The Game Point : Tournament Management System

A. N. Herath
Department of Computer Science and Engineering, University of Moratuwa
Sri Lanka
narmada.ah.13@cse.mrt.ac.lk

Abstract - **The paper discusses the development of the web application "The Game Point – Tournament Management System", the need of such an application, the nature of the application, tools and technologies used and the overall project development process. The main goal in designing this system is to computerize the draw making and match scheduling process in badminton - knockout or group round robin tournament while providing a consistent and reliable platform to store and manage player and team information.**

Keywords – Badminton, Knockout draw, Round robin groups, Match schedule, Online entry, Web application

I. INTRODUCTION

"University of Moratuwa International Shuttlers Fest (UMiSF)" is an annual badminton tournament organized by the University Badminton team & the Physical Education Division of the university. The tournament which spans over 10 days' time consists of an inter school tournament for age categories from under 11 to 19 and an inter university tournament. Altogether it consists of 27 events and has a participation of over 1000 players. The objective of the proposed software is to automate the management of player information, draw making and the match scheduling activities of UMiSF. Currently, all activities related to organizing the tournament are carried out manually.

The motivation behind automating the manual system is to increase the efficiency and reliability of the organizing work. The manually carried out process of sorting 1000+ tournament entries and preparing match draws and schedules are an excessive time and effort consuming task for the organizing committee.

Currently all tournament entries are sent via post. Then the player and event details from entries are manually entered in to excel sheets. It has often proved to be an unreliable storage medium. And synchronizing the entries sorted by 30+ organizing committee members has also proven to be a difficult task.

The match draws play a vital role in this highly competitive tournament. Hence possible human errors and shortcomings in the manual process are highly unwelcome. There is also a need of an efficient match scheduling protocol since many categories of matches are played in parallel, so that the players/teams have enough recovery time.

The objective of the proposed online entry system and the computerized draw making and match scheduling process is to provide a reliable alternative for this manual procedure.

Furthermore the implemented draw making and match scheduling algorithms could be extended to create draws and schedules of similar sports such as tennis, squash etc.

The paper is structured as follows. Section II includes the literature review followed by Section III on significant system models. Section IV describes the details of system implementation. Section V is on testing and analysis followed by the conclusion.

II. LITERATURE REVIEW

There exist a number of sports tournament management systems in the internet for assisting organizing activities of an event such as UMiSF.

A major difficulty in using such a system is the need to purchase the software, as its services are not free nor are their algorithms open sourced. 'Tournament software' website [1] provides management systems for several sports including badminton that are available on purchase.

"Badminton England" web application [2] provides a well-structured design to create draws & schedules for singles, doubles & mix double events. The app provides mechanism to add player entries and tournament dates. And it also creates schedules depending on the type of match round: knockout or round robin etc. Here the player draws are created via obtaining the current national rankings of the players. Using national rankings is not feasible methodology for an inter school tournament such as UMISF. In "The Game Point" the system admin can specify a set of previous tournaments and rank players depending on their performances in them. Importance is that these algorithms could be extended for such similar sport tournaments where the admin can specify their own player ranking pointing scheme.

"playpass" web app [3] provides one of the best match scheduling algorithms, complete with interfaces for specifying different details for scheduling, such number of badminton courts, starting time, time for a single match, match round etc. It outputs a complete schedule demonstrating the teams, court numbers and the match times for each.

The above app can be used to create the inter university team tournament schedule. But it does not provide the ability to create match schedules for individual events such as singles, doubles etc.

Similarly to draw making algorithm the "The Game Point" scheduling algorithm for individual & team events are extendable for other similar tournaments and sports events.

III. SYSTEM MODELS

A. *System Requirement*

1) Functional Requirements

The system will consist of three stakeholders. High level Admin, Low level Admin and Participant. Based on stake holders the system is divided in to three subsystems, namely the participant subsystem, Event Coordinator subsystem (Low level admin) & the project chairperson subsystem (High level admin)

Participant subsystem: The major functional requirement of the participant subsystem is to register inter school & inter university entries to the system, via filling the application mentioning the personal information as well as performances in other tournaments.

Event Coordinator Subsystem: The low level admin can enter entries to the system on behalf of players and append performances. They will have low level management over player database.

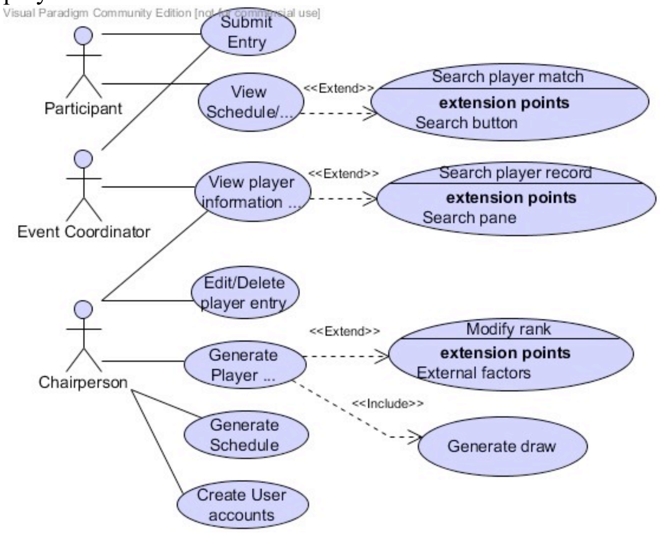


Fig. 1: Use Case Diagram

As shown in Figure 1, project Chairperson Subsystem: High level admin will have a higher level of control over the player database. (E.g.: delete players etc.). The main functional requirements of the system are generating the draws & schedules to which the orders and specifications are given by the high level admin. Also new user sign up requests should be approved by a high level admin who will specify their access levels.

2) Non - functional Requirements

User Friendly GUI: The participants (general users) have low computer literacy hence the web app for the online entry system must be highly user friendly. Simplicity and understandability are important. System Availability: During the time span of the tournament, the application will be accessed by large no. of general users as well as constantly accessed by admins; hence the ability to support multiple access as well as high performance is required. Accuracy: The accuracy and the consistency of handling entries as well the accuracy of algorithms to generate the draws/schedules is a main requirement of the system. Security: The privileges given to different admin levels should be preserved while disallowing any unauthorized access to player database, tournament details etc.

B. *System Design*

This system is designed using MVC architecture. The architecture comprises of model, view and controller, which separates system logic from interfaces and data access sections. The controller communicates between the view and data access, receives requests from the application and prepare the data needed via accessing the database through the data model and finally the view presents the prepared response.

Logical view describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages as shown in Figure 2. The three subsystems of the system are Participant, Event Coordinator and Chairperson Subsystems. Further the system has three main packages Entry package, Draw making package and Schedule making package.

Entry package: This package will support the functionality of registering school and university players to the system. Classes: *Player*, *UniTeam* (Registering players and university teams, have method to insert player's/team's past performances). *School* (Set of schools & their information). *AdminLow* (Methods to manage the player database).

Draw making package: This will support the creation of draws for each event. Classes: *EventEntry* (Contains individual entries of each player for each event) *EventDraw* (The combination of methods contains the main logic for creating the draw) *AdminHigh* (have the methods for giving the specifications for creating draws).

Schedule Making Package: This will support the creating of the match schedule for each event. Classes: *Match* (Contains details of matches in each round) *EventSchedule* (Contains the main logic for creating the schedule) *AdminHigh* (Give specifications for scheduling)

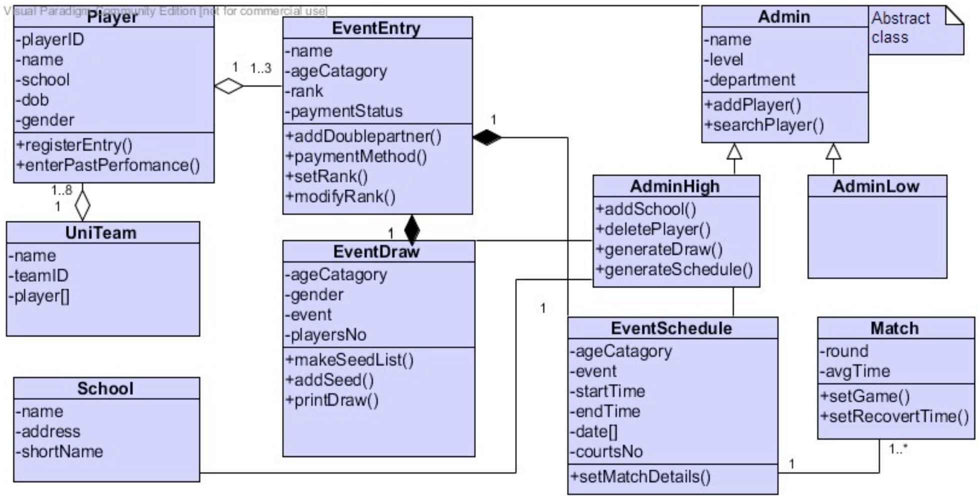


Fig. 2: UML Class Diagram

The sequence diagrams of the two main processes of generating the draw and schedule are shown in Figure 3. Admin high will give specifications step by step for creating the draw/schedule to which the system will function, validate requirements and respond whether the request can be fulfilled. Finally the system will generate the draw/ schedule accordingly.

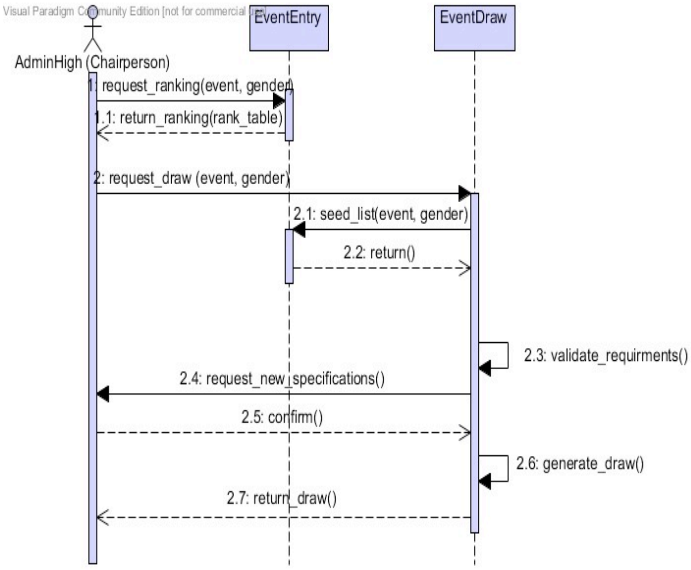

Visual Paradigm Community Edition [not for commercial use]
AdminHigh (Chairperson)
EventEntry
EventDraw
1: request_ranking(event, gender)
1.1: return_ranking(rank_table)
2: request_draw (event, gender)
2.1: seed_list(event, gender)
2.2: return()
2.3: validate_requirments()
2.4: request_new_specifications()
2.5: confirm()
2.6: generate_draw()
2.7: return_draw()

Fig. 3: Generating draw

The database view of the system is shown in Figure 4. Third normal form is used in normalizing the database design to reduce the duplication of data and ensure referential integrity by ensuring that the entity is in second normal form, and all the attributes in a table are determined only by the candidate keys of that table and not by any non-prime attributes. It was implemented using phpMyAdmin. It was ensured that redundancy was avoided while maintaining dependency preservation

.

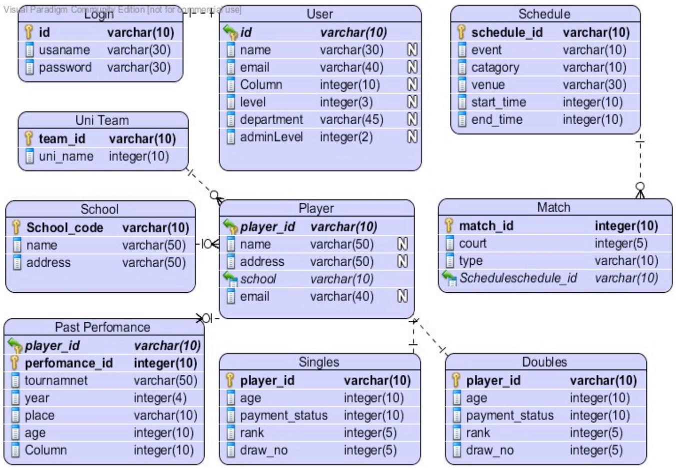


Fig. 4: Database view of the system

IV. System Implementation

A. *Implementation Procedure*

The system used several technologies. The database was implemented and designed using phpMyAdmin. Several programming languages were used in the system. The web interfaces were designed from the scratch using html, css and several available bootstrap libraries. Source codes for special design effects such as image sliders were obtained from online available resources. The project was mainly coded using php. The MVC architecture was implemented by separately defining php classes and files for Modal, Controller and Template directories for each of the three packages. The system consists of taking a series of commands from the user and feting data from the database, processing them and displaying them. JavaScript with AJAX were widely used throughout the implementation for interactive web designing. Several open source libraries such as JSPDF was also used in the system. The source code developed for this system was maintained in a configuration management tool. The source code is available on GitHub.

Several important strategies were followed during the requirement gathering phase of the system. The manual procedure of draw making was closely studied while the organizing committee was preparing draws. The pointing scheme for ranking was designed in collaboration with the organizing committee. Past draws and schedules were analyzed and studied during the requirement gathering phase. The project was implemented following the project schedule under rational unified process in close collaboration with the stakeholders.

The system logic consists mainly of different algorithms to generate outputs necessary for draw making and match scheduling. Following are the algorithms to generate a player rank in a particular event (e.g.: under 11- singles) utilizing the past performance-pointing scheme. To generate the players draw number depending on rank, total number of players, total seeding positions and other rules [5].

```
Player Ranking Algorithm - Singles

For each player in list
       Find  if  exist  past  performances: singles
              For each performance in
              performance list
                     Generate score from
                     pointing scheme
       Get overall player points
       Store points to database
Calculate player rank by points
```

Fig. 5: Player ranking algorithm

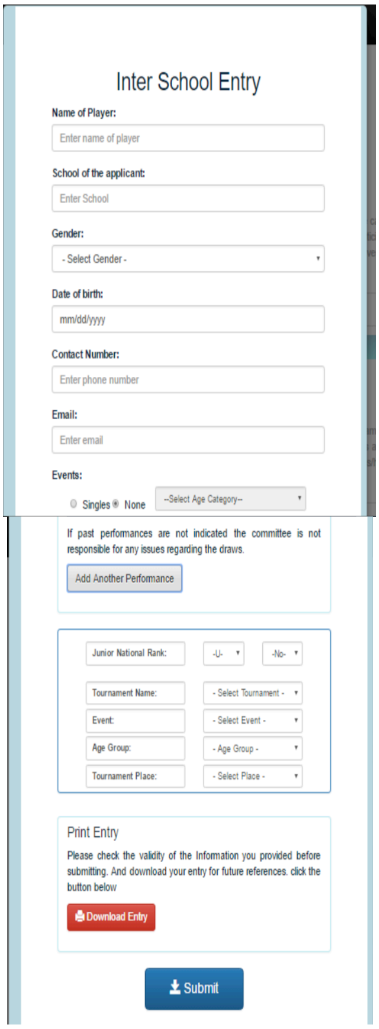
```
Draw Making Algorithm - Singles (Generate Draw Number)
Player_list_size // no of players
Draw_size // size of draw - calculated by another algorithm
rankNo[] // seeding positions - calculated by another algorithm
byeList[] // byes positions - calculated by another algorithm
random = 1;

For each in player list{
   If player.rank !=0 {//rank:player ranking algo
       Player.drawno = rankNo[i]
       takenList.add (drawno)
   }
   Else if (random <= Draw_size) {
       If (random is in takenList){
              Random++
       }
       Else if (random is in byeList){
              Random ++
       }
       Else {
              Player.drawno = random;
       }
       takenList.add (random)
       }
   }

write drawno to database
```

Fig. 6: Draw making algorithm

B. Main Interfaces

Participant interfaces: *Submit entry* – This interface contains a form with fields for player information, playing events (e.g.: singles/doubles/mix doubles), junior national ranking and other past performances. (Figure 5)

Admin interfaces: *University Team interface* – teams are ranked depending on the past team performance - point scheme (Figure 6). And final group draw is created, by assigning teams to the 4 groups according to the round robin group algorithm (Figure 7).



Inter School Entry

Name of Player: Enter name of player

School of the applicant: Enter School

Gender: - Select Gender -

Date of birth: mm/dd/yyyy

Contact Number: Enter phone number

Email: Enter email

Events:

○ Singles ● None --Select Age Category--

If past perfomances are not indicated the committee is not responsible for any issues regarding the draws.

Add Another Perfomance

Junior National Rank: -U- -No-

Tournament Name: - Select Tournament -

Event: - Select Event -

Age Group: - Age Group -

Tournament Place: - Select Place -

Print Entry

Please check the validity of the Infomation you provided before submitting. And download your entry for future references. click the button below

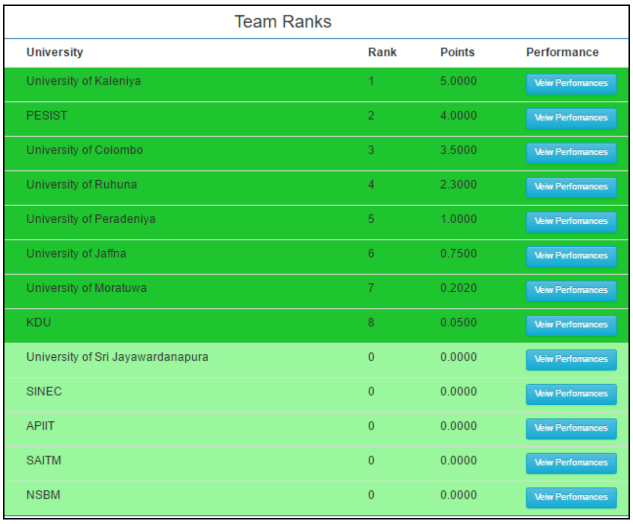
Download Entry

Submit

Fig. 7: Inter School events – entry

Individual event interface - individual event (singles, doubles, mix doubles) draws are created by seeding ranked players (Player ranking obtained via the past performances in relevant events) and by following other rules [5].

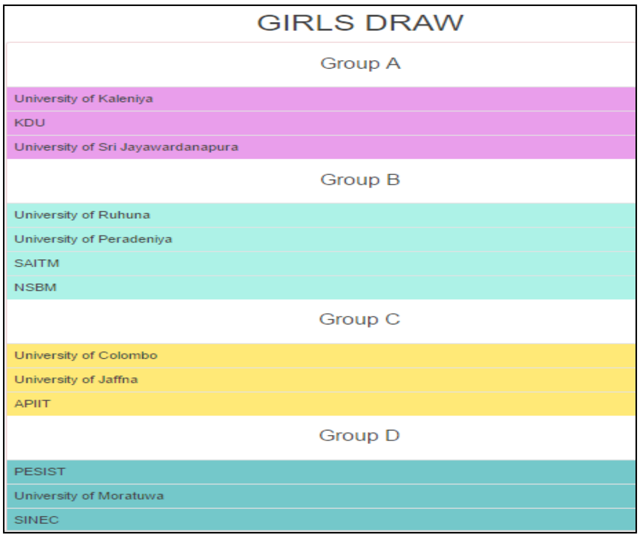
Schedule Interface – Match schedule for an event is created considering no. of courts, no. of matches, match round, recovery time and other specifications (Figure 8).



Team Ranks

University	Rank	Points	Performance
University of Kaleniya	1	5.0000	Veiw Perfomances
PESIST	2	4.0000	Veiw Perfomances
University of Colombo	3	3.5000	Veiw Perfomances
University of Ruhuna	4	2.3000	Veiw Perfomances
University of Peradeniya	5	1.0000	Veiw Perfomances
University of Jaffna	6	0.7500	Veiw Perfomances
University of Moratuwa	7	0.2020	Veiw Perfomances
KDU	8	0.0500	Veiw Perfomances
University of Sri Jayawardanapura	0	0.0000	Veiw Perfomances
SINEC	0	0.0000	Veiw Perfomances
APIIT	0	0.0000	Veiw Perfomances
SAITM	0	0.0000	Veiw Perfomances
NSBM	0	0.0000	Veiw Perfomances

Fig. 8: View Team points (past performance – pointing scheme) & Team ranks in order



GIRLS DRAW

Group A

- University of Kaleniya
- KDU
- University of Sri Jayawardanapura

Group B

- University of Ruhuna
- University of Peradeniya
- SAITM
- NSBM

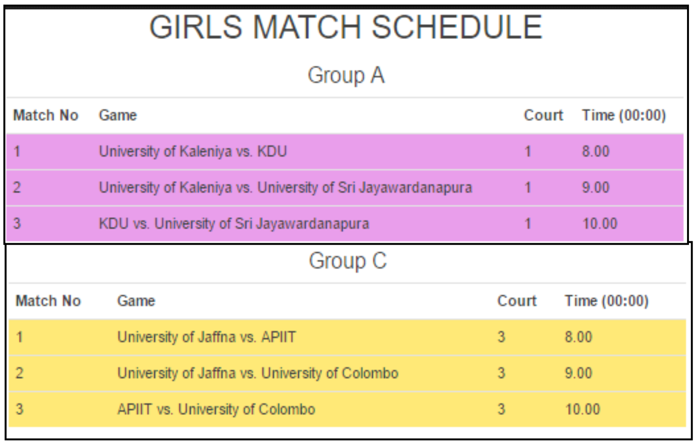
Group C

- University of Colombo
- University of Jaffna
- APIIT

Group D

- PESIST
- University of Moratuwa
- SINEC

Fig. 9: Generate round robin group – draw



GIRLS MATCH SCHEDULE

Group A

Match No	Game	Court	Time (00:00)
1	University of Kaleniya vs. KDU	1	8.00
2	University of Kaleniya vs. University of Sri Jayawardanapura	1	9.00
3	KDU vs. University of Sri Jayawardanapura	1	10.00

Group C

Match No	Game	Court	Time (00:00)
1	University of Jaffna vs. APIIT	3	8.00
2	University of Jaffna vs. University of Colombo	3	9.00
3	APIIT vs. University of Colombo	3	10.00

Fig. 10: University girls - Round robin group match schedule

V. System Testing and Analysis

Data and Database Integrity Testing: Database was tested to ensure that the data has been populated as intended and all database events such as edit, delete and insert have occurred properly. Each database access method was invoked and verified. Prevention of incorrect/ invalid data access was also tested.

Function Testing: Testing features/functionality of the system covering cover all the scenarios, including failure paths and boundary cases. Black box testing technique is followed, that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output. User interface testing: Selenium IDE in Mozilla Firefox was used to test interfaces.

Security and Access Control Testing: The system was tested for application-level security, including access to the data and business functions and for System-level Security, including logging into and remotely accessing to the system.

Automated tests are written to cover all the segments of code. Unit tests were written using php unit. The tests can be executed on the deployed or development environment by enabling the Testing module. Testing results were analyzed for validation & verification.

VI. Conclusion and Future Work

The automated tournament management system is able to facilitate the proceedings of the current manual system with less time and effort for the organizing committee and also with more reliability. Submitting online entries will be an easier task for the participants. And also the player information storing directly to the player database will be a valuable alternative as it prevents errors such as misplace of entries and human errors when entering player information from entries. The database is a reliable alternative to the current file storage system. The organizing committee can view all the player information sorted by underage, event, gender etc. The organizing committee can edit/ modify player information if a requirement arises.

The main functionality of the system is the computerized draw and schedule making process. The generated point scheme for ranking players is tested to be accurate using real data from entries. The draw making process utilizes the set of rules and constraints defined in the “Badminton Handbook” [5].The computerized algorithm are far more accurate, unbiased and efficient than the manual procedure. Similarly the scheduling algorithm is also high in reliability and effectiveness.

Future development for the project includes getting the system ready for deployment and implementing a payment gateway to pay tournament entry fees online. The system will be deployed in two phases, in the first phase it will run in parallel with the manual system and in the next phase it will completely replace the manual system. Further the algorithms will be open sourced to be used for any similar sport tournaments.

VII. References

[1] "tournament software," [Online]. Available: https://www.tournamentsoftware.com/product/demo.aspx?id=15&s=2..

[2] "Badminton England," [Online]. Available: http://www.badmintonengland.co.uk/text.asp?section=1227#.Vt3Dffl97IV..

[3] "playpass," [Online]. Available: https://playpass.com/schedule.

[4] "selenium," [Online]. Available: http://www.seleniumhq.org/download/.

[5] W. B. federation, BWF HANDBOOK II (Laws of Badminton & Regulations), 2010-2011.

[6] "UMiSF," [Online]. Available: http://badmintonmora.org/.

[7] "Model View Controller(MVC) in PHP," [Online]. Available: http://php-html.net/tutorials/model-view-controller-in-php/.

[8] "Generating PDFs from Web Pages on the Fly with jsPDF," [Online]. Available: https://www.sitepoint.com/generating-pdfs-from-web-pages-on-the-fly-with-jspdf/.

[9] "Round Robin Tournament," [Online]. Available: https://en.wikipedia.org/wiki/Round-robin_tournament.

[10] "jQuery.ajax()," [Online]. Available: http://api.jquery.com/jquery.ajax/