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

AITCS

Vol. 6 No. 2 (2025) 625-643
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Swimming Facilities Management System for Universiti Tun Hussien Onn Malaysia Sport Centre with Two-Factor Authentication

Muhamad Danial Shahrom[1], Khairul Amin Mohamad Sukri[1*]

[1] *Faculty of Science Computer and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: khairulm@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.034

Article Info

Received: 21 October 2025
Accepted: 19 November 2025
Available online: 30 November 2025

Keywords

Online Booking System, Two-Factor Authentication, One-Time Password (OTP), Access Control, Role-Based Access Control (RBAC)

Abstract

The Swimming Facilities Management System for Universiti Tun Hussein Onn Malaysia (UTHM) Sports Centre with Two-Factor Authentication was developed to address inefficiencies in the management of swimming pool facilities and improve the overall operations of the Sports Centre. The existing manual process such as an inefficient booking process for swimming pool activities, a poor data management for users and an inefficient payment process for walk-in users, have proven to be time-consuming and prone to errors. To resolve these issues, the project adopts the Waterfall methodology, which provides a structured, sequential approach with a focus on clear documentation and well-defined requirements at each stage of development. The project aims to design a proposed system by implementing Role-Based Access Control, developing secure management system with Two-Factor Authentication and testing the functionality of the developed system. The outcome is a user-friendly interface that allows students, staff, and visitors to efficiently manage bookings, process payments, and access services, while securely handling user data.

1. Introduction

The Swimming Facilities Management System for UTHM Sports Centre is a newly developed management platform designed to replace the current system and streamline operations at UTHM's swimming facilities. Currently, UTHM Sport Centre manages its swimming facilities through two methods which are a walk-in option for students and visitors who want to use the swimming pool for swimming and a written reservation for those who want to book for activities like tournaments or swimming classes.

Although the current management system is working, several issues have been identified that highlight the problem statement. One major issue is the inefficient booking process. Users who wanted to book the swimming pool for activities like competitions had to make written reservations. Another problem was poor data management. Walk-in users did not need to register to use the pool, resulting in a lack of record-keeping for facility use. The third problem faced by UTHM Sport Centre was inefficient payment process. Walk-in users require to pay with a debit card leading to long queues and congestion at the counter, especially during peak hours.

This project aims to design a Swimming Facilities Management System for UTHM Sports Centre combined with a central database and management securely with Role Based Access Control. It has been decided that to develop the secure management system with two-factor authentication for UTHM Sport Centre using web-based

approach. The system will be properly tested with all the functionality to make the swimming facilities management easy and reliable.

The project scope was designed for UTHM Sports Centre, serves students, staff, and visitors. It includes modules for user registration with role-based access (RBAC), facility bookings, and efficient payment options. Built with PHP, CSS, HTML, and MySQL, it ensures secure data storage and efficient management of bookings. The expected result was the system efficiently manage swimming pool facilities by storing data and information of registered users who utilize the swimming services. The system simplifies user transactions by offering a smooth and effective payment process. Overall, these improvements aim to enhance user satisfaction and attract new users and visitors to enjoy the swimming facilities at UTHM Sports Centre.

2. Literature Review

This section provides a comprehensive review of the literature relevant to the project. It covers topics including online booking systems, authentication mechanisms, access control methods, and an analysis of existing systems. Additionally, it includes a comparative study to evaluate the features, strengths, and limitations of these systems, highlighting their relevance to the proposed solution.

2.1 Online Booking System

Online booking systems are widely available online, offering convenience and efficiency for users in various areas. A prominent example is hotel booking platforms like Booking.com which provide users with diverse options tailored to their preferences. Similarly, government institutions, such as clinics, utilize online booking systems to streamline appointment scheduling, improving service satisfaction for patients. These systems benefit not only large organizations but also small businesses and freelancers, highlighting their universal applicability.

The advantages of online booking systems include time-saving convenience, as users can schedule services without the need to visit a location or wait in queues. As a result, it can enhance user experience as the system can provide. Offers a seamless and user-friendly experience leading to higher customer satisfaction and repeat bookings [1], and by minimizing manual errors, such as double bookings, these systems reduce potential losses and maintain consumer trust. They also contribute to environmental sustainability by reducing the reliance on paper for schedules and tickets, supporting eco-friendly practices.

2.2 Two-Factor Authentication (2FA)

Two-factor authentication is commonly used to strengthen system security and better protect user privacy. It offers an extra layer of defense in addition to traditional authentication methods [2]. Two-factor authentication schemes were developed to combat phishing attacks, although they have proven to be costly to implement in real-time applications [3]. The proposed system implements two-factor authentication (2FA) using a strong password and OTP. This combines something the user knows (password) with something they have (OTP), providing enhanced security by requiring both factors for access.

2.2.1 Strong Password Management

A strong password is typically long, complex, and unique, making it difficult for attackers to guess or crack. It serves as the first line of defense in 2FA, where the user must enter the correct password to gain access. Proper password management involves using passwords that are hard to predict, avoiding password reuse across sites, and regularly updating passwords. Strong passwords are often challenging for users to remember because their high entropy values make them complex and difficult to retain [4]. However, to ensure security, it is important to use strong passwords, even though they may be difficult to remember.

2.2.2 One-Time Password (OTP)

A one-time password is a unique password that can only be used once and is only valid for a single session. This human-centered security solution is frequently utilized for multi-factor authentication. These characteristics have made it well-known for being extremely secure, and it may be applied as an authentication service in information and communication technology (ICT) domains including mobile cloud computing and mobile payments [5]. For the proposed system, the OTP will be delivered via email. This means that when a user logs in, after entering their password, they will receive a unique, time-sensitive code sent to their registered email address.

2.3 Access Control

Access control helps keep data secure by preventing unauthorized users from accessing personal data and private information. It also prevent data leaks or alterations from attackers. Besides the well-known access control models, newer ones have been developed to meet modern security needs [6]. The initial forms of access control were Mandatory Access Control (MAC) and Discretionary Access Control (DAC). However, their limitations led to the development of advanced models, including Role-Based Access Control (RBAC), Dynamic Typed Access Control (DTIC), and Domain Type Enforcement (DTE), designed to meet more sophisticated security needs.

2.3.1 Role Based Access Control (RBAC)

Role-based access control (RBAC), also known as role-based security, is a technique used in computer systems security to limit system access to authorized individuals [7]. It simplifies permission management by grouping users into roles. In the proposed system, the admin is given control over all activities within the system. The admin has the authority to modify, delete, and approve documents. Meanwhile, regular users, such as students, UTHM staff, and visitors, are only granted rights to make bookings, view, and edit their own information.

The primary benefits of RBAC lie in its efficiency and scalability, making it ideal for large organizations with numerous users and complex access requirements [8]. By managing permissions centrally, RBAC ensures consistency and minimizes errors in assigning access rights. Furthermore, it adheres to the principle of least privilege, granting users only the permissions necessary for their specific roles. This minimizes unnecessary access, enhancing overall security.

2.4 Existing System of Swimming Pool Management System

This section will discuss the existing swimming pool management system, which is like the proposed system. The existing system are UTHM Swimming Pools, Seremban City Council Swimming Pools [10] and Greater Chennai Corporation Swimming Pool Booking.

2.4.1 UTHM Swimming Pools

The UTHM swimming pool currently relies on Google Forms for reservations related to tournament activities and swimming classes. For general swimming access, users and students can simply walk into the UTHM swimming centre without needing to register or make a booking. Payments are made exclusively through debit card. The swimming schedule, rules, and fees are available on the website sukan.uthm.edu.my. Additionally, information about payments and swimming centre usage is recorded manually in reports. Once a booking is made through Google Forms, users receive a confirmation email [9].

2.4.2 Seremban City Council Swimming Pools

The Seremban City Council Swimming Pools system, as displayed on their official portal, currently offers detailed information about swimming pool rental rates categorized by day and age group with separate pricing for children and adults on weekdays and weekends. For example, children are charged RM 1.50 on weekdays and RM 2.50 on weekends, while adults pay RM 3.00 on weekdays and RM 3.50 on weekends. Additionally, the portal uses the HTTPS protocol, which helps prevent content tampering or modification and enhances user trust by confirming the site's authenticity and security. However, despite providing clear pricing and operational details, the system does not offer any visible features or guidelines for users to make online reservations or book swimming slots directly through the portal. This lack of an integrated booking function may cause inconvenience for users who wish to secure their preferred time slots in advance. [10].

2.4.3 Greater Chennai Corporation Swimming Pool Booking

The Greater Chennai Corporation Swimming Pool Booking system is an online platform that allows users to make reservations without the need for registration. To book a pool session, users simply visit the homepage, where they must enter their name, phone number, gender, date, time, and the number of adults and children. After filling in these details, users can confirm their booking by pressing the "Confirm Booking" button. Then, a pop-up window that allows users to confirm their booking before proceeding to the payment process. Once the user is satisfied with their booking details, they need to click on the "Pay Now" button to continue. The payment process, which offers multiple payment options including UPI, cards, and net banking. Additionally, users are required to enter their phone number to complete and confirm the payment [11]. Additionally, the platform uses input validation to guarantee the integrity and accuracy of data submitted by users, lowering the possibility of mistakes or misuse. Furthermore, the system protects sensitive user and payment information by using the HTTPS protocol to encrypt data transmission.

2.4.4 Comparison Existing System with the Proposed System

Table 1 presents a comparison between existing swimming pool booking systems, such as the Greater Chennai Corporation Swimming Pool Booking, Seremban City Council Swimming Pools, and UTHM Swimming Pool, with the proposed system, which is the Swimming Facilities Management System for UTHM Sport Center.

Table 1 *Comparison between the existing system with the proposed system*

Features/ Module	UTHM Swimming Pool System	Seremban City Council Swimming Pool	Greater Chennai Corporation Swimming Pool Booking	Proposed System
Type of system	Google form	Informational System	Online System	Online System
Login	No	No	No	Yes
Sign-up	No	No	No	Yes
Booking	No	No	Yes	Yes
Payment	No	No	Yes	Yes
Input Validation	No	No	Yes	Yes
Audit Trail	No	No	Information not available	Yes
RBAC	No	No	No	Yes
Hashing and Salt	No	No	No	Yes
Forgot Password	No	No	No	Yes
reCAPTCHA	No	No	No	Yes
Two-Factor Authentication	No	No	No	Yes

3. Methodology

This section outlines the methodology used to develop the Swimming Facilities Management System for UTHM Sports Centre, focusing on a structured approach to ensure efficient development and high-quality outcomes. The chosen approach is the Waterfall Model, which follows a structured, step-by-step process with clearly defined stages. The section concludes by highlighting the methodology's importance in meeting the project's objectives and ensuring an adaptable and effective system development.

3.1 Waterfall model

The Waterfall model in Fig. 1 is a widely recognized type of software development process. Proposed by Winston W. Royce in 1970, it has become one of the most iconic frameworks in software engineering [12]. The Waterfall model is a Software Development Life Cycle (SDLC) framework with five sequential stages which are requirement, design, implementation, testing, and maintenance. Each stage must be completed before progressing to the next, following a structured, linear approach [13]. This approach was used to provide a clear and methodical process for the Swimming Facilities Management System. The requirement phase involved gathering information about user needs as well as system functionality including booking, payments, and role-based access. During the design phase, system architecture, database structure, and user interfaces were designed using diagrams such as ERD and DFDs. The implementation phase concentrated on developing the system using PHP, HTML, CSS, and MySQL in accordance with the design specs. During the testing phase, the system was validated through functional and user testing to confirm that it met requirements and worked properly. Finally, following deployment, the maintenance phase handles bug repairs to ensure the system's reliability and usability.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 625-643

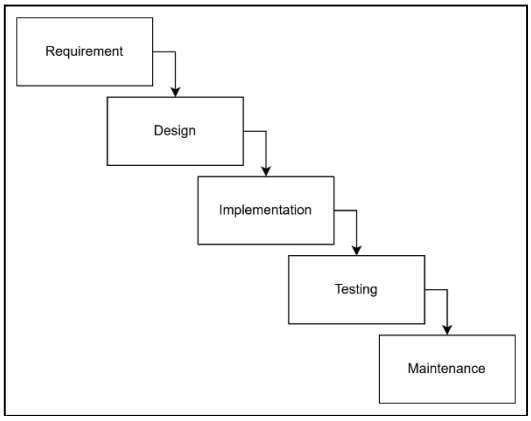


Fig. 1 *Waterfall model*

3.2 Requirement phase

The Waterfall model is a linear software development process, starting with requirements analysis and progressing through design, coding, and testing [14]. This phase involves engaging with stakeholders to gain insight into their expectations, challenges, and specific needs. For the Swimming Facilities Management System, methods such as interviews, surveys, and observations were used to identify key issues.

Discussions with UTHM Sports Centre management through WhatsApp highlighted the current challenges and areas for improvement. These include the manual booking process and lack of systematic data management. A visit to UTHM pool found inefficient payment methods for walk-in users. The analysis identified three main user needs which are an automated booking process, a user registration system for better data management, and an efficient payment system. These requirements form the foundation for the system's design and development, ensuring it addresses the center's challenges and meets its objectives.

3.3 Designing phase

The Design Phase is a crucial step in the Software Development Life Cycle (SDLC), as it shapes the architecture and functionality of the system. During this phase, the system's architecture, network infrastructure, databases, user interfaces, and system interfaces are carefully designed to meet both functional and non-functional requirements. Therefore, the design process necessitates a thorough analysis to understand user requirements and tailor the system design to meet those needs effectively [15]. The system architecture for the proposed system was designed to give an idea of how the system works. This includes the structure of the database, the communication between front end and back-end components, and how data flows through the system. This phase lays the foundation for building a system that is both functional and user-friendly, ensuring it meets user needs and expectations.

3.4 Implementation phase

The implementation phase, also known as the programming phase, involves transforming the designed system into functional program code using a suitable programming language [16]. The system is built using HTML for structure, CSS for design, JavaScript for interactivity, and PHP for database integration, ensuring a dynamic, visually appealing, and user-friendly platform.

3.5 Testing phase

The testing phase is a critical stage where the system developed by the developers is thoroughly verified and validated by a dedicated team, known as system analysts or web application testers [17]. This phase involves rigorous testing, including functional testing to verify that all system modules operate correctly, user acceptance testing to confirm the system meets user expectations, and security testing to safeguard against unauthorized access and cyber threats. For the Swimming Facilities Management System, critical features such as the booking process, payment methods, and user authentication are tested thoroughly. Any errors or vulnerabilities identified during testing are resolved to ensure the system is reliable, user-friendly, and secure before deployment.

3.6 Maintenance phase

According to the Waterfall model, the Maintenance Phase is the last phase of the software development lifecycle and handles making sure the program keeps working correctly after it has been deployed. It includes fixing any flaws or problems users might get into as well as providing updates and improvements. Furthermore,

security flaws are fixed as soon as they are discovered, and compatibility updates are performed to guarantee that the program functions.

4. Analysis and Design

This section includes several key components essential for the system's development. It begins with system requirements, followed by system analysis, system architecture, entity relationship diagram and interface design.

4.1 System Requirements

System requirements outline the essential capabilities the proposed system must achieve to meet user needs and expectations. This process ensures the system functions as intended and aligns with the project objectives. Table 2 and Table 3 show the functional requirements detailing the system's tasks, non-functional requirements emphasizing qualities like usability, performance, and security.

Table 2 *Functional Requirements*

Module	Functionality
Sign Up	Allow new users to register by providing personal details.
Login	Allow users to log in using their registered email and password.
User	Manage user profiles, including updating personal details and viewing booking.
Booking	Enable users to book swimming facilities.
Payment	Facilitate efficient payment methods
Admin Dashboard	Provide an admin interface for managing users, bookings and payments.
Security	Encrypt passwords using hashing and salting techniques for secure storage.
Schedule	Display facility schedules, including pool availability and class/tournament timings

Table 3 *Non-Functional Requirements*

Requirement	Description
Usability	The system interface should be intuitive and user-friend.
Security	The system must protect user data through encryption.
Error Handling	The system should provide meaningful error messages and handle exceptions gracefully without crashes.
Operational	The system can be accessed when connected to an internet connection.

4.2 System Analysis

System analysis involves breaking down the system into components to understand their roles in achieving objectives. This phase uses models like the Data Flow Diagram (DFD), context diagram, and system flowchart to identify potential issues and refine the system's design.

4.2.1 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a modelling tool that emphasizes the fundamental functional aspects of early software design. It illustrates how data moves from its origin to its endpoint, showing how it is processed and transformed as it progresses through different stages of the system. DFDs help understand how different parts of a system interact with each other and identify any issues, ensuring the system works efficiently. Fig.. 2 shows Data Flow Diagram.

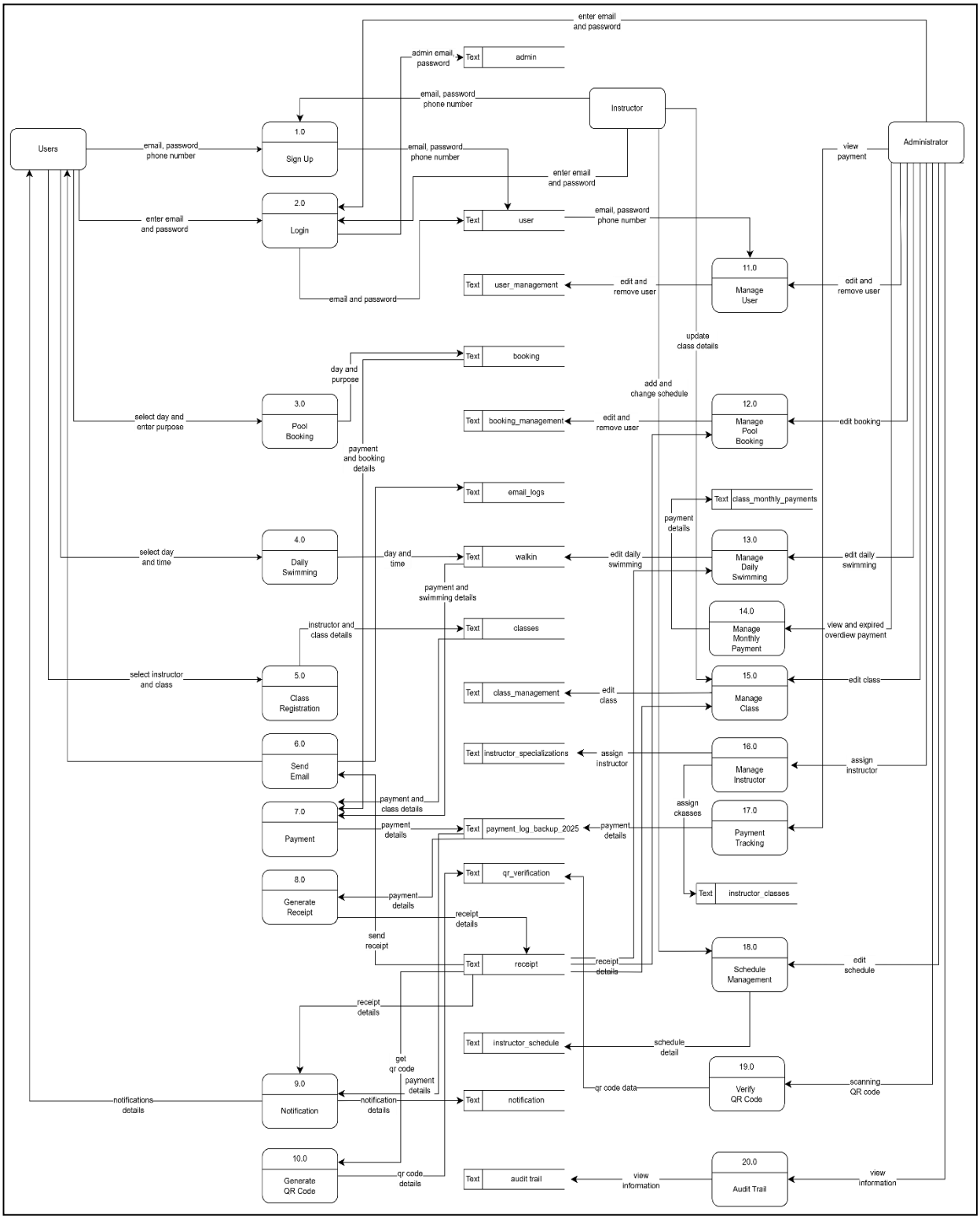


Fig. 2 *Data Flow Diagram*

4.2.2 System Flowchart

A system flowchart is a visual tool that shows the sequence of steps or processes within a system. It provides a clear, step-by-step view of the system's operations, helping to identify how information flows and where decisions are made. Fig. 3 presents the system flowchart for the Swimming Facilities Management System at UTHM Sport Center.

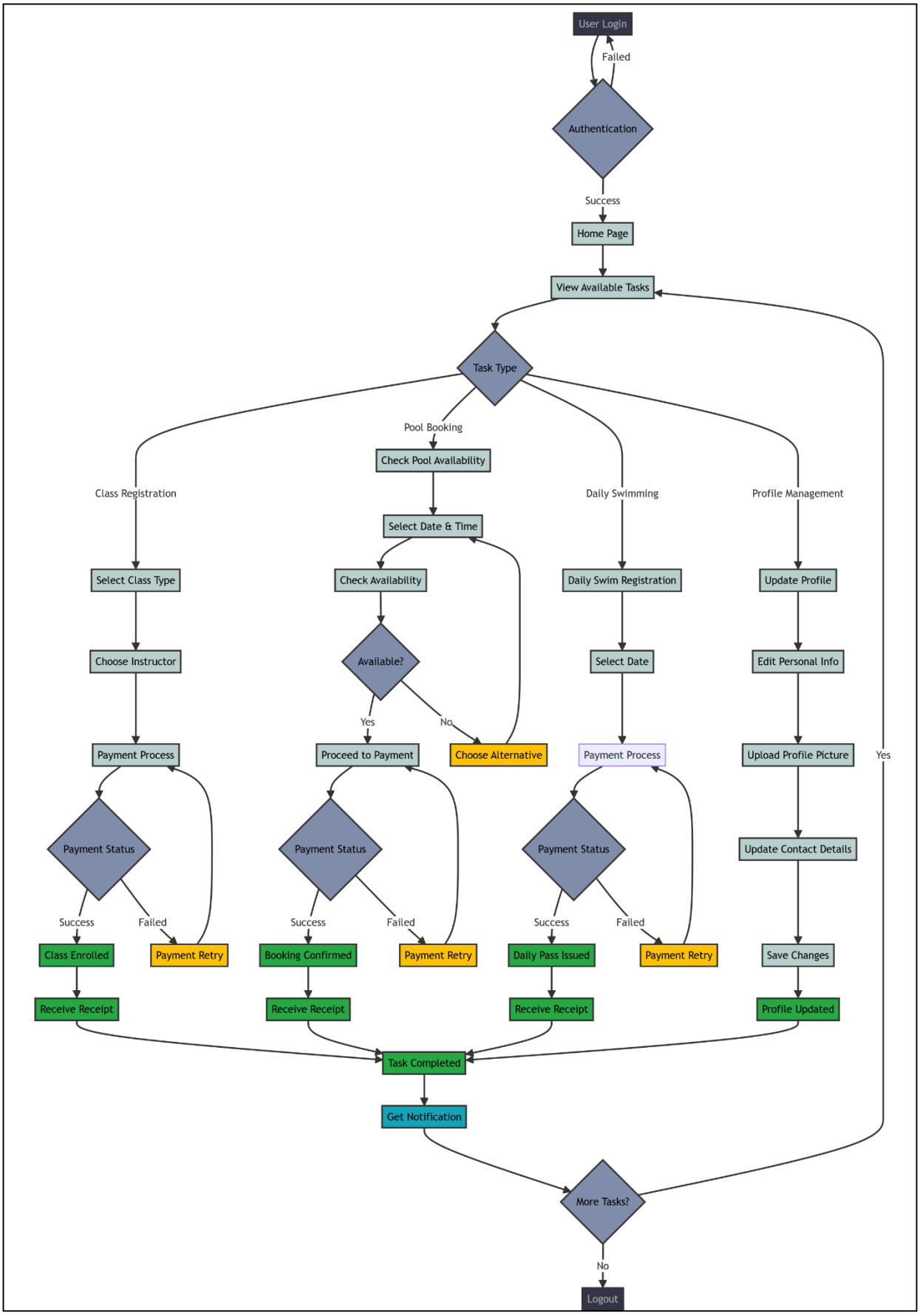


Fig. 3 *System Flowchart*

4.2.3 Entity Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) is an essential part of database design, visually representing the system's data entities and their relationships. For the Swimming Facilities Management System, the ERD in Fig. 4 shows how entities like users, bookings, registration, and payments are connected. It helps define the database structure, ensuring data is organized logically for efficient access and supporting system functions like booking management, scheduling, and payment processing.

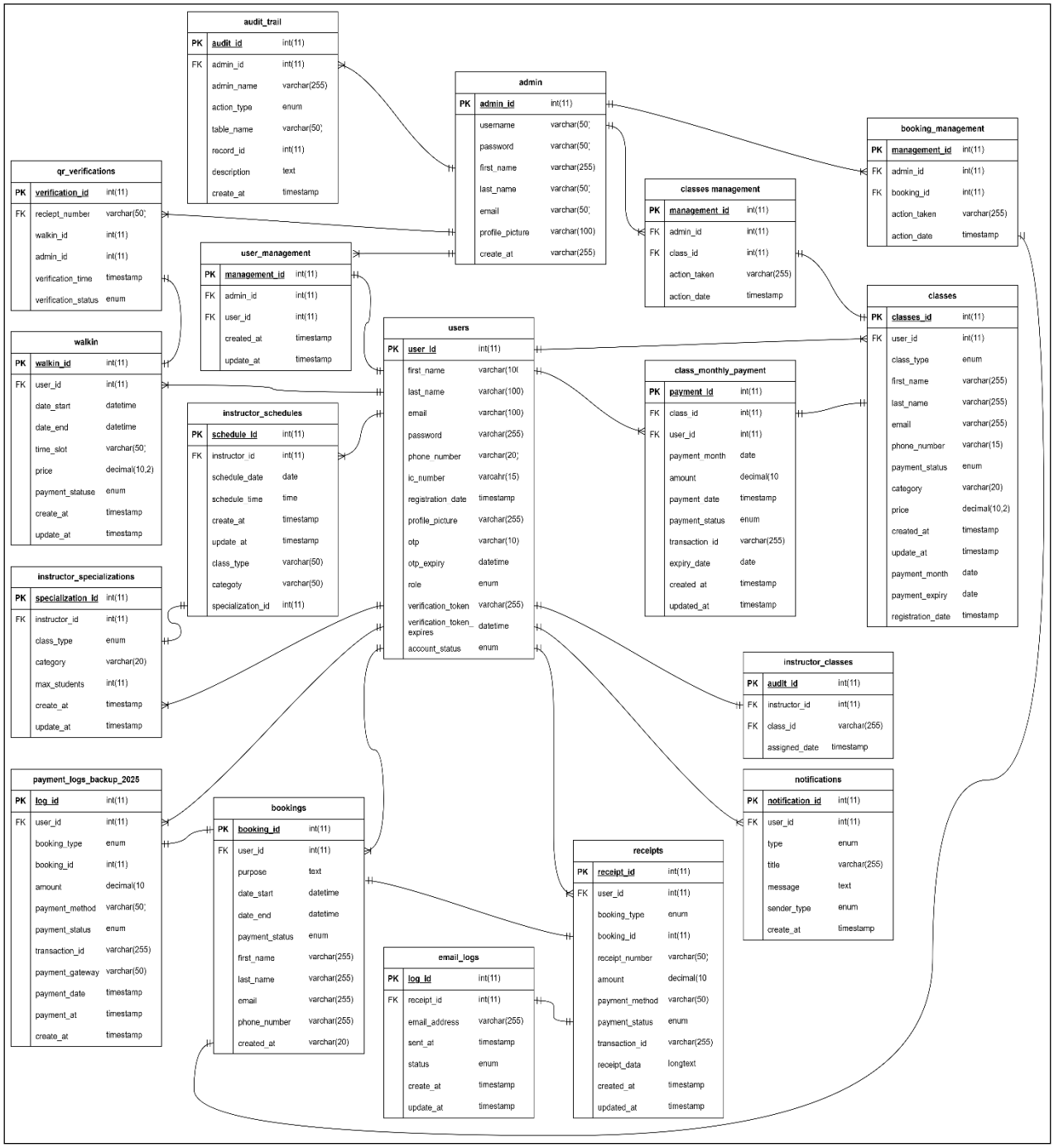


Fig. 4 *Entity Relationship Diagram*

4.2.4 Context Diagram

The context diagram provides a basic overview of the system, showing how data is collected, processed, and presented as output. This diagram offers a clear representation of the system's boundaries, inputs, and outputs, helping to clarify its role and integration with other components. Fig. 5 shows the context diagram for the Swimming Facilities Management System at UTHM Sport Center, emphasizing key external entities and their connections to the system.

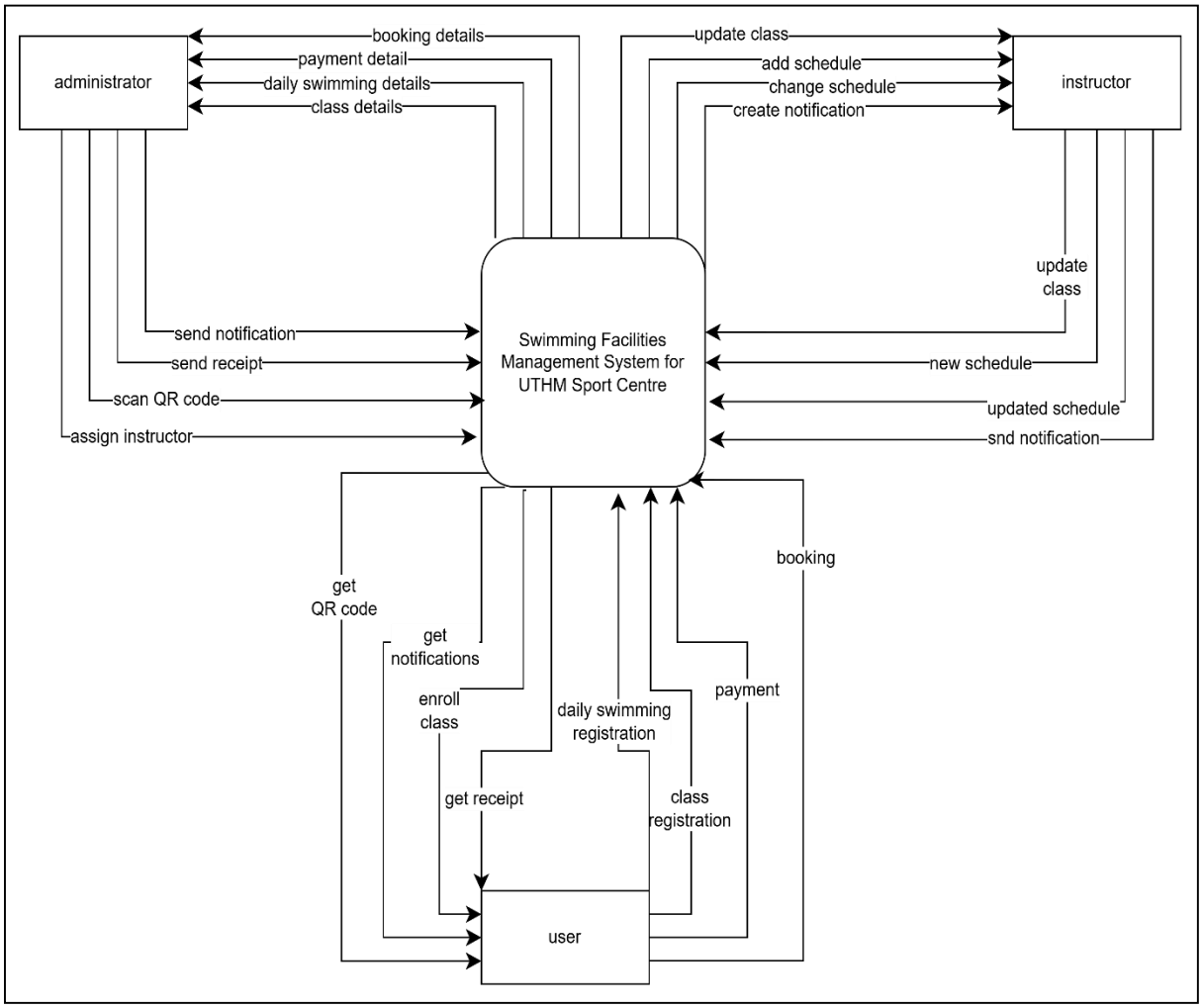


Fig. 5 *Context Diagram*

4.3 System Design

System design is the process of outlining the architecture, components, and interfaces of a system according to the identified requirements. It involves creating and developing a system that meets the user's needs and expectations.

4.4 System Architecture

Fig. 6 shows the system architecture outlines the structural design and interaction between various components of the Swimming Facilities Management System. It ensures seamless communication between users, administrators, and the database through well-defined modules.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 625-643

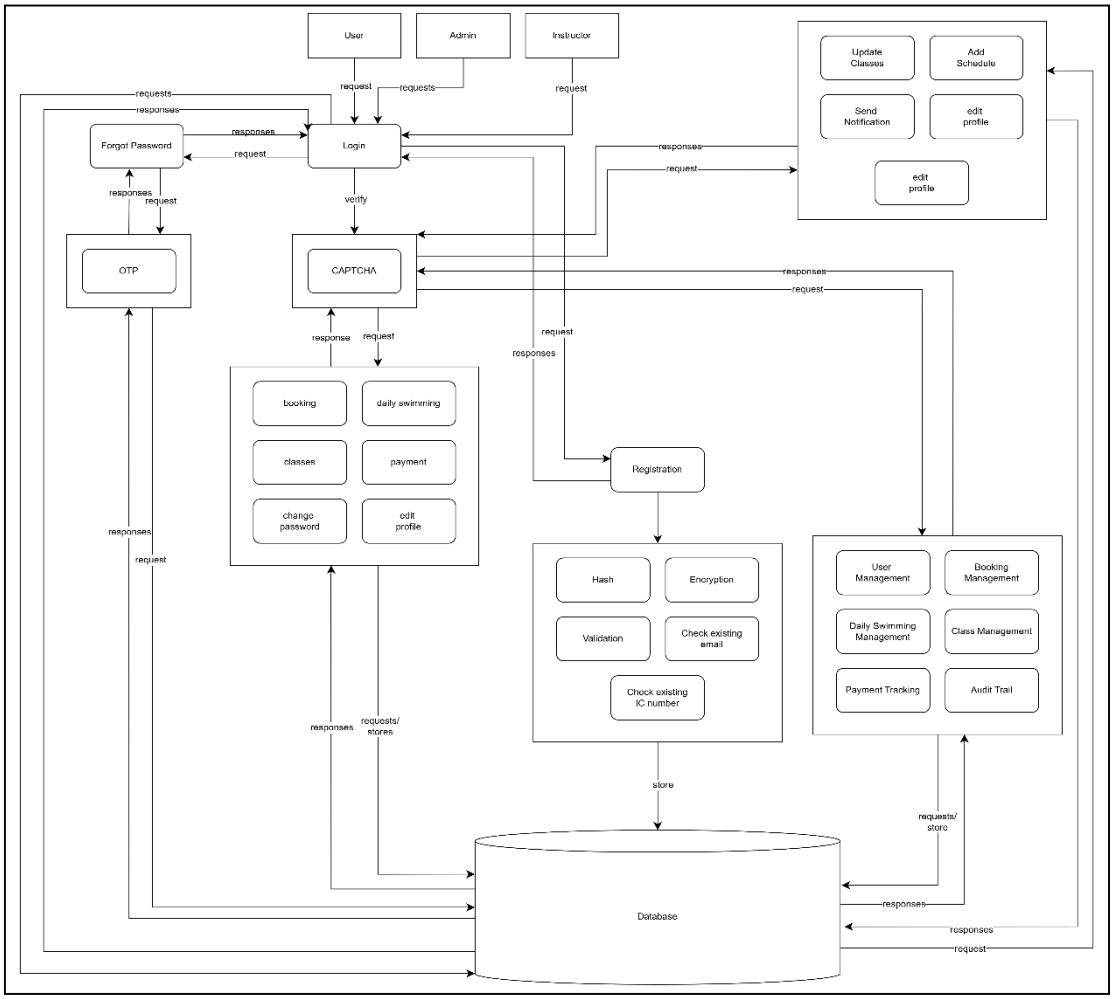


Fig. 6 *System Architecture*

4.5 Interface Design

Interface design focuses on creating a user-friendly layout that facilitates smooth interaction between the user and the system. It ensures that the system's features are easily accessible and intuitive, enhancing the overall user experience. For the Swimming Facilities Management System, the interface is designed to be simple and efficient, offering clear views of booking options, schedules, and payment processes.

Fig. 7 is a login page for users that already have an account. Users need to enter their email address and password to login. There is also a link to create an account for new users and forgot password options. Fig. 8 is a sign-up page where new users need to register their personal information to create an account. Fig. 9 shows the user profile that users can edit their personal details.

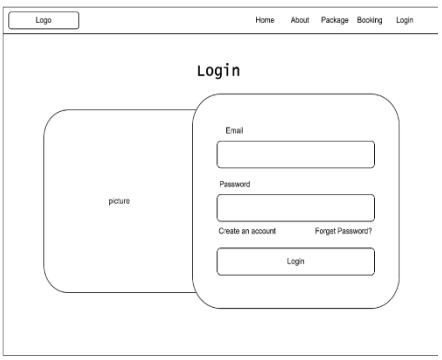


Fig. 7 *Login page*

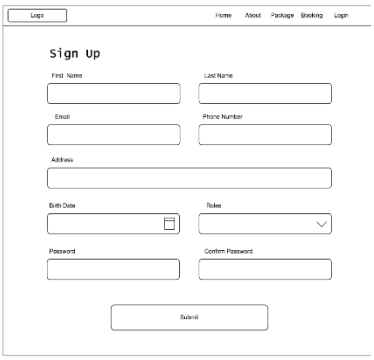


Fig. 8 *Sign-Up page*

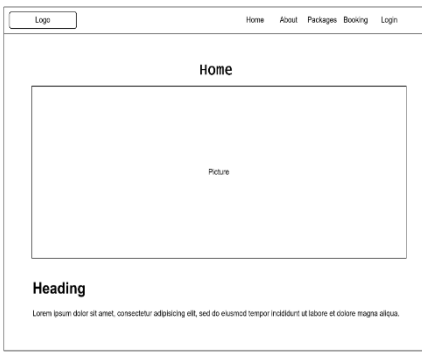


Fig. 9 *Home page*

Fig. 10 is a payment page after users have done the booking process. Fig. 11 shows the user profile that users can edit their personal details. Fig. 12 is the schedule section that users can access in home page.

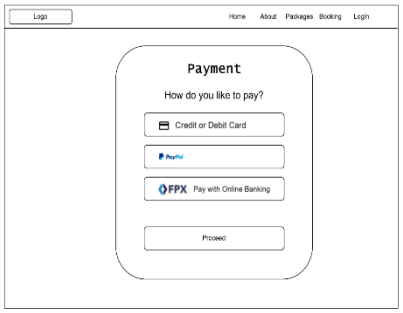


Fig. 10 *Payment page*

Fig. 11 *User profile page*

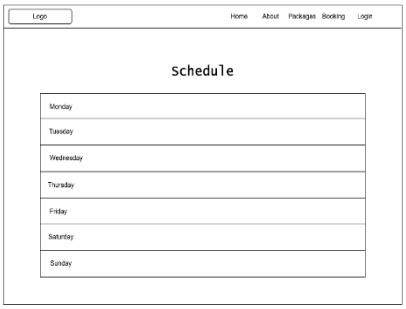


Fig. 12 *Schedule section*

Fig. 13 shows the admin dashboard and current information about the system. Fig. 14 shows the user management page. It allows admin to edit and remove the user in the system. Fig. 15 shows the booking management page for admin. It gives the latest information about swimming pool booking. Admin can accept or reject the booking.

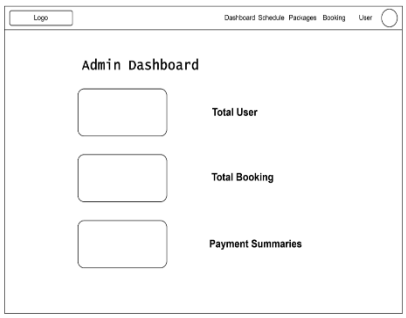


Fig. 13 *Admin Dashboard page*

Fig. 14 *User Management page*

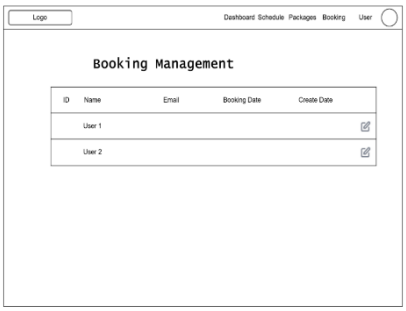


Fig. 15 *Booking Management page*

5. Implementation and Testing

Implementation and testing involve developing the system based on the design specifications and then verifying its functionality through various tests. During implementation, the system's features and components are coded and integrated. Testing follows to ensure that the system works correctly, meets user requirements, and is free of errors or bugs before deployment.

5.1 System Implementation

Fig. 16 shows existing users must access this page to log in to their accounts. They are required to input their email and password. While Fig. 17 shows the code segment for displaying the login page.

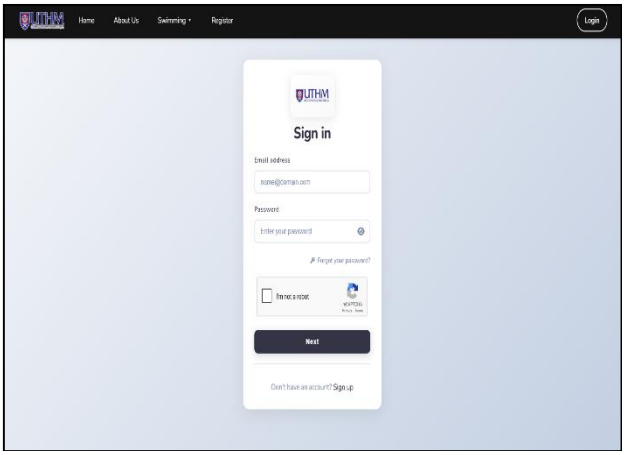
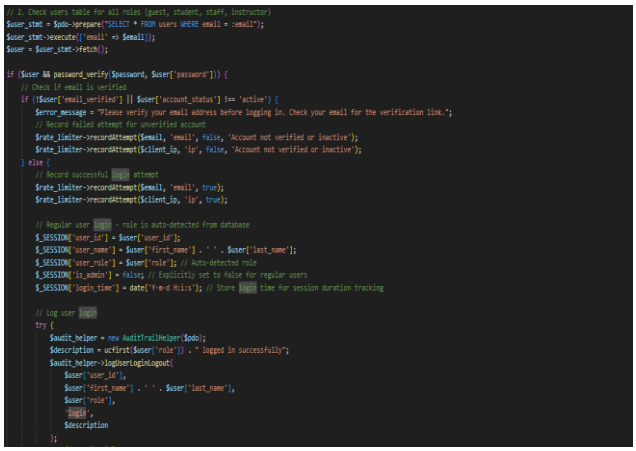


Fig. 16 *Login page*



```
// 2. Check users table for all roles (guest, student, staff, instructor)
$user_stmt = $pdo->prepare("SELECT * FROM users WHERE email = :email");
$user_stmt->execute(['email' => $email]);
$user = $user_stmt->fetch();

if ($user && password_verify($password, $user['password'])) {
    // Check if email is verified
    if (!$user['email_verified'] || $user['account_status'] !== 'active') {
        $error_message = "Please verify your email address before logging in. Check your email for the verification link.";
        // Record failed attempt for unverified account
        $rate_limiter->recordAttempt($email, 'email', false, 'Account not verified or inactive');
        $rate_limiter->recordAttempt($client_ip, 'ip', false, 'Account not verified or inactive');
    } else {
        // Record successful login attempt
        $rate_limiter->recordAttempt($email, 'email', true);
        $rate_limiter->recordAttempt($client_ip, 'ip', true);

        // Regular user login - role is auto-detected from database
        $_SESSION['user_id'] = $user['user_id'];
        $_SESSION['user_name'] = $user['first_name'] . ' ' . $user['last_name'];
        $_SESSION['user_role'] = $user['role']; // Auto-detected role
        $_SESSION['is_admin'] = false; // Explicitly set to false for regular users
        $_SESSION['login_time'] = date('Y-m-d H:i:s'); // Store login time for session duration tracking

        // Log user login
        try {
            $audit_helper = new AuditTrailHelper($pdo);
            $description = ucfirst($user['role']) . " logged in successfully";
            $audit_helper->logUserLoginLogout(
                $user['user_id'],
                $user['first_name'] . ' ' . $user['last_name'],
                $user['role'],
                'login',
                $description
            );
```

Fig. 17 *Code Segment for Login page*

The booking page shows the difference between slots that have already been booked, available and past as shown on Fig. 18. While Fig. 19 shows a code segment for booking page.

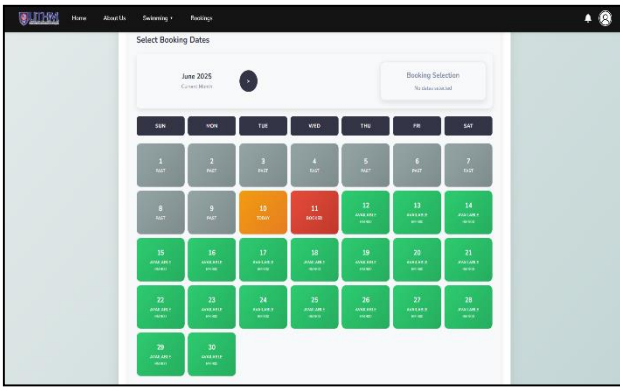
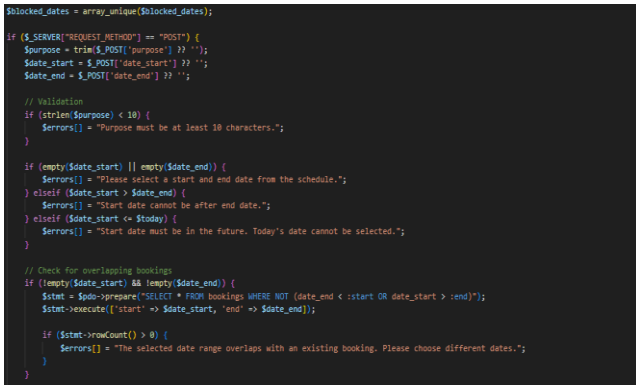


Fig. 18 *Booking page*



```
$blocked_dates = array_unique($blocked_dates);
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $purpose = trim($_POST['purpose'] ?? '');
    $date_start = $_POST['date_start'] ?? '';
    $date_end = $_POST['date_end'] ?? '';

    // Validation
    if (strlen($purpose) < 10) {
        $errors[] = "Purpose must be at least 10 characters.";
    }

    if (empty($date_start) || empty($date_end)) {
        $errors[] = "Please select a start and end date from the schedule.";
    } elseif ($date_start > $date_end) {
        $errors[] = "Start date cannot be after end date.";
    } elseif ($date_start <= $today) {
        $errors[] = "Start date must be in the future. Today's date cannot be selected.";
    }

    // Check for overlapping bookings
    if (!empty($date_start) && !empty($date_end)) {
        $stmt = $pdo->prepare("SELECT * FROM bookings WHERE NOT (date_end < :start OR date_start > :end)");
        $stmt->execute(['start' => $date_start, 'end' => $date_end]);

        if ($stmt->rowCount() > 0) {
            $errors[] = "The selected date range overlaps with an existing booking. Please choose different dates.";
        }
    }
```

Fig. 19 *Code Segment for Booking page*

Figure 20 shows the Stripe checkout page after user confirm booking the swimming pool. This is the payment phase and Fig. 21 is the successful payment after completion of payment.

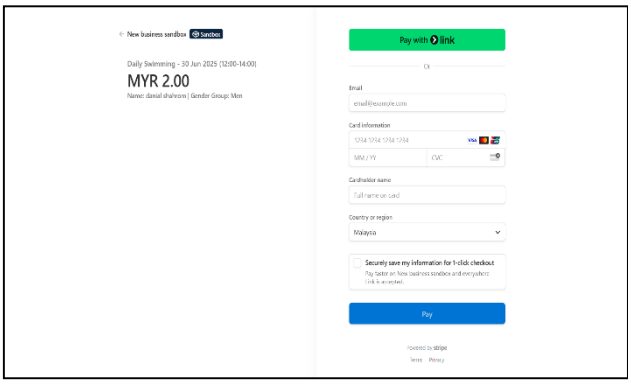


Fig. 20 *Stripe checkout page*

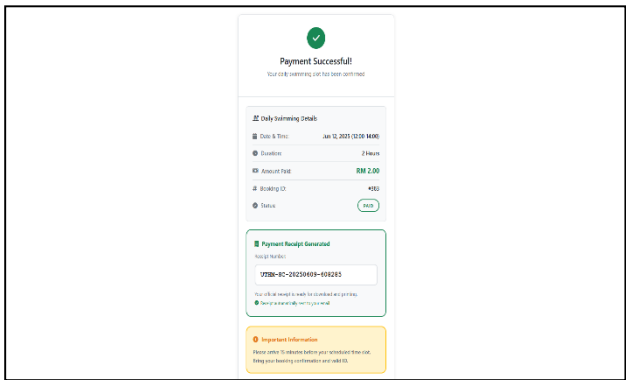


Fig. 21 *Payment Successful*

For daily swimming, user will get the receipt with QR code. Fig. 22 shows the user will scan the QR code in QR scan page. This process will help the administrator detect systematically the user that already pay for swim. Fig. 23 shows the booking management page for administrator to manage booking.

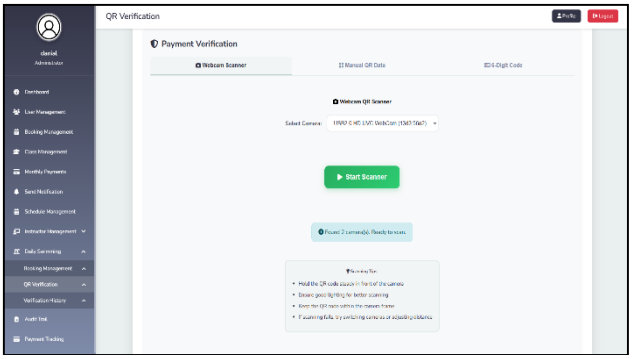


Fig. 22 *QR Scan page*

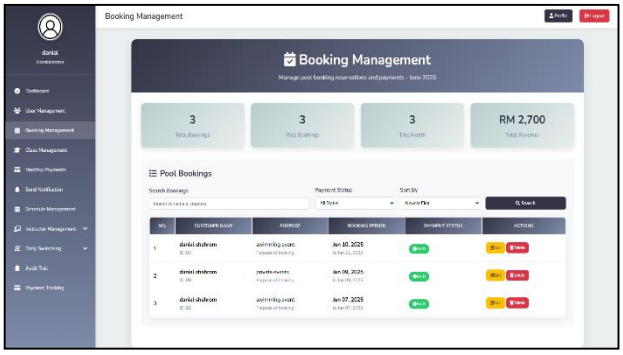


Fig. 23 *Booking Management page*

5.2 Security Implementation

For security implementation in the Swimming Facilities Management System, various measures are applied to protect user data and system operations. This includes secure user authentication such as strong passwords and role-based access control, encryption and hashing to protect credentials, and security testing to identify and fix vulnerabilities. These steps help ensure that sensitive information, such as personal details, remains safe from potential threats.

5.2.1 Password Hashing

Fig. 24 the system uses PHP's password_hash() and password_verify() functions to securely hash and verify passwords during registration and login. This approach prevents plain-text password storage and provides resistance against rainbow table attacks.

```
if ($user && password_verify($password, $user['password']))
$hashed_password = password_hash($password, PASSWORD_DEFAULT);
```

Fig. 24 *Password Hashing*

5.2.2 Role-Based Access Control

Fig. 25 shows role-based access control ensures that only authorized users can access specific parts of the system. Functions like isAdmin() and requireAdmin() in session_manager.php check the session to validate user roles, such as admin or instructor, and redirect unauthorized users.

```
function isAdmin() {
    return isset($_SESSION['is_admin']) && $_SESSION['is_admin'] === true;
}

/**
 * Check if user is instructor
 * @return bool
 */
function isInstructor() {
    return isset($_SESSION['user_role']) && $_SESSION['user_role'] === 'instructor';
}

/**
 * Get current user role
 * @return string|null
 */
function getUserRole() {
    return $_SESSION['user_role'] ?? null;
}
```

Fig. 25 *Role-Based Access Control*

5.2.3 Login Rate Limiter

Fig. 26 shows the system prevents brute force attacks using a rate limiter class, which limits login attempts to five within five minutes. If the limit is exceeded, the account is temporarily locked.

```
class LoginRateLimiter {
    private $pdo;
    private $max_attempts;
    private $lockout_duration; // in minutes
    private $cleanup_interval; // in hours

    public function __construct($pdo, $max_attempts = 5, $lockout_duration = 5, $cleanup_interval = 24) {
        $this->pdo = $pdo;
        $this->max_attempts = $max_attempts;
        $this->lockout_duration = $lockout_duration;
        $this->cleanup_interval = $cleanup_interval;
public function isLockedOut($identifier, $type = 'email') {
    try {
        // Check active lockouts
        $stmt = $this->pdo->prepare("
            SELECT locked_until, attempt_count
            FROM login_lockouts
            WHERE identifier = :identifier
            AND identifier_type = :type
            AND locked_until > NOW()
        ");
        $stmt->execute([
            ':identifier' => $identifier,
            ':type' => $type
        ]);

        $lockout = $stmt->fetch();

        if ($lockout) {
            return [
                'locked' => true,
                'locked_until' => $lockout['locked_until'],
                'attempt_count' => $lockout['attempt_count'],
                'remaining_time' => $this->getRemainingLockoutTime($lockout['locked_until'])
            ];
        }
```

Fig. 26 *Login Rate Limiter*

5.2.4 IC-Based OTP Format

Fig. 27 shows an OTP format based on the user's IC number is generated by combining segments of the IC with a random 4-digit code.

```php
// Split the last 4 IC digits: first 2 and last 2
$ic_first_2 = substr($last_4_ic, 0, 2);
$ic_last_2 = substr($last_4_ic, 2, 2);

// Generate a 4-digit OTP
$otp_4_digit = rand(1000, 9999);

// Combine: 2 IC digits + 4 OTP digits + 2 IC digits
$otp = $ic_first_2 . $otp_4_digit . $ic_last_2;

// Store the original 4-digit OTP for verification purposes
$_SESSION['otp_4_digit'] = $otp_4_digit;

// Set the OTP expiration time (in seconds) between 30 and 240
$otp_expiry_time = rand(30, 240); // Random expiration time between 30 and 240 seconds
$otp_expiry = date("Y-m-d H:i:s", time() + $otp_expiry_time); // Set expiry time
```

```php
// First, check if OTP columns exist in users table, if not add them
try {
    $pdo->exec("ALTER TABLE users ADD COLUMN otp VARCHAR(6) NULL");
} catch (Exception $e) {
    // Column might already exist
}

try {
    $pdo->exec("ALTER TABLE users ADD COLUMN otp_expiry TIMESTAMP NULL");
} catch (Exception $e) {
    // Column might already exist
}

// Update OTP and expiry in the users table
$sql_update_statement = "UPDATE users SET otp = ?, otp_expiry = ? WHERE user_id = ?";
$stmt_update = $pdo->prepare($sql_update_statement);
$stmt_update->execute([$otp, $otp_expiry, $user_id]);
```

Fig. 27 *IC-Based OTP Format*

5.2.5 Login Rate Limiter

Fig. 28 shows protect payment information displays, all dynamic outputs such as error messages and transaction IDs are sanitized using htmlspecialchars(). This ensures no script injection can occur in payment or receipt pages.

```php
$transaction_id = $receipt_data['payment_details']['transaction_id'] ?? null;
if (!empty($transaction_id)) {
    echo htmlspecialchars($transaction_id);
} else {
    echo '<span style="color: #6c757d; font-style: italic;">Not Available</span>';
}
```

Fig. 28 *Login Rate Limiter*

5.3 System Testing

Once the system has been fully developed, it proceeds to the testing phase to verify its functionality. This phase helps identify any bugs or errors present in the system. At the same time, any issues found can be addressed to ensure the system meets its intended scope and objectives. Table 4 presents a summary of the functional testing results.

Table 4 *Functional Test*

Test ID	Functional Testing	Test Case	Expected Result	Result
T001	User Registration	Register with valid UTHM email (student.uthm.edu.my)	Account created successfully, verification email sent	Success
T002	User registration	Register with an invalid email format	Registration fails with validation error	Success
T003	User login	Login with correct credentials	Successful login, redirect to home page	Success
T004	User login	Login with unverified email	Login blocked, verification required	Success
T005	Daily Swimming Booking	Book available time slot	Booking successful, redirect to payment	Success
T006	Daily Swimming Booking	Book duplicate time slot for same date	Booking blocked, duplicate prevention	Success
T007	Payment Processing	Process payment with valid card details	Payment successfully, receipt generated	Success
T008	Class Enrollment	Enroll in available swimming class	Enrollment successfully, instructor assigned	Success

Table 4 *(cont.)*

Test ID	Functional Testing	Test Case	Expected Result	Result
T009	Class Enrollment	Enroll in fully booked class	Enrollment blocked, class full	Success
T010	QR Code Verification	Scan valid QR code for payment verification	QR code verified, payment details displayed	Success
T011	QR Code Verification	Scan invalid or tampered QR code	Verification fails with security warning	Success
T012	Instructor Schedule Management	Instructor updates class schedule	Schedule updated, students notified	Success

5.4 User Testing

User testing was carried out to verify that the system meets the requirements and expectations of the target users. A total of 14 participants, including 5 normal users, 3 instructors, and 1 administrator, were selected to complete a detailed questionnaire. The questions covered areas such as ease of registration, login process, booking activities, and overall satisfaction with the system's security and usability. The results showed a 98% satisfaction rate, with all participants appreciating the two-factor authentication feature. This comprehensive user testing demonstrated the system's effectiveness in delivering a secure and user-friendly platform for UTHM students. Fig. 30 shows the testing results of the user. Fig. 31 shows the testing results of the instructor and Fig. 32 shows the results of the administrator. The figures display a graph rating where 1 represents "Yes" and 0 represents "No."

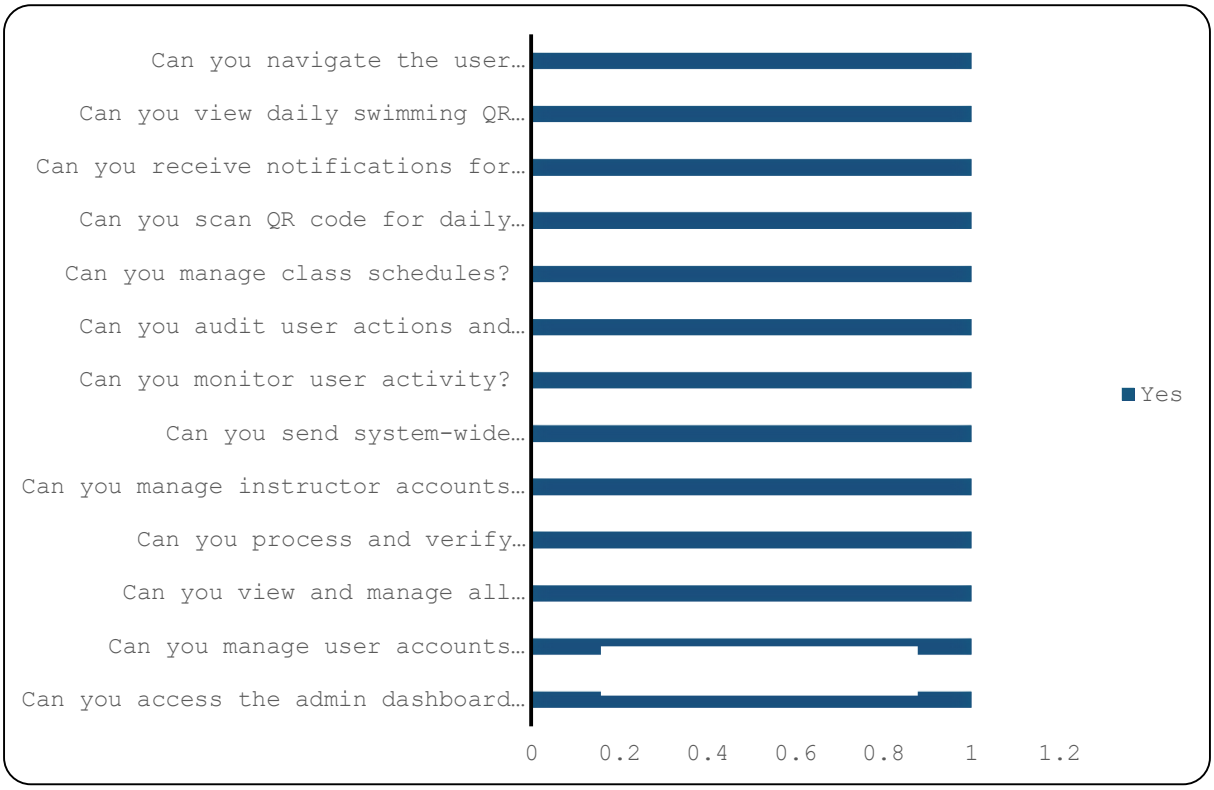


Fig. 30 *Result of Testing for User*

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 625-643

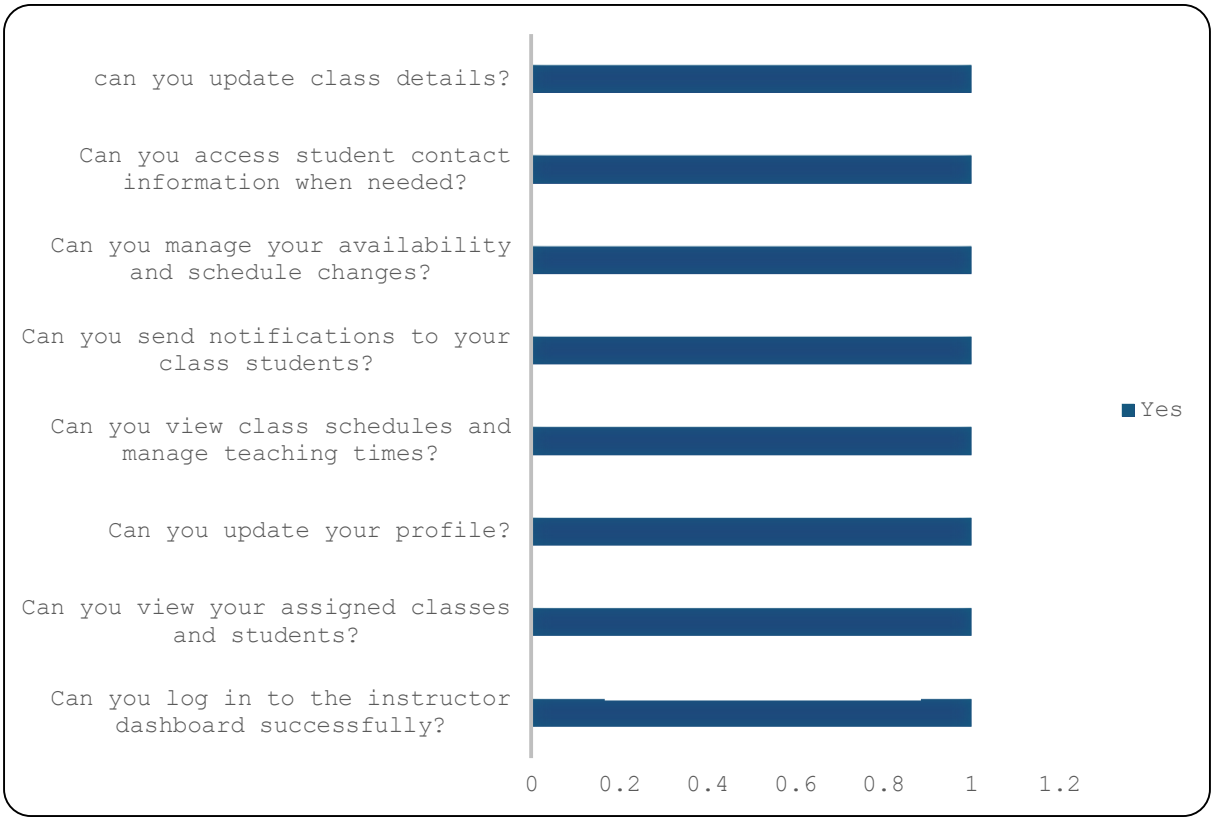


Fig. 31 *Result of Testing for Instructor*

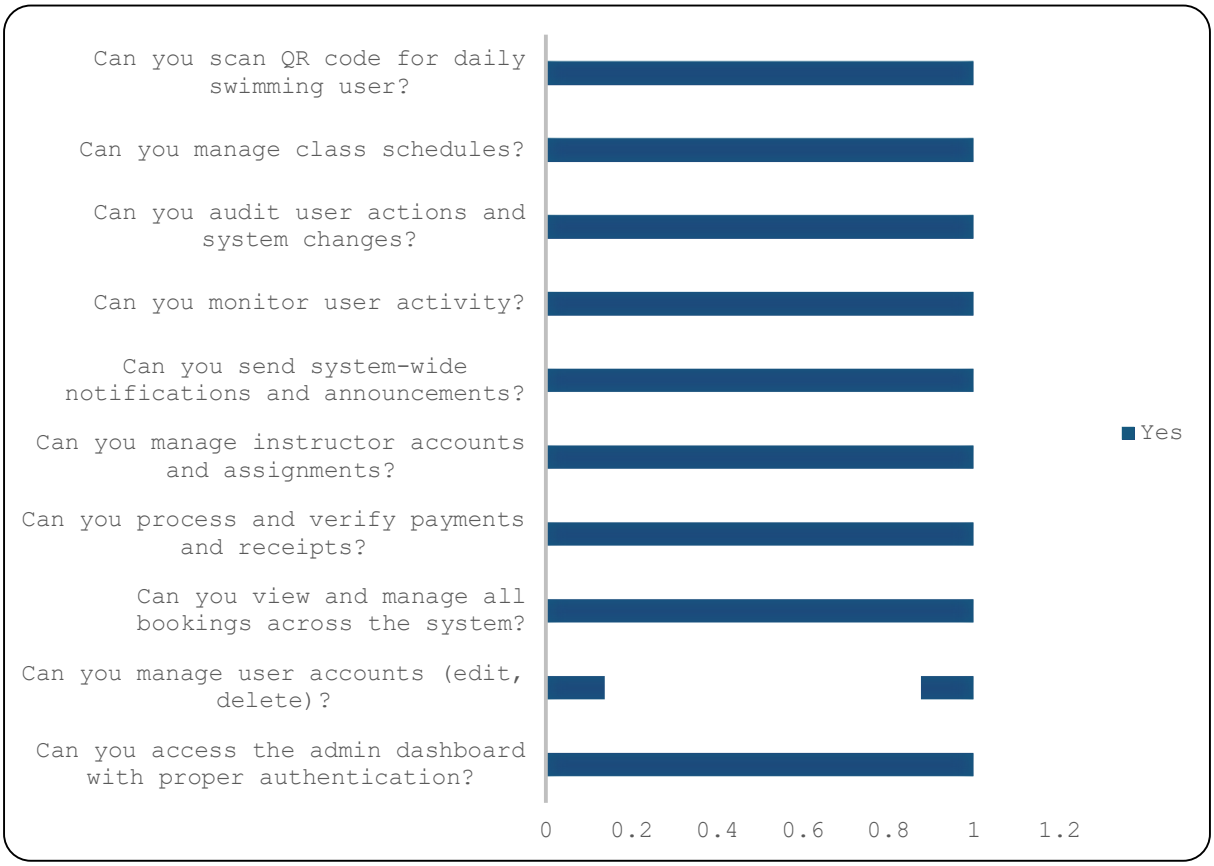


Fig. 32 *Result of Testing for Administrator*

6. Conclusion

In conclusion, the Swimming Facilities Management System for UTHM Sports Centre with Two-Factor Authentication addresses inefficiencies in the current manual processes by introducing a centralized, user-friendly platform built with PHP, HTML, CSS, and MySQL. The system integrates Role-Based Access Control, automated bookings, efficient payment options, and secure data management to enhance user experience and operational efficiency. Employing the Waterfall model, the project follows a structured development process, ensuring reliability and scalability. Key components such as system design, ERD, DFD, and interface design collectively create a robust solution that simplifies bookings, enables efficient transactions, and improves overall user satisfaction at UTHM Sports Centre. However, the system has some limitations, such as users only being able to see updates about their bookings or classes after logging in, which may cause them to miss important notifications, and the absence of a calendar feature for instructors to manage their class schedules. For future improvement, the system can be enhanced by integrating more payment methods such as FPX or e-wallets, developing a mobile-friendly version or dedicated mobile application for easier access, and enhancing the reporting module to support better monitoring and decision-making by administrators.

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 interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *M. D. Shahrom, K. A. Mohamad Sukri;* ***data collection:*** *M. D. Shahrom, K. A. Mohamad Sukri,* ***analysis and interpretation of results:*** *M. D. Shahrom, K. A. Mohamad Sukri****; draft manuscript preparation:*** *M. D. Shahrom, K. A. Mohamad Sukri. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Y. Guan, B. Wu, and J. Jia, "Does online ticket booking system make people better off? An empirical study on railway service," Transp Res Part F Traffic Psychol Behav, vol. 73, 2020, doi: 10.1016/j.trf.2020.03.014.
[2] Y. Cao, F. Li, Q. Zhang, S. Yang, and Y. Wang, "Towards Nonintrusive and Secure Mobile Two-Factor Authentication on Wearables," IEEE Trans Mob Comput, vol. 22, no. 5, 2023, doi: 10.1109/TMC.2021.3133275.
[3] K. M. Quadry, A. Govardhan, and M. Misbahuddin, "Design, analysis, and implementation of a two-factor authentication scheme using graphical password," International Journal of Computer Network and Information Security, vol. 13, no. 3, 2021, doi: 10.5815/IJCNIS.2021.03.04.
[4] Y. Cheng, C. Xu, Z. Hai, and Y. Li, "DeepMnemonic: Password Mnemonic Generation via Deep Attentive Encoder-Decoder Model," IEEE Trans Dependable Secure Comput, 10.1109/TDSC.2020.2987025.
[5] H. Kim, J. Han, C. Park, and O. Yi, "Analysis of vulnerabilities that can occur when generating one-time password," Applied Sciences (Switzerland), vol. 10, no. 8, 2020, doi: 10.3390/APP10082961.
[6] A. K. Y. S. Mohamed, D. Auer, D. Hofer, and J. Küng, "A systematic literature review for authorization and access control: definitions, strategies and models," 2022. doi: 10.1108/IJWIS-04-2022-0077.
[7] G. Rostami, "Role-based Access Control (RBAC) Authorization in Kubernetes," Journal of ICT Standardization, vol. 11, no. 3, 2023, doi: 10.13052/jicts2245-800X.1132.
[8] K. Gai, J. Yu, and L. Zhu, Introduction to Cybersecurity in the Internet of Things. CRC Press, 2024. doi: 10.1201/9781032694818.
[9] shuhada, "Aktiviti Kami," https://sukan.uthm.edu.my/index.php/kompleks-renang-uthm.
[10] "Swimming Pools," https://www.mbs.gov.my/en/citizens/services/public-amenities-and-bookings/swimming-pools.
[11] #, "Swimming Pool," https://gccservices.chennaicorporation.gov.in/swimmingpool.
[12] J. Rahmadoni, R. Akbar, and U. M. Wahyuni, "Web-Based Cooperation Information System At The Science Techno Park Technology Business Development Center," *Journal of Applied Engineering and Technological Science*, vol. 3, no. 2, 2022, doi: 10.37385/jaets.v3i2.806.
[13] D. Saripudin, W. I. Fauzi, and E. Nugraha, "The development of interactive E-book of local history for senior high school in improving local wisdom and digital literacy," *European Journal of Educational Research*, vol. 11, no. 1, 2022, doi: 10.12973/eu-jer.11.1.17.

[14] A. M. T. Nugraha, A. Y. Saputra, and R. S. Dewi, "Software Development Website Inventaris Pada Pusat Perbelanjaan XYZ," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 1, 2020, doi: 10.30865/jurikom.v7i1.1862.
[15] S. A. Sutresno and Y. A. Singgalen, "Analysis and Design of Morotai Tourism Village Information System (SIDEWITA) Based on Local Wisdom of Tokuwela and Babari Tradition," *Journal of Information Systems and Informatics*, vol. 5, no. 1, 2023, doi: 10.51519/journalisi.v5i1.441.
[16] N. Fajriati and K. Budiman, "Web-Based Employee Attendance System Development Using Waterfall Method," *Journal of Advances in Information Systems and Technology*, vol. 3, no. 2, 2022, doi: 10.15294/jaist.v3i2.52942.
[17] A. Sinha and P. Das, "Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry," in *2021 5th International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2021*, 2021. doi: 10.1109/IEMENTech53263.2021.9614779.