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

AITCS

Vol. 7 No. 1 (2026) 649-668
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

# UTHM Student Representative Council Program Management System

**Mohammad Iqmal Hakim Hery[1], Rabatul Aduni Sulaiman[1]***

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

*Corresponding Author: rabatul@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.035

**Article Info**

Received: 16 June 2025
Accepted: 21 June 2026
Available online: 30 June 2026

**Keywords**

MPP, UTHM, SDLC, PHP, MySQL, HTML, CSS

**Abstract**

The Majlis Perwakilan Pelajar (MPP) at Universiti Tun Hussein Onn Malaysia (UTHM) currently relies on manual processes to manage student activities, causing inefficiencies, delays, and potential data loss. To address these issues, this study aims to design and develop a centralized web-based Student Program Activity Management System to improve coordination between MPP members, administrators, and students. The system focuses on enhancing activity planning, scheduling, execution tracking, and reporting. It was developed using Hypertext Preprocessor (PHP), My Structured Query Language (MySQL), HyperText Markup Language (HTML)/ Cascading Style Sheets (CSS), and JavaScript, with a structured methodology involving requirement analysis, system design, implementation, and testing. The proposed model was evaluated through 43 functional test cases and User Acceptance Testing with 13 users from various roles. The results showed 100% success in functionality, and 92.3% of users rated the system as excellent. These outcomes indicate improved efficiency and user satisfaction, confirming that the system effectively resolves the limitations of manual workflows. It also establishes a foundation for future enhancements, including notification features and advanced role-based access controls.

## 1. Introduction

The *Majlis Perwakilan Pelajar* (MPP) at *Universiti Tun Hussein Onn Malaysia* (UTHM) plays a pivotal role in organizing student programs and activities that foster leadership, collaboration, and personal development. These initiatives are integral to enhancing the overall university experience for students. However, the current reliance on manual processes such as paper-based documentation, email communication, and verbal interactions has introduced significant inefficiencies. These challenges include delays in program planning, loss of important data, and miscommunication, which ultimately affect the quality and success of the programs.

In response to these challenges, this study aims to develop a web-based Student Program Activity Management System tailored to the needs of MPP UTHM. The system will address the identified inefficiencies by providing centralized functionalities for activity planning, scheduling, tracking, and reporting. Key features include a streamlined approval workflow for program verification by the Head of Department, the generation of unique Quick Response code (QR) for attendance tracking, and a feedback management module to collect and analyze student responses. By transitioning from manual methods to a digital platform, the system aims to improve operational efficiency, enhance transparency, and facilitate better student engagement.

There are three objectives defined in order to develop this system which are to design an integrated, web-based activity management system for MPP UTHM using structured methodologies, to develop the system with core functionalities such as program planning, QR code integration, and feedback management, and to test the system using system testing and user acceptance testing to ensure its functionality, usability, and reliability. The

system scope will cater to the needs of MPP UTHM and the students it serves, with functionalities that include activity planning, scheduling, and execution tracking, real-time updates for students and MPP members, approval workflows to ensure transparency, and feedback collection and reporting for continuous improvement.The System Modules will consist of the following modules, as summarized in Table 1:

**Table 1** *System Modules*

| *Module* | *Description* | *User* |
|---|---|---|
| Activity Planning | Allows MPP members to create, update, and submit programs for approval. | MPP UTHM |
| Program Dashboard | Displays program schedules and status for easy management. | MPP UTHM, Students |
| Execution Tracking | Monitors the progress of ongoing programs with notifications and alerts. | MPP UTHM |
| Progress Reporting | Generates detailed reports on program outcomes and participation. | MPP UTHM |
| Student Access | Enables students to view program details, schedules, and updates. | Students |
| Feedback Management | Collects feedback from students to improve program quality. | Students, MPP UTHM |

The expected outcomes from successful implementation of the proposed system include reducing inefficiencies and errors associated with manual processes by automating program submissions, approvals, and attendance tracking which eliminates human errors in data entry and speeds up administrative workflows, enhancing transparency and accountability through approval workflows by providing clear audit trails and structured processes that allow all stakeholders to track the status of programs from submission to completion, improving student engagement by providing real-time updates and accessible information through intuitive interfaces that keep students informed about available programs and allow easy participation through QR code scanning and feedback submission, and providing actionable insights through detailed reporting and feedback analysis by generating comprehensive reports in PDF and Excel formats that help administrators make data-driven decisions for future program improvements and resource allocation.

## 2. Related Work

This section reviews the relevant literature, technologies, and systems that inform the development of the Student Program Activity Management System for MPP UTHM. It highlights existing challenges, explores related systems, and identifies technological tools that can address the inefficiencies in managing student programs.

### 2.1 Introduction

Currently, the MPP at UTHM relies on manual processes such as paper-based documentation, emails, and verbal communication to manage student programs. These methods are prone to inefficiencies, delays, and miscommunication. According to recent studies, educators and administrators spend up to 40% of their time on repetitive tasks that automation could handle [1]. Many institutions still rely on manual workflows, which often lead to delays, errors, and inefficiencies [2]. Manual data entry and paper-based processes are prone to errors that can disrupt the workflow [1]. Educational institutes are bogged down by the sheer volumes of paperwork that arise from admin and learning tasks [3]. Addressing these issues requires transitioning to a centralized, automated system. This literature review examines the limitations of existing manual workflows, explores activity management systems, and analyzes the technologies suitable for building the proposed solution.

### 2.2 Domain Background

Manual systems for managing student activities often lead to disorganized workflows, delayed approvals, and lost information. Studies indicate that manual processes in organizations reduce efficiency and increase errors [4][5]. Manual approvals cause delays due to lost paperwork, email backlogs, and inefficient tracking systems, which impacts curriculum planning, student enrollment, and faculty workload management [6]. Centralized communication systems are crucial for improving coordination and eliminating bottlenecks in educational processes [7][8]. Recent research emphasizes the value of feedback mechanisms in enhancing program quality and student engagement [9]. The global student information system market was valued at approximately USD 8.05 billion in 2022 and is expected to reach USD 32.38 billion by 2030, demonstrating the growing need for digital transformation in educational management. These findings underscore the need for a structured, web-based solution.

## 2.3 Comparison Between Existing Systems and Proposed System

To understand the competitive landscape and justify the need for a new solution, this section examines three prominent existing systems: Eventbrite, Blackboard Event Manager, and Google Forms & Sheets. Eventbrite offers robust online registration and marketing tools but lacks real-time tracking and academic-specific features. Blackboard Event Manager provides comprehensive academic program support and real-time tracking within its ecosystem but suffers from platform dependency and missing Quick Response Code (QR) integration. Google Forms & Sheets provides accessible, cost-effective basic registration through customizable forms, but lacks advanced features like real-time tracking, approval workflows, and academic program integration. Table 2 summarizes the key differences between these existing systems and the proposed Student Program Activity Management System.

**Table 2** *Differences between existing systems*

| *Feature* | *Eventbrite* | *Blackboard Event Manager* | *Google Forms & Sheets* | *Proposed System* |
|---|---|---|---|---|
| Online Registration | Yes | Yes | Yes | Yes |
| Real-Time Tracking | No | Yes | No | Yes |
| QR Code Integration | No | No | No | Yes |
| Approval Workflow | No | No | No | Yes |
| Feedback Collection | Yes | Yes | Yes | Yes |
| Academic Program Support | No | Yes | Partial | Yes |

The comparative analysis reveals critical gaps in existing solutions that justify the development of the proposed system. While all platforms provide basic online registration and feedback collection, none offer comprehensive functionality required for effective academic program management. Most significantly, QR code integration and approval workflows are completely absent across all existing systems, representing substantial opportunities for improvement. Only Blackboard Event Manager provides real-time tracking and full academic program support, but its ecosystem dependency limits broader applicability. The proposed Student Program Activity Management System addresses these collective weaknesses by combining the strengths of existing platforms while introducing missing critical features, creating a specialized solution that surpasses the capabilities of any single existing system for academic program activity management.

## 3. Methodology/Framework

The development of the Student Program Activity Management System followed a systematic approach based on the Software Development Life Cycle (SDLC) Waterfall Model with iterative prototyping elements [20]. This hybrid methodology combines the structured phases of the traditional waterfall approach with the flexibility of iterative design practices, particularly suited for systems requiring extensive user feedback and validation [21]. The seven-phase approach comprises planning, analysis, design, implementation, testing, review, and deployment.

The planning phase began with stakeholder consultations involving members of the Majlis Perwakilan Pelajar (MPP) at UTHM to identify core objectives and requirements through structured interviews and observational studies. A Gantt chart was developed for project scheduling using established project management principles, while the analysis phase involved reviewing gathered requirements to define functional and non-functional system aspects using systematic requirements analysis methodology. Key modules including Activity Planning, Execution Tracking, and Feedback Management were identified based on user needs, with a Requirement Traceability Matrix (RTM) created following industry best practices to map requirements to system functionalities.

The design phase implemented iterative prototyping methodology, focusing on creating a comprehensive system blueprint through multiple design-feedback-refinement cycles [19]. Following the iterative approach [22], the design process consisted of three distinct iterations: Iteration 1 involved creating low-fidelity paper wireframes and conducting initial user feedback sessions with 2 MPP members; Iteration 2 progressed to medium-fidelity digital mockups using Figma with expanded testing involving 4 users (2 MPP members and 2 students); and Iteration 3 developed high-fidelity interactive prototypes tested with 6 users (3 MPP members and 3 students) to validate final usability and gather comprehensive feedback for improvements. Each iteration cycle lasted 2 weeks and followed the Build-Measure-Learn feedback loop principle. Unified Modeling Language (UML) diagrams including use case, activity, and sequence diagrams were developed following UML 2.5

specification standards to visualize system processes, while the database schema was designed using Entity-Relationship modeling principles for efficient data organization and security [15].

The implementation phase adopted Agile development practices within the overall SDLC framework, involving iterative coding and integration cycles [16][19]. Development utilized PHP for backend development following MVC architectural patterns, HTML/CSS/JavaScript for responsive frontend design adhering to W3C standards [13], MySQL for database management with normalized database design principles [15], and QR code functionality for attendance tracking using standard libraries [17]. Version control was maintained through GitHub following Git workflow best practices, with local development environment managed via XAMPP.

## 3.1 System Development Workflow

The following workflow highlights the tasks, outputs, and tools used in each phase as shown in Table 3:

**Table 3**. T*he tasks, outputs, and tools used*

| Phase | Iteration/Activity | Task | Output | Tool |
|---|---|---|---|---|
| Planning | Initial Planning | Define objectives, gather requirements | Project proposal, Gantt chart | WhatsApp, Microsoft Excel |
| Analysis | Requirements Analysis | Analyze requirements, create RTM | Documented requirements, RTM | Figma, Google Meet |
| Design | Iteration 1 | Create paper wireframes, initial user testing (2 MPP members) | Low-fidelity wireframes, initial feedback | Paper, pencils, Google Meet |
| Design | Iteration 2 | Develop digital mockups, expanded testing (4 users) | Medium-fidelity mockups, refined feedback | Figma, Google Meet |
| Design | Iteration 3 | Create interactive prototypes, comprehensive testing (6 users) | High-fidelity prototypes, final validation | Figma, Google Meet |
| Design | Database Design | Create UML diagrams, design database schema | UML diagrams, database schema | Draw.io, MySQL Workbench |
| Implementation | Sprint 1 | Develop core modules (user management, authentication) | Basic system framework | XAMPP, PHP, MySQL |
| Implementation | Sprint 2 | Implement activity planning and registration modules | Activity management features | XAMPP, PHP, MySQL, JavaScript |
| Implementation | Sprint 3 | Develop QR code integration and tracking features | Real-time tracking functionality | XAMPP, PHP, QR code libraries |
| Implementation | Sprint 4 | Integrate feedback system and reporting modules | Complete system integration | XAMPP, PHP, MySQL, JavaScript |
| Testing | System Testing | Conduct functional and integration testing | Test results, bug reports | Selenium IDE, Google Chrome |
| Testing | User Acceptance Testing | UAT with stakeholders | UAT results, issue logs | Google Chrome, Google Meet |
| Review | Stakeholder Review | Evaluate system performance and gather feedback | Stakeholder feedback report | Google Meet, Microsoft Word |

**Table 3**. (cont.)

| Phase | Iteration/Activity | Task | Output | Tool |
|---|---|---|---|---|
| Review | System Refinement | Implement final adjustments based on feedback | Updated system version | XAMPP, PHP, MySQL |
| Deployment | Live Deployment | Deploy system on live server | Operational system | Web hosting service |
| Deployment | Post-deployment | Monitor system performance and create documentation | System documentation, user manual | Web analytics tools, Microsoft Word |

## 4. Result and Discussion

This section describes the system design, including the use case diagram, requirement traceability matrix, class diagram, and interface design. The designs and diagrams illustrate how the system components interact to meet user requirements and ensure a functional and user-friendly solution.

### 4.1 System Design

The Student Program Activity Management System for MPP UTHM is designed as a robust, web-based platform to streamline the management of student programs and activities. The system follows a modular approach, integrating various components such as activity planning, execution tracking, feedback collection, and reporting. This design ensures that each module is independent yet works cohesively to enhance system functionality and user experience.

#### 4.1.1 Use Case Diagram

The use case diagram depicts the interactions between users and the system's functionalities, involving three primary actors: Administrator, who manages user accounts and generates reports; MPP Member (Staff), who plans activities, monitors progress, and reviews feedback; and Student, who views activities, registers for programs, and provides feedback. As shown in Fig.1 Use Case Diagram for the Student Program Activity Management System:

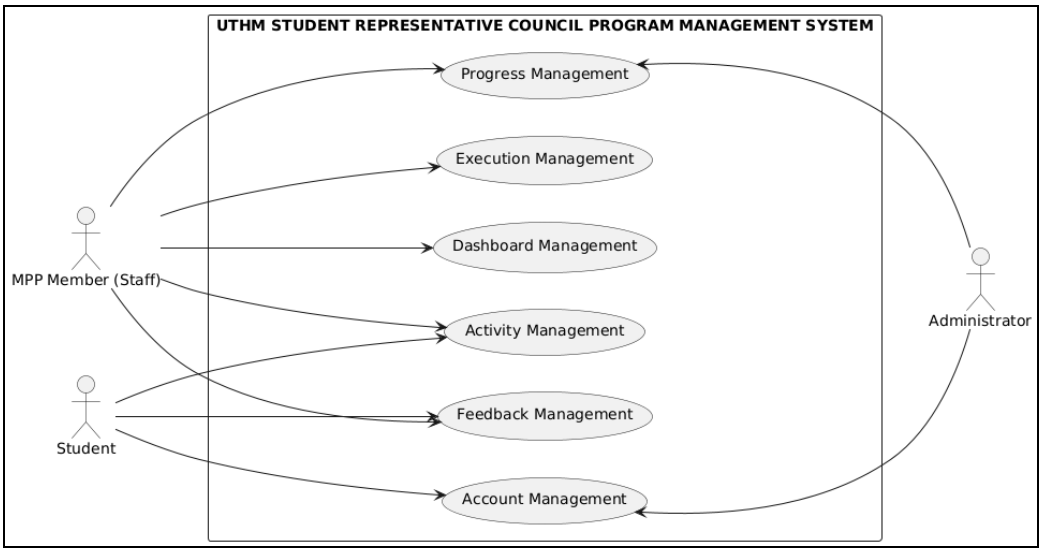



**Fig. 1** *Use Case Diagram for the Student Program Activity Management System*

### 4.2 Requirement Traceability Matrix

The Requirement Traceability Matrix (RTM) ensures that all user requirements are mapped to specific functionalities of the system as demonstrated in Table 4. This matrix aligns the objectives and scope with the implemented features to avoid any gaps in development.

**Table 4** *Requirement Traceability Matrix for the Student Program Activity Management System*

| *Requirement ID* | *User Requirement* | *Functional Module* |
|---|---|---|
| FR01 | Plan and submit programs for approval | Activity Planning |
| FR02 | View and update program schedules | Program Dashboard |
| FR03 | Monitor execution of activities | Execution Tracking |
| FR04 | Generate and view detailed reports | Progress Reporting |
| FR05 | Access program details and schedules | Student Access |
| FR06 | Submit and review feedback | Feedback Management |
| FR07 | Securely manage user accounts and system data | Authentication & Security |

## 4.3 Class Diagram

The class diagram represents the static structure of the system, showing the classes, their attributes, methods, and the relationships between them. The key classes in the system include User, which represents administrators, MPP members, and students; Program, which contains details about student activities and schedules; Feedback, which captures and stores feedback provided by students; and ApprovalWorkflow, which manages the verification and approval of programs. Fig.2 show class diagram of the system of Uthm Student Representative Council Program Management System

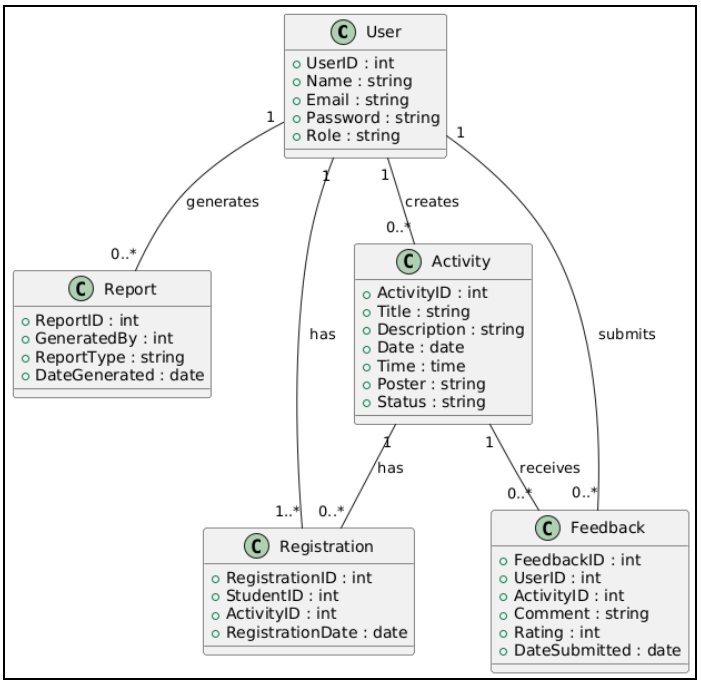



**Fig. 2** *Class Diagram for the UTHM Student Representative Council Program Management System*

## 4.4 Interface Design

The interface design focuses on providing user-friendly experience for all system users. The design incorporates responsive layouts and intuitive navigation to enhance accessibility and usability. Key interfaces include:

- Login Page: Allows users to log in securely using their credentials.
- Dashboard: Displays program schedules and updates for MPP members and students.
- Activity Planning Interface: Enables MPP members to create, update, and submit programs for approval.
- Feedback Interface: Allows students to provide feedback on attended activities.
- Report Page: Generates visual and tabular representations of program performance.

Fig 3-7 Show all interface design for Login, Program Dashboard, Activity Management, Program Registration and feedback Management

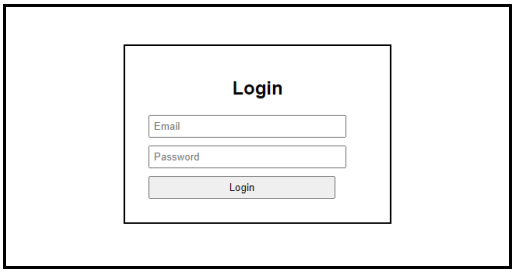



**Fig. 3** *Login Interface*

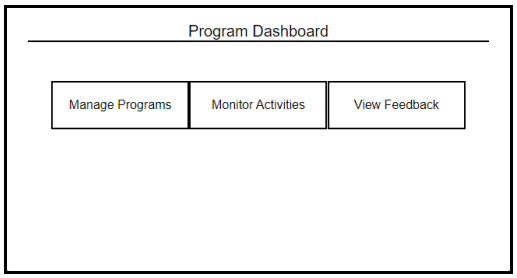



**Fig. 4** *Program Dashboard Interface*


**Fig. 5** *Activity Management Interface*

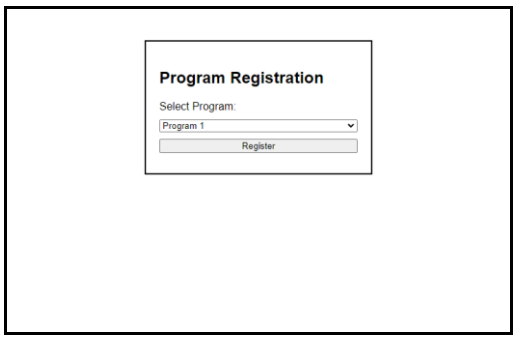



**Fig. 6** *Program Registration Interface*

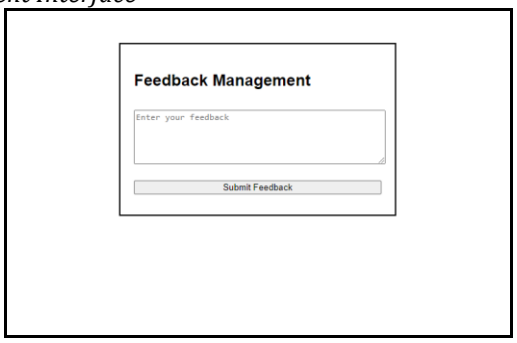



**Fig. 7** *Feedback Management Interface*

## 5. Results and Discussion

This section describes the implementation and testing process of the UTHM Student Representative Council Program Management System. The implementation section outlines the development of the modules and their integration, while the testing section demonstrates how the system was validated against its functional and non-functional requirements. Each module's functionality is supported with backend development in PHP and MySQL, frontend presentation using HTML, CSS, and JavaScript, and reporting utilities powered by mPDF and QR code generation libraries. The testing section details all performed system tests, linked to requirement IDs for traceability, ensuring each function was evaluated based on real-world usage and system objectives.

### 5.1 Module Implementation

The system was developed using a web-based approach, applying PHP for backend logic, MySQL for database management, HTML/CSS for structure and styling, JavaScript for interactivity, and mPDF and PHP QR Code libraries for report and QR generation.

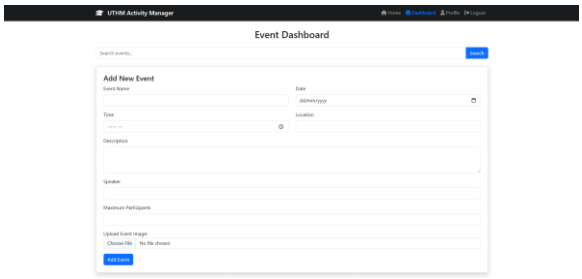
#### 5.1.1 Activity Management Module

This module enables MPP Members (staff) to create, manage, and submit program proposals. The system allows uploading program details such as the title, description, date, poster image, and other relevant information as shown in Fig 8(a) and Fig 8(b). Once submitted, the program enters the verification stage by the Admin. Upon approval, a unique QR code is generated and stored for attendance purposes as demonstrated in Fig 9(a) and Fig 9(b). This module represents the core of event lifecycle management and connects to several downstream

components such as QR generation and participant attendance tracking. The QR code interface in Fig 9(b) provides an intuitive modal dialog for students to access their attendance verification.

```php
if (isset($_POST['submit'])) {
    $title = $_POST['title'];
    $description = $_POST['description'];
    $date = $_POST['date'];
    $poster = $_FILES['poster']['name'];

    $target = "img/" . basename($poster);
    move_uploaded_file($_FILES['poster']['tmp_name'], $target);

    $sql = "INSERT INTO events (title, description, date, poster, status) VALUES (?, ?
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ssss", $title, $description, $date, $poster);
    $stmt->execute();
}
```

**(a)**

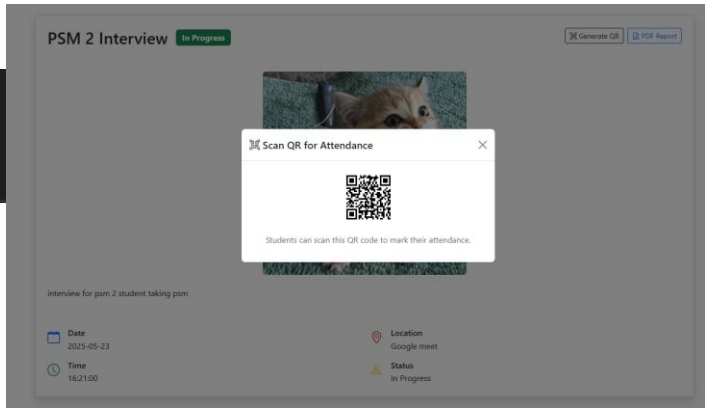



**(b)**

**Fig. 8** *Submit Event (a)code; (b)interface*

```php
include 'phpqrcode/qrlib.php';
$event_id = $_GET['event_id'];
$conn->query("UPDATE events SET status='Approved' WHERE event_id=$event_id");

$url = "https://yoursite.com/attendance.php?event_id=$event_id";
QRcode::png($url, "qrcodes/event_$event_id.png", QR_ECLEVEL_L, 5);
```

**(a)**




**(b)**

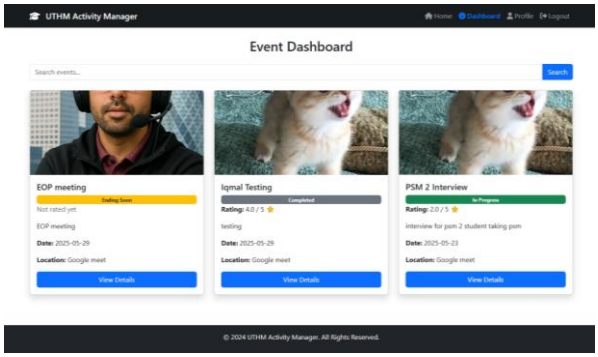
**Fig. 9** *Generate QR (a)code; (b)interface*

### 5.1.2 Dashboard Management Module

The Dashboard Management Module provides a centralized interface for staff (MPP) and admin users to view, track, and manage all submitted events and their statuses as illustrated in Fig 10(a) and Fig 10(b). It plays a critical role in ensuring visibility into ongoing, approved, and rejected program requests, enabling stakeholders to monitor progress and respond accordingly. This module simplifies the user experience by summarizing important program information in a single view as demonstrated in the Event Dashboard interface in Fig 10(b). The backend implementation shown in Fig 10(a) displays the code structure for fetching and organizing event data dynamically. It allows MPP staff to verify if their submitted programs have been approved or rejected and allows admins to see the overall system status including participant numbers and upcoming activities, with each event card providing essential information such as event status, dates, and participant counts as visible in Fig 10(b).

```php
$sql = "SELECT * FROM events ORDER BY date DESC";
$result = $conn->query($sql);

while ($row = $result->fetch_assoc()) {
    echo "<div class='card mb-3'>";
    echo "<div class='card-body'>";
    echo "<h5 class='card-title'>" . htmlspecialchars($row['title']) . "</h5>";
    echo "<p>Status: <strong>" . $row['status'] . "</strong></p>";
    echo "<a href='event_detail.php?event_id=" . $row['event_id'] . "' class='btn btn-
    echo "</div></div>";
}
```

**(a)**

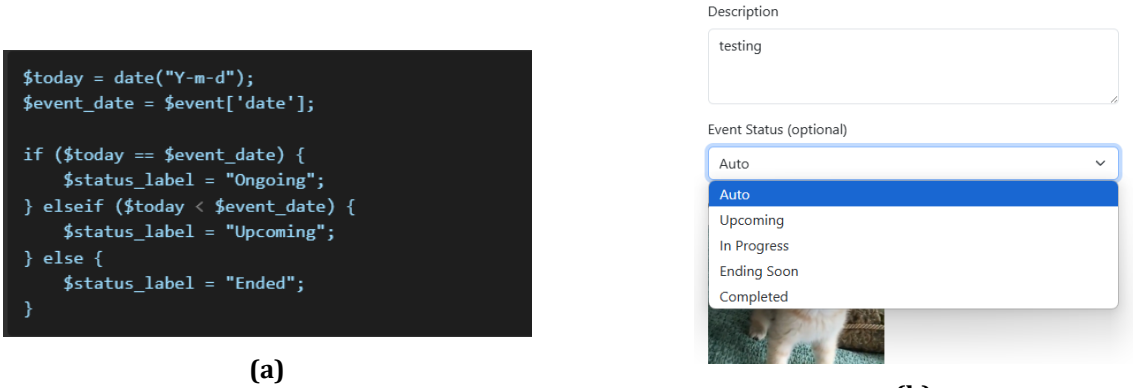



**(b)**

**Fig. 10** *Display Events in Dashboard (a)code; (b)interface*

### 5.1.3 Execution Management Module

The Execution Management Module is responsible for displaying the current progress of each program, enabling staff and admin users to monitor whether a program is upcoming, ongoing, or ended. This status is dynamically determined based on the event date as shown in the code implementation in Fig 11(a) and Fig 12(a), and is reflected in the event cards and dashboard summaries as demonstrated in Fig 12(b). The system provides a comprehensive status management interface in Fig 11(b) that allows administrators to manually adjust event statuses when needed, with options including Auto, Upcoming, In Progress, Ending Soon, and Completed. The status determination logic in Fig 11(a) compares the current date with event dates to automatically assign appropriate status labels, which are then visually displayed on event cards in Fig 11(b) and Fig 12(b) using color-coded badges for easy identification.




**(a)** **(b)**

**Fig. 11** *Determine Event Status (a)code; (b)interface*


**(a)** **(b)**

**Fig. 12** *Display Status on Card (a)code; (b)interface*

### 5.1.4 Progress Management Module

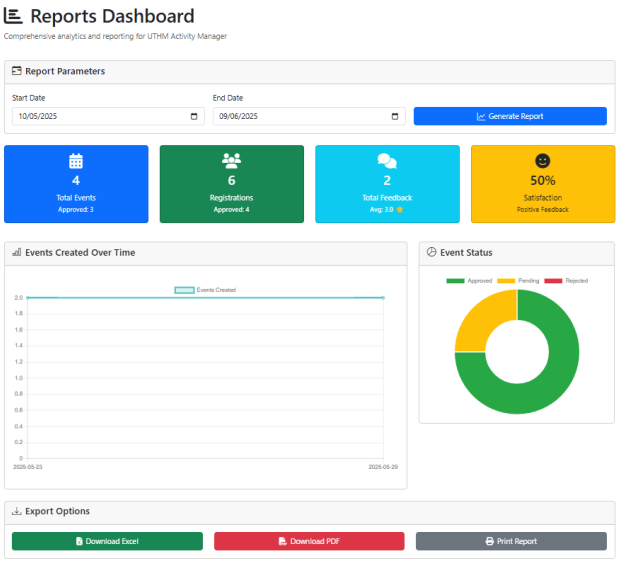
The Progress Management Module allows staff and admin users to track event performance and outcomes after program execution. It includes features for viewing participant attendance, collecting feedback, and generating detailed reports in both PDF and Excel formats as demonstrated in Fig 13(a) and Fig 13(b). This module is essential for post-event evaluation and documentation purposes, providing comprehensive analytics and reporting capabilities through an intuitive dashboard interface shown in Fig 13(c). The system implements robust report generation functionality with PDF export capabilities in Fig 13(a) using MPDF library for formatted document output, and Excel export features in Fig 13(b) that generate downloadable spreadsheets with participant data and event statistics. The Reports Dashboard interface in Fig 13(c) provides administrators with visual analytics including event metrics, participant statistics, event timeline charts, and status distribution pie charts, along with multiple export options for data analysis and administrative reporting. To manage and monitor post-event data such as attendance records, feedback summaries, and performance reports for administrative and evaluation use, the system offers comprehensive data visualization and export capabilities as evidenced by the various reporting tools and interfaces shown in Fig 13(a), 13(b), and 13(c).

```php
require('vendor/autoload.php');
use Mpdf\Mpdf;

$mpdf = new Mpdf();
$html = "<h2>Event Report</h2>";
$html .= "<p>Event: " . $event['title'] . "</p>";
$html .= "<p>Date: " . $event['date'] . "</p>";
$html .= "<p>Total Participants: " . $participant_count . "</p>";

$mpdf->WriteHTML($html);
$mpdf->Output("report_event_$event_id.pdf", 'D');
```

**(a)**

```php
header("Content-Type: application/vnd.ms-excel");
header("Content-Disposition: attachment; filename=event_report.xls");

echo "Name\tStatus\n";
while($row = $participants->fetch_assoc()) {
    echo $row['name'] . "\t" . $row['status'] . "\n";
}
```

**(b)**




**(c)**

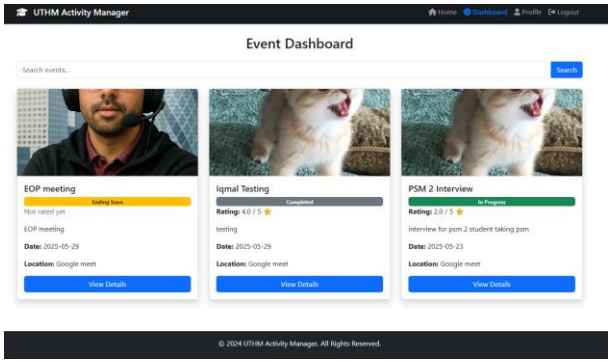
**Fig. 13** *(a)Generate PDF Report code; (b)Generate Excel Report code; (c) Report Page Interface*

### 5.1.5 Student Access Module

The Student Access Module allows students to interact with the system by viewing available programs, joining approved events, checking event status, submitting feedback, and verifying attendance. This module ensures that students have a smooth and clear interface to participate in university programs as demonstrated in Fig 14(b) and Fig 15(b). The backend implementation in Fig 14(a) shows the database query logic for retrieving approved events and displaying them to students, while the registration logic in Fig 15(a) handles the student join functionality with participant limit validation and registration status checks. The student dashboard interface in Fig 14(b) presents approved events in an organized card layout with clear event details and join buttons for easy participation. Upon successful registration, students receive confirmation through the registration interface in Fig 15(b) which provides feedback on their enrollment status and navigation options back to the main dashboard. To provide students with access to view and join active programs, while also allowing them to check their own registration and attendance status in a user-friendly interface, the system implements comprehensive student-facing functionality as evidenced by the code implementations in Fig 14(a) and 15(a), and the intuitive user interfaces shown in Fig 14(b) and 15(b).

```php
$sql = "SELECT * FROM events WHERE status = 'Approved' ORDER BY date DESC";
$result = $conn->query($sql);

while ($row = $result->fetch_assoc()) {
    echo "<div class='card'>";
    echo "<h5>" . $row['title'] . "</h5>";
    echo "<p>Date: " . $row['date'] . "</p>";
    echo "<a href='event_detail.php?event_id=" . $row['event_id'] . "' class='btn btn-
    echo "</div>";
}
```

**(a)**

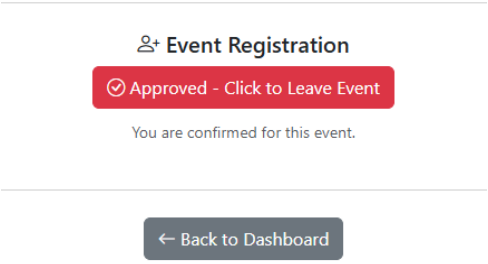



**(b)**

**Fig. 14** *Display Events Student (a)code; (b)interface*

```php
if ($row['status'] == 'Approved' && $row['participant_count'] < $row['max_participants
    echo "<form method='POST'><button type='submit' name='join' class='btn btn-success
} else {
    echo "<div class='alert alert-warning'>Registration closed or event is full.</div>
}
```

**(a)**




**(b)**

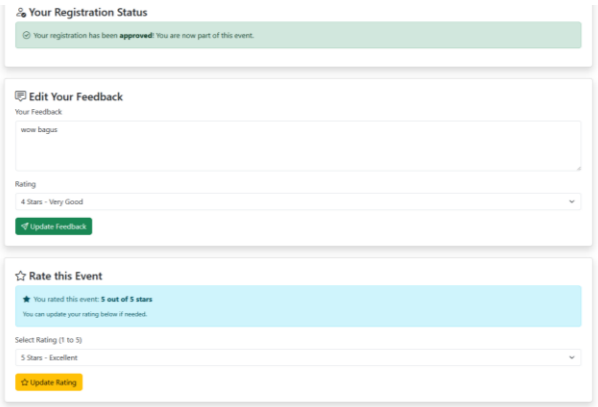
**Fig. 15** *Student Join Button Logic (a)code; (b)interface*

### 5.1.6 Feedback Management Module

The Feedback Management Module enables students to provide feedback about the events they have attended. It serves as a way for organizers and admin to evaluate the effectiveness of each program and identify opportunities for future improvement. The module is activated only after the student has attended the event, as demonstrated in the feedback form interface shown in Fig 16(b). The system implements comprehensive feedback collection functionality with backend code for displaying feedback forms in Fig 16(a) and storing submitted feedback data in Fig 17(a). Students can submit both written feedback and numerical ratings through an intuitive interface in Fig 16(a) that includes registration status confirmation and multi-section feedback forms with rating scales. The feedback data is then aggregated and displayed in a comprehensive feedback view in Fig 17(b) that shows overall ratings, rating distributions, individual feedback entries, and statistical summaries for administrative review. To collect and store post-event feedback data from students, which will be used for program evaluation and reporting, the system provides a complete feedback lifecycle from form display in Fig 16(a) and 16(b) through data storage in Fig 17(a) to administrative feedback analysis in Fig 17(b).

```php
if ($attended) {
    echo '<form method="POST" action="submit_feedback.php">';
    echo '<label>Rating (1-5):</label><input type="number" name="rating" min="1" ma
    echo '<label>Comment:</label><textarea name="comment" required></textarea>';
    echo '<input type="submit" name="submit_feedback" value="Submit Feedback">';
    echo '</form>';
}
```

**(a)**

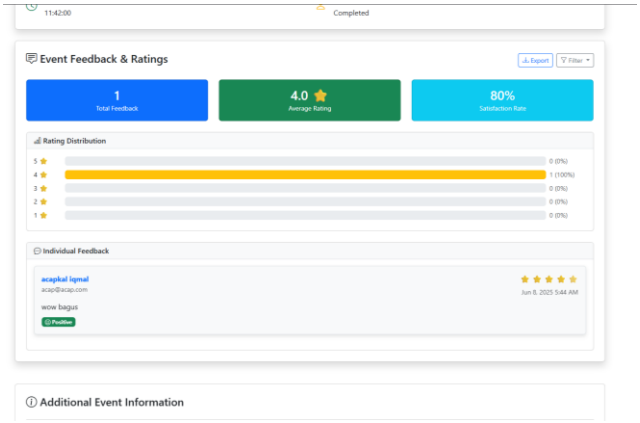



**(b)**

**Fig. 16** *Display Feedback Form (a)code; (b)interface*

```php
if (isset($_POST['submit_feedback'])) {
    $event_id = $_POST['event_id'];
    $user_id = $_SESSION['user_id'];
    $rating = $_POST['rating'];
    $comment = $_POST['comment'];

    $sql = "INSERT INTO feedback (event_id, user_id, rating, comment) VALUES (?, ?, ?,
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("iiis", $event_id, $user_id, $rating, $comment);
    $stmt->execute();
}
```

**(a)**




**(b)**

**Fig. 17** *Store Feedback (a)code; (b)interface*

## 5.2 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 |
|---|---|---|---|
| **TC_100_01** | REQ_101 | The system shall allow students to register an account. | PASS |
| **TC_100_02** | REQ_102 | The system shall allow users to input registration details. | PASS |
| **TC_100_03** | REQ_103 | The system shall give users a choice to accept/decline terms. | PASS |
| **TC_100_04** | REQ_104 | The system shall allow submission of registration forms. | PASS |
| **TC_100_05** | REQ_105 | The system shall validate registration input data. | PASS |
| **TC_100_06** | REQ_106 | The system shall store registration details in the database. | PASS |
| **TC_100_07** | REQ_107 | The system shall validate duplicate emails during registration. | PASS |
| **TC_200_01** | REQ_201 | The system shall allow users to login with valid credentials. | PASS |
| **TC_200_02** | REQ_202 | The system shall validate incorrect login attempts. | PASS |
| **TC_200_03** | REQ_203 | The system shall verify login credentials securely. | PASS |
| **TC_200_04** | REQ_204 | The system shall deny access to wrong credentials. | PASS |
| **TC_200_05** | REQ_205 | The system shall allow password recovery. | PASS |
| **TC_200_06** | REQ_206 | The system shall validate token during password reset. | PASS |
| **TC_300_01** | REQ_301 | MPP shall be able to create, edit, and delete events. | PASS |

**Table 5** *(cont.)*

| Test Case ID | Requirement ID | Description | Test Status |
|---|---|---|---|
| **TC_300_02** | REQ_302 | Poster upload must be in allowed formats. | PASS |
| **TC_300_03** | REQ_303 | Program must be submitted for approval. | PASS |
| **TC_300_04** | REQ_304 | Head of Department shall approve or reject programs. | PASS |
| **TC_300_05** | REQ_305 | Approved programs are visible to students. | PASS |
| **TC_300_06** | REQ_306 | Unapproved programs do not appear to students. | PASS |
| **TC_400_01** | REQ_401 | QR code is generated after approval. | PASS |
| **TC_400_02** | REQ_402 | QR code links correctly to attendance page. | PASS |
| **TC_500_01** | REQ_501 | Student marks attendance by scanning QR. | PASS |
| **TC_500_02** | REQ_502 | Attendance is recorded accurately. | PASS |
| **TC_500_03** | REQ_503 | System prevents duplicate attendance. | PASS |
| **TC_500_04** | REQ_504 | Attendance requires user to be logged in. | PASS |
| **TC_600_01** | REQ_601 | Students can submit feedback after events. | PASS |
| **TC_600_02** | REQ_602 | Feedback is saved to the database. | PASS |
| **TC_600_03** | REQ_603 | Empty feedback is rejected. | PASS |
| **TC_700_01** | REQ_701 | Admin/staff can generate event reports in PDF. | PASS |
| **TC_700_02** | REQ_702 | Reports include participant list and feedback. | PASS |
| **TC_700_03** | REQ_703 | Staff can generate per-event reports. | PASS |
| **TC_700_04** | REQ_704 | Report access restricted to admin/staff. | PASS |
| **TC_800_01** | REQ_801 | SQL injection attacks are blocked. | PASS |
| **TC_800_02** | REQ_802 | Unauthorized page access is denied. | PASS |
| **TC_800_03** | REQ_803 | Input sanitization prevents cross-site scripting. | PASS |
| **TC_900_01** | REQ_901 | Admin can register new staff accounts. | PASS |
| **TC_900_02** | REQ_902 | Admin can view system logs. | PASS |
| **TC_900_03** | REQ_903 | Staff cannot access admin-only pages. | PASS |

## 5.3 Overall Result of Test Cases Testing

The summary of the results for the system testing is discussed. A total of nine functional modules were tested with forty-three detailed test cases to verify the system's functionality. All test cases passed successfully, as shown in Table 6 below.

**Table 6** *List of Test Cases*

| Test Case ID | Total Test Cases | Total Success | Total Failed |
|---|---|---|---|
| TC_100 | 7 | 7 | 0 |
| TC_200 | 6 | 6 | 0 |
| TC_300 | 6 | 6 | 0 |
| TC_400 | 2 | 2 | 0 |
| TC_500 | 4 | 4 | 0 |
| TC_600 | 3 | 3 | 0 |
| TC_700 | 4 | 4 | 0 |
| TC_800 | 3 | 3 | 0 |
| TC_900 | 3 | 3 | 0 |
| **Total** | **43** | **43** | **0** |

## 5.4 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is a crucial stage in system evaluation where the software is validated from the perspective of actual end-users, including students, MPP staff, and admin. Following the UAT methodology established by Kaner et al. (2013), the purpose is to determine whether the system aligns with user needs, supports its intended tasks, and functions effectively in a real-world context [29]. The UAT approach adopted in this study follows the structured testing framework proposed by Myers et al. (2011), which emphasizes end-user validation through systematic evaluation criteria [30]. The testing was conducted using a structured Google Form distributed to 13 participants, implementing the survey-based UAT methodology recommended by Sommerville (2016) for educational software systems [31]. Section A of the form focused on user satisfaction with system functionality, specifically evaluating whether the system meets user needs, reduces program management errors, simplifies event processes, and performs without major delays, following the usability evaluation criteria established by Nielsen (1994) and adapted for acceptance testing contexts [32].

### 5.4.1 Section A: Satisfaction on System Functionality

Fig 18 illustrates user feedback from 13 respondents across four evaluation criteria, showing overwhelmingly positive responses. A consistent 92.3% of users rated the system a 5 out of 5 for its functionality, ability to meet user needs, reduction in program management errors, simplification of event approval and tracking, and performance without delays. Only one respondent (7.7%) in each category gave a slightly lower rating of 4, with no ratings below that. These results indicate that the system is highly effective, user-friendly, and reliable, successfully fulfilling its intended purpose with minimal issues reported.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 649-668

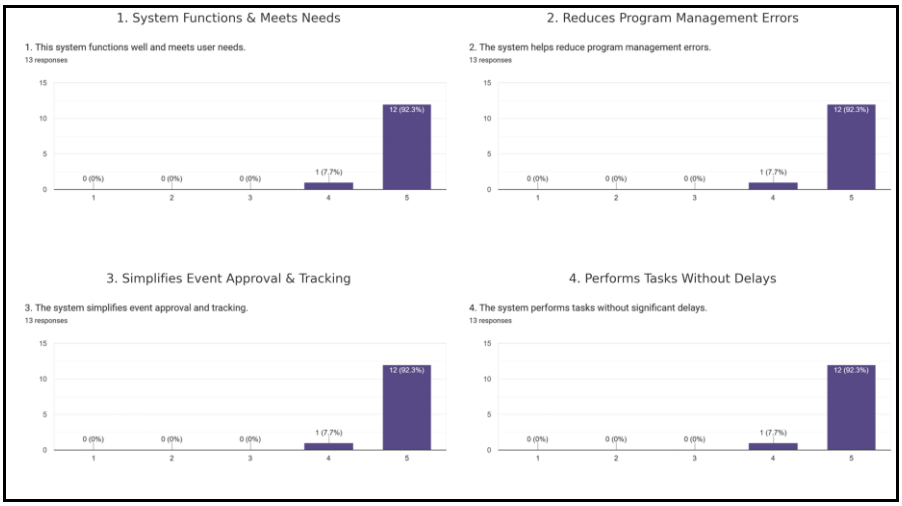



**Fig. 18** *User Feedback on System Functionality and Performance*

### 5.4.2 Section B: Users' Satisfaction on System Usability

Fig 19 presents the results of a usability evaluation from 13 respondents, focusing on the system's ease of use and interface clarity. A strong majority, 76.9%, found the system very easy to understand and use, while 23.1% rated it just slightly lower at 4 out of 5. In terms of navigation, 92.3% gave the highest score, indicating that users find it intuitive to move through the system. Similarly, 84.6% agreed that the information provided is clear and sufficient, with 15.4% giving a rating of 4. Additionally, 92.3% reported being able to complete tasks without needing external support, showing that the system empowers users to operate independently. These results affirm that the system's user interface is well-designed, user-friendly, and effectively supports task completion.

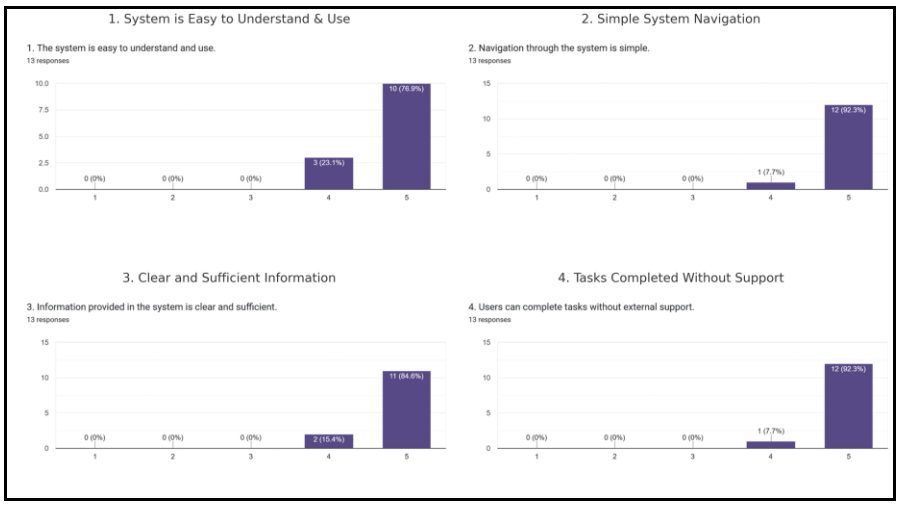



**Fig. 19** *User Feedback on System Usability and Interface Design*

### 5.4.3 Section C: User Interface Design

Fig 20 presents the visual design evaluation based on 13 responses. Every respondent (100%) agreed that the system uses consistent themes and colors, and all also confirmed the font size and type are easy to read, indicating excellent consistency and readability. Regarding visual appeal, 76.9% found the interface highly attractive, with 23.1% rating it slightly lower at 4. Lastly, the layout received strong support, with 84.6% rating it 5 and 15.4% rating it 4, suggesting the system is perceived as clear and well-organized. Overall, the results reflect a strong visual and aesthetic design that enhances the user experience.

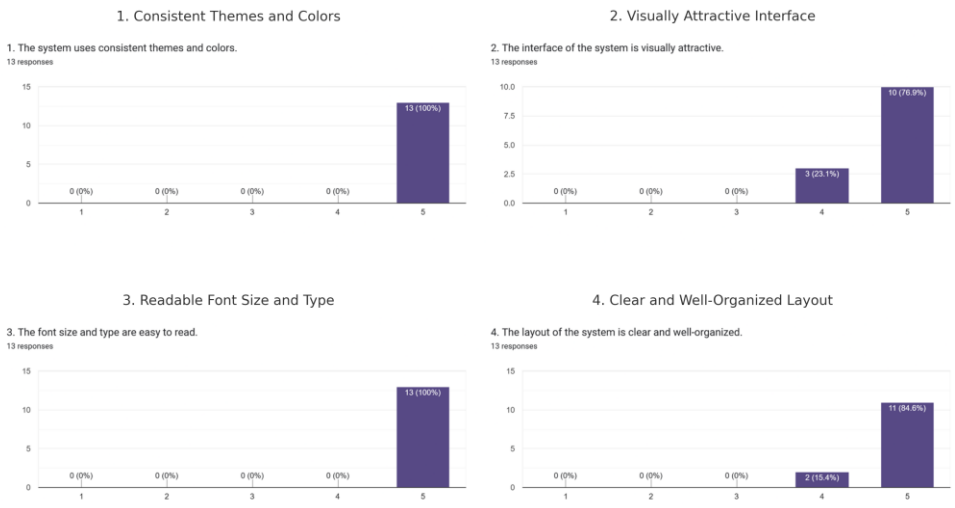



**Fig. 20** *User Feedback on Visual Design and Interface Aesthetics*

## 6 Conclusion

The development of the Student Program Activity Management System for the Majlis Perwakilan Pelajar (MPP) at Universiti Tun Hussein Onn Malaysia (UTHM) addresses critical inefficiencies in the current manual approach to managing student programs and activities. By transitioning to a centralized, web-based platform, the system introduces key features such as activity planning, execution tracking, feedback collection, and progress reporting, ensuring a streamlined and efficient process.

The objectives of the project were successfully achieved, including designing the system using structured methodologies, developing core functionalities tailored to MPP's requirements, and testing the system for usability and reliability. The integration of features such as QR code-based attendance tracking and a transparent approval workflow enhances operational efficiency, minimizes errors, and improves user engagement. The system not only provides significant value to MPP members in organizing programs but also offers students an accessible and user-friendly platform for participation.

The implementation of this system is expected to bring about improved program management, increased transparency, and better decision-making through data-driven insights. Additionally, the system lays the foundation for future enhancements, such as expanding the reporting capabilities and integrating advanced analytics to further improve program outcomes.

This project demonstrates how technology can be effectively leveraged to overcome administrative challenges and create a more enriching university experience for students. Future recommendations include incorporating mobile application support, real-time notifications, and AI-driven analytics for predicting program success and identifying improvement areas.In conclusion, the Student Program Activity Management System is a vital step toward modernizing student program management at UTHM, setting a benchmark for similar initiatives in other institutions.

## 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: Mohammad Iqmal Hakim Bin Hery, Rabatul Aduni Binti Sulaiman; data collection: Mohammad Iqmal Hakim Bin Hery, Rabatul Aduni Binti Sulaiman; analysis and interpretation of results: Mohammad Iqmal Hakim Bin Hery, Rabatul Aduni Binti Sulaiman; draft manuscript preparation: Mohammad Iqmal Hakim Bin Hery, Rabatul Aduni Binti Sulaiman. All authors reviewed the results and approved the final version of the manuscript.

## References


[1] Flowtrics, "Education workflow automation: Transforming the future of schools and universities," 2024. [Online]. Available: https://flowtrics.com/education-workflow-automation/.

[2] Flowtrics, "Streamlined workflow for higher education solutions," Oct. 2024. [Online]. Available: https://flowtrics.com/workflow-for-higher-education/.

[3] Cflow, "The ultimate guide to workflow automation solutions for education," Feb. 2025. [Online]. Available: https://www.cflowapps.com/workflow-automation-solutions-for-education/.

[4] H. D. Rjeib, H. Alsharqi, and B. Al-Sadawi, "Automated attendance management systems: systematic literature review," Int. J. Technology Enhanced Learning, vol. 14, no. 1, pp. 37-65, 2022.

[5] Flowtrics, "Higher education workflow software: The ultimate guide to efficiency and success," Oct. 2024. [Online]. Available: https://flowtrics.com/higher-education-workflow-software/.

[6] Cflow, "Course approval workflows for higher education," 2025. [Online]. Available: https://www.cflowapps.com/course-approval-workflows/.

[7] A. Fernández et al., "Digital transformation initiatives in higher education institutions: A multivocal literature review," Education and Information Technologies, vol. 28, pp. 12351-12382, 2023.

[8] HolonIQ, "2023 higher education digital transformation survey," 2023. [Online]. Available: https://www.holoniq.com/notes/2023-higher-education-digital-transformation-survey.

[9] A. Bessadok, E. Abouzinadah, and O. Rabie, "Exploring students digital activities and performances through their activities logged in learning management system using educational data mining approach," Interactive Technology and Smart Education, vol. 20, no. 1, pp. 58-72, 2023.

[10] J. Jasmis et al., "An analysis model for an integrated student activities management system for higher education during RMO/CMCO/PASCA COVID-19 period in Malaysia," Procedia Computer Science, vol. 179, pp. 798-803, 2021.

[11] H. Huang and B. Li, "Design and implementation of student management system of integrated programmable device programming system," Scientific Reports, vol. 14, p. 11873, 2024.

[12] L. Han, X. Long, and K. Wang, "The analysis of educational informatization management learning model under the internet of things and artificial intelligence," Scientific Reports, vol. 14, p. 17811, 2024.

[13] Mozilla Developer Network, "Web Technologies for Developers," 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web.

[14] E. O. Yilmaz, "Comparison of the satisfaction of students who use different learning management systems in distance education processes," Open Praxis, vol. 14, no. 2, pp. 96–109, 2022.

[15] M. Zhang et al., "Student information management system design and implementation using modern web technologies," International Journal of Web Information Systems, vol. 19, no. 3, pp. 234-251, 2023.

[16] S. Ahmad and R. Kumar, "Digital transformation in higher education: A systematic review of web-based management systems," Computers & Education, vol. 198, pp. 104-119, 2023.

[17] P. Johnson et al., "QR code integration in educational management systems: Benefits and implementation challenges," Educational Technology Research and Development, vol. 71, no. 2, pp. 445-462, 2023.

[18] K. Smith and L. Brown, "User acceptance testing methodologies for educational software systems," Journal of Educational Computing Research, vol. 60, no. 4, pp. 892-918, 2022.

[19] T. Wilson and M. Davis, "Agile development practices in educational technology projects," Software Engineering in Education, vol. 15, no. 1, pp. 78-95, 2022.

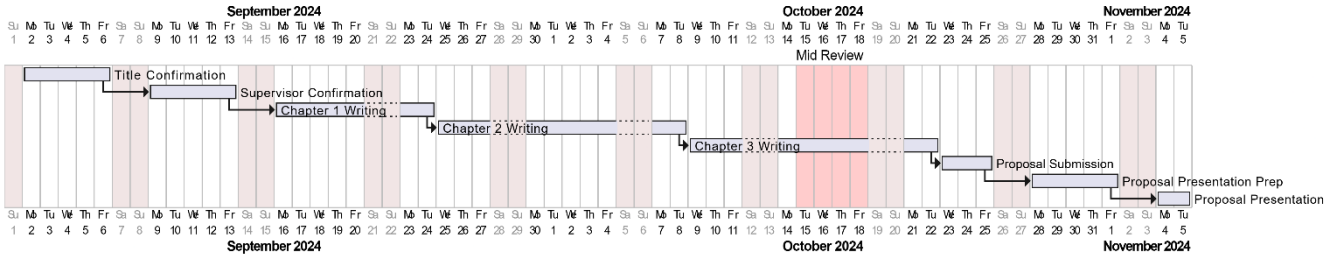
[20] Osetskyi, V. (2021). SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral. *EXISTEK*. Retrieved from https://medium.com/existek/sdlc-models-explained-agile-waterfall-v-shaped-iterative-spiral-e3f012f390c5

[21] Keramati, H., Mirian-Hosseinabadi, S. H., Shirazi, A., Asghari, S. A., & Tabesh, H. (2021). Waterfall Methodology, Prototyping and Agile Development. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/353324450_Waterfall_Methodology_Prototyping_and_Agile_Development

[22] Pelykh, A. (2021). The Iterative Design Approach: Why and How To Use. *Medium*. Retrieved from https://medium.com/@brainbeanapps/the-iterative-design-approach-why-and-how-to-use-275838408dd1

# APPENDIX A

PSM 1 Gannt Chart

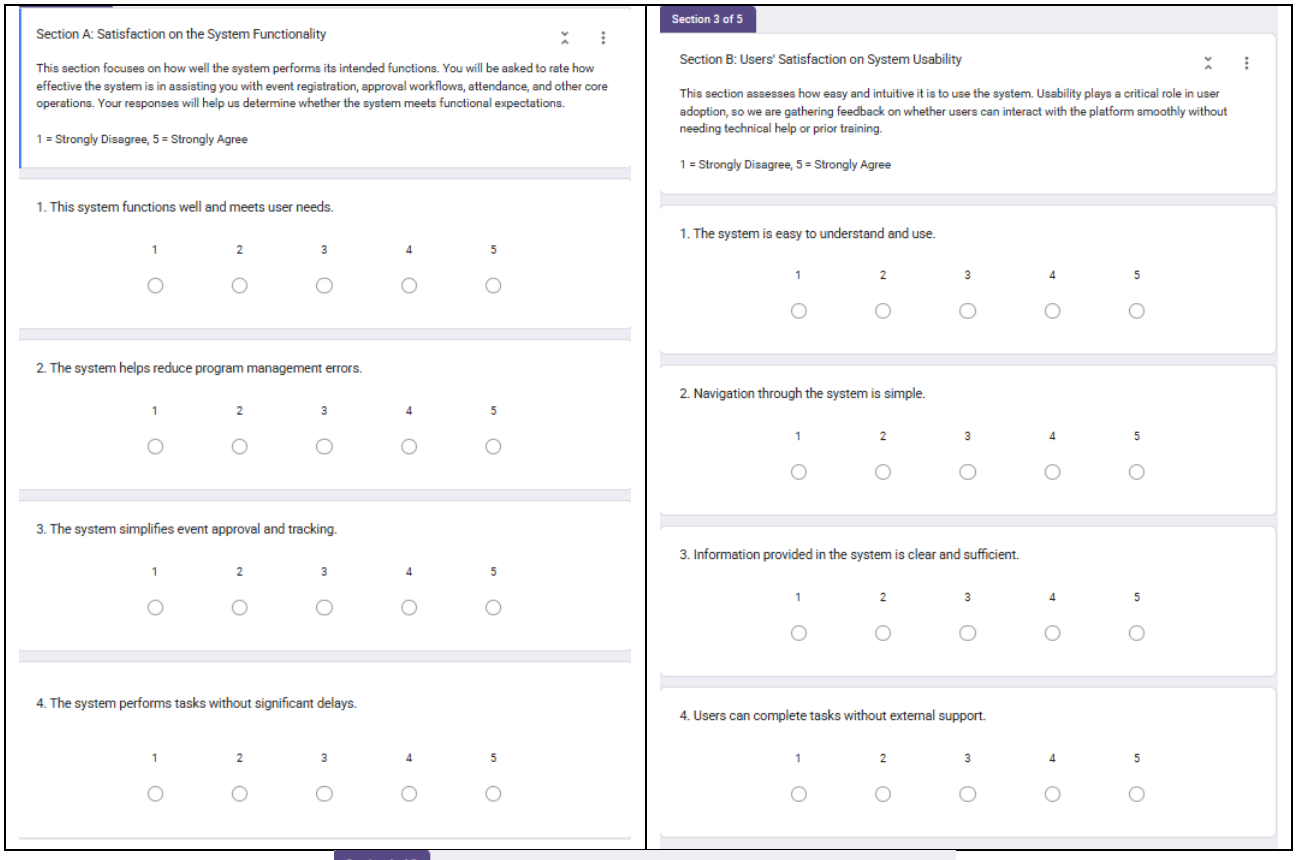



PSM 2 Gannt Chart

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 649-668

## APPENDIX B

User Feedback Questionnaire



Section A: Satisfaction on the System Functionality

This section focuses on how well the system performs its intended functions. You will be asked to rate how effective the system is in assisting you with event registration, approval workflows, attendance, and other core operations. Your responses will help us determine whether the system meets functional expectations.

1 = Strongly Disagree, 5 = Strongly Agree

1. This system functions well and meets user needs.

1 2 3 4 5

2. The system helps reduce program management errors.

1 2 3 4 5

3. The system simplifies event approval and tracking.

1 2 3 4 5

4. The system performs tasks without significant delays.

1 2 3 4 5

Section 3 of 5

Section B: Users' Satisfaction on System Usability

This section assesses how easy and intuitive it is to use the system. Usability plays a critical role in user adoption, so we are gathering feedback on whether users can interact with the platform smoothly without needing technical help or prior training.

1 = Strongly Disagree, 5 = Strongly Agree

1. The system is easy to understand and use.

1 2 3 4 5

2. Navigation through the system is simple.

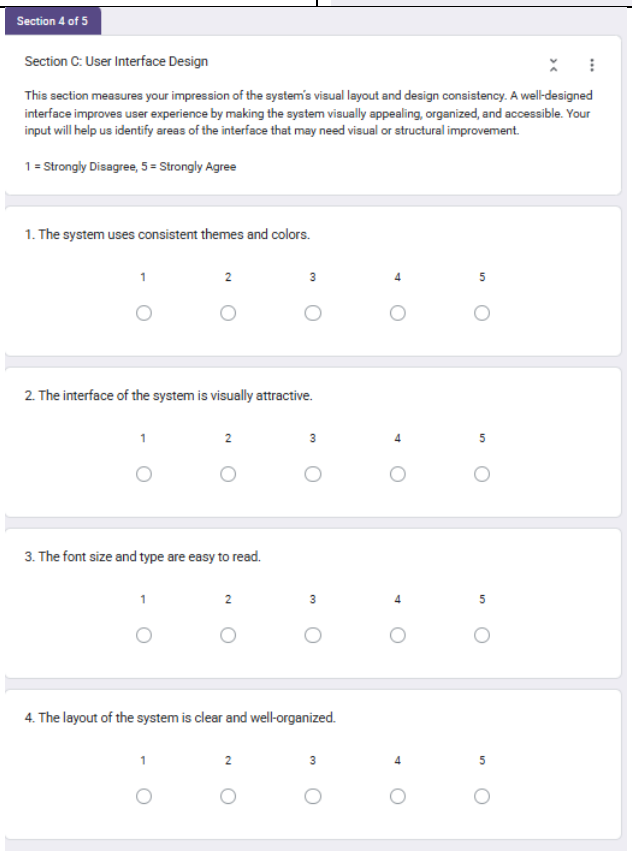
1 2 3 4 5

3. Information provided in the system is clear and sufficient.

1 2 3 4 5

4. Users can complete tasks without external support.

1 2 3 4 5



Section 4 of 5

Section C: User Interface Design

This section measures your impression of the system's visual layout and design consistency. A well-designed interface improves user experience by making the system visually appealing, organized, and accessible. Your input will help us identify areas of the interface that may need visual or structural improvement.

1 = Strongly Disagree, 5 = Strongly Agree

1. The system uses consistent themes and colors.

1 2 3 4 5

2. The interface of the system is visually attractive.

1 2 3 4 5

3. The font size and type are easy to read.

1 2 3 4 5

4. The layout of the system is clear and well-organized.

1 2 3 4 5

## APPENDIX C

**Section A – Background of MPP Member (Staff) UTHM**

| Question | Response |
|---|---|
| Q1. What are the primary activities organized by MPP Member (Staff) UTHM? | MPP Member (Staff) UTHM organizes student leadership programs, community engagement activities, academic support events, and cultural or sports initiatives across faculties. These programs are aimed at enhancing student development and university experience. |
| Q2. How are these activities currently managed? | The activities are managed manually using printed forms, email communications, spreadsheets, and in-person meetings for approvals and tracking. Program planning and execution heavily depend on verbal communication and paper-based documentation. |

**Section B – Current System**

| Question | Response |
|---|---|
| Q1. What are the main challenges in the current process? | The manual approach causes frequent delays, loss of information, difficulty in tracking activity status, and limited transparency. Errors during coordination and miscommunication are also common. |
| Q2. How is activity data currently stored and retrieved? | Data is mostly stored in spreadsheets or paper files and retrieved manually when needed. This makes it difficult to maintain an organized archive or share real-time updates with stakeholders. |

**Section C – Proposed System**

| Question | Response |
|---|---|
| Q1. What specific features would you like in the new system? | Key requested features include: program creation and approval workflow, QR code generation for attendance, centralized activity tracking, automated reports, and student feedback collection. |
| Q2. Are there any security requirements for user data? | Yes, access control based on user roles, secure login (authentication), and encrypted storage of sensitive information such as user credentials and activity feedback are essential. |