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

AITCS

Vol. 7 No. 1 (2026) p. 1051-1070
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Implementation of Swimming Class Booking Management System with Role-Based Access Control (RBAC) and Multi-Factor Authentication (MFA) For M&M Swimming Club

Nur Iman Sabreena Hasnan[1], Nur Ziadah Harun[1]*

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

*Corresponding Author: nurziadah@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.057

Article Info

Received: 17 July 2025
Accepted: 15 June 2026
Available online: 30 June 2026

Keywords

Web-based, administrative, Flexibility, Adaptability, Security

Abstract

In this era of technology, a web-based booking management system for classes or facilities has broadly been used by many sport clubs because it provides flexibility and adaptability to users. The paper-based method used by M&M Swimming club for tasks such as booking swimming classes and managing swimmers' attendance, increases higher risk for data loss and less flexibility for data resources. Therefore, A web-based booking system was developed using Agile methodology and embedded Role-Based Access Control (RBAC) with roles such as swimmer, coach, and admin to manage user privileges and Multi-Factor Authentication (MFA) using OTP via email to authenticate the users before gain access. The results have shown a successful implementation of the functionality and security of the system. The user acceptance result showed positive satisfaction with the usability and security of the system. In conclusion, the system conveniently assists users in booking swimming classes and keeping their data safe.

1. Introduction

The paper will discuss web-based management which is an essential element to manage daily tasks of an organization, including swimming clubs. As for The Mermaid & Mermen (M&M) Swimming Club, the administrator manages tasks such as membership registration, payment record, scheduling classes, and notifying events, all achieved using traditional methods. All these duties are quite out of hand if there is no proper management system. In which this will lead to inefficient handling of the administrative tasks. Therefore, a well-designed web-based online platform is needed to optimize the workflow of the club owner and staff. The web will have elements to assist them in completing their tasks. The implementation of security measurements for the web-based online platform will create a good impression from the club and users. Some of the security elements that can be embedded include multi-factor authentication (MFA) via one-time password (OTP), role-based access control (RBAC), hashing and more. These measurements will help to prevent unauthorized access from unwanted parties by verifying one's identity before accessing the web page and ensuring each entity has a suitable role to access the system with certain privileges. This will help to prevent exposure to sensitive information and other potential threats to the system. Thus, it will make the system look secure and genuine, which is important to gain the trustiness of the owner of the club and the users.

In summary, the project aim is to design, develop and test a swimming class booking management system that accommodates the admin, coach, and swimmer of M&M Swimming Club which highlights the importance of security features such as RBAC and MFA to ensure it will not violate any of the Confidentiality, Integrity, and Availability (CIA) triad of information security. CIA triad is a model of security strategy which includes guidelines

to developers before and during the project development process to filter out threats according to each element [1]. The main issue is to provide a stable web-based platform and eliminate inefficiency of manual methods used by the club.

2. Related Work

This section will discuss more about the swimming class booking management system along with security terms related to the project. This includes topics related to the sport club booking system, access control, authentication, security features, and comparison between existing systems.

2.1 Sport Club Booking Management System

A sport club booking management system is a structure that assists users to reserve facilities such as classes, courts, and equipment and ensuring efficient resource management and administrative tasks handled by manager [2]. For example, managing registration of new members, managing financial records for class fees, and recording athlete progress. It is important to have an efficient management system that helps us to collect, store and analyze data of users and at the same time identify the flow of the information as it will help in maintaining the operation of the sports club and provide satisfactory services to athletes and members of the club.

2.2 Role-Based Access Control (RBAC)

RBAC enables authorization of users to the system by determining each user's role. Every user has certain privileges that are aligned with the assigned role. For example, Alice holds a role as an admin, while Bob is a clerk. Bob may access the data entry and manage documents, but Bob is unable to access and perform other tasks outside of his job scope such as managing users, which is more to Alice's task. While Alice has the highest privilege in the system where she can perform more detailed tasks such as updating and managing the system. According to researchers, RBAC provides advantages to a large company that requires optimization in the operational flow [3]. This is because RBAC ensures it has great flexibility to adapt to new changes, it is easy to implement and understand, and provides accessibility to users according to their role, which can protect systems from being accessed by malicious actors. Therefore, this is the model that has been chosen to be embedded in the M&M Swimming Club Management System to prevent any breaches.

2.3 Multi-Factor Authentication (MFA)

MFA is a combination of more than one of SFA, it provides authentication by requesting users to provide a second source of validation before granting access to the system. MFA is needed especially for organizations who use digital platforms as a place to store and manage sensitive data. MFA consists of what you know, what you own and what you are. One example of MFA is a combination of password and a string of temporary passcodes which are a piece of possession that users have, to prove their claims. This temporary passcode is called OTP, and it is generated randomly at one login session. According to research, substituting SFA or 2FA to MFA, will reduce risk to 99.9% over password alone [4].

Table 1 *Components in MFA [5]*

Categories	Components	Methodologies
Knowledge	What you know	Passwords, digital signatures
Possession	What you own	OTP, smartcards
Inherence	What you are	Biometrics

Table 1 shows the three main components in Multi-Factor Authentication (MFA) which are knowledge, possession, and inherence. For knowledge, it refers to something the user knows such as passwords that the user themselves creates and digital signatures. Next, possession is what the user owns including smartcards and One-Time Password (OTP) that user can receive via email, SMS, or authenticator apps such as Google Authenticator. Lastly, inherence refers to something the user is, including biometrics traits such as fingerprints or facial recognition. Therefore, MFA combines two or more of these factors to provide multi-layer security that would make it harder to penetrate through the system.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1051-1070

2.4 M&M Management System

Almost every management task such as recording attendance, managing class schedule, and student's information are done manually. For example, to record swimmers' attendance, the process is done manually using papers. Fig. 1(a) shows how swimmers take their attendance before the swim lesson starts. Each swimmer will be categories based on the initial alphabet of their name and is separated by different file colors. For example, swimmers whose names start with 'M' will be in one file in yellow just like shown in Fig. 1(b).

(a) (b)

Fig. 1 *M&M Swim Club system (a) Paper-based; (b) Files for Attendance*

2.5 RNJ Swimming Academy

RNJ Swimming Academy has started their web-based management approach since 2016 [6] and has been ongoing until now. This system has a target user of coaches and swimmers, where the system aims is to assist users in achieving desired tasks such as registering new students, scheduling class and managing users. RNJ Swimming Academy has embedded Single Factor Authentication (SFA) method of authentication where users are required to insert username and password to verify their identity to access the system. There is no signup module, as new users who would like to register will be required to contact the staff themself via mobile phone or email.

This system has the advantage of simplicity and is adaptable for new users, the authentication method is just a short-period time and access to the system is easier than systems that implement MFA. Unfortunately, from aspect security, it will be considered as a bad practice especially if users have weak passwords, repeatedly use, and rarely change password. Attackers could easily crack passwords or do brute-force attacks to gain access of the account and do malicious acts.

2.6 USA Swimming SWIMS 3.0

USA Swimming is an organisation that manages all competitive swimmers and clubs across the United States and has been operating since 2002 until current. The system has undergone several redesign and updates since 2005 [7] and has achieved a latest version which is SWIMS 3.0. This system uses email verification and there will be a push-up notification for users to click to rest the forgotten passwords. In the signup module, this system implemented a strong password policy along with input validation before user signing in. In aspect, of access control, as for large organisations and in enterprise-level, the access-control is still basic and does not have the full features of RBAC. The use of push-up notification offers convenience and time saving for users who do not want an additional application for authentication, except users need to have a good internet connection so that the notifications can be received. This system also implements strong password policy, and input validation can prevent the system from attacks such as SQL injection and stolen identity credential.

2.7 Comparison with the Existing Systems

Table 2 shows the comparison between the existing system and proposed system. This table differentiates each system by comparing the features in each system along with the security element embedded. Based on Table 2, both existing systems have a login/sign up page except RNJ Swimming Academy that doesn't have a signup page. Every system except M&M Swim Club has database. Both M&M and RNJ Swimming Academy have no implementation of OTP and RBAC for authentication and access control, and strong password policy of the system except USA Swimming SWIMS 3.0. Next, RNJ Swimming Academy and USA Swimming SWIMS 3.0 have a similar goal to create a web-based platform to improve efficiency of the club management system, while M&M Swim Club is still using paper-based approach. Therefore, the proposed system will have similar features just like existing systems but with additional features such as payment record and improving the security defense

by implementing both OTP and RBAC into the proposed system.

Table 2 *Comparison of existing system*

Features In System	M&M Management System	RNJ Swimming Academy Management System	USA Swimming SWIMS 3.0	Proposed System
Login/signup	No	Yes (No signup)	Yes	Yes
Strong password policy	No	No	Yes	Yes
Database	No	Yes	Yes	Yes
RBAC	No	No	Yes	Yes
OTP	No	No	No	Yes
Platform	Paper-based	Web-based	Web-based	Web-based

3. Methodology

This section will discuss methodology that will be used to develop the proposed system. The chosen model for this project is agile development framework which is an approach that focuses on flexibility and enhances users experience and satisfaction by clarifying user's requirements from time to time. It is an iterative process and provides flexibility where developers can repeat the same process every time the prototype needs improvement required by the clients. Fig. 2 shows the Agile method used in this project.

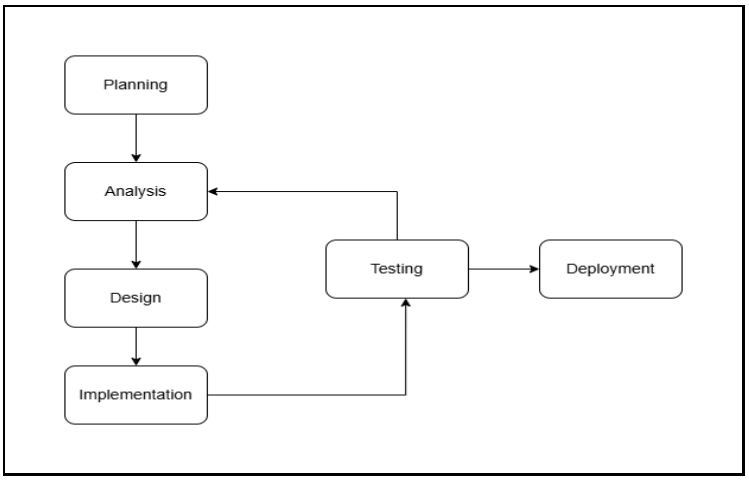


Fig. 2 *Agile methodology framework [8]*

Fig. 2 shows the flow of agile development methodology in developing a system. This method consists of six phases which are planning, analysis, design, implementation, testing, and deployment. Planning phase is a fundamental process of understanding the reason for building an information system and how to implement it [9] During this phase, it has been decided that M&M Swimming Club requires a management system which will help them to optimize their operation. This phase has been completed as all the information required to develop swimming class booking management system along with security features has been documented in the proposal, this includes problem statements, objectives, scope of users and system, and the importance to pursue this development. A timeframe of the project has been scheduled using Gantt Chart in Appendix A.

Analysis phase involves studying existing systems that relate to the M&M Swimming Club such as the USA Swimming club and RNJ Swimming club management system. Next, is also gathering details of all the requirements needed to build the proposed system such as users which are admin, coaches, and swimmers, along with functional and non-functional requirements. It also includes hardware and software requirements such as the type of languages used for developing the front-end and back-end of the system, the framework and what kind of database is applicable to the proposed system.

Design phase is where the proposed system components are designed such as database, interfaces and proposed system model. It is also where the security strategy is designed such as the access control matrix for Role-Based Access Control (RBAC) in the proposed system. The interfaces are designed with a low-fidelity wireframe and must be user-friendly to support users completing their tasks and improve user experience. The interface design will be drawn in *excalidraw* which is designing wireframe tools. For database design, the data

model will be designed using Entity Relationship Diagram (ERD) along with normalization technique. Lastly, Data Flow Diagram (DFD) will be drawn from a context diagram up to level 1.

Implementation phase is where all documented and discussed information in previous phases is carried out physically. Once the design phase is done, it is time to interpret it into a human readable and accessible interface. In this project, MySQL and PHP will be used as the back-end languages that handle the database. For the front-end, this project will use HTML, CSS, and JavaScript. This is done by thoroughly programming using safe code practice which aligns to OWASP secure practice checklist [10].

Testing phase is where the developers decide whether to repeat the cycle or to proceed to the deploy phase. It consists of several tests such as User Acceptance Testing (UAT), using a list of questionnaires for the owner of the club and using google form for coaches and swimmers as it requires many users. A functional and security test will also be done for users who will use the system and for this proposed system, which is to the M&M Swimming Club admin, coaches, and swimmers.

For deployment phase, once tests have been done and passed successfully, next is to gain agreement with the stakeholder to release the system using a domain. A final meeting can be held with M&M Swimming Club owner to determine whether to release the system or give more space for improvement. After the system has been released to the server, it is necessary to ensure the system is reliable and maintainable by updating frequently once vulnerabilities are detected and providing patches so that the proposed system won't be exposed to cyberattacks.

4. System Analysis and Design

In this section will discuss more further regarding M&M swimming class booking management system analysis and design.

4.1 System Development Workflow

The total phases of Agile development methodology are six. Table 3 shows the summary of each phase and the outcome that should be completed before, during, and after project development of the proposed system.

Table 3 *System Development Workflow*

Phase	Task	Outcomes
Planning	• Decide a suitable system to overcome M&M swimming club booking and data management issues. • Proposed a documented information about the proposed system specification. • Provide a time frame to coordinate activities in developing the system.	• Proposal • Gantt Chart
Analysis	• Identify requirements needed • Study existing system	• List of requirements such as user, functional, non-functional, and software and hardware requirements
Design	• Design DFD to show the system flow • Design ERD for data model • Design system architecture • Design user interface • Decide access control matrix for each role	• DFD diagram • ERD diagram • Proposed architecture • Access control matrix • Wireframes
Implementation	• Developing the proposed system	• Prototype
Testing	• Issued tests to the M&M Swimming Club Users.	• User Acceptance Testing • Functionality and security test
Deployment	• Improve the prototype • Final consent with M&M Swimming Club to release the system	• System able to run smoothly

Table 3 shows the outline of the system development workflow based on Agile methodology. The process begins with the first phase that is planning phase where the proposal and Gantt Chart that consists of timeline of the project has been produced. Then it will be followed by analysis phase where the list of requirements such as user requirements, functional and non-functional requirements, software and hardware requirements has been defined based on the questionnaire and existing system studies. The next phase is the design phase where DFD, ERD diagrams, system architecture, access control matrix and wireframes have been created before implementing the system. Implementation phase involves developing the system using technologies such as PHP, HTML, CSS, JavaScript, and MySQL. The next phase is the testing phase where functionality, security, and user acceptance test are carried out to ensure the system works as planned. The last phase is the deployment phase that focuses on ensuring the system able to run without problem. This workflow ensures a structured and flexible approach to reach the objective of the system.

4.2 System Requirements

Requirements analysis involves gathering details to determine what the system is required to do, expectation of the system, and where improvements can be made. This will assist users to complete their task using the system effectively and will improve user experience. Requirement analysis includes user requirements, functional requirement, and non-functional requirement.

Table 4 *Functional Requirement*

No.	Requirement	Users
1.	Users should be able to login to the system.	Admin, coach, swimmer
2.	Users should be able to create an account if they do not have an existing account.	Admin, coach, swimmer
3.	The system should be able to send One Time Password (OTP) to email for verification of the user and changing password.	Admin, coach, swimmer
4.	The system should separate access levels for each user.	Admin, coach, swimmer
5.	Users should be able to book and reschedule swim class.	Swimmer
6.	Users should be able to set their availability in schedule.	Coach
7.	The user should be able to receive notifications about events and competition.	Coach, swimmer
8.	The user should be able to manage user, swim class schedule, payment record, and notifications	Admin

Table 5 *Nonfunctional Requirement*

No.	Requirement	Description
1.	Usability	The system should provide a user-friendly interface that improves user experience while completing their task.
2.	Security	The system must provide assurance in securing system and data privacy in terms of confidentiality, integrity, and availability.
3.	Performance	The system should be able to perform optimum tasks and support users in completing tasks over a period.
4.	Reliability	The system should be functioning as expected without failure by keeping updating and maintenance.
5.	Compatibility	The system should be compatible with devices and common web browsers such as Chrome.

4.3 System Analysis

System analysis is a process to understand system structure, functions, and how it's supposed to behave to assist users to complete their task. This includes visualizing user requirements into diagrams such as Data Flow Diagram (DFD) and Entity Relationship Diagram (DFD) that will be explained further in the subsection outlined.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1051-1070

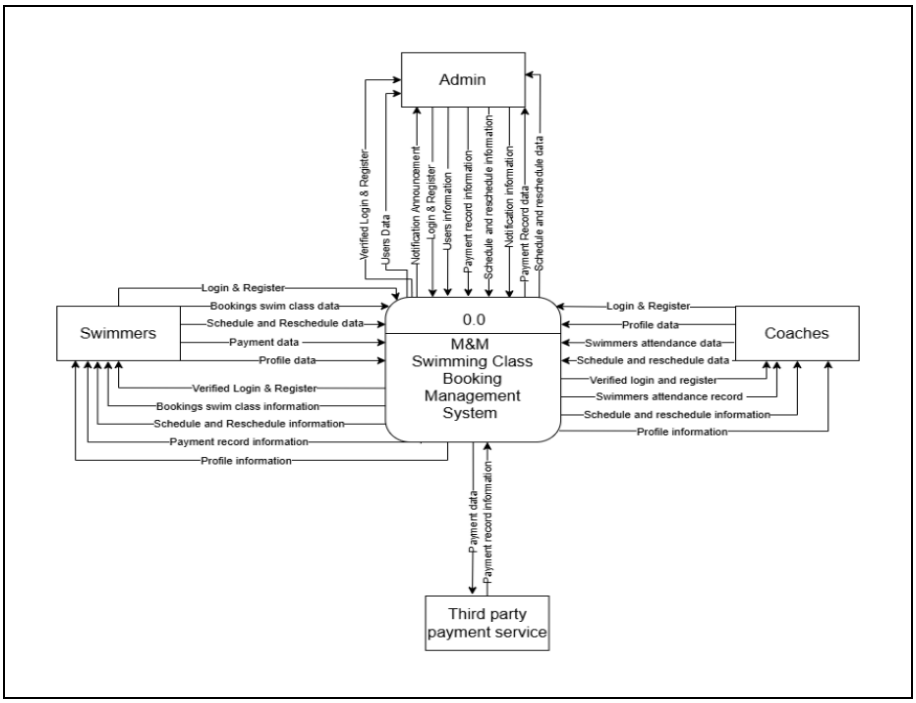
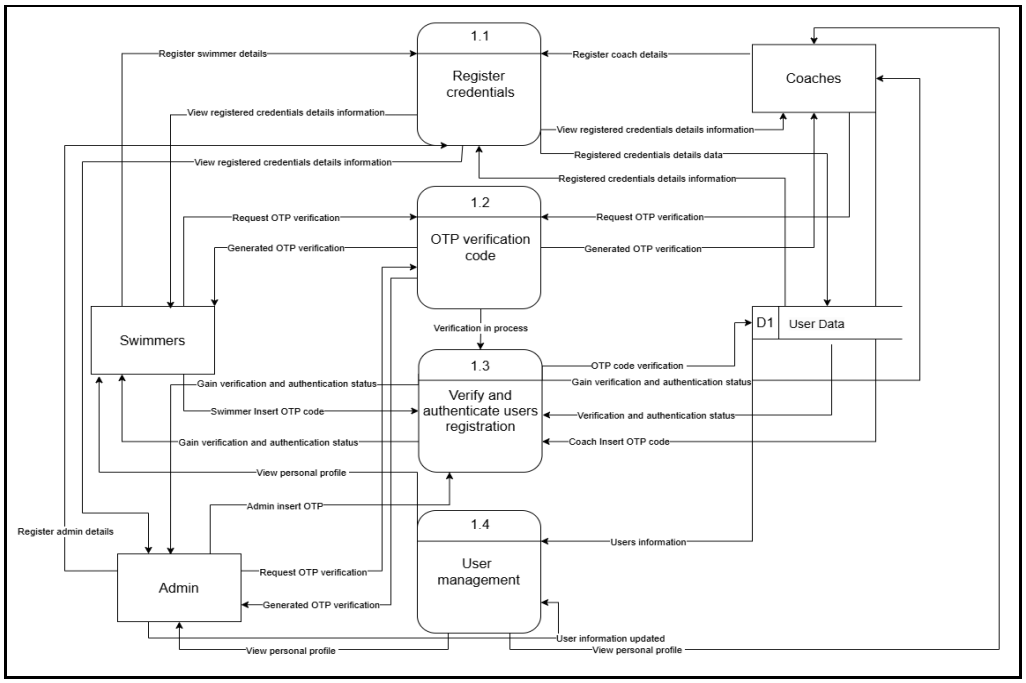


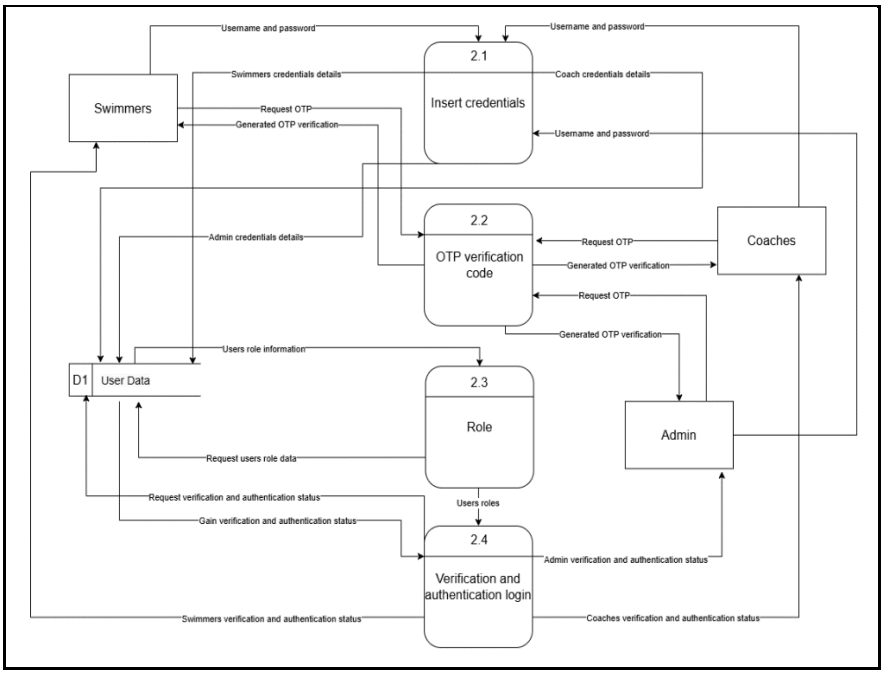
Fig. 3 *Context diagram for M&M Swimming Class Booking Management System*

The context diagram in Fig. 3 shows the main flow in the M&M Swimming Club Management System. It consists of four entities which are swimmers, admin, coaches, and third-party payment service. To access the system, all users are required to login and register. For swimmers, they will be able to book swim classes, view and edit profile data, book swim class schedules. Reschedule is also available if they want to change the swim class slot. For admin, they will be able to manage user's data, update and insert event announcements, manage payment record data, and manage schedule and reschedule class data. Coaches should be able to track swimmer attendance and create and view class schedules.

Fig. 4(a) shows DFD Level 1 for registration where all users must register before accessing the system. There are four processes in this level which are register credentials, OTP verification code, verify and authenticity user registration and user management. Fig. 4(b) shows DFD level 1 for the login module which consists of credentials requirements, OTP verification code, role, and verification and authentication login processes. Once users are registered, they are required to login to the system by inserting credentials info such as username and password, along with OTP code verification via email.



(a)



(b)

Fig. 4 *Level 1 DFD (a) registration module; (b) login module*

Fig. 5 shows ERD for database design. ERD is a visual representation of how each entity or table relates to each other. ERD contains components such as entities, attributes, and relationship of each entity.

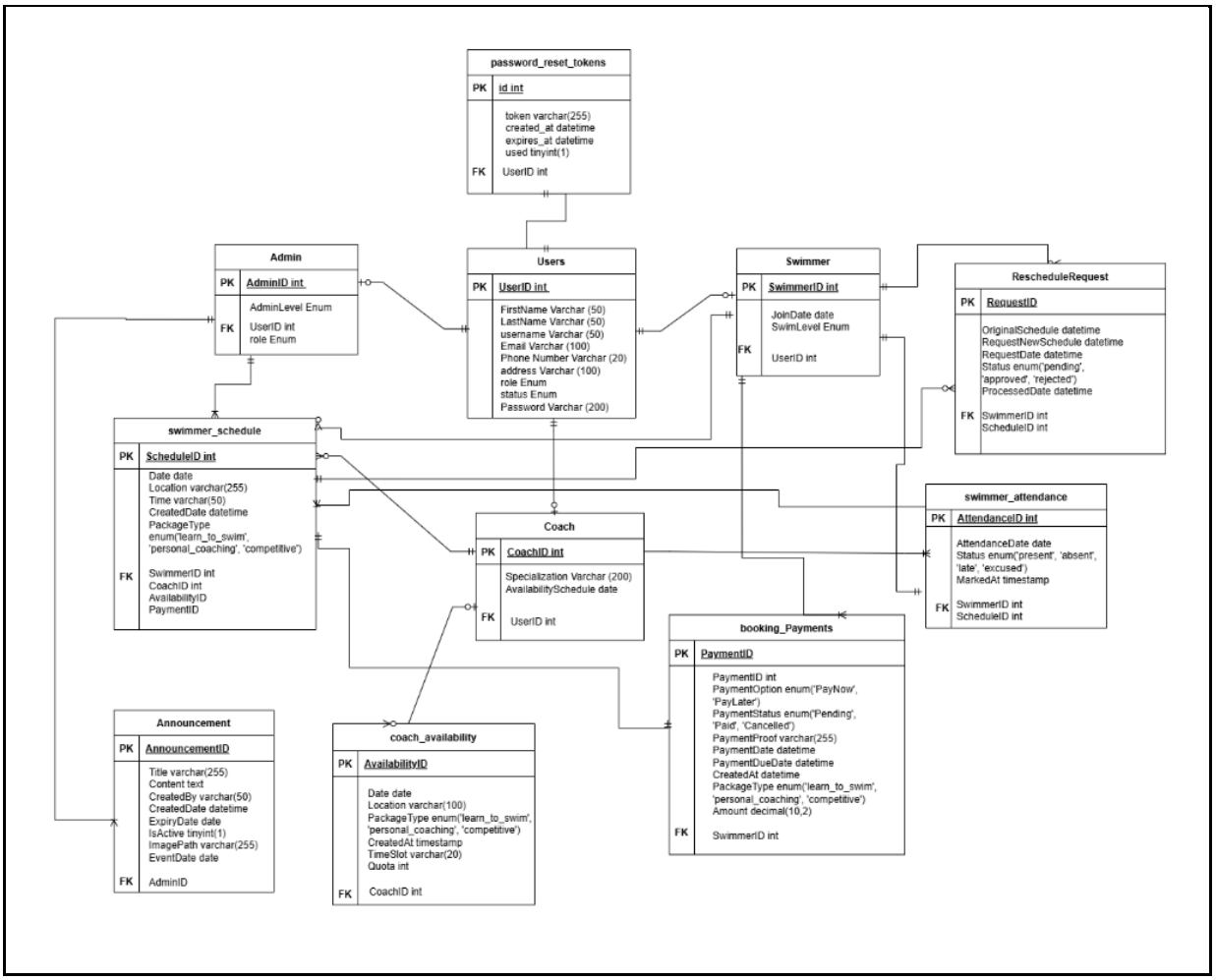
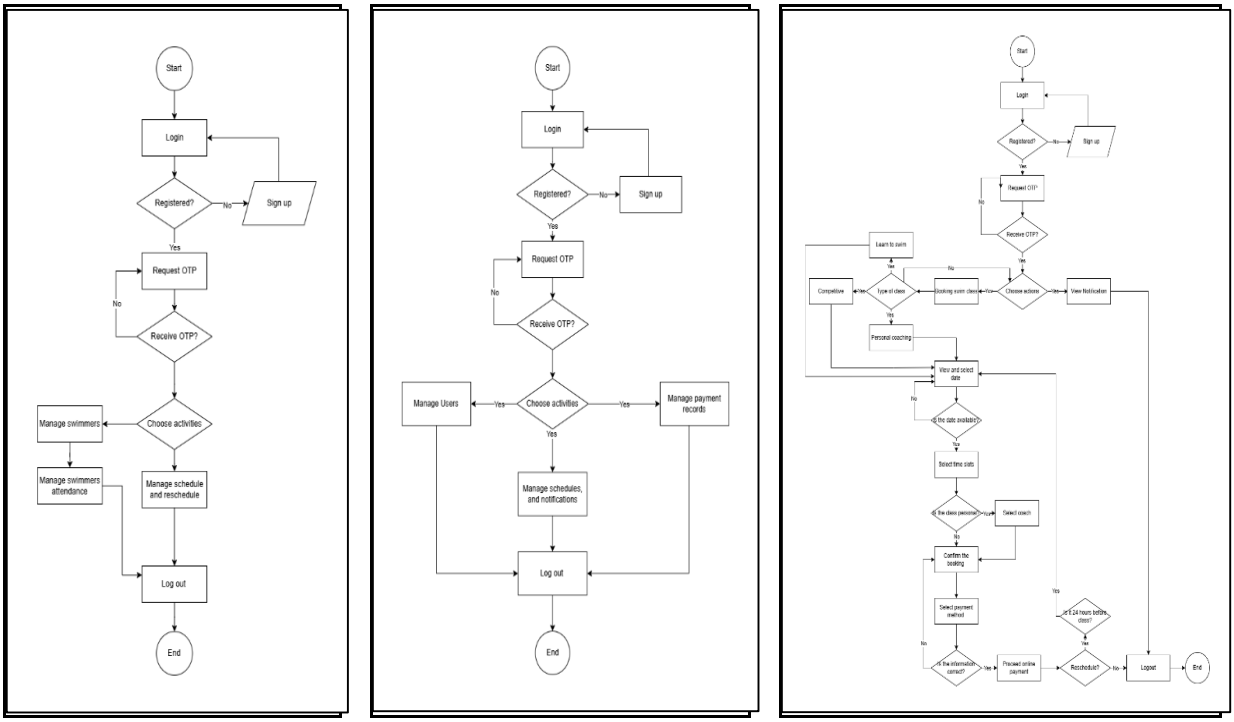


Fig. 5 *ERD of M&M Swimming Class Booking System*

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1051-1070

Fig. 6(a) shows flowchart for coaches. After successfully login to the system, coaches can do activities such booking availability for swimming class and reschedule it and tracking their swimmer's attendance. Fig. 6(b) shows flowchart for admin where they will be able to manage users, manage booking class records for both swimmers and coaches, manage notification, and payment records. Swimmers will be able to do tasks such as booking swimming class and reschedule it and making online payments as shown in Fig. 6(c).



(a) (b) (c)

Fig. 6: Flowchart (a) Coaches; (b) Admin; (c) Swimmer

4.4 System Design

System design is a process of visualizing the architecture, components, modules, interfaces, access controls and data flow of the system to ensure the requirements are fulfilled. Fig. 7 shows system architecture of M&M Swimming Class Booking Management System which shows the flow of users once successfully access to system.

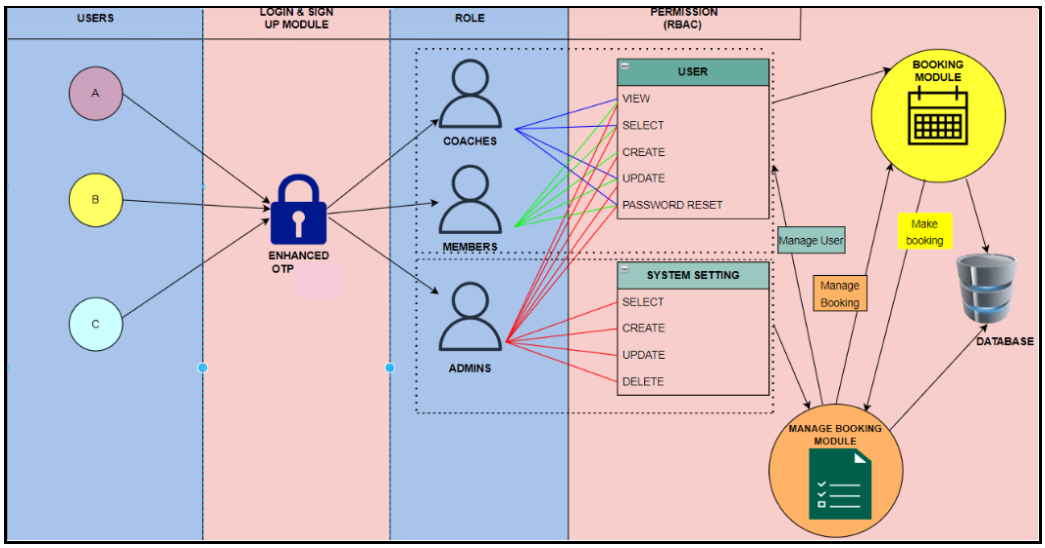


Fig. 7 *System architecture M&M Swimming Class Booking Management System*

The Access Role Matrix Table represents a row of subject or user and a column of object or function. The table will define permissions or access rights that are required according to the user's role just like shown in Table 6. There are three types of access rights, that is read which is represented in *r* and write represented in *w*,

Table 6 *Access Control Matrix*

Function\Users	Admin	Swimmer	Coach
Register and login	*rw*	*rw*	*rw*
Personal profile	*rw*	*rw*	*rw*
Booking Swim class	*rw*	*rw*	*rw*
Schedule and reschedule swim class	*rw*	*rw*	*rw*
Perform payment	*rw*	*rw*	---
Manage Users	*rw*	---	--
Manage announcements	*rw*	---	---
Manage schedule and reschedule swim class bookings	*rw*	---	---
Manage payment records	*rw*	---	---

5. Implementation

This section will discuss the implementation phase of the system. The implementation includes of security modules including RBAC and MFA, user modules which consist of three users that is swimmer, coach, and admin.

5.1 Implementation of Security Modules

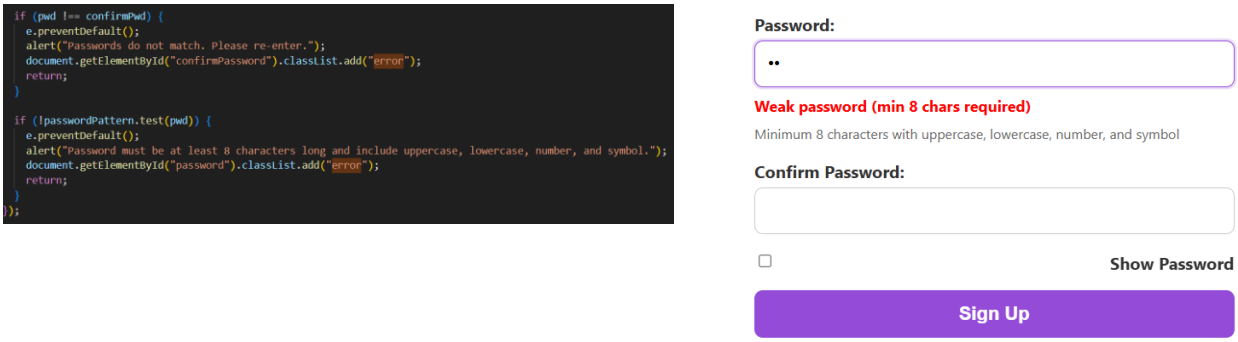
This section will discuss more onto security measurements implemented in the proposed system. Firstly, is the implementation of input validation in both sign up and log in modules. Input validation assists in preventing SQL injection in which happens if the attacker able to launch attack by putting malicious code in the input field if the fields are not validated. Input validation is not only implemented in sign up and log in form, but in any field that can be entered such as booking form field and user profile. Fig. 9(a) shows how input validation is set up in PHP which is in server-side and Fig.9(b) is where input validation shown in client-slide by using JavaScript. This prevents users from inserting malicious input in the form field and avoid gaining access to the system.

(a)

(b)

Fig. 9 *Input Validation (a) in PHP (b) in JavaScript*

The proposed system also has implemented strong password policy in where users are required to follow restricted rules in creating passwords. The password must have minimum of 8 characters with combination lower and uppercase, number and special symbol. Fig. 10(a) shows the server-side script to prevent users from gaining access if the password has not complied with the rules. Fig. 10(b) shows warning error to users when they enter simple password.



(a) (b)

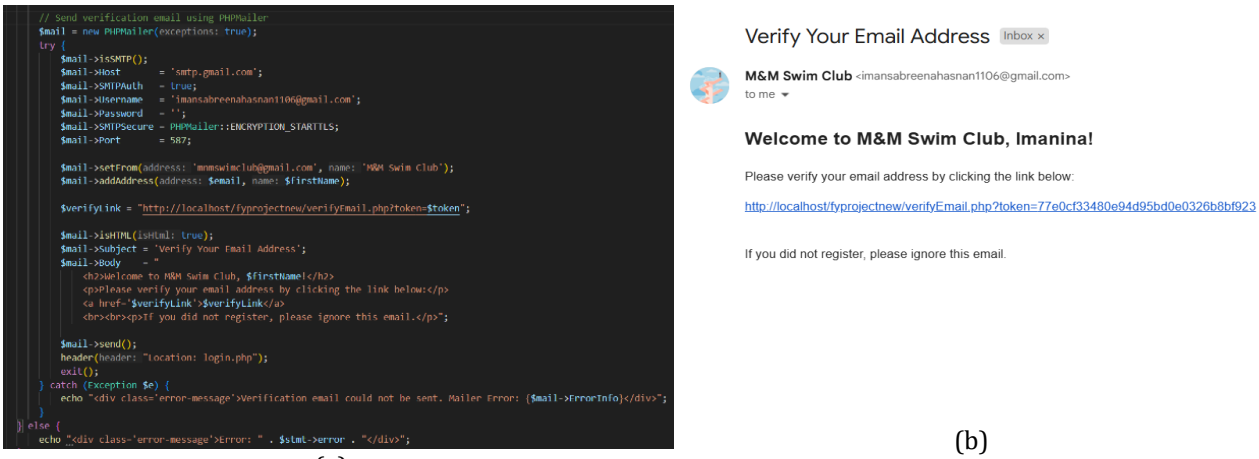
Fig. 10 *Strong Password Policy (a) in PHP (b) in sign up page*

Next, the system also implemented password hashing in the system during the signup module. Fig. 11(a) shows server-side scripting that protects password by converting them into a fixed-length string of characters by using cryptographic hash function. It is also combined with salting technique that prevents users to have same hash value even though they have same passwords.

(a) (b)

Fig. 11 *Hashed Password (a) in PHP (b) in database*

The proposed system also has embedded email verification to confirm that the user's email is valid and belongs to the correct owner. Fig. 12(a) shows how email verification is sent to user by using PHPMailer that connects to SMTP server using secure authentication protocols such as TLS. Fig. 12(b) represents user's email inbox when they receive email verification link that will then label them as verified before redirecting them to login page.

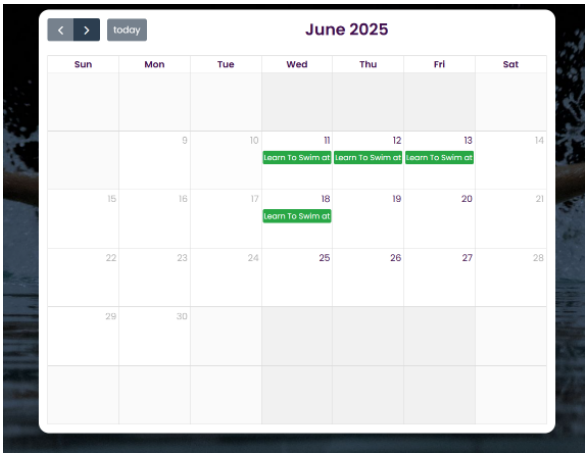


(a) (b)

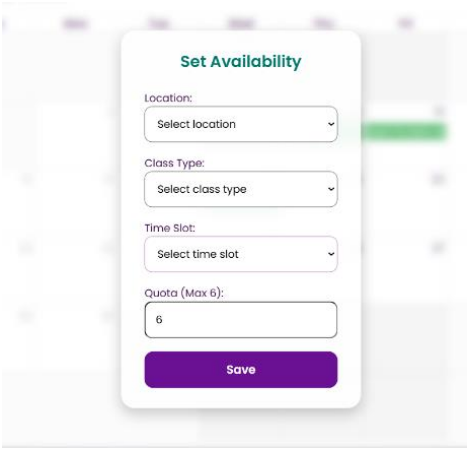
Fig. 12 *Email Verify (a) in PHP (b) in user's email*

5.2 Implementation of Coach Module

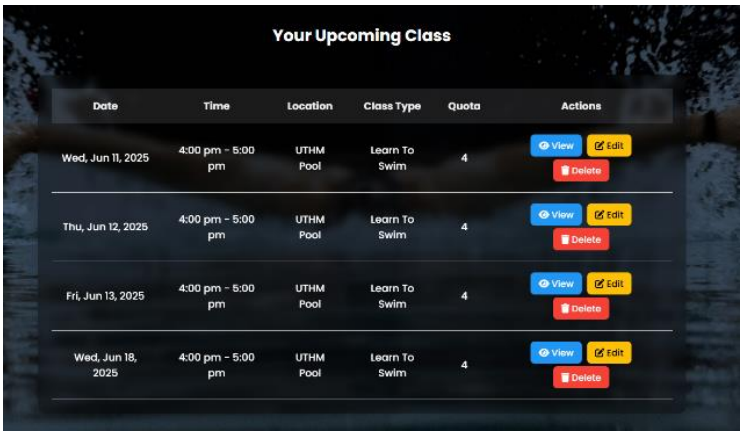
The implementation of coach module designed is to update coach availability for swimming class. It also includes user dashboard, booking class availability, receiving notifications, and update swimmer attendance. To book class availability, coach will need to go to page book availability and will display calendar as shown in Fig.13(a). Coach will need to choose any day from Wednesday until Friday. Past dates and any day besides the stated will be blocked. Then coach will need to fill in form including location, class type, and time slot as shown in Fig. 13(b). The quota has been set maximum to 6 swimmers per slot. Once the form has been filled, coach will need to click submit button and the calendar will be updated if the booking is successful. The coach availability which represents upcoming class will be display in coach dashboard as in Fig. 13(c).



(a)



(b)



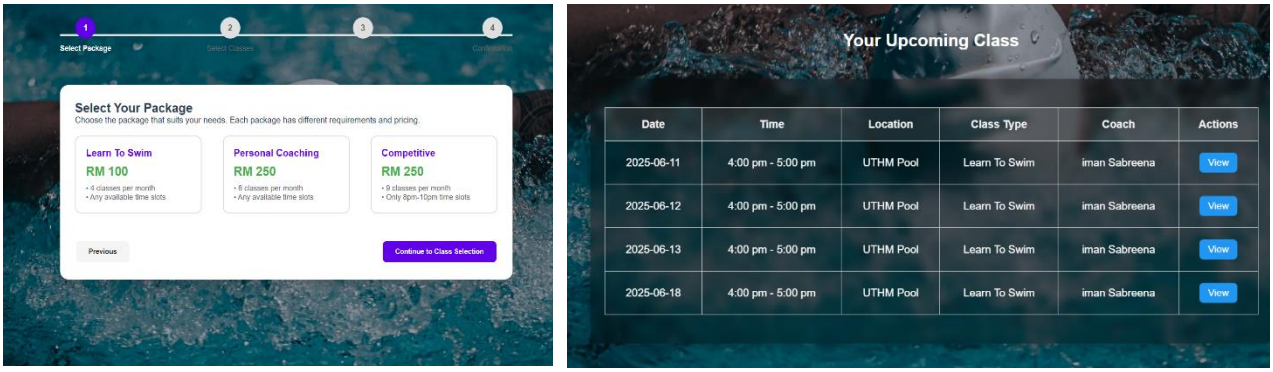
(c)

Fig. 13 *Coach Booking Swimming Class (a) calendar view; (b) set availability for swim class; (c) coach dashboard*

5.3 Implementation of Swimmer Module

The implementation of swimmer module is designed to book swimming classes and other functionalities such as viewing profile, dashboard, and online payments. To book a swim class, swimmers will have to choose type of class package since every package has different price as shown on Fig. 14(a). Once package has been selected, swimmer will have to follow requirements stated in the system. For example, learning to swim package has 4 classes, therefore, swimmer must choose 4 class sessions in any date, location, and time slot. Based on selected details, the system will display available coaches for the slot. If there is no coach the system will display "no coach available for the slot". Then swimmers will be redirect to payment method page and they can choose to pay now or pay later. Swimmers are also able to reschedule the class but must depend on coach availability and it must be done one day before class. Fig. 14(b) displays the swimmer's class booking after success full booking.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1051-1070

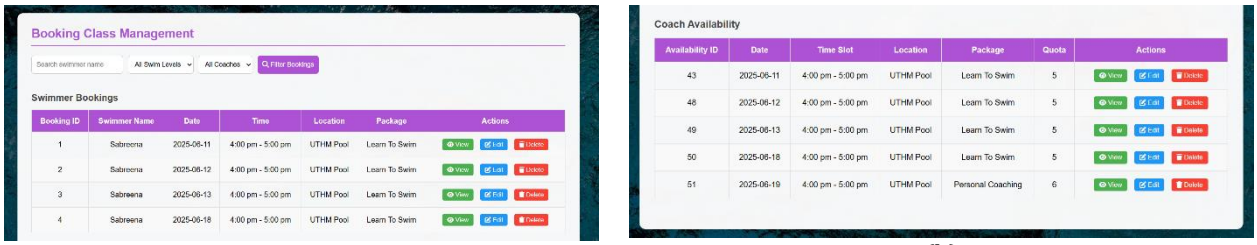


(a) (b)

Fig. 14 *Swimmer Booking Swimming Class (a) Upcoming class; (b) Booking Class Form*

5.4 Implementation of Admin Module

The implementation of the admin module is designed to manage users, manage swimming class bookings for both coach and swimmer, manage notifications, and manage payment record. Admin also has dashboard and profile which can be edited. For booking class management, the admin can view, edit, and delete any class booked by swimmers as shown in Fig. 15(a). Fig. 15(b) is for managing coach availability and shows quota of students available for the slot chosen by the coach. Admin also able to view, edit, and delete the coach availability if it is necessary. This interface will assist admin to track records and has much more organized management.

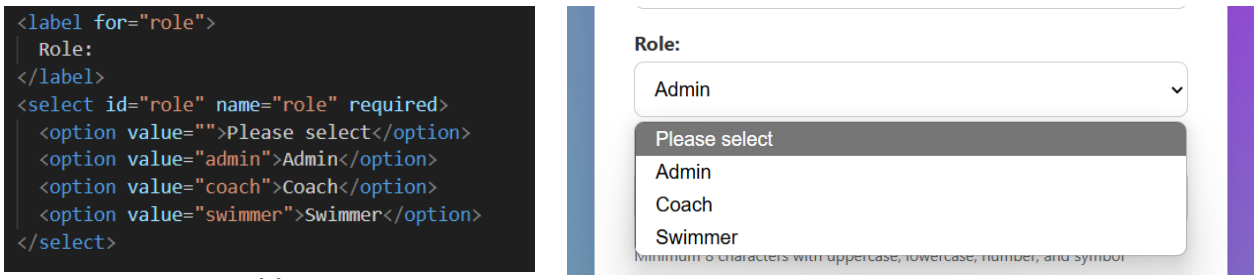


(a) (b)

Fig. 15 *Admin Booking Swimming Class Management interface*

5.5 Implementation of Role-Based Access Control (RBAC)

The setting up of Role-Based Access Control (RBAC) in the proposed system involves both frontend and backend scripting. It causes the system to restrict access based on user's role in the club which is swimmer, coach, and admin. Fig. 16(a) shows snippet of front-end scripting to enable user to choose their own role during signup. Fig. 16(b) is how the dropdown lists will be displayed to the client-side and assist to determine how the system will handle the user's account.



(a) (b)

Fig. 16 *Role-Based Access Control (a) in HTML ; (b) in Sign In Form*

Fig. 17(a) shows the back-end scripting executed in PHP in where after user succesfully verified their email, and based on their role selected, the system will insert the user's ID according to their role into specific role table such as admin, swimmer, and coach table in MySQL database. The prepared statement has been used to ensure database security by mitigating SQL injection attacks. Fig. 17(b) represents the PHP code that handles the user redirection after OTP verification during login.

```php
// Insert into role-specific table
if ($role == 'admin') {
    $roleStmt = $conn->prepare(query: "INSERT INTO admin (UserID) VALUES (?)");
} elseif ($role == 'coach') {
    $roleStmt = $conn->prepare(query: "INSERT INTO coach (UserID) VALUES (?)");
} else {
    $roleStmt = $conn->prepare(query: "INSERT INTO swimmer (UserID) VALUES (?)");
}
$roleStmt->bind_param(types: "i", var: &$userID);
$roleStmt->execute();
$roleStmt->close();
```

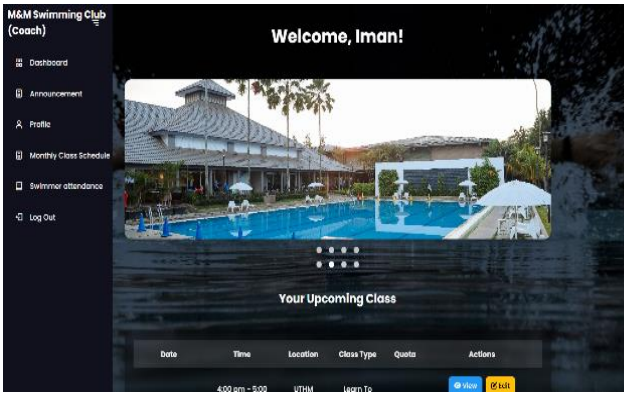
(a)

```php
switch ($role) {
    case 'admin':
        echo "<script>alert('Verification complete! Redirecting to Admin Dashboard...');
              window.location.replace('adminDashboard.php');</script>";
        break;
    case 'coach':
        echo "<script>alert('Verification complete! Redirecting to Coach Dashboard...');
              window.location.replace('coachDashboard.php');</script>";
        break;
    case 'swimmer':
        echo "<script>alert('Verification complete! Redirecting to Swimmer Dashboard...');
              window.location.replace('swimmerDashboard.php');</script>";
        break;
    default:
        echo "<script>alert('User role not recognized. Please contact support.');</script>";
}
```

(b)

Fig. 17 *Role-Based Access Control (a) in PHP after email verified; (b) after OTP verification during log in*

Depending on user's role, the system will redirect them to their dashboard as shown in Fig. 18(a) which is dashboard for coach and has access to announcement, view profile, updating monthly class schedule, and updating swimmer attendance. Fig. 18(b) is dashboard for swimmer that has access to swimmer's profile, announcement, making swim class booking and rescheduel, and also making payments. Fig. 18(c) dashboard for admin which has the highest privilege as they can read and write to any details of swimmer and coach, in where the manage users, class bookings, notifications, and also payment records.



(a)

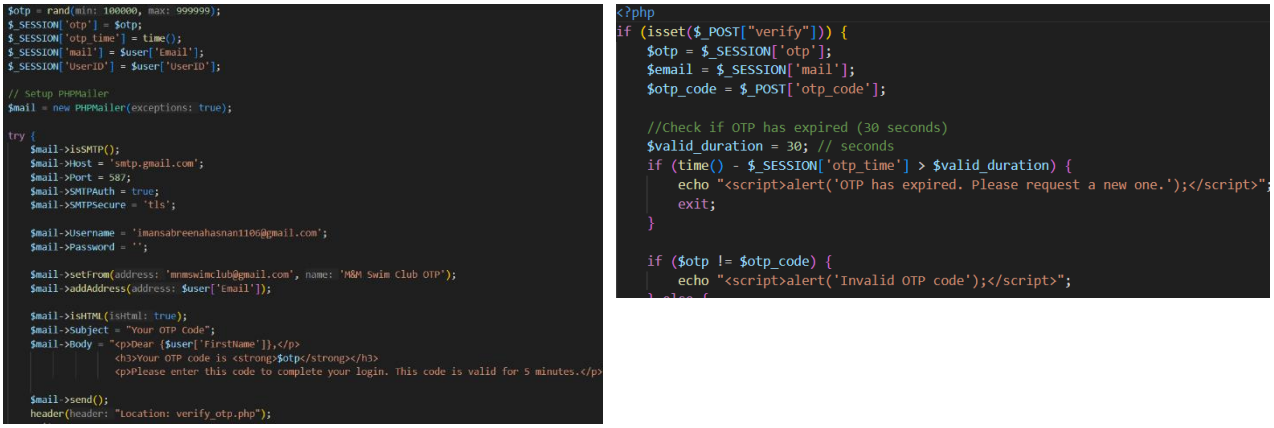
(b)

(c)

Fig.18 *Dashboard Interface of (a) Coach; (b) Swimmer; (c) Admin*

5.6 Implementation of Multi-Factor Authentication (MFA)

The proposed system also has implemented One-Time Password (OTP) as part of MFA to enhance the login security. Fig. 19(a) shows the server-side scripting generates a six-digits OTP using PHP rand( ) function and stores it along with other information such as current timestamp, user email, and user ID into session variables. The PHPMailer library has been used to configure. and send email that has OTP to users with email that has been verified during signup page. Fig. 19(b) shows on how the OTP verification has been done. Once user has entered the OTP number that they received via email, the system will compare the code with the one store in the session and if it matched, the user will be redirected to designated dashboard according to their role. Otherwise, error message will appear indicating the OTP entered is invalid due to expired code or wrong code. The OTP code will only be valid for 30 seconds and users will need to request new OTP.



```php
$otp = rand(min: 100000, max: 999999);
$_SESSION['otp'] = $otp;
$_SESSION['otp_time'] = time();
$_SESSION['mail'] = $user['Email'];
$_SESSION['UserID'] = $user['UserID'];

// Setup PHPMailer
$mail = new PHPMailer(exceptions: true);

try {
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->Port = 587;
    $mail->SMTPAuth = true;
    $mail->SMTPSecure = 'tls';

    $mail->Username = 'imansabreenahasnan1106@gmail.com';
    $mail->Password = '';

    $mail->setFrom(address: 'mnmswimclub@gmail.com', name: 'M&M Swim Club OTP');
    $mail->addAddress(address: $user['Email']);

    $mail->isHTML(isHtml: true);
    $mail->Subject = "Your OTP Code";
    $mail->Body = "<p>Dear {$user['FirstName']},</p>
                   <h3>Your OTP code is <strong>$otp</strong></h3>
                   <p>Please enter this code to complete your login. This code is valid for 5 minutes.</p>

    $mail->send();
    header(header: "Location: verify_otp.php");
```

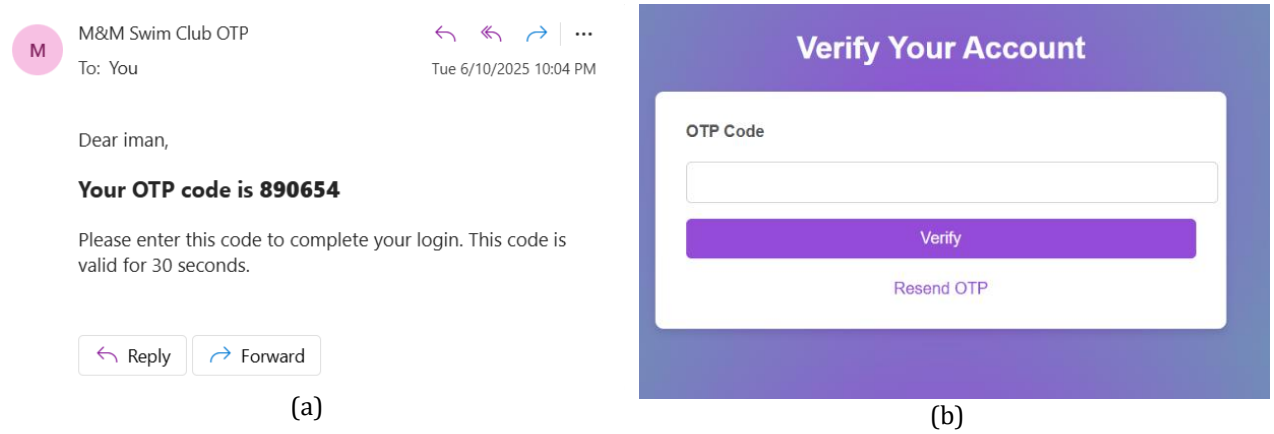
(a)

```php
<?php
if (isset($_POST["verify"])) {
    $otp = $_SESSION['otp'];
    $email = $_SESSION['mail'];
    $otp_code = $_POST['otp_code'];

    //Check if OTP has expired (30 seconds)
    $valid_duration = 30; // seconds
    if (time() - $_SESSION['otp_time'] > $valid_duration) {
        echo "<script>alert('OTP has expired. Please request a new one.');</script>";
        exit;
    }

    if ($otp != $otp_code) {
        echo "<script>alert('Invalid OTP code');</script>";
```

(b)

Fig.19 *OTP set up (a)Generating OTP in PHP; (b) Verifying OTP in PHP*

Fig. 20(a) shows how users will receive their OTP number in email and Fig.20(b) is where users will insert the code. This interface will appear once the user passes the admin's approval after signing up. Then, once username and password have been authenticated, then the system will send the OTP code to the user's email.



(a) (b)

Fig.20 *OTP for login (a) Receive in Email; (b) OTP interface*

6. Results and Discussion

This section will discuss the testing results of the proposed system. The tests conducted are functionality, security, and User Acceptance Testing (UAT). This testing phase assists with verifying the system functionality and security testing before being deployed.

6.1 Test Plan Result

Table 7 displays the result of the functionality test plan and Table 8 shows the results of the security test plan. Table 7 shows the result of the functionality test plan that has been carried out to check whether the system features have worked as intended. This includes scenarios such as user registration in sign up page, success or unsuccessful login attempts, booking and rescheduling swimming classes, processing payment records, and updating profiles. The result shows that all tests passed successfully indicate that the system has function as intended without errors.

Table 8 presents the security test plan which shows the effectiveness of the security measurements that have been embedded in the system. The test carried out includes enforcing user registration before login into system, validating credentials, implementation of RBAC, OTP verification, session destruction once logout, password masking and password policy during user registration, and email verification if user needs to recover forgotten password. The result shows that all security checks were passed where it verified that the system has met the intended security requirements.

Table 7 *Functionality Test Plan Result*

Check List	Expected Result	Actual Result
During register has correct input format	Input accepted and redirected to email verify	Pass
During register receive email verify	Register success and redirect to login module	Pass
During login has correct input format	Credentials validation passed and proceeded to OTP code request	Pass
During login receive OTP code via email	The OTP code will appear in email inbox	Pass
During login correct input format but wrong password	Login fail and a message will display "Incorrect username or password, please try again"	Pass
During login user has wrong input format and correct password	Login fail and a message will display "Incorrect username or password, please try again"	Pass
During login user has wrong input format and wrong password	Login fail and a message will display "Incorrect username or password, please try again"	Pass
During booking swim class, able to enter booking details	Booking successful and a message will display "Booking successful!"	Pass
Able to reschedule class according to policy	Reschedule successful and a message will display "Reschedule successful!"	Pass
Able to conduct online payment	Payment successful and a message will display "Payment Successful"	Pass
While updating profile, able to edit and save	Updating successful and a message will display "Updating successful!"	Pass

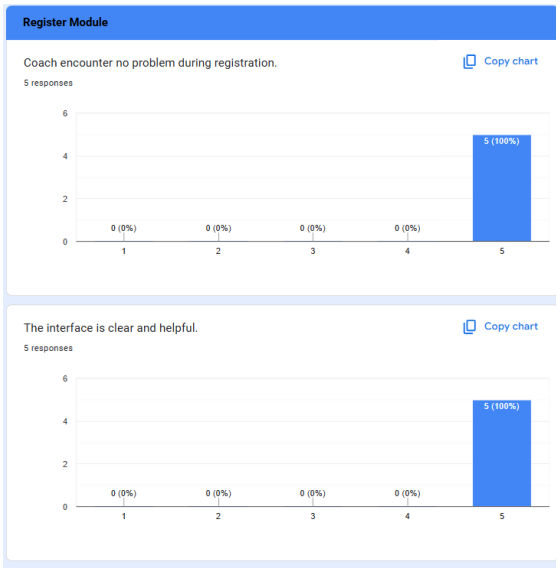
Table 8 *Security Test Plan Result*

Check List	Actual Result
Users should be able to register before accessing the system	Pass
Users should be able to log in according to a valid ID and password only	Pass
Each user has access to the system according to their role	Pass
Users should receive OTP code verification and get verified	Pass
Session is destroyed after logout	Pass
Password masking implemented during login and register	Pass
Password policy implemented during registration	Pass
Email verification is used for forgot passwords	Pass

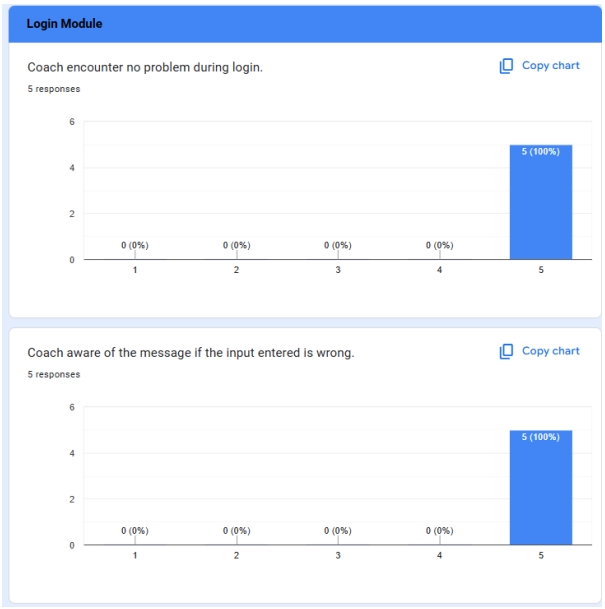
6.2 User Acceptance Form Results for Coach Module

The results of the user acceptance testing form for the coach are presented in both Fig. 21 and Fig. 22. The purpose of this testing is to gather user experience for the coach's interface while using the system. The test was implemented to the target user, which is coach of M&M swimming club. Fig. 21(a) indicates that 5 responders are satisfied with the registration process same goes with Fig. 21(b) where the responders are satisfied with the login process without any issues. Next, in Fig. 22(a) for Booking swim class module, the responders are satisfied with the class bookings and reschedule module and for the online payment in Fig. 22(b), all the coaches felt satisfied with the event announcement notification.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1051-1070

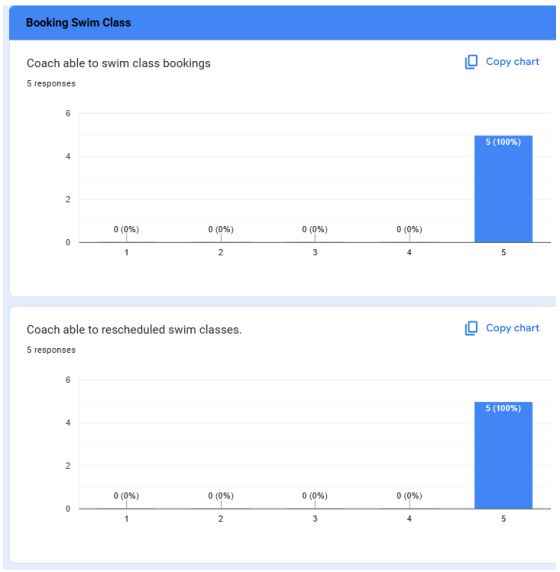


(a)

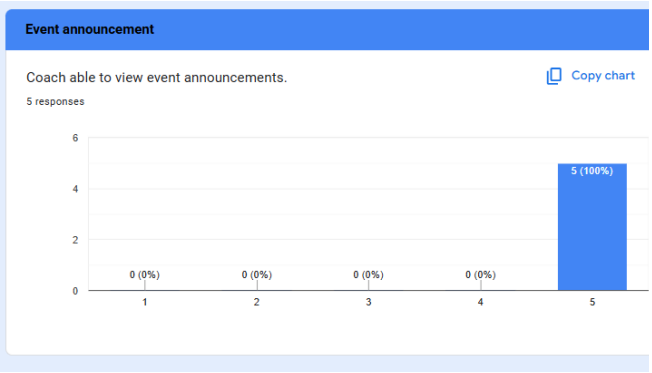


(b)

Fig. 21 *Result (a) Register Module; (b) login Module*



(a)

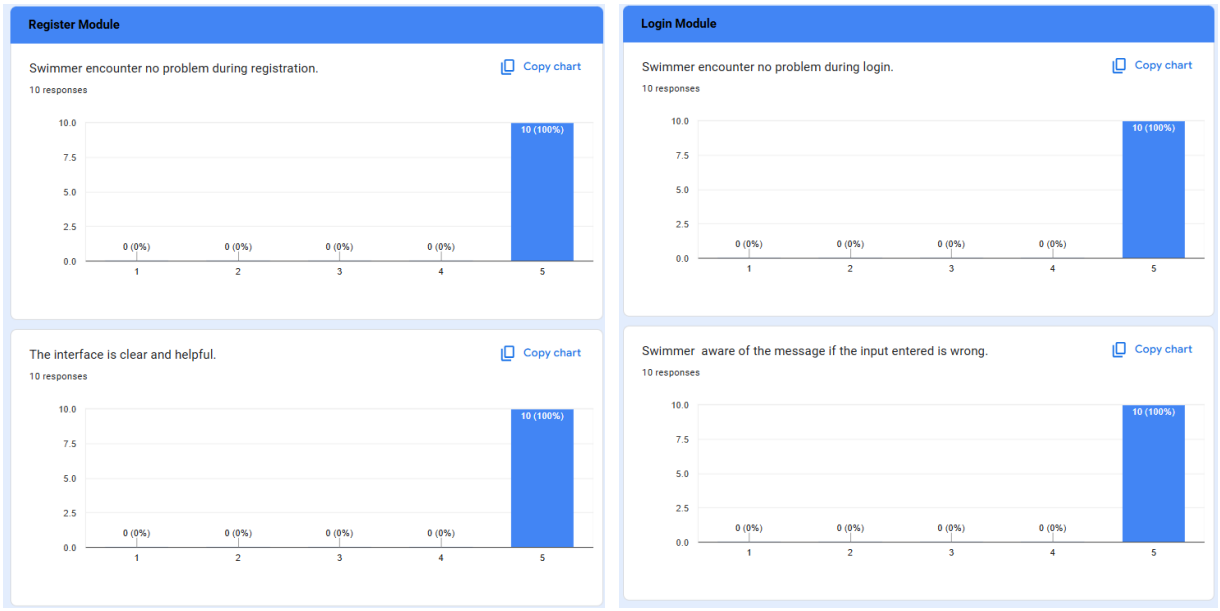


(b)

Fig. 22 *Result (a) Booking Swim Class Module; (b) Event Announcement*

6.3 User Acceptance Form Results for Swimmer Module

The results of the user acceptance testing form for the swimmer are presented in both Fig. 23 and Fig. 24. The purpose of this testing is to gather user experience for the swimmer's interface while using the system. The test was implemented to the target user, which is swimmers of M&M swimming club. Fig. 23(a) indicates that 10 responders are satisfied with the registration process, same goes with Fig. 23(b) where the responders are satisfied with the login process without any issues. Next in Fig. 24(a) for Booking swim class module, the responders are satisfied with the class bookings and reschedule module and for the online payment in Fig. 24 (b), 3 responders felt very satisfied, 6 responders were satisfied and 1 responder felt neutral. Lastly is for event announcement, 8 responders felt satisfied and 2 responders felt neutral



(a) **(b)**

Fig. 23 *Result (a) Register Module; (b) login Module*

(a) **(b)**

Fig. 24 *Result (a) Booking Swim Class Module; (b) Event Announcement*

6.4 User Acceptance Form Results for Admin Module

The result of the user acceptance test for the admin module has been shown in Table 9. The form was filled out by the admin of M&M Swimming Club and the results indicate that the admin was satisfied with the implementation of register module finding the interface is helpful and has encountered no problem during account registration. Same goes for the login module as the admin are aware of the error message if the inputs are entered wrong and have encountered no problem during login. Next, the admin is also satisfied with the profile presentation and has no problem updating their profile. For user management, the admin able to manage user information by viewing, editing, and deleting it. Admin is also able to be approved authorized users and navigate through the interface and buttons well. For swimming class booking management, the admin able to manage both coaches and swimmer by performing viewing, editing, and deleting. But the interface and the buttons of the swimming class booking management could be enhanced from aspect UI and UX design including table color and result displayed. For both payment record and notification management has overall scored satisfied by admin as it is functioning well by making the admin able to manage both payment record and notification. But for the notification display, they would prefer if it got notified via email as they can receive any notification much more directly rather than log in first.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1051-1070

Table 9 *User Acceptance Form Results for Admin*

No	Question	Result
		Dissatisfied 1 – 5 Satisfied
Register Module		
1.	Admin encounters no problem during registration	*5*
2.	The interface is clear and helpful	*5*
Login Module		
3.	Admin encounter no problem during login	5
4.	Admin aware of the message if the input entered is wrong	5
Admin Profile		
5.	Admin able to view their profile clearly	4
6.	Admin can update their profile	5
User Management		
7.	Admin able to manage user information by viewing editing and delete	5
8.	Admin able to approve authorised users	5
9.	Admin able to navigate to interface and buttons well	4
Swimming Class Booking Management		
10.	Admin able to manage swimming class booking for both coaches and swimmers	4
11.	Admin able to view, edit and delete booking for both coaches and swimmers	4
Payment record Management		
12.	Admin able to manage payment record	5
13.	Admin able to view and edit payment record	5
Notification Management		
14.	Admin able to manage notification	5
15.	Admin able to create, and display notification	3

7. Conclusion

In summary, the Swimming Class Booking Management System with security features of RBAC and MFA has been successfully developed with complete functionality. The objectives, scopes, system requirements, and user requirements have been successfully accomplished in the system developed. The Swimming Class Booking Management System has several advantages including the system that can separate privileges within three roles that are admin, coach, and swimmer. The system is also able to verify user's email and account via verification links and OTP sent in email. It also protected sensitive information such as password using hashing and salting and safely stored in database using prepared statement. However, there are also disadvantages to the system where since RBAC only assigns permissions using roles, if another user requires a slightly different privilege or permission under the same role, a new role must be created. If the same happens many times, too many roles will be created that will lead to difficulty handling and sustaining the system. Next, since the OTP code was sent via email, it may cause delays due to server issues, spam filters, and even poor internet connection. Based on the system's advantages and disadvantages, there are a few improvements that can be made to the system. Firstly, consider implementing RBAC along with permission sets instead of assigning privileges based on roles, we could define permissions in sets that can be assigned to any roles. This reduces the number of role creations and simplify the system. Next is to implement OTP using SMS-based or authenticator apps in which will improve OTP delivery speed even during offline.

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

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *N. I. S. Hasnan, N. Z. Harun;* ***data collection:*** *N. I. S. Hasnan, N. Z. Harun;* ***analysis and interpretation of results:*** *N. I. S. Hasnan, N. Z. Harun;* ***draft manuscript preparation:*** *N. I. S. Hasnan, N. Z. Harun. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Washington University, "Confidentiality, Integrity, and Availability: The CIA Triad | Office of Information Security | Washington University in St. Louis," Office of Information Security, Oct. 7, 2024. [Online]. Available: https://informationsecurity.wustl.edu/items/confidentiality-integrity-and-availability-the-cia-triad/

[2] B. Seijal Reino, *Web application for booking management and the organization of events in a sports town*, 2023.

[3] J. Zhao and J. Sun, "Research on access control model based on RBAC model in microservice environment," in Journal of Physics: Conference Series, vol. 1437, no. 1, p. 012031, 2020, IOP Publishing.

[4] M. Maynes, "IT executives prioritize Multi-Factor Authentication in 2020," Microsoft, Mar. 5, 2020. [Online]. Available: https://www.microsoft.com/en-us/security/blog/2020/03/05/it-executives-prioritize-multi-factor-authentication-2020/

[5] T. Suleski, M. Ahmed, W. Yang, and E. Wang, "A review of multi-factor authentication in the Internet of Healthcare Things," Digital Health, vol. 9, p. 20552076231177144, 2023.

[6] RNJ Swimming Academy, "PORTFOLIO," RNJ Swimming Academy, Nov. 14, 2024. [Online]. Available: https://rnjswimmingacademy.com/portfolio/

[7] D. J. McCubbrey, P. Bloom, and B. Younge, "USA Swimming: the data integration project," Communications of the Association for Information Systems, vol. 16, no. 1, p. 13, 2005.

[8] A. Oladele, "Top 15 Software Development Methodologies," Velvetech, Dec. 6, 2024. [Online]. Available: https://www.velvetech.com/blog/software-development-methodologies/

[9] G. Waja, J. Shah, and P. Nanavati, "Agile software development," International Journal of Engineering Applied Sciences and Technology, vol. 5, no. 12, pp. 73–78, 2021.

[10] OWASP, "OWASP Secure Coding Practices - Quick Reference Guide | Secure Coding Practices," OWASP Foundation. [Online]. Available: https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist