1. **User Model**:
   * Each user has a list of events they have some relationship with (**created**, **attending**, or **interested**).
   * It's embedded in the user's document, so fetching a user's profile gives instant access to this list.
2. **Event Model**:
   * The event retains the **createdBy** field pointing to the user who created it. This makes it easy to ensure only the creator can edit or delete the event.
   * Each event has a list of **attendees**, which points back to users and their status (**attending** or **interested**).

This setup gives me the flexibility to query the events a user is associated with and also fetch the users associated with an event (either attending or interested).

**For operations:**

1. **When a user creates an event**:
   * Add the event to the **Event** collection.
   * Update the user's **events** array to include this new event with the relationship **created**.
2. **When a user wants to attend or show interest in an event**:
   * Update the user's **events** array with the new event and the appropriate relationship (**attending** or **interested**).
   * Update the event's **attendees** array to include this user and their status.
3. **When a user wants to stop attending or remove interest**:
   * Remove the event from the user's **events** array.
   * Update or remove the user from the event's **attendees** array.
4. **When an event is deleted**:
   * Remove the event from the **Event** collection.
   * Update all users to remove this event from their **events** array.
   * (Note: I must ensure this operation is handled carefully, as multiple documents are being affected).