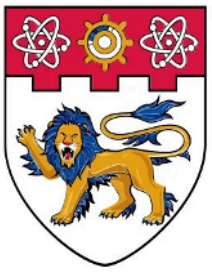


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 – Software Engineering

Lab 2 Deliverables

Lab Group	SCSC
Team	HobbyHive
Members	Teo Rong Xuan (U2421554F)
	Jain Divisha (U2423825L)
	Chua Jing Yi Jax (U2421642A)
	Ishita Dhananjaya (U2420909L)
	Palagiri Afreen Mahtaj (U2422632F)
	Swetha Sudhakar (U2420696G)

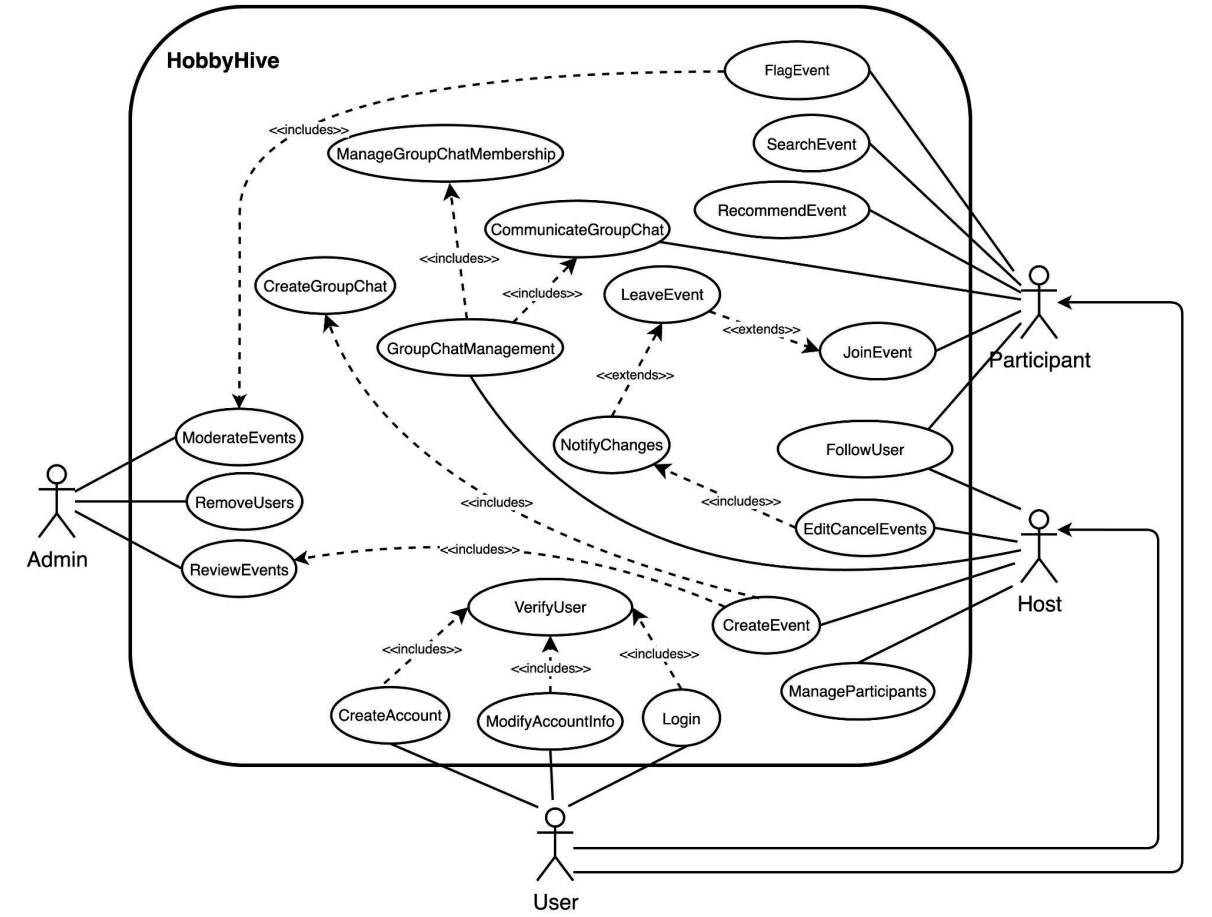
Table Of Contents

1. Use Case Diagram....4
2. Use Case Descriptions....5
I. For Functional Requirement #1 (Login)....5
I.I CreateAccount....5
I.II ModifyAccountInfo....7
I.III Login....9
I.IV VerifyUser....11
II. For Functional Requirement #2 (Participant)....13
II.I SearchEvent....13
II.II JoinEvent....15
II.III RecommendEvent....16
II.IV FollowUser....18
II.V LeaveEvent....20
II.VI FlagEvent....21
III. For Functional Requirement #3 (Host)....23
III.I CreateEvent....23
III.II ManageParticipants....25
III.III EditCancelEvents....26
III.IV NotifyChanges....27
IV. For Functional Requirement #4 (Admin)....28
IV.I RemoveUsers....28
IV.II ReviewEvents....30
IV.III ModerateEvents....32
V. For Functional Requirement #5 (Group Chats)....34
V.I GroupChatManagement....34
V.II CreateGroupChat....35
V.III ManageGroupChatMembership....36
V.IV CommunicateGroupChat....37
3. Class Diagram of Entity Classes....38
4. Key Boundary Classes and Control Classes....39
5. Sequence Diagrams of Some Use Cases....40
I. For Use Cases under #1 (Login)....40
I.I CreateAccount....40
I.II Login....41
II. For Use Cases under #2 (Participant)....42
II.I SearchEvent....42
II.II JoinEvent....43
II.III RecommendEvent....44
II.IV FollowUser....45
II.V LeaveEvent....46
II.VI FlagEvent....47
III. For Use Cases under #3 (Host)....48

III.I CreateEvent..... 48
III.II EditCancelEvent.....49
III.III ManageParticipants.....50
III.IV NotifyChanges..... 51
IV. For Use Cases under #4 (Admin)..... 52
IV.I RemoveUsers.....52
IV.II ReviewEvents.....53
IV.III ModerateEvents..... 54
V. For Use Cases under #5 (Group Chats)..... 55
V.I GroupChatManagement.....55
V.II ManageGroupChatMembership.....56
6. Initial Dialog Map.....57

1. Use Case Diagram

If the image is unclear, please refer to the raw png file that is uploaded together with this document.



2. Use Case Descriptions

I. For Functional Requirement #1 (Login)

I.I CreateAccount

Use Case ID:	#1-1		
Use Case Name:	CreateAccount		
Created By:	Jain Divisha	Last Updated By:	Jain Divisha
Date Created:	7 September 2025	Date Last Updated:	7 September 2025

Actor:	User
Description:	Allows a person to create an account with HobbyHive and be registered as a User.
Pre-Conditions:	None
Post-Conditions	An account with the details provided by the User is created.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. The system prompts the User to input particular details like role, username, password, email and profile picture. 2. The User enters the required information and clicks on “Create Account” to confirm details. 3. The system includes the VerifyUser use case to validate the entered information (check for unique email/username and proper format). 4. An account with the role and other provided details is created for the User.
Alternative Flows:	**AF-S2a**: If User inputs username that is being used by another account already 1. The system displays a message “Username is already taken. Please enter a new username.” 2. The system returns to step 2. **AF-S2b**: If User inputs email that is linked to a

	pre-existing account 1. The system displays a message “Email is already linked to an account” 2. The system returns to step 2.
Exceptions:	If any required information is missing or invalid, an error message “Please enter all valid details” is displayed.
Includes:	VerifyUser
Extends:	None
Special Requirements:	The system must validate all input data and store it securely.
Assumptions:	None
Notes and Issues:	None

I.II ModifyAccountInfo

Use Case ID:	#1-2		
Use Case Name:	ModifyAccountInfo		
Created By:	Jain Divisha	Last Updated By:	Jain Divisha
Date Created:	7 September 2025	Date Last Updated:	7 September 2025

Actor:	User
Description:	Allows User to change details linked to their account.
Pre-Conditions:	The User must have an account with HobbyHive and must be logged into the account.
Post-Conditions	The account details related to the User are modified.
Priority:	Medium
Frequency Of Use:	Medium
Flow Of Events:	1. The User opens the Account Menu and clicks on **Modify Account Info**. 2. The system redirects the User to a new page with options **Change Password**, **Change Email**, **Change Username**, **Change Profile Picture**. 3. If User clicks on **Change Password** 3.1. The system prompts User to enter **New Password** and **Confirm New Password**. This information is displayed as dots which can be unmasked by the user by clicking on an eye icon in the text bar. 3.2. User inputs required details and clicks on Confirm Changes. 3.3. The system includes the VerifyUser use case to validate matching fields and confirm the current password before saving changes. 3.4. The system updates the User's account information.
Alternative Flows:	**AF-S3a**: If User clicks on **Change Email** 1. The system prompts User to enter New Email and Confirm New Email. 2. User inputs required details and clicks on Confirm Changes.

	3. The system includes the VerifyUser use case to check if the email is not already associated with another account. 4. The system updates the User's account information. **AF-S3b**: If User clicks on **Change Username** 1. The system prompts User to enter New Username and Confirm New Username. 2. User inputs required details and clicks on Confirm Changes. 3. The system includes the VerifyUser use case to check if the username is not already associated with another account. 4. The system updates the User's account information. **AF-S3c**: If User clicks on **Change Profile Picture** 1. The system prompts User to select a new picture. 2. The User selects a picture with appropriate dimensions. 3. The system updates the User's account information
Exceptions:	If the data provided by the User in both the fields does not match, the system issues an error message "Mismatched Data! Please ensure correct entry."
Includes:	VerifyUser
Extends:	None
Special Requirements:	The system must verify data if both fields match and only then modify the Account Information.
Assumptions:	The User is already logged in.
Notes and Issues:	None

I.III Login

Use Case ID:	#1-3		
Use Case Name:	Login		
Created By:	Jain Divisha	Last Updated By:	Jain Divisha
Date Created:	7 September 2025	Date Last Updated:	7 September 2025

Actor:	User
Description:	Allows User to login to and access HobbyHive with their Account Information.
Pre-Conditions:	1. User has an account with HobbyHive that has been made through the CreateAccount use case. 2. User is logged out of the system.
Post-Conditions	The User is logged into and can access HobbyHive.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. The system prompts User to enter email/username and password. 2. The User provides the information. The password is masked as dots, but the User can choose to unmask it by clicking on the eye icon in the text bar. 3. The User clicks on Login. 4. The system includes the VerifyUser use case to verify the provided credentials against the database. 5. If credentials are correct, access is granted.
Alternative Flows:	**AF-S4**: If upon verification, the information does not match with any entry 1. The system outputs a message “This account does not exist. Please try again!” for 3 seconds. 2. The system returns to step 1
Exceptions:	If any required information is missing, an error message is displayed.
Includes:	VerifyUser

Extends:	None
Special Requirements:	System must validate user input data
Assumptions:	The User has an account with HobbyHive.
Notes and Issues:	None

I.IV VerifyUser

Use Case ID:	#1-4		
Use Case Name:	VerifyUser		
Created By:	Jain Divisha	Last Updated By:	Jain Divisha
Date Created:	7 September 2025	Date Last Updated:	7 September 2025

Actor:	System (triggered during user interactions in CreateAccount, ModifyAccountInfo and Login)
Description:	Ensures the user-provided information (email, username, password, or other credentials) is valid, unique (when required), and matches the system records before proceeding with account creation, modification, or login.
Pre-Conditions:	1. User is in the process of creating, modifying, or accessing their account. 2. Relevant input fields have been provided by the user.
Post-Conditions	1. User data is verified for correctness, validity, and uniqueness (as applicable). 2. If verification fails, an appropriate error is displayed.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. The system receives input data (email, username, password, etc.) from the primary use case. 2. The system checks for mandatory fields (none should be empty). 3. System validates data format: 3.1. Email is in valid format (or) Username follows allowed character rules. 3.2. Password meets security criteria (if relevant). 4. If invoked during CreateAccount: 4.1. The system checks that email and username is not already in use. 5. The system returns verification result (success or specific error)

Alternative Flows:	**AF-S1**: Missing mandatory fields 1. The system displays an error message. 2. It returns to the data entry step in the calling use case. **AF-S2**: Invalid email format 1. The system displays an error message 2. It returns to the data entry step in the calling use case. **AF-S4a**: If invoked during Login: 1. The system compares provided credentials with stored records. 2. Moves to step 5. **AF-S4b**: If invoked during ModifyAccountInfo: 1. The system checks for conflicts (e.g., email already in use, username already in use) and verifies if the current password is the same as the new password before applying changes. 2. Moves to step 5
Exceptions:	None
Includes:	None
Extends:	None
Special Requirements:	All passwords must be hashed. Verification must be completed in under 3 seconds.
Assumptions:	The database of users is always up to date and available for verification queries.
Notes and Issues:	Future versions can include multi-factor authentication (MFA).

II. For Functional Requirement #2 (Participant)

II.I SearchEvent

Use Case ID:	#2-1		
Use Case Name:	SearchEvent		
Created By:	Afreen Mahtaj	Last Updated By:	Jain Divisha
Date Created:	8 September 2025	Date Last Updated:	23 September 2025

Actor:	Participant
Description:	Allows a Participant to search for events using keywords, categories, dates, and locations. The system ensures valid inputs and displays only matching events.
Pre-Conditions:	1. Participant must be logged in. 2. Events must exist in the database.
Post-Conditions	1. A filtered list of events is displayed to the Participant.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. Participant navigates to the Search bar. 2. System prompts Participant to enter filters (keyword, category, date, location). 3. Participant enters one or more search criteria. 3.1. Keyword: text of length 1–255. 3.2. Category: selected from a predefined list. 3.3. Date: valid date in YYYY-MM-DD format. 3.4. Location: set by address and radius in km. 4. Participant clicks “Search.” 5. System validates inputs. 6. System retrieves and displays all events matching criteria within the specified circular boundary.
Alternative Flows:	**AF-S1**: If no results match, the system displays “No events found. Try adjusting filters.”
Exceptions:	1. Invalid category selection: 1.1. System shows error “Invalid category.”

	2. Invalid date format: 2.1. System shows error “Please enter date in YYYY-MM-DD format.” 3. Empty keyword or >255 characters: 3.1. System shows error “Keyword must be 1–255 characters.”
Includes:	None
Extends:	None
Special Requirements:	1. Search results must be displayed within 2 seconds. 2. Location-based search must calculate distance using defined radius.
Assumptions:	1. The event database is indexed and supports filtering efficiently.
Notes and Issues:	None

II.II JoinEvent

Use Case ID:	#2-2		
Use Case Name:	JoinEvent		
Created By:	Afreen Mahtaj	Last Updated By:	Jain Divisha
Date Created:	8 September 2025	Date Last Updated:	23 September 2025

Actor:	Participant
Description:	Allows a Participant to join an event. Joining grants access to group chat and updates.
Pre-Conditions:	1. Participant must be logged in. 2. Event must exist.
Post-Conditions	1. Participant becomes a Member of the event. 2. Participant gains access to group chat.
Priority:	High
Frequency Of Use:	Medium
Flow Of Events:	1. Participant views event details. 2. Participant clicks “Join” button. 3. Participant becomes a member of the Event and is added to the Group Chat.
Alternative Flows:	**AF-S2:** If the event already has enough members: 1. A message is sent to the participant “Event is Full”
Exceptions:	Event does not exist, system shows “This event is unavailable.”
Includes:	None
Extends:	None
Special Requirements:	None
Assumptions:	1. Groups may have maximum capacity set by Host.
Notes and Issues:	None

II.III RecommendEvent

Use Case ID:	#2-3		
Use Case Name:	RecommendEvent		
Created By:	Afreen Mahtaj	Last Updated By:	Jain Divisha
Date Created:	8 September 2025	Date Last Updated:	23 September 2025

Actor:	Participant
Description:	System recommends events to a Participant based on preferences (categories, hobbies), popularity, and proximity.
Pre-Conditions:	1. Participant must be logged in. 2. Event database must exist.
Post-Conditions	1. Recommended events are displayed on Participant's homepage.
Priority:	Medium
Frequency Of Use:	Medium
Flow Of Events:	1. Participant navigates to "Recommended Events." 2. System retrieves Participant's preferences (categories, hobbies, followed users). 3. System identifies events based on: 3.1. Matching categories/hobbies. 3.2. Event popularity. 3.3. Location proximity. 4. System displays recommended events in a ranked list.
Alternative Flows:	**AF-S1**: If no personalized matches exist, the system displays popular nearby events.
Exceptions:	If the system fails to fetch user preferences, fallback to general recommendations.
Includes:	None
Extends:	None
Special Requirements:	1. Recommendations should refresh daily. 2. Distance calculations must be accurate to within

	1 km.
Assumptions:	User preference data is up to date.
Notes and Issues:	None

II.IV FollowUser

Use Case ID:	#2-4		
Use Case Name:	FollowUser		
Created By:	Afreen Mahtaj	Last Updated By:	Jain Divisha
Date Created:	8 September 2025	Date Last Updated:	23 September 2025

Actor:	Participant
Description:	Allows a Participant to follow another User (Host or Participant) to see their updates and events more prominently in recommendations.
Pre-Conditions:	1. Participant must be logged in. 2. The other User exists in the system.
Post-Conditions	1. Followed User is added to Participant's "Following" list. 2. Participant receives updates about followed User's activities.
Priority:	Medium
Frequency Of Use:	Medium
Flow Of Events:	1. The participant visits another User's profile. 2. Participant clicks the "Follow" button. 3. System adds User to "Following" list. 4. System confirms: "You are now following this user." 5. The system notifies the followed User: "You have a new follower."
Alternative Flows:	**AF-S1**: If Participant clicks "Unfollow," the system removes User from the following list.
Exceptions:	1. If the target User no longer exists, the system displays: "This user is unavailable."
Includes:	None
Extends:	None
Special Requirements:	1. Notifications must be sent for new follows. 2. Unfollow should remove all related

	notifications/recommendations.
Assumptions:	All Users allow being followed unless the account is deleted.
Notes and Issues:	None

II.V LeaveEvent

Use Case ID:	#2-5		
Use Case Name:	LeaveEvent		
Created By:	Jain Divisha	Last Updated By:	Jain Divisha
Date Created:	11 September 2025	Date Last Updated:	23 September 2025

Actor:	Participant
Description:	Allows a Participant to leave an Event they are no longer attending.
Pre-Conditions:	Participant is part of an Event.
Post-Conditions	Participant is removed from the Event.
Priority:	Medium
Frequency Of Use:	Low
Flow Of Events:	1. Participant visits the group chat of the event that they have previously joined. 2. Participant clicks the LEAVE button at the top of the group chat. 3. Participant is removed from the Event and Groupchat.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Extends:	JoinEvent
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

II.VI FlagEvent

Use Case ID:	#2-6		
Use Case Name:	FlagEvent		
Created By:	Chua Jing Yi, Jax	Last Updated By:	Jain Divisha
Date Created:	11 September 2025	Date Last Updated:	23 September 2025

Actor:	Participant
Description:	Allows a Participant to flag a listed event for inappropriate content.
Pre-Conditions:	1. Participant must be logged in. 2. The event must exist in the system.
Post-Conditions	1. The event is flagged and recorded in the system. 2. Admins are notified of the flagged event. 3. Participant may see a confirmation message that the event has been flagged.
Priority:	High
Frequency Of Use:	Low–Medium
Flow Of Events:	1. Participant visits an Event page. 2. Participant click the “Flag Event” option. 3. The system prompts Participant to select a reason for flagging. 4. Participant selects a reason and confirms submission. 5. System records the flag in the database. 6. System confirms: “This event has been flagged for review.” 7. System notifies Admin for review and action.
Alternative Flows:	**AF-S1:** If the Participant cancels after being prompted for a reason, no flag is recorded. **AF-S2:** If the Participant has already flagged this event, the system prevents duplicate flags and displays: “You have already flagged this event.”
Exceptions:	If the Event no longer exists “This event is unavailable.”
Includes:	ModerateEvents

Extends:	None
Special Requirements:	The system should log reasons and timestamps for auditing purposes. A single Participant should not be able to repeatedly flag the same event.
Assumptions:	Participants can flag any event that exists in the system.
Notes and Issues:	None

III. For Functional Requirement #3 (Host)

III.I CreateEvent

Use Case ID:	#3-1		
Use Case Name:	CreateEvent		
Created By:	Swetha Sudhakar	Last Updated By:	Chua Jing Yi Jax
Date Created:	8 September 2025	Date Last Updated:	21 September 2025

Actor:	Host
Description:	Allows Hosts to create an event on HobbyHive by providing event details such as title, description, category, date, time, and location.
Pre-Conditions:	The Host must be logged in.
Post-Conditions	A new event is created and visible to Participants for joining.
Priority:	High
Frequency Of Use:	Medium–High
Flow Of Events:	1. Host clicks on "Create Event. 2. The system prompts the Host to input event details (title, description, category, date, time, location, capacity). 3. Host waits for Admin to review the event via ReviewEventst enters all required details and uploads an optional image/banner. 4. Host confirms creation. 5. The system validates details and creates the event in a Pending State. 6. If approved, the event becomes visible to Participants for joining. 7. If rejected, the Host is notified of the reason for rejection.
Alternative Flows:	**AF-S1:** If mandatory details are missing, the system shows “Please fill in all required fields.” **AF-S2:** If invalid date/time is entered, the system shows “Invalid scheduling information.”

Exceptions:	If there is a Database error, the system sends out a message “Event could not be created. Please try again later!”
Includes:	ReviewEvent, CreateGroupChat
Extends:	None
Special Requirements:	Events must have a unique title and valid date/time.
Assumptions:	Hosts have stable network access.
Notes and Issues:	Need to consider timezone handling for events.

III.II ManageParticipants

Use Case ID:	#3-2		
Use Case Name:	ManageParticipants		
Created By:	Swetha Sudhakar	Last Updated By:	Jain Divisha
Date Created:	8 September 2025	Date Last Updated:	23 September 2025

Actor:	Host
Description:	Allows Hosts to manage event participants by and viewing participant lists.
Pre-Conditions:	Host must be logged in and have at least one active event.
Post-Conditions	Participant statuses are updated based on Host's actions.
Priority:	High
Frequency Of Use:	Medium
Flow Of Events:	1. Host selects their event from the dashboard. 2. Host opens the Participants tab. 3. Host reviews participants 4. Host can remove participants if necessary. 5. System updates the participant list and sends notifications.
Alternative Flows:	**AF-S1:** If participant limit is reached, the system shows "Maximum participants reached."
Exceptions:	None
Includes:	None
Extends:	None
Special Requirements:	Notifications must be sent in real time.
Assumptions:	Hosts act fairly and follow community guidelines.
Notes and Issues:	Consider escalation to Admin if Host repeatedly removes participants unfairly.

III.III EditCancelEvents

Use Case ID:	#3-3		
Use Case Name:	EditCancelEvents		
Created By:	Swetha Sudhakar	Last Updated By:	Chua Jing Yi Jax
Date Created:	8 September 2025	Date Last Updated:	21 September 2025

Actor:	Host
Description:	Allows Hosts to edit or cancel existing events.
Pre-Conditions:	Host must have an active event.
Post-Conditions	Event details are updated OR event is cancelled. Participants are notified.
Priority:	High
Frequency Of Use:	Medium
Flow Of Events:	1. Host selects an event. 2. Host clicks Edit or Cancel. 3. If Edit, the system shows editable fields. 4. Host updates fields and confirms. 5. If Cancel, Host confirms cancellation. 6. System applies changes and notifies participants via NotifyChanges.
Alternative Flows:	**AF-S1:** If edited event date is in the past, system shows “Invalid event date.”
Exceptions:	Database error during update.
Includes:	NotifyChanges
Extends:	None
Special Requirements:	Event cancellation must automatically close group chats.
Assumptions:	Hosts edit responsibly.
Notes and Issues:	Ensure data consistency when updating/cancelling events.

III.IV NotifyChanges

Use Case ID:	#3-4		
Use Case Name:	NotifyChanges		
Created By:	Swetha Sudhakar	Last Updated By:	Jain Divisha
Date Created:	8 September 2025	Date Last Updated:	23 September 2025

Actor:	System
Description:	Automatically notifies Participants of event updates or cancellations.
Pre-Conditions:	An event must exist and have Participants.
Post-Conditions	Notifications are delivered to all Participants.
Priority:	High
Frequency Of Use:	Medium
Flow Of Events:	1. Host edits or cancels an event. 2. The system detects the change. 3. The system generates notifications. 4. All Participants receive the notification via dashboard and group chat.
Alternative Flows:	**AF-S1:** If Participant has disabled notifications, updates are shown only in dashboard.
Exceptions:	Notification delivery fails due to connectivity issues.
Includes:	None
Extends:	LeaveEvent
Special Requirements:	Notifications must be near real-time (< 1 min).
Assumptions:	Participants check their dashboard/group chat regularly.
Notes and Issues:	Consider adding email alerts as backup.

IV. For Functional Requirement #4 (Admin)

IV.I RemoveUsers

Use Case ID:	#4-1		
Use Case Name:	RemoveUsers		
Created By:	Jain Divisha	Last Updated By:	Ishita
Date Created:	7 September 2025	Date Last Updated:	20 September 2025

Actor:	Admin
Description:	Allows Admin to remove a User by their username for a specified duration.
Pre-Conditions:	Admin is logged into their HobbyHive account.
Post-Conditions	A User account is removed from HobbyHive and the email can not be used to make future accounts.
Priority:	High
Frequency Of Use:	Low
Flow Of Events:	1. Admin inputs the username of the user that the Admin wants to ban. 2. Admin inputs the duration of the ban. 3. Admin confirms the ban by selecting **Confirm**.
Alternative Flows:	1. If the username is not found, the system shall display an error message and prompt the Admin to input another username. 2. Admin cancels the ban by selecting CancelBan
Exceptions:	None
Includes:	None
Extends:	None
Special Requirements:	None
Assumptions:	None

Notes and Issues:	None

IV.II ReviewEvents

Use Case ID:	#4-2		
Use Case Name:	ReviewEvents		
Created By:	Jain Divisha	Last Updated By:	Ishita
Date Created:	7 September 2025	Date Last Updated:	21 September 2025

Actor:	Admin
Description:	Allows Admin to review event creation requests submitted by Users and decide whether to approve, reject, or request more information based on the event's description and purpose.
Pre-Conditions:	1. Admin is logged into their HobbyHive account. 2. One or more event creation requests have been submitted by Users.
Post-Conditions	1. The event creation request is approved or rejected or set to "pending additional information." 2. The requesting User is notified of the respective decisions.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. The system displays a list of event creation requests to the Admin. 2. Admin selects a event creation request to review. 3. The system displays the event details, including its description. 4. Admin reviews the details and chooses one of the following actions: 4.1. **Approve** the request 4.2. **Reject** the request 4.3. **Request more information** from the Host 5. If the Admin selects **Approve**, the system updates the event's status accordingly and notifies the User of the decision.
Alternative Flows:	**AF-S5a**: Request more information 1. Admin determines that the event description is insufficient or unclear. 2. Admin selects "Request More Information."

	3. The system notifies the User to provide additional details. **AF-S5b**: Reject offensive or unrelated event 1. Admin identifies offensive, inappropriate, or unrelated content in the event description. 2. Admin selects “Reject.” 3. The system notifies the User of rejection and provides a reason.
Exceptions:	None
Includes:	None
Extends:	None
Special Requirements:	1. All event descriptions must be stored securely and remain editable only until Admin takes action. 2. Notifications to Users must be sent in real-time.
Assumptions:	User has submitted the complete initial event creation request before Admin review.
Notes and Issues:	None

IV.III ModerateEvents

Use Case ID:	#4-3		
Use Case Name:	ModerateEvents		
Created By:	Jain Divisha	Last Updated By:	Ishita
Date Created:	7 September 2025	Date Last Updated:	23 September 2025

Actor:	Admin
Description:	Allows Admin to review and moderate inappropriate or **already created** problematic events on HobbyHive by removing those that violate guidelines, promote illegal activities, contain misleading information, or receive multiple user reports.
Pre-Conditions:	1. Admin is logged into their HobbyHive account. 2. One or more events are flagged or reported for review.
Post-Conditions	1. The inappropriate event is removed, or its status is updated to “Reviewed”. 2. Relevant Users (e.g., event creators or reporters) are notified of the action.
Priority:	High
Frequency Of Use:	Medium
Flow Of Events:	1. The system displays a list of flagged or reported events to the Admin. 2. Admin selects a event for moderation. 3. The system shows details including description, schedule, reports, and flagged content. 4. Admin reviews the details and chooses one of the following actions: 4.1. Remove the event 4.2. Keep the event 5. If the Admin clicks on **Remove**, the system removes the event listing from HobbyHive and notifies the creator and, if applicable, the reporting Users.
Alternative Flows:	**AF-S5**: If the admin clicks on **Keep** 1. The system updates the event status to “Reviewed” and notifies the creator and, if

	applicable, the reporting Users.
Exceptions:	None
Includes:	None
Extends:	None
Special Requirements:	All moderation actions must be logged with timestamps and Admin’s username for audit purposes.
Assumptions:	Events can be reported by Users or flagged automatically by the system.
Notes and Issues:	None

V. For Functional Requirement #5 (Group Chats)

V.I GroupChatManagement

Use Case ID:	#5-1		
Use Case Name:	GroupChatManagement		
Created By:	Swetha Sudhakar	Last Updated By:	Swetha Sudhakar
Date Created:	8 September 2025	Date Last Updated:	8 September 2025

Actor:	System, Host, Participant
Description:	Handles the lifecycle of event group chats, including creation, membership, and closure.
Pre-Conditions:	An event must exist.
Post-Conditions	Group chat is created, managed, and eventually closed.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. When a Host creates an event, the system generates a group chat. 2. Participants are added once they join. 3. Participants are removed when they leave. 4. Group chat closes after event cancellation or one month post-event.
Alternative Flows:	**AF-S1:** If chat cannot be created, the system retries.
Exceptions:	Chat service failure will show “Group chat unavailable.”
Includes:	ManageGroupChatMembership, CommunicateGroupChat
Extends:	None
Special Requirements:	Chats must support real-time messaging.
Assumptions:	All users have access to messaging features.
Notes and Issues:	Need to consider data retention policy for chat logs.

V.II CreateGroupChat

Use Case ID:	#5-2		
Use Case Name:	CreateGroupChat		
Created By:	Swetha Sudhakar	Last Updated By:	Swetha Sudhakar
Date Created:	8 September 2025	Date Last Updated:	8 September 2025

Actor:	System
Description:	Automatically creates a group chat for each new event.
Pre-Conditions:	A Host successfully creates an event.
Post-Conditions	A group chat linked to the event is created.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. The Host creates an event. 2. The system generates a unique group chat ID. 3. The chat is linked to the event.
Alternative Flows:	**AF-S1:** If creation fails, retry the process.
Exceptions:	Chat service error.
Includes:	None
Extends:	None
Special Requirements:	Group chats must be unique per event.
Assumptions:	The chat system is integrated with the event service.
Notes and Issues:	Consider scalability if many events are created.

V.III ManageGroupChatMembership

Use Case ID:	#5-3		
Use Case Name:	ManageGroupChatMembership		
Created By:	Swetha Sudhakar	Last Updated By:	Swetha Sudhakar
Date Created:	8 September 2025	Date Last Updated:	8 September 2025

Actor:	System, Participant, Host
Description:	Controls adding/removing users from group chats based on join/leave actions.
Pre-Conditions:	Event exists and a group chat is created.
Post-Conditions	Group chat membership is updated accordingly.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. The participant joins an event, and the system adds them to the group chat. 2. Participant leaves, the system removes them.
Alternative Flows:	**AF-S1:** If the Host removes a participant, the system notifies them.
Exceptions:	Membership sync error, which leads to delayed updates.
Includes:	None
Extends:	None
Special Requirements:	Membership must be updated in real time.
Assumptions:	Hosts/Participants follow group chat rules.
Notes and Issues:	Ensure proper logging for auditing.

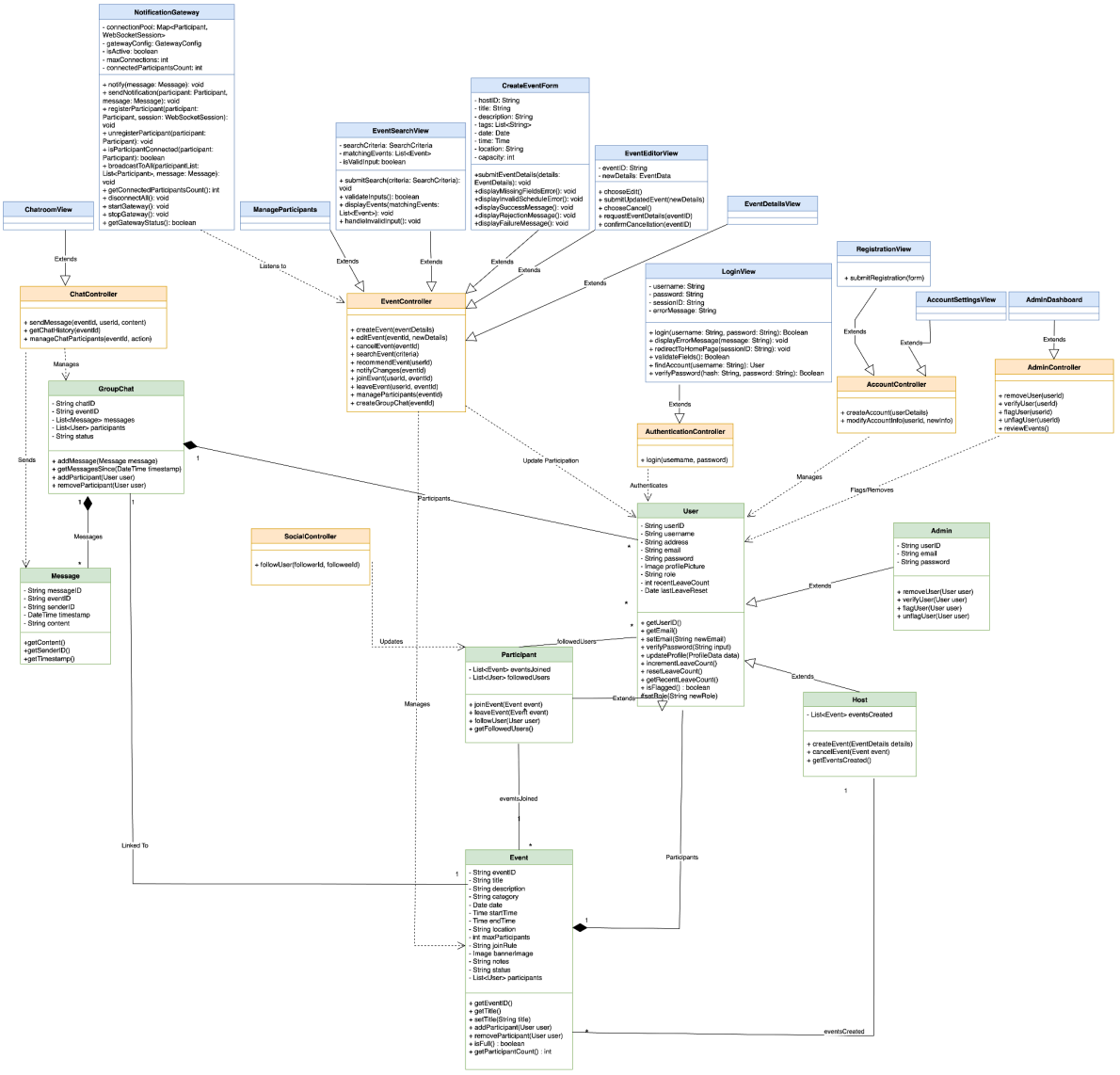
V.IV CommunicateGroupChat

Use Case ID:	#5-4		
Use Case Name:	CommunicateGroupChat		
Created By:	Swetha Sudhakar	Last Updated By:	Swetha Sudhakar
Date Created:	8 September 2025	Date Last Updated:	8 September 2025

Actor:	Host, Participant
Description:	Allows Hosts and Participants to exchange messages in the group chat.
Pre-Conditions:	User must be part of the group chat.
Post-Conditions	Messages are delivered to all chat members.
Priority:	High
Frequency Of Use:	High
Flow Of Events:	1. User types a message. 2. System transmits message. 3. All members see the message in real time.
Alternative Flows:	**AF-S1:** If message fails to send, system retries.
Exceptions:	Connectivity issue.
Includes:	None
Extends:	None
Special Requirements:	Must support text of up to 256 characters per message.
Assumptions:	Users communicate respectfully.
Notes and Issues:	Consider adding media support in the future.

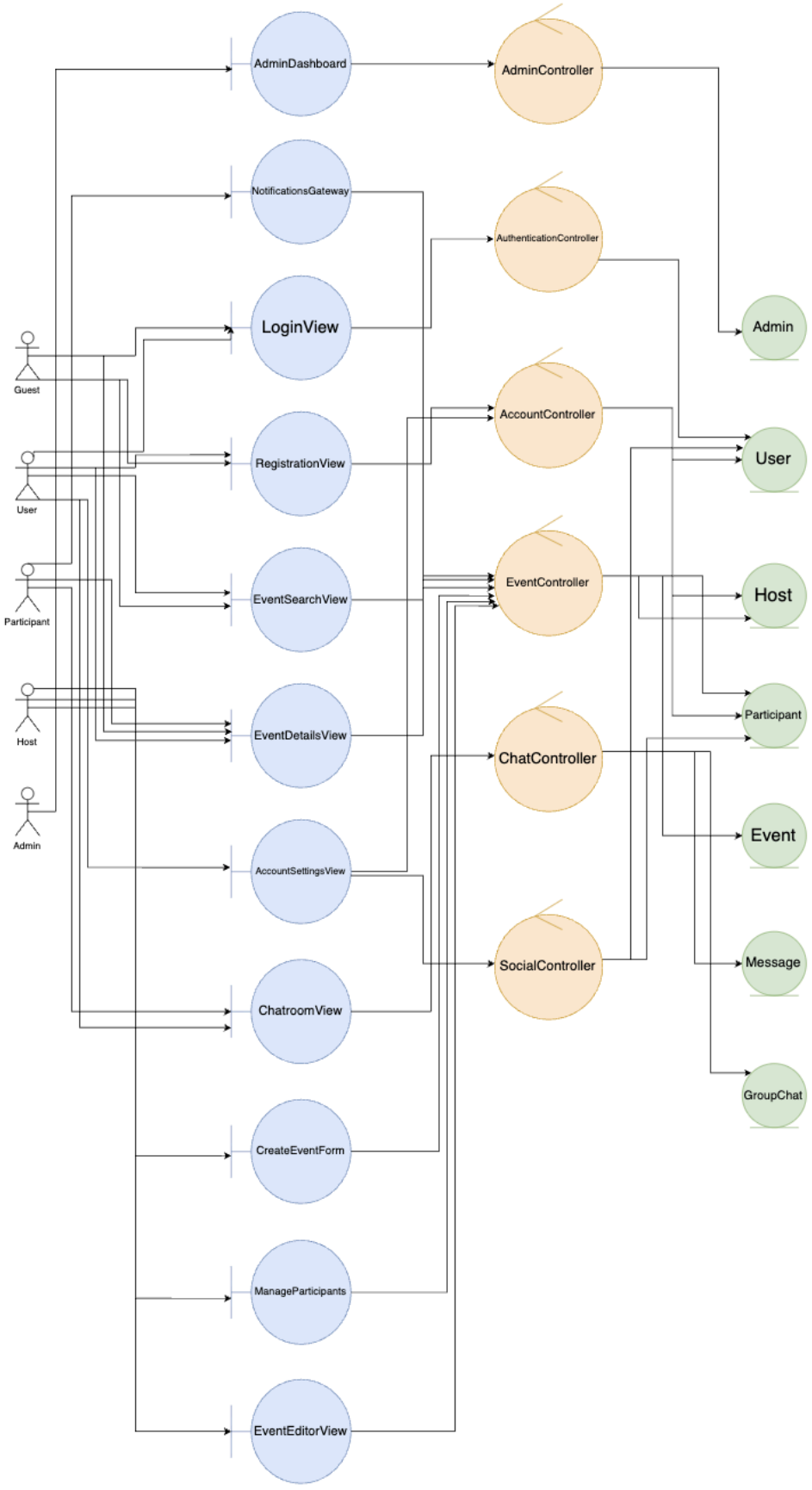
3. Class Diagram of Entity Classes

If the image is unclear, please refer to the raw png file that is uploaded together with this document.



4. Key Boundary Classes and Control Classes

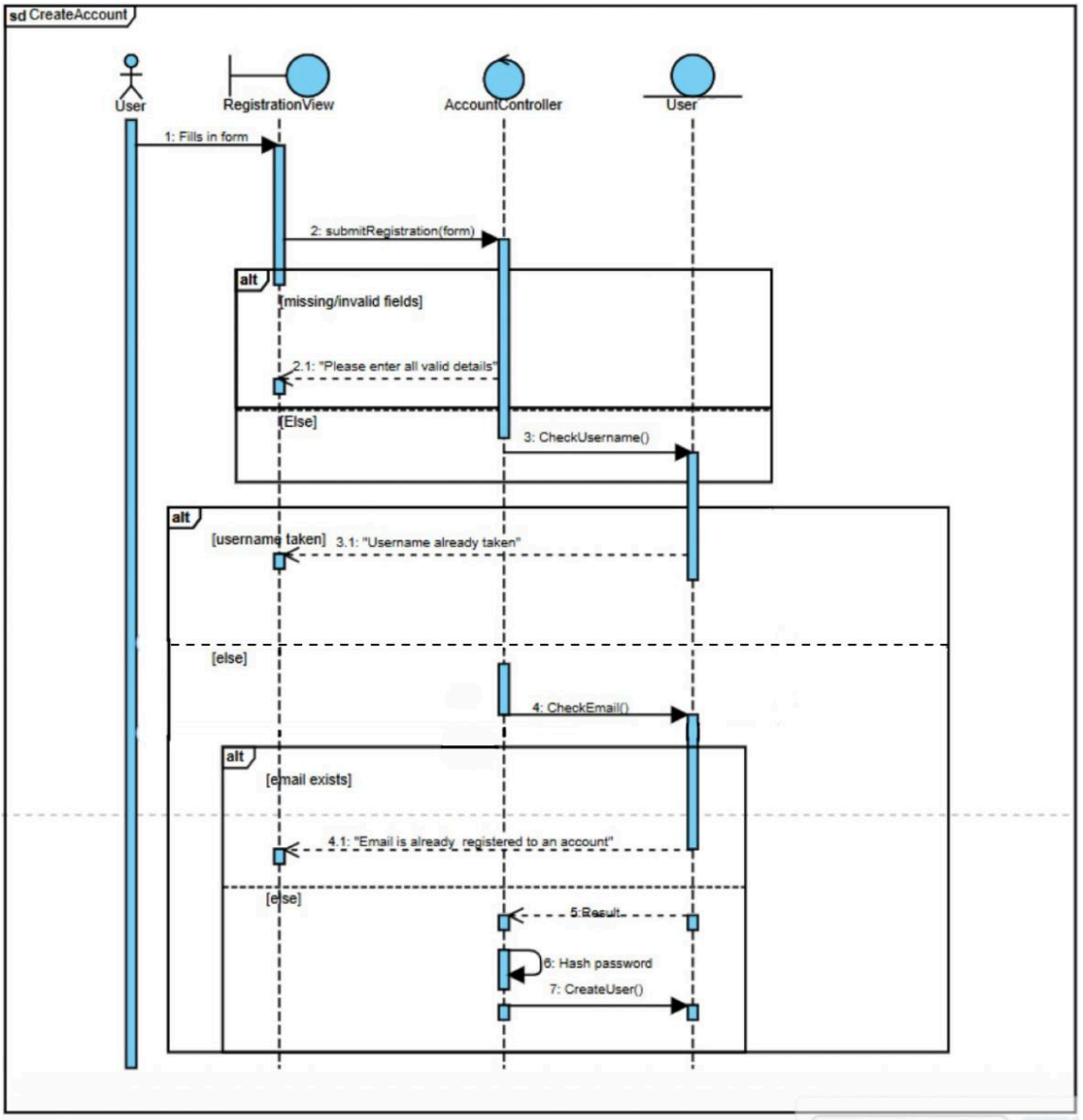
If the image is unclear, please refer to the raw png file that is uploaded together with this document.



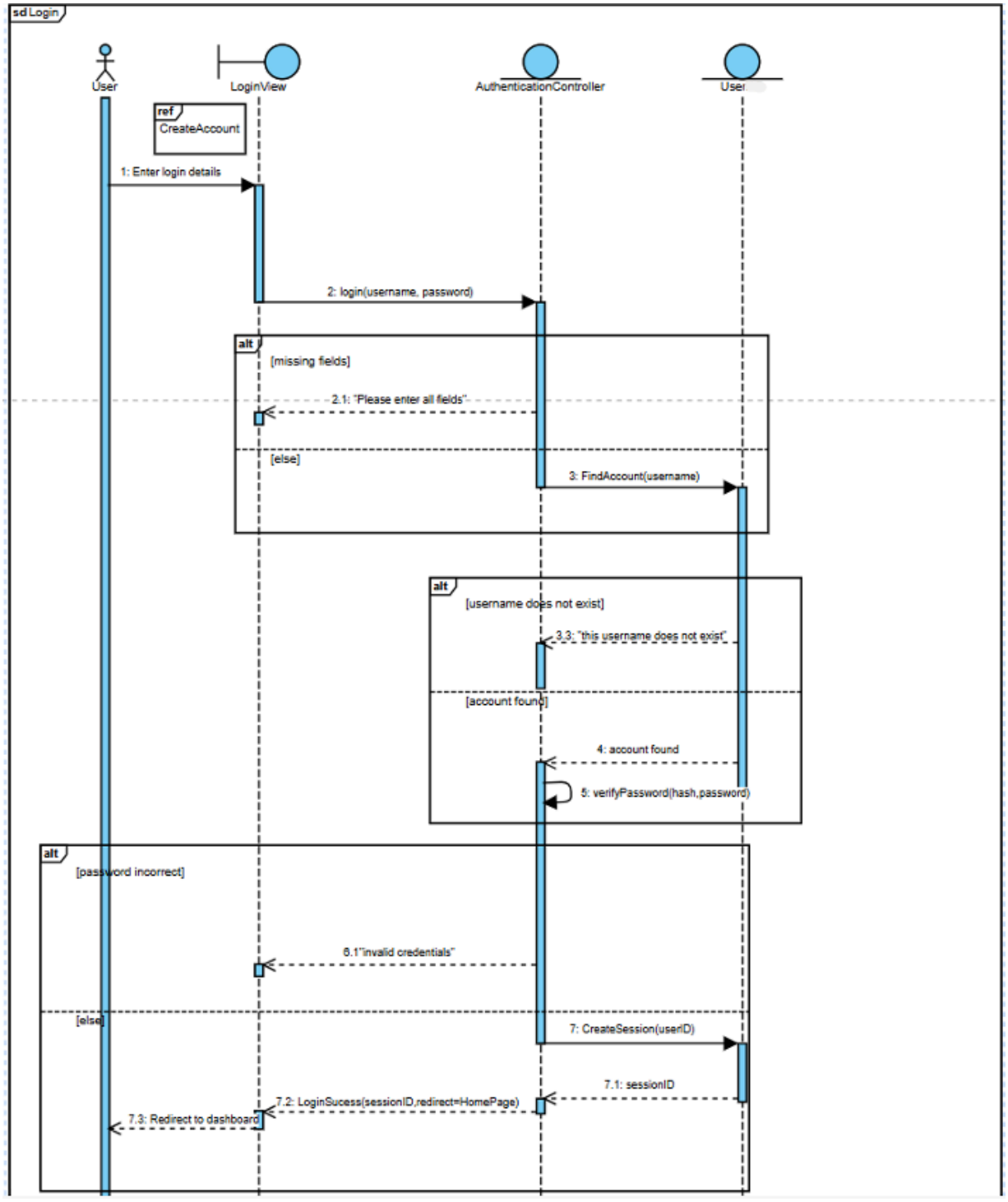
5. Sequence Diagrams of Some Use Cases

I. For Use Cases under #1 (Login)

I.I CreateAccount

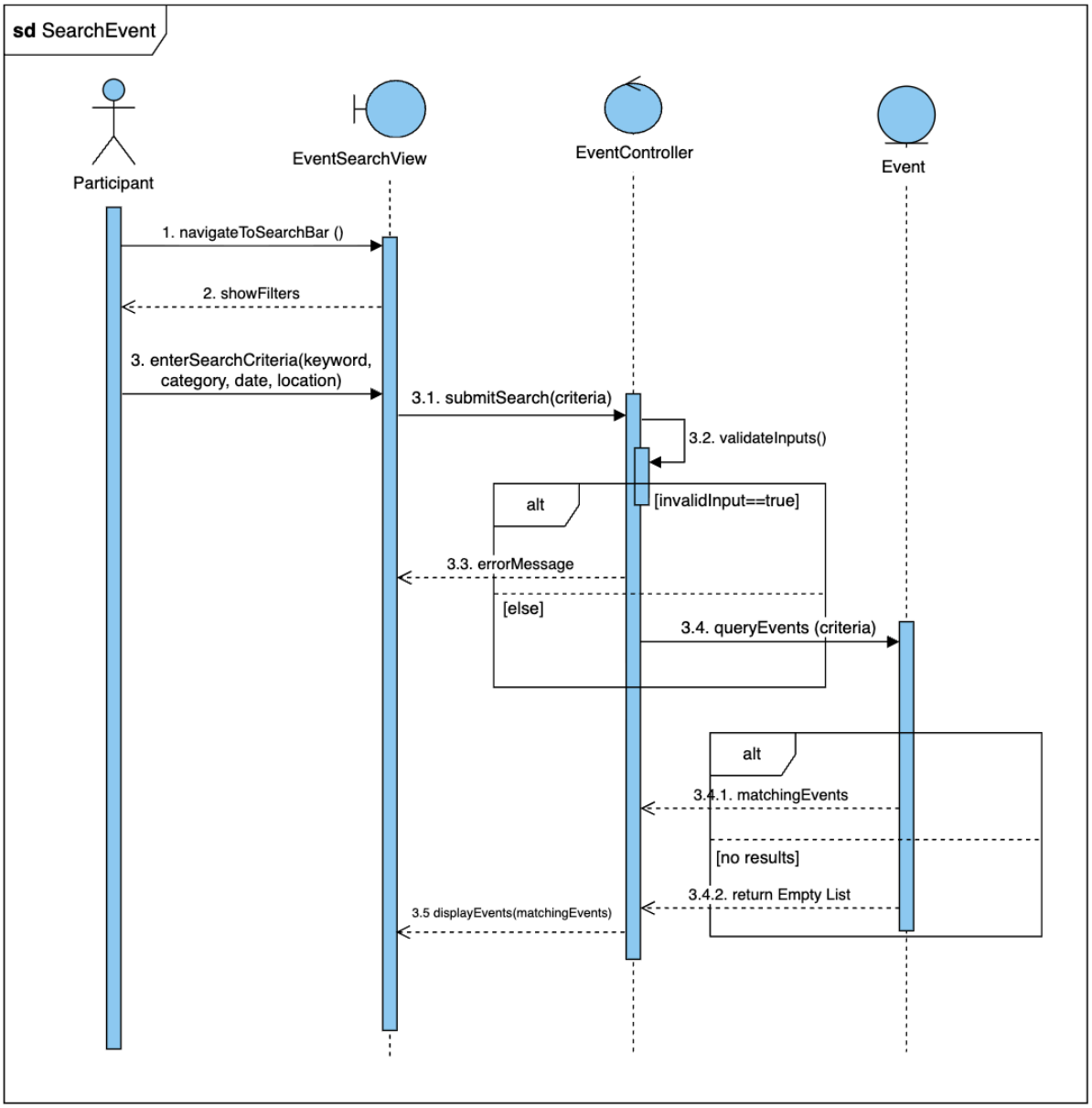


I.II Login

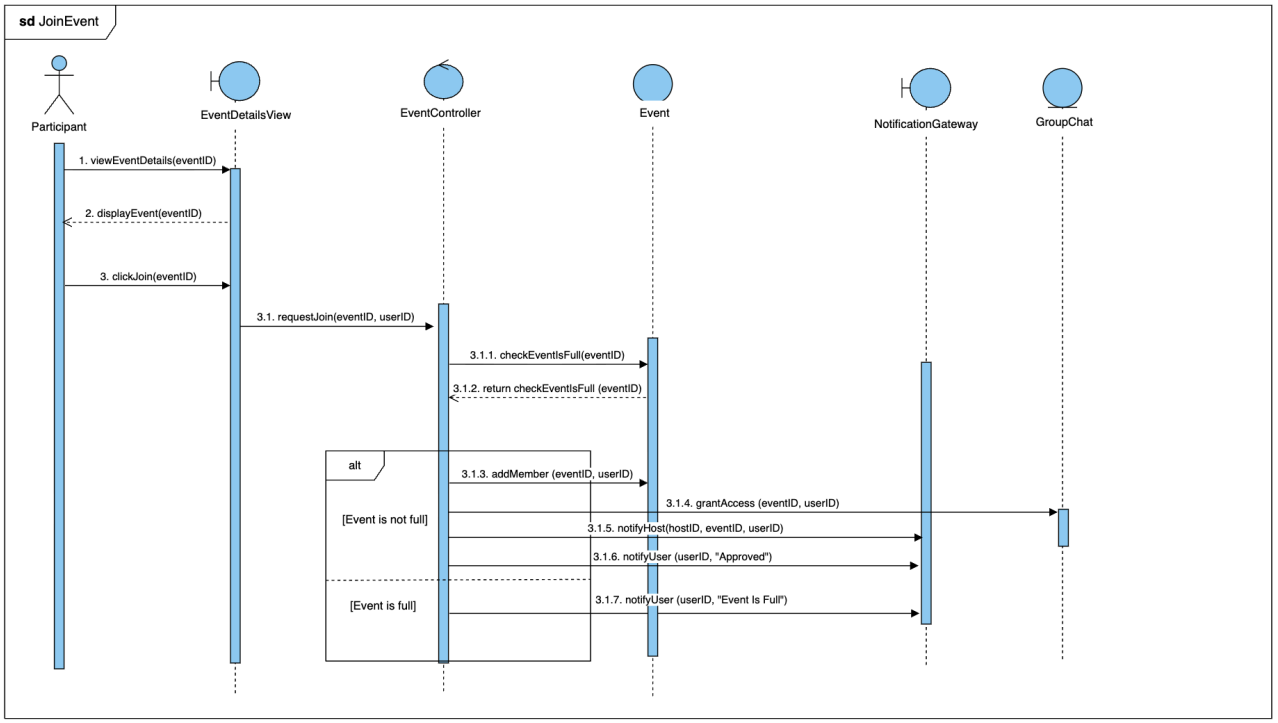

sd Login
User
LoginView
AuthenticationController
User
ref
CreateAccount
1: Enter login details
2: login(username, password)
alt
[missing fields]
2.1: "Please enter all fields"
[else]
3: FindAccount(username)
alt
[username does not exist]
3.3: "this username does not exist"
[account found]
4: account found
5: verifyPassword(hash,password)
alt
[password incorrect]
6.1"invalid credentials"
[else]
7: CreateSession(userID)
7.1: sessionID
7.2: LoginSucess(sessionID,redirect=HomePage)
7.3: Redirect to dashboard

II. For Use Cases under #2 (Participant)

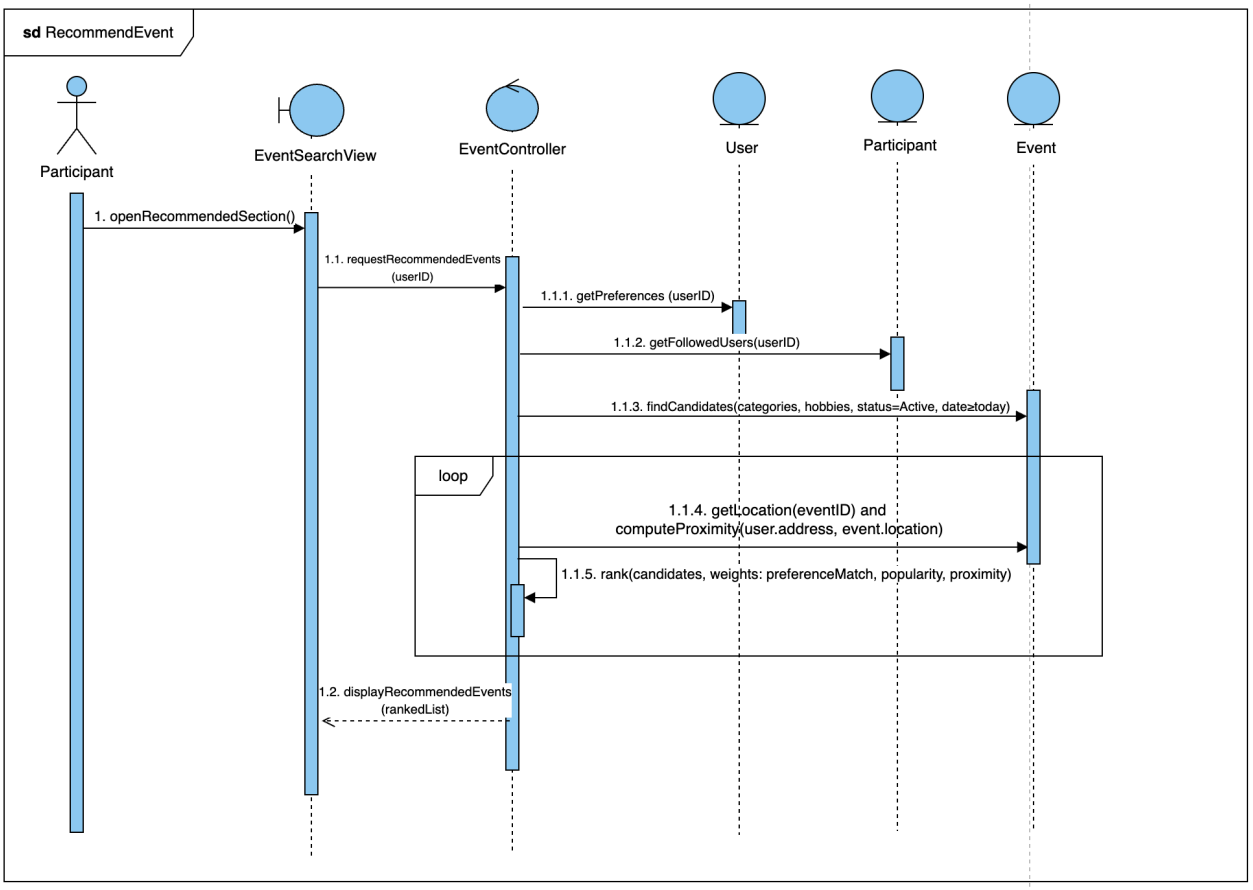
II.I SearchEvent



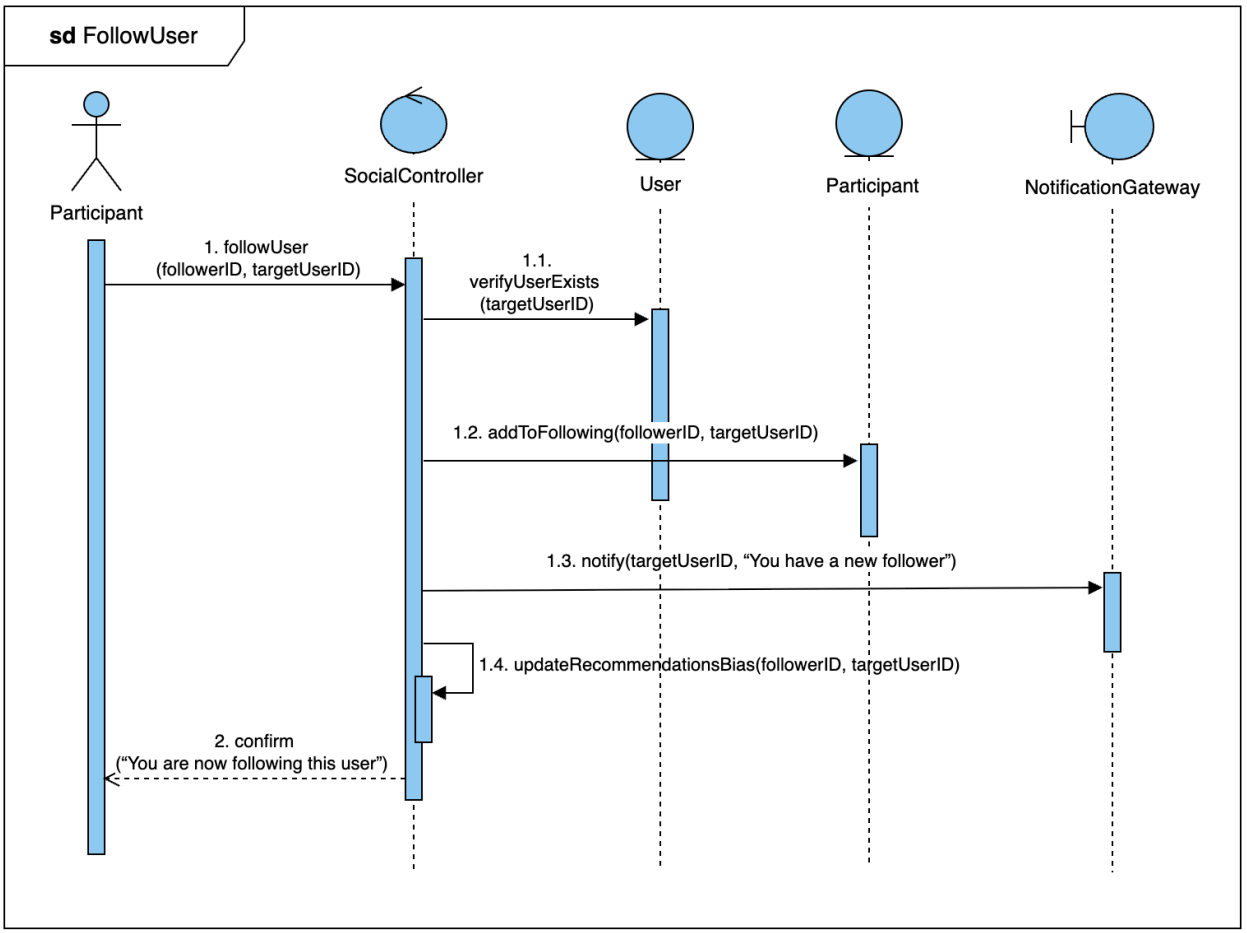
II.II JoinEvent



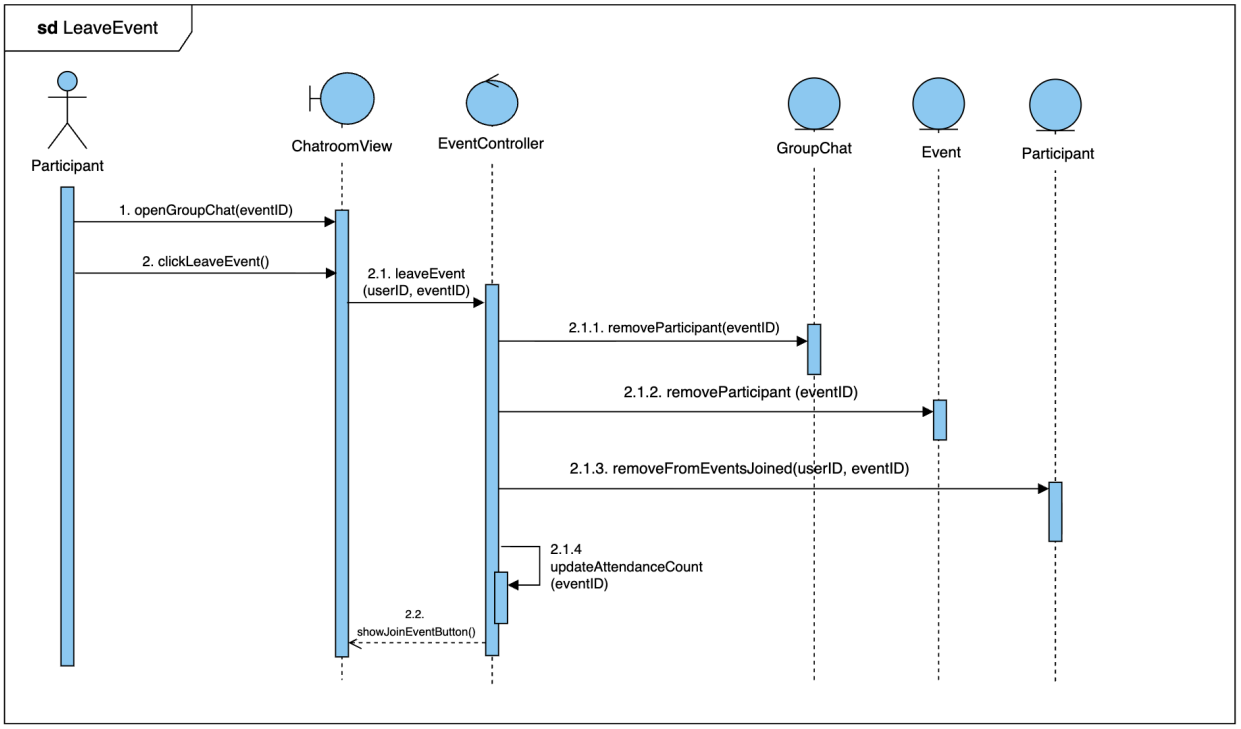
II.III RecommendEvent



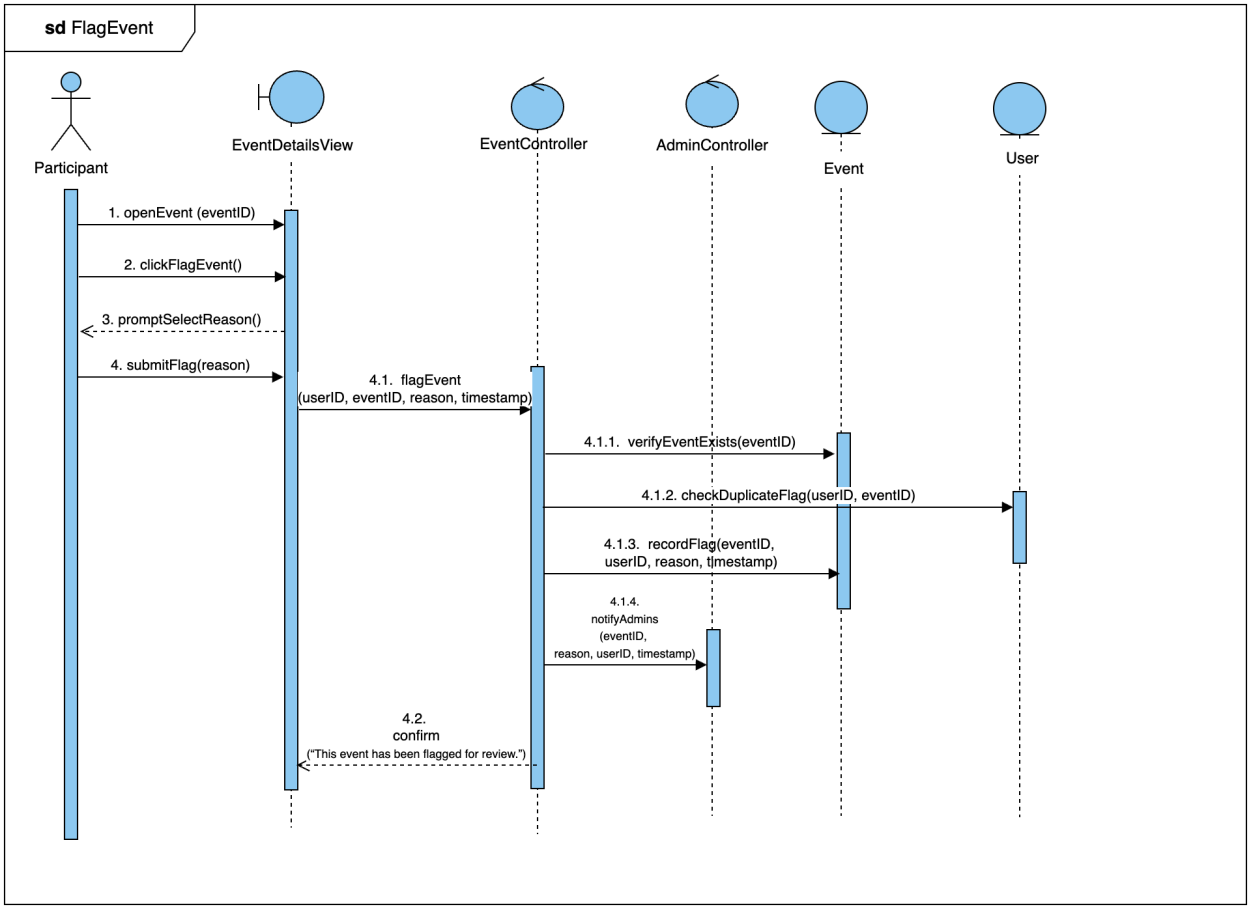
II.IV FollowUser



II.V LeaveEvent

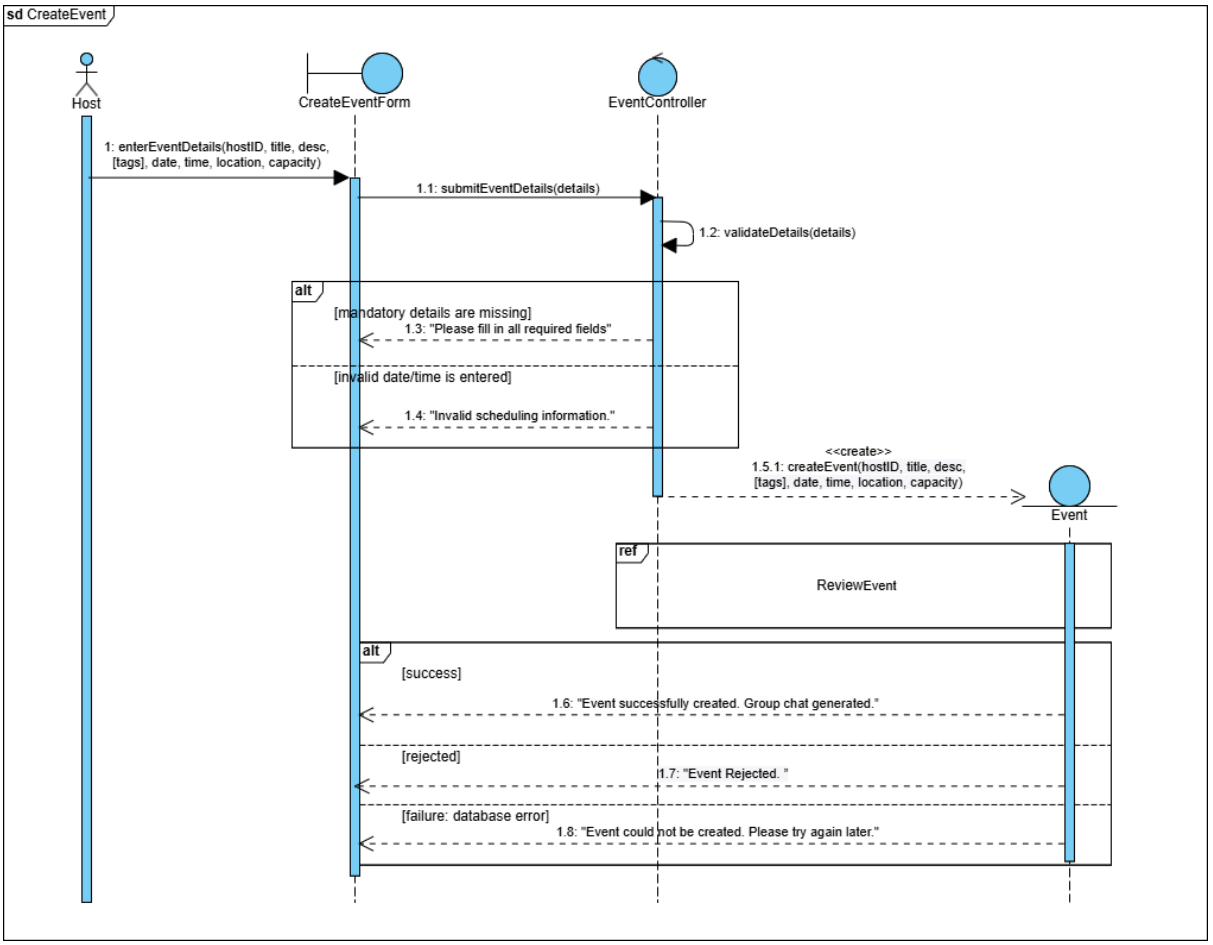


II.VI FlagEvent

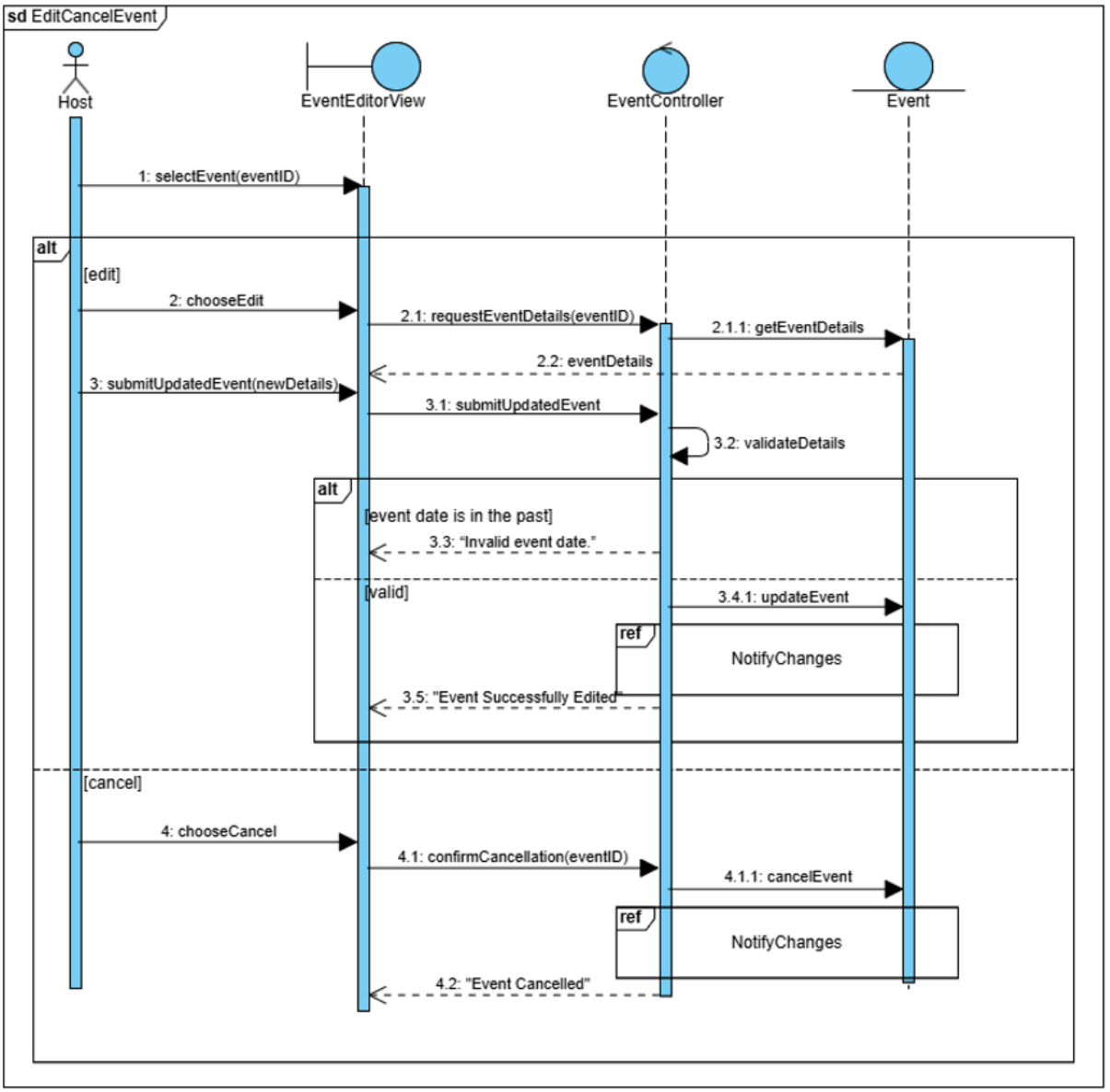


III. For Use Cases under #3 (Host)

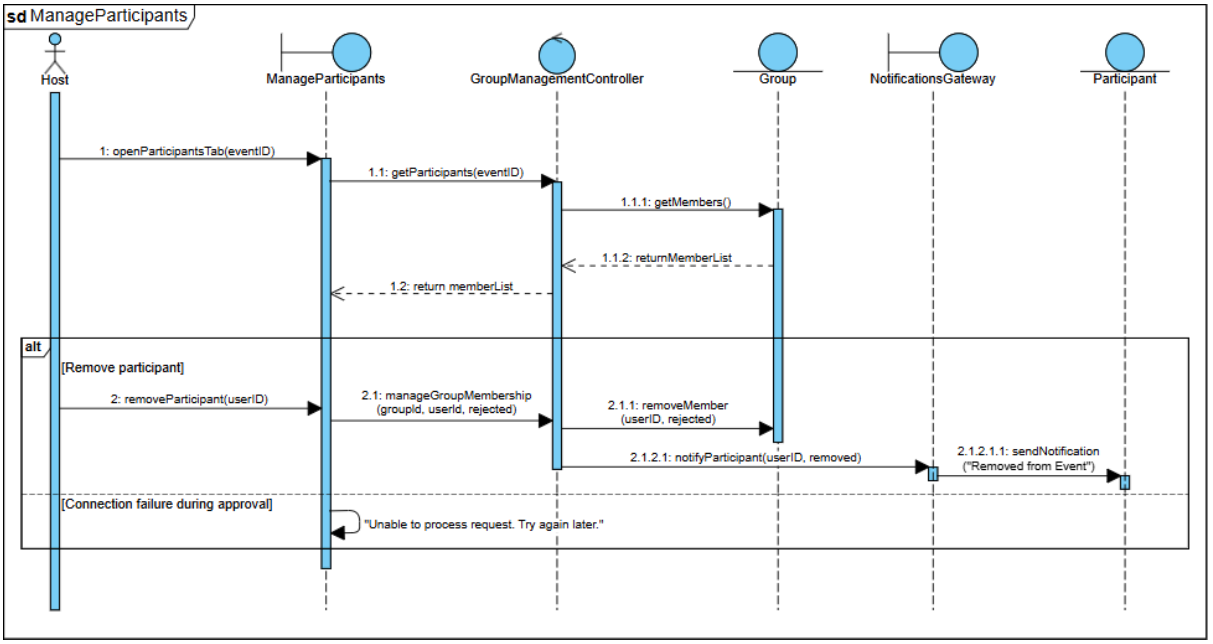
III.I CreateEvent



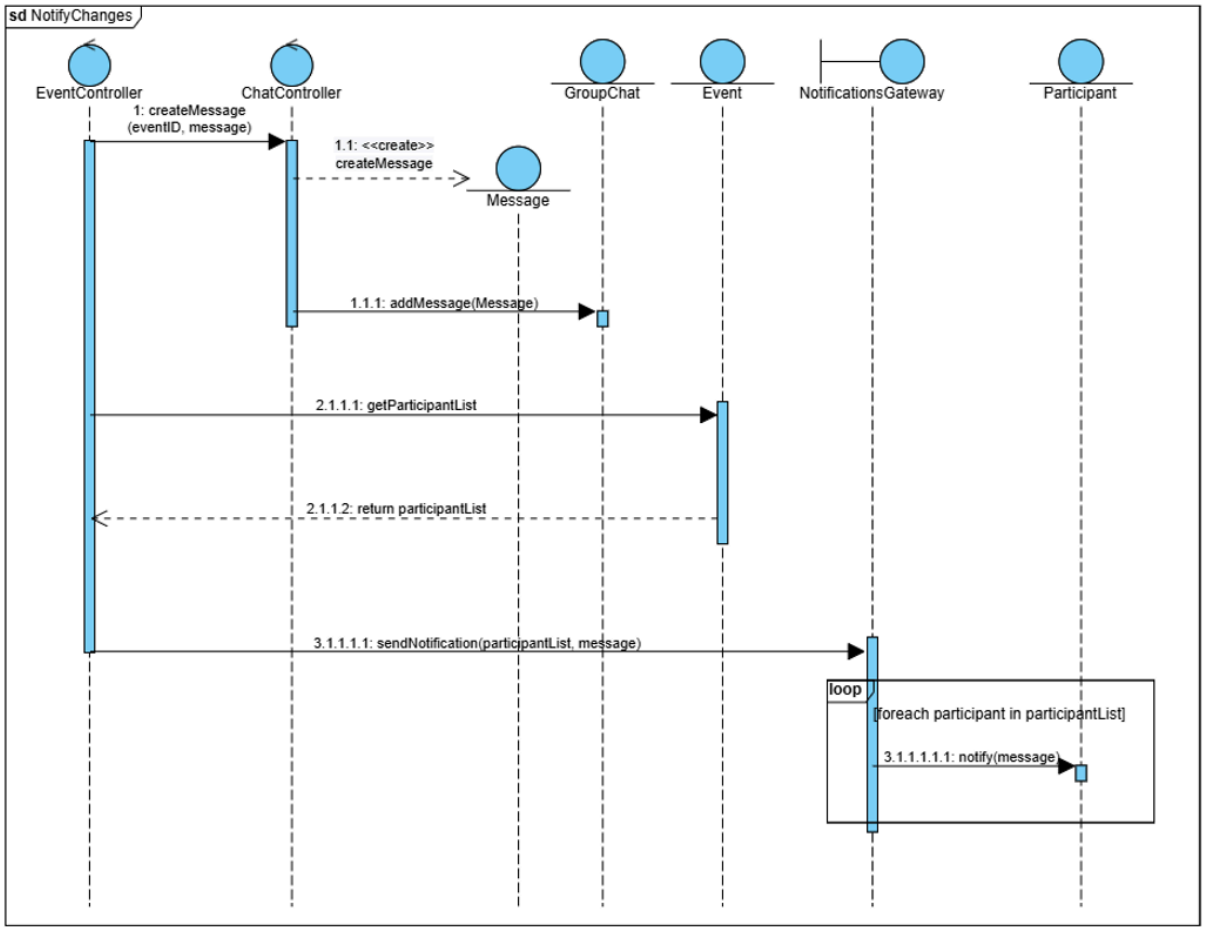
III.II EditCancelEvent



III.III ManageParticipants

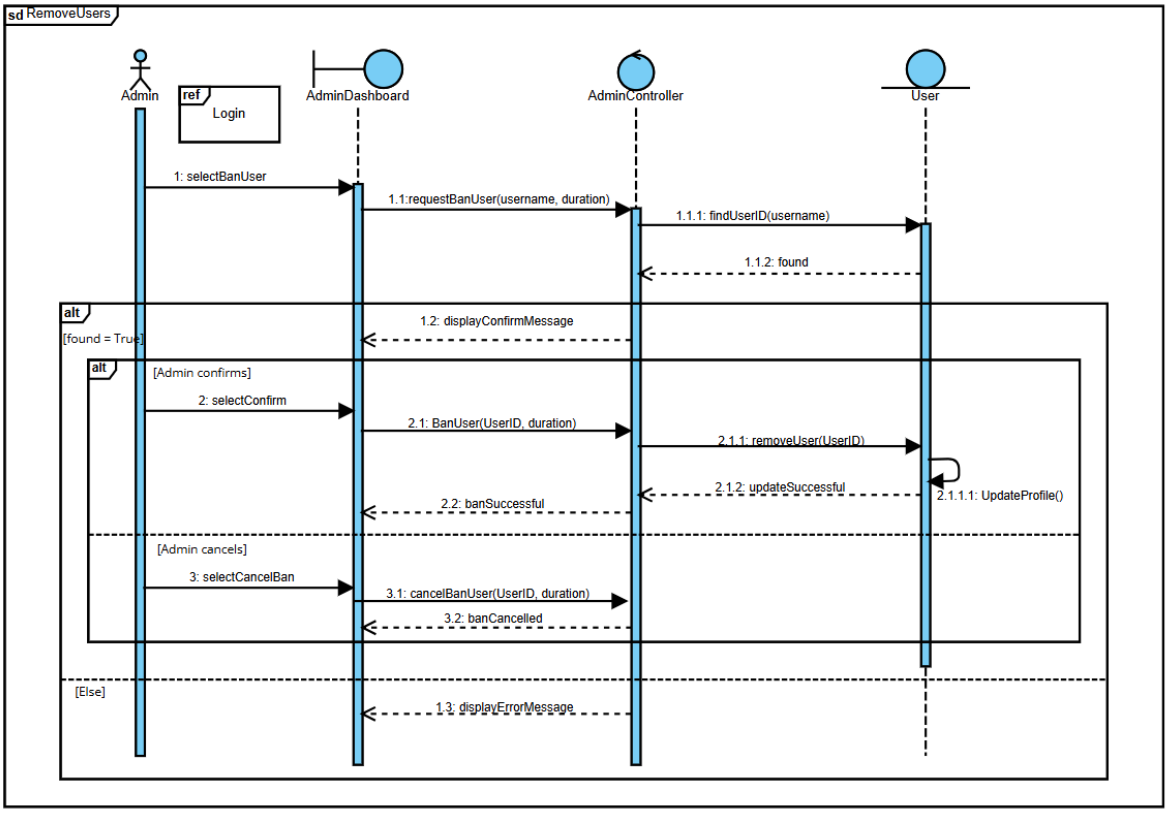


III.IV NotifyChanges

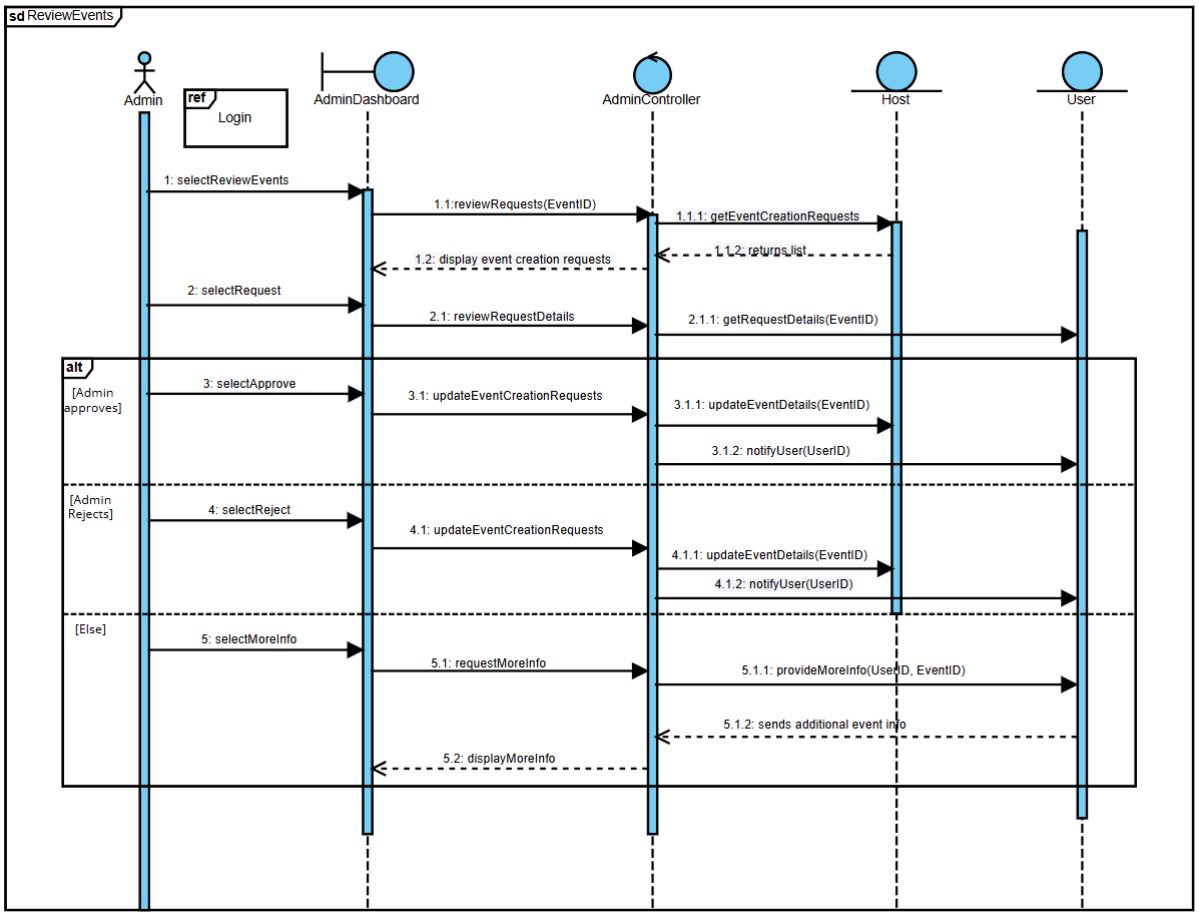


IV. For Use Cases under #4 (Admin)

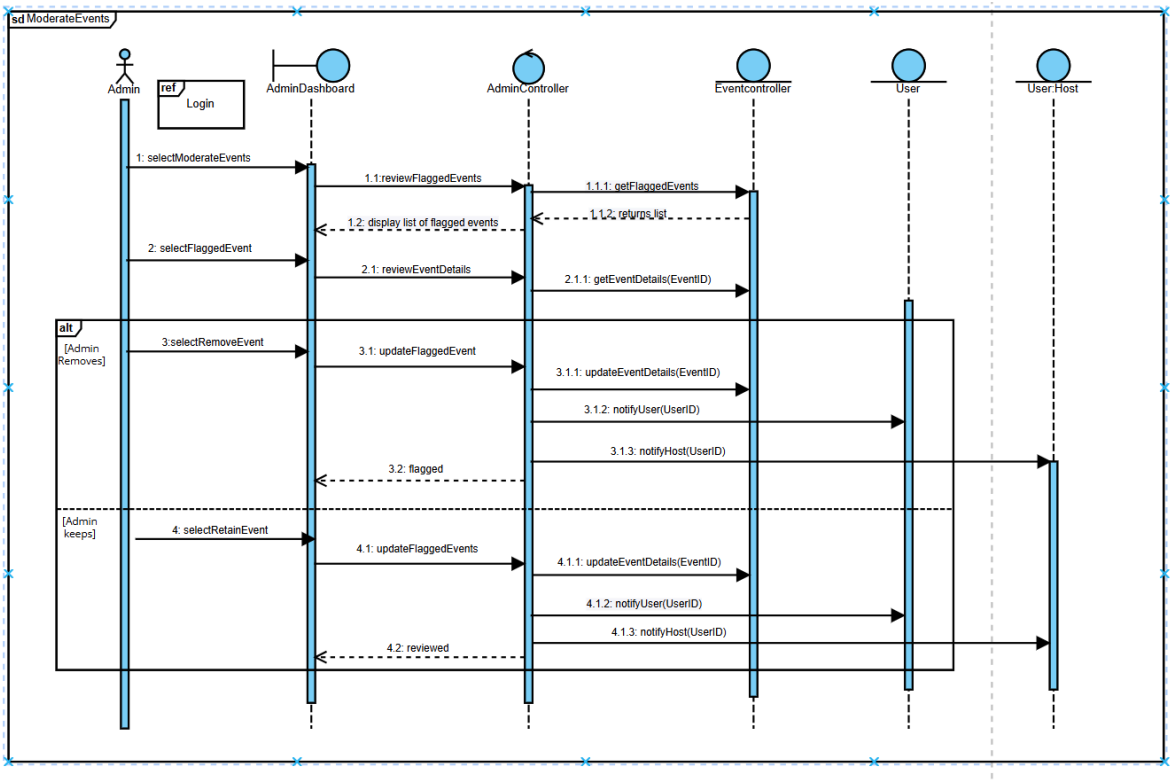
IV.I RemoveUsers



IV.II ReviewEvents

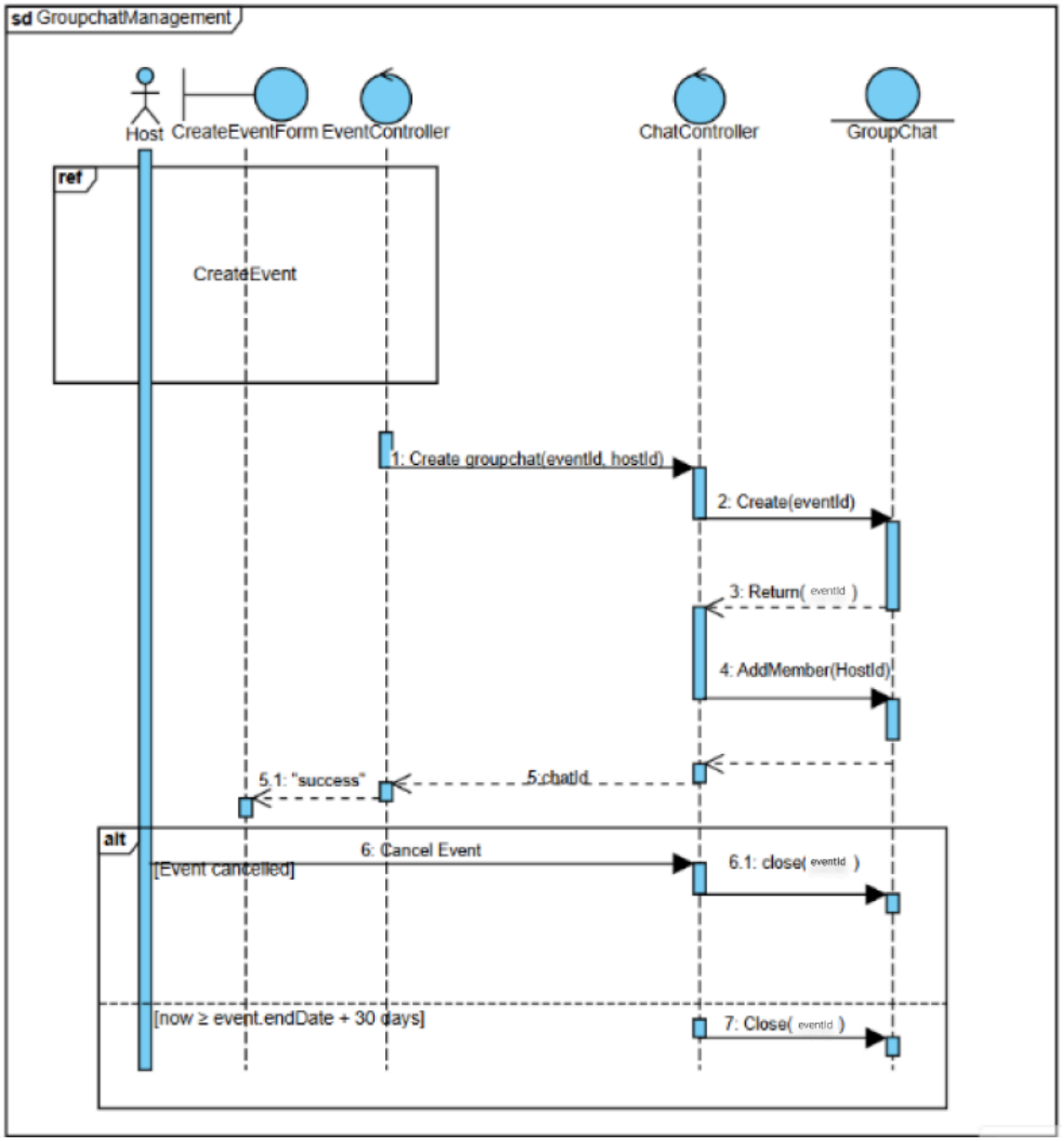


IV.III ModerateEvents

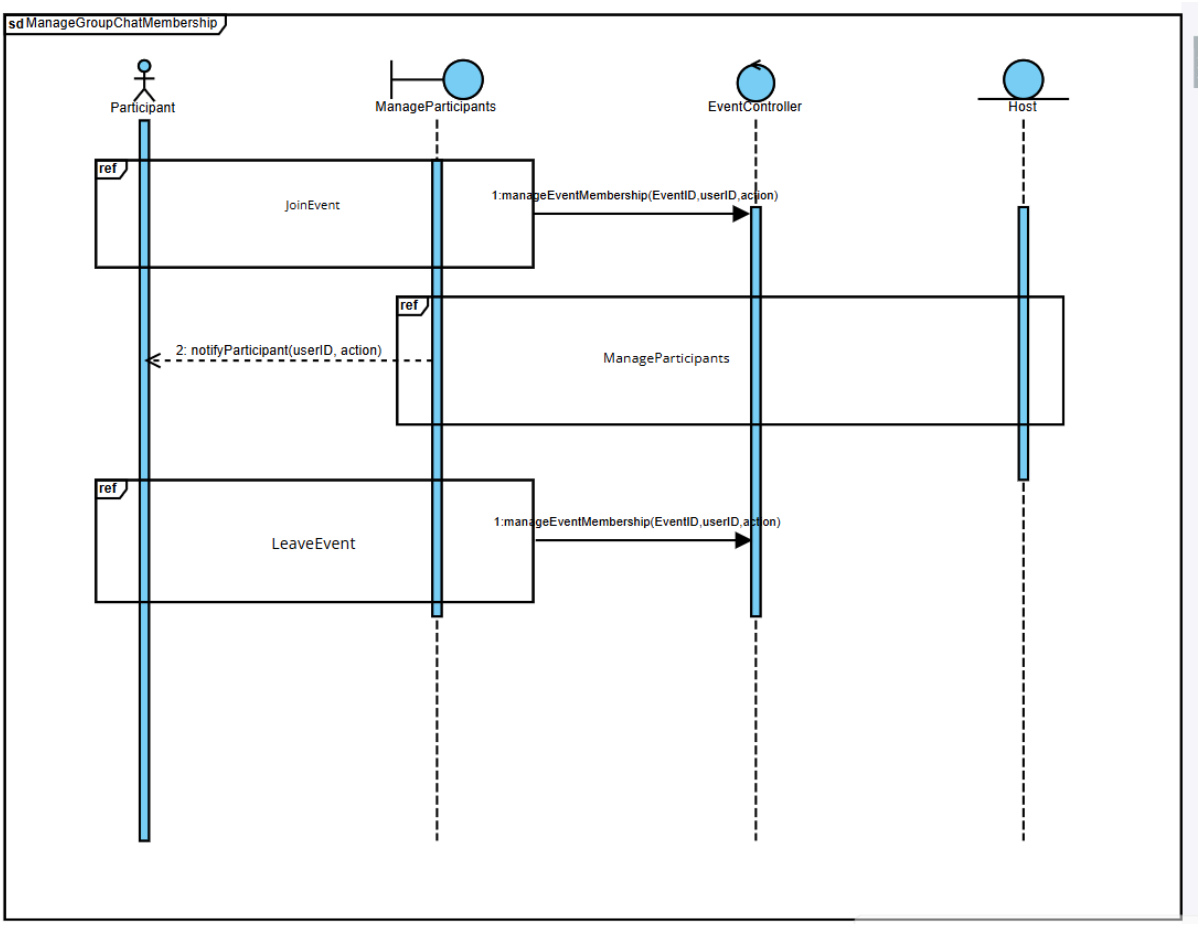


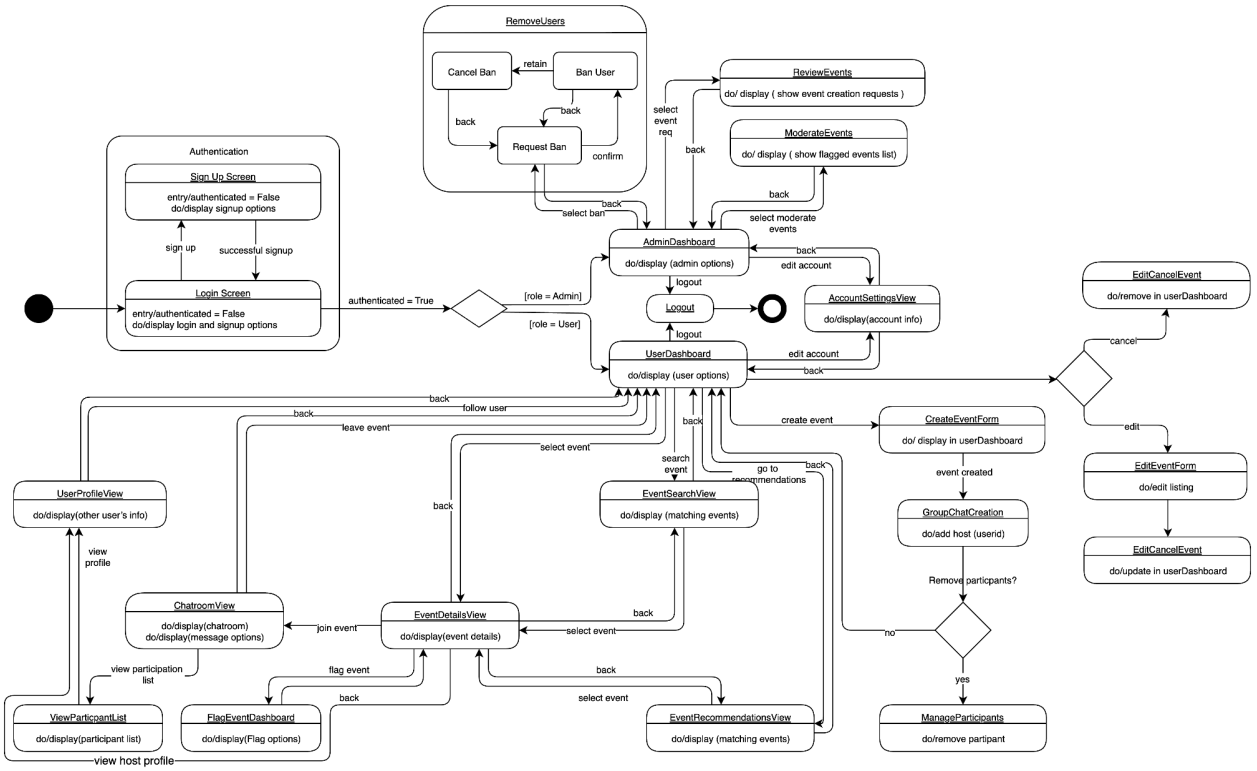
V. For Use Cases under #5 (Group Chats)

V.I GroupChatManagement



V.II ManageGroupChatMembership



6. Initial Dialog Map

If the image is unclear, please refer to the raw png file that is uploaded together with this document.