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

AITCS

Vol. 6 No. 1 (2025) 772-789
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

UTHM Tigris E-Sports Website

Muhammad Eilman Hasbollah[1], Noraini Ibrahim[1]*

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

*Corresponding Author: noraini@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.01.043

Article Info

Received: 15 June 2024
Accepted: 18 June 2025
Available online: 30 June 2025

Keywords

E-Sports, Web-based Management System, Tournament Management, Bracket and Schedule Generator.

Abstract

The UTHM Tigris E-Sports Club, engaged in the field of competitive gaming, faces challenges in its decentralized tournament management workflow, absence of an official website, and an inefficient talent identification process. The UTHM Tigris E-Sports Website project aims to address these issues by streamlining workflows, enhancing visibility, and implementing a reliable talent identification system. The project scope includes the development of ten modules: Manage Authentication, Manage Administrator and Participant Profiles, View Participant Performance, Manage Tournament, Manage Tournament Registration, Bracket, Schedule, and Result, and Generate Report. An object-oriented approach is implemented to ensure reusability and modularity, thereby facilitating future improvements. The system is designed to centralize event management, improve visibility, and revolutionize talent identification. Testing results indicate that all modules fulfill the expected outcomes, achieving a 100% pass rate. This project is expected to significantly enhance the efficiency of tournament management and talent identification processes for the UTHM Tigris E-Sports Club.

1. Introduction

A website is a collection of publicly accessible, interlinked web pages that share a single domain name [1]. A website is considered essential to every organisation nowadays since it provides exposure across all kinds of digital devices. Esports, also known as eSports, e-Sports, e-games, or electronic sports is an organized competitive video gaming. It primarily involves teams competing against each other in tournaments for a cash prize [2]. The UTHM Tigris E-Sports Club, formerly known as Blackmamba UTHM, transitioned under new leadership in May 2023, led by Chairman Mohamad Hazly bin Mohd Noh, with support from Muhammad Haris Izham bin Zali and Mohamad Ashraf bin Nordin. As the official E-Sports Club of Universiti Tun Hussien Onn Malaysia (UTHM), its mission is to promote e-sports among UTHM students. However, the club faces developmental challenges due to a lack of an official website. This deficiency needs to improve effective communication of the club's activities and the vibrant gaming culture it aims to foster. Utilising tools like Google Form and Challonge, the decentralised workflow during tournaments exacerbates issues with human errors and data integrity. Talent identification for e-sports tournaments is also hindered, relying on separate competitions and in-game statistics. To address these challenges, the UTHM Tigris E-Sports Website aims to centralise workflows, streamline talent identification, and establish a robust online presence, crucial for enhancing visibility and competitiveness within the e-sports community.

The paper will be organized as follows: Section 2 provides a thorough exploration of related works pertinent to the project. Section 3 outlines the methodology employed in the research. Moving on to Section 4, a detailed

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 772-789

examination of the system outcomes is presented. Finally, Section 5 engages in a discussion of the concluding aspects of the project.

2. Related Work

E-sports can be simply defined as video games played competitively in a highly organised setting [3]. It is a huge sector considering the global e-sports market size was valued at US$1.45 billion in 2022 according to Prof Ts Dr Tan Chi Ike [4]. Currently, highest prize pool for e-sports tournament was held by Valve's annual world championship in Dota 2, The International 10, that offered grand total of US$40 million prize pool [5]. As years goes on, organising e-sports tournament has become more popular internationally and locally due to its popularity among new generation, including UTHM students. Running an e-sports tournament is just like any other sports event; it all starts with a plan [6]. There are eight elements that usually being considered when organising an e-sports tournament which are video game selection, determining tournament format and rules, tournament schedule, promotion, registration, preparation, execution, and awards and prizes of the tournament. However, to centralise the workflow of the UTHM Tigris E-Sports club, most of these elements have been brought under one web-system which will also act as the club's official website.

The current process of UTHM Tigris E-Sports club's tournament handling can be divided into three separate phase which are tournament promotion, tournament handling, and athlete selection. For better visualization of the current processes, three swimlane diagrams are shown in APPENDIX A; represent each main phase of the tournament handling. Three similar tournaments management system have been examined to identify strengths of each system and determine how they can improve the system. The three selected systems are Challonge, Toornament, and Score7. Table 1 shows the comparison between existing systems and the developed system.

Table 1 *Comparison between existing systems and UTHM Tigris E-Sports Website*

Features/System	Challonge [7]	Toornament [8]	Score7 [9]	UTHM Tigris E-Sports Website
Authentication (login and sign-up)	√ combined organizer and player	√ organizer and player	√ organizer	√ administrator and participant
Administrator Profile	√	X	√	√
Participant Profile	√	√	X	√
Feed Post	X	√	X	√
Tournament Registration	√	√	X	√
Tournament Bracket	√	√	√	√
Tournament Schedule	√	√	√	√
Participant Performance	X	X	X	√ number of participation and win count are counted for each participant

The three mentioned web-system and the UTHM Tigris E-Sports Website are compared according to some features. All the systems have a login functionality which will grant the users access to the system functions. To have an account, they must first go through a registration process which the existing systems have option for free or paid account. Meanwhile for the developed system, there is no fees charged for the account creation. Both Toornament and Score7 have players or participants module which allow the participant to view the available tournament and track their tournament progress which have similarity to the UTHM Tigris E-Sports Website system's participant module. Tournament bracket and schedule generator are core functions for all the studied system, which allow organizer to easily executed these fundamental steps in handling tournament. For tournament registration, both Challonge and Toornament have the option to create a registration form through their system while Score7 require the admin to insert the participant details by themselves. Moreover, the three existing systems including the UTHM Tigris E-Sports Website allow organizers to update the tournament result on the system to be displayed to the participants. Lastly, the standout feature that the UTHM Tigris E-Sports Website system serves is it allows the administrator to track participant performance for UTHM e-sports athletes selection purpose based on tournament result recorded in the system.

3. Methodology

According to [10], prototyping model is a system development method which a prototype is built, tested, and then reworked as necessary until an acceptable outcome is achieved which the complete system or product can be developed. For this project there are 2 prototypes produced so far. The prototype acts as a testing ground for gathering stakeholders' and customers' feedback. Since the prototype model offer a smaller version of the actual finished product, precise feedback must be incorporated into the development procedure. This is to ensure the prototype undergoes iterative improvement and refinements, and gradually evolving into the final product. The prototyping model usually involves several phases which are planning, analysis, design, prototype development, user evaluation and refining, and implementation and testing. Table 2 list the phases and their associated tasks and output.

Table 2 *Software development activities and their task*

Phase	Task	Output
Planning	• Proposed the project • Determine the project schedule, activities, and output	• Project proposal • Gantt chart
Analysis	• Conduct interview session with the stakeholder • Observation on current workflow of the club	• Swimlane diagram (To-be-model) • Use case diagram • Use case specification • Activity diagram • Sequence diagram • Class diagram • Requirement Definition
Design	• Architectural design • Design the User Interface (UI) • Design the database	• System's architecture diagram • Wireframe of the system • Schema table and data dictionary
Prototype 1 Development	• Development of system prototype based on modules or function	• System prototype 1
User Evaluation (Prototype 1)	• Planning the user evaluation • Conduct the evaluation session • Collect the feedback of the evaluation session	• User evaluation feedback on Prototype 1
Refining Prototype 1	• Analysing feedback and data collected • Improvement of the prototype 1	• Updated design of prototype 1 • Updated Requirement Definition 1
Prototype 2 Development	• Development of system prototype based on modules or function	• System prototype 2
User Evaluation (Prototype 2)	• Planning the user evaluation • Conduct the evaluation session • Collect the feedback of the evaluation session	• User evaluation feedback on Prototype 2
Refining Prototype 2	• Analysing feedback and data collected • Improvement of the prototype 2	• Updated design of prototype 2 • Finalized Requirement Definition
Implementation and Testing	• Implement all the functions and modules into one system • Run a test case to the finished system	• Finished system • Test case

The analysis and design are detail out in the following sections.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 772-789

3.1 Analysis

This section outlines the results derived from the analysis of the system, presented through various visual representations. These include a swimlane diagram, a use case diagram, detailed use case specifications, the corresponding sequence diagram, an activity diagram, a class diagram, and a comprehensive requirement definition.

3.1.1 Swimlane diagram

The swimlane diagram illustrates who is responsible for each step in a process. It also describes the relationship between different actions and actors [11]. Fig. 1 shows the swimlane diagram of the UTHM Tigris E-Sports Website.

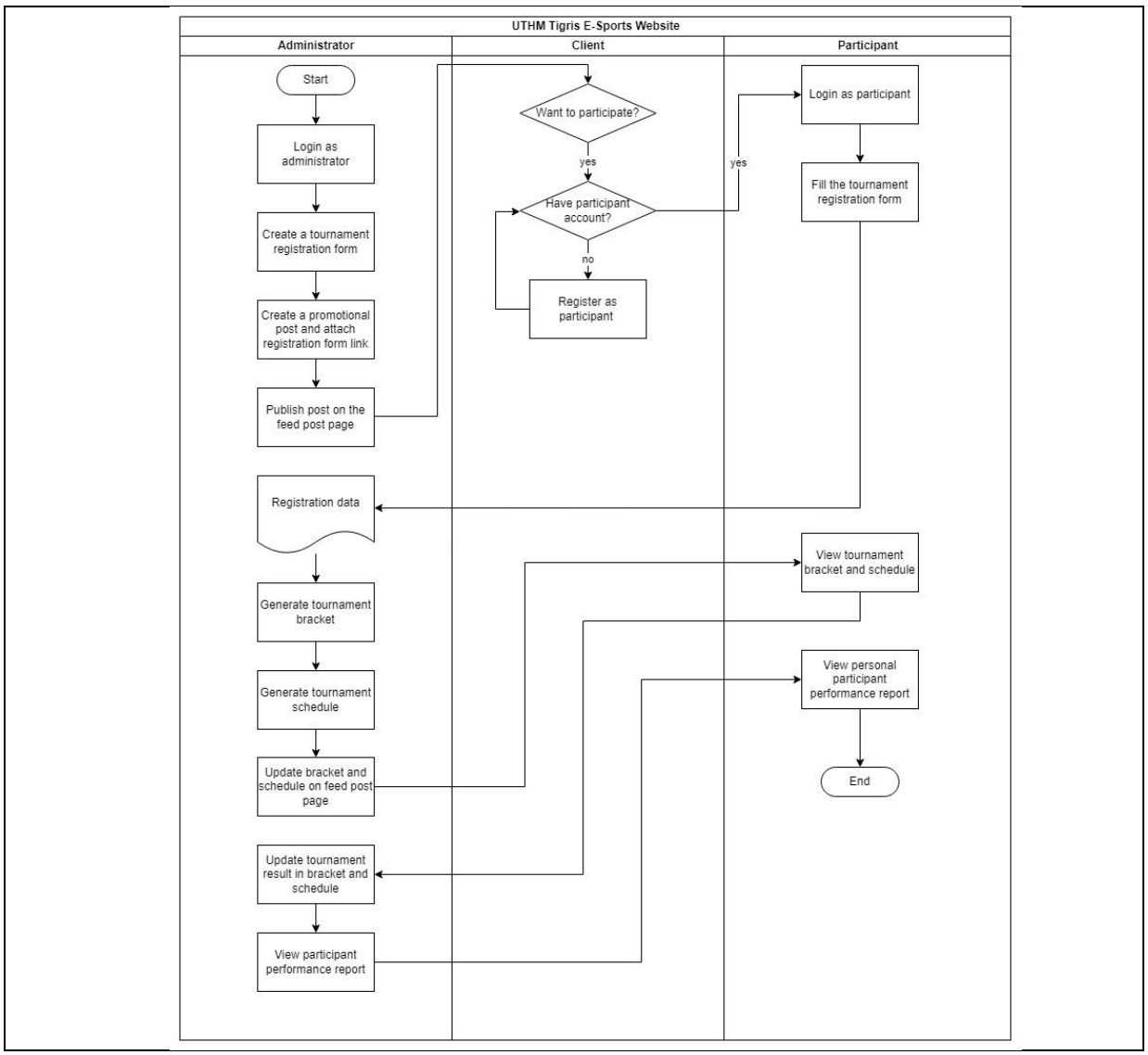


Fig. 1 *Swimlane diagram of UTHM Tigris E-Sports Website*

3.1.2 Use case diagram

A use case illustrates a unit of functionality provided by the system. The primary purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases [12]. Fig. 2 shows the use case diagram for the system.

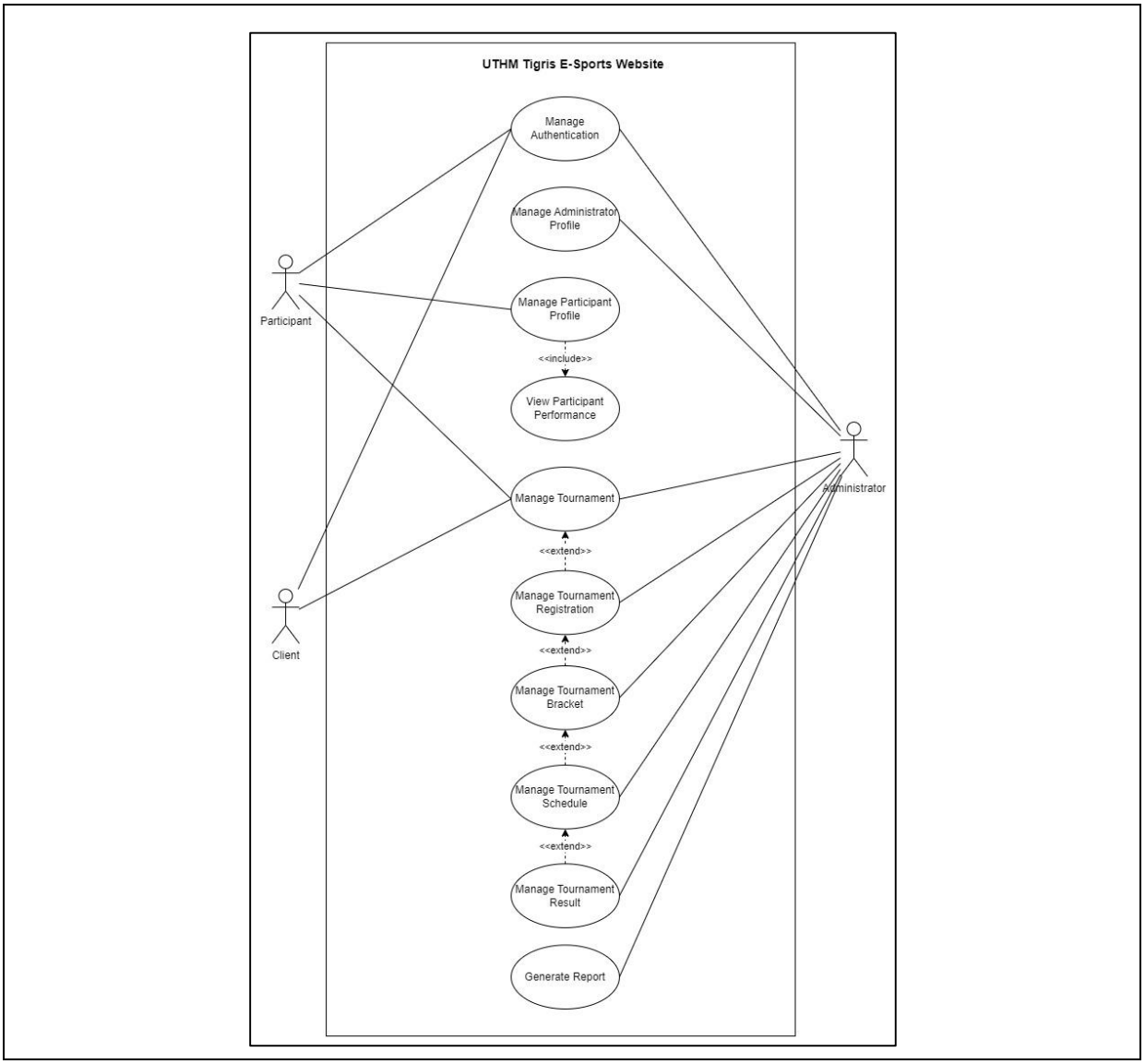


Fig. 2 *Use Case Diagram of UTHM Tigris E-Sports Website*

There are ten (10) main use cases for UTHM Tigris E-Sports: manage authentication, manage administrator profile, manage participant profile, view participant performance, manage tournament, manage tournament registration, manage tournament bracket, manage tournament schedule, manage tournament result, and generate report. The manage authentication module handles the login and registration of users. If the administrator enters a valid administrator ID and password, the system redirects them to the administrator dashboard, while clients can browse the website without logging in. If clients want to participate in a tournament, they must register and login as participants. Participants can access the participant profile module, which allow them to view and edit their personal information. The main page of the system for both clients and participants is the home page, handled by the manage tournament module. The tournament post consists of tournament details and links to the tournament registration. Administrators are responsible for generating, publishing, updating, and deleting elements in tournament registration, bracket, schedule, and results. The view participant performance allows participants to view their tournament participation. Lastly, the generate report module helps the club to identify potential talent among participants by generating essential charts.

3.1.3 Class diagram

The class diagram is frequently employed in object-oriented systems to demonstrate the system's static viewpoint. It presents a collection of classes, interfaces, and their interconnections [13]. Fig. 3 shows the class diagram of the UTHM Tigris E-Sports Website.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 772-789

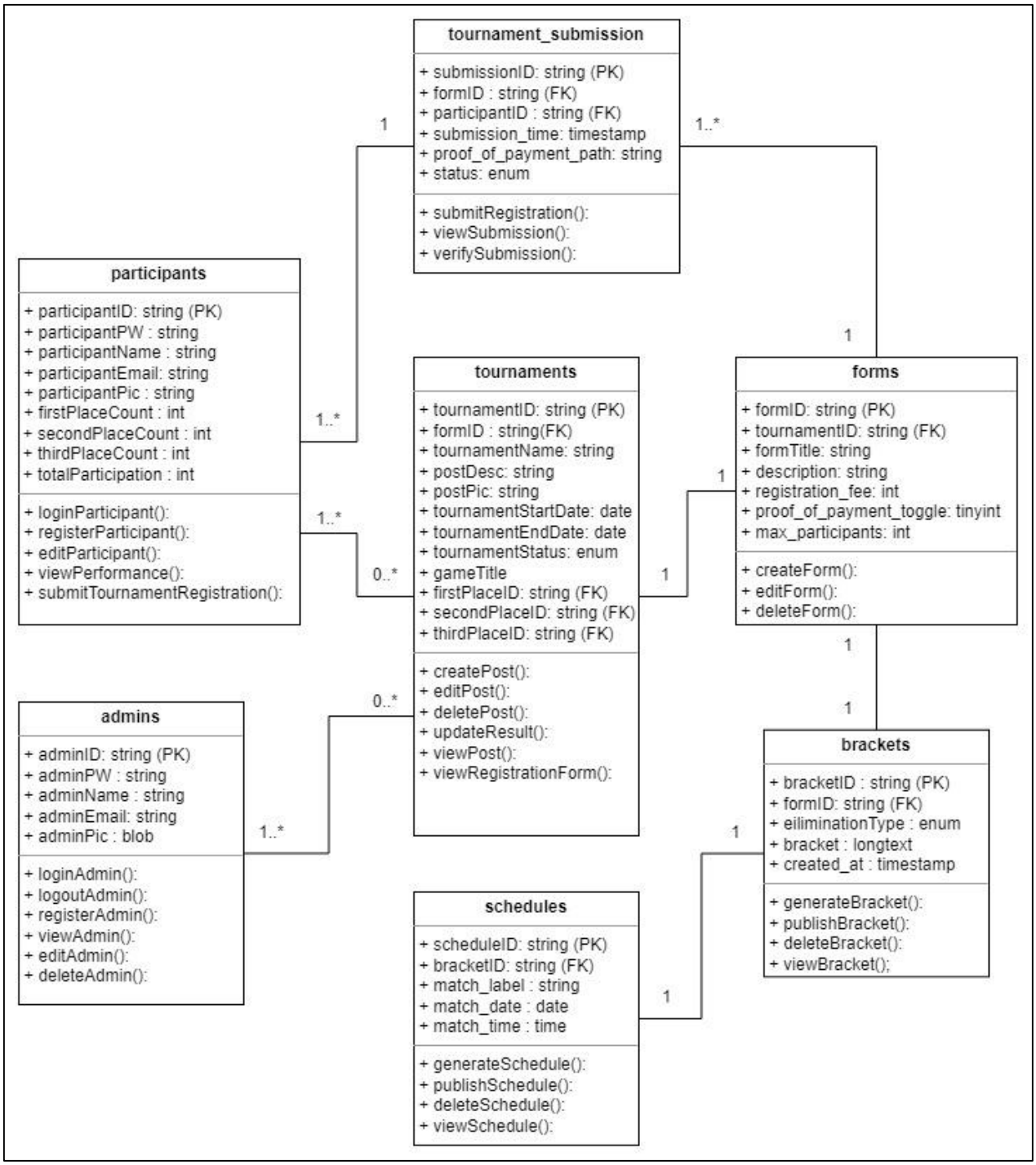


Fig. 3 *Class Diagram of UTHM Tigris E-Sports website*

3.1.4 Requirement Definition

Functional requirements refer to the features and operations that a system must have to meet functional perspectives [14]. Table 3 shows the functional requirements of the system.

Table 3 *Functional requirements of the system*

Modules	Functionalities
Manage Authentication Module	• The system shall display the login page. • The system shall allow administrators and participants to login to the system using Admin ID or Participant ID and password. • The system shall display the registration page. • The system shall allow student to register as participant by filling the registration form. • The system shall notify the user if there are missing data or incorrect input during the login and registration process.
Manage Administrator Profile Module	• The system shall allow the administrator to edit their personal information.
Manage Participant Profile Module	• The system shall allow participant to edit their personal information.
View Participant Performance	• The system shall allow the participants to view their performance.
Manage Tournament Module	• The system shall allow administrators to create, edit, and delete tournament post. • The system shall allow clients and participants to tournament post published by the administrator from the website.
Manage Tournament Registration Module	• The system shall allow administrators to create, edit, and delete registration forms on the website. • The system shall allow administrator to view and verify registration data. • The system shall allow participant who are interested in participating to register through the website.
Manage Tournament Bracket Module	• The system shall allow administrator to use the tournament bracket generator. • The system shall allow administrator to publish tournament bracket to the feed post. • The system shall allow clients and participants to view the tournament bracket generated on the website.
Manage Tournament Schedule Module	• The system shall allow administrator to use the tournament schedule generator. • The system shall allow administrator to publish tournament schedule to the feed post. • The system shall allow clients and participant to view the tournament bracket generated on the website.
Manage Tournament Result Module	• The system shall allow the administrator to update winner of ended tournament. • The system shall allow the clients and participants to view the winner of the ended tournament.
Generate Report Module	• The system shall allow administrator to view, print, or save system report.

Non-functional requirements define a software system's attributes and qualities, not its functionalities. They represent the constraints and standards that a software system should have, often related to performance, security, scalability, reliability, usability, and other non-functional aspects [15]. Table 4 shows the non-functional requirements of the system.

Table 4 *Non-functional requirements of the system*

Requirements	Description
Performance	• The website's load time should not be more than one second for users.
Operational	• The system should be able to function well at all web browsers. • The system should be able to be used anytime with the present of internet connection.
Security	• Administrator shall log in to the system by using Admin ID and password. • Participant shall log in to the system with Participant ID and password.

Table 4 *(Cont.)*

	• The password needs to be encrypted in the database.
Usability	• The system shall have a user-friendly and easy to understand interfaces.
Reliability	• The system should be able to perform tasks 97% without failure.
Availability	• The system shall always be available for access all the time.

3.2 Design

The design is represented in two (2) outcomes. They are the database schema and data dictionary.

3.2.1 Database Design

A database schema is a logical blueprint or structural design that represents the organization of data in a database. Following are the database schema for UTHM Tigris E-Sports website:

i. `Admins (adminID (varchar), adminPW (varchar), adminName(varchar), adminEmail (varchar), adminPic (varchar))`
ii. `Participants (participantID (varchar), participantPW (varchar), participantName (varchar),participantEmail (varchar), participantPic (varchar), firstPlaceCount (int), secondPlaceCount (int), thirdPlaceCount (int), totalParticipation (int))`
iii. `Tournaments (tournamentID (varchar), formID (varchar), tournamentName (varchar), postDesc (varchar), postPic (varchar), tournamentStartDate (date), tournamentEndDate(date), tournamentStatus (enum), gameTitle (varchar), firstPlaceID (varchar), secondPlaceID (varchar), thirdPlaceID (varchar))`
iv. `Forms (formID (varchar), tournamentID (varchar), formTitle (varchar), description (varchar), registration_fee (int), proof_of_payment_toggle (tinyint), max_participant(int))`
v. `Tournament_submission (submissionID (varchar), formID (varchar), participantID (varchar), submission_time (timestamp), proof_of_payment_path (varchar), status (enum))`
vi. `Brackets (bracketID (varchar), bracketTitle (varchar), numberOfMatches (int), tournamentFormat (enum))`
vii. `Schedules (scheduleID (varchar), scheduleTitle (varchar), numberOfDays (int), matchesDuration (int))`

4. Result and Discussion

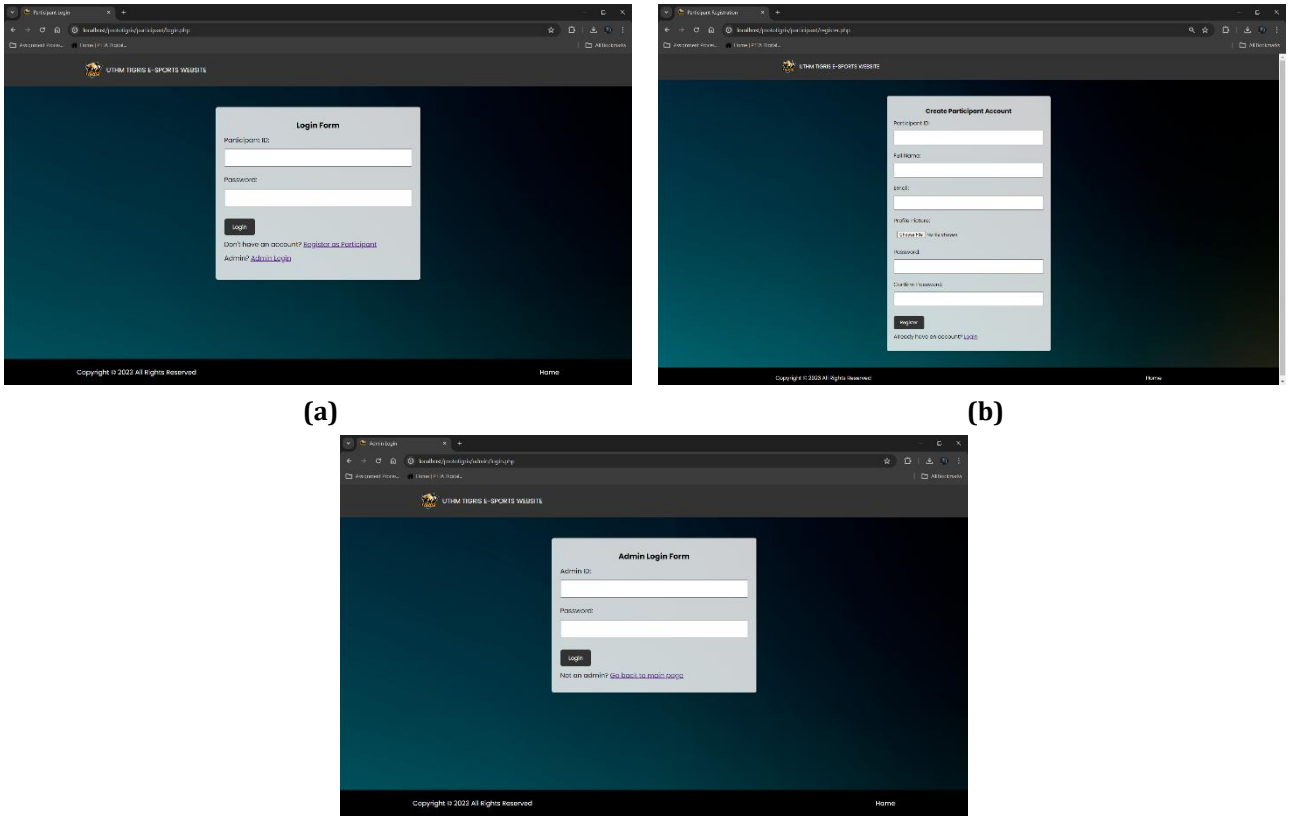
This section is divided into two parts: the implementation part, which showcases all the webpages created and the code segments used in the system, and the testing part, which summarizes the test results of the system.

4.1 Implementation

In this section, the created webpages and code segments for each module in the system are illustrated and explained.

4.1.1 Manage Authentication Module

Manage authentication module covers the login and registration of both participants and administrators. This module is crucial to determine users access to the system. Fig. 5 shows the interfaces of manage authentication module of UTHM Tigris E-Sports Website.



(a) **(b)**

(c)

Fig. 5 *(a) Participant Login Interface; (b) Participant Registration Interface; (c) Administrator Login Interface*

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $participantID = $_POST['participantID'];
    $participantPW = $_POST['participantPW'];

    // Validate login credentials
    $sql = "SELECT * FROM participants WHERE participantID='$participantID'";
    $result = $conn->query($sql);

    if ($result->num_rows == 1) {
        // Fetch the user data
        $row = $result->fetch_assoc();

        // Verify the password
        if (password_verify($participantPW, $row['participantPW'])) {
            // Successful login
            loginParticipant($row['participantID'], $row['participantName']);
            header("Location: ../client/index.php"); // Redirect to participant homepage
            exit();
        } else {
            // Invalid password
            $_SESSION['loginError'] = "Invalid participant ID or password";
            header("Location: login.php"); // Redirect back to the login page
            exit();
        }
    } else {
        // Invalid participant ID
        $_SESSION['loginError'] = "Invalid participant ID or password";
        header("Location: login.php"); // Redirect back to the login page
        exit();
    }
}

$conn->close();
?>
```

(a)

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $adminID = $_POST['adminID'];
    $adminPW = $_POST['adminPW'];

    // Validate login credentials
    $sql = "SELECT * FROM admins WHERE adminID='$adminID' AND adminPW='$adminPW'";
    $result = $conn->query($sql);

    if ($result->num_rows == 1) {
        // Successful login
        $row = $result ->fetch_assoc();
        loginAdmin($row['adminID'], $row['adminName']);
        header("Location: dashboard.php"); // Redirect to admin dashboard
        exit();
    } else {
        $loginError = "Invalid admin ID or password";
    }
}

$conn->close();
?>
```

(c)

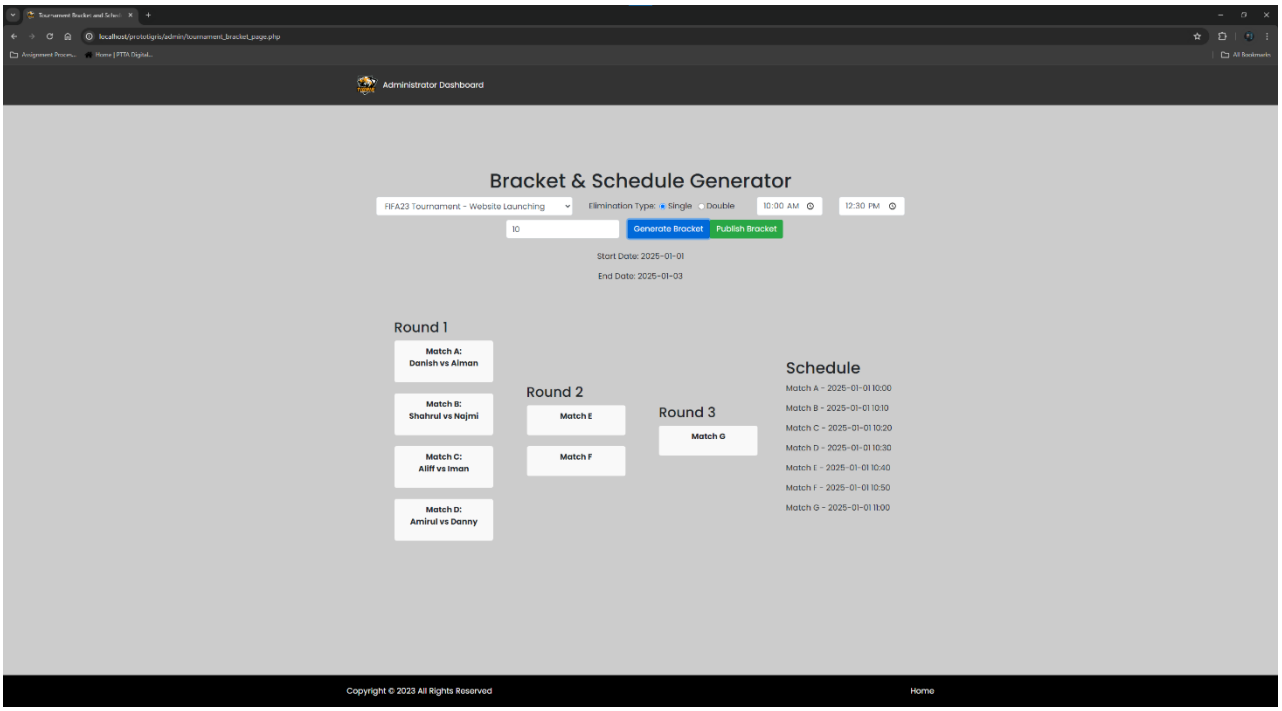
```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form data
    $participantID = $_POST['participantID'];
    $participantName = $_POST['participantName'];
    $participantEmail = $_POST['participantEmail'];
    $participantPW = $_POST['participantPW'];
    $confirmPassword = $_POST['confirmPassword'];

    // Check if passwords match
    if ($participantPW !== $confirmPassword) {
        echo "<script>alert("Passwords do not match"); window.history.back();</script>";
        exit();
    }

    // Check if participantID already exists
    $sql = "SELECT participantID FROM participants WHERE participantID = '$participantID'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        echo "<script>alert("Participant ID already exists"); window.history.back();</script>";
        exit();
    }

    // Check if participantEmail already exists
    $sql = "SELECT participantEmail FROM participants WHERE participantEmail = '$participantEmail'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        echo "<script>alert("Email already exists"); window.history.back();</script>";
        exit();
    }

    // Handle profile picture upload
    $targetDir = "../uploads/";
    $targetFile = $targetDir . basename($_FILES["participantPic"]["name"]);
    $uploadOk = 1;
    $imageFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));

    // Check if image file is a actual image or fake image
    $check = getimagesize($_FILES["participantPic"]["tmp_name"]);
    if ($check !== false) {
        $uploadOk = 1;
    } else {
        echo "<script>alert("File is not an image."); window.history.back();</script>";
        $uploadOk = 0;
    }

    // Check if file already exists
    if (file_exists($targetFile)) {
        echo "<script>alert("Sorry, file already exists."); window.history.back();</script>";
        $uploadOk = 0;
    }

    // Check file size
    if ($_FILES["participantPic"]["size"] > 500000) {
        echo "<script>alert("Sorry, your file is too large."); window.history.back();</script>";
        $uploadOk = 0;
    }

    // Allow only certain file formats
    if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType != "gif") {
        echo "<script>alert("Sorry, only JPG, JPEG, PNG & GIF files are allowed."); window.history.back();</script>";
        $uploadOk = 0;
    }

    // Check if $uploadOk is set to 0 by an error
    if ($uploadOk == 0) {
        echo "<script>alert("Sorry, your file was not uploaded."); window.history.back();</script>";
    } else {
        // If everything is ok, try to upload file
        if (move_uploaded_file($_FILES["participantPic"]["tmp_name"], $targetFile)) {
            // Hash the password before storing it in the database
            $hashedPassword = password_hash($participantPW, PASSWORD_DEFAULT);

            // File uploaded successfully, continue with database insert
            $sql = "INSERT INTO participants (participantID, participantName, participantEmail, participantPic, participantPW)
                    VALUES ('$participantID', '$participantName', '$participantEmail', '$targetFile', '$hashedPassword')";

            if ($conn->query($sql) === TRUE) {
                // Registration successful, redirect to login page
                echo "<script>alert("Registration successful"); window.location.href = "login.php";</script>";
                exit();
```

(b)

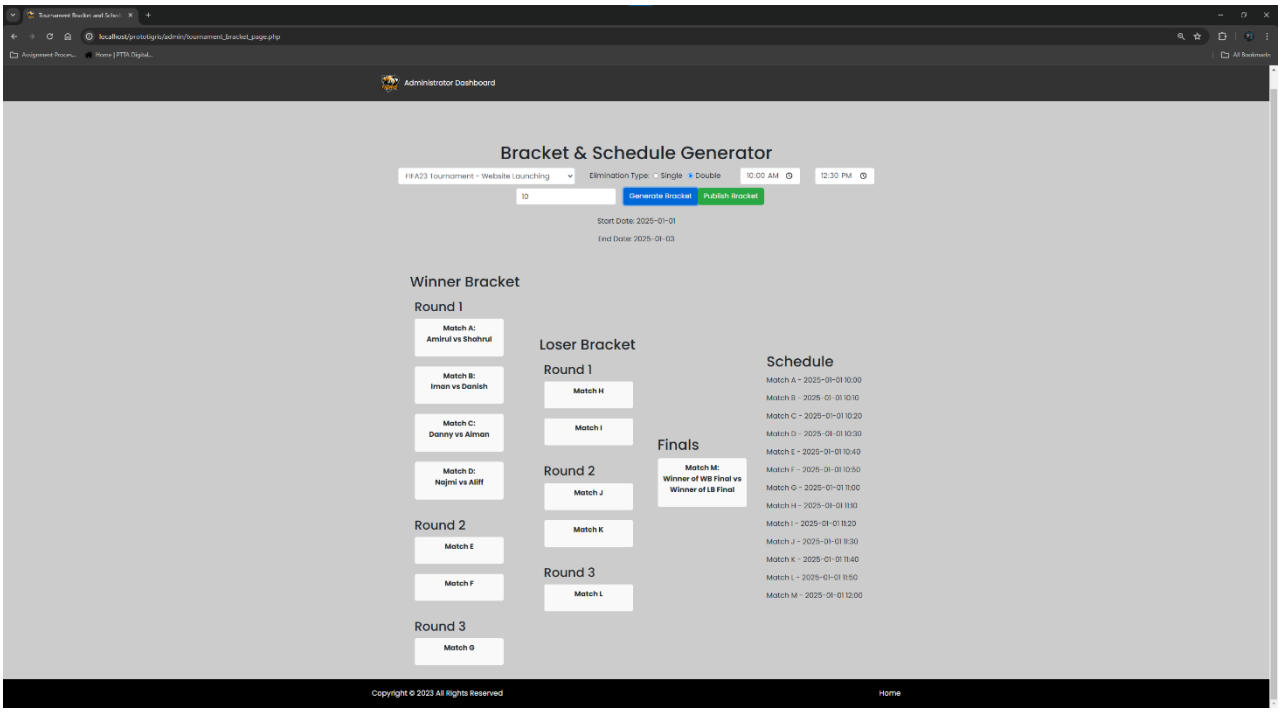
Fig. 6 *(a) Participant Login; (b) Participant Registration; (c) Administrator Login*

Fig 6 shows the code segment of manage authentication module. Users must enter valid credential which will allow them access to the system based on their credential level. Valid credentials for participant will redirect user to the homepage of the website, while for the administrator will be the administrator dashboard. New users can register as participants in the participant registration page while for the administrator, only existing administrators can access the administrator registration page for security reasons.

4.1.2 Manage Tournament Bracket and Schedule Module



(a)



(b)

Fig. 7 *(a) Single-Elimination Generated Bracket Interface; (b) Double-Elimination Generated Bracket Interface*

Fig. 7 shows the interfaces of tournament bracket and schedule generator page for administrator. Manage Tournament Bracket and Schedule are considered essential modules of the system. These modules provide bracket and schedule generator functions which streamline the workflow of the club when handling an e-sport tournament. The administrator can select the tournament format with time constraint that is necessary to generate precise schedule. Fig. 8 shows the code segment of bracket and schedule generator.

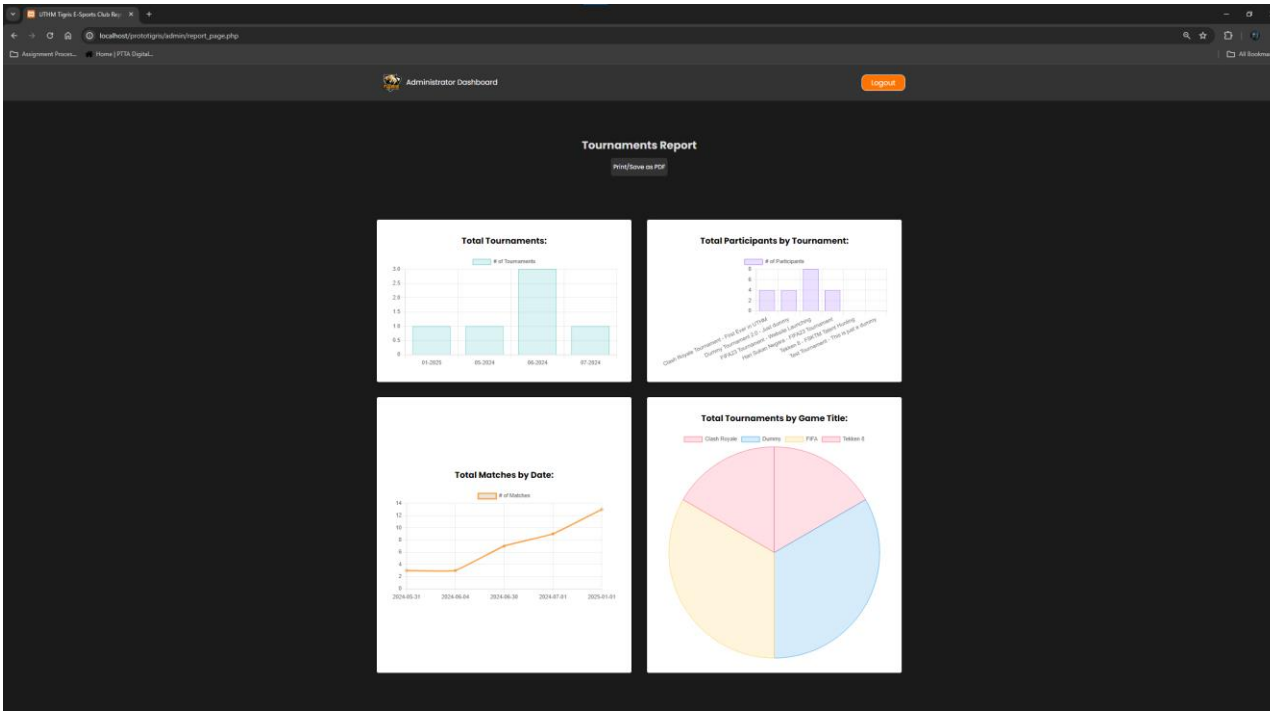
```php
<?php
include('../includes/db.php');
include('../includes/session.php');

function generateSingleEliminationBracket($participants) {
    $rounds = [];
    $numParticipants = count($participants);

    // Handle byes for non-power of 2 participants
    $numRounds = ceil(log($numParticipants, 2));
    $totalSlots = pow(2, $numRounds);
    $numByes = $totalSlots - $numParticipants;

    for ($i = 0; $i < $numByes; $i++) {
        array_push($participants, null);
    }

    while (count($participants) > 1) {
        $round = [];
        for ($i = 0; $i < count($participants); $i += 2) {
            $round[] = [$participants[$i], $participants[$i + 1]];
        }
        $rounds[] = $round;
        $participants = array_map(function($match) {
            return 'Winner of Match';
        }, $round);
        $participants = array_filter($participants);
    }

    return $rounds;
}

function generateDoubleEliminationBracket($participants) {
    $winnerBracket = generateSingleEliminationBracket($participants);
    $numRounds = count($winnerBracket);
    $loserBracket = array_fill(0, $numRounds, []);

    for ($round = 0; $round < $numRounds; $round++) {
        $numMatchesWB = count($winnerBracket[$round]);

        if ($round == 0) {
            // First round in loser's bracket gets half the matches of the first round in winner's bracket
            for ($i = 0; $i < $numMatchesWB / 2; $i++) {
                $loserBracket[$round][] = ['', ''];
            }
        } else if ($round == $numRounds - 1) {
            // Last round in loser's bracket has 1 match
            $loserBracket[$round][] = ['', ''];
        } else {
            // Middle rounds in loser's bracket match the count of the previous round in loser's bracket
            for ($i = 0; $i < count($loserBracket[$round - 1]); $i++) {
                $loserBracket[$round][] = ['', ''];
            }
        }
    }

    // Final match between the winners of the winner and loser brackets
    $finals = [['Winner of WB Final', 'Winner of LB Final']];

    return [
        'winnerBracket' => $winnerBracket,
        'loserBracket' => $loserBracket,
        'finals' => $finals
    ];
}

function generateMatchSchedule($numMatches, $startDate, $endDate, $startTime, $endTime, $matchDuration) {
    $schedule = [];
    $currentDate = new DateTime($startDate);
    $endDate = new DateTime($endDate);
    $currentDate->setTime(...explode(':', $startTime));
    $endDate->setTime(...explode(':', $endTime));
    $interval = new DateInterval('PT' . $matchDuration . 'M');
    $dayEndTime = clone $currentDate;
    $dayEndTime->setTime(...explode(':', $endTime));

    for ($i = 1; $i <= $numMatches; $i++) {
        if ($currentDate > $dayEndTime) {
            $currentDate->modify('+1 day');
            $currentDate->setTime(...explode(':', $startTime));
            $dayEndTime = clone $currentDate;
            $dayEndTime->setTime(...explode(':', $endTime));
        }

        if ($currentDate > $endDate) {
            break;
        }

        $schedule[] = [
            'match' => 'Match ' . chr(64 + $i),
            'date' => $currentDate->format('Y-m-d'),
            'time' => $currentDate->format('H:i')
        ];
        $currentDate->add($interval);
    }

    return $schedule;
}
```

Fig. 8 *Code Segment of Bracket and Schedule Generator*

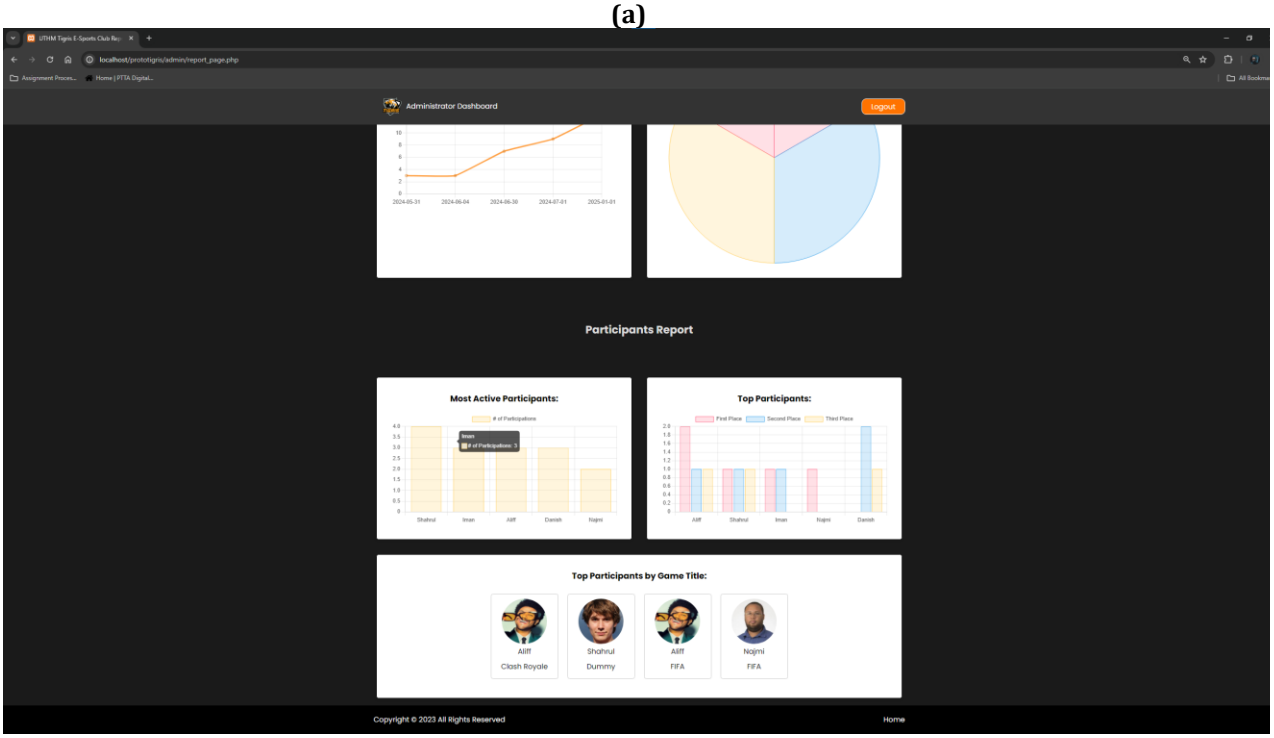
Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 772-789

4.1.3 Generate Report

Generate report module allows administrators to generate tournament and participant reports which give essential information and overview of the data acquired from the system. The participants report helps administrators to identify talented participants that have potential to become UTHM e-sports athletes. Fig. 9 shows the interfaces of the generate report module.



(a)



(b)

Fig. 9 *(a) Generate Tournament Report Interface; (b) Generate Participant Report Interface*

This code segment shown in Fig. 10 is responsible for fetching data from the server and creating various charts using the Chart.js library to visualize the fetched data. The fetchChartData function asynchronously retrieves data from the server's get_chart_data.php endpoint. Then, each chart is created using specific data arrays for labels and corresponding values. For example, createTotalTournamentsChart generates a bar chart showing the total number of tournaments per month, while createParticipantsPerTournamentChart creates a bar chart

displaying the number of participants per tournament. Similarly, other functions like createMatchSchedulesChart, createTopParticipantsChart, createTournamentsByGameTitleChart, createMostActiveParticipantsChart, and createTopParticipantsByGameTitle are responsible for creating different types of charts based on their respective datasets. Finally, the initializeCharts function orchestrates the entire process by fetching data, then calling each chart creation function to render the charts on the webpage.

```
// Function to create Total Tournaments Chart by Month
function createTotalTournamentsChart(tournaments) {
    const labels = tournaments.map(t => t.month);
    const data = tournaments.map(t => t.totalTournaments);
    const ctx = document.getElementById('totalTournamentsChart').getContext('2d');
    new Chart(ctx, {
        type: 'bar',
        data: {
            labels: labels,
            datasets: [{
                label: '# of Tournaments',
                data: data,
                backgroundColor: 'rgba(75, 192, 192, 0.2)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true
                }
            }
        }
    });
}

// Function to create Participants Per Tournament Chart
function createParticipantsPerTournamentChart(participants) {
    const labels = participants.map(p => p.tournamentName);
    const data = participants.map(p => p.participantCount);
    const ctx = document.getElementById('participantsPerTournamentChart').getContext('2d');
    new Chart(ctx, {
        type: 'bar',
        data: {
            labels: labels,
            datasets: [{
                label: '# of Participants',
                data: data,
                backgroundColor: 'rgba(153, 102, 255, 0.2)',
                borderColor: 'rgba(153, 102, 255, 1)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true
                }
            }
        }
    });
}
```

Fig. 10 *Code Segment of Generate Report*

4.2 Testing

In this section, functionality testing assesses each module's functionality. The system is also tested by the end user through user acceptance testing. The test cases are executed to compare the actual results with the expected results. Table 5 shows the list of test cases for this system.

Table 5 *List of Test Cases*

Test Cases	Description	Status
TC_100	**Manage Authentication**	**Fail/Pass**
TC_100_01	The registered user logs in with valid participantID or adminID and password and is redirected to the website homepage or administrator dashboard.	Pass
TC_100_02	The user registers with valid participantID, email, and password will be redirected to the login page.	Pass

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 772-789

Table 5 *(Cont.)*

TC_200	**Manage Administrator Profile**	**Fail/Pass**
TC_200_01	The administrator edits the user profile.	Pass
TC_300	**Manage Participant Profile**	**Fail/Pass**
TC_300_01	The participant edits the user profile.	Pass
TC_400	**View Participant Performance**	**Fail/Pass**
TC_400_01	The participant views the participant performance.	Pass
TC_500	**Manage Tournament**	**Fail/Pass**
TC_500_01	The administrator creates, edit, and delete tournament post.	Pass
TC_500_02	The participant and client view the published tournament post on the website homepage.	Pass
TC_600	**Manage Tournament Registration**	**Fail/Pass**
TC_600_01	The administrator creates, edit, delete, and publish tournament registration form.	Pass
TC_600_02	The administrator views registration data of the published tournament registration form.	Pass
TC_600_03	The administrator verifies proof of payment uploaded by the participants.	Pass
TC_600_04	The participant registers for tournament via published tournament registration form.	Pass
TC_700	**Manage Tournament Bracket**	**Fail/Pass**
TC_700_01	The administrator selects registration data from tournament registration form and select tournament format.	Pass
TC_700_02	The system generates tournament bracket based on administrator input.	Pass
TC_700_03	The administrator publishes the generated bracket to the website homepage.	Pass
TC_700_04	The administrator deletes published bracket from the system.	Pass
TC_700_05	The participant and client view the published tournament bracket on the website homepage.	Pass
TC_800	**Manage Tournament Schedule**	**Fail/Pass**
TC_800_01	The administrator input the start and end time of the tournament and match duration.	Pass
TC_800_02	The system generates tournament bracket based on administrator input and start and end date of the selected tournament.	Pass
TC_800_03	The administrator publishes the generated schedule to the website homepage.	Pass
TC_800_04	The administrator deletes published schedule from the system.	Pass
TC_800_05	The participant and client view the published tournament schedule on the website homepage.	Pass
TC_900	**Manage Tournament Result**	**Fail/Pass**
TC_900_01	The administrator selects 1st, 2nd, and 3rd place participant for selected tournament.	Pass
TC_900_02	The participant views 1st, 2nd, and 3rd place participant for end tournament on the website homepage.	Pass
TC_1000	**Generate Report**	**Fail/Pass**
TC_1000_01	The system provides reports regarding tournament and participant.	Pass
TC_1000_02	The administrator can determine the tournament with the most engagement and the number of tournaments held in a month.	Pass
TC_1000_03	The administrator able to determine participant with most involvement and success in participation.	Pass

A total of 26 test cases had been conducted to test the UTHM Tigris E-Sports website. The system has been passed successfully. Table 6 shows the overall result of the test cases.

Table 6 *Overall Result of Test Cases*

Test Case ID	Total Test Cases	Total Success	Total Fail
TC_100	2	2	0
TC_200	1	1	0

Table 6 *(Cont.)*

TC_300	1	1	0
TC_400	1	1	0
TC_500	2	2	0
TC_600	4	4	0
TC_700	5	5	0
TC_800	5	5	0
TC_900	2	2	0
TC_1000	3	3	0

Summary of the result in Table 6 is represented in a graph shown in Fig. 11.

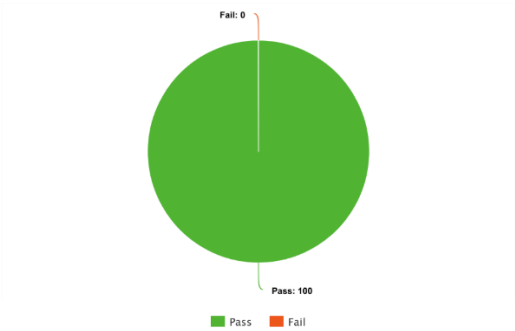


Fig. 11 ***Test Summary***

User acceptance testing had been undergone from the perspective of UTHM students as participants and UTHM Tigris E-Sports Club members as administrators. In general, this test is passed successfully. The test is completed via Google Form involving 12 respondents from the participants' perspective and 3 respondents from the administrators' perspective. There are 10 questions for both participant and administrator perspective, prepared to understand user's perception and satisfaction with the functionality and usability of UTHM Tigris E-Sports website. The scale that used in this test is 1 to 5 scale where 1 indicates strongly disagree and 5 indicates strongly agree.

5. Conclusion

In summary, the UTHM Tigris E-Sports website has successfully achieved its objectives of enhancing the club's operations and user engagement through its ten modules. While it offers numerous advantages such as secure authentication, administrator control, and streamlined registration, it also has some disadvantages, including complexity for new users and limited tournament formats for bracket generator. By implementing the recommended measures, the website can further improve its functionality and user experience, contributing to the continued growth and success of the UTHM Tigris E-Sports Club.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

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

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 772-789

Appendix A

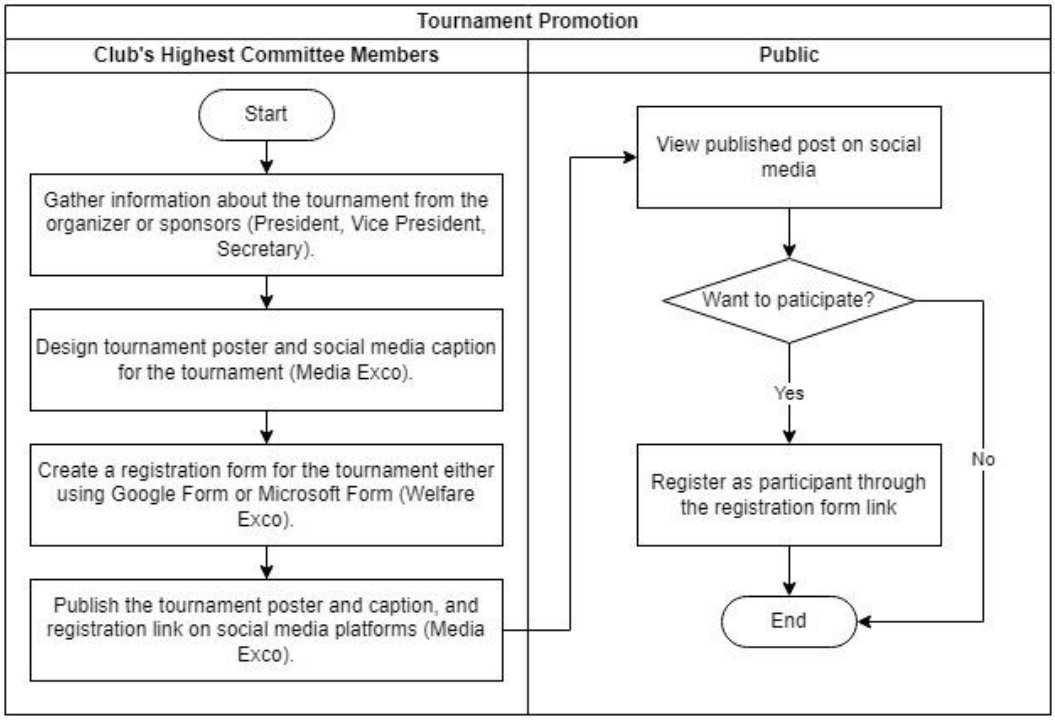


Figure Appendix A.1 *Swimlane diagram of current tournament promotion process*

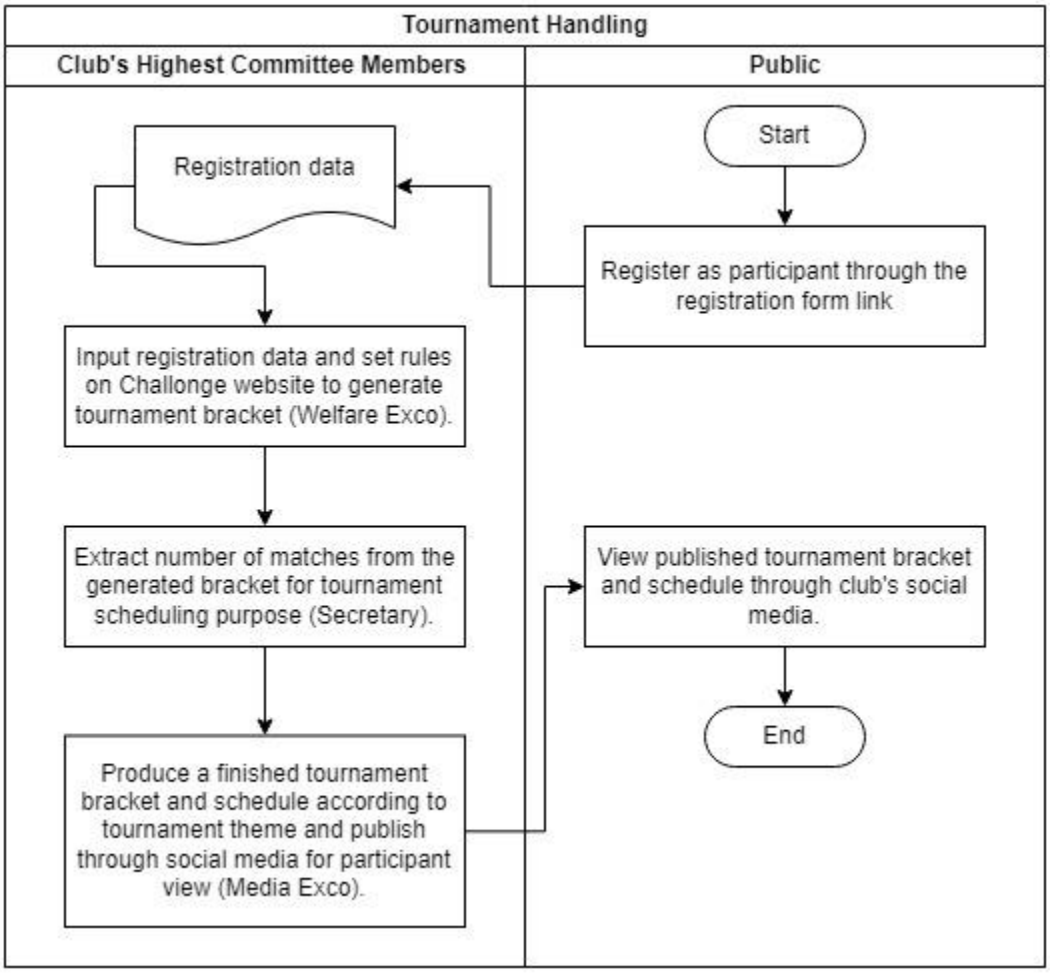


Figure Appendix A.2 *Swimlane diagram of current tournament handling process*

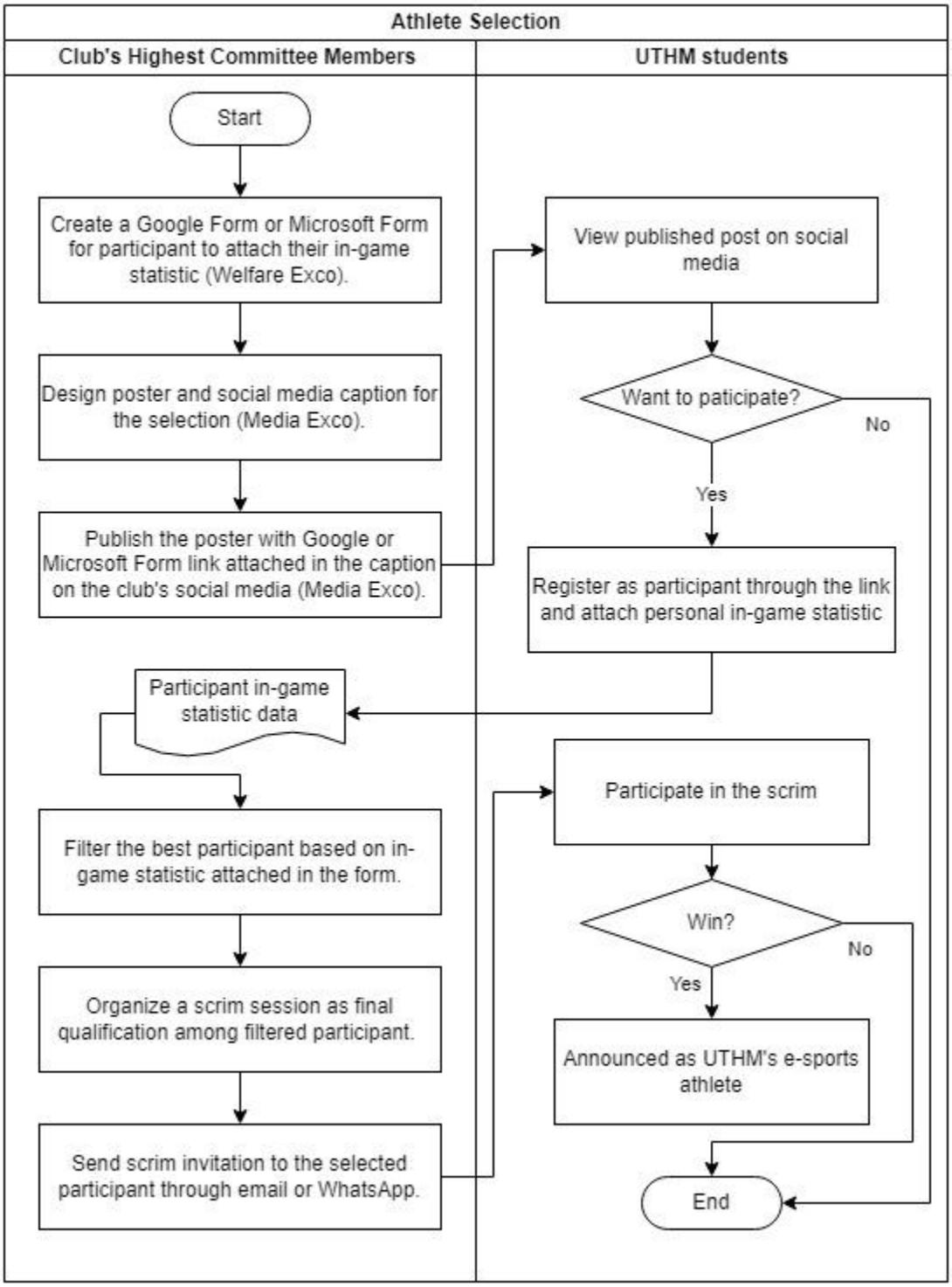


Figure Appendix A.3 *Swimlane diagram of current tournament athlete selection process*

References

[1] Rouse, M. (2020). *What is a website? - definition from Techopedia.* Techopedia. Retrieved February 12, 2024, from https://www.techopedia.com/definition/5411/website.

[2] Adair, C. (2023). *What is esports? history, top teams, revenues and risks.* Game Quitters. Retrieved February 18, 2024, from https://gamequitters.com/what-is-esports/.

[3] Leroux-Parra, M. (2020, August 2). *Esports part 1: What are esports?.* Harvard International. Retrieved February 18, 2024, from https://hir.harvard.edu/esports-part-1-what-are-esports/.

[4] Mardhiah, A. (2023, Sep 11). *Esports as emerging sector boosting Malaysia's economy.* The Malaysian Reserve. Retrieved February 18, 2024, from https://themalaysianreserve.com/2023/09/11/esports-as-emerging-sector-boosting-malaysias-economy/.

[5] Heath, J. (2023). *10 biggest prize pools in esports (updated 2023).* Dot Esports. Retrieved February 20, 2024, from https://dotesports.com/general/news/biggest-prize-pools-esports-14605.

[6] Vener, N. (2022, June 7). *Planning & executing your sports bar's esports event.* SportsTV Guide. Retrieved February 20, 2024, from https://sportsbarmarketing.com/planning-executing-sports-bars-esports-event/.

[7] Challonge. (2019). *Challonge Tournament Bracket.* Retrieved January 7,2024, from https://challonge.com/.

[8] Toornament. (2013). *Tournament – Esports tournament management software*. Retrieved January 7,2024, from https://www.toornament.com/en_GB/.

[9] Score7.io. (2020). Score7 *Tournament Generator and Bracket Maker.* Retrieved January 7,2024, from https://www.score7.io/.

[10] Lewis, S. (2023). *What is The prototyping model?.* CIO. Retrieved March 5, 2024, from https://www.techtarget.com/searchcio/definition/Prototyping-Model#:~:text=The%20prototyping%20model%20is%20a,or%20product%20can%20be%20developed.

[11] Miro. (2023). *What is a swimlane diagram.* Miro. Retrieved March 5, 2024, from https://miro.com/diagramming/what-is-a-swimlane-diagram/.

[12] Bell, D. (2003, June 14). *An introduction to the Unified Modeling Language.* IBM developer. Retrieved February 18, 2024, from https://developer.ibm.com/articles/an-introduction-to-uml/.

[13] Osis, J. & Donins, U. (2017). Class Diagram. *Topological UML modeling: An improved approach for domain modeling and software development, Elsevier,* 2017, 3–51.

[14] Petek, Z. (2023). *Functional and non-functional requirements in software architecture design.* Medium. Retrieved March 8, 2024, from https://ziga-petek.medium.com/functional-and-non-functional-requirements-in-software-architecture-design-aeb892546b8e.

[15] LTD, H. T. C. (2022). Abstract. *Database principles and technologies - based ONHUAWEI GAUSSDB, Springer.* 2 1–1.