

The Development of Santai Esports Tournament Management System

Zaliff Usri¹, Nurezayana Zainal^{1*}

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

*Corresponding Author: nurezayana@uthm.edu.my
DOI: <https://doi.org/10.30880/aitcs.2024.05.02.058>

Article Info

Received: 28 July 2024

Accepted: 30 October 2024

Available online: 15 December 2024

Keywords

Information Management System,
Bracket Generator, Certificate
Generator, Studio Reservation,
Centralized Hub

Abstract

The "Santai Esports Tournament Management System" tackles key issues in esports, notably inefficiencies and errors from traditional paper-based methods, challenges in social media promotion, and security risks from cash prize distribution. This project modernizes tournament management and financial transactions, enhancing transparency and streamlining processes for organizers and participants. By engaging the gaming community more effectively and providing a secure financial system, it eliminates paper-based work. Using the Prototyping Model, the system ensures user-centered design through iterative improvements. Developed with PHP, HTML, JavaScript, and CSS, this web-based platform simplifies tournament management, facilitates financial transactions, and serves as a centralized information hub. It improves management, financial processes, community engagement, and ensures a transparent, engaging gaming environment, enhancing user experience and supporting the growth of the esports community within Santai Esports.

1. Introduction

In this dynamic world of competitive gaming, there's a growing need for a user-centric platform for organizing and participating in esports tournaments. Esports, or "electronic sports," is competitive video gaming involving professional players, teams, and organizations competing in various video games, often multiplayer-focused. In recent years, esports have become one of the most rapidly growing forms of new media driven by the growing provenance of online games and online broadcasting technologies [1]. It has been estimated that more than 70 million people watched esports during 2013 [2]. This growth has been attributed to the accelerated popularity of digital communication, which has boosted esports consumption and paved the way for potential investment opportunities and sponsorship [3]. Introducing the "Santai Esports Tournament Management System," emphasizing ease and inclusivity for all tournament organizers and participants. This platform simplifies tournament management, financial transactions, and updates about upcoming events, all in one digital space. Whether you are a dedicated gamer seeking competition, an esports organization looking to host tournaments, or an administrator, Santai Esports is designed to meet your needs. Additionally, the global esports audience grew to 495 million people in 2020, an 11.7% increase from the previous year, driven by streaming platforms like Twitch and YouTube Gaming [4]. This growth has continued due to the rising engagement levels and substantial investments from major brands, positioning esports as a major player in the global entertainment industry with promising growth prospects in the coming years [5].

1.1 Problem Statement

Santai Esports faces several critical challenges in managing and promoting their esports tournaments. Firstly, they rely on outdated manual and paper-based methods for recording information and managing participant registrations, resulting in inefficiencies and errors. This outdated method makes their operations slower and brings the risk of mistakes.

Secondly, Santai Esports faces a challenge with its advertising approach, relying heavily on social media platforms. However, the gaming community may not be effectively engaged through this method. The sheer volume of content on social media means that esports promotions can easily get lost among unrelated posts. This challenge limits their ability to reach and engage with their target audience, ultimately hindering their efforts to increase awareness and participation in their tournaments.

Thirdly, Santai Esports currently distributes tournament winnings in cash, which is inconvenient, less secure, and can lead to potential disputes. This method poses hardships for participants who can't wait until the end of the competition to receive their earnings. The lack of a digital payment system also raises financial concerns for both organizers and participants, impacting the platform's user-friendliness.

1.2 Objective

The objectives of this project are to:

- (a) Design Santai Esports Tournament Management System based on object-oriented approach.
- (b) Develop a tournament management system by using a web-based application.
- (c) Test the Santai Esports Tournament Management System through user acceptance testing

1.3 Scope

The system used by three users, which are the administrator, organizer, and participant of the Santai Esports Tournament Management System. The system will have nine modules: Sign up, Log in, Tournament Management, Reservation Studio, Tournament Registration, Certificate, Profile, Report and History. The details of the modules are shown in Table 1.

Table 1 Modules system for Santai Esports Tournament Management System

Modules	Description	Targeted User
Sign up	<ul style="list-style-type: none"> • New users can create an account and choose to become organizer or participant. 	Organizer and Participant
Login	<ul style="list-style-type: none"> • Users can log into their accounts by using a username and password. 	Administrator, Organizer, and Participant
Manage Profile	<ul style="list-style-type: none"> • User can access their profile and edit. 	Organizer and Participant
Manage Studio Reservation	<ul style="list-style-type: none"> • Organizer can reserve Santai Esports Studio to the administrator. • Administrator can approve the request for the reservation from the organizer. 	Administrator and Organizer
Manage Tournament	<ul style="list-style-type: none"> • Can create the esports tournaments. • Manage tournaments by defining tournament name, game, venue, total participants, poster, description, format and start time. • All users can see the information and progress of the tournaments. • Handle registration fees. 	Organizer and Participant
Registration Tournament	<ul style="list-style-type: none"> • Can participate to the tournament by join it. 	Participant
Manage Payment	<ul style="list-style-type: none"> • Participant can pay the registration fee of the tournament. • Organizer can pay the reservation of the studio to the organizer. 	Organizer and Participant
Manage News	<ul style="list-style-type: none"> • Can upload, update and remove any news related to esports to the user of the system. 	Administrator
Generate Certificate	<ul style="list-style-type: none"> • Participant can download the certificate that have been generate after the tournament ends. 	Participant

Table 1 Modules system for Santai Esports Tournament Management System (cont.)

Generate Report	<ul style="list-style-type: none"> Administrator can generate all the report of the reservation and tournament in the system. Organizer can generate the tournament report that they have been create. 	Administrator and Organizer
Access History	<ul style="list-style-type: none"> Trace back the previous tournament that they have created or joined. 	Organizer and Participant

1.4 Expected Result

The Santai Esports Tournament Management System provides tailored accounts for administrators, organizers, and participants, ensuring specific functionalities for each role. Participants can register for and participate in tournaments, manage their profiles, make payments for fees, monitor tournament progress, and generate certificates. Organizers have the capability to create and manage tournaments and events, handle financial transactions, generate tournament reports, reserve the PS5 studio, and manage news content. Administrators oversee the system's growth, manage payment features, monitor user activities, tournament data, and financial transactions, ensuring smooth and efficient operations across the platform.

1.5 Project Significance

This project creates a new system to manage tournaments better. It helps organizers by making it easier to sign up participants and keep track of who's joining. It stores all tournament details, making it easier to plan future events. For players, it is a place to find out about upcoming tournaments and keep track of their achievements.

Additionally, it lets organizers book dates for tournaments, making it easier for everyone to plan and work together. Overall, it makes running tournaments smoother and brings people together online.

1.6 Report Organization

There are 6 sections in this report. Section 1 includes the project background, problem statement, objectives, scope, expected result, project significance, and report organization. Section 2 provides more details of the concepts, technologies, terminology and existing systems relevant to this project. Section 3 covers the methodology used to develop the proposed system and explains the plans for each phase. Section 4 discusses the analysis and design of the proposed system. Section 5 explains the implementation and testing of the proposed system. Section 6 concludes the entire report by highlighting the advantages, disadvantages of the developed system, and offering valuable recommendations for future system improvements.

2. Literature Review

This section will discuss the literature review that had been done for the proposed system and the current system that is being used by Santai Esports.

2.1 Domain Background

Established on March 8, 2023, Santai Esports operates as an esports center located at No. 7 (Tingkat 1), Jalan Universiti 1, Taman Universiti, Parit Raja, with a prominent presence within the Parit Raja community and the students of Universiti Tun Hussein Onn Malaysia (UTHM). The center serves as a hub for esports activities, providing a platform for gaming enthusiasts to engage in competitive gaming and tournaments. Santai Esports plays a pivotal role in promoting esports within the region, functioning as a focal point for gaming events, fostering community participation, and enhancing the overall esports experience for both local and university-based gamers. The center's vision revolves around creating an interactive and inclusive gaming environment, leveraging technology and innovation to streamline tournament organization, player engagement, and community-building initiatives. With a focus on enhancing tournament management practices and community engagement, Santai Esports endeavors to contribute significantly to the dynamic landscape of esports in Parit Raja and UTHM, aspiring to become a leading entity in the regional esports sphere.

2.2 Key Technologies in the Esports Tournament Management System

The Esports Tournament Management System is like a specialized toolbox for running esports competitions smoothly. It helps at every stage, from planning events to handling sign-ups and organizing matches. There are tools for setting up tournaments, managing who joins, and even creating schedules automatically. It keeps everyone updated in real-time about matches and results. It's not just about organizing, it also helps keep the community engaged with bonuses and special platforms for players to stay in the loop. Table 2 shows the technologies and their role in the Santai Esports Tournament Management System.

Table 2 Technologies and Their Roles in the Santai Esports Tournament Management System

Aspect	Technology	Role and Description
Backend Development	PHP	<ul style="list-style-type: none"> Server-side scripting language. Handles backend tasks such as data processing, form submissions, and user authentication. Interacts with the database to generate dynamic page content and manage server-side logic. Manages user authentication and role-based access control.
Frontend Structure	HTML	<ul style="list-style-type: none"> Standard markup language for creating the structure of web pages. Defines the layout of the user interface including forms, tables, and navigation menus. Organizes content and ensures accessibility. Forms used for user inputs (e.g., tournament registration, profile updates).
Styling and Presentation	CSS	<ul style="list-style-type: none"> Used for styling HTML elements. Controls visual presentation, including colors, fonts, spacing, and layout. Ensures a consistent look and feel across all pages of the system. Separates content from presentation, making it easier to maintain and update visual aspects.
Interactivity and Responsiveness	JavaScript	<ul style="list-style-type: none"> Client-side scripting language. Enhances interactivity and functionality of web pages. Validates form inputs, creates dynamic content updates without reloading the page (using AJAX), and provides interactive elements such as modals and tooltips. Enhances user experience by making the system more responsive and engaging. Enables real-time interactivity and responsiveness (e.g., updating match scores, displaying notifications without page reloads).
Dynamic Content Generation	PHP & HTML	<ul style="list-style-type: none"> Combines PHP and HTML to generate dynamic content based on user interactions and database queries. Enables features such as real-time updates on tournament status, participant lists, and match results.
Form Handling and Data Processing	HTML, PHP & JavaScript	<ul style="list-style-type: none"> HTML forms are used for user inputs. PHP processes these forms on the server side, validates inputs, and stores the data in the database. JavaScript enhances this process by providing client-side validation for a smoother user experience.
User Authentication and Role Management	PHP	<ul style="list-style-type: none"> PHP scripts handle user authentication processes, ensuring authorized access. Implements role-based access control to differentiate between participants, organizers, and administrators.
Styling and Layout	CSS	<ul style="list-style-type: none"> Ensures visually appealing and consistent interface. Separates content from presentation for easier maintenance and updates.
Real-Time Interactivity	JavaScript	<ul style="list-style-type: none"> Enables real-time interactivity such as updating match scores or displaying notifications without page reloads. Creates a dynamic and user-friendly environment to keep participants and organizers engaged.

2.3 Study of Santai Esports Manual Tournament Process

The study looks at how Santai Esports runs its tournaments step by step. First, they rent a studio for the event. Then, they plan the tournament, deciding the games, rules, and schedules. People who want to join sign up at the studio. The organizer manually creates match brackets, which takes time and can have mistakes. Updates about matches and results are done manually during the tournament. After it ends, prizes and certificates are given out manually, which can cause delays or mistakes. The study wants to find what is not working well, like taking too much time or causing problems. It aims to suggest ways for Santai Esports to manage tournaments better, making things smoother and more efficient.

2.4 Comparison with the Existing System

In the preceding section, the comparison among the three systems is condensed and presented in Table 3. This section focuses on comparing the features of the existing system or application with those of the proposed system. Table 2 shows a comparison between three existing systems and proposed system.

Table 3 Systems' comparison

Module	Toornament	Challenge	Score7	Santai Esports Tournament Management System
Platform	Web-Based System	Web-Based System	Web-Based System	Web-Based System
Sign up	Yes	Yes	Yes	Yes
Login	Yes	Yes	Yes	Yes
Manage Profile	Yes	Yes	No	Yes
Manage Studio Reservation	No	No	No	Yes
Manage Tournament	Yes	Yes	Yes	Yes
Registration Tournament	Yes	Yes	No	Yes
Manage Payment	Yes	Yes	No	Yes
Manage News	No	No	No	Yes
Generate Certificate	No	No	No	Yes
Generate Report	No	No	No	Yes
Access History	Yes	Yes	Yes	Yes

3. Methodology

A methodology comprises a range of concepts, tools, techniques, and principles applied in the planning, execution, management, and advancement of a project [6]. The term "methodology" refers to comprehending the approach employed in the case study to achieve the research goal [7]. Methodologies offer a roadmap for tasks starting from the initial idea phase to the final implementation, presenting a structured method for problem-solving and decision-making. The selection of a methodology significantly impacts on project outcomes, influencing elements like cost, timeline, and overall quality [8]. Therefore, the prototype model is the chosen software process model for developing the proposed system.

The Prototyping Model in software development process involves creating an initial system version, called a prototype, aimed at collecting user feedback and enhancing requirements prior to the development of the final system. This prototype acts as a testing ground for gathering feedback from stakeholders and users. Based on this feedback, the prototype undergoes iterative improvements and refinements, gradually evolving into the final product. Reviewing the prototype serves as a preventive measure against misunderstandings that commonly occur when there is an assumption of mutual understanding between parties [9].

However, managing changes and preventing scope creep during iterations can pose challenges in this method. This methodology consists of six stages which are planning, design, development, testing, refinement, and implementation. Figure 1 shows the phases in the Prototype Model.

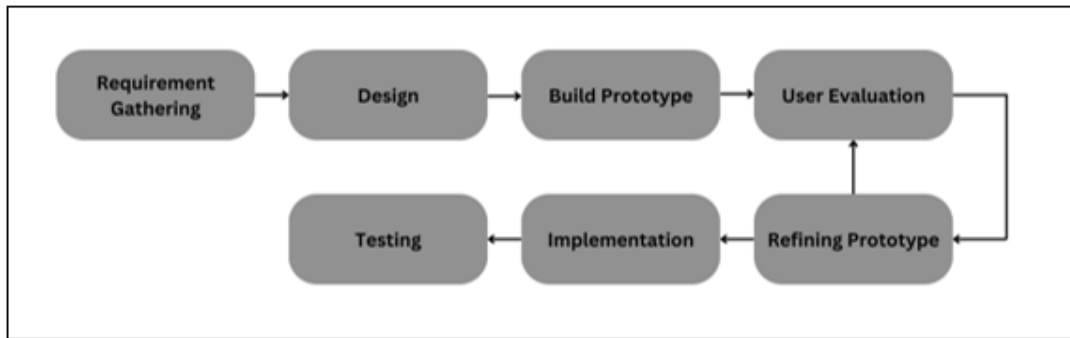


Fig. 1 Phases in Prototype Model

3.1 System Development Workflow

There are total of seven phases that are involved in this prototyping model. Table 4 shows the system development activities and the deliverables for each phase.

Table 4 Software Development Phases Activities

Phase	Task	Output
Requirement Gathering	<ul style="list-style-type: none"> Meeting and interviews Brainstorming and idea generation Requirement documentation Requirement review 	<ul style="list-style-type: none"> Requirements document Feedback reports Refined requirements and updates
Design	<ul style="list-style-type: none"> Architectural design Detailed design Interface design and data design 	<ul style="list-style-type: none"> System architecture diagrams Detailed design documents User interface Database and data models
Build Prototype	<ul style="list-style-type: none"> Prototyping development Integration of components 	<ul style="list-style-type: none"> Prototype system Codebase, source files, documentation and manuals
User Evaluation	<ul style="list-style-type: none"> Planning user evaluation Recruitment of participants Observation and data collection Analyzing feedback and data Generating reports and findings 	<ul style="list-style-type: none"> User evaluation report Usability findings and recommendation User feedback documentations
Refining Prototype	<ul style="list-style-type: none"> Reviewing user feedback Prioritizing enhancement areas Improve the prototype Continues feedback integration Stakeholder review and approval 	<ul style="list-style-type: none"> Revised prototype versions Documentation of changes Updated design Functional specifications
Implementation	<ul style="list-style-type: none"> Setting up development environments Coding and programming Database development User interface (UI) development Functionality integration 	<ul style="list-style-type: none"> Implemented system Codebase and source files Database and data structure User interface (UI) elements
Testing	<ul style="list-style-type: none"> Test planning Test environment setup User acceptance testing Test result analysis and reporting Test closure 	<ul style="list-style-type: none"> Test plan Test cases and scenario Test environment setup documentation User acceptance testing reports Test summary and closure report

4. Analysis and Design

Analysis involves understanding the current situation and spotting problems to figure out what the system needs. Design takes these findings and creates a detailed plan for how the system will work, including interactions and the look of the user interface. This process helps catch risks early, reducing issues later in development. A solid plan from the design phase cuts down development time and resources, making the process smoother. Overall, this method boosts the chances of delivering a successful system that works well for its users.

4.1 Functional and Non-Functional Requirements

Functional requirements define the specific functionalities and capabilities that a system, software, or product must perform. They outline what the system should do, detailing the essential tasks, operations, and interactions it needs to support to meet user needs and organizational objectives. Table 5 shows the modules and functionalities of the proposed system.

Table 5 *Functional Requirement of the proposed system*

No	Module	Functionalities
1	Sign Up	<ul style="list-style-type: none"> The system shall allow organizer and participant to create new accounts by providing necessary details.
2	Login	<ul style="list-style-type: none"> The system shall verify user credentials for access to the system.
3	Manage Profile	<ul style="list-style-type: none"> The system shall enable organizer and participant to manage and update personal details like contact information or profile pictures.
4	Manage Studio Reservation	<ul style="list-style-type: none"> The system shall allow organizer to reserve the Santai Esports studio for tournament hosting. The system shall allow organizer to see the status of the reservation. The system shall allow administrator to approve/reject the reservation from the organizer.
5	Manage Tournament	<ul style="list-style-type: none"> The system shall enable organizer to create new tournaments. The system shall provide tools to generate tournament brackets based on participant numbers and formats. The system shall allow organizer to set and modify match s The system shall update participant with the latest match results, schedules, or any changes.
6	Registration Tournament	<ul style="list-style-type: none"> The system shall enable participant to join the tournaments. The system shall allow participant to pay registration fees if needed.
7	Manage Payment	<ul style="list-style-type: none"> The system shall allow participant to pay the registration fee. The system shall allow organizer to distribute prize payment to the participant. The system shall allow organizer pay for the reservation studio.
8	Manage News	<ul style="list-style-type: none"> The system shall allow administrator to post the news. The system shall allow administrator to edit news. The system shall allow administrator to remove news.
8	Generate Certificate	<ul style="list-style-type: none"> Participant should be able to receive certificates for tournament participation.
9	Access History	<ul style="list-style-type: none"> The system shall allow organizer to review past tournaments they have organized, including details such as tournament formats, participants, and outcomes. The system shall enable participant to view their tournament participation history, including past tournament and achievements.

While non-functional requirements encompass the criteria that define the system operation and behaviour rather than its specific functionalities. Non-functional requirements provide the quality standards against which the system performance is evaluated, ensuring it meets user expectations, regulatory compliance, and organizational needs while addressing aspects beyond mere functionality. Table 6 below shows the non-functional requirements of the proposed system.

Table 6 Non-Functional Requirement of the proposed system

No	Requirements	Descriptions
1	Operational	The system should be able to function well at all web browsers.
2	Performance	The reasonable operation and response time of the operating system should be expected.
3	Security	The system is accessed only by authenticated users with valid username and password.
4	Availability	The system can be operated and available 24 hours a day.

4.2 User Requirement Analysis

User requirement analysis involves understanding what end-users need, expect, and prefer in a system or product. It includes directly engaging with stakeholders and users to learn about their workflows, challenges, and desired features. Techniques like interviews, surveys, and observations help gather and refine these needs. This ensures the final solution aligns with user goals and improves their experience. This analysis is essential for designing and developing systems that are user-friendly and effective. Table 7 shows the user requirements for the proposed system.

Table 7 User requirements of the proposed system

No	User Requirements
1.	Participant should be able to create a new account and switch their role to be organizer.
2.	Participant should be able to update preferences in their profiles.
3.	Participant should be able to join for tournaments conveniently and can pay easily for registration fee.
4.	Participant should be able to access real-time updates on tournament schedules, match results, and progress.
5.	Participant should be able to receive certificates for tournament participation.
6.	Organizer should be able to create new account and switch their role to be participant.
7.	Organizer should be able to manage tournament by create and update the tournament.
8.	Organizer should be able to manage studio reservation for hosting tournaments.
9.	Organizer should be able to handle registration fee collections and prize distributions.
10.	Organizer should be able to pay after the reservation studio.
11.	Organizer should be able to generate their tournament report.
12.	Administrator should be able to monitor user statistic, track the reservation statistic and tournament statistic.
13.	Administrator should be able to manage the permissions by the organizer to reserve the studio.
14.	Administrator should be able to access and generate comprehensive reports for reservation and tournament, including revenue and profit analysis.

4.3 Use Case Diagram

A use case diagram is a visual representation within the Unified Modelling Language (UML) that illustrates how a system interacts with its users or external entities. It showcases the various functionalities or actions a system performs in response to specific requests from actors (users, systems, or other entities). Figure 2 shows the use case diagram.



Fig. 2 Use case diagram

4.4 Domain Class Diagram

A class diagram is a visual representation in Unified Modelling Language (UML) that illustrates the structure and relationships within a system. It showcases classes, their attributes, methods, and the associations between different classes. Fig. 3 shows the domain class diagram of the proposed system.

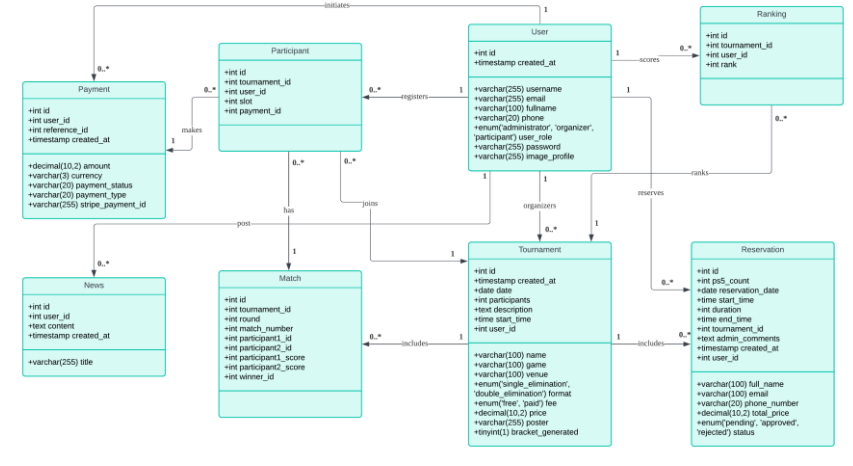


Fig. 3 Class diagram of the proposed system

4.5 Interface Design

Interface design involves creating visually and functionally cohesive systems that facilitate user interaction with the system. Interface design of the proposed system shown in the Figures 4 to 12.

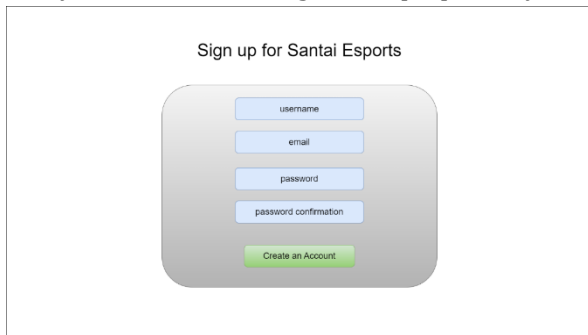


Fig. 4 Sign Up interface

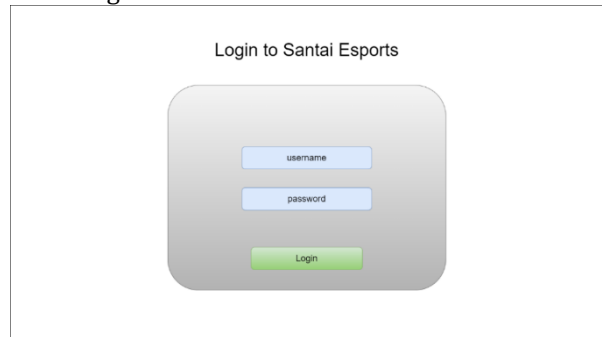


Fig. 5 Login interface

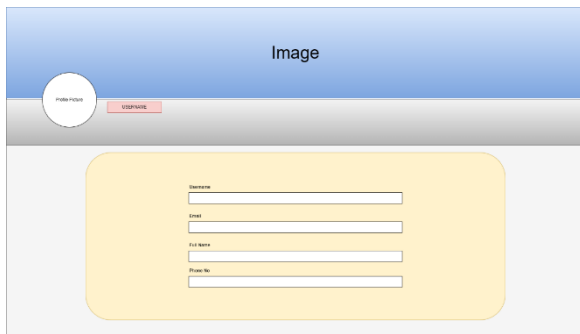


Fig. 6 User Profile interface

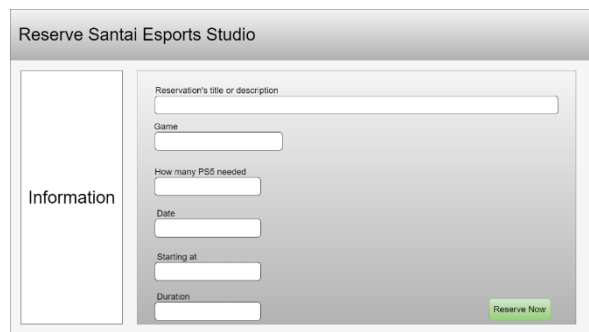


Fig. 7 Reservation Studio interface

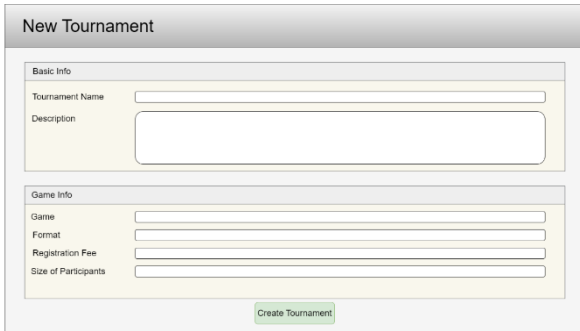


Fig. 8 Manage Tournament interface

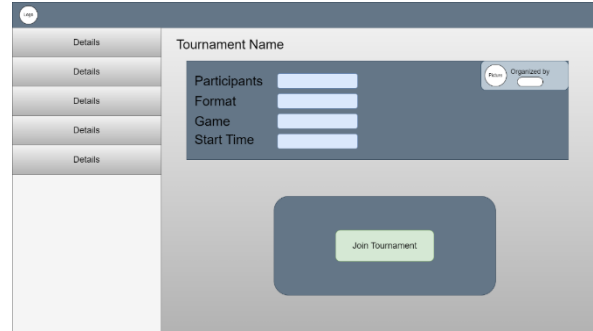


Fig. 9 Tournament Registration interface



Fig. 10 Manage Certificate interface

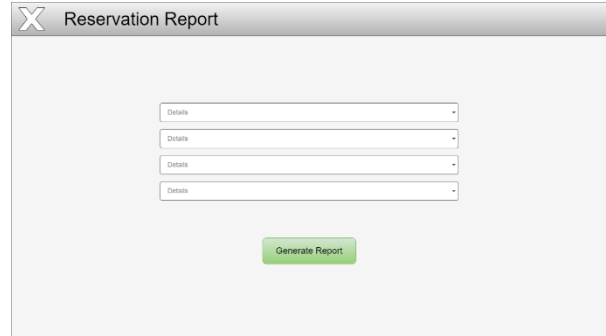


Fig. 11 Manage Reservation Report interface

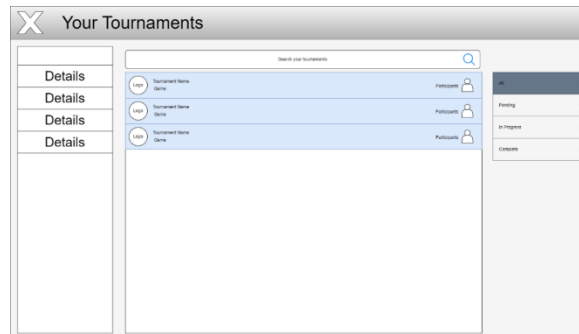
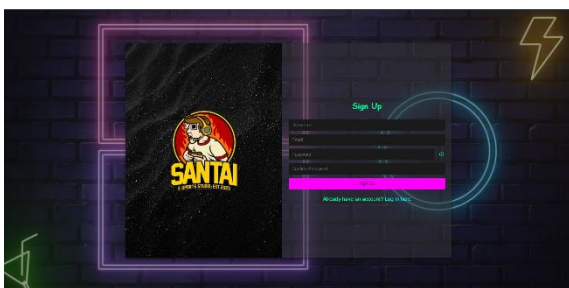


Fig. 12 Created Tournament History interface

5. Implementation and Testing

The implementation of the Santai Esports Tournament Management System involved several key stages. Initially, the system architecture was designed to accommodate three user roles: administrators, organizers, and participants. The development process utilized HTML, CSS, JavaScript, PHP, and MySQL to build the system's front-end and back-end functionalities. Key features such as user registration, profile management, tournament creation, and scheduling were implemented. Additionally, the system incorporated modules for managing news content, generating reports, download certificates, reserving studios, and facilitating secure payment transactions. The interface is depicted in Fig. to Fig. with the accompanying code segment.



```

<div id="signup-form">
  <div>Sign Up</div>
  <form action="authenticate.php" method="post">
    <input type="text" id="new_username" name="username" placeholder="Username" required>
    <input type="email" id="email" name="email" placeholder="Email" required>
    <div class="password-input-container">
      <input type="password" id="new_password" name="password" placeholder="Password" required oninput="validatePassword()">
      <button type="button" class="hint-button">
        <span class="iconify" data-icon="mdi:information-outline" data-inline="false"></span>
      </button>
      <input id="password-hint" class="password-hint">At least 8 characters, 1 uppercase, 1 lowercase, 1 number, 1 special character/</input>
    </div>
    <span id="strength-indicator" class="strength-indicator">Password is not strong enough</span>
    <input type="password" id="confirm_password" name="confirm_password" placeholder="Confirm Password" required>
    <button type="submit" id="signup-button" disabled=Sign Up</button>
    <span id="password-hint" class="password-hint">At least 8 characters, 1 uppercase, 1 lowercase, 1 number, 1 special character/</span>
  </form>
  <div>Already have an account? <a href="#" onclick="toggleForms()">log in here</a></div>
</div>

```

Fig. 13 Sign Up interface

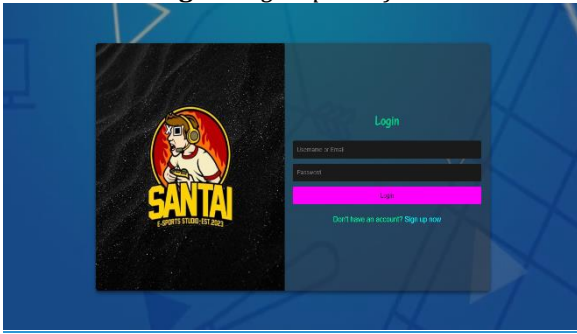


Fig. 15 Login interface

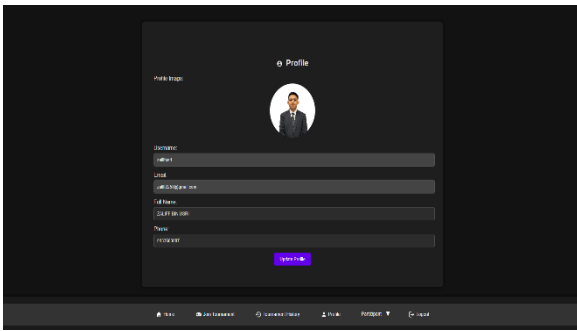


Fig. 17 Manage Profile interface

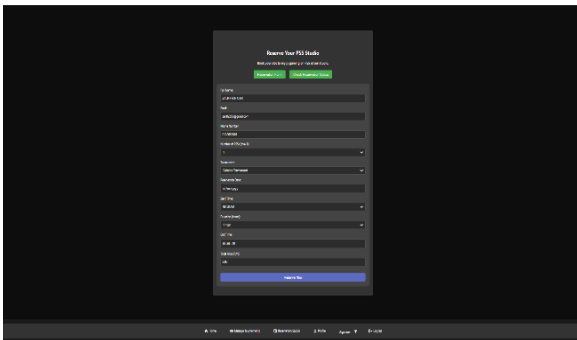


Fig. 19 Manage Reservation interface

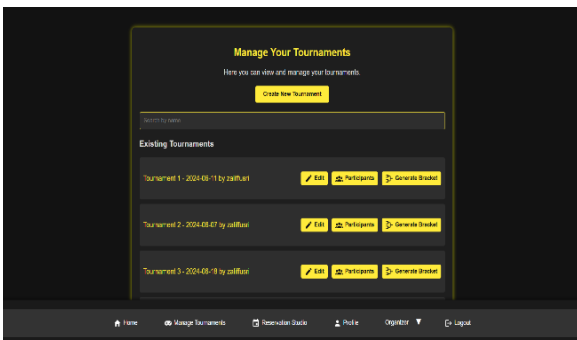


Fig. 21 Manage Tournament interface

Fig. 14 Sign Up code segment

```
<section id="login-form" style="display: none;">
  <h2>Login</h2>
  <form action="authenticate.php" method="post">
    <input type="text" id="username" name="username" placeholder="Username or Email" required>
    <input type="password" id="password" name="password" placeholder="Password" required>
    <button type="submit">Login</button>
    <p>Don't have an account? <a href="#" onClick="toggleForms()">Sign up now</a></p>
  </form>
</section>
```

Fig. 16 Login code segment

```
<script id="profile-section" class="section">
  <div class="profile-section">
    <div class="profile-info">
      <div class="profile-image">
        <img alt="Profile Image" data-id="profile-image" />
      </div>
      <div class="profile-name">
        <input type="text" id="username" name="username" value="" />
      </div>
      <div class="profile-email">
        <input type="text" id="email" name="email" value="" />
      </div>
      <div class="profile-full-name">
        <input type="text" id="full-name" name="full-name" value="" />
      </div>
      <div class="profile-phone">
        <input type="text" id="phone" name="phone" value="" />
      </div>
      <div class="profile-address">
        <input type="text" id="address" name="address" value="" />
      </div>
      <button type="submit">Update Profile</button>
    </div>
  </div>
</script>
```

Fig. 18 Manage Profile code segment

```
function showDetails(reservation) {
  document.getElementById('modal-name').innerText = reservation.full_name;
  document.getElementById('modal-email').innerText = reservation.email;
  document.getElementById('modal-phone').innerText = reservation.phone_number;
  document.getElementById('modal-date').innerText = reservation.reservation_date;
  document.getElementById('modal-start-time').innerText = reservation.start_time;
  document.getElementById('modal-duration').innerText = reservation.duration + " hours";
  document.getElementById('modal-end-time').innerText = reservation.end_time;
  document.getElementById('modal-tournament-id').innerText = reservation.tournament_id;
  document.getElementById('modal-total-price').innerText = "RM " + reservation.total_price;
  document.getElementById('modal-status').innerText = reservation.status;
  document.getElementById('modal-comments').innerText = reservation.admin_comments;

  document.getElementById('details-modal').style.display = 'block';
}
```

Fig. 20 Manage Reservation code segment

```
<script id="manage-tournaments">
  <div class="manage-tournaments">
    <div class="create-new-tournament">
      <button type="button" class="btn btn-primary">Create New Tournament</button>
    </div>
    <div class="existing-tournaments">
      <table border="1">
        <thead>
          <tr>
            <th>Tournament ID</th>
            <th>Date</th>
            <th>Name</th>
            <th>Format</th>
            <th>Participants</th>
            <th>Price</th>
            <th>Start Time</th>
            <th>Description</th>
            <th>Poster</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Tournament 1 - 2024-08-10 by zayn.ah</td>
            <td><input type="button" value="Edit" /><input type="button" value="Participate" /><input type="button" value="Generate Bracket" /></td>
          </tr>
          <tr>
            <td>Tournament 2 - 2024-08-07 by azzfian</td>
            <td><input type="button" value="Edit" /><input type="button" value="Participate" /><input type="button" value="Generate Bracket" /></td>
          </tr>
          <tr>
            <td>Tournament 3 - 2024-08-11 by azzfian</td>
            <td><input type="button" value="Edit" /><input type="button" value="Participate" /><input type="button" value="Generate Bracket" /></td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</script>
```

Fig. 22 Manage Tournament code segment

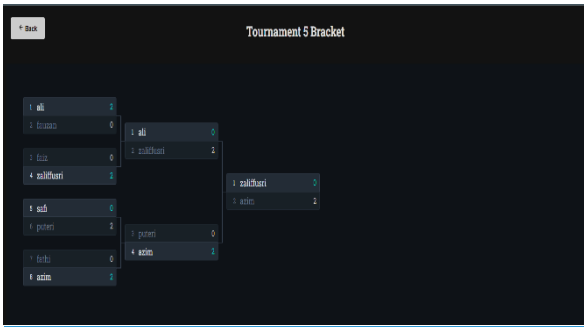


Fig. 23 Bracket interface

```

<div class="theme-dark">
  <div class="bracket disable-image">
    <div class="tournament">
      <div class="match">
        <div class="match-top-team">
          <div class="img"></div>
          <div class="name"></div>
          <div class="score"></div>
        </div>
        <div class="match-bottom-team">
          <div class="img"></div>
          <div class="name"></div>
          <div class="score"></div>
        </div>
        <div class="match-lines">
          <div class="line-one"></div>
          <div class="line-two"></div>
        </div>
        <div class="match-lines-all">
          <div class="line-one"></div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Fig. 24 Bracket code segment

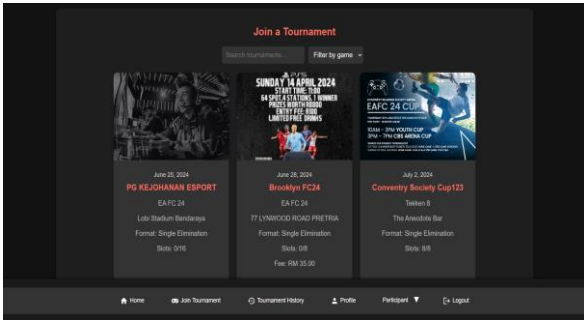


Fig. 25 Join Tournament interface

```

<div class="tournament-card">
  
    <div class="tournament-date"></div>
    <div class="tournament-game"></div>
    <div class="tournament-venue"></div>
    <div class="tournament-format"></div>
    <div class="tournament-price"></div>
  </div>
  <div class="tournament-actions">
    <div class="join-tournament-button"></div>
  </div>
</div>

```

Fig. 26 Manage Profile code segment

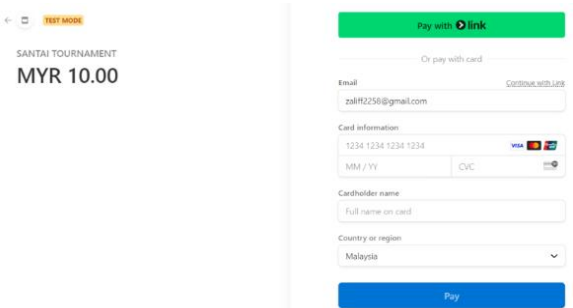


Fig. 27 Manage Payment interface

```

<div class="payment-form">
  <div class="card-info">
    <div class="card-number"></div>
    <div class="card-expiry"></div>
    <div class="card-cvc"></div>
  </div>
  <div class="cardholder-info">
    <div class="cardholder-name"></div>
    <div class="country-region"></div>
  </div>
  <div class="payment-button">
    <div class="pay-button"></div>
  </div>
</div>

```

Fig. 28 Manage Payment code segment

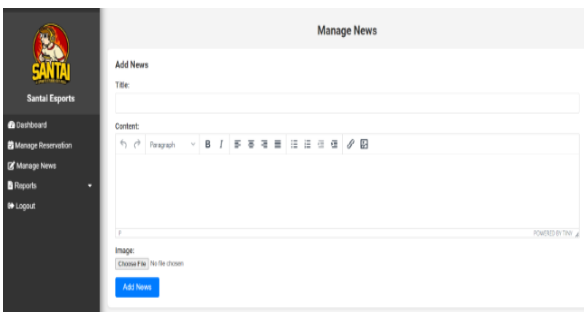


Fig.29 Manage News interface

```

<div class="form-container">
  <div class="add-news-button"></div>
  <div class="form">
    <div class="form-group">
      <div class="title-label"></div>
      <div class="title-input"></div>
    </div>
    <div class="form-group">
      <div class="content-label"></div>
      <div class="content-textarea"></div>
    </div>
    <div class="form-group">
      <div class="image-label"></div>
      <div class="image-input"></div>
    </div>
    <div class="submit-button"></div>
  </div>
</div>

```

Fig. 30 Manage News code segment



Fig. 31 Certificate interface

```

<div class="certificate">
  <div class="certificate-image"></div>
  <div class="certificate-text">
    <div class="certify-text"></div>
    <div class="participant-name"></div>
    <div class="event-name"></div>
    <div class="event-date"></div>
    <div class="participant-role"></div>
  </div>
</div>

```

Fig. 32 Generate Certificate code segment

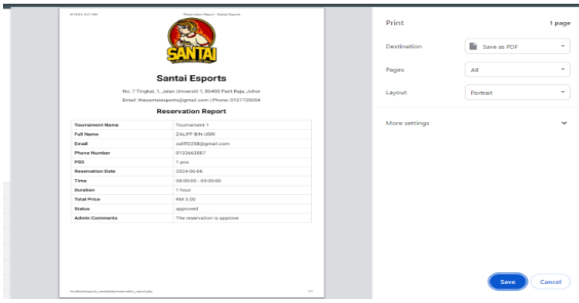


Fig. 33 Report interface

```
function printTournament($id) {
    const $tournament = $tournaments->find($id);
    if (!$tournament) return;

    const printContent = function() {
        <div class="print-report">
            <div style="text-align: center; margin-bottom: 20px;">
                
                <h3 style="margin: 0;">Santai Esports</h3>
                <p style="font-size: 0.8em; margin: 0;">Jl. Pongkor, 1, Jalan Universitas 1, BKM8 Perif Risy, Bogor</p>
                <p style="font-size: 0.8em; margin: 0;">☎️0811-11630101 | 📧thesantai.esports@gmail.com | 📞Phone: 01277326542</p>
            </div>
            <table style="width: 100%; border-collapse: collapse; margin-top: 20px;">
                <tr>
                    <td style="text-align: left; padding: 5px; background-color: #f2f2f2;">Name</td>
                    <td style="padding: 5px;">{$tournament->name}</td>
                </tr>
                <tr>
                    <td style="text-align: left; padding: 5px; background-color: #f2f2f2;">Venue</td>
                    <td style="padding: 5px;">{$tournament->venue}</td>
                </tr>
                <tr>
                    <td style="text-align: left; padding: 5px; background-color: #f2f2f2;">Participant</td>
                    <td style="padding: 5px;">{$tournament->participants}</td>
                </tr>
                <tr>
                    <td style="text-align: left; padding: 5px; background-color: #f2f2f2;">Total Price (Rp)</td>
                    <td style="padding: 5px;">{$tournament->total_price}</td>
                </tr>
            </table>
        </div>
    };
    printContent();
}
```

Fig. 34 Generate Report code segment

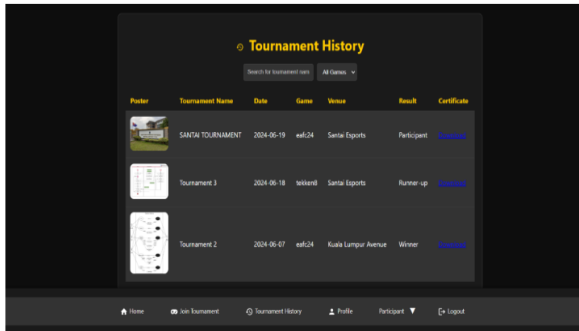


Fig. 35 Access History interface

```
<tr class="clickable-row" data-name="{$row['name']}" data-game="{$row['game']}">
    <td style="text-align: center; width: 100px;"></td>
    <td style="width: 200px;">{$row['name']}</td>
    <td style="width: 100px;">{$row['date']}</td>
    <td style="width: 100px;">{$row['game']}</td>
    <td style="width: 100px;">{$row['venue']}</td>
    <td style="width: 100px;">{$row['result_text']}</td>
    <td style="width: 100px; text-align: center;"><a href="generate_certificate.php?tournament_id={$row['id']}" class="btn btn-download">Download</a></td>
</tr>
```

Fig. 36 Access History code segment

5.1 Test Case

The test cases are divided into 11 categories, corresponding to each module of the system. The functionalities tested are derived from the functional requirements outlined in Section 4. Ensuring comprehensive coverage and verification of all specified requirements. This systematic approach helps track whether each requirement has been met. The detailed test case and the results are presented in Tables 8 through Table 18.

Table 8 Test Case for sign up

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_100	TEST_100	Sign Up	
SRS_REQ_101	TEST_101	The system shall allow organizer and participant to create new account by entering username, email, password and password confirmation	Pass
SRS_REQ_102	TEST_102	The system shall redirect the organizer and participant to the login page after the sign up successful.	Pass

Table 9 Test Case for Login

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_200	TEST_200	Login	
SRS_REQ_201	TEST_201	The system shall verify user credentials for access to the system.	Pass
SRS_REQ_202	TEST_202	The system shall allow redirect to their specified homepage or dashboard based on the roles of the user after successful login	Pass
SRS_REQ_203	TEST_203	The system shall allow the user to logout of the system after the session.	Pass

Table 10 Test Case for *Manage Profile*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_300	TEST_300	Manage Profile	
SRS_REQ_301	TEST_301	The system shall allow the organizer and participant to manage their profile such as update their personal information like profile image, name and phone number.	Pass
SRS_REQ_302	TEST_302	The system shall allow the organizer and participant to view their profile.	Pass

Table 11 Test Case for *Manage Studio Reservation*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_400	TEST_400	Manage Studio Reservation	
SRS_REQ_401	TEST_401	The system shall allow the organizer to reserve the Santai Esports studio for their tournament.	Pass
SRS_REQ_402	TEST_402	The system shall allow the organizer to see the status of the reservation.	Pass
SRS_REQ_403	TEST_403	The system shall allow the administrator to approve the reservation from the organizer.	Pass
SRS_REQ_403	TEST_404	The system shall allow the administrator to reject the reservation from the organizer.	Pass

Table 12 Test Case for *Manage Tournament*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_500	TEST_500	Manage Tournament	
SRS_REQ_501	TEST_501	The system shall allow the organizer to create tournament.	Pass
SRS_REQ_502	TEST_502	The system shall allow the organizer to edit the details about the tournament.	Pass
SRS_REQ_503	TEST_503	The system shall allow the organizer to shuffle the slow of the participants in the tournament.	Pass
SRS_REQ_504	TEST_504	The system shall allow the organizer to generate single elimination match bracket for the tournament.	Pass
SRS_REQ_505	TEST_505	The system shall allow the organizer to generate double elimination match bracket for the tournament.	Fail
SRS_REQ_506	TEST_506	The system shall allow the organizer to update the match in the bracket of the tournament.	Pass

Table 13 Test Case for *Registration Tournament*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_600	TEST_600	Registration Tournament	
SRS_REQ_601	TEST_601	The system shall enable participant to join the tournaments.	Pass
SRS_REQ_601	TEST_602	The system shall allow participant to pay registration fees if needed.	Pass

Table 13 Test Case for *Registration Tournament (cont.)*

SRS_REQ_601	TEST_603	The system shall allow the participant to see the progress of the tournament that they have join.	Pass
SRS_REQ_601	TEST_604	The system shall not allow the participant to join the same tournament multiple times.	Pass

Table 14 Test Case for *Manage Payment*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_700	TEST_700	Manage Payment	
SRS_REQ_701	TEST_701	The system shall allow participant to pay the registration fee.	Pass
SRS_REQ_701	TEST_702	The system shall allow organizer to distribute prize payment to the participant.	Pass
SRS_REQ_701	TEST_703	The system shall allow organizer pay for the reservation studio.	Pass

Table 15 Test Case for *Manage News*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_800	TEST_800	Manage News	
SRS_REQ_801	TEST_801	The system shall allow the administrator to post the news.	Pass
SRS_REQ_802	TEST_802	The system shall allow the administrator to edit the news.	Pass
SRS_REQ_803	TEST_803	The system shall allow the administrator to delete the news.	Pass
SRS_REQ_804	TEST_804	The system shall allow the organizer and participant to see the details of the news.	Pass

Table 16 Test Case for *Generate Certificate*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_900	TEST_900	Generate Certificate	
SRS_REQ_901	TEST_901	The system shall allow participant to download the certificate that tournament they have joined.	Pass

Table 17 Test Case for *Generate Report*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_1000	TEST_1000	Generate Report	
SRS_REQ_1001	TEST_1001	The system shall allow the administrator to generate all the reservation report.	Pass
SRS_REQ_1002	TEST_1002	The system shall allow the administrator to generate all the tournament report.	Pass
SRS_REQ_1003	TEST_1003	The system shall allow the organizer to generate their tournament report.	Pass

Table 18 Test Case for *Access History*

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass/Fail)
SRS_REQ_1100	TEST_1100	Access History	
SRS_REQ_1101	TEST_1101	The system shall allow the organizer to access the history of the tournament that they have created.	Pass
SRS_REQ_1102	TEST_1102	The system shall allow the participant to access the history of the tournament that that they have joined.	Pass

5.2 Overall Test Case Result

The Santai Esports Tournament Management System is a web-based tournament management for esports. It makes joining tournaments easy and keeps everything organized. It helps players sign up, organizers run events, and admins manage everything smoothly. With features like easy registration, tournament planning tools, and money management, it makes esports more enjoyable for everyone involved. This system aims to make gaming competitions simpler and more fun. Table 19 shows the overall test cases and Figure 37 shows the overall test case result.

Table 19 Overall Test Cases

Test CASE ID	Total Test Cases	Total Passes
TEST_100	2	2
TEST_200	3	3
TEST_300	2	2
TEST_400	4	4
TEST_500	6	5
TEST_600	4	4
TEST_700	3	3
TEST_800	4	4
TEST_900	1	1
TEST_1000	3	3
TEST_1100	2	2
Total	34	33

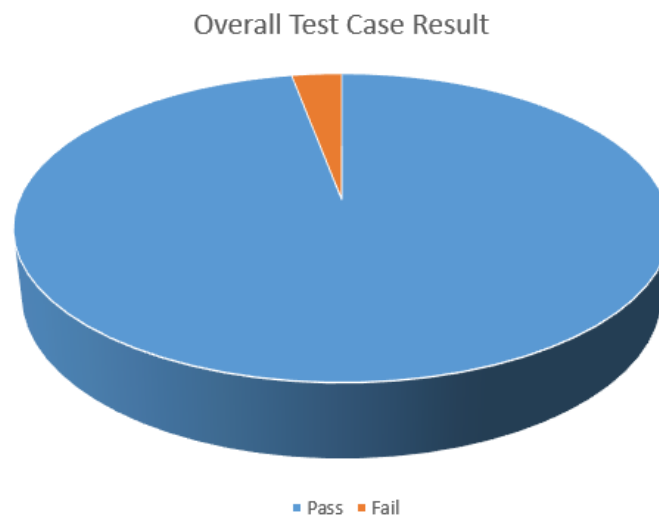


Fig. 37 Overall Test Case Result

5.3 User Acceptance Testing

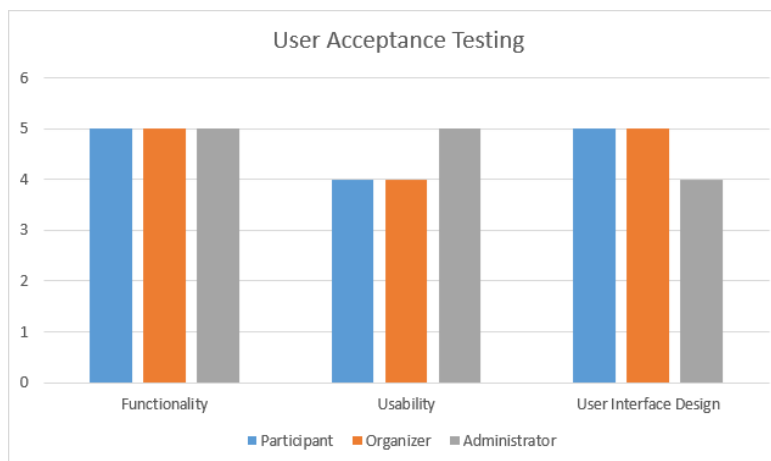


Fig. 38 User Acceptance Testing Result

Figure 38 shows the bar chart presents the user acceptance testing results for three categories: Functionality, Usability, and User Interface Design, evaluated by three user roles: Participant, Organizer, and Administrator. For Functionality, all three roles rated the system highly, giving it a score of 5, indicating that the system meets the required functional specifications effectively. In terms of Usability, Participants and Organizers gave it a score of 4, while Administrators rated it slightly higher at 5, suggesting a generally user-friendly experience with some room for improvement. For User Interface Design, Participants and Organizers again gave it high scores of 5, reflecting a positive perception of the design's aesthetics and intuitiveness, while Administrators rated it a bit lower at 4. This analysis highlights the system's strong functionality and appealing user interface, with usability being well-regarded but with potential areas for enhancement.

6. Conclusion

The Santai Esports Tournament Management System provides a comprehensive solution for managing esports tournaments, significantly enhancing the processes of organization and execution. By integrating automation, accessibility, and centralized management, the system addresses many of the challenges faced by tournament organizers. Its user-friendly web-based interface allows users to access the system from anywhere with an internet connection, making it a versatile tool for both participants and organizers.

While the system presents numerous advantages, such as the automated generation of tournament brackets and the centralization of studio reservations, it is not without its limitations. Challenges such as dependency on a stable internet connection and potential performance issues during high traffic periods can hinder the user experience. Furthermore, the initial setup requires a certain level of technical knowledge, which may deter some users. Security remains a concern as well, given the risks associated with web-based applications.

To maximize the system's effectiveness and utility, it is essential to address these limitations. Implementing offline capabilities could significantly enhance accessibility for users in areas with poor connectivity. Providing detailed documentation and user-friendly guides will facilitate the initial setup, ensuring that all users can navigate the system with ease. Additionally, enhancing scalability will allow the system to handle increased traffic more efficiently, while regular updates can mitigate security vulnerabilities.

In conclusion, the Santai Esports Tournament Management System has the potential to be an invaluable resource in the esports industry. By continuously improving its features and addressing user feedback, the system can evolve to better meet the needs of its users and maintain its relevance in an ever-changing landscape.

Acknowledgement

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

References

- [1] Hamari, J. and Sjöblom, M. (2017), "What is eSports and why do people watch it?", *Internet Research*, Vol. 27 No. 2, pp. 211-232. <https://doi.org/10.1108/IntR-04-2016-0085>
- [2] P. Warr, "eSports in numbers: Five mind-blowing stats," *Red Bull*, Jul. 27, 2016. <https://www.redbull.com/my-en/esports-in-numbers-five-mind-blowing-stats>
- [3] D. Lee and L. J. Schoenstedt, "Comparison of eSports and traditional sports consumption motives," *ICHPER-SD Journal of Research*, vol. 6, no. 2, pp. 39-44, 2011. [Online]. Available: <https://eric.ed.gov/?id=EJ954495>.
- [4] Newzoo, "Global Esports Market Report 2020," 2020. [Online]. Available: <https://newzoo.com/insights/articles/newzoos-global-esports-market-report-2020-light-version>. [Accessed: 12-Jul-2024].
- [5] Emergen Research, "Esports Market Size, Share, Growth & Trends Report," 2020. [Online]. Available: <https://www.emergenresearch.com/industry-report/esports-market>. [Accessed: 12-Jul-2024].
- [6] R. Joslin and R. Müller, "Relationships between a project management methodology and project success in different project governance contexts," *International Journal of Project Management*, vol. 33, no. 6, pp. 1377-1392, Aug. 2015. <https://doi.org/10.1016/j.ijproman.2015.03.005>.
- [7] Shanti Bhushan Mishra, *Handbook of Research Methodology*. Educreation Publishing, 2017.
- [8] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61-72, May 1988, doi: <https://doi.org/10.1109/2.59>.
- [9] G. Rajendra, A. Sabale, and Dani, "Comparative Study of Prototype Model For Software Engineering With System Development Life Cycle," *IOSR Journal of Engineering (IOSRJEN)*, vol. 2, no. 7, pp. 21-24, 2012.