MainActivity

- TAG : String
- PREFS_NAME : String
- KEY_REMEMBER_ME : String
- NOTIFICATION_PERMISSION_CODE : int
- btnLogin : Button
- btnAcctCreate : Button
- mAuth : FirebaseAuth
- sharedPreferences : SharedPreferences

- onCreate(savedInstanceState : Bundle) : void
- onRequestPermissionsResult(requestCode : int, permissions : String[], grantResults : int[]) : void
- onDestroy() : void
- checkAutoLogin() : void
- setupMainActivityUI() : void
- startNotificationListener() : void
- requestNotificationPermission() : void
- createNotificationChannels() : void

LoginActivity

- etEmail : EditText
- etPassword : EditText
- btnLogin : Button
- btnBack : ImageButton
- cbRememberMe : CheckBox
- mAuth : FirebaseAuth
- sharedPreferences : SharedPreferences
- PREFS_NAME : String
- KEY_REMEMBER_ME : String

- onCreate(savedInstanceState : Bundle) : void
- initViews() : void
- setupClickListeners() : void
- loginUser() : void
- saveRememberMePreference(rememberMe : boolean) : void
- validateInputs(email : String, password : String) : boolean
- resetButton() : void
- getErrorMessage(firebaseError : String) : String

AppCompatActivity

CreateAcctActivity

- etFullName : EditText
- etDisplayName : EditText
- etEmail : EditText
- etPassword : EditText
- etConfirmPassword : EditText
- etMobileNumber : EditText
- btnCreateAccount : MaterialButton
- btnBack : ImageButton
- mAuth : FirebaseAuth
- db : FirebaseFirestore
- accountsRef : CollectionReference

- onCreate(savedInstanceState : Bundle) : void
- initViews() : void
- setupClickListeners() : void
- createAccount() : void
- saveUserToDatabase(userId : String, fullName : String, displayName : String, email : String, mobileNumber : String) : void
- resetButton() : void
- validateInputs(fullName : String, displayName : String, email : String, password : String, confirmPassword : String, mobileNumber : String) : boolean
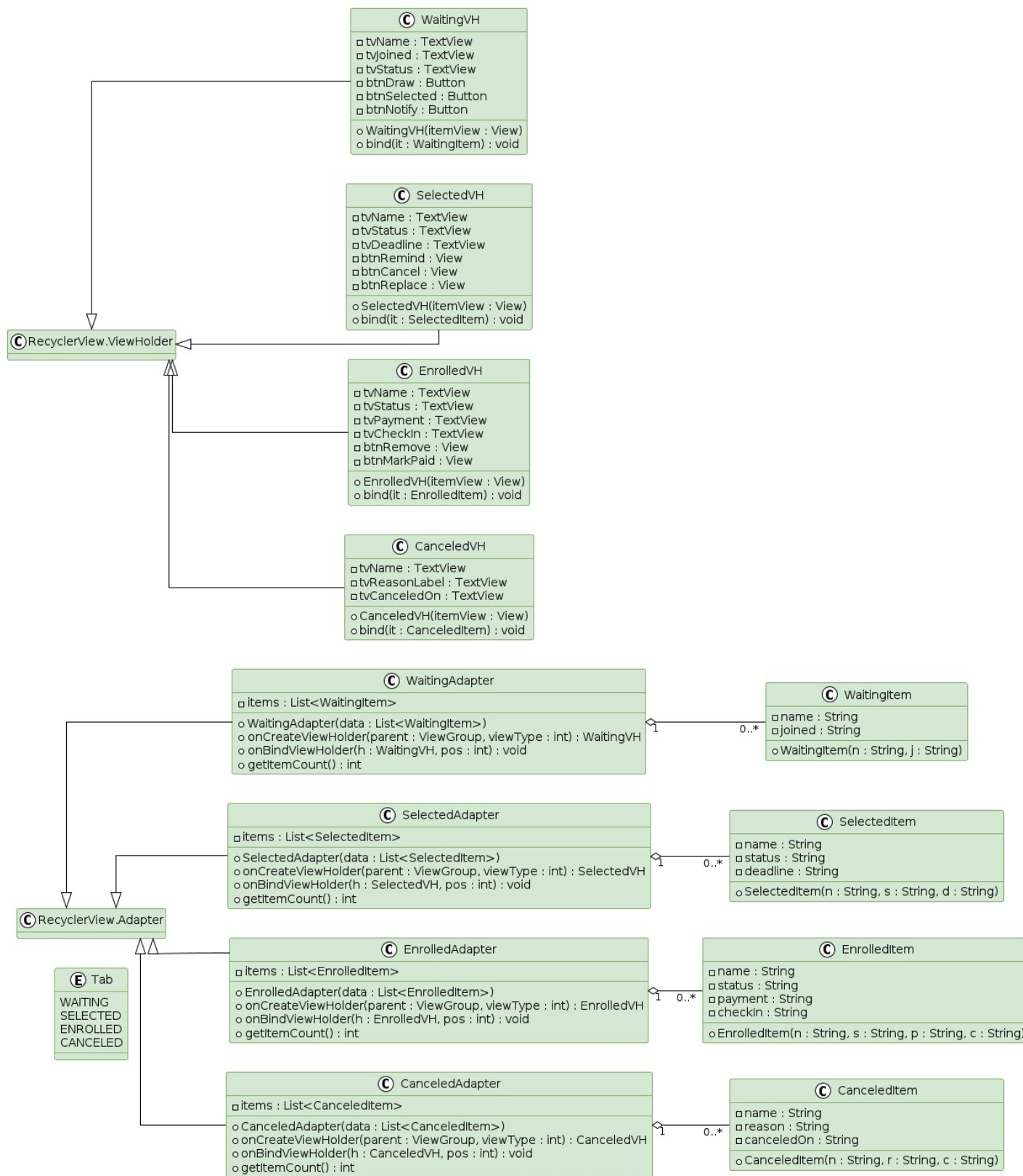
The light blue diagram covers the app's core login and account management system. It includes MainActivity, LoginActivity, and CreateAcctActivity, outlining how users register, sign in, and navigate through authentication screens.
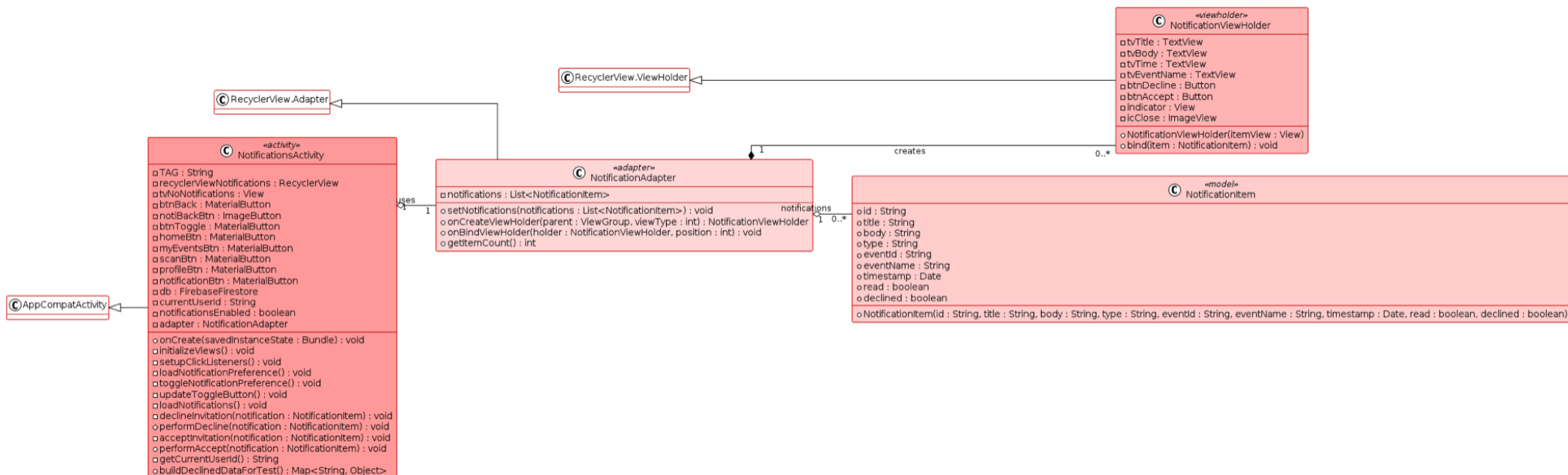
CreateEvent

- TAG : String
- etEventName : EditText
- etDescription : EditText
- etDrawCapacity : EditText
- etWaitingList : EditText
- etLocation : EditText
- etPrice : EditText
- btnBack : Button
- btnStartDate : Button
- btnStartTime : Button
- btnEndDate : Button
- btnEndTime : Button
- btnRegistrationOpens : Button
- btnRegistrationCloses : Button
- btnSaveDraft : Button
- btnPublishQR : Button
- ivEventImage : ImageView
- ivAddImage : ImageView
- switchGeolocation : SwitchMaterial
- startDateTime : Calendar
- endDateTime : Calendar
- registrationOpens : Calendar
- registrationCloses : Calendar
- dateFormat : SimpleDateFormat
- timeFormat : SimpleDateFormat
- dateTimeFormat : SimpleDateFormat
- selectedImageUri : Uri
- imagePickerLauncher : ActivityResultLauncher<Intent>
- db : FirebaseFirestore
- auth : FirebaseAuth
- currentUserId : String
- organizerName : String
- qrDialog : AlertDialog

- onCreate(savedInstanceState : Bundle) : void
- fetchOrganizerName() : void
- initializeViews() : void
- initializeDateTimeFormats() : void
- setupImagePicker() : void
- setupClickListeners() : void
- openImagePicker() : void
- showDatePicker(calendar : Calendar, button : Button, isStart : boolean) : void
- showTimePicker(calendar : Calendar, button : Button, isStart : boolean) : void
- showDateTimePicker(calendar : Calendar, button : Button, label : String) : void
- validateInputs() : boolean
- saveDraft() : void
- publishAndGenerateQR() : void
- createWaitingList(eventId : String) : void
- getWaitingListCapacity() : int
- showQRDialog(eventId : String, qrData : String) : void
- copyLinkToClipboard(qrData : String) : void
- shareQRCode(qrData : String) : void
- createEventData(isDraft : boolean) : Map<String, Object>
- onDestroy() : void

AppCompatActivity

EventDetails

- loadingSpinner : ProgressBar
- scrollContent : ScrollView
- btnBack : ImageView
- eventImage : ImageView
- eventTitle : TextView
- tvOrgName : TextView
- tvDateTime : TextView
- tvLocation : TextView
- tvPrice : TextView
- tvRegWindow : TextView
- tvWaitingList : TextView
- description : String
- btnInfo : Button
- btnJoinList : Button
- btnLeaveList : Button
- db : FirebaseFirestore
- eventId : String

- onCreate(savedInstanceState : Bundle) : void
- initializeViews() : void
- setupClickListeners() : void
- loadEventDetails(eventId : String) : void
- displayEventDetails(eventName : String, organizationName : String, dateTime : String, location : String, price : String, registrationWindow : String, waitingListCount : String) : void
- formatDateTime(dateTimeObj : Object) : String
- formatRegistrationDate(dateString : String) : String
- joinWaitingList() : void
- leaveWaitingList() : void
- hideContent() : void
- showContent() : void

The light yellow diagram focuses on event management. It contains CreateEvent, and EventDetails, showing how organizers create, modify, and display event information in the app.
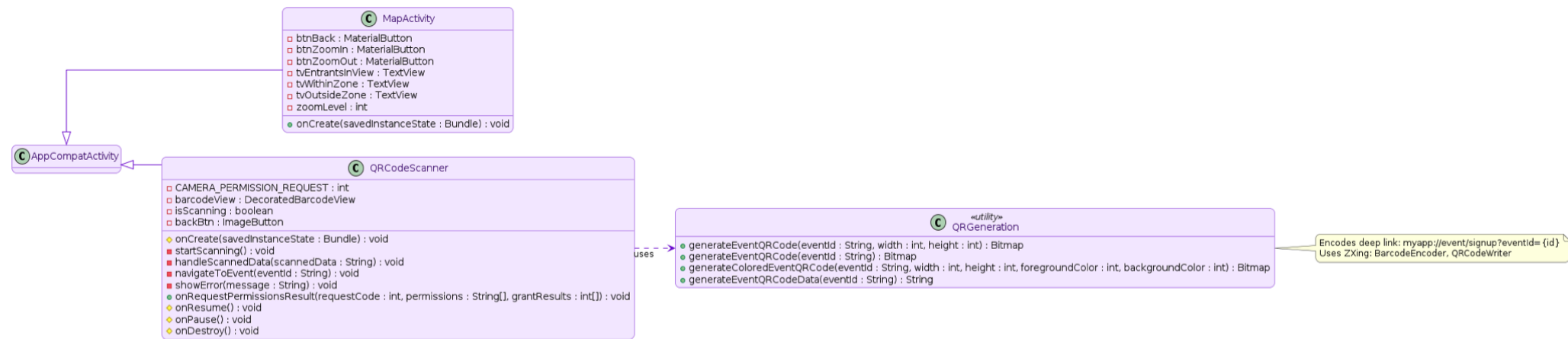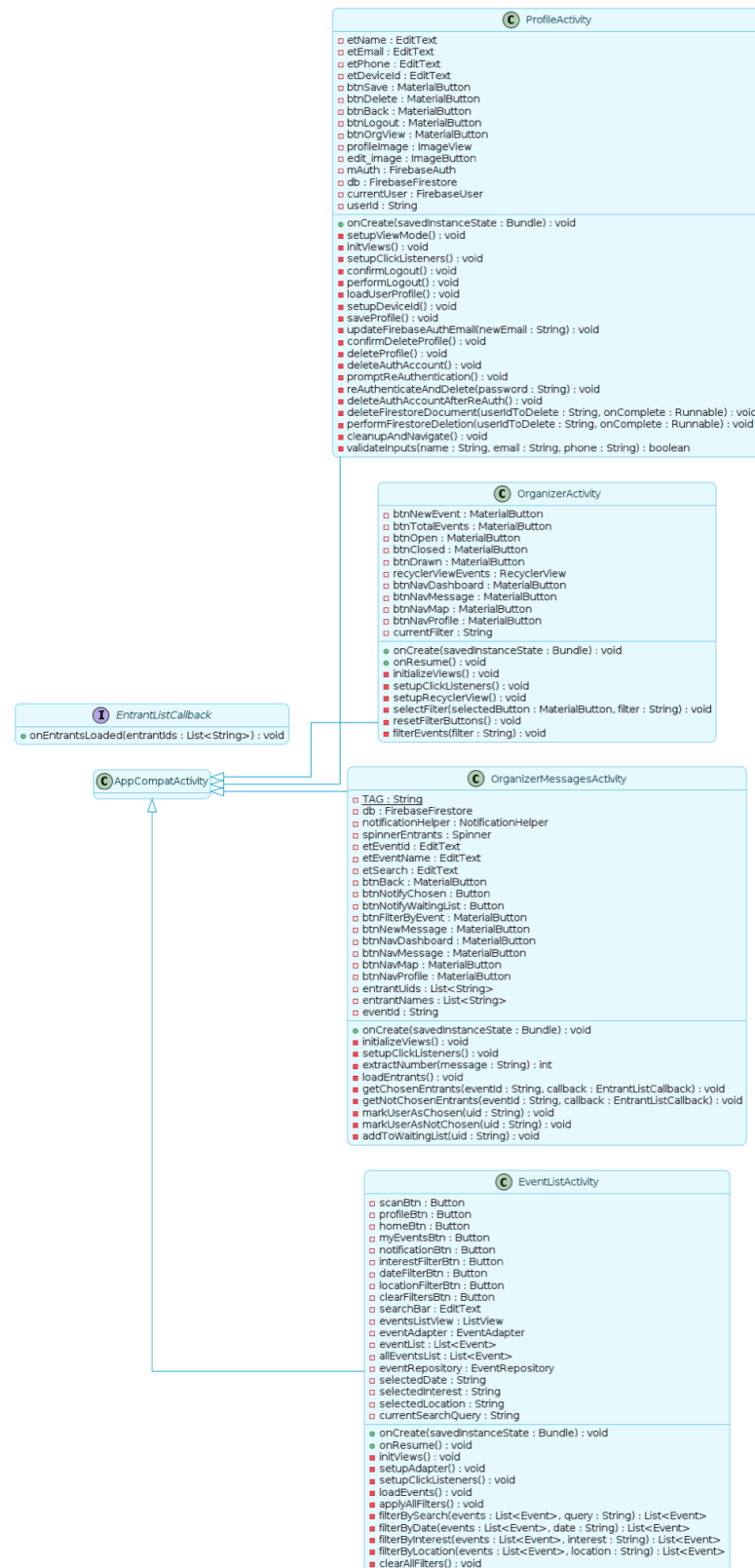
WaitingVH

- tvName : TextView
- tvJoined : TextView
- tvStatus : TextView
- btnDraw : Button
- btnSelected : Button
- btnNotify : Button

- WaitingVH(itemView : View)
- bind(it : WaitingItem) : void

SelectedVH

- tvName : TextView
- tvStatus : TextView
- tvDeadline : TextView
- btnRemind : View
- btnCancel : View
- btnReplace : View

- SelectedVH(itemView : View)
- bind(it : SelectedItem) : void

EnrolledVH

- tvName : TextView
- tvStatus : TextView
- tvPayment : TextView
- tvCheckIn : TextView
- btnRemove : View
- btnMarkPaid : View

- EnrolledVH(itemView : View)
- bind(it : EnrolledItem) : void

CanceledVH

- tvName : TextView
- tvReasonLabel : TextView
- tvCanceledOn : TextView

- CanceledVH(itemView : View)
- bind(it : CanceledItem) : void

RecyclerView.ViewHolder

The light green diagram represents the structure for participant selection and waiting lists. It includes adapters like WaitingAdapter and SelectedAdapter, and their item and view holder classes that handle waiting, selected, enrolled, and canceled participants.

WaitingAdapter

- items : List<WaitingItem>

- WaitingAdapter(data : List<WaitingItem>)
- onCreateViewHolder(parent : ViewGroup, viewType : int) : WaitingVH
- onBindViewHolder(h : WaitingVH, pos : int) : void
- getItemCount() : int

WaitingItem

- name : String
- joined : String

- WaitingItem(n : String, j : String)

1   0..*

SelectedAdapter

- items : List<SelectedItem>

- SelectedAdapter(data : List<SelectedItem>)
- onCreateViewHolder(parent : ViewGroup, viewType : int) : SelectedVH
- onBindViewHolder(h : SelectedVH, pos : int) : void
- getItemCount() : int

SelectedItem

- name : String
- status : String
- deadline : String

- SelectedItem(n : String, s : String, d : String)

1   0..*

RecyclerView.Adapter

Tab (E)

WAITING
SELECTED
ENROLLED
CANCELED

EnrolledAdapter

- items : List<EnrolledItem>

- EnrolledAdapter(data : List<EnrolledItem>)
- onCreateViewHolder(parent : ViewGroup, viewType : int) : EnrolledVH
- onBindViewHolder(h : EnrolledVH, pos : int) : void
- getItemCount() : int

EnrolledItem

- name : String
- status : String
- payment : String
- checkIn : String

- EnrolledItem(n : String, s : String, p : String, c : String)

0..*

CanceledAdapter

- items : List<CanceledItem>

- CanceledAdapter(data : List<CanceledItem>)
- onCreateViewHolder(parent : ViewGroup, viewType : int) : CanceledVH
- onBindViewHolder(h : CanceledVH, pos : int) : void
- getItemCount() : int

CanceledItem

- name : String
- reason : String
- canceledOn : String

- CanceledItem(n : String, r : String, c : String)

1   0..*

The light red diagram highlights the app's communication system. It shows NotificationsActivity, NotificationAdapter, and NotificationItem, illustrating how messages and notifications are displayed and managed between organizers and entrants.

«viewholder»
NotificationViewHolder

- tvTitle : TextView
- tvBody : TextView
- tvTime : TextView
- tvEventName : TextView
- btnDecline : Button
- btnAccept : Button
- indicator : View
- icClose : ImageView

- NotificationViewHolder(itemView : View)
- bind(item : NotificationItem) : void

RecyclerView.ViewHolder

RecyclerView.Adapter

«activity»
NotificationsActivity

- TAG : String
- recyclerViewNotifications : RecyclerView
- tvNoNotifications : View
- btnBack : MaterialButton
- notiBackBtn : ImageButton
- btnToggle : MaterialButton
- homeBtn : MaterialButton
- myEventsBtn : MaterialButton
- scanBtn : MaterialButton
- profileBtn : MaterialButton
- notificationBtn : MaterialButton
- db : FirebaseFirestore
- currentUserId : String
- notificationsEnabled : boolean
- adapter : NotificationAdapter

- onCreate(savedInstanceState : Bundle) : void
- initializeViews() : void
- setupClickListeners() : void
- loadNotificationPreference() : void
- toggleNotificationPreference() : void
- updateToggleButton() : void
- loadNotifications() : void
- declineInvitation(notification : NotificationItem) : void
- performDecline(notification : NotificationItem) : void
- acceptInvitation(notification : NotificationItem) : void
- performAccept(notification : NotificationItem) : void
- getCurrentUserId() : String
- buildDeclinedDataForTest() : Map<String, Object>

«adapter»
NotificationAdapter

- notifications : List<NotificationItem>

- setNotifications(notifications : List<NotificationItem>) : void
- onCreateViewHolder(parent : ViewGroup, viewType : int) : NotificationViewHolder
- onBindViewHolder(holder : NotificationViewHolder, position : int) : void
- getItemCount() : int

AppCompatActivity

uses

creates

notifications
1 0..*

1

0..*

«model»
NotificationItem

- id : String
- title : String
- body : String
- type : String
- eventId : String
- eventName : String
- timestamp : Date
- read : boolean
- declined : boolean

- NotificationItem(id : String, title : String, body : String, type : String, eventId : String, eventName : String, timestamp : Date, read : boolean, declined : boolean)
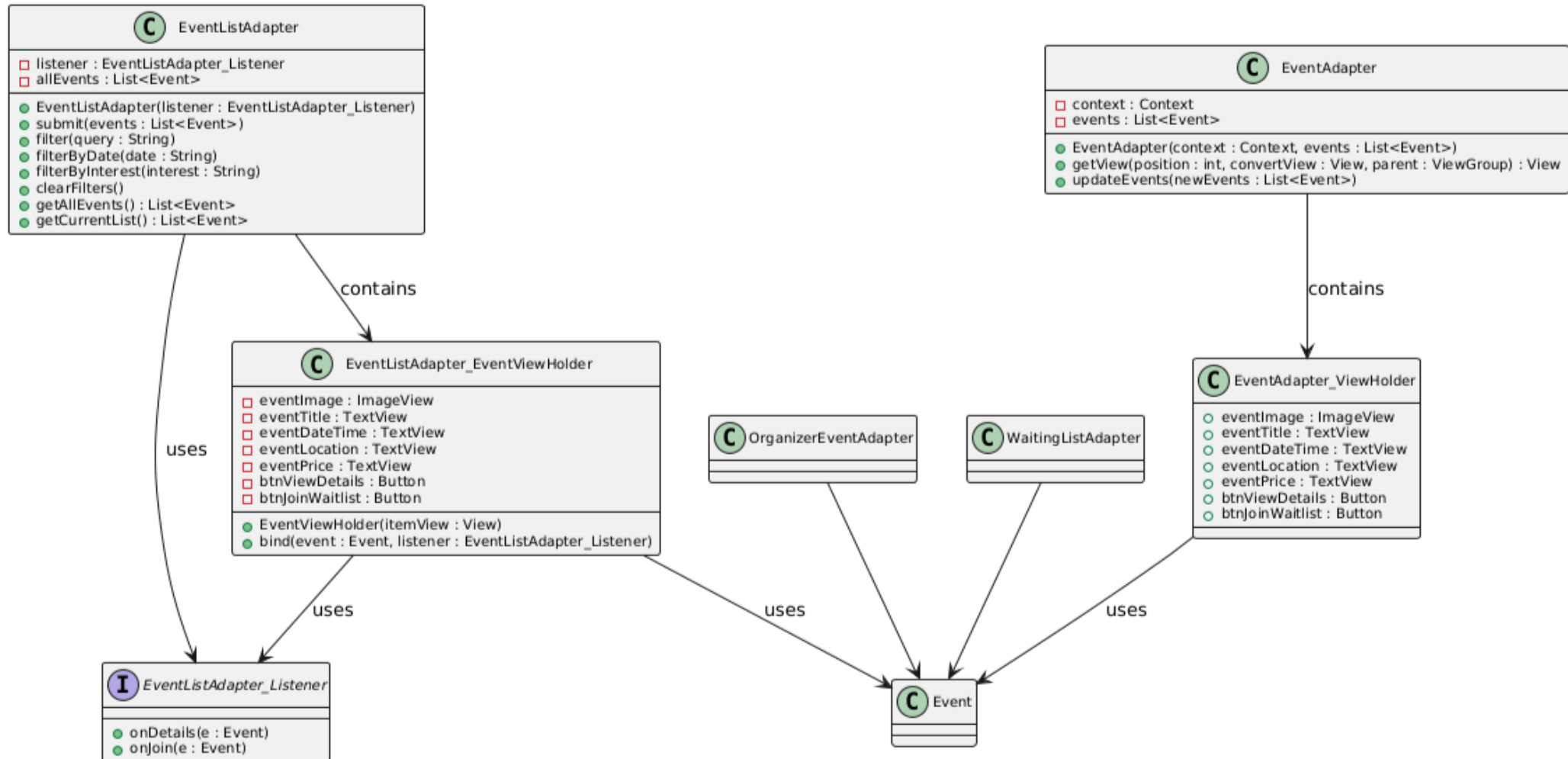
The lavender diagram visualizes the QR and map features. It includes QRCodeScanner, and QRGeneration, showing how the app generates, scans, and processes QR codes along with map-based location features.

MapActivity

- btnBack : MaterialButton
- btnZoomIn : MaterialButton
- btnZoomOut : MaterialButton
- tvEntrantsInView : TextView
- tvWithinZone : TextView
- tvOutsideZone : TextView
- zoomLevel : int

- onCreate(savedInstanceState : Bundle) : void

AppCompatActivity

QRCodeScanner

- CAMERA_PERMISSION_REQUEST : int
- barcodeView : DecoratedBarcodeView
- isScanning : boolean
- backBtn : ImageButton

- onCreate(savedInstanceState : Bundle) : void
- startScanning() : void
- handleScannedData(scannedData : String) : void
- navigateToEvent(eventId : String) : void
- showError(message : String) : void
- onRequestPermissionsResult(requestCode : int, permissions : String[], grantResults : int[]) : void
- onResume() : void
- onPause() : void
- onDestroy() : void

uses

«utility»
QRGeneration

- generateEventQRCode(eventId : String, width : int, height : int) : Bitmap
- generateEventQRCode(eventId : String) : Bitmap
- generateColoredEventQRCode(eventId : String, width : int, height : int, foregroundColor : int, backgroundColor : int) : Bitmap
- generateEventQRCodeData(eventId : String) : String

Encodes deep link: myapp://event/signup?eventId= {id}
Uses ZXing: BarcodeEncoder, QRCodeWriter

ProfileActivity

- etName : EditText
- etEmail : EditText
- etPhone : EditText
- etDeviceId : EditText
- btnSave : MaterialButton
- btnDelete : MaterialButton
- btnBack : MaterialButton
- btnLogout : MaterialButton
- btnOrgView : MaterialButton
- profileImage : ImageView
- edit_image : ImageButton
- mAuth : FirebaseAuth
- db : FirebaseFirestore
- currentUser : FirebaseUser
- userId : String

- onCreate(savedInstanceState : Bundle) : void
- setupViewMode() : void
- initViews() : void
- setupClickListeners() : void
- confirmLogout() : void
- performLogout() : void
- loadUserProfile() : void
- setupDeviceId() : void
- saveProfile() : void
- updateFirebaseAuthEmail(newEmail : String) : void
- confirmDeleteProfile() : void
- deleteProfile() : void
- deleteAuthAccount() : void
- promptReAuthentication() : void
- reAuthenticateAndDelete(password : String) : void
- deleteAuthAccountAfterReAuth() : void
- deleteFirestoreDocument(userIdToDelete : String, onComplete : Runnable) : void
- performFirestoreDeletion(userIdToDelete : String, onComplete : Runnable) : void
- cleanupAndNavigate() : void
- validateInputs(name : String, email : String, phone : String) : boolean

OrganizerActivity

- btnNewEvent : MaterialButton
- btnTotalEvents : MaterialButton
- btnOpen : MaterialButton
- btnClosed : MaterialButton
- btnDrawn : MaterialButton
- recyclerViewEvents : RecyclerView
- btnNavDashboard : MaterialButton
- btnNavMessage : MaterialButton
- btnNavMap : MaterialButton
- btnNavProfile : MaterialButton
- currentFilter : String

- onCreate(savedInstanceState : Bundle) : void
- onResume() : void
- initializeViews() : void
- setupClickListeners() : void
- setupRecyclerView() : void
- selectFilter(selectedButton : MaterialButton, filter : String) : void
- resetFilterButtons() : void
- filterEvents(filter : String) : void

EntrantListCallback *(interface)*

- onEntrantsLoaded(entrantIds : List<String>) : void

AppCompatActivity

OrganizerMessagesActivity

- TAG : String
- db : FirebaseFirestore
- notificationHelper : NotificationHelper
- spinnerEntrants : Spinner
- etEventId : EditText
- etEventName : EditText
- etSearch : EditText
- btnBack : MaterialButton
- btnNotifyChosen : Button
- btnNotifyWaitingList : Button
- btnFilterByEvent : MaterialButton
- btnNewMessage : MaterialButton
- btnNavDashboard : MaterialButton
- btnNavMessage : MaterialButton
- btnNavMap : MaterialButton
- btnNavProfile : MaterialButton
- entrantUids : List<String>
- entrantNames : List<String>
- eventId : String

- onCreate(savedInstanceState : Bundle) : void
- initializeViews() : void
- setupClickListeners() : void
- extractNumber(message : String) : int
- loadEntrants() : void
- getChosenEntrants(eventId : String, callback : EntrantListCallback) : void
- getNotChosenEntrants(eventId : String, callback : EntrantListCallback) : void
- markUserAsChosen(uid : String) : void
- markUserAsNotChosen(uid : String) : void
- addToWaitingList(uid : String) : void

EventListActivity

- scanBtn : Button
- profileBtn : Button
- homeBtn : Button
- myEventsBtn : Button
- notificationBtn : Button
- interestFilterBtn : Button
- dateFilterBtn : Button
- locationFilterBtn : Button
- clearFiltersBtn : Button
- searchBar : EditText
- eventsListView : ListView
- eventAdapter : EventAdapter
- eventList : List<Event>
- allEventsList : List<Event>
- eventRepository : EventRepository
- selectedDate : String
- selectedInterest : String
- selectedLocation : String
- currentSearchQuery : String

- onCreate(savedInstanceState : Bundle) : void
- onResume() : void
- initViews() : void
- setupAdapter() : void
- setupClickListeners() : void
- loadEvents() : void
- applyAllFilters() : void
- filterBySearch(events : List<Event>, query : String) : List<Event>
- filterByDate(events : List<Event>, date : String) : List<Event>
- filterByInterest(events : List<Event>, interest : String) : List<Event>
- filterByLocation(events : List<Event>, location : String) : List<Event>
- clearAllFilters() : void

The aqua blue diagram presents the profile and organizer interfaces. It includes ProfileActivity, OrganizerActivity, OrganizerMessagesActivity, and EventListActivity, illustrating how users and organizers manage profiles, messages, and event listings.
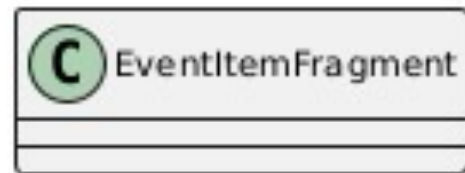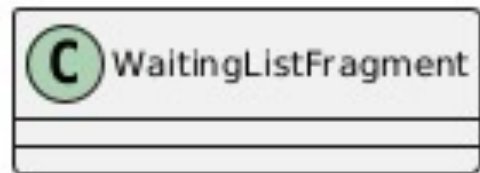
Adapters for Event App

EventListAdapter

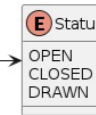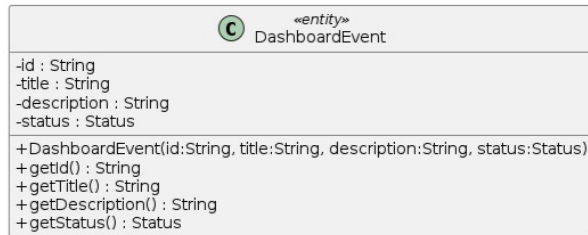- listener : EventListAdapter_Listener
- allEvents : List<Event>

- EventListAdapter(listener : EventListAdapter_Listener)
- submit(events : List<Event>)
- filter(query : String)
- filterByDate(date : String)
- filterByInterest(interest : String)
- clearFilters()
- getAllEvents() : List<Event>
- getCurrentList() : List<Event>

EventAdapter

- context : Context
- events : List<Event>

- EventAdapter(context : Context, events : List<Event>)
- getView(position : int, convertView : View, parent : ViewGroup) : View
- updateEvents(newEvents : List<Event>)

contains

EventListAdapter_EventViewHolder

- eventImage : ImageView
- eventTitle : TextView
- eventDateTime : TextView
- eventLocation : TextView
- eventPrice : TextView
- btnViewDetails : Button
- btnJoinWaitlist : Button

- EventViewHolder(itemView : View)
- bind(event : Event, listener : EventListAdapter_Listener)

uses

EventAdapter_ViewHolder

- eventImage : ImageView
- eventTitle : TextView
- eventDateTime : TextView
- eventLocation : TextView
- eventPrice : TextView
- btnViewDetails : Button
- btnJoinWaitlist : Button

contains

OrganizerEventAdapter

WaitingListAdapter

EventListAdapter_Listener

- onDetails(e : Event)
- onJoin(e : Event)

Event

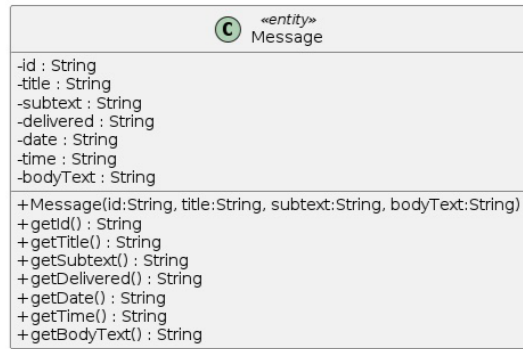*uses*

App Constants and Colors

AppColors

- ○ static int F1 = #B7E4EA «color»
- ○ static int F2 = #94B1B5 «color»
- ○ static int F3 = #507D82 «color»
- ○ static int F4 = #0B2E32 «color»

AppConstants

- ○ static int MAX_EVENT_PARTICIPANTS = 20
- ○ static int MIN_AGE_LIMIT = 5
- ○ static boolean FEATURE_ENABLE_LOTTERY = true

- ○ static String PREF_USER_ID = "pref_user_id"
- ○ static String PREF_PROFILE_COMPLETED = "pref_profile_completed"

- ○ static String COL_EVENTS = "events"

- ○ static String F_NAME = "name"
- ○ static String F_DATE = "date"
- ○ static String F_CAPACITY = "maxParticipants"
- ○ static String F_REG_OPENS = "regOpens"
- ○ static String F_REG_CLOSES = "regCloses"
- ○ static String F_DRAW_COUNT = "drawCount"
- ○ static String F_WAITING_LIST = "waitingList"

- ○ static String EXTRA_EVENT_ID = "EXTRA_EVENT_ID"

- ■ AppConstants() // private constructor

Event planner Fragments

C	WaitingListFragment

C	EventItemFragment

Models

«entity» Message

-id : String
-title : String
-subtext : String
-delivered : String
-date : String
-time : String
-bodyText : String

+Message(id:String, title:String, subtext:String, bodyText:String)
+getId() : String
+getTitle() : String
+getSubtext() : String
+getDelivered() : String
+getDate() : String
+getTime() : String
+getBodyText() : String

«entity» DashboardEvent

-id : String
-title : String
-description : String
-status : Status

+DashboardEvent(id:String, title:String, description:String, status:Status)
+getId() : String
+getTitle() : String
+getDescription() : String
+getStatus() : Status

«E» Status

OPEN
CLOSED
DRAWN

DashboardEvent —— uses ——> Status

«entity» Event

-eventId : String
-name : String
-description : String
-dateTime : String
-location : String
-price : String
-maxParticipants : int
-currentParticipants : int
-organizerId : String
-imageUrl : String
-category : String
-regStart : String
-regStop : String
-waitingListCount : Long

+Event()
+Event(name:String, dateTime:String, maxParticipants:int)
+getEventId() : String
+setEventId(eventId:String) : void
+getName() : String
+setName(name:String) : void
+getDescription() : String
+setDescription(description:String) : void
+getDateTime() : String
+setDateTime(dateTime:String) : void
+getLocation() : String
+setLocation(location:String) : void
+getPrice() : String
+setPrice(price:String) : void
+getMaxParticipants() : int
+setMaxParticipants(maxParticipants:int) : void
+getCurrentParticipants() : int
+setCurrentParticipants(currentParticipants:int) : void
+getOrganizerId() : String
+setOrganizerId(organizerId:String) : void
+getImageUrl() : String
+setImageUrl(imageUrl:String) : void
+getCategory() : String
+setCategory(category:String) : void
+getRegStart() : String
+setRegStart(regStart:String) : void
+getRegStop() : String
+setRegStop(regStop:String) : void
+getWaitingListCount() : Long
+setWaitingListCount(waitingListCount:Long) : void
+isFull() : boolean

«valueObject» WaitingListEntry

Event 1 —— waiting list entries —— 0..* WaitingListEntry

«entity» User

-userId : String
-name : String
-email : String
-phone : String
-profileImageUrl : String
-rememberMe : boolean

+User()
+User(userId:String, name:String, email:String, phone:String)
+User(userId:String, name:String, email:String, phone:String, profileImageUrl:String)
+getUserId() : String
+setUserId(userId:String) : void
+getName() : String
+setName(name:String) : void
+getEmail() : String
+setEmail(email:String) : void
+getPhone() : String
+setPhone(phone:String) : void
+getProfileImageUrl() : String
+setProfileImageUrl(profileImageUrl:String) : void
+isRememberMe() : boolean
+setRememberMe(rememberMe:boolean) : void
+toString() : String

Event organizer 1 —— 1 User

Network Layer Classes

EventRepository

- ☐ TAG : String
- ☐ COLLECTION_EVENTS : String
- ☐ db : FirebaseFirestore

- ● EventRepository()
- ● getAllEvents(callback : EventCallback) : void
- ● searchEventsByName(searchQuery : String, callback : EventCallback) : void
- ● getEventsByCategory(category : String, callback : EventCallback) : void
- ● getEventsByLocation(location : String, callback : EventCallback) : void
- ● getEventsByDateRange(startDate : String, endDate : String, callback : EventCallback) : void

EventCallback (Interface)

- ● onSuccess(events : List<Event>) : void
- ● onFailure(e : Exception) : void

uses →

NotificationListenerService

- ☐ TAG : String
- ☐ CHANNEL_ID : String
- ☐ SERVICE_CHANNEL_ID : String
- ☐ FOREGROUND_NOTIFICATION_ID : int
- ☐ notificationListener : ListenerRegistration
- ☐ db : FirebaseFirestore

- ● onCreate() : void
- ● onStartCommand(intent : Intent, flags : int, startId : int) : int
- ● startForegroundService() : void
- ● startListeningForNotifications() : void
- ● showNotification(title : String, body : String, type : String) : void
- ● createNotificationChannel() : void
- ● createServiceChannel() : void
- ● getCurrentUserId() : String
- ● onDestroy() : void
- ● onBind(intent : Intent) : IBinder

uses → FirebaseFirestore

uses → FirebaseFirestore

uses → FirebaseAuth

launches → MainActivity

DatabaseHelper

Utils

Validator
—
+validateDrawCount(raw:String, capacity:Integer) : Result

Result
—
+ok : boolean
+message : String
+value : int

returns

NotificationHelper
—
+notifyChosenEntrants(eventId:String, chosenEntrantIds:List<String>, eventName:String, cb: NotificationCallback) : void
+notifyNotChosenEntrants(eventId:String, notChosenEntrantIds:List<String>, eventName:String, cb: NotificationCallback) : void
+notifyAllWaitingListEntrants(eventId:String, eventName:String, cb: NotificationCallback) : void
·····················internal·····················
-sendNotificationsToUsers(...)
-markEntrantsAsNotified(...)

Holds FirebaseFirestore.
Reads DocumentSnapshot.
Uses FieldValue.serverTimestamp().

callback

NotificationCallback
—
+onSuccess(message:String) : void
+onFailure(error:String) : void

1   0..1

BottomNavHelper
—
+attach(a: Activity, root: View) : void
·····················internal·····················
-navigate(a: Activity, cls: Class<?>) : void

Uses Android Activity & View.
May navigate to OrganizerActivity,
MessagesActivity, MapActivity, ProfileActivity.

1   may use (optional)   0..1

DateUtils