**Object-Oriented Analysis: CRC Cards for Event Lottery System**

**Overview**

Based on the Event Lottery System Application requirements, I've identified the key entity, control, and boundary objects through analyzing nouns and verbs from the problem description. Below are CRC cards for the most important anticipated classes.

**1. Event**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Maintain event details (name, description, date/time, location) | • WaitingList |
| • Store registration period (start/end dates) | • Organizer |
| • Track participant capacity | • QRCode |
| • Hold event poster image | • EventPoster |
| • Manage geolocation requirement setting | • Facility |
| • Provide event information for QR scanning |  |

**Back:** Core entity representing events that entrants can join. Contains all event metadata and links to its waiting list and organizer.

**2. User (Abstract)**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Store device ID for identification | • Profile |
| • Maintain user role (Entrant/Organizer/Admin) |  |
| • Provide authentication |  |

**Back:** Abstract base class for all system users. Device-based identification eliminates need for username/password.

**3. Entrant (extends User)**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Join and leave waiting lists | • WaitingList |
| • View available events | • Event |
| • Accept or decline invitations | • Notification |
| • Maintain registration history | • Profile |
| • Manage notification preferences | • LotterySystem |
| • Scan QR codes to view events | • QRCode |

**Back:** Represents users who sign up for events. Tracks their participation across multiple events and manages their responses to lottery selections.

**4. Organizer (extends User)**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Create and manage events | • Event |
| • Generate QR codes for events | • QRCodeGenerator |
| • Set registration periods and capacity limits | • WaitingList |
| • Trigger lottery draws | • LotterySystem |
| • View entrant lists and geolocation data | • Entrant |
| • Send notifications to entrants | • NotificationManager |
| • Upload and update event posters | • EventPoster |
| • Export final attendee lists (CSV) |  |

**Back:** Entity that creates and runs events. Has full control over event lifecycle including lottery execution and communication with entrants.

**5. Administrator (extends User)**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Remove events | • Event |
| • Remove user profiles | • Profile |
| • Remove inappropriate images | • EventPoster |
| • Browse all events, profiles, and images | • Organizer |
| • Remove policy-violating organizers | • NotificationManager |
| • Review notification logs |  |

**Back:** System administrator with elevated privileges to moderate content and manage system integrity.

**6. WaitingList**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Maintain list of interested entrants | • Event |
| • Track entrant status (waiting/selected/declined/cancelled/enrolled) | • Entrant |
| • Enforce capacity limits (optional) | • LotterySystem |
| • Provide total entrant count |  |
| • Record geolocation data when entrants join |  |
| • Support viewing on map |  |

**Back:** Manages the pool of entrants for a specific event. Tracks all state transitions from joining to final enrollment or cancellation.

**7. LotterySystem**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Randomly sample entrants from waiting list | • WaitingList |
| • Select specified number of participants | • Event |
| • Draw replacement applicants when invitations declined | • NotificationManager |
| • Maintain lists of selected/cancelled/enrolled entrants | • Entrant |
| • Explain selection criteria/process |  |
| • Support multiple draw rounds |  |

**Back:** Control object that implements the lottery mechanism. Ensures fair random selection and handles the full lifecycle of draws and replacements.

**8. Profile**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Store personal information (name, email, phone) | • Entrant |
| • Maintain event registration history | • Event |
| • Track notification preferences | • NotificationManager |
| • Support profile updates |  |
| • Support profile deletion |  |
| • Store device identification |  |

**Back:** Holds all personal information for an entrant. Supports full CRUD operations and maintains historical data.

**9. QRCode**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Store unique code for event | • Event |
| • Link to event details page | • QRCodeGenerator |
| • Enable entrants to join waiting list via scan | • Entrant |
| • Display event information when scanned |  |

**Back:** Promotional QR code that serves as entry point for entrants to discover and join event waiting lists.

**10. NotificationManager**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Send lottery win notifications | • Entrant |
| • Send lottery lose notifications | • Organizer |
| • Send bulk notifications to waiting list/selected/cancelled groups | • LotterySystem |
| • Respect user notification opt-out preferences | • Administrator |
| • Log all notifications sent |  |
| • Support notification review by admins |  |

**Back:** Control object managing all system communications. Ensures proper notification delivery while respecting user preferences and maintaining audit logs.

**11. EventPoster**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Store event promotional image | • Event |
| • Support image upload | • Organizer |
| • Support image update | • Administrator |
| • Provide image for display in event details |  |

**Back:** Manages event poster images. Supports upload, update, and removal operations.

**12. QRCodeGenerator**

| **Responsibilities** | **Collaborators** |
| --- | --- |
| • Generate unique QR codes for events | • Event |
| • Create scannable promotional codes | • QRCode |
| • Link QR code to event details | • Organizer |

**Back:** Control object responsible for creating unique QR codes when organizers publish events.

**Design Principles Applied**

**Entity Objects** (Problem Domain Model):

* Event, Entrant, Organizer, Administrator, Profile, WaitingList, QRCode, EventPoster

**Control Objects** (Coordinate Services):

* LotterySystem, NotificationManager, QRCodeGenerator

**Boundary Objects** (System Interface):

* Not detailed in CRC cards but would include UI components for scanning, event browsing, and profile management

**Key Design Decisions:**

1. **User hierarchy**: Abstract User class with Entrant, Organizer, and Administrator as specializations
2. **Separation of concerns**: LotterySystem handles selection logic, NotificationManager handles communication
3. **Device-based auth**: User identified by device ID (US 01.07.01)
4. **WaitingList as separate entity**: Manages all entrant states and capacity constraints
5. **High cohesion**: Each class has clear, focused responsibilities
6. **Low coupling**: Classes interact through well-defined interfaces