
Software Requirements Specification

for

SportSync

Ahmad Rashad 22I-1175

Muhammad Salman Javed 22I-1298

Haseeb Sultan 22I-0874

CS-B

Table of Contents

1. Introduction.....	1
1.1 Purpose.....	1
1.2 Project Scope.....	1
2. Overall Description.....	1
2.1 Objective.....	1
2.2 Problem Statement and Description.....	1
2.3 Use Cases.....	1
2.4 Extended Use Cases.....	2
3. System Sequence Diagrams.....	10
4. Sequence Diagram.....	19
5. Component Diagram.....	24
6. Class Diagram.....	25
7. Deployment Diagram.....	26

1. Introduction

1.1 Purpose

SportSync offers an app that makes it simple for athletes to join or request sports events nearby.

1.2 Project Scope

Athletes and sports fans who want to plan or take part in informal indoor or outdoor sporting events can use SportSync. While there are currently few sports venues booking services available, none of them combine team formation, site time voting and game scheduling into a single, cohesive app.

2. Overall Description

2.1 Objective

- *To provide a smooth platform that makes it simple for athletes to plan, schedule, or participate in sports.*
- *To encourage community involvement by incorporating voting and team-building elements on a digital platform.*
- *A digital platform is being created to facilitate the organization of tournaments and friendly competitions.*

2.2 Problem Statement and Description

Whether indoors or outdoors, planning informal sporting events can be difficult. Delays, cancellations, or incomplete teams may result from the process of locating players, verifying their availability, and arranging a venue. SportSync offers an app that makes it simple for athletes to join or request sports events nearby. The software makes sure that games are well-organized by giving teams or individuals the ability to complete slots and cast votes for the final venue and time. It simplifies the online venue reservation process, making it as simple as scheduling a movie. Due to the growing popularity of social sports and fitness, as well as the growing use of smartphones and online booking, the likelihood of success is strong. SportSync can fill a void in the market for sports fans with user engagement features and strategic relationships with sports facilities.

2.3 Use Cases

- *Create a Sports Event*
- *Join a Sports Event*
- *Vote on Event Venue and Time*
- *Search for Events by Location*
- *Invite Friends to Events*

- *Create or Join a Team*
- *View Event Details*
- *Track Event Participation*
- *Book a Venue*
- *View Venue Availability*
- *Receive Notifications for Events*
- *Rate an Event or Venue*

2.4 Extended Use Cases

Use Case 1: Register a User

Actors: User (Host or Participant), System

Description: A user registers as either a host or a participant by providing necessary details.

Preconditions:

- The user is not already registered.

Triggers:

- User clicks "Register" on the registration page.

Main Success Scenario:

User	System
The system displays a registration form with options to select "Host" or "Participant."	
The user selects either "Host" or "Participant" role.	
The user fills in the required details (name, email, password, etc.).	
The user submits the form.	
	The system validates the information and creates a user account with the selected role (Host or Participant).

Confirmation is shown, and the user is redirected to the login page or dashboard.	
---	--

Postconditions:

- User account is created, and the user can log in with the appropriate role (Host or Participant).

Use Case 2: Login as Host

- **Actors:**
 - Host (primary), System
- **Description:**
 - The host logs into the system to access their dashboard.
- **Preconditions:**
 - Host must be registered in the system.
- **Triggers:**
 - Host enters their login credentials (email/username and password) and clicks the "Login" button.

Main Flow:

User	System
Enters login credentials and submits the form.	
	<p>Validates the credentials.</p> <ul style="list-style-type: none">○ If valid:<ul style="list-style-type: none">▪ System grants access and redirects to the host dashboard.○ If invalid:<ul style="list-style-type: none">▪ System displays an error message ("Invalid username or password").

Postconditions:

- **Successful Login:**
 - Host is logged into the system and redirected to the dashboard.
- **Failed Login:**
 - Host remains on the login page with an error message.

Use Case 3: Create Event (Continued)

- **Triggers:**
 - Host clicks the "Create Event" button on the dashboard.
- **Main Flow:**

User	System
Host is redirected to the venue booking page.	
Host selects a venue and confirms the booking.	
	System redirects the host to the main event creation form.
Host enters event details (name, description, date, time).	
	System saves the main event.
Host is redirected to the sub-event creation page.	
Host adds sub-events (name, description, participants).	
Host submits the event setup.	

	System confirms the event creation and shows a success message.
--	---

- **Postconditions:**

- 0. The event, along with its sub-events, is created and stored in the system.

Use Case 4: Book Venue

- **Actors:** Host (primary), System
- **Description:** The host books a venue for personal use or event creation.
- **Preconditions:**
 - Host must be logged in.
- **Triggers:**
 - Host clicks the "Book Venue" button on the dashboard.
- **Main Flow:**

User	System
	System displays available venues.
Host selects a venue and specifies the booking time and date.	
Host confirms the booking.	
	System processes the booking and updates venue availability.
	System shows a booking confirmation message.

- **Postconditions:**

- 0. Venue is booked, and the system marks it as unavailable for the selected time.

Use Case 5: Update Event

- **Actors:** Host (primary), System
- **Description:** The host updates an existing event by modifying sub-events or deleting the event.
- **Preconditions:**
 - Host must have created events.
- **Triggers:**
 - Host clicks the "Update Events" button on the dashboard.
- **Main Flow:**

User	System
	System displays a list of the host's events.
Host selects an event to update.	
Host can: <ul style="list-style-type: none">▪ Add a new sub-event.▪ Delete an existing sub-event.▪ Delete the main event.	
Host confirms the changes.	
	System processes the updates and shows a success message.

- **Postconditions:**
 - 0. The selected event or sub-event is updated in the system.

Use Case 6: View Notifications

- **Actors:** Host (primary), System
- **Description:** The host views notifications related to their events.
- **Preconditions:**
 - Host must be logged in.
- **Triggers:**
 - Host clicks the "Notifications" button on the dashboard.
- **Main Flow:**

User	System
	System retrieves unread notifications for the host.
Notifications are displayed in a list.	
Host can mark notifications as read or delete them.	
	System updates notification status.

- **Postconditions:**
 - 0. Notifications are marked as read or deleted.

Use Case 7: Join an Event

Actors: Participant (primary), System

Description: The participant browses and joins an event or subevent.

Preconditions:

- The participant must be logged in.
- Events and subevents must be available.

Triggers:

- The participant selects the "Join an Event" button from the dashboard.

Main Success Scenario:

User	System
Participant clicks "Join an Event" on the dashboard.	
	System displays a list of available main events.
Participant selects a main event.	
	System displays the subevents for the selected main event.
Participant selects a subevent: <ul style="list-style-type: none">○ If individual: System joins the participant to the subevent.○ If team-based: System navigates to the "Create Team" interface.	
Participant creates a team and registers for the subevent.	
	System confirms the registration.

Extensions:

- **Event Full:** System displays an "Event is full" message.
- **Network Error:** System displays a "Network error, try again" message.

Use Case 8: View My Teams

Actors: Participant (primary), System

Description: The participant views their registered teams and events.

Preconditions:

- The participant must be logged in.

Triggers:

- The participant selects the "My Teams" button from the dashboard.

Main Success Scenario:

User	System
Participant clicks "My Teams" on the dashboard.	
	System retrieves and displays a list of teams the participant has joined.
	System displays events associated with the participant's teams.

Extensions:

- **No Teams:** System displays "You have not joined any teams."

Use Case 9: Rate an Event

Actors: Participant (primary), System

Description: The participant rates an event and provides feedback.

Preconditions:

- The participant must have participated in the event.

Triggers:

- The participant selects the "Rate an Event" button from the dashboard.

Main Success Scenario:

User	System
Participant clicks "Rate an Event" on the dashboard.	
	System displays a list of events the participant has joined.

Participant selects an event to rate.	
Participant submits a rating (1 to 5 stars) and feedback.	
	System stores the rating and feedback.

Extensions:

- **Invalid Input:** System displays "Invalid rating, please try again."

Use Case 10: View Notifications

Actors: Participant (primary), System

Description: The participant views notifications related to events.

Preconditions:

- The participant must be logged in.

Triggers:

- The participant selects the "Notifications" button from the dashboard.

Main Success Scenario:

User	System
Participant clicks "Notifications" on the dashboard.	
	System displays a list of notifications: <ul style="list-style-type: none"> ○ New events available for participation. ○ Updates on ongoing events.

Extensions:

- **No Notifications:** System displays "No new notifications."

Use Case 11: Logout**Actors:** Participant (primary), System**Description:** The participant logs out of the system.**Preconditions:**

- The participant must be logged in.

Triggers:

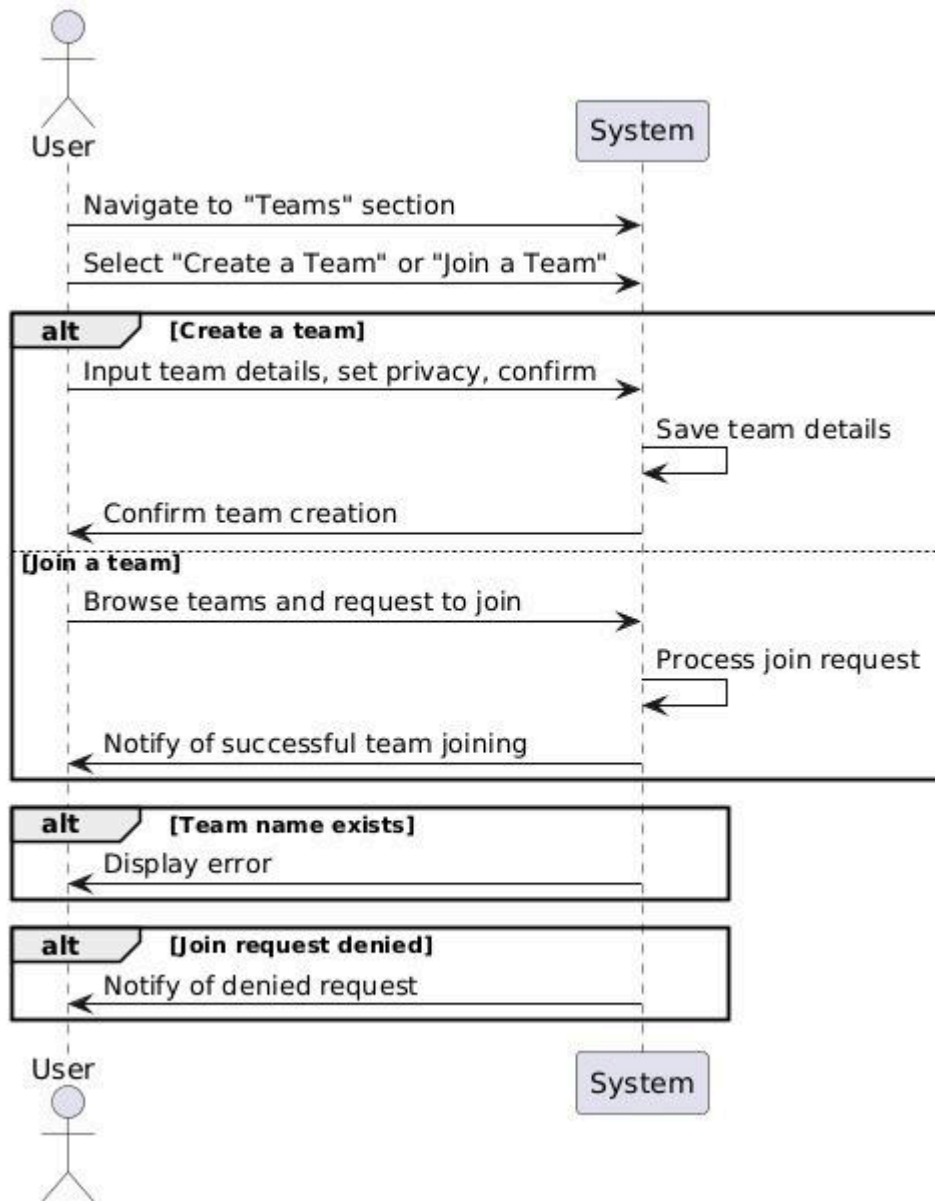
- The participant selects the "Logout" button from the dashboard.

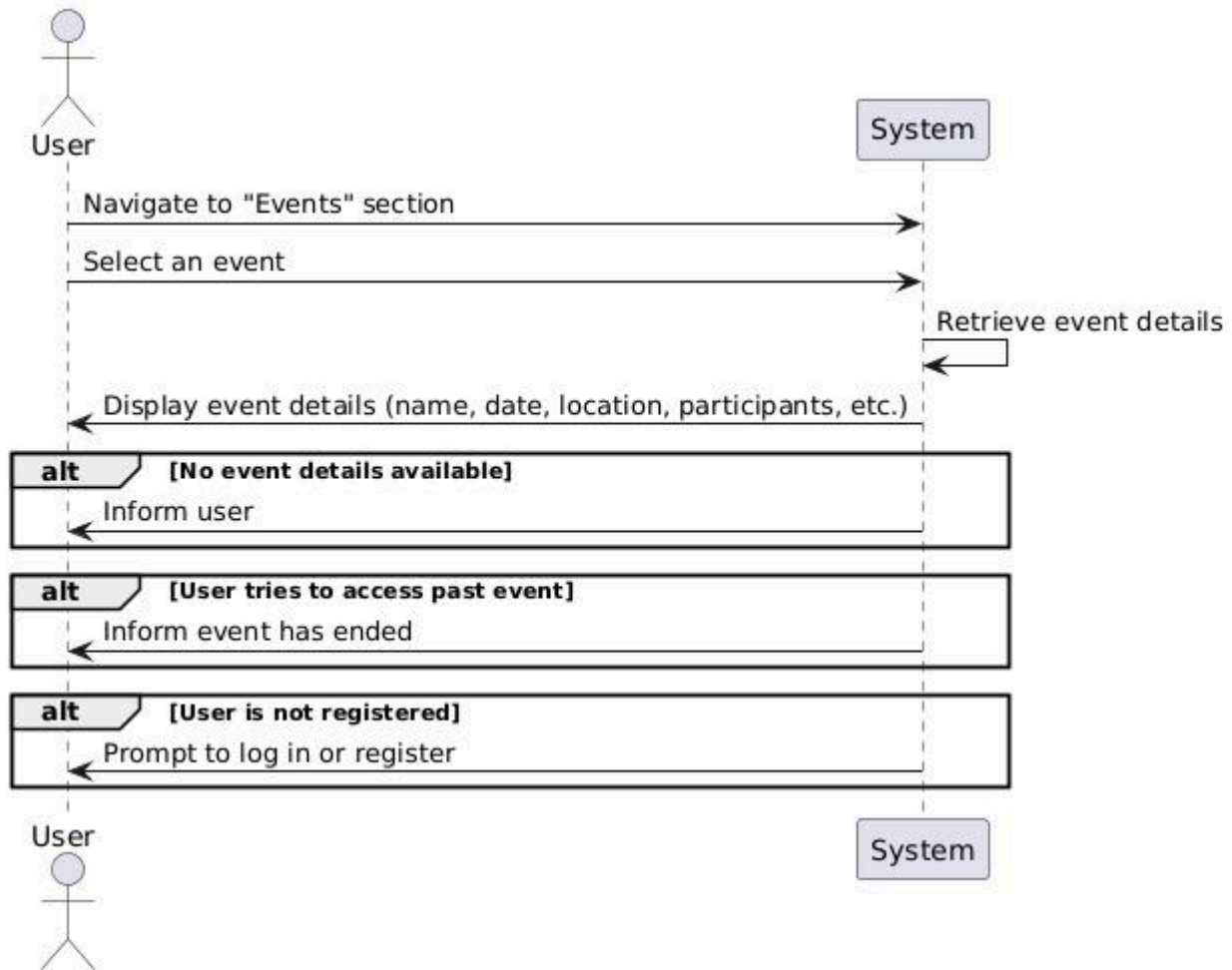
Main Success Scenario:

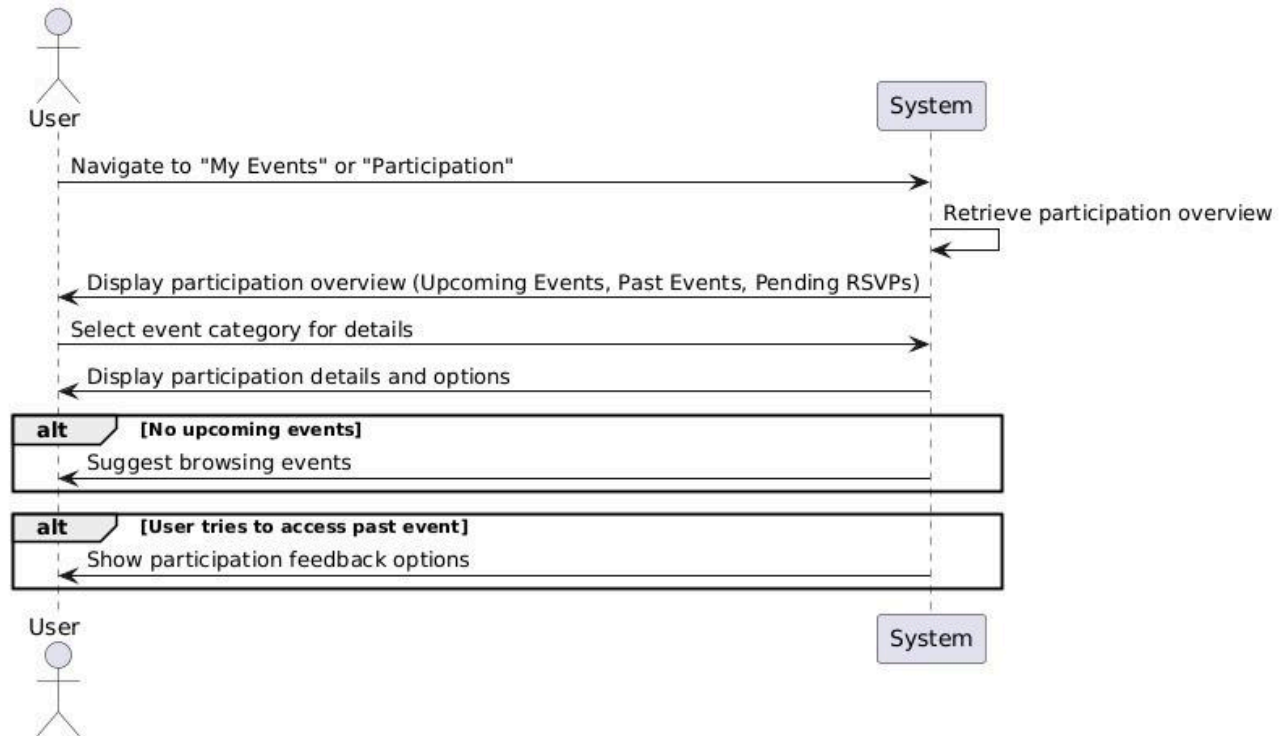
User	System
1. Participant clicks "Logout" on the dashboard.	
	2. System logs out the participant and redirects to the login page.

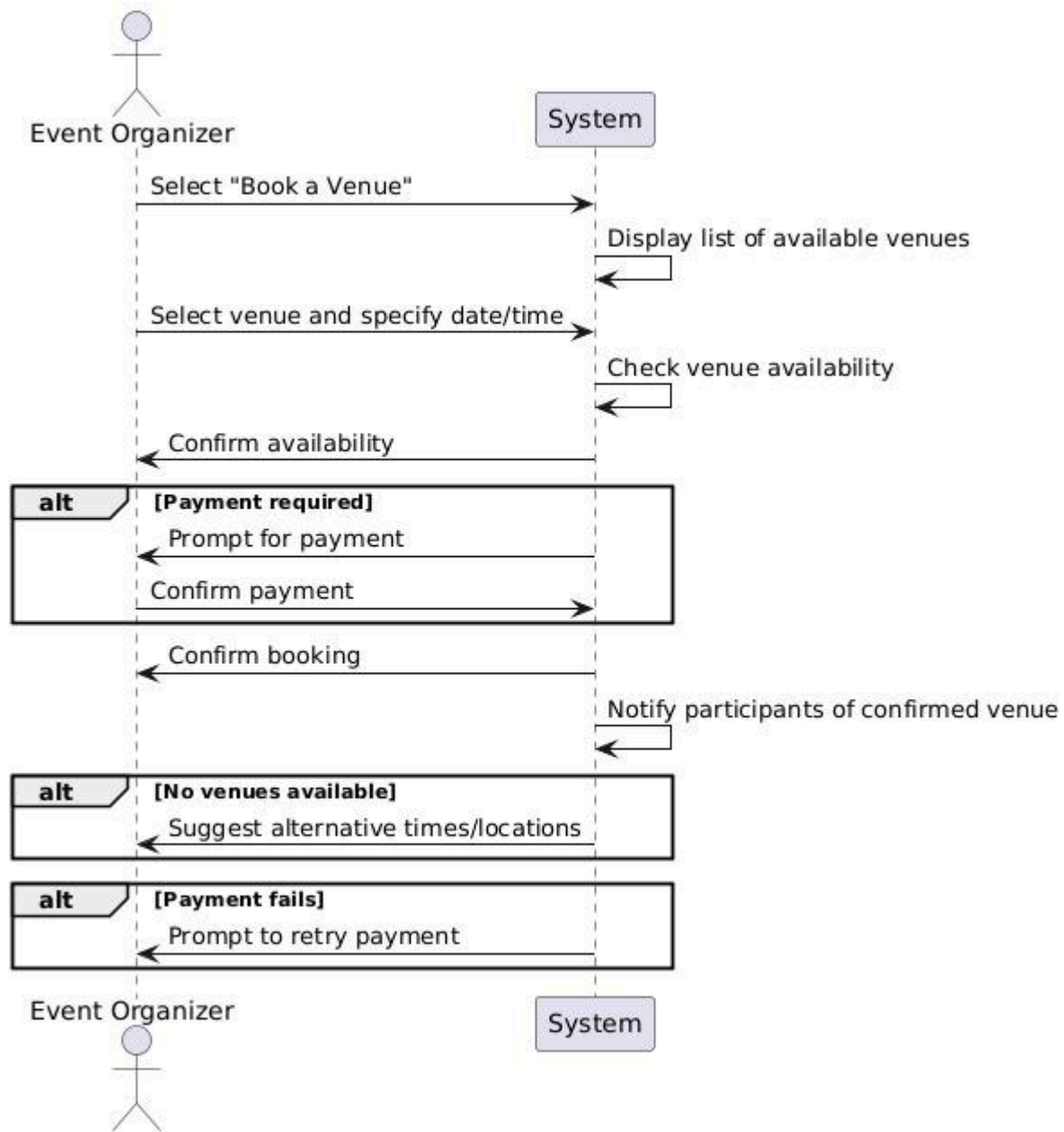
3. System Sequence Diagrams

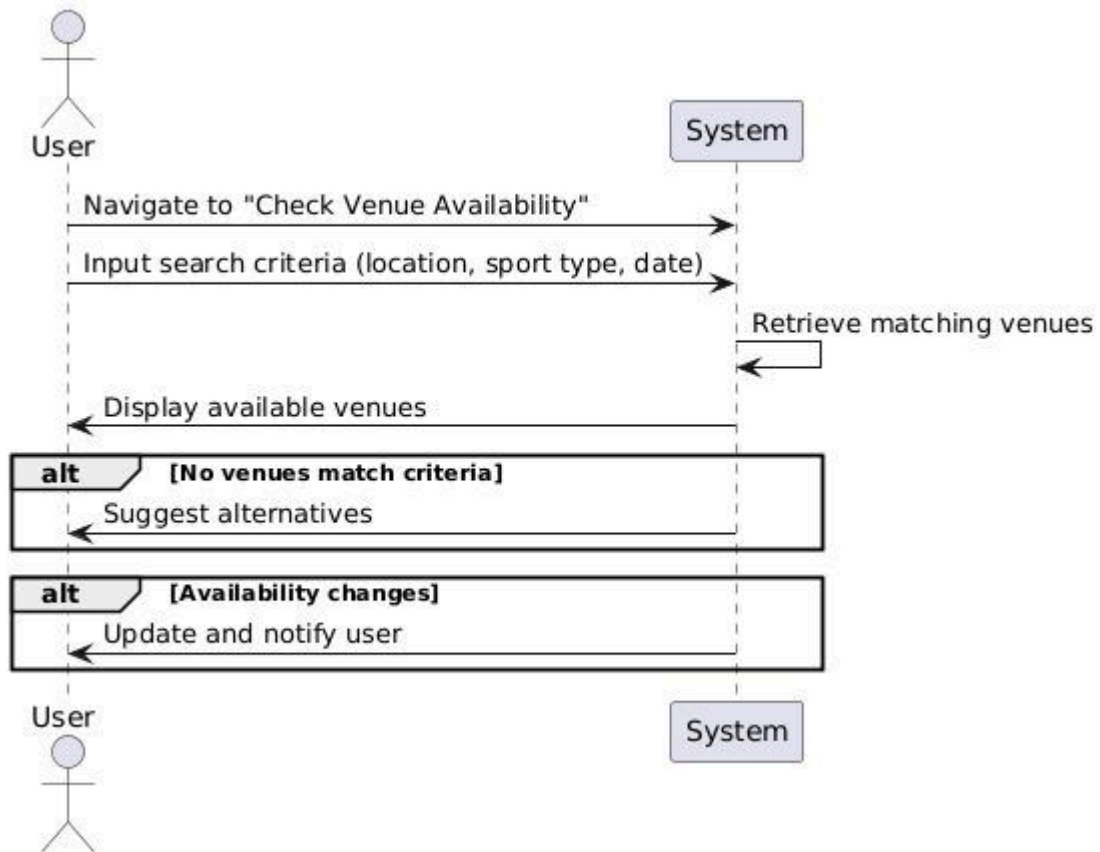
Use Case 1:

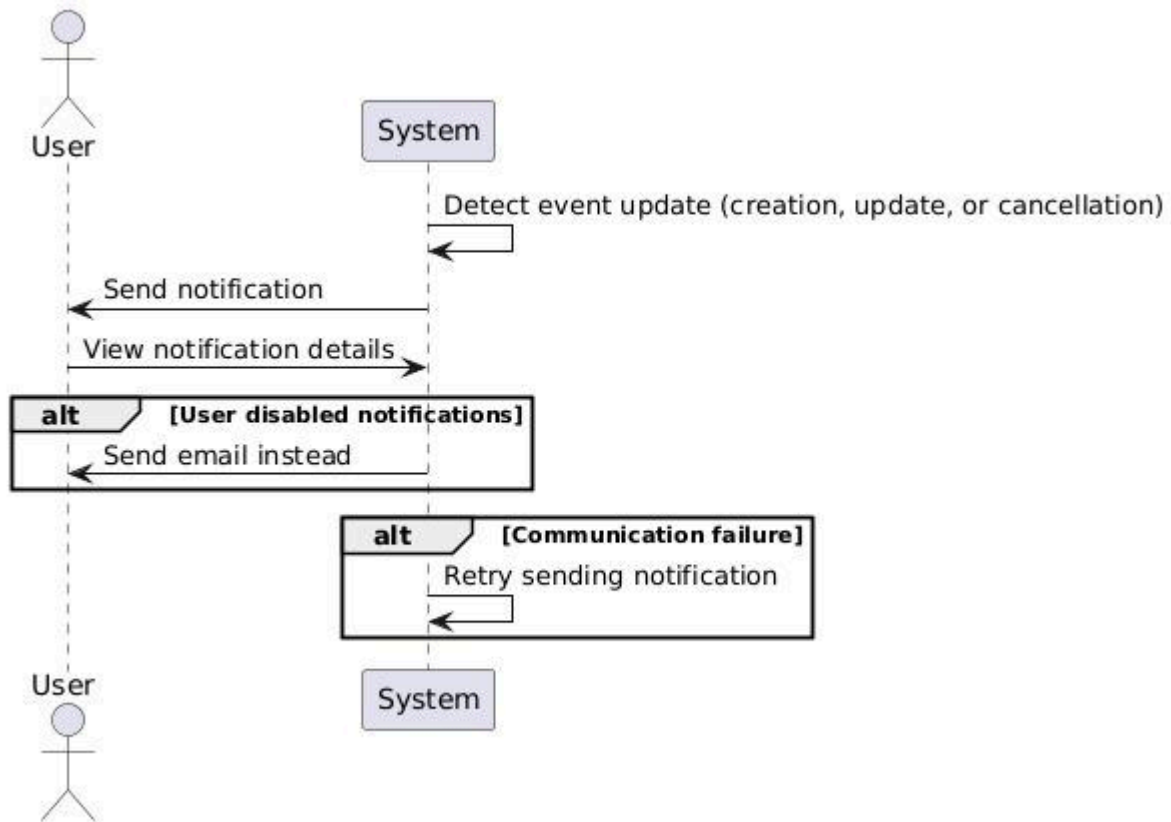


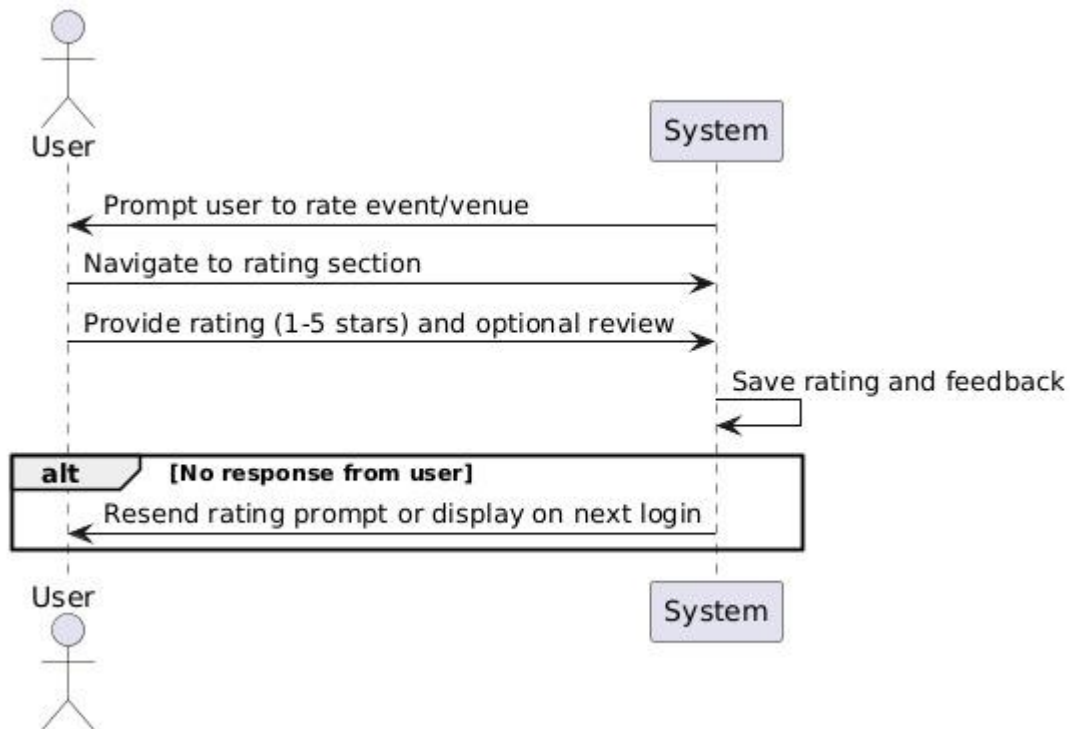
Use Case 2:

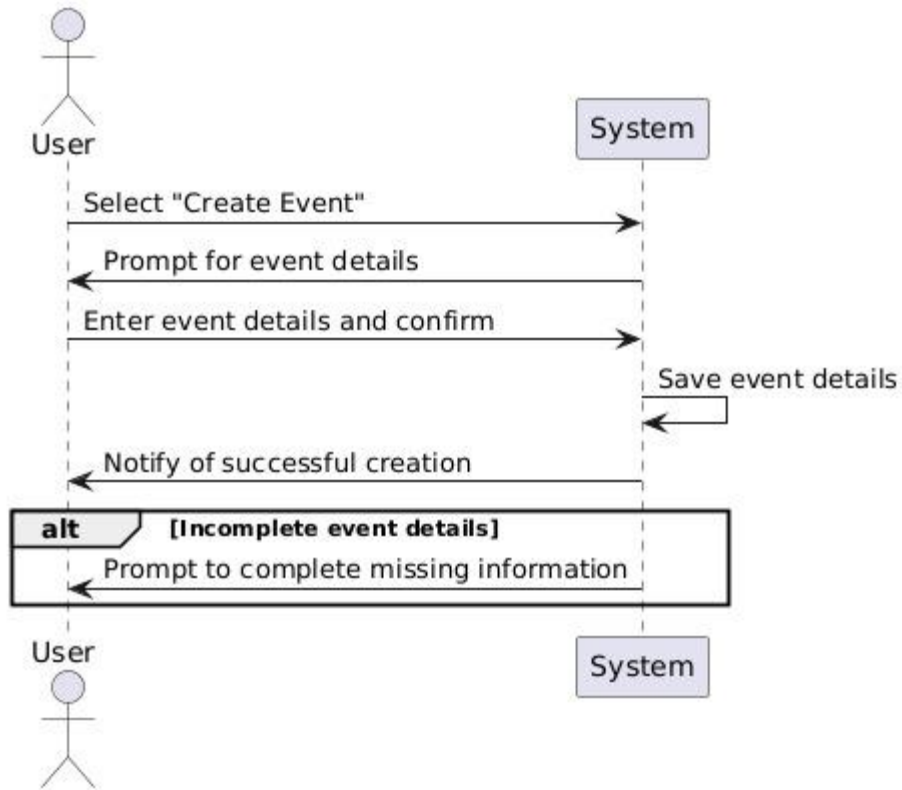
Use Case 3:

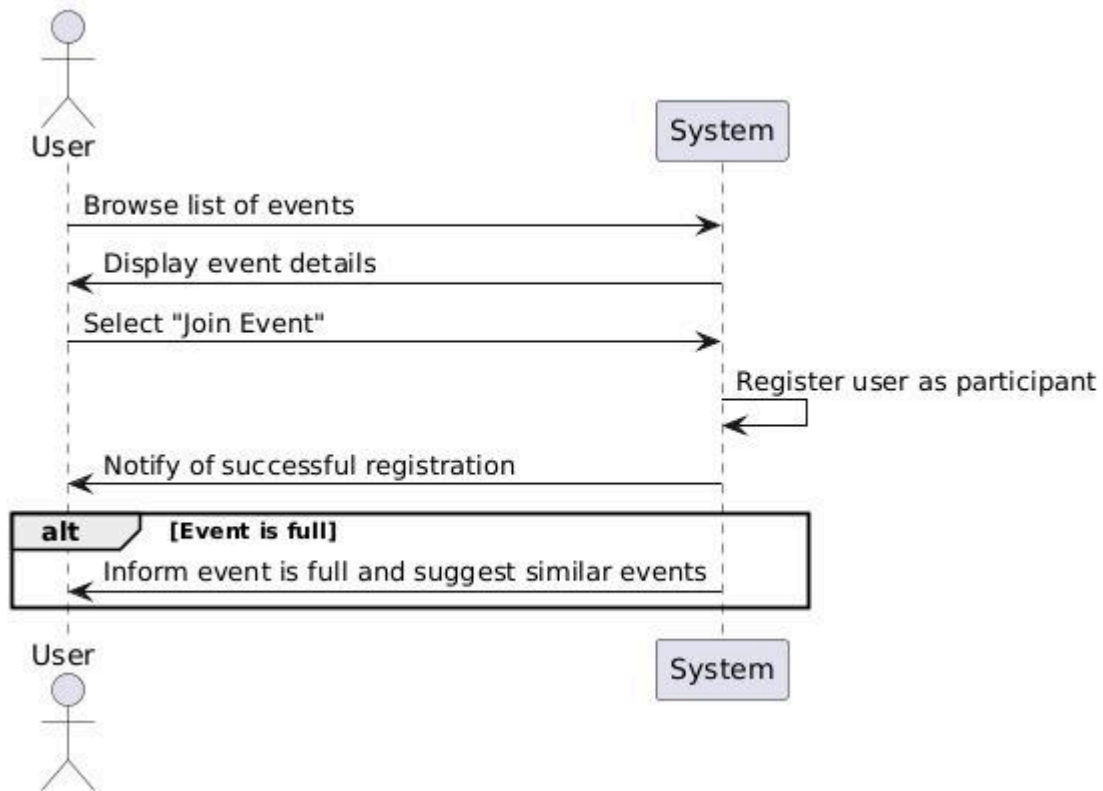
Use Case 4:

Use Case 5:

Use Case 6:

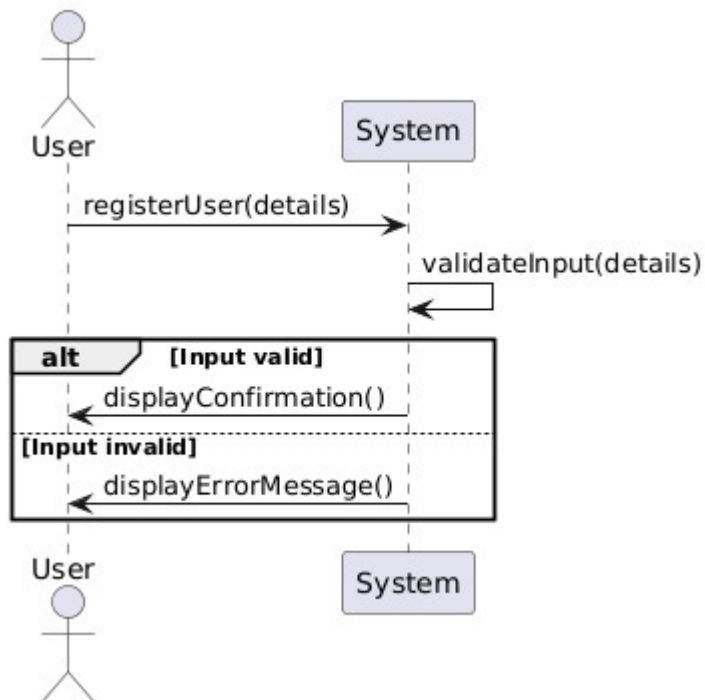
Use Case 7:**Use Case 8:**

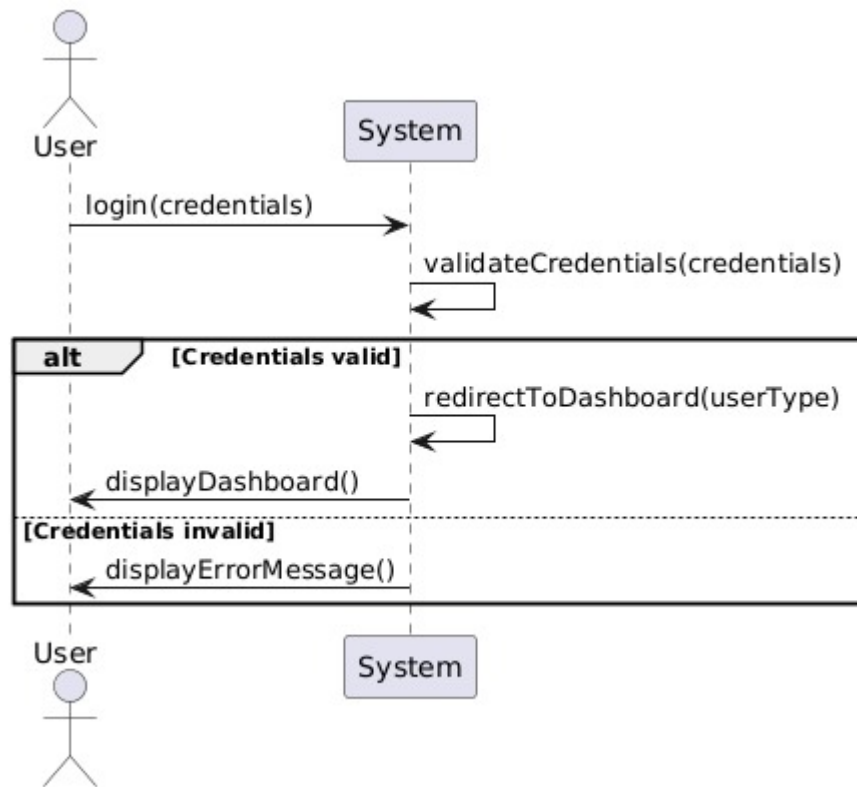


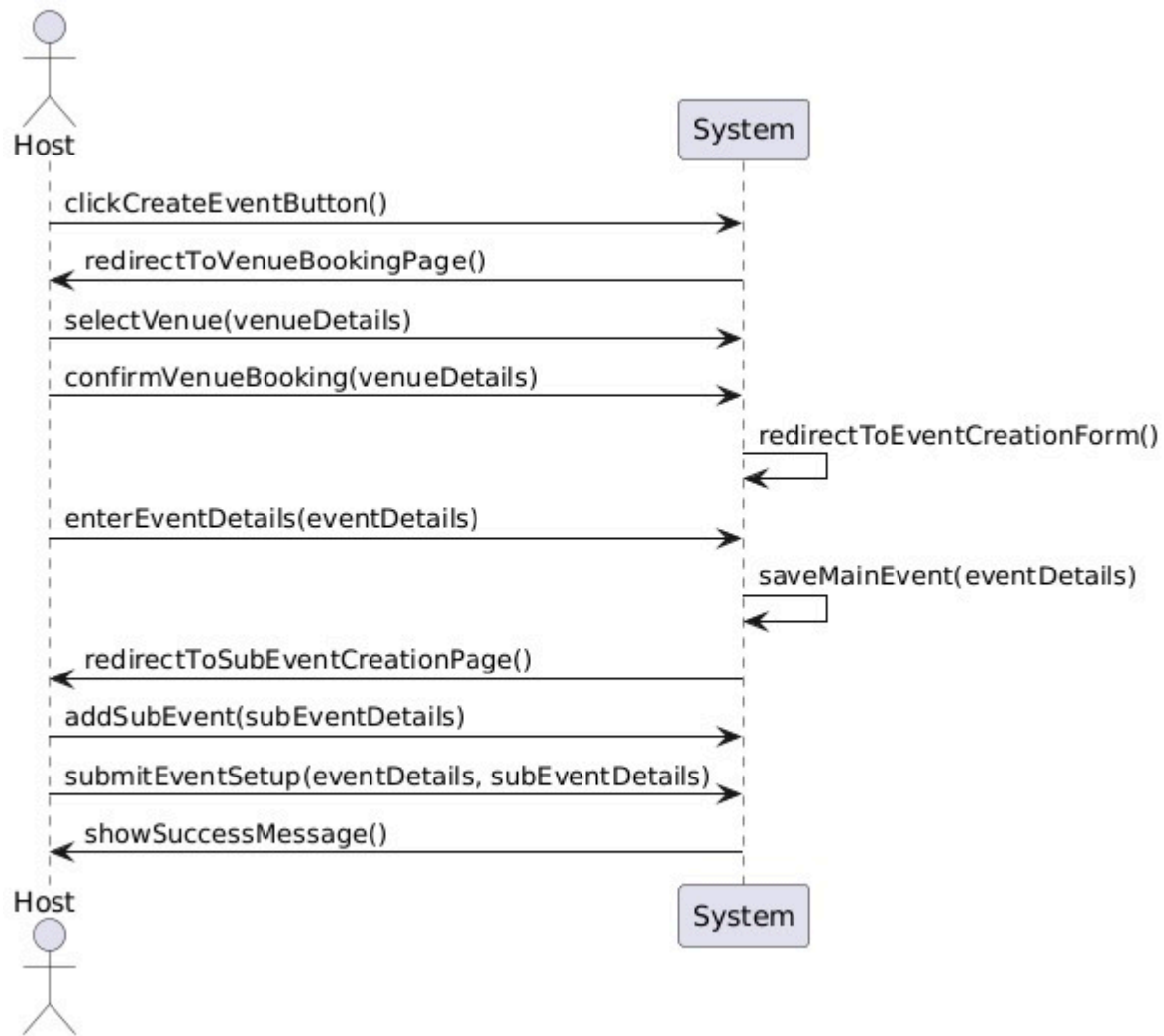
Use Case 9:

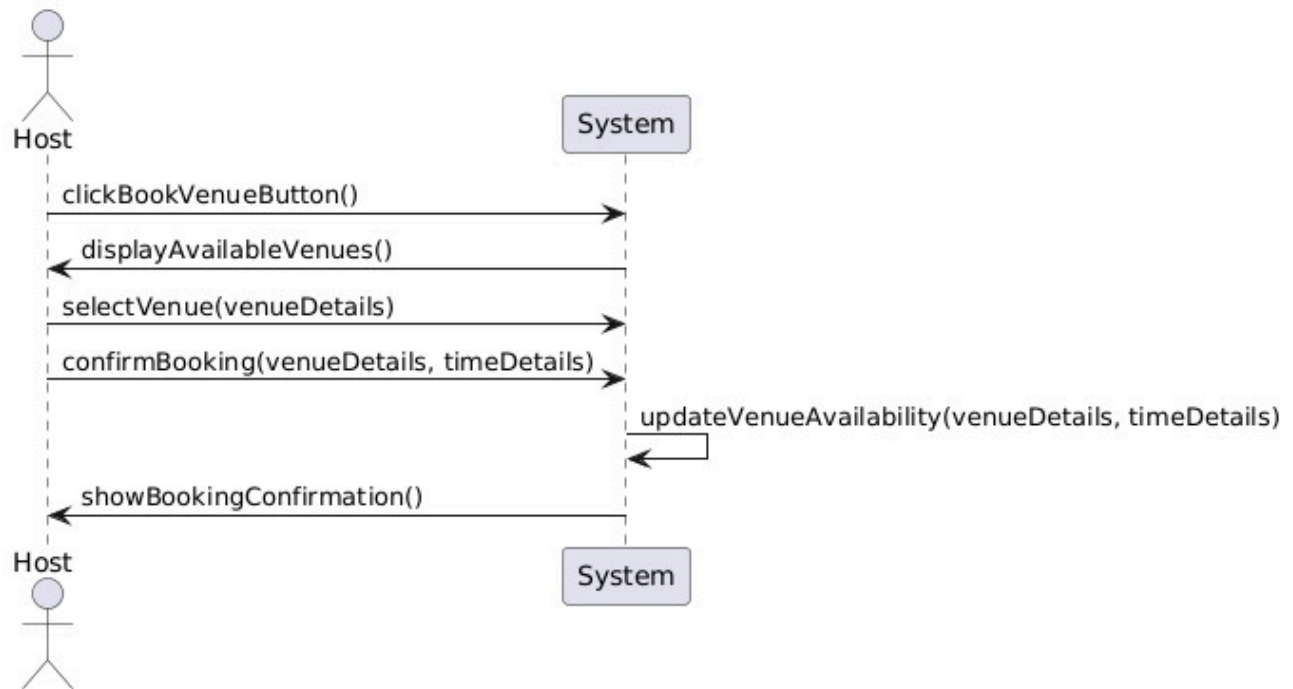
4. Sequence Diagram

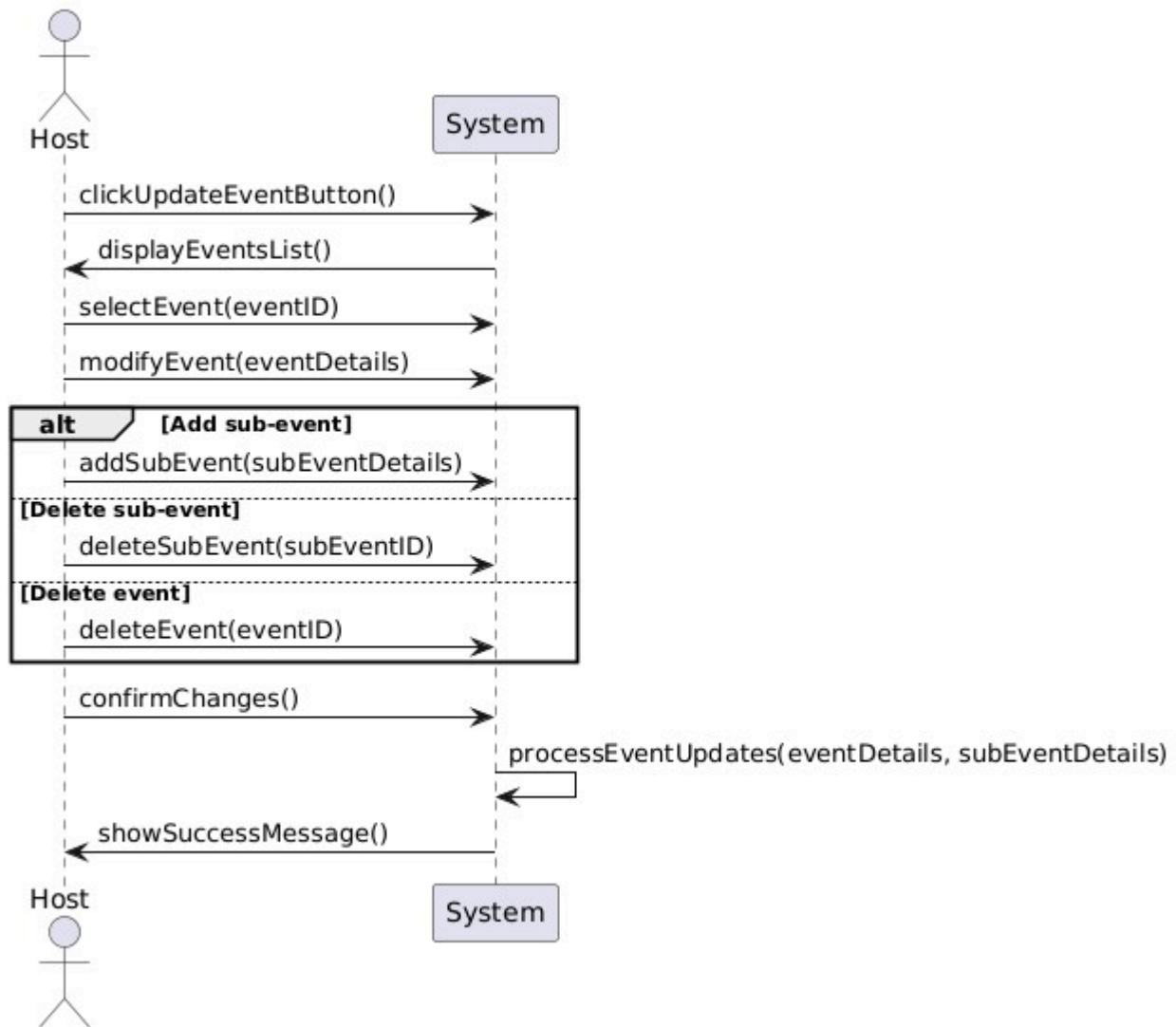
Use Case 1:

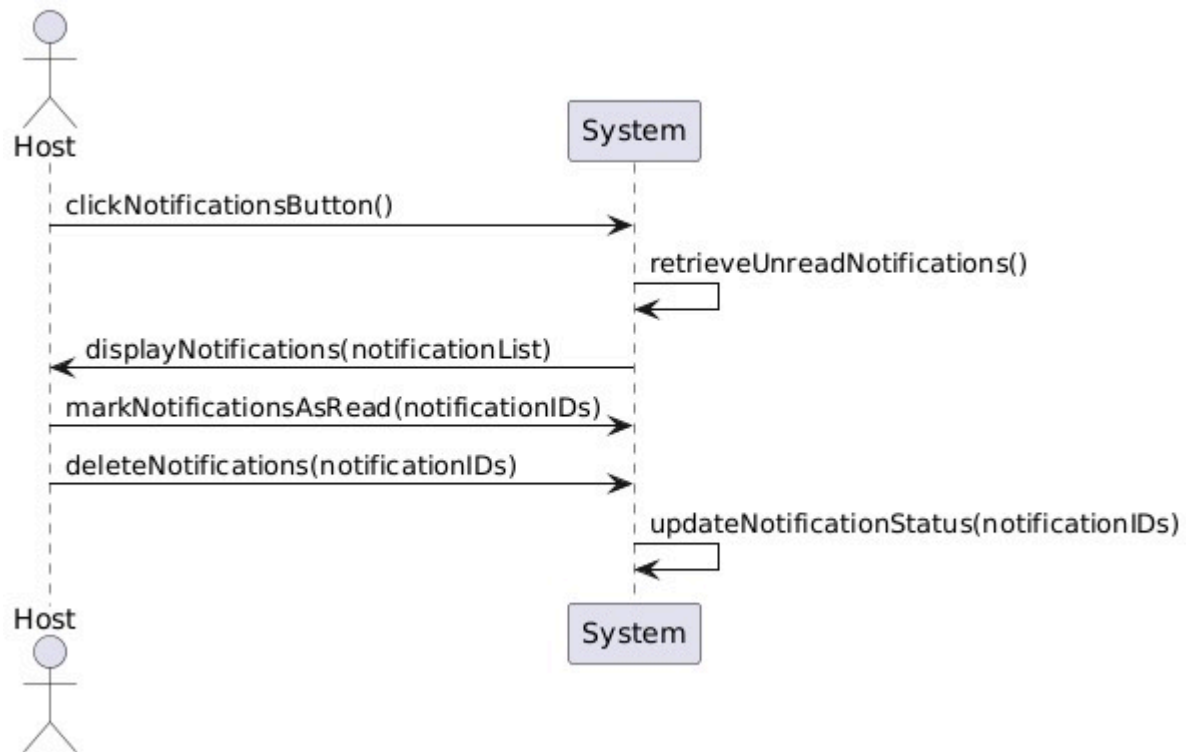


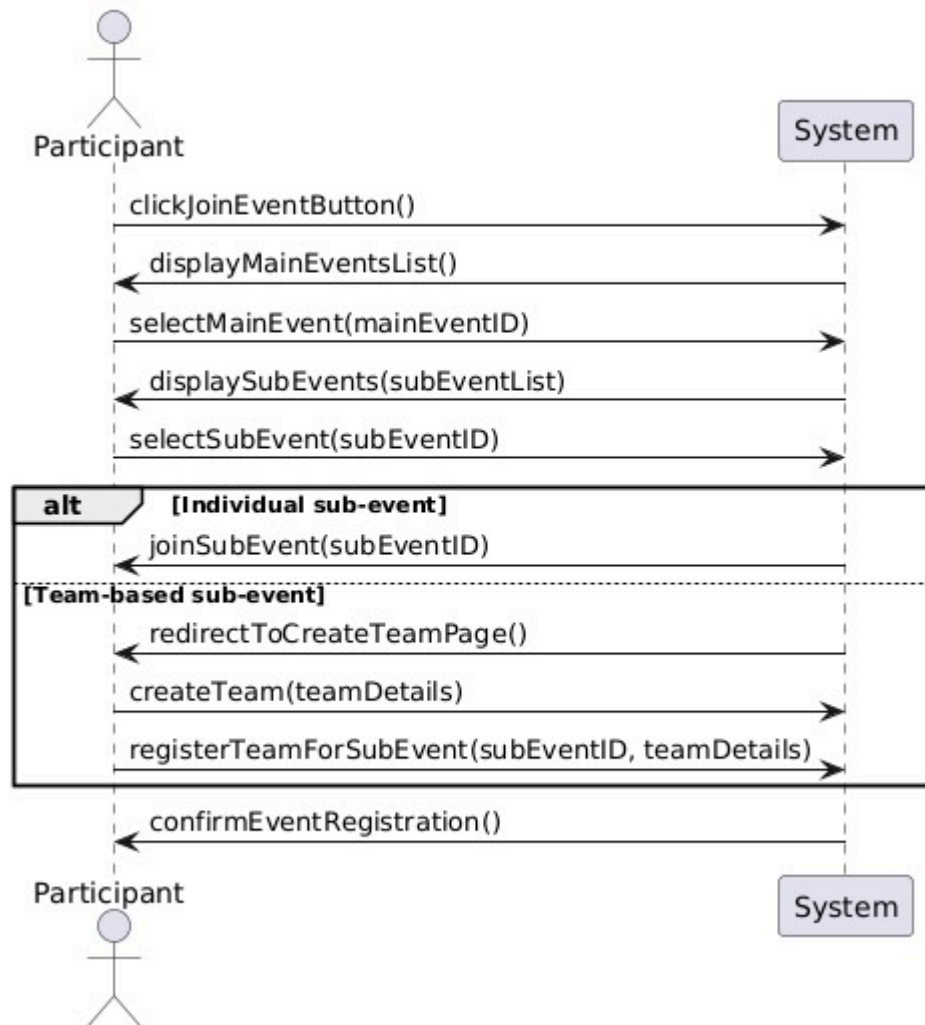
Use Case 2:

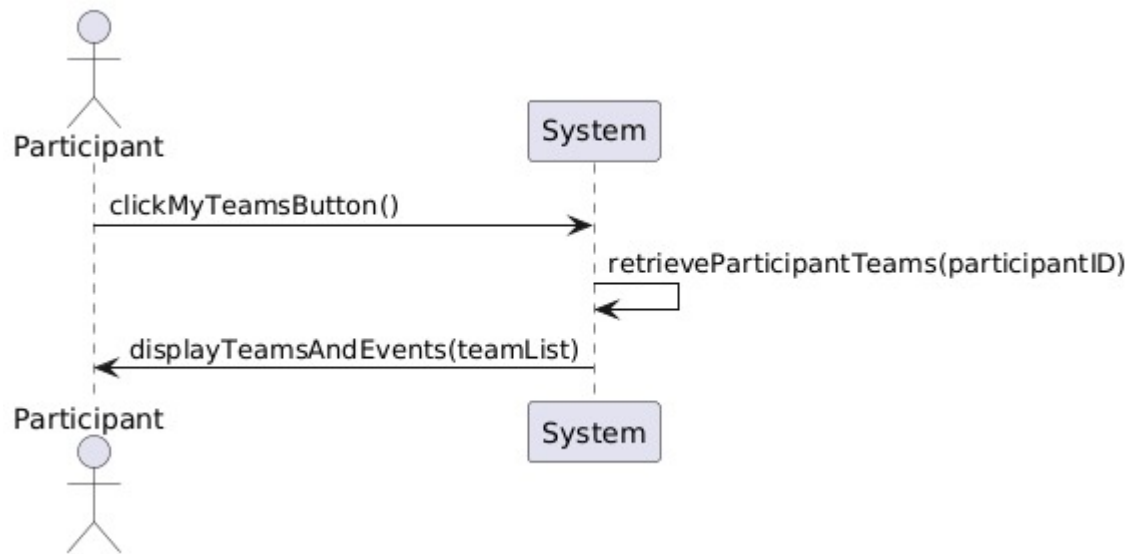
Use Case 3:

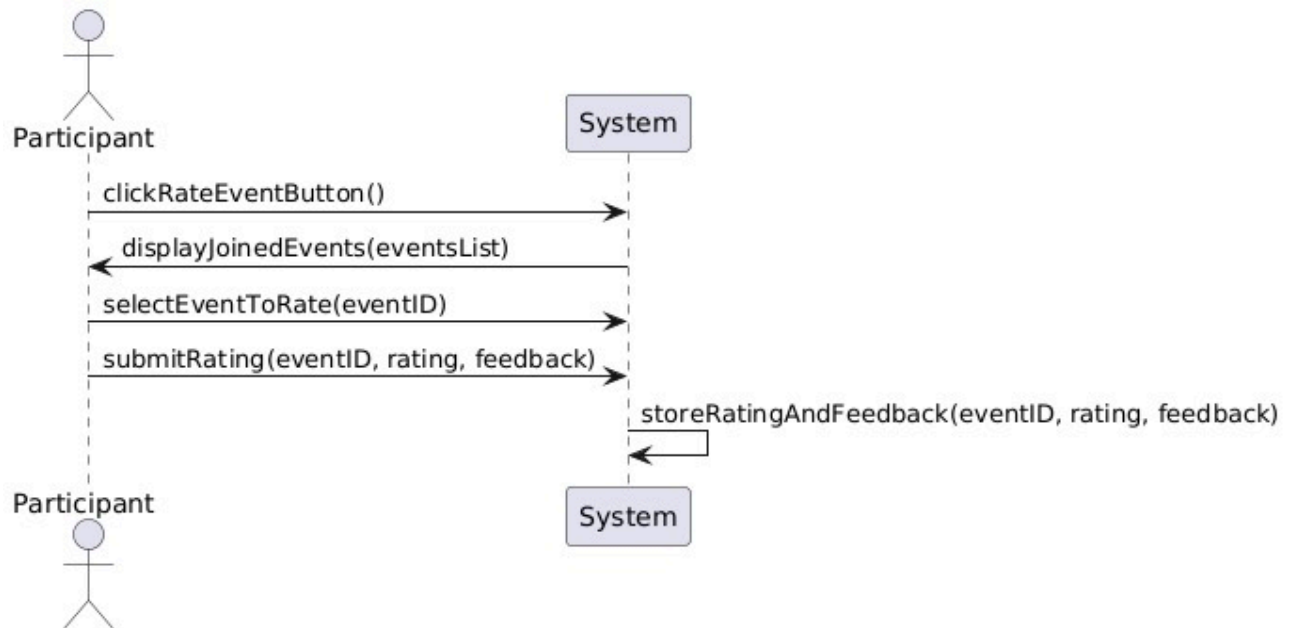
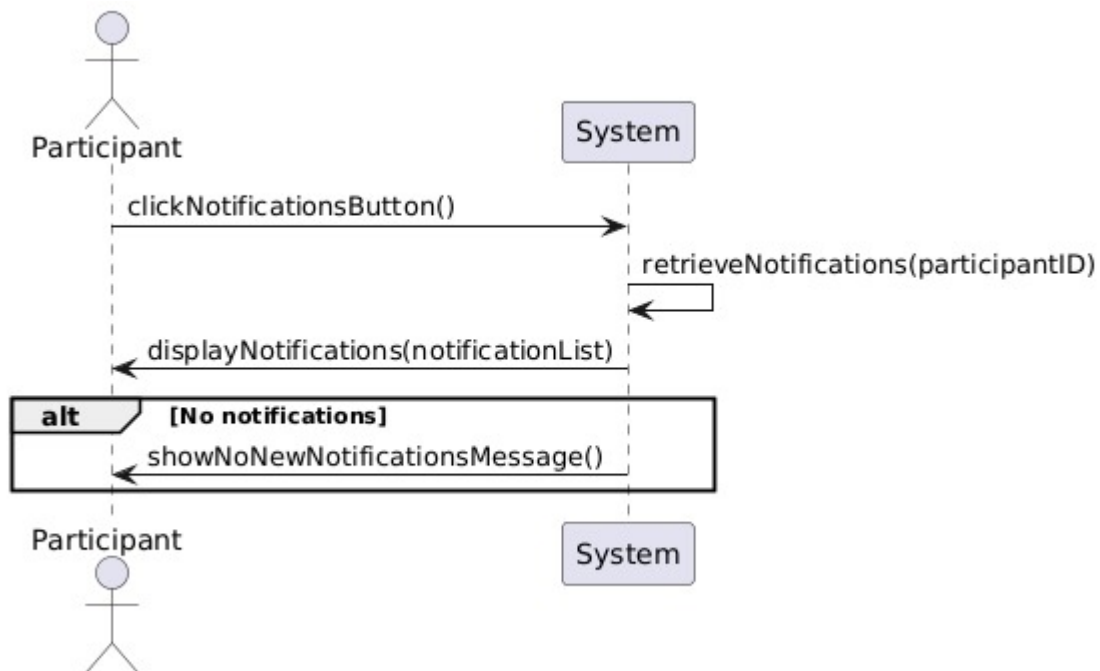
Use Case 4:

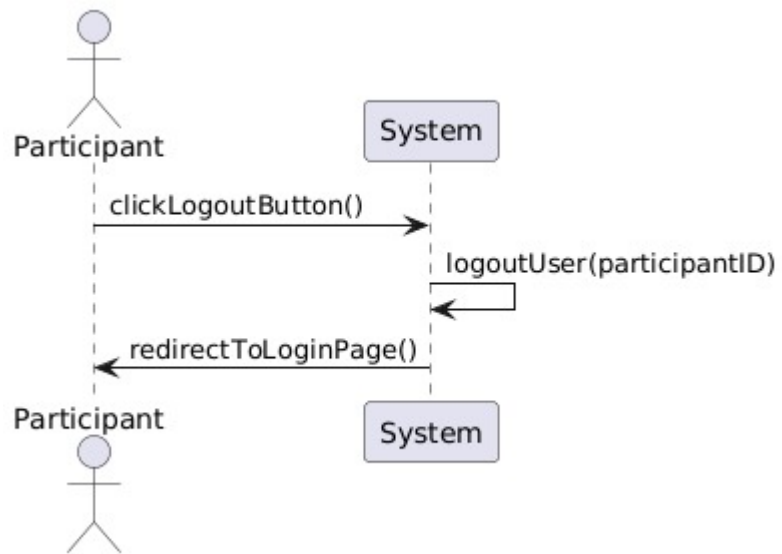
Use Case 5:

Use Case 6:

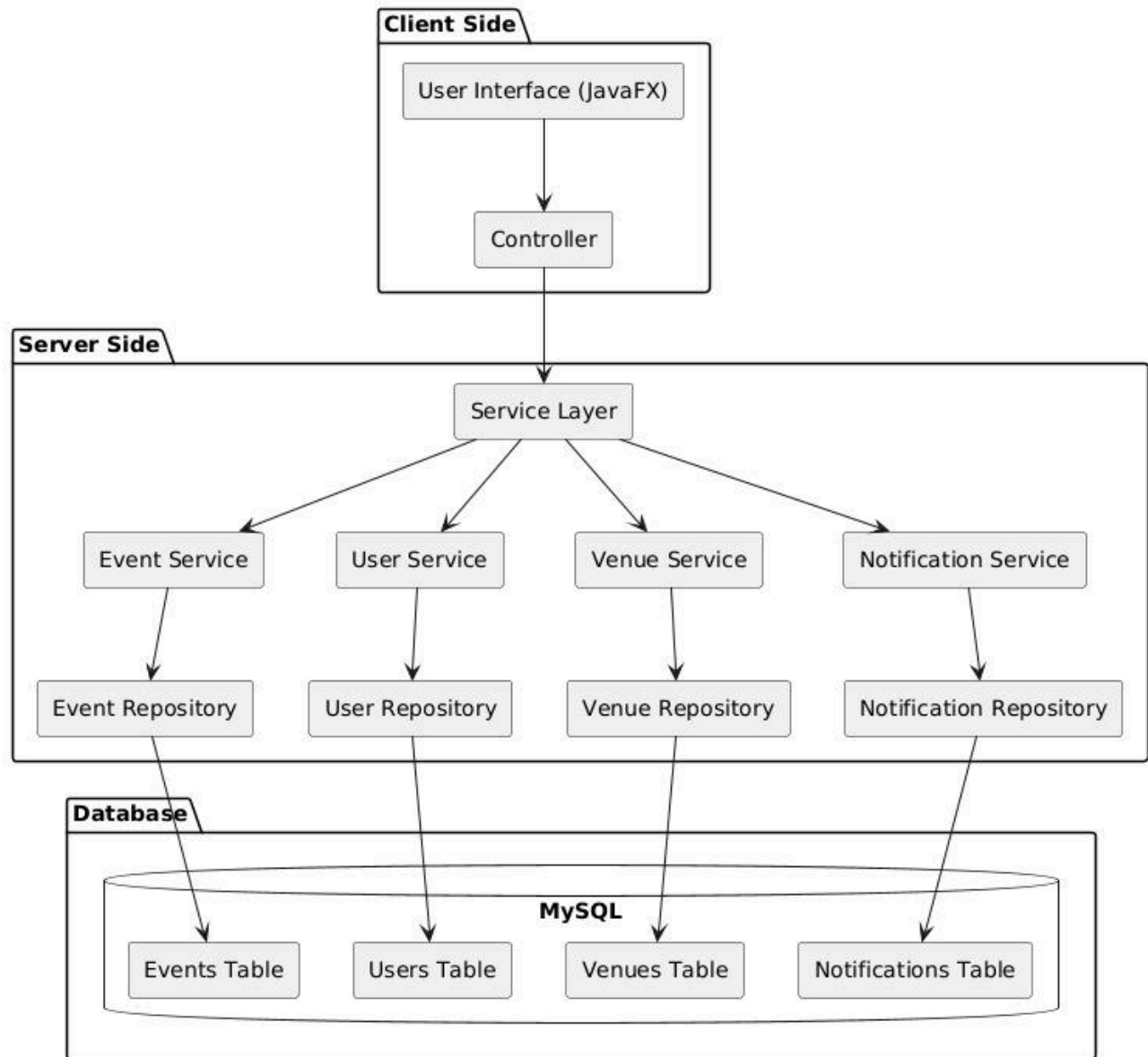
Use Case 7:

Use Case 8:

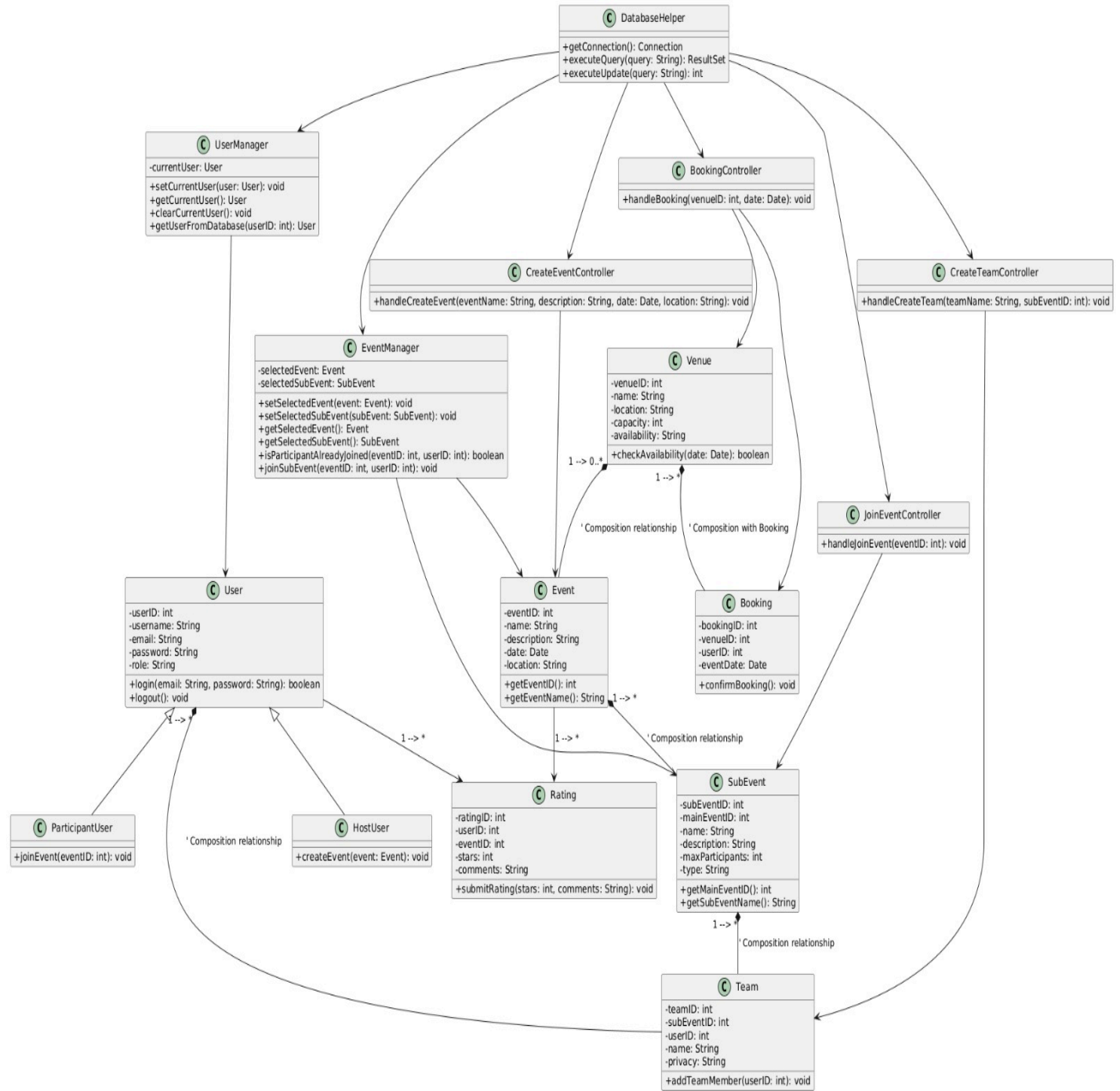
Use Case 9:**Use Case 10:**

Use Case 11:

5. Component Diagram



6. Class Diagram



7. Deployment Diagram

