PROJECT PART 3
CRC CARDS

MainActivity

Responsibilities

- Initialize and manage app cycle
- Load and manage user theme preferences using interactions
- Initialize and display Navigation Host
- Coordinate between theme changes/storage and UI

Collaborators

- ThemePreferences
- ThemeManager
- PLuckNavHost
- PluckTheme

PLuckApp

Responsibilities

- Entry point for global application startup
- Configure Cloudinary SDK for image uploads
- Provides a foundation for future application-wide initializations

Collaborators

- CloudinaryConfig

AdminAccessRepository

Responsibilities

- Verify if device is registered as admin
- Register new admin devices with password (validated)
- Remove Admin devices
- Store Admin metadata and convert Firestore documents to AdminConfig

Collaborators

- FirebaseFirestore
- AdminConfig

AppealRepository

Responsibilities

- Manages full lifecycle of organizer appeals
- Validate user ban status before appeal submission
- Retrieve all pending appeals for admin view
- Reject and Approve appeals (reinstate access)
- Observe real-time appeal updates

Collaborators

- FirebaseFirestore
- OrganizerAppeal
- AppealStatus
- UserRole

CloudinaryUploadRepository

Responsibilities

- Upload profile images and posters to Cloudinary
- Update Firestore with generated image URLs
- Organize uploads into folders (profiles / events)
- Track upload progress with error handling and cancellation
- Initialize Cloudinary MediaManager

Collaborators

- FirebaseFirestore
- MediaManager
- CloudinaryConfig

CsvExportRepository

Responsibilities

- Manages the creation and saving of CSV files
- Takes a list of WaitlistEntry objects and formats them into a CSV string

Collaborators

- Event
- WaitlistStatus
- MediaStore
- WaitlistEntry
- FileProvider

EventRepository

Responsibilities

- Manages the complete lifecycle of events in Firestore
- Creates new Events and updates in real-time
- Retieve Available Events (open Waitlist)
- Retrieves Events with applied filters
- Handle soft deletes and hard deletes
- Maintain and update Waitlist counts
- Convert between Event and FirebaseEvent models

Collaborators

- FirebaseFirestore
- FirebaseEvent
- Event

InvitationRepository

Responsibilities

- Run lottery draws from final waitlist
- Create event invitations for selected users
- Handles the logic for a user to accept an invitation
- Handles the logic for a user to decline an invitation
- Support wave-based lottery system
- Update waitlist and event statuses atomically

Collaborators

- FirebaseFirestore
- FirebaseEvent
- EventInvitation
- InvitationStatus
- WaitlistStatus

NotificationRepository

Responsibilities

- Send event invitations via email or phone
- Accept / Decline invitation notifications
- Integrate with waitlist on accept/decline
- Delete notifications for users (on account deletion)
- Delete notifications for events (on event deletion)
- Convert Firebase notifications to domain models

Collaborators

- WaitlistRepository
- EventRepository
- FirebaseFirestore
- EntrantProfile
- NotificationItem
- NotificationCategory
- WaitlistStatus
- InviteContactType
- FirebaseNotification

OrganizerAccessRepository

Responsibilities

- Manages the process of registering users as organizers
- Validate user registration (ban / duplicates)
- Manages the process of downgrading organizers to entrants
- Delete all events when downgrading organizer
- Update user roles in Firestore in real-time

Collaborators

- FirebaseFirestore
- UserRole

WaitlistRepository

Responsibilities

- Manages all user interactions with an event's waitlist and enrollment process.
- Accept / Decline Lottery invitations
- Run lottery draws and replacement for declines
- Send custom notifications to users (chosen + non-chosen)
- Update entrant display names across waitlists
- After account deletion, purge all waitlist entries for a user
- After event deletion, purge all waitlist entries for an event

Collaborators

- FirebaseFirestore
- FirebaseWaitlistEntry
- EventRepository
- WaitlistEntry
- WaitlistStatus
- Event

UserRepository

Responsibilities

- Retrieve all users from Firestore by role
- Retrieve a single user by ID
- Delete user profiles
- Remove organizer role and ban users
- Convert Firestore documents to FirebaseUser models

Collaborators

- FirebaseFirestore
- FirebaseUser
- UserRole

AdminViewModel

Responsibilities

- Manages all Admin dashboard states
- Observe organizers and appeals in real-time
- Delete events, profiles, and images with cleanup
- Revoke organizer privileges and close their events

Collaborators

- EventRepository
- UserRepository
- AppealRepository
- NotificationRepository
- FirebaseFirestore

EventViewModel

Responsibilities

- Manage all Event states
- Observe events and specific event in real-time
- Create, update, and delete events
- Run lottery draw and update draw status

Collaborators

- Event
- EventRepository
- WaitlistViewModel

NotificationsViewModel

Responsibilities

- Manage all Notification states
- Observe notifications in real-time for a user profile
- Track processing states to prevent duplicate actions
- Send event invitations via email/phone with feedback
- Handle invitation/notification logic

Collaborators

- NotificationRepository
- EntrantProfile
- NotificationItem
- AcceptNotificationResult

WaitlistViewModel

Responsibilities

- Manage all Waitlist states
- Observe waitlist and chosen entries in real-time
- Handle waitlist joins/leaves and run draws
- Export chosen entrants to csv
- Notify non-chosen entrants
- Check and refresh user waitlist membership

Collaborators

- WaitlistRepository
- CsvExportRepository
- Event
- WaitlistEntry
- WaitlistStatus
- UserEventHistory

PLuckNavHost

Responsibilities

- Acts as the central navigation controller for the entire application
- Manage navigation state and back stack via PLuckNavigator
- Manage all Viewmodel / local UI and session states
- Observe real-time profile changes & Waitlist updates
- Load and persist theme preferences (handle changes)
- Verify and provide admin dashboard access
- Manages all Account and Event operations
- Request location services for geolocation-required events
- Capture user location when joining events

Collaborators

- NavHostController
- PLuckNavigator
- PLuckDestination
- DeviceAuthenticator
- DeviceAuthPreferences
- ThemePreferences
- AdminAccessRepository
- OrganizerAccessRepository
- AppealRepository
- EventRepository
- EventViewModel
- WaitlistViewModel
- NotificationsViewModel
- AdminViewModel
- EntrantProfile
- Event
- NotificationItem
- WaitlistEntry
- WaitlistStatus
- UserRole
- OrganizerStats
- EventStatus

DeviceAuthenticator

Responsibilities

- Manage device-based authentication flow
- Register new users or sign in existing users
- Create and update user profiles in Firestore
- Handles data access/deletion across multiple repositories
- Resolve and provide device IDs

Collaborators

- FirebaseAuth
- FirebaseFirestore
- DeviceIdProvider
- EventRepository
- WaitlistRepository
- NotificationRepository
- AdminAccessRepository
- DeviceAuthPreferences
- EntrantProfile

DeviceAuthPreferences

Responsibilities

- Store and retrieve auto-login preferences
- Be consistent with authentication preferences across sessions
- Manage SharedPreferences for authentication settings

Collaborators

- Context

DeviceIdProvider

Responsibilities

- Retrieve/Generate stable unique device identifiers
- Ensure device ID consistency across app sessions

Collaborators

- Context

NotificationPreferences

Responsibilities

- Store and retrieve notification preference settings
- Control push notification preferences
- Control email notification preferences
- Control entrant/organizer notification preferences
- Determines if a notification should be sent based on user preferences.
- Persist notification settings across sessions

Collaborators

- NotificationType
- Context

CloudinaryConfig

Responsibilities

- Initialize Cloudinary MediaManager SDK
- Provide Cloudinary configuration
- Generate public IDs/URLs for uploaded images
- Define folder paths for profile images and event posters
- Handle singleton lifecycle management

Collaborators

- MediaManager
- Context

FirebaseEvent

Responsibilities

- Represent event data in Firestore format
- Store Event details
- Store User Information
- Store WaitList and Lottery configuration
- Store geolocation data
- Track creation and update/convert Timestamps

Collaborators

- Event
- Timestamp
- DrawStatus

EventInvitation

Responsibilities

- Represent event invitations in Firestore format
- Store invitation details
- Track invitation status and update timestamps
- Support wave-based lottery draws

Collaborators

- InvitationStatus
- Timestamp

FirebaseUser

Responsibilities

- Represent user profile data in Firestore format
- Store User role (Entrant, Organizer, Admin) and their contact info
- Store profile image URL
- Track account status and update timestamps

Collaborators

- Timestamp
- UserRole

FirebaseWaitlistEntry

Responsibilities

- Represent waitlist entry data in Firestore format
- Store entrant information
- Store entrant's waitlist status, queue position and geolocation data (optional)
- Track join, selection, and declined timestamps
- Convert between Firebase Timestamp and LocalDate
- Convert to UI WaitlistEntry model

Collaborators

- WaitlistEntry
- Timestamp
- WaitlistStatus

OrganizerAppeal

Responsibilities

- Represent organizer appeal data in Firestore format
- Store appeal details
- Track appeal status
- Store admin review information

Collaborators

- Timestamp
- AppealStatus

EntrantProfile

Responsibilities

- Represents the current user's identity within the UI layer
- Stores user-specific data and identifiers for app logic
- Stores ban/appeal flags with updated Timestamps

Collaborators

- Instant
- UserRole
- PluckNavHost

DrawStatus

Responsibilities

- Define draw states (Pending/Completed/Cancelled)
- Provide type-safe draw status categorization

Collaborators

Event

Responsibilities

- Represents a single event within the application's UI layer
- Store event Event/Organizer data
- Compute Lottery draws and update

Collaborators

- DrawStatus
- LocalDateTime
- DateTimeFormatter

NotificationItem

Responsibilities

- Represents a singular, user-facing notification in a list or feed.
- Consolidates all data needed to render a notification card
- Stores invitation (Accept/Reject) status

Collaborators

- FirebaseNotification
- NotificationCategory
- NotificationButtons
- NotificationStatus
- InviteContactType

NotificationCategory

Responsibilities

- Define selection category
- Define daadline category
- Define waitlist/replacement category
- Provide default accent color for each category

Collaborators

- NotificationItem

NotificationStatus

Responsibilities

- Define notification read/unread state
- Provide type-safe read status categorization

Collaborators

- NotificationItem

NotificationButtons

Responsibilities

- Provides a simple, immutable container of boolean flags
- Defines visibility of the CTA buttons set

Collaborators

- NotificationItem

InviteContactType

Responsibilities

- Specifies the method used for a direct invitation
- Provide type-safe contact method categorization

Collaborators

- NotificationItem

CustomTheme

Responsibilities

- Store theme metadata and predefined themes
- Store light mode color palette
- Store dark mode color palette
- Convert theme to PluckColors for light/dark mode

Collaborators

- PluckColors
- ContrastUtils
- Color

PluckTheme

Responsibilities

- Retrieve and convert CustomTheme to PluckColors
- Calculate contrast-safe "on" colors and ensureContrast
- Create material color scheme and apply theme

Collaborators

- ThemeManager
- CustomTheme
- PluckColors

PluckColors

Responsibilities

- Store color palette
- Store isDark flag for theme mode

Collaborators

- Color

ThemeManager

Responsibilities

- Store and Retrieve themes by ID
- Provide all predefined themes

Collaborators

- CustomTheme

ThemePreferences

Responsibilities

- Persist and retrieve dark mode setting
- Persist and retrieve selected theme ID
- Clear custom theme and reset to default

Collaborators

- CustomTheme

NotificationType

Responsibilities

- Provide type-safe notification categorization

Collaborators

- PLuckNavHost

UserRole

Responsibilities

- Defines User Status (Entrant / Organizer / Admin)

Collaborators

- PLuckNavHost

WaitlistStatus

Responsibilities

- Define waitlist state (Waiting/ Invited/ Selected/ Accepted/ Declined/ Cancelled)

Collaborators

- PLuckNavHost
- WaitlistRepository

AppealStatus

Responsibilities

- Define appeal states (Pending/ Approved/ Rejected)

Collaborators

InvitationStatus

Responsibilities

- Define invitation states (Pending, Accepted / Declined / Expired)

Collaborators

PLuckNavigator

Responsibilities

- Provide type-safe navigation methods wrapping NavHostController
- Navigate to specific screens with/without parameters

Collaborators

- NavHostController
- PLuckDestination
- PLuckNavHost

PLuckDestination

Responsibilities

- Define type-safe navigation routes for all app screens
- Provide route strings and parameterized route generation

Collaborators

EventStatus

Responsibilities

- Define event states (Upcoming/ Past/ Confirmed)

Collaborators

- PLuckNavHost