University of the West Indies

Capstone Project COMP3901

The Red Stripe Premier League Fantasy Football Application

(RSPA)

Supervisor: Dr. Simon Ewedafe

Group Members: Jodi- Marie Taylor

Romario Tomlin

Nathan Nash

Leon Facey

##### Acknowledgments

We, the members of The Red Stripe Premier League Fantasy Football Application (RSPA) would like to express our my deepest gratitude to our supervisor, Dr. Simon Ewedafe , for his thought out guidance and constant support. We place on record as well, our sincerest thanks to Dr. Coore who gave a considerable amount of his time helping in this project.

We would also like to express special appreciation to The English Premier League Fantasy Football Application for this amazing idea. Also, to our UWI colleagues for helping us to test our application.

Finally, we thank our friends and fellow final year computing majors for lending a supportive hand and giving their valued opinions and expertise throughout the duration of the project.

**Table of Contents**

1. Introduction………………………………………………………………………... 4
   1. Purpose……………………………………………………………………..4
   2. Problem Definition………………………………………………………....4
   3. Proposed Solution……………………………………………………….....4
   4. Product Scope………………………………………………………...........5
   5. Problem Motivation and Justification……………………………………...5
2. Background………………………………………………………………………….6
   1. Objective…………………………………………………………………….8
   2. Requirements……..………………………………………………………....9
      1. Non- Functional Requirements……………………………………...9
      2. Functional Requirements…………………………………………….9
   3. Implementation Constraints…………………………………………………10
   4. Limitations and Issues………………………………………………………10
3. Methodology………………………………………………………………………....11
   1. Architectural Design………………………………………………………...12
   2. System Overview…………………………………………………………....14
   3. Use Case Diagram…………………………………………………………...15
   4. Sequence Diagram…………………………………………………………...16
   5. Deployment Diagram………………………………………………………...17
   6. Activity Diagram……………………………………………………………..18
   7. Entity Relationship (ER) Diagrams………………………………………....19
   8. Flow Chart……………………………………………………………...…...20
   9. Computation………………………………………………………………....21
4. Results…………………………………………………………………………….....23
   1. Accomplishments…………………………………………………………...23
   2. Screenshots of Product……………………………………………………...25

5. Conclusion…………………………………………………………………………...26

* 1. Problems Encountered and Limitations……………………………………..31
  2. Project Scope That Was Not Accomplished…………………………………31
  3. Improvements……………………………………………………………......32
  4. Concluding Words…………………………………………………………...32

6.Appendix.………...………………………………………………………………...33

##### 

##### Introduction

**Purpose**

This document gives the project report for The Red Stripe Premier League Fantasy Football Application. The scope of the application covered in this document addresses the lack of knowledge and exposure of the Red Stripe Premier League in jamaica .

**Problem Definition**

The Red Stripe Premier League is the biggest league in Jamaica however it is is not well publicized across Jamaica or the Caribbean. Therefore, the Premier League is not maximizing its audience which affects the league with funding and other areas. Also, Individuals are unable to access up to date information on scores, matches and breaking news easily. There is lack of awareness , knowledge and exposure of the league.

**Proposed Solution**

The solution to the problem is to create a website that will engage its target audience and allow them to become more involved in the sport virtually. This application will allow them to create their own team with their own players and compete with their peers for higher rankings. This will make users feel apart of and more interested in the league. The website, along with creating teams, will allow users to view matches, scores, news and other things that will captivate its users and inform them on what is happening in the league to keep them updated.

##### 

##### Product Scope

The football fantasy league app will be a web application for the local Red Stripe Premier League. Users will only be allowed to select 1 team with 11 players from a list of all the players participating in the league as well as three bench players. The user can select among 3 formations which will dictate how they choose their players. Depending on the formation, users can select a certain amount of Goal Keepers, Midfielders, Defenders and Strikers.The balance given will be a fixed $1 million JMD on sign up. Players will be priced differently depending on how well they performed the previous season and based on popularity. Users are only allowed to make changes to their teams 24 hours before each match is played. The project will be conducted over a period of 3 months.

##### Justification and Motivation

Fantasy football games are becoming more popular with several games being developed for different leagues and sports around the world today. There has yet to be one developed in Jamaica, specifically for the Red Stripe Premier Football League. Being football enthusiasts, the group wanted to develop an application that would be fun , interesting and geared toward the development of Jamaican football. Therefore, it would be a great opportunity to bring an application to Jamaica’s biggest league. We believe it will be an exciting, educational and enlightening experience for users and players in Jamaica. Having this application will aid in in improving the league and alleviating the aforementioned issues.

##### Background

The implementation and design of RSPA required the use of various technologies and tools. The implementation was carried out using major tools such as Cloud9, Django and Python. Cloud9 is an open source, online integrated development environment which supports numerous programming languages. The latter is a Python-based web framework that provides tools and libraries to build an application. It is a minimal framework so it does not impose its own structure on web applications thereby giving its users more flexibility to integrate new components with ease.

Additionally, Django depends on a modern and designer-friendly templating language known as J[inja](http://jinja.pocoo.org/). This language promotes the division of templates into components, allowing for code reuse on the front end. Jinja also provides helpers for iteration and conditionals, making it easier to display dynamic content. The popular Hypertext Markup Language (HTML) and Cascading Style Sheet (CSS) are widely used in this application. MYSQL database is integrated in the software. It is an open source C library that doesn’t require a separate server process.

There are countless products that are similar to what this project aimed to accomplish. These include Fantasy Football Manager,Goal Fantasy Football, English Premier League, Serie A Fantasy Football and RealFevr Fantasy Leagues, to name a few. The following table is a summary of each product’s functionalities/features inclusive of RSPA.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | EPL | Football  Manager | RealFevr Fantasy Leagues | Serie A Fantasy Football | Goal Fantasy Football | RSPA |
| Logging in |  |  |  |  |  |  |
| Create a team |  | × |  | × |  |  |
| Buying Players |  | × |  | × |  |  |
| Shows match  Venue and time |  |  | × |  |  |  |
| Video Highlights |  |  |  |  |  | × |
| Weekly Score |  | × |  | × |  |  |
| Injury Update |  | × |  | × |  |  |
| Live News |  |  | × |  |  | × |
| Forum |  |  | × |  |  | × |
| Online Support |  | × |  | × |  | × |

**Objectives**

* To allow users create an account.
* Allowing users to choose between three different formations.
* To purchase players for your team
* To view league table
* To view total and weekly scores
* To allow users to transfer and substitute players
* To have an attractive and user friendly interface for ease of use
* To provide an interactive fantasy premier league for Jamaica
* To give the students of UWI an opportunity to interact with the UWI football team
* To provide exposure for the local Red Stripe Premier League

**Functional Requirements**

1. To allow users to create an account and login.To calculate the player points, the total points for all prior gameweeks and the user balance.
2. To allow admin to manipulate the data for every player in the database.
3. To allow users to transfer and substitute players
4. To rank users in terms of place in the fantasy league table so they can see their current positioning.
5. To view up to date information on matches and news in the league

**Non-Functional Requirements**

1. System should be password protected
2. System should be accessible where there is internet
3. The interface should be the same throughout different web browsers
4. To have an attractive and user friendly interface for ease of use
5. To provide an interactive fantasy premier league for Jamaica
6. To provide news, updates and fixtures for the Red Stripe Premier League.

##### 

##### 

##### Implementation Constraint

* Time - The time to build this project is limited to 3 months. This was cut even shorter due to outside interference with other school work
* Cost- There is a cost for cloud9 which was used for collaboration
* Programming Language/ Web technologies - It will be built with python 3.5 and web framework django
* Database storage - the database used is mysql 5.7 and it will serve as the SQL engine
* The application will be constantly maintained by administrators in order to keep player scores, matches and relevant news up to date. However, most of the information will be done through web crawling <http://uk.livescore.eurosport.com/football/digicel-premier-league/> which is the official site for the premier league
* Collaboration - cloud9

**Limitations and Issues**

* People are allowed to join only up to 4 weeks - Reason make it more competitive and allow for every player to have a fair and realistic chance of winning the league
* Cost of labour to watch the matches and enter the information into the system
* The points of users and league tables are only confirmed 2 hours after the last match of the week - reason is because the system has to wait on input from the admin and the sites to get updated
* Only allowed to transfer players a day before the first match of the game week .
* Player prices will not change until the end of the season

##### 

##### Method

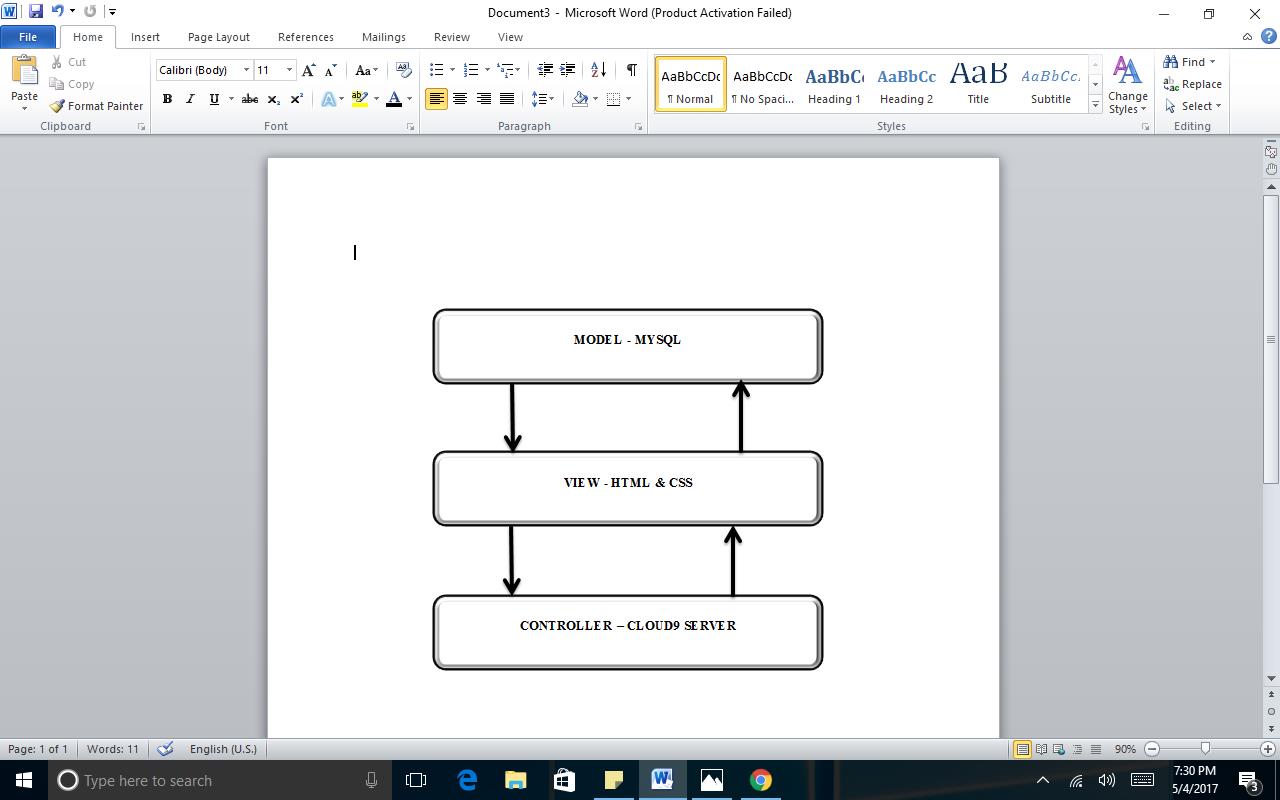
RSPA was done using a waterfall approach and it also incorporated a slightly agile method to create a hybrid-like structure (shown in figure below). Many requirements were established before the start of the project, however others were identified during the implementation and design stages. This approach led to the development of a useful and functional system.

Hybrid-like Structure



##### Design and Architecture

In developing RSAP, the widely used Model-View-Controller (MVC) architecture pattern for web applications was utilized to achieve each function. A view layer was added on top of the database layer to allow the users to interact with the database and access pertinent data. Django, a python driven application was used to process requests from the view layer to the Database layer, by generating the specific queries and operations on the database. MYSQL was the database used in the development of the system. The diagram below shows a simple representation of the MVC pattern.



* The model contains the application’s data
* A view which displays some portion of the underlying data and interacts with the user
* The controller mediates between the model and the view and manages the notifications of state changes.

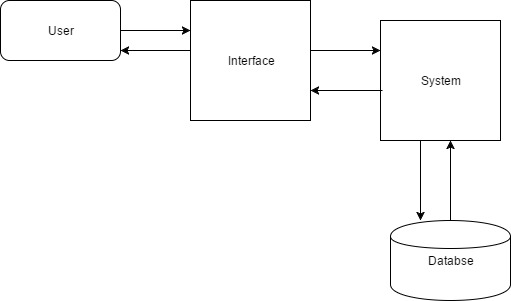
The MVC pattern thus separates user interface functionality from application functionality while being responsive to user input or to changes in the underlying application’s data. It also allows for multiple views of the user interface thereby giving each of the three user types different access or functions within the system even if underlying application data changes.

##### 

##### 

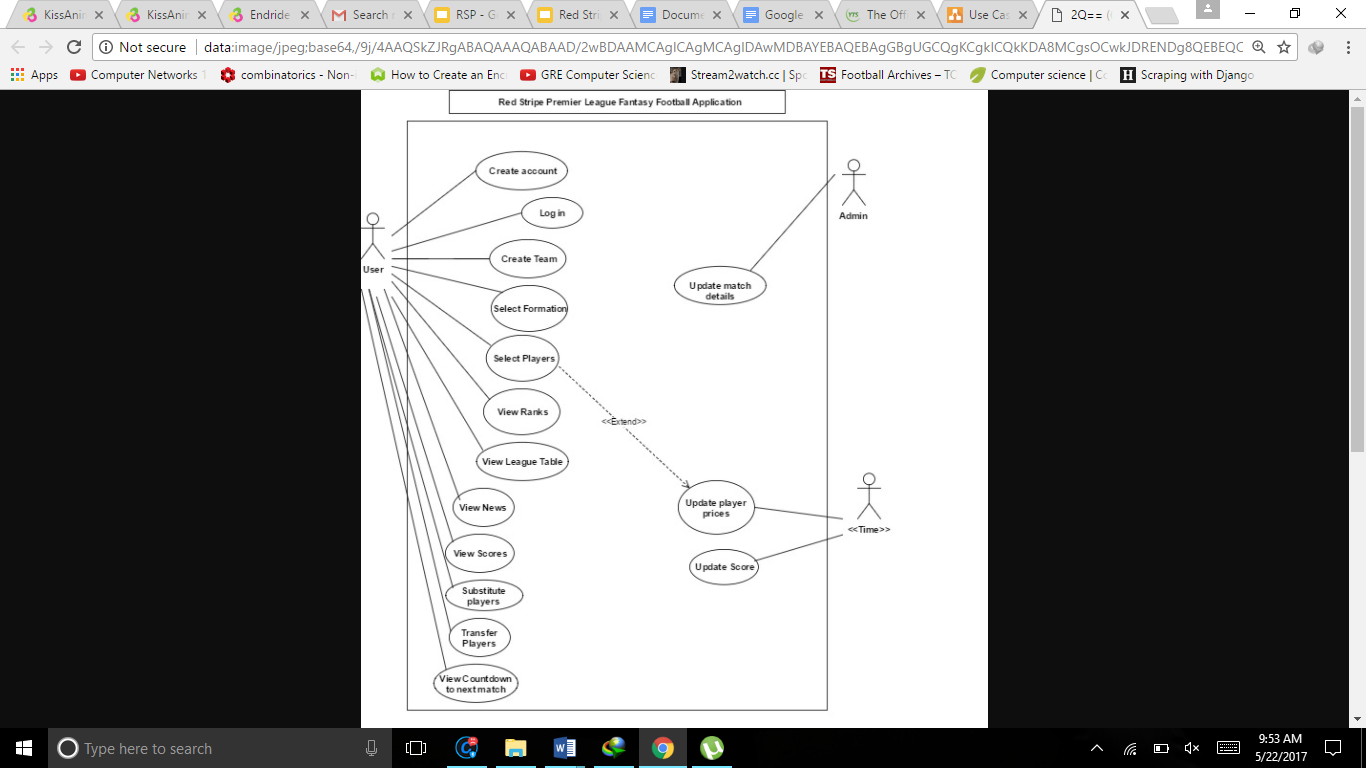
##### Diagrams

**System Overview**



This diagram shows the building blocks of the system. The user interacts with the interface of the system. The system communicates with the database to get and store information.

**USE CASE DIAGRAM**

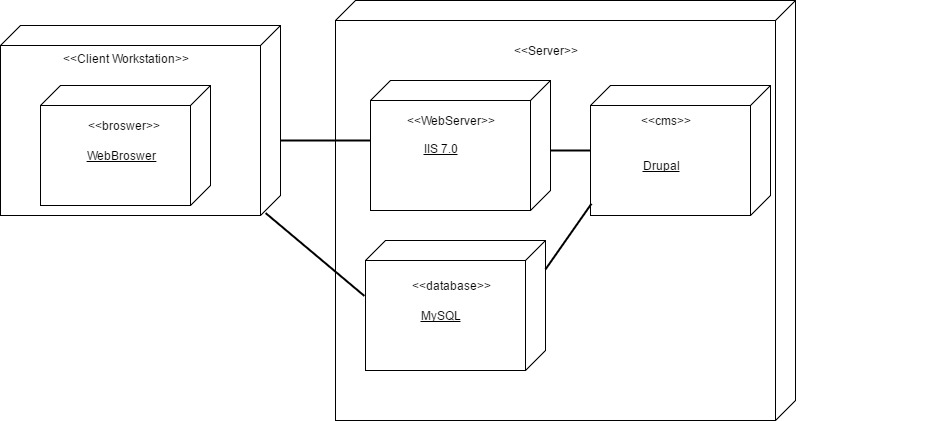


**Sequence Diagram**

Sequence ( Rank Report ).png

Above is the sequence diagram for ranking reports

**Deployment Diagram**



This diagram shows how the components of the system are deployed in hardware.

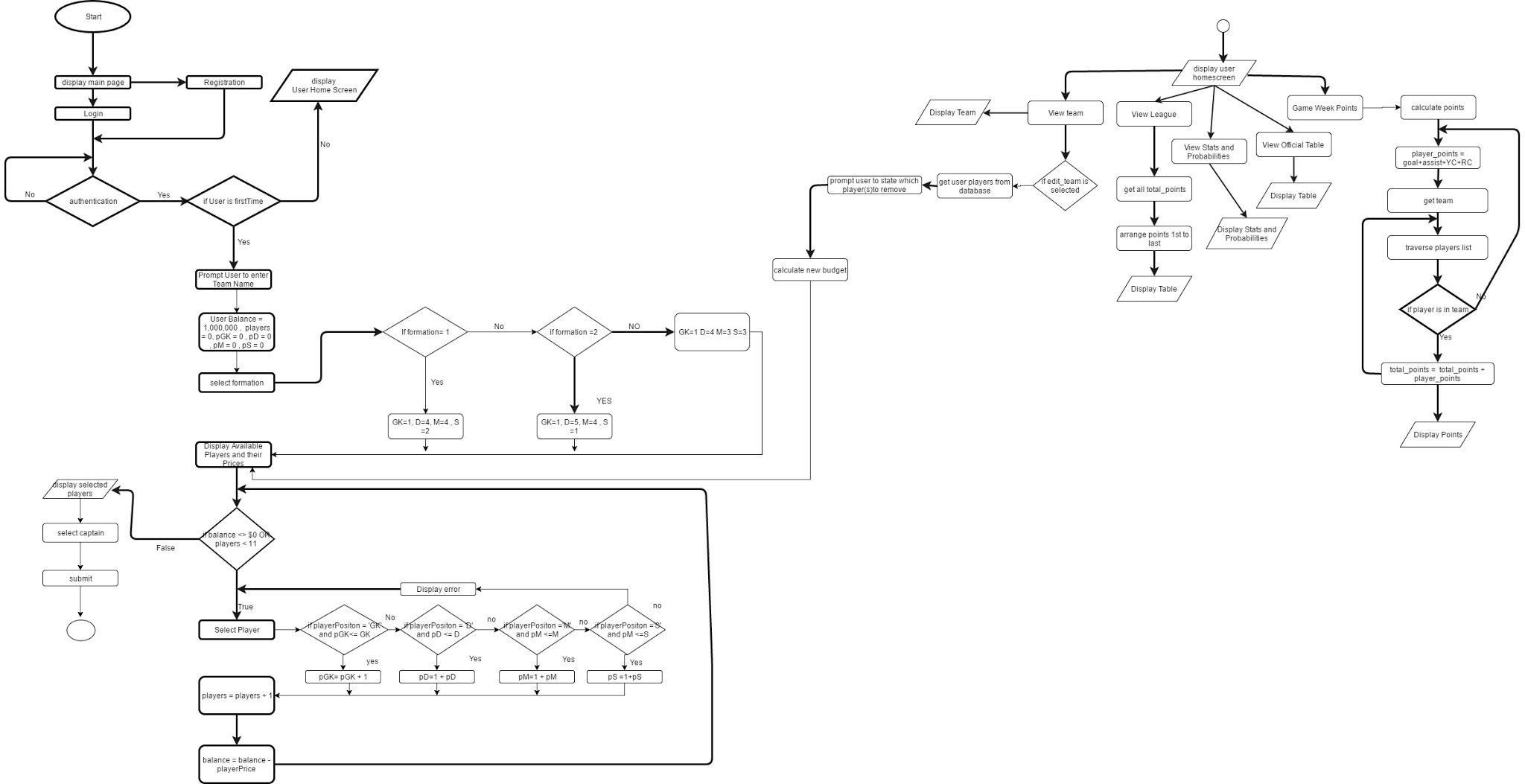
**Activity Diagram** - Calculating User Total Score



**Entity Relationship Diagram**



**Flow Chart**



See Appendix for details on Flow Chart

**Computation**

Players gain points based on goals, assists, appearances and clean sheets and lose points for yellow cards, red cards and own goals. Strikers are most likely to score so the points allocated for a Striker goal is 5 while midfielders, defenders and goalkeepers are allocated 6,7 and 7 points respectively based on likeliness to score. Midfielders are most likely to get an assist so the points allocated for a midfielder assist is 4 while strikers, defenders and goalkeepers are allocated 5, 6 and 6 points respectively, based on likeliness to get an assist. Defenders and goalkeepers only can receive clean sheets which is allocated 4 points. All types of players get a point each once the appear in a match.

-When calculating the points each player receives per match all statistic fields are set to 1 and the formulas below are used based on player position**.**

**Points calculation for Strikers**

Points received = (yellowcard \* -2) + (red card \* -5) + (assist \* 5) + (appearance) + (goal \* 5) + (own goal \* -2)

**Points calculation for Defenders**

points = (yellow card \* -2) + (red card \* -5) + (assist \* 6) + (appearance) + (goal \* 7) + (cleansheet) + (own goal \* -2)

**Points calculation for Midfielders**

points = (yellow card \* -2) + (red card \* -5) + (assist \* 4) + (appearance) + (goal \* 6) + (own goal \* -2)

**Points calculation for Goalkeepers**

points = (yellow card \* -2) + (red card \* -5) + (assist \* 6) + (appearance) + (goal \* 7) + (cleansheet) + (own goal \* -2)

**Note: player\_score = player\_score + points**

##### 

##### 

##### Results

##### Accomplishments

The succeeding sections present a concise overview of The Red Stripe Premier League Application and its capabilities.

RSPA facilitates two users- The Administrator and the actual user of the application. The functions of the system are broken down into the following sections:

* System Access - All users are able to login and logout, that is, they are able to access the system with their respective credentials that direct them to their corresponding pages. Users are allowed to register to join RSPA and are given a balance of $1 million JMD.
* Administrator - The Administrator is the backbone of the system in terms the administrator enters vital statistics about the matches played for a specific week and ensure the full functionality of the system is deployed.
* User - Individuals can create accounts and then create their own teams, naming them whatever they like. Users are allowed to select 1 team with 11 players from a list of all the players participating in the league as well as three bench players. They can select among 3 formations which will dictate how they choose their players. Users can select a certain amount of Goal Keepers, Midfielders, Defenders and Strikers. Users are only allowed to make changes to their teams 24 hours before each match is played. A user is able to view the total scores, weekly scores, position ranks and team members on their profile. They are also allowed to substitute and transfer their player as well as view the a countdown time to the next match and as result the time remaining to substitute and transfer players. They are able to view news and updates and they are able to view information such as scores and league teams on the previous matches played in the League. Users are able to view their rank on the RSPA League table as well as the real Red Stripe Premier League leaderboard table to see persons in first.

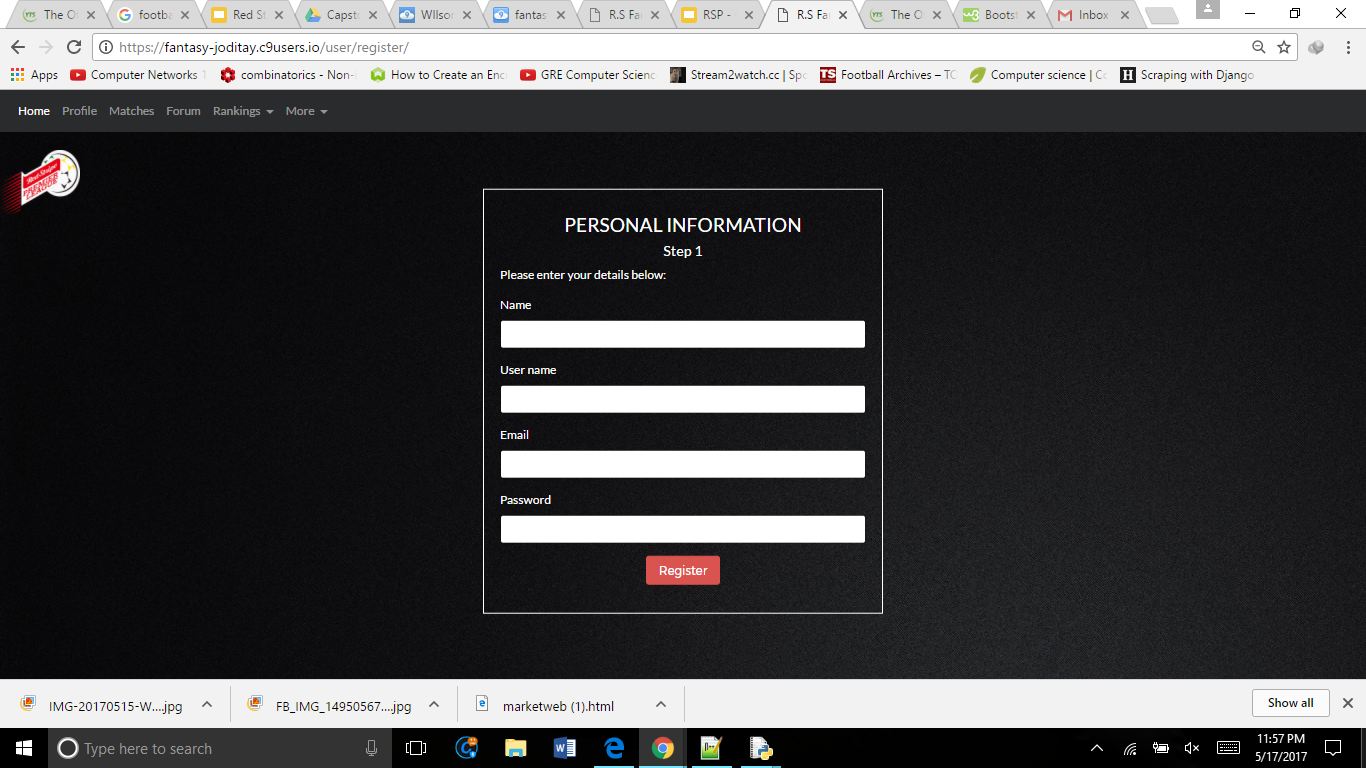
System - The system ranks users as well as updates and computes players scores, update sand computes user total and weekly scores, ranks users depending on highest scores

##### Screenshots

Home Page

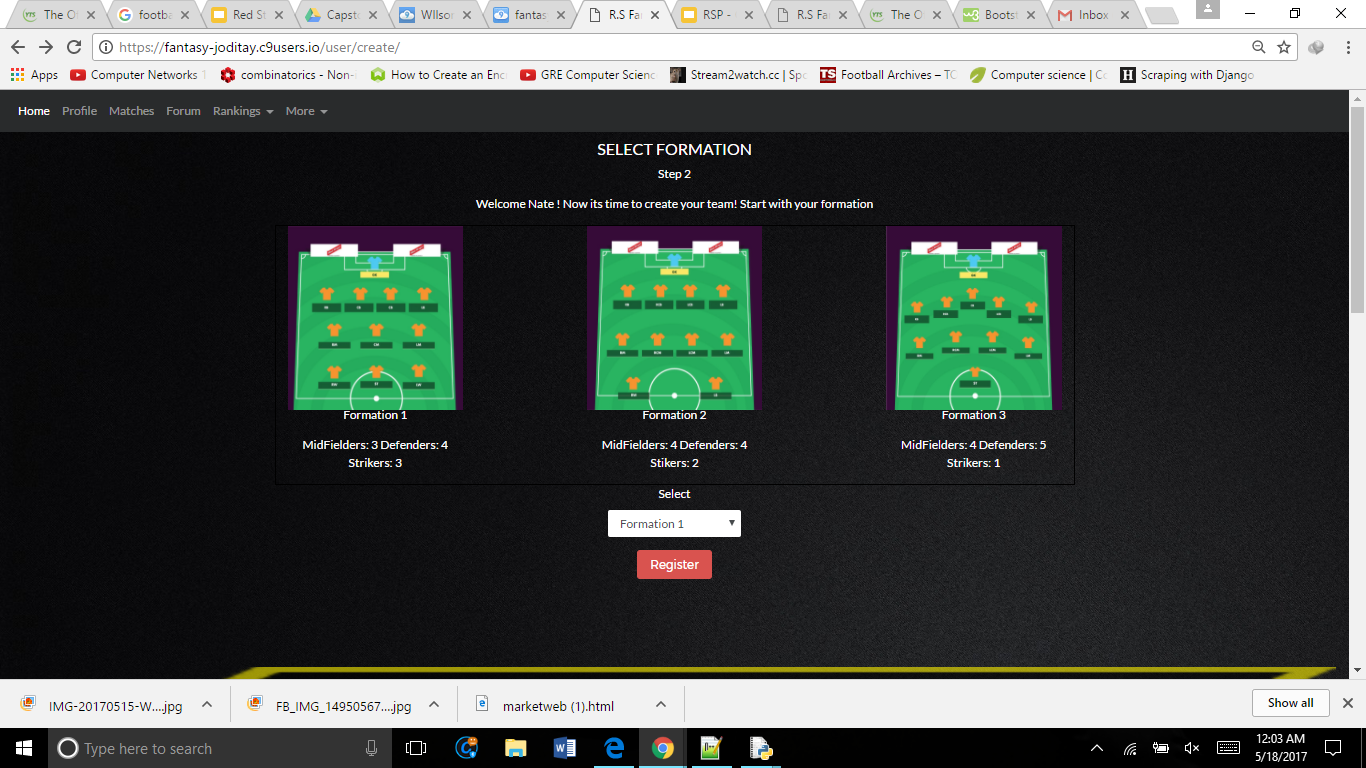
##### 

User Registration



##### User Login

##### 

User Formation selection

##### 

##### 

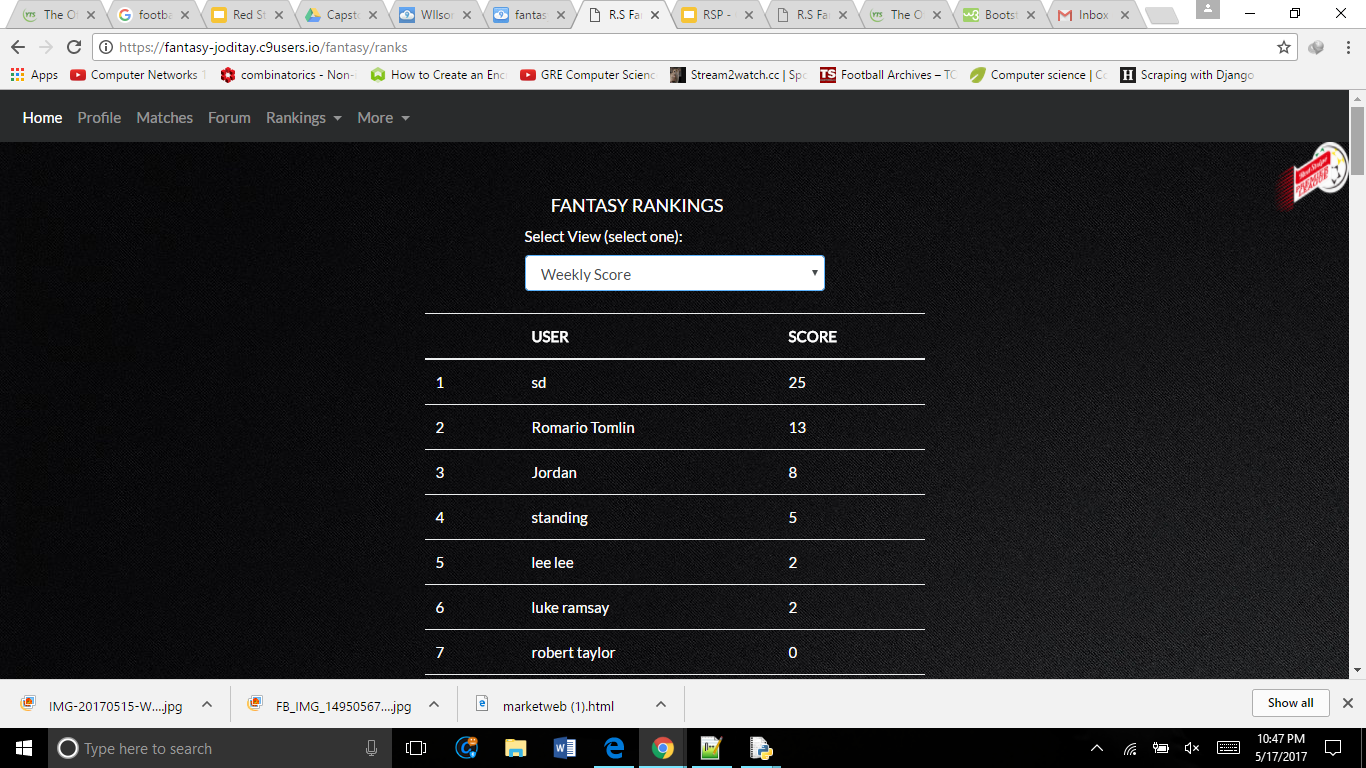
##### Player Selection

##### 

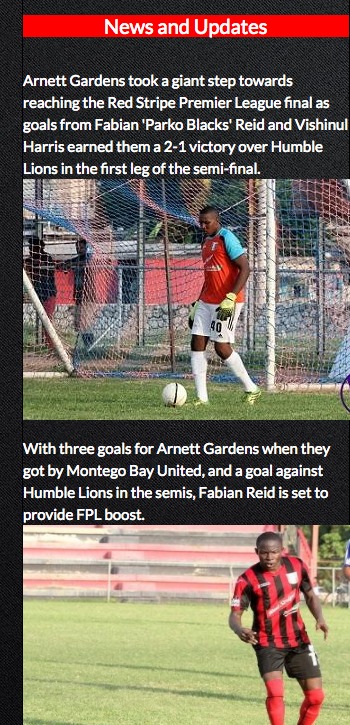
User Profile



Player Rankings



News and Updates



Matches



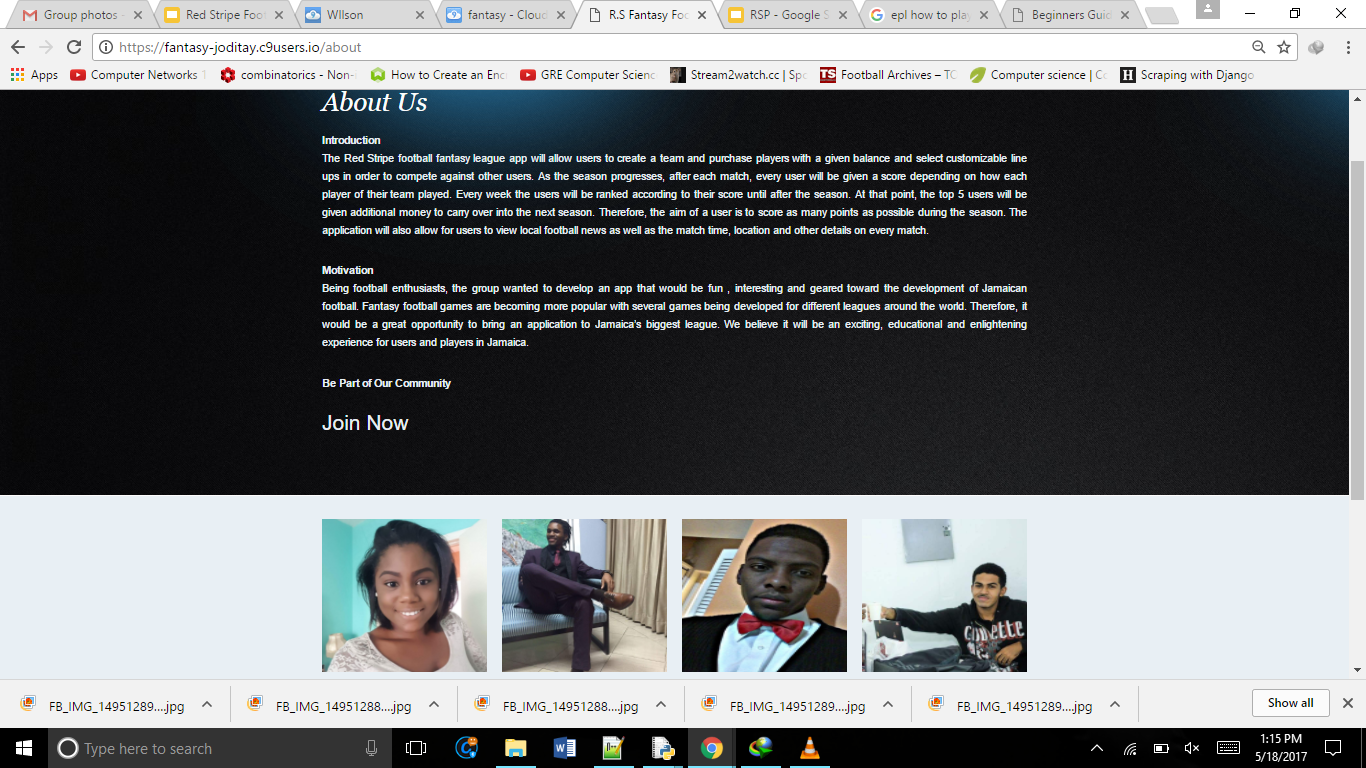
League Table



##### 

##### 

About Us



##### 

##### 

##### Conclusion

##### Problems Encountered and Limitations

Some of the problems encountered with creating this application included finding a source with all required information from the matches which take place in the Red Stripe Premier League. No website was found that displayed several key statistics such as key passes, interceptions and saves. This would be solved by having an administrator input all these statistics. This affected web crawling with django which proved more complicated than expected and required cost in some case.

##### Project Scope That Was Not Accomplished

Due to time constraints there were sections of the project that have yet to be accomplished. These include implementing a Forum, allowing users to interact and communicate and an Administrative Interface that would allow the administrator to input statistical data about the match into the system. Also, getting Live Scores was not able to done due the fact that web scrawling was an issue.

##### Improvements that can be made

##### Several improvements that can be made to the system in the near future include:

##### Allowing users a wider range of football formations to choose from.

##### Allowing users to create their own personal leagues with their friends.

##### Implementing a system which advises users on the best players to choose.

##### Make User interface more user friendly by adding drag and drop players to manipulate movement of players (creating team, transferring players and substituting players)

##### Adding a search block to with various types of searches to find players. (Eg. Find players by name, position, team and price limit)

##### 

##### Conclusion

To conclude, the implementation of the Red Stripe Premier League Fantasy Football Application was an enlightening experience. Our problem was to create the first fantasy football application for a major sports league in Jamaica and this was accomplished.

**Appendix**

**Some Functions to note:**

**Function: Update Total Score - Updates the total score of a user, based on the amount of points they have received thus far.**

def update\_total\_score(user\_id):

user = User.objects.get(id=user\_id)

team = Teams.objects.get(id=user.team\_number)

players\_json = json.loads(team.players)

players = players\_json['players']

temp\_total\_score = 0

for player\_id in players:

player = Players.objects.get(id = player\_id)

temp\_total\_score = temp\_total\_score + player.score

user.total\_score = temp\_total\_score

user.save()

return

**Function: Update Weekly Score - Updates the total score of a user, based on the amount of points they have received for that particular week.**

def update\_weekly\_score(user\_id):

user = User.objects.get(id=user\_id)

team = Teams.objects.get(id=user.team\_number)

players\_json = json.loads(team.players)

players = players\_json['players']

temp\_weekly\_score = 0

for player\_id in players:

player = Players.objects.get(id = player\_id)

temp\_weekly\_score = temp\_weekly\_score + player.weekly\_score

user.weekly\_score = temp\_weekly\_score

user.save()

return

**Function: Update player score**

def update\_players\_score(weekgiven):

matches = Matches.objects.filter( week=weekgiven , status = 'true' , updated = 'false' )

for match in matches:

results = Results.objects.filter(match\_id = match.id)

for result in results:

player = Players.objects.filter(id=result.player\_id)

if player[0].position == 'S':

points = (result.yc \* -2) + (result.rc \* -5) + (result.assist \* 4) + (result.appearance) + (result.goal \* 5) + (result.own\_goal \* -2)

elif player[0].position == 'D':

points = (result.yc \* -2) + (result.rc \* -5) + (result.assist \* 5) + (result.appearance) + (result.goal \* 6) + (result.cleansheet) + (result.own\_goal \* -2)

elif player[0].position == 'M':

points = (result.yc \* -2) + (result.rc \* -5) + (result.assist \* 5) + (result.appearance) + (result.goal \* 5) + (result.own\_goal \* -2)

else:

points = (result.yc \* -2) + (result.rc \* -5) + (result.assist \* 3) + (result.appearance) + (result.goal \* 5) + (result.cleansheet) + (result.own\_goal \* -2)

Players.objects.filter(id=result.player\_id).update(weekly\_score=F('weekly\_score')+ points)

Players.objects.filter(id=result.player\_id).update(score= F('score')+ points)

**Other functions include:**

**Function: Create user**-Creates an account for a new user.

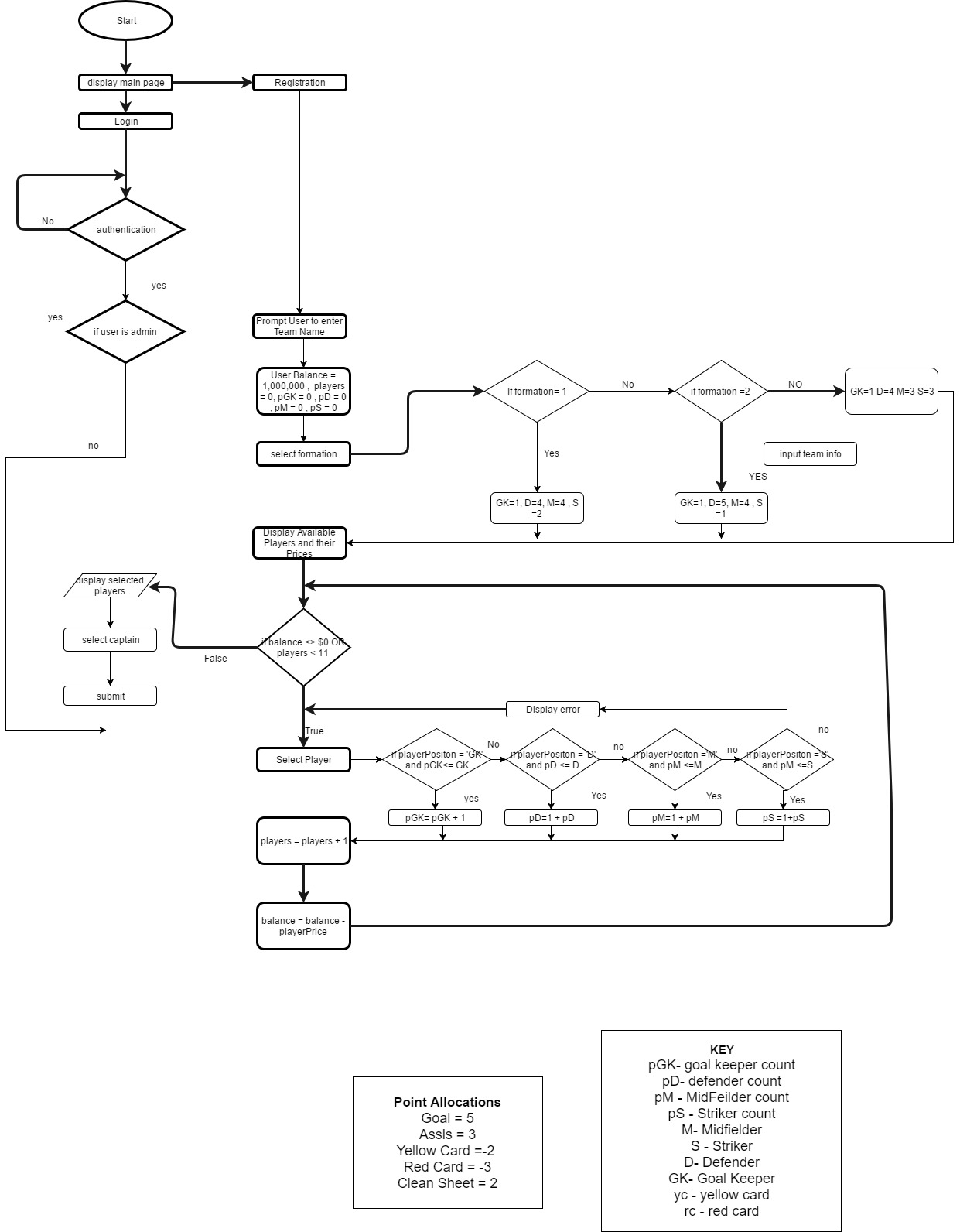
**Function: Select Formation -** Allows users to choose from 3 given formation to choose the amount of players they want playing in each position.

**Function: Add Players -** Allows users to add players to their squad

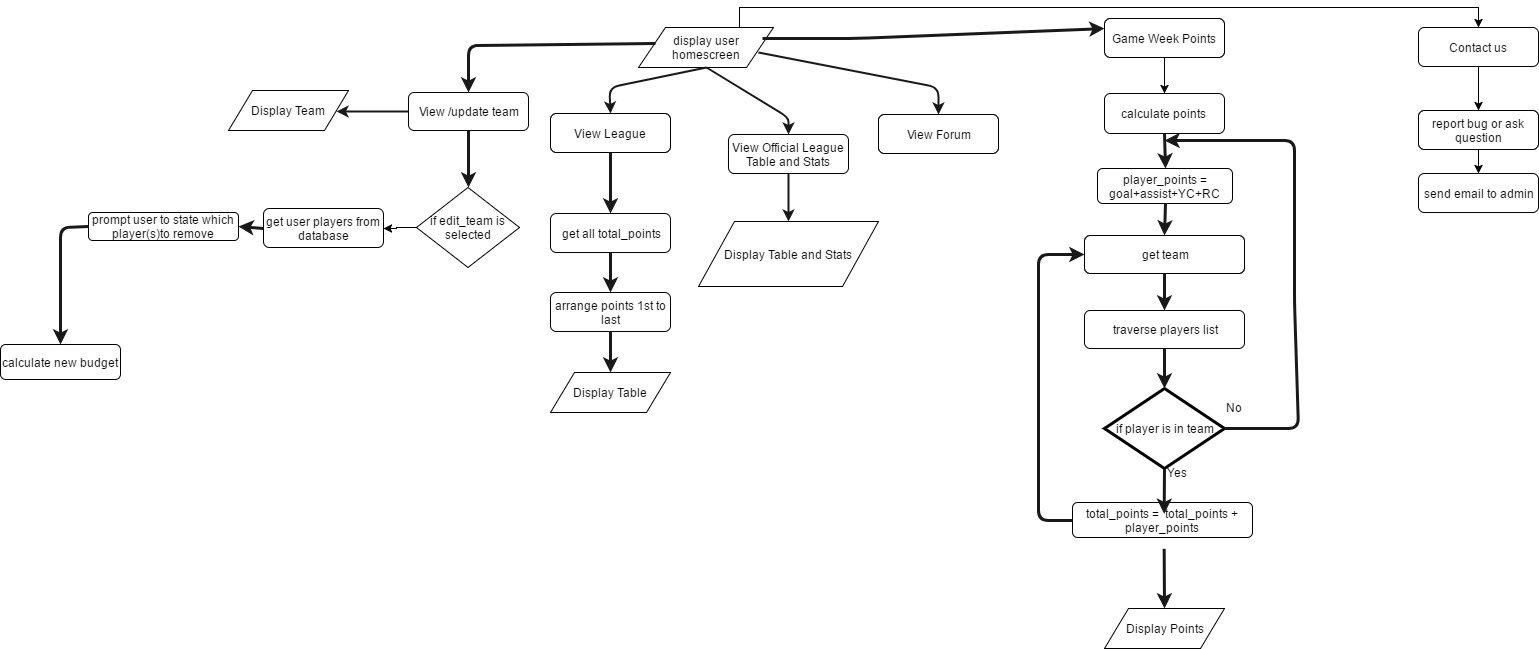
**Function: Remove Players -** Allows users to remove players that have previously been added to their squad.

**Function: Transfer Players -** Allows users to transfer players they have already selected so that they may choose others.

**Function: Substitute Players -** Allows users to switch players between being active and substitutes



FlowChart in details



Flow Chart in details part 2