APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 7 No. 1 (2026) 669-686
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Blackened Futsal Tournament Management System

Muhamad Akmal Irfan Lokman[1], Mohd Zainuri Saringat[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: zainuri@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.036

Article Info

Received: 15 June 2025
Accepted: 21 June 2026
Available online: 30 June 2026

Keywords

Futsal Tournament, Scheduling, Result Tracking, Progressive Web App (PWA), Centralised Platform, Management System, Bracket Management, Match Management

Abstract

Futsal tournaments have gained popularity, yet many still rely on manual processes, such as pen and paper or spreadsheet-based systems that lead to inefficiencies and organizational issues. The primary issue is the absence of an integrated platform to run tournaments, which causes delays, chaos, and dissatisfaction among organisers, players, and fans. This project aims to develop the Blackened Futsal Tournament Management System to automate and streamline tasks such as scheduling, score tracking, and team management. The system was developed using the Prototyping Model and built with web technologies including HTML, CSS, JavaScript, PHP, and MySQL, with Stripe API integration for payments. Results show improved scheduling accuracy, real-time score updates, and enhanced communication among stakeholders. Findings suggest that digital automation significantly reduces administrative workload and improves the overall user experience. Future improvements may include mobile-native integration and broader support for multi-stage tournaments.

1. Introduction

The Blackened Futsal Tournament Management System is a systematic process for planning, documenting, and evaluating futsal tournament-related data, such as schedules, team registration, match results, and score and rankings. The system's functionality, adaptability, and ease of use can help with the tiresome information management tasks during the games, lessen the event planners' workload, and increase productivity and quality of work [1]. The futsal tournament management at Blackened Futsal, which is situated in Batu Pahat, Johor, depends on manual procedures and simple equipment like paper and pen, which guarantees that this approach would waste time. This procedure requires careful record-keeping and repetitive procedures, which could result in errors and inefficiencies [2]. Inefficient data retrieval and a higher chance of errors result from the use of paper-based records and disorganized digital documents. Scheduling conflicts, delays, and data inconsistencies can emerge from tournament organizers' frequent struggles with scheduling, selecting which team will play, and monitoring results.

Three problems were taken into consideration before the Blackened Futsal Tournament Management System was developed in order to ensure that it would address all of the problems and provide solutions. Firstly, match scheduling and grouping are done manually, which increases the risk of error and irate participants. For instance, a match may take place twice due to a management error or incorrect information that causes player misunderstanding. Secondly, the absence of instantaneous communication between the tournament management and participants may lead to participant misconceptions. For instance, players are unaware of real-time information on match scheduling, results, and updates during the competition. Thirdly, when the match is over, the management will manually record the result, which will take time because they have to store the result securely so that the previous winner may move on to the next round. Additionally, this method may be prone to errors, such as losing the match result.

Therefore, the objective of the project includes designing a Blackened Futsal Tournament Management System using an object-oriented approach, developing the system using Progressive Web App (PWA) approach, and testing the system functionality using alpha and beta testing. The proposed system will implement a total of ten modules which are login, register, participant management, score and standings management, match management, tournament setup, scheduling, bracket management, tournament information, and report. This project aims to organize and oversee sporting events online, ensuring their success and facilitating smooth management from beginning to end. This project gives significant to the management by streamlining scheduling and organization, making the schedule match easier and faster, reducing human error that can result in inaccurate results, and saving money and time.

2. Related Work

In this section, the domain background of Blackened Futsal Tournament Management System, study of Blackened Futsal manual process, and the comparison with the existing systems are discussed.

2.1 Domain Background

A famous form of football that has become quite popular all around the world is called futsal. It can be difficult to organise and manage futsal tournaments, particularly big events. Ineffective manual procedures can result in errors, delays, and annoyance for teams, fans, organisers, and players. The Blackened Futsal Tournament Management System is crucial for optimising tournament administration and improving the overall experience, as it facilitates the futsal tournament with simplicity. The system will offer features like schedule generation, team registration management, match results, and real-time score and ranking updates. The Blackened Futsal Tournament Management System is accessible from any device connected to the internet using an HTTP browser and an active internet connection. Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript, and Hypertext Preprocessor (PHP) are some of the tools used in the development of this system. This system uses MySQL as a database to hold the data.

2.2 Study of Blackened Futsal Manual Process

Blackened Futsal continues conducting futsal tournaments by hand, such as with pen and paper. As part of the management team's strategy for running the futsal tournaments, all of the documentation related to the tournaments will be kept in a folder. The Blackened Futsal administrator will write down all the information needed on paper to set up a futsal tournament, including the date, time, format, rules, and prizes. Using all the information, the administrator will make a poster and post it on social media. Through phone calls or the Whatsapp messaging app, the administrator is able to obtain participation details, including team and player names. The participants' information is written down and kept in a file. Online banking is used by the participants to pay the entry fees, which are then entered into the logbook. The administrator assigns teams at random to make match schedules and list them on paper once registration is closed. In the course of the tournament, the administrator manually keeps track of the match scores on a scoreboard and records the results on paper. Participants can observe the knockout bracket displayed on a board. The administrator will gather all the files, including match schedules, results, and participant information, when the tournament is over and put them in a folder for future use. The goal of the study is to determine what isn't functioning well, such as taking too long or creating issues. It seeks to offer suggestions on how Blackened Futsal might improve tournament management, making things more seamless and effective.

2.3 Comparison with the Existing Systems

This section aims to illustrate the differences between the three existing systems and the proposed Blackened Futsal Tournament Management System. All of the features listed in Table 1 will serve as a guide for all systems, allowing them to determine which ones offer or satisfy all of the requirements. Table 1 shows the comparison between the existing systems and the proposed system.

Table 1 *System's Comparison*

Login	Provide account login	Provide account login	No	Allow to log in securely
Register	Allow users to create accounts	Allow users to create accounts	No	Allow administrator to create an account
Participant Management	Handle participant registration and allow register independently	Handle participant registration and allow register independently	Handle participant registration and allow register independently	Allow participants to sign up for the tournaments
Score and Standings Management	Automatically create standings and manage score after match end	Manage the score at the end of the match.	Automatically create standings and manage score after match end	Provide a real-time score and standings update
Match Management	No	No	No	Manage match by entering score and statistics
Tournament Setup	Set up a tournament	Set up for knockout round only.	No	Set up a tournament
Scheduling	Schedule matches automatically	No	Schedule matches automatically	Schedule matches automatically
Bracket Management	Handle bracket for knockout	Manage the knockout tournament's bracket.	Handle bracket for knockout	Manage the knockout tournament bracket.
Tournament Information	No	No	Save the previous tournament details	Record information on previous tournaments
Report	No	No	Produce a report	Generate reports

3. Methodology

The Blackened Futsal Tournament Management System was developed using the Prototyping Model methodology. The Prototyping Model is a software development process that aims to better understand user requirements by developing an early approximation of a system or component, called a prototype. Model prototyping is a method for rapidly obtaining specific data regarding user information requirements [3]. Prototypes facilitate communication between developers and users throughout the manufacturing process, enabling developers to more readily model the software that will be produced [4].

There are seven iterative phases in the prototyping model. Requirement gathering and analysis is the first step by asking the users to describe their expectations and desires from the system. The design phase, which comes next, will create a fundamental system design that reflects important features and workflows. Rapid design, a translation process based on user requirements or preferences, and a design that is presented as an intuitive interface plan [5]. The next step is to build a prototype, which will include all the necessary features to illustrate the main features of the system. In the following phase, known as user evaluation, users will test the prototype and provide feedback on its features, design, and usability, highlighting its strengths and shortcomings. The user evaluated the prototype application and then used it to build applications that use it as a

production end product [6]. The next step is to refine the prototype, which will be done in response to user feedback in order to meet user expectations. This lowers the possibility of expensive modifications and rework later on in the project, which saves money and makes the development process run more smoothly [7]. Implementation is the following phase, during which the prototype will serve as a guide for creating the actual system. The final phase is testing, during which the system will be examined to ensure that it is completely operational, meets user needs, and operates as anticipated. The prototype model works best with customised software or systems, which are created in response to particular needs and demands (or even circumstances or conditions) [8]. Fig. 1 shows the phases involved in prototyping model.

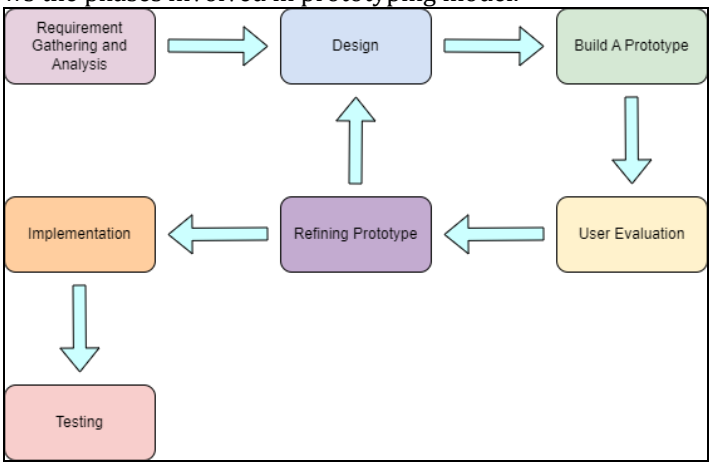


Fig. 1 *Prototyping Model*

3.1 System Development Workflow

There are total of seven phases from the prototyping model. As shown in Table 2, each phase has its own task and output that need to produce during the entire project development.

Table 2 *System Development Workflow*

Phase	Task	Output
Requirement Gathering and Analysis	• Identify stakeholders and collect information through interviews. • Develop use case diagram. • Documenting the detailed requirements in Software Requirements Specification (SRS) document.	• Use case diagram. • Software Requirements Specification (SRS) document.
Design	• Define the overall structure of the system. • Create the database schema. • Develop the user interface design. • Design the individual components or modules of the system. • Review and validate designs	• Use case diagram • Activity diagram • Sequence diagram • Wireframes or Mockups design • Database schema
Build A Prototype	• Develop a prototype • Integrate basic functionality in the prototype • Demonstrate prototype to stakeholder	• Initial prototype
User Evaluation	• Organize testing sessions with stakeholder • Gather feedback through surveys, interviews, and observation • Record feedback in a document	• Stakeholder evaluation report • Usability findings • Stakeholder feedback documentations
Refining Prototype	• Analyze the documented feedback • Modify the prototype based on stakeholder feedback • Add new features or improve existing features	• Refined prototype • Documentation of changes

Table 2 *System Development Workflow (cont)*

	• Retest prototype to ensure the changes meet stakeholder expectations • Documenting all modifications made to the prototype	
Implementation	• Code development • Integrate all components and subsystems • Prepare for the deployment of the system	• Completed system • System's database • Codebase and source files
Testing	• Test system's functionalities based on the test case • Debugging system's errors • Gather feedback forms from stakeholder • Alpha testing • Test result analysis and reporting	• Test case • Error and bug fix • Feedback forms • Alpha testing reports

4. Analysis and Design

The analysis and design of the Blackened Futsal Tournament Management System are described in this section. The functional and non-functional requirements of the proposed system will be covered in section 4.1. The user requirements analysis of the proposed system will be covered in section 4.2. The use case diagram that illustrates how users interact with the system will be covered in section 4.3. The domain class diagram, which shows a system's structure, will be covered in section 4.4. Section 4.5 will go over the proposed system's interface design.

4.1 Functional and Non-Functional Requirements

Functional requirements define what a system must be able to perform and how it should act in particular scenarios. A system's features must be implemented by the developer so that users can accomplish their objectives. Table 3 lists the functional requirements for every Blackened Futsal Tournament Management System module.

Table 3 *Functional Requirements*

Modules	Functionalities
Login	• The system should allow user to login using user identification (ID) and password. • The system should verify user credentials for access to the system.
Register	• The system should allow administrator to create a new account by providing necessary details.
Participant Management	• The system should allow player to join the tournament. • The system should allow player to insert necessary details for registration. • The system should allow player to pay registration fee if needed. • The system should allow administrator to view the list of participants of the tournament.
Score and Standings Management	• The system should allow player to view the match score and standings of the tournament. • The system should allow viewer to view the match score and standings of the tournament.
Match Management	• The system should allow administrator to input score for each match. • The system should allow administrator to record match statistics, including goalscorer, player substitution, and yellow or red cards.
Tournament Setup	• The system should allow administrator to create a new tournament. • The system should allow administrator to edit tournament details. • The system should allow administrator to delete tournament.

Table 3 *Functional Requirements (cont)*

Scheduling	• The system should automatically generate match schedules based on the number of teams. • The system should allow administrator to edit match details if needed. • The system should allow player to view the match schedule for upcoming matches. • The system should allow viewer to view the match schedule.
Bracket Management	• The system should generate tournament bracket for knockout tournament. • The system should allow administrator to manage a bracket for knockout tournament. • The system should allow player to view their team for the knockout tournament. • The system should allow viewer to view the bracket for the knockout tournament.
Tournament Information	• The system should display general information about the tournament. • The system should allow administrator to review past tournaments, including details such as tournament formats, number of participants, and match results.
Report	• The system should allow administrator to generate tournament report. • The system should allow administrator to export the report as a PDF file.

Non-functional requirements are a collection of specifications that outline the constraints and operational capabilities of the system. Additionally, it defines a system's quality rather than its functionality. The non-functional requirements for the Blackened Futsal Tournament Management System are displayed in Table 4.

Table 4 *Non-Functional Requirements*

Requirements	Description
Operational	• The mobile devices and desktop computer should be operated with internet availability. • The system should be able to read from the database and write to the equivalent database.
Performance	• The interaction between the system and user should not be longer than 5 seconds.
Security	• The system can be access only by authenticated administrator with valid user ID and password.
Availability	• The system can be operated and available 24 hours a day.

4.2 User Requirement Analysis

The process of determining what users anticipate from a new or modified system is known as user requirements analysis. It entails engaging with users and stakeholders face-to-face to find out about their needs, problems, and preferred features. This guarantees that the final result enhances the user's experience and is in line with their goals. Table 5 lists the user requirements for the Blackened Futsal Tournament Management System.

Table 5 *User Requirements*

No	User Requirements
1.	Administrator should be able to access the system using authorized account.
2.	Administrator should be able to create a new account.
3.	Administrator should be able to manage match by entering score and statistics.
4.	Administrator should be able to create a new tournament.
5.	Administrator should be able to generate match schedule for each team.
6.	Administrator should be able to generate and handle bracket for knockout phase.
7.	Administrator should be able to keep track of the tournament information, including match results and tournament schedules.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 669-686

Table 5 *User Requirements (cont)*

8.	Administrator should be able to access and generate comprehensive reports for the tournament and convert to PDF file.
9.	Player should be able to join for tournaments by entering details and pay registration fee.
10.	Player should be able to access real-time updates on match results, standings, tournament schedules, and knockout bracket.
11.	Viewer should be able to access real-time updates on match results, standings, tournament schedules, and knockout bracket.

4.3 Use Case Diagram

A use case diagram is a visual representation that shows how users interact with a system. By showing the different ways, users can interact with the system, a use case diagram offers a high-level overview of its capabilities. The actors in a use case diagram stand in for users, the use cases for modules or system functionality, and the system boundary for the system's limits. Three actors and 10 use cases are depicted in the use case diagram of the proposed system. Fig. 2 shows the Blackened Futsal Tournament Management System use case diagram.

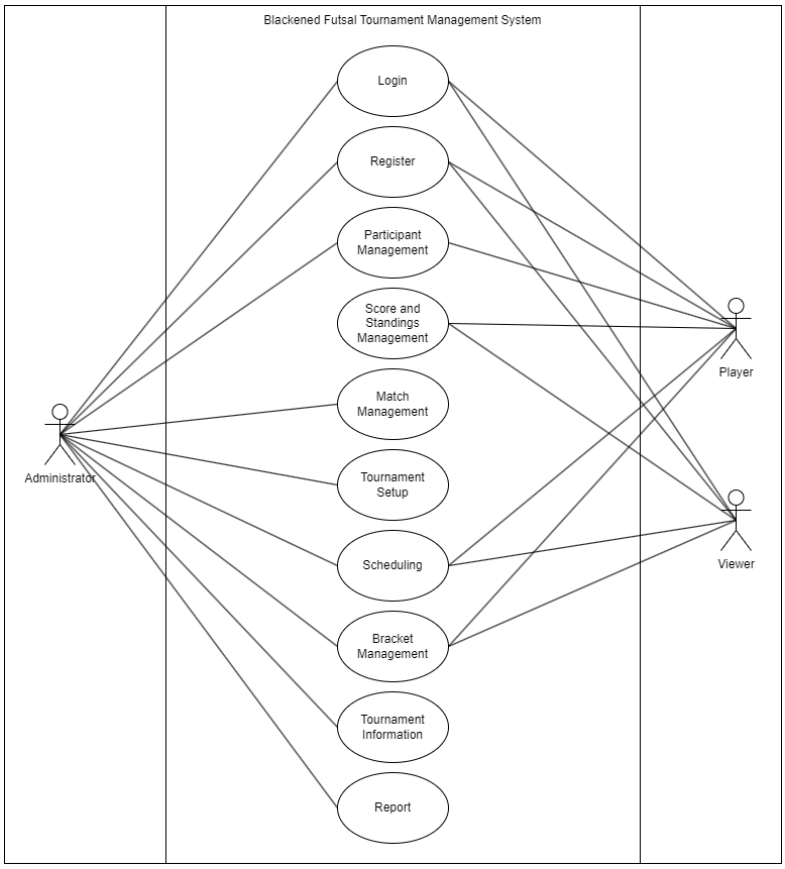


Fig. 2 *Use case diagram*

4.4 Domain Class Diagram

A class diagram will assist in representing the structure of a system by displaying its classes, characteristics, methods, and the links between them. Fig. 3 shows the class diagram of the proposed system.

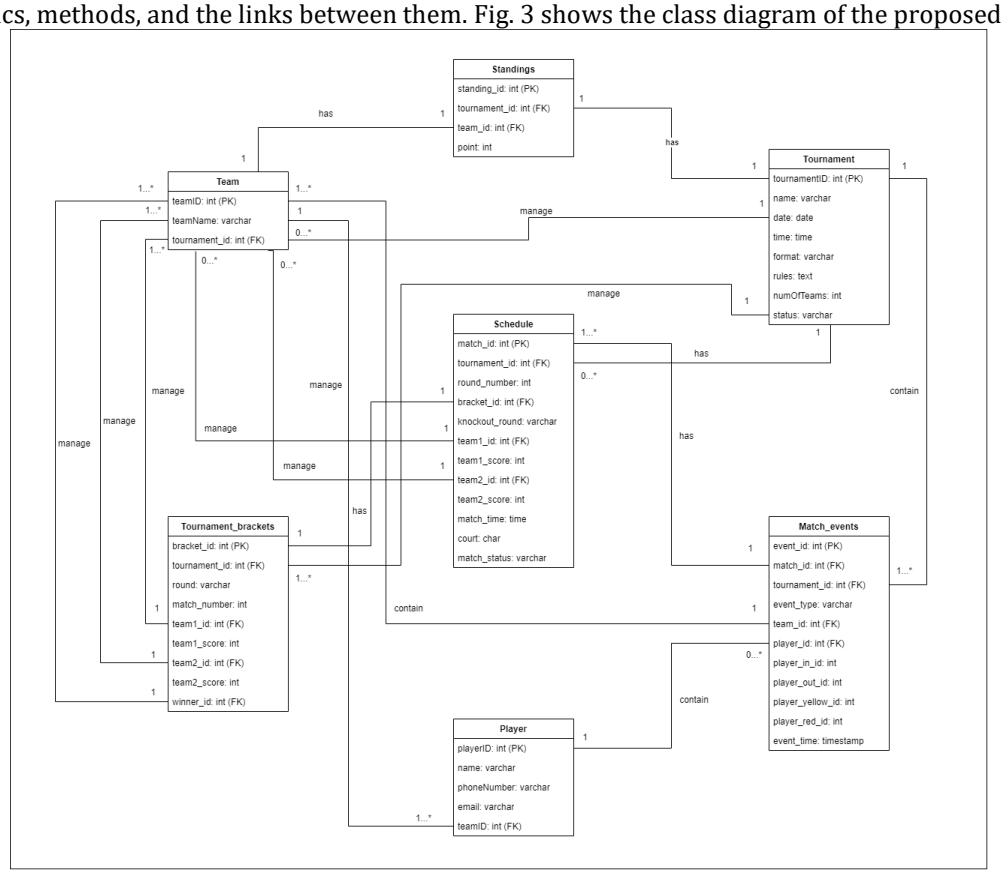
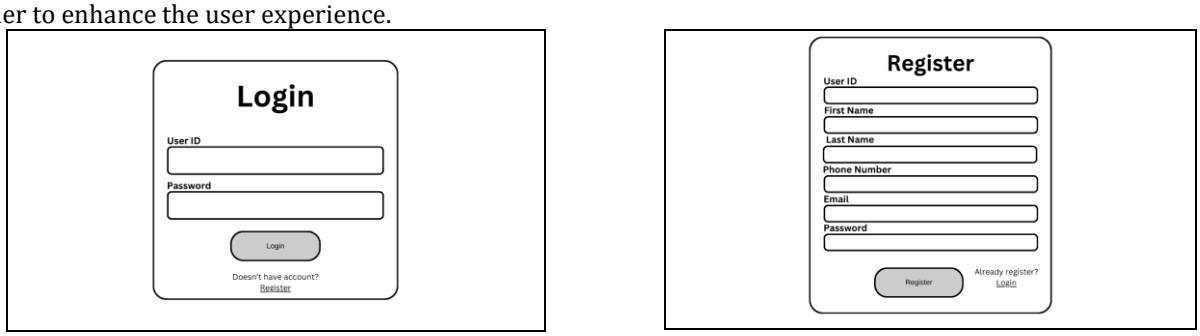


Fig. 3 *Class Diagram*

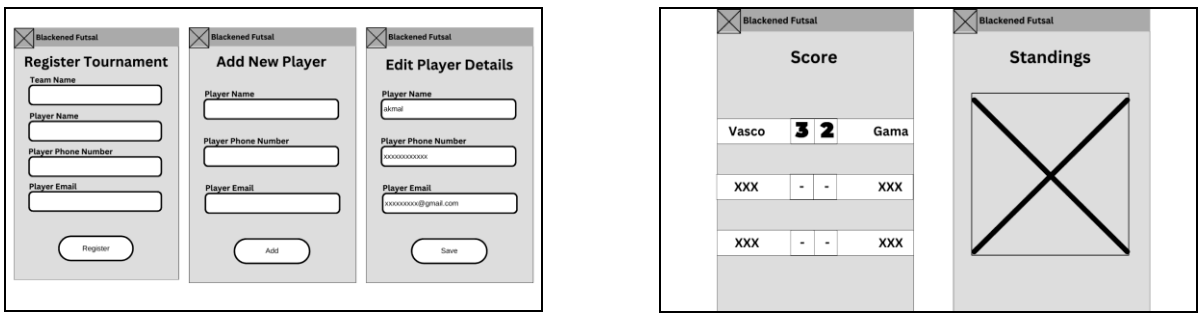
4.5 Interface Design

Interface design is the process of developing the interactive and visual elements of a product's interface in order to enhance the user experience.



(a) **(b)**

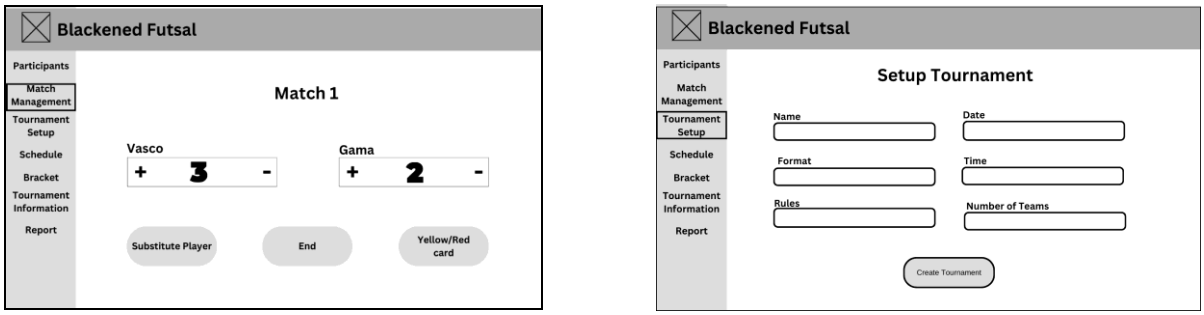
Fig. 4 (a) *Login Interface;* **(b)** *Register Interface*



(a) **(b)**

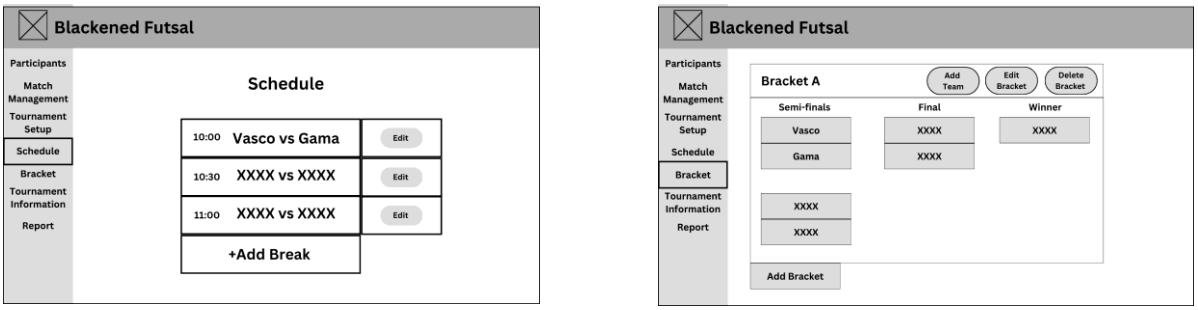
Fig. 5 (a) *Participant Management Interface;* **(b)** *Score and Standings Management Interface*

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 669-686



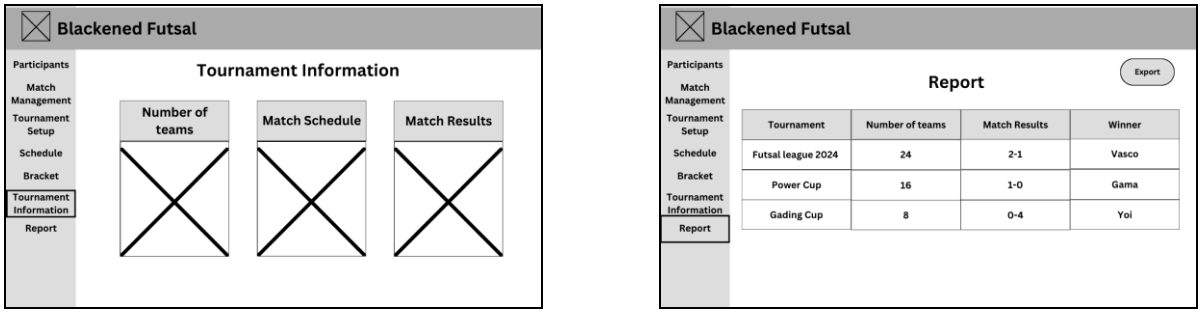
(a) **(b)**

Fig. 6 (a) *Match Management Interface;* **(b)** *Tournament Setup Interface*



(a) **(b)**

Fig. 7 (a) *Scheduling Interface;* **(b)** *Bracket Management Interface*



(a) **(b)**

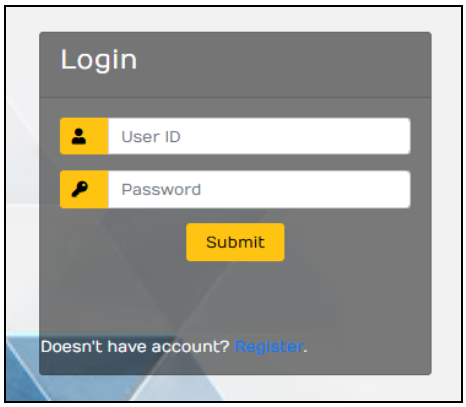
Fig. 8 (a) *Tournament Information Interface;* **(b)** *Report Interface*

5. Implementation and Testing

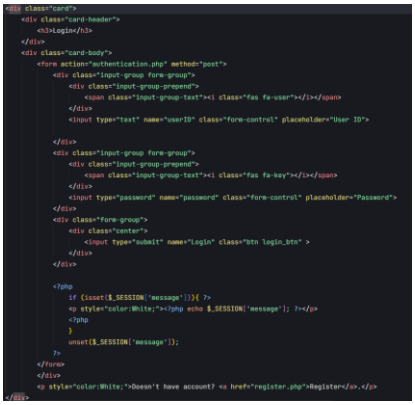
This section describes the Blackened Futsal Tournament Management System's implementation and testing. Section 5.1 will address the code implementation for the proposed system to ensure that it can connect to the database and display output. Section 5.2 will address the test case for evaluating the proposed system's functionality. Section 5.3 will discuss the test case's overall result. Section 5.4 will discuss user acceptance testing, which allows end users to test the proposed system.

5.1 Implementation

Each module's source code and user interface will be described in this section to provide a quick overview of its capabilities and to illustrate how each module will be shown. While HTML, CSS, and JavaScript will be used for the front end to create interactive and responsive interfaces, PHP will be used for the backend to create the system, allowing it to add, remove, and modify data in the database.

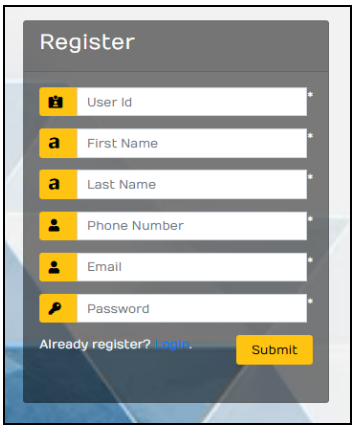


(a)



(b)

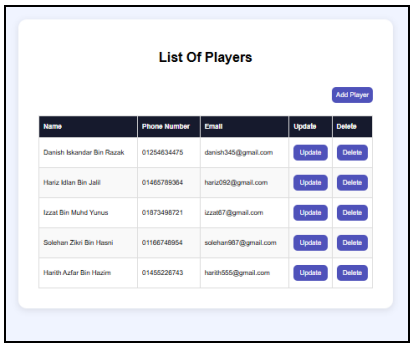
Fig. 9 (a) *Login Interface;* **(b)** *Login Code Segment*



(a)

(b)

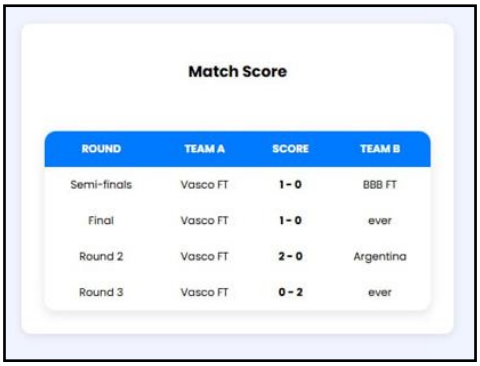
Fig. 10 (a) *Register Interface;* **(b)** *Register Code Segment*



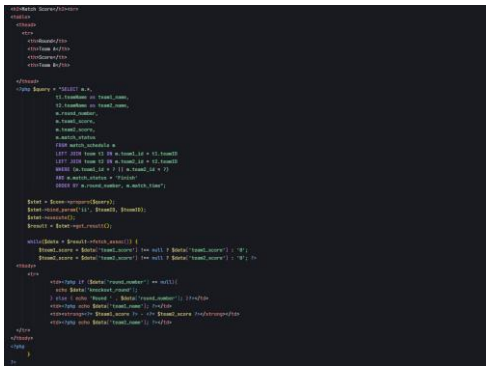
(a)

(b)

Fig. 11 (a) *Participant Management Interface;* **(b)** *Participant Management Code Segment*



(a)



(b)

Fig. 12 (a) *Match Score Interface;* **(b)** *Match Score Code Segment*

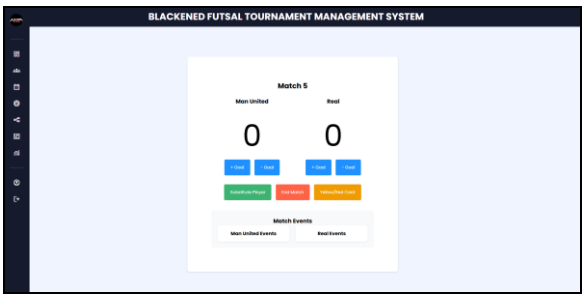
Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 669-686

(a)

```php
<h2>Standings</h2><br>
<table>
  <thead>
    <tr>
      <th>POS.</th>
      <th>Team</th>
      <th style="text-align: center;">Points</th>
  </thead>
  <tbody>
    <?php
    $i = 1;
    $standings = mysqli_query($conn, "SELECT
    t.teamName AS team_name,
    me.point AS point_count
    FROM standings me
    JOIN team t ON me.team_id = t.teamID
    WHERE me.tournament_id = '$tournamentID'
    ORDER BY point_count DESC;
    ");
    while($event = mysqli_fetch_array($standings)) {
      ?>
        <tr>
                <td><?php echo $i;?></td>
                <td><?php echo $event['team_name'];?></td>
                <td style="text-align: center;"><?php if ($event['point_count'] == null){
                    echo 0;
                }else{
                    echo $event['point_count'];
                }?></td>
    </tr>
    <?php
    $i++;
      }

    ?>
```

(b)

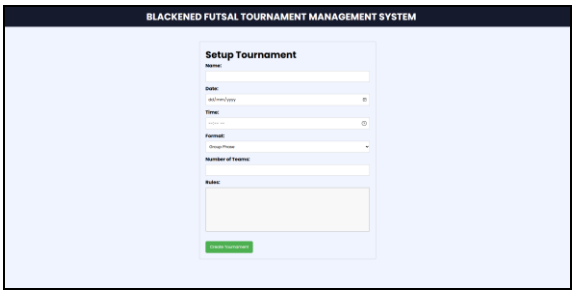
Fig. 13 (a) *Standings Interface;* **(b)** *Standings Code Segment*



(a)

(b)

Fig. 14 (a) *Match Management Interface;* **(b)** *Match Management Code Segment*

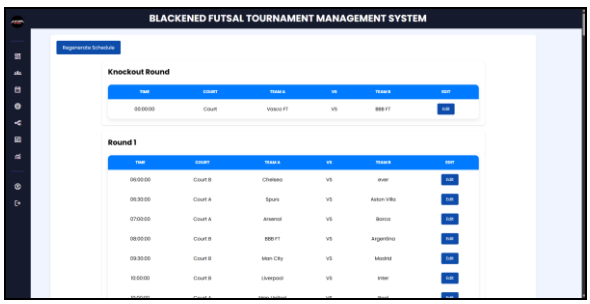


(a)

```php
<?php
session_start();
include "connection.php";


if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $name = $_POST['name'];
    $date = $_POST['date'];
    $time = $_POST['time'];
    $format = $_POST['format'];
    $teams = $_POST['teams'];
    $rules = $_POST['rules'];
}

$sql = "INSERT INTO tournament VALUES ('', '$name', '$date', '$time', '$format', '$rules', '$teams')";


$conn->query($sql);
header('location:list_tournament.php');

?>
```

(b)

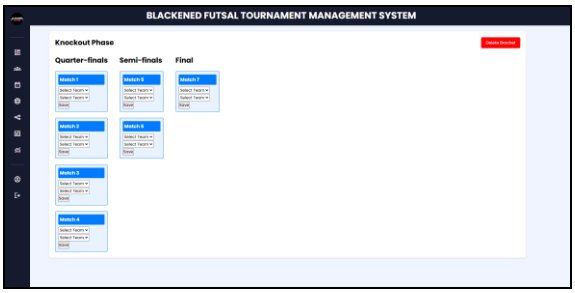
Fig. 15 (a) *Tournament Setup Interface;* **(b)** *Tournament Setup Code Segment*



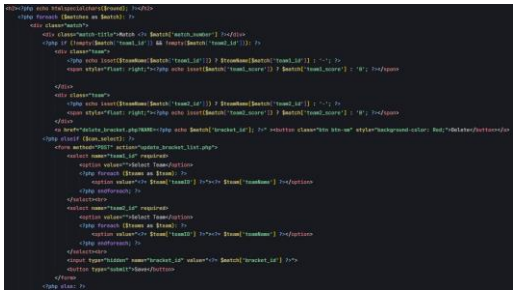
(a)

(b)

Fig. 16 (a) *Scheduling Interface;* **(b)** *Scheduling Code Segment*

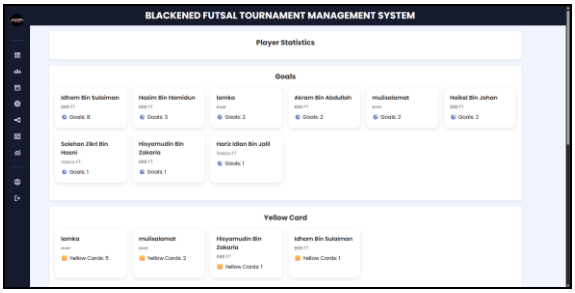


(a)



(b)

Fig. 17 (a) *Bracket Management Interface;* **(b)** *Bracket Management Code Segment*



(a)

(b)

Fig. 18 (a) *Tournament Information Interface;* **(b)** *Tournament Information Code Segment*

(a)

(b)

Fig. 19 (a) *Report Interface;* **(b)** *Report Code Segment*

5.2 Test Case

The purpose of system testing is to verify that the developed system satisfies both functional and non-functional requirements. This requirement is crucial for the system because it allows to identify conditions in which it has failed to perform specific tasks. All modules must perform properly and in accordance with the defined functional and non-functional requirements for system testing to be considered completed. The detailed test case and the results are presented in Table 6 until Table 15.

Table 6 *Test Case for Login*

Test Cases ID	Requirements ID	Description	Status
TC_100	**REQ_100**	**Login**	
TC_100_01	REQ_101	The system should allow user to login using user identification (ID) and password.	PASS
TC_100_02	REQ_102	The system should verify user credentials for access to the system.	PASS
TC_100_03	REQ_103	The system should display error message if the login details is invalid.	PASS
TC_100_04	REQ_104	The system should be access using any web browser.	PASS
TC_100_05	REQ_105	The system should display the homepage of the system in less than 5 second after the user enter correct user identification (ID) and password.	PASS

Table 7 *Test Case for Register*

Test Cases ID	Requirements ID	Description	Status
TC_200	**REQ_200**	**Register**	
TC_200_01	REQ_201	The system should allow user to create a new account by providing necessary details.	PASS
TC_200_02	REQ_202	The system should save the registration details into the database.	PASS
TC_200_03	REQ_203	The system should display error message of empty field detected.	PASS
TC_200_04	REQ_204	The system should display error message of password that less than 6 characters.	PASS

Table 8 *Test Case for Participant Management*

Test Cases ID	Requirements ID	Description	Status
TC_300	**REQ_300**	**Participant Management**	
TC_300_01	REQ_301	The system should allow administrator to view the list of participants of the tournament.	PASS
TC_300_02	REQ_302	The system should allow player to join the tournament.	PASS
TC_300_03	REQ_303	The system should allow player to insert necessary details for registration.	PASS
TC_300_04	REQ_304	The system should allow player to add new player in their team.	PASS
TC_300_05	REQ_305	The system should allow player to edit the player details in their team.	PASS
TC_300_06	REQ_306	The system should allow player to remove selected player in their team.	PASS
TC_300_07	REQ_307	The system should allow player to pay registration fee if needed.	PASS
TC_300_08	REQ_308	The system should display error message if the payment process fails.	PASS
TC_300_09	REQ_309	The system should update list of players after every operation in less than 3 seconds.	PASS

Table 9 *Test Case for Score and Standings Management*

Test Cases ID	Requirements ID	Description	Status
TC_400	**REQ_400**	**Score and Standings Management**	
TC_400_01	REQ_401	The system should allow player to view the match score and standings of the tournament.	PASS
TC_400_02	REQ_402	The system should allow viewer to view the match score and standings of the tournament.	PASS
TC_400_03	REQ_403	The system should automatically update match score and match statistics.	PASS
TC_400_04	REQ_404	The system should automatically update standings based on match results.	PASS
TC_400_05	REQ_405	The system should automatically update match score and standings in less than 3 seconds.	PASS

Table 10 *Test Case for Match Management*

Test Cases ID	Requirements ID	Description	Status
TC_500	**REQ_500**	**Match Management**	
TC_500_01	REQ_501	The system should allow administrator to input score for each match.	PASS
TC_500_02	REQ_502	The system should allow administrators to input goalscorer for each match.	PASS
TC_500_03	REQ_503	The system should allow administrators to make player substitutions for each match.	PASS
TC_500_04	REQ_504	The system should allow administrators to input yellow or red cards for each match.	PASS

Table 10 *Test Case for Match Management (cont)*

TC_500_05	REQ_505	The system should display error messages if the match score is invalid.	FAIL
TC_500_06	REQ_506	The system should allow the administrator to record match results after the match end.	PASS
TC_500_07	REQ_507	The system should store match details securely in the database.	PASS

Table 11 *Test Case for Tournament Setup*

Test Cases ID	Requirements ID	Description	Status
TC_600	**REQ_600**	**Tournament Setup**	
TC_600_01	REQ_601	The system should allow administrators to create a new tournament.	PASS
TC_600_02	REQ_602	The system should allow the administrator to set the number of teams for the tournament.	PASS
TC_600_03	REQ_603	The system should allow administrators to choose the format for the new tournament.	PASS
TC_600_04	REQ_604	The system should display error messages if the required field is empty.	PASS
TC_600_05	REQ_605	The system should save the tournament details securely in the database.	PASS
TC_600_06	REQ_606	The system should allow administrators to edit tournament details.	PASS
TC_600_07	REQ_607	The system should allow administrators to delete tournament.	PASS

Table 12 *Test Case for Scheduling*

Test Cases ID	Requirements ID	Description	Status
TC_700	**REQ_700**	**Scheduling**	
TC_700_01	REQ_701	The system should automatically generate match schedules based on the number of teams.	PASS
TC_700_02	REQ_702	The system should allow the administrator to edit match details if needed.	PASS
TC_700_03	REQ_703	The system should allow the administrator to add break in the match schedule.	FAIL
TC_700_04	REQ_704	The system should allow the administrator to assign court futsal for each match.	PASS
TC_700_05	REQ_705	The system should prevent schedule conflicts from happen.	PASS
TC_700_06	REQ_706	The system should allow players to view the match schedule for upcoming matches.	PASS
TC_700_07	REQ_707	The system should allow viewers to view the match schedule.	PASS

Table 13 *Test Case for Bracket Management*

Test Cases ID	Requirements ID	Description	Status
TC_800	**REQ_800**	**Bracket Management**	
TC_800_01	REQ_801	The system should generate tournament brackets for knockout tournament.	PASS
TC_800_02	REQ_802	The system should allow administrators to manage a bracket for knockout tournament.	PASS
TC_800_03	REQ_803	The system should allow the administrator to add team into the bracket.	PASS
TC_800_04	REQ_804	The system should update brackets automatically as matches conclude.	PASS
TC_800_05	REQ_805	The system should allow players to view their team for the knockout tournament.	PASS
TC_800_06	REQ_806	The system should allow viewers to view the bracket for the knockout tournament.	PASS
TC_800_07	REQ_807	The system should display error messages if match schedule is missing.	PASS

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 669-686

Table 14 *Test Case for Tournament Information*

Test Cases ID	Requirements ID	Description	Status
TC_900	**REQ_900**	**Tournament Information**	
TC_900_01	REQ_901	The system should display general information about the tournament.	PASS
TC_900_02	REQ_902	The system should allow administrators to review past tournaments, including details such as tournament formats, number of participants, and match results.	PASS
TC_900_03	REQ_903	The system should allow administrators to filter tournament information.	FAIL

Table 15 *Test Case for Report*

Test Cases ID	Requirements ID	Description	Status
TC_1000	**REQ_1000**	**Report**	
TC_1000_01	REQ_1001	The system should allow administrators to generate tournament report.	PASS
TC_1000_02	REQ_1002	The system should allow the administrator to export the report as a PDF file.	PASS
TC_1000_03	REQ_1003	The system should display the report in less than 3 seconds.	PASS

5.3 Overall Test Cases Result

The summary of the system testing results will be presented and discussed in this section to determine the number of test cases that passed and failed the system's testing. In order to test the system and determine which test cases pass and which fail, 57 test cases total across 10 modules have been compiled. The test is passed by 54 test cases and failed by 3 test instances. The system has met all requirements for every test case, according to the system testing results. The entire outcome of system testing is displayed in Table 16. Fig. 20 shows the overall test cases result in pie chart.

Table 16 *Overall Test Cases Result*

Test Cases	Total Test Cases	Total Success	Total Failed
TC_100	5	5	-
TC_200	4	4	-
TC_300	9	9	-
TC_400	5	5	-
TC_500	7	6	1
TC_600	7	7	-
TC_700	7	6	1
TC_800	7	7	-
TC_900	3	2	1
TC_1000	3	3	-
Total	**57**	**54**	**3**

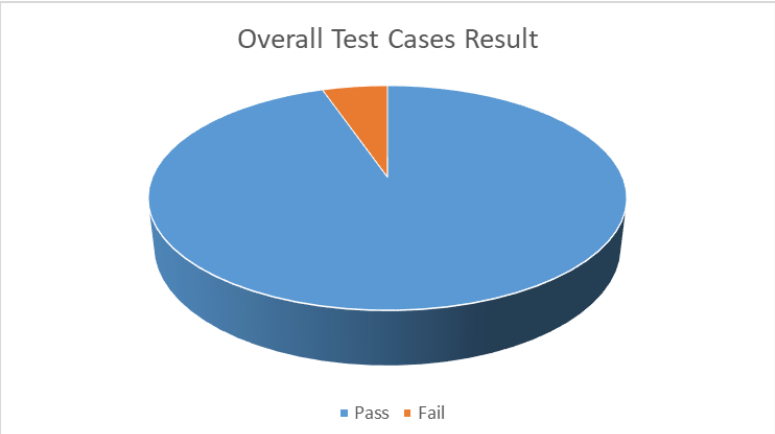


Fig. 20 *Overall Test Cases Result*

5.4 User Acceptance Testing

A type of testing called User Acceptance Testing (UAT) assesses the system's ability to carry out practical tasks and meet development goals. Before the system is formally launched, end users are given the chance to interact with it to see if there are any bugs or features that have been overlooked. User acceptance testing, which enables users to engage with the system prior to its official establishment, was used to test the Blackened Futsal Tournament Management System.

Users were given a questionnaire, and they were asked to provide input on the system in three different categories: usability, interface design, and functionality. The system's usability feedback is displayed in Fig. 21. Most respondents gave the system a high rating, ranging from 4 to 5, indicating that they found it easy and comfortable to use. This is because the system's flow encourages users to follow it through from start to finish without realising they are having problems. One respondent gave a score of three, indicating that different people had different experiences.

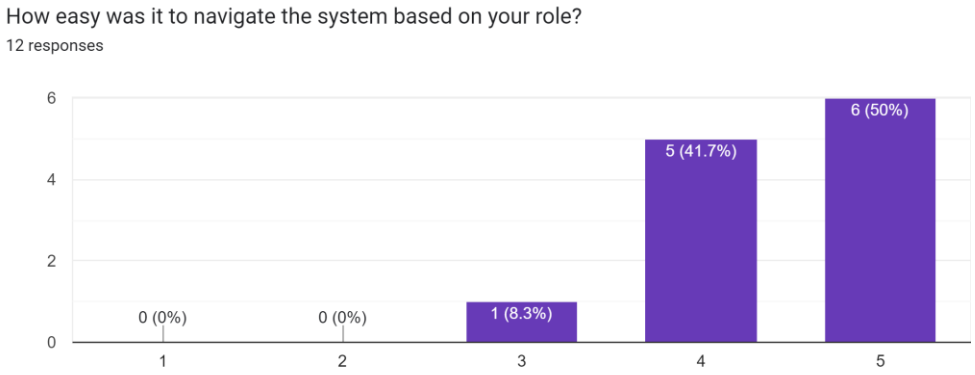


Fig. 21 *Usability Feedback*

The input that respondents provided regarding the system's interface design is displayed in Fig. 22. Additionally, the respondents gave the feedback a high score of 4 to 5, indicating that they are satisfied with the system's display and appearance and that the interface design is user-friendly. However, two respondents gave the system a score of 3, indicating that they believe it needs to be adjusted in terms of style, colours, or buttons to make it more professional.

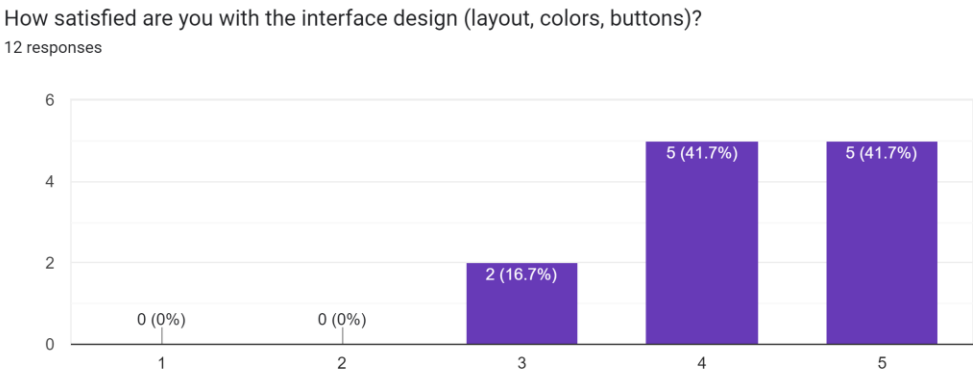


Fig. 22 *Interface Design Feedback*

Feedback from respondents regarding the system's functionality is displayed in Fig. 23. Many users, like nine people, gave the system's functionality a high rating because they had a positive experience with it, citing things like no delays or slow response times. Even with a good score, two respondents said that the system was unresponsive or took too long to load. This can be taken into consideration while adjusting for future developments.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 669-686

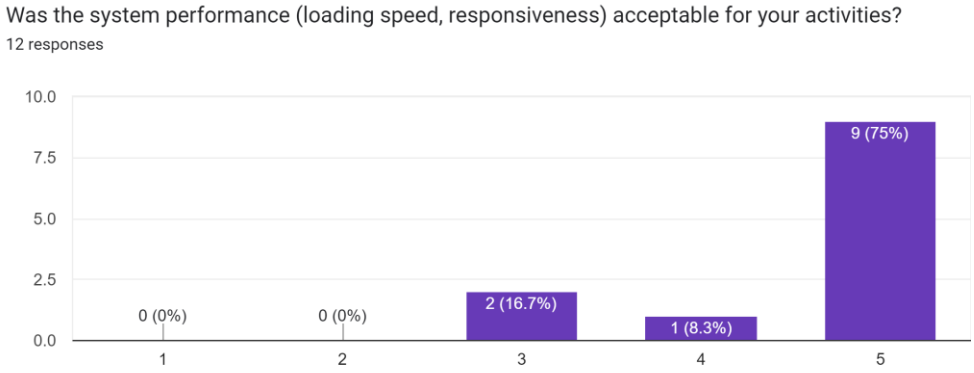


Fig. 23 *Functionality Feedback*

6. Conclusion

The Blackened Futsal Tournament Management System has demonstrated its potential to revolutionize futsal tournament operations by providing a centralized, efficient, and error-minimizing solution. This system addresses key challenges such as manual scheduling, result tracking, and lack of real-time communication, offering functionalities like automated scheduling, real-time score updates, and comprehensive reporting. The system's integration of Progressive Web App (PWA) technology guarantees device accessibility, improving the administrator, player, and fan experience. Iterative development was made possible by the implementation of a prototyping methodology, which made sure that user input was methodically taken into account to improve the system. The analysis emphasised the benefits of switching to an integrated platform and the inefficiencies of the current manual methods. By adding features like AI-driven statistics for player performance, interaction with live streaming platforms, and improved security measures, future research could broaden the reach of this system. Maintaining cooperation with stakeholders will guarantee that the system adapts to new requirements, enhancing its standing as a standard for sports management systems.

Acknowledgement

The authors extend their deepest gratitude to the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, for their invaluable support and resources throughout the duration of this project. The faculty's commitment to fostering innovation and providing technical guidance has been instrumental in the successful completion of this research. The author is also quite appreciative of the mentors and advisors, whose knowledge and perceptions have substantially improved the calibre of the work. They have inspired and motivated me with their helpful criticism, support, and dedication to greatness. Sincere gratitude is extended to the administrative staff and colleagues for helping to ensure seamless coordination and access to the facilities required for this project. Furthermore, special acknowledgement should be given to the stakeholders and research participants who gave their time, opinions, and experiences, as they were crucial in determining the results of the study. Finally, the author acknowledges the unwavering support of his family and friends, whose patience and understanding have been a pillar of strength throughout this journey. To everyone who played a part in the realization of this project, the author extends his deepest appreciation.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: study conception and design: Muhamad Akmal Irfan Lokman, Mohd Zainuri Saringat; data collection: Muhamad Akmal Irfan Lokman, Mohd Zainuri Saringat; analysis and interpretation of results: Muhamad Akmal Irfan Lokman, Mohd Zainuri Saringat; draft manuscript preparation: Muhamad Akmal Irfan Lokman, Mohd Zainuri Saringat. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Y. Zhang, X. Zhao, J. Shen, K. Shi, and Y. Yu, "Optimization of sports event management system based on wireless sensor network," Journal of Sensors, vol. 2021, no. 1, p. 1174351, 2021.

[2] P. R. M. I. Rashad, "Automating student registration and placement for seamless athletic tournaments," National EngiTech Digest, vol. 1, no. 1, 2024.
[3] Y. Nugraha, "Information system development with comparison of waterfall and prototyping models," JURNAL RISTEC: Research in Information Systems and Technology, vol. 1, no. 1, pp. 126-131, 2020.
[4] L. Andraini, "Pengelolaan surat menyurat dengan sistem informasi (Studi kasus: Kelurahan Gunung Terang)," Jurnal Portal Data, vol. 2, no. 1, 2022.
[5] S. Sauda and E. P. Agustini, "Implementasi prototype model dalam pengembangan aplikasi smart cleaning sebagai pendukung aplikasi smart city," MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 20, no. 1, pp. 73-84, 2020.
[6] I. R. Munthe, B. H. Rambe, R. Pane, D. Irmayani, and M. Nasution, "UML modeling and black box testing methods in the school payment information system," Jurnal Mantik, vol. 4, no. 3, pp. 1634–1640, 2020.
[7] M. I. Hossain, "Software development life cycle (SDLC) methodologies for information systems project management," International Journal for Multidisciplinary Research, vol. 5, no. 5, pp. 1–36, 2023.
[8] V. A. Kurniyanti and D. Murdiani, "Perbandingan model waterfall dengan prototype pada pengembangan system informasi berbasis website," Jurnal Syntax Fusion, vol. 2, no. 08, pp. 631–637, 2022.