

SportSync: A Social Sports Networking and Ground Booking App

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.

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.

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.

1. Use Case

Name: Create a

Sports Event **Scope:**

SportSync

Level:

User Goal

Stakeholders and Interests:

User (Event Organizer): Wants to create and organize a sports event efficiently. Participants: Interested in joining well-organized sports events.

Preconditions:

The user must be registered and logged into the app.

Postconditions:

The event is successfully created and available for others to join.

Main Success Scenario:

The user selects 'Create Event' from the app's menu. The system prompts the user to enter event details (sport type, date, venue preferences). The user enters the required details

and confirms. The system saves the event and notifies the user that it has been successfully created. The event becomes available for other users to join.

Extensions:

If the event details are incomplete, the system prompts the user to fill in the missing information.

Special Requirements:

The app should be responsive and ensure fast event creation without delays

2. Use Case Name:

Join a Sports Event

Scope:

SportSync

Level:

User Goal

Stakeholders and Interests:

User (Participant): Wants to easily find and join sports events that match their preferences.

Event Organizer: Wants to fill the event slots with participants.

Preconditions:

The user must be registered and logged into the app.

Postconditions:

The user successfully joins the event. Main Success Scenario:

The user browses the list of available sports events. The system displays event details.

The user selects 'Join Event' and confirms their participation.

The system registers the user as a participant and notifies them

of their successful registration.

Extensions:

If the event is full, the system informs the user and suggests similar events.

Special Requirements:

The system should be able to handle multiple users joining events simultaneously.

3. Use Case Name:

Vote on Event Venue and Time

Scope:

SportSync

Level:

User Goal Stakeholders and Interests: User (Participant): Wants to have a say in the selection of the event venue and time. Event Organizer: Wants to ensure participants agree on a venue and time that works for most.

Preconditions:

The user must have joined the event and voting must be open.

Postconditions:

The event's venue and time are successfully chosen based on majority voting.

Main Success Scenario:

The system notifies participants that voting is open. The user views venue and time options and submits their vote. The system collects votes from all

participants. The system calculates the most popular choice and notifies participants of the final venue and time.

Extensions:

If the user does not vote within the time limit, the system automatically assigns them the most popular choice.

Special Requirements:

Voting should be transparent and updates should be reflected in real-time.

4. **Use Case Name:** Search for Events by Location

Scope:

SportSync

Level:

User Goal

Stakeholders and Interests:

User: Wants to find sports events happening in a specific location.

Preconditions:

The user must be registered and logged into the app.

Postconditions:

The user views a list of sports events happening in the specified location.

Main Success Scenario:

The user selects 'Search' and enters the desired location. The system displays a list of events happening in the specified location. The user browses the events and selects one to view details.

Extensions:

If no events are found in the specified location, the system suggests nearby locations or future events.

Special Requirements:

The search feature should support filtering by location and display results quickly.

5. Use Case

Name: Invite Friends to

Events **Scope**

SportSync

Level

User Goal **Primary**

Actor User of the

app

Stockholders and Interests:

User- Invites teammates to an event.

Server- Processes the invite

Pre-Conditions:

User is logged in and successfully registered.

Post-Construction:

The Teammate of the user successfully joins the event.

Main Success Scenario:

i) **User selects an event:**

The user navigates to the "Events" section of the app and selects a specific event they

want to invite friends to.

ii) **User initiates**

invitation process: The user clicks on the "Invite Friends" button associated with the selected event.

iii) **User selects friends:** The app displays a list of the user's friends or teammates. The user can scroll through the list, search for specific friends, or select from frequently contacted friends.

status in the app.

iv) **User confirms selection:**

After selecting friends, the user clicks on the "Send Invitations" button.

v) **App sends invitations:** The app sends notifications or messages to the selected friends with details of the event (date, time, location, and description).

vi) **Friends receive notifications:** Friends receive invitations via in- app notifications, email, or SMS, depending on their notification settings.

vii) **Friends view event details:** Friends can click on the invitation to view more details about the event.

viii) **Friends RSVP:** Friends have the option to accept, decline, or maybe attend the event directly from the invitation.

ix) **User sees RSVP status:** The user receives updates on who has accepted or declined the invitation and can view the RSVP

Extensions:

- i) If the user has no friends to invite: The app displays a message indicating that there are no friends available to invite.
- ii) If a friend is not reachable: If the app cannot send a notification to a selected friend (due to lack of internet, privacy settings, etc.), the app informs the user of the issue and allows them to try again.
- ii) If friends decline the invitation: The user receives a notification indicating which friends declined the invitation.

Special Requirements:

- i) The app should ensure that notifications are sent in real-time.
- ii) The app must comply with privacy regulations regarding sharing user information.

6. Use Case

Name: Create or Join

a Team **Scope**

SportSync

Level

User Goal **Primary**

Actor User of the

app

Stockholders and Interests:

User- Creates or Join a Team

Server- Creates a new team or adds the user in an already existing one.

Pre-Conditions:

User is logged in and successfully registered.

User has access to “Teams” section of the application.

Post-Construction:

The User is either creator of a new team or a member of an existing team.

Team members can view team

details and participate in scheduled activities.

Main Success Scenario:

i) User navigates to Teams section: The user selects the "Teams" option from the main menu of the app.

ii) User decides to create or join a team:

- The app presents two options: "Create a Team" and "Join a Team."

iii) If the user chooses to create a team:

- a. User clicks on "Create a Team": The app prompts the user to fill out a team creation form.
- b. User enters team details: The user inputs the team's name, description, sport type, location, and any other relevant information.
- c. User sets privacy settings: The user selects whether the team will be public (anyone can join) or private (invitation only).

D. User submits the form: The user clicks the "Create Team" button.

e. App confirms creation: The app confirms that the team has been successfully created and displays the team page,

allowing the user to invite members.

iv) If the user chooses to join a team:

- a. User clicks on "Join a Team": The app displays a list of available teams that match the user's interests or sport type.
- b. User browses teams: The user can scroll through the list or use filters to find a suitable team.
- c. User selects a team: The user clicks on a team to view more details.
- d. User requests to join: The user clicks on the "Join Team" button.
- e. App confirms request: The app sends a request to the team administrator if the team is private or directly adds the user if the team is public.

f. User receives notification: The user receives a notification about the request status (approved or denied).

iv) User views team details: Regardless of whether they created or joined a team, the user can now access the team page, which includes:

- Team roster
- Upcoming events and matches
- Team chat or communication features

v) User interacts with team: The user can participate in team activities, communicate with teammates, and manage their involvement in events.

Extensions:

i) **If the team's name already exists:** The app displays an error message if the user attempts to

create a team with a name that is already in use.

ii) **If the user cannot find a suitable team:** The app provides an option to search for teams by keywords or create a new team if no suitable options are available.

iii) **If a team administrator denies the join request:** The user receives a notification indicating that their request to join the team has been denied.

Special Requirements:

i) The app should allow for easy updates to team information and provide notifications for team activity changes.

ii) The app must ensure that user data privacy is maintained during team

interactions.

7. **Use Case Name:**

View Event Details **Scope**

SportSync

Level

User Goal

Primary Actor

User of the app

Stockholders and Interests:

User- wants to view detailed information about an event.

Event Organizer- Wants potential attendees to have information about the event.

Server- Enhance user experience by providing clear and accessible information.

Pre-Conditions:

User is logged in and successfully registered.

User has access to “Events” section of the application.

There are events listed in the app for the user to view.

Post-Construction:

The user has viewed all relevant details about the event, including time, location, participants, and any other pertinent information

Main Success Scenario:

i) **User navigates to the Events section:** The user selects the "Events" option from the main menu of the app.

ii) **User browses events:** The app displays a list of upcoming sports events that the user can scroll through.

iii) **User selects an event:** The user clicks on a specific event they are interested in to view more details.

iv) **App retrieves event details:** The app fetches the event information from the database.

v) **App displays event details:** The app presents a detailed view of the event, which includes:

- Event name
- Date and time
- Location (with a map link)
- Description of the event
- List of participating teams or players
- RSVP or registration status

- Contact information for the event organizer

vi) **User can perform additional actions:** The user has the option to:

- RSVP or register for the event
- Share the event details with friends or teammates
- Add the event to their personal calendar
- Return to the events list

Extensions:

i) **If the event has no details available:** The app displays a message indicating that detailed information is not available for this event and suggests checking back later.

ii) **If the user tries to access a past event:** The app informs the user that the event has already occurred and offers options to view upcoming events.

iii) **If the user is not registered for a required event:** The app prompts the user to register or log in to access certain event details.

Special Requirements:

- i) The app should ensure that event details are updated in real-time, especially for changes in scheduling or location.
- ii) The app must have user-friendly navigation and display event information clearly.

8. Use Case Name:

Track Event Participation

Scope

SportSync

Level

User Goal

Primary Actor

User of the app

Stockholders and Interests:

User- wants to keep track of the events they have participated in or plan to attend.

Event Organizer- Wants to manage attendance and gather participation statistics.

Server- Aims to provide a user-friendly experience that encourages engagement and participation tracking.

Pre-Conditions:

User is logged in and successfully registered.

User has previously registered for or participated in one or more events.

Post-Construction:

The user has an updated view of their event participation history, including upcoming and past events.

Main Success Scenario:

- i) **User navigates to the Participation section:** The user selects the "My Events" or

"Participation" option from the main menu of the app.

ii) **App displays**

participation overview: The app presents a summary view of the user's participation, categorizing events into:

- Upcoming Events
- Past Events
- Pending RSVPs

iii) **User selects an event**

category: The user clicks on "Upcoming Events" to view events they are registered for or planning to attend.

iv) **App displays upcoming**

events: The app lists all upcoming events the user is participating in, showing key details such as:

- Event name
- Date and time
- Location
- Status (Registered, Interested, etc.)

v) **User selects a specific**

event: The user clicks on an upcoming event to view more details about their participation.

vi) **App retrieves and displays event participation**

details: The app shows:

- User's registration status (confirmed, pending, etc.)
- Option to cancel participation if applicable
- Event details (as described in the "View Event Details" use case)
- Option to add the event to a personal calendar

vii) **User views past**

participation: The user can click on the "Past Events" category to view events they have attended, including:

- Event name
- Date and time of participation

- Feedback option (to rate the event or leave comments)

Viv) **User tracks participation metrics:** The app may provide statistics related to the user's participation, such as:

- Total events attended
- Types of sports or events engaged in
- Average participation per month

xi) **User exits participation tracking:** The user can return to the main menu or continue exploring other features of the app

Extensions:

i) **If the user has no upcoming events:** The app displays a message indicating that there are no upcoming events and suggests exploring the events list to find new opportunities.

ii) **If the user tries to access a past event:** The app shows details of the past event and may provide a summary of the user's participation feedback or notes.

iii) **If the user has a pending RSVP:** The app informs the user of the pending status and may allow them to confirm their attendance.

Special Requirements:

i) The app should update participation statuses in real-time and ensure that users can easily navigate between past and upcoming events.

ii) The app must provide an intuitive interface for users to track and manage their event participation effectively.

9. Use Case Name:

Book a Venue

Primary Actor: Event Organizer

Stakeholders & Interests:

Event Organizer: Wants to ensure a venue is booked for the scheduled event.

Venue Owner: Ensures venue bookings are handled and tracked properly.

Participants: Want assurance that the event venue is confirmed.

Preconditions:

The event organizer has created or is in the process of creating a sports event.

The app displays a list of available venues.

The user is authenticated (logged in).

Triggers: The event organizer selects the option to book a venue after finalizing the event details.

Main Success Scenario:

I. The event organizer navigates to the "Book a Venue" section of the app.

II. The app displays a list of available venues, filtered by location and availability.

III. The organizer selects a venue and specifies the date and time of the event.

IV. The app checks the availability of the selected venue and provides confirmation.

V. The organizer confirms the booking and makes a payment if required.

VI. A confirmation notification is sent to the organizer and the venue is added to the event details.

VII. Participants are notified of the confirmed venue.

Extensions:

III.a. If no venues are available, the app suggests alternative times or locations.

V.a. If payment fails, the organizer is prompted to retry or use a different method.

VII.a. If booking fails, the organizer is notified with an error message.

Postconditions:

The venue is successfully booked and tied to the event.
All participants are notified of the confirmed venue.

10. Use Case Name:

View Venue Availability

Use Case Name: View Venue Availability

Primary Actor: User (Event Organizer or Participant)

Stakeholders & Interests:

User: Wants to check venue availability to organize or suggest a sports event.

Venue Owner: Wants to provide accurate and up-to-date availability information.

Preconditions:

The user is logged in to the app.
The app has access to the venue database with up-to-date availability.

Triggers: The user selects the option to check venue availability.

Main Success Scenario:

- I. The user navigates to the "Check Venue Availability" section.
- II. The app prompts the user to input search criteria (e.g., location, type of sport, date, and time).
- III. The app retrieves and displays a list of venues with availability based on the criteria.
- IV. The user reviews the availability and selects a venue for more details.
- V. The user can view venue details (e.g., amenities, location, pricing) and proceed to book the venue if desired.

Extensions:

- III.a. If no venues match the criteria, the app suggests alternatives.
- IV.a. If availability changes during the session, the app updates the displayed information and notifies the user.

Postconditions:

The user successfully views the availability of selected venues.

The user can proceed to book the venue if needed.

11. Use Case Name:

Receive Notifications for Events

Use Case Name: Receive Notifications for Events

Primary Actor: User (Event Organizer, Participant)

Stakeholders & Interests:

User: Wants to stay informed about event updates like schedule changes or cancellations.

Event Organizer: Wants to notify participants of any changes or confirmations.

App System: Ensures notifications are timely and relevant.

Preconditions:

The user has registered for a sports event.

Push notifications or in-app alerts are enabled.

Triggers:

An event is created, updated, or cancelled, triggering an alert.

A change in event status (e.g., time or venue) occurs.

Main Success Scenario:

- I. The app detects an update related to an upcoming event.
- II. The system automatically generates a notification message (push or in-app alert).
- III. The notification is sent to all relevant users (event organizer, participants).
- IV. The user views the notification, which contains details of the update (e.g., time change, venue change).
- V. The user acknowledges the notification and can take action if needed (e.g., confirm participation, view new details).

Extensions:

- II.a. If the user has disabled notifications, an email is sent instead.
- III.a. If there's a communication failure, the system retries sending the notification.

Postconditions:

Users are informed of any

changes or updates to the event.
Participants stay updated on key
event details (schedule, venue).

12. Use Case Name:

Rate an Event or Venue

Use Case Name: Rate an Event or Venue

Primary Actor: User (Event Participant)

Stakeholders & Interests:

User: Wants to provide feedback on the event or venue.

Event Organizer: Interested in receiving feedback to improve future events.

Venue Owner: Interested in receiving feedback to improve services.

Preconditions:

The user participated in an event that has concluded.

The user has access to the rating feature.

Triggers:

After the event ends, the app prompts the user to rate the venue and/or event.

Main Success Scenario:

I. The user receives a notification prompting them to rate the event and venue.

II. The user navigates to the rating section within the app.

III. The app displays the event details along with the option to provide a star rating (1-5 stars) and a text review.

IV. The user submits their rating and optional written feedback.

V. The app confirms the submission and displays an acknowledgment message.

VI. The rating is saved and becomes visible to future users browsing the venue or event.

Extensions:

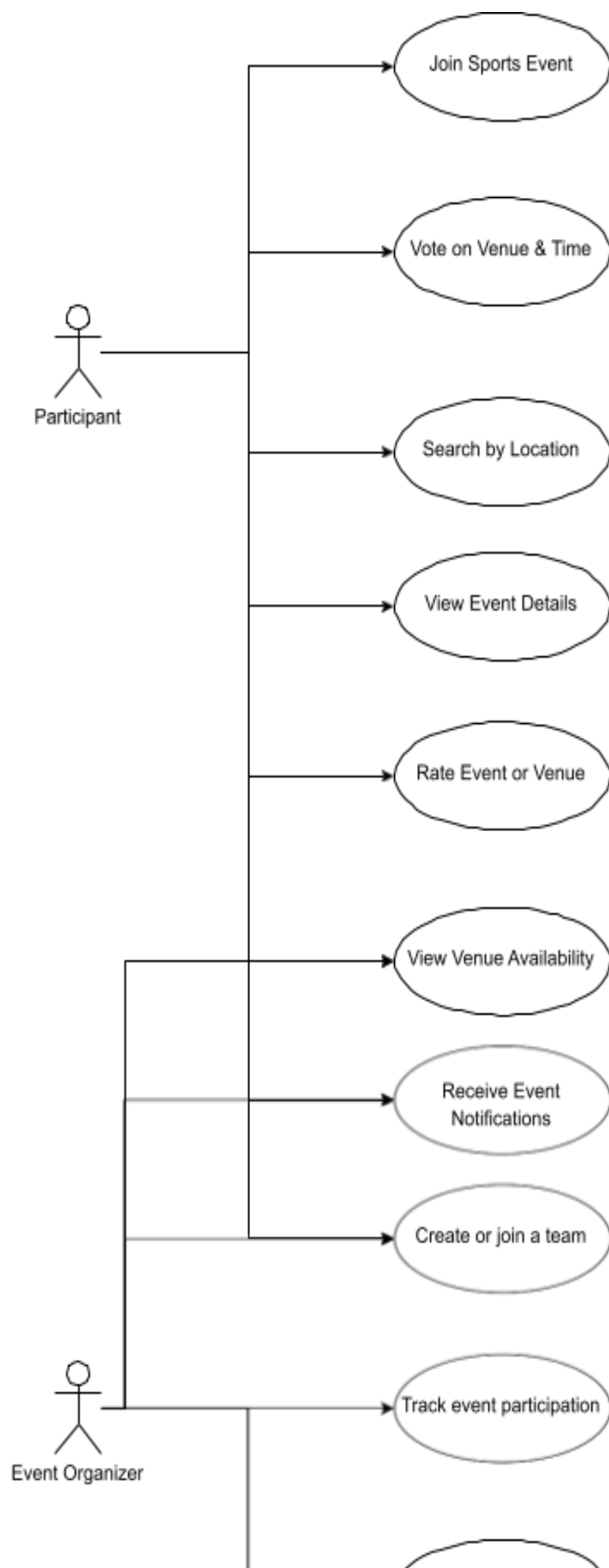
IV.a. If the user attempts to submit an inappropriate review, the system prompts them to revise their feedback.

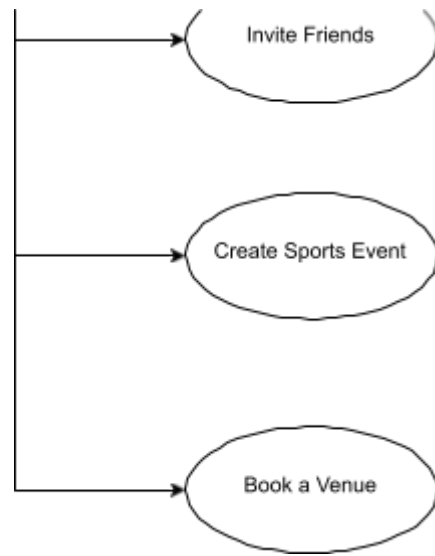
V.a. If submission fails, the app notifies the user and prompts a retry.

Postconditions:

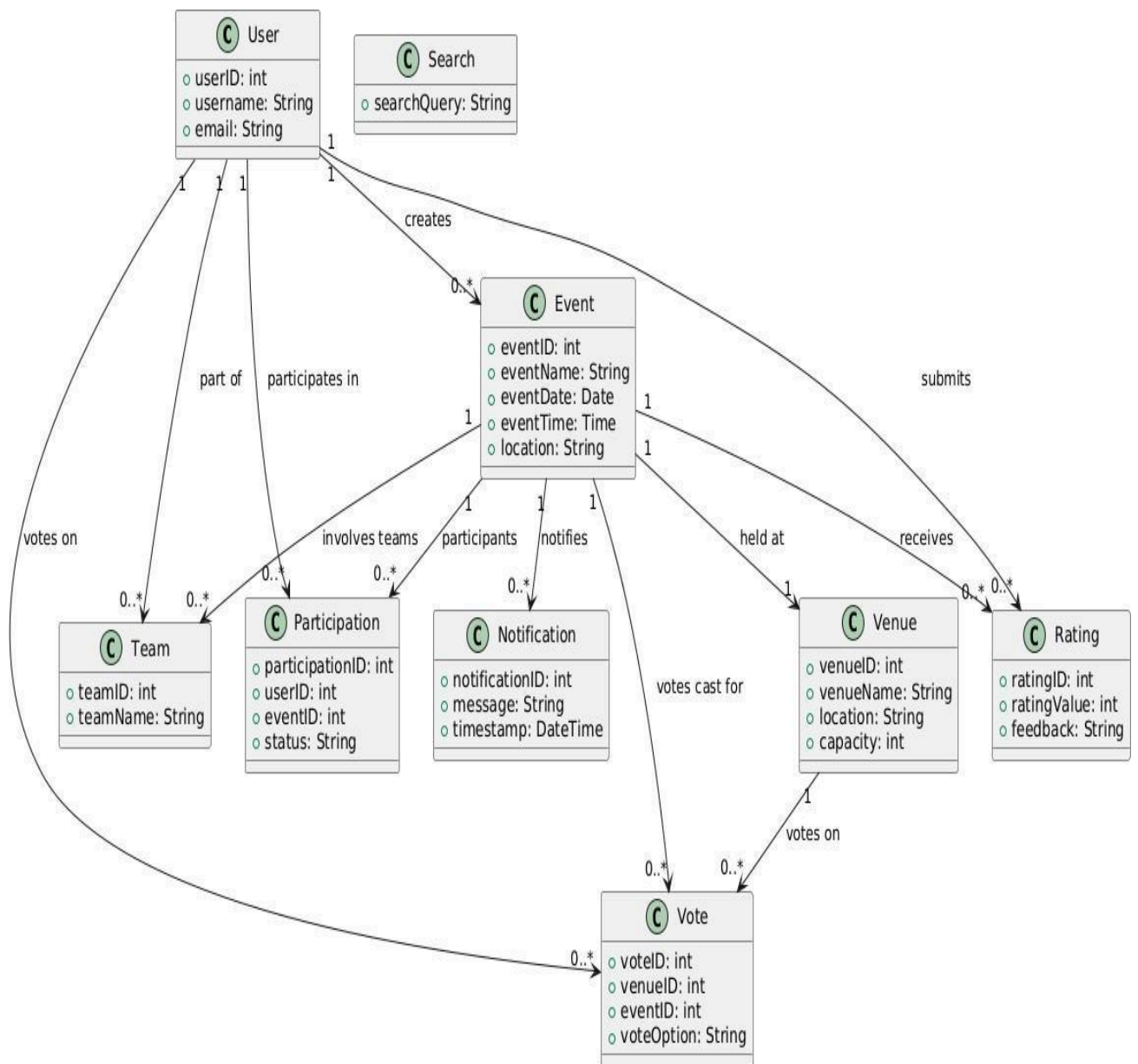
The user's feedback is stored in the system.

The rating and review are available to help other users and event organizers improve future experiences.



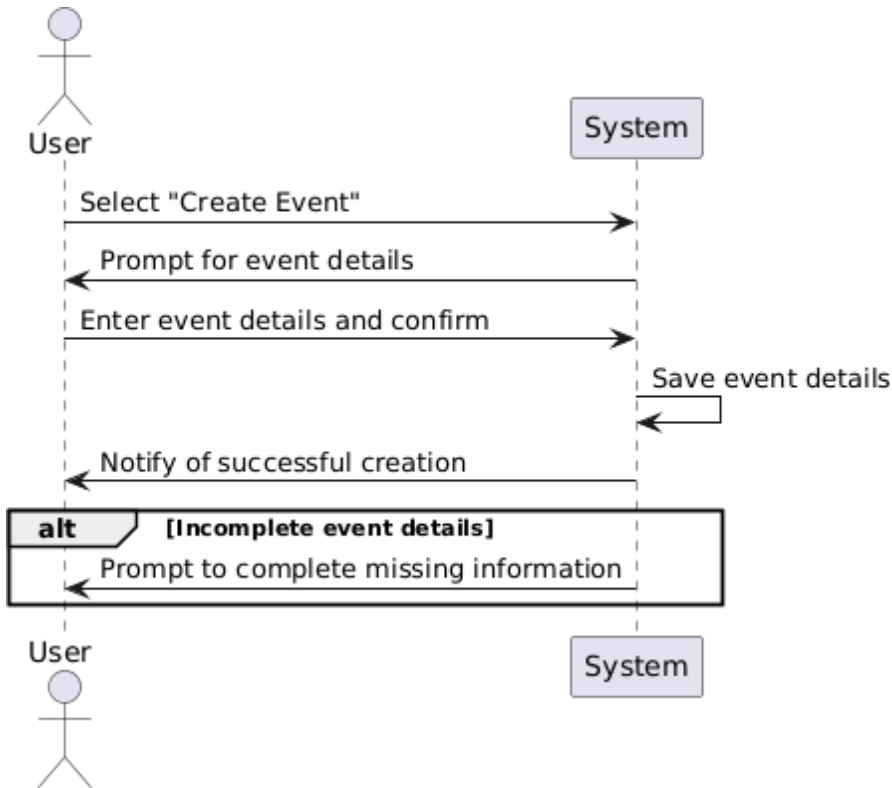


DOMAIN MODEL DIAGRAM

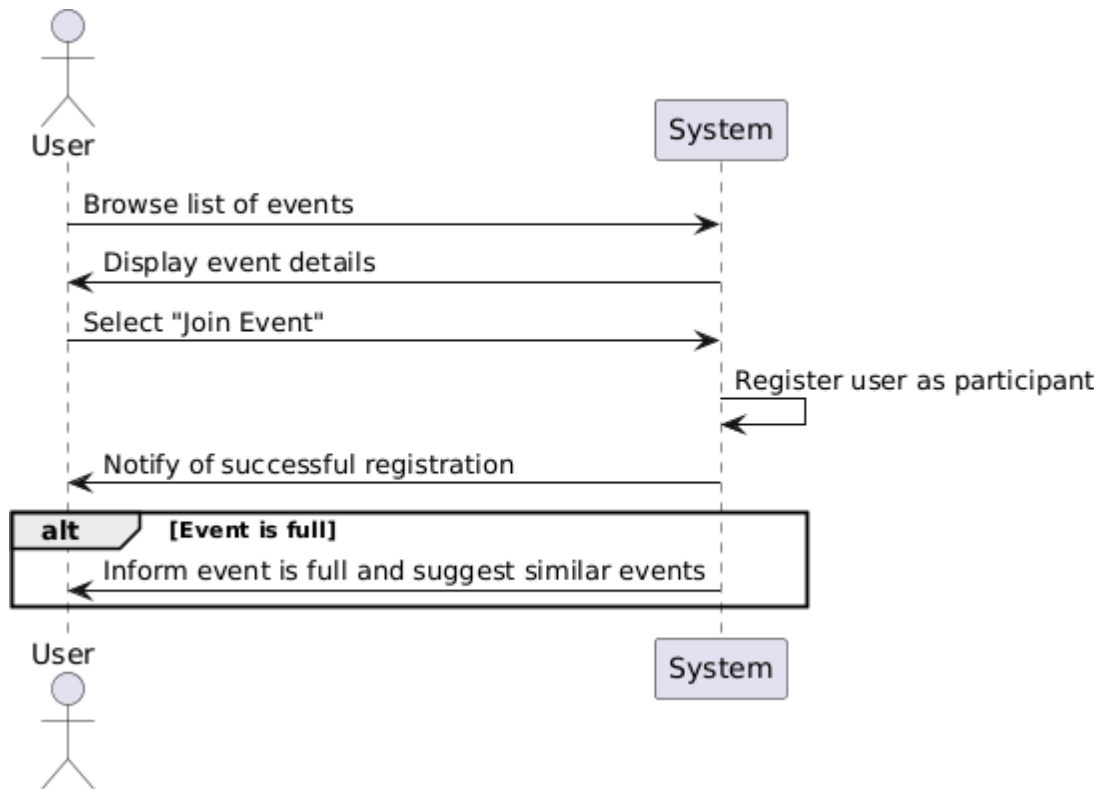


System Sequence Diagrams:

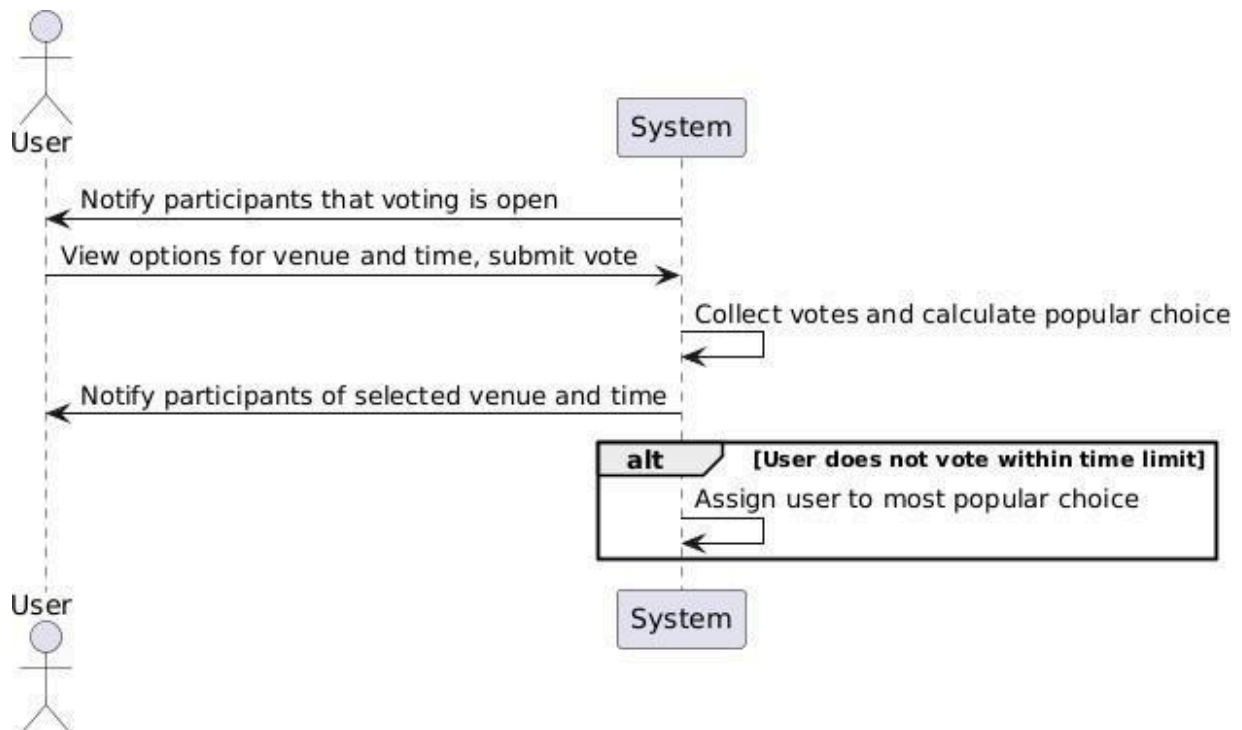
Use case 1:



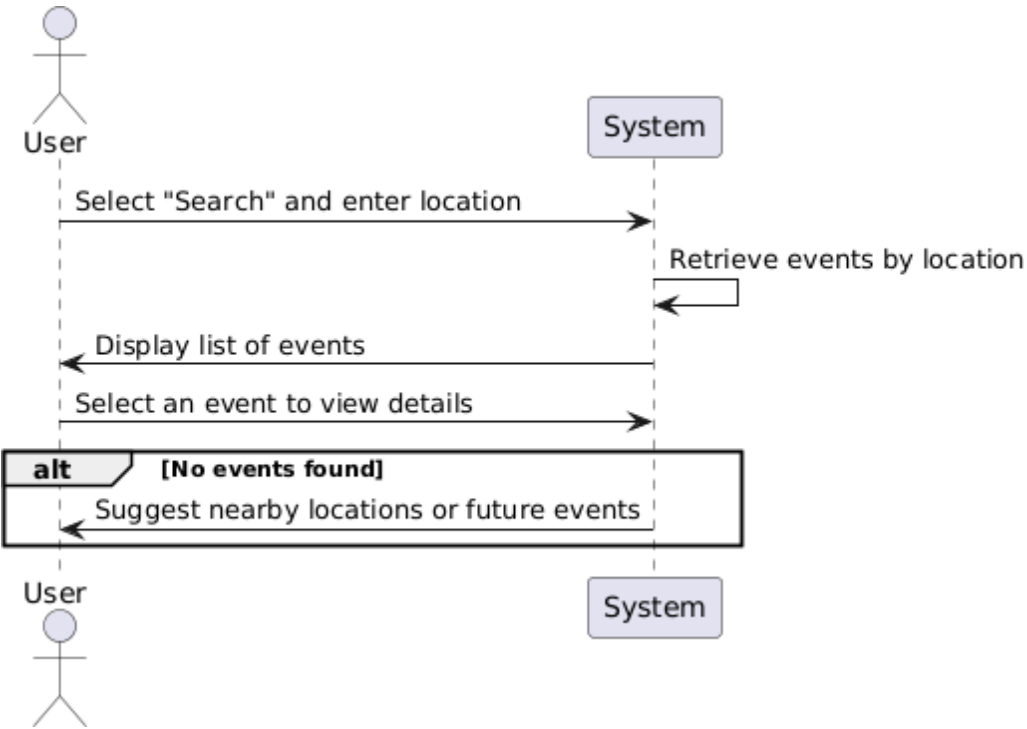
Use case 2:



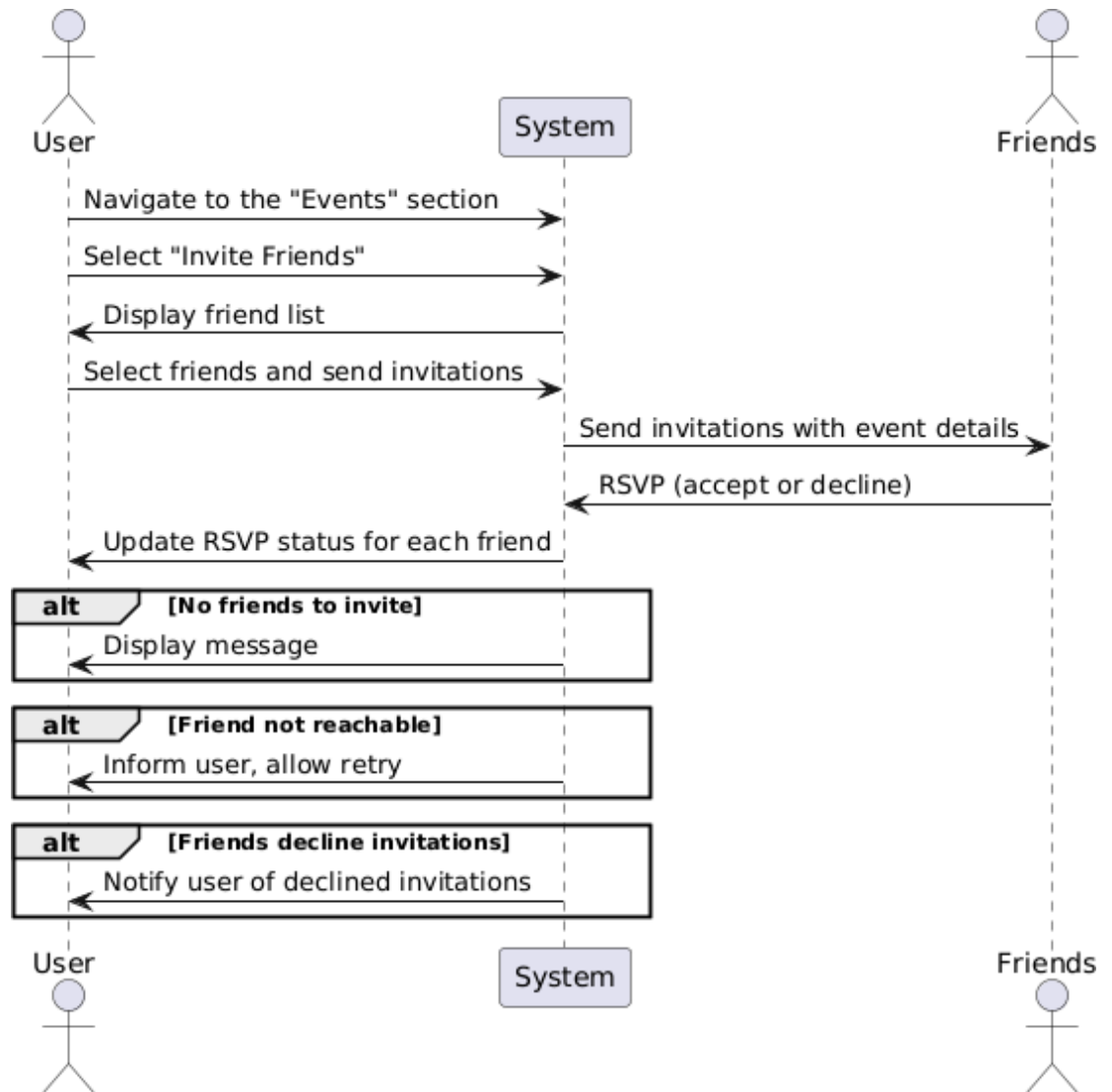
Use Case 3:



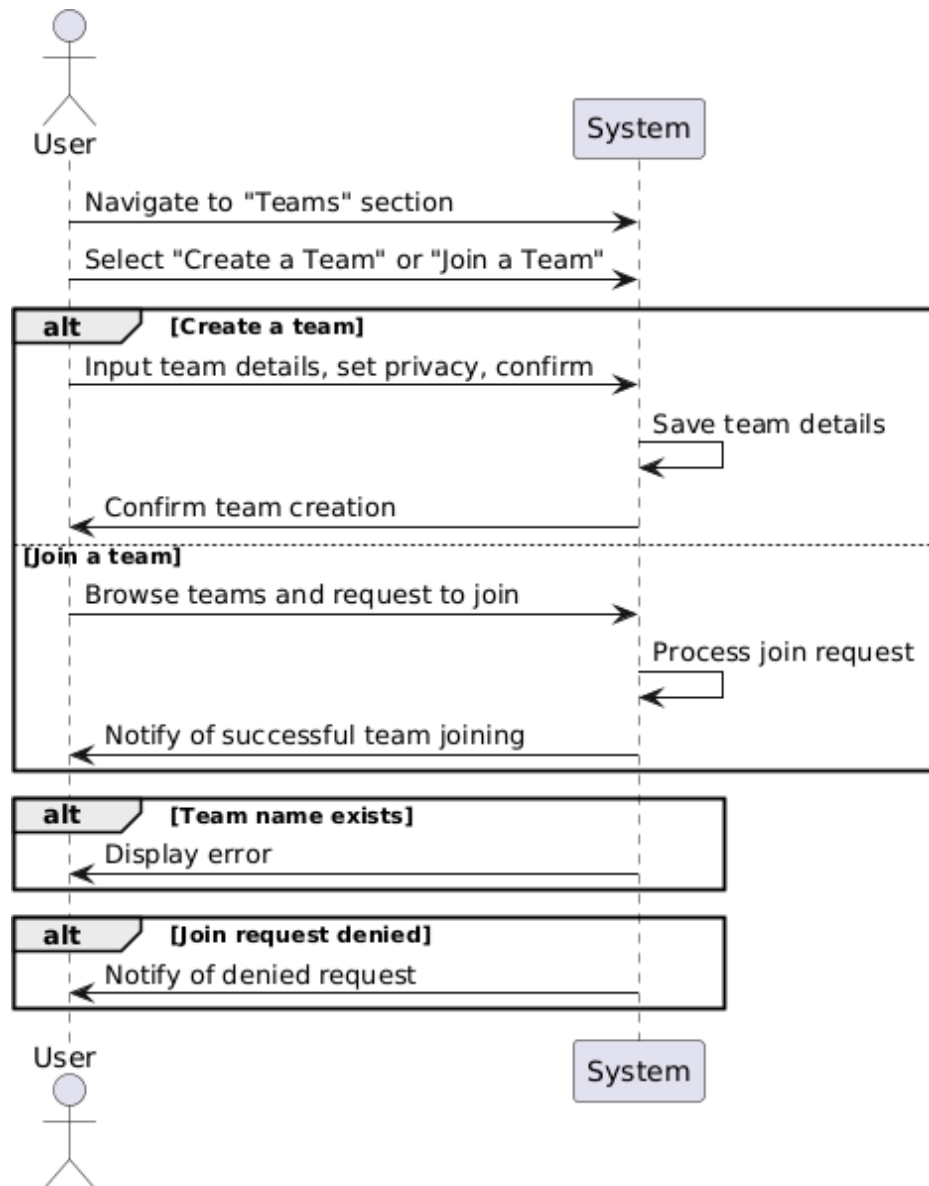
Use case 4:



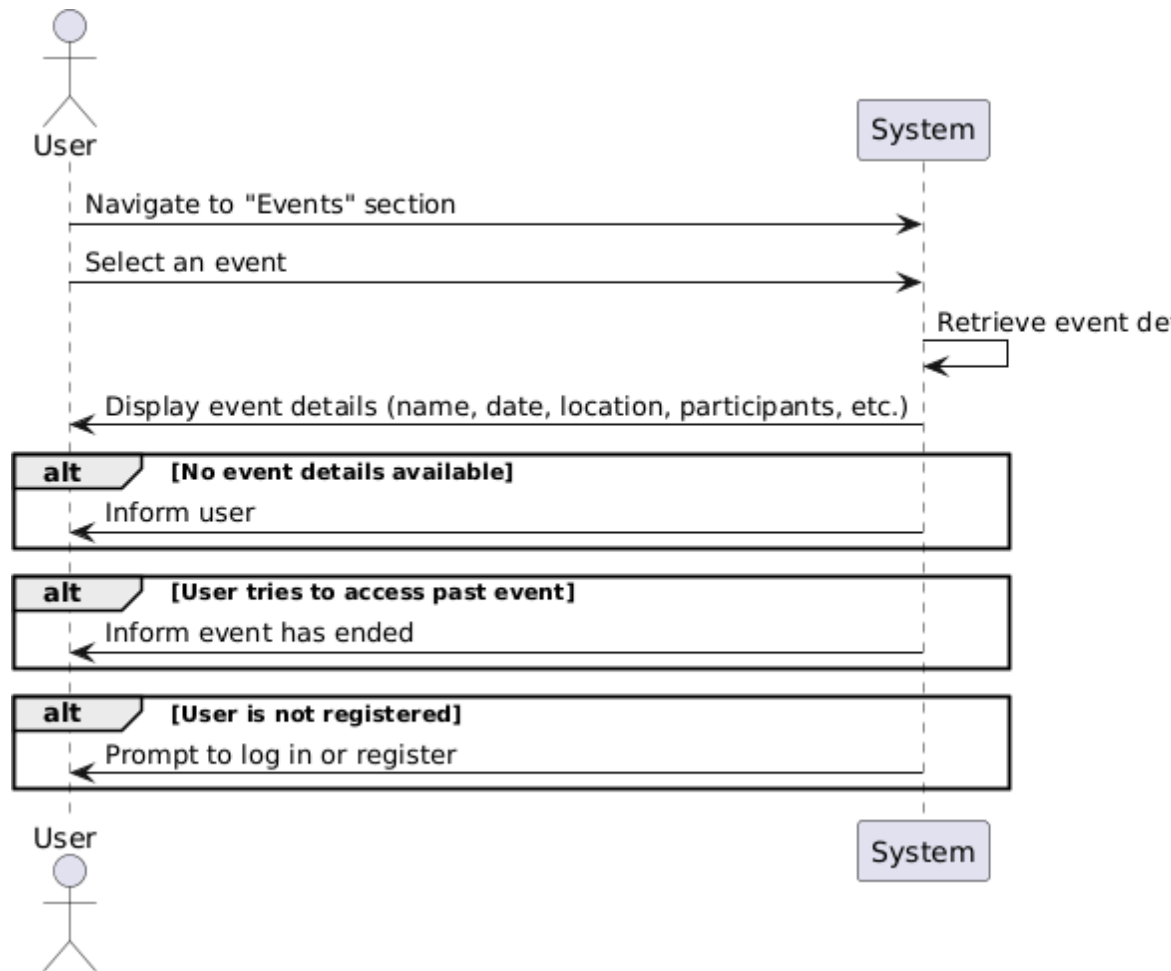
Use case 5:



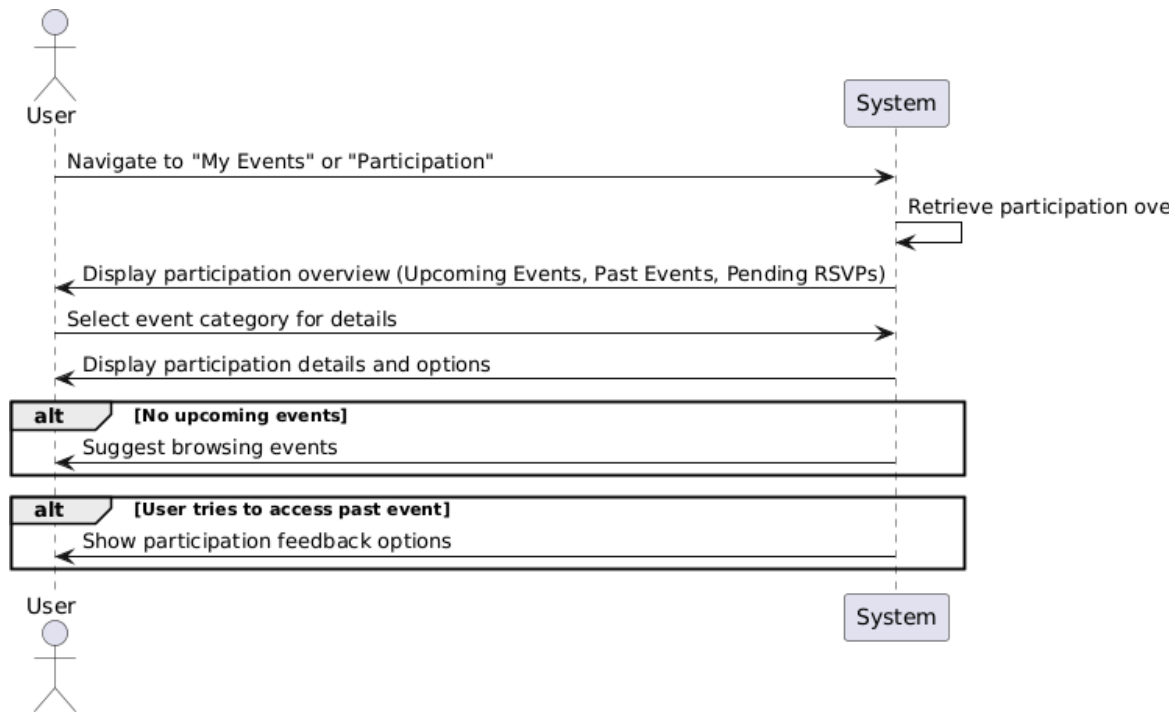
Use case 6:



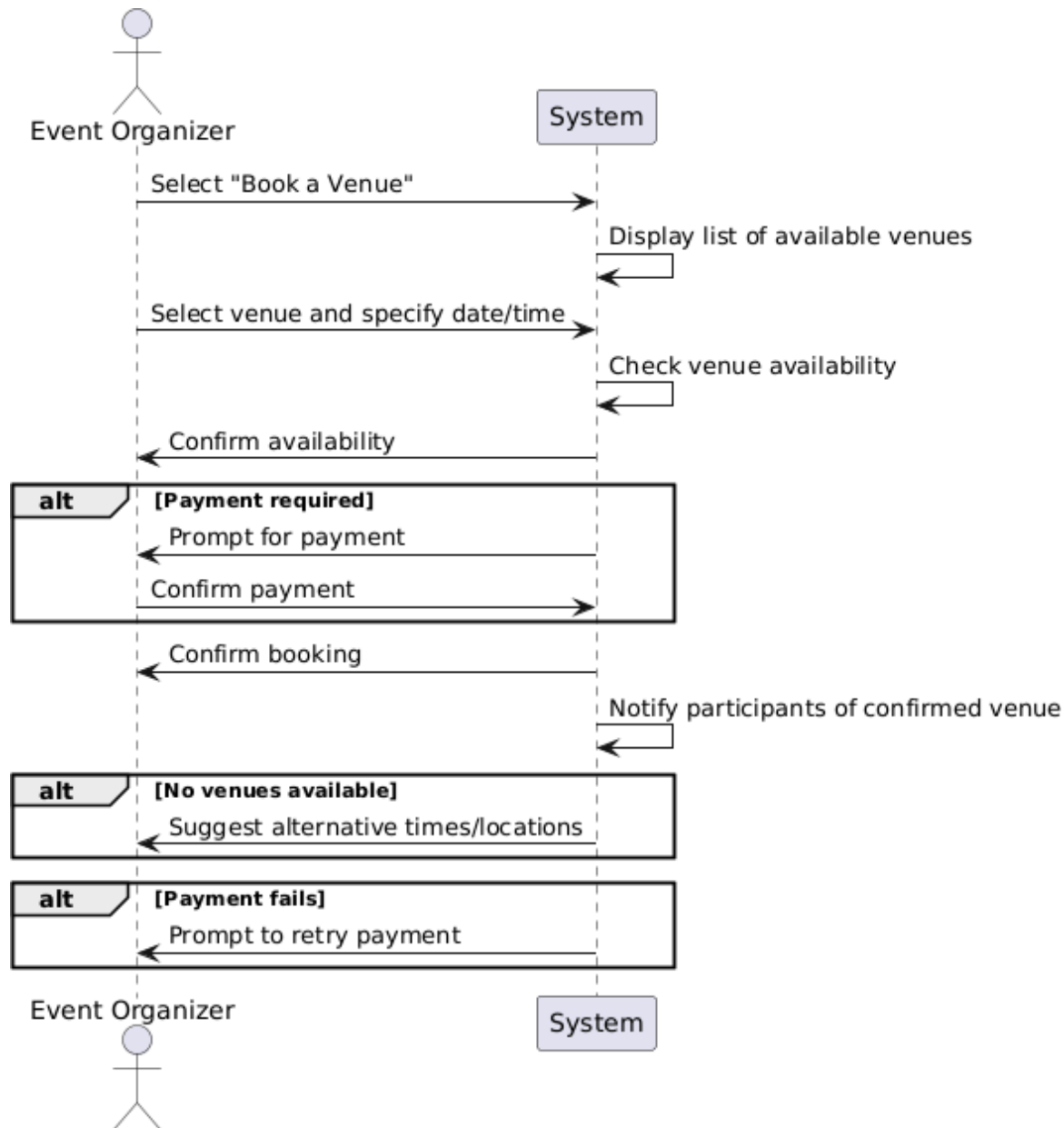
Use case 8:



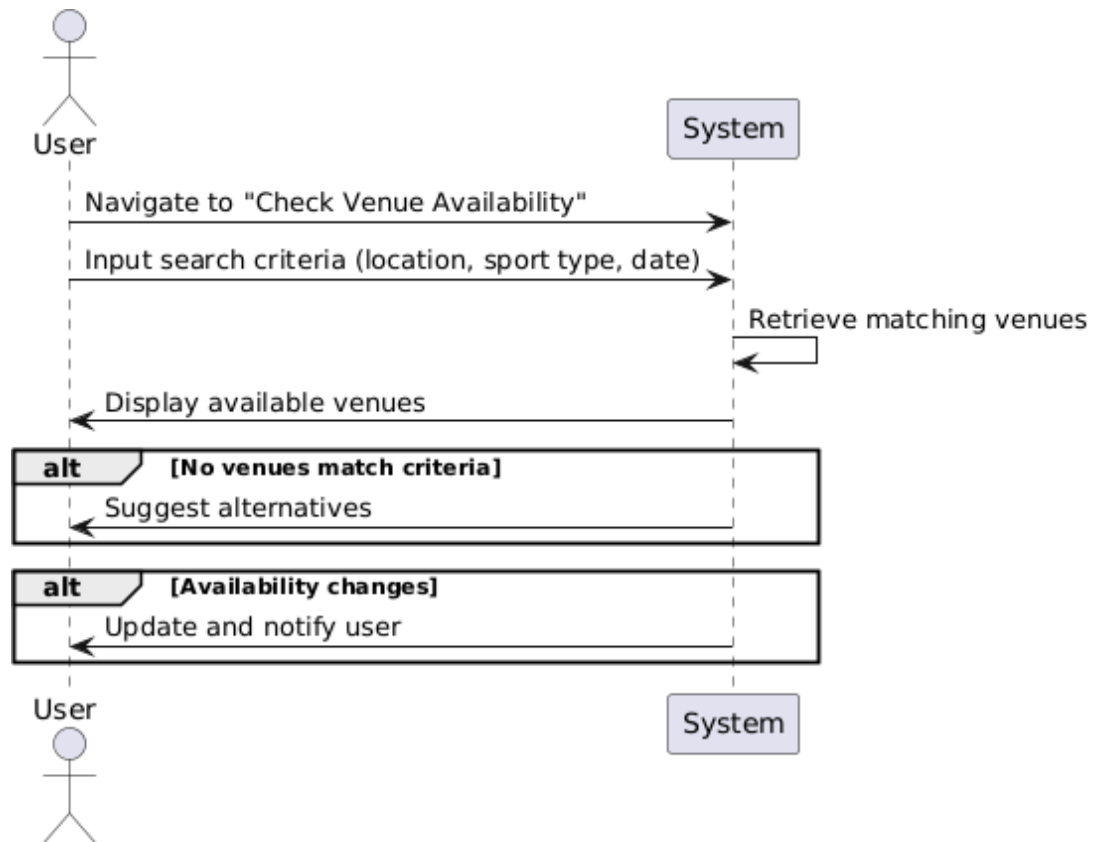
Use case 9:



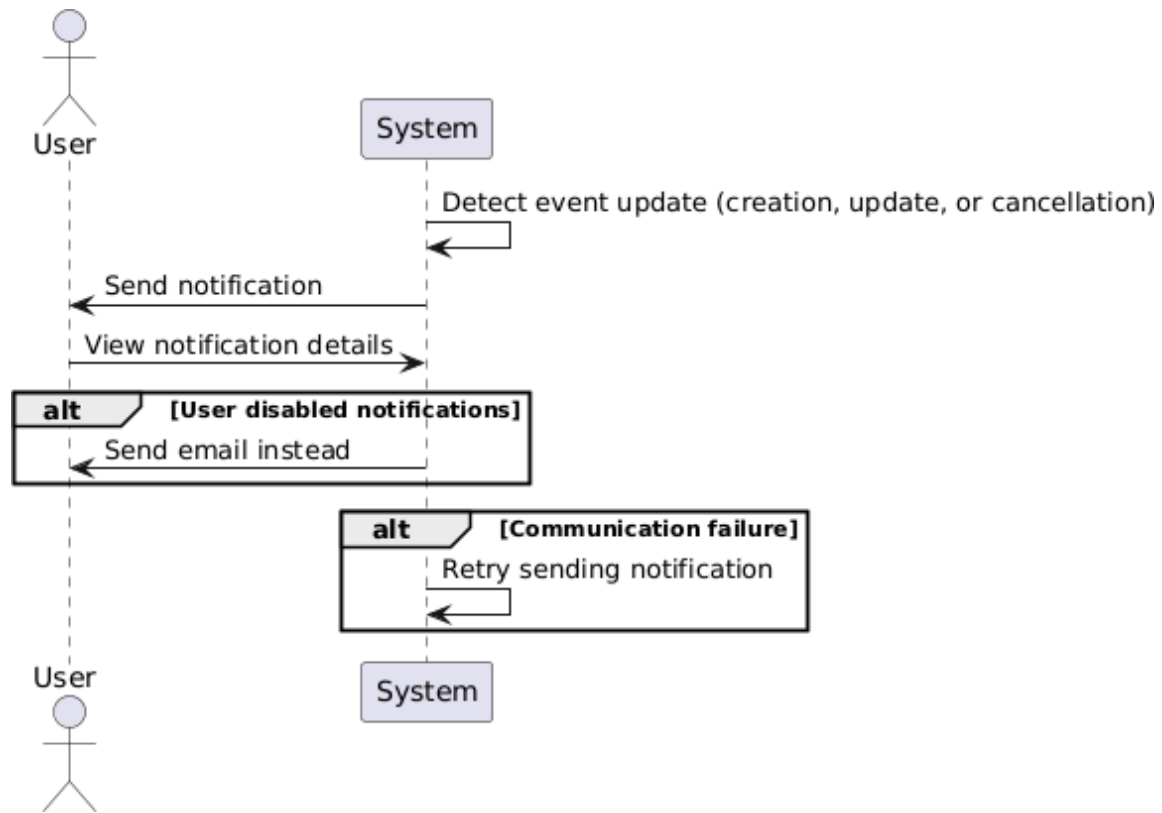
Use case 10:



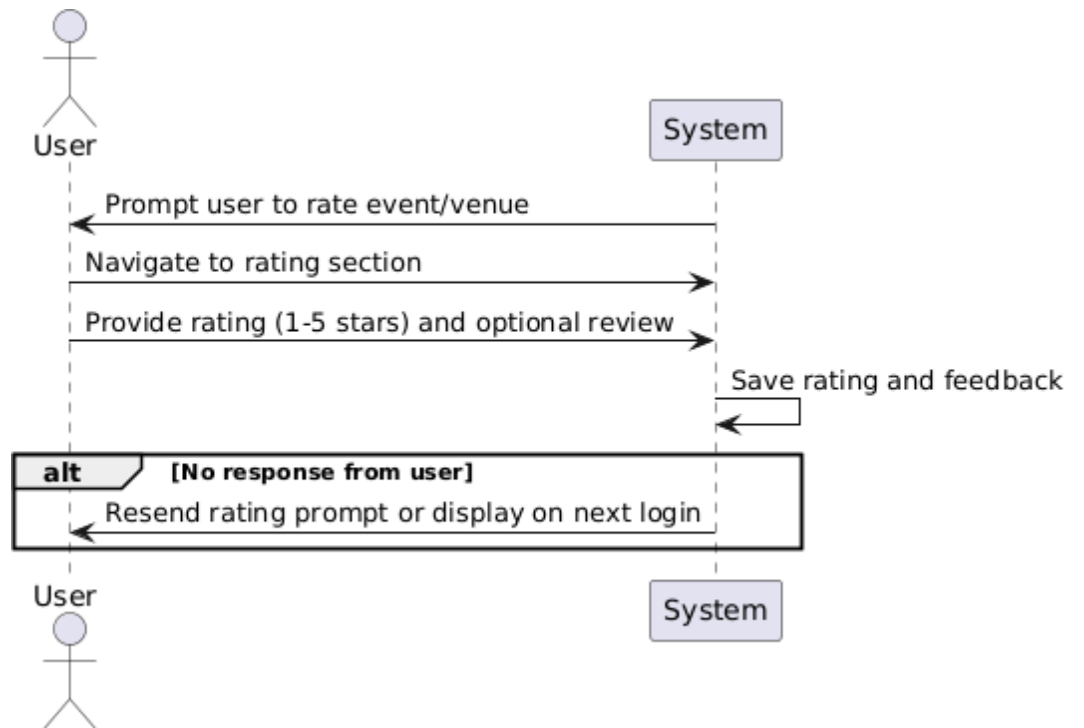
Use case 11:



Use case 12:

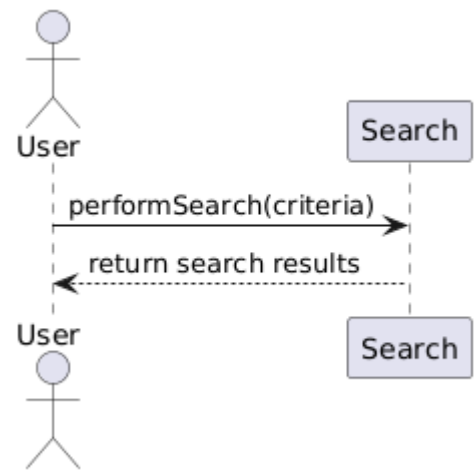


Use case 13:

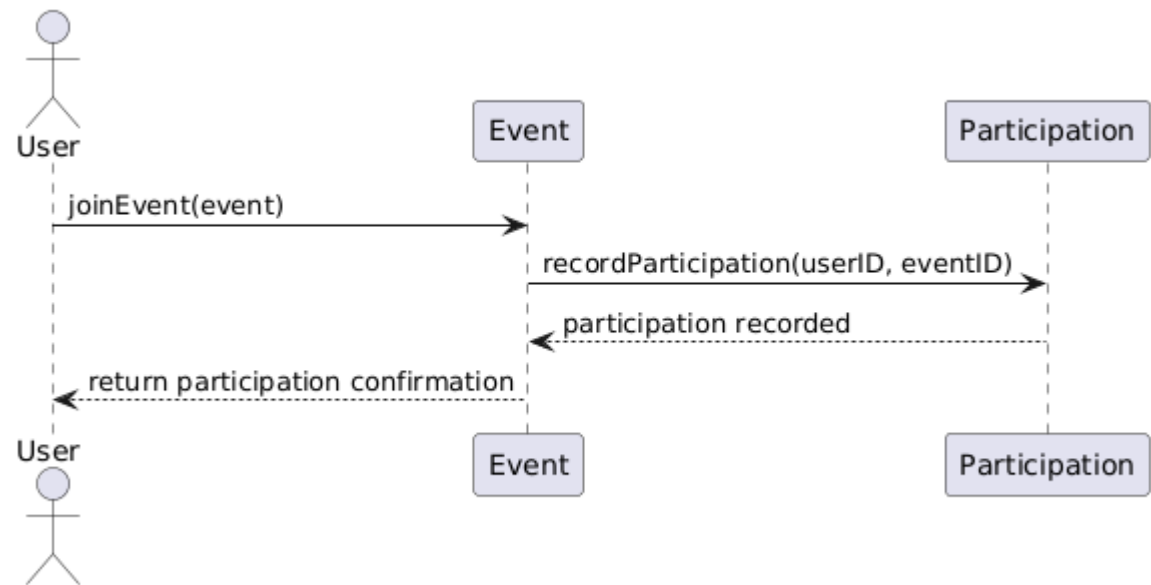


Sequence Diagram:

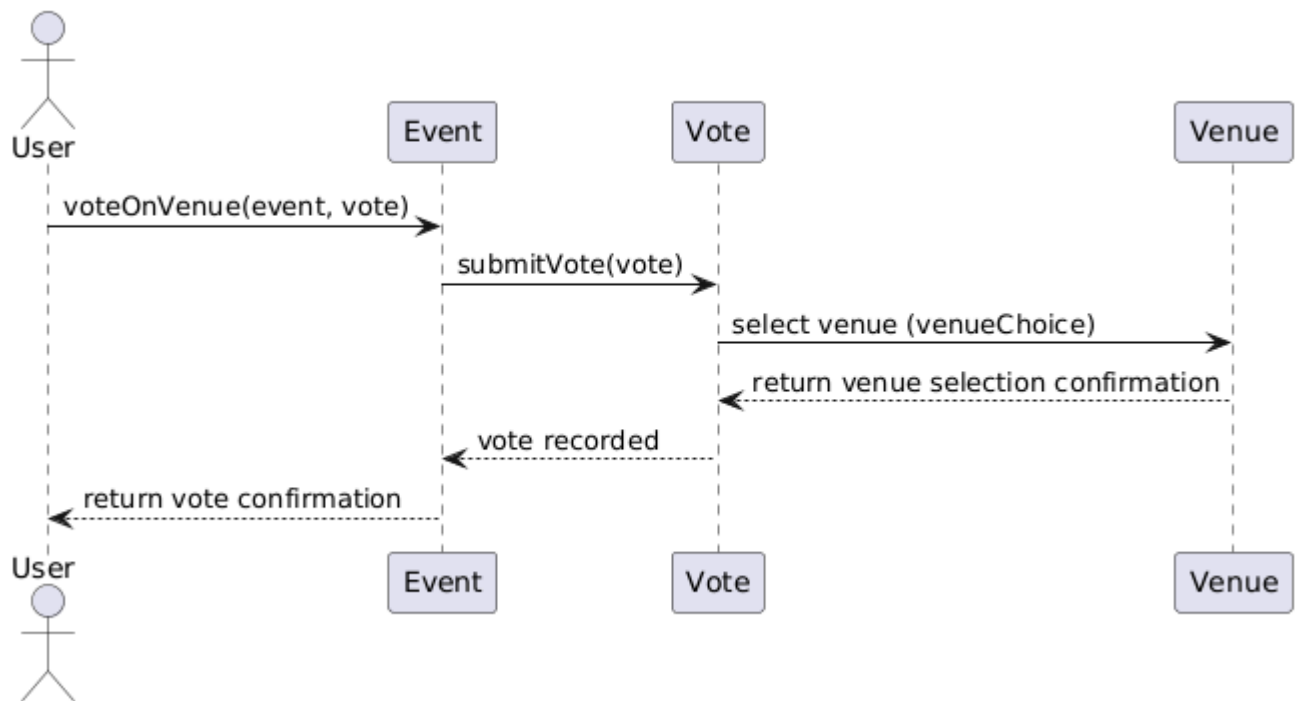
Use Case 1:



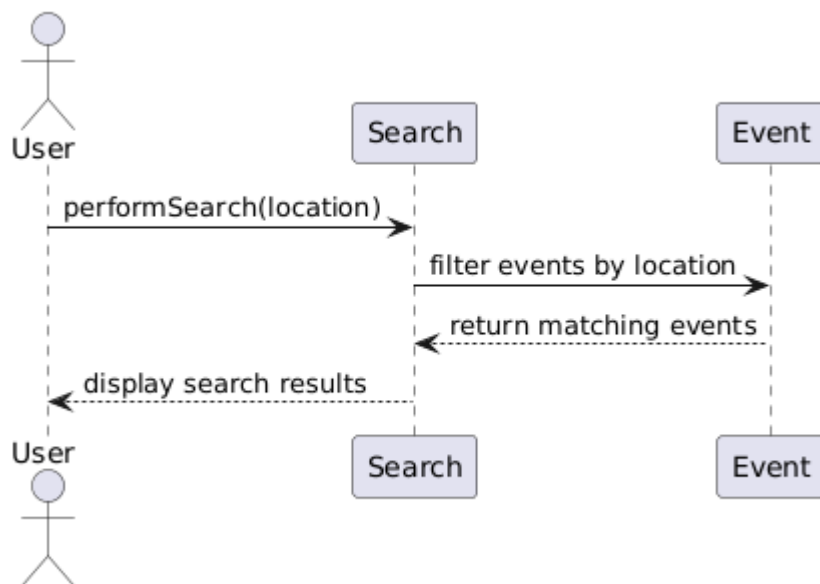
Use case 2:



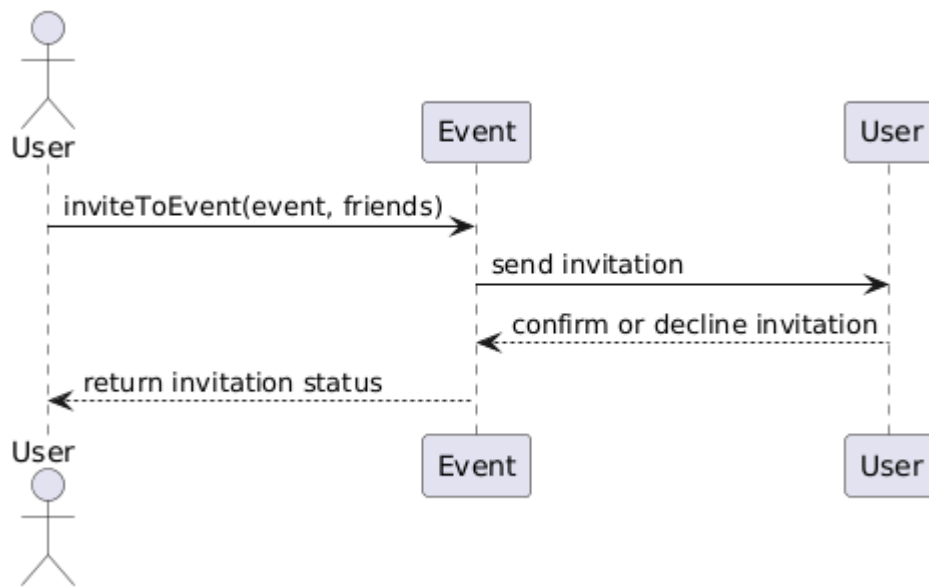
Use case 3:



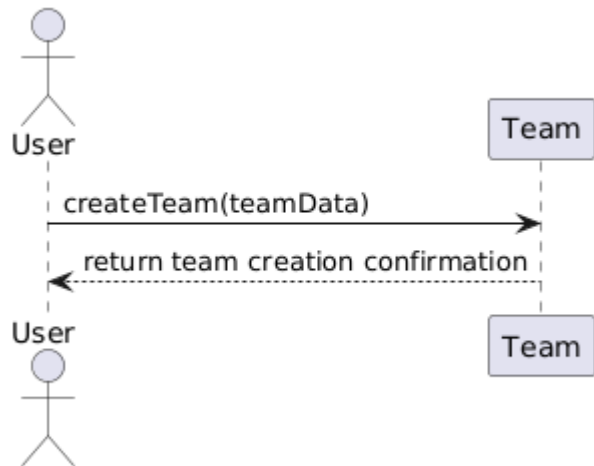
Use Case 4:



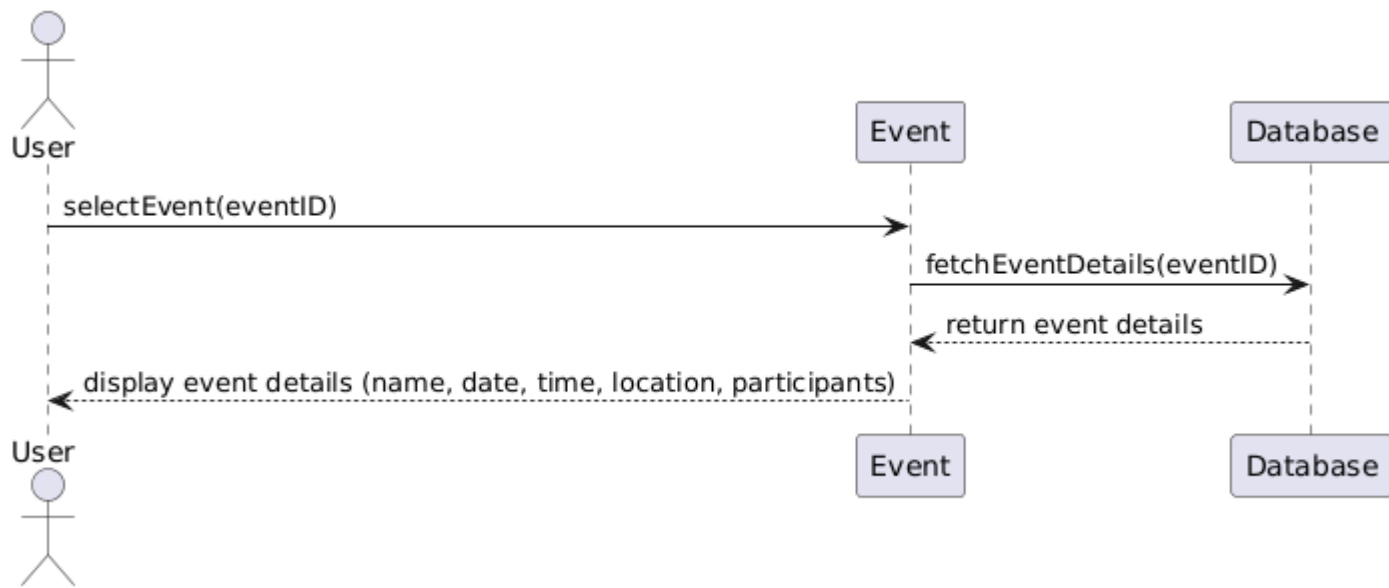
Use Case 5:



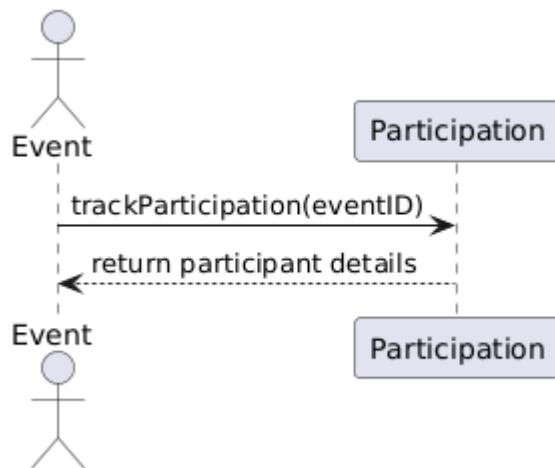
Use Case 6:



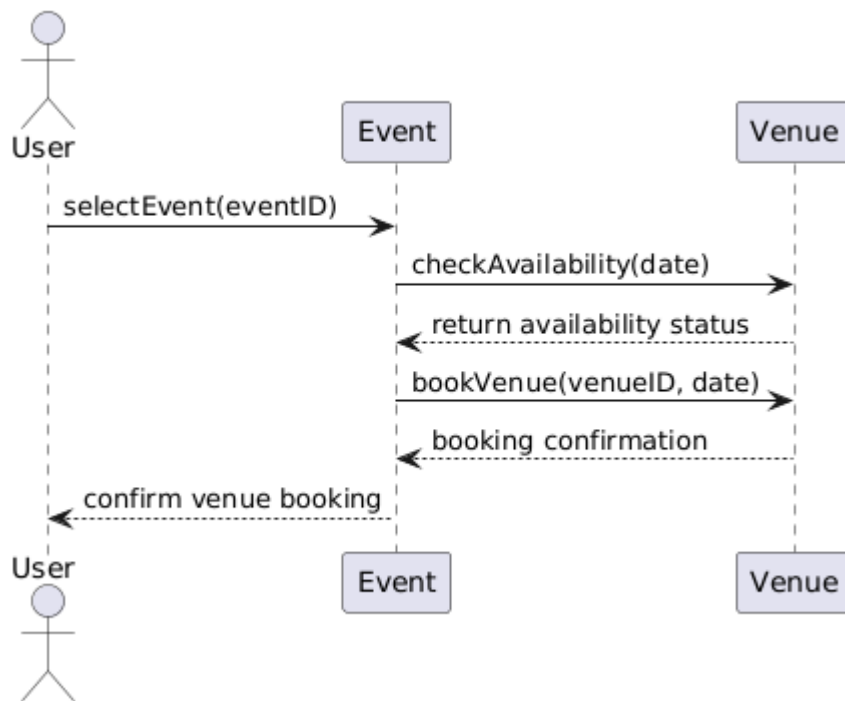
Use Case 7:



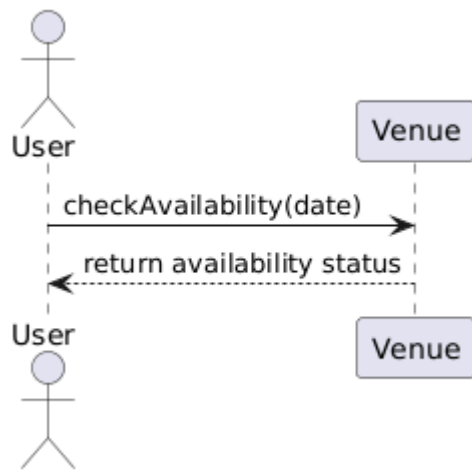
Use Case 8:



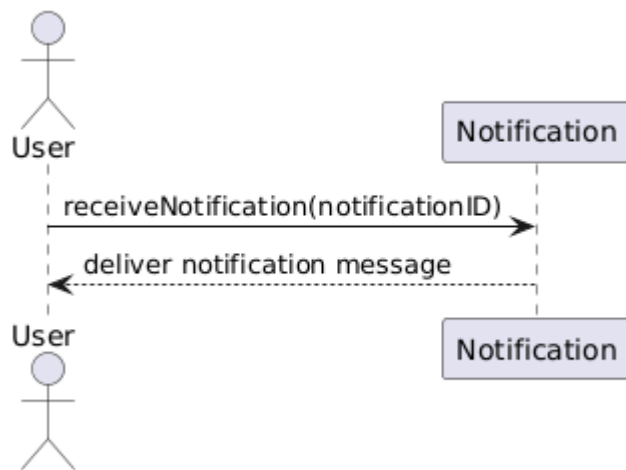
Use Case 9:



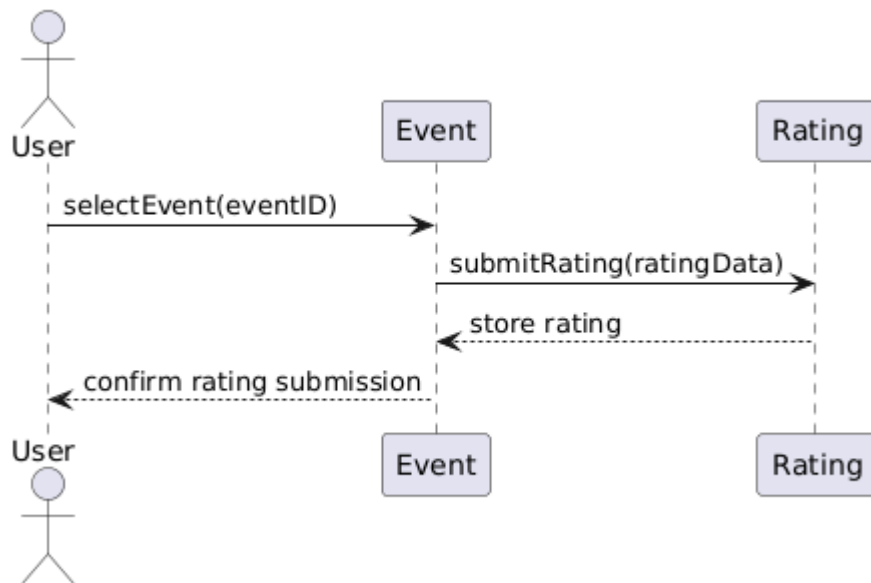
Use Case 10:



Use Case 11:



Use Case 12:



Class Diagram:

