Class Name	**Attendee**
Responsibilities	Collaborations
- Maintain a record of events the attendee has registered for, including status (confirmed, waiting, or cancelled). This information will also be stored on Firebase - Store and update the attendee's opt-in/opt-out preferences for receiving notifications about events.	- **Event** - **Notification Service**

Class Name	**EntrantList**
Responsibilities	Collaborations
- Manage the list of entrants who have registered for an event. - Update and display the status of each entrant (e.g., confirmed, waiting, or cancelled). Enable real-time updates to entrant statuses based on their registration progress or actions taken by organizers. For example, entrants can be marked as "confirmed" when their attendance is guaranteed, "waiting" if they're on the waitlist, or "cancelled" if their registration is withdrawn. - Provide functionality to add and remove entrants. Allow event organizers to add new entrants to the list or remove existing ones. This can include handling last-minute registrations, managing replacements, or removing entrants who cancel their participation.	- **Event** - **NotificationService** - **WaitingList**

Class Name	**Facility**
Responsibilities	Collaborations
- Store information about the location for the event and the facility name. Maintain a detailed database of event locations, including facility name, location, and facility image - Manage facility availability for event scheduling allowing users to check the availability of facilities in real-time.	- **Event** - **StorageImage**

Class Name	**Event**
Responsibilities	Collaborations
- Store information such as event name, date, time, maximum attendees, registration dates, poster, and geolocation requirements. - Add or remove entrants from the waitlist. - Maintain real-time information about attendees (waiting, cancelled, confirmed). - Track event status: (open/closed, participant limit) - Handle QR Code Data: Link generated QR codes to specific events. - Validate whether geolocation requirements are met for participants. - Integrate with NotificationService to send updates.	- **WaitingList** - **NotificationSystem** - **User** - **StorageImage**

Class Name	**StorageImage**
Responsibilities	Collaborations
- Manage image uploads for event posters and profile pictures and facility images. Enable users to easily upload and preview images for event promotion, attendee profiles, and Facility visuals, ensuring the images meet format and size requirements. - Store images in Firebase	- **Event** - **User** - **StorageImage**

Class Name	**NotificationService**
Responsibilities	Collaborations
- Send notifications to entrants when they are chosen from the waitlist (win/lose lottery). Automatically notify entrants of their selection or rejection from the waitlist, ensuring clear communication about their status. - Handle opt-in/opt-out preferences for notifications from organizers and admins. Allowing users to control the type of notifications they receive to respect their preferences and reduce unwanted alerts. - Send notifications to entrants about event updates. Keep entrants informed about changes, cancellations, or reminders for events they are participating in or waitlisted for.	- **Attendee** - **Event** - **User**

Class Name	**User**
Responsibilities	Collaborations
- Store information such as name, email, birthdate, optional phone number, country, notification preferences. Profile picture can be updated or a unique one will be assigned according to first letter of name entered. - Update profile information, including the profile picture (either uploaded or automatically generated from the user's name). - Opt-in or opt-out of receiving notifications from event organizers and administrators. - When in admin mode, user will be able to browse events, profiles, and images with the power to remove them. User will also be able to remove facilities that violate app policy	**Event** **NotificationService** **StorageImage** **Facility**