



PROJECT PROPOSAL

Group 52 - CS

Details of Project Supervisor and Co-supervisor

Proposed Project Supervisor (Academic Staff of UCSC)

Name of the supervisor: Ms. J G S Rupasinghe

Signature of the supervisor:

Date:

Proposed Project Co-Supervisor (Assigned by Course Coordinator)

Name of the co-supervisor: Ms. H.P.N.M. Jayathilake

Signature of the co-supervisor:

Date:



Table of Contents

1. Introduction.....	4
1.1. Problem Statement.....	4
1.2. Our Solution.....	4
1.3. How Is It Different.....	5
2. Project Goals.....	5
3. Scope of the project.....	6
3.1. Users.....	6
3.2. In Scope.....	6
3.3. Out Scope.....	6
4. Objectives of the project.....	7
5. Project Feasibility.....	7
5.1. Technical Feasibility.....	7
5.2. Economical Feasibility.....	8
5.3. Social Feasibility.....	8
5.4. Legal and Ethical Feasibility.....	16
5.5. Schedule Feasibility.....	17
6. Deliverables of the project.....	17
7. Project Constraints and Assumptions.....	18
7.1 Project Constraints.....	18
7.2 Project Assumptions.....	18
8. Requirements.....	19
8.1. Functional Requirements.....	19
8.2. Quality Attribute Requirements.....	21
8.3. Use case diagram.....	23
8.4. Use case description.....	29
8.4.1. Admin.....	29
8.4.2. Moderator.....	31
8.4.3. User.....	34
8.4.4. Event Publisher.....	40
8.4.5. Sponsor.....	48
9. Technologies to be used.....	54
10. Project Timeline.....	56
11. Declaration.....	57

1. Introduction

1.1. Problem Statement

At universities, events are important for academic, cultural, and social growth. However, the current ways of sharing and promoting event information are often messy and scattered. Most details are communicated through informal channels like WhatsApp groups, Facebook pages, or word of mouth. This makes it hard for students and others to stay informed about or find events that matter to them.

Because there isn't a centralized platform, students often miss opportunities, and event organizers struggle to reach their target audience. Organizers also have a tough time securing funding since there's no formal way to request sponsorships or accept donations. Finding sponsors can take a lot of time and is often unreliable. There are few organized ways to raise financial support from the community, especially for events led by students.

Moreover, poorly defined access control makes it difficult to manage visibility for internal (university) and external (non-university) participants. To address this, the proposed system is designed to be implemented across all government universities in Sri Lanka. It will allow internal users-registered university students and staff to view all events (both public and university/faculty-specific) relevant to their university and faculty, while external users will only be able to view and register for public events.

1.2. Our Solution

UniPulse is a centralized Event Marketing System aimed at addressing challenges in university settings. It serves as a digital hub for promoting, browsing, and registering for all upcoming events academic, cultural, social, and more.

The system is being developed for all government universities in Sri Lanka. This will create a unified platform for managing and promoting events at a national level within higher education.

When an event is published and limited to a specific university, only verified users from that university can view and interact with it. External users, like the general public or those not affiliated with the university, can only view and register for public events. This setup improves audience targeting and security.

A key feature of UniPulse is its support for sponsorships and donations. Event organizers can request sponsorships directly through the platform by sending proposals for sponsors to review and respond to. Individuals or organizations can also make direct donations to support events they care about. This feature creates new funding opportunities, allowing more events to take place and enhancing their quality.

The system includes event registrations, ticket bookings, real-time notifications, and feedback submissions. Event organizers can create and manage their event listings easily through their dashboards. Users can search for events by category, date, or department. UniPulse minimizes reliance on scattered communication channels and establishes a structured, user-friendly digital space for university events.

1.3. **How Is It Different**

Currently, universities promote events across various informal platforms, making it tough to track, find, and manage events properly. Most universities use WhatsApp and Facebook for updates, but these platforms are not built for organized event management. They lack features like registration tracking, feedback systems, and audience segmentation.

UniPulse stands out by providing a secure, role-based solution. Unlike social media or bulletin boards, it organizes events by type, visibility, and audience. It also includes tools for both users and organizers, offering features like ticketing, automatic reminders, and feedback collection.

The platform fosters greater student engagement, reduces information gaps, and helps event organizers plan better. Whether it's a small departmental seminar or a large inter-university festival, UniPulse is built to handle events of all sizes, bringing clarity, convenience, and teamwork to university event culture.

2. Project Goals

The goal of the UniPulse Event Marketing System project is to simplify the discovery, promotion, and participation of university-related events through a centralized, user-friendly web platform. Designed for implementation across all government universities in Sri Lanka, the system will act as a unified hub where university students and external users can find upcoming events, get updates, and interact with event organizers easily.

To enhance access control and engagement, the application will provide internal users such as university students and staff with access to both public and internal university events. External users, who are not affiliated with the university system, will be limited to viewing and registering for public events only. The platform will also include features such as event registration, feedback forms, ticket booking, and real-time notifications to ensure a smooth and inclusive experience for all participants.

3. Scope of the project

3.1. Users

- Registered inside users(University students)
- Registered outside users
- System Admin
- Moderators
- Event Publishers
- Sponsors

3.2. In Scope

- User roles: System Admin, Moderator, Publisher, Registered users (Inside/Outside), Sponsor
- Role-based access control and permissions
- Secure login & registration (for all users)
- Event publishing workflow (with approval by moderators)
- Public and internal event visibility (based on user type)
- Ticket booking and payment system
- Volunteer, donation and sponsor request features
- Event filtering, search, and categorization
- Event boost/promotion (via paid ads)
- Report generation
- Content moderation
- Event rating and feedback system (History management)
- Sponsorship features (branding, promotion, messaging)
- Notification system (alerts, reminders, errors)

3.3. Out Scope

- Real-time Offline ticketing systems or QR scanning for physical check-ins
- Multi-language support
- Mobile application development

4. Objectives of the project

- To create a web application that shows all upcoming university events with detailed descriptions.
- To provide a clear interface for students to view, filter, and register for events.
- To let clubs and event organizers post, edit, and manage their event listings easily.
- To integrate features like ticket booking, notifications, feedback forms, and sponsor visibility in the system.
- To decrease reliance on informal communication platforms like WhatsApp and Facebook for promoting events.
- To boost overall student engagement and attendance at university events by providing better access to event information.

5. Project Feasibility

5.1. Technical Feasibility

- The system is built **entirely from scratch** using core web technologies with **no frameworks or external libraries**:
 - **Frontend:** Plain HTML, CSS, and Vanilla JavaScript will be used to create a responsive and interactive user interface.
 - **Backend:** PHP will handle all server-side logic, user session management, file uploads, role-based access, and data validation.
 - **Database:** Direct SQL queries will be used to interact with a MySQL database for CRUD operations, relationship management, and data storage.
- Manual handling of routing, form validation, security mechanisms, and error handling will be implemented to ensure full control over the development.
- While this increases development time, it ensures a deeper understanding of core technologies and better control over the system behavior.
- **Deployment** will be managed using basic virtual machines and hosting services provided by platforms such as **AWS**, **Azure**, or **Google Cloud**, without relying on any managed PaaS services or containerized environments.
- **Conclusion:** Technically feasible using core web development tools and programming logic, though development will be more time-intensive.

5.2. Economical Feasibility

- There is no **development cost** as the project is implemented by students without the need for paid frameworks or development tools.
- **Infrastructure expenses** are limited to:
 - Basic hosting services
 - Domain registration (if deployed publicly)
 - Optional cloud storage and backup
- Since payment functionality is only required in sandbox mode, no real payment processing fees will be incurred.
- The system has potential for real-world application within the university, which could reduce the need for paid third-party event platforms.
- Revenue can be generated through:
 - 10% Commission on ticket sales
 - Promotional event placements
 - Sponsorship space sales
- **Conclusion:** Economically viable for academic development and potentially profitable in real-world application.

5.3. Social Feasibility

The University Event Marketing System demonstrates strong social feasibility, as validated through a user research survey conducted among university students and event participants.

- **Identified Need:** According to our survey, over **80%** of students reported difficulties in discovering or staying updated on university events due to scattered announcements across informal platforms like WhatsApp and Facebook. This highlights a clear demand for a centralized system.
- **High Interest in the System:** More than **85%** of respondents expressed interest in a platform that offers centralized access to event listings, registration, and updates. This confirms the system's relevance and appeal to the target audience.
- **Preferred Features:** Students prioritized features such as **reminders for events**, **easy ticket booking**, and **filtering events by faculty or interest**. These preferences have been directly incorporated into the system design.
- **Adoption Readiness:** The majority of respondents were already familiar with web applications and expressed confidence in using a new event platform. There were minimal concerns regarding ease of use, showing strong readiness for adoption.

- **Community Engagement:** The system encourages greater participation in university life by improving communication between clubs, organizers, sponsors, and students. It also facilitates volunteer and sponsor opportunities, supporting deeper engagement within the academic community.

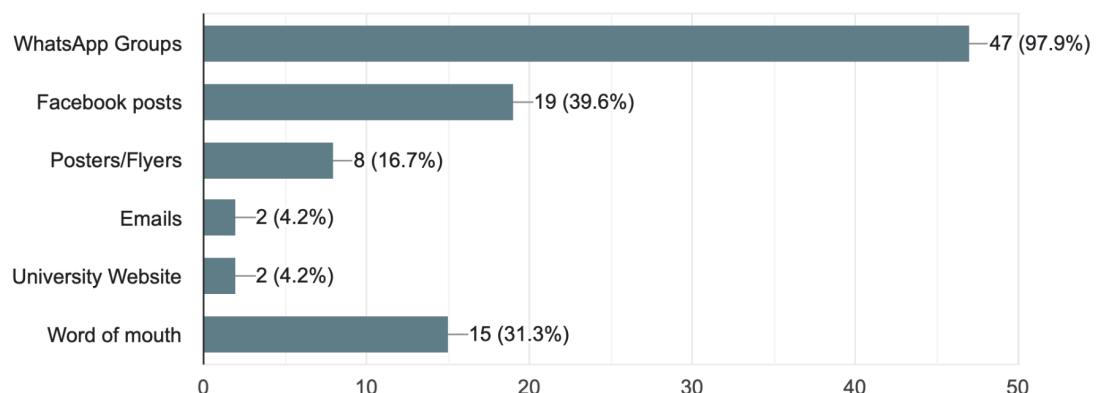
In summary, the system aligns well with current user behavior and fills a gap in university event management. The survey results indicate strong social support and a high likelihood of user adoption once the platform is deployed.

Results of the Survey Conducted for Feasibility Study

Users

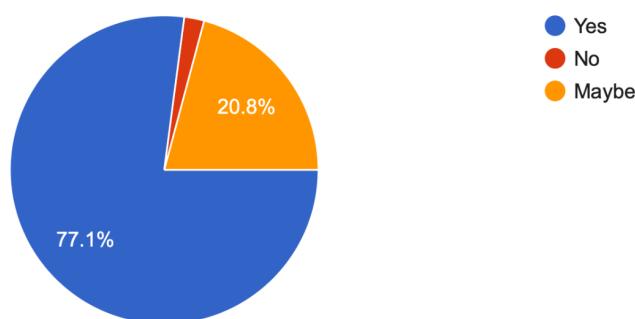
How do you usually hear about university events?

48 responses



Would you trust a university-approved system to handle ticket payments and donations securely?

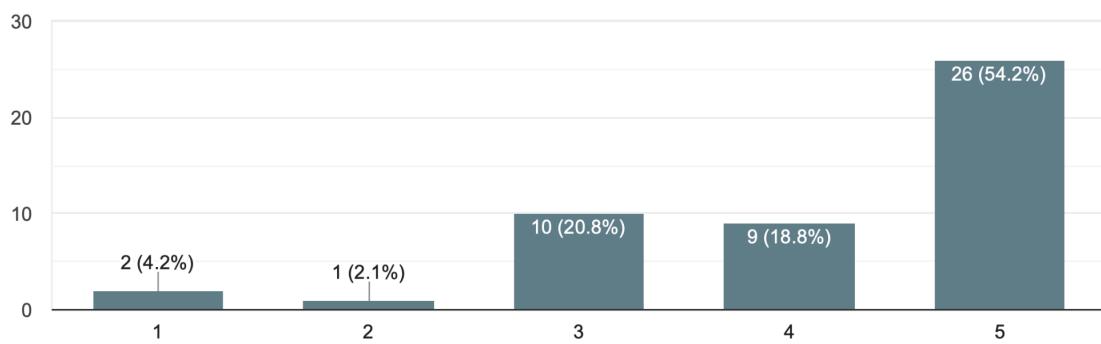
48 responses



Platform features expected by users

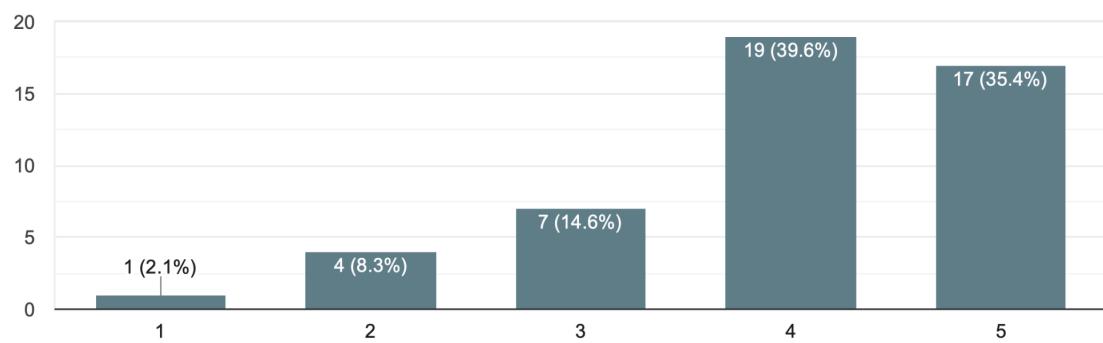
Browsing a list of upcoming events

48 responses



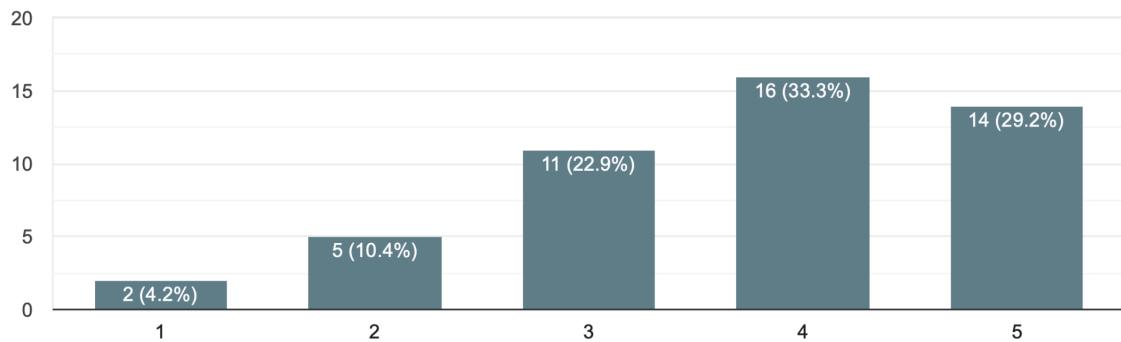
Filtering events by type (Free/Paid, Public/Internal)

48 responses



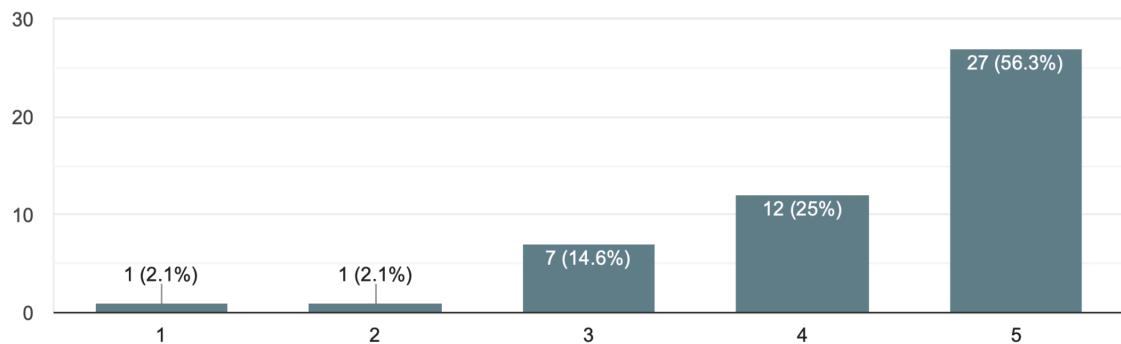
Asking questions about a event through a comment section

48 responses



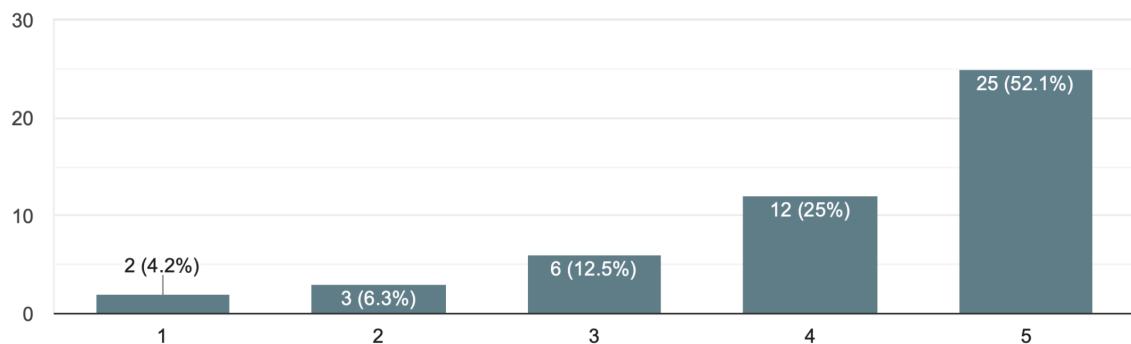
Booking tickets online

48 responses



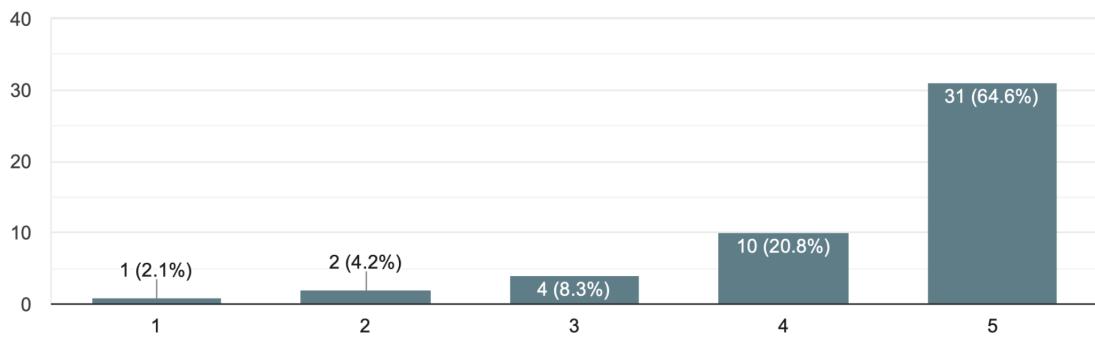
Receiving email/SMS event reminders

48 responses



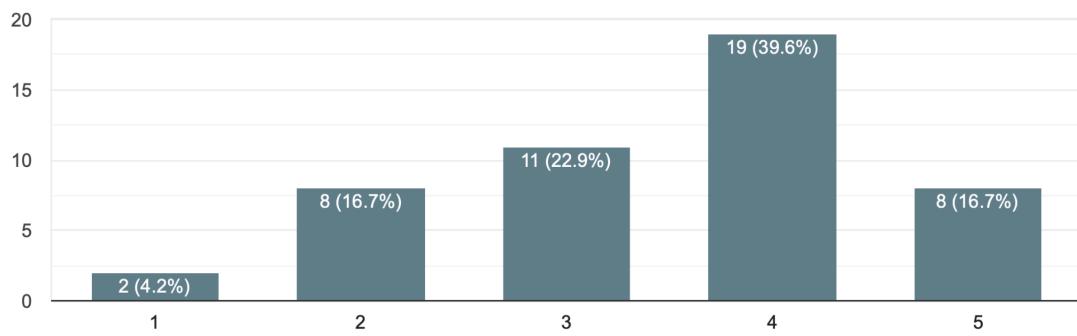
Getting student discounts

48 responses



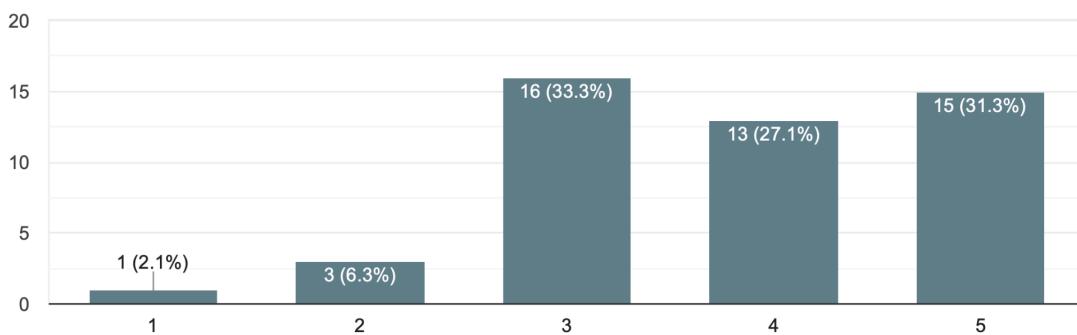
Viewing past events and reviews(Organized by a certain club/organizer)

48 responses



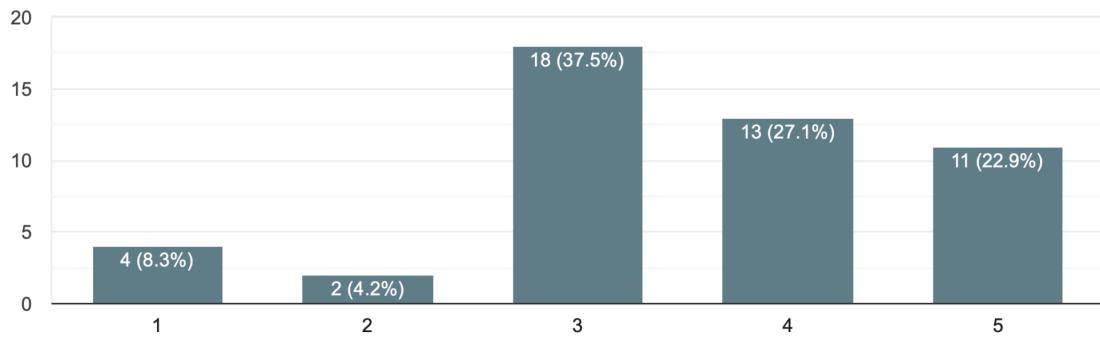
Rating clubs/organizers

48 responses



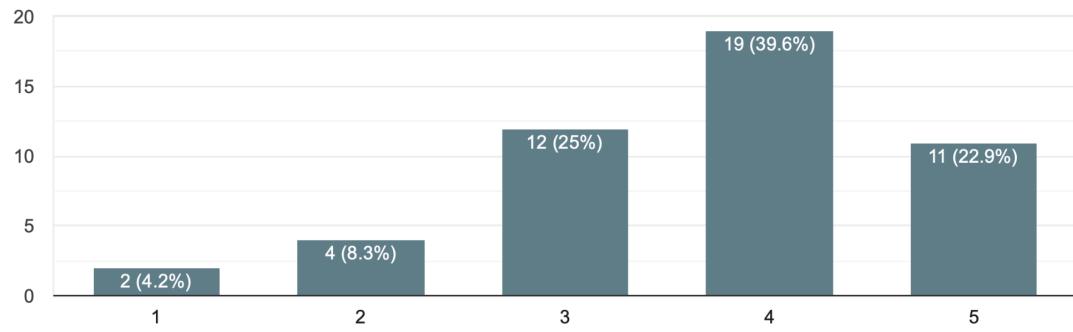
Support events through donations

48 responses



Applying as a volunteer for events

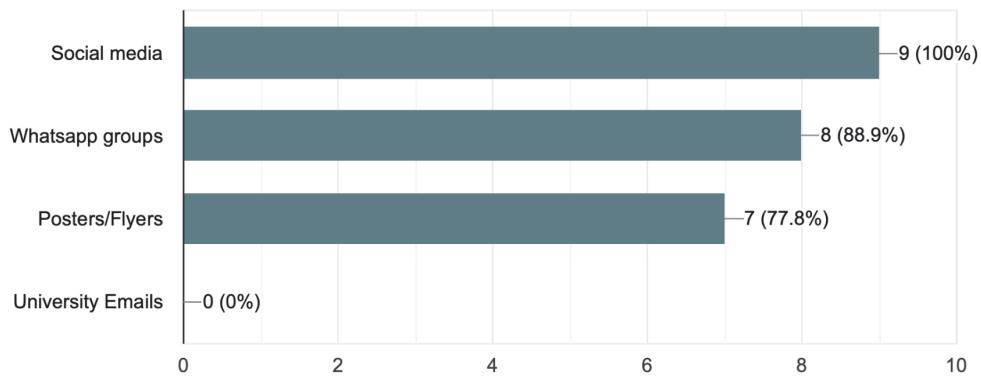
48 responses



Publisher

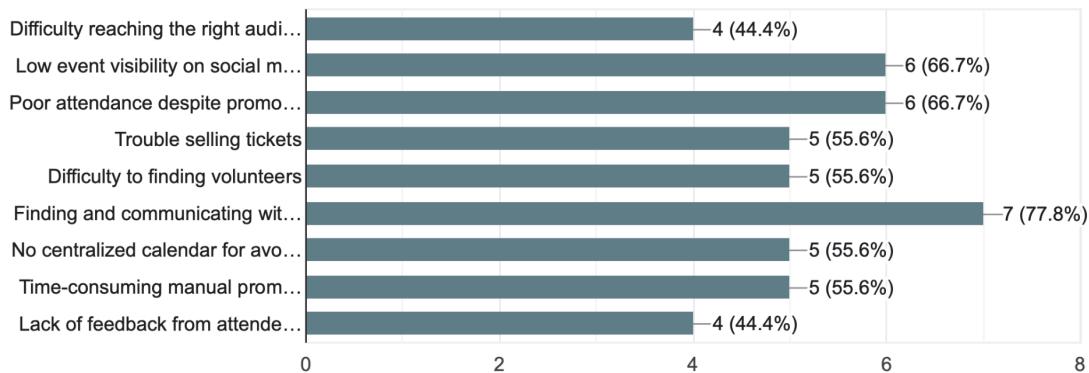
How do you currently promote your events?

9 responses



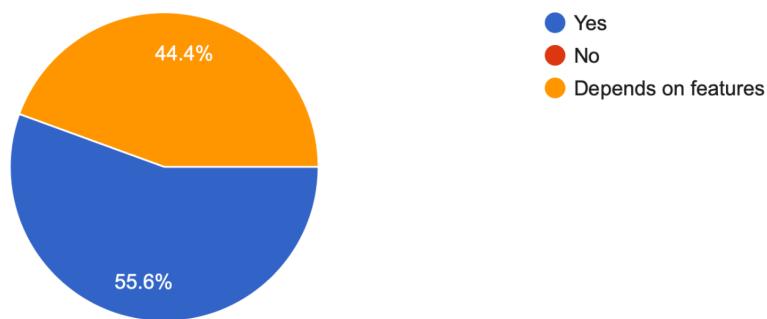
What challenges do you face in organizing/promoting events?

9 responses



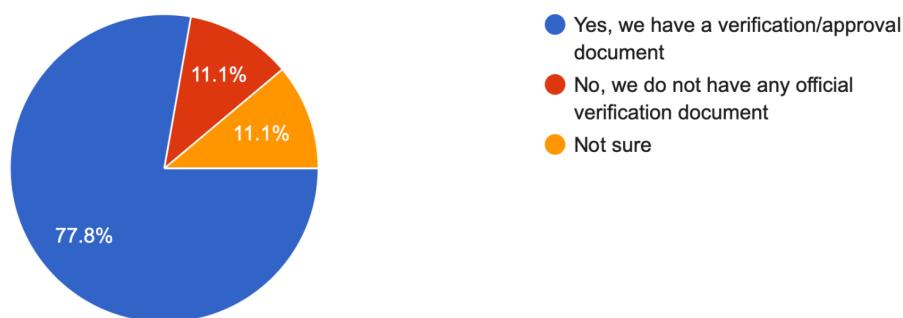
Would you be willing to pay a small commission on ticket sales to use the platform?

9 responses



Do you have an official document that verifies your club or union is recognized by your university or faculty?

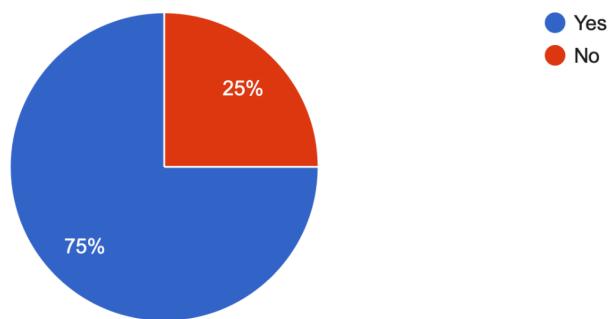
9 responses



Moderator

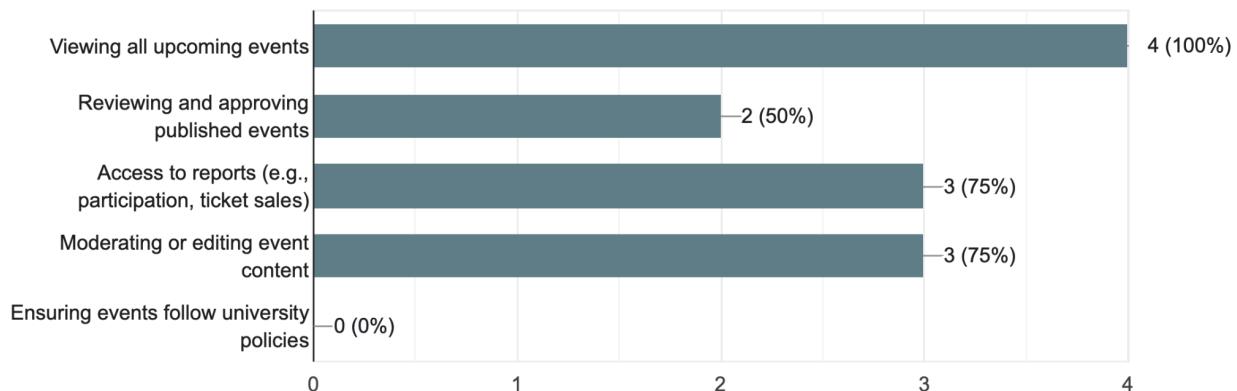
Do you oversee or coordinate any student-run events at your faculty/department?

4 responses



Which of the following would be helpful to you in a staff/admin role?

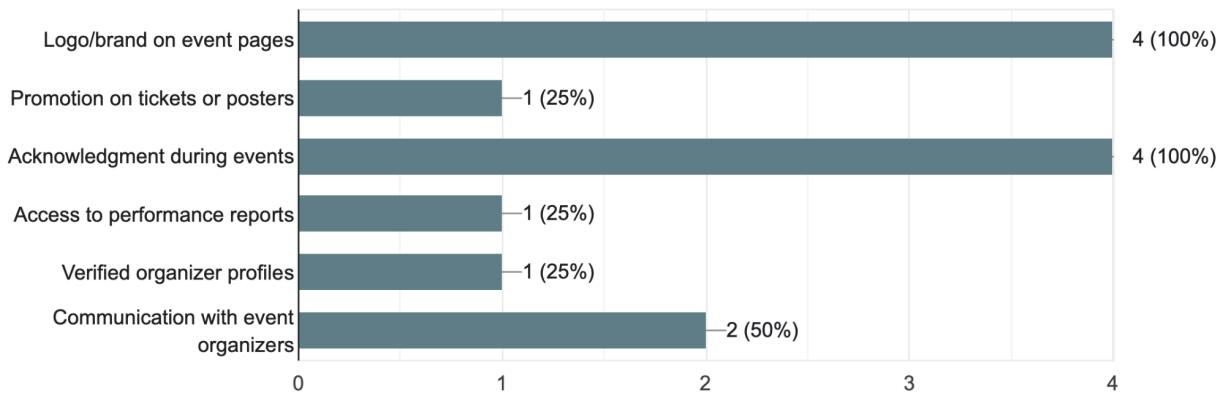
4 responses



Sponsor

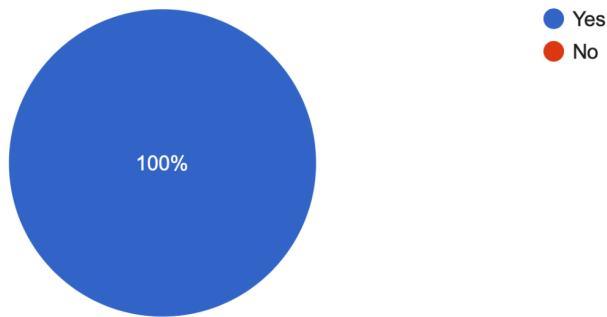
What benefits would you expect as a sponsor?

4 responses



Would a secure, centralized platform encourage you to sponsor more events?

4 responses



5.4. Legal and Ethical Feasibility

The project complies with necessary legal and ethical considerations:

- **Data Privacy and Protection:**

- The system will store personal data such as email addresses, names, and club details. Proper measures will be taken to **encrypt sensitive data** and protect user privacy.
- The project will adhere to general data protection guidelines applicable in Sri Lanka (or international equivalents like GDPR, where applicable in future deployments).

- **User Verification and Consent:**
 - Registration requires verified email addresses and consent to platform policies, ensuring ethical data collection.
 - Publishers and sponsors are required to provide documentation to verify their legitimacy, adding a layer of responsibility and transparency.
- **Intellectual Property and Academic Integrity:**
 - The system is developed from scratch using core technologies, avoiding third-party frameworks that may have restrictive licenses.
 - All contributors have signed a declaration of originality, ensuring academic honesty.
- **Content Moderation and Misuse Prevention:**
 - Moderators will have tools to remove or hide inappropriate content (e.g., fake events, spam, or offensive materials), aligning with university policies and ethical standards.

5.5. Schedule Feasibility

- The project spans a duration of **10 months**.
- Estimated man-hours for project completion are calculated as:
 - Weekday working hours = 5 hours
 - Weekend working hours = 5 hours
 - Number of group members = 4
 - Number of active project weeks = 36
 - **Total man-hours = $(5 + 5) \times 4 \times 36 = 1,440$ hours**
- With a clearly defined scope and fixed requirements, the **Waterfall model** is selected to manage development in sequential phases.
- The **requirement gathering** phase is nearly complete. System design and development phases are expected to begin immediately after.
- Based on the project's scale and team availability, we are confident in completing the development and deployment **by mid-april 2026**

6. Deliverables of the project

- **Web Application**
- **Documentation**

7. Project Constraints and Assumptions

7.1 Project Constraints

→ Time Constraints

- ◆ The project must be completed within a 1 year academic year.
- ◆ Planning, designing, development and implementation has fixed deadlines.

→ Resource Constraints

- ◆ Only 4-5 members are available to work on the project.

→ Technical Constraints

- ◆ Frameworks are not allowed. Development must be done in pure HTML, PHP, JavaScript and MYSQL.

→ Scope Constraints

- ◆ Only government university-level events will be published.

→ Security Constraints

- ◆ Must handle basic user authentication and input validation.
- ◆ No handling of real payment gateways.

7.2 Project Assumptions

→ User Assumptions

- ◆ All users have access to a website with internet connection.
- ◆ Users will provide valid email addresses and identities during registration.

→ Publisher Verification

- ◆ Publishers will upload valid documents or use university verified email to verify their identity.
- ◆ The admin will manually verify and approve publishers within 24-48 hours.

→ Event Scope

- ◆ Events will include only university related events such as workshops, cultural events, musical events, hackathons and sports events.
- ◆ Publishers will provide correct and complete information about the events.

→ Platform Usage

- ◆ Users will register and purchase tickets using the website.

→ Sponsorship

- ◆ Sponsors will contact publishers directly through this website.

8. Requirements

8.1. Functional Requirements

1. System Administrator

- Admin shall be able to add and remove user accounts.
- Admin shall be able to view and edit user details.
- Admin shall be able to manage website content (add, edit, delete).
- Admin shall be able to view the system dashboard with key statistics.
- Admin shall be able to change user roles (e.g., assign publisher, moderator).
- Admin shall be able to monitor system logs, errors, and perform backups.

2. Moderator

- Moderator shall be able to register and create a moderator account.
- Moderator shall be able to securely log in and log out.
- Moderator shall be able to view and review pending club/union registration requests.
- Moderator shall be able to accept or reject verification requests with reason.
- Moderator shall be able to view published event posts.
- Moderator shall be able to delete or hide inappropriate event posts.
- Moderator shall be able to contact event publishers.
- Moderator shall be able to manage and moderate user comments.
- Moderator shall be able to generate and export event reports.

3. General Users (Students, Staff, Guests)

- Guest users shall be able to register with role-specific details (inside/outside user).
- Registered users shall be able to log in and log out.
- Users shall be able to visit the home page and navigate the site.
- Users shall be able to view public events.
- Internal users (students/staff) shall be able to view university-specific (internal) events.
- Users shall be able to filter events based on various criteria.
- Users shall be able to check ticket availability.
- Users shall be able to book tickets for events and receive confirmation.

- Internal users shall be able to receive discounts on tickets if eligible.
- Users shall be able to make secure payments via the system.
- Users shall be able to provide feedback on events.
- Users shall be able to rate clubs and organizers.
- Users shall be able to make donations to clubs/events.
- Users shall be able to receive notifications related to bookings, payments, or updates.
- Users shall be able to apply as volunteers for events.

4. Event Publisher (Club/Union/Faculty Organizer)

- Event publishers shall be able to register with club/union details.
- Event publishers shall be able to log in and log out securely.
- Publishers shall be able to create and update a detailed profile.
- Publishers shall be able to publish events with complete details.
- Publishers shall be able to manage event visibility (public/internal).
- Publishers shall be able to boost event visibility via paid promotions (sandbox mode).
- Publishers shall be able to manage ticket sales (pricing, discounts, attendee list).
- Publishers shall be able to request volunteers and manage applications.
- Publishers shall be able to request sponsors for events.
- Publishers shall be able to request donations for events.
- Publishers shall be able to manage event history (past, ongoing, drafts).
- Publishers shall be able to make payments for platform services (e.g., boosting).

5. Sponsor / Donator

- Sponsors shall be able to register by submitting business/organization details.
- Sponsors shall be able to log in and log out securely.
- Sponsors shall be able to create and update a sponsor profile.
- Sponsors shall be able to view event listings with sponsorship needs.
- Sponsors shall be able to send messages to event organizers.
- Sponsors shall be able to promote their brand through sponsored posts.
- Sponsors shall be able to propose custom sponsorship terms/conditions.
- Sponsors shall be able to make donations to events.
- Sponsors shall be able to view and manage their sponsorship/donation history.

8.2. Quality Attribute Requirements

1. Usability

- Ensure the interface is intuitive and easy to use for all roles (students, publishers, sponsors, moderators, admins).
- Design user-friendly forms for event creation, ticket booking, sponsorship requests, and feedback.
- Maintain consistent navigation and design throughout the platform.
- Provide user manuals, FAQs, and in-app help for common features.
- Include tutorials or onboarding guidance for new users.

2. Reliability

- Implement proper error handling for issues like failed submissions, payment failures, or system errors.
- Show clear error messages with helpful instructions.
- Validate user inputs and ensure data consistency across modules.
- Keep accurate records of all transactions and updates.
- Perform regular data backups (events, user data, payments, etc.).
- Include a reliable recovery mechanism for restoring data in case of failure.

3. Performance

- Ensure quick and responsive interactions for users across all major functions.
- Design the system to handle growth in users, events, and activity without slowing down.
- Use load distribution techniques to maintain system responsiveness during high traffic.

4. Security

- Use secure login mechanisms and verify users based on roles.
- Apply role-based access control to prevent unauthorized access or actions.
- Encrypt sensitive data (personal info, payment data) both in storage and transmission.
- Maintain activity logs for critical actions like role changes, event approvals, and transactions.

5. Maintainability

- Use a modular system design (separate modules for users, events, payments, etc.).
- Write clean, well-documented code following standard best practices.
- Implement automated testing for key features to prevent bugs during updates.

6. Scalability

- Optimize the database to handle large volumes of data efficiently.
- Ensure the platform can grow with increased demand from users and events.

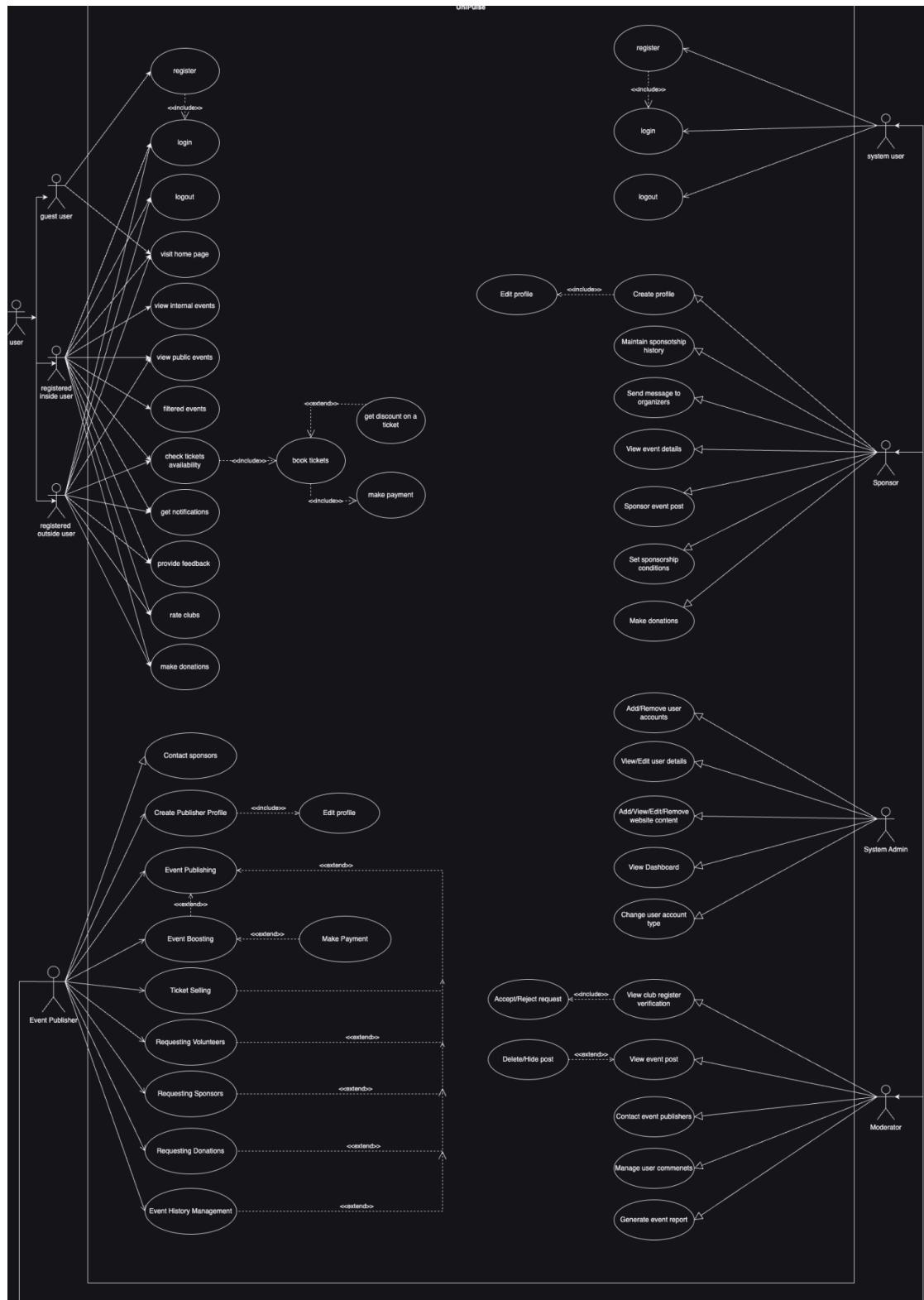
7. Compliance

- Adhere to relevant data protection laws and digital transaction regulations.
- Keep detailed records and logs for transparency and accountability.
- Ensure event data and financial transactions are traceable and audit-ready.

8. Flexibility

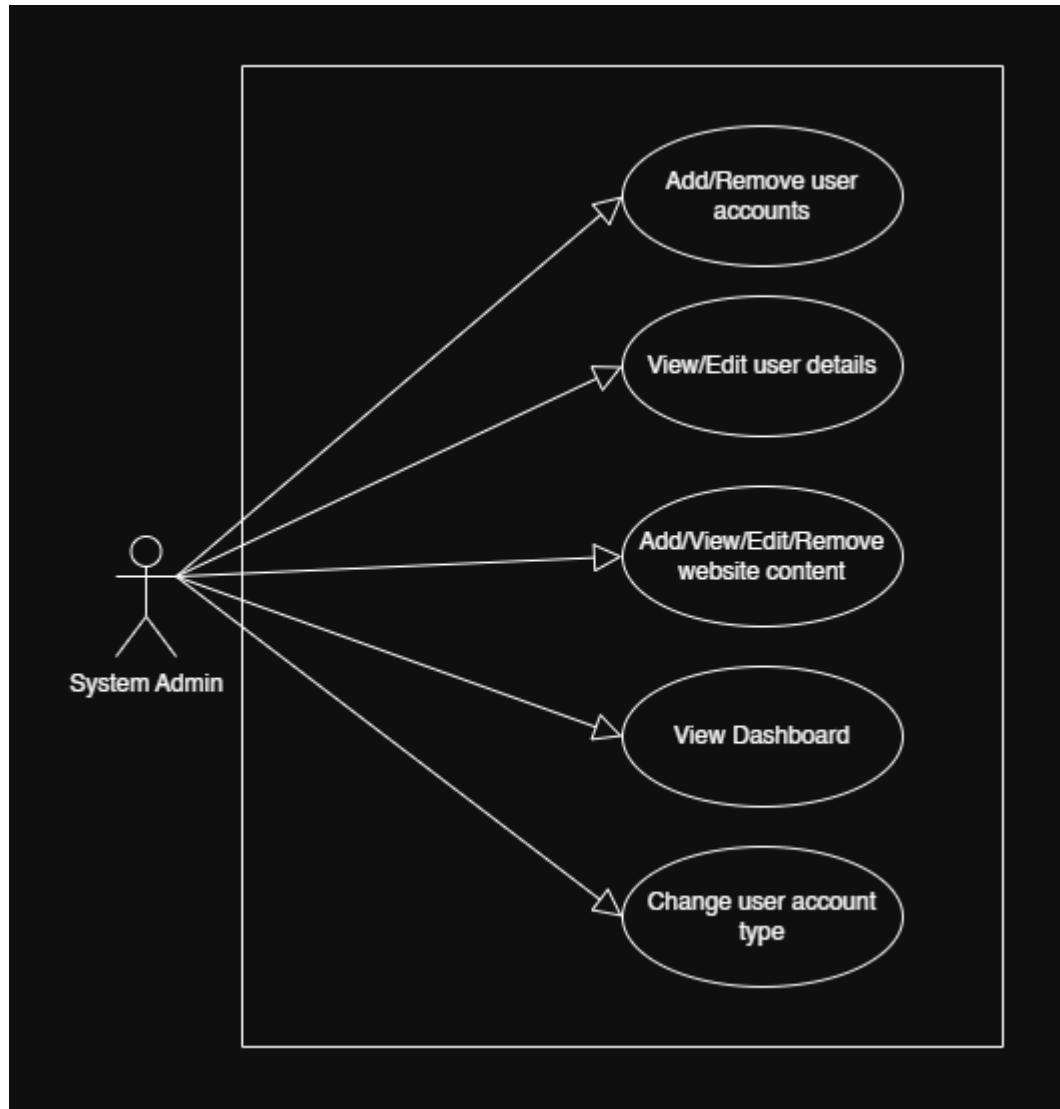
- Allow customizable event options (e.g., public/private, ticket pricing, volunteer/sponsor requests).
- Support adaptable workflows based on different event types or publisher preferences.
- Use infrastructure that can be easily scaled or modified as platform needs evolve.

8.3. Use case diagram

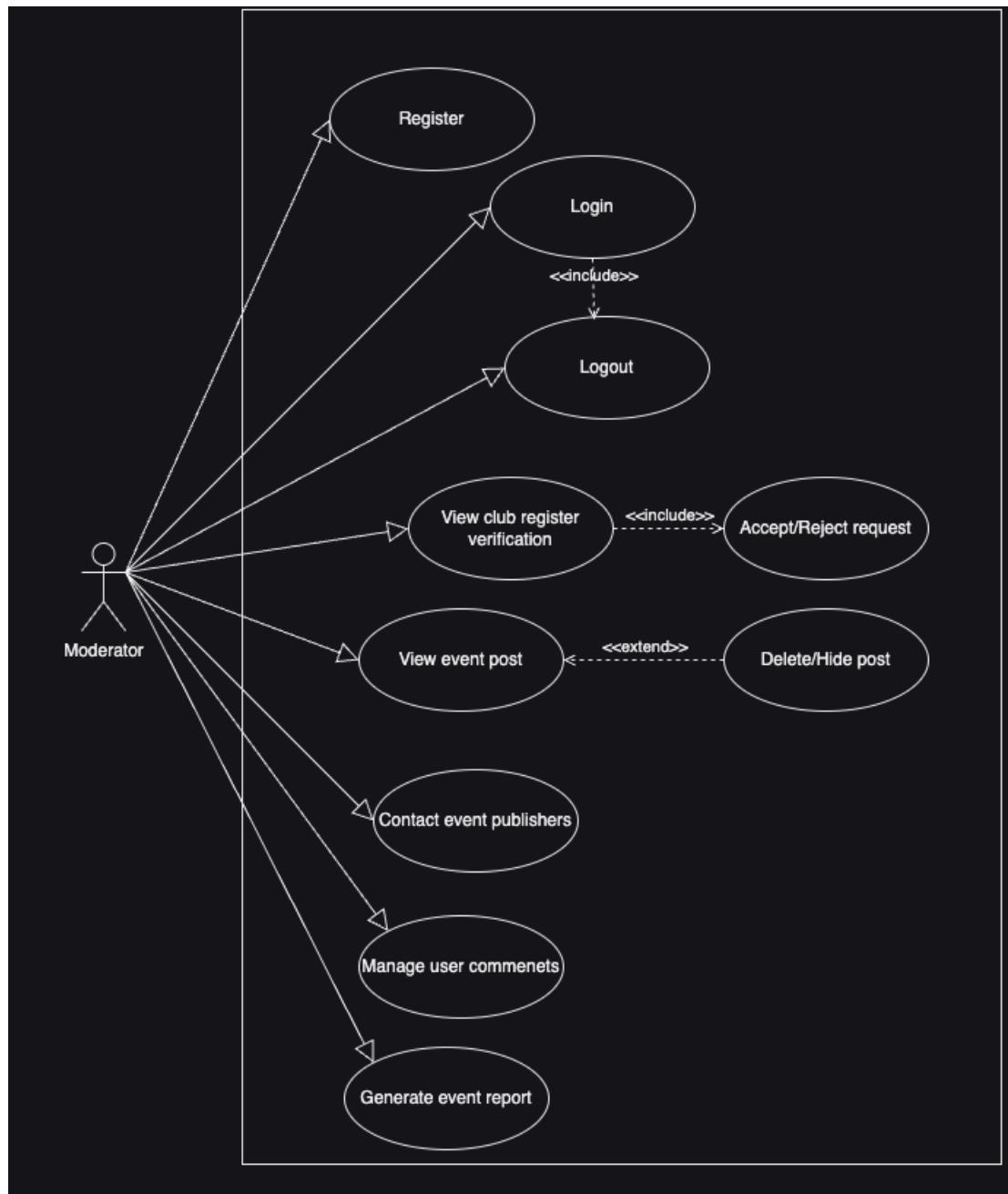


https://drive.google.com/drive/folders/1PoB8unRbMGsO0iLHSpjEjN59x85_Hryv?usp=sharing
Here is the link for clear image.

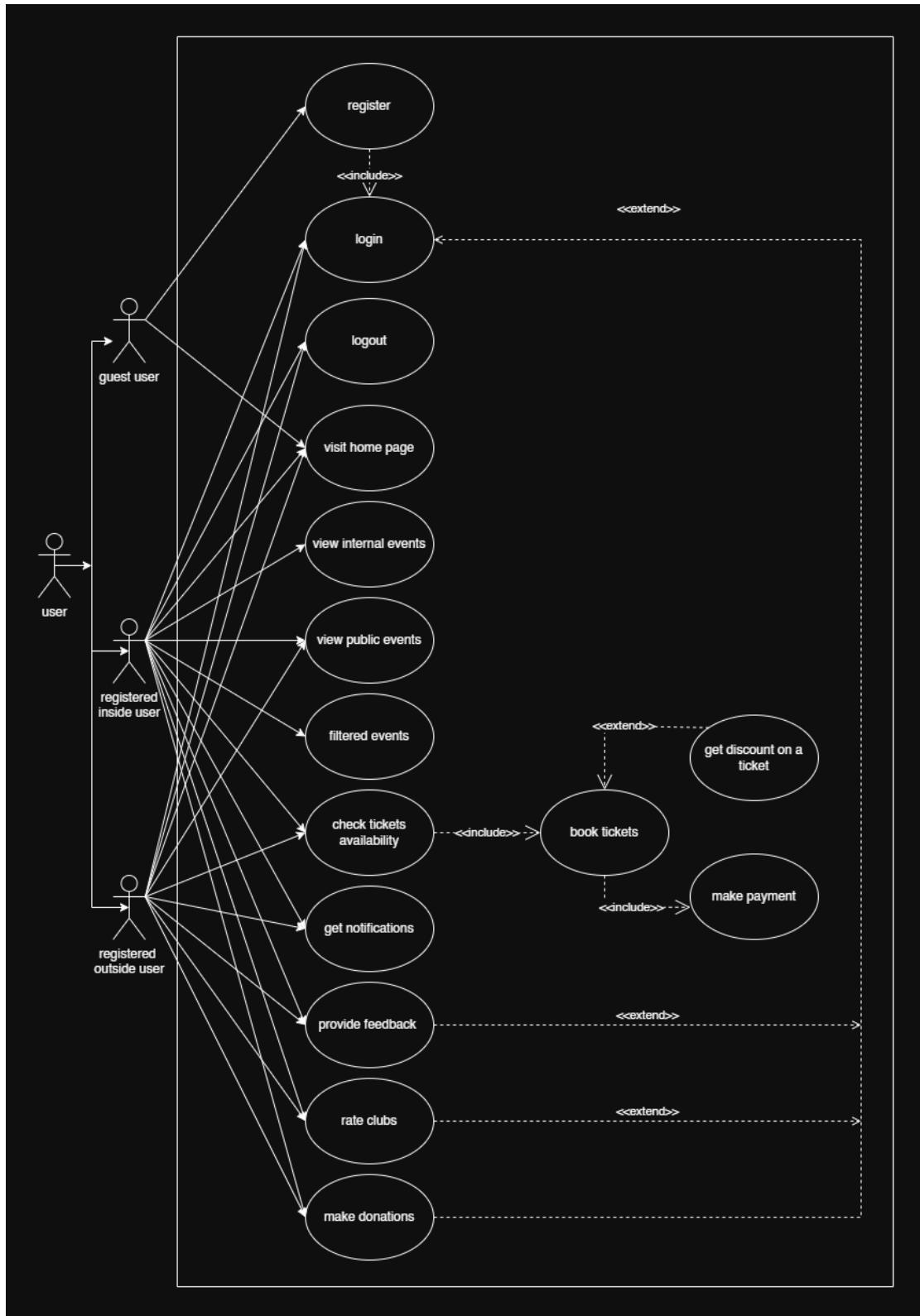
8.3.1 System Admin



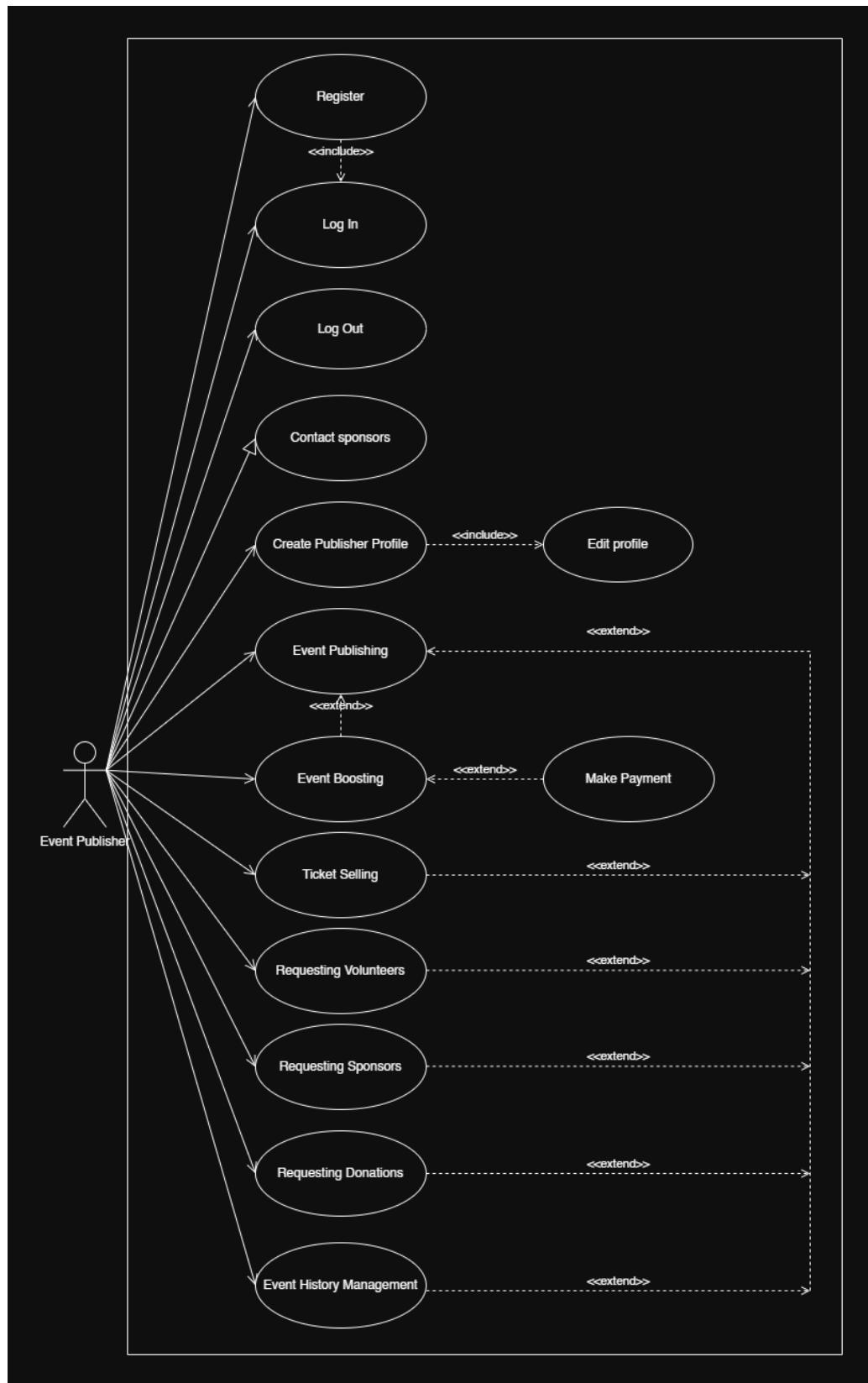
8.3.2 Moderator



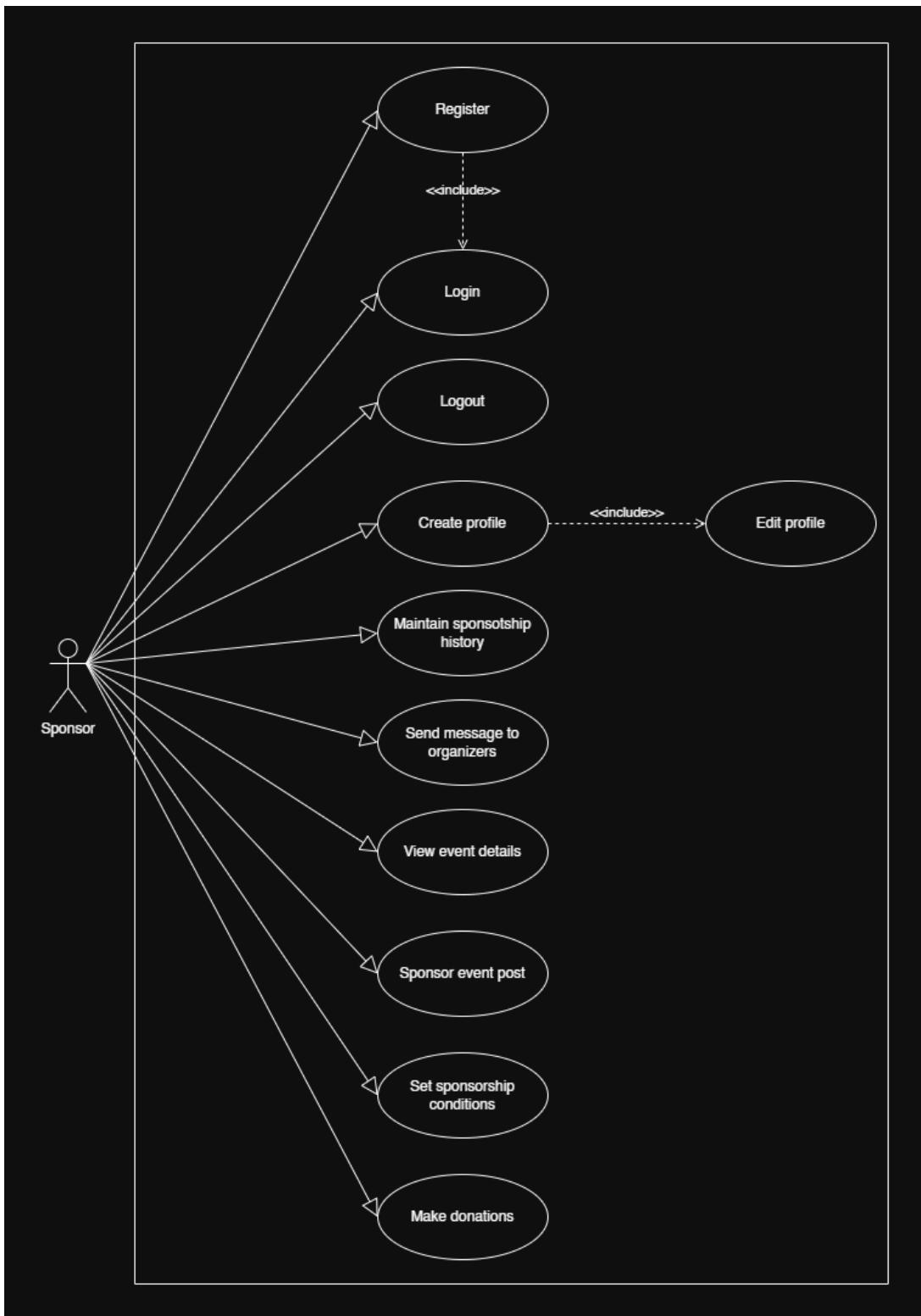
8.3.3 User



8.3.4 Event Publisher



8.3.5 Sponsor



8.4. Use case description

8.4.1. Admin

Use Case ID	SA-01
Use Case Name	Add/Remove user accounts
Primary Actor/s	System admin
Description	The system admin can create new user accounts or delete existing ones from the system.
Pre-Conditions	Admin must be logged in with appropriate privileges
Main scenario	<ol style="list-style-type: none"> 1. Navigates to user management section 2. Selects option to add new user or remove existing user 3. For adding : Enters required user details 4. For removing : Select user and confirms deletion
Exception	<ul style="list-style-type: none"> • Invalid user data entered • Attempt to delete non-existent user
Post-Conditions	User account is created/deleted in system

Use Case ID	SA-02
Use Case Name	View/Edit user details
Primary Actor/s	System admin
Description	Enables admin to view and modify user information
Pre-Conditions	Admin logged in. User exists
Main scenario	<ol style="list-style-type: none"> 1. Access user list 2. Select user 3. View details 4. Edit the details 5. Update the system
Exception	<ul style="list-style-type: none"> • User not found • Invalid edits attempted
Post-Conditions	User details are updated

Use Case ID	SA-03
Use Case Name	Manage Website Content
Primary Actor/s	System admin
Description	Allows admin to add, view, edit or remove website content
Pre-Conditions	Admin must be logged in with appropriate privileges
Main scenario	<ol style="list-style-type: none"> 1. Go to content management 2. Select action 3. Perform operation 4. Save changes
Exception	<ul style="list-style-type: none"> • Content validation fails • Unauthorized changes attempted
Post-Conditions	Website content is modified

Use Case ID	SA-04
Use Case Name	View dashboard
Primary Actor/s	System admin
Description	Provides system overview and metrics
Pre-Conditions	Admin must be logged in with appropriate privileges
Main scenario	<ol style="list-style-type: none"> 1. Access dashboard 2. View system statistics
Exception	No data available
Post-Conditions	Admin views system status

Use Case ID	SA-05
Use Case Name	Change user role
Primary Actor/s	System admin
Description	Allows modification of user privilege levels
Pre-Conditions	Admin must be logged in with appropriate privileges, user exists
Main scenario	<ol style="list-style-type: none"> 1. Select user 2. Choose new type 3. Confirm change
Exception	<ul style="list-style-type: none"> • Invalid privilege level selected • User not found
Post-Conditions	User privileges are updated

8.4.2. Moderator

Use Case ID	MOD-01
Use Case Name	Register
Primary Actor/s	Moderator
Description	Allows user to create moderator account
Pre-Conditions	User not registered
Main scenario	<ol style="list-style-type: none"> 1. Select registration 2. Enter details 3. Submit form 4. Receive confirmation
Include	MOD-02: Login
Exception	<ul style="list-style-type: none"> • Invalid registration data • Account already exists
Post-Conditions	New moderator account created

Use Case ID	MOD-02
Use Case Name	Login
Primary Actor/s	Moderator
Description	Authenticates moderator into system
Pre-Conditions	Account exists
Main scenario	<ol style="list-style-type: none"> 1. Go to login page 2. Enter credentials 3. System verifies
Exception	<ul style="list-style-type: none"> • Invalid credentials • Account locked
Post-Conditions	Moderator gains access

Use Case ID	MOD-03
Use Case Name	Logout
Primary Actor/s	Moderator
Description	Ends current session
Pre-Conditions	Moderator logged in
Main scenario	<ol style="list-style-type: none"> 1. Select logout 2. System terminates session

Exception	Session already expired
Post-Conditions	Moderator logged out and returns to homepage

Use Case ID	MOD-04
Use Case Name	View club registration verification
Primary Actor/s	Moderator
Description	Review pending club registration request
Pre-Conditions	Moderator logged in, request exists
Main scenario	<ol style="list-style-type: none"> 1. Access verification queue 2. View request details
Include	MOD-05: Accept/Reject request
Exception	No pending requests
Post-Conditions	Requests are visible for review

Use Case ID	MOD-05
Use Case Name	Accept/Reject request
Primary Actor/s	Moderator
Description	Approves or denies club verifications
Pre-Conditions	Viewing requests
Main scenario	<ol style="list-style-type: none"> 1. Select request 2. Choose action 3. Provide reason if rejecting 4. Confirm
Exception	Invalid selection
Post-Conditions	Request status updated
Include	View club register verification

Use Case ID	MOD-06
Use Case Name	View event posts
Primary Actor/s	Moderator
Description	Inspects publishes events
Pre-Conditions	Moderator logged in, events exist
Main scenario	<ol style="list-style-type: none"> 1. Navigate to events 2. View post details

Include	MOD-07: Delete/Hide post
Exception	No events available
Post-Conditions	Events are visible

Use Case ID	MOD-07
Use Case Name	Delete/Hide post
Primary Actor/s	Moderator
Description	Removes or hides inappropriate posts
Pre-Conditions	Viewing event post
Main scenario	<ol style="list-style-type: none"> 1. Select post 2. Choose action 3. Preview reason 4. Confirm
Exception	<ul style="list-style-type: none"> • Post already removed • Not an inappropriate post
Post-Conditions	Post is deleted/hidden

Use Case ID	MOD-08
Use Case Name	Contact event publishers
Primary Actor/s	Moderator
Description	Communicates with event organizers
Pre-Conditions	Moderator logged in, publishers exist
Main scenario	<ol style="list-style-type: none"> 1. Select publisher 2. Choose method 3. Send message
Exception	Publisher unavailable
Post-Conditions	Message delivered

Use Case ID	MOD-09
Use Case Name	Manage user comments
Primary Actor/s	Moderator
Description	Monitors and moderates user comments
Pre-Conditions	Comments exists

Main scenario	<ol style="list-style-type: none"> 1. Access comments 2. Review 3. Take action if needed
Exception	No comments available
Post-Conditions	Comments are moderated

Use Case ID	MOD-10
Use Case Name	Generate event report
Primary Actor/s	Moderator
Description	Creates event activity reports
Pre-Conditions	Moderator logged in, event exists
Main scenario	<ol style="list-style-type: none"> 1. Select report option 2. Set parameters 3. Generate 4. Save/export
Exception	No data available
Post-Conditions	Report is created

8.4.3. User

Use Case ID	U-01
Use Case Name	Register
Primary Actor/s	Guest User
Description	A guest user creates a new account to become a registered user (either inside or outside user).
Pre-Conditions	User has not registered before using the same email.
Main Scenario	<ol style="list-style-type: none"> 1. Visit registration page 2. Enter required details (name, email, password, etc.) 3. Submit registration form
Include	U-02: Login

Exceptions	<ul style="list-style-type: none"> • Email already exists: Show error • Validation failed: Highlight missing/invalid fields
Post-Conditions	<ul style="list-style-type: none"> • User account is created • User can now log in

Use Case ID	U-02
Use Case Name	Login
Primary Actor/s	All Registered Users
Description	Allows users to authenticate and access features relevant to their roles.
Pre-Conditions	Users must have a registered account.
Main Scenario	<ol style="list-style-type: none"> 1. Visit login page 2. Enter email and password 3. Submit form
Extend	U-12: Provide Feedback
Extend	U-13: Rate Clubs
Extend	U-14: Make Donations
Exceptions	Invalid credentials: Show error
Post-Conditions	User is logged into the system

Use Case ID	U-03
Use Case Name	Logout
Primary Actor/s	All Registered Users
Description	Terminate current user session and exit the system
Pre-Conditions	User is logged in
Main Scenario	<ol style="list-style-type: none"> 1. Click logout button 2. System terminates session 3. System redirects to login page
Exceptions	Session timeout, network error

Post-Conditions	User session ended, user redirected to public pages
-----------------	---

Use Case ID	U-04
Use Case Name	Book Tickets
Primary Actor/s	Registered Inside User, Registered Outside User
Description	The user books tickets for selected events after checking availability and applying discounts (if available).
Pre-Conditions	Users must be logged in and select an event with available tickets.
Main Scenario	<ol style="list-style-type: none"> 1. Select an event 2. Check ticket availability 3. Apply discount (if eligible) 4. Proceed to payment 5. Confirm booking
Include	U-09: Make payment
Extend	U-10: Get discount on a ticket
Exceptions	<ul style="list-style-type: none"> ● No tickets available: Show alert ● Payment failure: Display error and retry option
Post-Conditions	<ul style="list-style-type: none"> ● Ticket is booked successfully ● Confirmation and ticket details are sent to the user

Use Case ID	U-05
Use Case Name	Visit Home Page
Primary Actor/s	All Users

Description	Opens the home page showing event highlights and general navigation
Pre-Conditions	None
Main Scenario	<ol style="list-style-type: none"> 1. Open application 2. Navigate to homepage
Exceptions	None
Post-Conditions	Home page is displayed

Use Case ID	U-06
Use Case Name	View Public Events
Primary Actor/s	Registered Users
Description	Allows all registered users to view public events.
Pre-Conditions	None
Main Scenario	<ol style="list-style-type: none"> 1. Visit events page 2. Browse public event listings
Exceptions	None
Post-Conditions	List of public events is shown

Use Case ID	U-07
Use Case Name	View University Events
Primary Actor/s	Registered Inside User
Description	Allows inside users to access internal university events.
Pre-Conditions	Must be logged in as a registered inside user
Main Scenario	<ol style="list-style-type: none"> 1. Login 2. Go to university events section
Exceptions	Unauthorized access (for outside users)
Post-Conditions	Internal events are visible

Use Case ID	U-08
Use Case Name	Filtered Events
Primary Actor/s	Registered Inside Users
Description	Users can filter events based on criteria

Pre-Conditions	Events must be available
Main Scenario	<ol style="list-style-type: none"> 1. Go to events page 2. Apply filter options
Exceptions	No events found matching filters
Post-Conditions	Filtered list of events is displayed

Use Case ID	U-09
Use Case Name	Check Ticket Availability
Primary Actor/s	Registered Users
Description	Checks whether tickets are available for a selected event.
Pre-Conditions	User must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Select event 2. Check availability
Include	U-03: Book tickets
Exceptions	Tickets sold out
Post-Conditions	Availability status is shown

Use Case ID	U-10
Use Case Name	Make Payment
Primary Actor/s	Registered Users
Description	Finalizes the ticket booking by making an online payment.
Pre-Conditions	Must have selected tickets
Main Scenario	<ol style="list-style-type: none"> 1. Choose payment method 2. Enter details 3. Confirm payment
Exceptions	Payment declined or failed
Post-Conditions	Payment success is recorded

Use Case ID	U-11
Use Case Name	Get Discount on a Ticket
Primary Actor/s	Registered Users

Description	Eligible users can apply a discount to reduce ticket price.
Pre-Conditions	Discount criteria must be met
Main Scenario	<ol style="list-style-type: none"> 1. Add ticket to cart 2. Apply discount code/auto-discount
Exceptions	Invalid or expired discount
Post-Conditions	Discount is applied

Use Case ID	U-12
Use Case Name	Provide Feedback
Primary Actor/s	Registered Users
Description	Users can submit feedback about events.
Pre-Conditions	Must be logged in and have attended an event
Main Scenario	<ol style="list-style-type: none"> 1. Open feedback form 2. Fill comments 3. Submit feedback
Exceptions	Empty feedback submission
Post-Conditions	Feedback is stored

Use Case ID	U-13
Use Case Name	Rate Clubs
Primary Actor/s	Registered Users
Description	Users can give ratings to clubs or organizers
Pre-Conditions	Must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Visit club/organizer profile 2. Select rating 3. Submit
Exceptions	Duplicate rating attempts
Post-Conditions	Rating is recorded

Use Case ID	U-14
Use Case Name	Make Donations
Primary Actor/s	Registered Users
Description	Users can donate to clubs/events
Pre-Conditions	Must be logged in

Main Scenario	<ol style="list-style-type: none"> 1. Choose event/club 2. Enter donation amount 3. Make payment
Exceptions	Payment failed
Post-Conditions	Donation recorded

Use Case ID	U-15
Use Case Name	Get Notifications
Primary Actor/s	Registered Users
Description	Users receive updates about events, bookings, and payments
Pre-Conditions	Must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Log in 2. Notifications appear in user dashboard
Exceptions	None
Post-Conditions	Notifications displayed or emailed

8.4.4. Event Publisher

Use Case ID	EP-01
Use Case Name	Register
Primary Actor/s	Event Publisher
Description	The event publisher registers to the website by providing club/union details
Pre-Conditions	The event publisher has not already registered with the same email or name
Main Scenario	<ol style="list-style-type: none"> 1. Open registration page 2. Filling details (email, password, club/union name etc.) 3. Submit form
Include	EP-02: Log In
Exceptions	<ul style="list-style-type: none"> • If email or password already exists show error • If required field is missing display validation alerts

Post-Conditions	<ul style="list-style-type: none"> • Event publisher account is created • Event publisher now can login to the website
-----------------	--

Use Case ID	EP-02
Use Case Name	Log In
Primary Actor/s	Event Publisher
Description	The event publisher logs in using email and password
Pre-Conditions	The event publisher must already be registered
Main Scenario	<ol style="list-style-type: none"> 1. Visit log in page 2. Enter details 3. Click log in
Exceptions	<ul style="list-style-type: none"> • Incorrect email or password • Account is inactive
Post-Conditions	Event publisher accesses their dashboard

Use Case ID	EP-03
Use Case Name	Log Out
Primary Actor/s	Event Publisher
Description	The event publisher logs out from the system
Pre-Conditions	The event publisher must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Click log out button 2. Session ends
Exceptions	Session timeout or error
Post-Conditions	<ul style="list-style-type: none"> • Sponsor is logged out • Session is /terminated securely

Use Case ID	EP-04
Use Case Name	Create Publisher Profile

Primary Actor/s	Event Publisher
Description	The event publisher sets up their profile by entering club/union logo, description, links etc.
Pre-Conditions	The event publisher must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Navigate to profile section 2. Enter details 3. Upload logo, contact info and bank details etc. 4. Save profile
Include	EP-05: Edit Publisher Profile
Exceptions	<ul style="list-style-type: none"> • Missing mandatory fields • Invalid URL or image file
Post-Conditions	Event publisher profile is created and visible to others

Use Case ID	EP-05
Use Case Name	Edit Publisher Profile
Primary Actor/s	Event Publisher
Description	The event publisher edits previously entered profile information
Pre-Conditions	Profile must already exist
Main Scenario	<ol style="list-style-type: none"> 1. Go to profile page 2. Update fields 3. Save Changes
Exceptions	Invalid or incomplete data entered
Post-Conditions	Event publisher profile is updated successfully
Include	<>include>> Create Publisher Profile – because editing requires a profile to have already been created

Use Case ID	EP-06
Use Case Name	Event Publishing
Primary Actor/s	Event Publisher

Description	The event publisher publishes their upcoming events
Pre-Conditions	The event publisher must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Navigate to profile section 2. Enter upcoming event details 3. Save form
Extend	EP-07: Event Boosting
Extend	EP-08: Ticket Selling
Extend	EP-09: Requesting Volunteers
Extend	EP-10: Requesting Sponsors
Extend	EP-11: Requesting Donations
Extend	EP-12: Event History Management
Exceptions	<ul style="list-style-type: none"> • Missing mandatory fields • Invalid URL or image file
Post-Conditions	Upcoming events are visible to others

Use Case ID	EP-07
Use Case Name	Event Boosting
Primary Actor/s	Event Publisher
Description	Allows publishers to promote their events for greater visibility
Pre-Conditions	User has published at least one event
Main Scenario	<ol style="list-style-type: none"> 1. User selects "Boost Event" from their event management dashboard 2. System displays boosting options (duration, target audience, budget) 3. User selects boosting parameters 4. System calculates cost and displays payment requirements 5. User confirms and completes payment 6. System applies boosting to selected event
Extend	EP-13: Make Payment

Exceptions	<ul style="list-style-type: none"> • If payment fails, system returns to step 4 • If event doesn't meet boosting criteria, system displays requirements
Post-Conditions	Event receives enhanced visibility according to boosting parameters

Use Case ID	EP-08
Use Case Name	Ticket Selling
Primary Actor/s	Event Publisher
Description	Allows publishers to manage ticket sales for their events
Pre-Conditions	Event is published with ticket sales enabled
Main Scenario	<ol style="list-style-type: none"> 1. User accesses event management dashboard 2. User selects "Ticket Sales" for specific event 3. System displays current sales metrics and management options 4. User can: <ol style="list-style-type: none"> a. Adjust ticket prices/quantities b. Create discount codes c. View sales reports 5. System saves all changes
Exceptions	If changes violate platform policies, system displays warning
Post-Conditions	Ticket sales configuration is updated

Use Case ID	EP-09
Use Case Name	Requesting Volunteers
Primary Actor/s	Event Publisher
Description	Allows publishers to recruit volunteers for their events
Pre-Conditions	Event is published

Main Scenario	<ol style="list-style-type: none"> 1. User selects "Request Volunteers" for an event 2. System displays volunteer request form 3. User enters: <ol style="list-style-type: none"> a. Number of volunteers needed b. Roles/responsibilities c. Requirements/qualifications d. Benefits 4. System publishes volunteer opportunity 5. System notifies potential volunteers
Exceptions	If requirements are incomplete, system prompts for more information
Post-Conditions	Volunteer request is active and visible to users

Use Case ID	EP-10
Use Case Name	Requesting Sponsors
Primary Actor/s	Event Publisher
Description	Allows publishers to seek sponsors for their events
Pre-Conditions	Event is published
Main Scenario	<ol style="list-style-type: none"> 1. User selects "Request Sponsors" for an event 2. System displays sponsorship proposal form 3. User enters: <ol style="list-style-type: none"> a. Sponsorship levels b. Benefits for sponsors c. Contact information 4. System publishes sponsorship opportunity 5. System notifies potential sponsors

Exceptions	If proposal is incomplete, system prompts for more information
Post-Conditions	Sponsorship request is active and visible to potential sponsors

Use Case ID	EP-11
Use Case Name	Requesting Donations
Primary Actor/s	Event Publisher
Description	Allows publishers to collect donations for their events/causes
Pre-Conditions	Event is published
Main Scenario	<ol style="list-style-type: none"> 1. User selects "Request Donations" for an event 2. System displays donation configuration form 3. User enters: <ol style="list-style-type: none"> a. Donation purpose b. Target amount c. Suggested donation levels 4. System enables donation functionality for the event
Exceptions	If purpose description is insufficient, system prompts for more details
Post-Conditions	Event page shows donation options

Use Case ID	EP-12
Use Case Name	Event History Management
Primary Actor/s	Event Publisher
Description	Allows publishers to track, analyze, and manage their past and ongoing events.
Pre-Conditions	<ul style="list-style-type: none"> • User is logged in with a publisher profile. • Has created at least one event.

Main Scenario	<ol style="list-style-type: none"> 1. Publisher navigates to "My Events" dashboard. 2. System displays: <ol style="list-style-type: none"> a. Past Events (completed) b. Upcoming Events (scheduled) c. Draft Events (unpublished) d. Cancelled Events 2. Publisher selects an event to: <ol style="list-style-type: none"> 1. View detailed analytics (attendees, ticket sales, revenue). 2. Export event data (CSV/PDF reports). 3. Clone/Reuse event setup for future events. 4. Edit event details (if unpublished). 3. System saves any changes made.
Exceptions	<ul style="list-style-type: none"> • If no events exist, the system suggests "Create New Event" with a quick-start guide. • If publisher requests deletion, system archives the event instead
Post-Conditions	<ul style="list-style-type: none"> • Publisher gains insights into event performance. • Event data is preserved for future reference.

Use Case ID	EP-13
Use Case Name	Make Payment
Primary Actor/s	Event Publisher
Description	Handles financial transactions initiated by publishers (e.g., event boosting, fees, refunds).
Pre-Conditions	<ul style="list-style-type: none"> • Publisher is logged in. • Payment method is registered (credit card, PayPal, etc.)

Main Scenario	<ol style="list-style-type: none"> 1. Payment Trigger: <ol style="list-style-type: none"> a. Publisher initiates a paid action (e.g., "Boost Event," "Pay Platform Fee"). 2. System displays: <ol style="list-style-type: none"> a. Payment amount (with breakdown of costs/taxes). b. Selected payment method (with option to change). 3. Publisher confirms payment. 4. System processes payment: <ol style="list-style-type: none"> a. Charges the publisher's account. b. Generates a receipt. 5. System notifies publisher of successful payment.
Extend	EP-07: Event Boosting
Exceptions	<ul style="list-style-type: none"> • Insufficient Funds: System prompts to update payment method. • Partial Refunds: Publisher can issue partial refunds to attendees (e.g., for cancelled events). • Failed Transaction: System retries once, then aborts and logs the error.
Post-Conditions	<ul style="list-style-type: none"> • Payment is recorded in the publisher's transaction history. • Paid services (e.g., event boosting) are activated.

8.4.5. Sponsor

Use Case ID	SPON-01
Use Case Name	Register
Primary Actor/s	Sponsor/Donator
Description	The sponsor registers an account by providing organization/business details
Pre-Conditions	The sponsor has not already registered with the same email or business ID

Main Scenario	<ol style="list-style-type: none"> 1. Open registration page 2. Fill in details (company name, email, password, business ID etc.) 3. Submit Form
Include	SPON-01: Login
Exceptions	<ul style="list-style-type: none"> • If email already exists, show error • If required fields are missing, display validation alerts
Post-Conditions	<ul style="list-style-type: none"> • Sponsor account is created • Sponsor can now log in to the system

Use Case ID	SPON-02
Use Case Name	Login
Primary Actor/s	Sponsor/Donator
Description	Sponsor logs in using their email and password
Pre-Conditions	Sponsor must already be registered
Main Scenario	Visit login page Enter details Click login
Exceptions	<ul style="list-style-type: none"> • Incorrect email or password • Account is inactive
Post-Conditions	Sponsor accesses their dashboard

Use Case ID	SPON-03
Use Case Name	Logout
Primary Actor/s	Sponsor/Donator
Description	Sponsor logs out of the system
Pre-Conditions	Sponsor must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Click logout button 2. Session ends
Exceptions	Session timeout or error
Post-Conditions	<ul style="list-style-type: none"> • Sponsor is logged out • Session is terminated securely
Include	<>include>> Login – because after logout, the system may redirect or require login to resume

Use Case ID	SPON-04
Use Case Name	Create Profile
Primary Actor/s	Sponsor/Donator
Description	Sponsor sets up their profile by entering the company logo, description, links, etc.
Pre-Conditions	Sponsor must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Navigate to profile section 2. Enter details 3. Upload logo and contact info 4. Save profile
Include	SPON-05: Edit profile
Exceptions	<ul style="list-style-type: none"> • Missing mandatory fields • Invalid URL or image file
Post-Conditions	Sponsor profile is created and visible to event organizers

Use Case ID	SPON-05
Use Case Name	Edit Profile
Primary Actor/s	Sponsor/Donator
Description	Sponsor edits previously entered profile information
Pre-Conditions	Profile must already exist
Main Scenario	<ol style="list-style-type: none"> 1. Go to profile page 2. Update fields 3. Save changes
Exceptions	Invalid or incomplete data entered
Post-Conditions	Profile is updated successfully
Include	<<include>> Create Profile – because editing requires a profile to have already been created

Use Case ID	SPON-06
Use Case Name	Send Message to Organizers

Primary Actor/s	Sponsor/Donator
Description	Allows the sponsor to communicate with event organizers via internal messaging
Pre-Conditions	Sponsor must be logged in. Organizer must be associated with an event
Main Scenario	<ol style="list-style-type: none"> 1. View organizer details from an event 2. Click "Send Message" 3. Type message and send
Exceptions	<ul style="list-style-type: none"> ● Organizer is not available ● Message content is empty
Post-Conditions	<ul style="list-style-type: none"> ● Message is delivered to organizer's inbox ● Sponsor sees confirmation or thread

Use Case ID	SPON-07
Use Case Name	View Event Details
Primary Actor/s	Sponsor/Donator
Description	Displays available university events and their sponsorship opportunities
Pre-Conditions	Sponsor must be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Navigate to event listing 2. Click an event title 3. View date, venue, audience, sponsorship needs
Exceptions	<ul style="list-style-type: none"> ● No events found ● Event details page fails to load
Post-Conditions	<ul style="list-style-type: none"> ● Event information is displayed ● Sponsor may choose to take further action (message, donate, etc.)

Use Case ID	SPON-08
Use Case Name	Sponsor Event Post
Primary Actor/s	Sponsor/Donator
Description	Allows sponsors to promote their brand through featured posts on the event page

Pre-Conditions	Sponsor must have a complete profile and be logged in
Main Scenario	<ol style="list-style-type: none"> 1. Select an event 2. Click “Sponsor Post” 3. Submit promotional content 4. Admin approval (if required)
Exceptions	<ul style="list-style-type: none"> • Content violates guidelines • Post not approved
Post-Conditions	Sponsored post becomes visible on the event page

Use Case ID	SPON-09
Use Case Name	Set Sponsorship Conditions
Primary Actor/s	Sponsor/Donator
Description	Allows sponsors to propose or customize the terms for sponsoring an event
Pre-Conditions	Sponsor must be logged in and viewing a specific event
Main Scenario	<ol style="list-style-type: none"> 1. Select an event 2. Click “Set Conditions” 3. Input budget, expectations, terms 4. Submit for review
Exceptions	<ul style="list-style-type: none"> • Terms conflict with organizer policies • Incomplete input
Post-Conditions	Conditions are saved and shared with the organizer

Use Case ID	SPON-10
Use Case Name	Make Donations
Primary Actor/s	Sponsor/Donator
Description	Sponsor provides either monetary or in-kind contributions to support an event.
Pre-Conditions	Sponsors must be logged in and have access to event donation options.

Main Scenario	<ol style="list-style-type: none"> 1. Visit the event page 2. Click “Donate” 3. Choose donation type (money/in-kind) 4. Enter details and confirm
Exceptions	<ul style="list-style-type: none"> ● Invalid payment method ● Donation exceeds limit or is not permitted
Post-Conditions	<ul style="list-style-type: none"> ● Donation is recorded in system ● Organizer is notified ● Sponsor receives confirmation receipt

Use Case ID	SPON-11
Use Case Name	Maintain Sponsorship History
Primary Actor/s	Sponsor/Donator
Description	Displays a log of all past events sponsored or donated to by the sponsor
Pre-Conditions	A sponsor profile must be created and used in sponsorship/donation actions.
Main Scenario	<ol style="list-style-type: none"> 1. Navigate to “Sponsorship History” 2. View list of events with sponsorship details 3. Click on any event for more info
Exceptions	<ul style="list-style-type: none"> ● No past records ● Data not available or temporarily hidden
Post-Conditions	Sponsor can analyze past support and involvement in university events
Include	<<include>> Create Profile – since history tracking is only available to registered profiles

9. Technologies to be used

Backend Development

- PHP
- MySQL



Frontend Development

- HTML
- CSS
- JavaScript



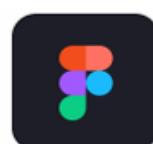
Diagrams and Modeling

- Draw.io



UI/UX and Prototyping

- Figma



Documentation

- Google Docs
- Google Slides



IDE

- VS Code



Version Control

- GitHub



Group Collaboration and Communication

- Whatsapp
- Google Meet



Payment Gateway

- PayHere



10. Project Timeline

Our project, **University Event Management System**, is planned to span the entire academic year and follows the Iterative Waterfall Model to ensure a structured, systematic, and quality-driven development process. The project begins with a comprehensive requirements gathering phase, followed by detailed system design, implementation, and testing phase. By the conclusion of the project, we aim to deliver a fully functional, well-tested, and well-documented event management platform tailored for university needs.

Below is the Gantt chart that illustrates our project's timeline and phase-wise task distribution.

Task	June				July				August				September				October				November				December				January				February				March				April			
	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4				
Project Selection	Y																																											
Requirement gathering			Y																																									
Requirement Analysis		Y	Y																																									
Project Proposal			Y	Y																																								
UML Diagrams					Y	Y	Y	Y																																				
SRS Documentation						Y	Y	Y	Y	Y	Y	Y																																
UI Design							Y	Y	Y	Y	Y	Y	Y																															
Frontend Development								Y	Y	Y	Y	Y	Y	Y																														
Backend Development									Y	Y	Y	Y	Y	Y	Y																													
System Testing										Y	Y	Y	Y	Y	Y																													
Deployment											Y	Y	Y	Y	Y																													
Maintenance												Y	Y	Y	Y																													

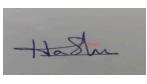
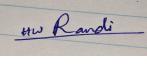
11. Declaration

We, as the members of the project titled **UniPulse**, hereby certify that we will carry out this project in accordance with the guidelines and instructions provided by the course coordinators and supervisors.

We further declare that, to the best of our knowledge and belief, this project:

- Does not contain any material that has been previously submitted for any degree or diploma at any university or institution without proper acknowledgment.
- Does not include any content that has been previously published or written by another individual or ourselves, except where appropriate references have been clearly cited within the text.

We accept full responsibility for the integrity and originality of the work presented in this project and commit to maintaining academic honesty throughout its development and submission.

Name of the student	Index Number	Signature
1. R.P.M.A Perera	23001445	
2. V.V Wakishta	23002141	
3. H.G Suraweera	23002001	
4. H.W Randi	23001585	
5. S. Kokulan	23000945	