Member Calendar Synchronization

Client (Society Member) — Event Manager — Calendar Generator — Database Connector — Database

- requestCalendarSync(societyID)
- getUpcomingEvents(societyID)
- execute("SELECT * FROM events WHERE date > NOW()...")
- return eventList
- return eventList
- generateCalendarExport(eventList)
- formatToICS(eventList)
- return calendarFile
- downloadCalendar(calendarFile)
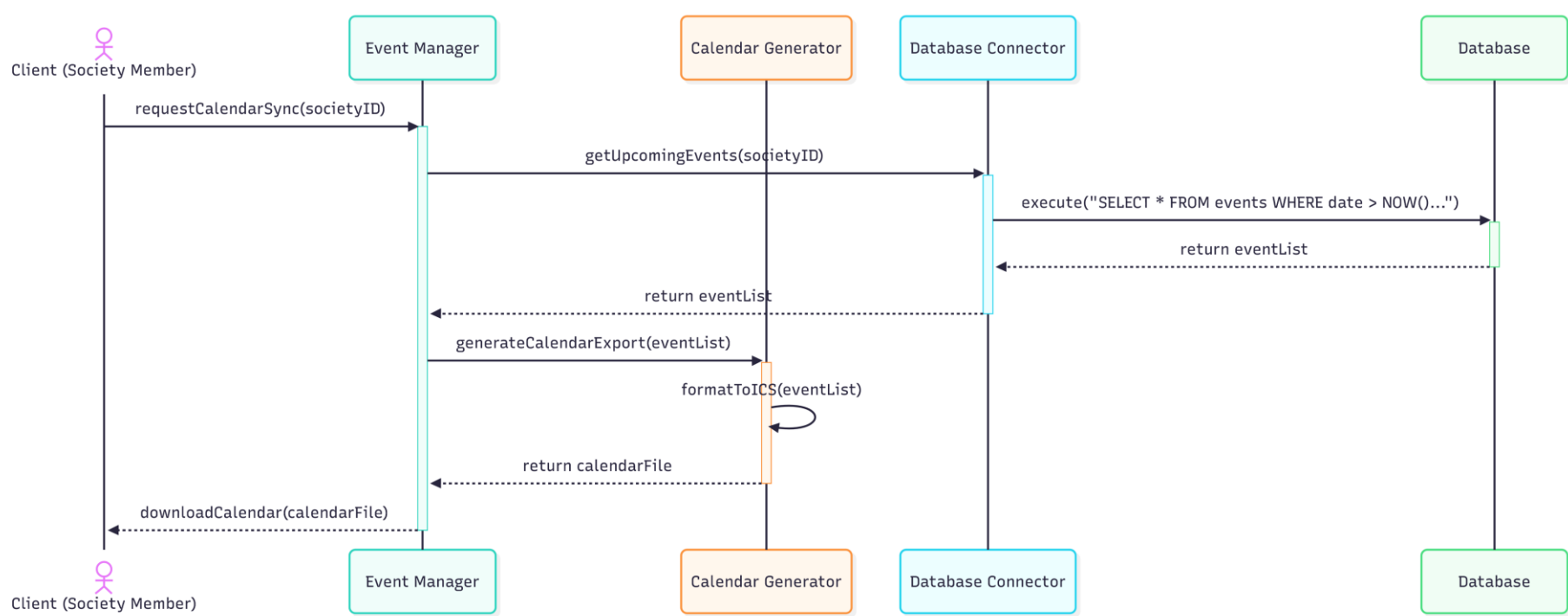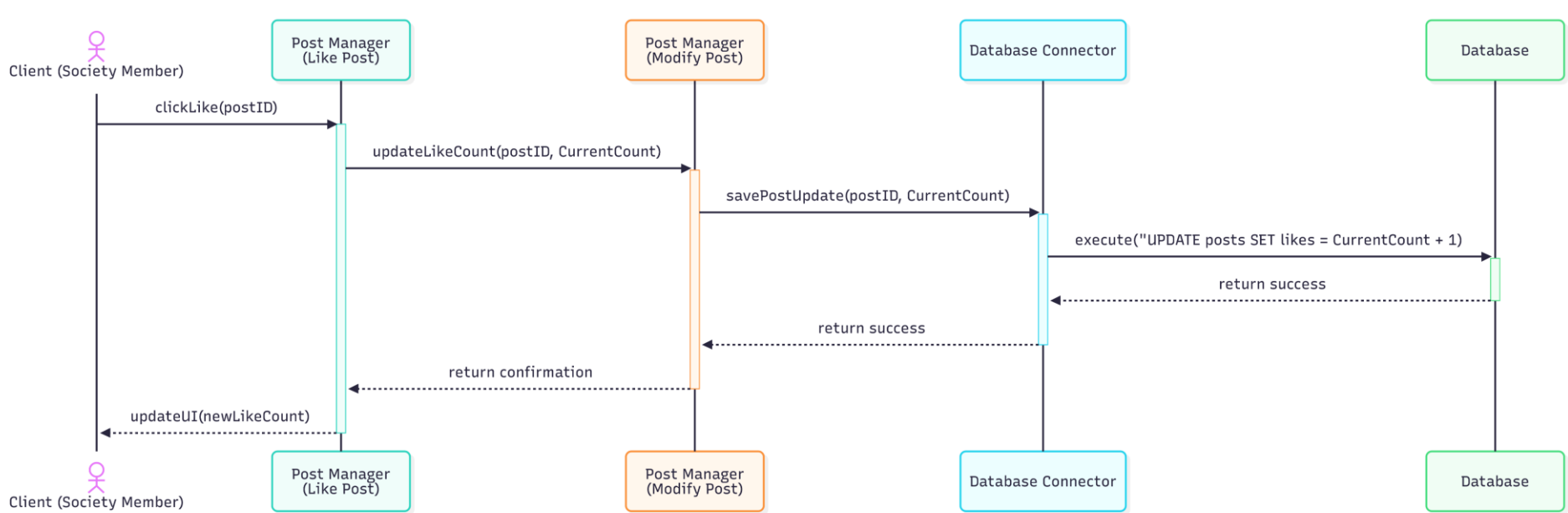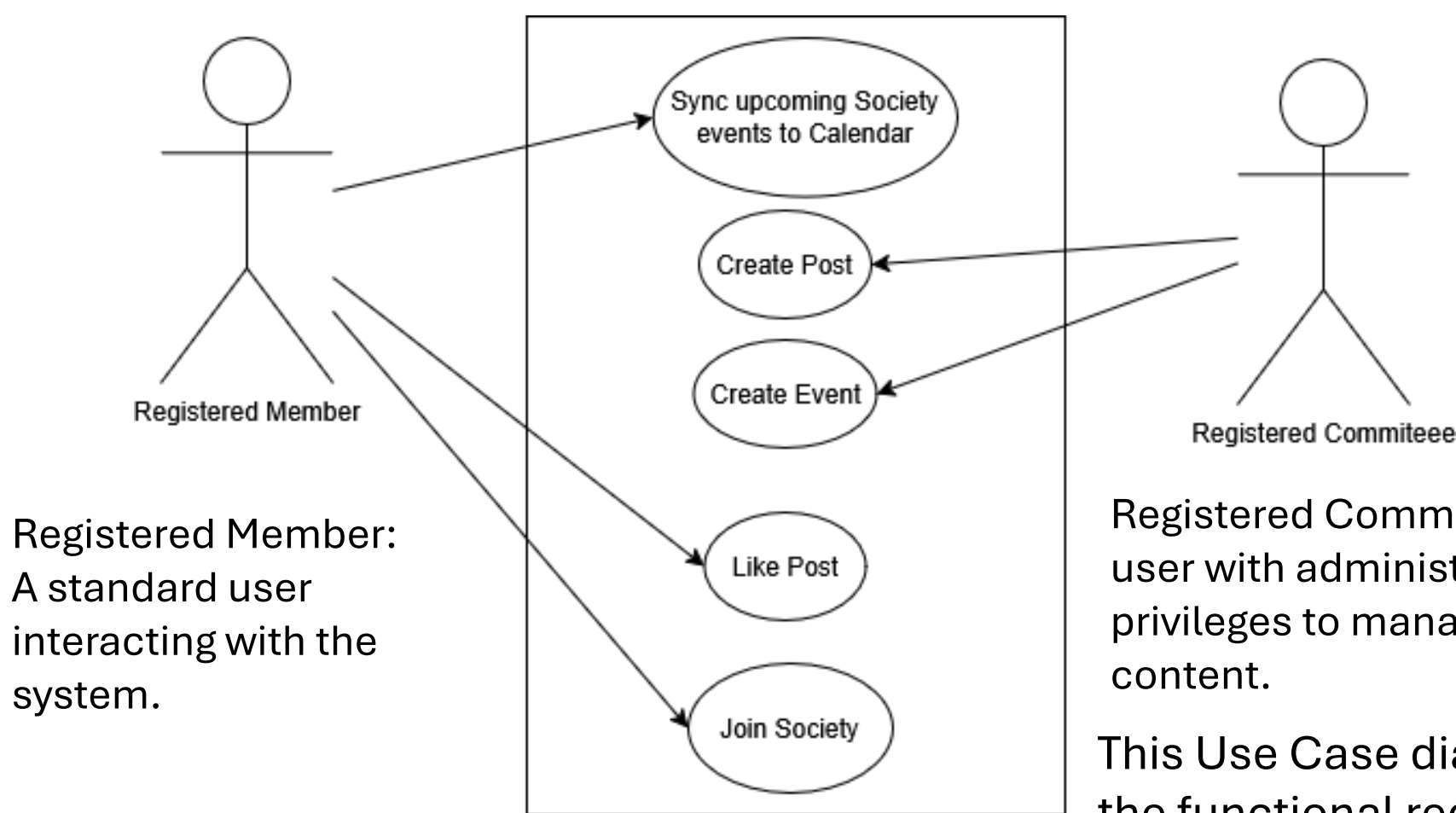
When a user clicks to sync, the system finds all future events in the database. It converts these records into a standard file format that personal calendars understand. The user then downloads this file to import the schedule into their own app.

Member Content Engagement

Client (Society Member) — Post Manager (Like Post) — Post Manager (Modify Post) — Database Connector — Database

- clickLike(postID)
- updateLikeCount(postID, CurrentCount)
- savePostUpdate(postID, CurrentCount)
- execute("UPDATE posts SET likes = CurrentCount + 1")
- return success
- return success
- return confirmation
- updateUI(newLikeCount)

A user clicks the like button on a post. The system calculates the new total and saves the update in the database. The screen immediately refreshes to display the new number of likes.

Use Case Diagram

Registered Member
- Sync upcoming Society events to Calendar
- Create Post
- Create Event
- Like Post
- Join Society

Registered Committee

Registered Member:
A standard user interacting with the system.

Registered Committee: A user with administrative privileges to manage content.

Registered Member can:
Join Society: Enrols the user in a specific society.
Like Post: Allows the user to engage with content posted on the platform.
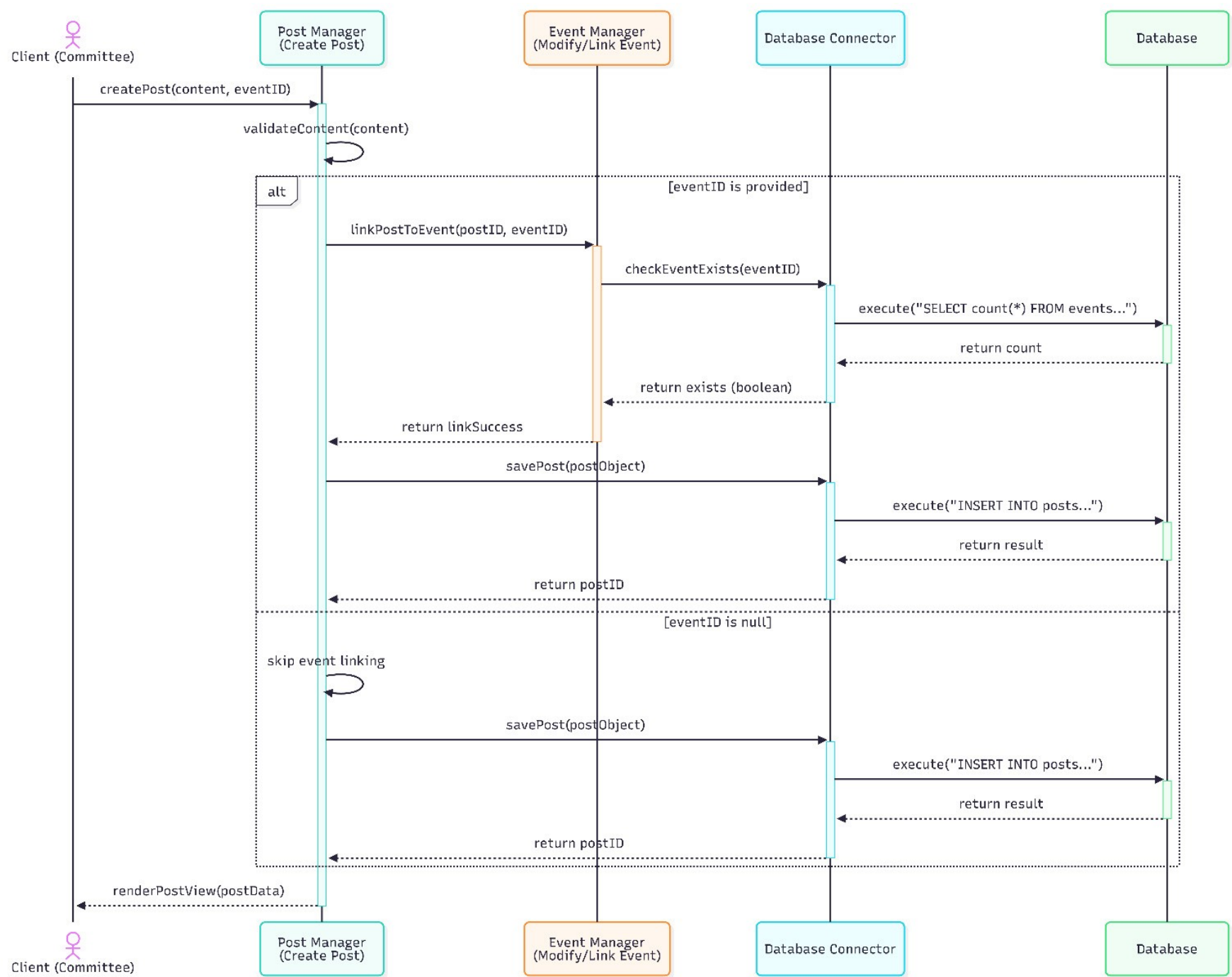Sync upcoming Society events to Calendar: Exports event data to the user's personal calendar.

Registered Committee can:
Create Post: Publishes announcements or updates for members to view.
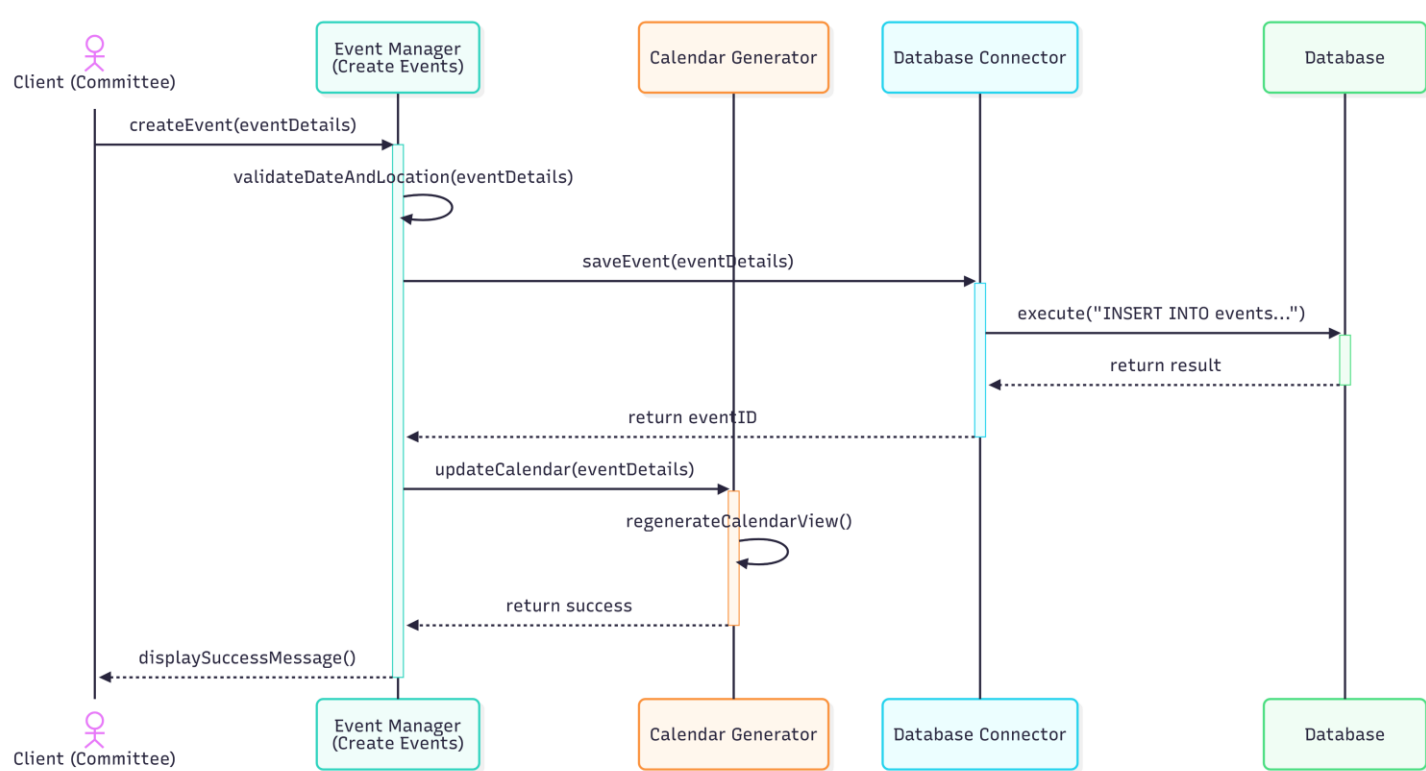Create Event: Schedules and publishes new society activities.

This Use Case diagram outlines the functional requirements for a society management system involving two distinct actors. Registered Members engage with the platform by joining societies, liking posts, and syncing upcoming events to their personal calendars. The Registered Committee holds administrative privileges, focusing on content generation by creating posts and scheduling new events for the society.

Committee Post Publishing

Client (Committee) — Post Manager (Create Post) — Event Manager (Modify/Link Event) — Database Connector — Database

- createPost(content, eventID)
- validateContent(content)

alt
[eventID is provided]
- linkPostToEvent(postID, eventID)
- checkEventExists(eventID)
- execute("SELECT count(*) FROM events...")
- return count
- return exists (boolean)
- return linkSuccess
- savePost(postObject)
- execute("INSERT INTO posts...")
- return result
- return postID

[eventID is null]
- skip event linking
- savePost(postObject)
- execute("INSERT INTO posts...")
- return result
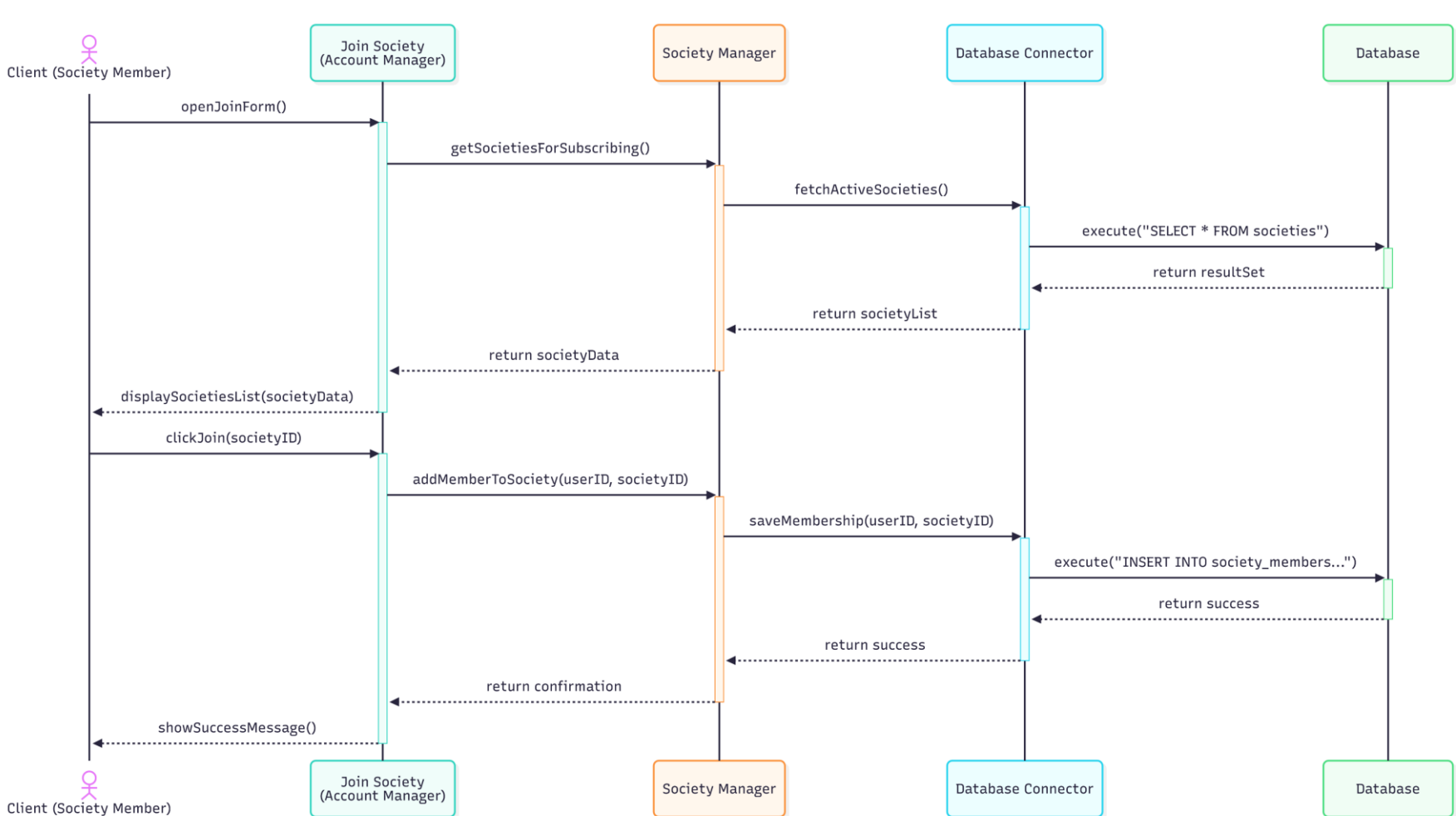- return postID

- renderPostView(postData)

Committee members use this tool to write posts. If the author links an event to the post, the system checks that the event exists. Once the check passes, the post is saved to the database.

Committee Event Management

Client (Committee) — Event Manager (Create Events) — Calendar Generator — Database Connector — Database

- createEvent(eventDetails)
- validateDateAndLocation(eventDetails)
- saveEvent(eventDetails)
- execute("INSERT INTO events...")
- return result
- return eventID
- updateCalendar(eventDetails)
- regenerateCalendarView()
- return success
- displaySuccessMessage()

A committee member enters event details like time and location. The system validates the information and saves it to the database. This action also triggers an update to the main calendar file, so it remains accurate for future user downloads.

Society Discovery and Enrolment

Client (Society Member) — Join Society (Account Manager) — Society Manager — Database Connector — Database

- openJoinForm()
- getSocietiesForSubscribing()
- fetchActiveSocieties()
- execute("SELECT * FROM societies")
- return resultSet
- return societyList
- return societyData
- displaySocietiesList(societyData)
- clickJoin(societyID)
- addMemberToSociety(userID, societyID)
- saveMembership(userID, societyID)
- execute("INSERT INTO society_members...")
- return success
- return success
- return confirmation
- showSuccessMessage()

The system presents a list of active societies from the database. When a user selects a group to join, the system creates a formal link between the user and that society. This action instantly grants them member access.