

# Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS)

Wong Lik<sup>1</sup>, Rabatul Aduni Sulaiman<sup>1\*</sup>

<sup>1</sup> Faculty of Computer Science and Information Technology,

Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

Corresponding Author: [rabatul@uthm.edu.my](mailto:rabatul@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.067>

## Article Info

Received: 16 June 2025

Accepted: 20 November 2025

Available online: 30 November 2025

## Keywords

SJMS, SJAM SMKJPP, Membership Management System, Prototyping

## Abstract

The Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS) is a web-based platform tailored to address the administrative needs of St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJAM SMKJPP). The current manual system for tracking member details, certifications, and activities is inefficient, prone to errors, and lacks centralised access. The project aims to design, develop, and test the SJMS, leveraging the Prototyping Software Development Lifecycle (SDLC) model to ensure iterative feedback from stakeholders. The system incorporates modules such as login, register, profile management, activity management, badge management, reporting, and learning materials, and it is designed for different user roles (super admin, admins, and members). With deployment targeted for SJAM SMKJPP in Ipoh, Perak, SJMS enables real-time data access and streamlined management tasks. As a result, the system successfully enhances task automation, improves data accuracy, and simplifies member tracking and reporting. Its implementation is significant in transforming manual processes into a centralized digital system, ultimately supporting better decision-making and promoting a more efficient, transparent, and engaging administrative environment.

## 1. Introduction

St. John Ambulance of Malaysia (SJAM), a voluntary, non-profit organisation established in 1908, has been dedicated to humanitarian and charitable services [1]. Its SMK Jalan Pasir Puteh division (SJAM SMKJPP), founded in 1983 [2], currently comprises 180 active members. The division's membership management relies on manual processes, requiring members to fill out forms with personal information, which are subsequently compiled into spreadsheets by committee members. These spreadsheets are used to track personal details, certifications, and participation in events. However, this approach has significant limitations, including inefficiencies, errors, and delays, particularly when managing large volumes of data. These challenges hinder the division's ability to maintain accurate records and streamline operations effectively. To address these issues, the Membership Management System (SJMS) is proposed as a centralised, web-based platform. The SJMS aims to automate data collection, track member participation, and provide real-time access to information for both administrators and members. It incorporates features such as user authentication, activity and badge management, and robust reporting tools, all tailored to improve administrative efficiency, reduce manual errors, and enhance member engagement. Additionally, the system will empower division executives with data-driven insights to better plan events and manage operations, marking a significant advancement over the existing manual processes.

There are three objectives for the project:

This is an open access article under the CC BY-NC-SA 4.0 license.



1. To design the Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS) using an object-oriented approach.
2. To develop the Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS) using a web-based approach.
3. To test the developed Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS) using system testing and user acceptance testing.

The target users for this project are the super admin, admins, and members of the St. John Ambulance of Malaysia SMK Jalan Pasir Puteh division. The system will feature role-based access control, allowing super admin to oversee overall system management, admins to handle activities and member data, and members to access their profiles and learning materials. The system will be developed using web technologies and will incorporate various modules to enhance administrative efficiency and member engagement. These modules include login, registration, profile management, activity management, badge management, report management, and learning materials, listed as shown in Table 1.

**Table 1** *System Modules*

Module	Description
Login	Allows secure user authentication for super admin, admins, and members, with self-service options for password resets.
Profile Management	Enables super admin to manage user accounts, including creating, updating, and deleting profiles and managing user permissions.
Registration	Allows super admin to register admins, who can then register members. An IC card reader is used for faster and more accurate data entry.
Activity Management	Enables admins to create, update, and track activities, assign members to specific tasks, and monitor participation.
Badge Management	Allows admins to manage cadet proficiency badge records, enabling members to view their achievements.
Report Management	Allows super admin to generate insightful reports on member data, activities, and other metrics to support decision-making and improve division operations.
Learning Materials	Allows admins to manage and update learning resources and quizzes related to first aid, which members can access to enhance their knowledge and skills.

The successful implementation of the Membership Management System (SJMS) is anticipated to significantly enhance administrative efficiency and member engagement for the St. John Ambulance of Malaysia, SMK Jalan Pasir Puteh division. This system will streamline processes such as registration, activity tracking, and badge management, while also reducing manual errors and improving data accuracy. Members will benefit from a clear view of their achievements and access to learning materials, while super admin and admins will gain robust reporting tools to facilitate data-driven decision-making.

Furthermore, the SJMS is a crucial initiative to enhance the division's operational effectiveness by addressing challenges such as manual errors, data inconsistencies, and inefficiencies. By centralising member data and event participation records, the system will enable better resource allocation and event planning. Its user-friendly interface will foster member engagement by providing an organised and transparent platform that reflects the division's commitment to service excellence.

This paper will continue with several sections. Section II will present the literature review, Section III will describe the methodology used for this project, Section IV will cover system analysis and design, Section V will cover implementation and testing, and Section VI will conclude the paper.

## 2. Literature Review

This section will address the study of the membership management system, and the comparison of similar systems.

### 2.1 Domain Background

St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJAM SMKJPP) is a division located in Ipoh, Perak, that currently manages its membership through manual processes. These processes involve members submitting personal information forms, which are reviewed and manually entered into spreadsheets by committee members. This system is used to track certifications, event participation, and other member-related data. However, as the organisation has grown to 180 active members, the manual approach has become increasingly inefficient and error-prone. The reliance on spreadsheets for data management leads to challenges in maintaining accuracy, consistency, and timely updates, underscoring the need for a more efficient and centralised solution.

## 2.2 Membership Management System

Digital membership management systems have revolutionised organisations to manage member operations by automating processes such as data management, event coordination, and financial transactions. These systems address inefficiencies, reduce errors, and improve scalability, which are common challenges in manual approaches [3]. Web-based systems, such as those developed using Node.js, were highlighted as effectively managing member data, finances, and activities, thereby streamlining administrative workloads [4]. Additionally, integrating tools like Quick Response codes (QR) can enhance attendance tracking and event management, boosting operational efficiency [5]. These systems allow admins to manage member data and enable members to update profiles, participate in events, and engage more actively, fostering involvement and improving efficiency compared to traditional methods.

## 2.3 Similar System Comparison

The three existing systems are thoroughly examined and analysed in this section. The features of three existing systems are compared with the proposed system. Table 2 compares the three existing systems and the proposed system.

**Table 2** Similar System Comparison

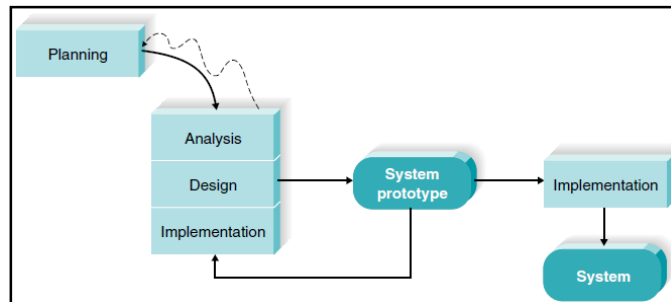
No.	Features	MACMA	Join It	Mighty Networks	SJMS
1	Login	Yes	Yes	Yes	Yes
2	Registration	Yes	Yes	Yes	Yes
3	Manage Member Detail	Yes	Yes	No	Yes
4	Activity Management	No	Yes	No	Yes
5	Badge Management	No	No	No	Yes
6	Generate Report	Yes	Yes	No	Yes
7	Learning Materials	No	No	No	Yes
8	User	Admin, Member	Admin, Member	Admin, Member	Super admin, Admin and Member

The comparison between the proposed SJMS system and three existing systems—MACMA Membership Management System with QR Code, Join It, and Mighty Networks—revealed that SJMS offers the most comprehensive set of features. All systems support basic login and registration functionalities. However, SJMS stands out by integrating additional modules such as badge management, activity tracking, report generation, and access to learning materials, which are not fully available in the other systems. While Join It and MACMA offer partial administrative capabilities, they lack badge tracking and learning resources. Mighty Networks focuses more on community engagement rather than administrative functions and does not support member detail management or reporting. Overall, SJMS improves upon the limitations of the existing systems by offering a centralized, role-based solution tailored specifically to the needs of St. John Ambulance SMK Jalan Pasir Puteh.

## 3. Methodology

The Prototyping Model involves the simultaneous execution of analysis, design, and implementation to develop simplified system versions for user evaluation and iterative refinement [6]. This model was chosen to develop the

SJMS due to its ability to incorporate continuous stakeholder feedback throughout the process. With distinct user categories—super admin, admins, and members, the model enables the creation of early prototypes for features such as profile management, login, and registration. Stakeholders can review and test these prototypes, providing feedback to refine functionalities and enhance usability. This iterative approach ensures that the system effectively addresses the challenges of manual management processes while aligning with user requirements and improving overall system design. Figure 1 shows the prototyping model used in the project.



**Fig. 1 Prototyping Model [6]**

The planning phase defines the project's objectives and establishes its foundation. Initial meetings with stakeholders are held to present the project proposal and outline its scope and goals. Requirements are gathered through interviews and observations, providing insights into stakeholder needs and priorities. The current membership management process is documented to identify areas for improvement. Tools such as Microsoft Excel are used to create a project schedule and Gantt chart, ensuring tasks are well-organized and timelines are effectively managed. This phase is crucial for aligning stakeholder expectations and setting a clear direction for the project's development.

In the analysis phase, stakeholder requirements are examined to define the system's functional modules. Interviews and observations are conducted to gather detailed insights into user needs. The identified modules include login, registration, profile management, activity management, badge management, report management and learning materials. Effective communication with stakeholders is prioritized to ensure clarity and alignment. Visual models of the system, created using tools like Figma, provide stakeholders with a clear preview of the system's functionality and design, ensuring that the requirements are accurately captured for subsequent phases.

The design phase focuses on creating a comprehensive blueprint for the system. Unified Modeling Language (UML) diagrams, including use case, activity, and sequence diagrams, are developed using tools like Draw.io to outline the system's architecture and interactions. The database structure and user interface are also designed during this phase, ensuring alignment with stakeholder expectations. Feedback is collected to validate the design, and detailed software design documentation captures key decisions. This phase establishes a structured foundation, enabling a systematic approach to the system's development.

The implementation phase involves building the system based on the design specifications. Specific modules are developed, and the database is established using tools such as XAMPP, PHP, HTML, CSS, JavaScript, and MySQL. Initial testing and debugging are conducted throughout the process to identify and resolve issues early, ensuring the system's reliability and performance. Version control is managed using GitHub to track changes and maintain consistency. This iterative approach ensures that each component is systematically developed and refined, resulting in a stable and functional final product.

Table 3 shows the system development workflow, which discusses each phase's tasks, outputs, and tools used during the process. The prototyping model has four phases, including planning, analysis, design, and implementation.

**Table 3 Task and output of each phase**

Phase	Task	Output	Tool
Planning	• Proposed idea and title for the final year project	• Project Proposal	• Microsoft Word
	• Initial meeting with the stakeholders	• Gantt Chart	• Microsoft Excel
	• Determine problem statements, objectives, and project scopes		• WhatsApp
	• Document the current system		

**Table 3: (Cont).**

	<ul style="list-style-type: none"> <li>• Study the related topic of the proposed system</li> </ul>		
Analysis	<ul style="list-style-type: none"> <li>• Interview with stakeholders</li> <li>• Observation</li> <li>• Requirement gathering</li> <li>• Identify the hardware and software requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Functional requirement</li> <li>• Non-functional requirement</li> <li>• Hardware and software requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• WhatsApp</li> </ul>
Design	<ul style="list-style-type: none"> <li>• Design UML diagrams</li> <li>• Design database structure</li> <li>• Design user interface</li> </ul>	<ul style="list-style-type: none"> <li>• Use case diagram</li> <li>• Activity diagram</li> <li>• Sequence diagram</li> <li>• Domain Class diagram</li> <li>• Interface design</li> <li>• Schema table</li> </ul>	<ul style="list-style-type: none"> <li>• Draw.io</li> <li>• Figma</li> </ul>
Implementation	<ul style="list-style-type: none"> <li>• Code implementation</li> <li>• Develop the system database</li> <li>• Testing</li> <li>• System evaluation by stakeholders</li> </ul>	<ul style="list-style-type: none"> <li>• System database</li> <li>• Final complete system</li> <li>• Test case</li> <li>• System evaluation result</li> </ul>	<ul style="list-style-type: none"> <li>• VS Code</li> <li>• XAMPP</li> <li>• PHP</li> <li>• HTML</li> <li>• CSS</li> <li>• JavaScript</li> <li>• MySQL</li> <li>• Google Chrome</li> </ul>

## 4. Analysis and Design

This section describes the system requirements analysis and design.

### 4.1 Requirement Traceability Matrix

This section provides an overview of the Requirement Traceability Matrix for the proposed system. It documents the specific requirements outlined in Table 4.

**Table 4 Requirement Traceability Matrix**

Requirement ID	Description
<b>REQ_100</b>	<b>Login</b>
FR01-01	The system shall allow users to log in.
FR01-02	The system shall allow users to input login details.
FR01-03	The system shall allow users to submit the login details.
FR01-04	The system shall verify the user login details.
FR01-05	The system shall allow users to input the login ID associated with their account.
FR01-06	The system shall allow users to reset their account password.
FR01-07	The system shall allow users to submit the reset password details.

**Table 4:** (Cont).

FR01-08	The system shall verify the reset password email address.
CR01-01	The system must accept only valid user credentials to log in.
CR01-02	The system must accept only valid email address matching the database to reset the account password.
QR01-01	The system should have an easy layout and navigation structure.
<b>REQ_200</b>	<b>Registration</b>
FR01-01	The system shall allow super admin to register admins.
FR01-02	The system shall allow the super admin to input the admin registration details.
FR01-03	The system shall allow the super admin to submit the admin registration details.
FR01-04	The system shall allow the super admin and admin to register members.
FR01-05	The system shall allow the super admin and admin to input the member registration details.
FR01-06	The system shall allow the super admin and admin to submit the member registration details.
FR01-07	The system shall verify the admin and member registration details.
FR01-08	The system shall store the admin and member details in the database.
CR01-01	The system must accept only unregistered member details.
QR01-01	The system should have an easy layout and navigation structure.
<b>REQ_300</b>	<b>Profile Management</b>
FR01-01	The system shall allow users to update their profile details.
FR01-02	The system shall allow users to change their account password.
FR01-03	The system shall allow users to submit the updated profile details.
FR01-04	The system shall allow users to submit a new password.
FR01-05	The system shall verify the updated profile details.
FR01-06	The system shall verify the new password input.
FR01-07	The system shall allow the super admin to delete admin accounts.
FR01-08	The system shall allow the super admin and admin to delete member accounts.
FR01-09	The system shall update the profile details in the database.
FR01-10	The system shall update the user password in the database.
CR01-01	The system must only accept valid input details for profile details.
CR01-02	The system must only accept valid passwords to change account passwords.
QR01-01	The system should have an easy layout and navigation structure.
<b>REQ_400</b>	<b>Activity Management</b>
FR01-01	The system shall allow the super admin and admin to create, edit, and delete the activity records.
FR01-02	The system shall allow the super admin and admin to input the activity record.
FR01-03	The system shall verify the activity record.
FR01-04	The system shall allow the super admin and admin to submit the activity record.
FR01-05	The system shall be able to store the activity record in the database.

**Table 4:** (Cont).

FR01-06	The system shall be able to update the activity record in the database.
FR01-07	The system shall be able to remove the activity record in the database.
CR01-01	The system must only accept valid input details for activity records.
QR01-01	The system should have an easy layout and navigation structure.
<b>REQ_500</b>	<b>Badge Management</b>
FR01-01	The system shall allow the super admin and admin to create, edit, and delete the badge records.
FR01-02	The system shall allow the super admin and admin to input the badge record.
FR01-03	The system shall verify the badge record.
FR01-04	The system shall allow the super admin and admin to submit the badge record.
FR01-05	The system shall be able to store the badge record in the database.
FR01-06	The system shall be able to update the badge record in the database.
FR01-07	The system shall be able to remove the badge record in the database.
CR01-01	The system must only accept valid input details for badge records.
QR01-01	The system should have an easy layout and navigation structure.
<b>REQ_600</b>	<b>Learning Materials</b>
FR01-01	The system shall allow the super admin and admin to create, edit, and delete the learning materials.
FR01-02	The system shall allow the super admin and admin to input the learning materials.
FR01-03	The system shall verify the learning materials.
FR01-04	The system shall allow the super admin and admin to submit the learning materials.
FR01-05	The system shall be able to store the learning materials in the system folder.
FR01-06	The system shall be able to update the learning materials in the system folder.
FR01-07	The system shall be able to remove the learning materials in the system folder.
CR01-01	The system must only accept valid input details for learning materials.
QR01-01	The system should have an easy layout and navigation structure.
<b>REQ_700</b>	<b>Report Management</b>
FR01-01	The system shall create an Excel report file.
FR01-02	The system shall allow the super admin to convert the selected report to a PDF file.
FR01-03	The system shall allow the super admin to download the PDF file.
CR01-01	The system must only accept valid input details for learning materials.
QR01-01	The system should have an easy layout and navigation structure.

## 4.2 Use Case Diagram

A use case diagram visually represents the interaction between users and the information system [7]. The Membership Management System for St. John Ambulance of Malaysia at SMK Jalan Pasir Puteh (SJMS) involves three main user roles: super admin, admins, and members. The system consists of seven modules that are assigned according to the user roles. These modules are depicted in the use case diagram shown in Figure 2.

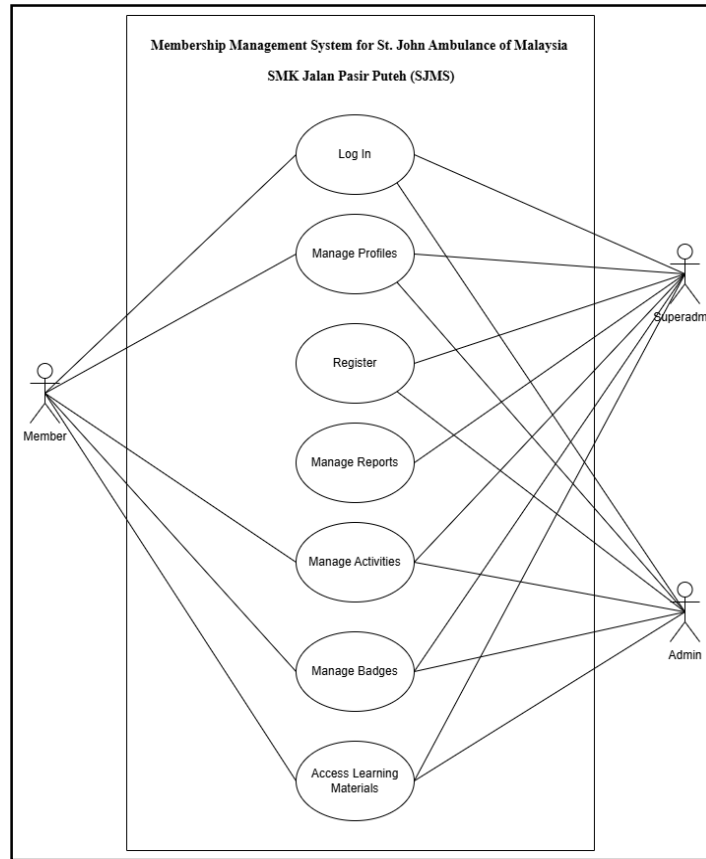


Fig. 2 Use Case Diagram

### 4.3 Class Diagram

The class diagram represents the conceptual model in database modelling, as illustrated in Figure 3. It includes users, profile, activity, activity\_record, badge, badge\_record, learning\_material and learning\_materials\_record database tables.

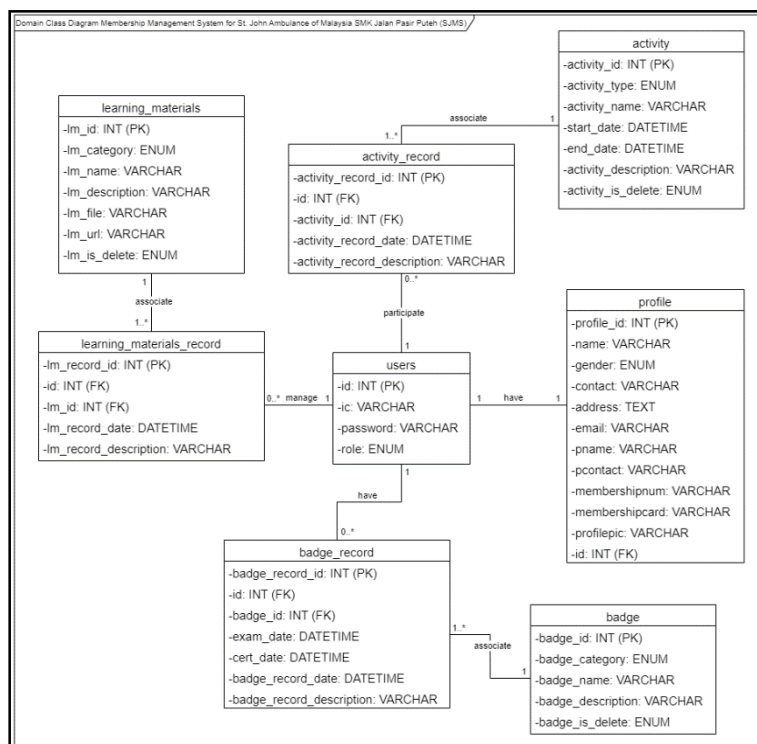


Fig. 3 Class Diagram

## 5. Implementation and Testing

This section will discuss the implementation and testing of SJMS.

### 5.1 Implementation

In the login module, super admin, admins, and members can login to the system using their login credentials. Figure 4.1(a) shows the Login interface. Super admin, admins, and members need to input their Login ID and password to be able to login to the system. Figure 4.1(b) shows the code segments for Login. The system will validate the login input provided by the super admin, admins, and members. If all of the input is valid, the system will redirect the super admin, admins, and members to the respective Dashboard Page.

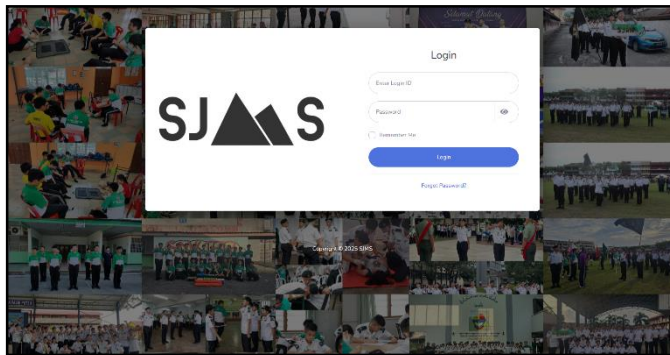


Fig. 4.1(a): Login Interface



Fig. 4.1(b): Login Code Segment

In registration module, super admin and admins can register an account for members to access the system. To register an account, super admin and admins need to go to the Register member Page and fill out the required details for the registration process. Figure 4.2(a) shows the interface for Member Registration. Figure 4.2(b) shows the code segments for Member Registrations. The system will validate the input provided by the super admin and admin. If all the inputs are valid, the system will store the member registration details in the database.

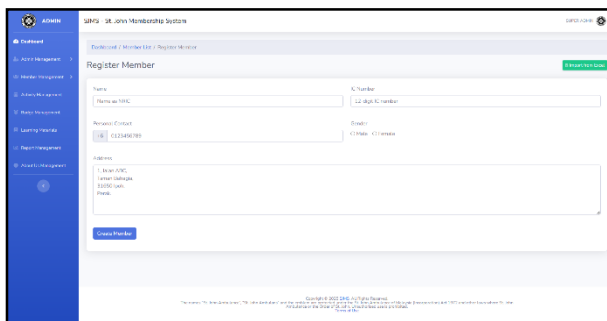
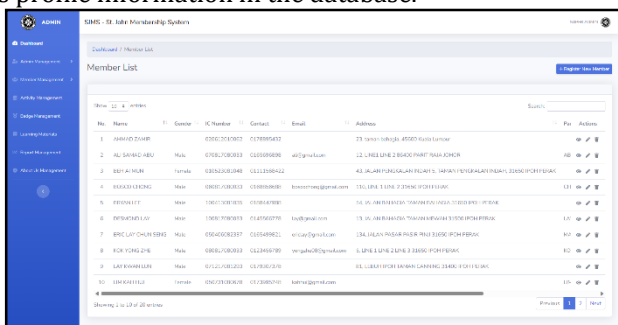


Fig. 4.2(a): Registration Interface



Fig. 4.2(b): Registration Code Segment

In profile management module, all users—including super admin, admins, and members—can manage their personal profiles within the system. Super admin also have the ability to update or delete member and admin accounts, while admins can only update or delete member accounts. To manage a profile, users navigate to the Profile Page, where they can edit personal information and change their account password. Figure 4.3(a) shows the interface for Profile Management. Figure 4.3(b), 4.3(c) and 4.3(d) presents the code segments used to handle profile updates. The system performs validation on the submitted data and, upon successful validation, updates the user's profile information in the database.



**Fig. 4.3(a): Profile Mangement Interface**

**Fig. 4.3(b): Profile Mangement Code Segment**

```
// Update profile picture in database
$stmt = $conn->prepare(query: "UPDATE profile SET profilepic = ? WHERE id = ?");
$stmt->bind_param('si', vars: [$upload_path, vars: $user_id]);
if (!$stmt->execute()) {
    throw new Exception(message: "Error updating profile picture");
}

// Execute the first membership card upload statement
if (!$stmt->execute()) {
    throw new Exception(message: "Error uploading membership card photo");
}

// Handle membership card photo upload
if (isset($_FILES['membership_card']) && $_FILES['membership_card']['error'] == 0) {
    $allowed = ['jpg', 'jpeg', 'png'];
    $filename = $_FILES['membership_card']['name'];
    $filetype = pathinfo(path: $filename, flags: PATHINFO_EXTENSION);

    if (in_array(strtolower($filetype), haystack: $allowed)) {
        throw new Exception(message: "Only JPG, JPEG, and PNG files are allowed for membership card photo");
    }

    // Create directory
    $upload_dir = "../uploads/file/membership_card_photo";
    if (!file_exists($filename: $upload_dir)) {
        if (!mkdir(directory: $upload_dir, permissions: 0755, recursive: true)) {
            throw new Exception(message: "Failed to create upload directory");
        }
    }

    $newname = "membership_card_" . $user_id . "_" . time() . "." . $filetype;
    $upload_path = $upload_dir . "/" . $newname;

    // Check if file was actually uploaded
    if (!is_uploaded_file(filename: $_FILES['membership_card']['tmp_name'])) {
        throw new Exception(message: "File upload failed");
    }

    // Add error checking for move uploaded file
    if (!move_uploaded_file(from: $_FILES['membership_card']['tmp_name'], to: $upload_path)) {
        throw new Exception(message: "Failed to move uploaded file");
    }
}
```

```
// Update membership card photo in database
$stmt = $conn->prepare(query: "UPDATE profile SET membershipcard = ? WHERE id = ?");
$stmt->bind_param('si', vars: [$upload_path, vars: $user_id]);
if (!$stmt->execute()) {
    throw new Exception(message: "Error uploading membership card photo");
}

$stmt->commit();

// Redirect with success message
header(header: "location: member-edit.php?id=" . $user_id . "&success=1");
exit();

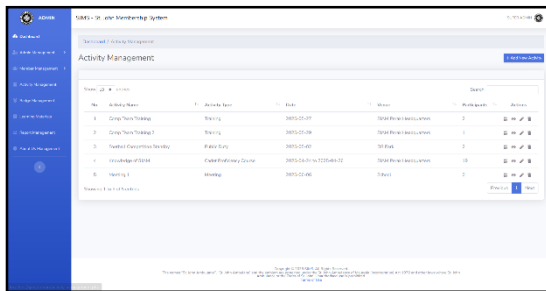
} catch (Exception $e) {
    $conn->rollback();
    header(header: "location: member-edit.php?id=" . $user_id . "&error=0");
    exit();
}

// If not POST request, redirect to member-edit page
header(header: "location: member-edit.php");
exit();
```

**Fig. 4.3(d): Profile Mangement Code Segment**

**Fig. 4.3(c): Profile Mangement Code Segment**

The Activity Management module allows super admin and admins to create, update, and delete records of activities or events. These activities are crucial for tracking member participation. To manage activities, users go to the Activity Management Page and input the necessary event details. Figure 4.4(a) illustrates the interface for Activity Management, while Figure 4.4(b) displays the code segments responsible for managing activity records. The system validates the data before storing or updating it in the database to ensure accuracy and completeness.



**Fig. 4.4(a): Activity Management Interface**

```
try {
    // Insert data for database
    $datetimeformatted = $datetime->format(format: 'Y-m-d H:i:s');
    $datetimeformatted = $datetime->format(format: 'Y-m-d H:i:s');

    // Convert participants array to comma-separated string
    $participantsString = implode(separator: ',', array: array_map(callback: 'intval', array: $participants));

    // Insert into activity table
    $query = "INSERT INTO activity (
        activity_category,
        activity_name,
        activity_description,
        start_date,
        end_date,
        activity Venue,
        activity participants,
        activity record date
    ) VALUES ( ?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = mysqli_prepare(mysqli: $conn, query: $query);
    if (!$stmt) {
        error_log(message: "Insert prepare failed: " . mysqli_error(mysqli: $conn));
        throw new Exception(message: "Database prepare failed");
    }

    mysqli_stmt_bind_param(statement: $stmt, types: 'sssssss',
        vars: [$activityCategory,
            $description,
            $datetimeformatted,
            $datetimeformatted,
            $venue,
            $participantsString
        ]);

    if (!mysqli_stmt_execute(statement: $stmt)) {
        error_log(message: "Insert execute failed: " . mysqli_stmt_error(statement: $stmt));
        throw new Exception(message: "Database execute failed");
    }

    // Redirect to success page
    header(header: "location: activity-add.php?success=1");
    exit();
}
```

**Fig. 4.4(b): Activity Mangement Code Segment**

Badge management module enables super admin and admins to manage cadet proficiency badge records for each member. They can add, edit, or delete badge achievements, and members can view the badges they have earned. To perform badge management, users access the Badge Management Page and input the relevant information. Figure 4.5(a) shows the interface for Badge Management, and Figure 4.5(b) contains the corresponding code segments. Input validation is conducted before the badge data is saved or updated in the system database.

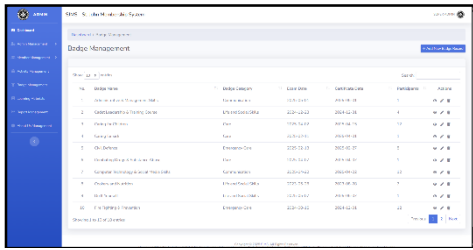


Fig. 4.5(a): Badge Mangement Interface



Fig. 4.5(b): Badge Mangement Code Segment

The Learning Materials module is designed to allow super admin and admins to upload and manage educational content, such as documents or links related to first aid. Members can browse and access these resources to enhance their knowledge. To manage learning content, users navigate to the Learning Materials Page and provide the necessary file or link details. Figure 4.6(a) shows the interface for the Learning Materials module, and Figure 4.6(b) presents the code that handles file and link management. The system verifies the input before storing the learning materials in the designated system folder or database.

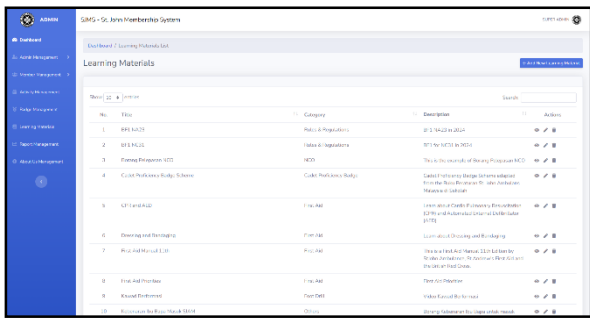


Fig. 4.6(a): Learning Materials Interface

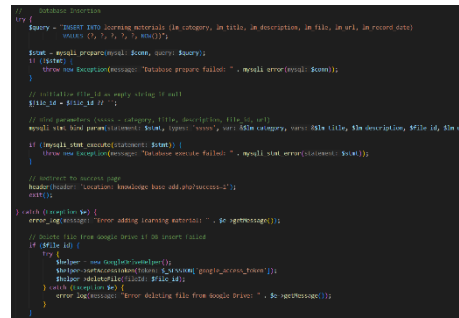


Fig. 4.6(b): Learning Materials Code Segment

The Report Management module enables the super admin to generate reports on member data, activity participation, and badge achievements. These reports help division executives make informed decisions and evaluate member engagement. To generate a report, the super admin accesses the Report Management Page, selects the desired report type, and exports it in Excel or PDF format. Figure 4.7(a) illustrates the interface for Report Management, and Figure 4.7(b) shows the code used to create and export reports. The system compiles the data based on current database records and generates the report output accordingly.

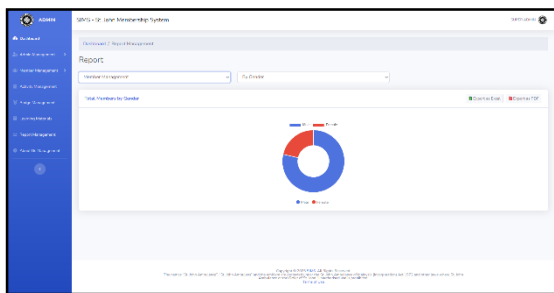


Fig. 4.7(a): Report Mangement Interface



Fig. 4.7(b): Report Mangement Code Segment

### 5.2 Testing

The testing phase focuses on assessing the functionality of the system. This section involves two different testing methods, which are system testing and user acceptance testing.

System testing is conducted to ensure that the developed system meets both functional and non-functional requirements. The completion of system testing is determined by the proper functionality of all modules, aligned with the specified functional and non-functional requirements. The list of test cases and results is presented in Table 5.

Table 5 List of Test Cases

Test Case ID	Requirement ID	Description	Test Status
--------------	----------------	-------------	-------------

<b>TC_100</b>	<b>REQ_100</b>	<b>Login</b>	
TC_100_01	REQ_101	Verify user can access the login page	PASS
TC_100_02	REQ_102	Verify user can input login details	PASS
TC_100_03	REQ_103	Verify user can submit login form	PASS
TC_100_04	REQ_104	Verify system validates user credentials	PASS
TC_100_05	REQ_105	Verify user can input associated login ID	PASS
TC_100_06	REQ_106	Verify user can request password reset	PASS
TC_100_07	REQ_107	Verify system accepts submitted reset password	PASS
TC_100_08	REQ_108	Verify only valid credentials are accepted	PASS
TC_100_09	REQ_109	Verify only valid email addresses for password reset	PASS
TC_100_10	REQ_110	Verify login interface has user-friendly layout	PASS
<b>TC_200</b>	<b>REQ_200</b>	<b>Registration</b>	
TC_200_01	REQ_201	Verify super admin can register admins and members	PASS
TC_200_02	REQ_202	Verify system accepts only unregistered member details	PASS
TC_200_03	REQ_203	Check UI layout and navigation for registration	PASS
<b>TC_300</b>	<b>REQ_300</b>	<b>Profile Management</b>	
TC_300_01	REQ_301	Verify user can update profile details	PASS
TC_300_02	REQ_302	Verify user can change password	PASS
TC_300_03	REQ_303	Validate only valid profile inputs are accepted	PASS
TC_300_04	REQ_304	Validate only valid password is accepted for updates	PASS
TC_300_05	REQ_305	Check UI layout and structure for profile page	PASS
<b>TC_400</b>	<b>REQ_400</b>	<b>Activity Management</b>	
TC_400_01	REQ_401	Verify super admin/admin can manage activity records (CRUD)	PASS
TC_400_02	REQ_402	Validate only valid activity input is accepted	PASS
TC_400_03	REQ_403	Check UI layout and navigation for activity management	PASS
<b>TC_500</b>	<b>REQ_500</b>	<b>Badge Management</b>	
TC_500_01	REQ_501	Verify super admin/admin can manage badge records (CRUD)	PASS

Table 5 (Cont).

TC_500_02	REQ_502	Validate only valid badge input is accepted	PASS
TC_500_03	REQ_503	Check UI layout and navigation for badge management	PASS
<b>TC_600</b>	<b>REQ_600</b>	<b>Learning Materials</b>	
TC_600_01	REQ_601	Verify super admin/admin can manage learning materials (CRUD)	PASS
TC_600_02	REQ_602	Validate only valid learning material input is accepted	PASS
TC_600_03	REQ_603	Check UI layout and navigation for learning materials	PASS
<b>TC_700</b>	<b>REQ_700</b>	<b>Report Management</b>	
TC_700_01	REQ_701	Verify Excel report generation and PDF conversion/download	PASS
TC_700_02	REQ_702	Validate only valid inputs for report generation	PASS
TC_700_03	REQ_703	Check UI layout and navigation for report management	PASS

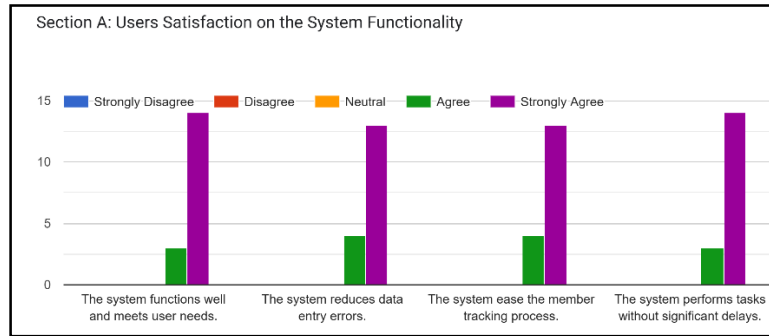
In this section, the summary of the results for the system testing will be shown and discussed. There are a total of seven modules with thirty test cases that have been carried out to test the system. Table 6 shows the overall result of test cases.

**Table 6 Overall Results of Test Cases**

Test Case ID	Total Test Cases	Total Success	Total Failed
TC_100	10	10	-
TC_200	3	3	-
TC_300	5	5	-
TC_400	3	3	-
TC_500	3	3	-
TC_600	3	3	-
TC_700	3	3	-
Total	30	30	-

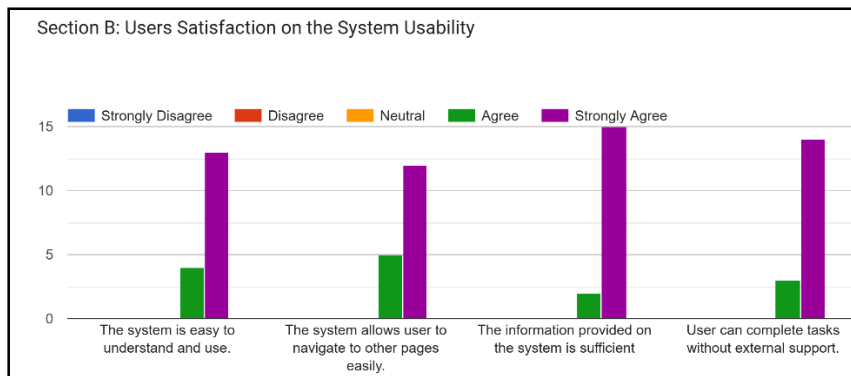
User Acceptance Testing (UAT) is a vital phase in software development that assesses the system from the perspective of end-users and stakeholders. Its primary objective is to verify that the system meets user expectations, satisfies the documented requirements, and performs effectively in real-world scenarios. The UAT form used for feedback collection is provided in Appendix B. The evaluation was conducted with 17 users across three main aspects: system functionality, usability, and user interface design.

Figure 5.1(a) displays the UAT results for System Functionality. The majority of users strongly agree that the SJMS functions well and meets their needs. They also affirm that the system significantly reduces data entry errors, simplifies the member tracking process, and executes tasks without noticeable delays. This indicates that the core system modules—such as login, registration, badge and activity management, and reporting—are operating as intended and provide high value to users.



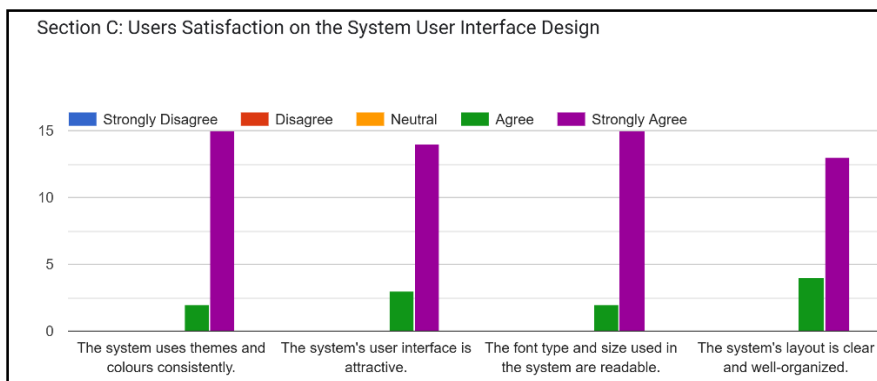
**Fig. 5.1(a):** Users Satisfaction on the System Functionality

Figure 5.2(b) presents the feedback on System Usability, which received highly positive ratings. Most respondents strongly agree that the system is easy to understand and navigate, allowing users to complete tasks efficiently without requiring additional support. Furthermore, users confirmed that the information displayed in the system is sufficient for them to carry out their roles effectively. This demonstrates that the system is user-friendly and intuitive for both admins and members.



**Fig. 5.2(b):** Users Satisfaction on the System Usability

Figure 5.3(c) shows the UAT findings on User Interface Design, which also reflect a highly favorable response. Most users strongly agree that the interface uses consistent themes and colors, features an attractive layout, employs readable fonts, and is overall well-organized. These aspects contribute to a positive user experience and reflect good design practices in interface development.



**Fig. 5.3(c):** Users Satisfaction on the System User Interface Design

Figure 5.4(d) summarises the open-ended feedback collected from users, which was overwhelmingly positive. Users praised the system for its clean and attractive interface, fast response time, and overall ease of use. Several suggestions for improvement were also provided, including implementing two-factor authentication (2FA) to enhance security, integrating Telegram for system notifications, developing a mobile app for iPhone users, adding multi-language support to increase accessibility, and including recommendations for learning materials. These suggestions align well with the system’s future enhancement roadmap and demonstrate strong user engagement and satisfaction.

Do you have any additional comments, suggestions or feedback about the Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS)? Your insights are valuable and will help us make further improvements.

8 responses

- enhance the system security like 2fa
- The ui is attractive
- have learning materials recommendation
- Add telegram into the system
- can have mobile app for iPhone
- Multi languages support
- nice and clean interface
- system respond is fast

**Fig. 5.4(d):** Comments, Suggestions, and Feedback about SJMS

## 6. Conclusion

The development of the Membership Management System for St. John Ambulance of Malaysia SMK Jalan Pasir Puteh (SJMS) has successfully met all defined project objectives, including the design, implementation, and testing of a centralized, web-based platform. The system has addressed critical limitations of the previous manual approach by introducing key modules such as registration, profile management, activity tracking, badge management, reporting, and learning materials. It streamlines operations, reduces human errors, and enhances accessibility for super admin, admins, and members. Feedback gathered through user acceptance testing (UAT) confirmed high user satisfaction in terms of functionality, usability, and interface design. Although the system currently has limitations, such as lacking mobile application support and advanced notification features, these can be improved in future iterations. Recommended future enhancements include the development of a mobile-friendly version or dedicated mobile app, integration with messaging platforms like Telegram for real-time notifications, multi-language support to accommodate a broader user base, and the implementation of two-factor authentication (2FA) for improved security. Overall, the implementation of SJMS significantly improves administrative efficiency, ensures data accuracy, and promotes engagement, making it a valuable solution for the ongoing management and growth of the organization.

## Acknowledgement

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

## Conflict of Interest

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

## Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Wong Lik, Rabatul Aduni Sulaiman; **data collection:** Wong Lik, Rabatul Aduni Sulaiman; **analysis and interpretation of results:** Wong Lik, Rabatul Aduni Sulaiman; **draft manuscript preparation:** Wong Lik, Rabatul Aduni Sulaiman. All authors reviewed the results and approved the final version of the manuscript.

## References

- [1] Wikipedia Contributors (2024). St. John Ambulance of Malaysia. Wikipedia; Wikimedia Foundation. [https://en.wikipedia.org/wiki/St.\\_John\\_Ambulance\\_of\\_Malaysia](https://en.wikipedia.org/wiki/St._John_Ambulance_of_Malaysia)
- [2] Wikipedia Contributors (2011). Sekolah Menengah Kebangsaan Jalan Pasir Puteh. Wikipedia.org; Yayasan Wikimedia Bhd. [https://ms.wikipedia.org/wiki/Sekolah\\_Menengah\\_Kebangsaan\\_Jalan\\_Pasir\\_Puteh](https://ms.wikipedia.org/wiki/Sekolah_Menengah_Kebangsaan_Jalan_Pasir_Puteh)
- [3] Hambali, M. H., Alsamman, M., Osman, B., & Mohammed, F. (2024). *DESIGN AND EVALUATION OF A WEB-BASED MEMBERSHIP MANAGEMENT SYSTEM*. Journal of Digital System Development, 2(1), 17–29. <https://doi.org/10.32890/jdsd2024.2.1.2>
- [4] Sihotang, J. I., Richel, Y., & Pakpahan, A. F. (2019). *Membership Information System Using Node JS*. Abstract Proceedings International Scholars Conference, 7(1), 1729–1740. <https://doi.org/10.35974/isc.v7i1.1372>
- [5] Yap, S., Yap, S., & Aziz, R. A. (2020). *MACMA Membership Management System with QR Code*. In International Journal of Advanced Computing Science and Engineering 2(1), 1-13.

- [6] Dennis, A., Wixom, B. H., & Roth, R. M. (2018). *Systems Analysis and Design Seventh Edition*.
- [7] Scott Tilley. (2020). *System Analysis and Design, Twelfth Edition*. <https://www.cengageasia.com>