
Software Requirements and Design Document

For

NASCON EVENT MANAGEMENT SYSTEM

Prepared by

Waseem Zahid (22i-1355)

Ibtisam Shehzad (22i-1201)

Haider Zia (22i-1196)

Section F

23 Nov 2024

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Product Scope	1
1.3 Title	Error! Bookmark not defined.
1.4 Objectives1	
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	5
2.4 Extended Use Cases	5
2.5 Use Case Diagram.....	12
3. Other Nonfunctional Requirements	13
3.1 Performance Requirements	13
3.2 Safety Requirements	14
3.3 Security Requirements	15
3.4 Software Quality Attributes	Error! Bookmark not defined.
3.5 Business Rules	17
3.6 Operating Environment.....	19
3.7 User Interfaces	Error! Bookmark not defined.
4. Domain Model	25
5. System Sequence Diagram	25
6. Sequence Diagram	32
7. Class Diagram	41
8. Package Diagram	Error! Bookmark not defined.
9. Deployment Diagram.....	Error! Bookmark not defined.

1. Introduction

1.1 Purpose

The purpose of this project is to develop a streamlined and efficient event management system tailored specifically for NASCON events. The system will simplify the process of registering, enrolling, and organizing various events, such as Olympiads, providing a robust platform for both participants and organizers.

1.2 Product Scope

The project will focus on developing an event management system exclusively for NASCON, covering all major events including technical and non-technical competitions such as Sports. The system will support registration for participants, event enrollment, and offer organizers a comprehensive tool for managing event schedules, participants, and results. This system will also address communication between organizers and participants.

Previous systems, if any, have lacked the unified platform needed to manage the diverse range of events held by NASCON. The project will integrate key features like real-time updates, notifications, and user-friendly dashboards.

1.3 Title: SmartSphere

SmartSphere is a comprehensive management system designed to streamline event operations for NASCON. It connects organizers, attendees, and venue managers, ensuring efficient event scheduling, seamless registration, and smooth coordination. The system simplifies event management processes, enhances user experience, and promotes effective communication among all stakeholders.

1.3 Objectives

1. Create a user-friendly registration and enrollment system for participants.
2. Develop a dashboard for organizers to manage event logistics efficiently.
3. Streamline event scheduling, communication, and results reporting.
4. Reduce manual processes and improve event participation management.
5. Ensure scalability to support various NASCON events.

1.4 Problem Statement

<i>The problem of</i>	<i>Inefficient and manual management of NASCON events, including registration, enrollment, and event organization</i>
Affects	Event participants, organizers, and staff
The impact of which is	Confusion, delays, and errors in the event management process, leading to reduced satisfaction for participants and increased workload for

A successful solution would be

organizers
A centralized, digital event management system that automates registration, enrollment, and communication. The system would provide a user-friendly interface for participants to register and enroll, while also offering organizers tools to manage schedules, participant data, and event logistics efficiently. This solution will enhance communication and minimize manual processes.

2. Overall Description

2.1 Product Perspective

The Event Management System for NASCON is a **new, self-contained product** developed specifically to address the inefficiencies in managing NASCON events. This system will replace the existing manual and disjointed processes for registration, enrollment, and event organization with an automated, centralized platform. It will provide a unified system for both participants and organizers to manage all aspects of NASCON events, from registration to results reporting, while also facilitating seamless communication.

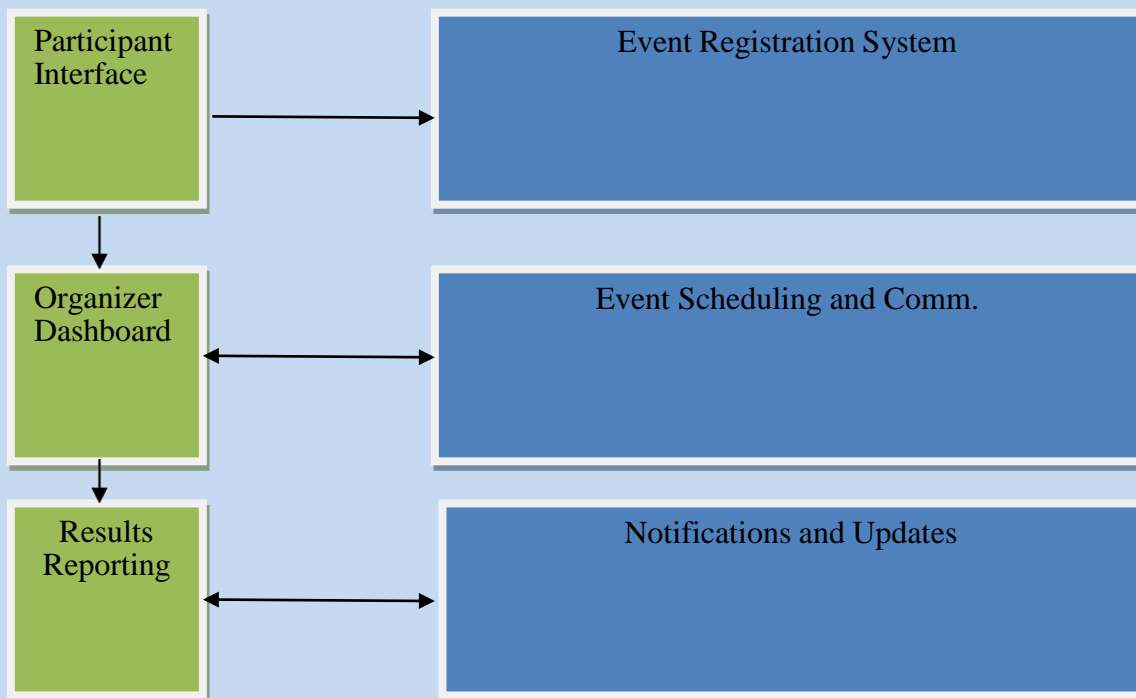
This product is designed to cater to the wide variety of events held by NASCON, such as Olympiads, technical competitions, and sports events. As a **standalone solution**, it will be capable of handling the complexities and requirements unique to NASCON, while offering scalability for different types of events in the future.

While the system will function independently, it may be integrated with existing infrastructure such as email servers or communication tools to send notifications and updates to participants. The system will also allow easy updates to the event schedules, handle participant registrations, and provide organizers with efficient tools for managing logistics, communication, and results reporting.

Major System Components:

- **Participant Registration & Enrollment:** A user-friendly interface for participants to register and enroll in events.
- **Organizer Dashboard:** A comprehensive tool for event organizers to manage schedules, track participant data, and handle event logistics.
- **Event Scheduling and Communication:** A feature to streamline scheduling and ensure effective communication between organizers and participants.

- **Results Reporting:** Automated and efficient tools for generating and sharing event results.
- **Notifications & Updates:** Real-time notifications to keep participants and organizers informed about important events or changes.



2.2 Product Functions

The Event Management System for NASCON must support the following major functions to ensure efficient management of events, registration, and communication between participants and organizers:

1. Participant Registration and Enrollment:

- Allow participants to register for events.
- Enable participants to select and enroll in multiple events (technical, non-technical, sports, etc.).

- Verify participant details and event eligibility.

2. Organizer Dashboard:

- Provide a central platform for event organizers to view and manage event details.
- Allow organizers to create, update, and schedule events.
- Enable organizers to monitor registration progress and manage participant data.
- Generate reports on event status, registrations, and results.

3. Event Scheduling:

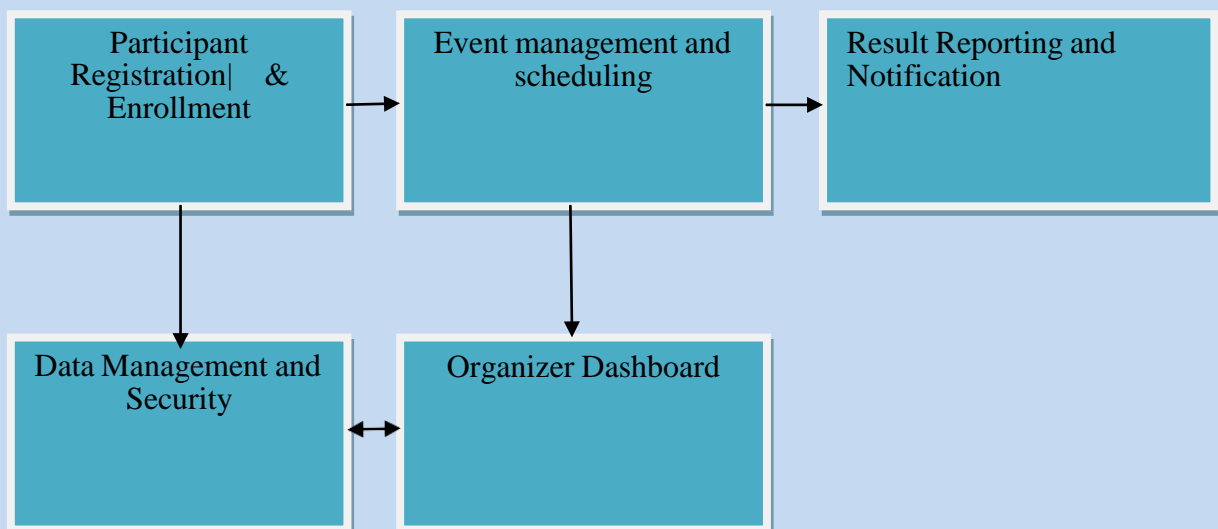
- Allow event organizers to schedule and reschedule events.
- Enable the creation of time slots and venues for each event.
- Display event schedules to participants in real time.

4. Communication and Notifications:

- Send automated notifications and updates to participants and organizers (e.g., event reminders, changes in schedules).
- Facilitate communication between organizers and participants, such as sending messages or announcements.

5. Results Reporting:

- Collect, process, and display results for each event.
- Allow organizers to update event results manually or automatically.
- Display real-time results to participants and other relevant stakeholders.



2.3 List of Use Cases

Create Event
 Modify Event
 View Event details
 Enroll Event
 Send Feedback
 Review feedback report
 Add Announcement
 View notification
 Register User
 Login
 Generate report

2.4 Extended Use Cases

2.4.1 create event:

Use Case Name	Create Event
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actor	Organizer
Stakeholders and Interests	- Organizer: Wants to create and manage an event. - Participants: Benefit from well-organized events. - Admin: Monitors the event creation process for compliance.
Preconditions	- The organizer must be logged into the system.
Postconditions	- A new event is successfully created and stored in the system.
Main Success Scenario	
Participant/Actor Actions	System Responses
1. The organizer selects the "Create Event" option from the dashboard.	2. Displays the event creation form.
3. The organizer fills out the event details (e.g., event name, date, time, venue, description, team size, team limit, registration fee, and account number).	4. Validates the provided information.
5. The organizer submits the event details.	6. Generates a unique

	event ID.
	7. Saves the event details in the system database.
	8. Displays a confirmation message with event details.

Extensions

- **4a:** If any mandatory field is left empty, prompt the organizer to complete the missing information.
- **4b:** If the venue is unavailable for the selected date/time, suggest alternative venues or times.
- **4c:** If the account number is invalid, prompt the organizer to enter a valid account number.
- **7a:** If there is a system error while saving, display an appropriate error message and allow retrying.

2.4.2 Register new user

Use Case Name	Register New User
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actor	User (Admin/Participant/Organizer)
Stakeholders and Interests	- User: Wants to register with the system for appropriate access. - Admin: Oversees user registrations for system integrity. - System: Ensures no duplicate or invalid registrations.
Preconditions	- The system must be operational and connected to the database.
Postconditions	- The user is successfully registered in the system with a unique ID.

Main Success Scenario

Participant/Actor Actions	System Responses
1. The user selects the "SignUp" option on the application interface.	2. Displays the user registration form.
3. The user enters their details, including first name, last name, email, password, and role (Admin/Participant/Organizer).	
4. The user submits the registration form.	5. Validates the provided information. 6. Checks if a user with the same email and role already exists.
	7. Generates a unique user ID if the user does not exist.

8. Saves the new user details to the system database.

Extensions

- **3a:** If any mandatory field is left empty, prompt the user to complete the missing information.
- **3b:** If the provided role is invalid (not Admin/Participant/Organizer), display an error message and reject the registration.
- **6a:** If a user with the same email and role already exists, display a message: "User already registered with this role. Registration skipped."
- **8a:** If there is a database error while saving the user, display an appropriate error message and allow retrying.

2.4.3 Login

Use Case Name	Login
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actor	User (Admin/Participant/Organizer)
Stakeholders and Interests	- User: Wants secure access to the system.
	- System: Ensures valid credentials for controlled access.
Preconditions	- The system must be operational and connected to the database.
	- Users must be registered in the system with valid credentials.
Postconditions	- The user is successfully authenticated and granted access to their role-specific functionalities.
Main Success Scenario	
Participant/Actor Actions	System Responses
1. The user selects the "Login" option on the application interface.	2. Displays the login form.
3. The user enters their email, password, and role (Admin/Participant/Organizer).	4. Validates the input for completeness and role validity.
5. The user submits the login form.	6. Checks the credentials against the registered user database.
	7. If the credentials match, logs the

user into the system.

Extensions

- **3a:** If any field is left empty, prompt the user to complete the missing information.
- **4a:** If the role is invalid or does not match any valid types (Admin/Participant/Organizer), display an error message: "Invalid role selected. Please try again."
- **6a:** If the email or password does not match any user in the specified role, display an error message: "Invalid credentials. Please try again."
- **6b:** If the system encounters a database error during validation, display an appropriate error message and allow retrying.
- **7a:** If the user is locked out due to multiple failed login attempts (if implemented), display a message: "Account locked. Contact support."

2.4.4 Modify Event

Use Case Name	Modify Event
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actor	Organizer
Stakeholders and Interests	- Organizer: Wants to make changes to event details.
	- Participants: Expect accurate and updated event details.
	- System: Ensures that modifications are authorized and logged.
Preconditions	- The event must already exist in the system.
	- The organizer must be logged in and authorized to modify the event.
Postconditions	- The event details are successfully updated and saved in the system.
Main Success Scenario	
Participant/Actor Actions	System Responses
1. The organizer selects the modify event interface.	2. Displays the list of created events.
3. The organizer selects the event to modify from the list of created events.	4. Displays the event details.
5. The organizer updates the desired details (e.g., name, date, venue, etc.).	
6. The organizer submits the modified details.	7. Validates the updated information for completeness and accuracy
	8. Saves the changes to the database and

updates the event.

Extensions

- **2a:** If no events are found for the organizer, display a message: "No events available for modification."
- **5a:** If the organizer tries to modify restricted fields (e.g., event ID), display a message: "This field cannot be modified."
- **7a:** If validation fails (e.g., invalid date or missing details), display an error message: "Invalid input. Please correct the highlighted fields."
- **8a:** If the system fails to save the changes due to a database error, display an error message: "An error occurred while updating the event. Please try again later."
- **8b:** If the organizer cancels the modification, discard changes and return to the event dashboard.

2.4.5 View Event Details

Use Case Name	View Event Details
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actors	Participants, Organizers
Stakeholders and Interests	- Participant: Wants to view accurate and complete event details.
	- Organizer: Wants to verify or review details of their events.
	- System: Ensures quick and accurate access to event information.
Preconditions	- The actor must be logged into the system.
Postconditions	- The actor successfully views the details of the selected event.
Main Success Scenario	
Actor Actions	System Responses
1. The actor navigates to the "View Events" section.	2. Displays a list of events available to the actor.
3. The actor selects an event from the list.	4. Retrieves the event details from the database.
	5. Displays the event details, including name, date, venue, description, team size, registration fee, and organizer details (for participants).

Extensions

- **2a:** If no events are available:
 - Display a message: "No events available at the moment."
- **3a:** If the event is canceled:

- Display a notification: *"This event has been canceled."*
- **4a:** If the system fails to retrieve event details:
 - Display an error message: *"Unable to load event details. Please try again later."*
- **5a:** If additional details are requested:
 - Provide further options, such as "Contact Organizer" for participants or "Modify Event" for organizers.

2.4.6 Add Announcement

Use Case Name	Add Announcement
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actor	Organizer
Stakeholders and Interests	- Organizer: Wants to inform participants about updates or instructions.
	- Participant: Expects clear and timely announcements.
	- System: Ensures announcements are properly stored and accessible.
Preconditions	- The event must exist in the system.
	- The organizer must be associated with the event.
Postconditions	- The announcement is successfully added to the event.
	- The announcement is updated in the database.
Main Success Scenario	
Actor Actions	System Responses
1. The organizer navigates to the "Manage Announcements" section.	2. Displays a list of events organized by the user.
3. The organizer selects an event and enters an announcement message.	4. Validates the selected event and checks if the organizer is authorized.
5. The organizer confirms the addition of the announcement.	6. Adds the announcement to the event object.

Extensions

- **1a:** If no events are available for the organizer:
 - Display a message: *"No events available for announcements."*
- **2a:** If the event does not exist or cannot be found:
 - Display an error message: *"Event not found!"*

- **2b:** If the organizer is not authorized to manage the event:
 - Display an error message: *"You are not authorized to manage announcements for this event."*

- **3a:** If the announcement message is empty:

Prompt the organizer to enter a valid message

2.4.7 Generate Full Event Report

Use Case Name	Generate Full Event Report
Scope	NASCON Event Management System (Desktop Application)
Level	User goal
Primary Actor	Admin
Stakeholders and Interests	- Admin: Requires a comprehensive overview of all events.
	- Organizer: Relies on the report for event performance review.
	- System: Ensures data accuracy and consistency for reporting.
Preconditions <ol style="list-style-type: none"> 1. Events must exist in the system and admin must be logged in. 2. The admin must have access to all related event data (e.g., participants, transactions, announcements, feedback). Postconditions <ol style="list-style-type: none"> 1. A comprehensive report of all events is generated successfully. 2. The report is available for view, print, or export. Main Success Scenario	
Actor Actions	System Responses
1. The admin selects the option to generate a full event report.	2. The system retrieves all events from the database.
3. The system iterates through each event and generates individual reports.	4. Combines all individual reports into a comprehensive report.

Extensions

- **2a:** If no events exist in the system:
 - Display an error message: *"No events found! Cannot generate the report."*
- **2b:** If data for any event is incomplete or corrupted:
 - Display a warning message: *"Some events have incomplete data. Partial report generated."*

Submit Feedback

Use Case Name: **Submit Feedback**

Scope: **NASCON Event Management System**

Level: **User Goal**

Primary Actor: **Participant**

Stakeholders and Interests:

- Participant: **Wants to provide feedback for an event to share their experience.**
- Organizer: **Gains insights into the participants' experiences to improve future events.**

Preconditions:

- **The participant must be logged in.**
- **The participant must have attended the event.**

Postconditions:

- **Feedback is successfully stored in the database and linked to the corresponding event and participant.**

Main Success Scenario

Participant/Actor Actions	System Responses
1. The participant selects the "Submit Feedback" option.	2. Displays a list of events the participant attended.
3. The participant selects an event to submit feedback for.	4. Displays a feedback form for the selected event.
5. The participant enters their feedback and submits the form.	6. Validates the feedback and saves it to the database.

--	--

Extensions:

- **2a:** If the participant has not attended any events:
 - 2a1: The system displays a message indicating no eligible events for feedback.
 - 2a2: Ends the use case.
- **5a:** If the feedback form is incomplete:
 - 5a1: The system prompts the participant to complete all required fields.

View Feedback report

Scope: **NASCON Event Management System**

Level: **User Goal**

Primary Actor: **Organizer**

Stakeholders and Interests:

- Organizer: **Wants to view feedback to evaluate participants' experiences and identify areas for improvement.**
- Admin: **Ensures the feedback system operates smoothly and provides meaningful insights.**

Preconditions:

- **The organizer must be logged into the system.**
- **Feedback must exist for at least one event organized by the organizer.**

Postconditions:

- **Feedback is successfully displayed to the organizer for the selected event.**

Main Success Scenario

Organizer/Actor Actions	System Responses
1. The organizer selects the "View Feedback report" option.	2. Displays a list of events managed by the organizer.
3. The organizer selects an event to view feedback.	4. Retrieves and displays feedback report for the selected event.

Extensions:

- **1a:** If the organizer has not organized any events:
 - 1a1: The system displays a message indicating no events are available for feedback.
 - 1a2: Ends the use case.
- **5a:** If no feedback exists for the selected event:
 - 3a1: The system notifies the organizer that no feedback is available for the event.

- 3a2: Ends the use case.

View Notification

Use Case Name: **View Notification**

Scope: **NASCON Event Management System**

Level: **User Goal**

Primary Actor: **Participant**

Stakeholders and Interests:

- Participant: **Wants to stay informed about updates, changes, or important information related to events they are registered for.**
- Organizer: **Ensures participants are well-informed about the event details.**

Preconditions:

- **The participant must be logged into the system.**
- **The participant must be registered for at least one event.**
- **Announcements must exist for the selected event.**

Postconditions:

- **The participant successfully views all announcements for the selected event.**

Main Success Scenario

Participant/Actor Actions	System Responses
1. The participant selects the "View Announcements" option.	2. Displays a list of events the participant is registered for.
3. The participant selects an event to view announcements.	4. Retrieves all notifications for the selected event.

Extensions:

- **1a:** If the participant is not registered for any events:
 - 1a1: The system displays a message indicating no registered events.
 - 1a2: Ends the use case.
- **3a:** If no announcements exist for the selected event:
 - 3a1: The system notifies the participant that no announcements are available for the event.
 - 3a2: Ends the use case.

Enroll Event

Use Case Name: **Enroll Event**

Scope: **NASCON Event Management System**

Level: **User Goal**

Primary Actor: **Participant**

Stakeholders and Interests:

- Participant: **Wants to register and participate in desired events.**
- Organizer: **Ensures participants enroll correctly to manage event logistics.**

Preconditions:

- **The participant must be logged into the system.**
- **The event must be open for enrollment.**

Postconditions:

- **The participant is successfully enrolled in the selected event, and their details are updated in the system.**

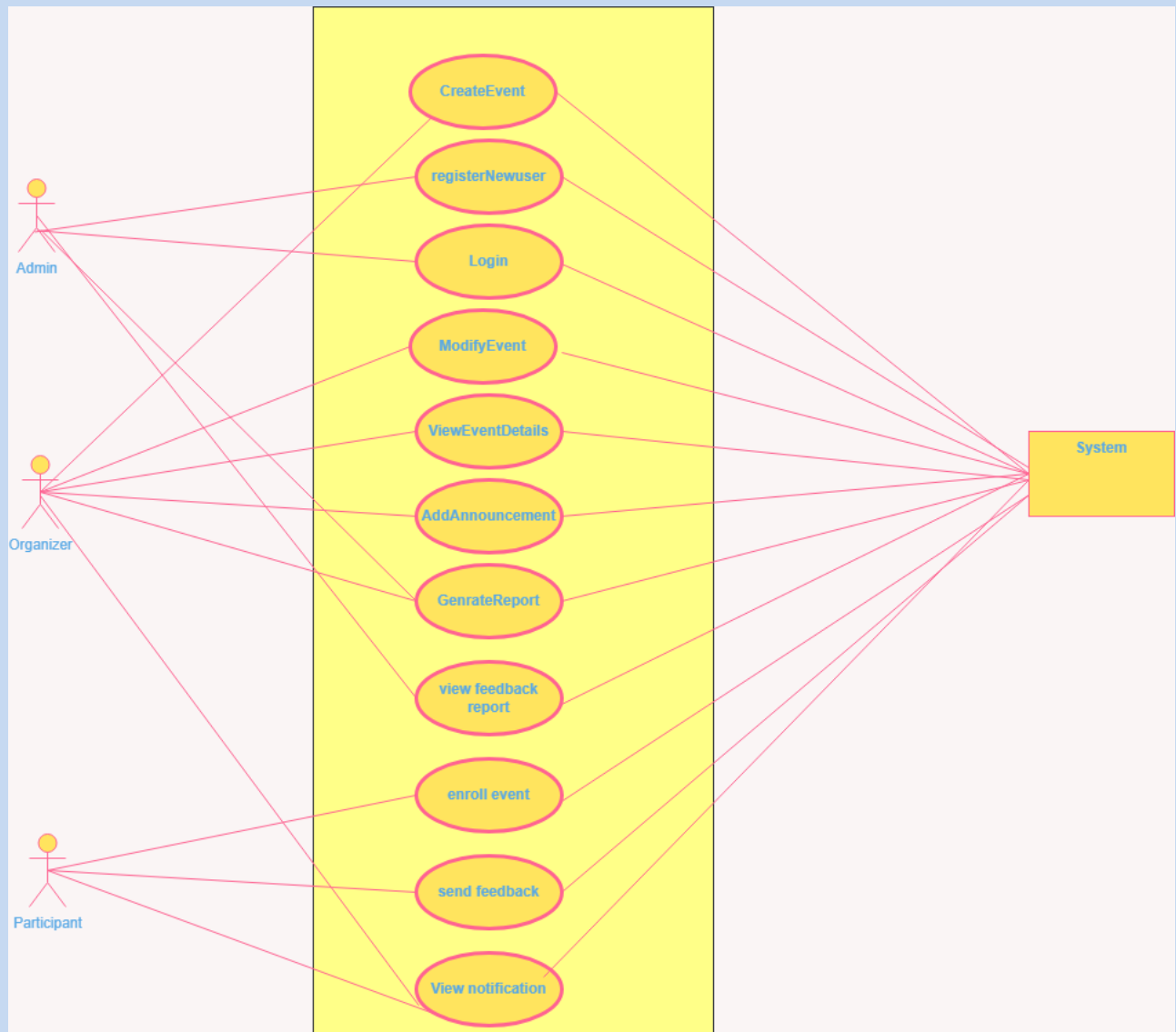
Main Success Scenario

Participant/Actor Actions	System Responses
1. The participant selects the "Enroll Event" option.	2. Displays a list of events available for enrollment.
3. The participant selects an event to enroll in.	4. Displays an enrollment form .
4. The participant fills the detail.	5. System ensures that fields are filled.
5. The participant confirms enrollment and payment details.	

Extensions:

- **1a:** If no events are open for enrollment:
 - 1a1: The system displays a message indicating no available events.
 - 1a2: Ends the use case.
- **3a:** If the participant is not eligible for the event (e.g., team size limit reached):
 - 3a1: The system notifies the participant of the ineligibility.
 - 3a2: Ends the use case.
- **5a:** If the payment process fails:
 - 5a1: The system notifies the participant of the payment failure.
 - 5a2: The participant can retry or cancel the enrollment process.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

1. Moderate Response Time

- Our system should respond to user actions like login or registration within **5-10 seconds**.
- Pages and dashboards should load within **10 seconds**.

2. Support for Multiple Users

- Our system should handle **a moderate number of users** simultaneously without significant delays.
- It should scale to support **20% more users** for future events.

3. Periodic Updates

- Notifications and event updates in our system should occur within **10-15 seconds**.

4. Moderate Data Processing

- Registrations, event updates, and results should process within **10 seconds** in our system.

5. Reasonable Scheduling

- Our system should resolve scheduling conflicts or venue issues within **30 seconds or less**.

6. Secure Transactions

- User logins and payments in our system should complete within **10 seconds**.

7. Report Generation

- Our system should generate reports like results or stats within **30 seconds**.

8. Moderate Uptime

- Our system should be available **98% of the time**.

- Maintenance downtime should not exceed **2 hours per month**, and unexpected downtime should remain under **1 hour monthly**.

9. Database Performance

- Our system should manage **a reasonable number of records** per event, with data queries taking less than **5 seconds**.

This revision reflects a more balanced approach, ensuring the system meets basic functional needs without targeting high-end performance.

3.2 Safety Requirements

1. Basic Data Privacy and Protection

- Our system should use basic encryption methods to protect user data (e.g., personal details, payment information) and minimize unauthorized access risks.
- Ensure compliance with **local data protection laws** where applicable, though advanced regulations like GDPR may be optional.

2. User Authentication

- Users should access participant or organizer dashboards through secure login credentials.
- Multi-factor authentication can be optional but encouraged for high-privilege accounts.

3. Data Loss Prevention

- Backups of event data, participant details, and results should occur **weekly** to reduce data loss risks during unexpected failures.
- Basic recovery mechanisms should restore data within **reasonable timeframes** during crashes.

4. Error and Conflict Management

- Inputs should be validated to reduce scheduling conflicts, duplicate registrations, or incorrect data entries.
- Errors should be flagged for manual correction by admins or users.

5. Basic Access Control

- User roles (participants, organizers, admins) should have basic restricted permissions to prevent accidental changes to key data.
- Full audits and advanced restrictions may not be necessary.

6. Payment Security

- Internal security mechanisms for payment processing can be basic.

7. System and Physical Security

- Servers should have basic protections against physical damage and unauthorized access.
- Basic firewalls should be enabled to mitigate potential cyberattacks, but intrusion detection systems may not be required.

8. Load and Crash Prevention

- The system should monitor performance and apply simple measures, like **basic request limits**, to avoid overloads during peak times.

This version focuses on providing **essential security features** for an average system while avoiding unnecessary complexities or high compliance standards.

3.3 Security Requirements

3.3.1 User Authentication

- We ensure that all users (participants, organizers, admins) log in with unique usernames and passwords.
- For added security, we will implement **optional multi-factor authentication** for admin accounts.

3.3.2 Data Privacy

- We will use things to secure the transmission of sensitive data, such as user credentials and payment details.
- Sensitive information stored in our database, like passwords or payment information, will be encrypted using similar standards.
- To comply with privacy regulations such as **GDPR**, we will provide users the option to view and delete their personal data upon request.

3.3.3 Access Control

- We will implement **role-based access control** to manage user permissions:
 - Admins will have full system access.
 - Participants and organizers will have limited access based on their roles.
- We will enforce access control checks at both the application and database levels to prevent unauthorized actions.

3.3.4 Data Integrity

- We will use database-level constraints like foreign keys, unique keys, and NOT NULL constraints to ensure data integrity.
- Periodic data consistency checks will be performed using scripts or stored procedures in MySQL.

3.3.5 Secure Payments

- We will integrate secure payment gateways that comply with **PCI-DSS standards**.
- Basic fraud detection mechanisms, such as payment validation and logging suspicious transactions, will be implemented.

3.3.6 System Monitoring

- We will set up real-time logging in our JavaFX application to track significant actions, such as logins, updates, and deletions.
- Our MySQL database activities, such as failed queries or slow execution times, will be monitored using tools like **MySQL Workbench** or custom scripts.

3.3.7 Data Backup

- We will schedule **daily automated backups** of our MySQL database using tools like **mysqldump**.
- Backups will be securely stored on a separate server or in cloud storage, with restricted access to authorized personnel.

3.3.8 Session Management

- We will implement session timeout functionality in our JavaFX application to log out inactive users after a set period, such as 15 minutes.
- User sessions will be explicitly invalidated upon logout to ensure they cannot be reused.

3.3.9 Audit Trails

- We will maintain detailed **audit logs** in our MySQL database to track important user activities, such as logins, updates, and payment transactions.
- Logs will include timestamps, user IDs, and descriptions to help us investigate any security incidents effectively.

By adhering to these security measures, we will ensure the safety of our users' data, provide a secure experience, and protect our system from unauthorized access or potential threats.

3.4 Software Quality Attributes

3.4.1 Adaptability

- The system is designed to easily add new event types or categories by utilizing reusable Java classes like `EventType` or `EventCategory`.
- Event schedules or formats are managed dynamically through the MySQL database, ensuring minimal code changes and downtime during updates.

3.4.2 Availability

- The system maintains **99.9% uptime**, particularly during peak event periods.
- Basic failover mechanisms ensure smooth operation by switching to a backup database or application server during failures.

3.4.3 Correctness

- Accuracy in event registrations, scheduling, and payment processing is ensured through data validation at the JavaFX form level and enforced constraints in MySQL, such as `NOT NULL`, `UNIQUE`, and `CHECK`.

3.4.4 Flexibility

- Modifications to event types, participant categories, or venue details are handled dynamically through the **JavaFX admin panel**, with changes stored in the database.
- Core functionality remains unaffected by updates or adjustments to event requirements.

3.4.5 Interoperability

- The system integrates seamlessly with email and SMS notification tools using Java APIs.
- Data exchange is enabled by exporting MySQL records to CSV files, ensuring compatibility with external platforms.

3.4.6 Maintainability

- Clean coding practices are followed, including meaningful variable names and modularized code structures, simplifying updates.
- The system architecture separates **UI (JavaFX)**, **Business Logic (Java)**, and **Data (MySQL)** layers for easier debugging and maintenance.

3.4.7 Portability

- Java ensures compatibility across operating systems, including **Windows** and **Linux**.
- The application is portable and deployable on various devices without additional modifications.

3.4.8 Reliability

- The system operates consistently under normal conditions, handling multiple event registrations and updates without crashes.
- Automatic recovery mechanisms, such as retries or error messages, are implemented to maintain stability during network instability.

3.4.9 Reusability

- Core components, such as the registration module and notification system, are designed as independent and reusable modules.
- Reusability is achieved through modular and loosely coupled design, making them adaptable for different event scenarios.

3.4.10 Usability

- The system provides a user-friendly interface with clear navigation and intuitive designs in JavaFX.
- Users perform tasks like registration, event management, and report generation easily, reducing the learning curve.

This version aligns with the use of **JavaFX, Java, and MySQL** while focusing on practicality and system quality attributes.

3.5 Business Rules

1. Participant Registration

- Only participants who are eligible and have provided the necessary information (e.g., name, age, event preferences) can register for an event.
- A participant can register for multiple events, but the system must ensure there are no conflicts in event schedules.

2. Event Enrollment

- Event organizers must define the maximum capacity for each event. Once the capacity is reached, no more participants can enroll unless there is an increase in the event's capacity.
- Participants cannot enroll in an event after the registration deadline has passed.

3. Role-Based Access Control (RBAC)

- Only authorized organizers and administrators can manage event details, schedules, and participant data.
- Organizers can only access and modify the details for events they are responsible for.
- Administrators have full access to all event data, participant information, and can make system-wide changes, such as modifying event settings or adding new events.

4. Event Scheduling

- Event schedules must avoid conflicts with other events, especially for participants enrolled in multiple events. The system should automatically flag scheduling conflicts when an event overlaps with a participant's other registered events.
- Event organizers can set specific time slots for events, but the system must ensure that the event duration aligns with the available venue's capacity.

5. Notification System

- Participants should receive an email/SMS notification confirming their registration upon successful enrollment.
- Organizers must be able to send mass notifications about event updates, changes, or cancellations to all registered participants.

6. Payment and Fees

- Payment for events must be completed before registration is finalized. If payment is not confirmed, the registration is considered incomplete.
- The system should allow for different fee structures for various types of events (e.g., technical, sports).

7. Results Reporting

- After an event, organizers are responsible for entering the results into the system. These results must be confirmed by the organizer before being published on the platform.
- Results should be available to participants within 24 hours of event completion.

8. Data Privacy and Security

- Personal data collected from participants, such as name, contact information, and event participation, must be securely stored and accessible only by authorized users.
- Data will not be shared with third parties without the explicit consent of the participant, except for necessary communications regarding the event.

9. User Authentication and Access

- All users (organizers, participants, administrators) must create accounts to access the system.
- Users must authenticate with a secure username and password before accessing event-related features.

10. System Availability

- The system should be available 24/7, except during scheduled maintenance windows. Maintenance periods must be communicated to users in advance.

These business rules help establish the operational framework for the event management system, ensuring smooth and efficient event organization while maintaining security and proper access control.

3.6 Operating Environment

The event management system operates in a multi-platform environment, designed for optimal performance and compatibility across various hardware and software configurations.

3.6.1.1 Hardware Platform

- **Servers:**
 - Cloud-based infrastructure or on-premises servers with adequate resources to support up to 100 concurrent users.
 - Minimum specifications: 8GB RAM, quad-core processor, and 500GB storage capacity.
- **Workstations:**
 - Desktop or laptop computers for event organizers, administrators, and participants.
 - Minimum requirements: 4GB RAM, 1 GHz processor speed, and 200MB available disk space for system access.

3.6.1.2 Operating System

- **Client Side (Participants and Organizers):**
 - Compatible with:
 - Windows 10 or newer.
 - Linux distributions (Ubuntu 18.04 or newer).
- **Server Side:**
 - Primary: Ubuntu 20.04 or later, preferred for deployment ease and compatibility with Java and MySQL.
 - Alternative: Windows Server 2019 or later.

3.6.1.3 Web Browser

- Designed for compatibility with modern web browsers:
 - **Google Chrome** (latest version).
 - **Microsoft Edge** (latest version).
- Responsive design ensures usability on both desktop and mobile devices.

3.6.1.4 Software Components

- **Frontend:** Developed using **JavaFX**, providing a robust and interactive user interface.
- **Backend:** Powered by **Java**, ensuring a seamless connection between the application logic and database.
- **Database Management System (DBMS):** **MySQL** is used to manage event and participant data securely and efficiently.
- **Authentication Services:** Integrated with secure authentication mechanisms to protect user credentials and ensure controlled access.

This configuration ensures adaptability, reliability, and security in various environments while supporting a diverse user base.

3.7 User Interfaces

The user interface (UI) of the event management system will consist of several logical components to ensure ease of use and accessibility for different user types (participants, organizers, administrators). Below are the key characteristics for each interface:

1. Participant Interface:

- **Login/Registration Screen:**
 - Simple fields for username, email, password.
 - Option for password recovery.
- **Event Dashboard:**
 - List of events with details such as event name, date, and status (open for registration, full, or closed).

- A search bar to filter events based on type (technical, sports, etc.).
- Option to register and enroll in events with clear call-to-action buttons.
- **Notifications:**
 - Display notifications for successful registration, event updates, or cancellations.
 - Clear and concise messaging with action buttons (e.g., "View Event", "Cancel Registration").

2. Organizer Interface:

- **Login/Authentication Screen:**
 - Standard username and password input fields.
 - Two-factor authentication (optional).
- **Dashboard:**
 - Overview of all events: search and filter options for events, registrations, schedules.
 - Event creation and management features (add, modify, or cancel events).
 - Ability to view participant lists, update event details, and confirm results.
- **Schedule Management:**
 - Calendar-based scheduling interface for event timings.
 - Automatic conflict detection to avoid overlapping events.
- **Results Reporting:**
 - A dedicated form to enter results for each event, with validation checks for completeness.

3. Administrator Interface:

- **Admin Dashboard:**
 - Comprehensive control panel to manage the system, including user roles, permissions, and access to all events.
 - Ability to monitor system health (server status, user count, etc.).
- **User Management:**

- Option to create, modify, or delete user accounts (organizers, participants).
- Assign roles and permissions to users.

System Settings:

- Ability to configure email/SMS notifications, payment integration, and security settings.
- System maintenance options, including backup and data restore.

4.

General UI Standards:

- **Consistency:** All screens will follow a consistent color scheme, font styles, and button placement to ensure ease of navigation.
- **Navigation:** Clear navigation menus for quick access to all major sections (Dashboard, Events, Profile, etc.).
- **Error Handling:** Standard error messages will be displayed for common issues (e.g., incorrect input, network errors). The system will guide users to resolve issues with friendly error messages.
- **Help/Support:** An easily accessible help section with FAQ and contact details for support.
- **Mobile-Responsive Design:** The interface will be fully responsive and optimized for mobile users, ensuring that participants and organizers can access the platform on any device.

5.

Sample Screen Layout:

- **Header:** Includes logo, navigation menu, and user profile icon (for logged-in users).
- **Sidebar:** Quick links to major sections such as events, notifications, and dashboard.
- **Main Area:** Displays detailed content relevant to the current section (event list, registration forms, etc.).
- **Footer:** Contains links to help, privacy policy, terms of use, and contact support.

```

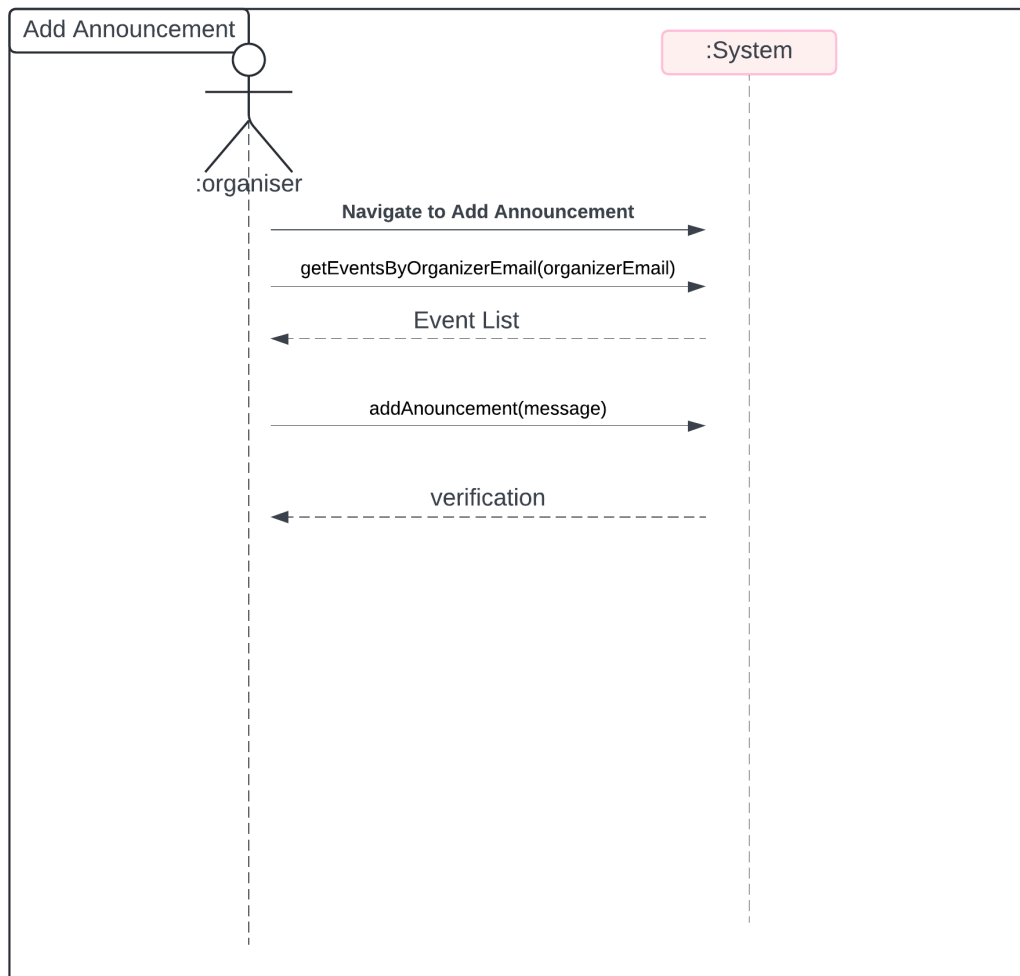
classDiagram
    class Admin {
        +id
        +firstName
        +lastName
        +email
        +password
    }
    class User {
        +userId
        +firstName
        +lastName
        +email
        +password
    }
    class TimeSlot {
        +dateandtime
    }
    class Payment {
    }
    class Team {
        +teammember
        +participant list membes
        +verified
    }
    class Participants {
        +registeredEvents
    }
    class GenerateReport {
        +reportId
        +title
        +dateGenerated
    }
    class Event {
        +eventId
        +name
        +organizerEmail
        +description
        +venue
        +dateTime
        +feedbackList
        +verifiedTeams
        +teamSize
        +teamlimit
        +registrationFee
        +accountNumber
        +announcements
    }
    class Feedback {
        +eventID
        +Comments
        +ratings
    }
    class Organizer {
        +managedEvents
    }
    class Announcement {
    }
    class Venue {
        +listofvenues
    }

    Admin "1" -- "*" User
    Admin "1" -- "1" TimeSlot
    Admin "1" -- "1" Payment
    Admin "1" -- "*" Participants
    Admin "1" -- "*" GenerateReport
    Admin "1" -- "0" Event
    Admin "1" -- "*" Feedback
    Admin "1" -- "1" Organizer
    Admin "1" -- "*" Announcement
    Admin "1" -- "1" Venue

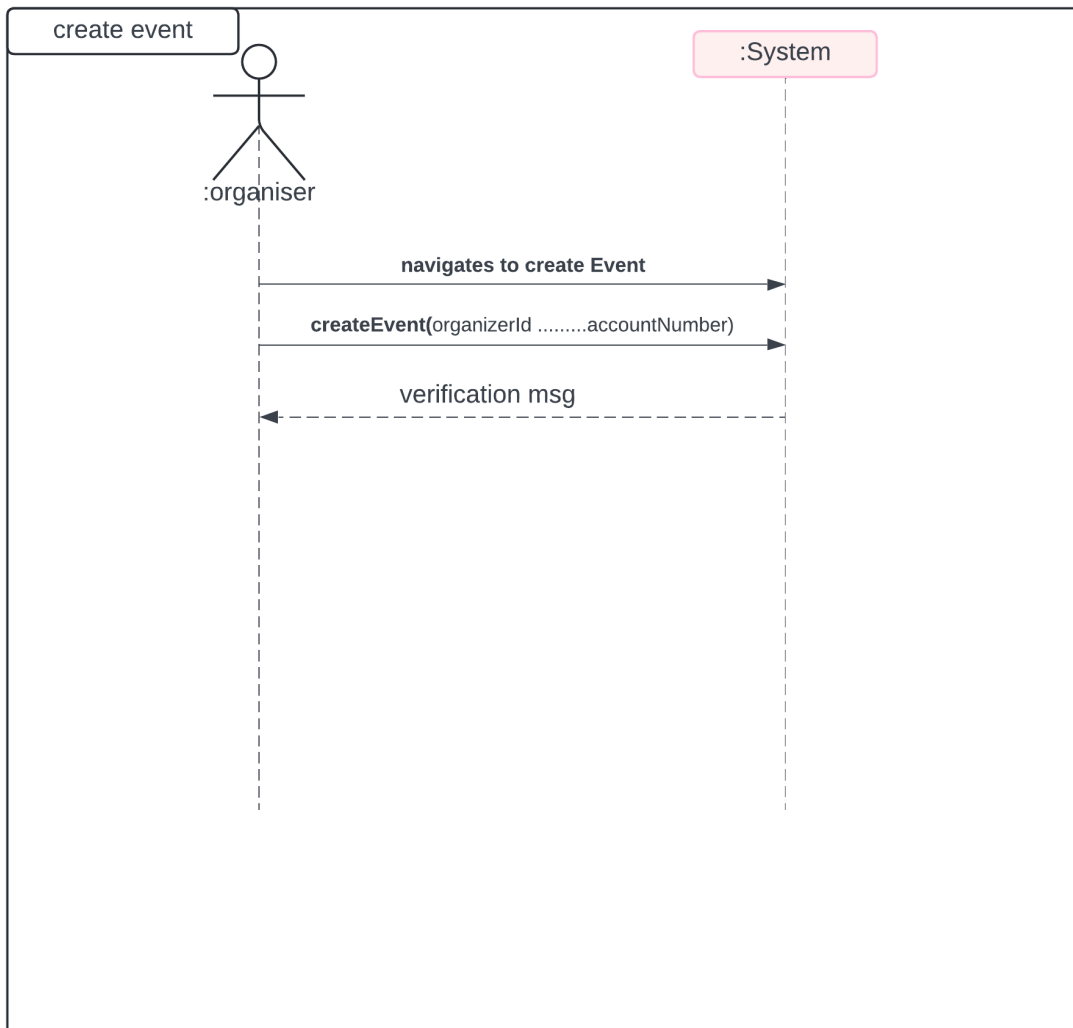
    User "*" -- "*" Event
    TimeSlot "1" -- "*" Event
    Payment "1" -- "*" Event
    Team "*" -- "*" Event
    Participants "*" -- "*" Event
    GenerateReport "*" -- "*" Event
    Event "0" -- "*" Event
    Feedback "*" -- "*" Event
    Organizer "1" -- "*" Event
    Announcement "*" -- "*" Event
    Venue "1" -- "*" Event
  
```

5. System Sequence Diagram

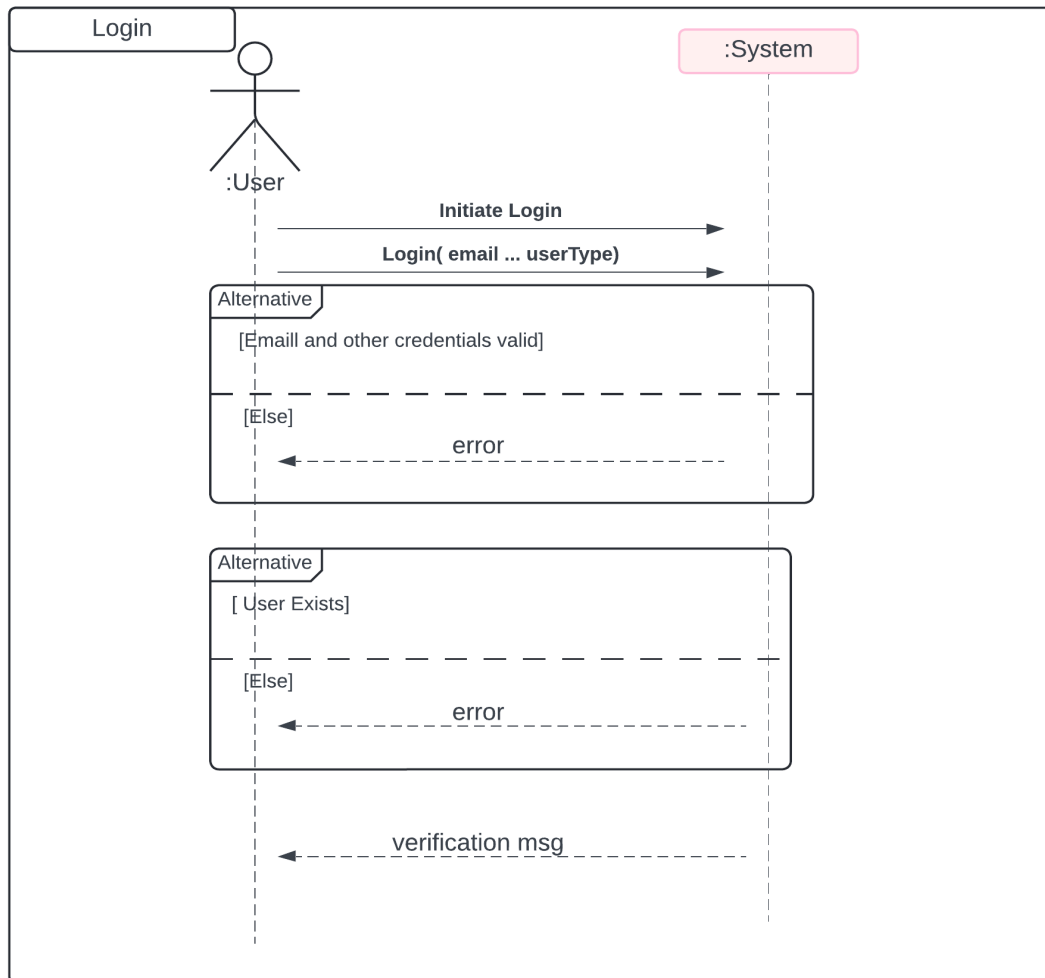
1. Add Announcement



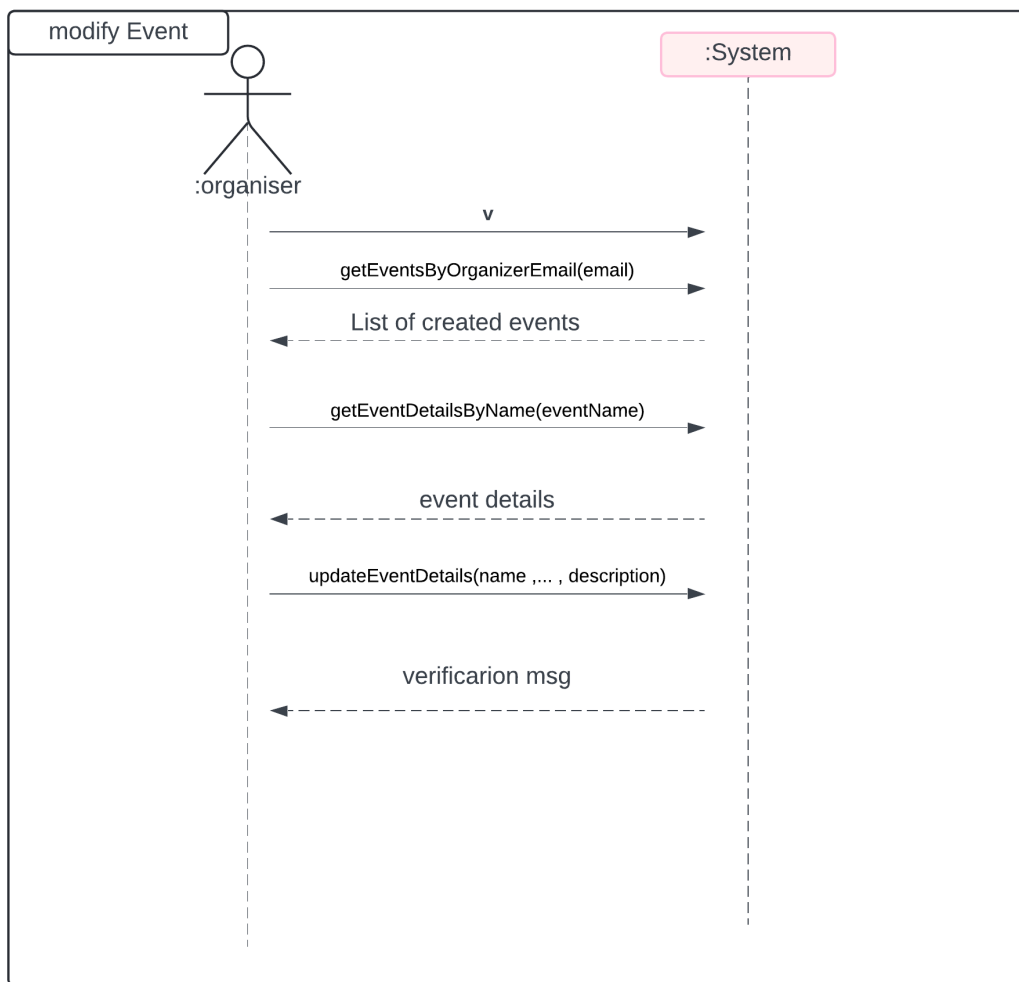
4.1 Create Event



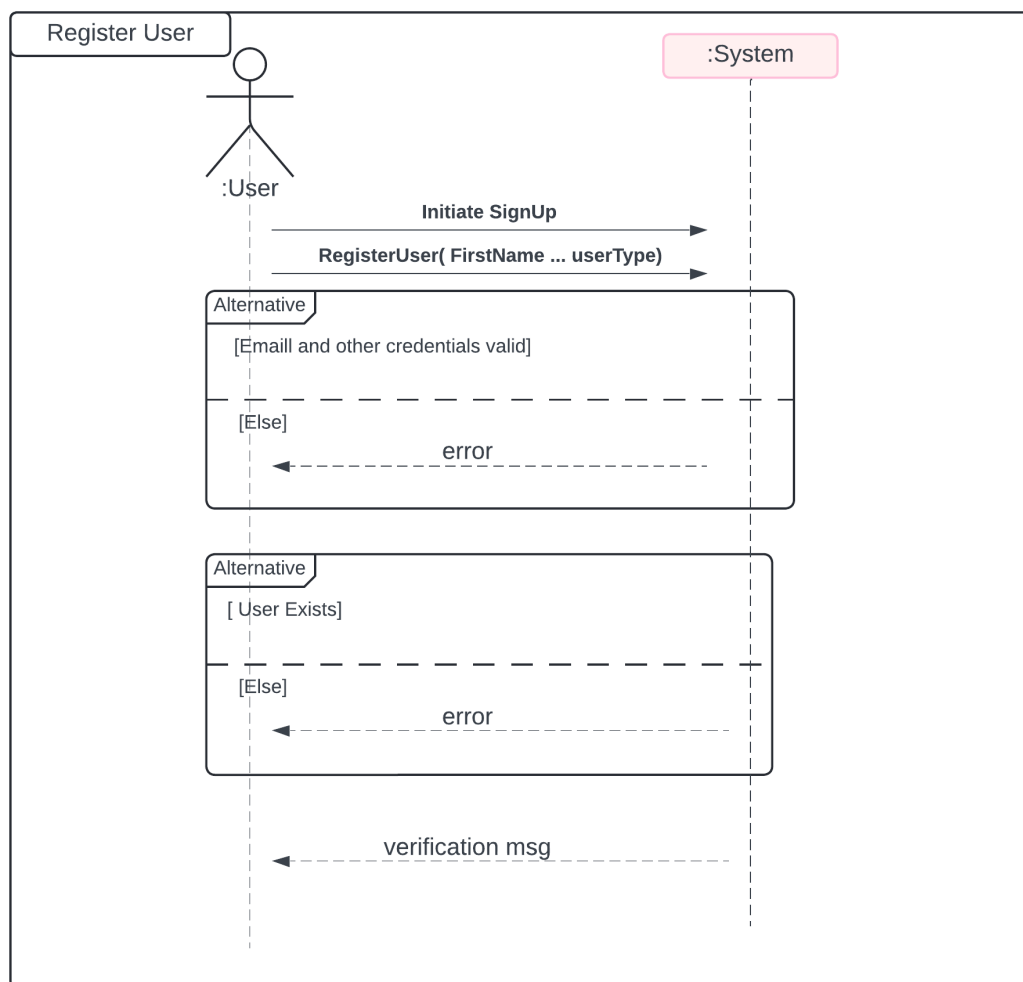
4.2 Login



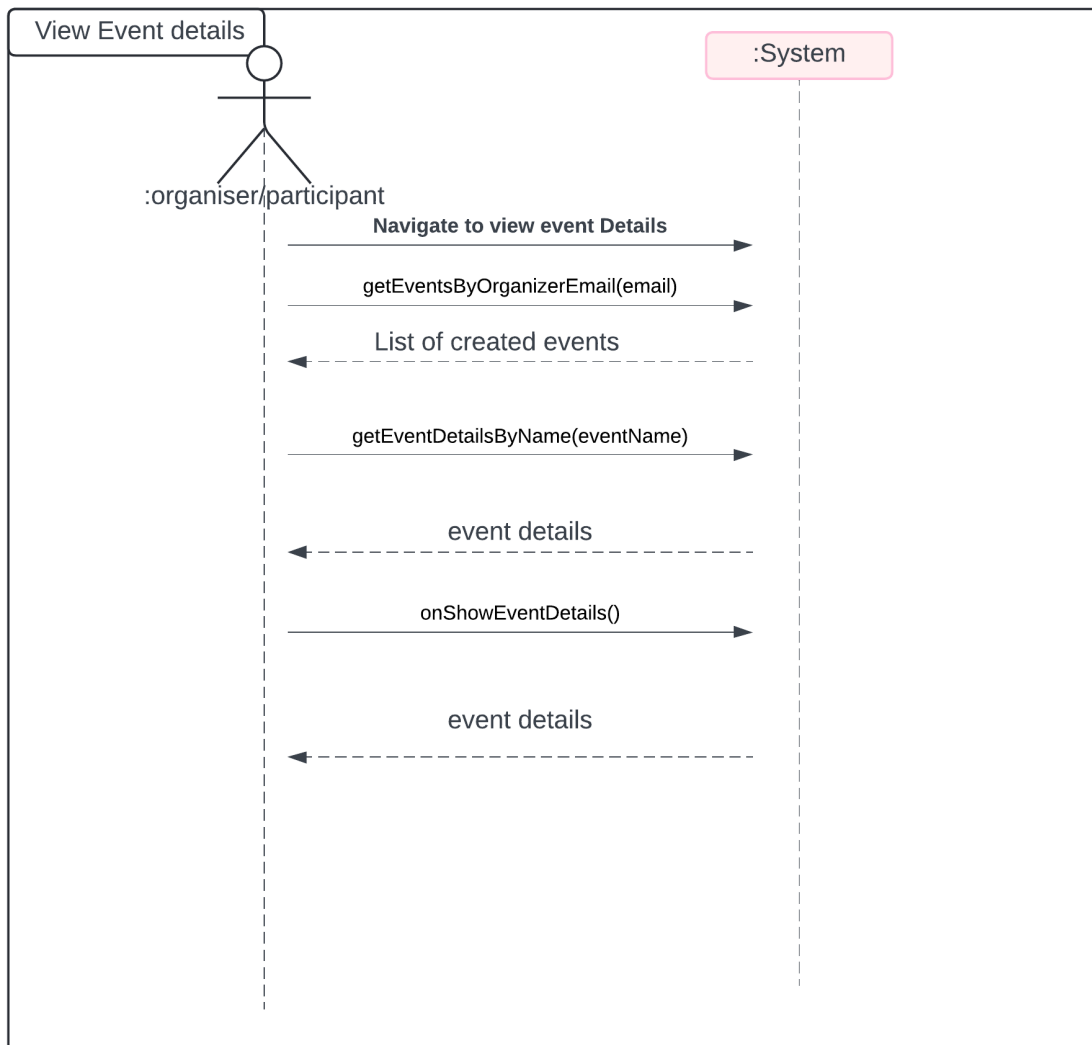
4.3 Modify Event



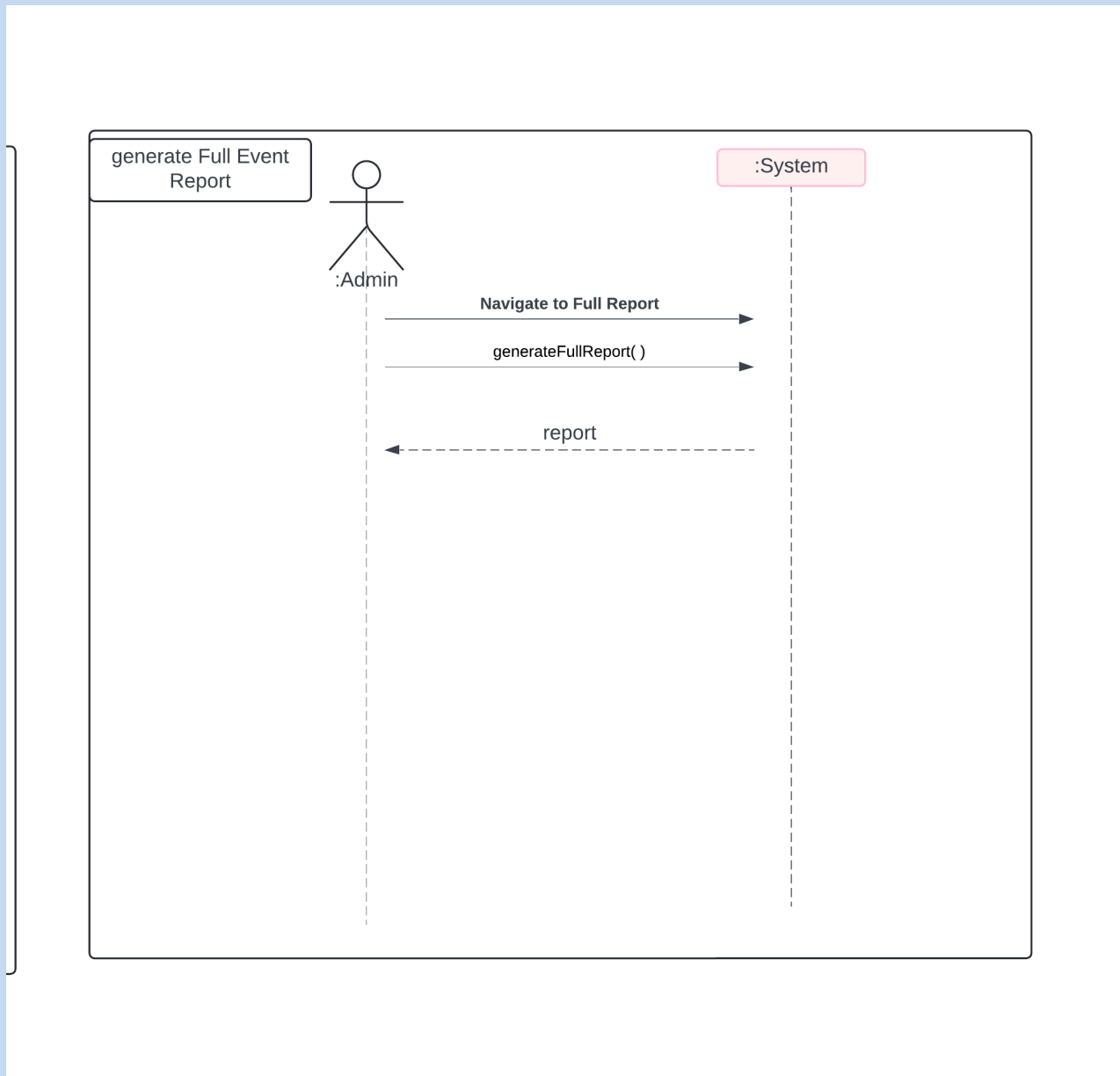
4.4 Register User



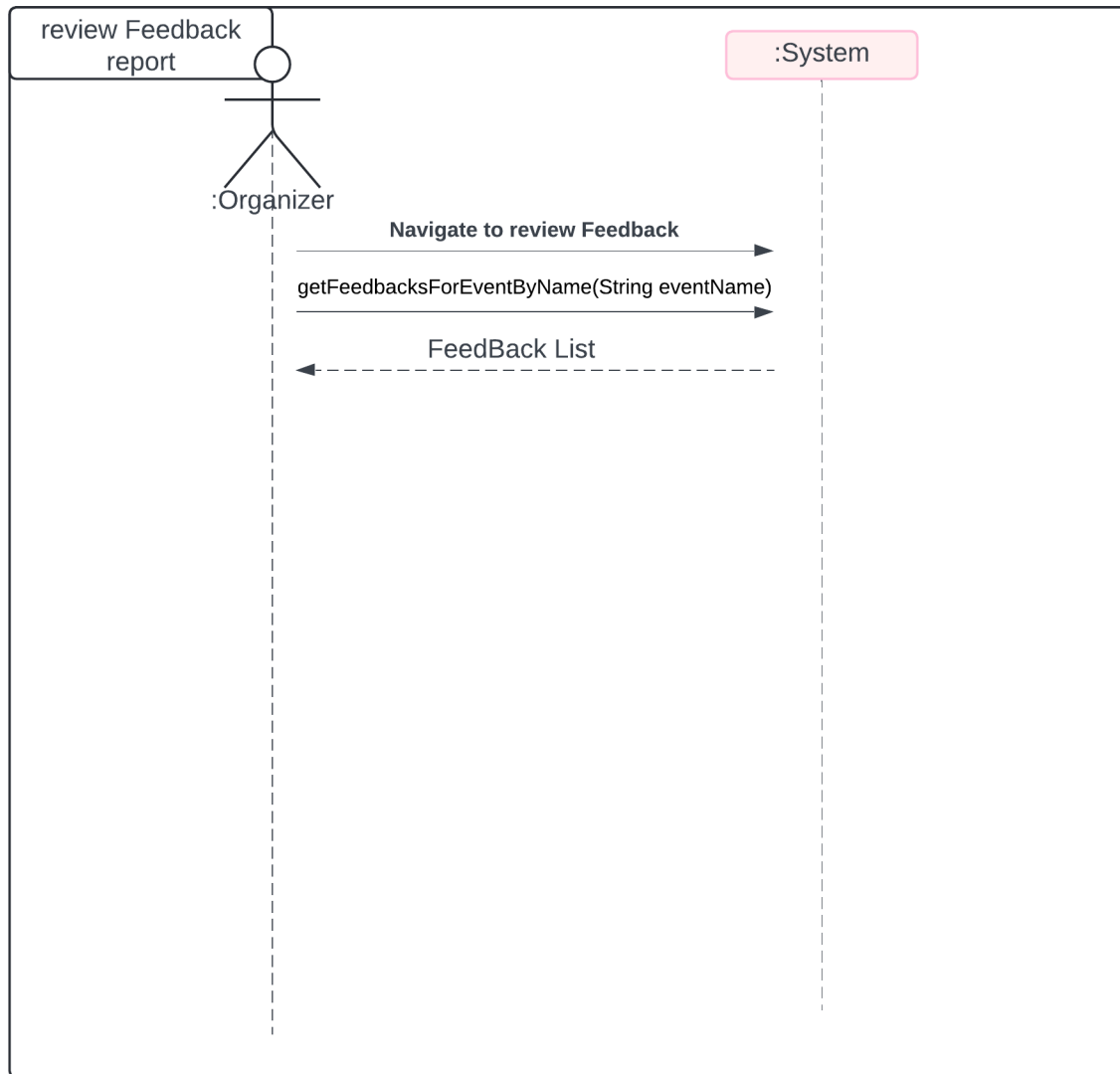
4.5 View Event Details



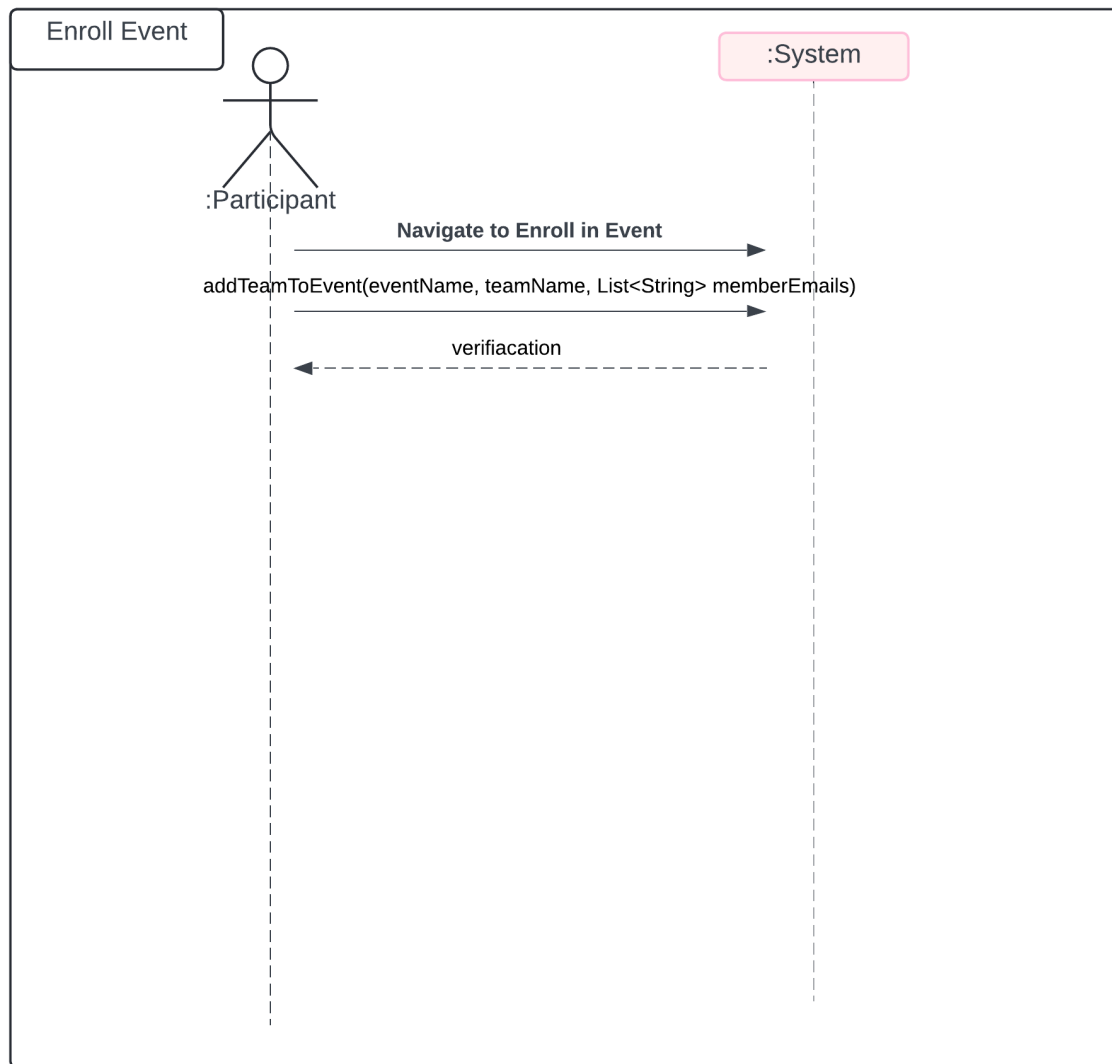
4.6 Generate Full Event Report



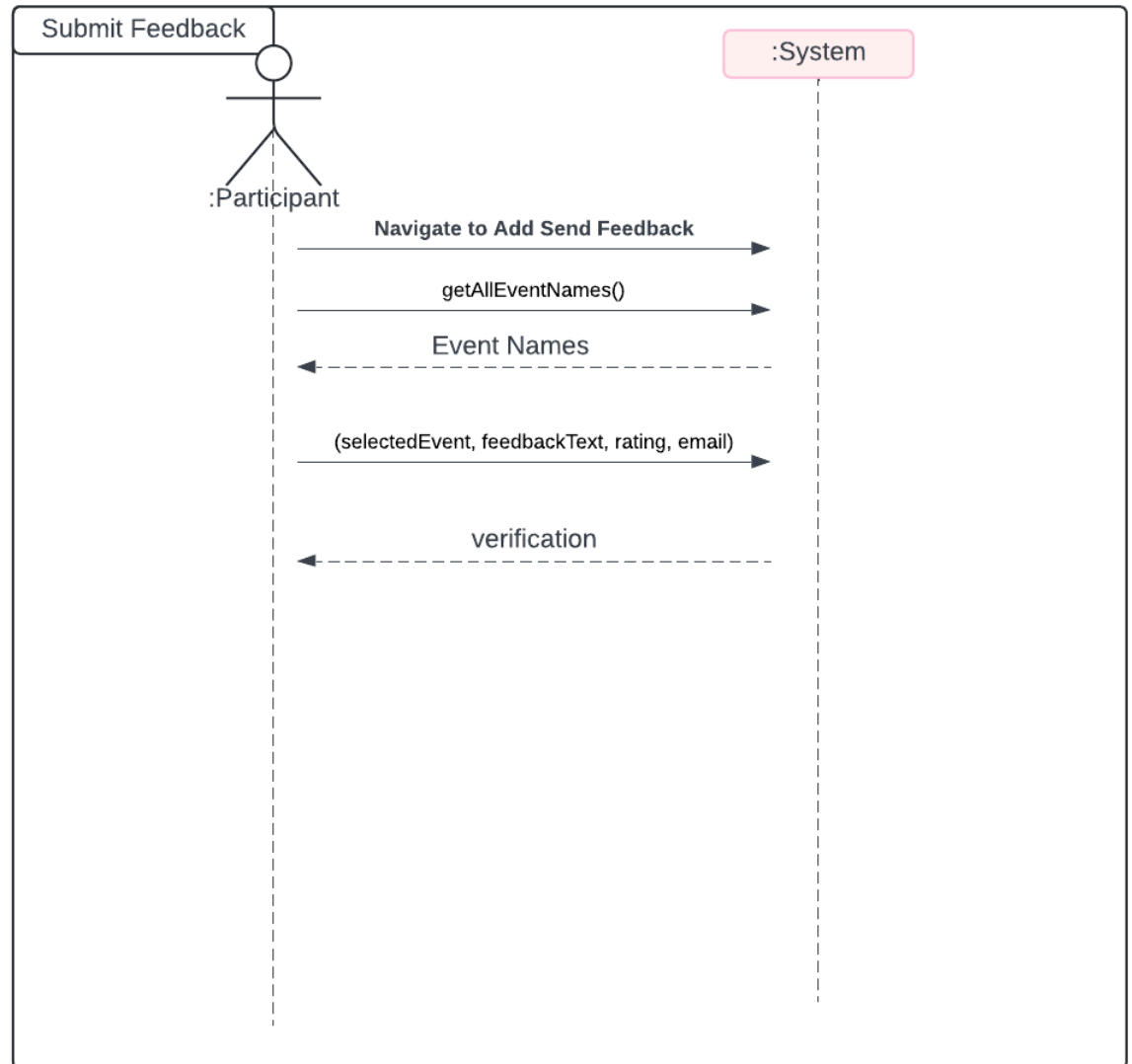
4.7. Review Feedback Report



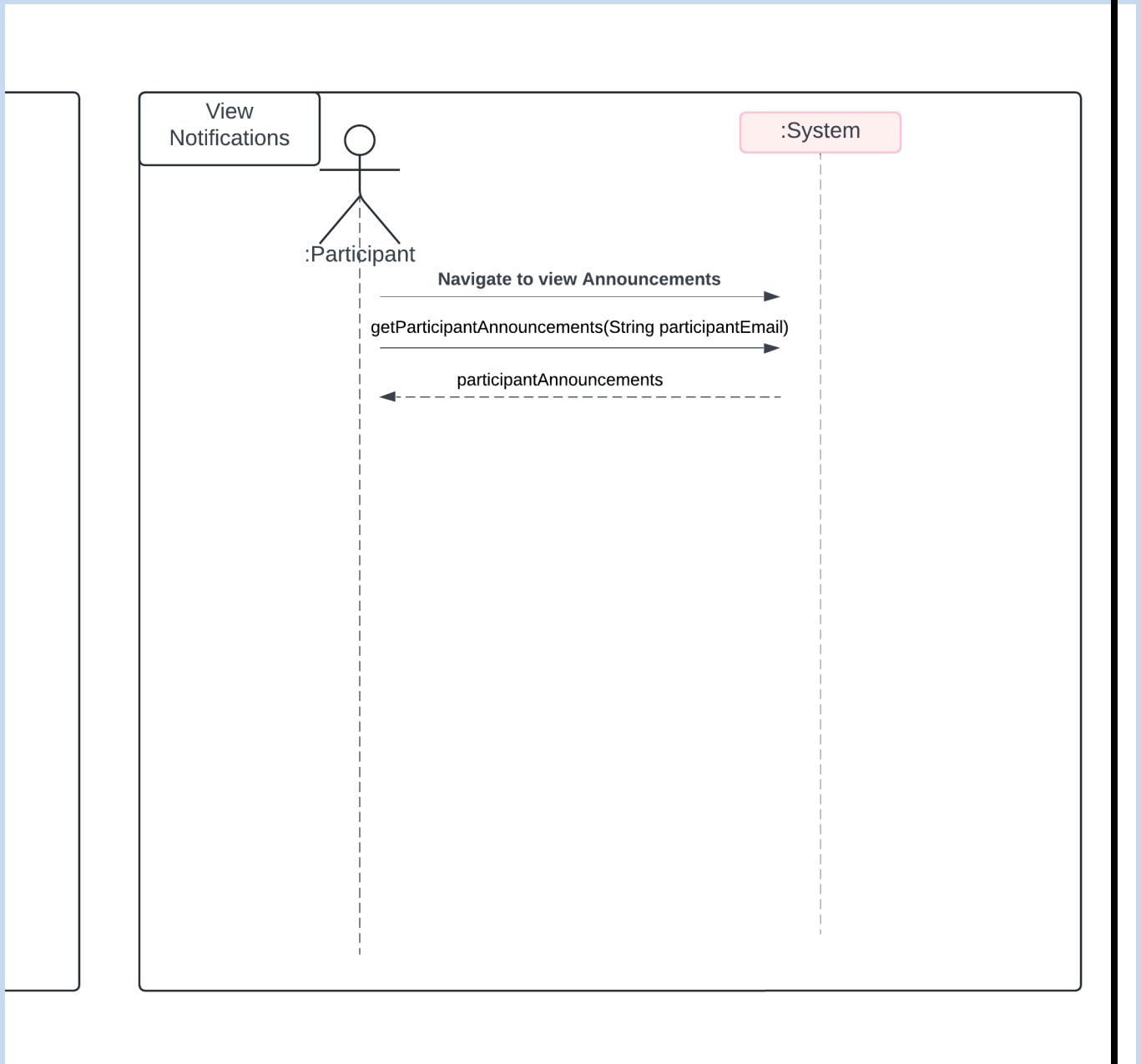
4.8 Enroll Event



Submit FeedBack

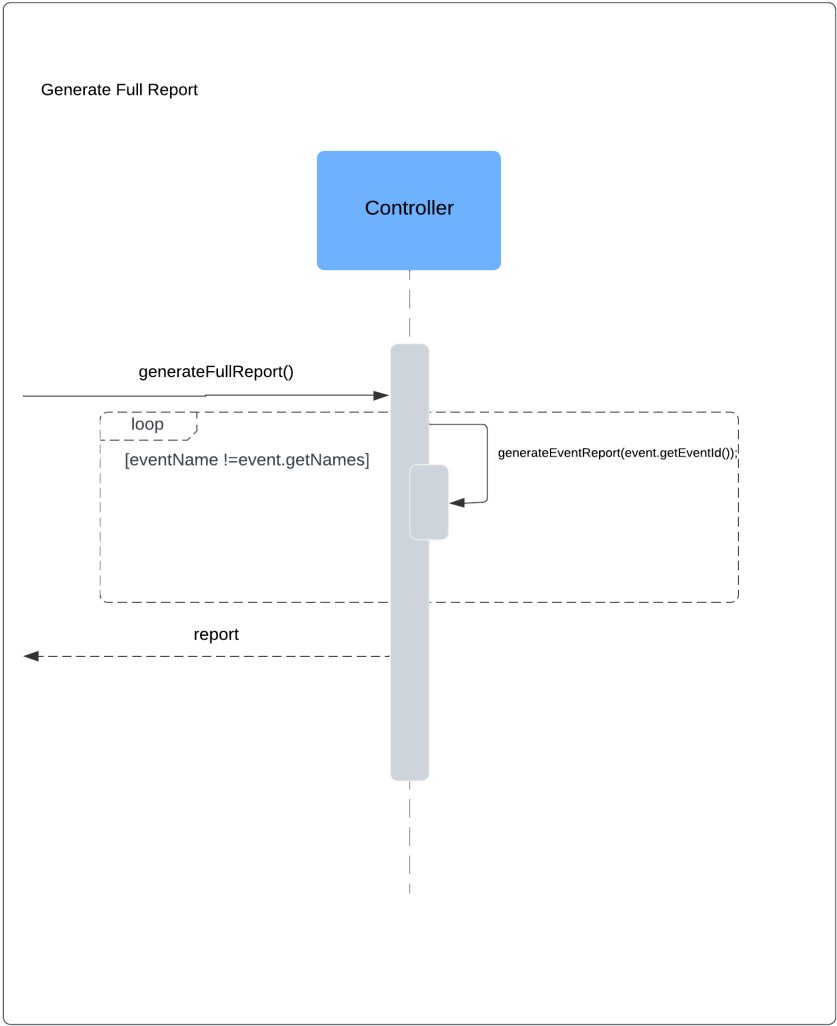


View Notification

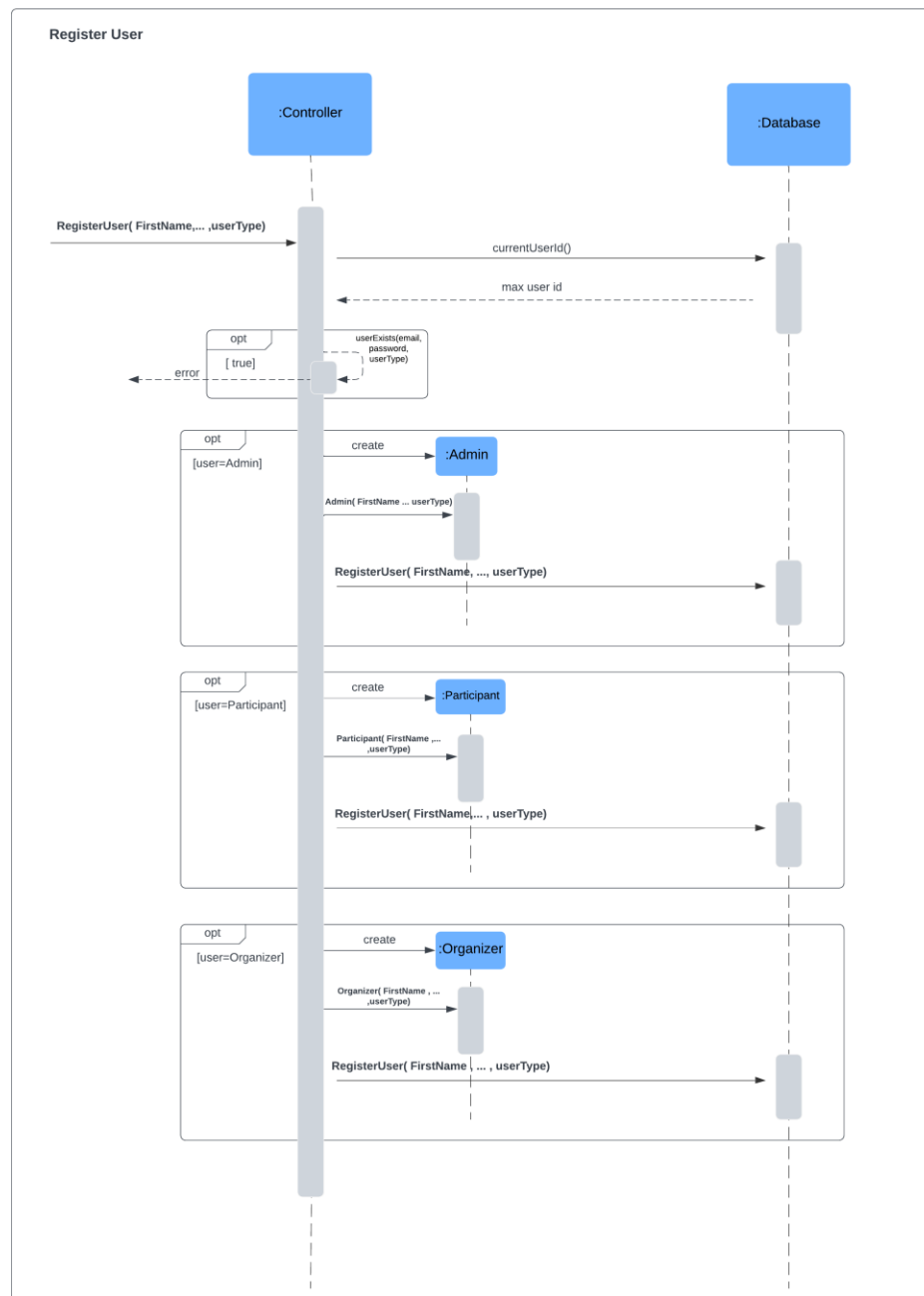


5. Sequence Diagram

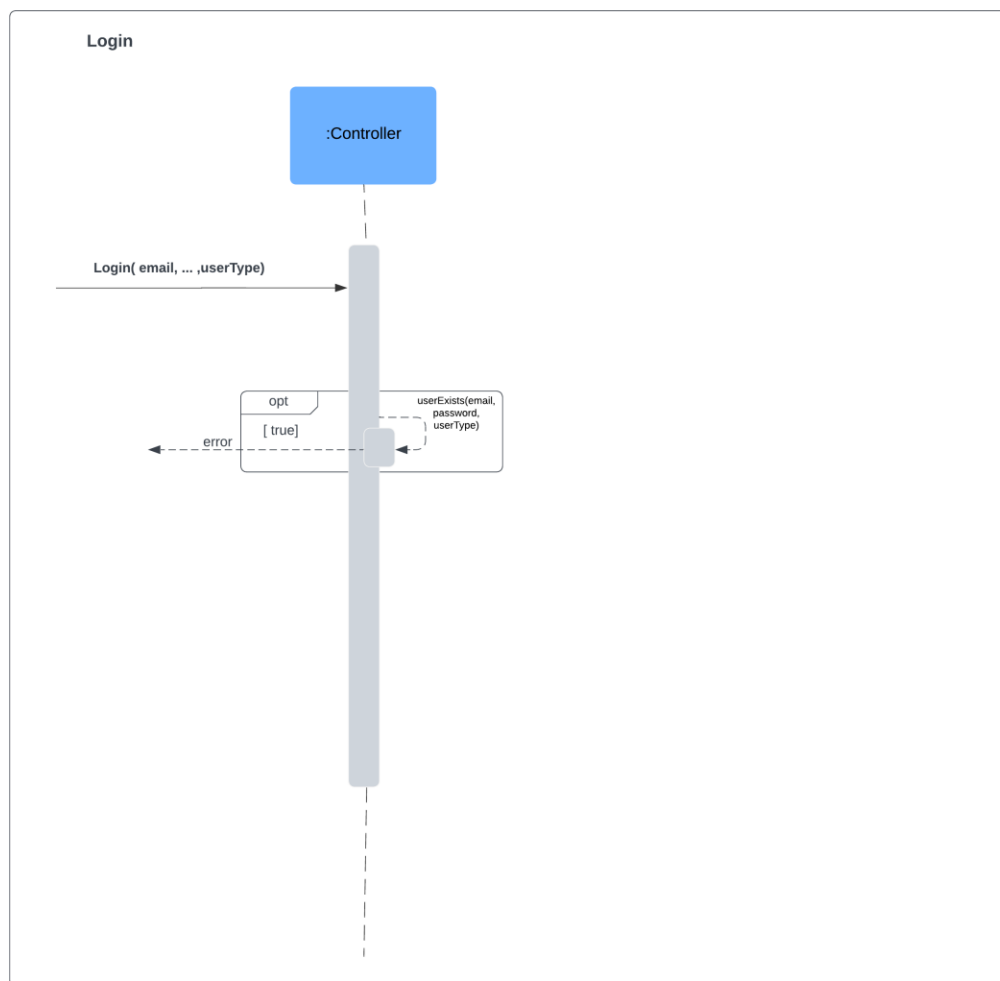
5.1 Generate Full report



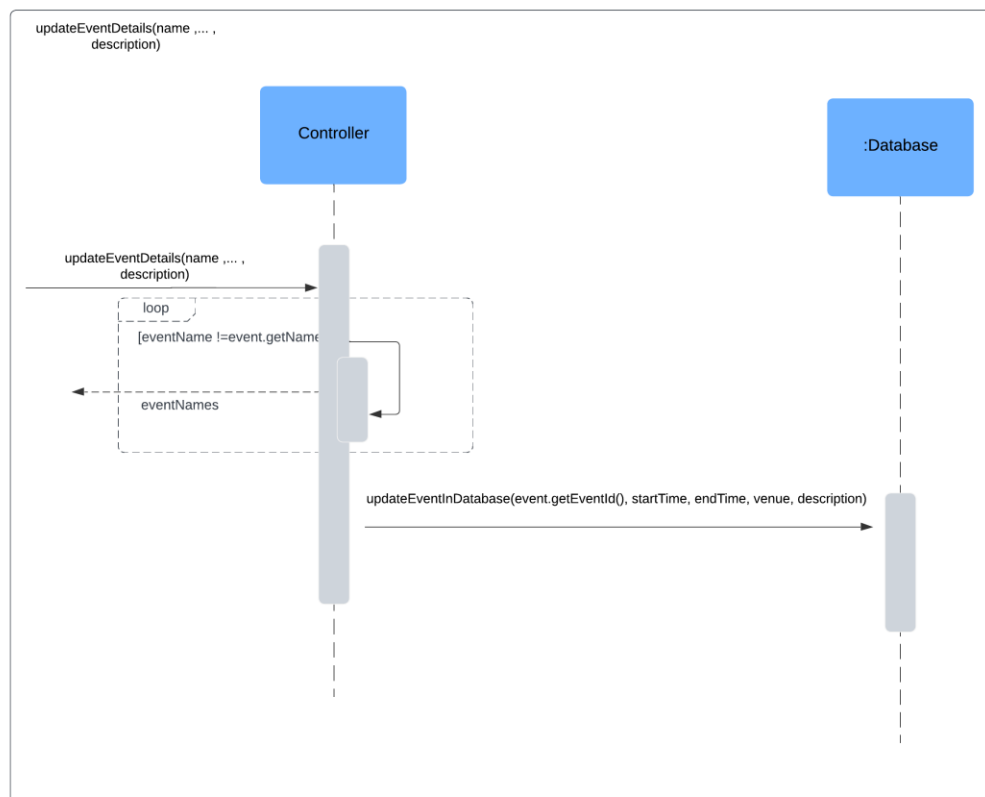
5.2 Register User



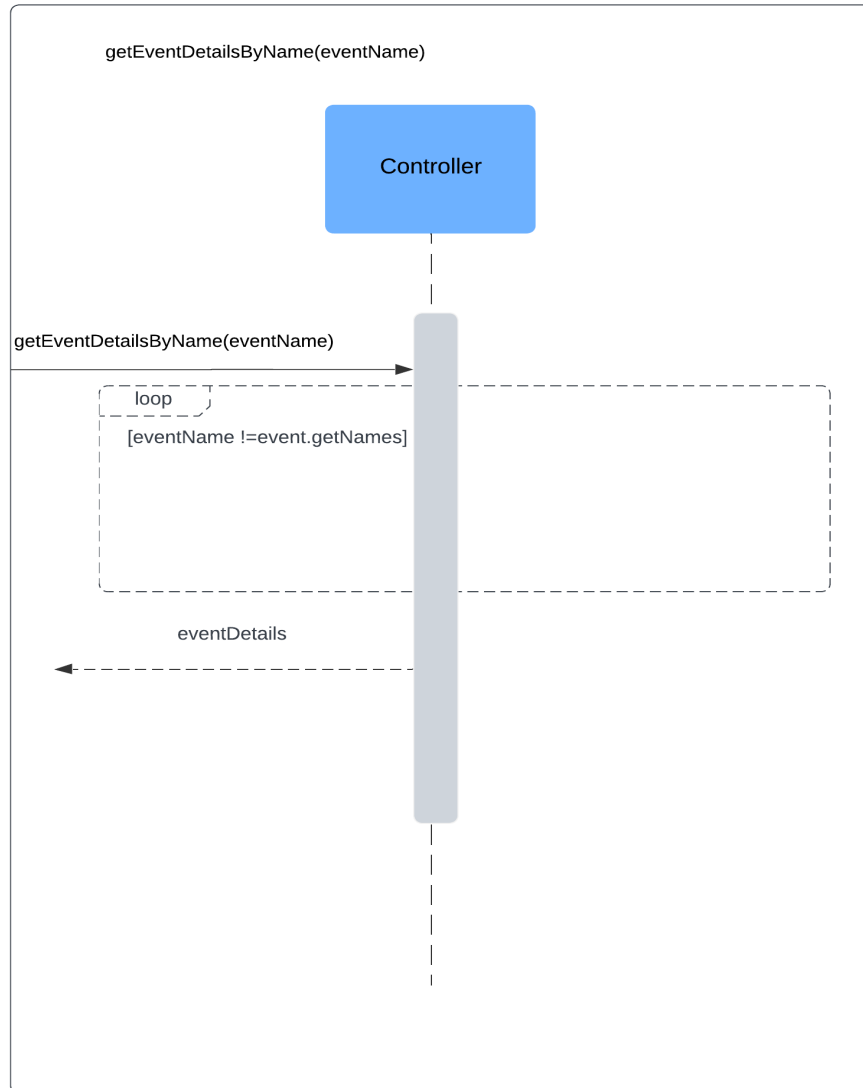
5.3 Login

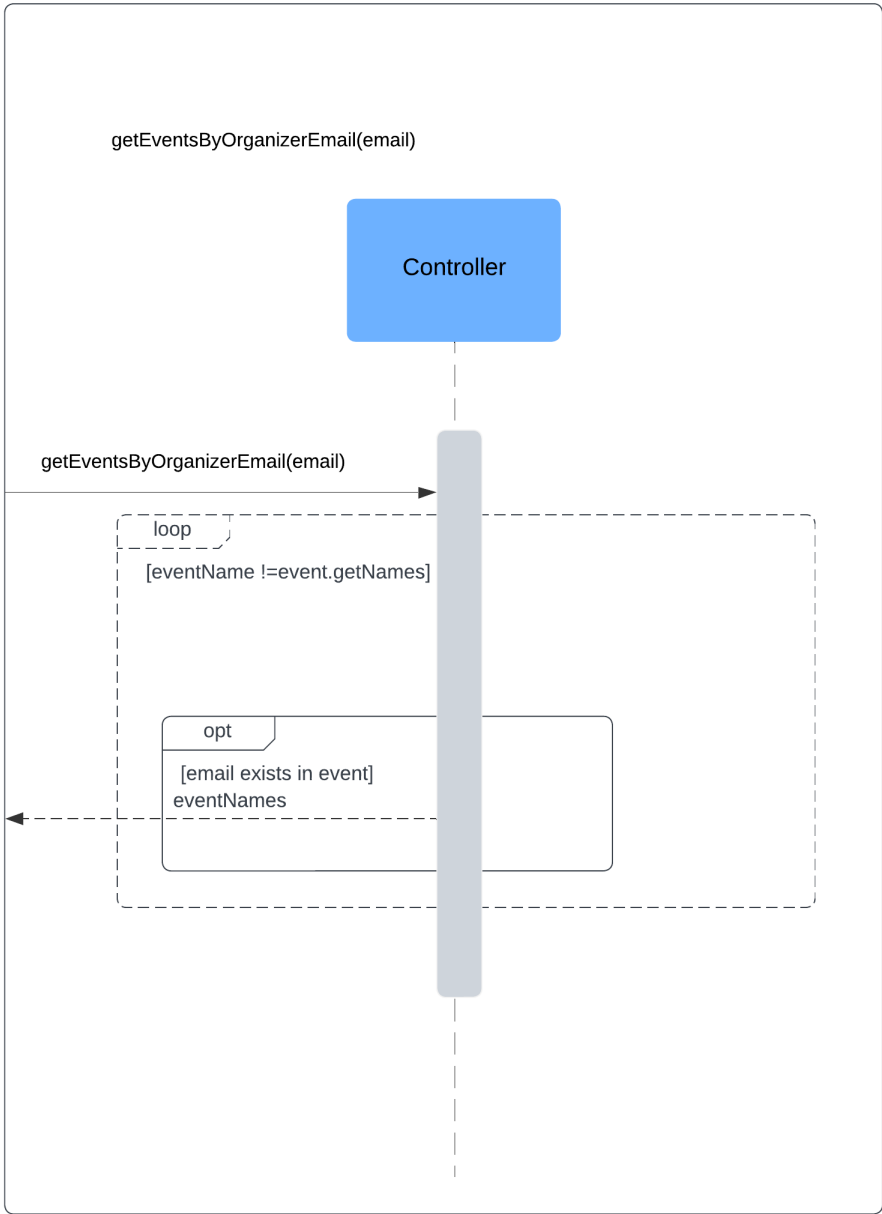


5.4 Modify Event

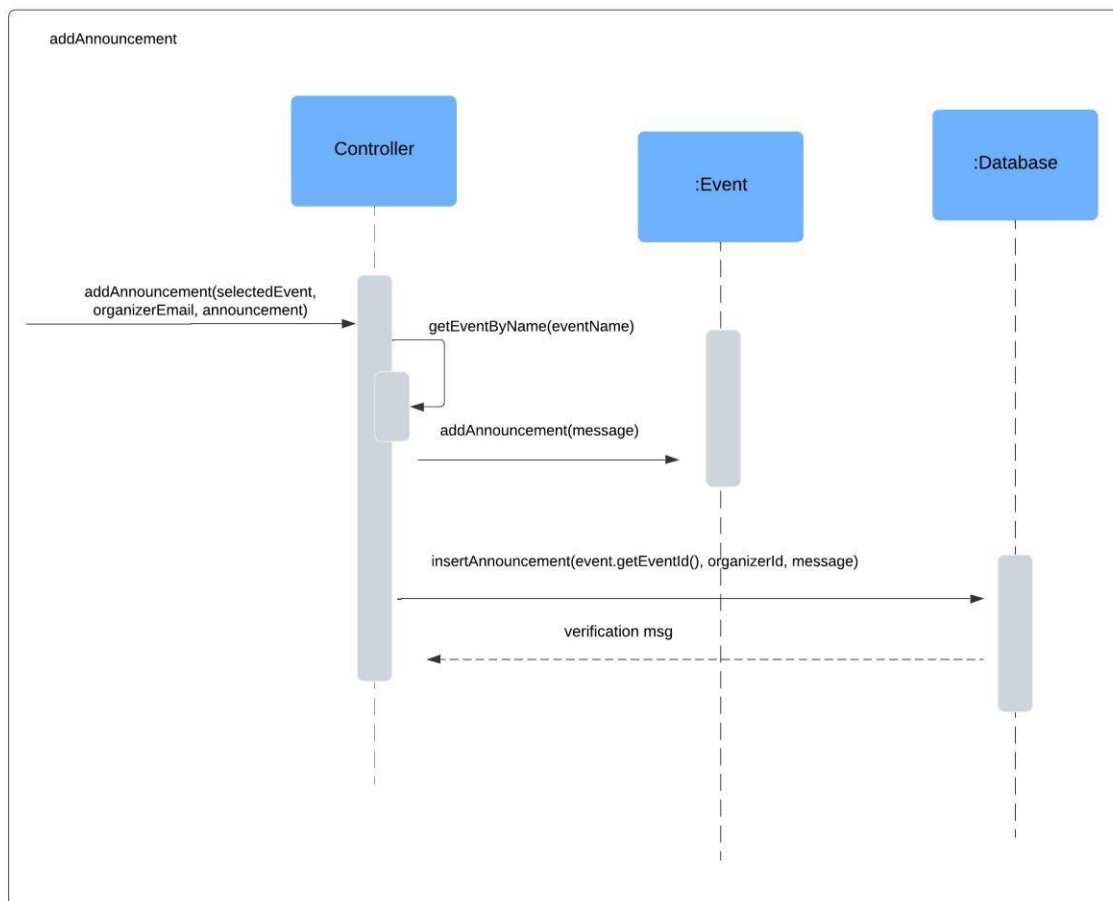


5.5 View Event

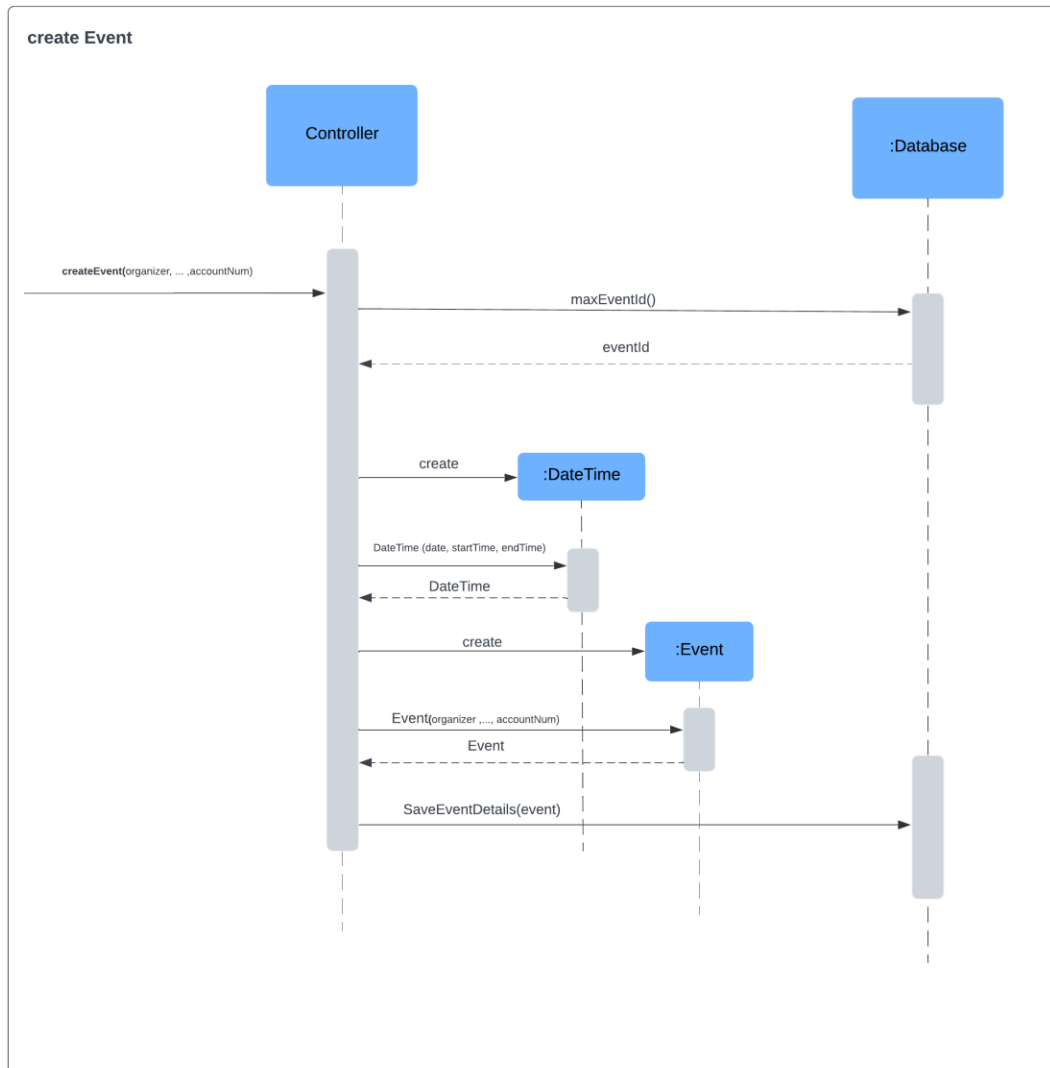


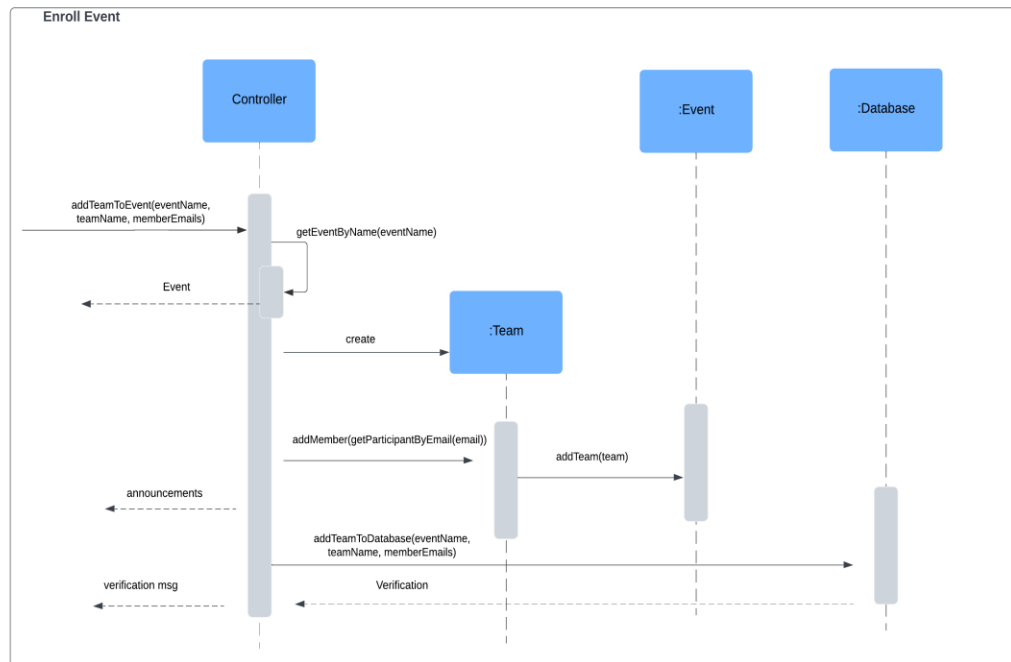


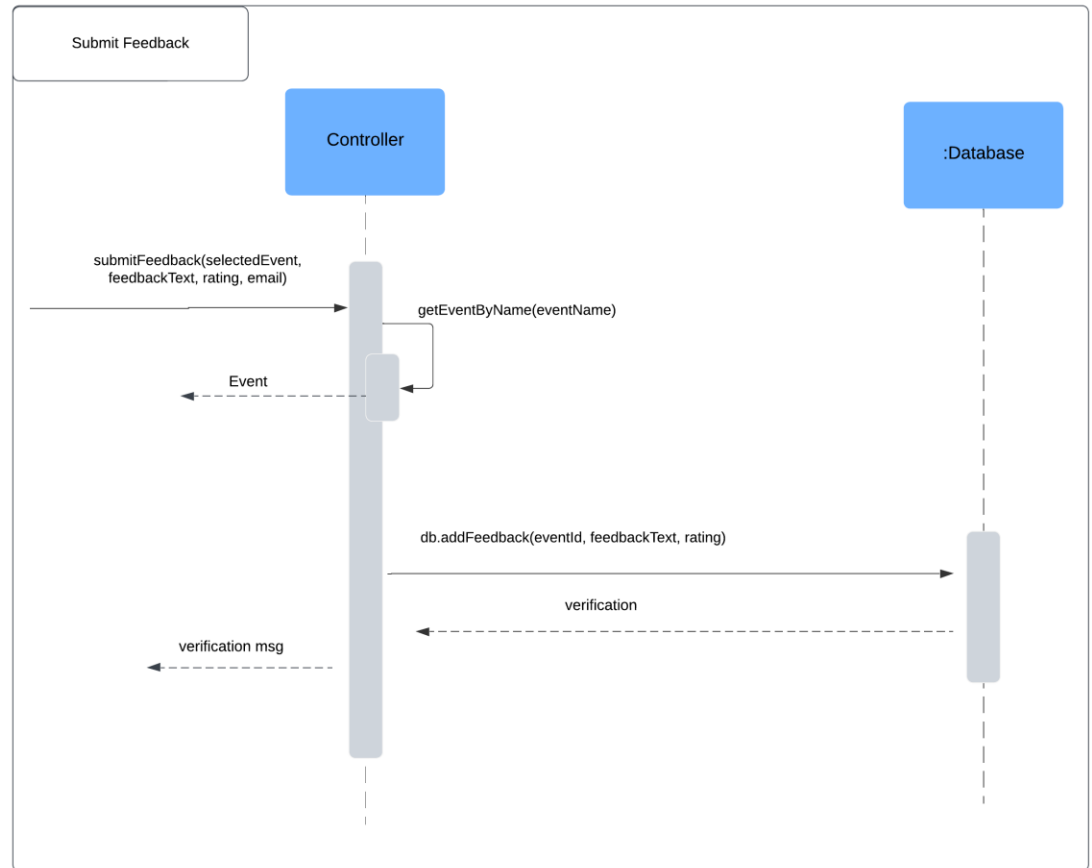
5.6 Add Announcement



5.7 Create Event



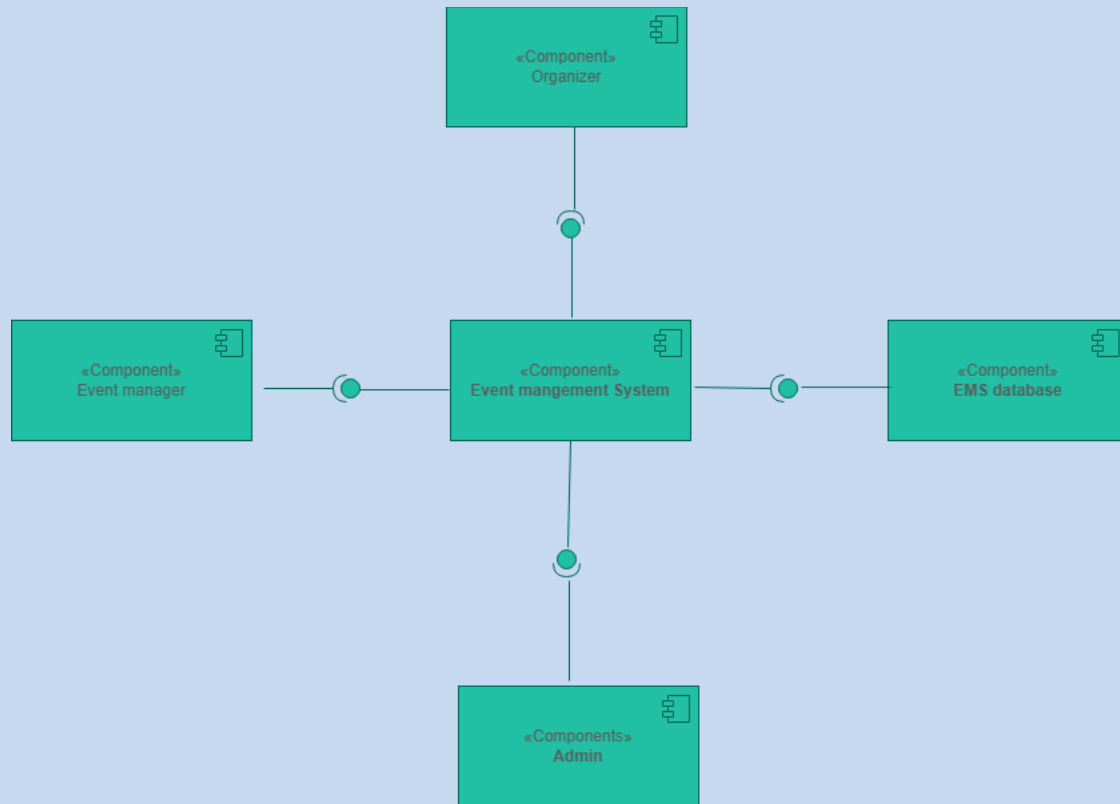




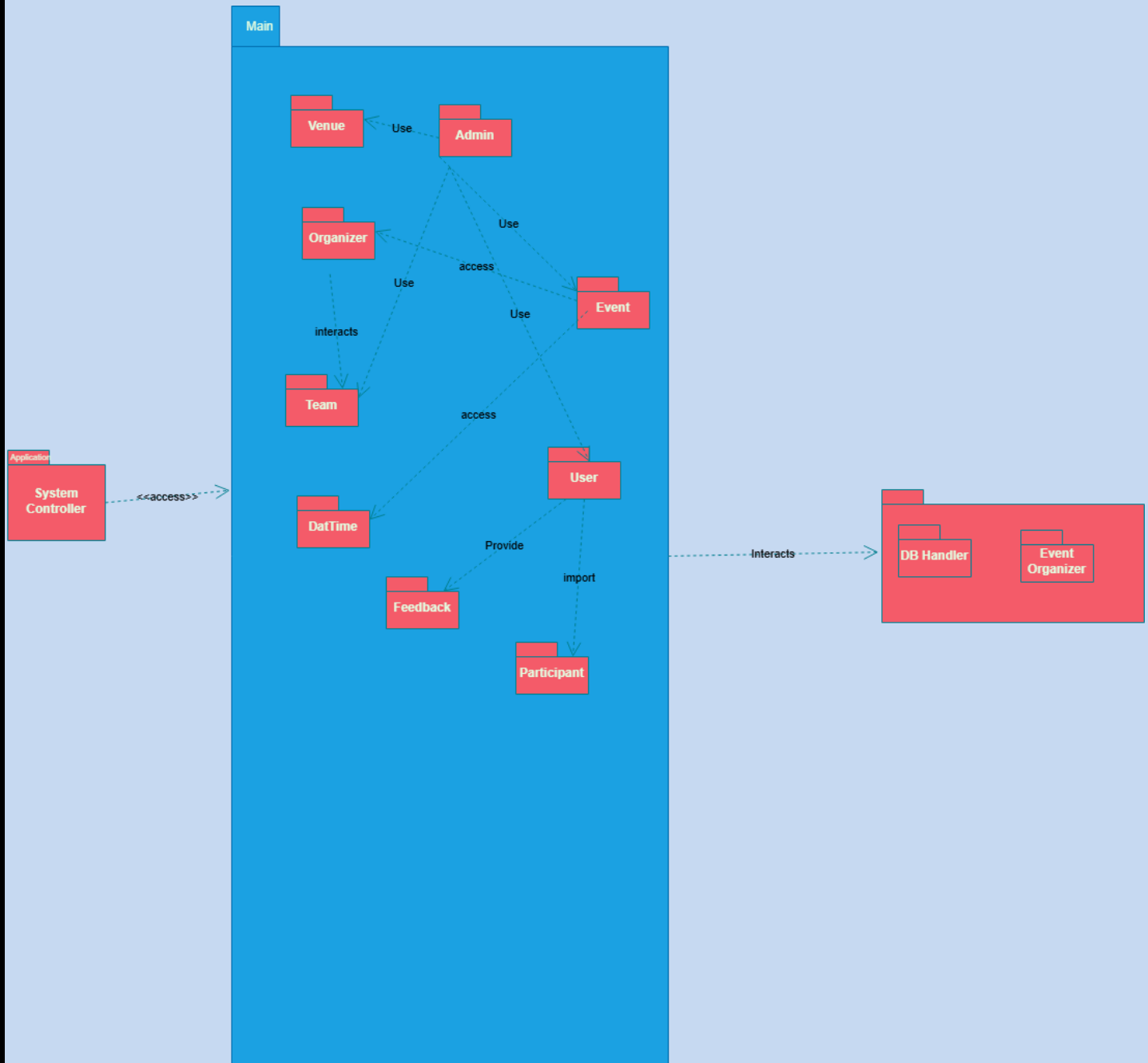
6. Class Diagram



7. Component Diagram



8. Package Diagram



9. Deployment Diagram

