

## Club Event Management System

---

[GITHUB LINK](#)

Group #9

Eddie Wang(YW6490)

Jerry Zhang(xz3956)

Scott He(th2904)

Table of Work

	Eddie Wang	Scott He	Jerry Zhang
Project Description			x
Uses Case Diagrams	x	x	x
Sequence Diagrams	x	x	
Class Diagrams		x	x
Implementation	x	x	x
Conclusion	x		

Description:

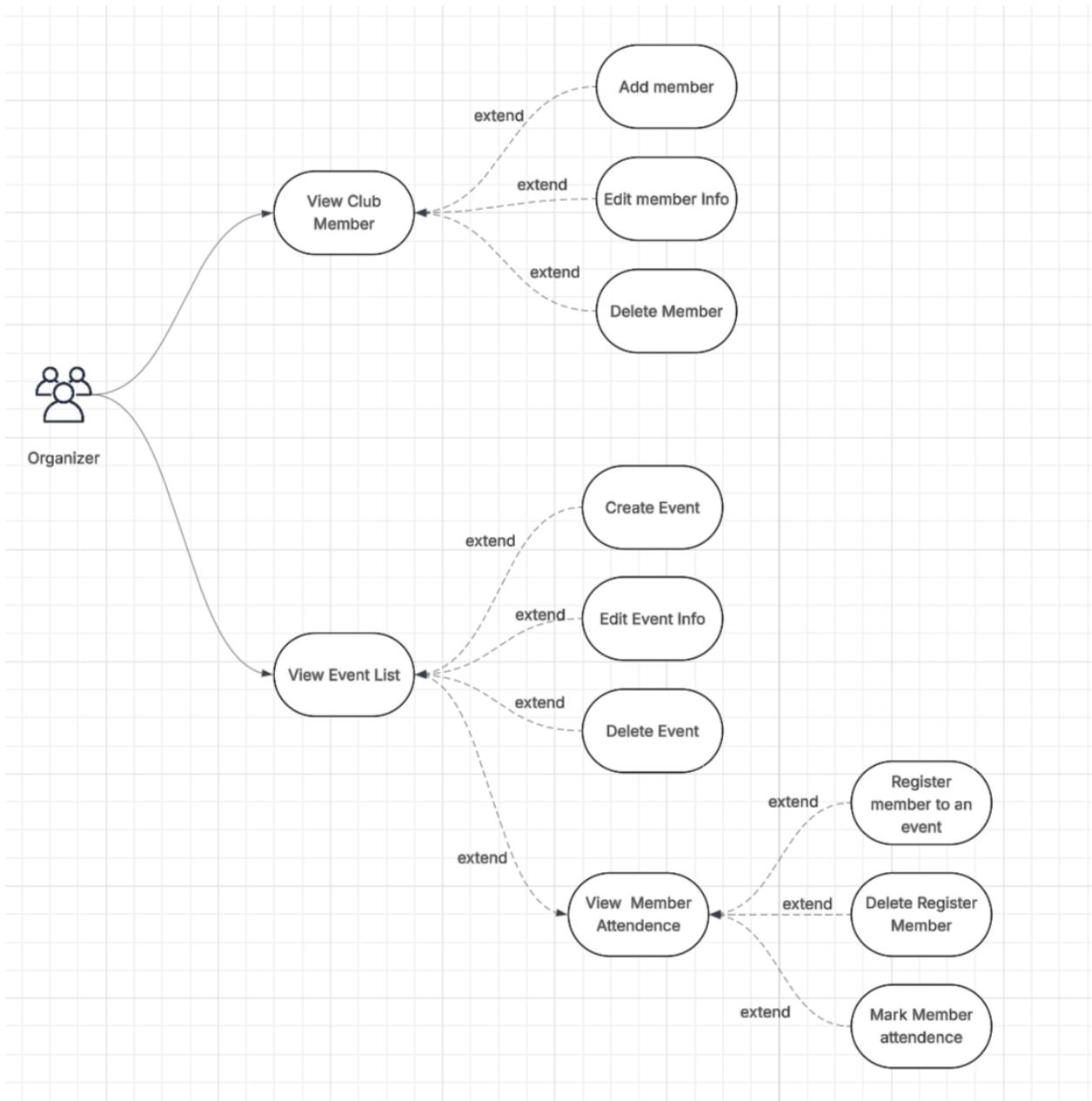
We want to create a campus event management system that allows the student club to manage events and attendees. Organizers are able to create and manage events, handle club members and registration, and also track attendance in this application. The entire system is designed to support student club activities by giving organizers a place to manage event details and overall tracking.

Core features:

1. Event Management: create, edit and delete events where each event has a title, time, date, location, description and capacity limits (organizers can update even information if details change)
2. Member Management: add, delete and edit members where each member has name, phone number, student ID and email. (the system keeps all member information in one place so organizers don't have to manage different spreadsheets for members)
3. Event registration: member can be registered to an event (prevent duplicate registration for the same event), this will ensure the sign up list are accurate and events do not go beyond maximum limits.,
4. attendance tracking, for each single event, mark and track which registered members actually attended
5. Event and attendance views, list upcoming events and view event details including the list of registered members and number of attendees

The Campus Event Management system will be implemented in Java using OOD and will create corresponding UML crafts. we plan to use simple GUI(Java swing) for the user interface

## Usecase Diagram:



Use case description:

UC Reference Name/Number: UC-01 View Club Member	
Overview	This use case enables the organizers to view the club member list and basic member information.
Related use cases	UC-02 Add Member (members appear after creation) UC-03 Edit Member Info (edit is initiated after selecting a member) UC-04 Delete Member (delete is initiated after selecting a member) UC-10 Register Member to an Event (organizer may select a member before registering).
Actors	Organizer

UC Reference Name/Number: UC-02 Add members	
Overview	This use case enables the organizers to add a new club member into the system.
Related use cases	UC-01 View Club Member (new member becomes visible) UC-03 Edit Member Info (modify later) UC-04 Delete Member (remove member) UC-09 View Member Attendance (existing members appear in attendance lists) UC-10 Register Member to an Event (member must exist before registering) UC-11 Delete Registered Member (registered member must exist) UC-12 Mark Member Attendance (attendance can only be marked for an existing member).
Actors	Organizer

UC Reference Name/Number: UC-03 Edit member Info	
Overview	This use case enables the organizers to edit an existing member's information.
Related use cases	UC-01 View Club Member (select member to edit) UC-02 Add Member (member must exist first)

	UC-04 Delete Member (alternative action)
Actors	Organizer

UC Reference Name/Number: UC-04 Delete Member	
Overview	This use case enables the organizers to delete a member record from the system.
Related use cases	UC-01 View Club Member (select member to delete) UC-02 Add Member (member lifecycle) UC-03 Edit Member Info (alternative action) UC-10 Register Member to an Event (preventing deletion if member is registered) UC-11 Delete Registered Member (needed before deletion if registrations must be cleared)
Actors	Organizer

UC Reference Name/Number: UC-05 View Event List	
Overview	This use case enables the organizers to view the list of events and can access event-related operations
Related use cases	UC-06 Create Event (events appear after creation) UC-07 Edit Event Info (edit can be initiated after selecting an event) UC-08 Delete Event (delete can be initiated after selecting an event) UC-09 View Member Attendance (attendance is viewed per event) UC-10 Register Member to an Event (registration requires choosing an event)
Actors	Organizer

UC Reference Name/Number: UC-06 Create Event	
Overview	This use case enables the organizers to create a new event.

Related use cases	UC-05 View Event List (new event becomes visible) UC-07 Edit Event Info (modify later) UC-08 Delete Event (remove event) UC-09 View Member Attendance (attendance is managed per event) UC-10 Register Member to an Event (event must exist before registering members) UC-11 Delete Registered Member (requires an existing event) UC-12 Mark Member Attendance (requires an existing event)
Actors	Organizer

UC Reference Name/Number: UC-07 Edit Event Info	
Overview	This use case enables the organizer to edit an existing event's details.
Related use cases	UC-05 View Event List (select event to edit) UC-06 Create Event (event must exist first) UC-08 Delete Event (alternative action) UC-09 View Member Attendance (attendance depends on the event)
Actors	Organizer

UC Reference Name/Number: UC-08 Delete Event	
Overview	This use case enables the organizers to delete an event from the system.
Related use cases	UC-05 View Event List (select event to delete) UC-06 Create Event (event lifecycle) UC-07 Edit Event Info (alternative action) UC-09 View Member Attendance (attendance records will be removed) UC-10 Register Member to an Event (registrations depend on the event) UC-11 Delete Registered Member (may be needed before deletion if registrations must be cleared)

	UC-12 Mark Member Attendance (attendance depends on the event)
Actors	Organizer

UC Reference Name/Number: UC-09 View Member Attendance	
Overview	This use case enables the organizers to view registration/attendance information for a selected event.
Related use cases	UC-05 View Event List (choose event) UC-06 Create Event (event must exist) UC-02 Add Member (member must exist) UC-10 Register Member to an Event (adds members to the list) UC-11 Delete Registered Member (removes members from the list) UC-12 Mark Member Attendance (updates attendance status)
Actors	Organizer

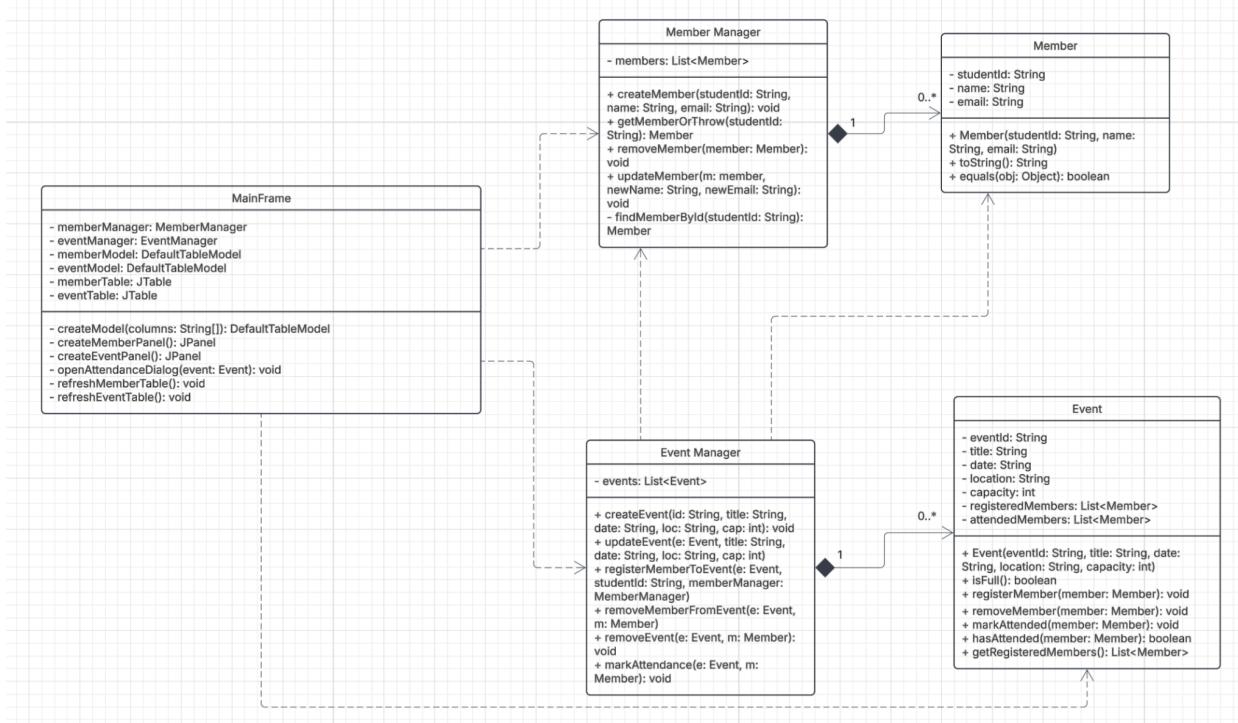
UC Reference Name/Number: UC-10 Register member to an event	
Overview	This use case enables the organizers to register a member for a chosen event.
Related use cases	UC-01 View Club Member (select member) UC-02 Add Member (member must exist) UC-05 View Event List (select event) UC-06 Create Event (event must exist) UC-09 View Member Attendance (registration appears in this use case) UC-11 Delete Registered Member (undo registration), UC-12 Mark Member Attendance (attendance can be marked after registration)
Actors	Organizer

UC Reference Name/Number: UC-11 Delete Register Member

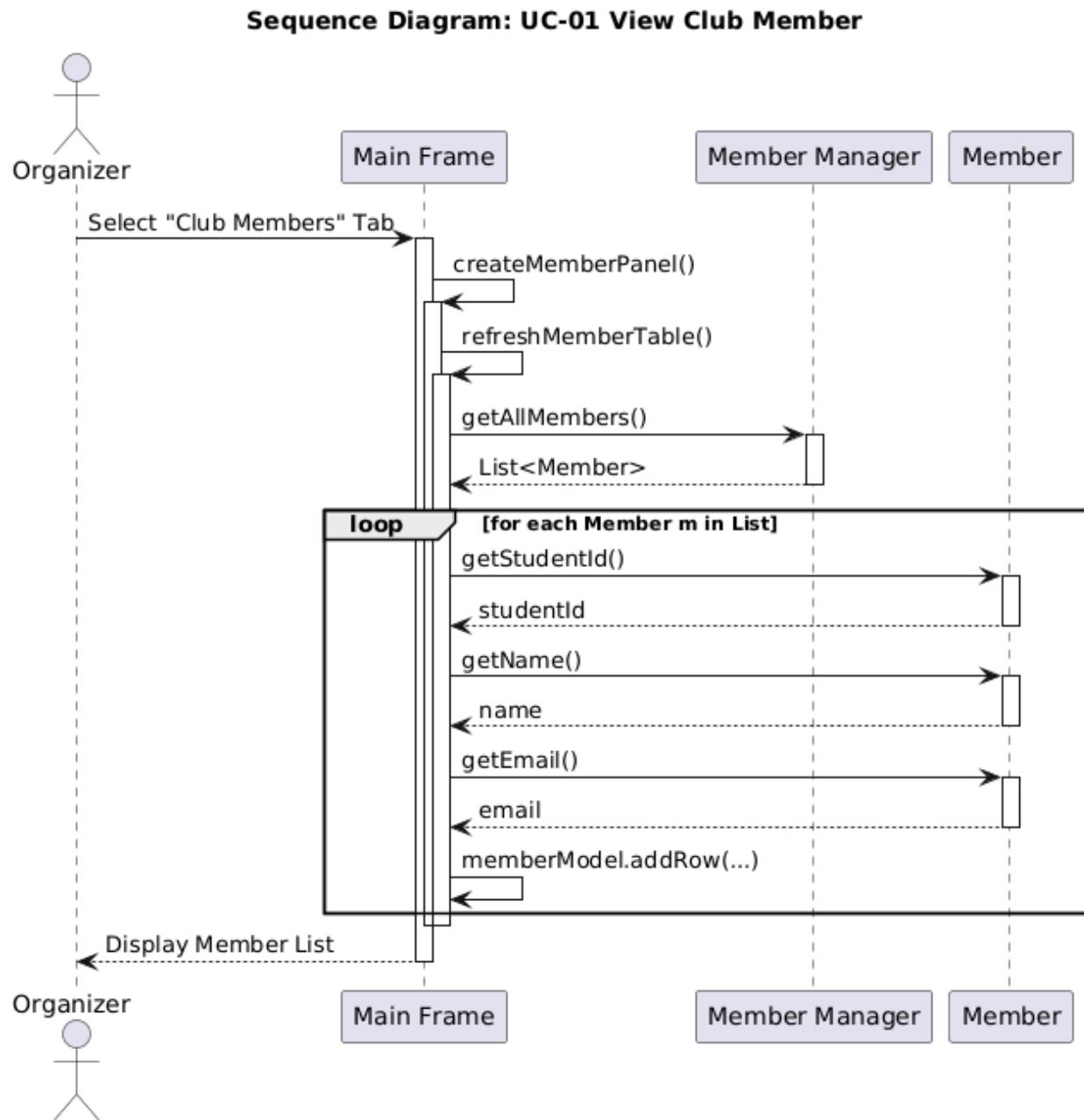
Overview	The use case enables the organizer to remove a member from an event's registration list.
Related use cases	UC-02 Add Member (member must exist) UC-06 Create Event (event must exist) UC-09 View Member Attendance (select registration to remove) UC-10 Register Member to an Event (member must be registered first for removal) UC-12 Mark Member Attendance (attendance may no longer be applicable after removal)
Actors	Organizer

UC Reference Name/Number: UC-12 Mark Attendance	
Overview	The use case enables the organizers to mark whether registered members attended the event.
Related use cases	UC-02 Add Member (member must exist) UC-06 Create Event (event must exist) UC-09 View Member Attendance (marking attendance happens within attendance list) UC-10 Register Member to an Event (a member must be registered first) UC-11 Delete Registered Member (corrective action if wrongly registered/recording).
Actors	Organizer

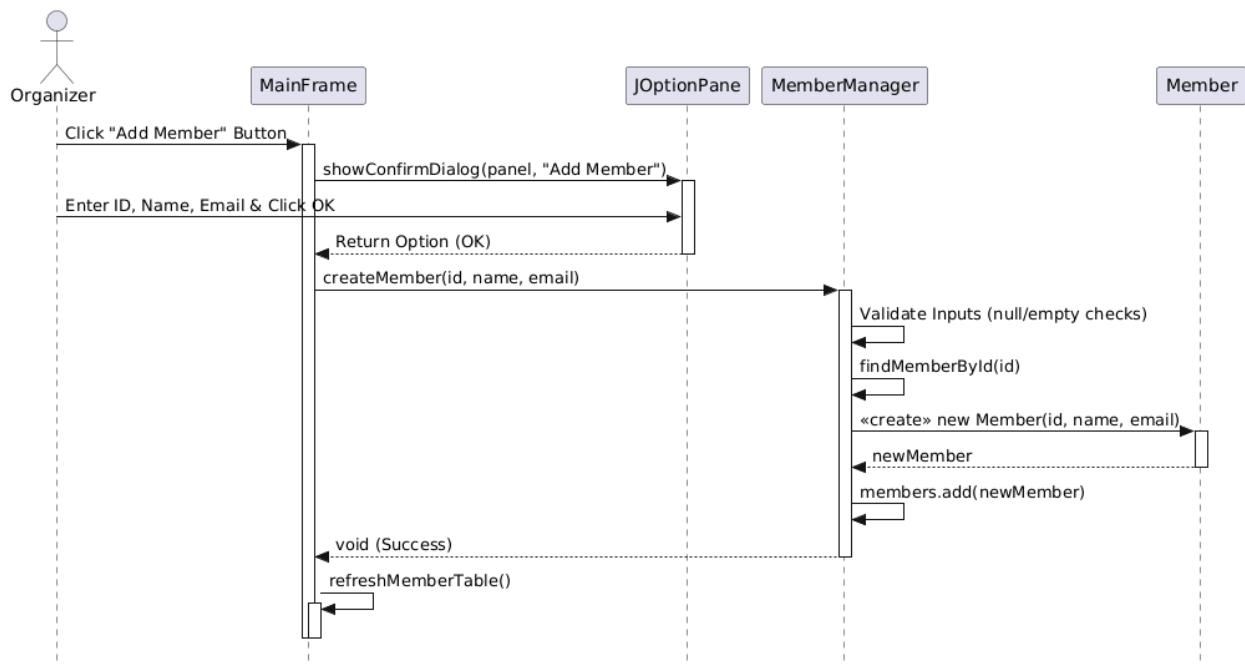
## Class Diagram:



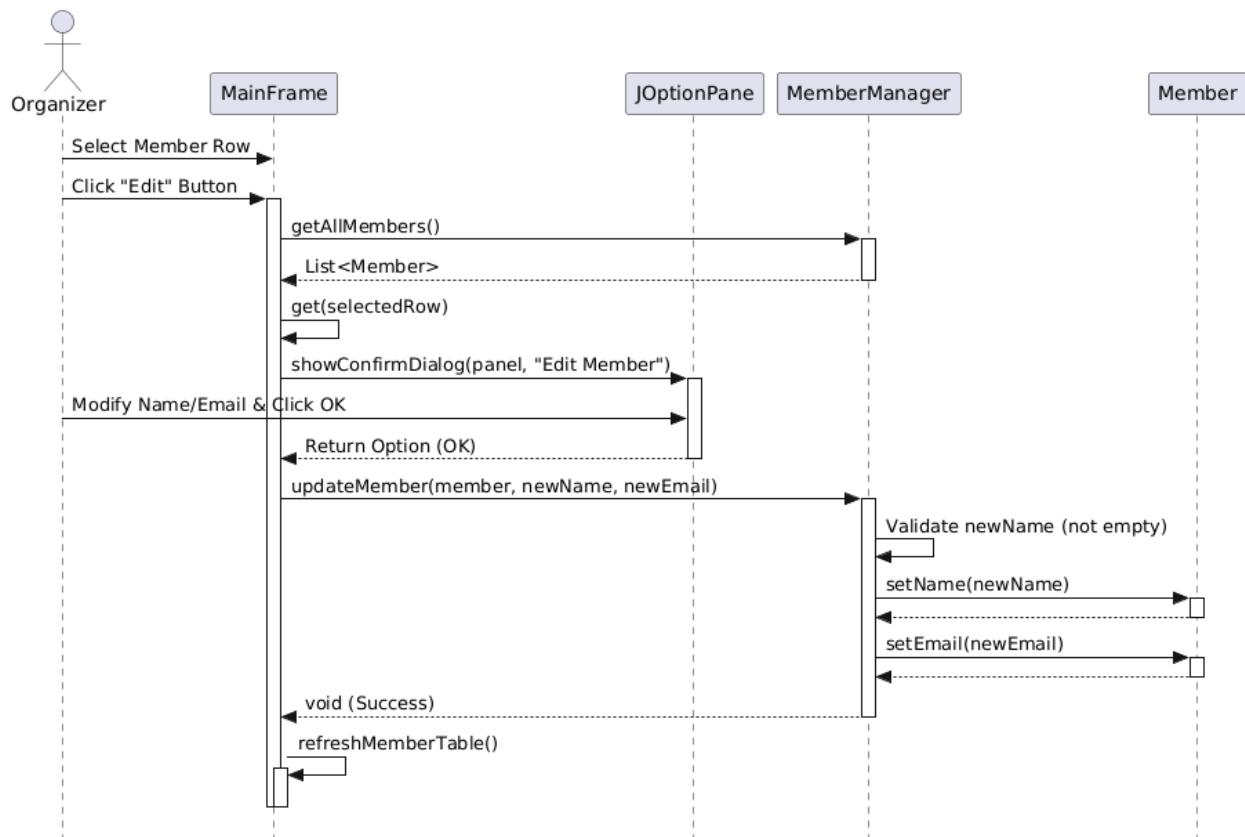
Sequence Diagrams:



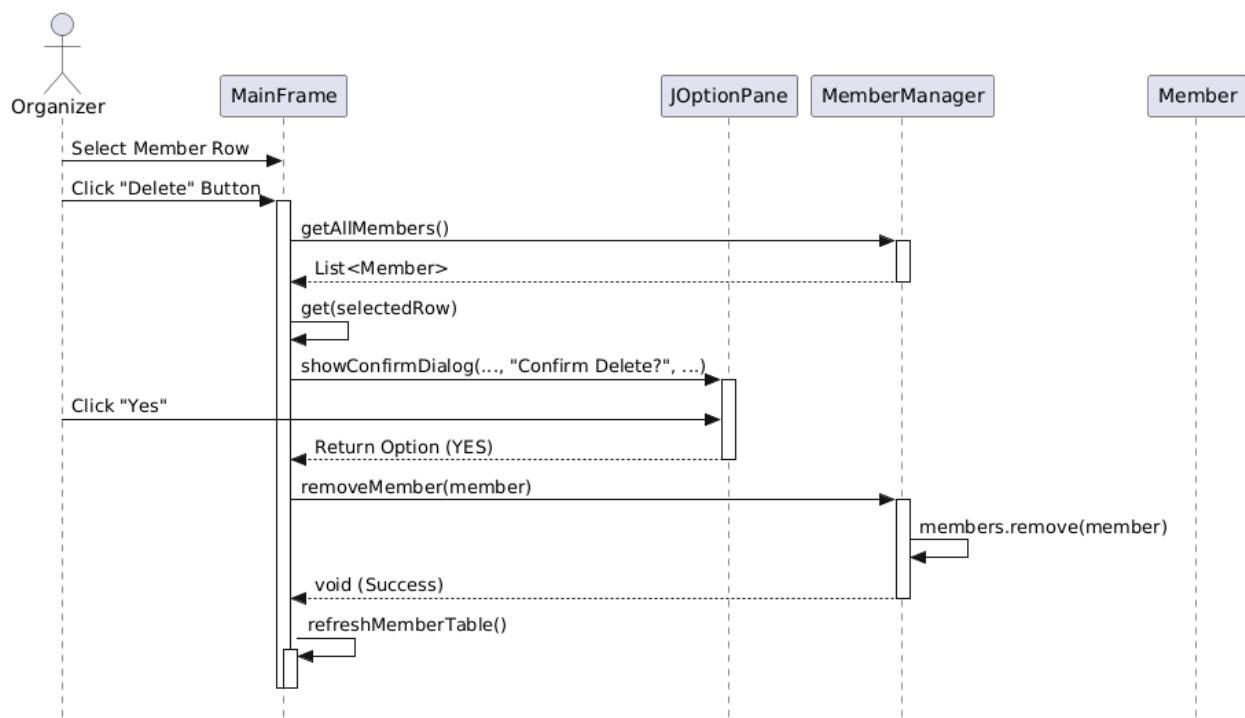
### Sequence Diagram: UC-02 Add Members



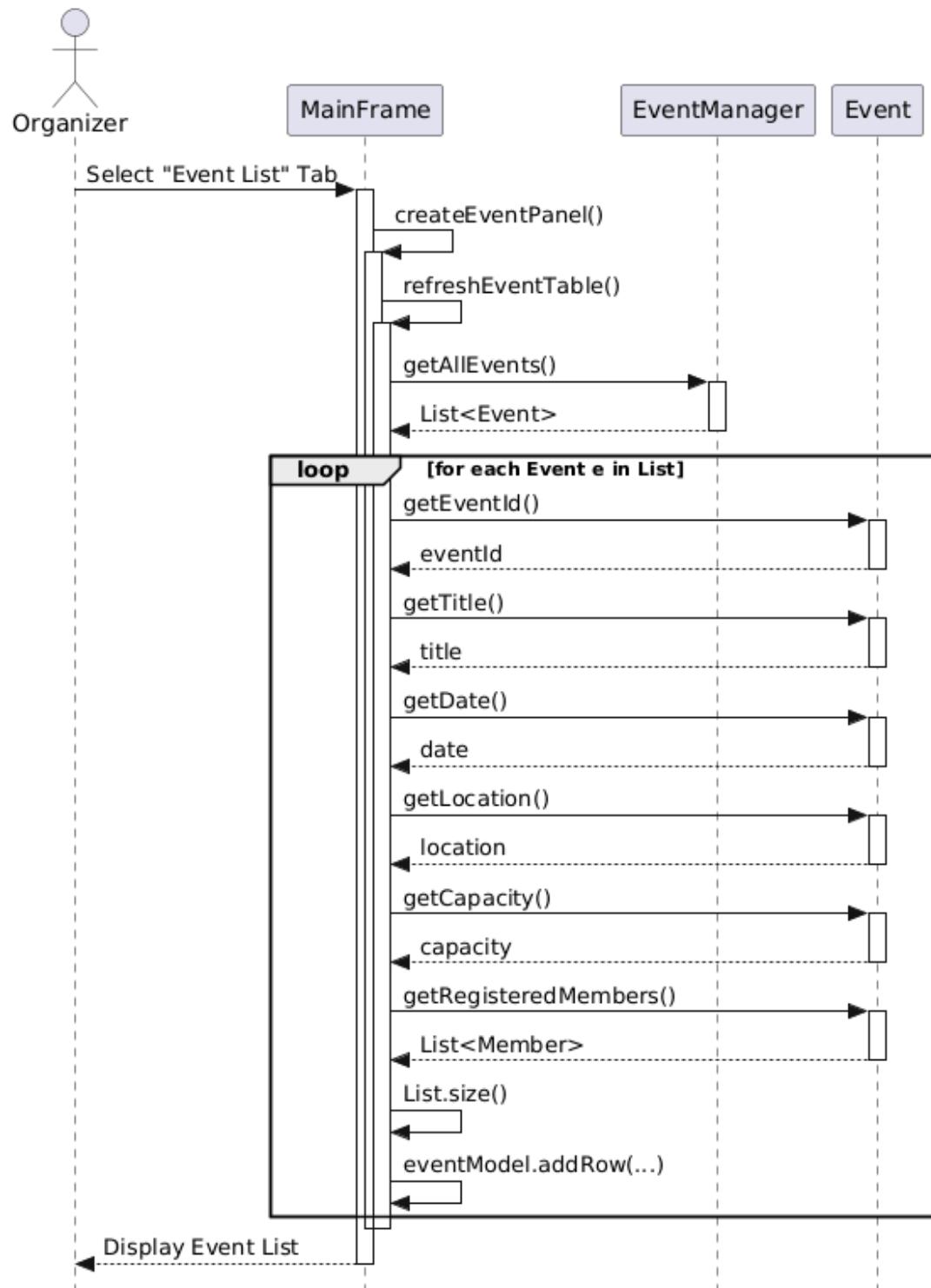
### Sequence Diagram: UC-03 Edit Member Info



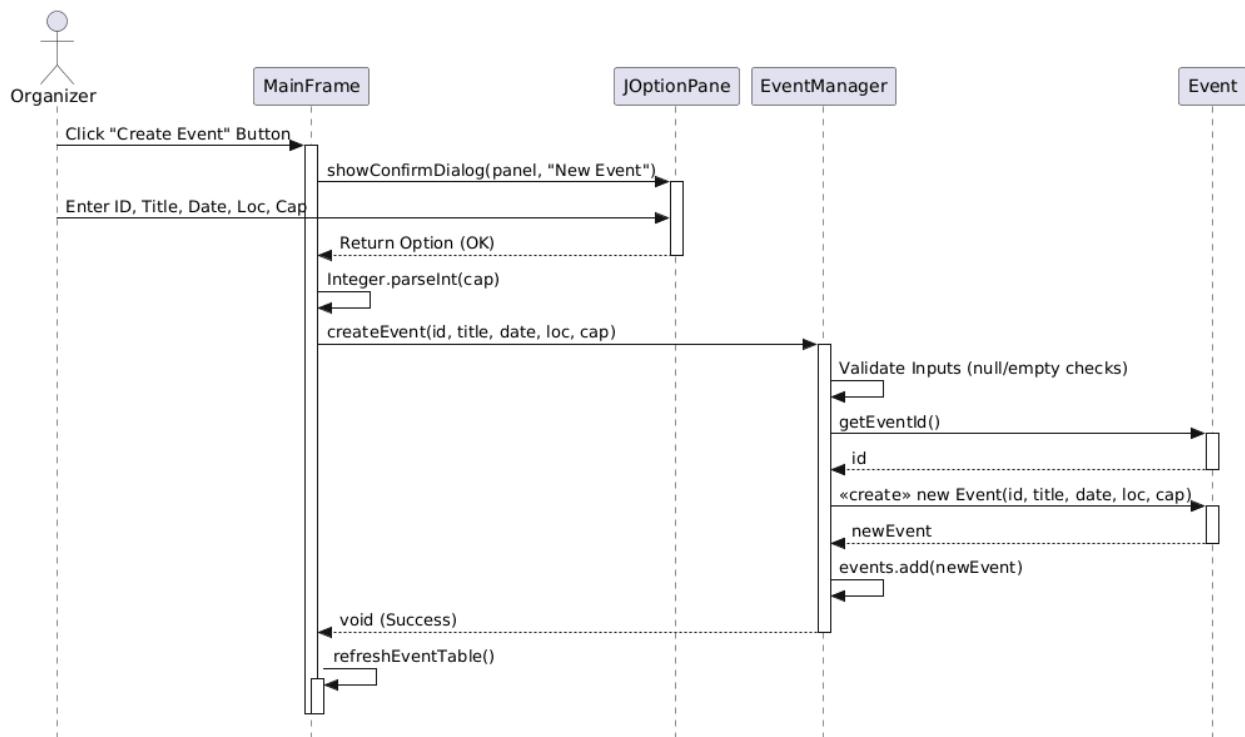
### Sequence Diagram: UC-04 Delete Member



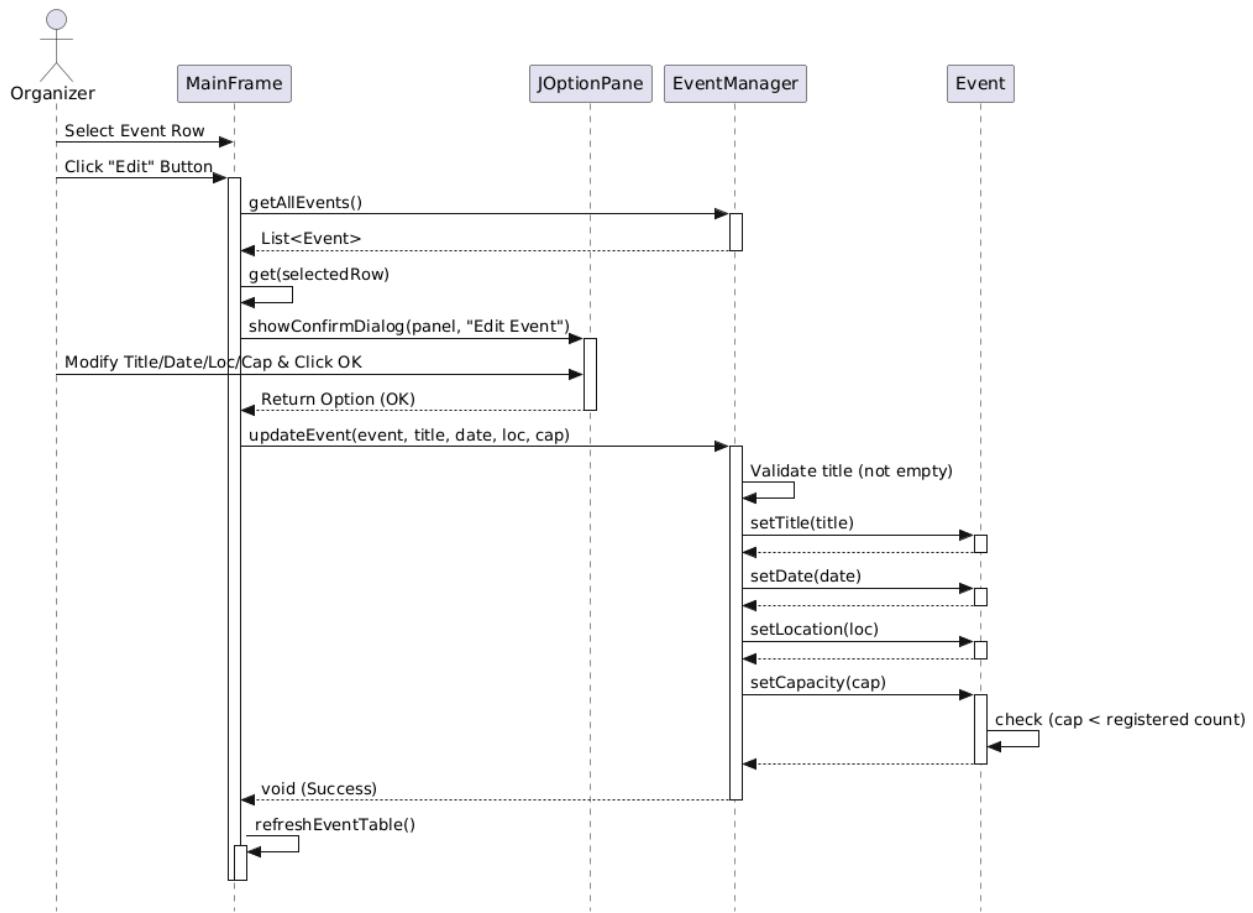
### Sequence Diagram: UC-05 View Event List



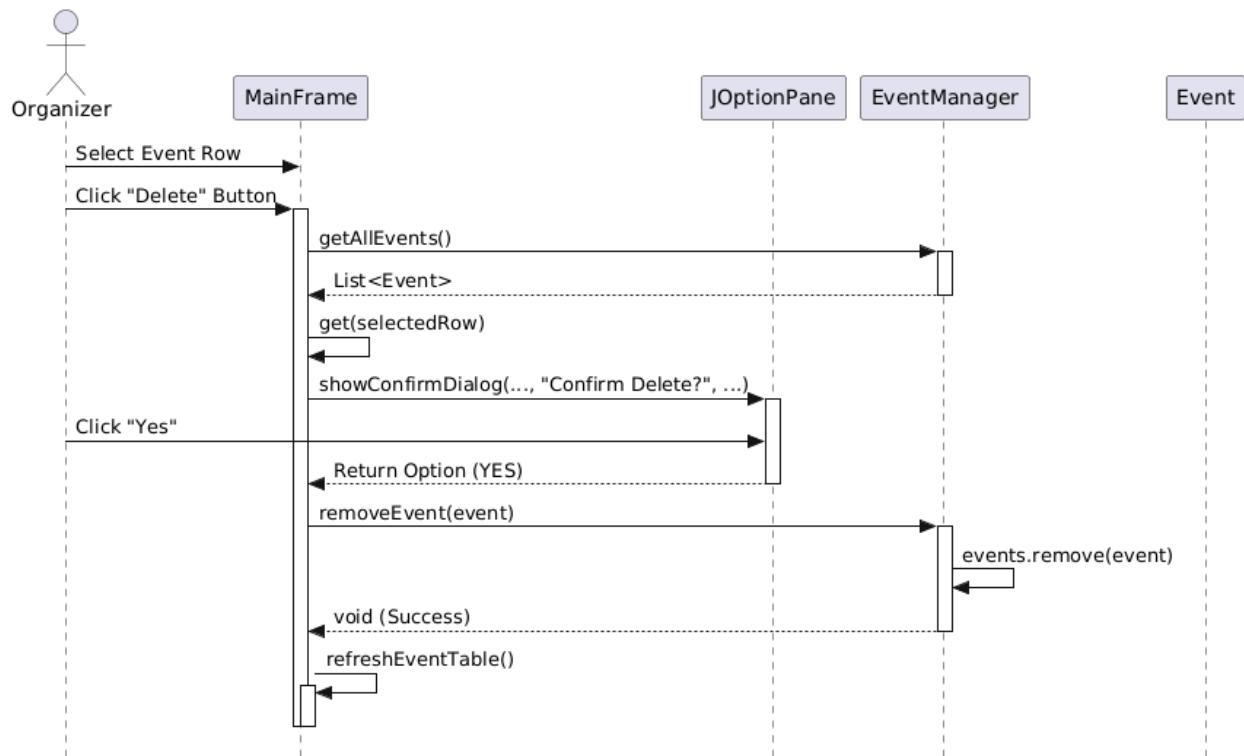
**Sequence Diagram: UC-06 Create Event**



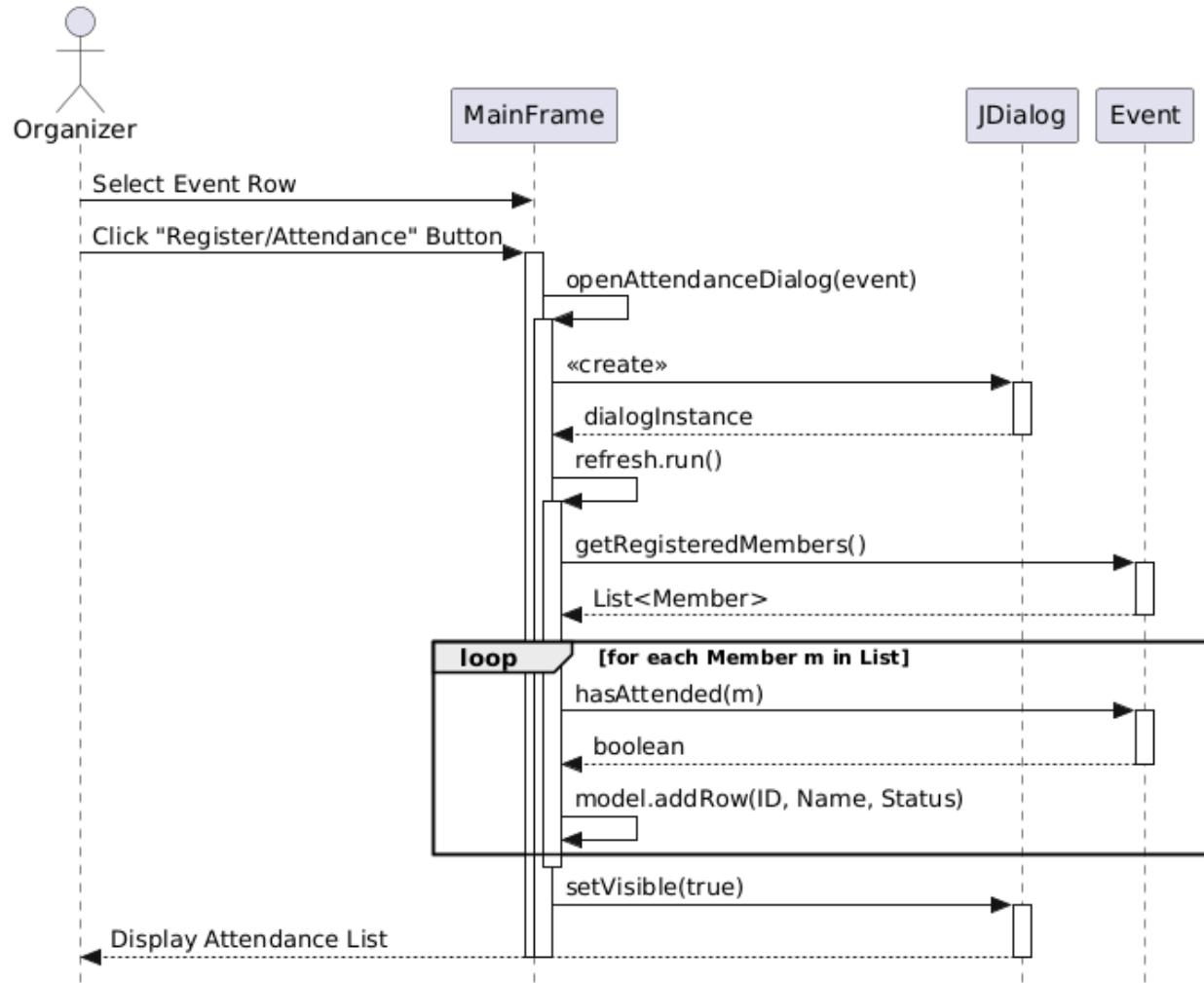
**Sequence Diagram: UC-07 Edit Event Info**



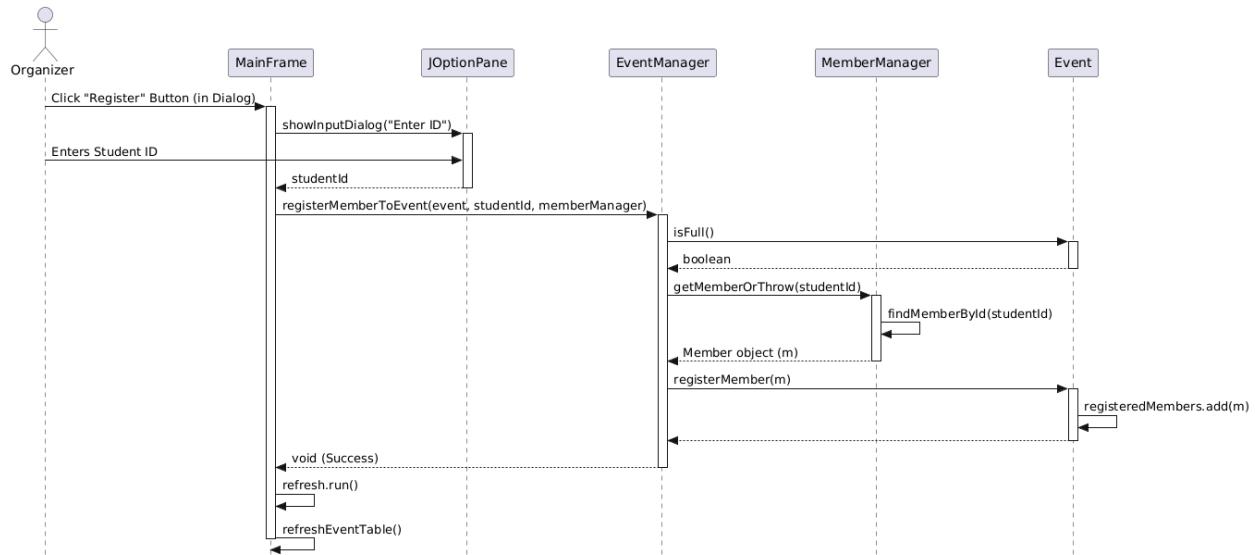
### Sequence Diagram: UC-08 Delete Event



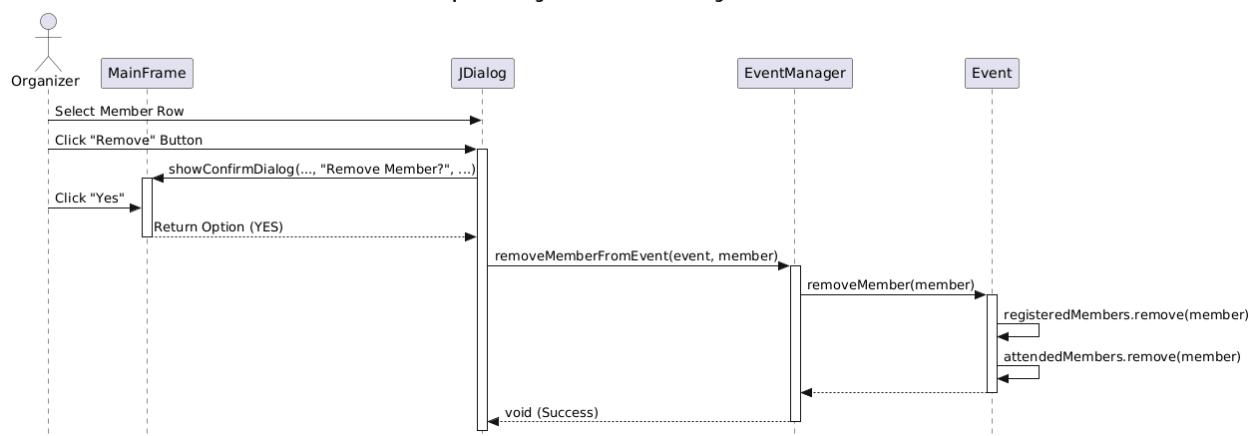
### Sequence Diagram: UC-09 View Member Attendance



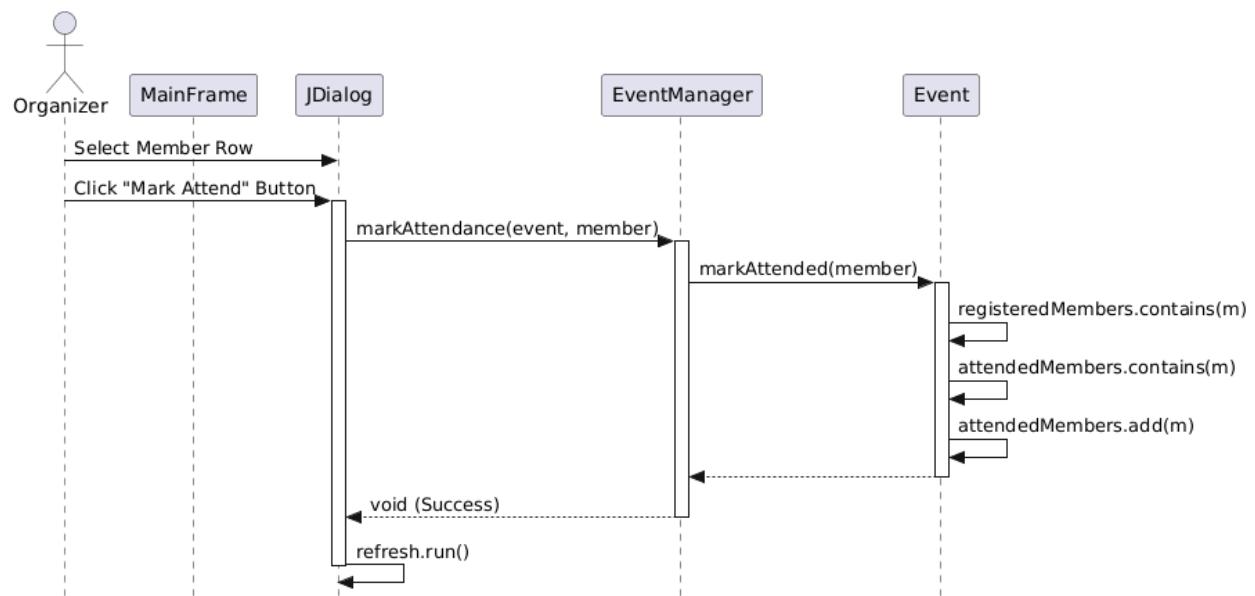
### Sequence Diagram: UC-10 Register Member to an Event



**Sequence Diagram: UC-11 Delete Registered Member**



**Sequence Diagram: UC-12 Mark Member Attendance**



## Conclusion:

We successfully developed a Campus Event Management System in Java that effectively handles the core lifecycle of club organization. By strictly separating our data entities from the management logic and user interface, we achieved a modular design that supports all twelve planned use cases, from basic member and event CRUD operations to complex registration and attendance tracking. The system ensures data integrity through real-time capacity checks and validation, providing organizers with a responsive tool for managing student engagement.

However, making the two "Managers" talk to each other was confusing at first. Since MemberManager holds the students and EventManager holds the events, we had to be careful when registering a student. We had to ensure the system didn't just take a string ID, but actually looked up the real Member object to store it in the event list, in the meantime handling different conditions to make sure our input id holds. Moving forward, we would need to implement a database that supports remote storage. This way we could actually put our product in use. Additionally, implementing stricter input validation, particularly for dates, would further enhance the system's robustness and prevent user errors.