Class Performance Report 01

Only for course Teacher						
		Needs Improvement	Fair	Good	Excellent	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	15
Level of Content	3					
Analysis	5					
Development	4					
Accuracy	3					
					Total obtained mark	
Comments						

Semester: Spring 2026

Student Name: K.M. Fathum Mubin Sachcha **ID: 241-35-554**

Student Name: Md. Rakibul Islam **ID: 241-35-099**

Student Name: Md. Mahmud Rana Tonmoy **ID: 241-35-062**

Batch: 42 **Section:** E1 **Course Code:** SE 231

Course Name: System Analysis & Design Capstone Project

Course Teacher Name: Most. Munira Tabassum

Designation: Lecturer

Submission Date:

EventHive – Event Coordinator System

Table of Contents

Introduction

EventHive is a comprehensive, web-based event coordination and management system designed to streamline the planning, execution, and monitoring of events of various scales and categories including social, academic, promotional, corporate, and political events.

Traditional event management relies heavily on manual coordination, scattered communication, and disconnected tools for budgeting, vendor handling, scheduling, and attendee management. EventHive solves this problem by providing a centralized digital platform where organizers can manage the entire event lifecycle from concept planning to post-event analytics.

The system focuses on automation, collaboration, transparency, and scalability, making it suitable for universities, corporations, agencies, NGOs, and political organizations.

Objectives

- To develop a web and mobile-based event management system for easy access from any device.
- To provide a centralized platform for planning and managing different types of events.
- To reduce manual coordination and paperwork through digital automation.
- To enable organizers to manage events anytime and anywhere using a mobile app.
- To improve communication among organizers, team members, and attendees.
- To simplify event scheduling, budgeting, and task assignment.
- To allow online registration and attendance management through both web and app.
- To store event data securely and systematically in a database.
- To generate basic reports to evaluate event performance.
- To design a scalable system that can be expanded with more features in the future.

Goals

- To create a reliable cross-platform event management solution.
- To ensure seamless synchronization between the web application and mobile app.
- To make event management faster, more organized, and more efficient.
- To enhance teamwork and collaboration through real-time updates.
- To improve user experience with a simple and intuitive interface.
- To support multiple event categories in a single system.
- To minimize errors in scheduling, budgeting, and coordination.
- To increase accessibility by supporting both desktop and mobile users.
- To help organizations monitor events using reports and feedback.
- To establish a foundation for a professional event management platform.

Scope

- User registration and secure login system
- Role-based access for admin, organizer, team member, and attendee
- Event creation, update, and management
- Support for multiple event types (social, academic, corporate, promotional, political)
- Event scheduling with date, time, and venue details
- Budget estimation and expense tracking
- Vendor information management
- Task assignment and progress tracking
- Team and volunteer management
- Online attendee registration
- Attendance tracking system
- Notification and reminder system
- Dashboard with event overview
- File upload and document management
- Feedback and review collection
- Report generation (attendance, expenses, event summary)
- Data synchronization between web and mobile app
- Secure database management

Key Features

1. Event Creation & Management: Organizers can create, update, and manage different types of events from one place.

2. Support for Multiple Event Types: Handles social, academic, corporate, promotional, and political events.

3. User Registration & Login: Users can sign up and log in securely to access the system.

4. Role-Based User Access: Different roles such as Admin, Organizer, Team Member, and Attendee with specific permissions.

5. Event Scheduling: Set event dates, time, venue, and basic event timeline.

6. Budget Planning: Allows organizers to estimate and track event expenses.

7. Vendor Management: Store and manage vendor details like caterers, decorators, sound systems, etc.

8. Task Assignment: Organizers can assign tasks to team members and track their progress.

9. Employee Management: Add, remove, and manage event team members easily.

10. Attendee Registration: Participants can register for events online.

11. Attendance Tracking: Track who attended the event manually or digitally.

12. Notification System: Send important updates and reminders to users.

13. Event Status Tracking: View event progress such as planned, ongoing, or completed.

14. Report Generation: Generate simple reports like attendance list and expense summary.

15. Feedback Collection: Collect feedback from attendees after the event.

16. Document Upload: Upload and store event-related files such as schedules or agreements.

17. Dashboard Overview: Simple dashboard showing total events, upcoming events, and completed events.

18. Multiple Event Handling: Manage more than one event at the same time.

19. Data Security: Basic security features like password protection and user authentication.

20. Web-Based Access: The system can be accessed from any device with an internet connection.

Scenario Wise Features

Scenario Writing

Scenario writing describes specific situations or use cases that show how users interact with the system. The following scenarios explain typical user interactions with the EventHive: Event Coordinator system.

Scenario-1: User Registration & Login

- Request for user registration
- Provide required information (name, email, password, role)
- Submit registration form
- System validates information
- Account created successfully
- User logs into the system

Scenario-2: Create Event

- Request for creating a new event
- Select event type (social, academic, corporate, etc.)
- Enter event details (title, date, time, venue)
- Add basic event description
- Submit event information
- Event created successfully

Scenario-3: Event Scheduling

- Request for event scheduling
- Select event from event list
- Set event date and time
- Assign venue/location
- Save schedule
- Event schedule confirmed

Scenario-4: Budget Planning

- Request for budget creation
- Select related event
- Enter estimated expenses
- Update cost details
- Save budget information
- Budget plan created successfully

Scenario-5: Vendor Management

- Request for adding vendor
- Enter vendor details
- Select service type (catering, decoration, sound, etc.)
- Save vendor information
- Vendor added successfully

Scenario-6: Task Assignment

- Request for task assignment
- Select event
- Assign task to team member
- Set deadline and priority
- Save task details
- Task assigned successfully

Scenario-7: Team Member Management

- Request for adding team member
- Enter team member details
- Assign role and responsibilities
- Save team information
- Team member added successfully

Scenario-8: Attendee Registration

- Request for attendee registration
- Fill registration form
- Submit registration
- System verifies information
- Registration completed successfully

Scenario-9: Attendance Tracking

- Request for attendance marking
- Select event
- View attendee list
- Mark present or absent
- Save attendance record
- Attendance updated successfully

Scenario-10: Notification & Announcement

- Request for sending notification
- Select event and target users
- Write notification message
- Send notification
- Notification delivered successfully

Scenario-11: Feedback Collection

- Request for feedback collection
- Select completed event
- Provide feedback form
- Attendees submit feedback
- Feedback stored successfully

Scenario-12: Generate Report

- Request for report generation
- Select event
- Choose report type (attendance, budget, summary)
- Generate report
- Report generated successfully

Scenario-13: Event Completion

- Request to close event
- Verify all tasks are completed
- Finalize budget and attendance
- Mark event as completed
- Event closed successfully

Requirement Elicitation Techniques

Requirement elicitation techniques are methods used to collect, analyze, and understand user requirements for developing the EventHive system.

1. Interviews

One-to-one or group discussions with potential users such as event organizers, students, teachers, and volunteers.

Purpose in EventHive:

- Understand problems in manual event management
- Identify required features like scheduling, budgeting, and registration
- Gather expectations for both web and mobile usage

2. Questionnaires & Surveys

Structured forms with multiple-choice and short questions distributed to a large group of users.

Purpose in EventHive:

- Collect opinions from students, event attendees, and organizers
- Identify common event management issues
- Decide which features are most important

3. Observation

Observing how events are currently planned and managed in real environments.

Purpose in EventHive:

- Study how organizers coordinate tasks manually
- Identify inefficiencies in communication and attendance tracking
- Understand real-world workflows of events

4. Document Analysis

Reviewing existing documents related to event planning.

Purpose in EventHive:

- Analyze event schedules, budgets, attendance sheets
- Understand how information is currently recorded
- Identify data fields needed in the system

5. Brainstorming Sessions

Group discussions among project team members and stakeholders to generate ideas.

Purpose in EventHive:

- Identify possible features for web and mobile platforms
- Propose simple and user-friendly solutions
- Improve system design ideas

6. Use Case Analysis

Defining interactions between users and the system through use cases.

Purpose in EventHive:

- Identify system functionalities clearly
- Define user roles and actions
- Convert requirements into scenarios and features

7. Prototyping

Creating basic UI mockups or wireframes of the system.

Purpose in EventHive:

- Show early design of dashboards and forms
- Collect feedback before final development
- Improve usability for both web and app users

8. Stakeholder Meetings

Meetings with different stakeholders involved in event management.

Purpose in EventHive:

- Align system features with user needs
- Resolve conflicting requirements
- Validate system scope and limitations

9. Market & System Study

Studying existing event management systems and apps.

Purpose in EventHive:

- Identify missing or weak features in existing systems
- Avoid repeating common design mistakes

10. Feedback & Review Sessions

Collecting feedback after reviewing requirement drafts or prototypes.

Purpose in EventHive:

- Validate collected requirements
- Make necessary changes
- Ensure system meets user expectations

Feasibility Study

The feasibility study evaluates whether the EventHive system can be developed successfully.

1. Technical Feasibility

- The system will be developed using commonly available technologies such as HTML, CSS, JavaScript, backend framework, and MySQL.
- The mobile application can be developed using Android or cross-platform tools.
- Required hardware and software resources are available to the development team.
- The project is technically feasible for a 3-member student team.

2. Economic Feasibility

- The project does not require expensive hardware or paid software tools.
- Open-source tools and frameworks will be used to reduce cost.
- Development cost is minimal and suitable for an academic project.
- The system is economically feasible within a limited budget.

3. Operational Feasibility

- The system is easy to use and does not require extensive user training.
- Users can access the system through web browsers and mobile devices.
- The system improves efficiency and reduces manual effort.
- The system is operationally feasible for real-world use.

4. Schedule Feasibility

- The project can be completed within the allocated academic time frame.
- Tasks will be distributed among team members to meet deadlines.
- The development schedule is realistic and achievable.

Non-Functional Requirements

Non-functional requirements define how the system performs rather than what it does.

1. Usability

- The system shall provide a simple and user-friendly interface.
- Both web and mobile interfaces shall be easy to navigate.

2. Performance

- The system shall respond to user actions within acceptable time limits.
- The system shall handle multiple users simultaneously.

3. Security

- User authentication shall be required to access the system.
- Passwords shall be securely stored.
- User data shall be protected from unauthorized access.

4. Availability

- The system shall be available whenever users need to access it.
- Minimal downtime shall be ensured.

5. Scalability

- The system shall support future expansion such as more users and events.
- New features can be added without major changes.

6. Reliability

- The system shall function correctly without failure during normal use.
- Data shall not be lost in case of minor system issues.

7. Compatibility

- The system shall work on major web browsers.
- The mobile app shall be compatible with common Android devices.

8. Maintainability

- The system shall be easy to update and maintain.
- Code shall be well-structured and documented.

Project Time Frame

Phase	Activity	Duration
Phase 1	Requirement Analysis & Planning	2 weeks
Phase 2	System Design (UI, Diagrams, DB)	2 weeks
Phase 3	Web Application Development	3 weeks
Phase 4	Mobile App Development	2 weeks
Phase 5	Integration & Testing	2 weeks
Phase 6	Documentation & Presentation	1 week

Description: The total estimated duration of the project is 12 weeks, which is divided into multiple phases to ensure systematic development and timely completion.

The project begins with requirement analysis and planning, where system requirements are gathered, analyzed, and documented. This phase ensures a clear understanding of project objectives and user expectations.

Next, the system design phase focuses on creating user interface designs, database structure, and system diagrams to define how the system will function.

The web application development phase involves implementing core functionalities such as user management, event creation, scheduling, and reporting. In parallel, the mobile application development phase ensures that similar features are accessible through a mobile-friendly interface.

After development, integration and testing are performed to verify that the web and mobile applications work seamlessly together and meet functional requirements.

Finally, the project concludes with documentation and presentation, where system manuals, reports, and final project presentations are prepared.

This phased approach ensures efficient task distribution among team members and helps maintain project quality and deadlines.

Project Budget

Item	Description	Estimated Cost (BDT)
Project Planning & Requirement Analysis	Stakeholder meetings, requirement elicitation, documentation	60,000
UI/UX Design	Professional UI/UX design for web & mobile app (wireframes, prototypes)	80,000
Web Application Development	Frontend & backend development of web system	2,50,000
Mobile Application Development	Android / Cross-platform mobile app development	2,00,000
Database Design & Management	Database architecture, optimization, backups	60,000
API Development & Integration	Web–app synchronization, API services	70,000
Hosting & Cloud Infrastructure	Cloud server, database hosting, storage, bandwidth	90,000
Security Implementation	Authentication, authorization, data security measures	50,000
Testing & Quality Assurance	Functional testing, bug fixing, performance testing	60,000
Deployment & Configuration	Production deployment and environment setup	40,000
Maintenance & Support (Initial)	Post-deployment support and minor updates	50,000
Documentation & Training	User manuals, technical documents, training materials	30,000
Contingency Cost	Unexpected expenses	30,000
Total Estimated Budget		**10,00,000 BDT (Approx.)**

Description: The budget includes costs for project planning and requirement analysis, ensuring that stakeholder needs are properly identified and documented before development begins. A significant portion of the budget is allocated to UI/UX design, which focuses on creating an intuitive and user-friendly experience for both web and mobile platforms. Development costs cover web application development, mobile application development, database design, and API integration, enabling seamless communication between the web system and mobile app. Investment in cloud hosting and infrastructure ensures high availability, data storage, and system scalability. To ensure data protection and system reliability, the budget also includes security implementation, testing, and quality assurance, which are critical for handling sensitive user and event data. Additional costs are allocated for deployment, documentation, user training, and initial maintenance, ensuring smooth system launch and early operational support. A contingency cost is included to handle unexpected technical or operational challenges.

Stakeholders (Users)

EventHive has six main stakeholders:

1. Admin
2. Event Organizer / Client / Manager
3. Team Member / Employees
4. Vendor / Service Provider / Convention Hall / Resort
5. Sponsor
6. Attendee / Participant

User Profile:

User Profile-01: Admin

User Class	Notes on Characteristic	Requirement Implied
Type of User	Admin	Verification
Age Range	28–45	Verification
Frequency of Use	Daily	Performance, Operation
Mandatory	Yes	
Computer Experience	Highly Experienced	Documentation
Education	B.Sc / M.Sc	
Goals	Manage system, ensure security	Security, Performance, Acceptance
Language Skills	Bangla, English	
Number of Users	2—3	Performance, Operation
Training	Minimal training required	Documentation
Other System Used	No	
Ways of Working	Full system control	Security, Operation, Maintenance

User Profile-02: Event Organizer / Manager

User Class	Notes on Characteristic	Requirement Implied
Type of User	Event Organizer / Manager	Verification
Age Range	25—50	Verification
Frequency of Use	Most of the time per day	Performance, Operation
Mandatory	Yes	
Computer Experience	Experienced	Documentation
Education	Graduate	
Goals	Plan events and hire vendors	Resource, Performance, Acceptance
Language Skills	Bangla, English	
Number of Users	10—20	Performance, Operation, Portability
Training	May accept basic training	Documentation
Other System Used	Sometimes	
Ways of Working	Manage events digitally	Acceptance, Security, Operation

User Profile-03: Team Member / Employee

User Class	Notes on Characteristic	Requirement Implied
Type of User	Team Member / Employee	Verification
Age Range	20–40	Verification
Frequency of Use	Regular	Performance
Mandatory	Yes	
Computer Experience	Moderate	Documentation
Education	Any	
Goals	Complete assigned tasks	Performance, Operation
Language Skills	Bangla, English	
Number of Users	20–50	Performance, Portability
Training	Short training acceptable	Documentation
Other System Used	No	
Ways of Working	Task-based operation	Acceptance, Operation

User Profile-04: Vendor / Service Provider / Convention Hall / Resort

User Class	Notes on Characteristic	Requirement Implied
Type of User	Vendor / Service Provider	Verification
Age Range	25–55	Verification
Frequency of Use	Occasionally	Performance
Mandatory	Yes	
Computer Experience	Moderate	Documentation
Education	Any	
Goals	Provide services and get bookings	Performance, Acceptance
Language Skills	Bangla, English	
Number of Users	30—100	Performance, Portability
Training	Minimal training required	Documentation
Other System Used	Yes	
Ways of Working	Service-based interaction	Acceptance, Operation

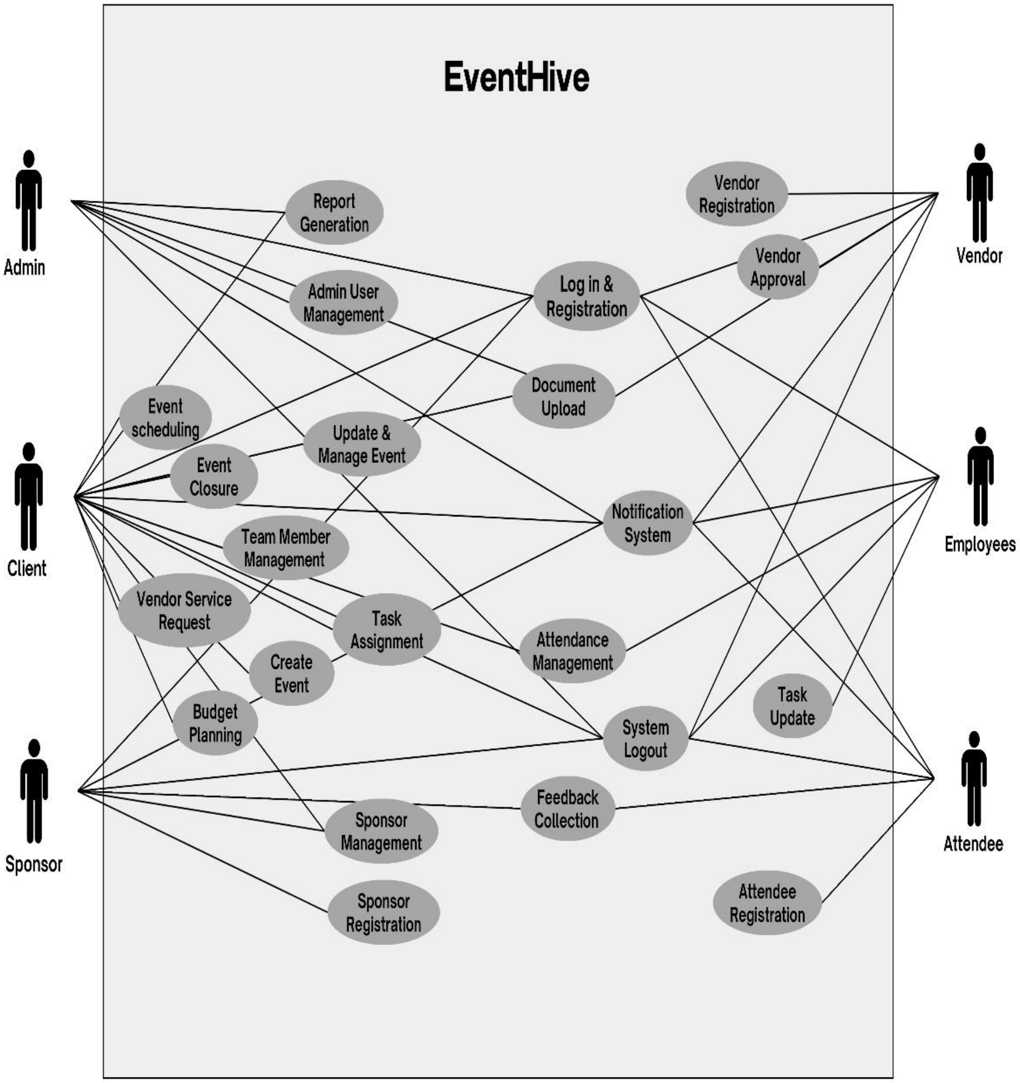
User Profile-05: Sponsor

User Class	Notes on Characteristic	Requirement Implied
Type of User	Sponsor	Verification
Age Range	30–55	Verification
Frequency of Use	Occasionally	Performance
Mandatory	No	
Computer Experience	Experienced	Documentation
Education	Graduate	
Goals	Promote brand through events	Performance, Acceptance
Language Skills	Bangla, English	
Number of Users	5—15	Performance
Training	No training required	
Other System Used	Yes	
Ways of Working	Monitor sponsorship details	Acceptance, Security

User Profile-06: Attendee / Participant (Guest)

User Class	Notes on Characteristic	Requirement Implied
Type of User	Attendee / Participant	Verification
Age Range	18–60	Verification
Frequency of Use	Rare	Performance
Mandatory	Yes	
Computer Experience	Basic	Documentation
Education	Any	
Goals	Attend event smoothly	Acceptance, Performance
Language Skills	Bangla, English	
Number of Users	100–1000+	Performance, Portability
Training	No training required	
Other System Used	No	
Ways of Working	Simple interaction	Acceptance, Security

Use Case Diagram



Use Case Description

UC-01: Login & Registration

<table>
<tr><td><b>Use Case</b></td><td>Login &amp; Registration</td></tr>
<tr><td><b>Goal</b></td><td>To allow users (Admins, Organizers, Vendors, Attendees) to register for an account and log in to access system features.</td></tr>
<tr><td><b>Preconditions</b></td><td>System is online and accessible. User has a valid email address (for registration).</td></tr>
<tr><td><b>Success End Condition</b></td><td>User is successfully authenticated and redirected to their respective dashboard.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>User remains on the login/registration page with an error message displaying the reason for failure.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Guest (Unregistered User), Registered User</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Email Service Provider (for verification)</td></tr>
<tr><td><b>Trigger</b></td><td>User navigates to the EventHive homepage and selects "Login" or "Register".</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>User opens the EventHive application.</td></tr>
<tr><td>2</td><td>User travels to the Login/Register page.</td></tr>
<tr><td>3</td><td>User selects "Register" and fills in required details (Name, Email, Password, Role).</td></tr>
<tr><td>4</td><td>User submits the registration form.</td></tr>
<tr><td>5</td><td>System validates the input and sends a verification email.</td></tr>
<tr><td>6</td><td>User verifies email and returns to the login page.</td></tr>
<tr><td>7</td><td>User enters valid credentials (Email, Password) and clicks "Login".</td></tr>
<tr><td>8</td><td>System authenticates the user and retrieves their role.</td></tr>
<tr><td>9</td><td>System redirects the user to the appropriate dashboard.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>5a</td><td>Invalid Input Data: System displays validation errors (e.g., "Email already in use", "Weak password"). User corrects information and resubmits.</td></tr>
<tr><td>8a</td><td>Invalid Credentials: System displays "Incorrect email or password". User retries.</td></tr>
<tr><td>8b</td><td>Account Suspended: System displays "Account suspended, contact Admin". Use Case ends.</td></tr>
</table>
</td></tr>
</table>

	8c	Forgot Password: User clicks "Forgot Password". System initiates password recovery flow (UC-Recovery).
Quality Requirements	**Step**	**Requirement**
	5	Validation feedback should appear within 2 seconds of submission.
	8	Authentication process should take no longer than 3 seconds.
	9	Dashboard should load within 5 seconds after successful login.

UC-02: Admin User Management

<table>
<tr><td><b>Use Case</b></td><td>Admin User Management</td></tr>
<tr><td><b>Goal</b></td><td>To allow the Administrator to manage system users (approve, block, delete, or update roles).</td></tr>
<tr><td><b>Preconditions</b></td><td>Admin is logged in and has "Super User" privileges.</td></tr>
<tr><td><b>Success End Condition</b></td><td>User status is updated in the system database.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>User status remains unchanged; Admin receives an error notification.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Administrator</td></tr>
<tr><td><b>Secondary Actors</b></td><td>System Database</td></tr>
<tr><td><b>Trigger</b></td><td>Admin navigates to the "User Management" section of the dashboard.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Admin selects "User Management" from the navigation menu.</td></tr>
<tr><td>2</td><td>System displays a list of all registered users with their statuses.</td></tr>
<tr><td>3</td><td>Admin searches for a specific user or filters by role/status.</td></tr>
<tr><td>4</td><td>Admin selects a user and chooses an action (e.g., "Approve Vendor", "Ban User").</td></tr>
<tr><td>5</td><td>System prompts for confirmation.</td></tr>
<tr><td>6</td><td>Admin confirms the action.</td></tr>
<tr><td>7</td><td>System updates the user's record in the database.</td></tr>
<tr><td>8</td><td>System notifies the affected user via email (if applicable).</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a</td><td>User Not Found: System displays "No users matching criteria". Admin clears filters.</td></tr>
<tr><td>7a</td><td>Database Error: System fails to update record. Displays "System Error, try again later".</td></tr>
<tr><td>7b</td><td>Self-Modification: Admin tries to ban themselves. System allows "Logout" but prevents "Ban" on self-session.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>2</td><td>User list should load 50 records within 3 seconds.</td></tr>
<tr><td>7</td><td>Database update must be atomic and confirmed immediately.</td></tr>
</table>
</td></tr>
</table>

UC-03: Create Event

<table>
<tr><td><b>Use Case</b></td><td>Create Event</td></tr>
<tr><td><b>Goal</b></td><td>To allow an Event Organizer to create a new event instance in the system.</td></tr>
<tr><td><b>Preconditions</b></td><td>Organizer is logged in based on a valid subscription plan (if applicable).</td></tr>
<tr><td><b>Success End Condition</b></td><td>A new event is created with a unique ID and is visible in the draft/published list.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Event is not created; input data is preserved for correction.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>None</td></tr>
<tr><td><b>Trigger</b></td><td>Organizer clicks the "Create New Event" button.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer clicks "Create Event" on the dashboard.</td></tr>
<tr><td>2</td><td>System presents the Event Creation Wizard (Basic Info, Date/Time, Venue).</td></tr>
<tr><td>3</td><td>Organizer enters Event Title, Description, Type (Corporate, Social, etc.).</td></tr>
<tr><td>4</td><td>Organizer selects Date, Start Time, and End Time.</td></tr>
<tr><td>5</td><td>Organizer enters Venue details or selects "To be decided".</td></tr>
<tr><td>6</td><td>Organizer passes to the next step and saves as "Draft" or clicks "Publish".</td></tr>
<tr><td>7</td><td>System validates all mandatory fields.</td></tr>
<tr><td>8</td><td>System saves the event details and generates a unique Event ID.</td></tr>
<tr><td>9</td><td>System displays "Event Created Successfully" message.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>4a</td><td>Date Conflict: System detects another event at the same venue/time (if venue managed internally). Alerts Organizer.</td></tr>
<tr><td>6a</td><td>Cancel Creation: Organizer clicks "Cancel". System prompts "Save as Draft?".</td></tr>
<tr><td>7a</td><td>Missing Fields: System creates alert highlighting missing mandatory fields.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>7</td><td>Field validation must be real-time or occur immediately upon submission.</td></tr>
<tr><td>8</td><td>Event creation confirmation must appear within 2 seconds.</td></tr>
</table>
</td></tr>
</table>

UC-04: Update & Manage Event

<table>
<tr><td><b>Use Case</b></td><td>Update & Manage Event</td></tr>
<tr><td><b>Goal</b></td><td>To allow Organizers to modify details of an existing event.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event exists and Organizer has ownership permissions.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Event details are updated and saved.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Changes are discarded; Event remains in previous state.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Registered Attendees (Notification recipients)</td></tr>
<tr><td><b>Trigger</b></td><td>Organizer selects an event and clicks "Edit".</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer navigates to "My Events" list.</td></tr>
<tr><td>2</td><td>Organizer selects the specific event and clicks "Edit".</td></tr>
<tr><td>3</td><td>System loads current event details into editable fields.</td></tr>
<tr><td>4</td><td>Organizer modifies necessary information (e.g., changes time, updates description).</td></tr>
<tr><td>5</td><td>Organizer clicks "Save Changes".</td></tr>
<tr><td>6</td><td>System validates the changes.</td></tr>
<tr><td>7</td><td>System updates the event record.</td></tr>
<tr><td>8</td><td>IF event is published, System prompts "Notify Attendees of changes?".</td></tr>
<tr><td>9</td><td>Organizer confirms. System triggers notifications.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>6a</td><td>Validation Failure: New data is invalid (e.g., End time before Start time). System shows error.</td></tr>
<tr><td>9a</td><td>No Notification: Organizer chooses not to notify attendees. System saves silently.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>3</td><td>Form should pre-fill with existing data in < 1 second.</td></tr>
<tr><td>9</td><td>Notifications should be queued for delivery within 1 minute of confirmation.</td></tr>
</table>
</td></tr>
</table>

UC-05: Event Scheduling

<table>
<tr><td><b>Use Case</b></td><td>Event Scheduling</td></tr>
<tr><td><b>Goal</b></td><td>To define the detailed itinerary or agenda within an event.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event container exists (UC-03).</td></tr>
<tr><td><b>Success End Condition</b></td><td>A detailed schedule (sessions, breaks, speakers) is attached to the event.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Schedule is incomplete or not saved.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Speakers/Performers (for assignment)</td></tr>
<tr><td><b>Trigger</b></td><td>Organizer selects "Manage Schedule" for an event.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer enters the "Schedule/Agenda" section of the event.</td></tr>
<tr><td>2</td><td>Organizer adds a "Session" or "Activity".</td></tr>
<tr><td>3</td><td>Organizer defines Session Title, Time Slot, and Description.</td></tr>
<tr><td>4</td><td>Organizer assigns a Speaker/Performer to the session (optional).</td></tr>
<tr><td>5</td><td>Organizer repeats steps 2-4 for all activities.</td></tr>
<tr><td>6</td><td>Organizer clicks "Publish Schedule".</td></tr>
<tr><td>7</td><td>System saves the schedule items and updates the public event page.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a</td><td>Time Overlap: System detects overlapping sessions in the same track. Warns user.</td></tr>
<tr><td>4a</td><td>Speaker Conflict: Selected speaker is booked elsewhere. System flags conflict.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>7</td><td>Drag-and-drop schedule adjustments (if available) should react under 100ms.</td></tr>
</table>
</td></tr>
</table>

UC-06: Budget Planning

<table>
<tr><td><b>Use Case</b></td><td>Budget Planning</td></tr>
<tr><td><b>Goal</b></td><td>To create and track a financial budget for the event, including projected and actual expenses.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event exists.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Budget is defined and balanced.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Budget data is lost.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>None</td></tr>
<tr><td><b>Trigger</b></td><td>Organizer accesses "Budget" tab.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer opens "Budget Manager".</td></tr>
<tr><td>2</td><td>System displays detailed categories (Venue, Catering, Marketing).</td></tr>
<tr><td>3</td><td>Organizer sets "Total Budget Amount".</td></tr>
<tr><td>4</td><td>Organizer adds line items for estimated costs.</td></tr>
<tr><td>5</td><td>System calculates "Remaining Budget" in real-time.</td></tr>
<tr><td>6</td><td>As expenses occur, Organizer enters "Actual Cost".</td></tr>
<tr><td>7</td><td>System updates variance (Estimated vs Actual).</td></tr>
<tr><td>8</td><td>Organizer saves the budget report.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>5a</td><td>Over Budget: Total estimated costs exceed Total Budget. System highlights negative balance in red.</td></tr>
<tr><td>6a</td><td>Unexpected Expense: Organizer adds a new category "Miscellaneous".</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>5</td><td>Calculations must be accurate to 2 decimal places.</td></tr>
<tr><td>7</td><td>Visual indicators (charts/graphs) should update immediately upon data entry.</td></tr>
</table>
</td></tr>
</table>

UC-07: Vendor Registration & Approval

<table>
<tr><td><b>Use Case</b></td><td>Vendor Registration & Approval</td></tr>
<tr><td><b>Goal</b></td><td>To allow third-party vendors to sign up and offer services on the platform.</td></tr>
<tr><td><b>Preconditions</b></td><td>System is online.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Vendor account is created and pending approval or active.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Registration rejected or abandoned.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Vendor</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Administrator (Approver)</td></tr>
<tr><td><b>Trigger</b></td><td>Vendor clicks "Become a Vendor".</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Vendor navigates to "Vendor Registration" page.</td></tr>
<tr><td>2</td><td>Vendor enters business details (Name, Service Type, License No., Location).</td></tr>
<tr><td>3</td><td>Vendor uploads identification/verification documents.</td></tr>
<tr><td>4</td><td>Vendor submits application.</td></tr>
<tr><td>5</td><td>System changes status to "Pending Approval".</td></tr>
<tr><td>6</td><td>Administrator reviews application (linked to UC-02/Admin Work).</td></tr>
<tr><td>7</td><td>Administrator approves vendor.</td></tr>
<tr><td>8</td><td>System notifies Vendor of success.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a</td><td>Incomplete Data: System prevents submission.</td></tr>
<tr><td>7a</td><td>Rejection: Admin rejects application due to "Invalid Documents". System emails Vendor with reason.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>3</td><td>Document upload supports PDF/JPG up to 10MB.</td></tr>
<tr><td>8</td><td>Approval notification email sent within 5 minutes of Admin action.</td></tr>
</table>
</td></tr>
</table>

UC-08: Vendor Service Request

<table>
<tr><td><b>Use Case</b></td><td>Vendor Service Request</td></tr>
<tr><td><b>Goal</b></td><td>To allow Organizers to request services (catering, photography) from registered vendors.</td></tr>
<tr><td><b>Preconditions</b></td><td>Vendor is active; Organizer has an active event.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service request is sent to the Vendor.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Request fails to send.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Vendor</td></tr>
<tr><td><b>Trigger</b></td><td>Organizer browses Vendor Marketplace.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer browses/searches for vendors by category (e.g., Catering).</td></tr>
<tr><td>2</td><td>Organizer views Vendor Profile (Reviews, Pricing, Portfolio).</td></tr>
<tr><td>3</td><td>Organizer clicks "Request Quote" or "Book Service".</td></tr>
<tr><td>4</td><td>Organizer provides event details (Date, Pax, Special Requirements).</td></tr>
<tr><td>5</td><td>Organizer sends request.</td></tr>
<tr><td>6</td><td>System notifies Vendor of new lead.</td></tr>
<tr><td>7</td><td>Vendor accepts request and provides quote.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1a</td><td>No Results: System suggests "Broaden your search".</td></tr>
<tr><td>7a</td><td>Vendor Unavailable: Vendor declines request. System notifies Organizer to find alternative.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>6</td><td>Vendor must receive notification via Dashboard and Email instantly.</td></tr>
</table>
</td></tr>
</table>

UC-09: Team Member Management

Use Case	Team Member Management	
Goal	To allow Organizers to form a team and assign roles for event execution.	
Preconditions	Organizer has an event.	
Success End Condition	Team members are added and have appropriate access rights.	
Failed End Condition	Invitation fails.	
Primary Actors	Event Organizer	
Secondary Actors	Invited User	
Trigger	Click "Manage Team".	
Description / Main Success Scenario	**Step**	**Action**
	1	Organizer goes to "Team" settings.
	2	Organizer enters email of user to invite.
	3	Organizer selects role/permissions (e.g., "Moderator", "Check-in Staff").
	4	System sends invitation link.
	5	Invited User clicks link and accepts.
	6	System adds User to the Event Team list.
Alternative Flows	**Step**	**Branching Action**
	2a	User Unregistered: System prompts "User needs to register first" or sends registration invite.
	5a	Invite Expiration: Link expires after 48h. User requests new link.
Quality Requirements	**Step**	**Requirement**
	6	Permissions must update immediately upon acceptance.

UC-10: Task Assignment

<table>
<tr><td><b>Use Case</b></td><td>Task Assignment</td></tr>
<tr><td><b>Goal</b></td><td>To assign specific operational tasks to team members.</td></tr>
<tr><td><b>Preconditions</b></td><td>Team members exist (UC-09).</td></tr>
<tr><td><b>Success End Condition</b></td><td>Task is assigned to a specific member with a deadline.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Task remains unassigned.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer / Team Lead</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Team Member (Assignee)</td></tr>
<tr><td><b>Trigger</b></td><td>"Add Task" button.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer opens "Kanban Board" or "Task List".</td></tr>
<tr><td>2</td><td>Organizer creates new task (e.g., "Confirm Sound Check").</td></tr>
<tr><td>3</td><td>Organizer assigns task to Team Member "Alex".</td></tr>
<tr><td>4</td><td>Organizer sets due date and priority (High/Medium/Low).</td></tr>
<tr><td>5</td><td>System saves task.</td></tr>
<tr><td>6</td><td>System notifies "Alex" of new assignment.</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a</td><td>Reassignment: Organizer changes assignee from "Alex" to "Sam". System updates notifications.</td></tr>
</table></td></tr>
<tr><td><b>Quality Requirements</b></td><td><table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>6</td><td>Push notif. sent to mobile app (if applicable) within 5 seconds.</td></tr>
</table></td></tr>
</table>

UC-11: Task Update

<table>
<tr><td><b>Use Case</b></td><td>Task Update</td></tr>
<tr><td><b>Goal</b></td><td>To track progress of tasks by updating their status.</td></tr>
<tr><td><b>Preconditions</b></td><td>Task exists.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Task status moves (e.g., To Do -> In Progress -> Done).</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Status update fails.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Team Member (Assignee)</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Team Lead (Watcher)</td></tr>
<tr><td><b>Trigger</b></td><td>Member works on task.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Team Member logs in and views "My Tasks".</td></tr>
<tr><td>2</td><td>Member selects a task ("Confirm Sound Check").</td></tr>
<tr><td>3</td><td>Member adds a comment or attachment (proof of work).</td></tr>
<tr><td>4</td><td>Member changes status to "Completed".</td></tr>
<tr><td>5</td><td>System timestamps the completion.</td></tr>
<tr><td>6</td><td>System notifies Team Lead.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>4a</td><td>Blocker: Member marks task "Blocked". enters reason. System alerts Lead.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>5</td><td>Audit log of task changes must be preserved.</td></tr>
</table>
</td></tr>
</table>

UC-12: Sponsor Registration & Management

<table>
<tr><td><b>Use Case</b></td><td>Sponsor Registration & Management</td></tr>
<tr><td><b>Goal</b></td><td>To manage event sponsors, their tiers (Gold, Silver), and logo visibility.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event is created.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Sponsors are listed on the event page.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Sponsor data not saved.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Event Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Sponsor (Entity)</td></tr>
<tr><td><b>Trigger</b></td><td>"Add Sponsor" action.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer accesses "Sponsors" section.</td></tr>
<tr><td>2</td><td>Organizer creates sponsorship tiers (e.g., Gold - $5000).</td></tr>
<tr><td>3</td><td>Organizer adds Sponsor Name, Logo, and Website URL.</td></tr>
<tr><td>4</td><td>Organizer assigns Sponsor to a Tier.</td></tr>
<tr><td>5</td><td>System formats and displays Sponsor Logos on the Event Landing Page.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a</td><td>Logo Resize: System auto-resizes large images to fit standardized slots.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>5</td><td>Logos must differ in size based on Tier (Gold > Silver) automatically.</td></tr>
</table>
</td></tr>
</table>

UC-13: Attendee Registration

<table>
<tr><td><b>Use Case</b></td><td>Attendee Registration</td></tr>
<tr><td><b>Goal</b></td><td>To allow users to purchase tickets or RSVP for an event.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event is Published. Registration is Open.</td></tr>
<tr><td><b>Success End Condition</b></td><td>User receives a ticket/QR code. Organizer list updates.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Transaction fails; Ticket not issued.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Attendee</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Payment Gateway</td></tr>
<tr><td><b>Trigger</b></td><td>"Get Tickets" button.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Attendee visits Event Page.</td></tr>
<tr><td>2</td><td>Attendee selects Ticket Type and Quantity.</td></tr>
<tr><td>3</td><td>Attendee fills in checkout details.</td></tr>
<tr><td>4</td><td>IF paid event: User is directed to Payment Gateway.</td></tr>
<tr><td>5</td><td>User completes payment.</td></tr>
<tr><td>6</td><td>System generates unique Ticket ID/QR Code.</td></tr>
<tr><td>7</td><td>System emails Ticket to Attendee.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a</td><td>Sold Out: System disables selection. Shows "Join Waitlist".</td></tr>
<tr><td>4a</td><td>Payment Fail: Gateway returns error. System prompts "Retry Payment".</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>7</td><td>Ticket email must arrive within 2 minutes of confirmed payment.</td></tr>
</table>
</td></tr>
</table>

UC-14: Attendance Management (Check-in)

<table>
<tr><td><b>Use Case</b></td><td>Attendance Management</td></tr>
<tr><td><b>Goal</b></td><td>To track usage of tickets and actual entry of attendees.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event is Live. Attendee has valid QR code.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Attendee is marked "Checked In".</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Entry denied.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Check-in Staff</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Attendee</td></tr>
<tr><td><b>Trigger</b></td><td>Attendee arrives at venue.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Check-in Staff opens "Scanner" feature on Check-in App.</td></tr>
<tr><td>2</td><td>Staff scans Attendee's QR Code.</td></tr>
<tr><td>3</td><td>System verifies code against database.</td></tr>
<tr><td>4</td><td>System returns "Valid Ticket" & "Attendee Name".</td></tr>
<tr><td>5</td><td>System marks ticket as "Used".</td></tr>
<tr><td>6</td><td>Staff allows entry.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>4a</td><td>Duplicate Entry: System shows "Already Checked In". Staff investigates.</td></tr>
<tr><td>4b</td><td>Invalid Code: System shows "Invalid Ticket". Entry denied.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>3</td><td>Scan verification time < 1 second to prevent queue buildup.</td></tr>
</table>
</td></tr>
</table>

UC-15: Feedback Collection

<table>
<tr><td><b>Use Case</b></td><td>Feedback Collection</td></tr>
<tr><td><b>Goal</b></td><td>To gather ratings and reviews from attendees after the event.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event has ended.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Feedback is stored and aggregated.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Form submission fails.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Attendee</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Organizer</td></tr>
<tr><td><b>Trigger</b></td><td>Event End Time passed + 2 hours.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>System automatically emails "Rate your Experience" link to Attendees.</td></tr>
<tr><td>2</td><td>Attendee clicks link.</td></tr>
<tr><td>3</td><td>Attendee rates aspects (Venue, Content, Food) 1-5 stars.</td></tr>
<tr><td>4</td><td>Attendee leaves text comments.</td></tr>
<tr><td>5</td><td>Attendee submits.</td></tr>
<tr><td>6</td><td>System aggregates data for Organizer Report.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1a</td><td>Manual Trigger: Organizer manually clicks "Send Feedback Request".</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>6</td><td>Aggregated score updates on Dashboard instantly.</td></tr>
</table>
</td></tr>
</table>

UC-16: Notification System

<table>
<tr><td><b>Use Case</b></td><td>Notification System</td></tr>
<tr><td><b>Goal</b></td><td>To broadcast important updates to relevant users.</td></tr>
<tr><td><b>Preconditions</b></td><td>Users have opted-in (usually default).</td></tr>
<tr><td><b>Success End Condition</b></td><td>Message delivered via App/Email/SMS.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Message bounces.</td></tr>
<tr><td><b>Primary Actors</b></td><td>System / Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>All Users</td></tr>
<tr><td><b>Trigger</b></td><td>Status change, Reminder, or Manual Announcement.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Trigger Event occurs (e.g., "Event Starts in 1 Hour").</td></tr>
<tr><td>2</td><td>System identifies target audience (e.g., Registered Attendees).</td></tr>
<tr><td>3</td><td>System constructs message payload.</td></tr>
<tr><td>4</td><td>System dispatches to configured channels (Push, Email).</td></tr>
<tr><td>5</td><td>User receives notification.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>4a</td><td>Channel Fail: Push fails -> Fallback to Email.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>4</td><td>Bulk dispatch throughput > 1000 messages/minute.</td></tr>
</table>
</td></tr>
</table>

UC-17: Document Upload

<table>
<tr><td><b>Use Case</b></td><td>Document Upload</td></tr>
<tr><td><b>Goal</b></td><td>To share resources (slides, brochures) with attendees.</td></tr>
<tr><td><b>Preconditions</b></td><td>Organizer has file.</td></tr>
<tr><td><b>Success End Condition</b></td><td>File hosted and downloadable.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Upload fail.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Organizer</td></tr>
<tr><td><b>Secondary Actors</b></td><td>Attendee (Consumer)</td></tr>
<tr><td><b>Trigger</b></td><td>"Upload Resource".</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Organizer selects event session.</td></tr>
<tr><td>2</td><td>Organizer drags and drops file (e.g., "Presentation.pdf").</td></tr>
<tr><td>3</td><td>System scans for viruses.</td></tr>
<tr><td>4</td><td>System uploads to cloud storage.</td></tr>
<tr><td>5</td><td>System generates download link on Event Page.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a</td><td>Virus Detected: System rejects file. Warns User.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>4</td><td>Upload progress bar shown for files > 1MB.</td></tr>
</table>
</td></tr>
</table>

UC-18: Report Generation

<table>
<tr><td><b>Use Case</b></td><td>Report Generation</td></tr>
<tr><td><b>Goal</b></td><td>To export data regarding event performance (Sales, Attendance, Feedback).</td></tr>
<tr><td><b>Preconditions</b></td><td>Event has data.</td></tr>
<tr><td><b>Success End Condition</b></td><td>PDF/CSV file downloadable.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Generation error.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Organizer / Admin</td></tr>
<tr><td><b>Secondary Actors</b></td><td>None</td></tr>
<tr><td><b>Trigger</b></td><td>"Export Report" button.</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>User selects Report Type (e.g., "Financial Summary").</td></tr>
<tr><td>2</td><td>User selects Date Range.</td></tr>
<tr><td>3</td><td>System queries database and compiles statistics.</td></tr>
<tr><td>4</td><td>System generates visual charts and tables.</td></tr>
<tr><td>5</td><td>System formats into PDF/CSV.</td></tr>
<tr><td>6</td><td>User downloads file.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a</td><td>No Data: System generates empty report with "No Data Found" watermark.</td></tr>
</table>
</td></tr>
<tr><td><b>Quality Requirements</b></td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>5</td><td>Report generation should typically take < 10 seconds for standard events.</td></tr>
</table>
</td></tr>
</table>

UC-19: Event Closure & Logout

<table>
<tr><td><b>Use Case</b></td><td>Event Closure & Logout</td></tr>
<tr><td><b>Goal</b></td><td>To formally close an event and securely terminate the user session.</td></tr>
<tr><td><b>Preconditions</b></td><td>Event date passed OR User wants to leave.</td></tr>
<tr><td><b>Success End Condition</b></td><td>Event state = "Closed" OR User session destroyed.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>None.</td></tr>
<tr><td><b>Primary Actors</b></td><td>Organizer / Any User</td></tr>
<tr><td><b>Secondary Actors</b></td><td>System</td></tr>
<tr><td><b>Trigger</b></td><td>"Close Event" or "Logout".</td></tr>
<tr><td><b>Description / Main Success Scenario</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>(Flow A: Close Event) Organizer clicks "Close Event".</td></tr>
<tr><td>2</td><td>System archives event data.</td></tr>
<tr><td>3</td><td>System restricts further registrations/edits.</td></tr>
<tr><td>4</td><td>(Flow B: Logout) User clicks "Logout".</td></tr>
<tr><td>5</td><td>System invalidates session token.</td></tr>
<tr><td>6</td><td>System redirects to Homepage.</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>4a</td><td>Auto-Logout: System detects inactivity > 30mins. Logs out user automatically.</td></tr>
</table></td></tr>
<tr><td><b>Quality Requirements</b></td><td><table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>5</td><td>Session data must be completely cleared from browser cache/storage directives.</td></tr>
</table></td></tr>
</table>

Software Requirement Specification

FR001 – Registration & Login

Field	Description
Description	After accessing the system through a web browser or mobile application, users must register using valid information. After successful registration, users must log in using their credentials. Only authorized users are allowed to access the system based on their assigned roles.
Stakeholder	Admin, Event Organizer, Team Member, Vendor, Sponsor, Attendee

FR002 – Admin User Management

Field	Description
Description	The admin can view, add, update, block, or delete any user account from the system database. Admin has full control over user access and role management.
Stakeholder	Admin

FR003 – Create Event

Field	Description
Description	Event organizers can create a new event by selecting event type, entering event details such as title, date, time, venue, and description. The system stores the event information in the database.
Stakeholder	Event Organizer / Manager

FR004 – Update & Manage Event

Field	Description
Description	Event organizers can edit event information, update schedules, or cancel events. All changes are reflected in real time for related users.
Stakeholder	Event Organizer / Manager

FR005 – Event Scheduling

Field	Description
Description	The system allows organizers to schedule events by setting date, time, and location. Scheduling conflicts are minimized by proper validation.
Stakeholder	Event Organizer / Manager

FR006 – Task Assignment

Field	Description
Description	Event organizers can assign tasks to team members with specific deadlines and priorities. Assigned tasks are visible on team member dashboards.
Stakeholder	Event Organizer / Manager

FR007 – Task Update

Field	Description
Description	Team members can view assigned tasks and update task status such as pending, in progress, or completed.
Stakeholder	Team Member / Employee

FR008 – Team Member Management

Field	Description
Description	The system allows organizers to add, remove, or update team member roles and responsibilities for specific events.
Stakeholder	Event Organizer / Manager

FR009 – Vendor Registration

Field	Description
Description	Vendors can register in the system by providing service details such as catering, decoration, sound system, convention hall, or resort information.
Stakeholder	Vendor / Service Provider

FR010 – Vendor Service Request

Field	Description
Description	Event organizers can request services from registered vendors and view vendor details before selection.
Stakeholder	Event Organizer / Vendor

FR011 – Vendor Approval

Field	Description
Description	Vendors can accept or reject service requests sent by event organizers. Approved services are linked to the event.
Stakeholder	Vendor / Service Provider

FR012 – Budget Planning

Field	Description
Description	Event organizers can create and manage budgets by entering estimated expenses and tracking actual costs.
Stakeholder	Event Organizer / Manager

FR013 – Sponsor Registration

Field	Description
Description	Sponsors can register and provide sponsorship details including contribution amount and sponsorship type.
Stakeholder	Sponsor

FR014 – Sponsor Management

Field	Description
Description	Event organizers can manage sponsor details and associate sponsors with specific events.
Stakeholder	Event Organizer / Sponsor

FR015 – Attendee Registration

Field	Description
Description	Attendees can register for events using the web or mobile app by submitting required personal details.
Stakeholder	Attendee / Participant

FR016 – Attendance Management

Field	Description
Description	Organizers can track attendance by marking present or absent attendees for each event.
Stakeholder	Event Organizer / Team Member

FR017 – Notification System

Field	Description
Description	The system sends notifications and reminders related to event updates, task deadlines, and announcements.
Stakeholder	All Users

FR018 – Document Upload

Field	Description
Description	Users can upload and download event-related documents such as schedules, agreements, and reports.
Stakeholder	Admin, Organizer, Vendor

FR019 – Feedback Collection

Field	Description
Description	Attendees can submit feedback after event completion. Feedback data is stored for analysis.
Stakeholder	Attendee / Organizer

FR020 – Report Generation

Field	**Description**
Description	The system generates reports such as attendance reports, budget summaries, and event performance reports.
Stakeholder	Admin, Event Organizer

FR021 – Event Closure

Field	**Description**
Description	Event organizers can mark events as completed after verifying all tasks, attendance, and budgets.
Stakeholder	Event Organizer / Manager

FR022 – System Logout

Field	**Description**
Description	Users can securely log out from the system to protect their account information.
Stakeholder	All Users

Tools & Technique

Tools

Design, Prototyping & Diagramming Tools

- **Canva** – Used to design project prototypes, presentation visuals, use case diagrams and structured documentation layouts
- **Figma** – Used for UI/UX wireframes, interactive prototypes, and interface design planning
- **Draw.io** – Used to create system diagrams including use case diagrams, flowcharts, and architectural diagrams

Frontend Technologies

- **HTML** – Defines the structure of web pages
- **CSS** – Handles layout, styling, and responsiveness
- **JavaScript** – Enables client-side interactivity

Backend Technology

- **PHP** – Manages server-side logic and request processing

Database

- **MySQL** – Stores user data, event details, tasks, vendors, sponsors, and reports

Version Control & Collaboration

- **Git** – Tracks source code changes and manages development history
- **GitHub** – Hosts the project repository and supports team collaboration

Deployment Platform

- **Vercel** – Used for deploying and hosting the web application with continuous deployment support

Documentation Tools

- **Notion** – Used for requirement documentation, planning notes, and project workflow management
- **Microsoft Word** – Used for formal project documentation and report preparation

Research & Planning Tools

- **ChatGPT** – Assisted in requirement analysis, feature ideation, and documentation drafting
- **Wikipedia** – Used for conceptual understanding and background research
- **Notebook (Google / Local)** – Used to organize research notes and planning ideas

Techniques

Requirement Engineering Techniques

- **Requirement Elicitation** – Gathering system requirements through research and analysis
- **Use Case Modeling** – Identifying user roles and system interactions

System Design Techniques

- **Role-Based Access Control (RBAC)** – Provides different access levels for system users
- **Modular Design Approach** – Divides the system into manageable modules

Development Techniques

- **CRUD Operations** – Implements Create, Read, Update, and Delete operations
- **Form Validation** – Prevents invalid and incomplete data input

Security Techniques

- **Authentication & Authorization** – Ensures secure login and controlled system access

Project Management Techniques

- **Version Control Management** – Tracks development progress using Git and GitHub
- **Iterative Development** – Develops the system in phases with continuous improvement

Testing & Maintenance Techniques

- **Functional Testing** – Verifies system features work as expected
- **Error Handling** – Detects and manages system errors effectively

References

- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education Limited.
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.
- IEEE (Institute of Electrical and Electronics Engineers) Computer Society. (2014). *IEEE Standard for Software Requirements Specifications (IEEE Std 830-1998)*. IEEE.
- UML Documentation. (2023). *Unified Modeling Language (UML) Overview*. Retrieved from https://www.uml.org
- Wikipedia. (2024). *Event Management*. Retrieved from https://en.wikipedia.org/wiki/Event_management
- Wikipedia. (2024). *Use Case Diagram*. Retrieved from https://en.wikipedia.org/wiki/Use_case_diagram
- GeeksforGeeks. (2024). *Software Requirement Specification (SRS)*. Retrieved from https://www.geeksforgeeks.org/software-requirement-specification-srs/
- TutorialsPoint. (2024). *UML Use Case Diagrams*. Retrieved from https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm
- Lucid Software. (2024). *Use Case Diagram Tutorial*. Retrieved from https://www.lucidchart.com/pages/uml-use-case-diagram
- Eventbrite. (2024). *Event Management Basics and Best Practices*. Retrieved from https://www.eventbrite.com/blog/
- Primary Data Collection **(Personal Research)**. (2026).
  Data collected through face-to-face interviews, informal discussions, and direct observation of local event organizers, vendors, and attendees in the surrounding area for requirement elicitation of the *EventHive* system.