

The Development of UTHM Campus Event Management System using Laravel Framework

Ainul Aisyah Ainul Hisham¹, Nur Ariffin Mohd Zin^{1*},

¹ Faculty Computer Science and Information Technology

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

*Corresponding Author: ariffin@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.095>

Article Info

Received: 17 July 2025

Accepted: 20 November 2025

Available online: 30 November 2025

Keywords

Event Management, Campus,
System, Prototype

Abstract

Universiti Tun Hussein Onn Malaysia (UTHM) hosts over 70 clubs, yet the MyAktiviti portal is managed by Pejabat Hal Ehwal Pelajar lacks public events registration features and detailed club information. This limitation reduces student engagement and clubs' visibility, as essential data such as club objectives, description, contact details, and list of upcoming events are not readily accessible. Thus, the UTHM Campus Event Management System was developed to centralize stakeholder activities and improve efficiency across the university. Built using the Prototype Model, the system uses Laravel-PHP for the backend development, MySQL for database management, and HTML, CSS and JavaScript for a dynamic frontend. It was refined iteratively based on feedback from stakeholders. User-acceptance testing involving representative stakeholders reported an 84.8% satisfaction rate, indicating improved event planning, coordination and student engagement. Future development will focus on adding mobile access and analytics modules to personalize recommendations and support larger user loads.

1. Introduction

Universiti Tun Hussein Onn Malaysia (UTHM) is an institution that offers vibrancy in campus life as it organizes many programs and activities such as academic conferences, sports competitions, cultural festivals, student organization activities, and many others that make university life worth experiencing for the students [1]. The event on campus creates amazing opportunities for intellectual growth, career networking, and community-building among students. Through participation in these events, students will gain vital skills in organizing, leading and managing time, helping students be consistent by extending professional networks in response to future careers [2]. Activities and events of various kinds are proposed and implemented with the guidance and support of the student affairs office through the process for students and organizations. The office plays an important role in approving the paperwork for events to ensure a safe and structured environment for students.

It is a website page for organizers planning programs or events that require permission from HEP's administrative staff and Majlis Perwakilan Pelajar (MPP). Starting in May 2023, the organizer of every club and associations need to submit their paperwork by following the procedures and budget for the event. For events with a budget below RM5,000, paperwork is submitted to MPP. However, if the budget of the event is equal to or above RM5,000, the organizer needs to submit the paperwork to HEP's administrative staff. If the paperwork gets approval by HEP administrative staff or MPP, the organizer can continue with the program by creating a Google Form for the students to register for the event.

However, this system does not involve students where the organizer needs to do a registration form on another platform than the MyAktiviti system, such as Google Form. This shows that the current existing system is

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



inefficient in event planning and coordination. In addition, the organizer could not get feedback from students who participated in this event making the process less efficient. The students also need to visit the social media pages of each club or association, for further information which often leads to feel unmotivated to seek out event details and may end up not participating in events due to the hassle of searching through various social media platforms.

To solve this problem, the UTHM Campus Event Management System was developed to streamline the process, which makes it more efficient and effective for every target user. This system will improve access to event information that can boost student engagement and participation in campus activities. UTHM Campus Event Management System will provide feedback for the students who participate in the event, and the organizers will automatically generate report analytics to meet student expectations and better organize activities. This system will be a user-friendly design interface that can increase satisfaction and interaction among every user, which will give a positive campus experience and reduce misunderstandings.

This paper contains several sections. Section 2 is about related work that discusses developing system and study about others three existing system. Section 3 describes methodology that has been used to develop the system and system analysis. Section 4 explains about the result and discussion of the output creating the system and lastly, Section 5 is discussing about the conclusion of this project.

2. Literature Review

This section explains the method and technology used to develop the UTHM Campus Event Management System. To successfully develop the UTHM Campus Event Management System, it is essential to have the best method and technology to have a dynamic and better user-friendly interaction. The method and technology used to develop the system are Laravel Framework as a modern web application that uses PHP language, database management that stores the data safely and a web-based management system that uses modern web technologies to streamline and centralize event planning in the campus environment.

2.1 Laravel Framework

The Laravel framework is an advanced and popular software for developing systems using PHP-based web development. This is convenient for developers due to its better performance with Model-View-Controller (MVC) architecture. Using other software applications, such as Brackets, can be time-consuming when finding errors. On the back end of the system, the UTHM Campus Event Management System uses Laravel framework with PHP-based and a structured approach that allows integration with the database seamlessly. It also gives a smooth communication that can manage and process event information effectively. Using Laravel as the framework of the system, it secures the all the data by adding input validation and password hashing to protect the data from unauthorized access [3].

2.2 Database Management System

UTHM Campus Event Management System uses a relational database provided to keep data safe and organized which is called MySQL. MySQL offers structured data storage, facilitating efficient data retrieval and analysis for event information, user profiles, calendars, and feedback. It provides better security by encrypting data such as passwords, and only admins or authorized users can modify the data (4). The database also reduces data duplication, avoiding confusion caused by similar records and saving storage space that could be taken up by redundant data. Furthermore, the database ensures integrity and consistency which enables an up-to-date and accurate information that is essential for developing a successful project.

2.3 Web-Based Management System

The web-based management system approach offers adoption due to the ability to meet the dynamic needs of event coordination on a university campus [5]. By using this method, UTHM's Campus Event Management system can allow users to access system flexibility and convenience. It collects and saves data more quickly can improve the communication between all the target users and reduce delays and misunderstanding [6]. Using a web-based management system, UTHM will have an event management platform that is adaptable to growth and sustainable for campus needs. This method can be a powerful way to increase the volume and complexity of campus events while streamlined experience for all stakeholders. Using web-based supports all different role of user by customized the permission in database and it will provide a real time updates about system that been developed.

2.4 Study of Existing Related System

2.4.1 HEP UiTM Selangor

HEP UiTM Selangor is a system website for UiTM students at the Selangor branch who want to apply for program activities. The source link of the system is at [7]. This system helps provide guidance in terms of the steps to easily make an activity or event application for the organizer such as provides the format of official letter of application for paperwork, application to hold student activities, and speaker biodata format [8]. The documents need to submit completely through the Google Form link that has been provided in the system no later than 30 days for programs or activities in the country or 90 days for programs or activities abroad before the program date.

2.4.2 Eventzilla

Eventzilla was founded in 2008, is an event platform created registration and event marketing with thousands of users worldwide. The official website can be accessed at [9] and this system aims to be a platform that easily and seamlessly sets up a custom event website, manages event registrations, markets the events, collects payments, and publishes the event publicly [10]. This website is available 24 hours per day and has used its own specialties with many companies such as media, education, information technology, manufacturing, and government. Eventzilla enhance experience for both organizer and participant experiences by offering feedback features that make event management more systematic and accessible and responsive to participants' opinions and event quality.

2.4.3 Eventbrite

Eventbrite is a website create in 2008 and allow users to create and manage an event by provide an easy-to-use process which makes it popular for individuals and organization [11]. The website can be accessed at [12] and offering features such as AI-assisted event creation, registration forms, payment setup, and analytics.

Table 1 summarizes the features of three event management systems and the UTHM Campus Event Management System. This comparison will show the advantages and disadvantages of every system and the UTHM Campus Event Management System has the most unique features for a system that can be used in the campus environment. The three existing related system that have been chosen are HEP UiTM Selangor, Eventzilla and Eventbrite. Based on the review, the comparison is shown in Table 1.

Table 1 Comparison of three existing system and UTHM Campus Event Management System

Features/ System	HEP UiTM Selangor	Eventzilla	Eventbrite	UTHM Campus Event Management System
Register and Log In	Admin staff only	Organizer	Organizer & Participant	All target users
Event Creation Management	Manual by involves admin staff and organizer	Simple event setup with custom forms	Quick setup with AI- assisted options and custom forms	Centralized event management for all target user
Target User	Students, staff	Public, organizations	Public, organizations	Students, staff, student representative, organizer
Feedback	-	Auto generated for organizer	Auto generated for organizer	Support giving & receiving feedback
Event Analytic and Report	-	Basic attendance and revenue	Advanced engagement analytics	Generate report on club association and events
Event Reminder	-	Basic email reminder	Customizable email reminders	Auto email reminders for registered students
Event Calendar	List event on main page	Calendar for upcoming event	Calendar for organizer that created events	Centralized calendar displays upcoming events for all users
Role-based Access Control	Manual control	Basic control	Limited for organizers	Advanced role-based access for all users

Event Payment	Support multiple gateways	Integrated payment processors	Support card, PayPal and GooglePay	Support card and GooglePay
---------------	---------------------------	-------------------------------	------------------------------------	----------------------------

Based on the comparison shown in Table 1, each system has its strengths and weaknesses. However, UTHM’s Campus Event Management System stands out with the most strengths among other systems and builds on suitable and specialized features for the university environment. This system can enhance event planning, coordination, and engagement and improve the overall campus experience for all the target users who may be involved in event management at UTHM. Organizers of clubs and associations, HEP’s administrative staff and Majlis Perwakilan Pelajar (MPP) can view and generate event reports such as list of total events organized, feedback details, the total number of clubs and associations and lists of events held by each club and association. This system also has event calendar features for all users to prevent scheduling conflicts. The student will receive a feedback form that can be a place for them to share their opinion and thoughts on the event they attended. Students also will receive automatically generated email reminders of the event they have registered for. This feature can help students manage their time well and reduce the chance of missing events. In general, UTHM’s Campus Event Management System will be a platform that brings all users and manages their event process in one platform. This system will make it easier to organize campus events and improve the overall event experience at UTHM.

3. Methodology

In this project, a Prototyping Model was used for the software development life cycle which is suitable for developing a system or website by testing functionalities and modifying it until approved by the target users. The prototype is chosen for the system because its flexibility can be changed and improved user satisfaction which ensuring it meets user needs. The prototype model can reduce risks allows for early testing and feedback from target users throughout the development process, which can help avoid repeated mistakes. By using this model, the prototype provides a preview of the system to the target user that makes it easier to ensure meets user expectations and understand the system better [13].

Fig. 1 shows the Prototype Model that has six phases, which start with requirement gathering, design, prototype, customer evaluation, refined prototype, and final product.

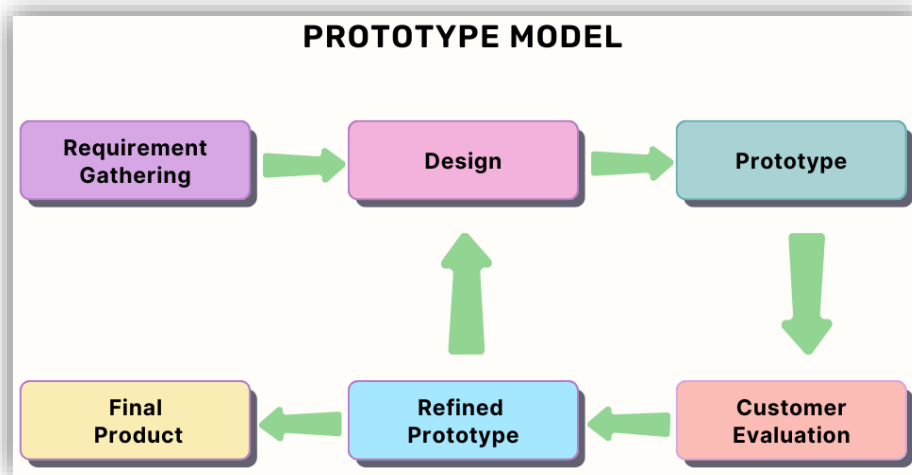


Fig 1 Prototyping Model

3.1 Methodology and System Development Workflow

There is a total of six phases from the prototype model. As shown in Table 2, each phase has its own assignment and output that need to be produced during the entire project development. Besides that, the output had been completed within the specific days that had been given.

Table 2 Software development activities and their task

Phase	Task	Output
Requirement Gathering and Analysis	<input type="checkbox"/> Proposed the project	<input type="checkbox"/> Project proposal
	<input type="checkbox"/> Determine the project schedule, activities and output	<input type="checkbox"/> Gantt chart
	<input type="checkbox"/> Analysis the existing system	<input type="checkbox"/> Interview with stakeholder
Design	<input type="checkbox"/> Identify stakeholder	
	<input type="checkbox"/> Collect Information	
	<input type="checkbox"/> Collect requirements of each module	<input type="checkbox"/> System architecture
Prototype	<input type="checkbox"/> Design user interface	<input type="checkbox"/> Database schema and data dictionary
		<input type="checkbox"/> User interface design
		<input type="checkbox"/> Flowchart
		<input type="checkbox"/> ERD and DFD
Customer Evaluation	<input type="checkbox"/> Build prototype of system with features	<input type="checkbox"/> Prototype
	<input type="checkbox"/> Analyze the feedback from stakeholders	<input type="checkbox"/> Changes and improve the prototype
Refined Prototype	<input type="checkbox"/> Re-test prototype with the stakeholders	
	<input type="checkbox"/> Test the system	<input type="checkbox"/> Updated the prototype
	<input type="checkbox"/> Identify the issues	<input type="checkbox"/> Adding missing features
Final Product	<input type="checkbox"/> Conduct more testing	
	<input type="checkbox"/> Final testing all functionalities in the system	<input type="checkbox"/> Publish the system
	<input type="checkbox"/> Make improvements to the system	

3.2 System Requirements Functional and Non-Functional Requirements Table

Functional requirements describe the features of the system and what tasks the system must perform to meet user needs in the system [14]. These functional requirements cover essential features to make the system robust, user-friendly, and effective for managing campus events. Table 3 explains the functional requirements of the UTHM Campus Event Management System.

Table 3 Functional requirements

No	Requirement	Description
1	User Management Module	<ul style="list-style-type: none"> • New users such as students, can create accounts with a unique username and password. • All user roles can log in to access their account. • Users can view and update their profiles. • Have different levels of access for administrators, organizers, and students.
2	Event Calendar Module	<ul style="list-style-type: none"> • Display a calendar of all approved events by administrators for all users
3	Event Creation Management Module	<ul style="list-style-type: none"> • Organizers can submit event application with details like name, description, date, time, and location • HEP’s administrative staff and Majlis Perwakilan Pelajar (MPP) can view, add comments and download paperwork of requested events and change the event status. • The system allow organizer to track the event status whether is pending, approved, or rejected. • The system allows students to view a list of clubs and associations with details and register for events that interest them.

Table 3 *Continued*

No	Requirement	Description
4	Feedback Module	<ul style="list-style-type: none"> The system should allow students to submit feedback and give overall rating for the events based on their experience. The system should allow organizers access to feedback reports to analyze event performance and areas for improvement
5	Payment Module	<ul style="list-style-type: none"> Students can make online payments for events directly through the system. Organizers can view the payment status
6	Report and Analytics Module	<ul style="list-style-type: none"> Display statistics on attendee numbers and demographics. Generate insights from feedback forms to measure event success. Analyze trends in event popularity and participation over time.
7	Notification and Reminder Module	<ul style="list-style-type: none"> Students receive reminders via email for upcoming events they have registered for

Non-functional requirements describe what the system will serve and specify how a system should perform and its quality attributes [15]. Table 4 explains the non-functional requirements of the developed system.

Table 4 *Non-functional requirements*

No	Requirement	Description
1	Performance	The system should be always usable and response time for all user requests should be within 2 seconds under normal load conditions
2	Operational	The loading time required for a website is no more than 1 minute
3	Scalability	The system should be scalable to accommodate future increases in users and events without requiring significant redesign
4	Availability	The system must ensure to support student and organizer access and minimize downtime in case of server failure
5	Security	The system should be user friendly, and all data must be encrypted and stored securely
6	Compatibility	The system should be able to work on any web browser

3.3 System Analysis

The results of the system analysis, including the data flow diagram and the entity relationship diagram, are outlined in this section. A systematic approach is used to perform system analysis

3.3.1 Context Diagram

Fig. 2 shows about Context diagram that present the overview of the interaction between the system and the users, which are HEP's Administrative Staff, Majlis Perwakilan Pelajar, Organiser's Club and Association and Student. Context diagrams also show the input and output to and from the user and the system.

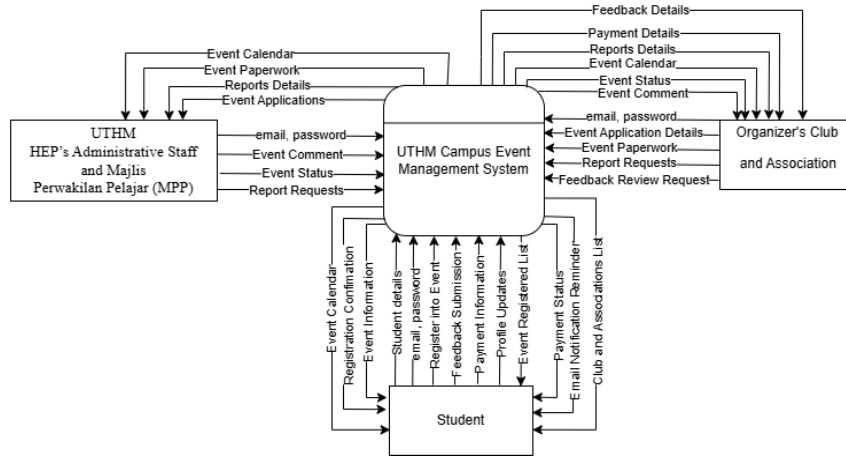


Fig 2 Context Diagram

3.3.2 Data Flow Diagram Level 1

Data Flow Diagram (DFD) is a graphical representation of the flow of data or input from an entity through a process, which then generates output either to another entity or stored in data storage. Fig. 3 shows each input and output of DFD level 1 for each entity and process.

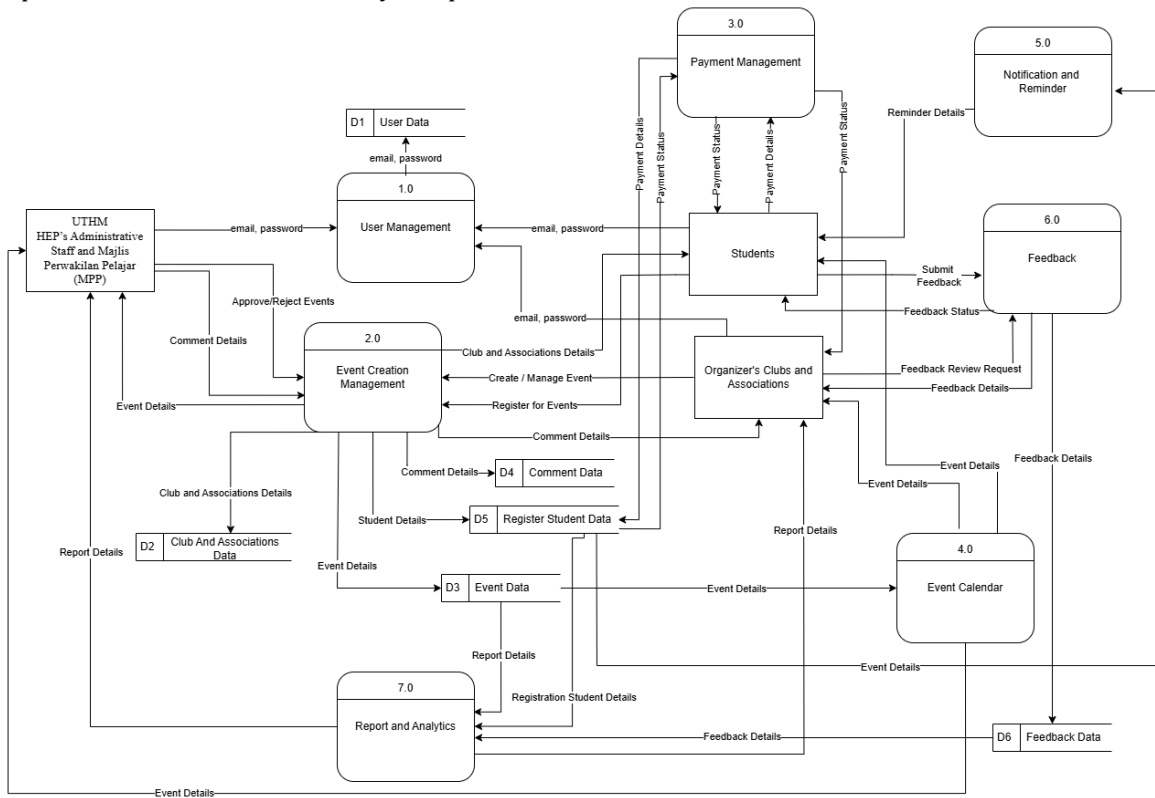


Fig. 3 Data Flow Diagram level 1

3.3.3 Entity Relationship Diagram

Fig. 4 shows the Entity Relationship Diagram (ERD) which visually represents how different entities relate to each other in a system. ERDs are commonly used in database design to model the structure and ensure data integrity.

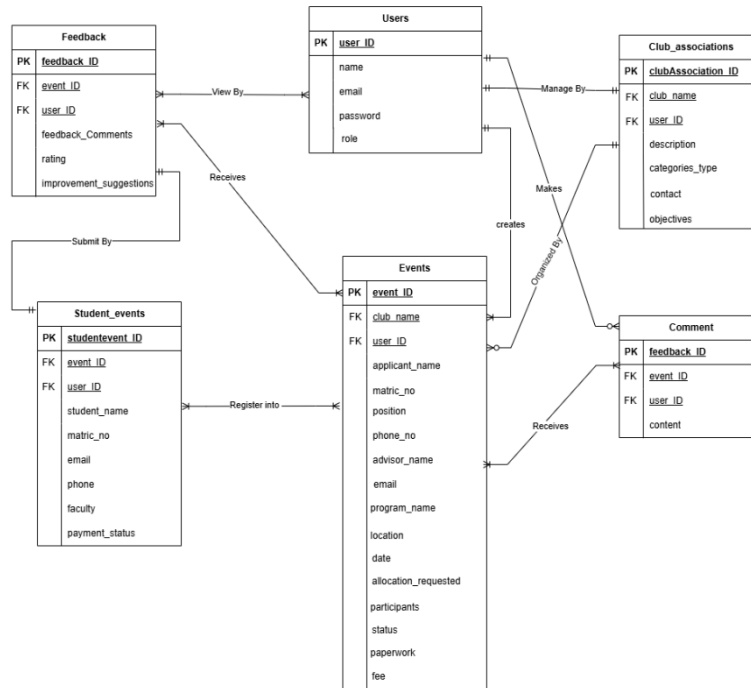


Fig. 4 Entity Relationship Diagram

3.3.4 Flowchart

A flowchart is a diagram that shows the steps in a process. The flowchart diagram process of every user which is HEP’s Administrative Staff, Majlis Perwakilan Pelajar, Organizers’ Club and Association and Student is shown in Appendix A.

3.4 System Design

After all the users’ requirements were successfully analyse, the project will proceed in design phase. In this phase, both interface and database had been designed to help visualize the system before proceed with the coding the system. This section includes of System Architecture, Relational Schema and User Interface Design.

3.4.1 System Architecture

Fig. 5 shows the system architecture of UTHM Campus Event Management System. The diagram shows that all users, which are HEP Administrative Staff, Majlis Perwakilan Pelajar, Organizer Club and Association, and Student, are connected to the UTHM Campus Event Management System. The backend Server connects to Database, Payment Gateway and Notification Server. The database is for storing data detailed about the event and users. Payment Gateway is for transaction processing, and the Notification Server is for students who will receive notifications to alert them via email.

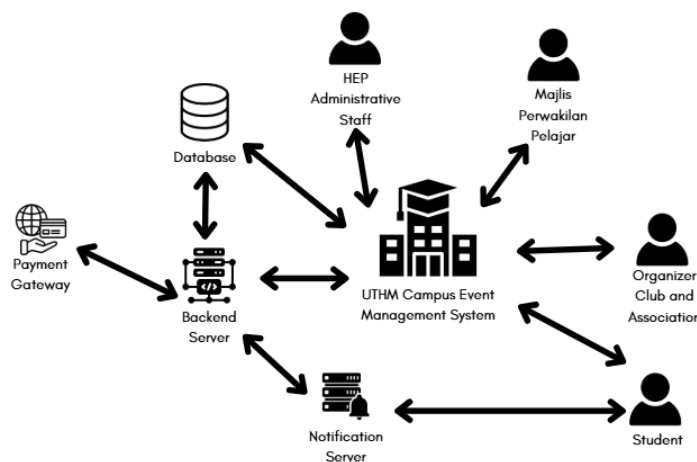


Fig. 5 System Architecture

3.4.2 Relational Schema

The following are the tables from the database that have been designed and extracted from the entity relationship diagram. They were created using MySQL Server 2014.

- i. User (user_ID, name, email, password, role)
- ii. Event (event_ID, applicant_name, matric_no, position, phone_no, advisor_name, email, program_name, club_name, user_ID, location, date, allocation_requested, participants, status, paperwork, fee)
- iii. Club and Association (clubAssociation_ID, club_name, description, categories_type, contact, objectives event_ID, user_ID)
- iv. Feedback (feedback_ID, event_ID, user_ID, feedback_Comments, improvement_suggestions rating)
- v. Comments (comment_ID, event_ID, user_ID, content)
- vi. Student_events (studentE_id, user_ID, event_ID, student_name, matric_no, email, phone, faculty, payment_status)

3.4.3 User Interface Design

Fig. 6 shows the interface for all target users logging into the system and a list of the upcoming events and programs that will be held. Fig. 7 shows the list of clubs and associations under UTHM and feedback form for students to give compliments and suggest improvements for the next events.

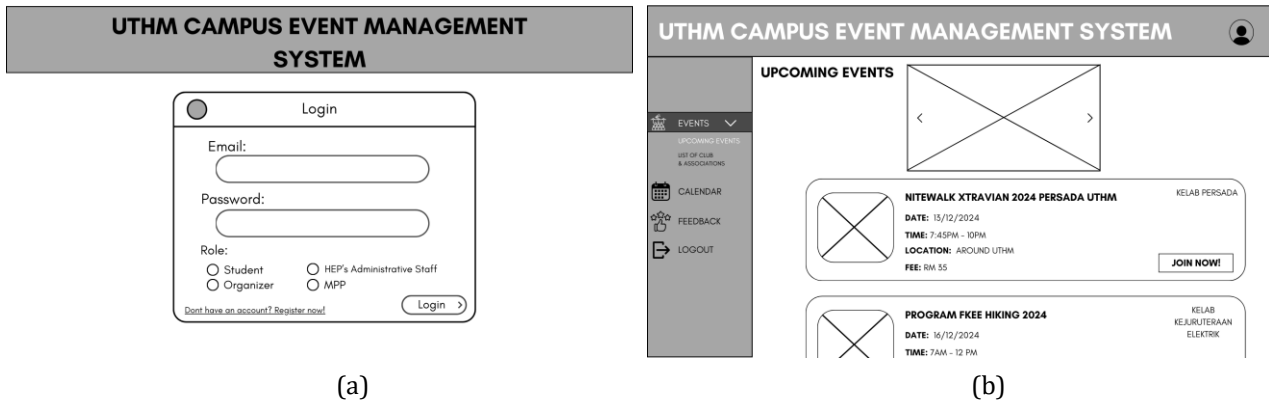


Fig. 6 (a) The Login Page and (b) Upcoming Events Page for students

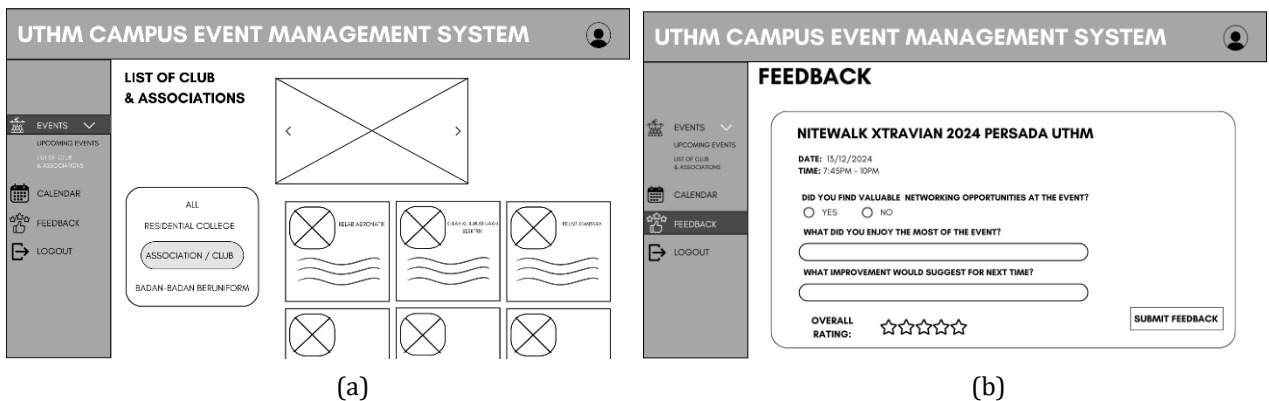


Fig. 7 (a) List of Club and Association Page and (b) Feedback Form for students.

4. Result & Discussion

Fig. 8 shows the login interface where users enter their email and password to access the system. Fig. 9 shows the registration page, which is only available to students, as all other users are pre-registered to ensure security and accurate participation tracking.

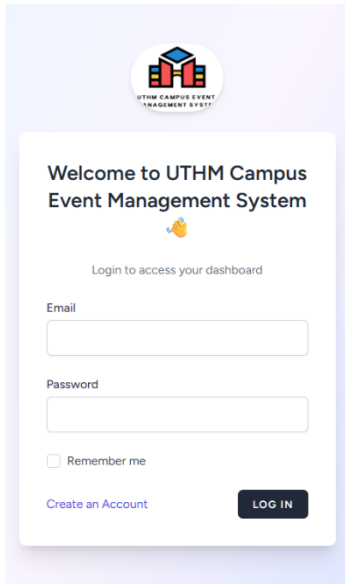


Fig. 8 Login Interface of the system for all users

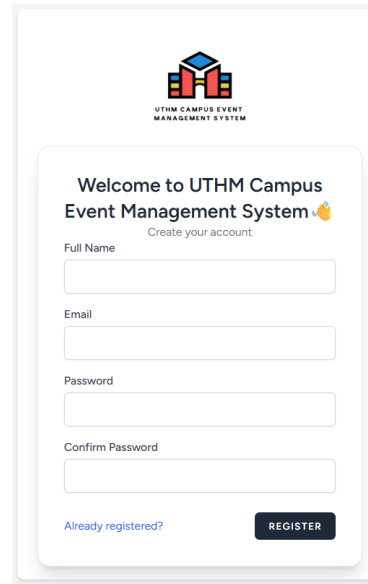
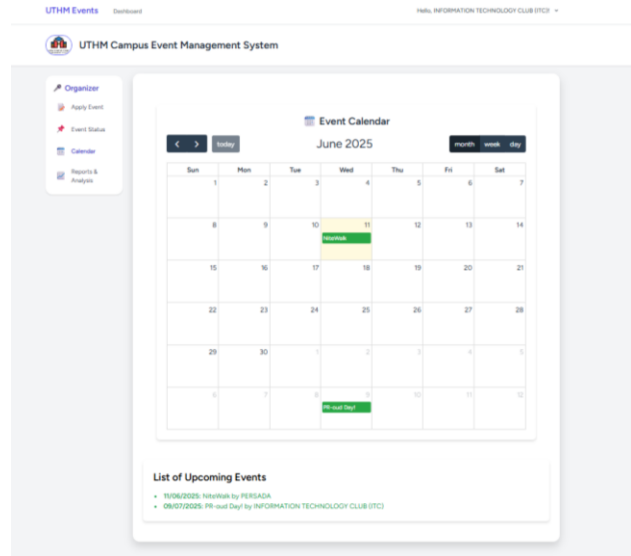


Fig. 9 Account Registration Interface for Student

Fig. 10 shows the code and interface of event calendar page that is accessible by all users and provides a comprehensive calendar view of upcoming and past events that have been organised by various clubs and associations and approved by HEP’s administrative staff and Majlis Perwakilan Pelajar (MPP)

```
public function eventCalendar()
{
    // Fetch events from the database (replace this with your actual logic)
    $events = Event::where('status', 'approved')->get();

    // Return the view with the events data
    return view('events.event_calendar', ['events' => $events]);
}
```



(a) (b)

Fig. 10 (a) eventCalendar function in EventController and (b) Event Calendar interface for all user

Fig. 11 shows the store function in the Event Controller and Event Application Form interface that needs be submitted to Majlis Perwakilan Pelajar (MPP) and HEP’s administrative staff depending on the event’s budget allocation.

```

// Handle form submission and store event in the database
public function store(Request $request)
{
    $request->validate([
        'applicant_name' => 'required|string|max:255',
        'matric_no' => 'required|string|max:50',
        'position' => 'required|string|max:255',
        'phone_no' => 'required|string|max:15',
        'fee' => 'required|numeric|min:5',
        'club_name' => 'required|string|max:255',
        'advisor_name' => 'required|string|max:255',
        'email' => 'required|email',
        'program_name' => 'required|string|max:255',
        'location' => 'required|string|max:255',
        'date' => 'required|date|after:today',
        'allocation_requested' => 'required|integer|min:0',
        'participants' => 'required|integer|min:1',
        'paperwork' => 'required|file|mimes:pdf|max:2048', // Ensure it's a PDF
    ]);
    $userId = Auth::id();
    if (!$userId) {
        return redirect()->route('login')->with('error', 'You must log in to apply for an event.');
```

(a)

(b)

Fig. 11 (a) store function in Event Controller and (b) Event Application Form Interface

Fig. 12 displays display the updateStatus function in Event Controller and the “Event Requested” page in Majlis Perwakilan Pelajar and HEP’s administrative staff interface.

```

public function updateStatus(Request $request, Event $event)
{
    // Validate the status
    $request->validate([
        'status' => 'required|in:pending,approved,rejected',
    ]);

    // Ensure there is at least one comment before allowing a status change
    if ($event->comments->count() == 0) {
        return redirect()->back()->with('error', 'Please add a comment before updating the status.');
```

(a)

(b)

Fig. 12 (a) updateStatus function in Event Controller and (b) Event Requested List Interface.

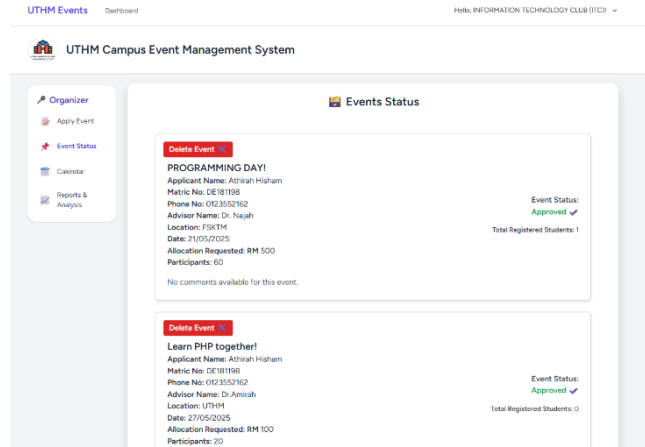
Fig. 13 shows the code of the blade view handle the “Event Status” page for organisers to display a list of events they have submitted along with each event’s current status and given comments by Majlis Perwakilan Pelajar and HEP’s administrative staff.

```

<div class="bg-white shadow-md rounded-lg mb-6 p-6 relative border border-gray-300">
<div style="float: right; text-align: right; font-size: 0.8em; font-weight: bold; color: #6c757d; margin-bottom: 10px;">
<span style="float: right; font-weight: normal; color: #6c757d;">Delete Event X
</div>
</div>
<div class="font-weight-normal text-gray-700">
<strong style="font-weight: normal; color: #6c757d;">Event Status
</div>
<div class="font-weight-normal text-gray-700">
<strong style="font-weight: normal; color: #6c757d;">Event Status: <span style="font-weight: normal; color: #6c757d;">Approved <span style="font-weight: normal; color: #6c757d;">✔Total Registered Students: <span style="font-weight: normal; color: #6c757d;">1Event Status: <span style="font-weight: normal; color: #6c757d;">Approved <span style="font-weight: normal; color: #6c757d;">✔Total Registered Students: <span style="font-weight: normal; color: #6c757d;">0

```

(a)



(b)

Fig. 13 (a) Event status source code and (b) Event Status Interface.

Fig. 14 displays the `upcomingEvents` function in EventController and Upcoming Events page in student's interface. This page displays list of upcoming events, including those the student has already registered for.

```

public function upcomingEvents(Request $request)
{
    // Get upcoming events
    $events = Event::where('status', 'approved')
        ->whereDate('date', '>', now())
        ->orderBy('date', 'asc')
        ->get();

    $registeredEvents = collect(); // Empty collection by default

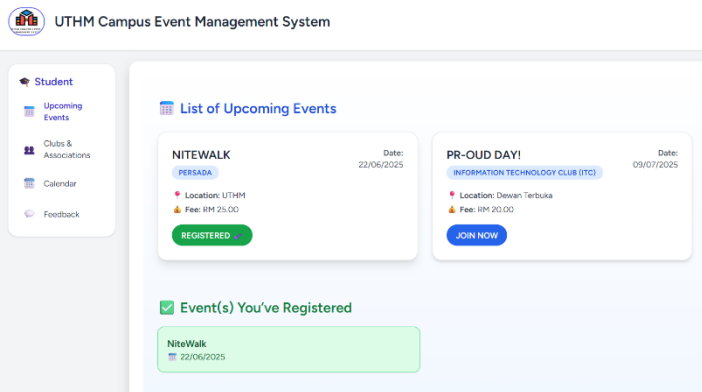
    if (auth()->check()) {
        $email = auth()->user()->email;
        $registeredEvents = \App\Models\StudentEvent::with('event')
            ->where('email', $email)
            ->get();
    }

    // Check if matric_no is stored in session (after registration)
    elseif ($request->session()->has('registered_matric_no')) {
        $registeredEvents = \App\Models\StudentEvent::with('event')
            ->where('matric_no', $request->session()->get('registered_matric_no'))
            ->get();
    }

    return view('events.upcomingEvents', compact('events', 'registeredEvents'));
}

```

(a)



(b)

Fig. 14 (a) `upcomingEvents` function in Event Controller and (b) Upcoming Events Interface.

Fig. 15 shows the `register` function in Event Controller and the registration event form for students who want to register for the selected event by filling in the registration form with their personal information.

```

public function register($id)
{
    $event = Event::findOrFail($id);

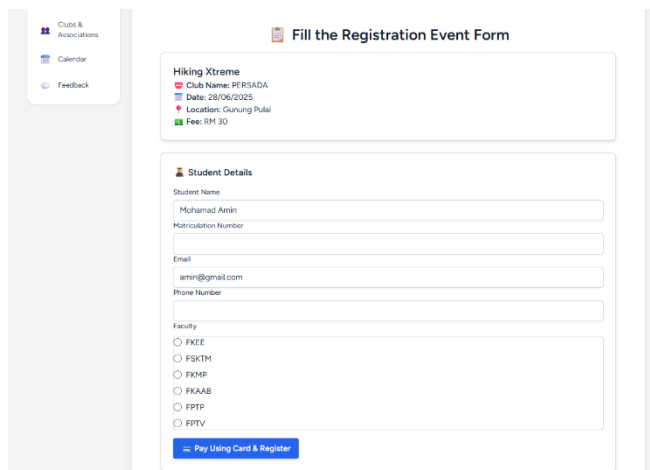
    // Get student identity (from authenticated user or session)
    $user = Auth::user();
    $email = $user->email;
    $matricNo = session('registered_matric_no');

    // Check if student has already registered for this event
    $alreadyRegistered = false;

    if ($email) {
        $alreadyRegistered = StudentEvent::where('event_id', $id)
            ->where('email', $email)
            ->exists();
    } elseif ($matricNo) {
        $alreadyRegistered = StudentEvent::where('event_id', $id)
            ->where('matric_no', $matricNo)
            ->exists();
    }

    if ($alreadyRegistered) {
        return redirect()->route('events.upcoming')->with('success', 'You have already registered for this event.');
```

(a)



(b)

Fig. 15 (a) `register` function in Event Controller and (b) Event Registration Form Interface.

Fig. 16 shows checkout function and Payment Checkout Interface that manages event registration payment by using Stripe. Student need to fill out the form with their email, card information and cardholder name.

```
public function checkout(Request $request)
{
    Stripe::setApiKey(env('STRIPE_SECRET'));

    $eventId = $request->input('event_id');
    $amount = $request->input('amount'); // in cents

    // Save the student details in session to retrieve after payment
    session(['registration_data' => $request->only(['student_name', 'matrix_no', 'email', 'phone', 'faculty', 'event_id'])]);

    try {
        $session = Session::create([
            'payment_method_types' => ['card'],
            'line_items' => [
                [
                    'price_data' => [
                        'currency' => 'myr',
                        'product_data' => [
                            'name' => 'Event Fee for Event ID ' . $eventId,
                        ],
                        'unit_amount' => $amount,
                    ],
                    'quantity' => 1,
                ],
            ],
            'mode' => 'payment',
            'success_url' => route('payment.success'),
            'cancel_url' => route('payment.cancel'),
        ]);

        return redirect($session->url, 303);
    } catch (\Exception $e) {
        return back()->withErrors(['error' => 'Unable to create payment session: ' . $e->getMessage()]);
    }
}
```

(a)

(b)

Fig.16 (a) checkout function in Payment Controller and (b) Payment Checkout Interface.

Fig. 17 shows the EventNotification class in Mailable and Email notification interface that sends a confirmation email to student after successfully making a payment in event registration.

```
class EventNotification extends Mailable
{
    use Queueable, SerializesModels;

    public $event;
    public $studentName;

    public function __construct(Event $event, $studentName)
    {
        $this->event = $event;
        $this->studentName = $studentName;
    }

    public function build()
    {
        return $this->subject('Event Registration Confirmation: ' . $this->event->program_name)
            ->view('emails.event-notification')
            ->with([
                'event' => $this->event,
                'studentName' => $this->studentName,
            ]);
    }
}
```

(a)

(b)

Fig. 17 (a) EventNotification function in Mailable and (b) Email Notification Interface in Gmail.

Fig. 18 display the list of Clubs and Associations Interface for student access more details about a specific club by clicking “Read More” button for further information and Feedback Form Interface for student give their comments, suggestion improvement for future event and manage feedback submission.

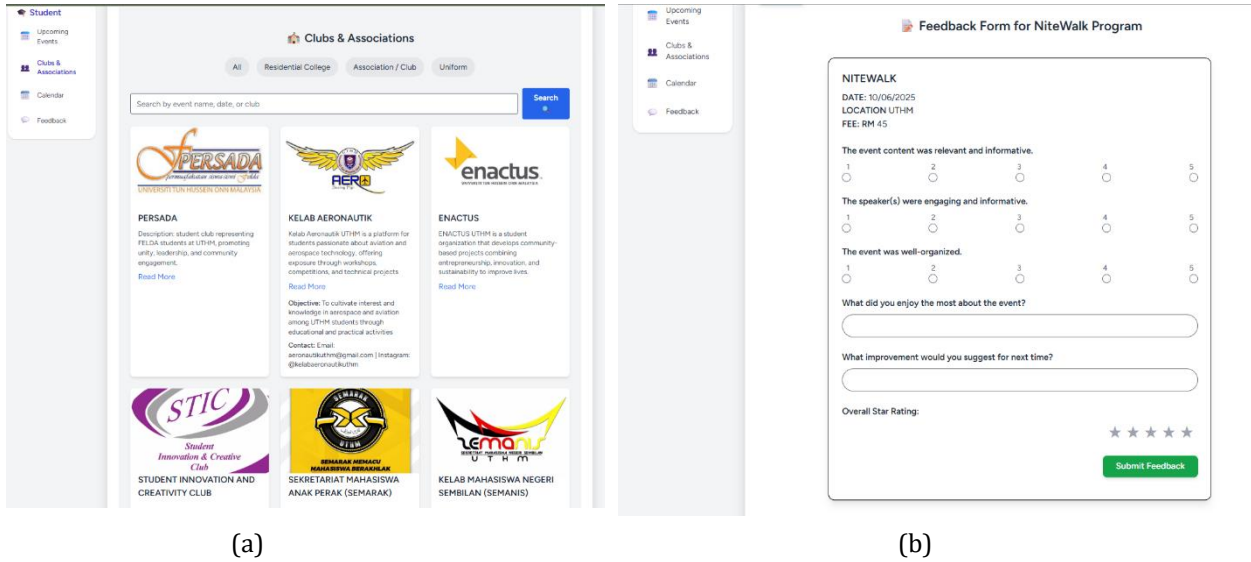


Fig. 18 (a) Club and Associations Interface and (b) Feedback Form Interface.

Report and Analytics modules in this system provides three types of reporting function which are a reporting of total number of events organized by individual clubs which only accessible for organizers, overview of total events and registered clubs, and a list of all events conducted by all clubs and only accessible to MPP and HEP’s administrative staff. Fig. 19 shows main interface of Report and Analytics page, as well as View Report Page which displays detailed reports of specific events in organizer’s interface. This page includes selected event data, such as the payment list, participant feedback, bar chart and pie chart of the total overall feedback rating of the event and registrations of students by faculty.

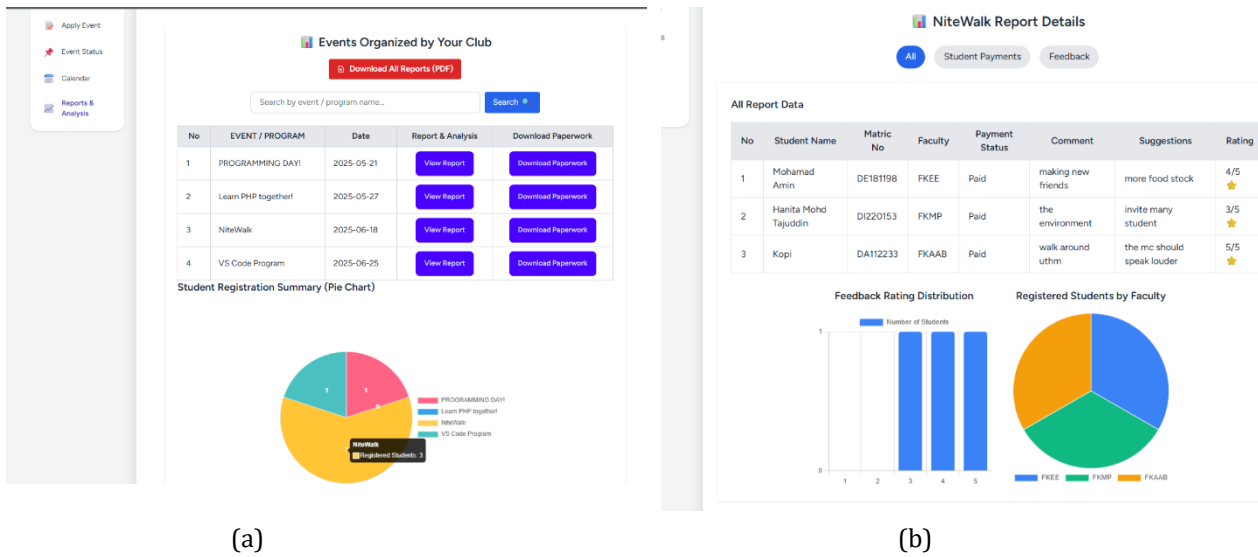


Fig 19 (a) Main interface of Report and Analytics Page for Organizer (b) and View Report Page that display Report Details on specific events.

4.1 Test Case

The system requirements that involved several test cases which are listed in Table 5. The test cases include every module such as User Management, Event Calendar, Event Creation Management, Payment, Notification and Reminder, Feedback and Report and Analytics.

Table 5 Test Case for User Management Module

Test Case ID	Description	Actual Result	Pass/Fail
User Management			
M1-1	All users can login into the system	All users have successfully logged into the system	Pass
M1-2	Students can register for an account	Student successfully created for an account	Pass
M1-3	The system will display error message when login with invalid credentials	The system restricted login when enter invalid credentials and display error message "These credentials do not match our records"	Pass
M1-4	The system prevents registration using an existing email.	The system blocks the registration and display error message "The email has already been taken"	Pass
Event Calendar			
M2-1	The system displays the event calendar on every user's page	Users can view approved events on the calendar in monthly, weekly, and daily views.	Pass
M2-2	Users can navigate to previous and next months	The calendar successfully displays the previous and next month	Pass
Event Creation Management			
M3-1	Organizer can submit event based on specific budget and admin	Organizer successfully submitted event based on the total allocation requested	Pass
M3-2	The event status is updated after being modified by MPP or HEP's administrative staff	The system successfully displays the updated event status to organizer's interface	Pass
M3-3	MPP or HEP's administrative staff can submit comments.	The comments are successfully saved and displayed under the event	Pass
M3-4	Approved events are shown on the calendar.	Event calendar successfully displays approved events according to scheduled dates.	Pass
M3-5	Approved events appear on the Upcoming Events page in student's interface	The Upcoming Events page displays approved events, and students can register by filling out the form.	Pass
M3-6	The system successfully displays the list of clubs and associations	The system displays all the active clubs and associations in UTHM including their description, objective and contact details of every clubs.	Pass
Payment			
M4-1	Students can successfully submit event registration form by making a payment	Students can fill in the registration form and redirected to Stripe checkout to make payment and return with a success message	Pass
M4-2	The payment process is completed successfully	The student that made payment for event registration has successfully stored in database and receive email confirmation	Pass
Notification and Reminder			
M5-1	Students receive email notifications after registering and making payment	Students receive email confirmation about event details after made payment	Pass
M5-2	Email has the correct content	Students successfully receive email confirmation with the correct content of the registered event	Pass

Table 5 *Continued*

Test Case ID	Description	Actual Result	Pass/Fail
M5-3	System handles scenarios where email cannot be delivered to temporary inactive email addresses.	the system received an email notification indicating a temporary delivery problem with a recipient's email	Pass
Feedback			
M6-1	Student can write feedback	Student successfully write, submit the feedback, give overall rating and store the data in database.	Pass
M6-2	Submit the feedback form without giving overall rating	Student cannot submit the feedback and store the feedback data in database	Pass
M6-3	Student can view the overall of submitted feedback	Student can view the submitted feedback on past event that been registered and display with star ratings	Pass
Report and Analytics			
M7-1	Organizer can view report summary	Organizer can view the report summary and details of every event by displaying pie chart of total registrations student by events, view student payment status and feedback.	Pass
M7-2	Organizer can download the summary report and paperwork in PDF format.	Organizer successfully download report summary include a pie chart and paperwork of specific event in PDF format.	Pass
M7-3	The total events and club counts is accurate in Event Total page.	The system displays counters of total events that has been made and total clubs and associations, residential college, and uniform with a pie chart of total events by every club.	Pass
M7-4	User can view all events details and status	MPP and HEP's administrative staff can view the total of events that been made in details and pie chart of total approved and rejected events.	Pass
M7-5	The search bar and filtering button is functionable	User can use the search function to search club or event name and filtering the event based on the club type.	Pass

4.2 User Acceptance Testing

This system conducted user testing feedback using Google Forms to collect feedback from students. A total of 30 students participated and the feedback form along with the response results is shown in **Appendix B**.

5. Conclusion

In conclusion, UTHM Campus Event Management System has successfully achieved the objectives and give many benefits such as centralized event data management and improved communication between student and organizers. This system is also has been tested by one of HEP's administrative staff, club's organizer and student and get a positive impact. With this system, it helps to simplify the event application registration and approval process, reduce manual paperwork and encourages student participation by making it easier to register into the event. Overall, UTHM Campus Event Management System enhances efficiency, promotes transparency and collaboration within the university community.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for their support and providing the opportunity to develop this project.

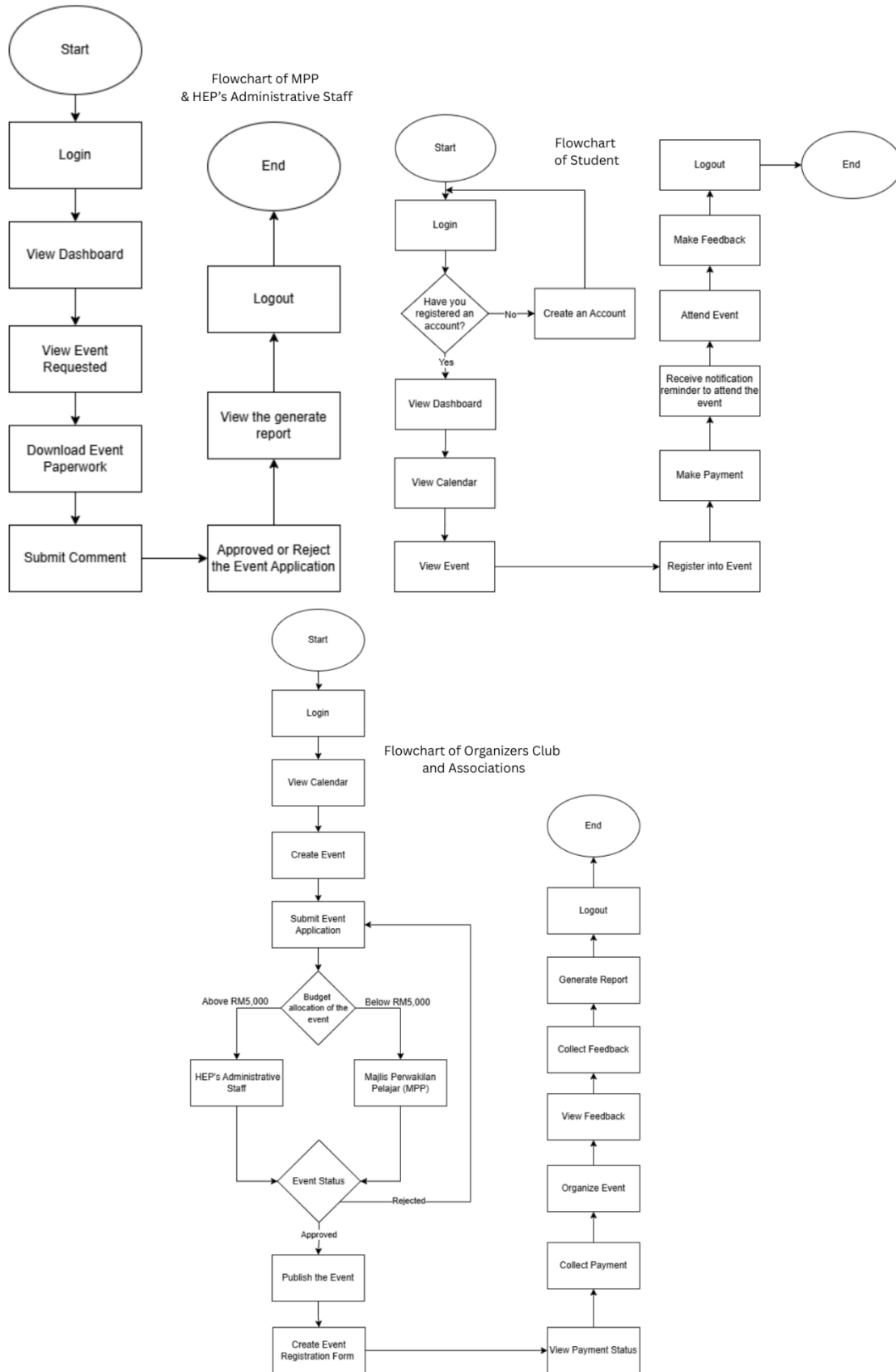
Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Ainul Aisyah Ainul Hisham, Nur Ariffin Mohd Zin; **data collection:** Ainul Aisyah Ainul Hisham, Nur Ariffin Mohd Zin; **analysis and interpretation of results:** Ainul Aisyah Ainul Hisham, Nur Ariffin Mohd Zin;

References

- [1] Kamaruddin, N. K. M., Wan Ahmad, W. M. R., Amin, Z. M., and Alias, M., "A study of the effectiveness of the contextual approach to teaching and learning statistics at the Universiti Tun Hussein Onn Malaysia (UTHM)," *International Journal of Arts & Sciences*, vol. 4, no. 25, p. 305, 2011.
- [2] Nutsugbodo, R. Y., Blankson-Stiles-Ocran, S., Osei, B. A., Afful, B. E. B., Kuuder, C. J. W., Alhassan, T. Z., and Amponsah, G. N., "University students' participation in campus-based events," *International Journal of Event and Festival Management*, 2024. doi: 10.1108/IJEFM-10-2023-0100.
- [3] Soegoto, E. S., "Implementing Laravel framework website as brand image in higher-education institution," *IOP Conference Series: Materials Science and Engineering*, vol. 407, no. 1, p. 012066, 2018. doi: 10.1088/1757-899X/407/1/012066.
- [4] Taipalus, T., "Database management system performance comparisons: A systematic literature review," *Journal of Systems and Software*, vol. 208, 111872, 2024. doi: 10.1016/j.jss.2023.111872.
- [5] Dobecki, M., and Zabierowski, W., "Web-based content management system," 2010.
- [6] Alawamleh, H. A., AlShibly, M. H. A. A., Tommalieh, A. F. A., Al-Qaryouti, M. Q. H., and Ali, B. J., "The challenges, barriers and advantages of management information system development: Comprehensive review," *Academy of Strategic Management Journal*, vol. 20, no. 5, pp. 1–8, 2021.
- [7] Universiti Teknologi MARA (UiTM), "Permohonan aktiviti – HEP UiTM Selangor." [Online]. Available: <https://sites.google.com/uitm.edu.my/hep-uitm-selangor/aktiviti-pelajar/permohonan-aktiviti>.
- [8] Universiti Teknologi MARA (UiTM), Selangor, *AKNC 2021 Universiti Teknologi MARA (Selangor): Pengurusan dan Pembangunan Pelajar* (Doctoral dissertation), 2021.
- [9] Eventzilla, "Eventzilla: Online event registration and ticketing." [Online]. Available: <https://www.eventzilla.net/us/home>.
- [10] Islam, S., Majumder, R., Sultana, S., Nasrin, S., and Islam, R., "Toward a generic event management system for academia," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 706–711, IEEE, 2019.
- [11] Chhetri, D. M., *Website redesign process and user experience analysis of new website, Case: Company X*, 2023.
- [12] Eventbrite, "Eventbrite: Discover great events or create your own." [Online]. Available: <https://www.eventbrite.com/>.
- [13] Bischofberger, W. R., and Pomberger, G., *Prototyping-oriented software development: Concepts and tools*. Springer, 2012. doi: 10.1007/978-3-7091-6794-7.
- [14] Malan, R., and Bredemeyer, D., *Functional requirements and use cases*. Bredemeyer Consulting, 2001, pp. 335–1653.
- [15] Zhou, Y., "Functional requirements and non-functional requirements: A survey," Doctoral dissertation, Concordia University, 2004. [Online]. Available: <https://spectrum.library.concordia.ca/id/eprint/7993>.

Appendix A: System's Flowchart of MPP & HEP's Administrative Staff, Student, Organizer's Club and Associations,



Appendix B: User Testing Response Results from Students with Signature and Official Stamp from HEP Administrative Staff

