

We can use Firebase Anonymous Auth to

Entity: User

Attributes:

- userID: String, document id autogenerated by Firestore OR we make the DeviceID the primary key (this is the document name)
- deviceID: String
- name: string
- phoneNumber: string
- email: string
- isAdmin: boolean
- notificationSettings: TBD

User is Many to Many with Event

Entity: Event

Attributes:

- organizerID: String, references UserID from User table
- eventID: String, document id autogenerated by Firestore (this is the document name)
- eventTags: Array of strings
- description: string
- **Registration Subcollection**
 - UserID: String, Document ID, references UserID from User table (when creating a document in this subcollection make the document ID the UserID that is registering)
 - Status: String, we will treat this as an enum ("waiting", "chosen", "cancelled", "finalized")
 - Location: the location where the user signed up for the event (likely just latitude and longitude floats or doubles stored in memory)
 - OrganizerID: String
- PosterImageURL: string to img url
- RegistrationStartDate: datetime
- RegistrationEndDate: datetime
- EventStartDate: datetime
- EventEndDate: datetime
- QRCode: either store the image or an encoding string that can be decoded to create the qr code later on the app
- WaitingListLimit: Number (if negative value then no limit)
- MaxFinalListCapacity: Number
- (Optional) GeolocationWaitingList: array of tuples (latitude, longitude) of entrants on the waiting list

User is Two to Many with Notification in SenderID, RecipientID

Note: We could put Notification as a subcollection under Event if notifications are only sent by event organizers

Entity: Notification

Attributes:

- NotificationID: String, document id autogenerated by Firestore
- SenderID: String, references UserID from the User collection
- RecipientID: String, references UserID from the User collection
- Message: string
- Timestamp: datetime

Listeners

- In order to have code guaranteed to run after a listener gets data, make sure to have the code be called in the listener
-