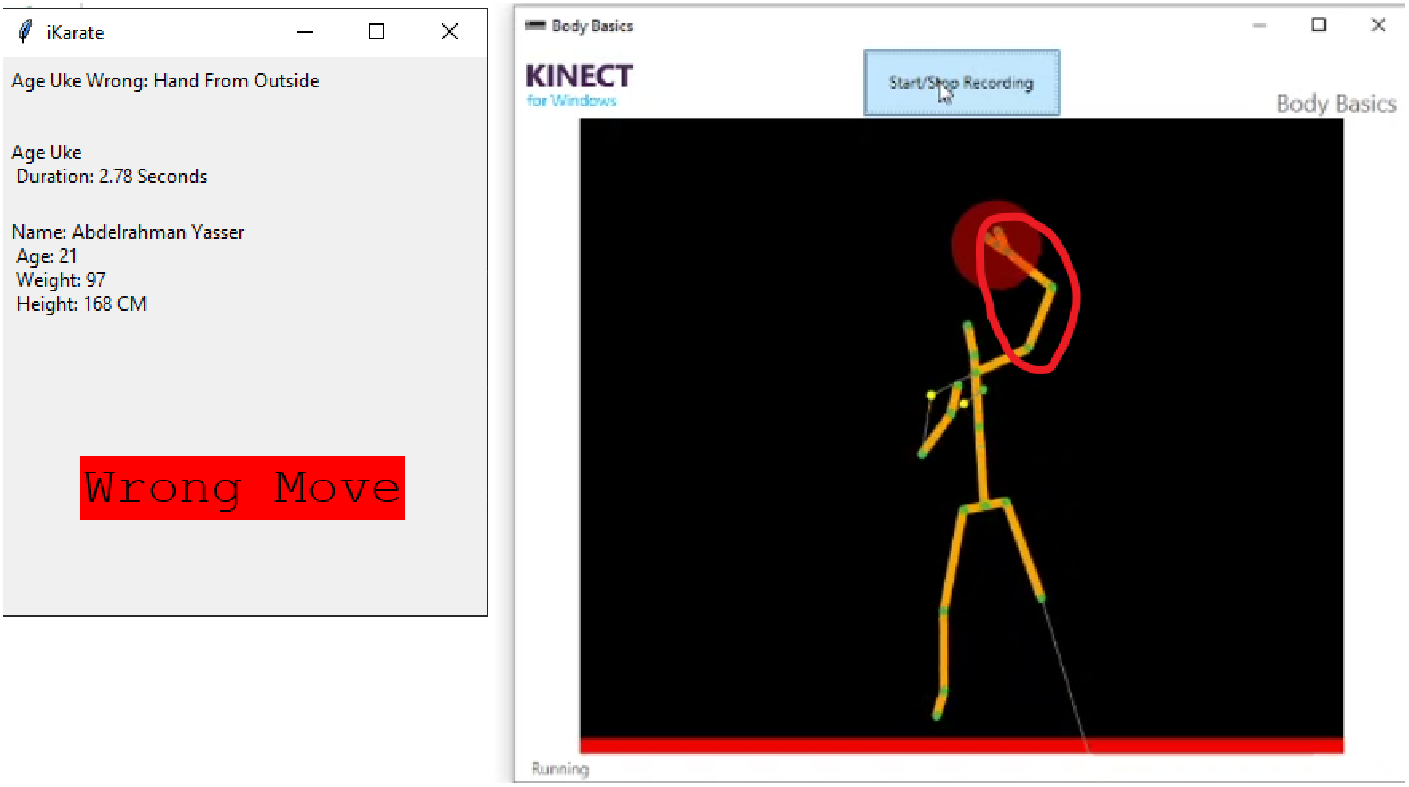

Figure 3: Real-Life System

4.3 Algorithmic components, if any

Fast-DTW is an algorithm for measuring similarities between two signals, each signal may have a different speed from the other signals. Fast-DTW is an alignment algorithm which is capable of classifying two different time signals. Fast-DTW could be applied to different types of data, like videos, audio and graphics data. Thus, any data that can be converted to a linear sequence could be analyzed using it. The algorithm was able to recognize the classes of the correct and wrong moves. Fast-DTW could be used with many different distance equations but the "Euclidean Distance" is the one used in this approach as shown in EQ. 1 to compute the distance between the classes.

$$D = \sum_{x_i, y_i}^{n} \sqrt{(x_i - y_i)^2} \tag{1}$$

Where "D" is the distance value, "X" represents the data-set joint position and "Y" represents the performer's joint position. Between each two moves, there is a small gap, this gap would be used to segment each movement and the movement data would be sent to F-DTW algorithm to be classified.

4.4 Innovative aspects of the design

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.

5 System Implementation

5.1 Hardware and software platforms

5.1.1 Kinect

The main hardware used in this paper is 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 Hardware and software development tools, languages, etc.

Kinect SDK 2.0: The Kinect for Windows Software Development Kit (SDK) 2.0 enables developers to create applications that support gesture and voice recognition, using Kinect sensor technology on computers running Windows 8, Windows 8.1, and Windows Embedded Standard 8.

Windows Development tools: C#, WPF, Python

5.3 Modules/components acquired from external sources

Microsoft skeleton tracking module

5.4 Innovative aspects of the implementation

Segmentation and analysis pose and dynamic

6 Other Relevant Issues and Challenges

6.1 Technical

Sending and receiving data between C# (Kinect SDK) and python (Pre-processing & Classification) was one of the challenges and we were able to solve this problem using the database. Another challenge was the segmentation of sequence of movements and we were able to solve this problem by visualizing the data on a 3D space so we can know where exactly to segment the move. Moreover, Live report analysis and real-time feedback is one of the major challenges and we were able to solve this problem by using threads and real-time database.