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

AITCS

Vol. 6 No. 2 (2025) 793-812
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

A Web Based Pool Facility Booking Application for UTHM Sport Centre with Device-Based Authentication

Jafni Hazwani Zakaria[1], Nurul Hidayah Ab Rahman[1]*

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

*Corresponding Author: hidayahar@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.043

Article Info

Received: 20 July 2025
Accepted: 19 November 2025
Available online: 19 November 2025

Keywords

Web-based application, Pool booking management, Device-based authentication, Online booking system, Digital payment gateway, User access control, Database management, Security implementation

Abstract

UTHM's pool facility relies on a manual, counter-based booking and payment system using paper records, creating three critical challenges: lack of online booking capabilities, inefficient user record management affecting emergency response, and inadequate access control. The project aims to develop a digital platform with secure authentication, online booking, and two-tier administrative access. Device-based authentication is implemented using the Fingerprint2.js method integrated with a database, allowing users to skip re-verification on previously verified devices for future. Following a six-phase methodology, from MoSCoW-based (Must have, Should have, Could have, Won't have) requirements gathering to post-implementation review, the system will implement user-friendly architecture, secure coding practices, and comprehensive testing protocols. The resulting platform will benefit the stakeholder by providing secure authentication for online payments, efficient booking management, improved emergency response capabilities, and enhanced reporting tools for management decision-making, effectively transitioning the facility from paper-based to digital operations.

1. Introduction

The swimming pool is a vital part of university life where students maintain their health through exercise and water sports while building community relationships [1]. Currently at UTHM, the pool facility operates through an inefficient manual system requiring in-person counter visits for bookings and payments, with staff recording transactions on paper receipts. This outdated approach has led to three major problems: the absence of an online booking system to systematically record usage and payment, inefficient manual management of user records creating safety concerns during emergencies, and poor control over different user access levels leading to potential unauthorized access.

To address these challenges, this project aims to design and develop a web-based pool facility booking application with device-based authentication for UTHM Sport Centre. The proposed system will implement differentiated access levels for regular users, pool staff, and administrators, with features including secure online booking and payment capabilities. This modernization will bring significant benefits: users will enjoy secure online payments and convenient booking management, staff will have better tools for emergency response and daily operations through quick access to user information, and management will gain access to detailed usage reports for better decision-making. The system will transform the facility's operations from paper-based to digital, enhancing security, efficiency, and user experience for the entire UTHM community.

2. Related Work

This section explores web-based booking systems, different authentication methods, and analyzes three existing pool booking platforms. By studying these elements, we can better understand how to develop an effective and secure pool booking system. The examination of current solutions helps identify both successful features to incorporate and gaps that need to be addressed in our proposed system.

2.1 Web Based Booking System

Web-based booking systems are online platforms that let users make reservations and payments through the internet. According to [2], these systems have transformed businesses from traditional methods to digital solutions. These systems offer 24/7 booking capabilities and secure payment processing [3]. They also generate useful data for analyzing booking patterns and customer preferences. For security, these systems need to follow best practices like collecting only essential information, implementing strict access controls, and following data privacy laws [4].

2.2 Authentication Methods

Traditional authentication typically uses usernames and passwords, which is simple but risky. Passwords can be compromised through hacking or phishing [4], [5]. While it's easy to implement and accessible to users, relying on just passwords makes systems vulnerable to attacks [5], [6]. To address these issues, Multi-Factor Authentication (MFA) requires two or more verification factors - something you know (password), have (security token/mobile), or are (biometrics) [4], [6]. MFA adds extra security layers and reduces unauthorized access risks [4], [5], though it can make systems more complex to integrate [4], [5].

2.2.1 Device Based Authentication

Device-based authentication is a security method used to uniquely identify a device by collecting various data points, such as the browser type, installed plugins, screen resolution, and operating system. These attributes are combined to create a unique "fingerprint" for each device, which is then stored and used to recognize the device in future logins or transactions. This method provides continuous device identification and helps prevent fraud, even when a user clears their browser cookies or browsing history, as the fingerprint remains linked to the device [7].

However, while device-based authentication is effective in identifying devices and enhancing security, it also raises privacy concerns. Users might not be fully aware that they are being tracked by these methods, as device fingerprinting can occur without explicit consent. Some people feel uncomfortable knowing that their device's unique identifier is being used to track them, even if they have not actively agreed to this [7].

To address these privacy concerns and improve security, experts recommend combining device fingerprinting with additional verification methods. For example, One-Time Passwords (OTPs) can be sent to the user to ensure that both the device and the person using it are authorized to perform sensitive actions, such as making an online payment. By combining device fingerprinting with OTPs, the security layer becomes stronger, ensuring only the rightful user is allowed to proceed with actions like transactions, thereby reducing the risk of fraud and unauthorized access [8].

2.3 Swimply

Swimply is a platform where pool owners can rent out their pools to swimmers, operating similarly to Airbnb but for swimming pools [9]. It allows users to create an account, browse available pools, view photos, and read user reviews. Users can book pools by the hour, providing flexibility for their needs. However, in terms of security, Swimply has some limitations. The platform implements a strong password policy, password masking, and limits login attempts, ensuring that users' accounts are well protected. However, Swimply lacks advanced security measures like OTP verification and session termination, which are critical for safeguarding user transactions and preventing unauthorized access. Furthermore, Swimply also lacks reCAPTCHA implementation and device-based authentication, which are essential for mitigating automated attacks and ensuring that only legitimate users can access the system. While the platform allows users to manage their bookings, it does not keep track of previous booking history, which may be a limitation for frequent users.

2.4 WET Deck

WET Deck simplifies the pool booking process by allowing users to book a pool session without the need to create an account [10]. Users can easily send a message via WhatsApp or fill out a form on their website to make a booking. While the system is very user-friendly, it falls short in terms of security features. WET Deck does not implement strong password policies, password masking, or limits on login attempts, making user accounts more

vulnerable to unauthorized access. The absence of reCAPTCHA also leaves the platform open to bot-driven attacks. Moreover, WET Deck does not support device-based authentication or OTP verification, which means that user transactions and sensitive data are not as securely protected as they could be. Although the system allows booking cancellations, there is no way for users to view their booking history, and once a payment is made, users cannot modify their booking time, which could be inconvenient for frequent users.

2.5 Aerotel Singapore

Aerotel Singapore's pool booking system targets travelers who need to quickly book a pool session during layovers at the airport [11]. Users can choose an available time slot and pay online. This simple, fast system is perfect for time-constrained travelers. However, it has significant limitations when it comes to security features. Like WET Deck, Aerotel Singapore lacks a strong password policy, password masking, and login attempt limitations, exposing users to potential security threats. Additionally, there is no implementation of reCAPTCHA, device-based authentication, or OTP verification, which leaves the system vulnerable to automated and unauthorized access. While it offers quick bookings, users cannot view their booking history or manage multiple bookings, which can be an inconvenience for those who regularly use the service. The lack of a session termination feature also raises concerns about the security of user accounts after they have logged in.

2.6 Comparison with the Existing Systems

The main differences between existing systems and current pool booking solutions lie in their features and security measures. Key features to consider include general and security features.

Table 1 *Comparison table between reviewed applications with the MyPool UTHM application*

Application	Swimply [9]	WET Deck [10]	Aerotel Singapore [11]	MyPool UTHM
General Features				
User Account	Yes	No	No	Yes
Booking Status	Yes	Yes	Yes	Yes
Show Available Slot	Yes	Yes	Yes	Yes
Booking History	No	No	No	Yes
Cancellation Features	Yes	No	Yes	Yes
Security Features				
Strong Password Policy	Yes	No	No	Yes
Password Masking	Yes	No	No	Yes
Limit Login Attempts	Yes	No	No	Yes
reCAPTCHA Implementation	Yes	No	No	Yes
Device based authentication (via email)	Yes	No	No	Yes
OTP verification	No	No	No	Yes
Session Termination	No	No	No	Yes

When comparing existing pool booking systems with the proposed UTHM Sport Centre system, several key differences emerge as in Table 1. While Swimply, Wet Deck, and Aerotel Singapore offer basic booking functionalities like availability checking and reservation making, they lack advanced features and comprehensive security measures. The proposed UTHM system stands out by offering enhanced user control and accessibility, including features like booking history tracking and profile management. It also implements a multi-tiered user account system with separate access levels for regular users, staff admins, and manager admins. Security-wise, the UTHM system provides robust protection through strong password policies, password masking, login attempt limits, reCAPTCHA, device-based authentication, and OTP verification features largely absent in existing systems. While some current platforms like Swimply and Wet Deck include basic security features such as session termination and email verification, the UTHM system's comprehensive approach to both user-friendly features and security measures makes it a more secure and flexible solution for pool facility management.

3. Methodology/Framework

Software development methodology is a systematic approach that guides the creation of software. It provides a structured framework of principles and tools for building reliable software solutions. For this project, the Agile methodology was selected because it offers significant advantages for web-based booking systems. According to research, "Agile offers flexibility in development, supports continuous user feedback, and allows for iterative development" [12]. This enables rapid iterations, easy integration of user experience requirements, and adaptability to changing project needs.

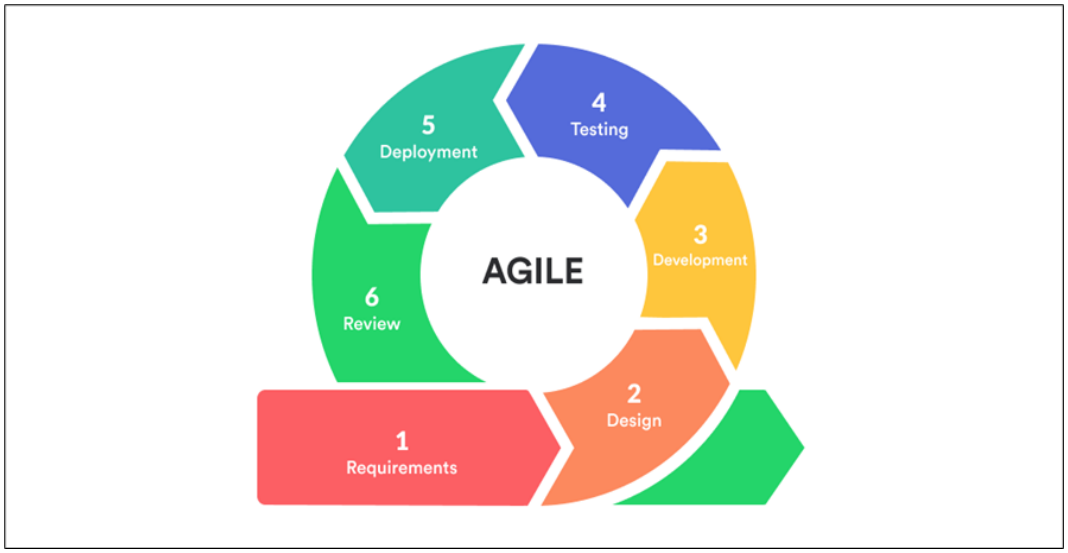


Fig. 1 *Agile Model [13]*

The development of the MyPool UTHM web-based booking system follows six key phases in the Agile methodology as in Fig. 1. The Requirements Phase begins with stakeholder interviews and uses the MoSCoW technique to prioritize features, focusing on both functional and non-functional requirements [14]. The MoSCoW technique categorizes features into Must have, Should have, Could have, and Won't have, helping the development focus on the most critical requirements first to align with Agile methodology. In the Design Phase, an iterative and adaptable approach is implemented, emphasizing collaboration and risk mitigation [15], while creating key components like ERD, DFD, and interface designs with a strong focus on security through device fingerprinting. The Development Phase, being the most extensive, encompasses UX/UI design and coding, utilizing PHP, MySQL, and Laravel for backend development, and HTML, CSS, and JavaScript for frontend development [16].

The Testing Phase serves as a comprehensive quality check, incorporating unit testing, security testing, integration testing, and a two-week on-site UAT with UTHM students/ staff, pool staff and admin. The Deployment Phase, known as the "end game" [16], focuses on transitioning the application to real-world use with robust security measures including SSL/TLS encryption and firewall protection, along with user documentation and training materials. Finally, the Review Phase serves as a reflection point where stakeholder meetings are conducted to gather feedback through comprehensive surveys, ensuring continuous improvement of the system's functionality and user experience [17].

4. Analysis and Design

The MyPool UTHM system is a web-based pool facility booking application designed for the UTHM Sport Centre. It focuses on simplifying the booking process by automating user registration, slot management, and device-based authentication. This section outlines the system analysis and design process, detailing the requirements, workflows, database structure, and user interface elements.

4.1 System Development Workflow

This system was designed based on detailed requirements to meet user expectations and ensure effective functionality. These requirements are categorized into user needs, functional operations, non-functional attributes, and security considerations. Table 2 shows the task and output of each phase.

Table 2 *Task and output of each phase*

Phase	Description	Output
Requirement	Talks with stakeholders and sort out what features are most important for the system.	A clear list of must-have features and system requirements is created
Design	Draw plans for how the system will look and work, with special attention to making it secure.	Detailed diagrams and sketches show how data flows through the system and how users interact with it.
Development	Build the system piece by piece, making sure the code is secure and properly reviewed.	A working version of the system that includes both the program logic and user interface.
Testing	Check every part of the system to make sure it works correctly and securely.	A list of any problems found and fixed, along with feedback from real users who tried the system.
Deployment	Set up the system on secure servers and prepare everything needed to teach people how to use it.	The system goes live with proper security measures and training materials ready for users.
Review	Gather feedback from users and look at how well the system is performing.	A report shows how satisfied users are and what improvements could be made to make the system better.

4.2 System Requirements Analysis

The MyPool UTHM system was designed based on detailed requirements to meet user expectations and ensure effective functionality. These requirements are categorized into user requirement (Table 3), functional operations (Table 4), non-functional attributes (Table 5), and security considerations (Table 6).

Table 3 *User Requirement*

No	Requirement
1	Users should be able to register accounts, make pool bookings through a user-friendly interface, and process payments securely with device authentication.
2	Staff users should have all general user capabilities plus the ability to manage bookings for dependents, create and manage dependent profiles, and handle family member bookings.
3	Staff should be able to manage and verify user bookings, monitor facility usage and capacity, and access user information for emergencies.
4	Admins should have the ability to manage staff accounts, monitor and maintain the system, and access reporting features.

Table 4 *Functional Requirement*

Application	Requirements	Users
Register	Users must create a new account by providing personal details with a strong password.	User, Admin
Login	Users authenticate using ID/username and password with password reset capability.	User, Staff, Admin
Home	Users can view facility information including available slots, fees, and rules.	User, Staff, Admin
Booking	Users can check slot availability, make bookings, manage payments, cancel reservations and view booking history.	User
Profile	Users can manage their personal information and dependent profiles.	User, Staff, Admin
Manage Booking	Staff can process booking requests, cancellations, and refunds while monitoring their capacity.	Staff
View User Booked	Users can access booking data, monitor facility usage, and generate reports for analysis.	Staff, Admin
Manage Staff	Admin can control staff accounts including adding, removing, and updating staff information.	Admin

Table 5 *Non-Functional Requirement*

Requirements	Description
Operation	The system must be consistently available, secure, and maintainable while supporting multiple users simultaneously.
Performance	The system must deliver fast response times and handle high traffic efficiently with minimal loading delays.

Table 6 *Security Requirement*

Requirements	Description
Authentication	The system must implement multi-factor authentication including username/password, device verification, and OTP.
Access Control	The system must manage user roles with appropriate privileges and automatic session timeouts.
Data Protection	The system must encrypt and securely store all sensitive user information.
System Security	The system must maintain regular backups, activity logs, and protect against common web vulnerabilities through input validation.

4.1.1 Software and Hardware Requirement Analysis

The system runs on a Dell Latitude 3440 with Windows 11 Pro, featuring an Intel Core i5 processor, 8GB RAM, and 512GB SSD. For development, it uses PHP and JavaScript for backend processing, and MySQL for database management. Frontend technologies include HTML, CSS, and JavaScript, with additional security provided by the Fingerprint2.js library for device-based authentication.

4.3 System Analysis

4.3.1 Context Diagram

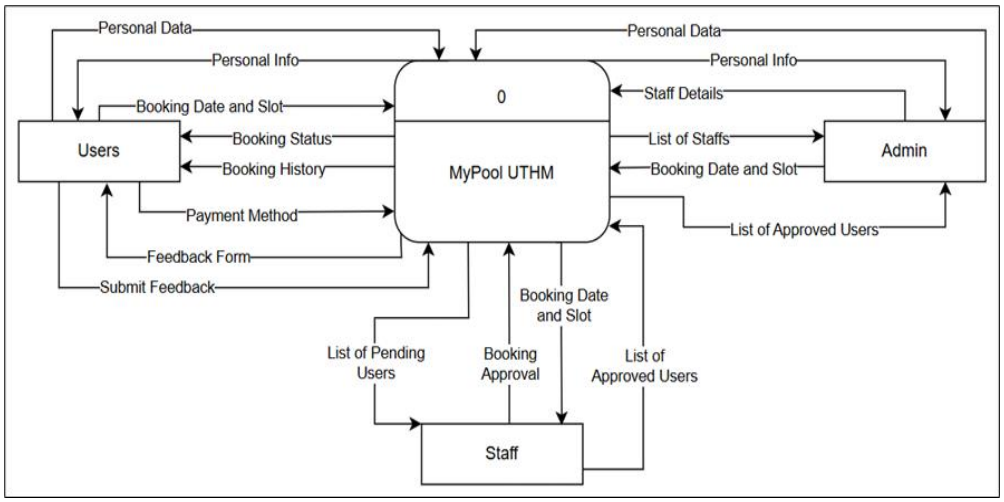


Fig. 2 *Context Diagram*

The MyPool UTHM is a website that helps people book swimming pool facilities at UTHM Sport Centre. Fig. 2 shows the context diagram for MyPool UTHM. There are three main groups who use the system: regular users (students, staff, and public), staff members who manage the pool, and administrators. Regular users can create accounts, book pool time slots, pay for their bookings, and leave feedback. They can also check their booking history and status. Staff members are in charge of approving bookings and making sure users are allowed to use the pool. The administrators focus on managing the staff, including adding new staff members and keeping track of their information. The whole system works together to make pool booking easier and more organized for everyone at UTHM Sport Centre.

4.3.2 Data Flow Diagram (DFD) Level 1

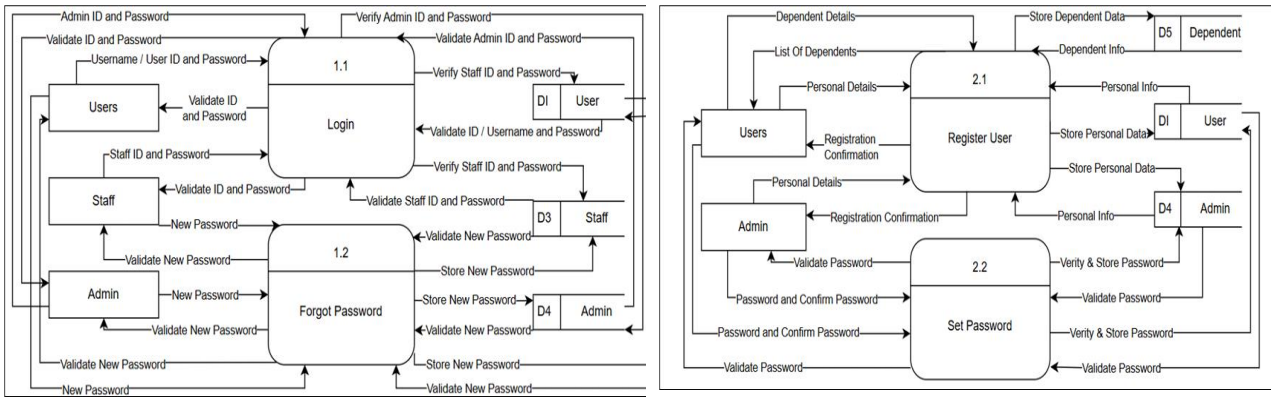


Fig. 3 *DFD Level 1 (a) Process 1; (b) Process 2*

Fig. 3 shows the DFD Level 1 for login, forgot password, register user and set password. The login process is how everyone (users, staff, and administrators) get into the MyPool UTHM system. They need to type in their username and password to get access. If someone forgets their password, they can use the "Forgot Password" feature to create a new one. The system makes sure the new password is saved properly in their account, replacing the old one. This helps people get back into their accounts safely if they can't remember their login details.

The registration process is how new people sign up to use the system. Users need to fill in their personal information like their name and contact details, which gets saved in the user database. Staff members can also add family members (dependents) to their accounts. Administrators can create new accounts too, and their information is kept in a separate admin database. When registering, everyone needs to create a strong password that meets the system's security requirements. Once everything is filled in correctly, people can start using the system.

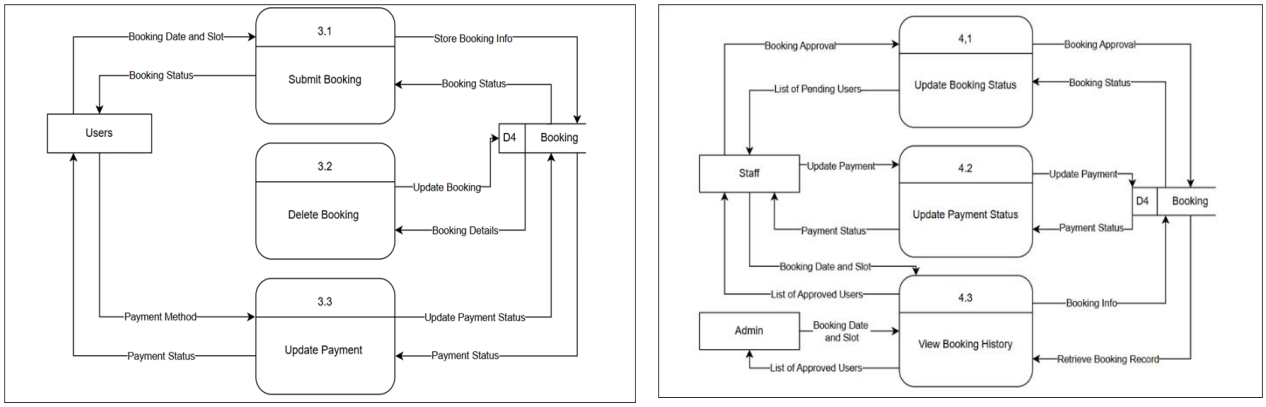


Fig. 4 *DFD Level 1 (a) Process 3; (b) Process 4*

When users want to book the pool, they can select their preferred date and time slot through the booking process. The system saves all booking details in a database, and users can check their booking status anytime. If needed, users can cancel their bookings or update their payment information. They can pay either online or in person at the facility, and the system keeps track of all payment statuses.

The staff members play a key role in managing bookings. They review new booking requests and can either approve or deny them. After approval, staff can update payment status, especially for those who choose to pay at the facility. Both staff and administrators can view booking histories and see a list of approved users, which helps them keep track of who's using the pool facility. This makes it easier for them to manage the facility and keep records of all pool usage.

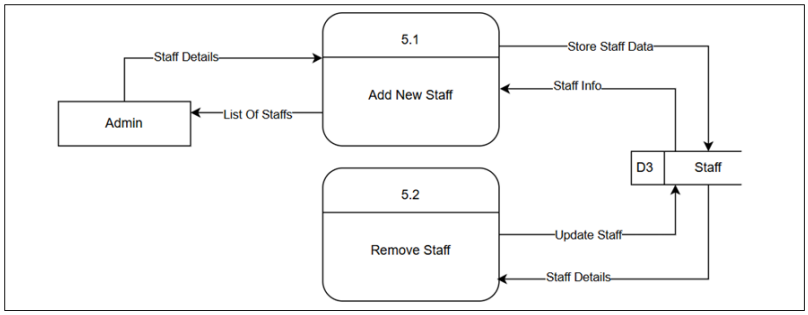


Fig. 5 *DFD Level 1 Process 5*

Administrators can add and remove staff members to the system (Fig. 5). When adding new staff, the admin enters their personal information like name and contact details, which gets saved in the staff database. This lets the new staff members log in and start managing pool bookings. If a staff member leaves their job or no longer works at the pool facility, the admin can remove them from the system. When this happens, their information is deleted from the database, and they can no longer access the system or manage any bookings.

4.3.3 Entity Relationship Diagram (ERD)

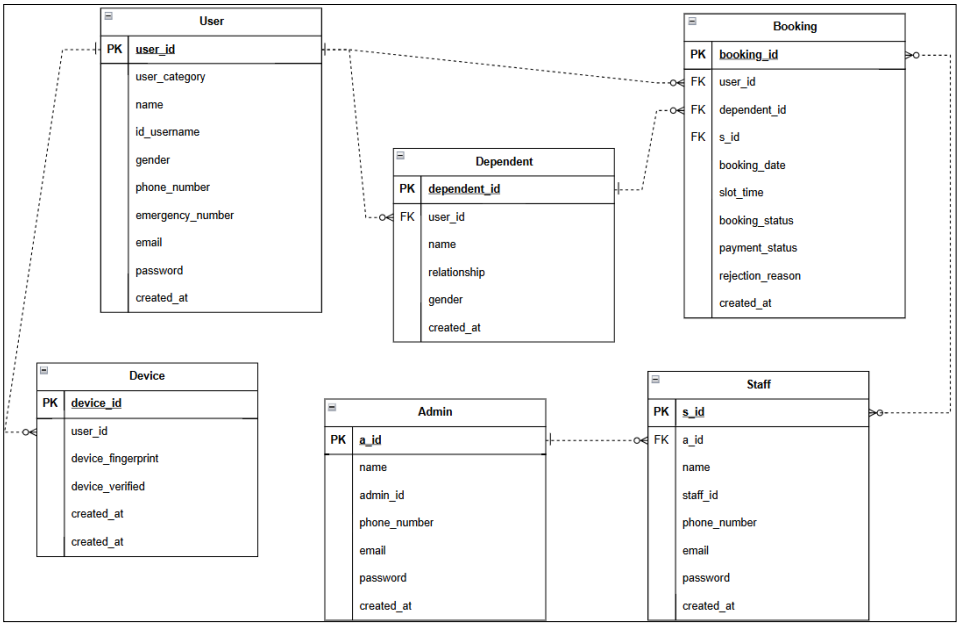


Fig. 6 *ERD for MyPool UTHM*

The MyPool UTHM system (Fig. 6) has six main connected parts: Users, Dependents, Bookings, Staff, Admins, and Device. Users can be students, staff, or public members, and each has their own profile with basic information like name, ID/username, contact details, and password. Staff members can add family members as dependents to their accounts. Any user or dependent can have bookings, which include details like the date, time slot, and payment status. Staff members, who are managed by administrators, handle these bookings. Both staff and admin profiles contain their personal information and login details. The Device table stores verified fingerprinted devices for each user, simplify for payment process on trusted devices without repeated verification in future. This structure allows the system to keep track of everyone who uses the pool, manage bookings efficiently, and maintain clear records of who's in charge of running the system.

4.3.4 Flowchart

The system flowcharts (Fig. 7) for MyPool UTHM outline how different user roles (User, Staff, Admin) interact with the pool booking application. Users begin by selecting their role. New users register, while existing ones log in with their credentials. Users can view pool details, book slots, check booking history, and manage profiles. Staff log in to process bookings by approving or rejecting requests, update payment statuses, and manage their profiles. Admins can register new staff, view staff lists, track bookings, and manage their profiles. The flowcharts provide a clear overview of user navigation and decision-making across the system.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 793-812

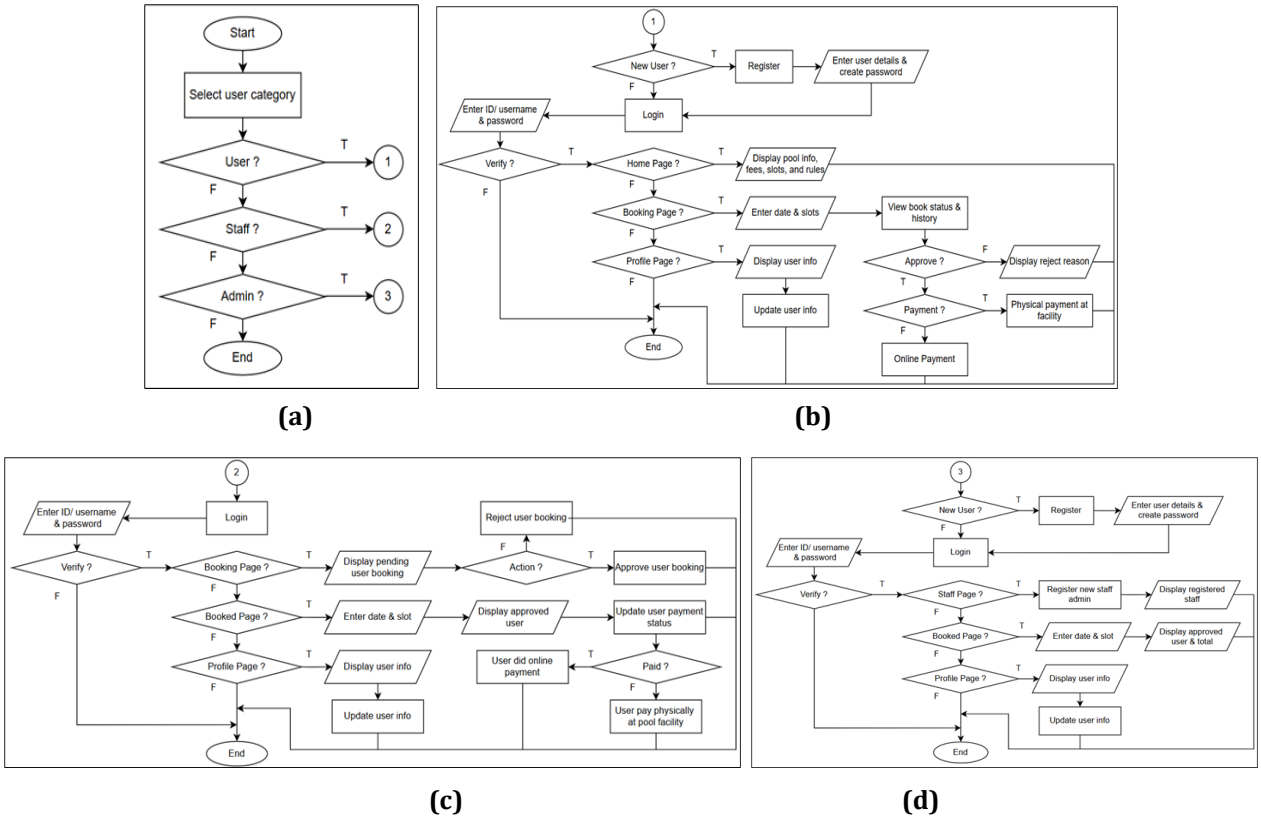


Fig. 7 *System Flowchart*

4.3.5 System Architecture Model

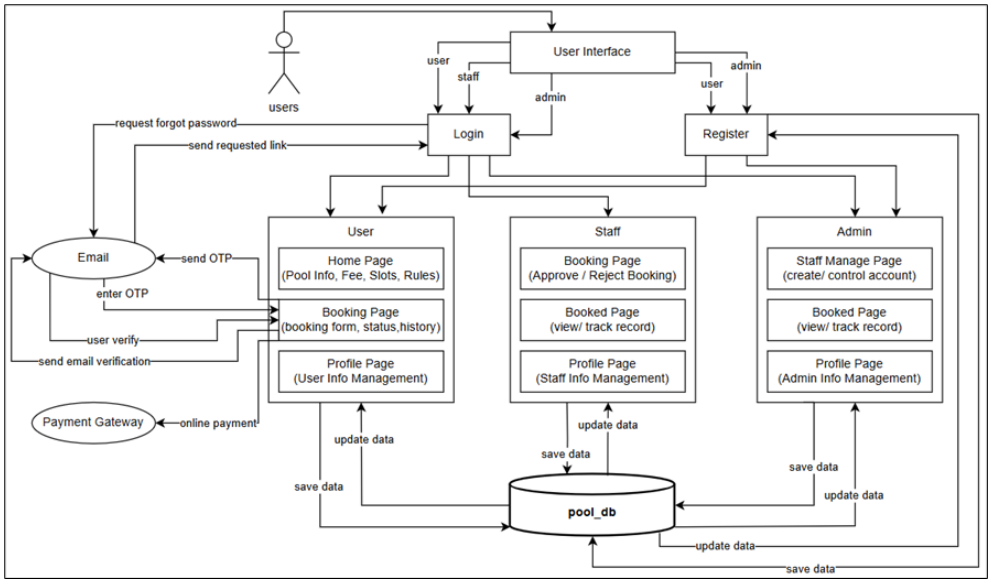
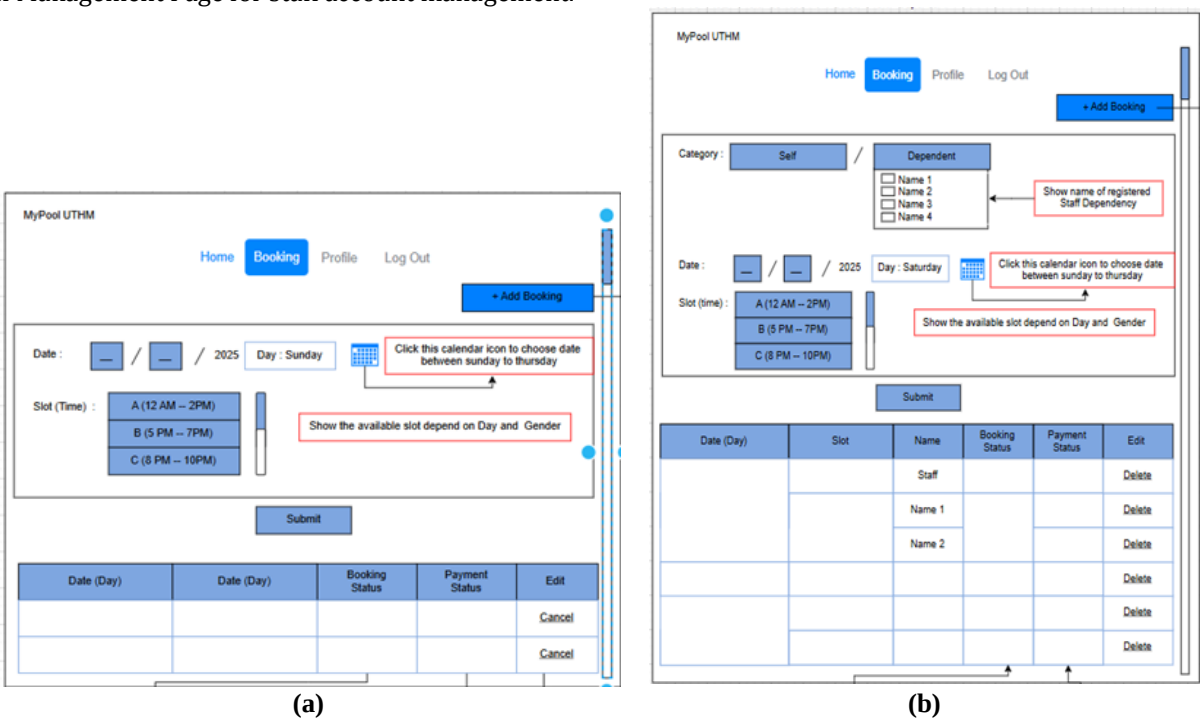


Fig. 8 *System Architecture Model*

Fig. 8 shows the system architecture model. When users first sign up, the system creates a unique "fingerprint" of their device (like taking a snapshot of their browser and computer settings). Every time someone tries to log in, the system checks if they're using the same device by comparing fingerprints. If the device matches the original fingerprint, the system sends a one-time password (OTP) via email for verification before allowing any payments. However, if someone logs in from a new device, they'll need to go through extra security steps - the system will send an email verification, and only after confirming their identity and entering the OTP can the user proceed with payment. This whole process acts like a digital security guard that ensures only legitimate users can make payments, especially when accessing from different devices than usual.

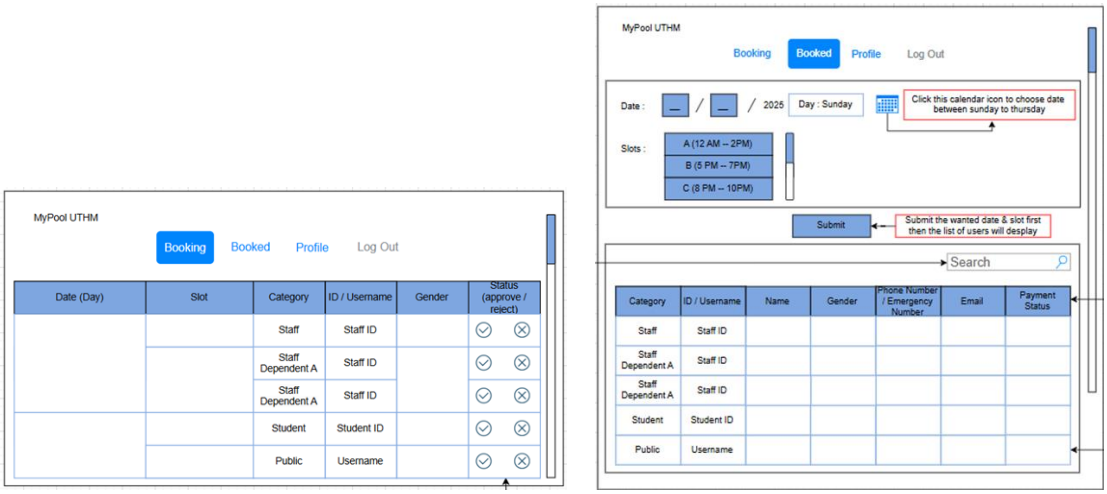
4.3.6 Interface Design

Users access the Home Page for facility details and booking options. The Booking Page in Fig 9 is for managing reservations, and the Profile Page for updating personal and dependent information. Staff interact with the Booking Management Page to handle bookings, the Booked Page to view reservations, with admins accessing the Staff Management Page for staff account management.



(a) **(b)**

Fig. 9 *User Interface (a) Booking Page (public /student); (b) Booking Page (staff user)*



(a) **(b)**

Fig. 10 *Staff Interface(a) Booking Management Page; (b)List Booked User Page*

5. Results and Discussion

5.1 Implementation of Security Module

To begin with, the pool facility booking system implements ten comprehensive security modules designed to protect against various cyber threats and unauthorized access. These include device-based authentication as the core security feature, input validation and sanitization to prevent malicious data entry, password masking with toggle functionality, live password policy indicators, secure password hashing using Bcrypt algorithms, reCAPTCHA protection against automated attacks, login attempt limitations to prevent brute force attacks, email OTP verification for multi-factor authentication, automatic session termination for idle users, and reliable session management.

5.1.1 Device Based Authentication

```
Start

    If user clicks "Pay Now" then
        Check if device fingerprint exists in database (device_statu

        If device_status = 0 then
            Send verification email
            While (count <= 5 minutes) do
                If user verify in email then
                    Update device_status = 1
                    Proceed to OTP generation
                    Exit While
                End If
            End While

            // If 5 minutes expire without verification
            If count > 5 minutes then
                // Redirect user back to page before
            End If

        Else if device_status = 1 then
            // Device is verified, proceed to OTP generation
            Proceed to OTP generation
        End If

End
```

(a)

```
<script src="https://cdn.jsdelivr.net/npm/fingerprintjs2@2.1.0/dist/fingerprint2.min.js"></script>
<script>
    // Capture device fingerprint
    Fingerprint2.get(function(components) {
        const deviceFingerprint = Fingerprint2.x64hash128(components.map(comp => comp.value).join(), 3
        const form = document.querySelector('form'); // Select form element
        const hiddenInput = document.createElement('input'); // Create hidden input
        hiddenInput.type = 'hidden'; // Set input type to hidden
        hiddenInput.name = 'device_fingerprint'; // Set input name
        hiddenInput.value = deviceFingerprint; // Set fingerprint value
        form.appendChild(hiddenInput); // Append input to form
    });
</script>
```

(b)

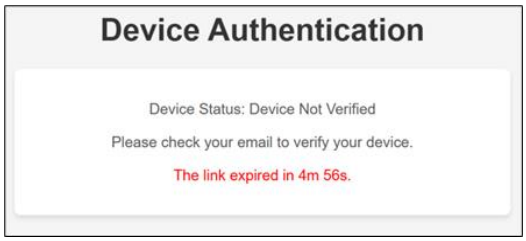
```
// Generate the next `device_id` with prefix
$device_id = generateNextDeviceId($db);

// Insert device fingerprint into the `device` table
$stmt = $db->prepare("INSERT INTO device (device_id, user_id, device_fingerprint, device_verifi
                      VALUES (:device_id, :user_id, :device_fingerprint, 0)");
$stmt->execute([
    ':device_id' => $device_id,
    ':user_id' => $user_id,
    ':device_fingerprint' => $deviceFingerprint
]);

$status = "Registration successful! <a href='../user/login_u.php'>Click here to login</a>";
```

(c)

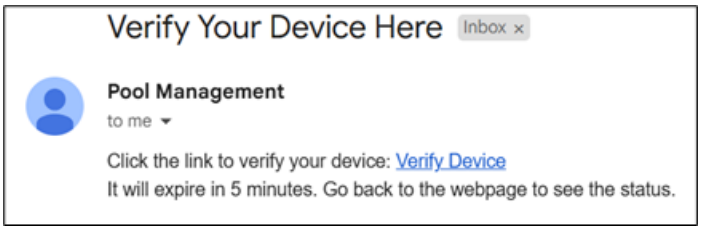
Fig. 11 *Device Authentication (a) Algorithm; (b) Fingerprint2.js; (b) Save Device Fingerprint*

This module creates unique digital fingerprints to enhanced security by following OWASP guidelines, which combine various device attributes such as operating system, browser version, screen resolution, installed fonts, and browser plugins to generate unique identifiers that can effectively detect unauthorized access attempts using the fingerprintjs2 JavaScript library [18].

In terms of implementation, the system utilizes the FingerprintJS library as shown in Fig. 11(b) to capture device-specific information including browser user agent, IP address, screen resolution, and timezone, subsequently generating a unique hash for each device. A similar approach was demonstrate by [19] that implement device-centric authentication for the web, where the user authenticates locally on their device. During the registration process, device fingerprints are automatically stored in the device table with an initial device_verified = 0, indicating that verification is required.



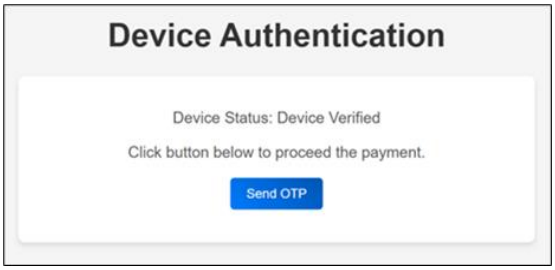
(a)



(b)

Fig. 12 *Device Authentication (a) Device Not Verified; (b) Email Verification*

When users attempt online payments, the system compares the current device fingerprint with stored records, and if the device is unverified, it triggers an email verification process that redirects users to device_authentication.php where they see a countdown timer as shown in Fig.12(a) The countdown ensures that only users who have verified their email can proceed and prevents repeated attempts from attackers.



(a)

```
// Update the device_verified status to 1 (verified) in the database
$query = "UPDATE device SET device_verified = 1 WHERE user_id = :user_id AND
device_fingerprint = :device_fingerprint";
$stmt = $db->prepare($query);
$stmt->execute([
    ':user_id' => $userId,
    ':device_fingerprint' => $deviceFingerprint
]);

// Redirect back to the device authentication page with a success message
header('Location: ../index/device_authentication.php');
exit;
?>
```

(b)

Fig. 13 *Device Authentication (a) Device Verified; (b) Update Device Status*

Once users click the verification link in verify_device.php, the system updates the device_verified status to 1 in the database as demonstrated in Fig.13(b), thereby marking the device as trusted. Finally, verified users are redirected back to device_authentication.php where they can click the "Send OTP" button shown in Fig. 13(a) to proceed with further authentication steps.

5.1.2 One Time Password

The system implements One-Time Password (OTP) verification via email as an additional security layer before transaction processing, also after staff or admin login as multi-factor authentication. Once device verification is completed, the system generates a random six-digit OTP using rand(100000, 999999) and sends it to the user's registered email address through the PHPMailer library as shown in Fig.14(a), while storing the OTP in the user's session, for 5 minutes before expired.

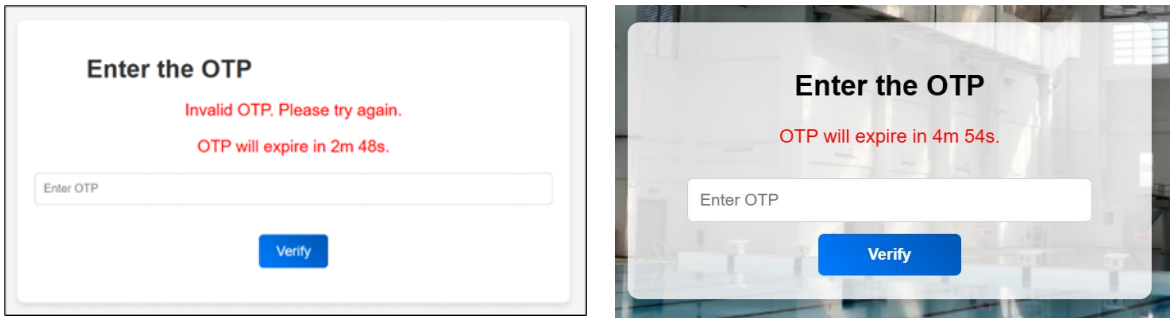
```php
// Device is verified, prepare for OTP email sending
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Generate OTP and store in session
    $otp = rand(100000, 999999);
    $_SESSION['user_otp'] = $otp;
    $_SESSION['otp_time'] = time(); // Store OTP generation time

    // Retrieve user email for sending OTP
    $query = "SELECT email FROM users WHERE user_id = :user_id";
    $stmt = $db->prepare($query);
    $stmt->execute([':user_id' => $userId]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    $userEmail = $user['email'];
```

(a)

```php
// Update the device_verified status to 1 (verified) in the database
$query = "UPDATE device SET device_verified = 1 WHERE user_id = :user_id AND
device_fingerprint = :device_fingerprint";
$stmt = $db->prepare($query);
$stmt->execute([
    ':user_id' => $userId,
    ':device_fingerprint' => $deviceFingerprint
]);

// Redirect back to the device authentication page with a success message
header('Location: ../index/device_authentication.php');
exit;
?>
```

(b)

Fig. 14 *OTP (a) OTP Generate; (b) OTP Verification*



(a) **(b)**

Fig. 15 *OTP (a) Authenticate User before Payment; (b) Authenticate Staff or Admin Before Login*

The OTP verification page displays an input field with a countdown timer as shown in Fig.15, and if users enter incorrect or expired codes, the system displays error messages and due to the expiry, the page automatically redirecting users back to the device authentication page to request a new OTP. This implementation follows OWASP recommendations for time-limited authentication codes, thereby ensuring codes have short lifespans and cannot be reused to address potential vulnerabilities [18].

5.1.3 Register Page

The user registration system (Fig. 16) implements input validation and sanitization using htmlspecialchars() and trim() functions to prevent malicious data entry, while email validation uses filter_var() with FILTER_VALIDATE_EMAIL to ensure proper format. The system includes duplicate checking to prevent existing usernames or emails from being registered.

The registration includes live indicator password policy that check for minimum eight characters, uppercase/lowercase letters, digits, and special characters, plus password masking with toggle functionality using JavaScript to switch between hidden and visible text. Validated passwords are securely hashed using password_hash() with PASSWORD_BCRYPT before database storage, while the system captures device fingerprints through FingerprintJS2 library to create unique device identifiers stored in the device table with device_verified status set to 0 for later verification during payment transactions.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 793-812

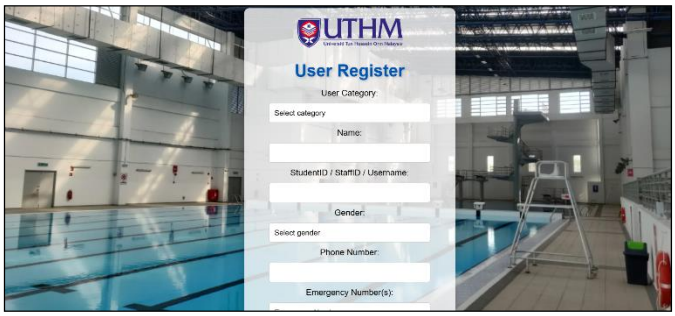


Fig. 16 *Register Page*

5.1.4 Login Page

Login page in Fig. 17 has input validation to ensure both username and password fields are required before processing, while input sanitization uses htmlspecialchars() and trim() functions to prevent malicious code injection. The system includes password masking with toggle functionality to protect against shoulder surfing attacks. Additionally, the login process incorporates reCAPTCHA verification through verifyRecaptcha() function that communicates with Google's servers to prevent automated bot attacks. On login, the script checks if the device fingerprint already exists in the device table. If not, it creates a new entry in the device table for the new device.

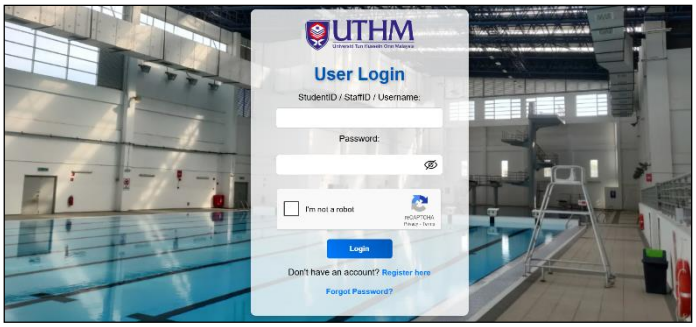


Fig. 17 *Login Page*

To prevent brute force attacks, the system implements a login attempt limitation mechanism that tracks failed login attempts in $_SESSION['user_failed_attempts'] and locks users out after 5 times failed attempts for 15 minutes. During lockout, the system calculates remaining time using time() function and displays a countdown timer that automatically refreshes the page when the lockout period expires.

For staff and admin login page, the system includes an additional security feature that forces password changes on first login by checking the force_password_change flag in the database, redirecting staff to change_password_s.php if the flag equals 1 to ensure generic passwords set by administrators are replaced with personalized secure passwords.

5.2 Implementation Of Booking Module

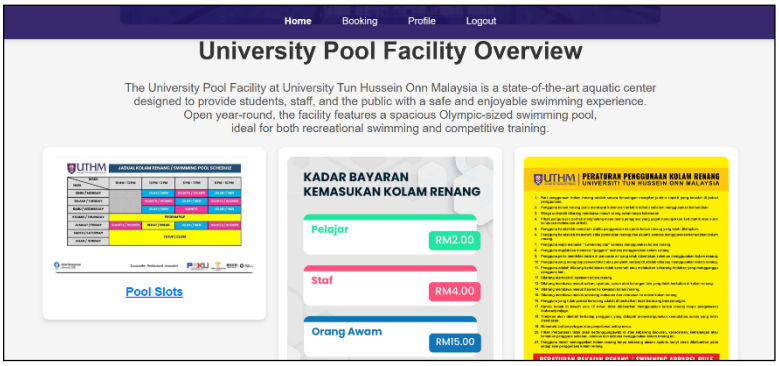
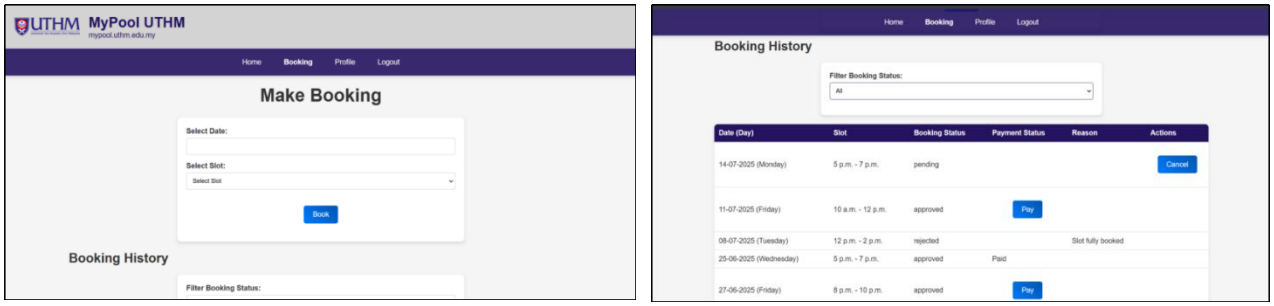


Fig. 18 *User Dashboard*

The User Dashboard in Fig. 18 is the main page where users can see everything about the pool facility such as booking schedule, prices and pool rules. The booking module (Fig. 19) helps pool users easily book their preferred time slots. First, the system checks if users already have a booking for the same time to prevent double bookings.

Then, it connects to the database to save and track all new reservations. The system also allows university staff to book slots for their family members or dependents. Hence, this feature manages bookings efficiently for individual users and families while avoiding booking conflicts.



(a) **(b)**

Fig. 19 *User Module (a) Booking Form; (b) Booking History*

The payment module in Fig. 20 offers users two payment options, offline (pay at premise) and online payments. The system automatically calculates fees based on user categories such as students, staff, or public users. For online payments, it uses secure platforms like Stripe with multi-factor authentication including device verification and OTP codes for safety. Users can also choose offline payments at the facility, and the booking status updates accordingly. Hence, this module provides flexible and secure payment choices for all users (Fig. 21).

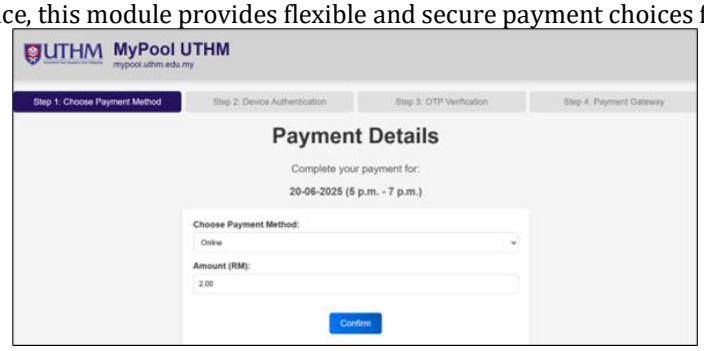
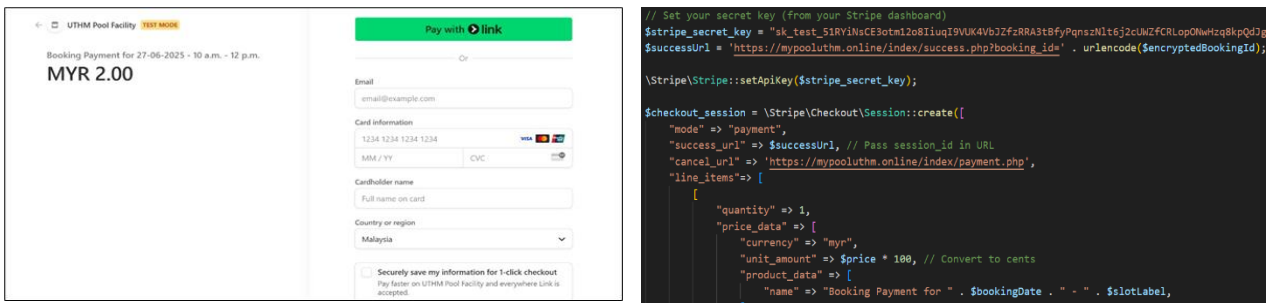


Fig. 20 *Payment Module*



(a) **(b)**

Fig. 21 *Stripe (a) Payment Gateway; (b) Coding for Payment Gateway*

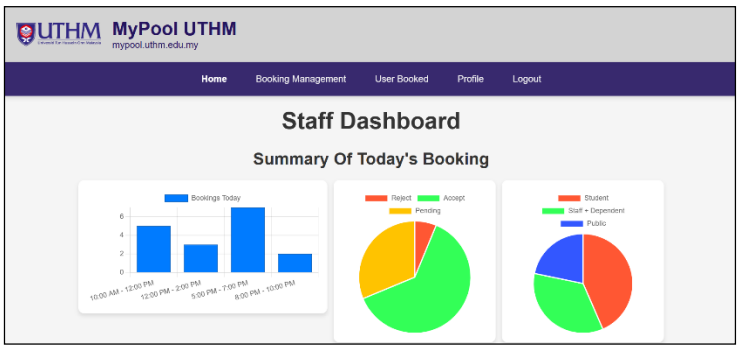
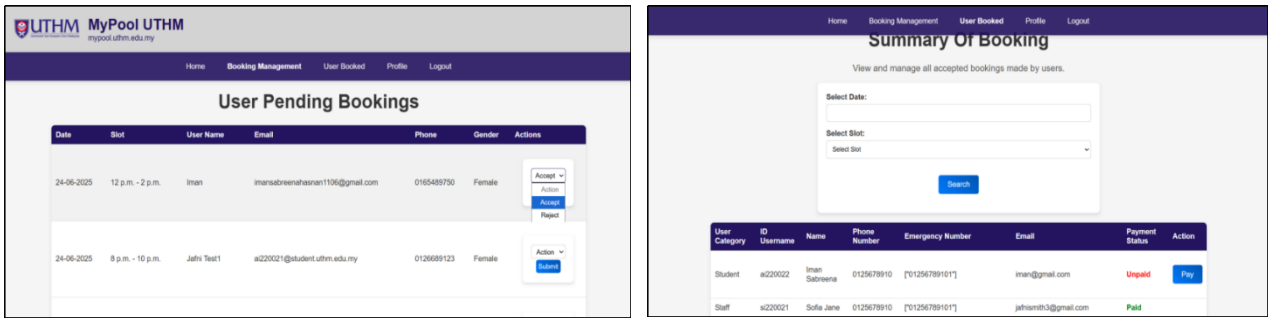


Fig. 22 *Staff Dashboard*

The Staff Dashboard in Fig. 22 shows staff an overview of daily bookings, including how many bookings exist for each time slot and their status (approved, rejected, or pending). It also displays user types like students, staff, or public users. The booking management module lets staff approve or reject bookings, view user details, and update booking statuses. When rejecting bookings, staff must provide reasons, and the system automatically sends email notifications to users about status changes. Hence, this helps staff manage bookings effectively and keeps users informed quickly.



(a) **(b)**

Fig. 23 *Staff Module (a) User Pending Booking; (b) Summary of Booking*

The Summary Module in Fig. 23 let the staff see detailed booking information for specific days or time periods within two months. Staff can filter bookings by date and time slots, and the system organizes records accordingly. It shows booking times, user details, and payment status from the database. User details are important for emergency assistance. Staff can also update payment status for users who choose to pay at the premises. Therefore, this gives staff a clear view of all bookings for easier daily management.

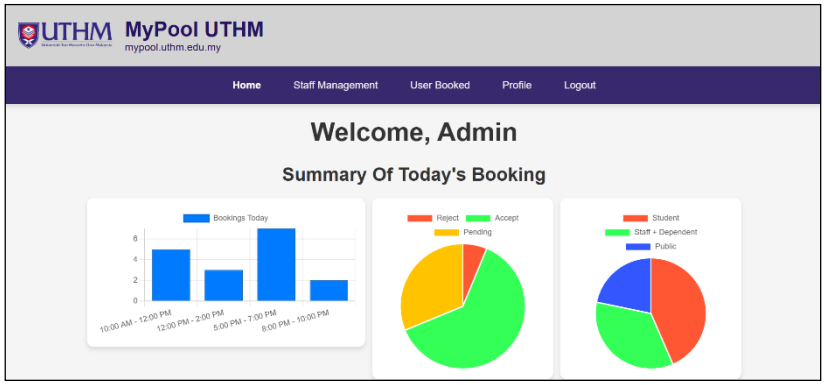
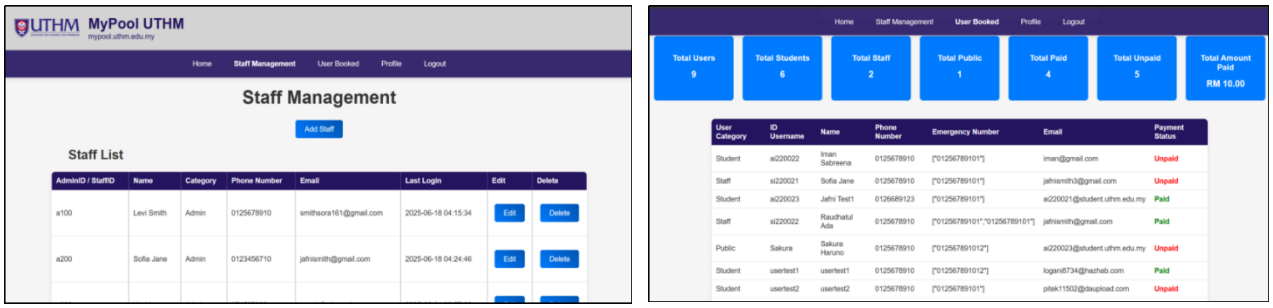


Fig. 24 *Admin Dashboard*

The Admin Dashboard as in Fig. 24 shows administrators real-time daily booking information organized by time slots, booking status, and user types. The Admin Summary Module provides a complete view of all approved bookings for any chosen date and time slot, including overall history. Each booking displays user details, categories, contact information, and payment status. Admins can update payment statuses directly from this page. Hence, this module makes booking management simpler by giving admins an easy way to track and update all bookings.



(a) **(b)**

Fig. 25 *Admin Module (a) Staff Management; (b) Summary of Booking*

5.3 Test Plan Result

5.3.1 Functional Test Plan Result

Table 7 *Functional Test Plan Result*

Application		Requirements	Result
Test registration page	User, Admin	Success: "Registration successful! Click here to login." Failure: Error message based on requirements (e.g., "Email already exists")	Pass
Test login authentication	User, Staff, Admin	Success: Redirect to dashboard. Failure: "Invalid login credentials!"	Pass
Test device verification	User	Success: "Device Status: Device Verified. Click button below to proceed." Failure: "Device Status: Device Not Verified. Please check your email to verify your device."	Pass
Test OTP verification	User Admin, Staff	Success: Redirect to payment gateway. Success: Redirect to dashboard. Failure: "Invalid OTP. Please try again."	Pass
Test booking page	User	Success: "Booking successful!" Failure: "You already have a booking for this date and time slot."	Pass
Test booking management page	Staff	Success: "Booking approved and user received email booking status." Failure: "Failed to approve booking."	Pass
Test staff management page	Admin	Success: "Staff added successfully!" Failure: Error message based on requirements (e.g., "Email already exists")	Pass

5.3.2 Security Test Plan Result

Table 8 *Security Test Plan Result*

Application		Requirements	Result
Authentication	Test Device-Based Authentication	When logging in from a new device, the device is initially unverified, and the system sends a verification email to the user.	Pass
		After the 5-minute countdown, if the device is still not verified, the user is redirected to the page before.	Pass
	Test reCAPTCHA Implementation	Success: "Device Status: Device Verified. Click button below to proceed."	Pass
		Failure: "Device Status: Device Not Verified. Please check your email to verify your device."	
	Test Email OTP Verification	Success: Redirect to payment gateway. Success: Redirect to dashboard. Failure: "Invalid OTP. Please try again."	Pass
User Input	Test Input Validation	If the user enters incorrect username or password, the system should show a generic error message like “Invalid login credentials!”	Pass
		If the user misses any required fields or enters invalid , the system should display a clear error message specifying the missing or invalid fields.	Pass
	Test Input Sanitization	The system should sanitize inputs to prevent SQL injection and Cross-Site Scripting (XSS).	Pass

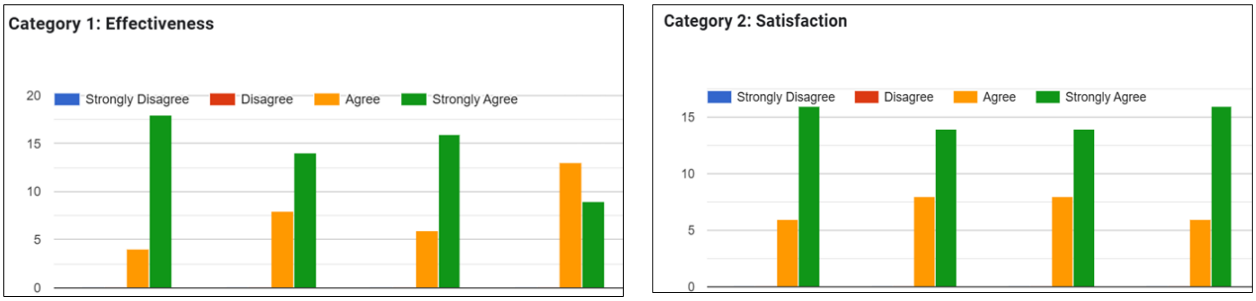
Table 8 *(Cont.)*

Application		Requirements	Result
Limit Login Attemps	Test with Brute Force	After 5 failed login attempts, the user should be temporarily locked out for 15 minutes. The system notifying them with lockout period.	Pass

Sessions	Test Session Termination	If the user is idle for 30 minutes, their session should automatically terminate, and they should be redirected to the login page. The system alert users 10 seconds before their session expires.	Pass
	Test Session Management	The system should handle multiple logins (same account, different devices) by ensuring session integrity. If another device logs in, the original session should be invalidated, or the user should be notified.	Pass

5.3.3 User Acceptance Form for User Module

The user acceptance testing with 22 respondents among UTHM students, staffs, and public user, showed very positive results for the pool booking system. Most users (18 out of 22) strongly agreed that the booking interface was easy to use and they could view their booking history without problems. Additionally, the payment process worked well, with 8 agreeing and 14 strongly agreeing it was straightforward. The OTP verification was also smooth, with 16 strongly agreeing and 6 agreeing about its effectiveness. Furthermore, users found the device-based authentication easy to use, with 13 agreeing and 9 strongly agreeing it was effective. Hence, users were highly satisfied with the registration, login, booking features, and overall security of the system.

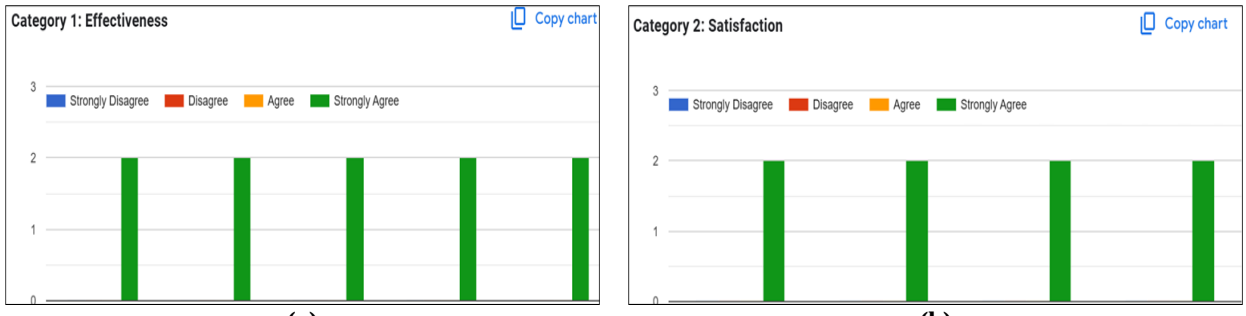


(a) **(b)**

Fig. 26 *User Questionnaire (a) Category Effectiveness; (b) Category Satisfaction*

5.3.4 User Acceptance Form for Pool Staff Module

Fig. 27 shows both staff members strongly agreed with all aspects of the staff module functionality and satisfaction. They could log in easily, change passwords after first login, and manage user bookings without any issues. The staff found it simple to approve or reject booking requests and track payment statuses effectively. Additionally, the OTP verification process worked smoothly for staff access. Therefore, both staff members were highly satisfied with the authentication process, booking management interface, and the user booking history features.



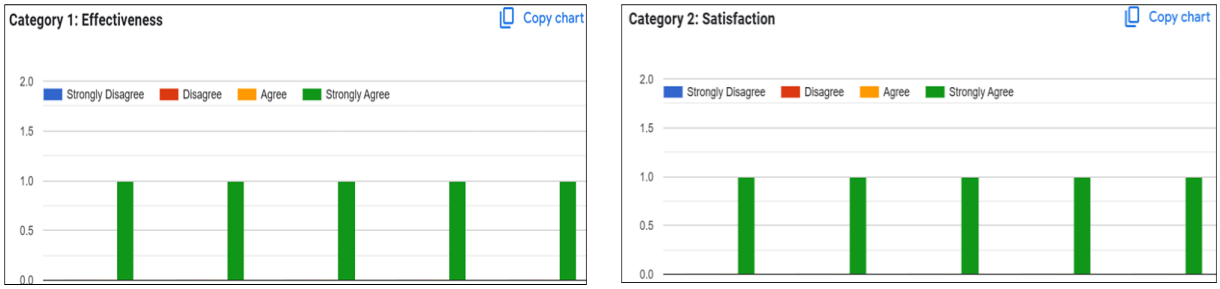
(a) **(b)**

Fig. 27 *Staff Questionnaire (a) Category Effectiveness; (b) Category Satisfaction*

5.3.5 User Acceptance Form for Admin Module

Fig. 28 shows the administrator strongly agreed with all functionality and satisfaction aspects of the admin module. They could log in effectively, manage staff accounts (create, update, delete), and monitor all bookings including payment statuses and usage patterns. The admin could also filter user booking records easily, and the OTP verification worked seamlessly. Furthermore, they were highly satisfied with the login process, staff

management features, and security systems including device-based authentication and OTP verification. Hence, the administrator was completely satisfied with the overall admin module experience and functionality.



(a) **(b)**

Fig. 28 *Admin Questionnaire (a) Category Effectiveness; (b) Category Satisfaction*

6. Conclusion

This project has successfully developed a web-based pool facility booking system for UTHM Sport Centre, transforming manual operations into an efficient digital platform. The implementation of device-based authentication and user access control has significantly improved facility security and management. The system successfully addresses previous issues by providing online booking capabilities, efficient user tracking, and proper access control for different user groups. However, the system presents some limitations, such as added complexity when UTHM staff user book on behalf of their family dependents, and a learning curve for staff and administrators to replace the manual system. Looking forward, future research could explore adding mobile app integration, implementing AI-based capacity prediction, and expanding the system to manage other sports facilities across campus.

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:*** *J. H. Zakaria, N. H. Ab. Rahman;* ***data collection:*** *J. H. Zakaria, N. H. Ab. Rahman;* ***analysis and interpretation of results:*** *M J. H. Zakaria, N. H. Ab. Rahman;* ***draft manuscript preparation:*** *J. H. Zakaria, N. H. Ab. Rahman. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] E. Lau, H. (Cynthia) Hou, J. H. K. Lai, D. Edwards, and N. Chileshe, "User-centric analytic approach to evaluate the performance of sports facilities: A study of swimming pools," *Journal of Building Engineering*, vol. 44, 2021, doi: 10.1016/j.jobe.2021.102951.

[2] Aurelio Maglione, "What Exactly Is an Online Booking System?" [Online].

[3] "Top 8 Benefits of Online Booking Systems for Business | Square." Accessed: Nov. 22, 2024. [Online]. Available: https://squareup.com/au/en/the-bottom-line/operating-your-business/benefits-of-online-booking-systems

[4] "What is Authentication? Different Types of Authentication." Accessed: Nov. 22, 2024. [Online]. Available: https://www.miniorange.com/blog/different-types-of-authentication-methods-for-security/

[5] "What is Authentication? | Definition from TechTarget." Accessed: Nov. 22, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/authentication

[6] "Use these 6 user authentication types to secure networks | TechTarget." Accessed: Nov. 23, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks

[7] "Device Recognition vs. User Authentication: What's the Difference?" Accessed: Nov. 22, 2024. [Online]. Available: https://www.lansweeper.com/blog/itam/device-recognition-vs-user-authentication/

[8] "Device Fingerprint: how does it work and what can it do?" Accessed: Nov. 27, 2024. [Online]. Available: https://trustdecision.com/resources/blog/what-is-device-fingerprint-how-does-it-work#getDem

[9] "Swimply - Rent Private Pools, Courts, and More by the Hour - Pools Near Me." Accessed: Nov. 23, 2024. [Online]. Available: https://swimply.com/

[10] "WET Deck | W Kuala Lumpur." Accessed: Nov. 24, 2024. [Online]. Available: https://www.wkualalumpur-wetdeck.com/

[11] "Swimming Pool by Aerotel Singapore (Transit Area, Terminal 1) | Plaza Premium Lounge." Accessed: Nov. 24, 2024. [Online]. Available: https://www.plazapremiumlounge.com/en-uk/find/asia/singapore/singapore/singapore-changi-airport/aerotel-singapore

[12] V. Yakovyna, M. Seniv, and I. Symets, "The Relation between Software Development Methodologies and Factors Affecting Software Reliability," in *International Scientific and Technical Conference on Computer Sciences and Information Technologies*, 2020. doi: 10.1109/CSIT49958.2020.9321937.

[13] "What is MoSCoW Prioritization? | Overview of the MoSCoW Method." Accessed: Dec. 06, 2024. [Online]. Available: https://www.productplan.com/glossary/moscow-prioritization/

[14] "SDLC methods and their advantages and disadvantages | by Sivasubramaniam Elankumaran | Medium." Accessed: Dec. 06, 2024. [Online]. Available: https://siva98kumarane.medium.com/sdlc-methods-and-their-advantages-and-disadvantages-ded47a32f1b1

[15] Prisca Amajuoyi, Lucky Bamidele Benjamin, and Kudirat Bukola Adeusi, "Agile methodologies: Adapting product management to rapidly changing market conditions," *GSC Advanced Research and Reviews*, vol. 19, no. 2, pp. 249–267, May 2024, doi: 10.30574/GSCARR.2024.19.2.0181.

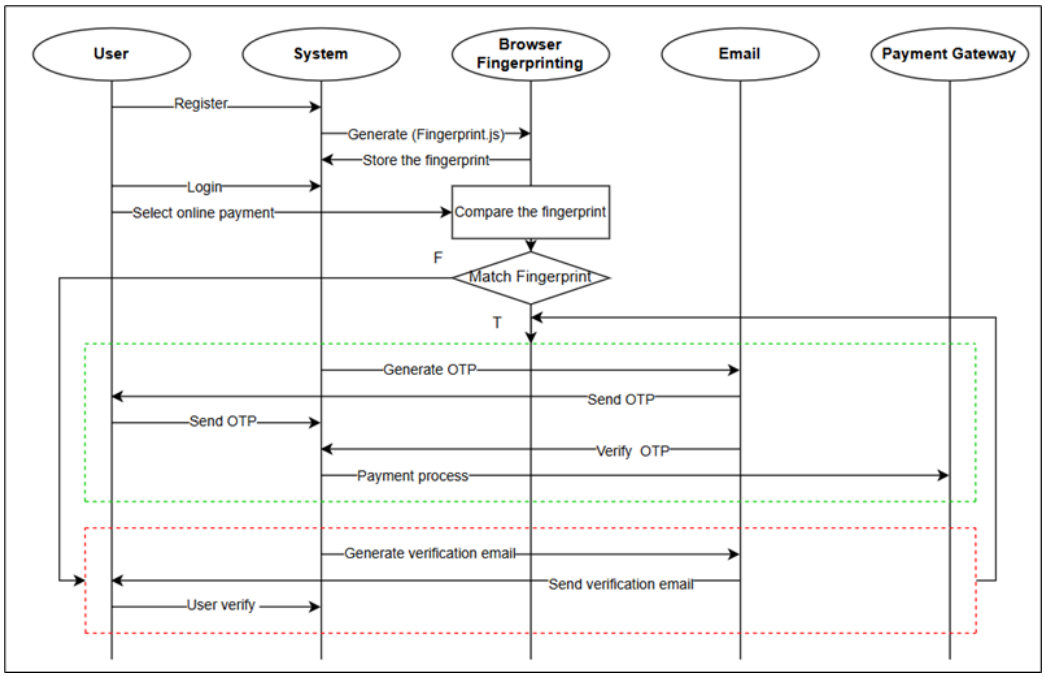
[16] "What is Agile software development? | Definition from TechTarget." Accessed: Dec. 06, 2024. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

[17] "The Agile System Development Lifecycle (SDLC)." Accessed: Dec. 06, 2024. [Online]. Available: https://ambysoft.com/essays/agileLifecycle.html#Deploy

[18] "CheatSheetSeries/cheatsheets at master · OWASP/CheatSheetSeries · GitHub." Accessed: Jun. 09, 2025. [Online]. Available: https://github.com/OWASP/CheatSheetSeries/tree/master/cheatsheets

[19] K. Papadamou et al., "Killing the Password and Preserving Privacy with Device-Centric and Attribute-based Authentication."

Appendix A: Device Based Authentication



Appendix B: Gantt Chart