

Project Requirements Analysis

1. Fully Dressed Use Cases

Use Case 1: Sign Up for Events

1. **ID:** UC01
2. **Title:** User Registration
3. **Primary Actor:** User
4. **Goal in Context:** User registers for the app to access event listings.
5. **Preconditions:** User has the app installed and is not logged in.
6. **Trigger:** The user opens the app and chooses to sign up.
7. **Main Success Scenario:**
 1. User opens the app.
 2. User selects "Sign Up."
 3. User enters required information (email, password, etc.).
 4. System creates the user account.
 5. System confirms the account registration.
8. **Extensions:**
 1. 4a: If the email is already registered, the system shows an error.
 2. 4b: If the password is weak, the system prompts the user to choose a stronger password.
9. **Special Requirements:** User data should be encrypted for security.
10. **Technology and Data Variations List:** Users may sign up using a social media account.
11. **Frequency of Use:** Low (once per user).
12. **Open Issues:** None.

Use Case 2: Discover Events

1. **ID:** UC02
2. **Title:** Event Discovery
3. **Primary Actor:** User
4. **Goal in Context:** User finds events based on preferences like location, type, or date.
5. **Preconditions:** The user is logged in.
6. **Trigger:** The user opens the app and chooses to browse events.
7. **Main Success Scenario:**
 1. User opens the event discovery page.
 2. User inputs search criteria (location, type, date, etc.).
 3. System filters and displays relevant events.
 4. User selects an event for more details.

8. **Extensions:**
 1. 3a: If no events match the search criteria, the system displays a message saying "No events found."
 2. 4a: If the event details are unavailable, the system prompts the user to try again later.
9. **Special Requirements:** The system should load results quickly.
10. **Technology and Data Variations List:** Users can filter results by distance.
11. **Frequency of Use:** High.
12. **Open Issues:** None.

Use Case 3: Post Event

1. **ID:** UC03
2. **Title:** Event Creation
3. **Primary Actor:** Venue Manager
4. **Goal in Context:** Venue Manager creates a new event listing.
5. **Preconditions:** Venue Manager is logged in with proper permissions.
6. **Trigger:** Venue Manager selects "Create New Event" option.
7. **Main Success Scenario:**
 1. Venue Manager fills in event details (name, date, time, location, etc.).
 2. Venue Manager uploads event images.
 3. Venue Manager sets ticket prices and availability.
 4. System validates the information.
 5. System creates and publishes the event listing.
8. **Extensions:**
 1. 4a: If information is incomplete, the system prompts for missing details.
 2. 4b: If image upload fails, the system allows retry or skip.
9. **Special Requirements:** Events should be immediately visible after creation.
10. **Technology and Data Variations List:** Option for recurring event creation.
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 4: Purchase Tickets

1. **ID:** UC04
2. **Title:** Ticket Purchase
3. **Primary Actor:** User
4. **Goal in Context:** User buys tickets for an event through the app.
5. **Preconditions:** User is logged in and has selected an event.
6. **Trigger:** User clicks the "Buy Tickets" button on the event page.
7. **Main Success Scenario:**
 1. System displays available ticket types and prices.
 2. User selects desired tickets and quantity.
 3. User enters payment information.

4. System processes payment.
5. System confirms purchase and provides digital tickets.
8. **Extensions:**
 1. 3a: If payment fails, the system prompts for an alternative payment method.
 2. 4a: If tickets sell out during the transaction, the system notifies the user and cancels purchase.
9. **Special Requirements:** Secure payment processing.
10. **Technology and Data Variations List:** Multiple payment options (credit card, PayPal, etc.).
11. **Frequency of Use:** High.
12. **Open Issues:** None.

Use Case 5: Receive Notifications

1. **ID:** UC05
2. **Title:** Event Notifications
3. **Primary Actor:** User
4. **Goal in Context:** User receives timely notifications about relevant events.
5. **Preconditions:** User is logged in and has enabled notifications.
6. **Trigger:** New event is added or existing event is updated.
7. **Main Success Scenario:**
 1. System identifies relevant events based on user preferences.
 2. System creates a notification with event details.
 3. System sends push notifications to the user's device.
 4. User receives and views notification.
8. **Extensions:**
 1. 3a: If push notifications are disabled, the system sends email instead.
9. **Special Requirements:** Notifications should be timely and not excessive.
10. **Technology and Data Variations List:** In-app notifications option.
11. **Frequency of Use:** High.
12. **Open Issues:** None.

Use Case 6: Leave Review

1. **ID:** UC06
2. **Title:** Event Review
3. **Primary Actor:** User
4. **Goal in Context:** User leaves a review and rating for an attended event.
5. **Preconditions:** User has attended the event and is logged in.
6. **Trigger:** User selects "Leave Review" option on past event page.
7. **Main Success Scenario:**
 1. System displays review form.
 2. User rates event (e.g., 1-5 stars).
 3. User writes a review text.

4. User submits review.
5. System publishes reviews on the event page.
8. **Extensions:**
 1. 4a: If the review text is too short, the system prompts for more details.
9. **Special Requirements:** Users can only review events they've attended.
10. **Technology and Data Variations List:** Option to add photos to review.
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 7: Save Event

1. **ID:** UC07
2. **Title:** Bookmark Event
3. **Primary Actor:** User
4. **Goal in Context:** User saves an event for later reference.
5. **Preconditions:** User is logged in and viewing an event.
6. **Trigger:** User clicks "Save" or bookmark icon on event page.
7. **Main Success Scenario:**
 1. User clicks the save button.
 2. System adds events to the user's saved list.
 3. System confirms the save with visual feedback.
8. **Extensions:**
 1. 2a: If an event is already saved, the system removes it from the saved list.
9. **Special Requirements:** Saved events should sync across devices.
10. **Technology and Data Variations List:** Option to organize saved events into lists.
11. **Frequency of Use:** High.
12. **Open Issues:** None.

Use Case 8: Share Event

1. **ID:** UC08
2. **Title:** Event Sharing
3. **Primary Actor:** User
4. **Goal in Context:** User shares an event with friends.
5. **Preconditions:** User is logged in and viewing an event.
6. **Trigger:** User clicks the "Share" button on the event page.
7. **Main Success Scenario:**
 1. System displays sharing options (social media, messaging apps, etc.).
 2. User selects the sharing method.
 3. System generates shareable links or content.
 4. User sends shared content through the chosen method.
8. **Extensions:**
 1. 2a: If the sharing method requires login, the system prompts for authentication.
9. **Special Requirements:** Shared content should include key event details.

10. **Technology and Data Variations List:** In-app sharing with other users.
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 9: Filter Events

1. **ID:** UC09
2. **Title:** Event Filtering
3. **Primary Actor:** User
4. **Goal in Context:** User applies filters to refine event search results.
5. **Preconditions:** User is logged in and on the event search page.
6. **Trigger:** User selects "Filter" option.
7. **Main Success Scenario:**
 1. System displays filter options (date range, event type, price, etc.).
 2. User selects desired filters.
 3. User applies filters.
 4. System updates event list based on selected filters.
8. **Extensions:**
 1. 4a: If no events match filters, the system suggests broadening criteria.
9. **Special Requirements:** Filtering should be fast and responsive.
10. **Technology and Data Variations List:** Save favorite filter combinations.
11. **Frequency of Use:** High.
12. **Open Issues:** None.

Use Case 10: View Event Map

1. **ID:** UC10
2. **Title:** Event Map View
3. **Primary Actor:** User
4. **Goal in Context:** User views events on a map interface.
5. **Preconditions:** User is logged in and on the event discovery page.
6. **Trigger:** User selects "Map View" option.
7. **Main Success Scenario:**
 1. System displays a map with event locations marked.
 2. User navigates the map (zoom, pan).
 3. User taps the event marker for brief details.
 4. User selects events for full details.
8. **Extensions:**
 1. 1a: If location services are disabled, system prompts to enable.
9. **Special Requirements:** Map should load quickly and be smooth to navigate.
10. **Technology and Data Variations List:** Integration with map providers (Google Maps, Apple Maps).
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 11: Set Event Reminder

1. **ID:** UC11
2. **Title:** Event Reminder
3. **Primary Actor:** User
4. **Goal in Context:** User sets a reminder for an upcoming event.
5. **Preconditions:** User is logged in and viewing an event.
6. **Trigger:** User selects "Set Reminder" option.
7. **Main Success Scenario:**
 1. System displays reminder options (time before event).
 2. User selects the reminder time.
 3. System confirms reminder set.
 4. System sends reminders at a specified time.
8. **Extensions:**
 1. 4a: If a user has notifications disabled, system prompts to enable.
9. **Special Requirements:** Reminders should be accurate and timely.
10. **Technology and Data Variations List:** Multiple reminder options (push, email, SMS).
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 12: Update User Profile

1. **ID:** UC12
2. **Title:** Profile Update
3. **Primary Actor:** User
4. **Goal in Context:** User updates their profile information.
5. **Preconditions:** User is logged in.
6. **Trigger:** User selects "Edit Profile" option.
7. **Main Success Scenario:**
 1. System displays editable profile fields.
 2. User modifies desired information.
 3. User saves changes.
 4. System updates user profile.
8. **Extensions:**
 1. 3a: If changes are invalid, the system highlights errors and prompts for correction.
9. **Special Requirements:** Profile updates should be instant.
10. **Technology and Data Variations List:** Option to sync profile with social media.
11. **Frequency of Use:** Low.
12. **Open Issues:** None.

Use Case 13: View Event Statistics (for Venues)

1. **ID:** UC13

2. **Title:** Event Analytics
3. **Primary Actor:** Venue Manager
4. **Goal in Context:** Venue Manager views statistics for their events.
5. **Preconditions:** Venue Manager is logged in.
6. **Trigger:** Venue Manager selects "Event Analytics" option.
7. **Main Success Scenario:**
 1. System displays a list of past and upcoming events.
 2. Venue Manager selects an event.
 3. System shows event statistics (views, ticket sales, etc.).
 4. Venue Manager reviews data.
8. **Extensions:**
 1. 3a: If data is still processing, the system shows a loading message.
9. **Special Requirements:** Data should be updated in real-time.
10. **Technology and Data Variations List:** Option to export data.
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 14: Contact Event Organizer

1. **ID:** UC14
2. **Title:** Organizer Contact
3. **Primary Actor:** User
4. **Goal in Context:** User sends a message to the event organizer.
5. **Preconditions:** User is logged in and viewing an event.
6. **Trigger:** User selects "Contact Organizer" option.
7. **Main Success Scenario:**
 1. System displays a message form.
 2. User writes a message.
 3. User sends a message.
 4. System delivers a message to the organizer.
8. **Extensions:**
 1. 4a: If a message fails to send, the system prompts retry.
9. **Special Requirements:** Messages should be delivered promptly.
10. **Technology and Data Variations List:** Option for public Q&A on event page.
11. **Frequency of Use:** Low.
12. **Open Issues:** None.

Use Case 15: View Ticket History

1. **ID:** UC15
2. **Title:** Ticket History
3. **Primary Actor:** User
4. **Goal in Context:** User views their past and upcoming event tickets.
5. **Preconditions:** User is logged in.

6. **Trigger:** User selects "My Tickets" option.
7. **Main Success Scenario:**
 1. System displays a list of user's tickets.
 2. Users can view ticket details.
 3. Users can access digital tickets for upcoming events.
8. **Extensions:**
 1. 3a: If the ticket is transferable, the user can transfer to another user.
9. **Special Requirements:** Digital tickets should be easily accessible.
10. **Technology and Data Variations List:** Option to add tickets to mobile wallet.
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 16: Report Inappropriate Content

1. **ID:** UC16
2. **Title:** Content Reporting
3. **Primary Actor:** User
4. **Goal in Context:** User reports inappropriate event or user content.
5. **Preconditions:** User is logged in and viewing content.
6. **Trigger:** User selects "Report" option.
7. **Main Success Scenario:**
 1. System displays reporting reasons.
 2. User selects reason for report.
 3. User submits a report.
 4. System confirms report receipt.
8. **Extensions:**
 1. 2a: If "Other" is selected, user can type custom reason.
9. **Special Requirements:** Reports should be reviewed promptly.
10. **Technology and Data Variations List:** Option to block user/event after reporting.
11. **Frequency of Use:** Low.
12. **Open Issues:** None.

Use Case 17: Customize Notification Settings

1. **ID:** UC17
2. **Title:** Notification Preferences
3. **Primary Actor:** User
4. **Goal in Context:** User customizes their notification preferences.
5. **Preconditions:** User is logged in.
6. **Trigger:** User selects "Notification Settings" option.
7. **Main Success Scenario:**
 1. System displays notification options.
 2. User toggles desired notifications on/off.
 3. User saves preferences.

4. System updates notification settings.
8. **Extensions:**
 1. 2a: User can set quiet hours for notifications.
9. **Special Requirements:** Changes should apply immediately.
10. **Technology and Data Variations List:** Separate settings for push, email, and in-app notifications.
11. **Frequency of Use:** Low.
12. **Open Issues:** None.

Use Case 18: Create Event Playlist

1. **ID:** UC18
2. **Title:** Event Playlist Creation
3. **Primary Actor:** User
4. **Goal in Context:** User creates a playlist of multiple events.
5. **Preconditions:** User is logged in.
6. **Trigger:** User selects "Create Playlist" option.
7. **Main Success Scenario:**
 1. User names the playlist.
 2. User searches for events to add.
 3. User adds events to the playlist.
 4. System saves the playlist.
8. **Extensions:**
 1. 3a: If an event is already in the playlist, the system notifies the user.
9. **Special Requirements:** Playlists should be shareable.
10. **Technology and Data Variations List:** Option to make a playlist public or private.
11. **Frequency of Use:** Medium.
12. **Open Issues:** None.

Use Case 19: View Event Venue Information

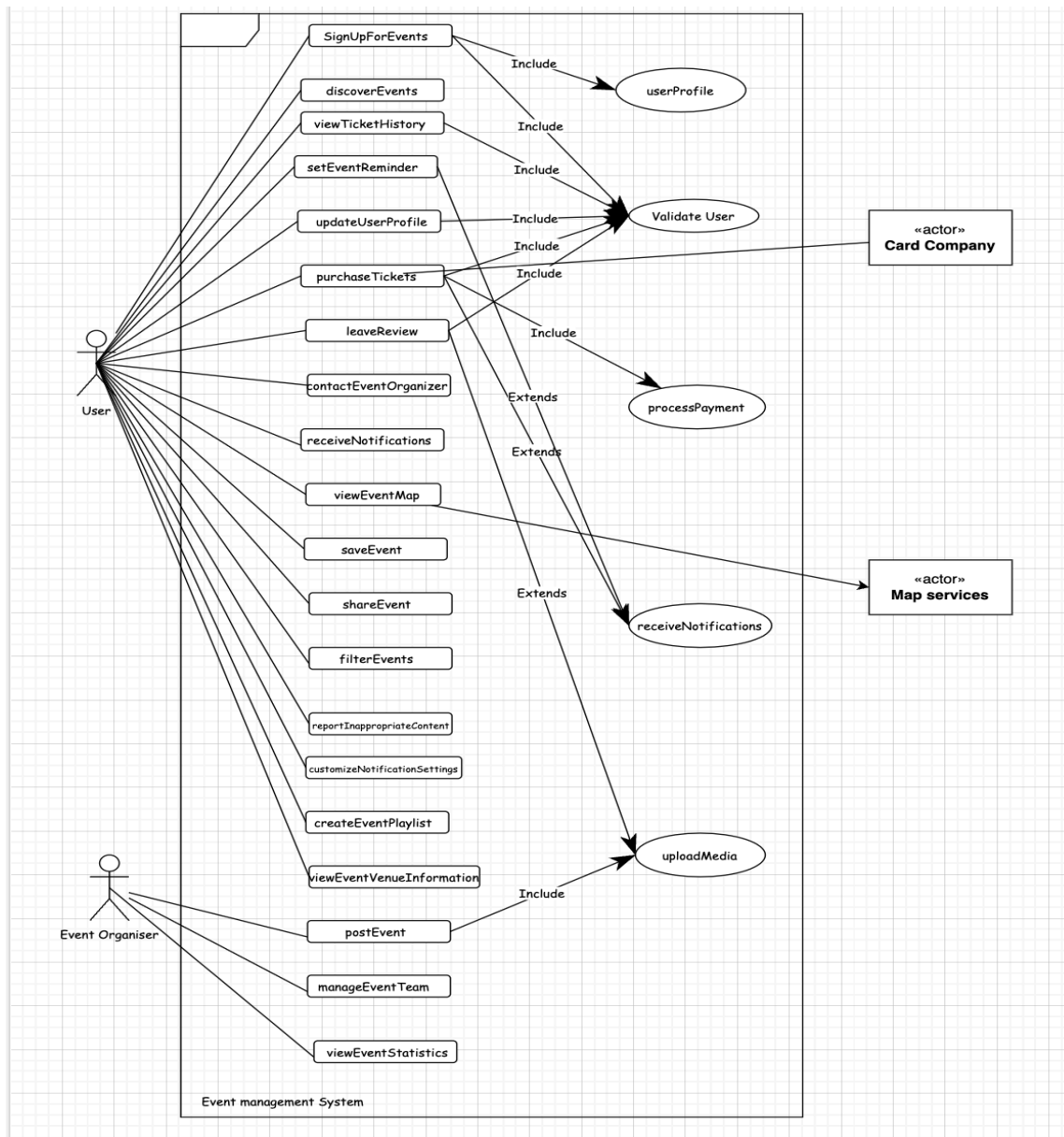
1. **ID:** UC19
2. **Title:** Venue Information
3. **Primary Actor:** User
4. **Goal in Context:** User views detailed information about an event venue.
5. **Preconditions:** User is viewing an event.
6. **Trigger:** User selects "Venue Info" option.
7. **Main Success Scenario:**
 1. System displays venue details (address, capacity, amenities).
 2. User can view venue photos.
 3. User can access venue contact information.
 4. User can view the map location of the venue.
8. **Extensions:**
 1. 4a: Users can get directions to the venue from current location.

9. **Special Requirements:** Venue information should be up-to-date.
10. **Technology and Data Variations List:** Virtual tour of venue if available.
11. **Frequency of Use:** High.
12. **Open Issues:** None.

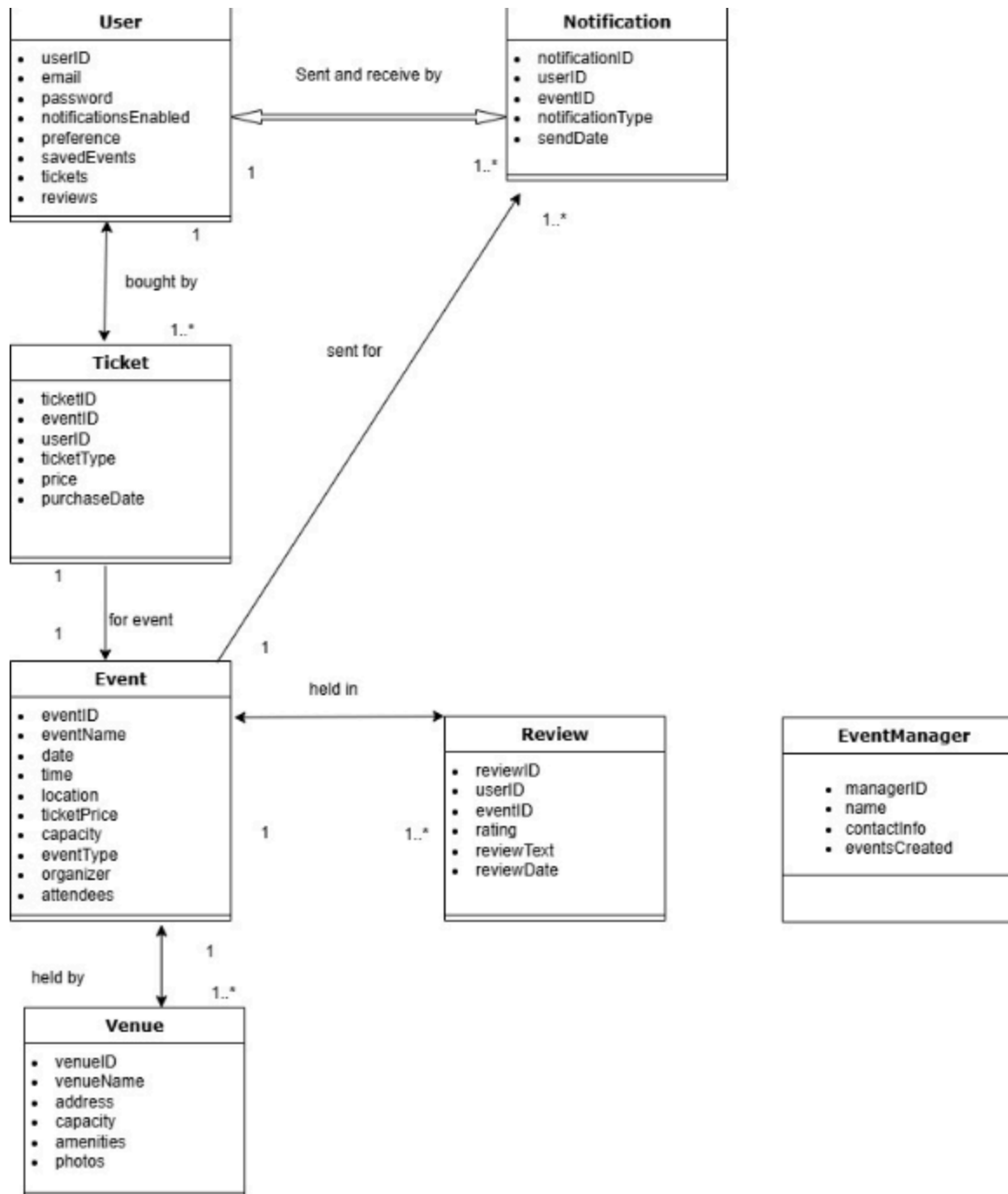
Use Case 20: Manage Event Team (for Organizers)

1. **ID:** UC20
 2. **Title:** Event Team Management
 3. **Primary Actor:** Event Organizer
 4. **Goal in Context:** Organizer manages team members for an event.
 5. **Preconditions:** Organizer is logged in and has created an event.
 6. **Trigger:** Organizer selects "Manage Team" for an event.
 7. **Main Success Scenario:**
 1. System displays current team members and roles.
 2. Organizers can add new team members.
 3. Organizers can assign or change roles.
 4. Organizers can remove team members.
 5. System updates team information.
 8. **Extensions:**
 1. 2a: If adding members by email, the system sends invitations.
 9. **Special Requirements:** Role changes should update permissions immediately.
 10. **Technology and Data Variations List:** Integration with staff management software.
 11. **Frequency of Use:** Low.
 12. **Open Issues:** None.
-

2. Use Case Diagrams



3. Conceptual Class Diagrams



4. Supplementary Specifications

Non-functional Requirements:

- App should handle many users concurrently without slowing down and ensure the features event filtering and playlist are fast and responsive.

- User data must be secure, applied to sensitive information such as login credentials and payments details.
- App should load quickly, provided responsive interaction for event discover and map, ensure time-sensitive functionalities like reminder are accurate
- App should work on both mobile and desktop, features like saving events and notification sync across all devices.
- Digital tickets and event management must be consistently available and easy to access, making sure users can rely on these features without delays or disruptions.
- Apps should support multiple languages, allowing users to choose their preferred language.