BADMINTON GAME ANALYSIS

BY

TEH KIAN CHONG

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF (HONS) COMPUTER SCIENCE

Faculty of Information and Communication Technology

(Perak Campus)

OCTOBER 2015

BADMINTON GAME ANALYSIS

By

Teh Kian Chong

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF (HONS) COMPUTER SCIENCE

Faculty of Information and Communication Technology

(Perak Campus)

OCTOBER 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled "**BADMINTON GAME ANALYSIS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________

Name : TEH KIAN CHONG

Date : 20TH APRIL 2015

ACKNOWLEDGEMENTS

I would like to take this opportunity to show my gratitude to those who gave their helping hands and make this project completed with success. First of all, I would like to express my sincere appreciation to my supervisor, Dr. Ooi Boon Yaik who gave me proper guidance, support and encouragement throughout the whole system development process. Besides, I would also like to thank my moderator, Dr. Ng Hui Fuang who has given me advices and opinions on the project.

Lastly, I am very thankful to Universiti Tunku Abdul Rahman for providing me learning opportunities to conduct the study. Also, an utmost appreciation is given to my families and friends who supported me throughout the project development.

ABSTRACTS

This project is a software development project designed for the badminton industry. Its main focus is on the data collection of important key points from a badminton match by video. The common method used in analysis for badminton is basically observation either by video or live observation itself. The current practice limits the ability to analyze large amount of data as ample time would be required. Therefore, this project is targeted to develop software where it would be able to analyze these videos of badminton matches allowing users to collect data of key points as fast as possible. This project utilizes image processing tools, OpenCv along with a JavaEE framework that allows the system to be rendered in a website. The project is divided into 2 subsystems where the first system functions to segment videos based on the indicator for each change in rally which is the change in angle of camera. The second system functions to render the videos segmented previously, allowing the users to validate each rally while allowing them to input important key notes for that particular segment of video.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Program Interface
BGA	Badminton Game Analysis
JPA	Java Persistence API
JPQL	Java Persistence Query Language
EE	Enterprise Edition
SQL	Structured Query Language
HTML	Hyper Text Markup Language
JSP	Java Server Pages
EJB	Enterprise Java Bean

Chapter 1: Introduction

1.1 Background and Motivation

Analysis of player's performance during match in sports is a common procedure conducted by coaches in order to further improve the player's performance. The specifics in the analysis would allow the coaches to further explore the weakness of the players and to work on that particular weakness (Performance Analysis, n.d.). The analysis procedure is not only limited to the player's performance but can also be used to analyze the opponents performance. This allows the coaches to strategize on how to counter the weakness of the opponent. Therefore, this clearly illustrates the importance of analysis in the sports industry (Emptage, 2014).

There are several methods that help coaches keep track of the player's performance, and at the same time analyze them. The most common method of analysis is by using the observation technique (Bishop, n.d.). This technique is applied where the coach observe the players live in action during the match itself. This allows the coach to take notes on the key points or weaknesses of the player. Besides that, another method used to aid in analysis, which is the highlight method of this project, is the use of video for analysis. The coaches use videos to compute the data of key points for the player's performance and then conduct the analysis. By using the videos, the coaches would have possibly 2 alternatives where one would be to analyze the video manually by taking notes while watching the video and the other would be to use software to generate key points automatically.

This concept is not excluded from badminton. Badminton is generally a fast paced sport where analysis and data collection has been a tedious task to do. The current general practice for analysis in the badminton industry would be to manually record down the key points in a match either by using video or by observation itself (Methods of Data Collection, n.d.). This, in general, is a very tedious and inefficient way of analyzing a badminton match, and hence would bring a need for a system where the ability to generate data of key points for a badminton match could be done as automated as possible.

The key points mentioned above, in the context of badminton would refer to important points on a particular game noting how players actually win or lose a rally. For example, it would be important to note on key points such as smashes, drop or even unforced errors on a particular game.

These key points when collected in a whole could show a certain pattern of play in a badminton game whether it’s in the context of the player or the opponent. For example, a group of smashes in particular area could note that this player has a pattern of smashing that area frequently. Therefore, this information could be used by opponents to counter-attack on the player.

This project aims to ease the collection of the key points in a badminton match as mentioned above. There are many currently development of systems that provide coaches method of recording key points in a particular match, but still it takes time to watch a particular play and record each of the key points. Therefore, there is a need for a system capable of easing the user in recording the key points of a badminton match. A comparison of the existing systems will be discussed in Chapter 2.

1.2 Problem Statement

The development of this system brings interest as international badminton matches require thorough analysis of key points. The development of a more efficient way to collect badminton game key points such as smashes and unenforced errors will ease the analysis process. An example of this case would be if a coach were to make use of an efficient system of collecting data, they could save more time in the data collecting which consumes time, and could spend more time on their analysis. With more time on analysis, a better plan to counter-attack the opponent could be devised.

Current general method of collecting data for analytical purposes in badminton is usually by observation. This method is flawed as observation by itself might produce errors in the data collection. For example, if a coach keep track of a player’s key points in match by observation, it is mostly likely that there might be an error in the data collection, maybe due to tiredness. However, if combined with a system capable of aiding the user in the data collection process, then the accuracy of the data collection could be increased as the coach would not have to look and search for the key points in a video manually, but have the system to generate the key points manually.

Not only that, data collected by observation is usually stored in hardcopy. Therefore, the data stored would be difficult to search for and retrieved. For example, if a coach were to keep track of several players, then he might have several copies of data of key points that refers to different players. There is a possibility he might get lost on retrieving data few years back. Hence, if combined with a system capable of storing data in a database, there is no problem for the coach to retrieve the data of that particular player, even, if it is from a few years back. The organization of the key points would be handled by the database management system.

Hence, in order to develop this system, a subsystem functioning to control the segmentation of the videos into key points are required, along with another subsystem that renders the data collection based on the videos segmented along with the users. The second subsystem would also have to store the data collected into the database. The implementation of this project will be further elaborated in Chapter 4.

1.3 Objectives

The objective of this project is to develop a system that can aid users to collect data of key points from a badminton match by video. The proposed solution is to develop two subsystems that function to segment videos and render videos separately.

The second objective is to develop a database that can help store and manage the data of key points of badminton matches. Instead of using hardcopies of the data stored physically, the use of database will help in reducing space consumption as well as improve the retrieving of the data of key points.

After the implementation of this system, an experimental phase will be conducted in order to calculate the accuracy and the speed of data collection. These tests will show the limitation of the system along with its weaknesses.

1.4 Project Scope

This project involves the development of a key points collection system by video for badminton. As such, the videos that are covered are only videos that represent a badminton tournament, and each video represents a game in a tournament. The system will only cover matches that are recorded in specific angles as shown below:

Figure 1-4-F1 Specific Angle

This is because, each of these matches will be processed separately based on each game, so that for every segmented video based on the game, the recorded key points from user can be stored in relation to that particular video segment. Apart from that, the system will only cover matches for men’s singles as the key points covered will be easier to implement. However, in future matches, it could be suggested that the match should also include other categories of badminton as well.

In addition, the system will only cover the data collection of key points for the player nearer to the camera as the key point detection further away from the camera might cause ambiguities and errors in data since it is not clear.

Besides that, the system implement shall have 2 subsystems that will run sequentially one after another where the storage of data will be at the host computer itself. This will allow the host system to access the data for retrieving and writing purposes easily.

1.5 Contribution

Currently in the context of badminton, there is a system that aids badminton aids in the key point data collection for badminton. It allows users to add points manually and then provide basic analytic information such as average drops, and graphs of winning points throughout the game. However, this method is still considered as a manual approach where the user has to visually observe matches whether relevant or irrelevant in order to record the key points of the match.

The development of this system managed to ease the data collection from a badminton match by video. The user would not have to replay each video to keep track of each point scored in a badminton match. The system will cater for all the key points of the badminton match, ready for the user to input data for that particular match.

Not only that, the users could use the system for analysis of previously recorded data. This system would show the user the basic analysis such as average number of drops per game as well as a more thorough analysis such as the technique and position where the player wins the most.

1.6 Summary

The badminton game analysis system is a method to record data from badminton video matches such as smashes and drops so that it could be analyzed, in order to form a strategy against opponents. For the implementation of the following system, 2 subsystems will be created where the first system will function to process the videos into segments containing key points in a match. The second system will render the videos segmented earlier, so that the user will be able to record the key point manually. In addition to that, the data collected will be stored in the database that will allow the user to retrieve and analyze the data easily the hassle of searching for the data. There is still much improvements to be made to cover the limitations that are omitted in the project scope, for example, the coverage for other categories of badminton such as men’s doubles.

Chapter 2: Literature Review

2.1 Overview

The current systems that are available in the market is no doubt proving that there is a major importance in computing statistics for a match. These statistics not only provide a detailed overview of the overall match, but also provides medium for players to improve their game play. There are a few number of software that computes statistics of a match by video analysis. Current practices are mainly focused on manual tagging where special key points are tagged in the video manually. The systems basically allow sophisticated analysis after the tagging process. However, there is software that practices automatic video analysis. In the context of badminton, there is currently one software the practices the use of manual tagging where the results will later be shown after the manual is done. The reviewed software of video analysis for different fields will be further elaborated in this chapter.

2.2 Solutions for other Sports

2.2.1 Soccer

SOCCERLAB VIDEO ANALYSIS PRO is software that allows the coaches of soccer teams to analyze a soccer match by video. Among the details features offered by the software is the flexibility to design the labels used for video tagging.

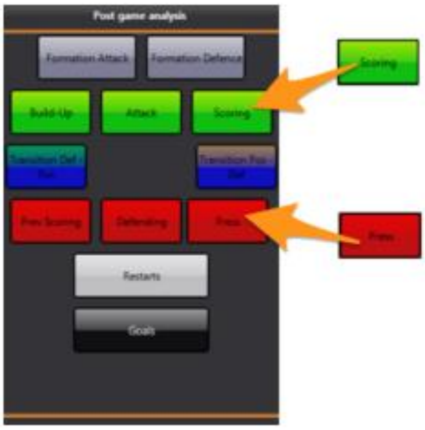

Figure 2-2-1-F1 SoccerLab customization interface

Besides that, another interesting feature provided by this software is the feature that allows the coaches to draw on top of the video that will be automatically mixed inside the resulting clip.

Figure 2-2-1-F2 Drawing feature

The manual tagging feature of this software is different as it allows multiple tagging by different people. The central server will then synchronize these tagging into the video. This manual tagging feature is also available on their mobile android version. The statistics computed would be able to be compiled into a video clip form, where each clip represents a certain key point of the match (soccerlab, n.d.).

VideoMatch is another match analysis software for soccer. This software is basically a software for manually tagging of the key points in the video of the match. It provides the most basic analysis where users can customize their own tags and tag them during the play of the video (videomatch, n.d.).

Figure 2-2-1-F3 Different kinds of tags

MatchAnalysis is an online software that provides analysis through access of their database. The statistics are basically computed manually by analysts on their side. Their basic flows of computing video statistics for their clients are as follows:

Figure 2-2-1-F4 Match Analysis logo

1. Video is uploaded to MatchAnalysis or received via satellite broadcast
2. Analysts record every touch, synchronizing data precisely with video
3. Data flows into their database
4. Coaches receive the statistics and analysis instantly.

(matchanalysis, n.d.)

2.2.2 Tennis

ITUSATENNIS is an online system where it analyses the video of the player's performance using state-of-the-art technology and results in a video compilation and breakdown of the player's strengths and weaknesses as well as effectiveness of the patterns of play during match. However, the method of computation of the statistics is not revealed to the user.

Figure 2-2-2-F1 ITUSA tennis logo

The system is basically divided into 3 levels of analysis, namely, winners and unforced errors, patterns of play and custom match analysis. An example of the results of the analysis is shown below (iTUSA, n.d.):

Figure 2-2-2-F2 Example of first level analysis

2.2.3 Multiple Sports

LONGOMATCH is a system that supports multiple video based sports analysis.

Figure 2-2-3-F1 LongoMatch logo

As mentioned above, one of the special features offered by this software is the ability to conduct analysis on different sports. Among the few already implemented areas of sports are soccer, rugby, volleyball, floorball, netball, basketball, field hockey, ice hockey, wrestling, and cricket.

The standard feature of customizable key points is also available throughout the software (logomatch, n.d.).

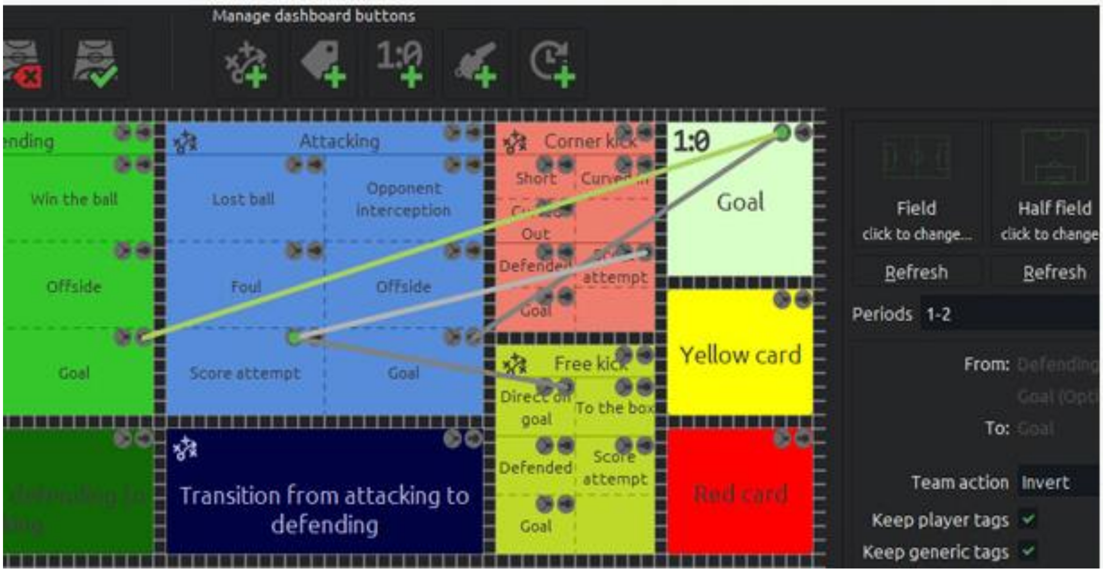

Figure 2-2-3-F2 Customization of own tagging panel

There is another alternative to video analysis where it requires extra hardware to capture more information live during the match. An example of such software is matchviewer by Prozone.

Figure 2-2-3-F3 Prozone logo

This software is unique as the analysis and the computation of the statistics for the match are done by a special team in the company. The team specializes in analyzing the video using different technologies which are not shared to the public. The computation of statistics will be ready made for the user and a wide range of analysis could be done on the statistics (prozone, n.d.).

2.2.4 Badminton

Badminton Game Analysis App is a system that allows users to key in points of a match. The key in process is as shown below:

Figure 2-2-4-F1 Badminton Game Analysis App

As shown in the image above, the statistics collection requires the user to manually input each point individually. The app can also provide basic analysis but are limited to only basic analysis such as:

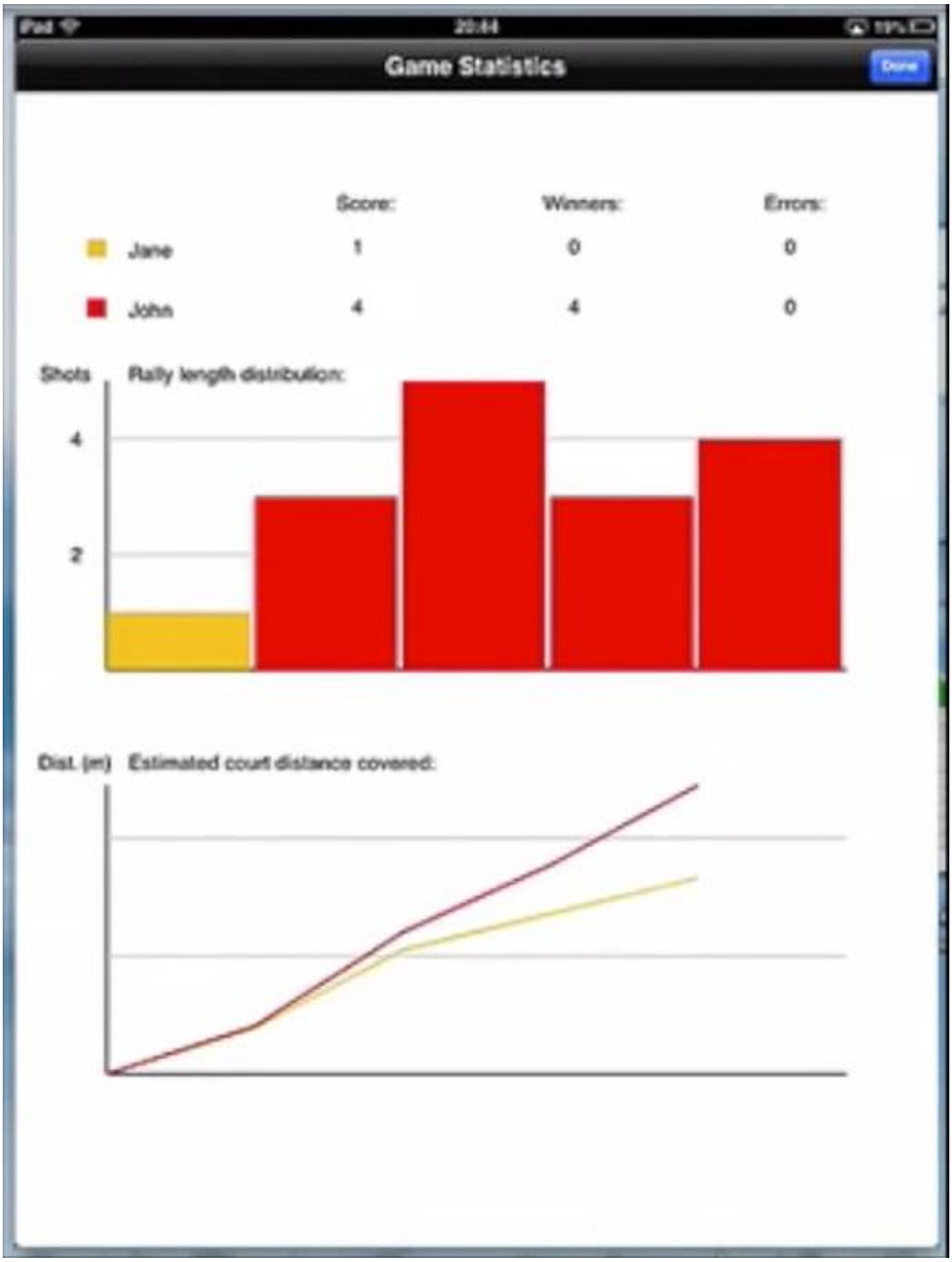

Figure 2-2-4-F2 Statistics for BGA

This system is flexible as it allows analysis of any players without the need of a video. It is also unique as it has a different kind of analysis where there is an estimation of the distance covered on the court.

2.3 White Space Analysis

System Features	Custom Tags	Manual Tagging	Automatic Tagging	Need of Special Device	Drawing on video	Different levels of Analysis
SoccerLab	√	√			√	
VideoMatch	√	√				
MatchAnalysis		√				√
LongoMatch	√	√				
ITUSA Tennis			√			√
Prozone			√	√		
Badminton Game Analysis App	√	√			√	√

Table 2-3-1 White Space Analysis

2.4 Conclusion

All of the above reviewed software were useful in their respective fields. However, the exploitation for video analysis for a badminton match is still not very widely used. Therefore, there will be a need for a system that is capable of aiding the users in recording key points for a match by video. The proposed solution is to use a system that could help identify the key points in a match by video in order to aid the user in recording the key points.

As a conclusion, the final proposed system shall consist of two subsystems where the first one will function to process the videos based on the key points of a match, and the second subsystem will render the segmented videos onto a website platform where the user can record down the important key points. From there, the key points recorded will be stored onto a database where post-analysis of a badminton match could be done. With such a system, the user would be able to speed up analysis time and store the records in a database, where future analysis could be done.

Chapter 3: System Design

3.1 Methodology and Technologies Used

3.1.1 Methodology

The methodology model that will be implemented for this software is the evolutionary prototyping model. This model was chosen because the software should be flexible to any extra requirements that are needed by the user. The most basic system should be developed first as prototype before the user can request for any further requirements for the software. Hence, the basic core subsystem that segments the videos into important key points will be developed first, followed on by the interface that will render the segmented videos.

Before the cycle of the further refinements start, a basic plan will be drawn for the software. This would include the use case diagram, the class diagram and ER diagram. This stage is estimated to take around 3 to 4 weeks.

Next, the system will be built according to the design drawn. The system logic will be built first and then followed on by the interface and the database for the system. The software interface and logic will be developed using OpenCv, C++ and JavaEE while the database used is the Oracle database. This part of the development will feature only the basic ability of the software where the videos are segmented and rendered to the website platform. This stage will roughly take around 4 to 5 weeks.

The following stage will be repeated until the user is satisfied with the software. This would include user feedback and further refinements added to the software. This cycle is estimated to repeat around 4 times to finally get the required results. Among the expected refinements are a more detailed statistics from the video match itself. This is estimated to take around 8 weeks.

The final product will then be delivered through by the 16^{th} week of the development period. The total estimated time would be around 4 months.

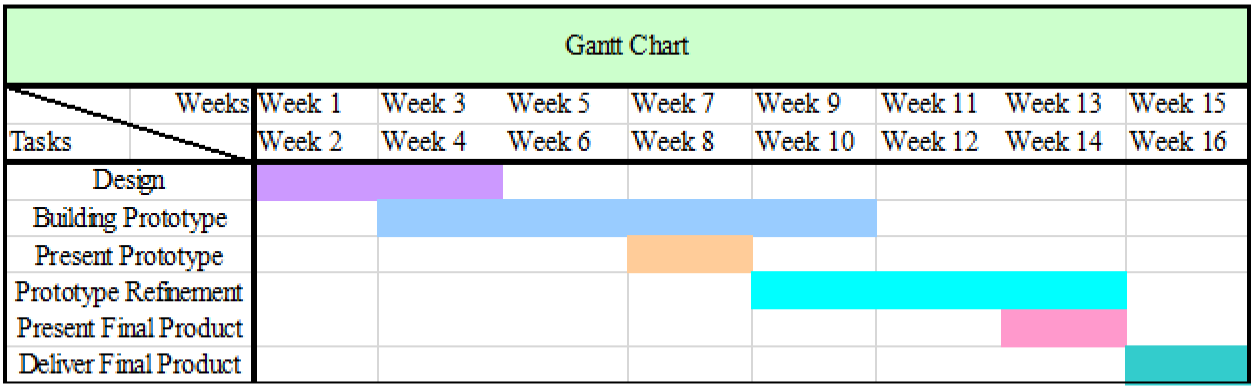

Table 3-1-1 Project Timeline

3.1.2 Software Technologies Used

1) Glassfish Server

Glassfish is an Application Server that was developed by Oracle and is Java EE compatible. It allows developers to develop and deploy their web based applications with the Java EE technology. Besides that, the server even supports the creation of connections to various kinds of databases such as Oracle, MySQL, and Derby. This server also allows developers to deploy their web developments under their own localhost which is useful for testing and debugging. These features are important to support the development of the badminton game analysis system. It acts as the middleman that supports both the rendering of the video to a website and the connection to the database.

2) HTML

HTML stands for “Hypertext Markup Language”. It is a language used to create websites and it forms the skeletal structure of the whole website. It involves markup language that includes tags such as <body>, <head>, and <p>. This language forms the main part of the badminton game analysis system where this language is used to render the segmented videos on the website.

3) Java EE

Java EE stands for Java Enterprise Edition. The Java EE provides an API and a runtime environment for executing a website that has complex logic. Under this platform, it provides several API's such as servlets, Java Server Pages (JSP), Enterprise Java Beans (EJB), validation and Java Persistence API (JPA). As for the badminton game analysis system, it utilizes most of the Java EE API's including the JSP, EJB, JPA and servlets. Generally, EJB and JPA will be used for the connection to the database where each entity in database will be mapped to a class in the java framework while the servlets will handle the logic of the website utilizing the JSP for displaying purposes. With the use of JPA, the Java Persistence Query Language (JPQL) will be used to query and store the data into the database. The use of JSP allows java codes to be rendered before the full markup page is sent to the browser.

4) Javascript

Javascript is a scripting language embedded inside a browser. With Javascript, the simple logic part of a website could be rendered. In the case of the badminton analysis system, the javascript language is used to store the data for the input of the user when the segmented videos are rendered. It would be used to collect the data and pass it to the servlets. Besides that, it is also used to store data in the web browser for different sessions.

5) Oracle Database

The oracle database is a database management system that utilizes the Structured Query Language (SQL) to allow modification, creation, adding and removing of data stored in the database. In the context of badminton game analysis, it will be used to manage the data of the input from user when the segmented videos are rendered. The operations will be rendered by the Java Persistence API using the Java Persistence Query Language.

6) C++

C++ is a general programming language that supports many type of programming such as object-oriented and even structured programing. As for the badminton game analysis, it is used in the first subsystem to process the raw videos in to segmented videos based on their important key points. C++ is used in this context because, OpenCv, an image processing library has full support for this language which makes the development process much faster.

7) OpenCv

OpenCv is an image processing library that allows many operations on images and videos. The images are represented in matrix forms where each of the cells in the matrix represents a pixel value. In the case of videos, OpenCv allows the use of the VideoCapture class which allows the user to read videos, frame by frame as a matrix. The operation could be done on each of the frames. For the system developed, the OpenCv library is used to read the video frames and then processed by using the histogram comparison function to differentiate between each frame in comparison to a template image.

8) Batch file

The batch file is a generally a script that is used in Disk Operating System (DOS) and windows. The file is meant to be used in the Command Line Interpreter (CLI) to run windows command. Due to the limitation of OpenCv in the support of the VideoWriter class which only supports writing videos in (.avi) format, an external library known as FFmpeg will be used in the batch file to convert the (.avi) videos into (.mp4) videos. The final video is required to be in a (.mp4) format because the HTML tags <video> can only support the (.mp4) extension.

9) Ffmpeg

Ffmpeg is a library used to handle multimedia data. This library can be integrated into the command line program. Therefore, this technology will be used to convert (.avi) videos into (.mp4) videos as mentioned earlier.

From the technologies written above, the first subsystem will start by processing raw videos using C++ and segmenting the videos based on changes in camera angle using OpenCv. From there on, since the videos produced are in a (.avi) extension, a batch file will be run, to convert the videos into a (.mp4) extension. The second subsystem will render the segmented videos by utilizing HTML, servlets, JSP and Javascript. After the user has input the data into the system, the EJB and JPA will be used to transfer the data in to the Oracle Database using the JPQL.

3.2 Requirement Specification

3.2.1 Use-Case Diagram

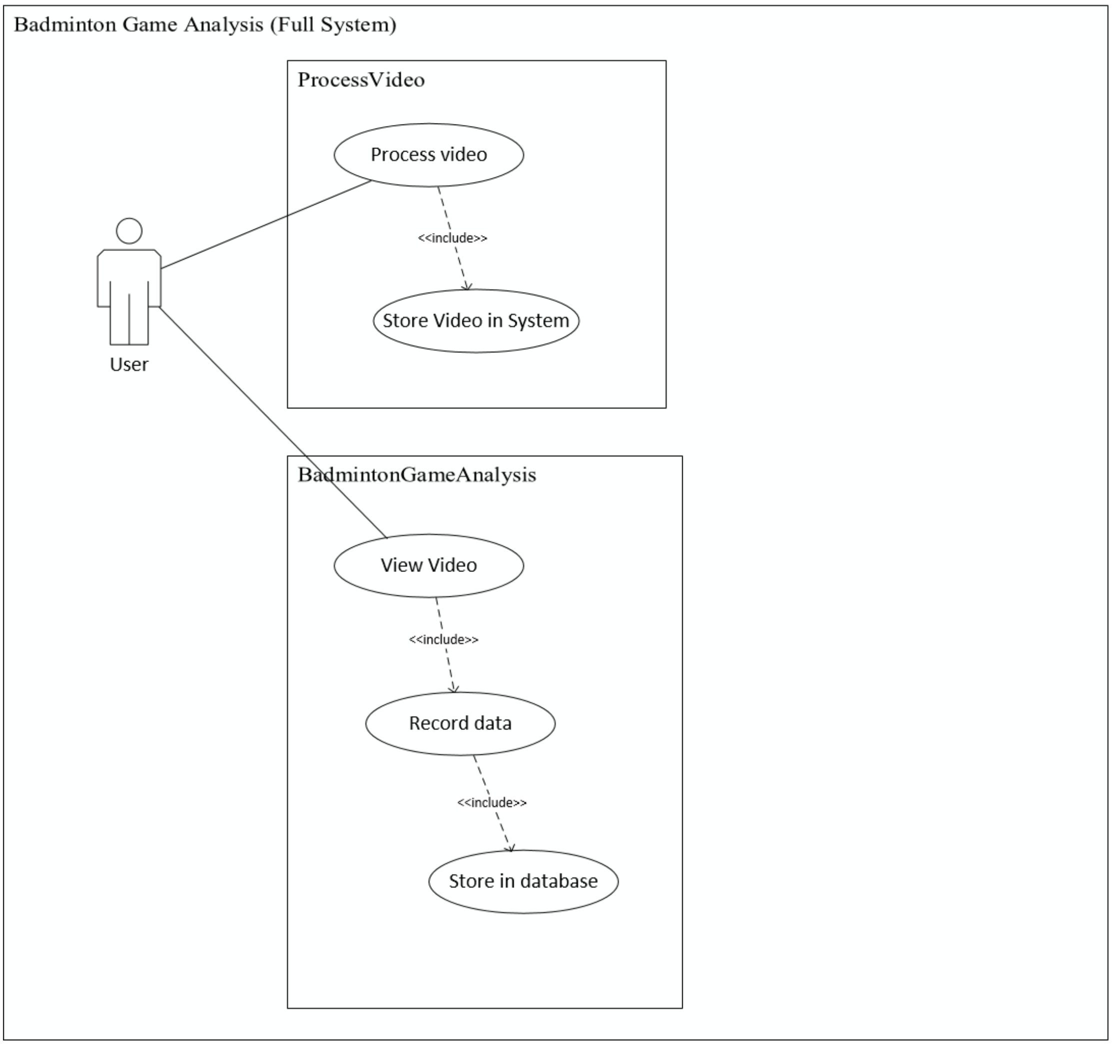

Figure 3-2-1-F1 Use Case Diagram

The Use-Case diagram illustrates the functionality of the Badminton Game Analysis System aids the user in data collection. According to the diagram above, the user will be the main actor in the system. The user must initially trigger the process video function in the first subsystem and the system will trigger the function to store the video in the system. After that, if the user deciders to view the video, the badminton game analysis system shall render the video to the user asking for relevant data from the user. The system will then trigger the record data function and the data will be stored in the database.

3.2.2 Activity Diagram

3.2.2.1 Process Video

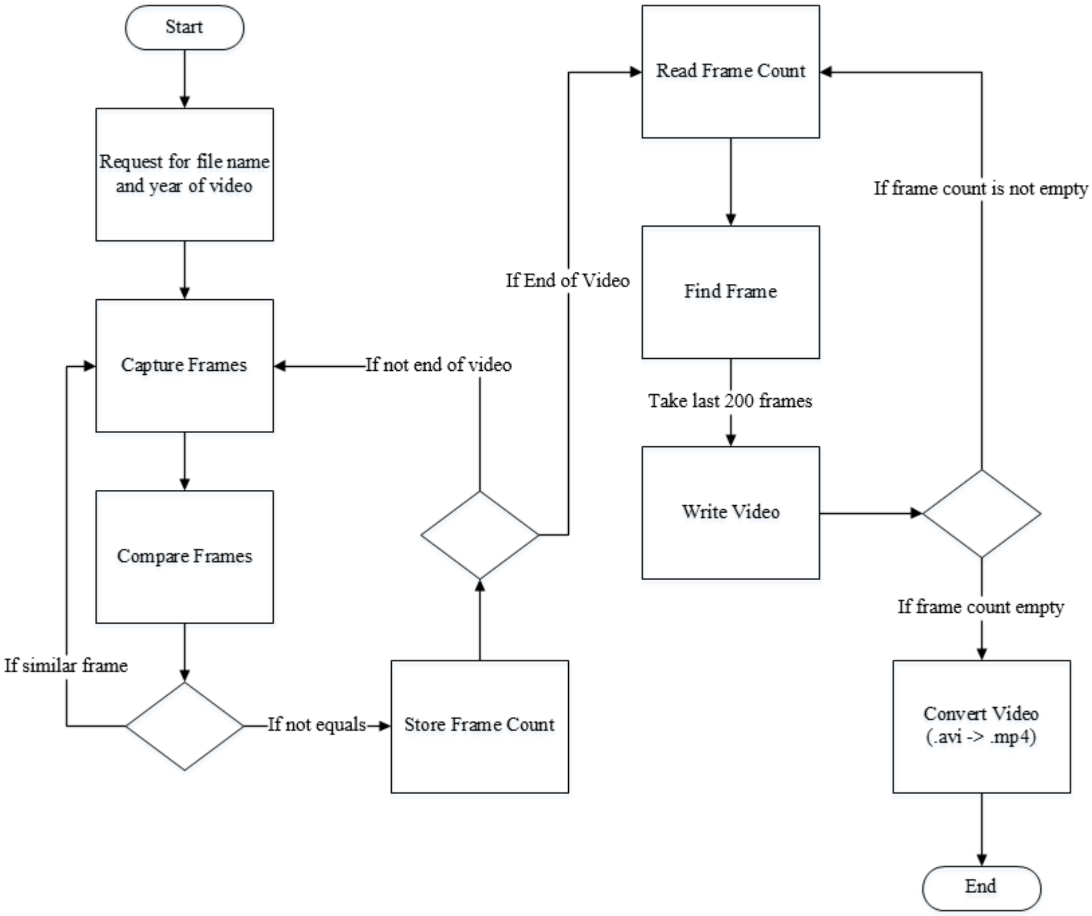

Figure 3-2-2-1-F1 Activity Diagram (Process Video)

The diagram above shows the flow of the program for process video. The first process is the request for file name and year of video. The file name is for the program to know which file to read from and the year of video is for the program to sort the videos segmented. These files or raw videos will be stored in a specific folder in the system where each file would represent a single game in a match. After the file or video has been read, the system will proceed to read the frames from the video. From there, it will start to compare the frames read with a template that is initially recorded. The template is a template of the badminton court as shown in the next page.

Figure 3-2-2-1-F2 Template Image

For the frame comparison, the color histogram for the template image above is first computed during startup of system and every time a frame is read from the current video, it's the color histogram for that particular frame is computed. The color histograms are computed by using the OpenCv function of calcHist. Then, both of the computed histogram will be compared using the compareHist function from OpenCv. The method used for comparison between the two frames is the compareHist function. A certain threshold is set where if the value returned by compareHist exceeds the original value, then the badminton court is still present. However, if the value returned does not exceed the original value, it would mean that there is a change in angle of a camera, thus recording it as an important frame. Note that only the frame count is stored into a list and not the whole frame. After all the important points for a video has been recorded, the system will begin to read each of the previously recorded values for important points. For each of the important points, the system will find that frame and take 200 frames before the current frame. This would mean that each video segmented would be of 200 frames length. For each 200 frames in a list, the system will proceed to write the values into a specific pre-specified path. The write function will be rendered by OpenCv under the VideoWriter class. Since the VideoWriter for OpenCv only supports writing videos in the (.avi) format, there will be a need to convert to videos into (.mp4) extension as the Java EE website only support certain extension for videos. Therefore, the ffmpeg video codec was added into the system. This codec will be used together in batch file to convert to videos from the (.avi) extension to the (.mp4) extension.

3.2.2.2 Badminton Game Analysis

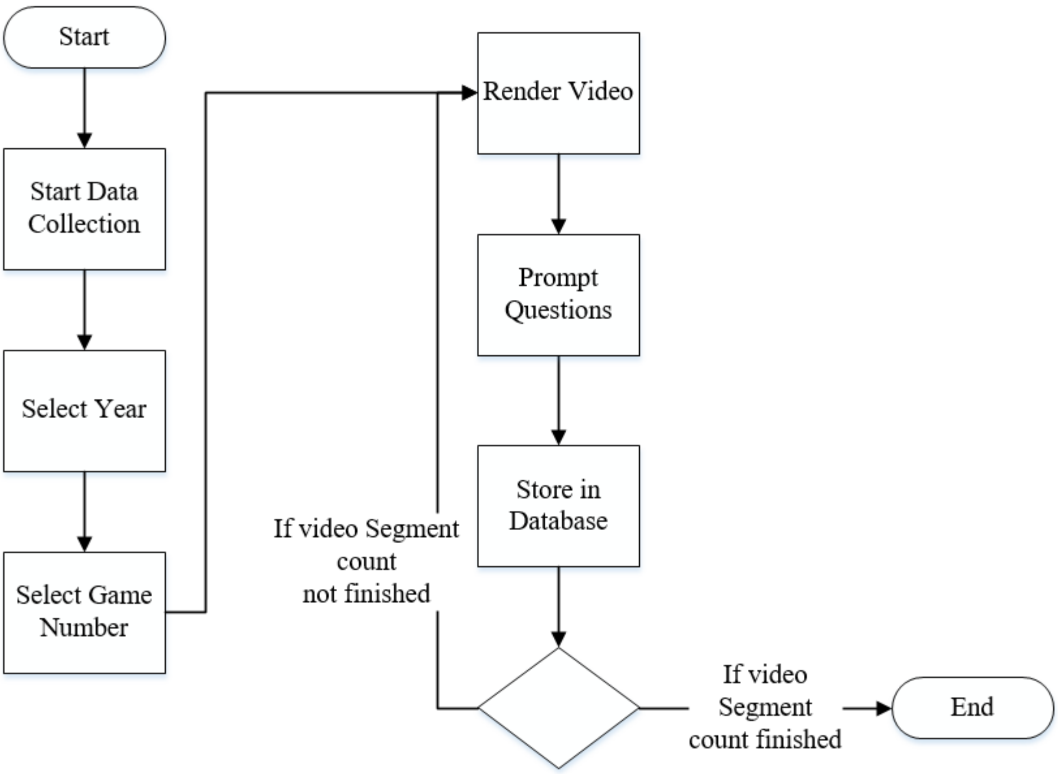

Figure 3-2-2-2-F1 Activity Diagram (BGA)

For the badminton game analysis system, the flow starts with the user initiating the data collection process. After the user has initiated the collection process, the user has to select which year he/ she would like to analyze on. After he/ she have clicked on the year, the system will show the number of games in that particular year. He/ she would then have to choose a game from that particular year. After he / she have done that, the system will retrieve the video segments for that particular game. At the end of each video, question will be prompt to the user in order to collect for data for that particular video segment. After the questions have been answered, the data will be stored on the database, and the system will loop to the next video in the segment queue. If the segment queue has ended, the system will terminate.

3.2.3 Class Diagram

3.2.3.1 Badminton Game Analysis

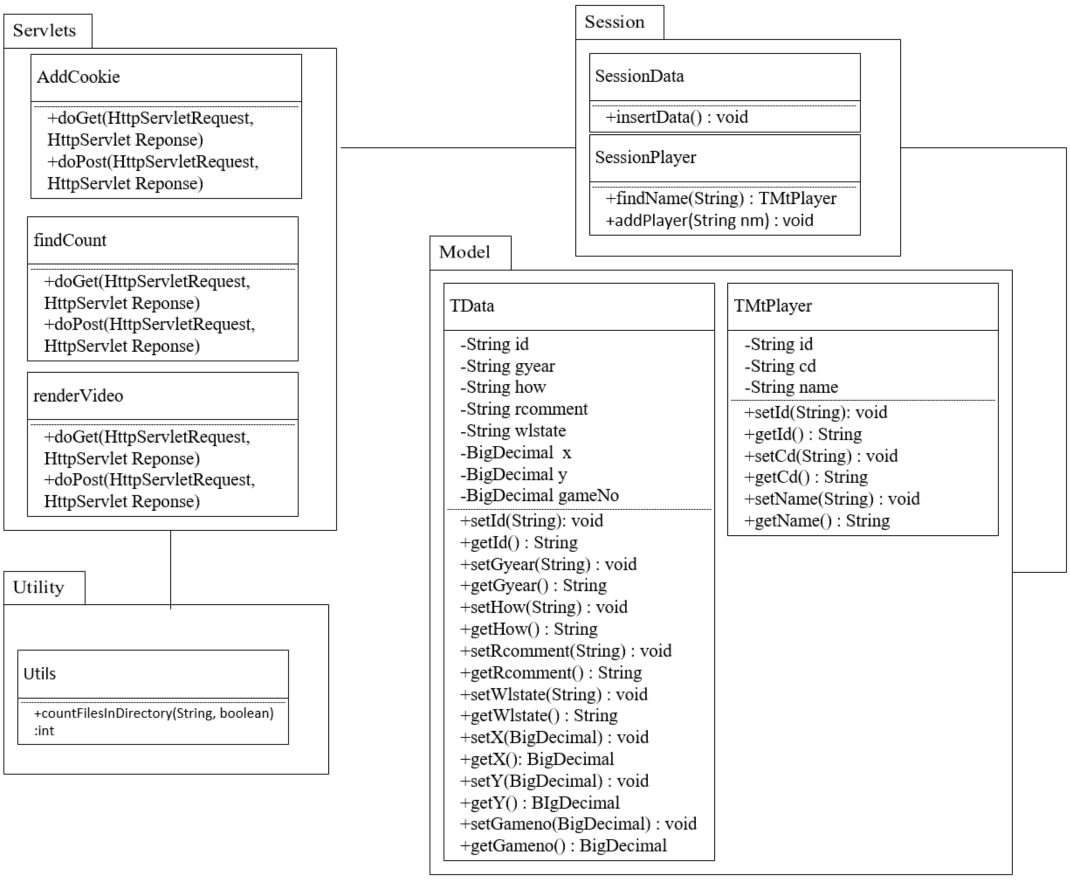

Figure 3-2-3-1-F1 Class Diagram

Based on the figure above, the servlets package will handle the logic of the system while the utility package will assist in counting the number of video segments in a particular directory. In the Session package, the classes are used to store data in the database while the Model package functions to handle the mapping between the database entity and the java class.

3.2.4 Entity Relationship Diagram

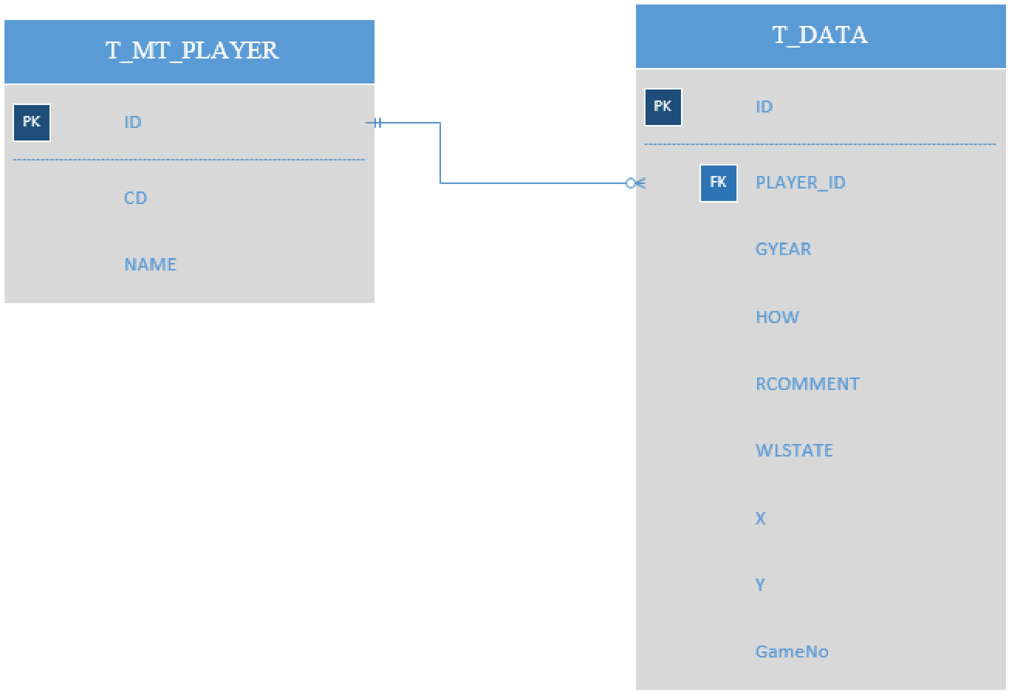

Figure 3-2-4-F1 Entity Relationship Diagram

The above diagram shows a simple relationship in the database where each player is related to none or more data in the data table. Each record in the data table however, can only relate to one and only one player.

Chapter 4: Implementation

4.1 System Architecture

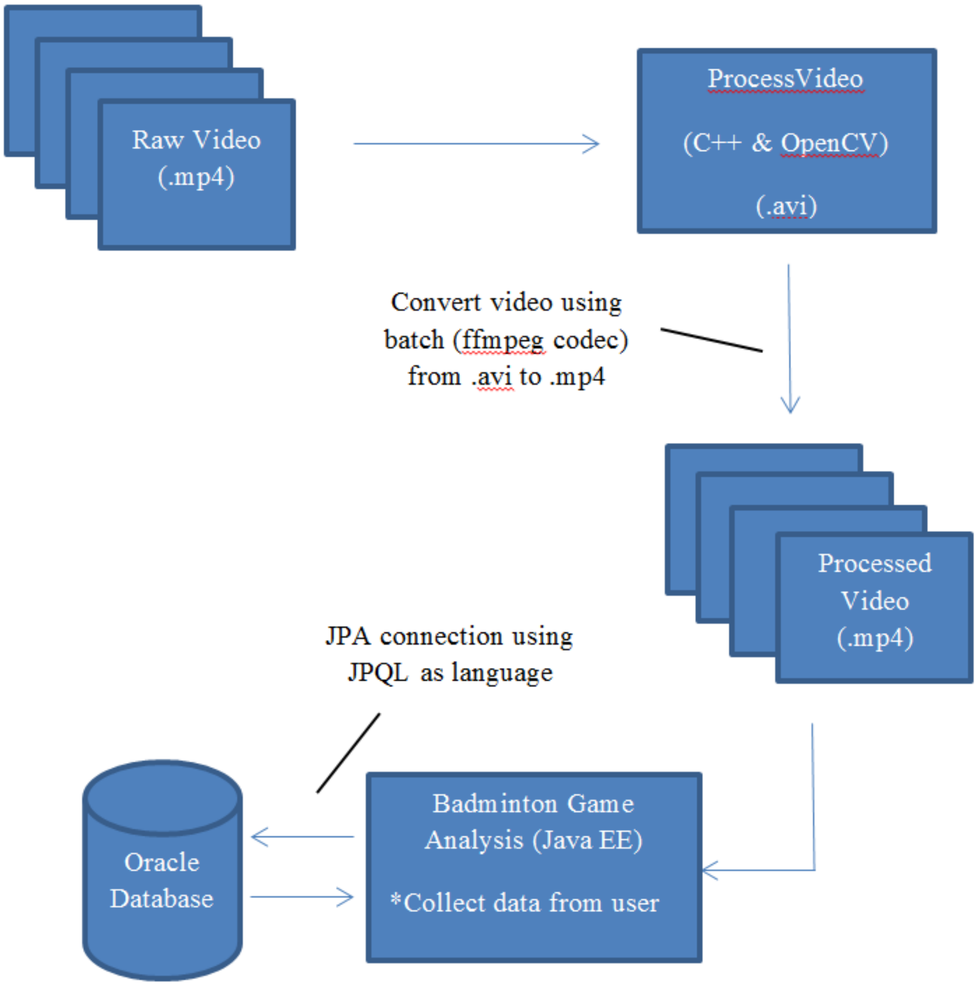

Figure 4-1-F1 System Architecture

The figure above shows two systems that provides different functionalities to the user. The first system segments the videos by using image processing tools (OpenCV) into several sections based on one main criterion which is the change in angle in a badminton game. This system takes one main assumption where it assumes that every change in angle in the video would represent an end in a rally as shown in the image below. Each video would be of a game's length and would be divided based on the above mentioned rule. In order to divide the videos, a frame comparison based on histogram matching will be performed to differentiate between the angle viewing the court and other angles.

Figure 4-1-F2 Rally Angle

Figure 4-1-F3 Change in Angle

The segment of videos generated by the first system will then be processed by the second system. The second system, developed using the Java Enterprise Edition technology as a dynamic web site, acts a platform that will allow the user to key in important notes that are relevant to the videos generated earlier. The videos would act as guidance for the coaches to keep track of the matches. For each of the videos, user would have to input on whether that particular video is the end of a rally. After that, the user would have to continue on to answer a few question to be listed here below:

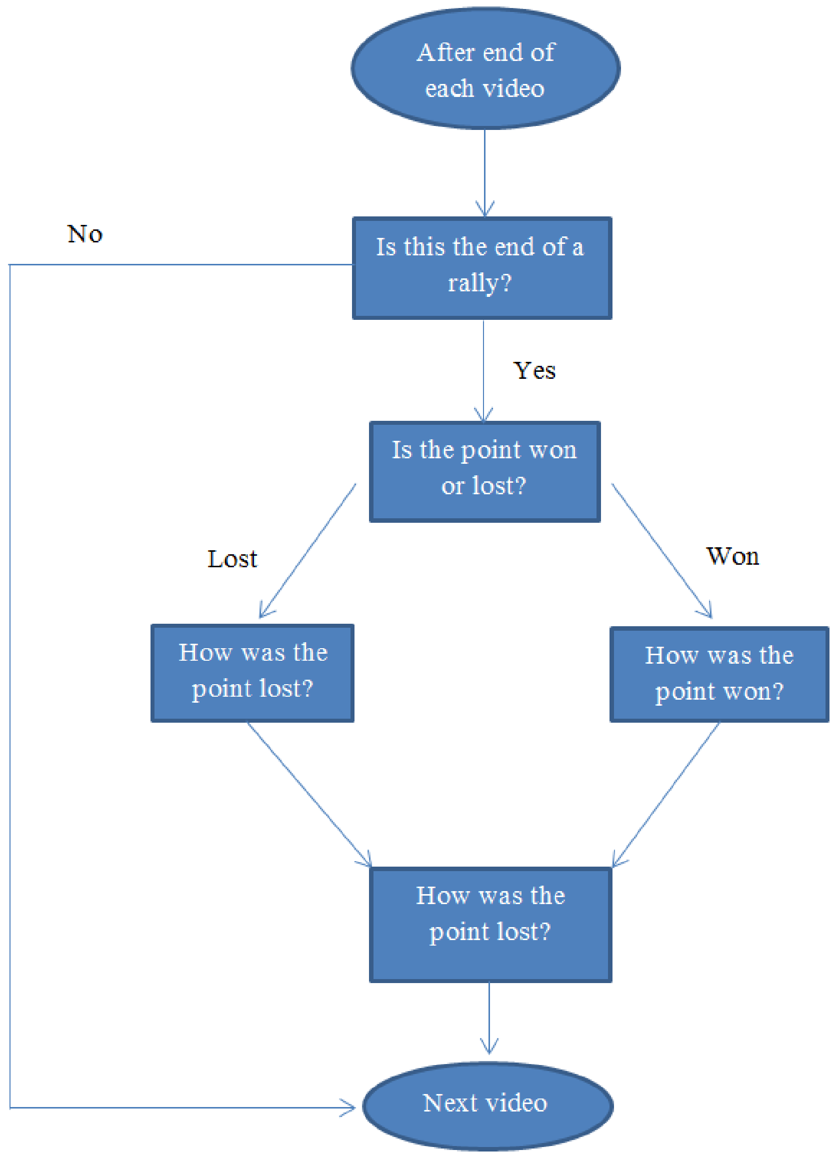

Figure 4-1-F4 Data Questions

These data obtained from the user would be stored in an oracle database by establishing a JPA connection. The data will be stored and retrieved using the JPQL through session beans that will handle the operations for the database. As for the analysis, the points previously recorded by the user would be shown and information such as most often technique used to win, area with most points won or lost, most often reason for losing points, and basic statistics such as total number of smash, drops and unforced error in a game will be shown.

4.2 Codes for The System (Process Video)

The system starts by requesting for the file name and year before it starts processing the video as shown below:

```
cout << "Please enter the video name" << endl;
getline(cin, videoName);
system("cls");
cout << "Please enter the year of the video" << endl;
getline(cin, year);
```

Figure 4-2-F1 Request Name and Year

After the both the values have been input, the system first try to open the video:

```
VideoCapture cap;
cap.open(READPATH + videoName);
if (!cap.isOpened()) {
        cerr << "Failed to open video file!\n" << endl;
        system("Pause");
        return 1;
}
```

Figure 4-2-F2 Read Video

Readpath is a constant variable that represents the location of the videofile. After that the system will first compute the histogram for the template of the court image as follows:

```
MatND computeTemp(){
        Mat temp = imread("template.jpg", 1);
        Mat tempHsv;
        MatND tempHist;

        cvtColor(temp, tempHsv, CV_BGR2HSV);
        calcHist(&tempHsv, 1, CHANNELS, Mat(), tempHist, 2, HISTSIZE, RANGES, true, false);
        normalize(tempHist, tempHist, 0, 1, NORM_MINMAX, -1, Mat());

        return tempHist;
}
```

Figure 4-2-F3 Compute Template

After the histogram has been computed, the system will continue to the index of the frames that record a change in angle as shown below:

```
while(1){
        //cap.set(CAP_PROP_POS_FRAMES, 2088);
        cap >> frame;
        if (frame.empty()){
                break;
        }

        if (compareframe(temp, frame)){
                first = true;
        }
        else {
                if (first){
                        first = false;
                        //getPrevious frame if current frame not = to template
                        endPos.push_back(cap.get(CAP_PROP_POS_FRAMES)-1);
                }

        }
}
```

Figure 4-2-F4 Store End Position

In the above figure, endPos represents a list with integer values. These integer values represent the intersection index when the camera has a change in angle. After all the values for the endPos has been stored, the system will loop endPos and write the previous 200 frames into a new list, toWrite.

```
while (!endPos.empty()){

        //set as frame before end frame
        cap.set(CAP_PROP_POS_FRAMES, endPos.front()-VIDEOFRAME);
        int loop = 0;
        while (loop < VIDEOFRAME){
                cap >> c;
                if (compareframe(temp, c)){
                        toWrite.push_back(c.clone());
                }
                else{
                        toWrite.clear();
                        writeValue = false;
                        break;
                }
                loop++;
        }

        if (writeValue){
                writeVideo(toWrite, S, videoCount, year);
                toWrite.clear();
                videoCount++;
        }
        writeValue = true;
        endPos.pop_front();
}
```

Figure 4-3-F5 Retrieve and Write

From the figure above, VIDEOFRAME is a constant that represents the amount of frames that should be in the video segment. After toWrite has been filled, the system called the writeVideo function that will write the video into the system as shown below:

```
bool writeVideo(list<Mat> temp, Size S, int count, string year){

        String name = WRITEPATH + year + "/" + to_string(count) + ".avi";
        cout << "Created video " << name << endl;
        VideoWriter output;
        output.open(name, CODEC , FPS, S, true);
        if (!output.isOpened())
        {
                std::cout << "Could not open the output video for write: " << endl;
                system("pause");
                return false;
        }

        for (int i = 0; i < temp.size(); i++){
                output << temp.front();
                temp.pop_front();
        }

        return true;
        output.release();

}
```

Figure 4-2-F6 Write Video

The above figure shows the write function of the system where WRITEPATH is a constant that represents the default storage for the video segments. After the video has been written into the file path, the system continues to convert the videos written into a (.mp4) extension.

```
cout << "Converting video to required format..." << endl;
convertVideo(year);
```

```
void convertVideo(string year){
        string convertBat = "convert.bat " + year;
        string arrangeBat = "arrange.bat " + year;
        system(convertBat.c_str());
        system(arrangeBat.c_str());
}
```

Figure 4-2-F7 Convert Video

From the previous figure, two batch files have been called in order to convert the videos into (.mp4) extension. The codes for both the batch files are shown below.

```
@echo off
setlocal enabledelayedexpansion
SET year=%1
cd "C:\Users\Simon\Desktop\SourceCode\ServerSide\BadmintonGameAnalysis\WebContent\Videos\"
if not exist %year% mkdir %year%
cd %year%
FOR %%F IN (*.avi) DO (
SET name=%%F
ffmpeg -i %%F !name:~0,-4!.mp4
del %%F
)
```

Figure 4-2-F8 convert.bat

```
@echo off
set year=%1
set count=-1
cd "C:\Users\Simon\Desktop\SourceCode\ServerSide\BadmintonGameAnalysis\WebContent\Videos"
cd %year%
for /d %%a in (dir *) do (
echo %%a
set /A count=count+1)
set /A cur=count+1
mkdir %cur%
set cur=%cur%%\%
move *.mp4 %cur%
```

Figure 4-2-F9 arrange.bat

After both the batches have been run, the segmented files will be stored in the path specified as shown below:

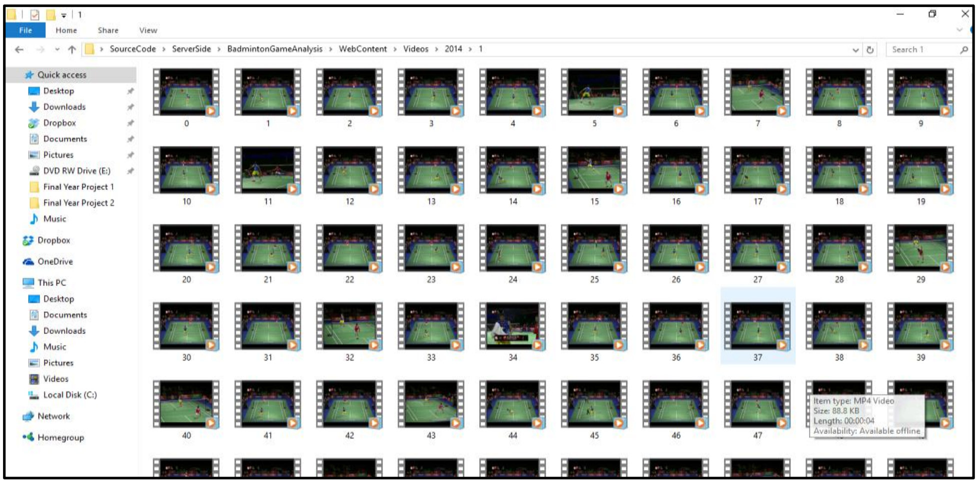

Figure 4-2-F10 final product of process video

4.3 Codes for The System (Badminton Game Analysis)

When the user starts the system, the HTML page will be rendered for the user to click. In this context, we will assume that the user will select to record data. The codes are shown below.

```
<h2><a href="instruction.jsp">Start your analysis now!</a></h2>
```

Figure 4-3-F1 HTML for starting data recording

After the user clicks on the button, the system will call instruction.jsp to render the next page. The codes for instruction.jsp are shown below.

```
<form action="findCount">
        <div>
                <h3>Please select the year of analysis</h3>
                <label>2014
                        <input type="radio" name="year" value="2014" >
                </label>
                <label>2015
                        <input type="radio" name="year" value="2015" >
                </label>
                <label>2016
                        <input type="radio" name="year" value="2016" >
                </label>
        </div>
        <br>
        <input type="submit" value="Continue">
</form>
```

Figure 4-3-F2 instruction.jsp

From the figure above, we assume that currently there are only 3 years to be analyzed. After the user clicks on one of the radio buttons above and submit, it will call the findCount servlet. The codes for findCount servlet are shown below.

```java
String year = request.getParameter("year");
List<Integer> videoCount = new ArrayList<Integer>();
int total = Utility.Utils.countFilesInDirectory("C:/Users/Simon/Desktop/SourceCode/ServerSide/FinalYearProject/WebContent/Videos/"
        + year + "/", true);
for(int i = 0; i < total; i++){
        videoCount.add(Utility.Utils.countFilesInDirectory("C:/Users/Simon/Desktop/SourceCode/ServerSide/FinalYearProject/WebContent/Videos/"
                + year + "/" + (i+1) + "/", false));
}


request.setAttribute("year", year);
request.setAttribute("total", total);
request.setAttribute("videoCount", videoCount);
RequestDispatcher req = request.getRequestDispatcher("gameNo.jsp");
req.forward(request, response);
```

Figure 4-3-F3 findCount.java

The utility function will calculate the number of folders inside the year selected by the user. It will then continue to call the gameNo.jsp to render the amount of games for each folder.

```jsp
<form id="gameCount" action="renderVideo" method="post">
        <div>
        <h3>Please select the which game</h3>
        <%

                if(year != null){
                        for(int i = 0; i < total; i++){
                        out.println("<label>" + (i+1));
                        out.println("<input type='radio' name='game' value='" + (i+1) + "'>");
                        out.println("</label>");
                        out.println("<input type='hidden' name='count' value='" + videoCount.get(i) + "' >");
                        out.println("<br>");
                        }

                }
                out.println("<input type='hidden' name='year' id='year' value='" + year + "' >");
        %>
        <br>
        </div>
        <input type="submit" value="Continue" onclick="storeData()">
</form>
```

Figure 4-3-F4 gameNo.jsp

For the above function, after the user has selected the game and click the submit button, the system will call the renderVideo servlet and at the same time call the storeData function to keep track of the year and game no that the user has selected.

```
function storeData(){
        var g = document.getElementsByName('game');
        for ( var i = 0; i < g.length; i++) {
        if (g.item(i).checked) {
            var newVal = g.item(i).value;
            var pos = i;
        }
    }

        var y = document.getElementById("year");
        var c = document.getElementsByName("count");
        var countVal = c.item(pos).value;

        sessionStorage.setItem('game', newVal);
        sessionStorage.setItem('year', y.value);
        sessionStorage.setItem('count', countVal);
}
```

Figure 4-3-F5 storeData

From the above figure, the sessionStorage is a function of javascript which helps store data in a web browser for different sessions. The following show the codes for renderVideo.

```java
            int videoCount = 0;
            HttpSession session = request.getSession();
            if(session.getAttribute("videoCount") == null){
                session.setAttribute("videoCount", videoCount);
            }
            else{
                videoCount = ((Integer)session.getAttribute("videoCount") + 1);
                session.setAttribute("videoCount", videoCount);
            }
            String game = request.getParameter("game");
            String year = request.getParameter("year");
            String total = request.getParameter("count");

            if(request.getParameter("rally") != null){

                if(request.getParameter("rally").equals("yes")){

                    String player = request.getParameter("player");
                    String x = request.getParameter("x");
                    String y = request.getParameter("y");
                    String state = request.getParameter("state");
                    String how = request.getParameter("how");
                    String comment = request.getParameter("com");
                    List<String> temp = new ArrayList<String>();
                    temp.add(year);
                    temp.add(how);
                    temp.add(comment);
                    temp.add(player);
                    temp.add(state);
                    temp.add(x);
                    temp.add(y);

                    data.insertData(temp);
                    /*System.out.println("Data for " + player + ": ");
                    System.out.println("Position: " + x + ", " + y);
                    System.out.println("Won/Loss: " + state);
                    System.out.println("How?: " + how);
                    System.out.println("Comments: " + comment);*/

                }
            }

            request.setAttribute("game", game);
            request.setAttribute("total", total);
            request.setAttribute("year", year);
            request.setAttribute("videoCount", videoCount);
            RequestDispatcher req = request.getRequestDispatcher("court.jsp");
            req.forward(request, response);
    }
```

Figure 4-3-F6 renderVideo.java

From the figure above, if the rally data has not been set, the servlet will automatically dispatch the request to court.jsp.

```
<%
    int videoCount= (Integer)request.getAttribute("videoCount");
    String  year = (String)request.getAttribute("year");
    String  game = (String)request.getAttribute("game");
    String  total = (String)request.getAttribute("total");

    String path = "Videos/" + year + "/" + game + "/" + videoCount + ".mp4";
    out.println("<canvas id='crt' class='left' width='400' height='600'");
    out.println("style='border:1px solid #c3c3c3;' onclick='getPoint(event)'></canvas>");
    out.println("<video id='display' class='center' controls>");
    out.println("<source src='" + path + "' type='video/mp4'>");
    out.println("</video>");
%>
```

Figure 4-3-F7 court.jsp

From the previous figure, there will be a canvas that will be hidden initially that represents the badminton court and a video tag that will show the videos. Handled by the javascript, if the video ended, there will be a prompt to the user asking if the rally has ended.

```
document.getElementById('display').addEventListener('ended',myHandler,false);
function myHandler(e) {
  DlgShow("Is it the end of a rally?");
}
```

Figure 4-3-F8 eventListener

The event listener shown above is a handler that's added to the video tag earlier. If the user clicks on yes, the court will be rendered out and several questions will be asked, namely,

1) Please point the position of the shuttle at the end of the rally.
2) Was the point won or lost?
3) How was the point won?
4) How was the point lost?
5) Any Comments?

After the final answer, the results will be stored and sent to renderVido again. If there is data, the sessionData will be called.

```
@EJB
private SessionData data;
```

```
data.insertData(temp);
```

Figure 4-3-F9 sessionData

```
    public void insertData(List<String> s){
        UID id =  new UID();
            TData d = new TData();
            TMtPlayer p = session.findName(s.get(3).toUpperCase());
            d.setId(id.toString());
            d.setGyear(s.get(0));
            d.setHow(s.get(1));
            d.setRcomment(s.get(2));
            if(p == null){
                session.addPlayer(s.get(3));
                p = session.findName(s.get(3));
            }
            d.setTMtPlayer(p);
            d.setWlstate(s.get(4));
            d.setX(BigDecimal.valueOf(Double.valueOf(s.get(5))));
            d.setY(BigDecimal.valueOf(Double.valueOf(s.get(6))));
            em.persist(d);
        }

}
```

Figure 4-3-F10 insertData

The above figure shows the data being insert into the database based on the model of TData created using the JPA connection.

4.4 Implementation Issues and Challenges

There are some amounts of issues encountered during the development of this system. The first was when developing the process video system. The project initially was targeted to utilize OpenCv to detect the motion of the shuttlecock. However, due to some limitations in my knowledge of image processing, the function could not be implemented. Therefore, we sorted for the technique to compare frames based on camera angles and divide them based on end of each rally.

Even after the changes in techniques, there were more issues to come. We found out that for each video, if it consists of 3 full games, then the interval between each game would have rendered out many useless segments of videos. Therefore, we sorted to cut the video into 3 segments where each represents a game.

After the issue with the raw videos was solved, there was an issue in writing the video in a format where the dynamic website could support. OpenCv only supports writing in a (.avi) extension. After some research, we found out that we could use an external library known as FFmpeg combined with a batch file to convert the (.avi) extension into (.mp4) extension.

During the second phase of development, we found out that the connection using JPA to the database had some issues. Initially, an Apache Tomcat server was used but during the process of development, we found out that the Tomcat server does not support Java EE. Hence, the system was not able to connect to the database. Therefore, we sorted for a different option and decided to use the glassfish server.

During the development of the whole system, I found out that combining all programming knowledges in the development process was a time consuming and difficult process. Therefore, the system could not be delivered fully. However, the basic functionalities of the system were managed to be developed. The final tests for the system will be discussed on the next chapter.

Chapter 5: Experimental Results

5.1 Experimental Environment Settings

The environment setup for this project is conducted on the host system itself. The first test shall test the error type for the system, namely type I error and type 2 errors. The percentage for the errors will be calculated. A predefined expected result will be recorded beforehand. For the system, the video segments produced are compared with the actual results to see if the key point selection by the system is accurate. If there was a point scored and the system did not record it, it would be considered as a type I error while if there was no point scored, but the system recorded it, it would be classified as a type II error.

In this setting, there will be a setup for both the system and manual methods of recording. Both times to record the data from each game will be recorded. For the system, the time for the first subsystem which is the process video will be excluded as it can be run before the test begins.

Using the same setting, the accuracy for both the methods shall be calculated as well. This will measure if the input given by the user are correct based on each points collected. The final data produced by both the methods will be compared with the actual results.

The tests were conducted for 5 games in a row. The average results were then calculated.

5.2 Control Tests

5.2.1 Results on Accuracy and Its Limitations

As far as the objective is concerned, during the tests, it did manage to help the user to key in the values faster than normal manual methods of recording. Time for each record of data was recorded. At real time, two people will keep track of the type I and type II errors for each record made by both the system and the manual methods.

The results of the tests are shown below:

Type I and Type II errors made my system when segmenting videos compared to manual methods of choosing key points.

		Type I error	Type II error
Game 1	System	0	30
	Manual	0	0
Game 2	System	1	20
	Manual	0	0
Game 3	System	2	10
	Manual	0	0
Game 4	System	1	20
	Manual	0	0
Game 5	System	1	30
	Manual	0	0

Table 5-2-1-1 Type I and Type II Error

The second test done was regarding the time to process each game. Hence, the results are as shown below:

		Time (avg)
Game 1	System	5s / record
	Manual	10s/ record
Game 2	System	5s / record
	Manual	12s/ record
Game 3	System	5s / record
	Manual	12s/ record
Game 4	System	6s / record
	Manual	13s/ record
Game 5	System	6s / record
	Manual	14s/ record

Table 5-2-1-2 Time for Each Record

By average,

System: (5 + 5 + 5 + 6 + 6)/5 = 5.4 s/record

Manual: (10 + 12 + 12 + 13 + 14)/5 = 12.2 s/record

The final test was to get the accuracy of the input given by user of each of the record made. It will be calculated based on the number of points in a game.

		Accuracy
Game 1	System	36/37
	Manual	36/37
Game 2	System	29/30
	Manual	30/30
Game 3	System	33/35
	Manual	35/35
Game 4	System	32/34
	Manual	33/34
Game 5	System	37/39
	Manual	38/39

Table 5-2-1-3 Accuracy of Data Collection

On average,

System: (36 + 29 + 33 + 32 + 37) / (37 + 30 + 35 + 34 + 39) * 100% = 95.43%

Manual: (36 + 30 + 35 + 33 + 38) / (37 + 30 + 35 + 34 + 39) * 100% = 98.28%

From the experiment, it can be seen that there certain pros and cons for the system in comparison to the manual method. It is found that the system is quite reliable in terms of speed performance. There is a significant difference between both the methods. It can be seen for the manual method, the speed of recording get longer and longer after each game. This proves that as the number of game increases, speed of recording gets longer. If compared to the system method, the speed of recording is quite constant. However, for the first test conducted, it can be seen that the Type II error rate is quite high. It terms of that, since the process video system only segments videos based on the change in angle for that particular game, there will certainly be quite an amount of segments that would not be used. It can also be seen that the Type 1 error was kept at minimum by using this system. In terms of accuracy of the results, the manual method was more accurate in comparison to the system.

The limitation of this system as can be seen in the above results is when the video segmented are more than the actual points scored in the match. This could be further improved by adding more constraints on the video segmenting by the process video system rather than just a change in angle.

Besides that, another limitation that can be observed is that the accuracy of the results. Even though it is still not very accurate, however if give compensation to time, the overall result of the final system is still quite reliable. Besides, the accuracy of the manual method was decreasing at the final stages of the test.

In addition to that, another limitation found is that this system is only limited to players in the men’s singles category. If further constraints can be formed during the processing stage, then more categories could be included into the analysis.

Last but not least, the system is only limited to recording data, while retrieving data for the purpose of analysis has not been implemented yet. Therefore, this could be done in the future work that will be discussed later on in Chapter 6.

5.3 Discussion

During the implementation of this system, it was found that recording data from badminton match by video using a system was not an easy task. Several methods of analyzing videos using image processing libraries were research but to no avail. For example, initially the project was targeted to detect the motion of the shuttlecock along with its final placement in the court after each rally. However, the object detection algorithms is not easy to be implement as the shuttlecock in the video is small and are subject to noises in the background.

Even though there were certain limitation discussed earlier, those limitations were irrelevant because it was out of the scope of this project. This project is only targeted to aid and ease the data recording process allowing users to store the data in the database. Since there was improvement in time for recording with constant accuracy, it can be seen that this system has accomplished its objectives.

Overall, I have developed a badminton game analysis system allowing user to record data from badminton matches by video. In addition, the system is able to store the data recording in a database. I have also experimented on the system where it shows that the use of this system speeds up the time for data recording of badminton matches by video without much sacrifice in accuracy.

Chapter 6: Conclusion

6.1 Conclusion

In a nutshell, there were no current systems that aid the user in data collection for badminton matches by video. This case leads to the need for a system that is capable of such capability. The present systems at the best provide an interface for the users to record the data. However, none were able to help in the data collection.

This development project was mainly focus on developing a system that is simple yet able to aid user in the collection of data such as smashes and drop for badminton matches by video. The proposed solution was to divide the system into 2 subsystems where the first subsystem functions to segment the raw video into video segments that represent each important key point of that particular match. The second subsystem’s function was to render the videos segmented by the first subsystem. The system would then continue on to prompt the user a few question regarding that particular key point. The data compiled will then be stored in a database.

Multiple tests were done in order to test the system to its limit. The details of the tests were discussed in Chapter 5. After the details were discussed, several reviews were done to ensure that the final system complies with the objective of the project. However, there were several limitations that were previously discussed. The future works on this project will be elaborated in the next section

6.2 Future Works

Referring to the previously limitations discussed; there are several future works that could be done. As previously suggested, the implementation of this project is only limited to the men's single category. Therefore, future works on this project could implement several more categories.

Another limitation of this project was the lack of constraints in the video processing part of the system. This system relies only on one constraint which is the change in angle. This lack of constraint causes a large number in the Type II error discussed in Chapter 5. Hence, future works could implement more sophisticated processing before the video is render to the user.

Another future work that could be done is, the system should include the analysis where user could have a look at the overall data recorded, including some analytics statistics.

As for the interface, a more sophisticated, more user-friendly interface could be added to the system as well.

REFERENCES

Badminton Game Analysis App Review 2012, (video recording). Available at: <https://www.youtube.com/watch?v=IX4laSqfYNY> [Accessed 20th March 2016].

Bishop, D. n.d., *How performance analysis can improve your coaching methods.* Available from: <http://www.pponline.co.uk/encyc/how-performance-analysis-can-improve-your-coaching-methods-39#>. [Accessed 5th August 2015].

Emptage, N. 2014, *Game Theory: How Do Opponents' Strengths Affect Your Team's Performance.* Available from: <http://puckprediction.com/2014/02/05/game-theory-how-do-opponents-strengths-affect-your-teams-performance/>. [5th August 2015].

iTUSA, n.d., *Match Analysis*. [Online] Available from: <http://www.itusatennis.com/> [Accessed 14th August 2015].

Logomatch, n.d., *Logomatch is a powerful, versatile, intuitive, multi-platform Video Analysis Software: Features*. [Online] Available from: <http://www.longomatch.com/> [Accessed 14th August 2015].

MatchAnalysis, n.d., *What We Do*. [Online] Available from: <http://matchanalysis.com/index.htm> [Accessed 14th August 2015].

Methods of Data Collection: *Physical and Mental Factors* n.d.. Available from: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj17breyr3JAhXHTI4KHQioBz0QFggoMAA&url=http%3A%2F%2Fkylenewhigherpe.wikispaces.com%2Ffile%2Fview%2FMethods%2Bof%2BData%2BCollection%2Bfor%2BMental%2Band%2BPhysical%2BFactors.docx&usg=AFQjCNHkhJYPdLFfpnZP7ShRAKL6mV7j-A&sig2=ZSMAKh4F6Eq3Sbz5DZ47SQ&bvm=bv.108538919,d.c2E&cad=rja>. [5th August 2015].

Performance Analysis, n.d.. Available from: <http://sini.co.uk/what-we-offer/performance-science/performance-analysis/>. [5th August 2015].

Prozone, n.d., *Matchviewer: Discover the Nature of Performance*. [Online] Available from: <http://www.prozonesports.com/product/matchviewer> [Accessed 14th August 2015].

Soccerlab, n.d., *Soccerlab Video Analysis Pro*. [Online] Available from: <http://www.soccerlab.com/> [Accessed 14th August 2015].

Videomatch, n.d., *Analyze.* [Online] Available from: <http://www.videomatch.it/?lang=en> [Accessed 14th August 2015].