Software Requirements and Design Document

for

Eventa X

Prepared by:

Muhammad Umer Khan

Mohammad Hamza Imran

Inam Ullah Shaikh

Aetherion

27 November, 2024

Table of Contents

1. Introduction....1
1.1 Purpose.... 1
1.2 Product Scope.... 1
1.3 Title....1
1.4 Objectives....1
1.5 Problem Statement.... 1
2. Overall Description....2
2.1 Product Perspective.... 2
2.2 Product Functions.... 3
2.3 List of Use Cases.... 3
2.4 Extended Use Cases.... 4
2.5 Use Case Diagram.... 32
3. Other Nonfunctional Requirements.... 33
3.1 Performance Requirements.... 33
3.2 Safety Requirements.... 34
3.3 Security Requirements.... 34
3.4 Software Quality Attributes.... 35
3.5 Business Rules.... 36
3.6 Operating Environment.... 37
3.7 User Interfaces.... 38
4. Domain Model.... 43
5. System Sequence Diagram.... 44
6. Sequence Diagram.... 46
7. Class Diagram.... 49
8. Component Diagram.... 50
9. Package Diagram.... 51
10. Deployment Diagram.... 52

1. Introduction

1.1 Purpose

*The product identified in this Software Requirements Specification (SRS) is **EventaX**, a Collaborative Community Event Management System. This document corresponds to **Release 1.0** of the system.*

1.2 Product Scope

EventaX simplifies event management by offering tools for planning, scheduling, and resource allocation, designed for local communities, organizations, and businesses. Core functionalities include venue booking, attendee registration, and resource coordination, along with integration with third-party payment systems for ticketing. This SRS focuses on the system's foundational features and subsystems, including Event Organizer tools, Admin controls, and User interfaces, setting the groundwork for future enhancements.

1.3 Title

EventaX: Collaborative Community Event Management System

1.4 Objectives

- *Event Creation and Scheduling*
- *Venue Management*
- *Attendee Registration*
- *Resource Coordination*
- *Reporting and Notifications*
- *Personalized invitations*
- *Dynamic Reporting and Analytics*
- *Interactive Attendee Experience*
- *Integrated Payment and Ticketing*

1.5 Problem Statement

Organizing community events is often inefficient due to the use of multiple tools and manual processes, which can lead to disorganization. EventaX solves these problems by providing a centralized platform that simplifies event planning with tools for scheduling, venue management, and attendee registration. This user-friendly system streamlines coordination, enhancing the event experience for both organizers and attendees.

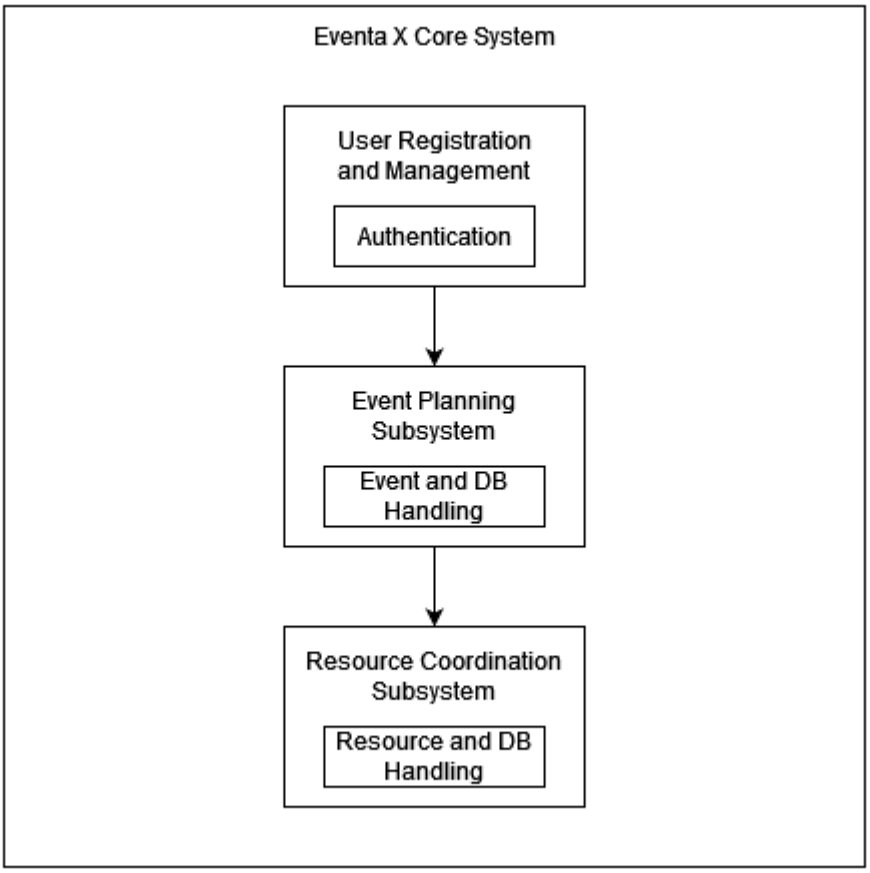
The feasibility of EventaX is high, given its straightforward approach and the widespread need for efficient event management in local communities and small organizations. The Java-based system

with a database backend provides a robust and scalable solution that can be easily adapted to various community needs.

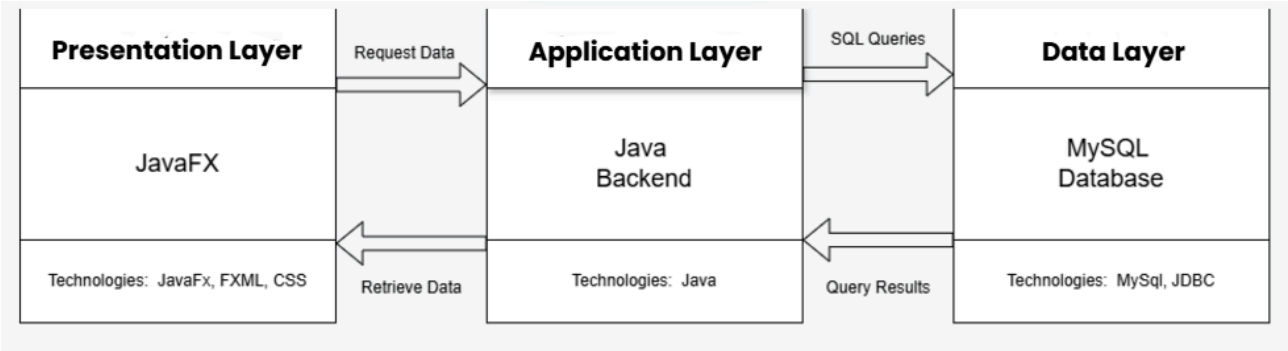
2. Overall Description

2.1 Product Perspective

EventaX is a new, self-contained product designed to address the growing need for an efficient and collaborative event management system tailored for local communities, organizations, and businesses. Unlike traditional systems, EventaX integrates modern tools and features to streamline event planning, scheduling, and resource coordination.
This system is not part of an existing product family but aims to bridge the gap between standalone tools and comprehensive event management solutions. EventaX operates as an independent platform.



3-Tier Architecture



2.2 Product Functions

The EventaX system is designed to streamline event management processes for local communities, organizations, and businesses. The primary functions include:

- *Event Planning and Scheduling:*
  Allows users to create, edit, and schedule events, ensuring efficient coordination.
- *Venue Booking and Resource Allocation:*
  Facilitates booking venues and allocating necessary resources, such as equipment or services, for events.
- *Attendee Registration Management:*
  Enables registration of attendees, tracking participation, and generating attendee reports.
- *Administrative Control and Monitoring:*
  Provides tools for administrators to oversee events, manage users, and ensure smooth execution.
- *Notifications and Reminders:*
  Sends timely updates and alerts to organizers, attendees, and Admin regarding schedules or changes.

2.3 List of Use Cases

- Register Attendees.
- Manage Event.
- Handling Tickets and Payments.
- Allocate and Track Event Resources.
- Remove Event Organizer.
- Add New Venue
- Add Event Organizer.
- Handle Event Budget.
- Manage Attendees.
- Remove Sponsorship for an Event.

- Remove Venue.
- Generate Event Analysis.
- Provide Feedback.
- Create an Event.
- Add Sponsorships.

2.4 Extended Use Cases

1) Name of Use Case: Register Attendees
Scope: EventaX
Level: User Goal
Primary Actor: Attendee
Preconditions:

- The event must be created and listed as available for registration.
- The attendee has access to initiate the registration webpage for the event.

Stakeholders and Their Interests:

- **Attendee**: Wants an easy and secure way to register, receive confirmation, and ensure correct handling of personal data.
- **Event Organizer**: Needs efficient attendee management and up-to-date registration data to prepare for the event.
- **EventaX Admin**: Ensures system reliability and compliance with data protection standards.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) Attendee heads to the sign up page.	2) System loads the registration form specific for the attendees.
3) Attendee fills out the registration form with personal details (name, email, phone, username, password).	4) System validates the form input and ensures all required fields are filled correctly.
5) Attendee submits the registration form	6) System saves the attendee's information and confirms account creation.
7) Attendee logs into the portal.	

Postconditions:

- Attendee information is saved and available in the EventaX database.

.

Extensions:

3a. Attendee provides incomplete or invalid information in the registration form.

1. System highlights the errors and prompts the attendee to correct them.
2. Attendee corrects the errors and resubmits the form.

6a. System experiences a temporary issue and cannot save the registration information.

1. System displays an error message and prompts the attendee to try again.
2. Attendee retries form submission after a brief wait.

Special Requirements:

- The user interface must be accessible on both desktop and mobile devices, with text and controls clearly visible from a distance of 1 meter on larger screens.
- Attendee information should comply with privacy standards and be encrypted for secure storage.
- All user-facing text should support multiple languages, with options for language selection at the start of the registration process.

Technology and Data Variations List:

3a. Attendee identification and verification can be completed via scanning a QR code or manually entering details through the keyboard.

2) Use Case Name: Manage Event
Scope: EventaX
Level: User Goal
Primary Actor: Event Organizer

Preconditions

- The event must already exist in the system.
- The Event Organizer must be authenticated and have access to the event management features.

Stakeholders and Their Interests

- **Event Organizer**: Wants an efficient way to update event details and manage resources for smooth event execution.
- **Attendees**: Expect accurate and up-to-date information about the events they are registered for.
- **EventaX Admin**: Ensures that updates are processed reliably and comply with system protocols.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) Event Organizer logs into the EventaX system.	2) System authenticates the Event Organizer and displays the event management dashboard.
3) Event Organizer checks the list of existing events.	4) System displays a list of created events with details (e.g., name, date, venue).
5) Event Organizer selects an existing event to manage.	6) System loads the selected event's details and displays editable fields (e.g., name, date, time, venue, resources).
7) Event Organizer updates the necessary event information, such as modifying details or removing events.	8) System validates the changes and checks for any conflicts or errors.
9) The Event Organizer confirms the updates to the event.	10) System saves the changes and updates relevant records. If applicable, it sends update notifications to registered attendees.
11) Event Organizer receives confirmation that the event details have been successfully updated.	

Postconditions

- Event details are saved and updated in the EventaX database.
- Notifications, if applicable, are sent to attendees informing them of any changes.
- Event resources are updated as needed, reflecting the latest adjustments.

Extensions

3a. Event Organizer attempts to manage an event that has been canceled or is fully booked.

1) System notifies the Event Organizer of the event's current status.

2) The Event Organizer decides whether to edit the event details or to take other actions based on the status.

7a. Event Organizer provides incomplete or conflicting information.

1) System highlights the errors and prompts the Event Organizer to correct them.

2) Event Organizer corrects the errors and re-submits the form.

9a. System experiences a temporary issue and cannot save the updates.

1) System displays an error message and prompts the Event Organizer to try again.

2) Event Organizer retries after a brief wait.

10a. System fails to notify attendees due to an email service error.

1) System logs the failure and attempts to resend the notifications.

Special Requirements

- The user interface should be accessible on both desktop and mobile devices, with controls optimized for touch and mouse input.
- All user-facing text should support multiple languages, with options for language selection at the start of the registration process.
- Event details and attendee data must comply with data privacy regulations, including secure storage and encrypted transmission.

Technology and Data Variations List

7a. Event updates may include digital asset uploads (e.g., promotional materials or event banners):

9a. Notifications may be sent via SMS in addition to email

3) Name of Use Case: Handling Tickets and Payments
Scope: EventaX
Level: User Goal
Primary Actor: Attendee

Preconditions:

- The event must be created, and tickets must be available for purchase.
- The Attendee has access to the ticket purchasing page for the event.

Stakeholders and Their Interests:

- **Attendee**: Wants a secure and straightforward process to purchase tickets, with a confirmation receipt.
- **Event Organizer**: Needs accurate ticket sales data, a reliable payment process, and aims to maximize revenue from ticket sales.
- **EventaX Admin**: Ensures system reliability, payment security, and compliance with data protection standards while aiming for revenue maximization.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) Attendee selects an event and chooses to purchase a ticket.	2) System calculates the total amount and asks payment details.
3) Attendee proceeds to payment.	4) System calculates the total amount and displays.
	5) System prompts the Attendee to enter payment details securely.
6) System prompts the Attendee to enter payment details securely.	7) System processes the payment.
	8) System generates a digital ticket with a unique ticket ID and sends it to the Attendee's notification.
9) Attendee receives confirmation with ticket details and payment receipt.	10) System updates the attendee list and reflects the ticket purchase in the event's database.

Postconditions

- The payment is processed, and the transaction details are saved in the EventaX system.
- The Attendee receives a digital ticket and payment confirmation.
- The available ticket count for the event is updated in real-time.

Extensions

1a. Attendee selects a ticket option that is no longer available.

1) System notifies the Attendee of the unavailability and prompts them to choose another option or exit the process.

7a. System experiences a temporary issue and cannot process the payment.

1) System displays an error message and prompts the Attendee to retry after a brief wait.

Special Requirements

- The system should support multiple payment options, including major credit cards and popular digital wallets.
- All user-facing text should be available in multiple languages with clear error handling in case of payment issues.

Technology and Data Variations List

8a. Ticket delivery options or Ticket collection point for events with physical tickets.

4) Name of Use Case: Allocate and Track Event Resources
Scope: EventaX
Level: User Goal
Primary Actor: Event Organizer

Preconditions:

- The event must be created and available in the system.
- The Event Organizer has access to the event management interface.

Stakeholders and Their Interests:

- **Event Organizer**: Wants to efficiently allocate and monitor resources, ensuring they are sufficient for the event's needs and available when needed.

- **EventaX Admin**: Ensures that the resource allocation process is smooth and accurate, supporting optimal event execution.
- **Attendees**: Indirectly interested, as proper resource allocation enhances their event experience.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) Event Organizer heads to the resource allocation page.	2) System displays a list assigned available resources (e.g., equipment, staff, seat count).
3) Event Organizer selects resources needed for the event.	4) System updates the event resource list and reserves the selected resources.
5) Event Organizer assigns quantities to resources.	6) System saves resource allocations, assigning resources to the event and updating availability status.
7) Event Organizer confirms the resource allocation.	8) System updates and displays real-time status of resources.

Postconditions:

- Resources are successfully allocated to the event and tracked in the EventaX database.
- Event Organizer can monitor resource statuses.

Extensions:

3a. **Required resources are not available**.

1) System alerts the Event Organizer of the resource unavailability.
2) Event Organizer either adjusts the allocation or reschedules conflicting resources.

Special Requirements:

- Resource data should be updated in real-time to reflect current availability.
- Resource allocation should comply with organizational policies and any limitations on resource usage.

Technology and Data Variations List:

3a. Resources may be booked through third-party integrations (e.g., external equipment providers).

8a. Resource tracking can be supported through barcode scanning or RFID, where applicable.

5) Name of Use Case: Remove Event Organizer
Scope: EventaX
Level: User Goal
Primary Actor: EventaX Admin

Preconditions:

- The Event Organizer exists in the system.
- The EventaX Admin has the appropriate permissions to manage user roles.

Stakeholders and Their Interests:

- **EventaX Admin**: Wants to efficiently manage user roles and ensure the platform has the necessary oversight and control over event organizers.
- **Event Organizer**: Interested in a transparent removal process, with adequate notice and explanations if removed.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) EventaX Admin heads to the user management section.	2) System displays a list of all registered Event Organizers with relevant details.
3) EventaX Admin selects the Event Organizer to be removed.	4) System shows detailed information about the selected Event Organizer, including their associated events.
5) EventaX Admin confirms the removal of the Event Organizer.	6) System removes the Event Organizer from the system and reassigns their events, as configured.

Postconditions:

- The Event Organizer is no longer able to access the platform.
- Events managed by the removed organizer are either reassigned to another organizer or archived.

Extensions:

3a. **Event Organizer is linked to active events**.

1) System prompts EventaX Admin to reassign the events to another organizer or archive them.
2) EventaX Admin completes reassignment.

5a. **EventaX Admin cancels the removal**.

1) System exits the removal process and returns to the user management section without making changes.

6a. **System encounters an error during removal**.

1) System displays an error message and logs the issue.
2) EventaX Admin retries or contacts support for assistance.

Special Requirements:

- All events linked to the removed Event Organizer must have clear reassignment or archiving protocols.
- Notifications should explain the reason for removal and provide a contact method for further questions.
- Data associated with the removed Event Organizer should be retained in compliance with data retention policies.

Technology and Data Variations List:

3a. The system may include an option to automatically reassign the removed organizer's events to another organizer based on pre-configured rules.

8a. Notifications to the removed organizer can be sent via email or in-app message, depending on their communication preferences.

6) Name of Use Case: Generate Event Analysis
Scope: EventaX
Level: User Goal
Primary Actor: Event Organizer/Admin

Preconditions:

- The event has been completed and all attendee data has been collected.
- The Event Organizer/Admin has access to the event's data and the analysis tools within EventaX.

Stakeholders and Their Interests:

- **Event Organizer**: Wants to generate insights based on the attendee data to evaluate event success and plan future events.
- **Sponsors**: May want a summary of attendee demographics and event engagement.
- **EventaX Admin**: Ensures accurate and timely generation of reports, while maintaining data privacy.

Main Success Scenario (Basic flow):

Actor Action	System Action
1.Admin navigates to the analytic dashboard for a specific event.	2. System displays event analytic options
3. Admin generated analysis	4. System generates the event analysis, visualizes it through charts or tables, and displays the results.

Postconditions:

- Event analysis is generated and displayed.
- The Event Organizer can export the analysis data for future use.

Extensions:

3a. Event Organizer tries to generate analysis for incomplete data.

1. System notifies the Event Organizer that some event data is missing or incomplete and offers to continue with partial data or wait for updates.

4a. System encounters an error while processing the analysis.

1. System displays an error message and prompts the Event Organizer to try again later.
2. System logs the error for administrators to resolve.

Special Requirements:

- Analysis generation should be fast and capable of handling large datasets without performance issues.
- Generated reports must adhere to data protection regulations, with any personal data anonymized where necessary.
- Visualizations should be clear, customizable, and exportable in multiple formats.

Technology and Data Variations List:

3a. The system may support different types of filters and customizable reports, depending on event size and complexity.

5a. Reports can be exported in formats like PDF, Excel, or CSV for external analysis or sharing.

7) Name of Use Case: Provide Feedback
Scope: EventaX
Level: User Goal
Primary Actor: Attendee

Preconditions:

- The event has concluded, and the feedback form is available.
- The attendee has access to the feedback form via a post-event link or email invitation.

Stakeholders and Their Interests:

- **Attendee**: Wants to provide feedback on the event, including rating their experience and offering suggestions for improvement.
- **Event Organizer**: Needs feedback to evaluate the event's success, improve future events, and address any issues raised by attendees.
- **EventaX Admin**: Ensures the feedback collection system functions smoothly and complies with data protection regulations.

Main Success Scenario (Basic flow):

Actor Action	System Action
1. Attendee navigates to Provide Feedback page	
	2. System displays the feedback form specific to the events attended by the attendee, including rating scales, text boxes, and optional fields.
5. Attendee fill out the feedback form, providing ratings, comments, and suggestions.	6. System validates the input (e.g., mandatory fields, proper formatting) and ensures all required fields are completed.
7. Attendee submits the feedback form.	8. System save the feedback in the system and confirms receipt.

Postconditions:

- Attendee feedback is saved in the EventaX system.
- Event Organizers has access to the feedback data for analysis and reporting.

Extensions:

3a. Attendee skips mandatory fields in the feedback form.

1. System highlights the missed fields and prompts the attendee to complete them before submission.

4a. System encounters a failure while saving feedback.

1. System displays an error message and logs the issue for administrators.
2. Attendee is prompted to retry submission.

6a. Organizer filters feedback by specific categories (e.g., ratings, comments) for more targeted analysis.

Special Requirements:

- Feedback form should be user-friendly and accessible on both desktop and mobile devices.
- System should support various feedback formats (e.g., multiple choice, text input, star ratings).
- Collected feedback should be anonymized to protect attendee identity where required.

Technology and Data Variations List:

3a. Feedback may be collected via different methods, including mobile app, web interface, or email.

5a. Confirmation messages may vary in format (e.g., in-app notification, email) based on user preferences.

8) Name of Use Case: Create Event
Scope: EventaX
Level: User Goal
Primary Actor: Event Organizer

Preconditions:

- The Event Organizer is logged into the system and has the necessary permissions to create an event.
- All necessary resources (venues, dates, etc.) are available for selection within the system.

Stakeholders and Their Interests:

- **Event Organizer**: Wants to create and manage a new event efficiently, including setting up schedules, venue details, and attendee registration options.
- **Attendee**: Wants clear and accurate event details when registering for the event.
- **EventaX Admin**: Ensures the event creation process runs smoothly and that data entered is accurate and meets platform standards.

Main Success Scenario (Basic flow):

Actor action	System Action
1.The event organizer heads to the event Creation page	2. System displays the event creation form, with fields for event name, date, time, venue, description, and registration details.
3. Event Organizer fills out the event details, including name, date, time, and venue.	4. System validates input fields (e.g., proper date format, venue availability) and ensures all mandatory fields are filled.
5. Event Organizer uploads any necessary materials (e.g., event banner, agenda).	6. System saves uploaded files and links them to the event details.
7. Event Organizer configures registration options (e.g., attendee limits, ticket prices, early bird discounts).	8. System loads registration options and ensures data is consistent (e.g., ticket pricing, availability).
9. Event Organizer reviews the event summary and submits the event for creation.	10. System confirms event details and saves them in the system. Sends a success confirmation to the organizer.

Postconditions:

- The event is successfully created and available for viewing and registration by attendees.
- Event details are stored in the system, accessible to both the organizer and attendees.

Extensions:

2a. Event Organizer selects an unavailable date or venue.

1. System notifies the organizer of the conflict and suggests alternative dates or venues.
2. Organizer selects an alternative, and the system updates the event details.

3a. Uploaded files are in an unsupported format or exceed size limits.

1. System displays an error and prompts the organizer to upload files in the correct format or within the size limit.

5a. System experiences a failure during event creation.

1. System displays an error message and prompts the organizer to try again.
2. Event details are saved as a draft for later submission.

Special Requirements:

- The user interface must be responsive and provide real-time feedback for venue and date availability.
- Event creation should support multiple languages for international organizers.
- Uploaded media files should be compressed for efficient storage while maintaining quality.

Technology and Data Variations List:

3a. Event media files can be uploaded via drag-and-drop or by selecting files manually.

5a. Event details may be integrated with third-party calendar systems or shared via social media platforms upon creation.

9) Name of Use Case: Add Sponsorship
Scope: EventaX
Level: User Goal
Primary Actor: Event Organizer

Preconditions:

- The event has already been created, and the Event Organizer has access to edit event details.
- Sponsorship packages are pre-defined or customizable by the Event Organizer.

Stakeholders and Their Interests:

- **Event Organizer**: Wants to secure sponsorship for financial support and brand partnerships for the event.
- **Sponsor**: Expects clear visibility, proper promotion, and returns on investment during the event.
- **Attendee**: May benefit from sponsored amenities or promotions during the event.
- **EventaX Admin**: Ensures that sponsorship details are properly integrated into the system and meet platform guidelines.

Main Success Scenario (Basic flow):

Actor Actions	System Actions
1. Event Organizer navigates to the "Sponsorship" section of an event.	2. System displays options for adding sponsorship, including predefined packages or custom sponsorship options.
3. Event Organizer selects a predefined sponsorship package or creates a custom package.	4. System loads the details of the selected package or prompts for custom sponsorship details.
5. Event Organizer fills in the sponsor details (e.g., sponsor name, contributing amount).	6. System validates the input, ensuring all required fields are complete.
9. Event Organizer reviews and submits the sponsorship details.	10. System confirms the sponsorship, saves the information in the system, and updates the event with sponsor details.

Postconditions:

- Sponsorship details are successfully added to the event.
- Sponsor information is visible to attendees (e.g., logos on event pages, promotional materials).

Extensions:

2a. The Event Organizer wants to modify an existing sponsorship.

1. System allows editing of sponsor details or package terms.
2. Organizer saves the updated sponsorship information, and the system reflects the changes.

3a. The sponsor does not have digital branding materials available.

1. System prompts the organizer to proceed without logos or upload them later.
2. Organizer submits sponsorship without branding materials, which can be updated later.

5a. System encounters an issue saving the sponsorship information.

1. System displays an error message.
2. Organizer is prompted to retry after a brief time.

Special Requirements:

- The sponsorship system must support multiple tiers of sponsorship with varying benefits (e.g., Platinum, Gold, Silver).
- Uploaded sponsor materials should adhere to platform standards (file size limits, supported formats).
- Sponsorship visibility (e.g., logo display, mentions in promotional content) must be customizable by the organizer.

Technology and Data Variations List:

4a. Sponsor materials may be provided via direct upload or linked from an external source (e.g., sponsor's website).

6a. Sponsorship details can be exported for sponsor reporting or promotional purposes.

10) Name of Use Case: Add Venue

Scope: EventaX

Level: User Goal

Primary Actor: Event Organizer

Preconditions:

- The event organizer has logged into their account.
- The event organizer has the necessary permissions to add a venue.

Stakeholders and Their Interests:

- **Event Organizer:** Needs to easily add new venue details, such as location, capacity, and amenities, to support event planning.
- **EventaX Admin:** Ensures that venue data is accurate and complies with system integrity standards.
- **Attendee:** Indirectly benefits from proper venue management, ensuring that events are well-organized.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) The Event Organizer chooses to add a new venue.	2) System prepares to receive the new venue information.
3) Event Organizer provides venue details (e.g., name, location, capacity).	4) System validates the provided details to ensure completeness and accuracy.
5) Event Organizer submits the venue information.	6) System saves the new venue data in the database.
	7) System confirms the successful addition of the venue.
	8) System updates the list of available venues for event bookings.

Postconditions:

- The new venue is successfully added to the EventaX database.
- The venue is available for event organizers to assign to future events.
- Confirmation is sent to the event organizer about the successful addition.

Extensions:

4a. Invalid or Incomplete Venue Information.

1. The system identifies missing or invalid data (e.g., blank fields, invalid capacity).
2. The system prompts the event organizer to correct the information before proceeding.

5a. The system fails during the submission process.

1. It notifies the event organizer of a temporary failure.
2. The system prompts the organizer to retry after a brief waiting period.

Special Requirements:

- The venue data should support multi-language input, allowing users from different regions to input venue information in their local language.
- The system must respond with confirmation within 10 seconds after a venue is added, with 95% success under normal load.
- The system should support bulk venue additions, allowing event organizers to upload multiple venues through an API.
- The system must support integration with external APIs for verifying venue addresses and geographical information.

Technology and Data Variations List:

3a. Venue location verification can be done via manual entry or integration with external verification services.

5a. Venue media uploads (e.g., images or supporting documents) may be stored internally or linked via external cloud services.

11) Name of Use Case: Remove Venue

Scope: EventaX

Level: User Goal

Primary Actor: Event Organizer

Preconditions:

- The event organizer has logged into their account.
- The event organizer has the necessary permissions to remove a venue.

Stakeholders and Their Interests:

- **Event Organizer:** Needs the ability to remove outdated or unused venues to keep the venue list accurate and manageable.
- **EventaX Admin:** Ensures that removing venues does not impact any ongoing events or system integrity.
- **Attendee:** Indirectly benefits from up-to-date venue management, ensuring venue availability is accurate.

Main Success Scenario (Basic flow):

Actor Action	System Action
1) Event Organizer selects a venue to remove from the system.	2) System checks if the venue is associated with any ongoing or upcoming events and sends notifications to attendees attending those events
3) Event Organizer confirms the removal of the venue.	4) System removes the venue from the database;
	5) System confirms that the venue has been successfully removed.
	6) System updates the list of available venues.

Postconditions:

- The venue is successfully removed from the EventaX database.
- The updated list of venues is available for future event planning.
- Confirmation is sent to the event organizer about the successful removal.

Extensions:

2a. Venue Linked to Ongoing or Upcoming Events.

1. The system detects that the venue is associated with one or more ongoing or scheduled events.

2. The system notifies the event organizer that the venue cannot be removed until those events are rescheduled or completed.

4a. The system fails during the removal process.

1. The System notifies the event organizer of a temporary failure.
2. The system prompts the organizer to retry after a brief waiting period.

Technology and Data Variations List:

1a. Venue selection and identification may be performed via manual search, or using unique venue IDs entered through keyboard.

Special Requirements:

- The system must check if the venue is linked to any ongoing events before allowing removal and notify the organizer within 5 seconds.
- System logs should capture the venue removal details, including timestamp, removed by, and associated events.
- The user interface must be accessible on both desktop and mobile devices, with text and controls clearly visible from a distance of 1 meter on larger screens
- The removal process must comply with regional data regulations, ensuring that any personal information linked to the venue is handled appropriately.

12) Name of Use Case: Add Event Organizer

Scope: EventaX

Level: User Goal

Primary Actor: Admin

Preconditions:

- The admin must be authenticated and logged into their account.

Main Success Scenario (Basic Flow):

Actor Action	System Action
1) Admin initiates the process to add a new event organizer.	2) System prompts for the event organizer's details (name, email, phone number etc).
3) Admin inputs the required information.	4) System validates the provided information (e.g., email format, required fields).
	5) System saves the new event organizer's information.
	6) System sends a notification with login credentials to the new event organizer.

Postconditions:

- The new event organizer's information is saved and available in the EventaX database.
- A confirmation message is sent to the new event organizer with their login credentials.

Extensions:

1a. Admin cancels the process.

1. System exits the add event organizer process without making any changes.

3a. Admin provides invalid information.

1. System highlights the errors and prompts the admin to correct them.

2. Admin corrects the errors and resubmits the information.

Special Requirements:

- The system must verify that the event organizer is not linked to any ongoing events before removal, notifying the admin within 5 seconds.
- Removal of event organizers should trigger an update in any related event schedules if applicable.
- The action must be logged, capturing details such as timestamp and admin who performed the removal.
- The system must ensure compliance with regional data regulations when handling personal information.
- User interface must be accessible on both desktop and mobile devices, with clear visibility of text from a distance of 1 meter.

Technology and Data Variations List:

***a**. The system must handle concurrent requests, ensuring multiple admins can add event organizers.

3a. The system must support multi-language input for event organizer details.

4a. Event organizer information can be validated using external APIs (e.g. phone number validation).

13) Name of Use Case: Handle Event Budget

Scope: EventaX

Level: User Goal

Primary Actor: Event Organizer

Preconditions:

- The event must be created and listed as available for budget management.
- The Event Organizer has access to the budget management section of the system.

Stakeholders and Their Interests:

- **Event Organizer**: Wants a simple and accurate way to input, track, and adjust event budgets.
- **EventaX Admin**: Ensures that all financial data is securely stored and properly tracked.
- **Finance Team**: Needs detailed and accurate financial reports for review.

Main Success Scenario (Basic Flow)

Actor Action	System Action
	1) System displays the current budget details for different events.
2) Event Organizer adjusts the budget.	3) System validates the input and ensures all required fields are completed.
4) Event Organizer finalizes and confirms the updated budget.	5) System saves the final budget and updates it.

Postconditions:

- The budget is saved and stored in the EventaX system.
- The total event cost is updated and available for future reference.
- A financial report is generated and can be downloaded.

Extensions:

3a. Event Organizer enters an invalid or incomplete budget entry.
1. System highlights the error and prompts the Event Organizer to correct it.
2. Event Organizer corrects the error and resubmits the budget.

6a. System experiences a temporary failure while saving the budget.
1. System displays an error message indicating the failure to save.
2. System prompts the Event Organizer to retry after a brief wait.
3. Event Organizer retries the submission, and the system attempts to save again.

Special Requirements

- Budget data must be stored securely, ensuring compliance with financial regulations.
- System must support various currencies and allow for currency conversion in real-time.
- Data encryption must be used for financial details during storage and transmission.

Technology and Data Variations List

***a**. Budget information can be manually entered via the keyboard or through a CSV file.
3a. Currency conversion rates can be manually entered by the Event Organizer.
7a. Financial reports can be displayed within the system.
7b. Event Organizer can choose to save the report locally.

14) Name of Use Case: Manage Attendees

Scope: EventaX

Level: User Goal

Primary Actor: Event Organizer

Preconditions:

- The event is created and attendees have been registered.
- The Event Organizer has access to the event's attendee management features.

Stakeholders and Their Interests:

- **Event Organizer**: Wants to efficiently manage attendees, including updating information, removing attendees, and handling attendance reports.
- **EventaX Admin**: Ensures proper data management and compliance with privacy standards.
- **Attendees**: Expect their personal information to be handled securely and be notified of any event updates.

Main Success Scenario (Basic Flow)

Actor Action	System Action
1. Event Organizer selects the event for attendee management.	2. System displays the current list of attendees for the selected event.
	3. System displays the attendees' information.
4. Event Organizer selects an attendee to update or remove.	
5. Event Organizer updates attendee details (e.g., name, email, phone) or removes the attendee from the event.	6. System validates and applies the changes.
7. Event Organizer confirms the updates or removal.	8. System saves the updated information.

Postconditions:

- The attendee information is updated or the attendee is removed from the event.
- The event's available attendee count is updated accordingly.
- An updated attendee list or report is generated and available for review.

Extensions:

3a. Event Organizer selects an attendee who has canceled their registration.
1. System notifies that the attendee is no longer part of the event.
2. Event Organizer selects another attendee or exits the management process.

5a. Event Organizer provides incomplete or invalid updates.
1. System flags the error and prompts for corrections.
2. The Event Organizer corrects the information and resubmits it.

9a. System encounters an issue generating the report.
1. System logs the issue and notifies the Event Organizer.
2. System prompts Event Organizer to retry after a brief wait.

Special Requirements

- The system must comply with data privacy laws and encrypt all attendee personal information.
- Real-time validation of attendee data should be implemented to ensure all fields are correctly filled before confirmation.
- The system should support bulk attendee updates or removals for efficiency.
- Attendance reports should be generated in under 30 seconds, 95% of the time.
- The attendee list and reports should support export options such as CSV, Excel, and PDF formats.

Technology and Data Variations List

***a.** Attendee information can be updated individually or via bulk upload through a CSV file.
5a. Attendee removals can be handled via single attendee selection or multiple selections at once.
9a. Reports can be reviewed directly within the system, or exported for external use (PDF, Excel, etc.).
9b. The updated attendee list can be sent to the Event Organizer or saved within the system as PDF for future reference.

15) Name of Use Case: Remove Sponsorship for an Event

Scope: EventaX

Level: User Goal

Primary Actor: Event Organizer

Preconditions:

- The event organizer has logged into their account.
- The event organizer has the necessary permissions to manage sponsorships for an event.
- The event sponsorships have already been added to the event.

Stakeholders and Their Interests:

Event Organizer: Needs to efficiently remove sponsorships from events to ensure accurate sponsorship management.

Sponsor: Wants confirmation if their sponsorship is removed and to ensure it is done according to agreements.

EventaX Admin: Ensures that sponsorship removal does not affect event integrity or cause discrepancies in sponsorship records.

Main Success Scenario (Basic Flow)

Actor Action	System Action
	1. System retrieves the list of current sponsorships associated with various events.
2. Event Organizer selects the event from which they wish to remove sponsorships.	
3. The Event Organizer chooses the sponsorship to remove.	
4. Event Organizer confirms the removal of the sponsorship.	5. System removes the sponsorship from the event and updates the event's sponsorship details.
	6. System confirms the successful removal of the sponsorship(s) and updates the list of sponsors for the event.

Postconditions:

- The sponsorship is successfully removed from the event's sponsorship list.
- The updated list of sponsors is available for event organizers to view.
- Confirmation is sent to the event organizer about the successful removal of sponsorship.
- Sponsors receive notifications if their sponsorship has been removed.

Extensions:

4a. Sponsorship Linked to Active Promotions or Agreements.

1. The system detects that the sponsorship is tied to ongoing promotions or agreements (e.g., advertising, special benefits).

2. The system notifies the event organizer that the sponsorship cannot be removed until the associated promotions or agreements are resolved or updated.

6a. The system fails during the removal process.

1. System notifies the event organizer of a temporary failure.

2. The system prompts the organizer to retry after a brief waiting period.

Special Requirements:

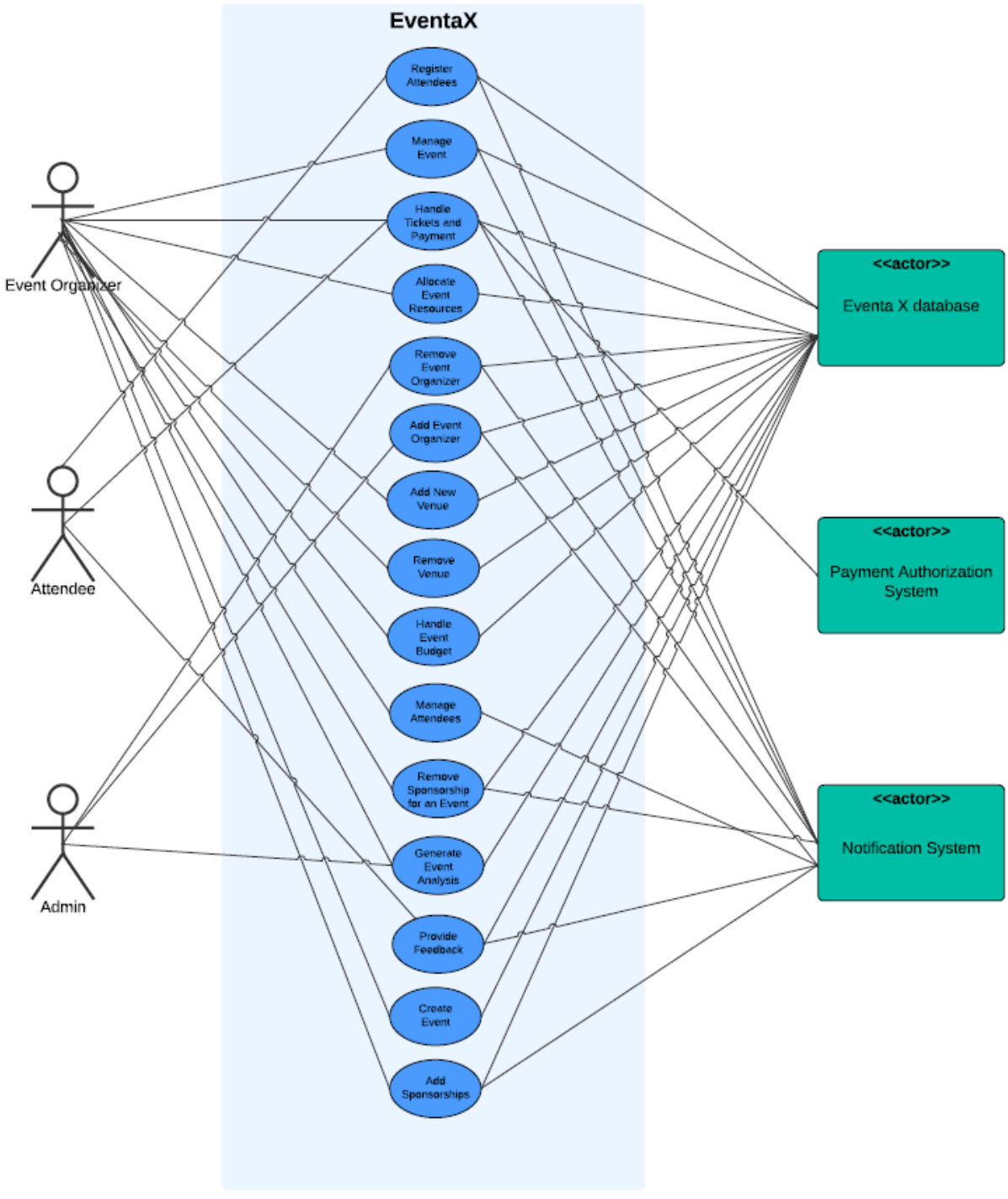
- The system must ensure that all agreements and promotional activities linked to a sponsorship are resolved before allowing removal.
- System logs should capture sponsorship removal details, including timestamps, sponsorship removed, and by whom.
- Notifications must be sent to both the event organizer and the sponsor when a sponsorship is removed.
- The removal process must comply with data privacy regulations concerning sponsor information and agreements.

Technology and Data Variations List:

3a. Sponsorship selection can be performed via a manual list or using unique sponsorship IDs entered through the keyboard.

5a. Removal of sponsorships can trigger automatic notifications to sponsors informing them of the change.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

The performance requirements for EventaX ensure the system operates efficiently under different circumstances and provides a seamless user experience. These requirements are as follows:

1. ***System Responsiveness**:*
    - *The system must respond to user actions (e.g., event creation, registration updates) within **2 seconds** under normal load conditions.*
    - *During peak usage (e.g., ticket sale openings), the response time should not exceed **5 seconds**.*
2. ***Data Processing**:*
    - *The system should process attendee registration updates and payment confirmations in real-time, with a maximum delay of **1 second** for database updates.*
3. ***Notification Delivery**:*
    - *Notifications must be sent within **5 seconds** of a triggering event, such as a schedule change or booking confirmation.*
4. ***Resource Optimization**:*
    - *The system should efficiently handle resource allocation for up to **500 events** scheduled within a single day.*
    - *Venue and resource conflict checks should be completed in under **3 seconds** per query.*
5. ***Availability**:*
    - *The system should maintain an uptime of **99.9%**, with planned maintenance limited to off-peak hours and announced at least **24 hours in advance**.*
6. ***Real-Time Features**:*
    - *For live updates (e.g., attendee count or resource changes), the system must notify the organizer within **1 second** of the change.*

These performance requirements are designed to ensure reliability, scalability, and responsiveness, meeting the needs of diverse users and scenarios while providing a robust event management experience.

3.2 Safety Requirements

The safety requirements for EventaX are designed to minimize the risk of data loss, unauthorized access, and harm resulting from system misuse or failure. These requirements ensure compliance with industry standards and user expectations.

1. ***Data Integrity and Security**:*
    - *The system must prevent unauthorized access to sensitive information, including attendee data, payment details, and event configurations.*
2. ***User Authentication**:*
    - *Regular audits of user permissions must be conducted to ensure only authorized personnel can access specific functions.*
3. ***Error Handling and Recovery**:*
    - *The system must include safeguards to prevent unintended actions, such as deleting critical data.*
    - *In the event of a system failure, users must be directed to a fallback interface, and all unsaved work must be recoverable.*
4. ***Safety Certifications**:*
    - *The system must pass security and usability testing by an accredited third party before deployment.*
5. ***Preventing Harmful Actions**:*
    - *Event organizers must not have access to delete attendee payment data after confirmation.*

These safeguards ensure the system operates securely and complies with relevant regulations, providing a safe environment for all users.

3.3 Security Requirements

The security and privacy requirements for EventaX ensure that all user data and system resources are adequately protected against unauthorized access, misuse, and breaches. These requirements are aligned with best practices and regulatory standards.

1. ***User Identity Authentication**:*
    - *All users must authenticate using unique credentials (username and password) before accessing the system.*
    - *Passwords must adhere to strong security standards, requiring a minimum length of 12 characters.*
2. ***Data Security**:*
    - *All sensitive data, including personal information, payment details, and event records, must be secured in the database.*
3. ***Privacy Protections**:*
    - *User data must be collected and processed in compliance with GDPR, CCPA, and other applicable data privacy laws.*
    - *The system must allow users to access, modify, or delete their personal data as required by applicable laws.*
4. ***Audit and Monitoring**:*
    - *All system notifications and changes must be logged for auditing purposes.*

These measures ensure the confidentiality, integrity, and availability of user data while fostering trust in the system.

3.4 Software Quality Attributes

The following quality attributes outline key expectations for the EventaX system, focusing on meeting user and developer needs. Each characteristic is described to ensure it is specific, measurable, and actionable.

1. ***Adaptability**:*
    - *The system must allow customization of event workflows, notifications, and templates for various event types.*
    - *New features or modules must integrate seamlessly without requiring major architectural changes.*
2. ***Availability**:*
    - *EventaX must maintain a system uptime of **99.9%**, allowing no more than **8 hours of downtime annually**, excluding scheduled maintenance.*
    - *Redundancy measures, including failover servers, must ensure uninterrupted access during critical operations.*
3. ***Correctness**:*
    - *All functionalities, including attendee registration, payment processing, and venue booking, must operate as specified with **zero known defects** in production releases.*
    - *Data accuracy must be ensured, with automated validation for user inputs, such as event dates and payment details.*
4. ***Flexibility**:*
    - *Organizers should have the ability to define custom roles and permissions within their events.*
5. ***Maintainability**:*
    - *The system's codebase must adhere to best practices and include comprehensive documentation for ease of maintenance.*
    - *Bug fixes and updates should be deployable with no more than **30 minutes of downtime per release**.*
6. ***Reliability**:*
    - *The system should recover from server failures within **5 seconds** without data loss.*
    - *Backup data must be stored in multiple geographic locations to prevent catastrophic loss.*
7. ***Reusability**:*
    - *Common modules, such as user authentication and notification services, must be designed for reusability in future enhancements or related projects.*
8. ***Robustness**:*
    - *EventaX must handle invalid inputs, such as duplicate attendee registrations or incorrect payment details, gracefully and provide meaningful error messages.*
    - *The system must withstand stress tests simulating **1,000 concurrent users** without degradation in performance.*
9. ***Testability**:*
    - *The system must include automated test suites for unit, integration, and regression testing, covering at least **90% of the codebase**.*

    - *Manual testing procedures must be documented for critical user workflows, such as event creation and payment processing.*
10. ***Usability**:*
    - *The user interface must enable organizers to set up events within **5 minutes** with minimal training.*
    - *The system must undergo usability testing with a target satisfaction score of **85% or higher** from participants.*

***Relative Preferences**:*

- ***Ease of Use** is prioritized over **Ease of Learning**, as the target audience is expected to be familiar with basic event management concepts.*
- ***Reliability** and **Security** are given precedence over features like **Flexibility** and **Adaptability**, ensuring a stable and secure platform.*

These attributes ensure a well-rounded, high-quality product that meets the needs of both end-users and developers.

3.5 Business Rules

The following operating principles define the roles and responsibilities within EventaX, outlining who can perform specific functions under various circumstances. These principles ensure clarity, accountability, and adherence to system policies.

1. ***Role-Based Access Control (RBAC)**:*
    - ***Event Organizers**:*
        - *Can create, modify, and delete events they own.*
        - *Have full control over attendee management, resource allocation, and ticket pricing.*
        - *Cannot access financial reports for events they do not own.*
    - ***Attendees**:*
        - *Can register for events, make payments, and view their registration details.*
        - *Cannot modify or delete events or access other attendees' information.*
    - ***EventaX Administrators**:*
        - *Can oversee all events on the platform, resolve disputes, and monitor system performance.*
        - *Have authority to suspend or terminate accounts for violations of both event organizers and attendees of terms of service.*
2. ***Data Visibility**:*
    - *Sensitive data, such as attendee details, can only be viewed by Event Organizers with appropriate permissions.*
    - *Public events must display relevant details (e.g., name, date, location) to all users.*
3. ***Resource Allocation**:*
    - *Shared resources, such as venues and equipment, must be booked on a first-come, first-served basis, subject to availability.*

4. ***User Actions**:*
    - *Users must confirm actions with significant impact (e.g., deleting an event, canceling registrations) through a secondary confirmation step.*
    - *Organizers cannot delete events that have active ticket sales or registered attendees without administrative approval.*
5. ***Audit and Accountability**:*
    - *All user actions, such as event creation, attendee registration, and resource booking, are logged with timestamps and user identifiers.*
    - *Logs must be retained for **1 year** and are accessible only to Administrators for auditing purposes.*

These operating principles ensure a structured, fair, and efficient use of the EventaX system while providing clear boundaries for user interactions and system functionality.

3.6 Operating Environment

The EventaX system is designed to operate in a modern, flexible, and scalable environment, ensuring compatibility with various hardware platforms, operating systems, and software components. Below is a detailed description of the environment in which EventaX will operate:

Hardware Platform

- ***Minimum Requirements**:*
    - ***Processor**: Intel Core i5 (or equivalent) with at least 2.5 GHz clock speed.*
    - ***Memory (RAM)**: 4 GB (8 GB recommended for optimal performance).*
    - ***Storage**: 5 GB free disk space (SSD recommended for faster read/write operations).*
- ***Recommended Server Specifications** (for cloud or on-premises deployment):*
    - ***Processor**: Intel Xeon or AMD equivalent, with multi-core support (8+ cores).*
    - ***Memory**: 32 GB or more.*
    - ***Storage**: 1 TB SSD or larger, with redundancy (RAID configuration preferred).*
    - ***Network**: 1 Gbps dedicated internet connection for server-side operations.*

Operating System

EventaX is designed to be cross-platform, ensuring compatibility with both web-based and on-premises installations.

- ***Web Client (End User)**:*
    - *Supported Operating Systems: Windows 10 or newer, macOS (10.12 or newer), Linux (Ubuntu 20.04 or newer, Fedora 34 or newer).*

- ***Server-Side (Backend)**:*
    - ***Primary Supported Operating Systems**:*
        - ***Linux**: Ubuntu 20.04 LTS, CentOS 8, Red Hat Enterprise Linux (RHEL) 8.x, or similar distributions.*
        - ***Windows Server**: Windows Server 2019 or newer.*

Software Components

EventaX is built on modern technologies, ensuring seamless integration and operation across various software components.

- ***Backend Framework**:*
    - *Java framework for efficient backend development and scalability.*
    - ***Database**:*
        - *Relational Database Management Systems (RDBMS), MySQL.*
- ***Frontend Framework**:*
    - *JAVA FX*

Other Software Dependencies

- ***Development Tools**:*
    - *Integrated Development Environment (IDE) such as IntelliJ IDEA.*
    - *Git for version control and GitHub/Bitbucket for repository management.*

This environment ensures that EventaX can operate smoothly, offering high availability, flexibility, and scalability while integrating with common industry tools and technologies.

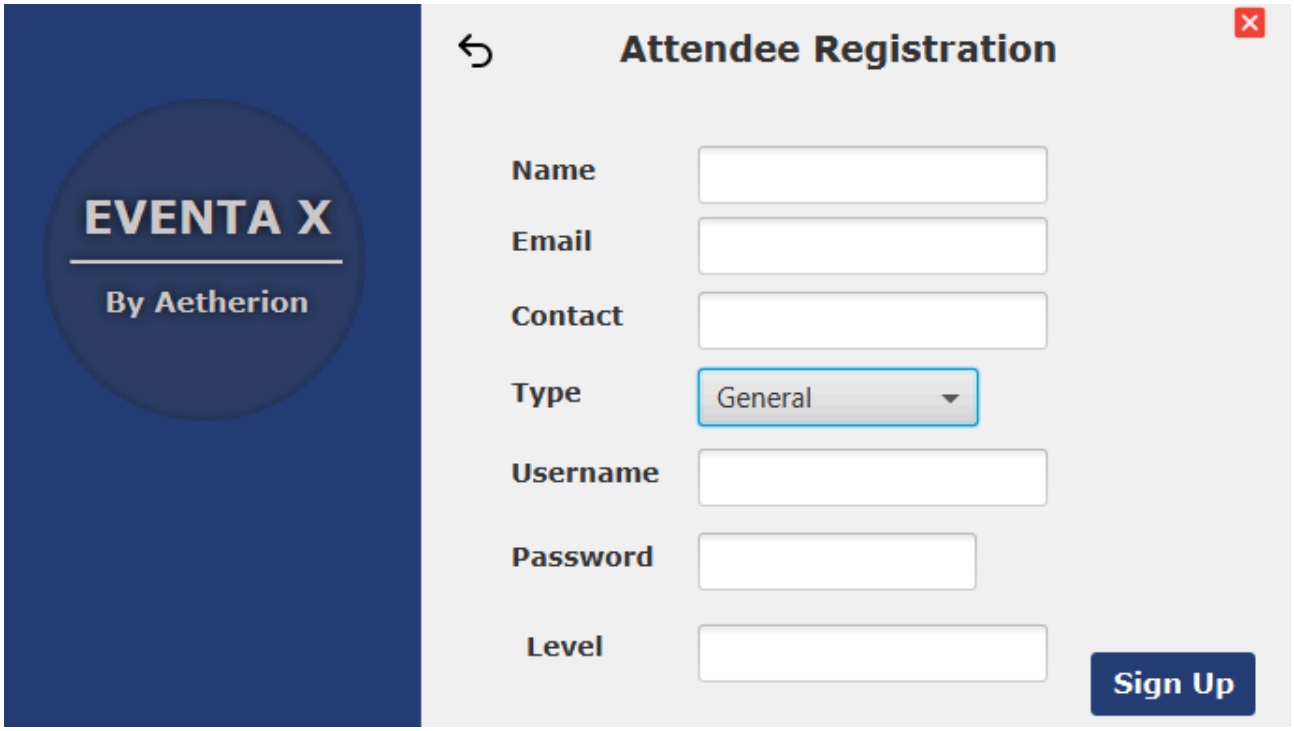
3.7 User Interfaces

The user interface (UI) of EventaX is designed to be intuitive, user-friendly, and accessible to a diverse audience. Below are the logical characteristics of the interfaces between the software and its users, focusing on key features, standards, and layout principles.

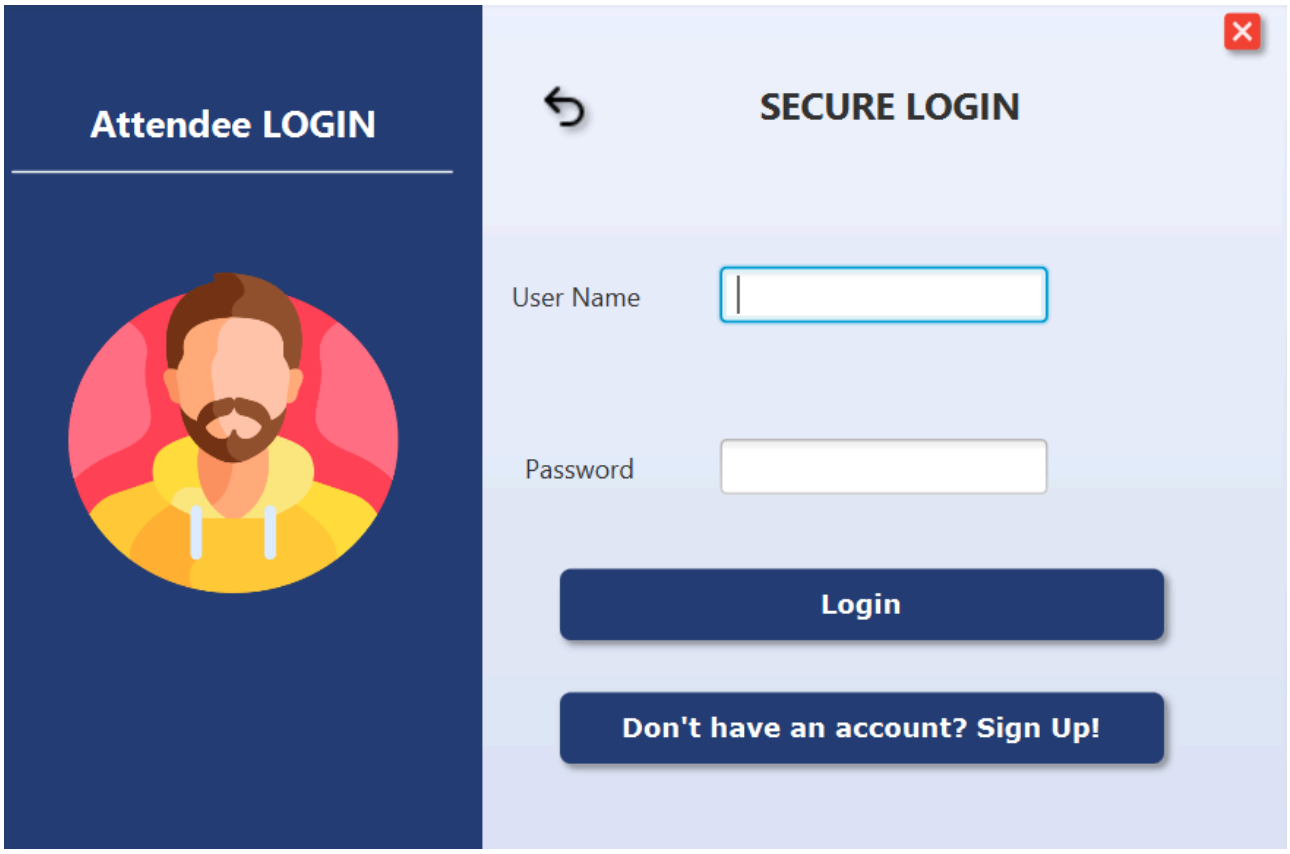
1. Interface Examples from the Program

- ***Home Page***

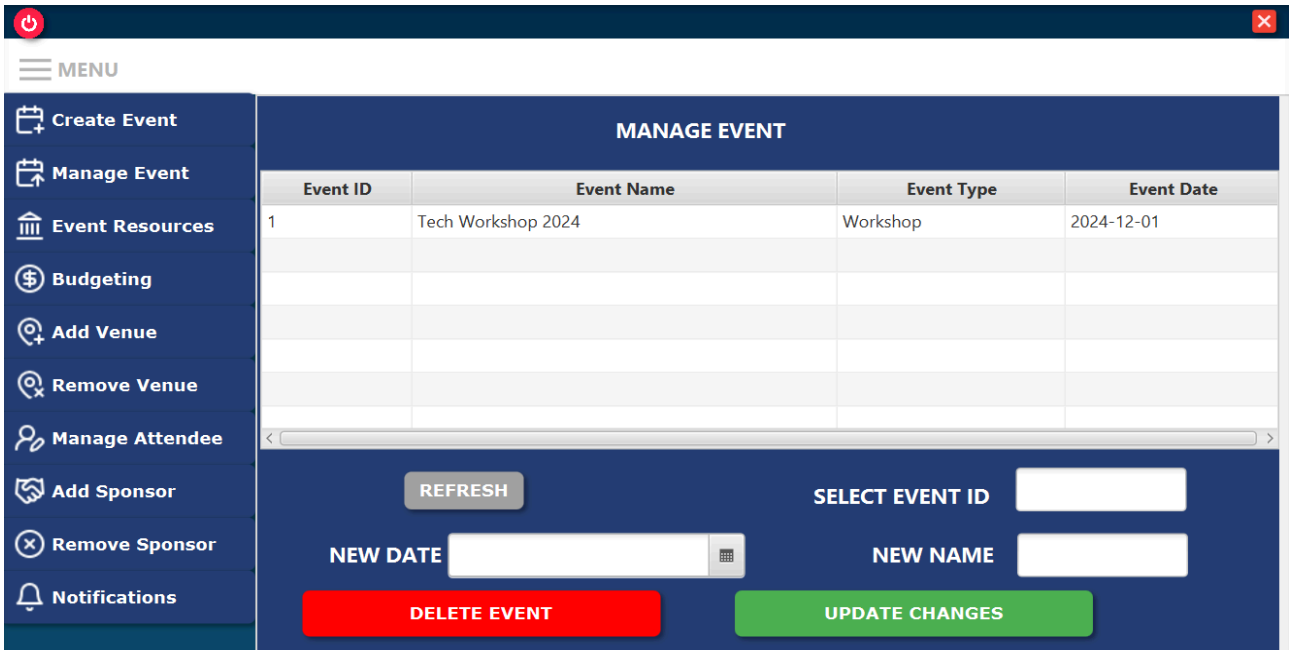
- ***Registration Page***



- ***Login Page***



- ***Dashboard Page***



1. User Interface Components

The system has different types of users, each with a dedicated interface. The following software components require user interfaces:

- ***Event Organizer Dashboard***
  - ***Features**: Event creation, event modification, resource management, attendee management, financial reports, and ticketing setup.*
  - ***Access**: Full access to events they create and manage, with options to create, update, and delete events.*
- ***Attendee Portal***
  - ***Features**: Registration for events, payment processing, personal account management, ticket viewing, and event notifications.*
  - ***Access**: Access to public events and their registration details. Restricted from modifying event data.*
- ***Administrator Panel***
  - ***Features**: Overview of all events, user management (suspension or termination), dispute resolution, audit logs, and system settings.*
  - ***Access**: Administrative control over the entire system for monitoring, user management, analytics, and system health.*

2. Screen Layout Constraints

- ***Consistent Layout**:*
  - *The UI follows a **three-column layout**:*
    - ***Left Sidebar**: Contains navigation options (e.g., Dashboard, Events, Users, Reports).*
    - ***Main Content Area**: Displays context-specific information and data.*
  - ***Responsiveness**: The UI will adapt to various screen sizes.*
  -

3. GUI Standards and Style Guides

- ***Color Scheme**:*
  - *The system adopts a modern **dark theme** (light text on dark background) as the primary design.*
- ***Typography**:*
  - ***Primary Font**: Verdana (for body text and headings), size 14px for content, 18px for titles.*
  - ***Secondary Font**: Roboto (for buttons and labels) for readability across platforms.*
- ***Button Style**:*
  - ***Standard Buttons**: Rounded corners (8px radius), with primary buttons in blue (#0056b3).*
- ***Icons**:*
  - *Use of flat, minimalistic icons to represent functions like "Save," "Cancel," "Edit," "Delete," etc.*
  - *Icons will align with the Material Design standard to ensure consistency and intuitiveness.*

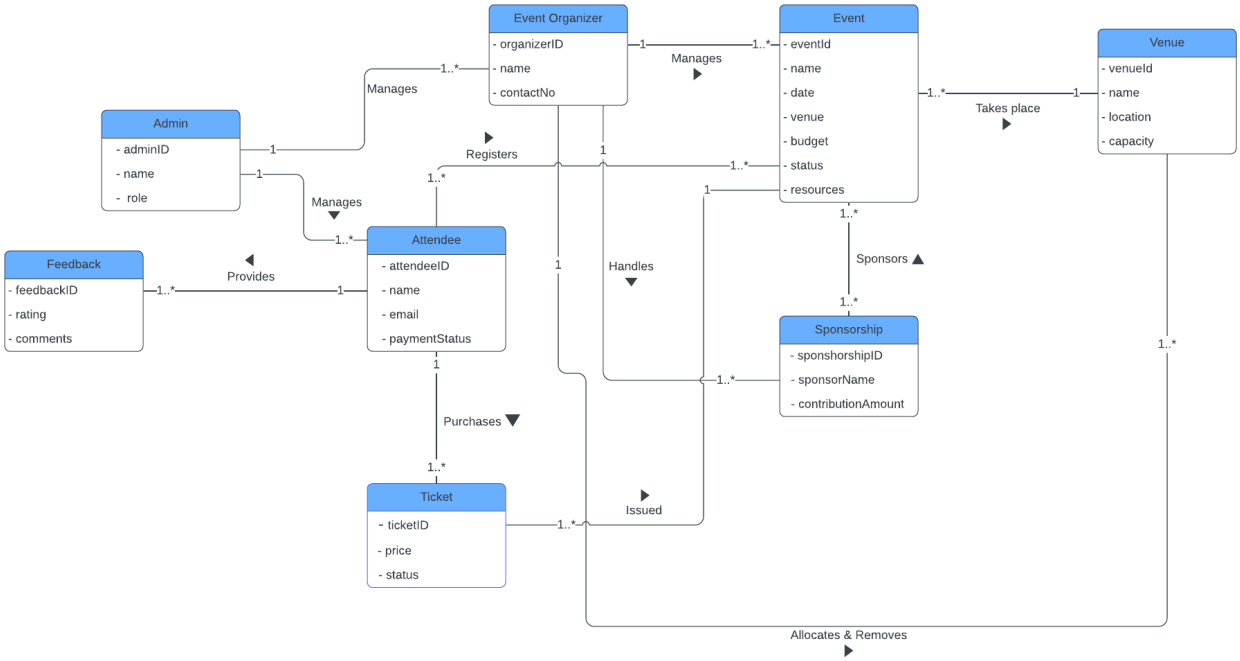
4. Error and Warning Standards

- ***Visual Indicators**:*
  - *Errors: Red icons, alerts, and icons at the top of the page.*
- ***Consistency in Language**:*
  - *All error and warning messages will follow a consistent tone, ensuring users are guided through troubleshooting steps with clarity and without confusion.*

5. User Interface for Different Roles

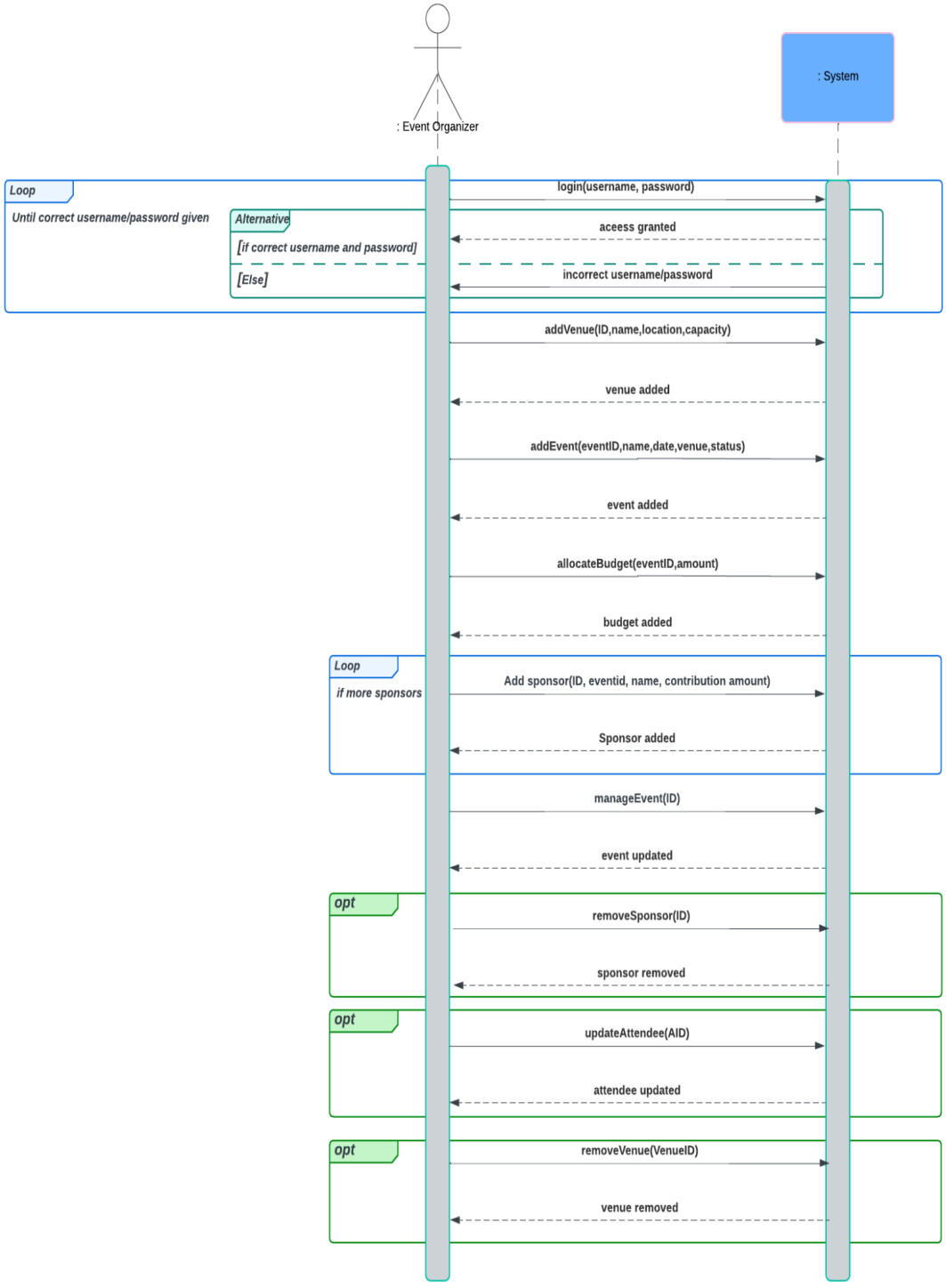
- ***Event Organizer Interface**:*
  - *A dashboard displaying upcoming events, analytical reports, and tools for managing attendee lists and resources.*
  - ***Add an Event** button prominently displayed for easy creation of new events.*
- ***Attendee Interface**:*
  - *A simple, minimalistic event registration page with clearly defined sections for event details, ticket types, and payment.*
  - *An **Event Reminders** section to notify users of upcoming events they are registered for.*
- ***Administrator Interface**:*
  - *An administrative panel displaying all events, users, and system logs.*
  - *A tool for quickly managing disputes, user suspensions, and system health monitoring.*

4. Domain Model

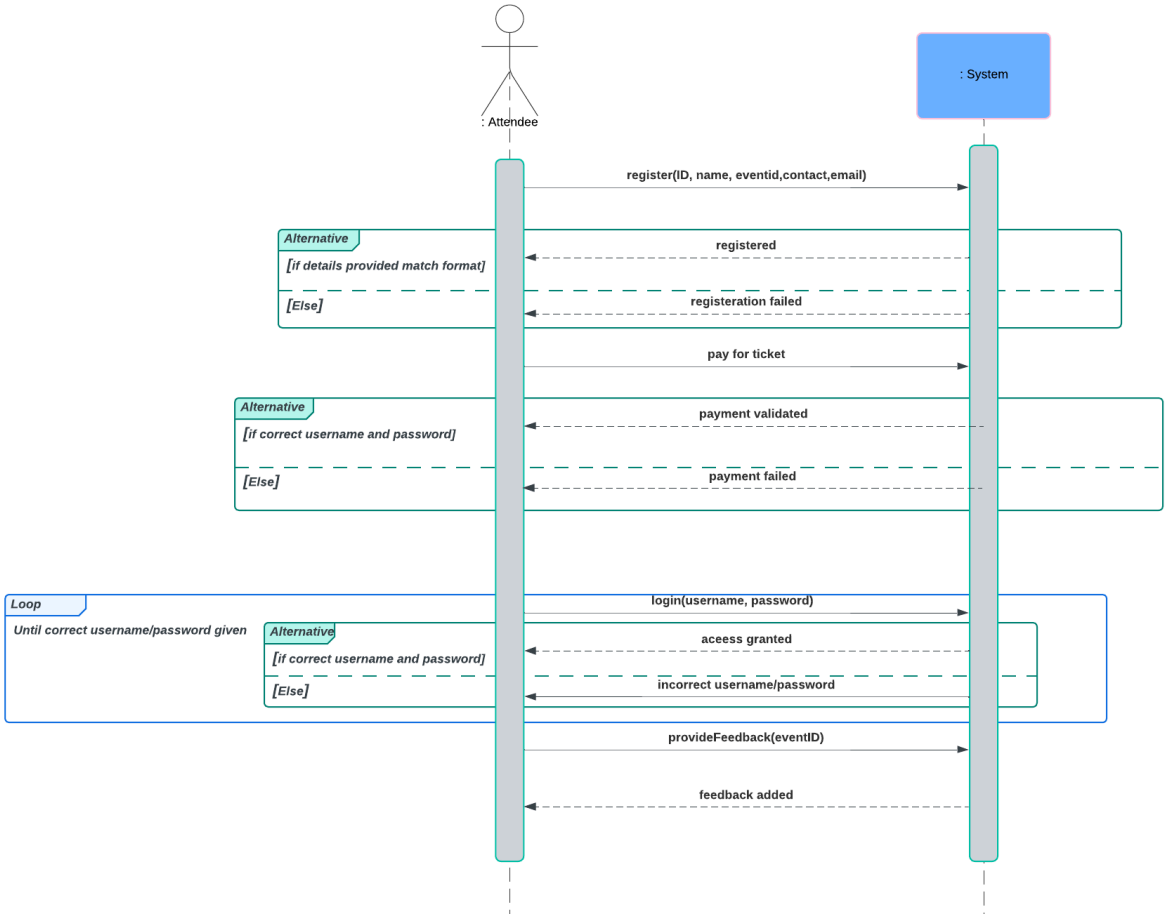


5. System Sequence Diagram

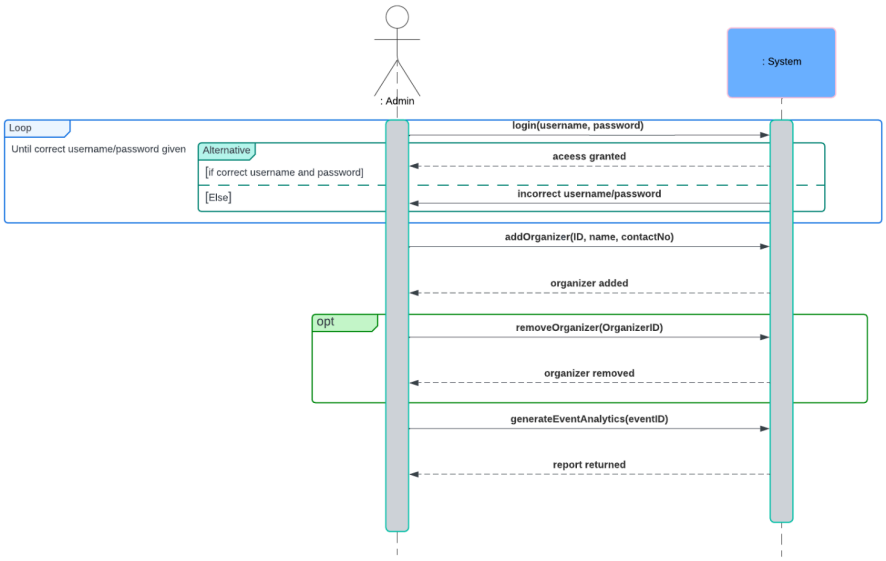
1. Event Organizer



2) Attendee

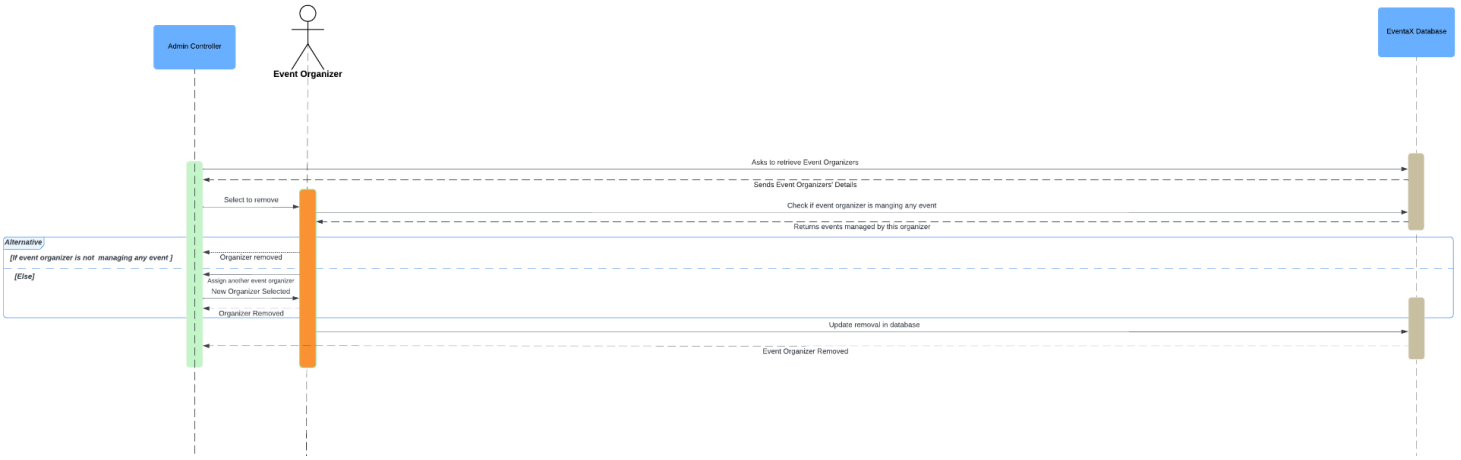


3) Admin



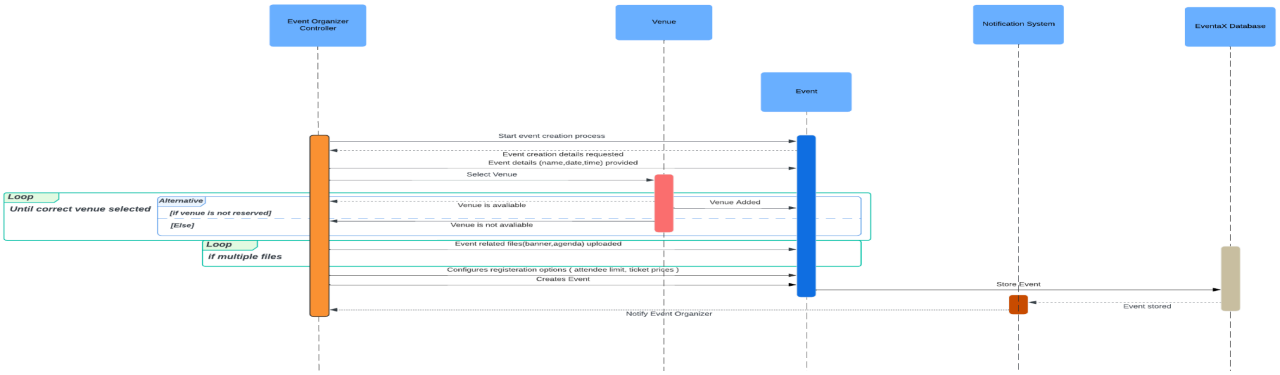
6. Sequence Diagram

1. Remove Event Organizer

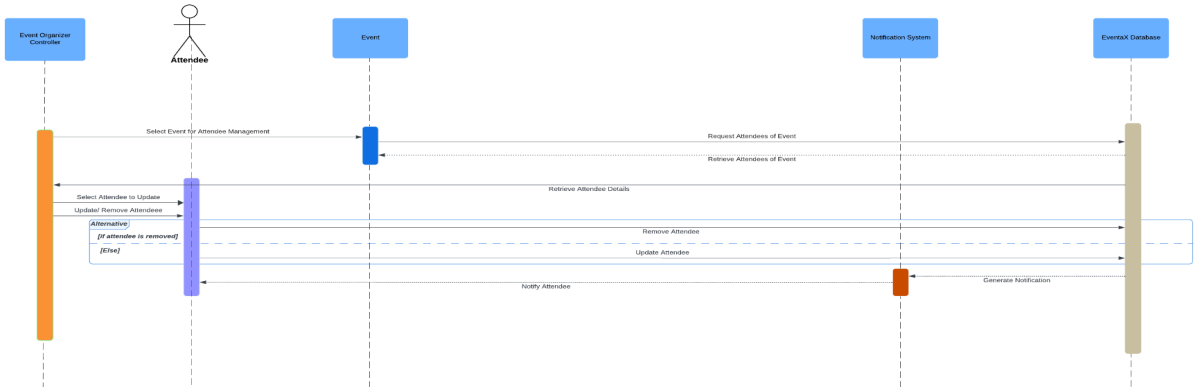


2. Remove Venue

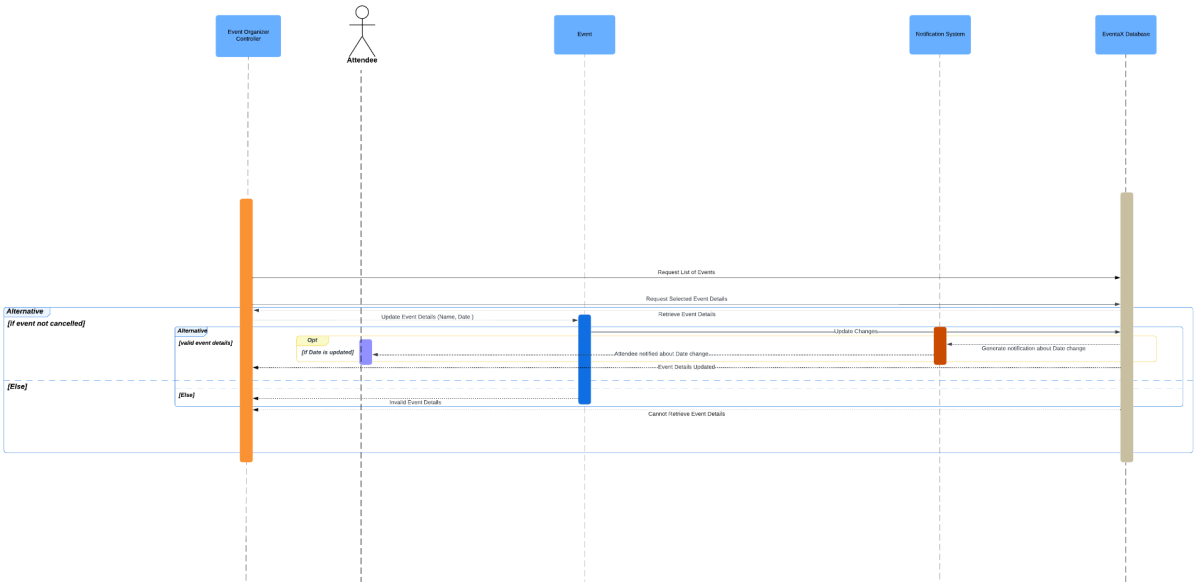
3. Create Event



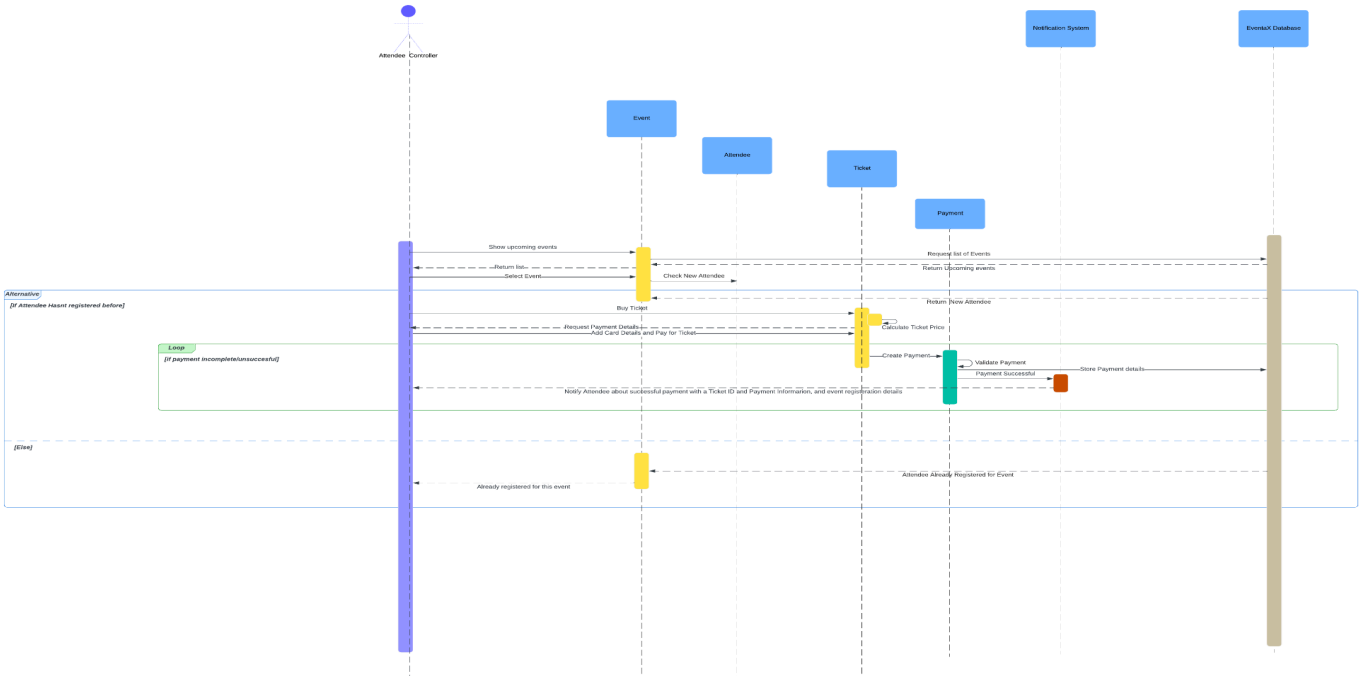
4. Manage Attendee



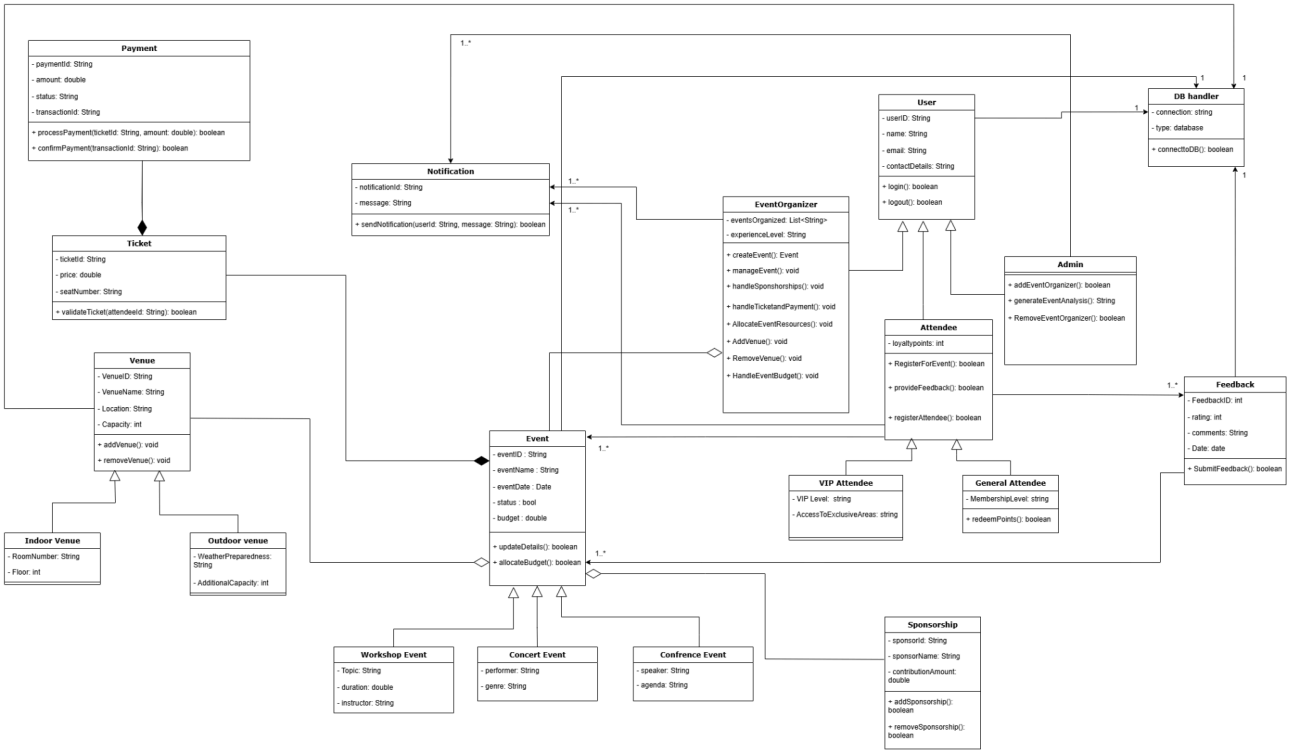
5. Manage Event



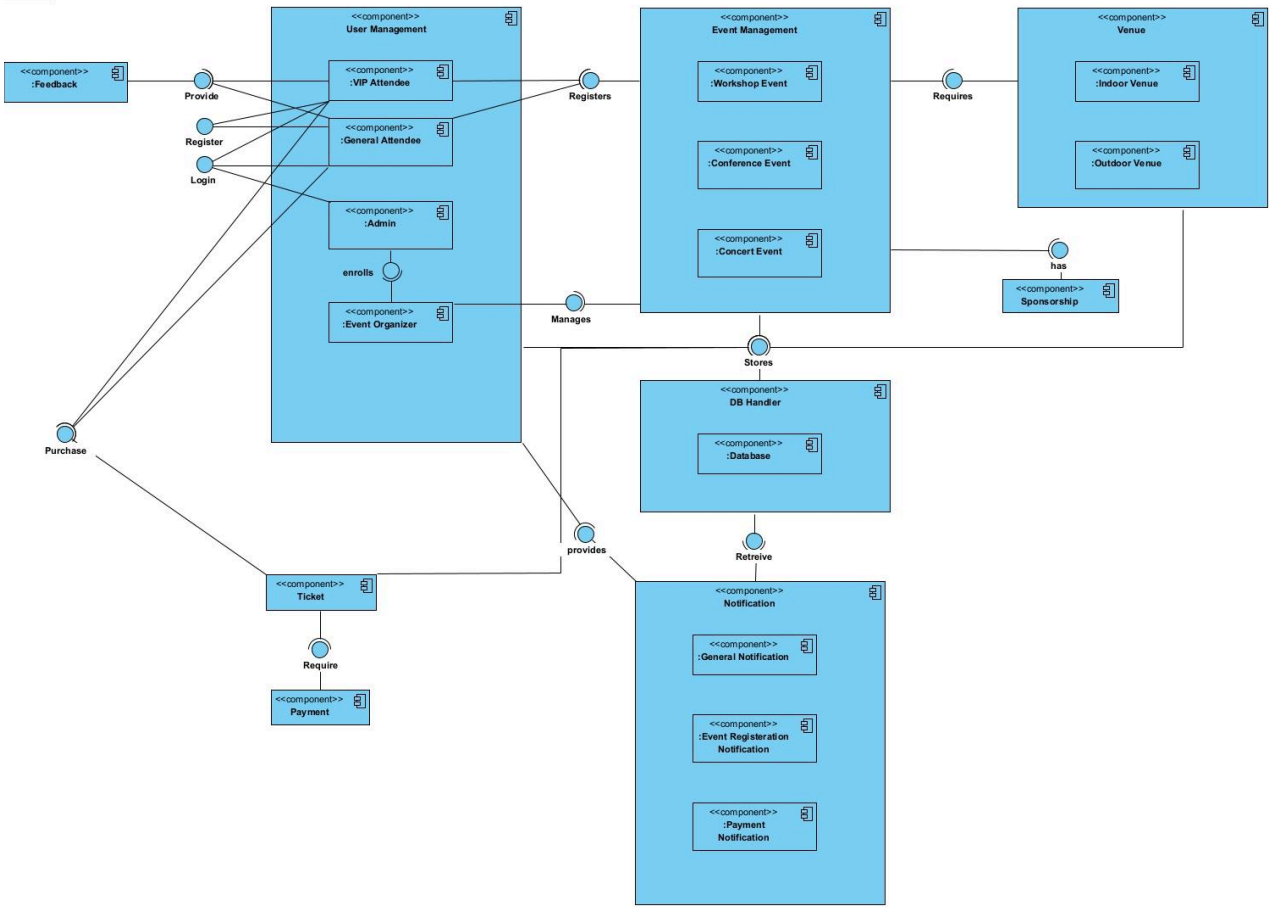
6. Handle Ticket and Payment



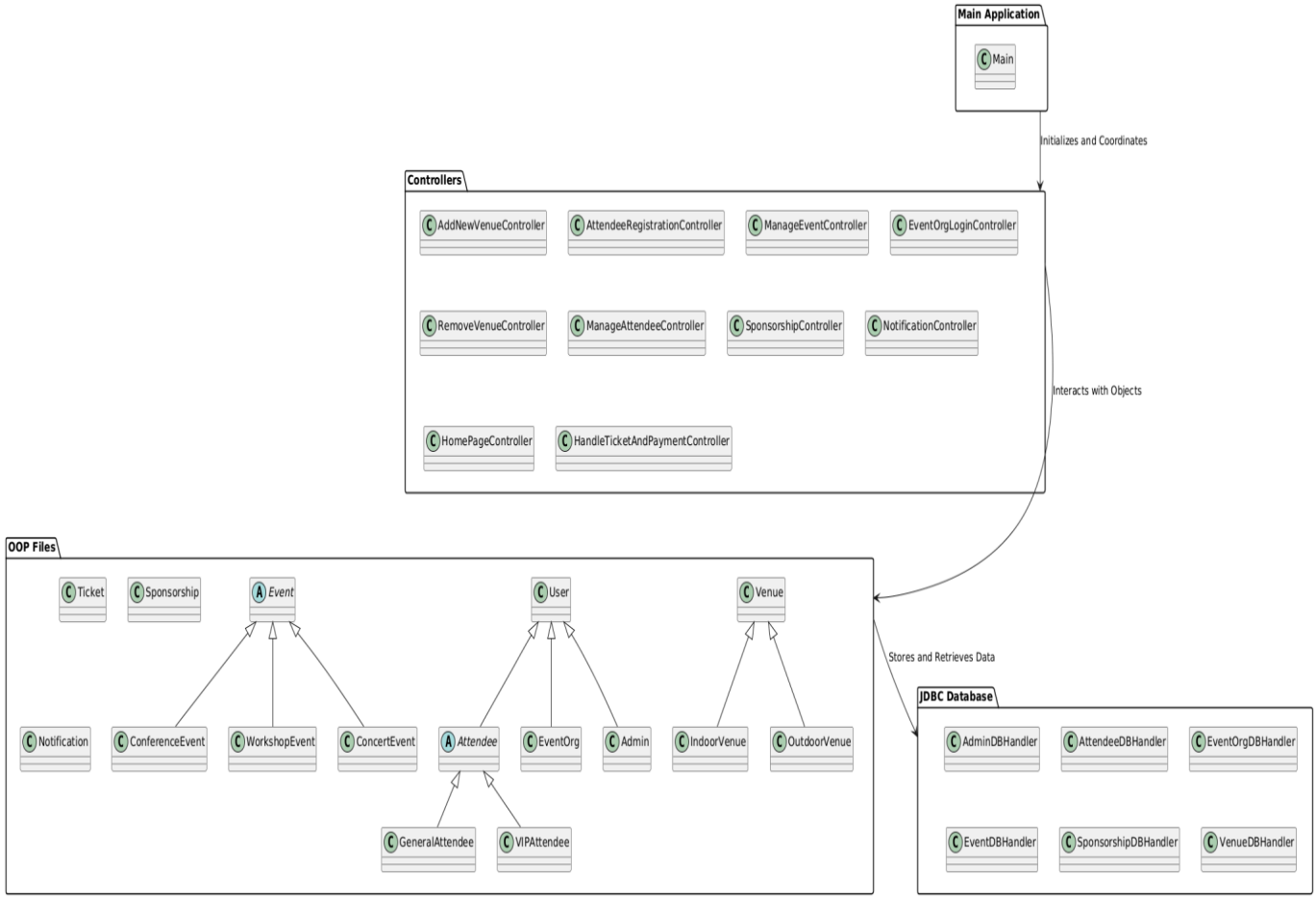
7. Class Diagram



8. Component Diagram



9. Package Diagram



10. Deployment Diagram