

Event Lottery System

CRC Cards – Full Release (Halfway + Final)

Covering all User Stories across Halfway Checkpoint and Final Release

Entity Classes

User «Entity» <i>User Stories: US 01.02.01, 01.02.02, 01.07.01</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Store user profile: name, email, optional phone number• Hold unique device ID for passwordless authentication• Track user role: Entrant / Organizer / Admin• Store notification preference flag• Provide profile data for display and update• Support profile deletion (cascade to registrations)	<ul style="list-style-type: none">• Entrant• Organizer• DeviceManager• FirebaseManager• EntrantRegistration• NotificationService
Entrant «Entity (Role)» <i>User Stories: US 01.01.01, 01.01.02, 01.05.02, 01.05.03, 01.06.01, 01.06.02, 01.04.01, 01.04.02 [Halfway] US 01.04.03, 01.02.04, 01.01.04, 01.05.05, 01.02.03 [Final]</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Extend User with entrant-specific behaviour• Join the waiting list for a specific event• Leave the waiting list for a specific event• Accept an invitation when selected by the lottery• Decline an invitation when selected by the lottery• Scan a QR code to view event details and join waiting list• Receive lottery win/loss notifications• View number of total entrants on a waiting list• Maintain list of events currently registered for• [Final] Opt out of receiving notifications from organizers and admins• [Final] Delete own profile and cascade remove registrations• [Final] Filter browsable events by interest and availability• [Final] View history of all registered events and selection outcomes• [Final] View lottery selection criteria and guidelines	<ul style="list-style-type: none">• User• WaitingList• EntrantRegistration• EventDetailActivity• QRCodeManager• NotificationService• FirebaseManager• EventHistoryActivity
Organizer «Entity (Role)» <i>User Stories: US 02.01.01, 02.01.04, 02.02.01, 02.04.01, 02.04.02, 02.05.01-03, 02.06.01, 02.06.03, 02.06.04, 02.07.01, 02.07.02 [Halfway] US 02.02.02, 02.02.03, 02.03.01, 02.06.02, 02.06.05, 02.07.03 [Final]</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Extend User with organizer-specific behaviour• Create a new event with details, registration period, and optional capacity• Upload and update a promotional poster image for an event• Trigger the lottery draw after registration closes• Specify number of entrants to sample in a draw• Trigger replacement draw when a selected entrant declines	<ul style="list-style-type: none">• User• Event• WaitingList• LotteryService• NotificationService• ImageManager• QRCodeManager• GeolocationManager

<ul style="list-style-type: none"> • View waiting list, selected list, and final enrolled list for owned events • Cancel individual entrants from an event • Send broadcast notifications to waiting-list or selected entrants • Manage multiple owned events from the organiser dashboard • [Final] Enable or disable geolocation requirement for an event • [Final] View map of where entrants joined from (geolocation) • [Final] Optionally limit the number of entrants on the waiting list • [Final] View list of cancelled entrants • [Final] Export final enrolled list as CSV • [Final] Send notifications to cancelled entrants 	<ul style="list-style-type: none"> • OrganizerDashboardActivity • FirebaseManager
---	---

Admin «Entity (Role)» <i>User Stories: US 03.01.01, 03.02.01, 03.03.01, 03.04.01, 03.05.01, 03.06.01, 03.07.01, 03.08.01 [Final]</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Extend User with administrator-specific privileges • Browse all events across the system • Remove any event that violates app policy • Browse all user profiles across the system • Remove any user profile that violates app policy • Remove organizers that violate app policy • Browse all uploaded images in the system • Remove any image that violates app policy • Review logs of all notifications sent by organizers to entrants 	<ul style="list-style-type: none"> • User • Event • FirebaseManager • ImageManager • NotificationService • AdminDashboardActivity

Event «Entity» <i>User Stories: US 02.01.01, 02.01.04, 02.04.01, 02.04.02, 01.01.03</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Store event details: name, description, date/time, location • Hold registration open/close timestamps • Maintain optional capacity limit for waiting list • Store reference to promotional poster image • Hold unique QR code payload (deep-link URL) • Track organizer (owner) reference • Expose event state: Draft / Open / Closed / Completed 	<ul style="list-style-type: none"> • FirebaseManager • QRCodeManager • ImageManager • WaitingList • LotteryService • User (Organizer)

WaitingList «Entity» <i>User Stories: US 01.01.01, 01.01.02, 01.05.04, 02.02.01, 02.06.01, 02.06.03</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Maintain ordered collection of EntrantRegistration records for an event • Enforce optional capacity limit on join • Prevent duplicate entries for the same user • Report total count of waiting entrants • Separate registrations by status: waiting / selected / accepted / declined / cancelled • Provide pool of eligible entrants for lottery draws • Return sub-lists for organiser views (waiting, selected, enrolled, cancelled) 	<ul style="list-style-type: none"> • Event • EntrantRegistration • LotteryService • FirebaseManager

EntrantRegistration «Entity» <i>User Stories: US 01.01.01, 01.01.02, 01.05.02, 01.05.03, 02.06.04</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> Record the association between a User and an Event on the waiting list Track registration status: WAITING / SELECTED / ACCEPTED / DECLINED / CANCELLED Store join timestamp and optional geolocation at join time Support status transitions driven by lottery and user responses Notify changes to WaitingList and LotteryService on status update 	<ul style="list-style-type: none"> User Event WaitingList NotificationService FirebaseManager

Notification «Entity» <i>User Stories: US 01.04.01, 01.04.02, 02.07.01, 02.07.02</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> Store notification payload: title, message body, event reference, recipient list Record send timestamp and delivery status Track notification type: SELECTED / NOT_SELECTED / ORGANIZER_MESSAGE Support opt-out check before delivery 	<ul style="list-style-type: none"> NotificationService User FirebaseManager

NotificationLog «Entity» <i>User Stories: US 03.08.01 [Final]</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> Record each notification send event: sender, recipient list, message, timestamp Store notification type and associated event reference Provide queryable log for admin review Support filtering by organizer, event, or date range 	<ul style="list-style-type: none"> Notification NotificationService FirebaseManager Admin

Controller / Service Classes

LotteryService «Controller» <i>User Stories: US 02.05.01, 02.05.02, 02.05.03, 01.05.01</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> Perform random sampling to select N entrants from the WaitingList Guarantee fair, unbiased random selection algorithm Trigger organizer-initiated draw after registration closes Handle replacement draws when selected entrant declines Update EntrantRegistration statuses to SELECTED after draw Invoke NotificationService to alert winners and non-winners Persist draw results via FirebaseManager 	<ul style="list-style-type: none"> WaitingList EntrantRegistration NotificationService FirebaseManager Event

NotificationService «Controller»

User Stories: US 01.04.01, 01.04.02, 02.07.01, 02.07.02

Responsibilities	Collaborators
<ul style="list-style-type: none">Send push notifications to selected / non-selected entrants after lotteryBroadcast organizer messages to all waiting-list or selected entrantsRespect per-user notification opt-out preference before sendingUse Firebase Cloud Messaging (FCM) to deliver messagesLog each notification send event for audit trailQueue and retry failed deliveries	<ul style="list-style-type: none">UserNotificationFirebaseManagerLotteryService

QRCodeManager «Controller»

User Stories: US 02.01.01, 01.06.01, 01.06.02

Responsibilities	Collaborators
<ul style="list-style-type: none">Generate a unique deep-link QR code payload for each new eventEncode event ID into QR code for device scanningInvoke device camera via ZXing / ML Kit for QR scanningParse scanned QR payload to extract event IDNavigate to Event Detail screen on successful scanReturn error state if QR code is invalid or event not found	<ul style="list-style-type: none">EventFirebaseManagerDeviceManagerEventDetailActivity

ImageManager «Controller»

User Stories: US 02.04.01, 02.04.02

Responsibilities	Collaborators
<ul style="list-style-type: none">Launch device image picker / camera intent for poster selectionCompress and validate image before uploadUpload event poster to Firebase Storage and return download URLReplace existing poster image on updateProvide placeholder image when no poster is availableSupport image deletion (admin moderation hook)	<ul style="list-style-type: none">EventFirebaseManagerDeviceManager

GeolocationManager «Controller»

User Stories: US 02.02.02, 02.02.03 [Final]

Responsibilities	Collaborators
<ul style="list-style-type: none">Request and obtain device GPS / network location on waiting list joinStore location coordinates in EntrantRegistration when geolocation is enabledProvide location data to OrganizerDashboardActivity for map displayRespect organizer toggle: only capture location when geolocation is enabled for eventHandle location permission requests gracefully with fallback	<ul style="list-style-type: none">EntrantRegistrationDeviceManagerOrganizerDashboardActivityFirebaseManager

Infrastructure / Boundary Classes

FirestoreManager «Boundary / Persistence» <i>User Stories: All user stories (cross-cutting)</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Serve as single access point for all Firestore read/write operations• Manage Firebase Authentication (anonymous / device-ID-based sign-in)• Interface with Firebase Cloud Messaging for push notifications• Interface with Firebase Storage for image upload/download• Provide real-time listeners for waiting list and event updates• Handle offline persistence and synchronisation• Abstract Firestore collection paths: users, events, waitingLists, notifications• Enforce Firestore security rules (role-based access)	<ul style="list-style-type: none">• User• Event• WaitingList• EntrantRegistration• Notification• NotificationService• ImageManager• LotteryService

DeviceManager «Boundary» <i>User Stories: US 01.07.01, 02.04.01, 01.06.01</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Retrieve or generate a stable unique device identifier (Android ID / UUID)• Provide device ID to FirestoreManager for passwordless authentication• Manage camera permission request and camera intent launching (QR scan)• Manage storage/media permission request for image picker• Expose device geolocation (GPS/network) for optional join-location capture• Cache device ID securely in SharedPreferences	<ul style="list-style-type: none">• FirestoreManager• QRCodeManager• ImageManager• User

UI Activity Classes (Boundary)

MainActivity «Boundary (UI)» <i>User Stories: US 01.07.01, 01.01.03</i>	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Entry point: detect device ID and authenticate with FirestoreManager• Determine user role and route to correct home screen• Display event browse list for entrants• Provide navigation to organiser dashboard or admin panel• Handle deep-link intents from QR code scans	<ul style="list-style-type: none">• DeviceManager• FirestoreManager• User• EventBrowseActivity• OrganizerDashboardActivity

ProfileActivity «Boundary (UI)»

User Stories: US 01.02.01, 01.02.02

Responsibilities	Collaborators
<ul style="list-style-type: none">• Display and allow editing of user name, email, phone• Validate form inputs before saving• Call FirebaseManager to persist profile updates• Display profile picture (auto-generated avatar or uploaded photo)• Provide option to delete account	<ul style="list-style-type: none">• User• FirebaseManager• ImageManager

EventBrowseActivity «Boundary (UI)»

User Stories: US 01.01.03, 01.05.04, 01.06.01

Responsibilities	Collaborators
<ul style="list-style-type: none">• Display scrollable list of all available open events fetched from Firestore• Show key event summary per item: name, date, location, waiting list count• Provide real-time updates to event list via FirebaseManager listener• Navigate to EventDetailActivity on event item tap• Launch QR scanner via QRCodeManager from browse screen• Show empty state when no events are available• Display waiting list count per event (US 01.05.04)	<ul style="list-style-type: none">• Event• FirebaseManager• EventDetailActivity• QRCodeManager• MainActivity

EventDetailActivity «Boundary (UI)»

User Stories: US 01.06.01, 01.06.02, 01.01.01, 01.01.02, 01.05.02, 01.05.03, 01.05.04

Responsibilities	Collaborators
<ul style="list-style-type: none">• Display full event details: name, description, date, location, poster• Show current waiting list count and capacity• Provide Join / Leave waiting list action buttons• Present Accept / Decline invitation buttons when user is SELECTED• Navigate to QR scanner via QRCodeManager• Refresh real-time status via FirebaseManager listener	<ul style="list-style-type: none">• Event• WaitingList• EntrantRegistration• QRCodeManager• FirebaseManager• NotificationService

NotificationInboxActivity «Boundary (UI)»

User Stories: US 01.04.01, 01.04.02, 02.07.01, 02.07.02

Responsibilities	Collaborators
<ul style="list-style-type: none">• Display list of all notifications received by the current entrant• Show notification type clearly: selected (won), not selected (lost), organizer message• Mark notifications as read on open• Show timestamp and associated event name for each notification• Provide navigation to relevant EventDetailActivity from a notification item• Show empty state when no notifications exist• Display unread notification badge count in navigation	<ul style="list-style-type: none">• Notification• FirebaseManager• EventDetailActivity• NotificationService• User

CreateEventActivity «Boundary (UI)»

User Stories: US 02.01.01, 02.01.04, 02.04.01

Responsibilities	Collaborators
<ul style="list-style-type: none">• Provide form for organiser to enter event name, description, date/time/location• Allow setting of registration open/close dates• Allow optional waiting list capacity input• Trigger ImageManager for poster upload• Call FirebaseManager to save event and trigger QRCodeManager to generate QR• Display generated QR code for sharing	<ul style="list-style-type: none">• Event• FirebaseManager• QRCodeManager• ImageManager• User (Organizer)

OrganizerDashboardActivity «Boundary (UI)»

User Stories: US 02.02.01, 02.05.01, 02.06.01, 02.06.03, 02.06.04, 02.07.01, 02.07.02 [Halfway] | US 02.02.02, 02.02.03, 02.03.01, 02.06.02, 02.06.05, 02.07.03 [Final]

Responsibilities	Collaborators
<ul style="list-style-type: none">• List all events owned by the organiser• Navigate to event-specific waiting list, selected list, and enrolled list views• Provide 'Draw Lottery' button to invoke LotteryService• Allow organiser to input number of entrants to select in draw• Provide 'Send Notification' controls for waiting-list and selected entrants• Allow organiser to cancel individual entrants• Show event status summary: waiting / selected / enrolled counts• [Final] Toggle geolocation requirement on/off per event• [Final] Display Google Map showing locations where entrants joined from• [Final] Set optional waiting list capacity limit on event creation/edit• [Final] Display list of cancelled entrants• [Final] Export enrolled list as CSV file• [Final] Send notification to cancelled entrants	<ul style="list-style-type: none">• Event• WaitingList• LotteryService• NotificationService• GeolocationManager• FirebaseManager• User (Organizer)

EventHistoryActivity «Boundary (UI)»

User Stories: US 01.02.03 [Final]

Responsibilities	Collaborators
<ul style="list-style-type: none">• Display full history of events the entrant has registered for• Show outcome per event: waiting / selected / accepted / declined / cancelled / not selected• Group or sort history by date• Navigate to EventDetailActivity for past event details• Fetch history from FirebaseManager using entrant's registration records	<ul style="list-style-type: none">• EntrantRegistration• Event• FirebaseManager• EventDetailActivity• Entrant

AdminDashboardActivity «Boundary (UI)»

User Stories: US 03.01.01, 03.02.01, 03.03.01, 03.04.01, 03.05.01, 03.06.01, 03.07.01, 03.08.01 [Final]

Responsibilities	Collaborators
<ul style="list-style-type: none">• Provide tabbed or sectioned view: Events / Profiles / Images / Logs• Browse and search all events across the system• Allow admin to remove any event• Browse and search all user profiles• Allow admin to remove any user profile or organizer• Browse all uploaded images in Firebase Storage• Allow admin to remove any image• Display notification log with filter by organizer, event, or date• Confirm destructive actions (delete) with dialog before executing	<ul style="list-style-type: none">• Admin• Event• User• ImageManager• NotificationLog• FirebaseManager