# Software Requirements and Design Document

# for

# Nescon Management System

## Prepared by

**Saif ul haq 22i-1306**

**Bilal Rasheed 22i-1337**

## Organization

**Fast University**

## Date Created

**27-11-2024**

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The **Nescon Event Management System** is a software application designed to streamline the planning and execution of events, specifically for the **NESCON conference** at FAST University. This system aims to facilitate efficient management of event-related tasks, enhancing the experience for organizers, participants, and sponsors.

## 1.2 Product Scope

*The **Nescon Event Management System** is designed to streamline the planning and execution of NESCON at FAST University. It automates key tasks like registration, scheduling, and resource management while providing real-time updates and notifications to participants, speakers, and sponsors. The system enhances user experience, optimizes resource usage, and improves data management for insights into event performance. By digitizing event operations, it aligns with FAST University's goal of fostering innovation and excellence, ensuring a smooth and professional event experience.*

## 1.3 Title

*NESCON  Management System*

## 1.4 Objectives

**Streamlined Event Registration**: Facilitate a user-friendly registration process for participants, speakers, and sponsors.
**Efficient Scheduling**: Organize event schedules, ensuring proper coordination of workshops, talks, and meetings.
**Resource Management**: Track and allocate venues, equipment, and staff effectively to optimize event operations.
**Enhanced User Experience**: Provide real-time event updates and accessible information to improve participant engagement.
**Task Automation**: Automate administrative tasks such as sending reminders, managing RSVPs, and generating reports.
**Improved Communication**: Enable smooth coordination and communication among stakeholders, fostering collaboration.
**Data Management and Analytics**: Collect and analyze event data to inform decision-making for future events.

## 1.5 Problem Statement

*Managing large-scale events like Nescon involves numerous challenges, including coordinating participant registrations, scheduling sessions, allocating resources, and ensuring smooth communication among stakeholders. Traditional methods often rely on manual processes and fragmented tools, which can lead to inefficiencies, errors, and delays. These issues negatively impact the overall experience for participants, speakers, and organizers.*

*The Nescon Management System aims to address these challenges by providing an integrated, automated solution for event management. By digitizing processes such as registration, resource allocation, and schedule management, the system minimizes manual effort, reduces errors, and enhances operational efficiency. Additionally, real-time updates and user-friendly interfaces improve participant engagement and satisfaction, ensuring the success of events. This system is both feasible and scalable, offering a sustainable solution for managing complex event operations effectively.*

# 2. Overall Description

## 2.1 Product Perspective

*The Nescon Management System is a new, standalone software product designed to simplify and enhance the management of large-scale events like Nescon. It is not a replacement for any existing system but rather a self-contained solution tailored to meet the specific needs of event organizers, participants, and Staff.*

*The system is designed to automate key event management processes, such as participant registration, session scheduling, resource allocation, and communication. It integrates seamlessly with common platforms like email and mobile notifications to ensure real-time updates and improved user interaction.*

*If incorporated into a larger event management ecosystem, the Nescon Management System can act as a central hub, interfacing with modules for payment processing, data analytics, and post-event feedback. Below is a simple diagram illustrating its core components and interfaces:*

***System Overview Diagram:***

- ***Core Modules**:*
    - *Registration Management*
    - *Schedule Management*
    - *Resource Allocation*
- ***External Interfaces**:*

    - *Email and SMS Notification Systems*
    - *Payment Gateways (optional integration)*

*This modular and flexible architecture ensures that the system can be adapted to different types of events while maintaining high performance and usability.*

## 2.2 Product Functions

*The Nescon Management System provides a comprehensive set of functionalities to streamline event management. Below is a high-level summary of its core features:*

- ***Registration Management**:*
    - *Easy registration for participants, Staff.*
    - *Automated confirmation emails and reminders.*
- ***Event Scheduling**:*
    - *Creation and management of event schedules, including workshops, talks, and meetings.*
    - *Real-time updates for schedule changes.*
- ***Resource Allocation**:*
    - *Tracking and assignment of venues, equipment, and staff for events.*
    - *Ensuring optimal use of resources based on event needs.*
- ***Participant Interaction**:*
    - *Notifications for updates and reminders through email or mobile.*
    - *Providing event information and resources in real time.*
- ***Automation of Administrative Tasks**:*
    - *Managing RSVPs, reminders, and post-event tasks efficiently.*
    - *Generating reports for event analysis.*

***Top-Level Diagram:***

*The system is designed with modular components to interact seamlessly:*

1. ***User Interface** (Participants, Staff)*
2. ***Back-End Modules** (Database, Scheduling, Resource Management)*
3. ***External Interfaces** (Notification Systems)*

*These functions aim to reduce the manual effort involved in event planning while enhancing the experience for all stakeholders.*

## 2.3 List of Use Cases

Here are the names of the use cases based on the detailed descriptions you provided:

1. Register for Event
2. View Event Schedule

3. Assign Resources
4. Send Event Reminders
5. Create Event Schedule
6. Allocate Staff
7. Register for Staff/Volunteer
8. Generate Certificates
9. Ask Event Query
10. Provide Feedback
11. Register for Transport Services

## 2.4 Extended Use Case

### Fully Dressed Use Case: Register for Event

**Use Case Name:** Register for Event

**Primary Actor:** Participant

**Scope:** Nascon Management System

**Level:** User-goal

**Stakeholders and Interests**

- Participant: Wants to easily register for the event and receive a confirmation.
- Event Organizer: Wants to efficiently manage participant registrations and ensure the registration process is seamless.
- Nascon Management System: Ensures accurate recording and processing of participant information.

**Preconditions**

- The participant has access to the Nascon Management System.
- The event details (including available slots) are available in the system.

## Success Guarantee (Postcondition)

- Participant registration is recorded successfully.
- Participant receives confirmation of registration.
- Event details are updated to reflect the newly registered participant.

## Main Success Scenario

| Actor Action | System Responsibility |
|---|---|
| **1.** Participant selects the event they want to register for. | **2.** System displays the registration form for the selected event. |
| **3.** Participant fills in the required information, such as name, email, and phone number. | **4.** System validates the provided information. |
| | **5.** System checks the availability of slots for the event. |
| **6.** Participant submits the registration form. | **7.** System registers the participant for the event. |
| | **8.** System sends a confirmation email/message to the Participant. |
| | **9.** System updates the event data to reflect the new registration. |

## Extensions

**1a. Event is full:**

System informs the Participant that the event is full.

**4a. Participant provides incomplete or invalid information:**

System displays an error message indicating the missing or incorrect information. Participant corrects the information and resubmits.

**8a. Email sending fails:**

System retries sending the email.

If still unsuccessful, System logs the issue and notifies the Event Organizer to manually contact the Participant.

# Fully Dressed Use Case: View Event Schedule

**Use Case Name:** View Event Schedule

**Primary Actor:** Participant, Event Organizer

**Scope:** Nascon Management System

**Level:** User-goal

## Stakeholders and Interests

- **Participant:** Wants to view the event schedule to know the timing of sessions they are interested in.
- **Event Organizer:** Wants to ensure participants have access to accurate and up-to-date event schedules.
- **Nascon Management System**: Provides a platform for managing and presenting event schedules.

## Preconditions

- The participant or event organizer has access to the Nascon Management System.
- The event schedule is already created and available in the system.

## Success Guarantee (Postcondition)

- The participant or event organizer successfully views and understands the required event schedule.

## Main Success Scenario

| Actor Action | System Responsibility |
|---|---|
| **1.** Participant selects the option to view the event schedule. | **2.** System displays a list of available events. |
| **3.** Participant selects the specific event they want to view. | **4.** System presents the schedule of the selected event, including sessions, timings, and venues. |

## Extensions

**1a. No events available:**

System displays a message informing that no events are currently scheduled.

**3a. Event details are missing or incomplete:**

System displays a message indicating the missing information and suggests contacting the event organizer for more details.

# Fully Dressed Use Case: Assign Resources

## Use Case Name: Assign Resources

## Primary Actor: Event Organizer

## Scope: Nascon Management System

## Level: User-goal

## Stakeholders and Interests:

- **Event Organizer:** Wants to assign resources such as staff, equipment, or venues for an event.
- **Participant:** Wants to be assigned to the appropriate resources or receive updates related to resource assignments.

## Preconditions

- The event has been created, and available resources have been listed in the system.

## Postconditions

- Resources are successfully assigned to the event, and the relevant stakeholders are notified of the assignments.

## Main Success Scenario

| Actor Action | System Responsibility |
|---|---|
| **1.** Organizer logs into the Nascon Management System. | **2.** System verifies credentials and grants access. |
| **3.** Organizer selects an event to assign resources. | **4.** System displays event details and available resources. |
| **5.** Organizer assigns staff, equipment, or venues to the event. | **6.** System records the assigned resources. |

**7.** Organizer reviews and confirms the assignments.

**8.** System saves the assignments and updates the event records.

**9.** System notifies assigned staff and stakeholders of the resource assignments.

**10.** Assigned resources receive notifications of their tasks or involvement in the event.

### Extensions

**4a. Resource not available:** System informs the organizer and provides alternate resource suggestions.

**7a. Organizer cancels the assignment:** System removes the resource from the event and notifies affected stakeholders.

## Fully Dressed Use Case: Send Event Reminders

**Use Case Name:** Send Event Reminders

**Primary Actor:** Event Organizer

**Scope:** Nascon Management System

**Level:** User Goal

### Stakeholders and Interests:

- **Event Organizer:** Wants to ensure participants are informed about upcoming events.
- **Participants**: Want to be reminded of upcoming events.

### Preconditions:

- Event details must be created and saved in the system.
- Participants must be registered for the event.
- Event Organizer must be logged into the system.

### Postconditions (Success Guarantee):

- Participants receive reminders about upcoming events.

- Event Organizer is notified that reminders were successfully sent.

### Main Success Scenario:

| Actor Action | System Responsibility |
|---|---|
| **1.** Event Organizer selects the event for which reminders need to be sent. | **2.** System displays the list of registered participants. |
| **3.** Event Organizer chooses the reminder template and the date/time to send reminders. | **4.** System schedules the reminder for the selected date and time. |
| **5.** Event Organizer confirms the list of participants and sends the reminders. | **6.** System sends the reminders to all registered participants. |
| **7.** Event Organizer is notified that reminders were sent. | |

### Extensions:

**1a. No participants registered:**

System notifies the Event Organizer that no participants are registered.

**2a. System error while sending reminders:**

System logs the error and notifies the Event Organizer.

Event Organizer retries or contacts support.

**3a. Reminder sent but participant did not receive it (email failure):**

System logs the failed delivery.

Event Organizer is notified to contact the participant.

## Fully Dressed Use Case: Generate Event Reports

**Use Case Name:** Generate Event Reports

**Primary Actor:** Event Organizer

**Scope:** Nascon Management System

**Level:** User Goal

### Stakeholders and Interests:

**Event Organizer:** Wants to generate detailed reports on events for analysis and decision-making.

**Management:** Wants to review event performance and participant engagement.

### Preconditions:

- Event data must be available in the system (e.g., participant data, event details, and feedback).
- Event Organizer must be logged into the system.

### Postconditions (Success Guarantee):

- Event Organizer receives a detailed report on the selected event.
- Report is available for download or further analysis.

### Main Success Scenario:

| Actor Action | System Responsibility |
|---|---|
| **1.** Event Organizer selects the event for which they want to generate a report. | **2.** System retrieves all relevant event data (e.g., participant list, feedback, registration numbers). |
| **3.** Event Organizer requests the report to be generated. | **4.** System generates the report and makes it available for download or viewing. |
| **5.** Event Organizer views the report. | **6.** System displays the report in the desired format (PDF, Excel, etc.). |
| **7.** Event Organizer downloads the report. | **8.** System ensures that the report is saved in the system for future access. |
| **9.** Event Organizer uses the report for analysis or decision-making. | |

### Extensions:

**1a. No events available:**

System notifies the Event Organizer that no events are available for reporting.

**3a. Error in report generation:**

System logs the error and notifies the Event Organizer to retry or contact support.

**4a. System generates report but some data is missing (e.g., incomplete feedback):**

System warns the Event Organizer that some data might be incomplete.

## Fully Dressed Use Case: Create Event Schedule

**Use Case Name:** Create Event Schedule

**Primary Actor:** Event Organizer

**Scope:** Nascon Management System

**Level:** User Goal

## Stakeholders and Interests:

- **Event Organizer:** Wants to create a detailed schedule for an event.
- **Participants:** Want to know the schedule to plan their participation.

## Preconditions:

- Event Organizer must be logged into the system.

## Postconditions (Success Guarantee):

- A detailed event schedule is created and saved in the system.
- Participants and speakers are notified of the schedule.
- System logs the creation of the event schedule.

## Main Success Scenario:

| Actor Action | System Responsibility |
|---|---|
| **1.** Event Organizer selects the event for which the schedule is to be created. | **2.** System retrieves event details (date, location, speakers, participants). |
| **3.** Event Organizer inputs the schedule details (time slots, sessions, breaks, etc.). | **4.** System validates the inputs and displays a preview of the schedule. |
| **5.** Event Organizer reviews and confirms the schedule. | **6.** System saves the schedule and notifies participants and speakers. |
| **7.** Event Organizer makes any necessary adjustments to the schedule. | **8.** System updates the schedule and re-notifies relevant parties. |
| | **9.** System generates the schedule in the required format (PDF, Excel, etc.). |

## Extensions:

- **1a. Incomplete or invalid schedule data:** System prompts Event Organizer to correct the inputs before saving.

- **3a. Event Organizer needs to make last-minute changes:** System allows real-time updates and re-notifies participants and speakers.
- **6a. System failure while saving schedule:** System logs the error and notifies the Event Organizer to retry or contact support.

## Fully Dressed Use Case: Allocate Staff

**Use Case Name:** Allocate Staff

**Primary Actor:** Event Organizer

**Scope:** Nascon Management System

**Level:** User Goal

### Stakeholders and Interests:

- **Event Organizer:** Wants to assign staff to specific roles and responsibilities for the event.
- **Staff:** Want to know their assigned roles and shifts.

### Preconditions:

- Event details (date, location, and staff availability) must be created.
- Event Organizer must be logged into the system.
- A pool of available staff members must be present in the system.

### Postconditions (Success Guarantee):

- Staff members are allocated to specific roles and responsibilities for the event.
- Staff members are notified of their assignments.

### Main Success Scenario:

| Actor Action | System Responsibility |
| --- | --- |
| **1.** Event Organizer selects the event and opens the staff allocation interface. | **2.** System retrieves the event details and available staff information. |

| | |
|---|---|
| **3.** Event Organizer assigns staff to various roles (e.g., registration, technical support). | **4.** System displays available staff and their qualifications. |
| **5.** Event Organizer confirms staff assignments for each role. | **6.** System validates the assignments and ensures no conflicts (e.g., double-booking staff). |
| **7.** Event Organizer saves the allocation. | **8.** System saves the staff allocation and notifies the staff of their assigned roles. |

### Extensions:

- **1a. Staff member unavailable**: System prompts the Event Organizer to select a different staff member or modify the schedule.
- **3a. Event Organizer changes staff assignments after confirmation:** System updates the staff allocation and re-notifies affected staff members.

## Fully Dressed Use Case: Register for Staff/Volunteer

**Primary Actor:** Participant

**Scope:** Nascon Management System

**Level:** User Goal

### Stakeholders and Interests:

- **Participant (Staff/Volunteer):** Wants to register as staff or volunteer for an event to take on specific responsibilities.
- **Event Organizer:** Wants to ensure that enough staff and volunteers are registered to manage the event.

### Preconditions:

- The event must be created and available for registration in the system.
- The event must have open positions for staff or volunteer roles.

### Postconditions (Success Guarantee):

- The participant is registered as staff or a volunteer for the selected event.

- The participant is notified of their role and any further instructions.
- The system logs the participant's registration.

## Main Success Scenario:

| Actor Action (or Intention) | System Responsibility |
|---|---|
| **1.** Participant selects an event to register for. | **2.** System displays available staff/volunteer roles for the selected event. |
| **3.** Participant chooses a staff or volunteer role and submits the registration. | **4.** System validates the registration and checks for role availability. |
| **5.** Participant confirms the registration details. | **6.** System saves the registration and sends confirmation with further instructions. |
| **7.** Participant can view their registered roles and responsibilities in their profile. | **8.** System logs the registration and allows participants to access role details later. |

## Extensions:

- **2a. Selected role is no longer available:** System notifies the participant and suggests alternative roles.
- **3a.Participant decides to withdraw from the role after registering:** System allows the participant to cancel their registration and notifies the Event Organizer.
- **4a.System failure during registration:** System logs the error and prompts the participant to retry or contact support.

# Fully Dressed Use Case: Generate Certificates

**Use Case Name:** Generate Certificates

**Primary Actor:** Event Organizer

**Scope:** Nascon Management System

**Level:** User Goal

## Stakeholders and Interests:

- **Event Organizer:** Wants to create and distribute certificates to participants, staff, or sponsors for their involvement in the event.
- **Participants/Volunteers:** Want to receive certificates as proof of their participation or contribution to the event.

## Preconditions:

- The event must be completed.
- Event Organizer must be logged into the system.
- Certificate templates must be pre-defined in the system.

## Postconditions (Success Guarantee):

- Certificates are generated and saved in the system.
- Participants, staff, or sponsors receive their certificates via the selected distribution method (email, download link, etc.).
- The system logs the certificate generation and distribution.

## Main Success Scenario:

| Actor Action (or Intention) | System Responsibility |
|---|---|
| **1.** Event Organizer logs into the system and selects the event. | **2.** System retrieves event details. |
| **3.** Event Organizer selects the certificate generation option. | **4.** System displays available certificate templates and participants. |

**5.** Event Organizer customizes certificate details (names, roles, etc.).

**6.** System validates the input and prepares the certificates for generation.

**7.** Event Organizer confirms the generation of certificates.

**8.** System generates the certificates and saves them in the system.

**9.** System sends certificates to participants via the selected method.

**10.** System logs the distribution and updates the status of certificates.

### Extensions:

**6a. Participant information is incomplete:**

System prompts the Event Organizer to correct or complete the information before generating certificates.

**9a. Event Organizer wants to regenerate certificates after modifications:**

System allows the organizer to update certificates and re-send them as needed.

## Fully Dressed Use Case: Ask Event Query

**Use Case Name:** Ask Event Query

**Primary Actor:** Participant

**Scope:** Nascon Management System

**Level:** User Goal

### Stakeholders and Interests:

- **Participant (User):** Wants to inquire about event details, schedules, or other relevant information.
- **Event Organizer:** Wants to ensure that participants have the information they need for successful event participation.

### Preconditions:

- Relevant event information must be available in the system.

### Postconditions (Success Guarantee):

- The participant receives a clear and accurate response to their query.

- The system logs the query for future reference and analysis.
- The participant can access additional information if needed.

### Main Success Scenario:

| Actor Action (or Intention) | System Responsibility |
|---|---|
| **1.** Participant navigates to the event query section. | **2.** System displays options for types of queries (e.g., schedule, location, speakers). |
| **3.** Participant selects a query type and submits their question. | **4.** System processes the query and searches for relevant information and sends query to Event Organizer. |
| | **5.** System displays the answer in a clear format and may suggest related information. |
| **6.** Participant reviews the response and can ask follow-up questions. | |

### Extensions:

1**a. System failure during the query process:**
System logs the error and notifies the participant to retry later or contact support.

## Fully Dressed Use Case: Provide Feedback

**Use Case Name:** Provide Feedback

**Primary Actor:** Participant

**Scope:** Nascon Management System

**Level:** User Goal

### Stakeholders and Interests:

**Participant (User):** Wants to share their thoughts or suggestions about the event to improve future events.
**Event Organizer**: Seeks constructive feedback to enhance event planning and execution.

**Management:** Interested in analyzing feedback to assess overall participant satisfaction and improve future strategies.

## Preconditions:

The event must have concluded for feedback to be relevant.

## Postconditions (Success Guarantee):

- The feedback is successfully submitted and stored in the system.
- The participant receives confirmation that their feedback has been recorded.
- The Event Organizer can access and review the feedback for future improvements.

## Main Success Scenario:

| Actor Action (or Intention) | System Responsibility |
|---|---|
| **1.** Participant navigates to the feedback section. | **2.** System displays the feedback form and instructions. |
| **3.** Participant fills out the feedback form with comments and ratings. | |
| **4**. Participant submits the feedback. | **5.** System stores the feedback. |
| | **6.** System allows the Event Organizer to access and analyze feedback reports. |

## Extensions:

1a. **Incomplete feedback form:**

System prompts the participant to fill in all required fields before submission.

# Fully Dressed Use Case: Register for Transport Services

**Use Case Name:** Register for Transport Services

**Primary Actor:** Participant

**Scope:** Nascon Management System

**Level:**User Goal

## Stakeholders and Interests:

**Participant (User):** Wants to secure transportation to and from the event.
**Event Organizer:** Aims to provide transport options for participants and manage logistics efficiently.
**Management:** Wants to track transportation registrations for planning and resource allocation.

## Preconditions:

- The participant must be logged into the system.
- The transport services must be available for the event.

## Postconditions (Success Guarantee):

- The participant is successfully registered for transport services.
- The participant receives a confirmation of the registration.
- The Event Organizer and Transport Service Provider are notified of the new registration.

## Main Success Scenario:

| Actor Action | System Responsibility |
|---|---|
| **1.** Participant navigates to the transport services registration section. | **2.** System displays available transport options and schedules. |
| **3.** Participant selects the desired transport service and fills out the registration form. | **4.** System validates the input data (e.g., name, pickup location, drop-off location). |
| **5.** Participant submits the registration form. | **6.** System processes the registration and sends confirmation message. |

**7.** Participant receives a confirmation notification.

## Extensions:

**5a. Form submission is incomplete:**

System prompts the participant to fill in all required fields before submission.

**6a. Transport service is at capacity:**

System informs the participant that the selected service is fully booked and suggests alternatives.

## 2.5 Use Case Diagram

# 3. Other Nonfunctional Requirements

## 3.1 Performance Requirements

1. **System Response Time for Event Registration:**
    - **Requirement**: The system should process event registrations (for participants, speakers, and sponsors) within **5 seconds** of submitting the registration form.
    - **Rationale**: Fast registration processing ensures a smooth and efficient user experience, especially when multiple participants might be registering at the same time.
2. **Real-Time Event Schedule Updates:**
    - **Requirement**: The event schedule should be updated in real time for all users (participants, organizers) with a maximum delay of **1 second** between updates and notifications.
    - **Rationale**: Real-time updates are critical to ensure that participants are informed about any last-minute changes, such as event time shifts or speaker changes.
3. **Resource Assignment and Availability Check:**
    - **Requirement**: Resource (venue, equipment, staff) assignment should be processed within **3 seconds** for each task.
    - **Rationale**: Quick allocation and checks for resource availability are essential to avoid delays in event planning and scheduling.
4. **System Load Capacity:**
    - **Requirement**: The system should be able to handle at least **500 concurrent users** (participants, speakers, staff, and organizers) without significant degradation in performance.
    - **Rationale**: Since the event may involve hundreds of users accessing the system simultaneously, the application must be scalable and able to handle traffic spikes, such as during event registration or feedback collection.
5. **Event Feedback Submission:**
    - **Requirement**: Event feedback submissions should be stored and processed in **less than 2 seconds** for each user's submission.
    - **Rationale**: Users will likely provide feedback during or after the event. Ensuring that their input is processed quickly helps in managing large volumes of feedback efficiently.
6. **System Downtime:**
    - **Requirement**: The system should have a maximum **downtime of 2 hours per month** for maintenance and updates.
    - **Rationale**: Minimizing downtime ensures that the event management platform remains operational and accessible, especially during critical phases like event registration and the event itself.
7. **Data Synchronization Across Devices:**
    - **Requirement**: The system must ensure that data (such as schedules, resources, and participant information) is synchronized across all devices within **5 seconds**.

    - **Rationale**: Users accessing the system from different devices (mobile, desktop, etc.) should see consistent data across platforms, and any updates should reflect immediately to maintain the accuracy of the information.
8. **Email and SMS Notifications:**
    - **Requirement**: Notifications sent via email or SMS should be delivered within **2 minutes** of an action (e.g., registration confirmation, event reminder).
    - **Rationale**: Timely notifications ensure that participants and staff are informed promptly about important events or actions, reducing delays in communication.
9. **Search and Filtering Capabilities:**
    - **Requirement**: The system's search functionality should return results for events, speakers, and participants within **3 seconds**.
    - **Rationale**: Efficient search and filtering functionalities are necessary for users to quickly find relevant information, especially during the event.
10. **Report Generation Time:**
    - **Requirement**: Reports (event details, participant feedback, resource usage) should be generated within **30 seconds** of the request.
    - **Rationale**: Fast report generation helps event organizers access crucial data quickly to make timely decisions or adjustments.

---

## Timing Relationships for Real-Time Systems

- **Real-Time Alerts and Updates**: Notifications regarding event changes (schedule, venue, etc.) should be pushed immediately when the update occurs, with a maximum delay of **1 second**.
- **Real-Time Event Query Handling**: User queries for event status or availability should be processed in real time, with a system response time of **less than 2 seconds** for each query.

## 3.2 Performance Requirements

**Data Security**: Encrypt sensitive data using industry standards
**User Authentication**: Implement strong authentication for admin users.
**Backup & Recovery**: Regular automated backups with secure storage for recovery in case of failure.
**Error Handling**: Gracefully handle errors without exposing sensitive information, logging and notifying admins as needed.
**Access Control**: Limit access to sensitive features based on user roles.
**System Redundancy**: Ensure critical components have redundancy to minimize downtime.
**Compliance**: Adhere to data privacy regulations.
**Data Retention**: Anonymize or delete personal data after it is no longer needed.

**Incident Reporting**: Provide mechanisms for reporting safety incidents

## 3.3 Security Requirements

**User Authentication**: All users must log in with a secure password. Admins should use two-factor authentication.
**Data Protection**: Encrypt sensitive data like passwords, payment info, and personal details both during transmission and while stored.
**Access Control**: Restrict access to sensitive data and features based on user roles.
**Data Privacy**: Follow privacy laws like GDPR and CCPA to protect personal information. Users must have control over their data, including the ability to delete it.
**Regular Security Updates**: The system should be regularly updated to fix known security vulnerabilities.
**Incident Response**: Have a clear plan in place for responding to data breaches or security threats, including user notifications.
**Security Certifications**: Comply with relevant security standards and certifications
**Secure Backups**: Ensure all backups are encrypted and stored securely to prevent data loss or theft.

## 3.4 Software Quality Attributes

**Usability**: The system should be easy to use, with a user-friendly interface. Users should be able to complete key tasks (e.g., registration, booking) in no more than 3 steps.
**Reliability**: The system should have an uptime of 99.9% or higher, ensuring minimal downtime. Critical errors should be rare and logged for future fixes.
**Maintainability**: The codebase should be modular and well-documented, allowing developers to fix bugs or add new features with minimal effort.
**Scalability**: The system should support at least 1,000 concurrent users without significant performance degradation.
**Portability**: The system should be able to run on both Windows and macOS platforms without major changes.
**Security**: The system should protect user data with encryption and secure authentication, adhering to relevant security standards.
**Interoperability**: The system should work seamlessly with external systems (e.g., payment gateways, email services) using industry-standard protocols.
**Adaptability**: The system should allow easy configuration changes without altering the core functionality.
**Testability**: The system should have unit tests for at least 80% of the codebase, allowing for efficient bug detection and fixing.
**Flexibility**: The system should easily adapt to future changes, such as adding new features or integrations, without major rewrites.

## 3.5 Business Rules

**User Roles**:

- **Admin**: Manages events, user registrations, and system settings.
- **Participant**: Registers for events, views schedules, and receives updates.
- **Speaker**: Manages session details, schedules, and interacts with participants.
- **Sponsor**: Views event analytics and provides sponsorship details.

**Registration**:

- Participants, speakers, and sponsors must register before attending.
- Registration is confirmed once payment or approval is processed.

**Scheduling**:

- Event schedules are fixed after approval and cannot be changed without admin consent.
- Speakers and sessions are assigned to specific time slots.

**Payments**:

- Payments for event participation must be processed before confirmation.
- Sponsors are invoiced for agreed amounts based on their sponsorship packages.

**Event Updates**:

- Participants and speakers receive real-time updates and notifications on event changes.

## 3.6 Operating Environment

**Hardware**:

- **For administrators and staff**:
  - **Processor**: 1.5 GHz dual-core or higher
  - **RAM**: 4 GB minimum
  - **Storage**: 500 MB for application and event data
- **For participants**:
  - Mobile devices (smartphones or tablets) and computers with internet access to interact with the system.

**Operating System**:

- **Web-based version**:
  - **Windows** 10 or higher
  - **macOS** 10.12 or higher
  - **Linux** (latest distribution)
- **Mobile version**:
  - **Android** version 8.0 or higher
  - **iOS** version 12 or higher

**Software Components**:

- **Frontend**: Web-based UI using HTML5, CSS3, JavaScript
- **Backend**: Java
- **Database**: MySQL or oracle for managing event data, schedules, and registrations
- **Web Server**: Apache Tomcat or Nginx

## 3.7 User Interfaces

**Event Organizer Interface:**

- **Dashboard**: View event status, schedules, and tasks.
- **Event Creation & Management**: Create events, manage registrations, and schedule sessions.
- **Notification System**: Alerts for upcoming events and tasks.

**2. Speaker Interface:**

- **Profile Management**: Edit bio and session details.
- **Session Scheduling**: Select time slots for presentations.
- **Notifications**: Alerts for session updates.

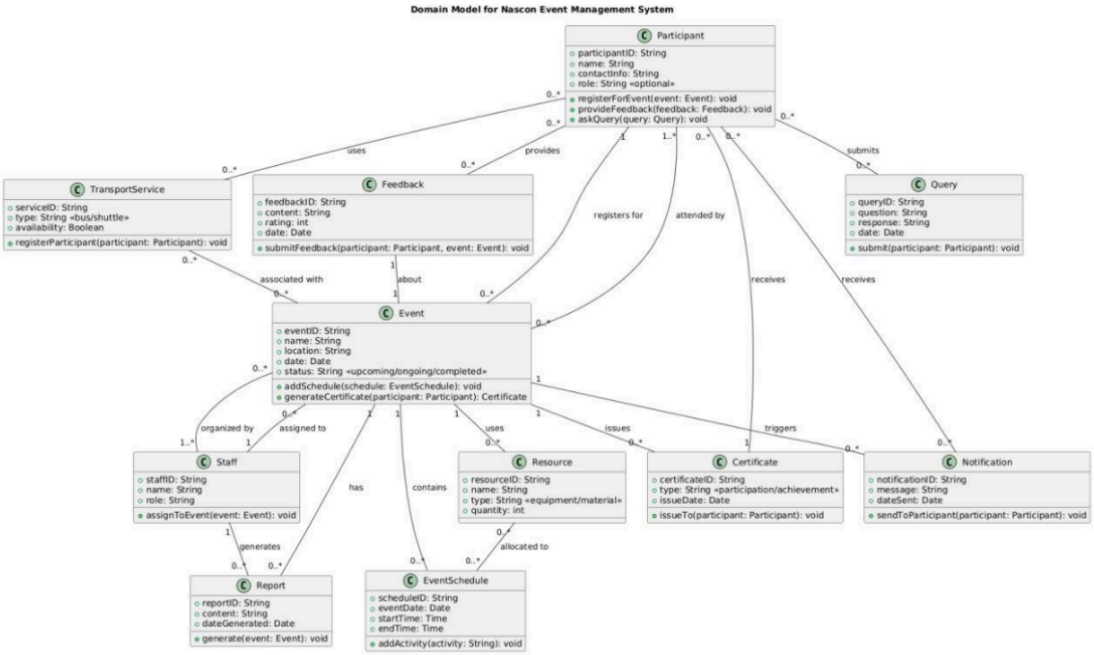
**3. Participant Interface:**

- **Event Registration**: Register for events and sessions.
- **Event Schedule**: Browse event agenda and speaker sessions.
- **Feedback Submission**: Provide feedback on sessions.

**Design Standards:**

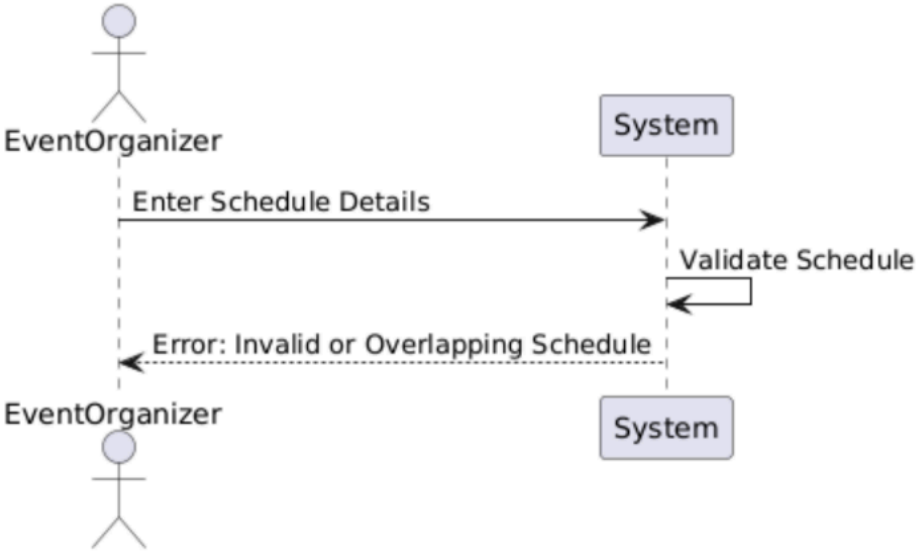
- **Consistency**: Unified color scheme and layout.
- **Responsiveness**: Works on desktop, tablet, and mobile.
- **Accessibility**: High contrast, keyboard navigation, and screen reader support.
- **Feedback**: Clear visual feedback for actions.
- **Navigation**: Simple and intuitive menus with common buttons like Save and Cancel.

# 1. Domain Model

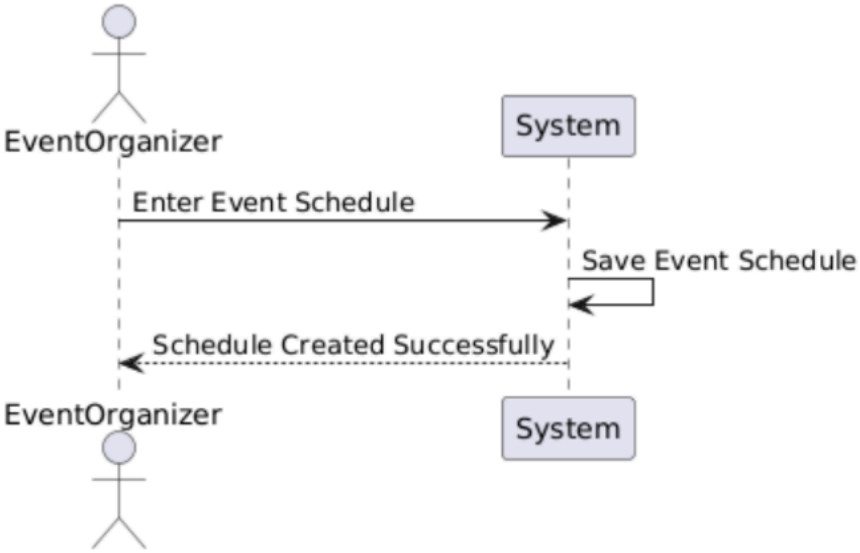



# 2. System Sequence Diagram

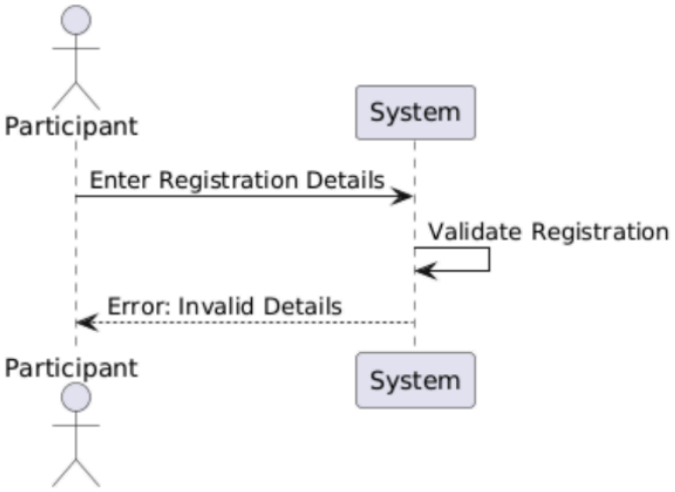
**Create Event Schedule - AS: Invalid Schedule Details**



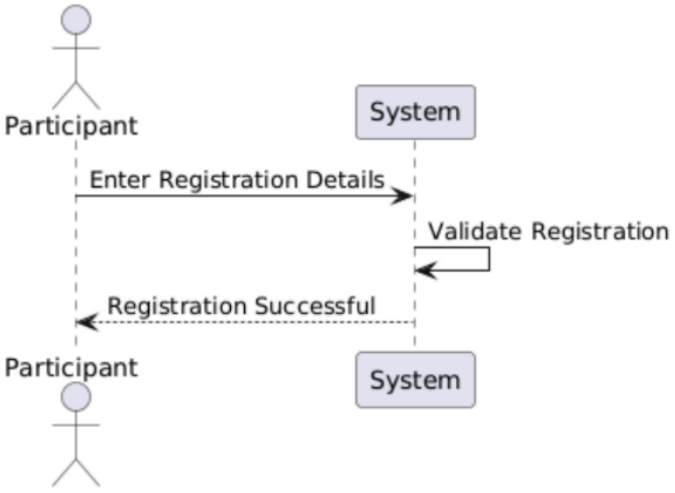

**Create Event Schedule - MSS**



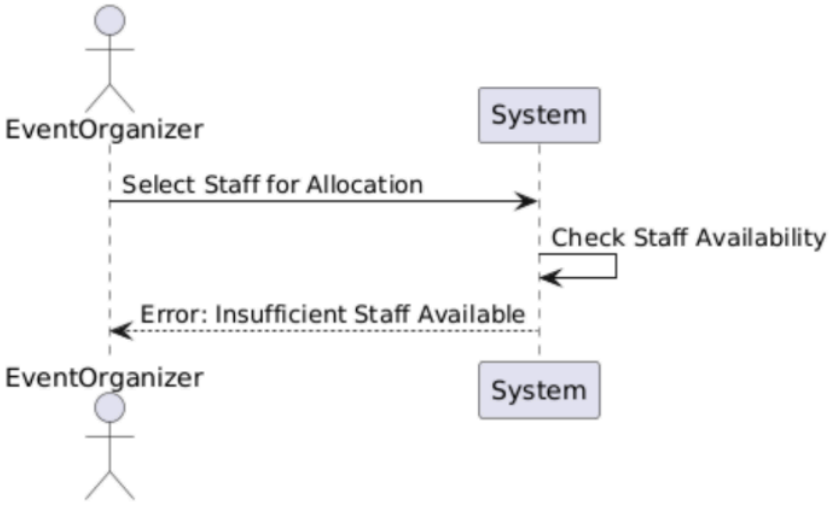
**Register for Event - AS**



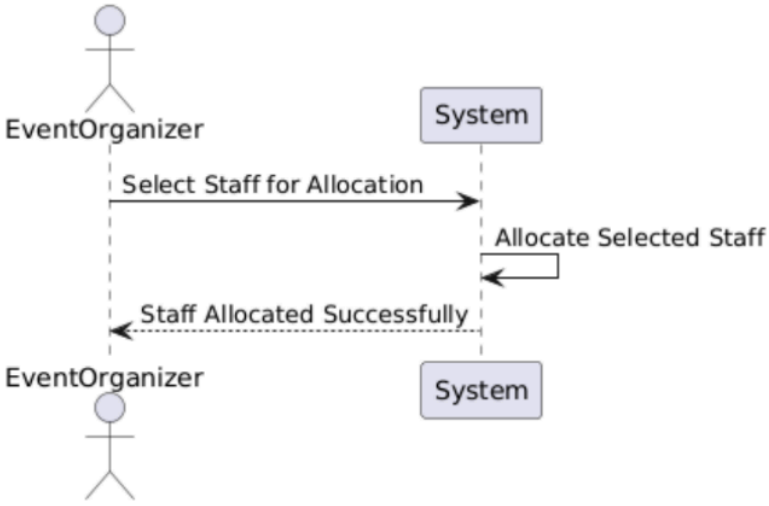

**Register for Event - MSS**



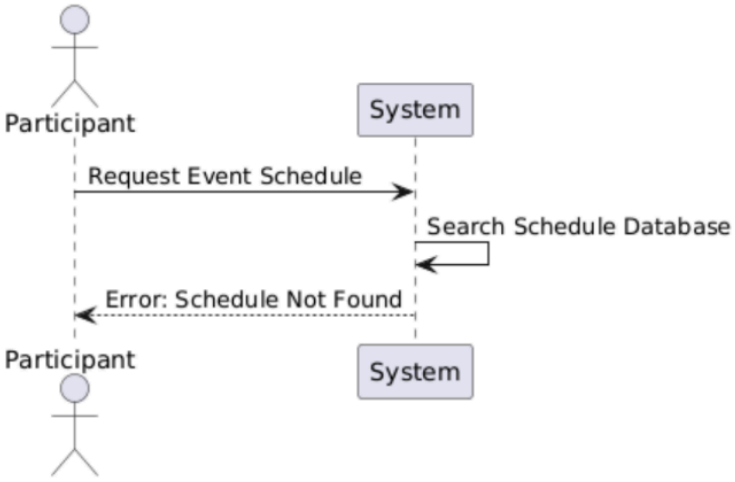
**Allocate Staff - AS: Insufficient Staff**



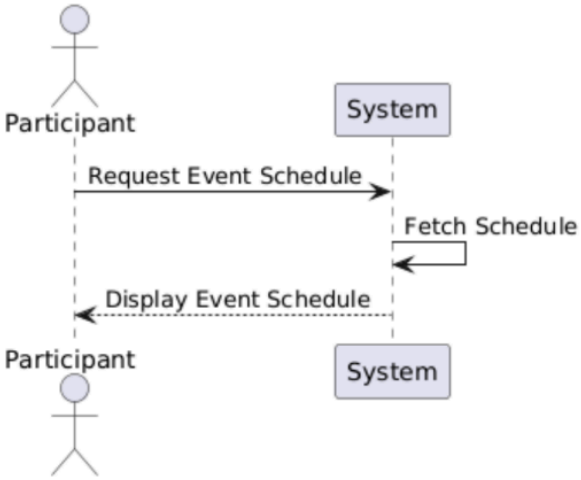

**Allocate Staff - MSS**



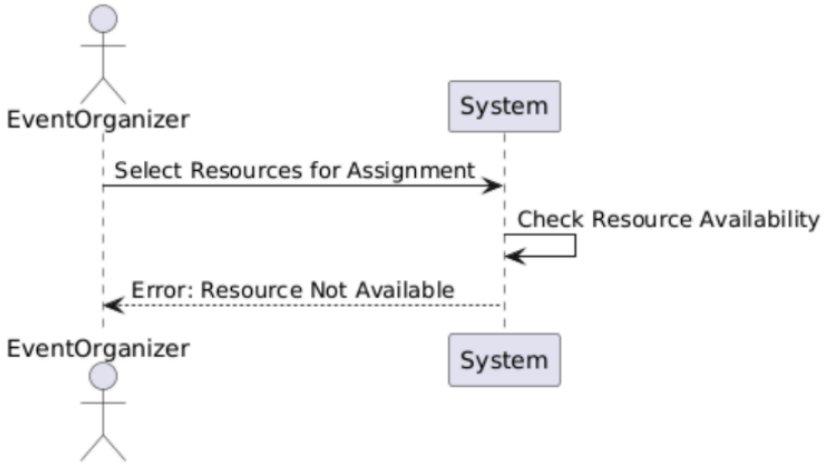
**View Event Schedule - AS: Schedule Not Available**



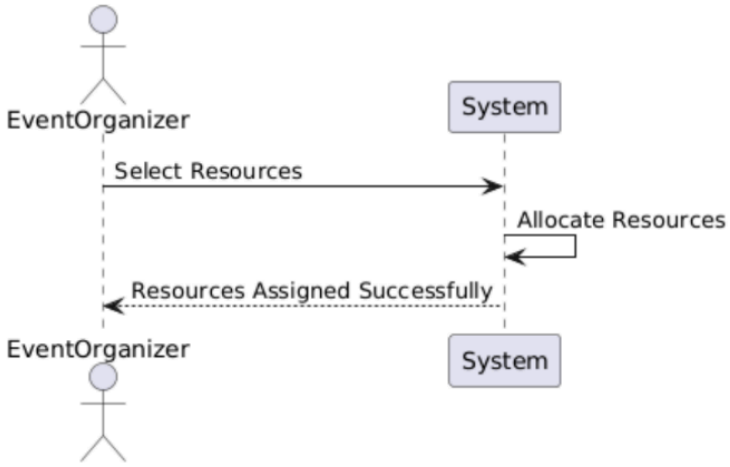

**View Event Schedule - MSS**



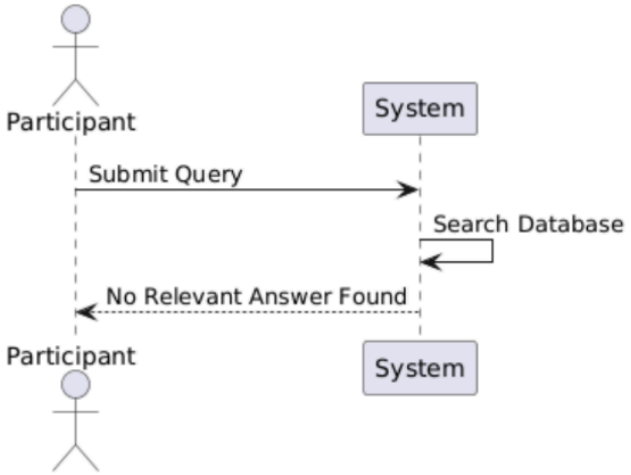
**Assign Resources - AS: Resource Allocation Failed**



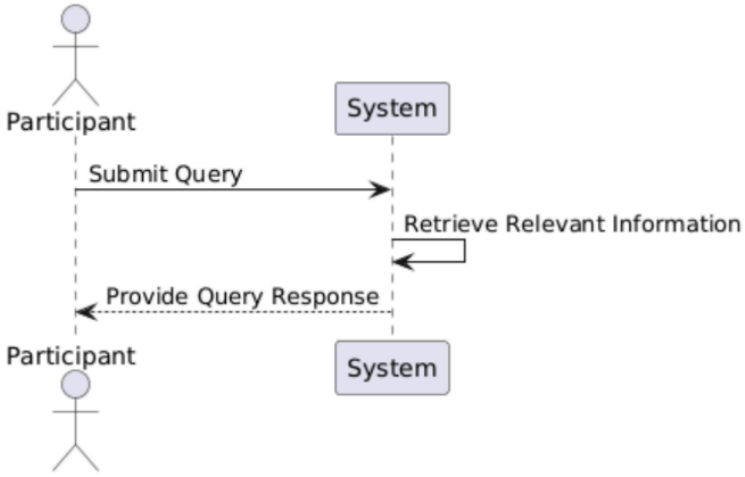

**Assign Resources - MSS**



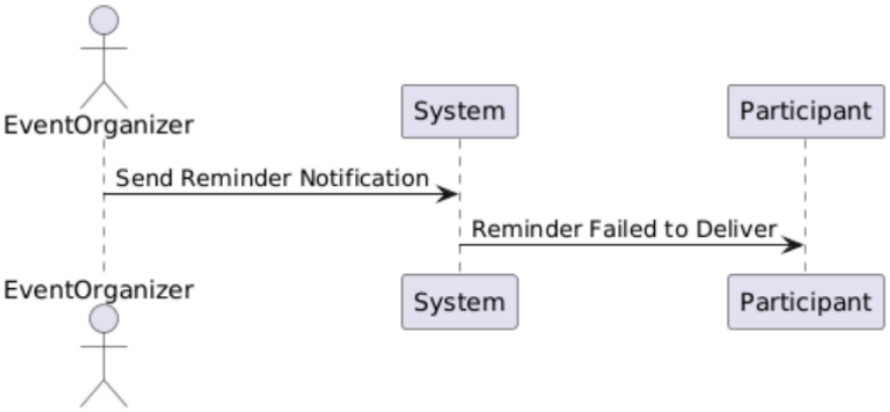
**Ask Event Query - AS**



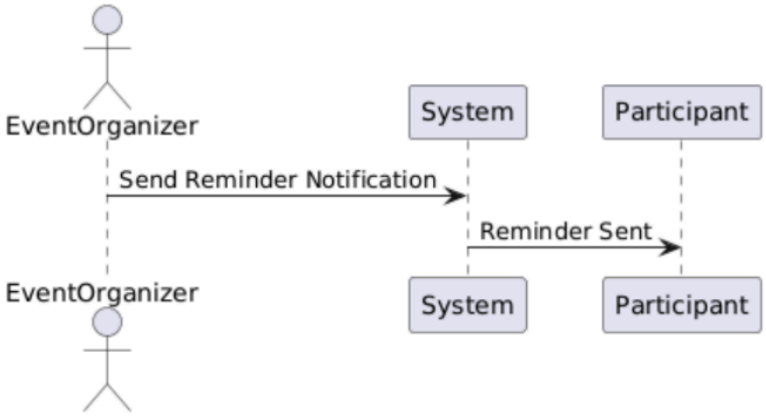

**Ask Event Query - MSS**



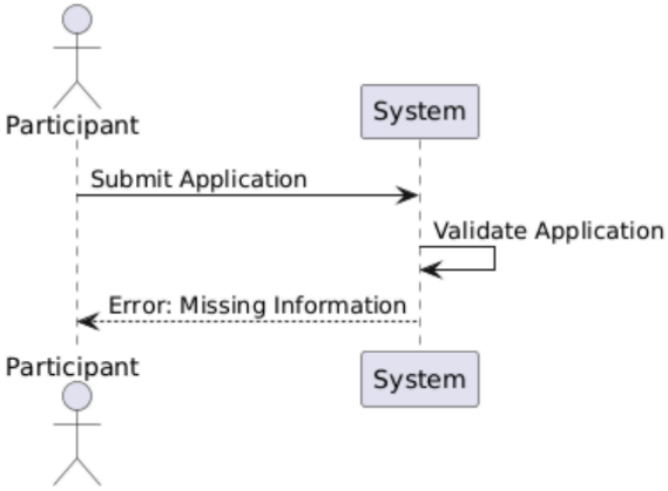
**Send Event Reminders - AS: Reminder Failure**



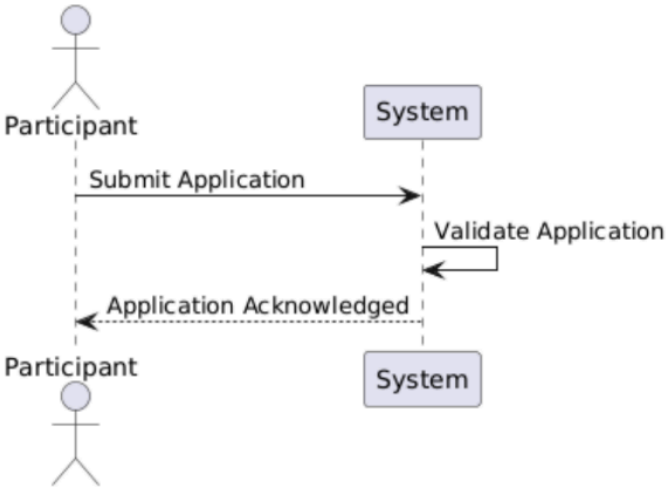

**Send Event Reminders - MSS**



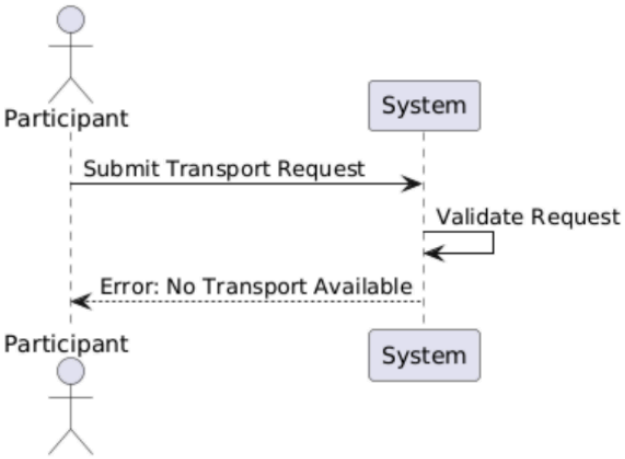

**Register for Staff/Volunteer - AS**



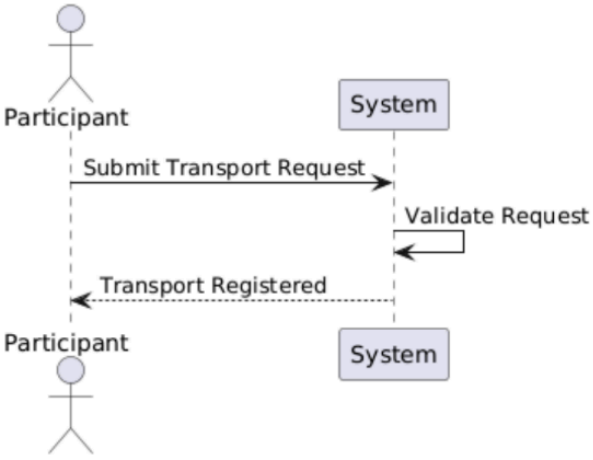
**Register for Staff/Volunteer - MSS**



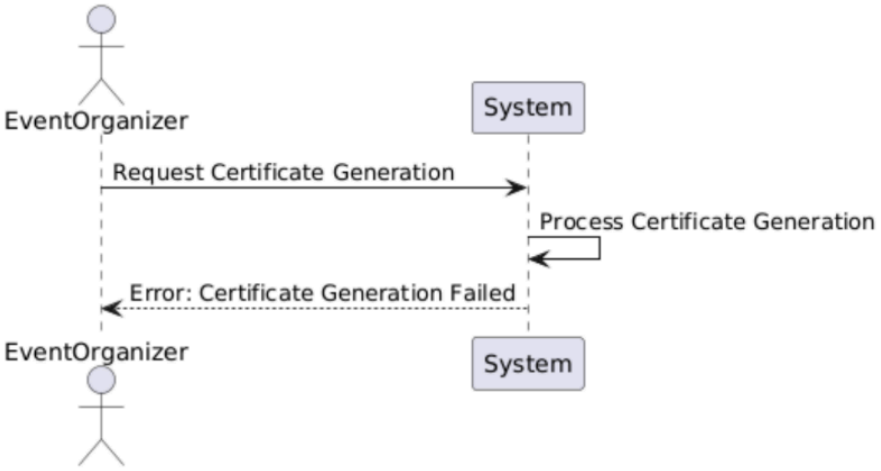
**Register for Transport Services - AS**



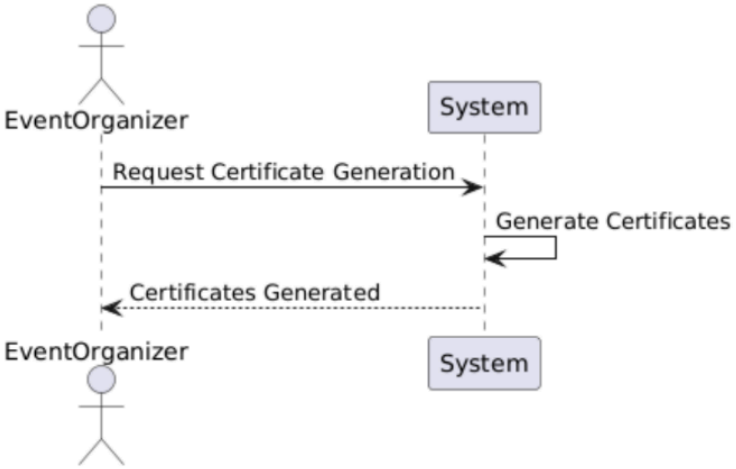

**Register for Transport Services - MSS**



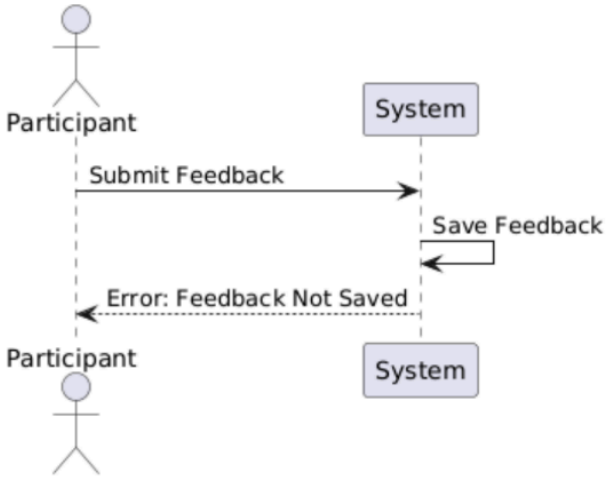
**Generate Certificates - AS: Certificate Generation Failed**



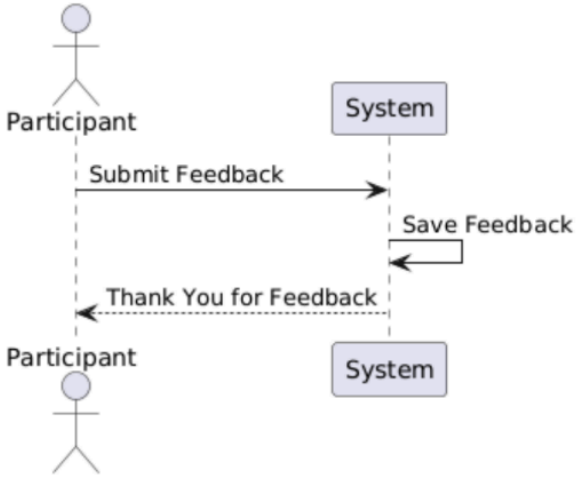

**Generate Certificates - MSS**



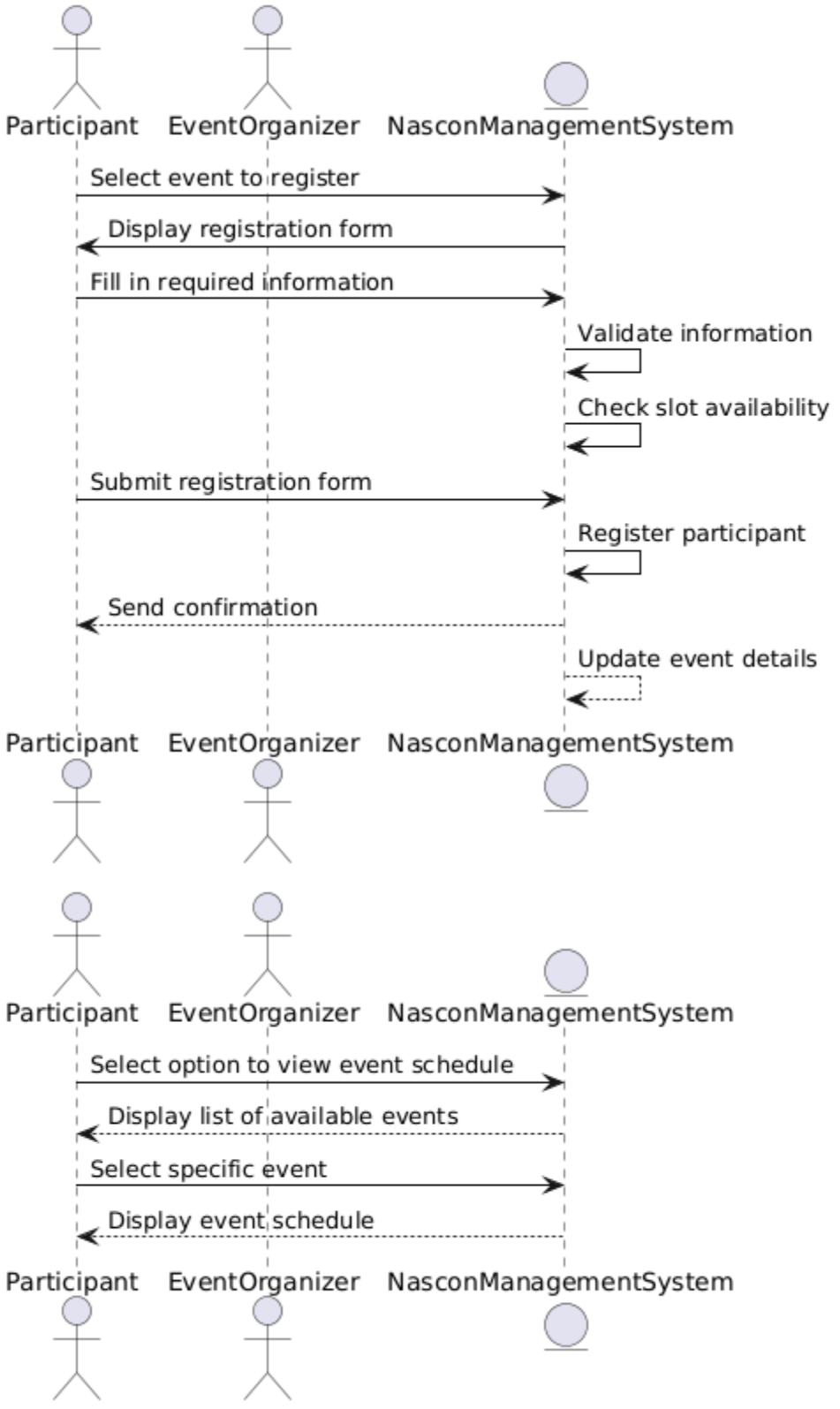
**Provide Feedback - AS**

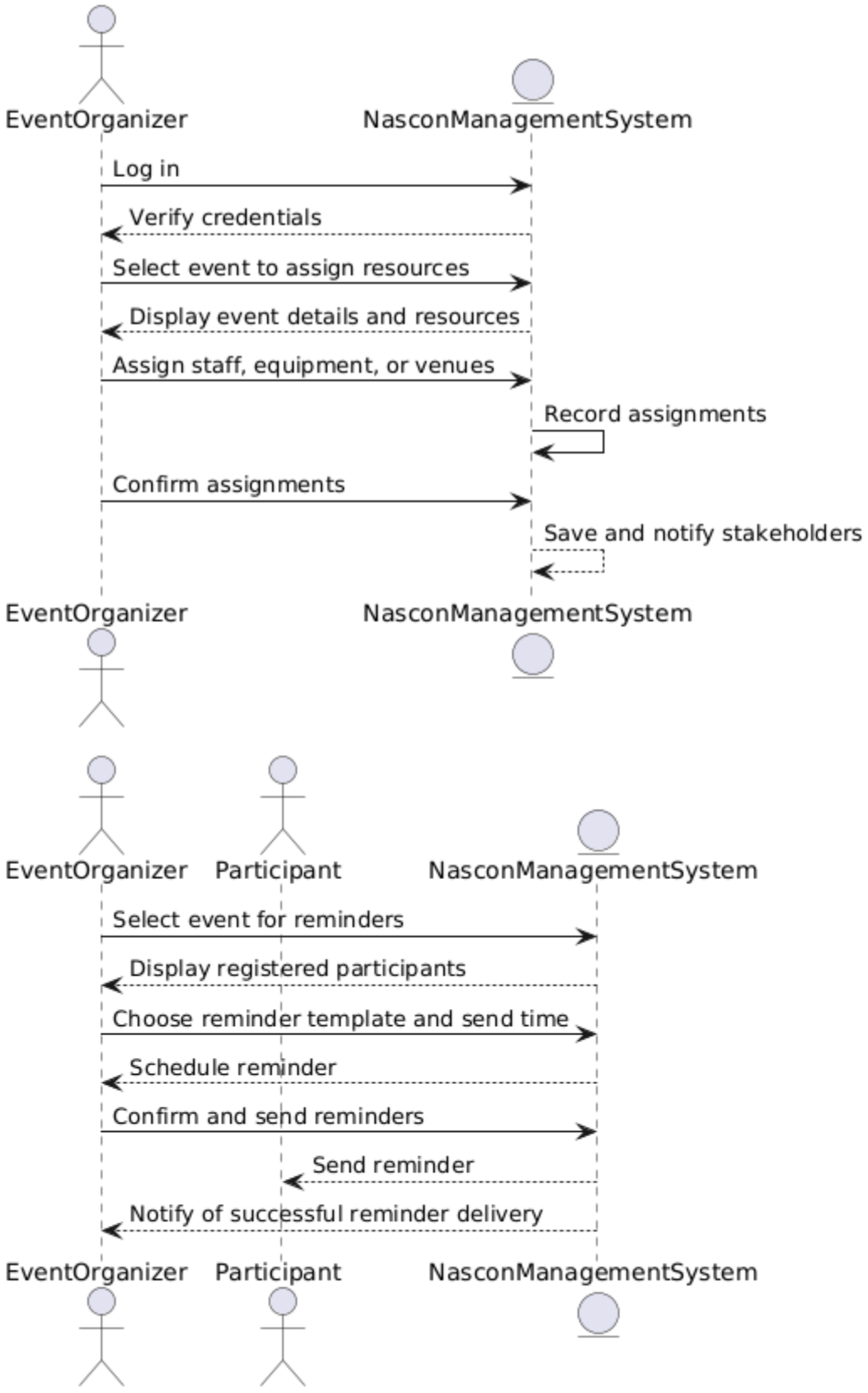


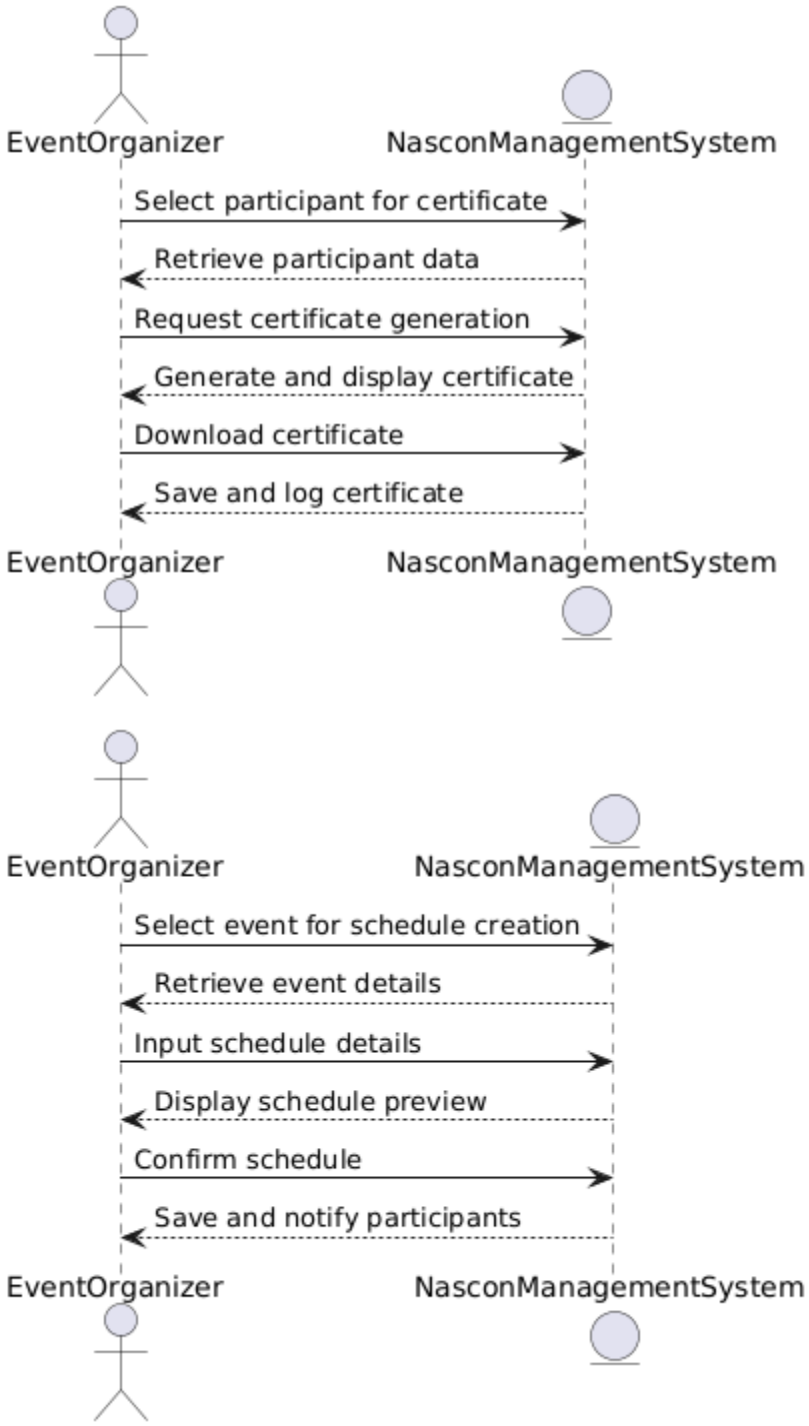

**Provide Feedback - MSS**

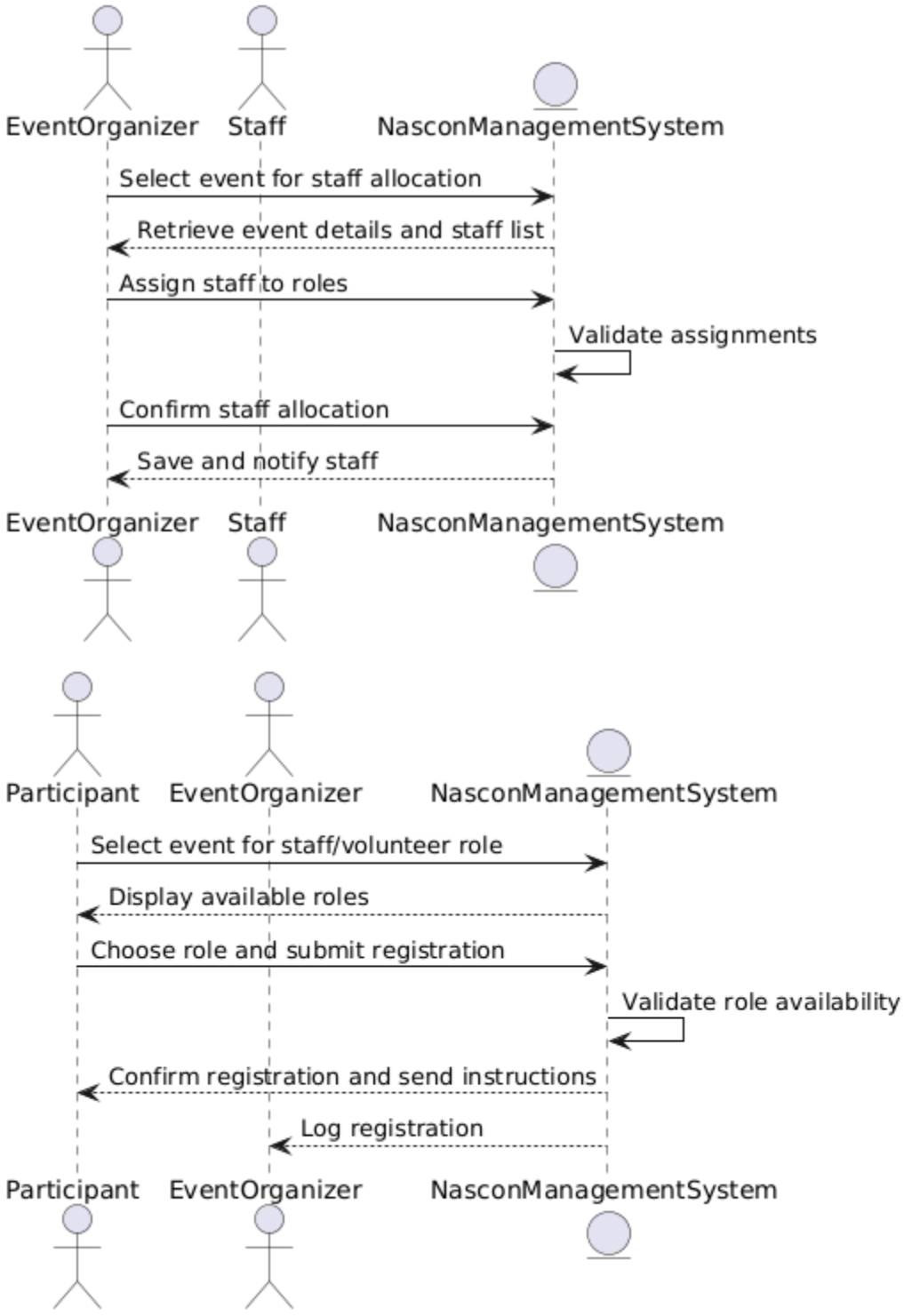


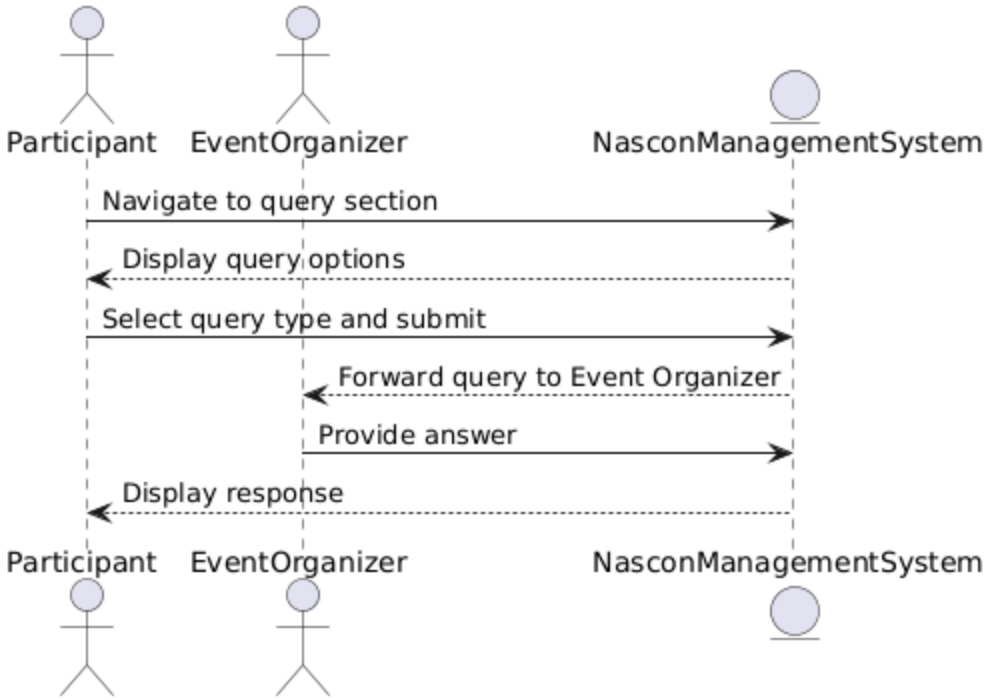
# 3. Sequence Diagram

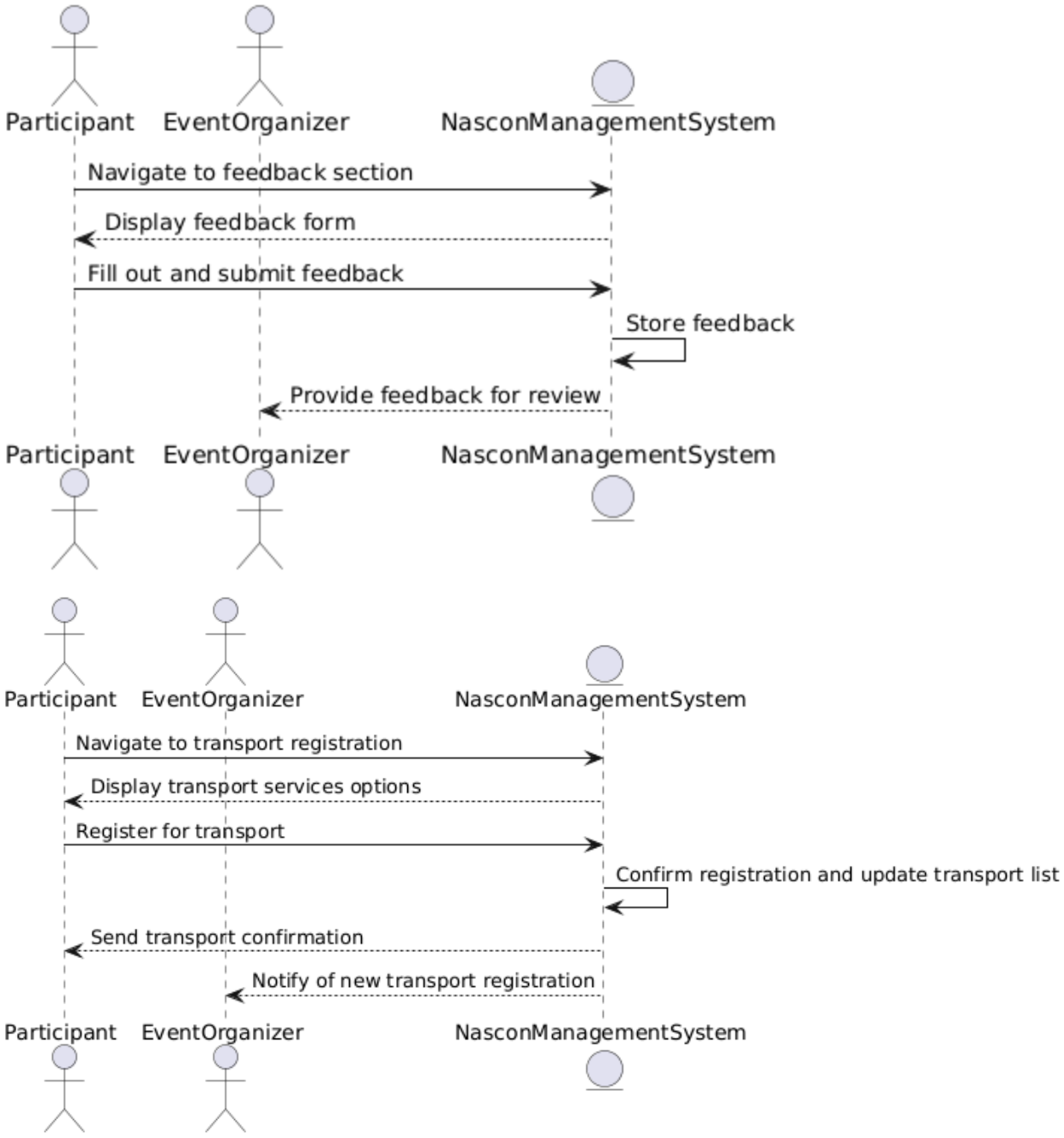



EventOrganizer
NasconManagementSystem
Log in
Verify credentials
Select event to assign resources
Display event details and resources
Assign staff, equipment, or venues
Record assignments
Confirm assignments
Save and notify stakeholders
EventOrganizer
NasconManagementSystem
EventOrganizer
Participant
NasconManagementSystem
Select event for reminders
Display registered participants
Choose reminder template and send time
Schedule reminder
Confirm and send reminders
Send reminder
Notify of successful reminder delivery
EventOrganizer
Participant
NasconManagementSystem

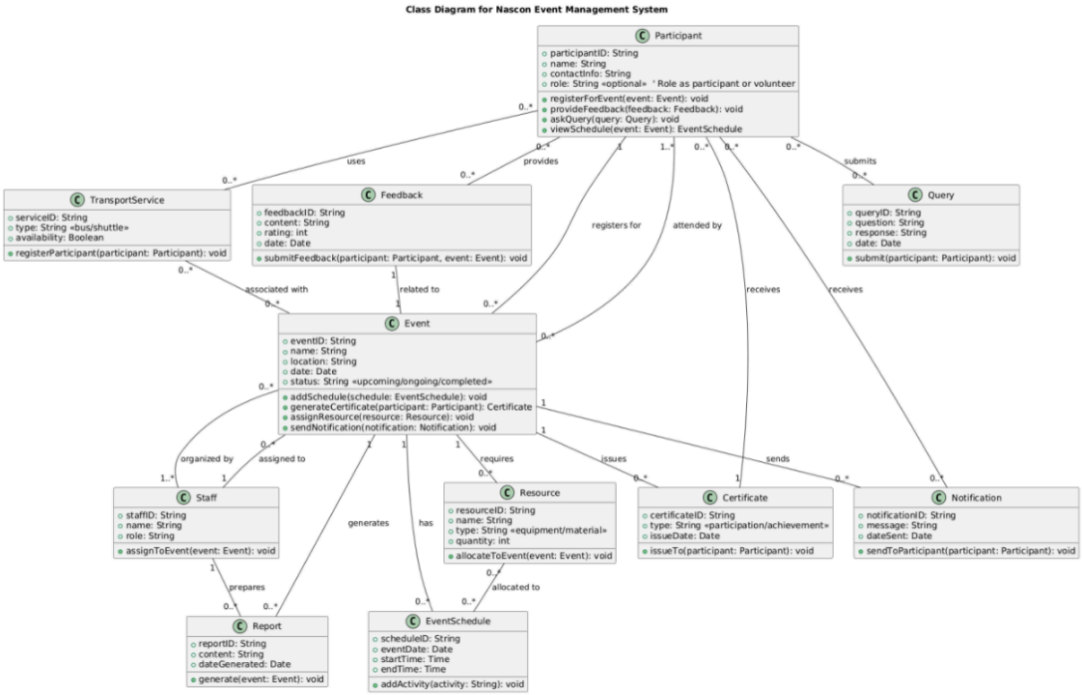

EventOrganizer
NasconManagementSystem
Select participant for certificate
Retrieve participant data
Request certificate generation
Generate and display certificate
Download certificate
Save and log certificate
EventOrganizer
NasconManagementSystem
EventOrganizer
NasconManagementSystem
Select event for schedule creation
Retrieve event details
Input schedule details
Display schedule preview
Confirm schedule
Save and notify participants
EventOrganizer
NasconManagementSystem


EventOrganizer
Staff
NasconManagementSystem
Select event for staff allocation
Retrieve event details and staff list
Assign staff to roles
Validate assignments
Confirm staff allocation
Save and notify staff
EventOrganizer
Staff
NasconManagementSystem
Participant
EventOrganizer
NasconManagementSystem
Select event for staff/volunteer role
Display available roles
Choose role and submit registration
Validate role availability
Confirm registration and send instructions
Log registration
Participant
EventOrganizer
NasconManagementSystem


Participant
EventOrganizer
NasconManagementSystem
Navigate to query section
Display query options
Select query type and submit
Forward query to Event Organizer
Provide answer
Display response
Participant
EventOrganizer
NasconManagementSystem

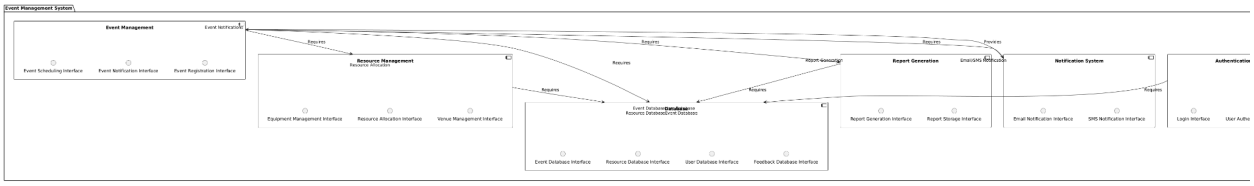

Participant
EventOrganizer
NasconManagementSystem
Navigate to feedback section
Display feedback form
Fill out and submit feedback
Store feedback
Provide feedback for review
Participant
EventOrganizer
NasconManagementSystem
Participant
EventOrganizer
NasconManagementSystem
Navigate to transport registration
Display transport services options
Register for transport
Confirm registration and update transport list
Send transport confirmation
Notify of new transport registration
Participant
EventOrganizer
NasconManagementSystem

# 4. Class Diagram

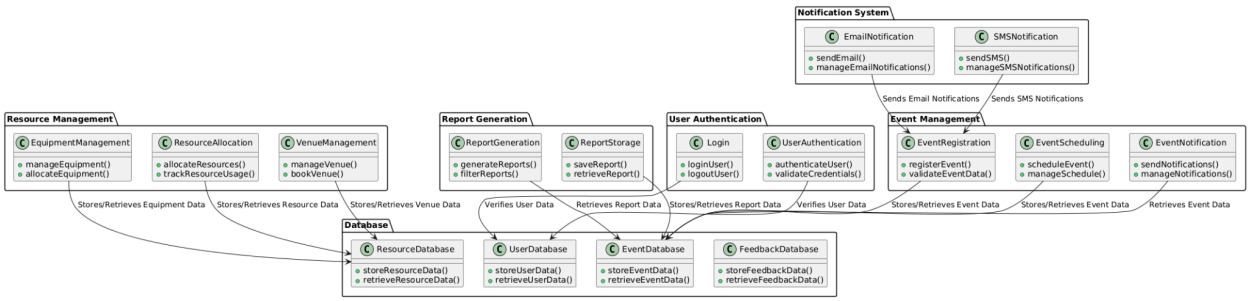



5.

# 6. Component Diagram




# 7. Package Diagram



# 8. Deployment Diagram