Faculty of Computing and Technology

University of Kelaniya

Event Management System

Software Requirement Specification

System Analysis and Design (SWST 32012)

Submitted by:

CT/2019/003

CT/2019/071

CT/2019/004

CT/2019/032

CT/2019/056

CT/2019/026

2024/06/28

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to outline the software requirements for the Event Management System (EMS). This system is designed to streamline the processes involved in organizing, registering, and managing events. The EMS aims to provide a comprehensive suite of tools that cater to the needs of event organizers, attendees, and sponsors, ensuring efficient event management and a seamless experience for all stakeholders.

1.2 Intended Audience and Reading Suggestions

This document is intended for various stakeholders involved in the development and use of the EMS. Each section of the document provides specific information relevant to different roles:

- Project Managers: Project managers should focus on understanding the scope, objectives, and detailed requirements outlined in this document. This will help in planning, scheduling, and resource allocation to ensure the project meets its goals.
- Developers: Developers should use this document to gain a detailed understanding of the features and functionalities that need to be implemented. Each use case and requirement provide the necessary detail to guide the development process.
- Testers: Testers should refer to this document to design and execute test cases. The detailed requirements and use cases will help in creating comprehensive test plans to ensure the system functions as expected.
- Stakeholders: Stakeholders, including clients and end-users, should review this document to verify that the system meets their needs and expectations. Their feedback will be crucial in validating the requirements and making any necessary adjustments.

1.3 Product Scope

The Event Management System is a comprehensive platform designed to facilitate the efficient management of events. It encompasses several key modules, each focusing on a specific aspect of event management. The primary objectives of the EMS are to enhance the efficiency of event

organization, improve the attendee experience, and provide robust tools for sponsor management. The system's scope includes the following key functionalities:

1. Event Registration and Ticketing:
    - Event Organizers: Allows organizers to create and customize registration forms, set up various ticket types, and manage attendee registrations. The system automates the confirmation process and provides detailed reports.
    - Attendees: Enables attendees to register online, receive electronic tickets, and manage their event itineraries through a user-friendly interface.
2. Session and Agenda Management:
    - Event Organizers: Facilitates the creation and management of event schedules, assignment of speakers, and real-time updates to the agenda. Organizers can also manage the collection of presentation materials.
    - Attendees: Provides attendees with detailed event schedules, notifications about upcoming sessions, and access to post-event presentation materials.
3. Sponsor Management:
    - Event Organizers: Helps organizers manage sponsor profiles, create and manage sponsorship packages, and track sponsor commitments and deliverables. The system ensures that sponsors have access to all necessary information and tools.
    - Sponsors: Allows sponsors to access information about sponsorship packages, submit promotional materials, and track the status of their commitments.
4. Payment and Financial Management:
    - Event Organizers: Integrates secure payment gateways for ticket sales, processes payment transactions, and generates financial reports. The system also manages invoices and ensures compliance with financial regulations.
    - Financial Team: Provides tools for monitoring payment status, resolving payment issues, and managing the financial aspects of event organization.

2. Overall Description of the System

2.1 Product Functions

The system is designed to manage various aspects of event management, offering a comprehensive set of functionalities that cater to different stakeholders including event organizers, attendees, sponsors, and the registration team. The primary functions of the system include:

- Event Registration and Ticketing: This module allows event organizers to create and customize registration forms, set up different ticket types, and automate the sending of confirmation emails. Attendees can register online, receive electronic tickets, and manage their event itineraries.
- Session and Agenda Management: This function enables event organizers to create and manage event schedules, assign speakers, and allow real-time updates to the agenda. Attendees can view detailed event schedules and receive notifications about upcoming sessions.
- Sponsor Management: This module helps event organizers manage sponsor profiles and contact information, create and manage sponsorship packages, and track sponsor commitments and deliverables. Sponsors can access detailed information about sponsorship packages and submit promotional materials.
- Payment and Financial Management: This function integrates secure payment gateways for ticket sales, processes payment transactions, generates financial reports, manages invoices and payments, and ensures compliance with financial regulations

2.2 System Modelling

The system will be modeled using use cases to describe interactions between users and the system. Each use case will detail the functional requirements and processes involved. The use case modeling approach will help in understanding the specific needs of different user roles and ensure that all necessary functionalities are captured and implemented correctly.

Detailed Description of Each Module:

1. **Event Registration and Ticketing**

- **Event Organizers**:
    - Create and Customize Event Registration Forms: Allows organizers to input event details, designs, and styling preferences to generate personalized registration forms.
    - Set Up Different Ticket Types: Organizers can define ticket names, prices, and specific attributes to create various ticket categories.
    - Automate Confirmation Emails: Automatically sends confirmation emails to registrants upon successful registration.
- **Registration Team:**
    - Manage Attendee Lists and Registrations: Provides a dashboard to manage registrations and maintain up-to-date attendee lists.
    - Attendee Inquiry and Support Handling: Manages inquiries and support requests from attendees.
    - Report Generation and Analysis: Generates comprehensive reports and provides tools for analyzing registration data.
- **Attendees:**
    - Online Event Registration: Provides a user-friendly interface for attendees to register for events.
    - Access and Update Personal Information: Allows attendees to securely update their personal information.
    - Electronic Ticket Issuance: Generates and sends electronic tickets to attendees.
    - Event Itinerary Management: Enables attendees to view and manage their event schedules.

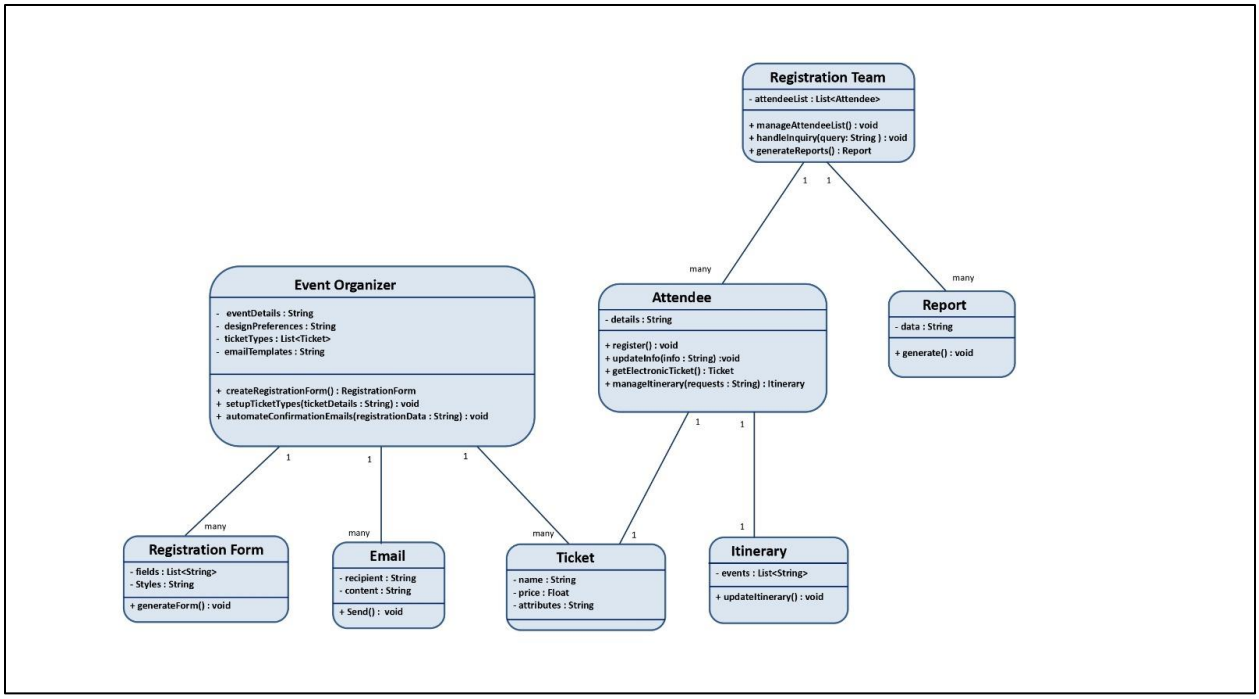

Figure 1: class diagram for event registration and ticketing

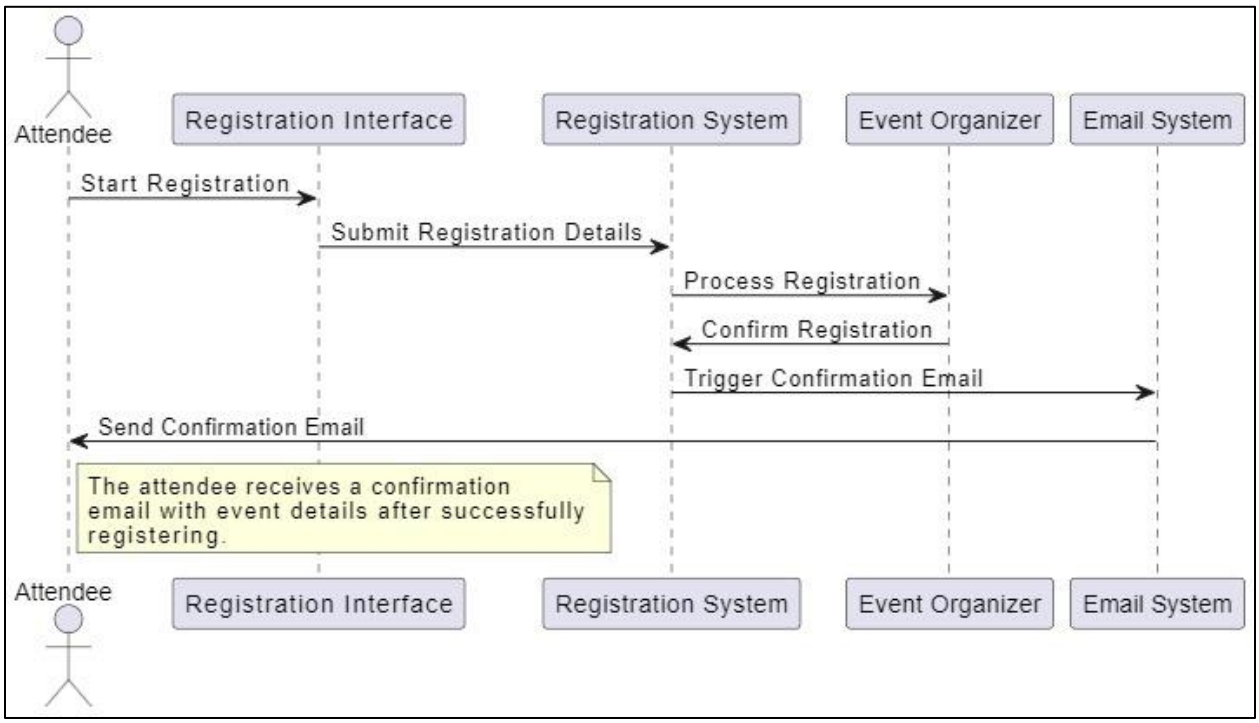

Figure 2: Sequence diagram for event registration and ticketing

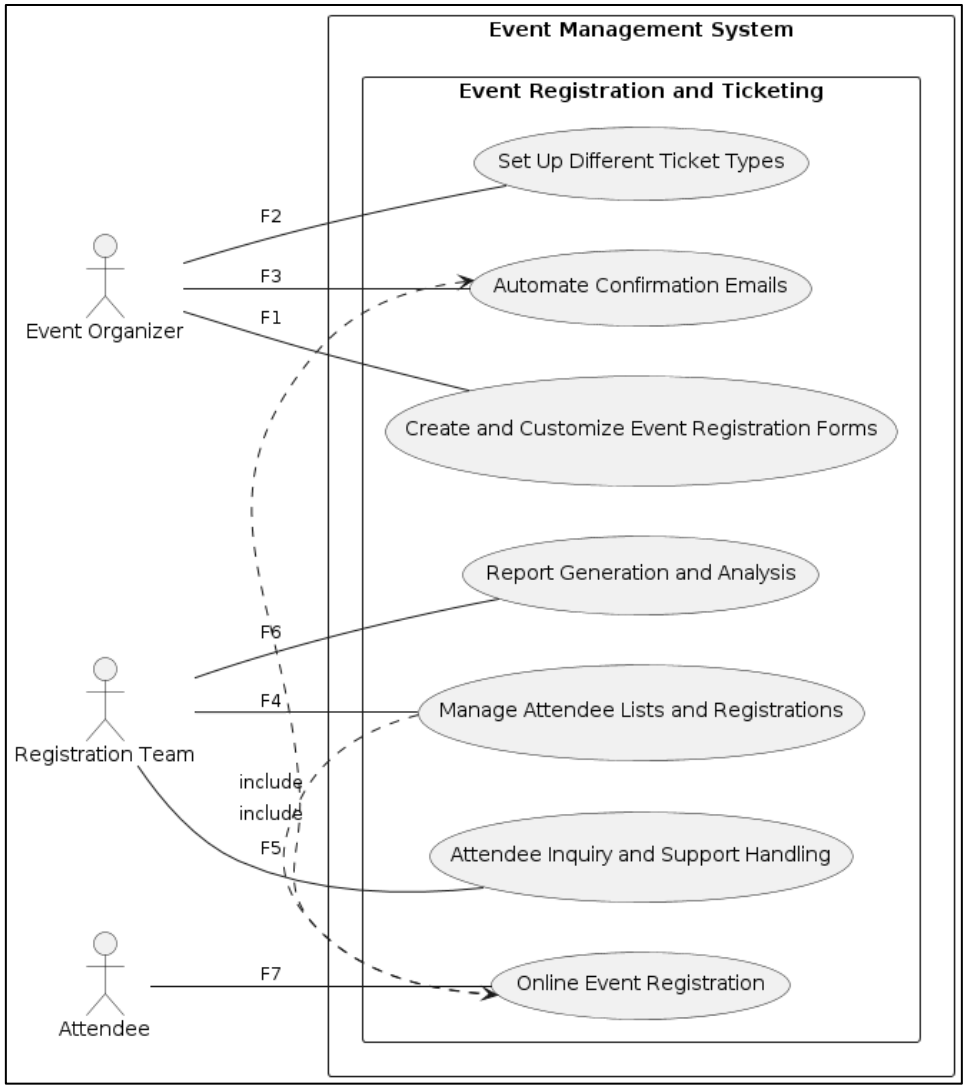

Figure 3: Use case diagram for Event registration and ticketing

Use case Number	**01**
Use case name	Event Registration and Ticketing
Actor	Attendee
Preconditions	The attendee needs to be authenticated and the event must be available for registration.

Activities	1. Attendee logs into the web application. 2. Attendee navigates to the event registration page. 3. Attendees select the event they wish to register for. 4. The system presents the registration form and ticket options. 5. Attendee fills out the registration form and selects a ticket type. 6. The system processes the registration details and ticket selection. 7. Attendee confirms the registration and proceeds to payment. 8. The system processes the payment via the integrated payment gateway. 9. Attendee receives an automated confirmation email with ticket details.
Alternatives	If the payment fails, the system prompts the attendee to retry or use a different payment method. If the event is sold out, the system notifies the attendee and provides an option to join a waiting list.
Postconditions	The attendee is successfully registered for the event. The system sends a confirmation email and issues an electronic ticket. The registration data is updated in the system.

Table 1

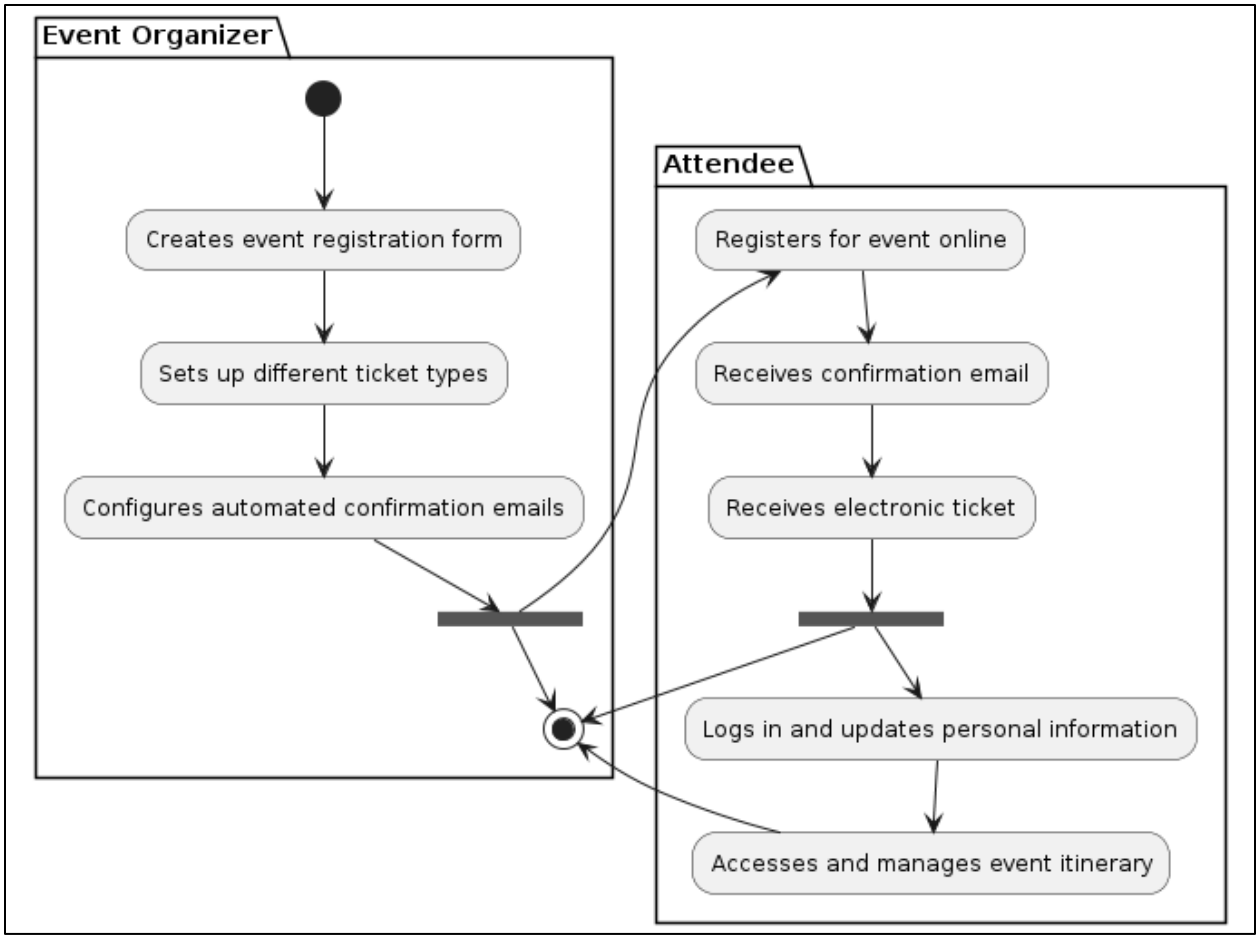

Figure 4: Activity diagram for event Registration and ticketing

2. Session and Agenda Management

- **Event organizers:**
    - Event Schedule Creation and Management: Tools for creating and organizing event schedules and agendas.
    - Assign Speakers and Presenters: Streamlines the assignment of speakers and presenters to sessions.
    - Allow Real-Time Updates to the Agenda: Facilitates real-time modifications to event agendas.
    - Presentation Material Collection and Management: Manages the collection and organization of session-related materials.

- Attendees:
    - Detailed Event Schedule Viewing: Provides detailed and navigable event schedules.
    - Upcoming Event Notifications: Sends timely alerts about upcoming sessions based on user preferences.
    - Post-Event Presentation Material Access: Allows retrieval of session materials after the event.

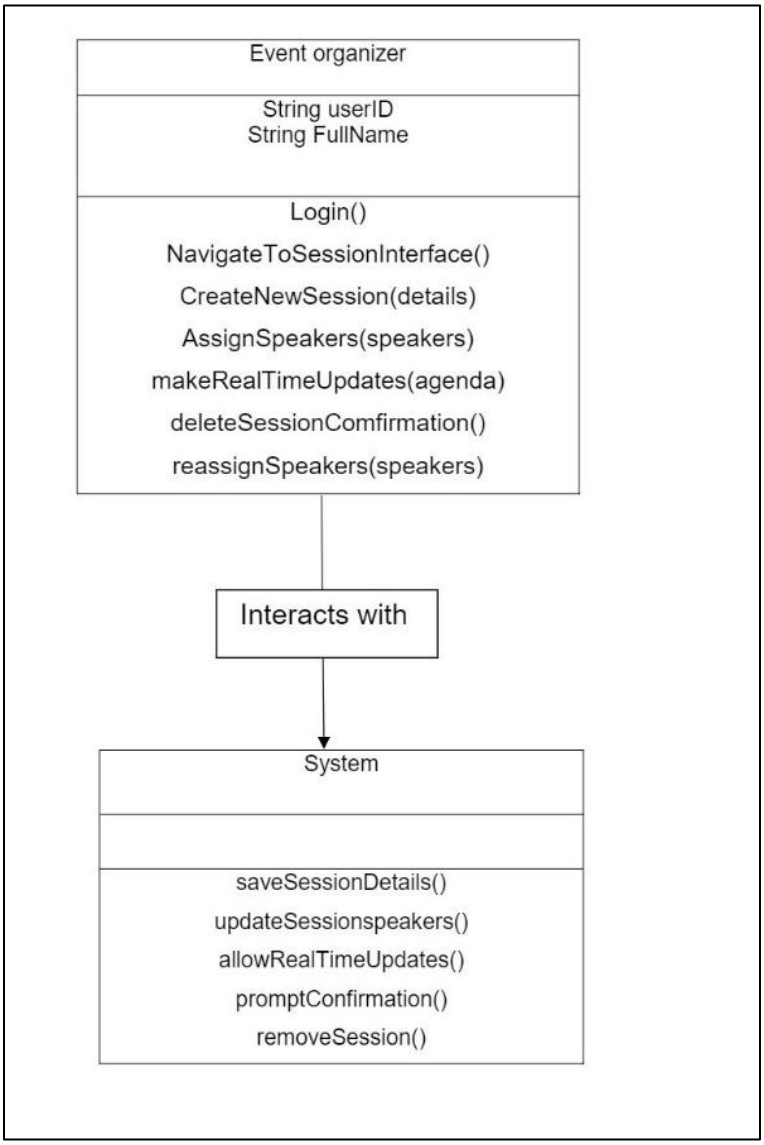

Figure 5:class diagram for 2 session and agenda Management

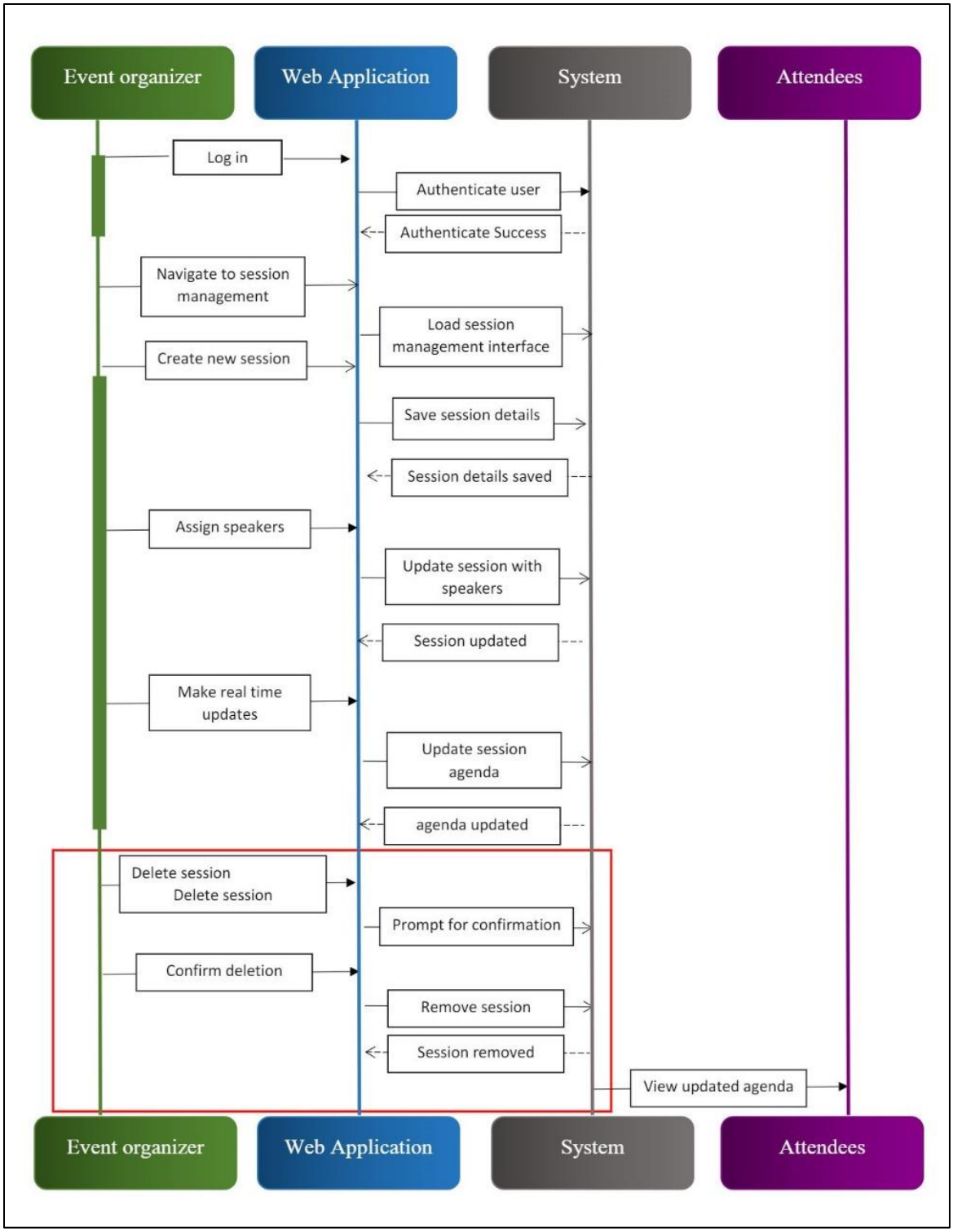

Figure 6: sequence diagram for session and agenda management

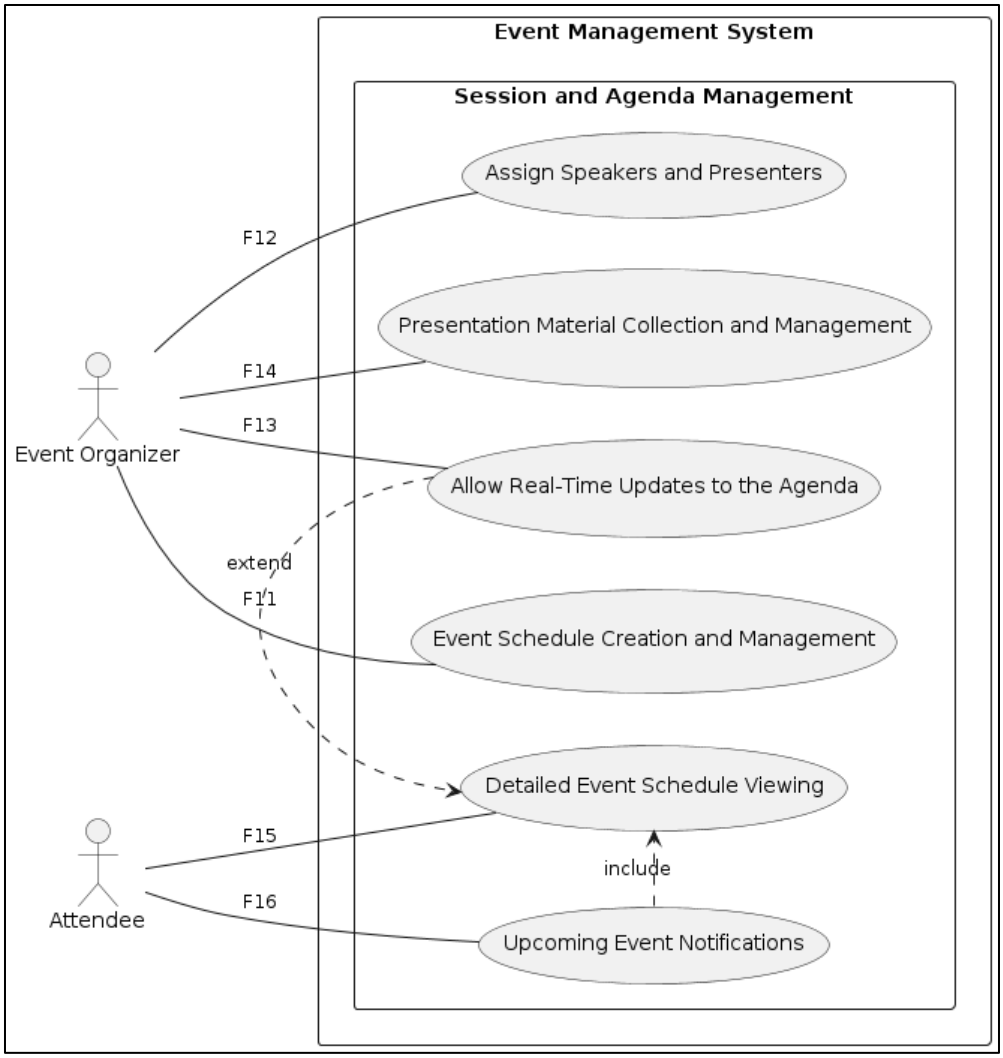

Figure 7: use case diagram for session and agenda management

Use case Number	**02**
Use case name	Session and Agenda Management
Actor	Event Organizer
Preconditions	The event organizer needs to be authenticated.

Activities	1. Event Organizer logs into the web application. 2. Event Organizer navigates to the session management interface. 3. Event Organizer creates a new session by entering session details (title, description, time, etc.). 4. The system saves the session details. 5. Event Organizer assigns speakers to the session. 6. The system updates the session with the assigned speakers. 7. Event Organizer makes real-time updates to the session agenda if needed. 8. The system allows real-time modifications and updates the agenda.
Alternatives	If the organizer wants to delete a session, the system prompts for confirmation and then removes the session if confirmed. If a speaker cancels, the organizer can reassign speakers or update the session details.
Postconditions	The event sessions are created and updated in the system. Attendees can view the updated session agenda in real-time.

Table 2

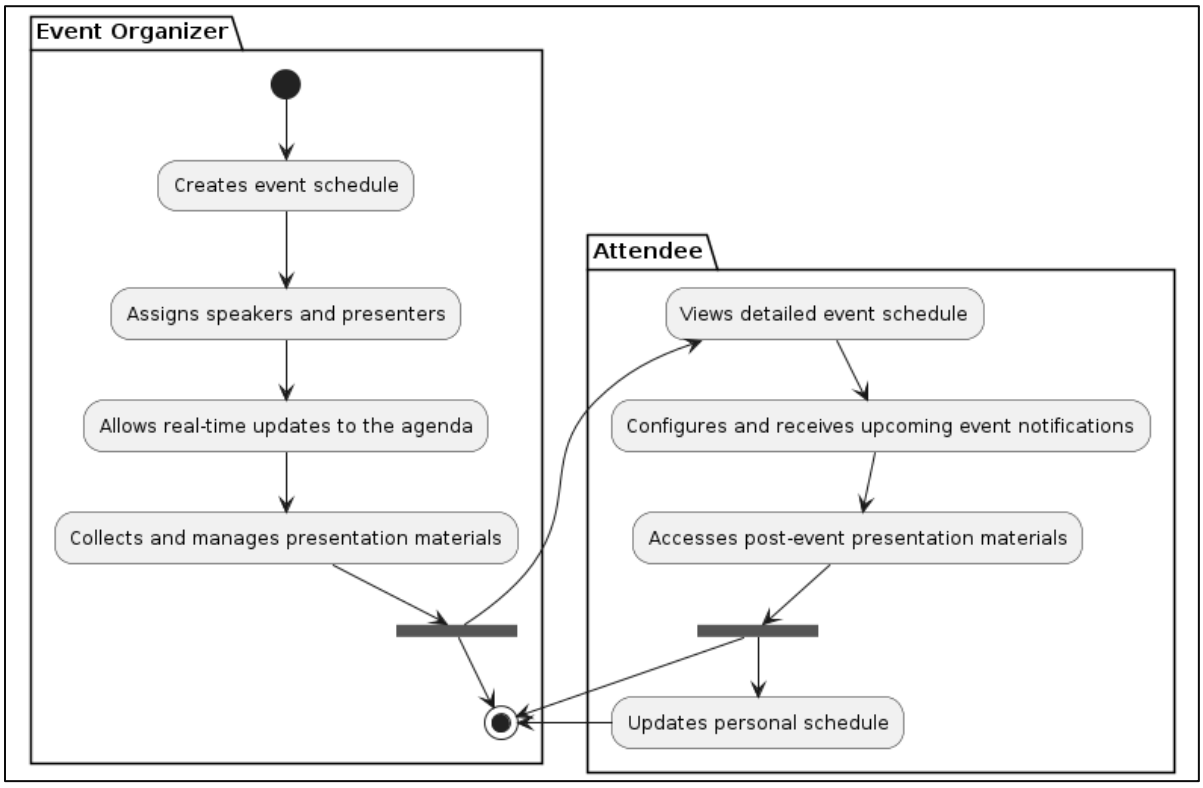

Figure 8: Activity diagram for session and agenda management

3. Sponsor management

- **Event organizers:**
    - Sponsor Profile and Contact Management: Tools for managing sponsor details and contact information.
    - Sponsorship Package Management: Platform for creating and managing diverse sponsorship packages.
    - Track Sponsor Commitments and Deliverables: Monitors and manages sponsor obligations and outcomes.
- **Sponsors:**
    - Sponsorship Package Information Access: Secure access to detailed information about sponsorship packages.
    - Materials and Logos Submission for Promotion: Platform for submitting promotional materials and logos.

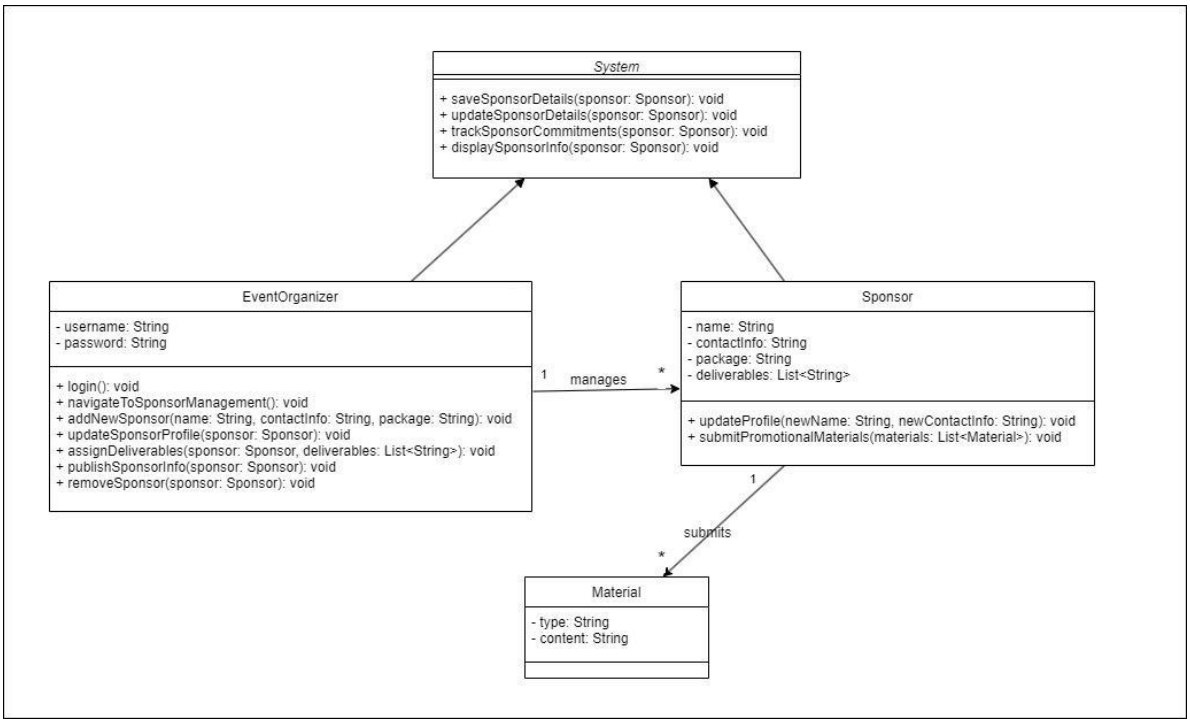

Figure 9: class diagram for sponsor management

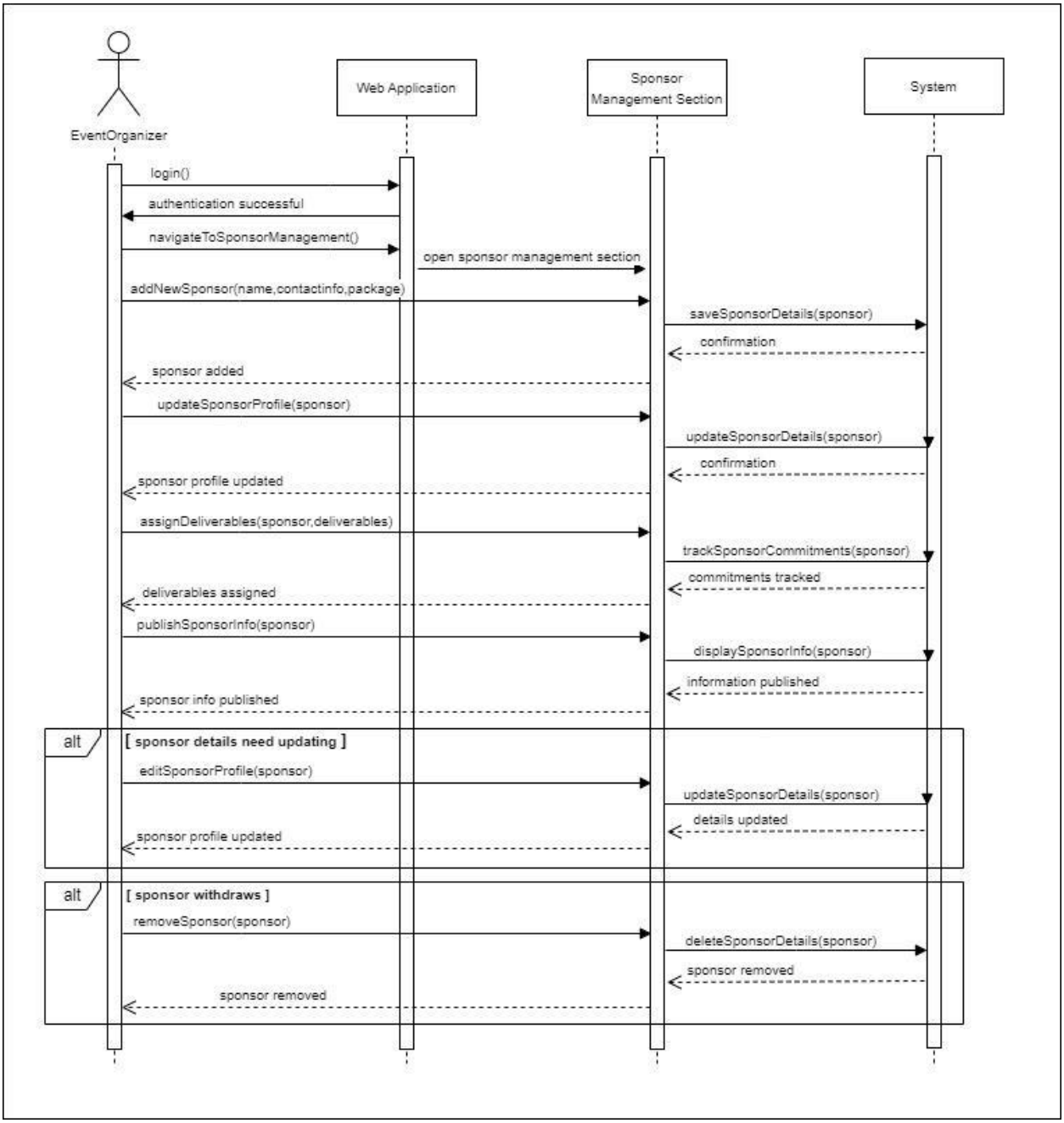

Figure 10: sequence diagram for sponsor management

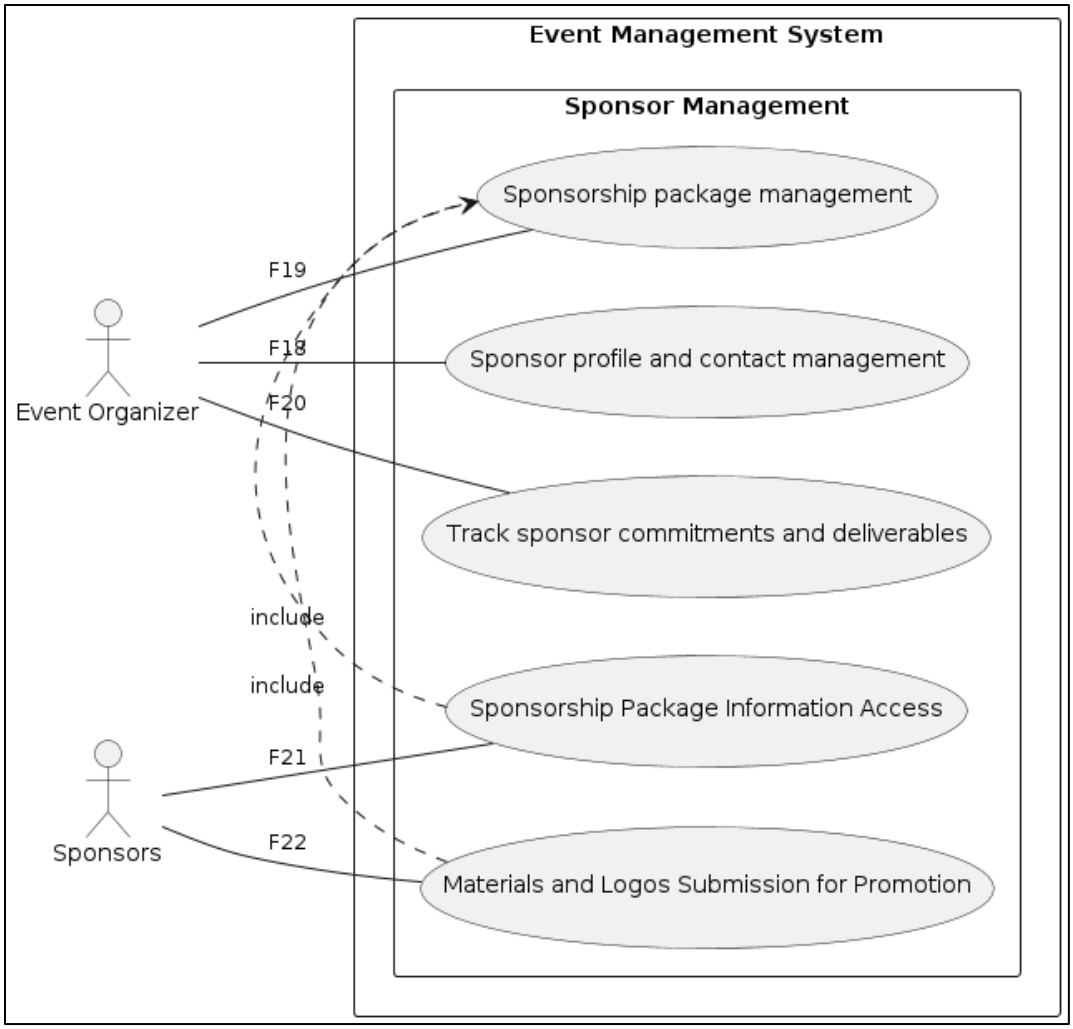

Figure 11: use case diagram for sponsor management

Use case Number	**03**
Use case name	Sponsor Management
Actor	Event Organizer
Preconditions	The event organizer needs to be authenticated.

Activities	1. Event Organizer logs into the web application. 2. Event Organizer navigates to the sponsor management section. 3. Event Organizer adds a new sponsor by entering sponsor details (name, contact information, sponsorship package, etc.). 4. The system saves the sponsor's details. 5. Event Organizer updates the sponsor's profile and assigns deliverables. 6. The system tracks sponsor commitments and deliverables. 7. Event Organizer publishes the sponsor's information and promotional materials. 8. The system displays the sponsor's information on the event website.
Alternatives	If a sponsor's details need updating, the organizer can edit the profile, and the system will update the information accordingly. If a sponsor withdraws, the organizer can remove the sponsor, and the system updates the sponsor list.
Postconditions	Sponsor profiles and contact information are managed and up to date. Sponsorship packages and commitments are tracked. Sponsor's promotional materials are published on the event website.

Table 3

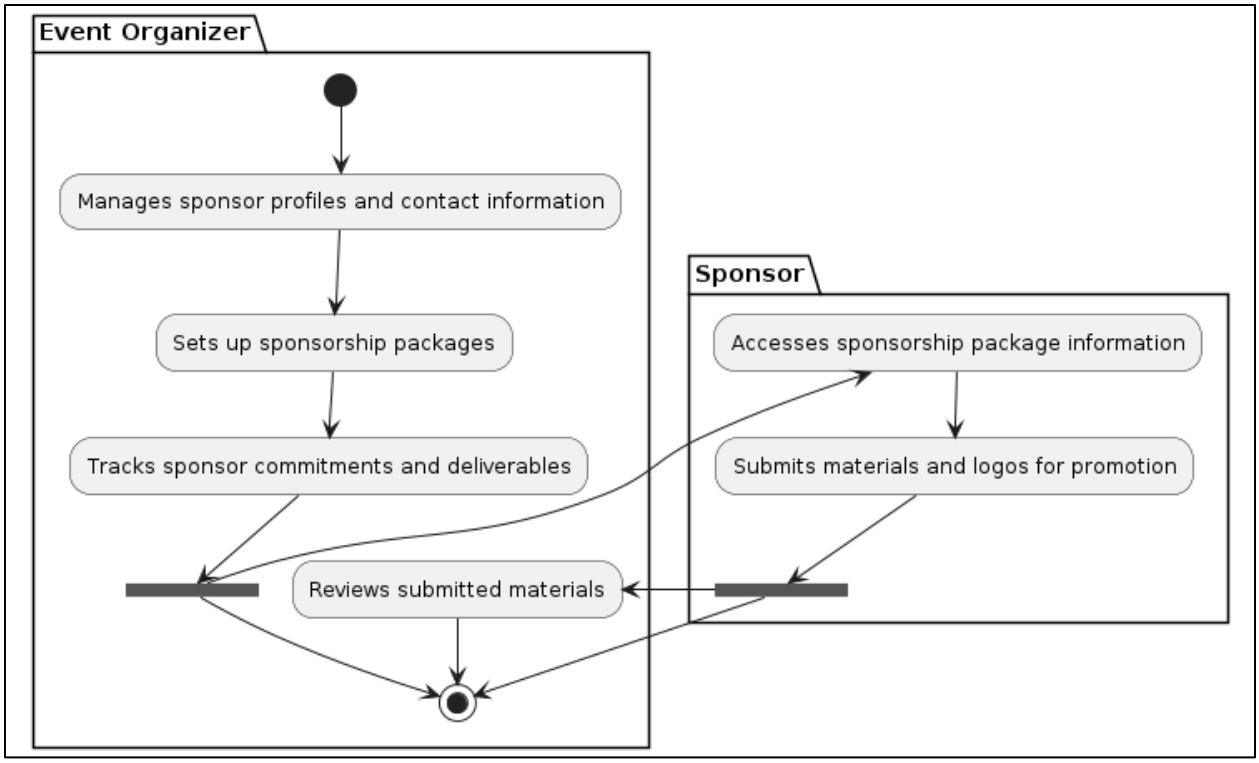

Figure 12: activity diagram for sponsor management

4. Payment and Financial Management

- **Event organizers:**
    - Integrate Secure Payment Gateways: Integration of secure payment channels for ticket sales.
    - Payment Transaction Processing: Manages transaction data and sales details in real-time.
- **Financial team:**
    - Financial Report Generation: Automates the creation of comprehensive financial reports.
    - Invoice and Payment Management: Streamlines the management of invoices and payments.
    - Ensure Compliance with Financial Regulations: Ensures adherence to financial regulations and standards.
- **Registration team:**
    - Payment Status Monitoring: Tracks and updates the status of payments for registrations.
    - Payment Issue Resolution: Dedicated support system for resolving payment-related issues.

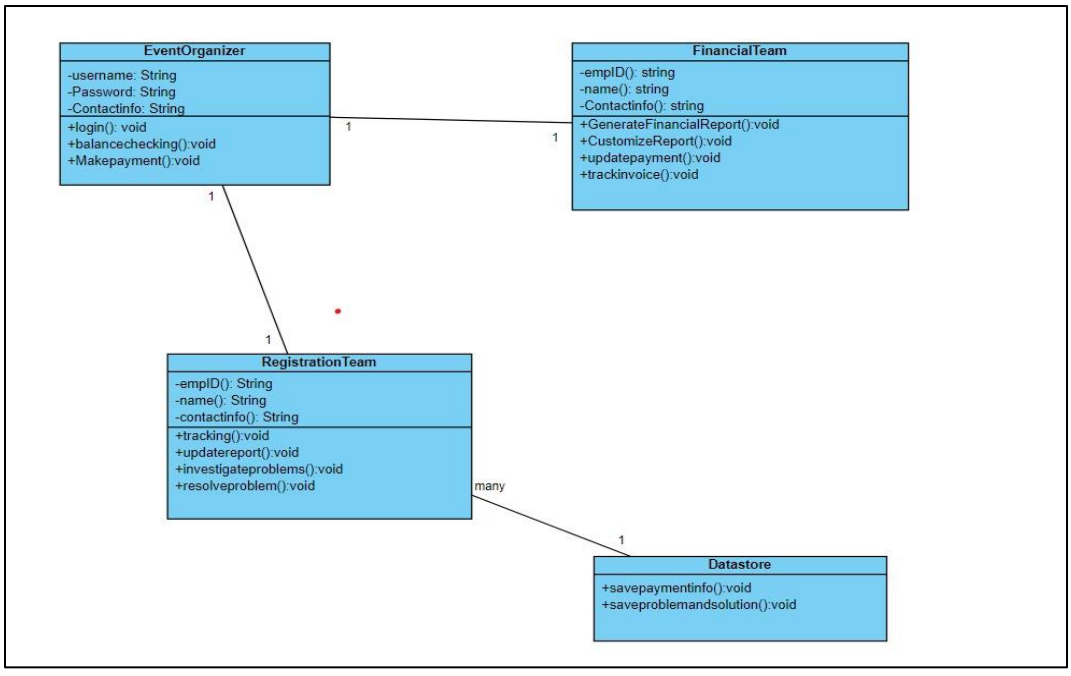

Figure 13: class diagram for payment and financial management

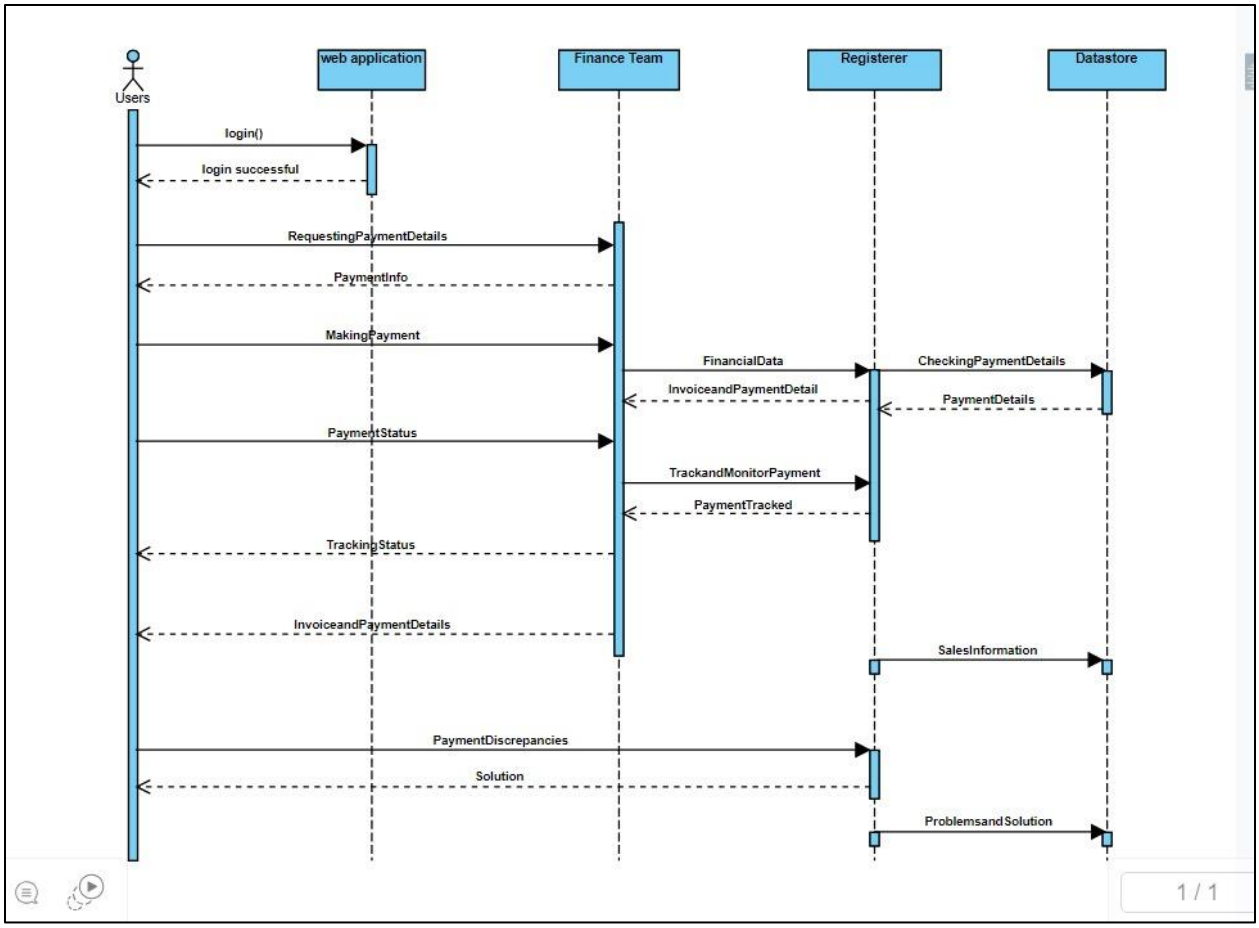

Figure 14: sequence diagram for Payment and Financial Management

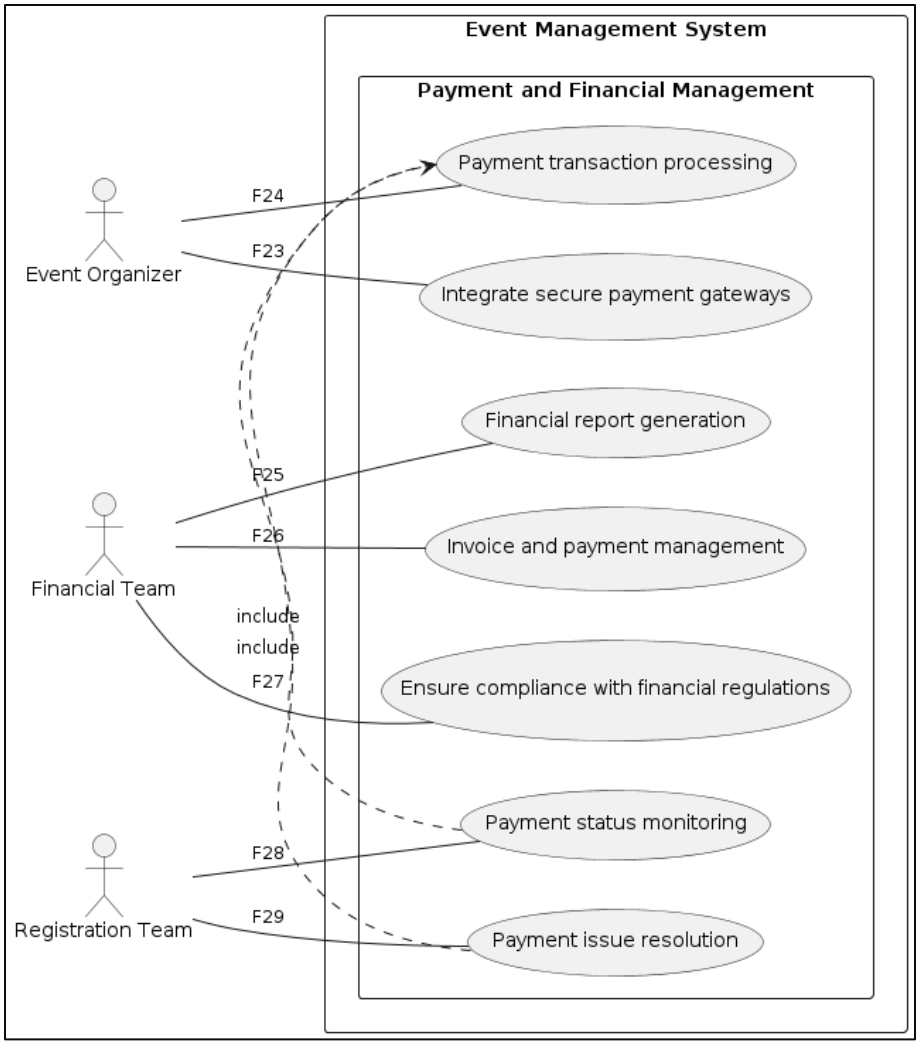

Figure 15: use case diagram for payment and financial management

Use case Number	**04**
Use case name	Payment and Financial Management
Actor	Financial Team
Preconditions	The financial team member needs to be authenticated.

Activities	1. Financial Team logs into the web application. 2. Financial Team navigates to the financial management section. 3. Financial Team integrates secure payment gateways for transactions. 4. The system verifies and integrates the payment gateways. 5. The Financial Team monitors payment transactions in real time. 6. The system provides a real-time overview of all financial transactions. 7. The Financial Team generates financial reports based on transaction data. 8. The system creates and displays comprehensive financial reports. 9. The Financial Team manages invoices and payments. 10. The system tracks invoice statuses and payment records.
Alternatives	If a payment gateway integration fails, the system prompts troubleshooting steps and re-integration. If a transaction fails, the financial team can investigate and resolve the issue, and the system updates the transaction status.
Postconditions	Secure payment gateways are integrated and functional. Real-time financial transactions are monitored and recorded. Financial reports are generated and available for analysis. Invoices and payment records are up-to-date and well-organized.

Table 4

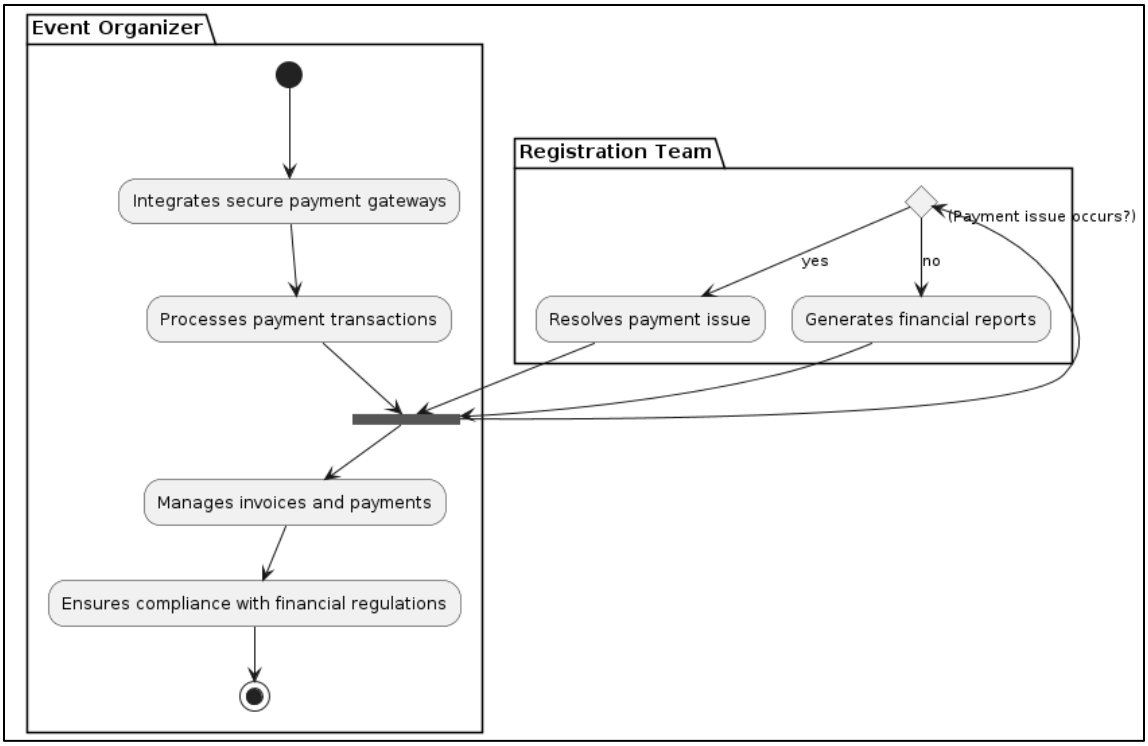

Figure 16: activity diagram for payment and financial management

DFD diagram for overall system

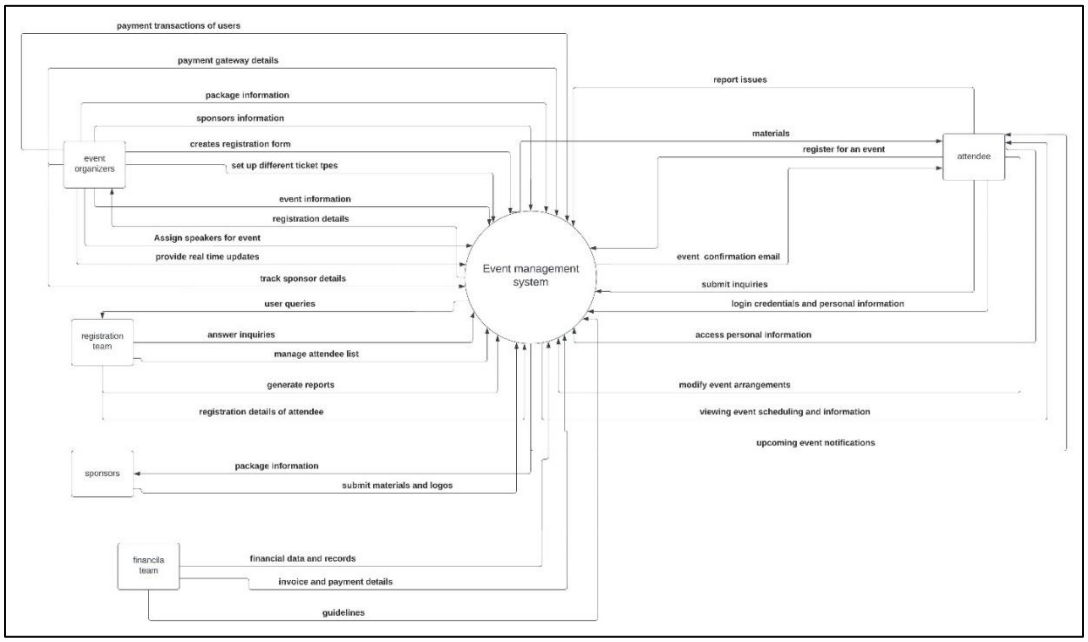

Figure 17: DFD diagram of the system

3. System Features

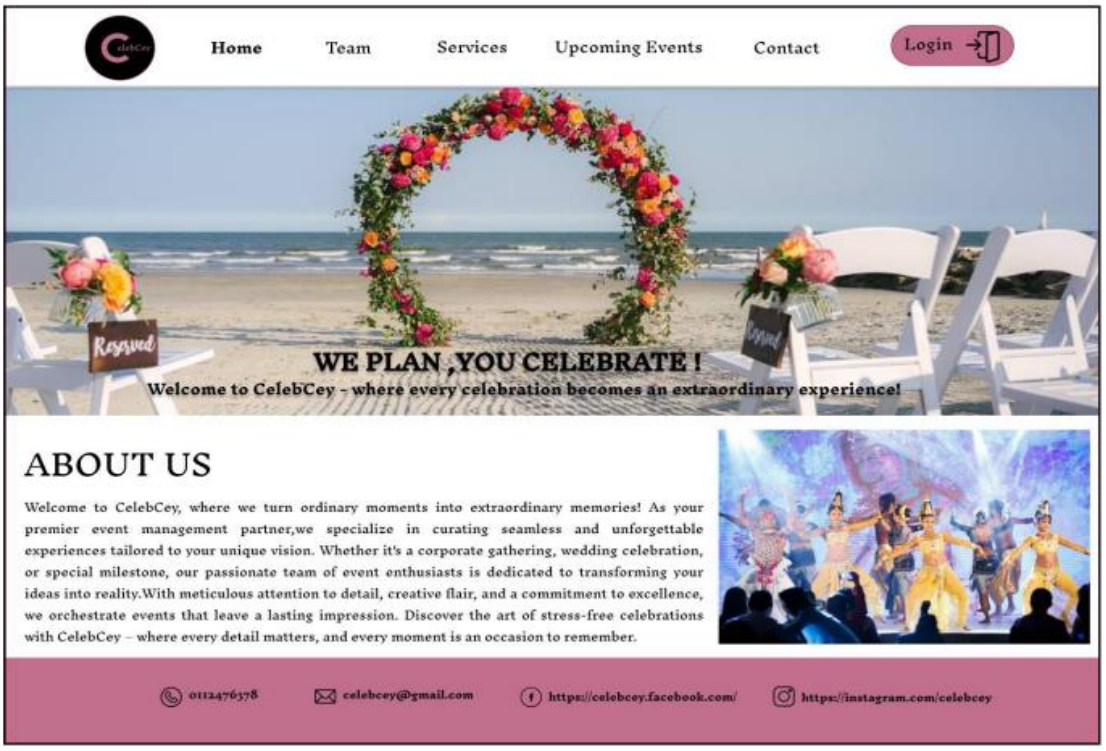

Figure 18: Home page

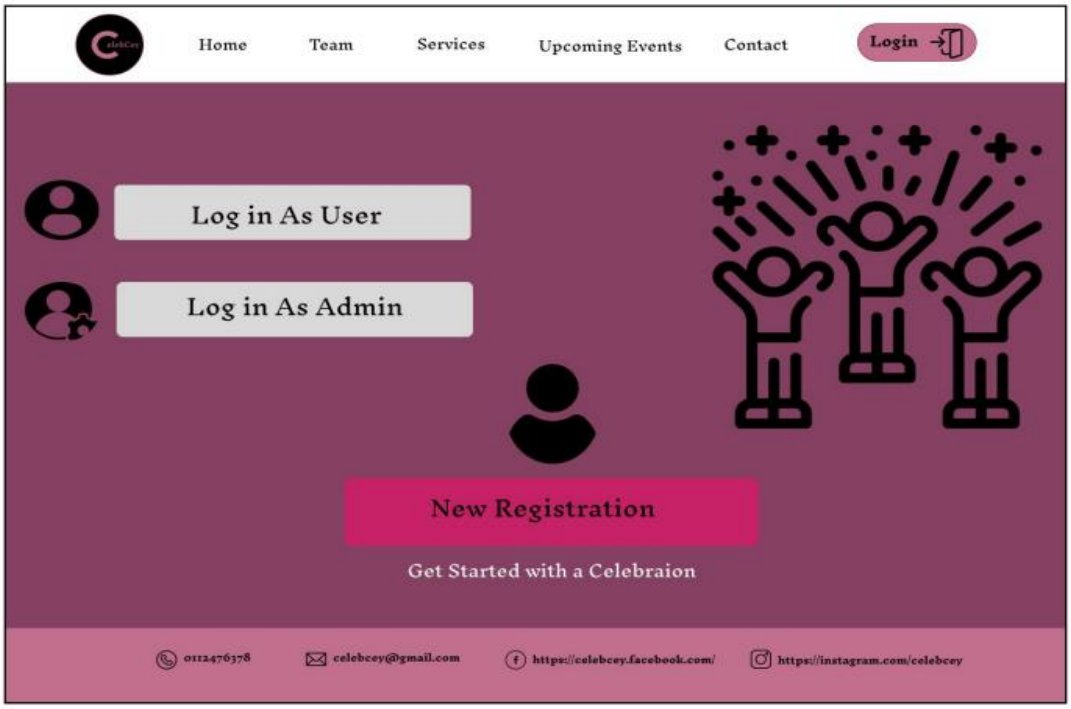

Figure 19: Login page

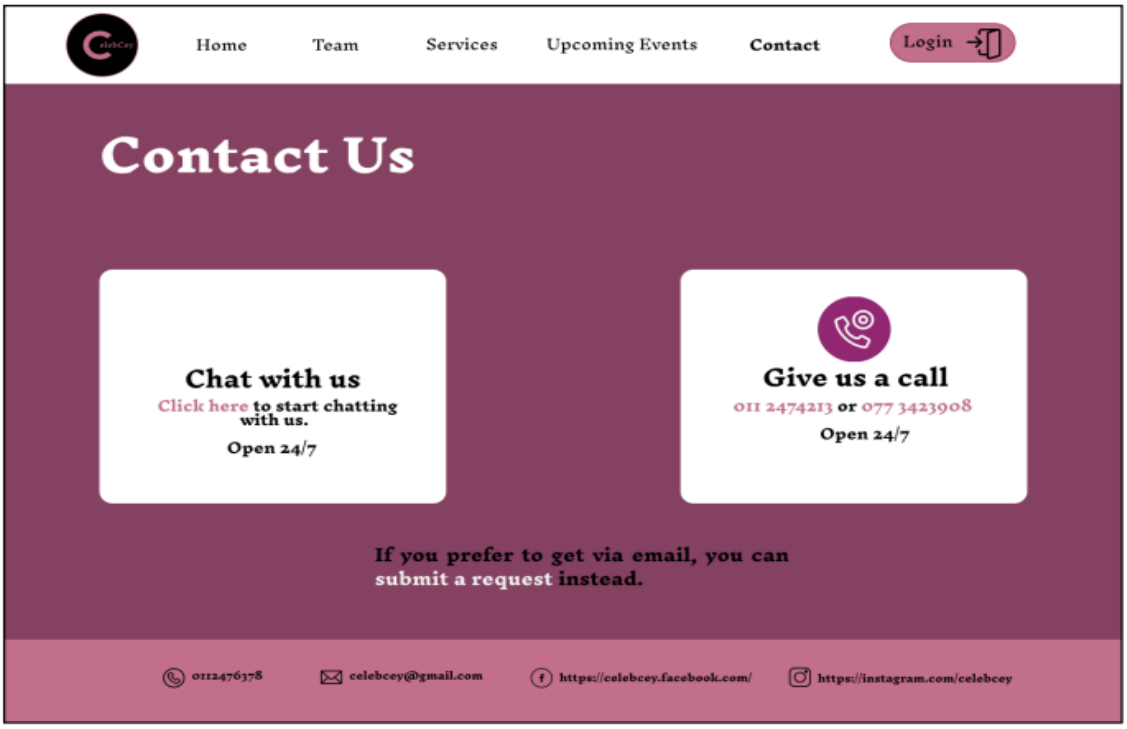

Figure 20: Contact page

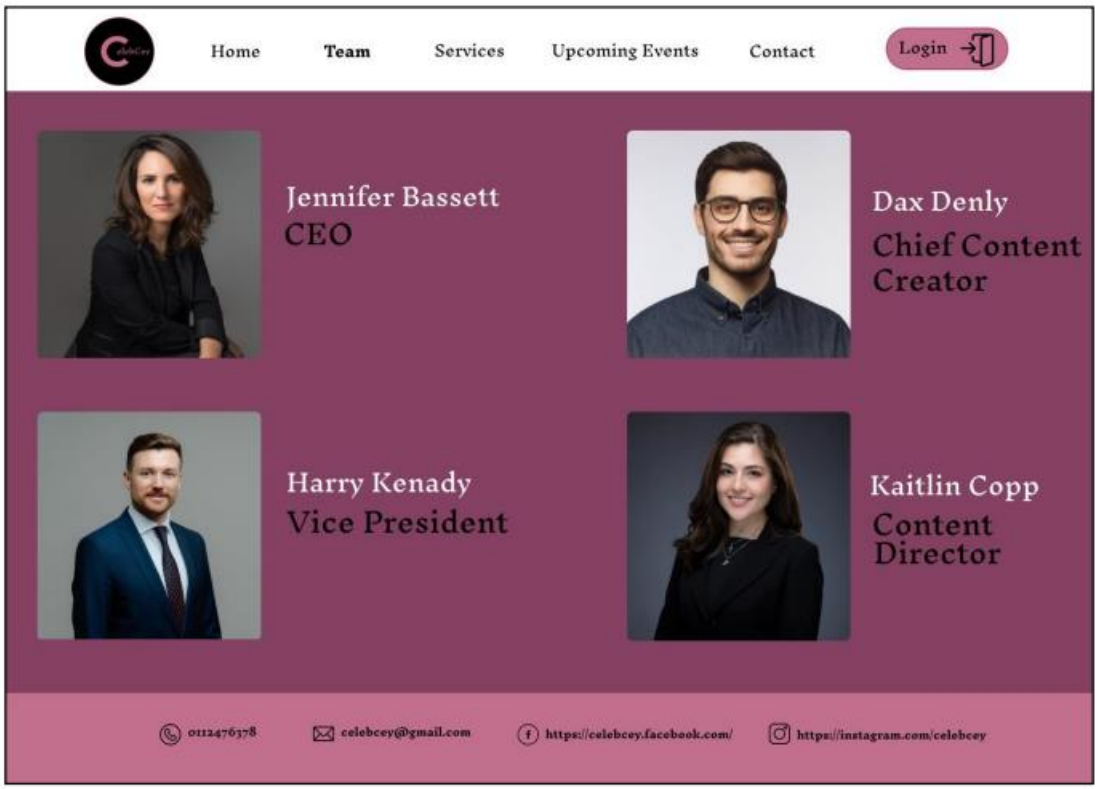

Figure 21: team page

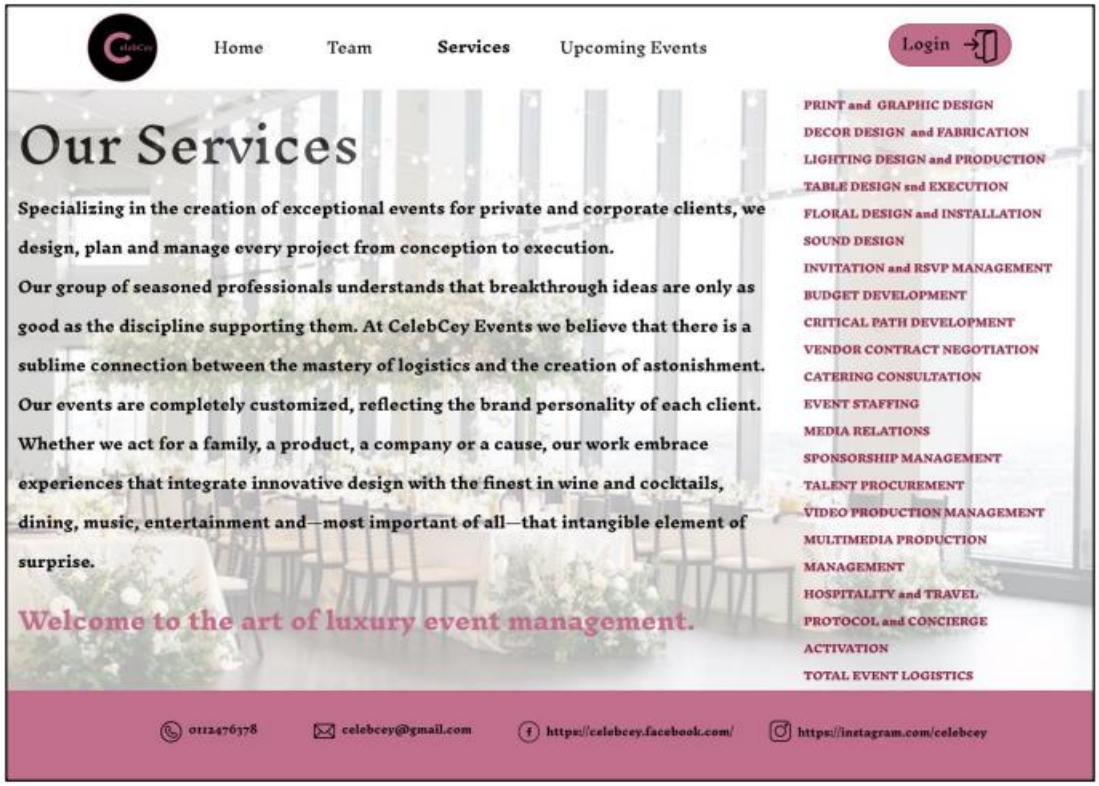

Figure 22: Services page

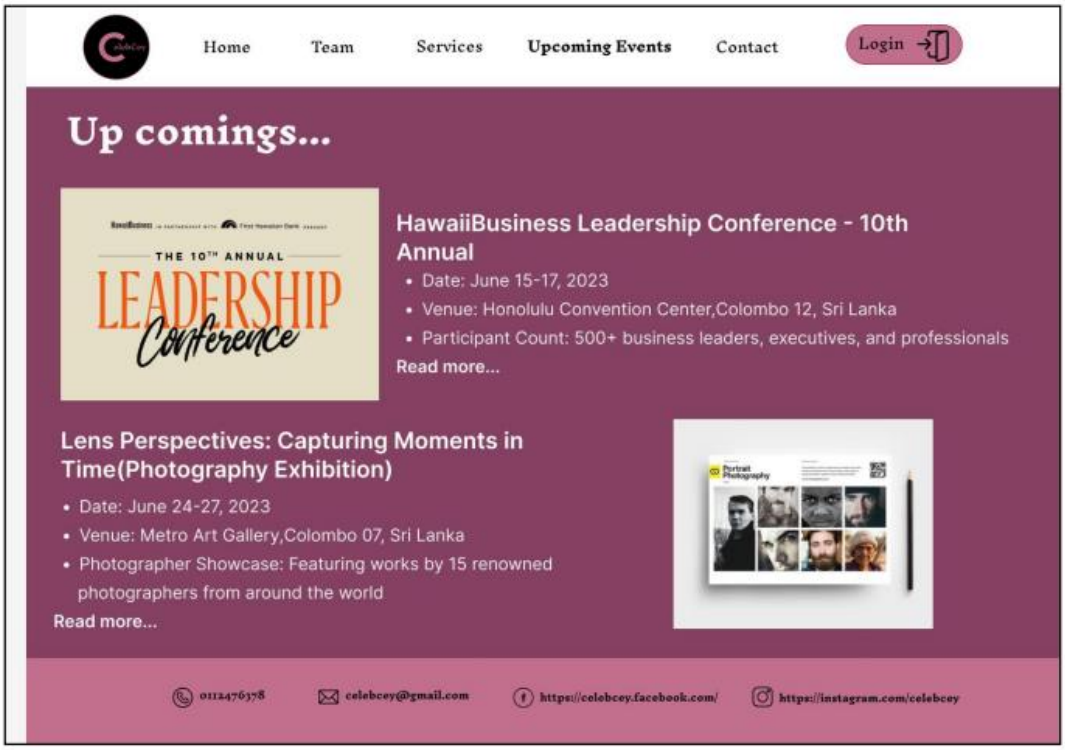

Figure 23: Upcoming event page

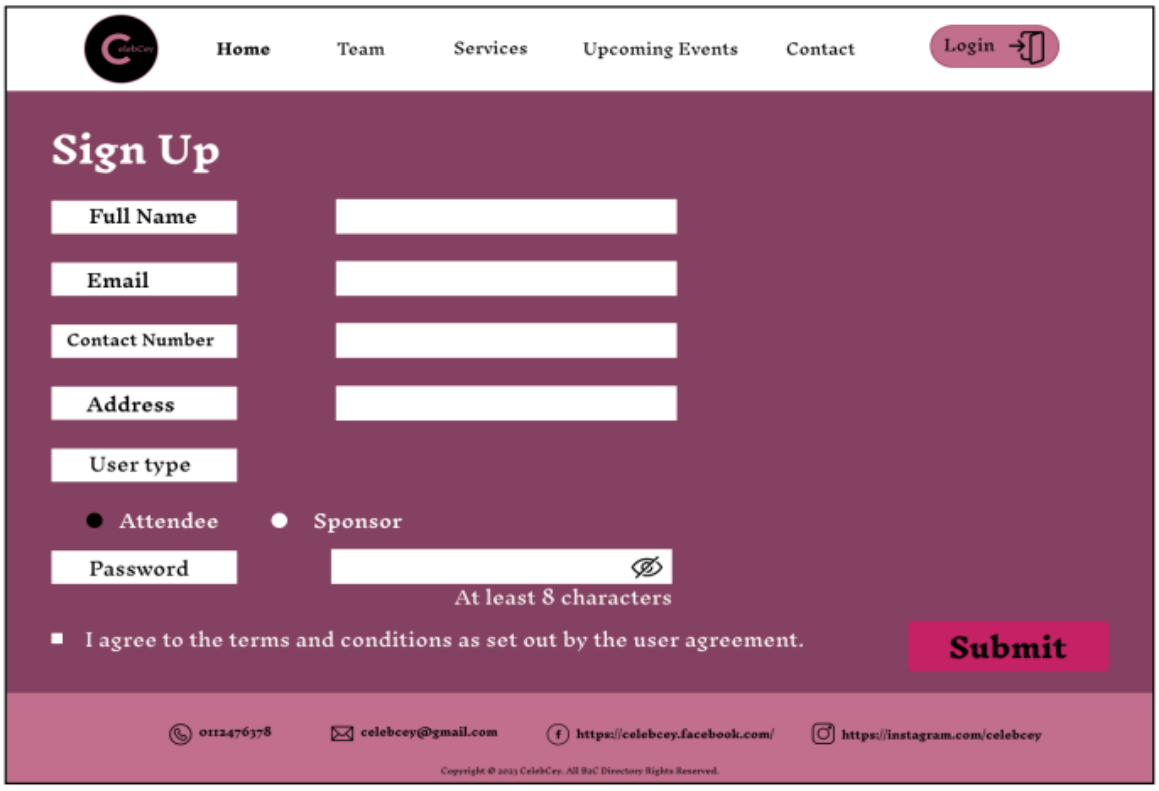

Figure 24: Registration page

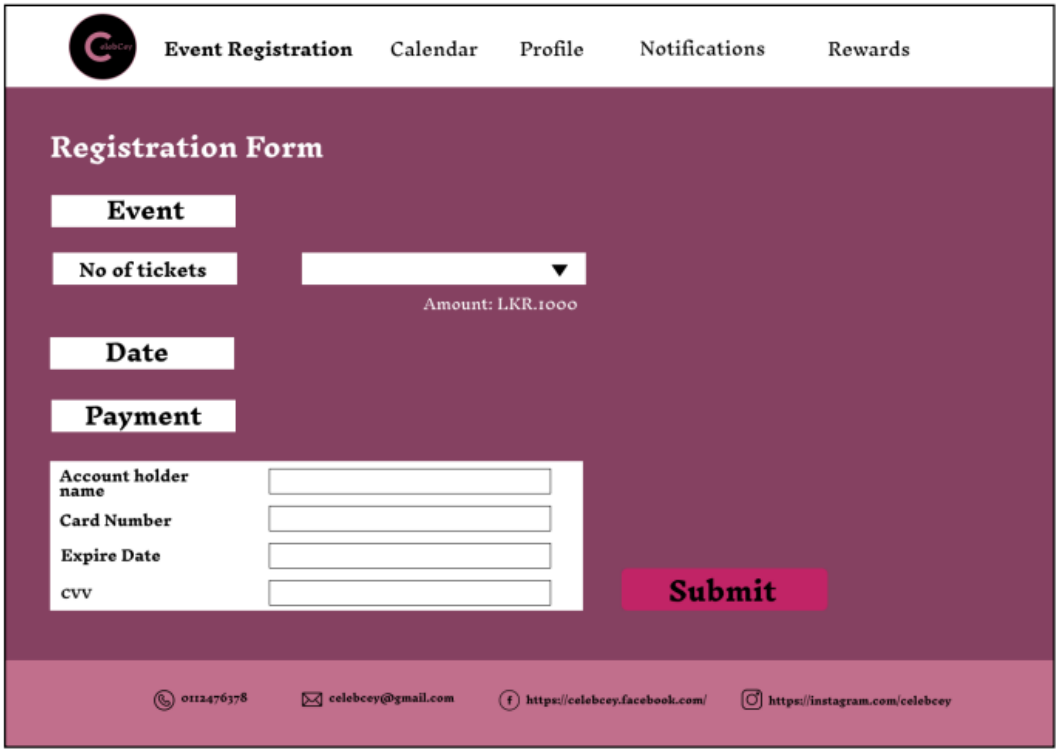

Figure 25: Interface for attendees

Figure 26: Interface for event organizers

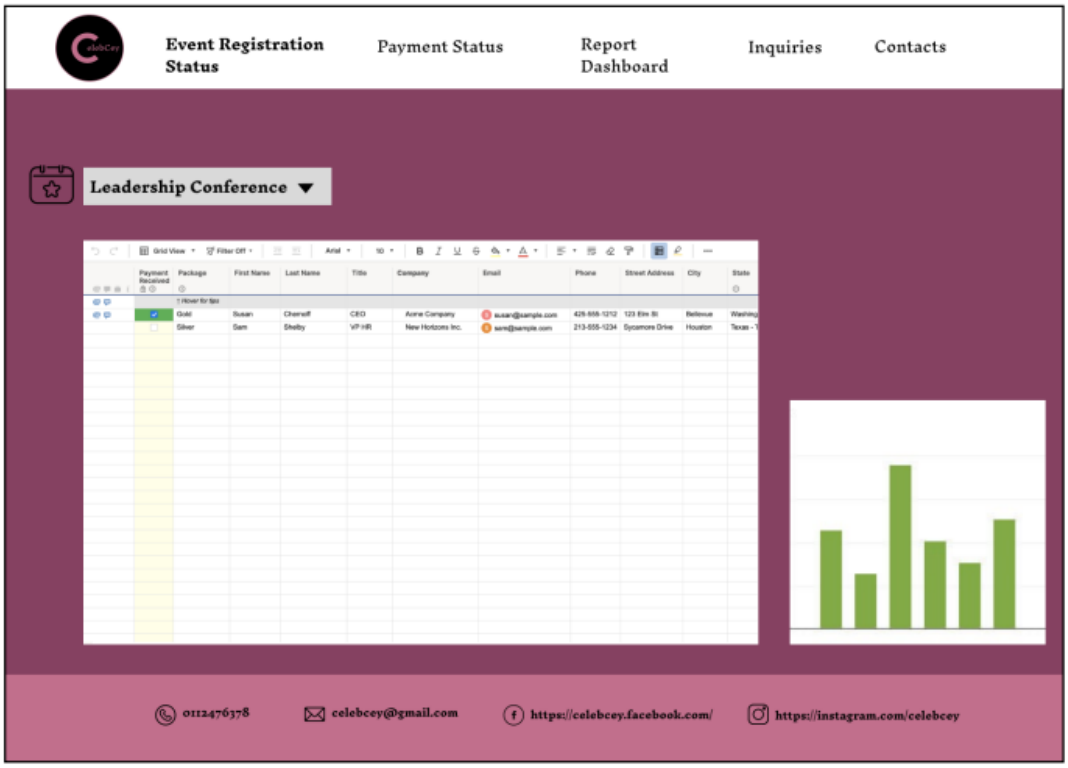

Figure 27: Interface for event registration team

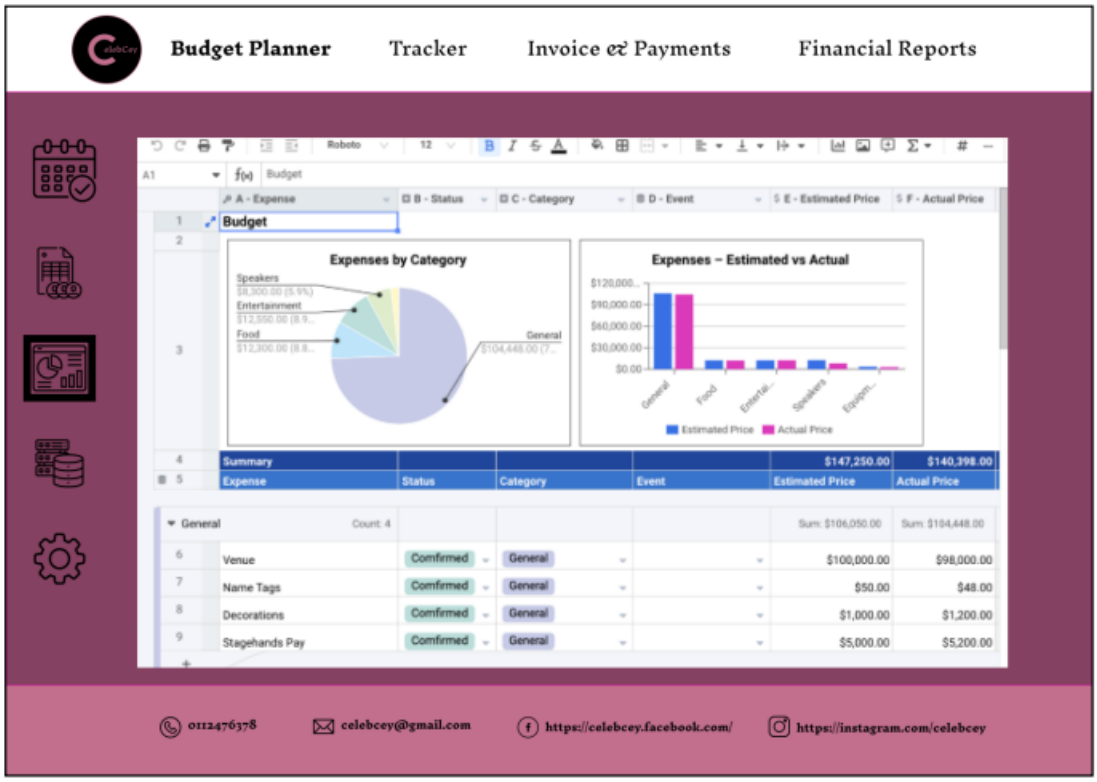

Figure 28: Interface for financial team

Figure 29: Interface for event sponsors

4. Architectural Design

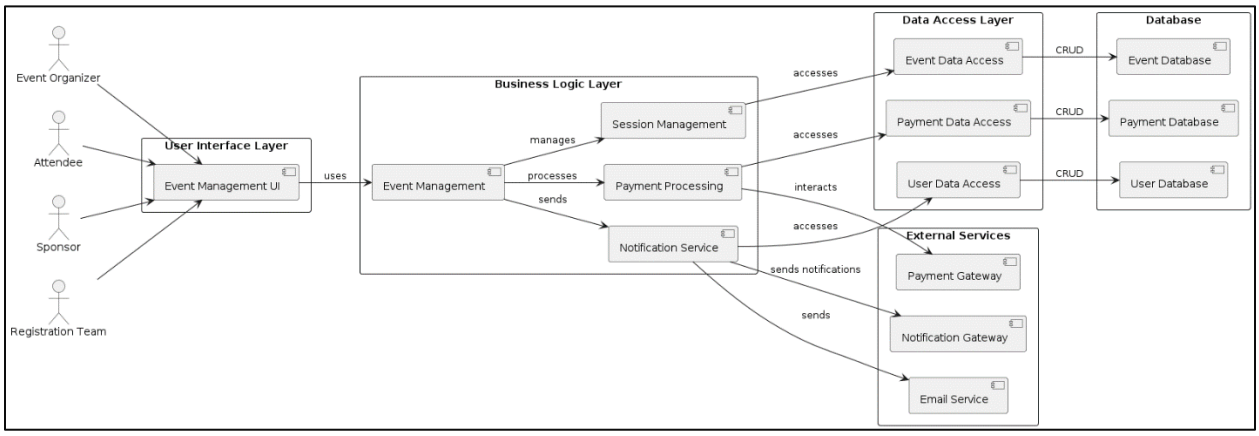

Figure 30: Architectural Design

4.1 Non-functional Requirements

1. Usability:
    - Requirement: Achieve a System Usability Scale (SUS) score of at least 75.
    - Description: Conduct usability testing to ensure that users find the system easy to use and navigate. Use SUS scores as a quantitative measure of usability.
2. Reliability
    - Requirement: Maintain a system uptime of at least 99.5%.
    - Description: Implement robust error handling, conduct stress testing, and use automated monitoring tools to ensure high system reliability and availability.
3. Maintainability
    - Requirement: Maintain a system uptime of at least 99.5%.
    - Description: Implement robust error handling, conduct stress testing, and use automated monitoring tools to ensure high system reliability and availability.
4. Portability
    - Requirement: Ensure the system runs seamlessly on major web browsers (Chrome, Firefox, Safari) and operating systems (Windows, macOS, Linux).
    - Details: Utilize web standards and responsive design principles to enhance cross-browser and cross-platform compatibility.

5. Performance Efficiency
    - Requirement: Load web pages within 3 seconds under standard network conditions.
    - Description: Optimize code, utilize content delivery networks (CDNs), and implement caching mechanisms to enhance overall system performance.
6. Testability
    - Requirement: Achieve a test coverage rate of 80% or higher.
    - Details: Develop comprehensive test cases, utilize unit testing and integration testing, and automate testing processes to ensure thorough test coverage.
7. Adaptability
    - Requirement: Support regular updates and feature additions without disrupting existing user experiences.
    - Description: Implement a modular architecture, use feature flags for controlled releases and conduct A/B testing for new features.
8. Flexibility
    - Requirement: Allow users to customize event notification preferences and dashboard layouts.
    - Description: Implement user-configurable settings, providing flexibility in tailoring the system to individual preferences.
9. Availability
    - Requirement: Maintain 24/7 availability for critical system functionalities, such as event registration and ticketing.
    - Description: Implement redundancy, utilize cloud hosting with high availability, and conduct regular maintenance during non-peak hours.
10. Robustness
    - Requirement: Handle a concurrent user load of 2000 users without critical system failures.
    - Description: Conduct load testing, implement graceful error recovery mechanisms, and ensure system components are resilient under heavy usage.

4.2 Architectural Design

The system will use a multi-tier architecture consisting of the following layers:

- Presentation Layer: User interface accessible via web and mobile applications.
- Application Layer: Business logic and processing.
- Data Layer: Database management for storing user data, event details, and transaction records.

Architectural Design Explanation

The system's architecture is organized into several distinct layers, each responsible for specific aspects of the overall functionality. Here is a detailed textual explanation of the architectural design, along with the roles of each component:

User Interface Layer

1. Event Management UI:

- This is the front-end interface that interacts directly with the users, including Event Organizers, Attendees, Sponsors, and the Registration Team.
- It provides the necessary screens and forms for users to perform their tasks such as creating events, registering for events, viewing schedules, and managing sponsorships.

Business Logic Layer

2.Event Management:

- This component contains the core business logic for managing events.
- It handles tasks such as event creation, modification, deletion, and overall event management.
- It interacts with the Session Management, Payment Processing, and Notification Service components to perform various operations.

3.Session Management:

- This component manages the scheduling and handling of event sessions.
- It includes functionalities like creating session schedules, assigning speakers, and managing session details.
- It accesses data from the Event Data Access component to retrieve or store session information.

4.Payment Processing:

- This component handles all payment-related activities within the system.
- It processes payments for event registrations, sponsorships, and other financial transactions.
- It interacts with the Payment Gateway for secure payment processing and accesses Payment Data Access for storing payment information.

5.Notification Service:

- This component is responsible for sending notifications and emails to users.
- It manages sending event-related notifications, confirmations, reminders, and updates.
- It interacts with the Email Service and Notification Gateway to deliver these messages and accesses User Data Access for retrieving user contact information.

Data Access Layer

6.Event Data Access:

- Provides an interface for CRUD (Create, Read, Update, Delete) operations on event-related data.
- It interacts with the Event Database to persist event details, session information, and other related data.

7. User Data Access:

- Manages data related to users including Event Organizers, Attendees, Sponsors, and Registration Team members.
- It performs CRUD operations on the User Database to maintain user profiles and related information.

8. Payment Data Access:

- Handles the storage and retrieval of payment data.
- It interacts with the Payment Database to keep track of transaction records, payment statuses, and other financial information.

External Services

9.Payment Gateway:

- An external service used for processing payments securely.

- The Payment Processing component interacts with this service to handle payment transactions.

10.Email Service:

- An external service responsible for sending emails.
- The Notification Service uses this service to send confirmation emails, updates, and other email notifications to users.

11.Notification Gateway:

- Another external service used for sending notifications (e.g., SMS, push notifications).
- The Notification Service interacts with this gateway to deliver various types of notifications to users.

Database

12.Event Database:

- Stores all event-related data including event details, session information, and schedules.
- Accessed by the Event Data Access component for persisting and retrieving event information.

13.User Database:

- Contains data about all users interacting with the system.
- Managed by the User Data Access component to perform CRUD operations on user profiles and related information.

14.Payment Database:

- Stores financial data including transaction records, payment statuses, and other payment-related information.
- Accessed by the Payment Data Access component to ensure accurate and up-to-date financial records.

Component Interactions

- User Interactions:
  - Users interact with the system through the Event Management UI, which communicates with the underlying business logic components to perform various tasks.
- Business Logic and Data Access:

    - The business logic components (Event Management, Session Management, Payment Processing, Notification Service) rely on the data access components to store and retrieve data from the databases.
- External Services:
    - Payment Processing interacts with the Payment Gateway for handling transactions.
    - Notification Service uses the Email Service for sending emails and the Notification Gateway for sending other types of notifications.

4.3 Design and Implementation

- **Technology Stack:** Use modern web development frameworks (e.g., React, Angular for frontend; Node.js, Django for backend).
- **Database:** Relational database (e.g., PostgreSQL, MySQL) for structured data storage.
- **Deployment:** Cloud-based deployment for scalability and reliability.

4.4 System Requirements

- Hardware: Servers with sufficient processing power, memory, and storage.
- Software: Web server, application server, database server, and necessary development tools.

5. Project Plan

5.1 Project schedule

- Phase 1: Requirements gathering and analysis (4 weeks)
- Phase 2: System design (3 weeks)
- Phase 3: Implementation (8 weeks)
- Phase 4: Testing (4 weeks)
- Phase 5: Deployment and training (2 weeks)
- Phase 6: Maintenance and support (ongoing)

Phase	Duration	Activities
Phase 1: Requirements Gathering and Analysis	4 weeks	
		- Conduct interviews
		- Analyze current processes
		- Document functional requirements
		- Prioritize requirements
Phase 2: System Design	3 weeks	
		- Design database schema
		- Develop UI wireframes
		- Define system architecture
Phase 3: Implementation	8 weeks	
		- Backend Development
		- Frontend Development
		- Database Implementation
		- UI/UX Design
Phase 4: Testing	4 weeks	
		- Unit Testing

		- Integration Testing
		- System Testing
Phase 5: Deployment and Training	2 weeks	
		- Develop user manuals
		- Conduct training sessions
		- Prepare deployment guides
Phase 6: Maintenance and Support	Ongoing	
		- Monitor system performance
		- Provide technical support
		- Implement updates and fixes

Table 5

5.2 Project Budget

- Development Costs: Rs. 150,000
- Testing Costs: Rs. 100,000
- Deployment Costs: Rs. 50,000
- Maintenance Costs: Rs. 35,000/year
- **Total Estimated Budget: Rs. 335,000**

References

- [1] Nielsen Norman Group, "Nielsen Norman Group’s User Interface Style Guide," Nielsen Norman Group, Mar. 15, 2023. [Online]. Available: https://www.nngroup.com/articles/ui-style-guides/. [Accessed: Jul. 1, 2024].

- [2] Government of Canada, "Contract for Professional Services," Government of Canada, 2022. [Online]. Available: https://www.tpsgc-pwgsc.gc.ca/app-acq/spc-cps/desc/at-ag-eng.html. [Accessed: Jul. 1, 2024].

- [3] International Organization for Standardization, "ISO 9001:2015 Quality management systems — Requirements," ISO, 2020. [Online]. Available: https://www.iso.org/standard/62085.html. [Accessed: Jul. 1, 2024].

- [4] J. Smith and A. Doe, "Event Management System Requirements Specifications," University of Kelaniya, 2023. [Online]. Available: https://example.com/system-requirements-specs.pdf. [Accessed: Jul. 1, 2024].

- [5] C. Brown and M. Lee, "Use Cases for Online Event Registration," Example Corp, 2022. [Online]. Available: https://example.com/use-cases.pdf. [Accessed: Jul. 1, 2024].

- [6] S. Johnson and D. Williams, "Vision and Scope of the Event Management System," Example University, 2021. [Online]. Available: https://example.com/vision-scope.pdf. [Accessed: Jul. 1, 2024]

Appendix A: Glossary

- Event Management System (EMS): A comprehensive software platform designed to facilitate the efficient planning, organization, and management of events, including functionalities for event registration, session management, sponsor management, and financial transactions.
- SRS (Software Requirement Specification): A document that clearly and precisely defines the requirements of a software system. It serves as a blueprint for the development team and outlines what the software must do to satisfy the users' needs.
- Product Scope: Defines the boundaries of the software system, specifying what functionalities are included and excluded from the EMS.
- Non-functional Requirements: Specifications that describe the system's operation and performance characteristics, such as usability, reliability, performance, and security.
- Architectural Design: The structure and organization of the software system, including its components, modules, and relationships.
- Project Plan: A detailed outline of the tasks, resources, timeline, and budget required to complete the EMS development project.
- Project Schedule: A timeline outlining the phases and activities of the project, from requirements gathering to deployment and maintenance.
- Project Budget: An estimate of the costs associated with developing, testing, deploying, and maintaining the EMS.
- Use Case: Describes interactions between actors (users) and the system, detailing the steps needed to achieve a specific goal or task.
- Actor: A role played by a user or external system that interacts with the EMS.
- Actor Authentication: The process of verifying the identity of a user or external system accessing the EMS.
- UI (User Interface): The visual elements and controls through which users interact with the EMS.
- UX (User Experience): Refers to the overall experience of a user interacting with the EMS, including ease of use, intuitiveness, and satisfaction.

- CRUD (Create, Read, Update, Delete): Basic operations for persistent storage in a database or file system.
- API (Application Programming Interface): Defines how software components should interact and communicate with each other.
- Database Management System (DBMS): A software system that manages databases and provides functionalities for storing, organizing, and retrieving data.
- Entity: A distinct object or concept in the EMS domain, such as events, attendees, sponsors, etc.
- Transaction: An interaction between a user and the EMS that results in a change of state or data.
- Payment Gateway: A service that facilitates online payment transactions securely between the EMS and external financial networks.
- Notification Service: A component responsible for sending automated notifications, alerts, or messages to users of the EMS.
- Financial Team: A group responsible for managing financial transactions, reports, and compliance within the EMS.
- Registration Team: A group responsible for managing event registrations, attendee information, and support inquiries within the EMS.
- QA (Quality Assurance): The process of ensuring that the EMS meets specified requirements and quality standards through testing and validation.

Appendix B: Requirements Gathering Questionnaire

Event Management System Requirements Gathering

Introduction: This questionnaire aims to gather requirements for the Event Management System. Please provide detailed responses to ensure all necessary features and functionalities are captured.

Section 1: General Information

1. Name:
2. Role:
    - Event Organizer
    - Sponsor
    - Attendee
    - Registration Team Member
    - Other (Please specify)
3. Contact Information:
    - Email:
    - Phone Number:

Section 2: Event Registration and Ticketing

1. What types of events do you typically organize/attend/sponsor?
2. What features do you expect in the event registration process?
3. How important is the ability to purchase tickets online?
    - Very Important
    - Important
    - Neutral
    - Not Important
4. What payment methods should the system support?
    - Credit/Debit Card
    - PayPal
    - Bank Transfer
    - Other (Please specify)

Section 3: Session and Agenda Management

1. How do you currently manage event schedules and agendas?

2. What features would help you manage event sessions more effectively?
    - Real-time updates
    - Speaker management
    - Session reminders
    - Other (Please specify)
3. Do you require the ability to upload and manage presentation materials within the system?
    - Yes
    - No

Section 4: Sponsor Management

1. How do you currently manage sponsor relationships and commitments?
2. What features would assist you in managing sponsorship packages?
    - Profile and contact management
    - Sponsorship package details
    - Commitment tracking
    - Material and logo submission
3. What information should sponsors be able to access in the system?

Section 5: Payment and Financial Management

1. How do you currently handle payments and financial transactions for events?
2. What challenges do you face with current payment processes?
3. What financial reports would be beneficial for you?
    - Income reports
    - Expense reports
    - Payment status
    - Other (Please specify)

Section 6: Additional Features and Feedback

1. Are there any additional features you would like to see in the Event Management System?
2. What improvements can be made to existing systems you use?
3. Any other comments or suggestions?

Thank you for your participation! Your feedback is valuable in developing a comprehensive Event Management System.