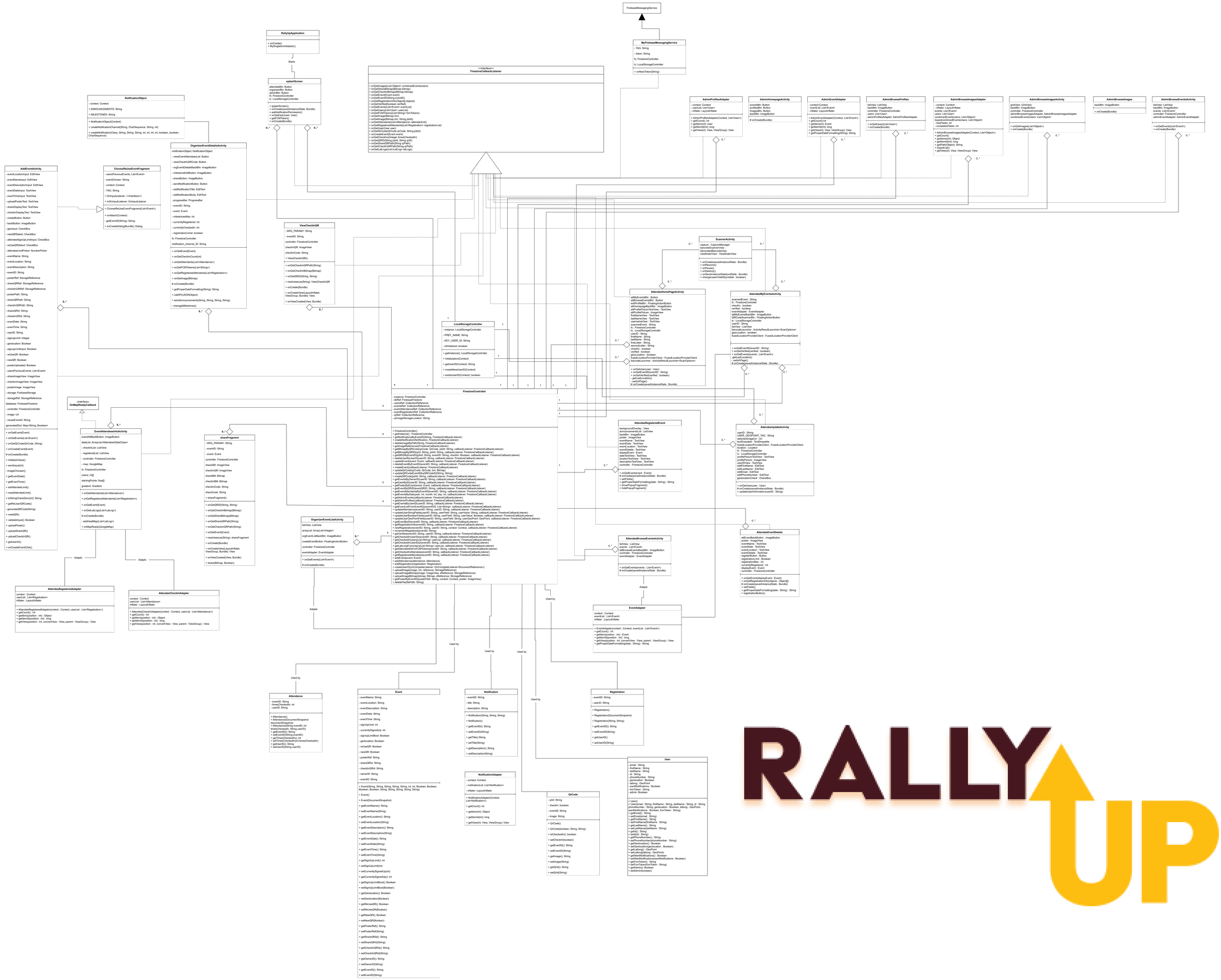
FirebaseMessagingService
MyFirebaseMessagingService
- TAG: String
- token: String
fc: FirestoreController
lc: LocalStorageController
+ onNewToken(String)
RallyUpApplication
+ onCreate()
+ MySingletonInitializer()
Starts
splashScreen
attendeeBtn: Button
organizerBtn: Button
adminBtn: Button
fc: FirestoreController
lc: LocalStorageController
+ splashScreen()
+ onCreate(savedInstanceState: Bundle)
+ askNotificationPermission()
+ onGetUser(user: User)
+ getFCMToken()
+ onCreate(Bundle)
<<Interface>>
FirestoreCallbackListener
+ onGetImages(List<Object> combinedEventsUsers)
+ onGetShareBitmap(Bitmap bitmap)
+ onGetCheckinBitmap(Bitmap bitmap)
+ onGetEvent(Event event)
+ onGetEventID(String eventID)
+ onGetRegistrationInfo(Object[] objects)
+ onGetEventsAndUsers(List eventsUsers)
+ onGetEventsUsersEvent(List<Event> eventList)
+ onGetUsers(List<User> userList)
+ onGetFCMTokens(List<String> fcmTokens)
+ onGetImage(Bitmap bm)
+ onGetImage(Bitmap bm, String jobId)
+ onGetAttendants(List<Attendance> attendantList)
+ onGetRegisteredAttendants(List<Registration> registrationList)
+ onGetUser(User user)
+ onGetEvent(Event event, String jobId)
+ onCreateEvent(Event event)
+ onGetCheckins(Integer timesCheckedIn)
+ onGetQRID(String jobId, String qrID)
+ onGetShareQRPath(String qrPath)
+ onGetCheckinQRPath(String qrPath)
+ onGetLatLngs(List<LatLng> latLngs)
NotificationObject
- context: Context
+ ANNOUNCEMENTS: String
+ MILESTONES: String
+ NotificationObject(Context)
+ createNotificationChannel(String, CharSequence, String, int)
+ createNotification(Class, String, String, String, int, int, int, int, boolean, boolean, CharSequence)
OrganizeEventDetailsActivity
notificationObject: NotificationObject
- viewEventAttendeesList: Button
- viewCheckinQRCode: Button
- orgEventDetailsBackBtn: ImageButton
- milestoneEditButton: ImageButton
- shareButton: ImageButton
- sendNotificationButton: Button
- editNotificationTitle: EditText
- editNotificationBody: EditText
- progressBar: ProgressBar
- eventID: String
- event: Event
- milestoneMax: int
- currentlyRegistered: int
- currentlyCheckedIn: int
- registrationLimit: boolean
fc: FirestoreController
notification_channel_ID: String
+ onGetEvent(Event)
+ onGetCheckinCount(int)
+ onGetAttendants(List<Attendance>)
+ onGetFCMTokens(List<String>)
+ onGetRegisteredAttendants(List<Registration>)
+ onGetImage(Bitmap)
# onCreate(Bundle)
+ getProperDateFormatting(String): String
+ callAPI(JSONObject)
+ sendAnnouncements(String, String, String, String)
- manageMilestones()
AdminProfilesAdapter
- context: Context
- userList: List<User>
- inflater: LayoutInflater
+ AdminProfilesAdapter(Context, List<User>)
+ getCount(): int
+ getItem(int): User
+ getItemId(int): long
+ getView(int, View, ViewGroup): View
AdminHomepageActivity
eventsBtn: Button
profilesBtn: Button
imagesBtn: Button
backBtn: ImageButton
onCreate(Bundle)
AdminEventAdapter
- context:Context
- eventList: List<Event>
- inflater: LayoutInflater
+ AdminEventAdapter(Context, List<Event>)
+ getCount():int
+ getItem(int): Event
+ getItemId(int): long
+ getView(int, View, ViewGroup): View
- getProperDateFormatting(String): String
AdminBrowseProfiles
listView: ListView
backBtn: ImageButton
controller: FirestoreController
users: List<User>
adminProfilesAdapter: AdminProfilesAdapter
+ onGetUsers(List<User>)
+ onCreate(Bundle)
AdminBrowseImagesAdapter
- context: Context
- inflater: LayoutInflater
- images: List<Object>
- users: List<User>
- combinedEventsUsers: List<Object>
- dupedcombinedEventsUsers: List<Object>
- totalTasks: int
- completedTasks: int
+ AdminBrowseImagesAdapter(Context, List<Object>)
+ getCount()
+ getItem(int): Object
+ getItemId(int): long
+ getItemID(Object): String
+ dupeList()
+ getView(int, View, ViewGroup): View
AdminBrowseImagesActivity
gridView: GridView
backBtn: ImageButton
controller: FirestoreController
adminBrowseImagesAdapter: AdminBrowseImagesAdapter
combinedEventsUsers: List<Object>
+ onGetImages(List<Object>)
+ onCreate(Bundle)
AdminBrowseImages
backBtn: ImageButton
+ onCreate(Bundle)
AdminBrowseEventsActivity
listView: ListView
backBtn: ImageButton
events: List<Event>
controller: FirestoreController
adminEventAdapter: AdminEventAdapter
+ onGetEventsLists(List<Event>)
+ onCreate(Bundle)
AddEventActivity
- eventLocationInput: EditView
- eventNameInput: EditView
- eventDescriptionInput: EditView
- eventDateInput: TextView
- eventTimeInput: TextView
- uploadPosterText: TextView
- shareDisplayText: TextView
- checkInDisplayText: TextView
- createButton: Button
- backButton: ImageButton
- geoInput: CheckBox
- newQRSelect: CheckBox
- attendeeSignUpLimitInput: CheckBox
- reUseQRSelect: CheckBox
- attendeeLimitPicker: NumberPicker
- eventName: String
- eventLocation: String
- eventDescription: String
- eventID: String
- posterRef: StorageReference
- shareQRRef: StorageReference
- checkInQRRef: StorageReference
- posterPath: String
- shareQRPath: String
- checkInQRPath: String
- shareQRId: String
- checkInQRId: String
- eventDate: String
- eventTime: String
- userID: String
- signupLimit: Integer
- geolocation: Boolean
- signupLimitInput: Boolean
- reUseQR: Boolean
- newQR: Boolean
- posterUploaded: Boolean
- usersPreviousEvents: List<Event>
- shareImageView: ImageView
- checkInImageView: ImageView
- posterImage: ImageView
- storage: FirebaseStorage
- storageRef: StorageReference
database: FirebaseFirestore
- controller: FirestoreController
- image: Uri
- reuseEventId: String
generatedDict: Map<String, Boolean>
+ onGetEvent(Event)
+ onGetEvents(List<Event>)
+ onGetQrCode(QCode, String)
+ onCreateEvent(Event)
# onCreate(Bundle)
+ initializeViews()
+ sendInput(int)
- imageChooser()
+ getEventDate()
+ getEventTime()
+ setAttendeeLimit()
+ resetAttendeeLimit()
+ toStringCheckZero(int): String
- getRaUserQRCode()
- generateQRCode(String)
+ resetQR()
+ validateInput(): Boolean
+ uploadPoster()
+ uploadShareQR()
+ uploadCheckInQR()
+ getUserID()
+ onCreateEventClick()
ChooseReUseEventFragment
- usersPreviousEvents: List<Event>
- eventChosen: String
- context: Context
- TAG: String
+ OnInputListener: <<Interface>>
+ onOnInputListener: OnInputListener
+ ChooseReUseEventFragment(List<Event>)
+ onAttach(Context)
- getEventID(String): String
+ onCreateDialog(Bundle): Dialog
ViewCheckInQR
- ARG_PARAM1: String
- eventID: String
controller: FirestoreController
checkInQR: ImageView
checkInCode: String
+ ViewCheckInQR()
+ onGetCheckinQRPath(String)
+ onGetCheckinBitmap(Bitmap)
+ onGetQRID(String, String)
+ newInstance(String): ViewCheckInQR
+ onCreate(Bundle)
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
+ onViewCreated(View, Bundle)
ScannerActivity
capture : CaptureManager
barcodeScannerView
DecoratedBarcodeView
viewfinderView : ViewfinderView
+ onCreate(savedInstanceState : Bundle)
+ onResume()
+ onPause()
+ onDestroy()
+ onSaveInstanceState(outState : Bundle)
+ changeLaserVisibility(visible : boolean)
LocalStorageController
- instance: LocalStorageController
- PREF_NAME: String
- KEY_USER_ID: String
- sInitialized: boolean
+ getInstance(): LocalStorageController
+ initialization(Context)
+ getUserID(Context): String
+ createNewUserID(Context)
+ existsUserID(Context): boolean
AttendeeHomePageActivity
attMyEventsBtn : Button
attBrowseEventsBtn : Button
addProfileBtn : FloatingActionButton
attHomepageBackBtn : ImageButton
attProfileUsernameTV : TextView
attProfilePicture : ImageView
firstNameView : TextView
lastNameView : TextView
usernameView : TextView
scannedEvent : String
fc : FirestoreController
ls : LocalStorageController
userID : String
firstName : String
lastName : String
firstLetter : String
secondLetter : String
checkIn : boolean
verified : boolean
geoLocation : boolean
fusedLocationProviderClient : FusedLocationProviderClient
barcodeLauncher : ActivityResultLauncher<ScanOptions>
+ onGetUserUser : User)
+ onGetEventID(eventID : String)
+ onGetVerified(verified : boolean)
- getLastLocation()
- switchPage()
onCreate(savedInstanceState : Bundle)
AttendeeMyEventsActivity
scannedEvent : String
fc : FirestoreController
checkIn : boolean
verified : boolean
eventAdapter : EventAdapter
attMyEventsBackBtn : ImageButton
QRCodeScannerBtn : FloatingActionButton
ls : LocalStorageController
userID : String
listView : ListView
barcodeLauncher : ActivityResultLauncher<ScanOptions>
geoLocation : boolean
fusedLocationProviderClient : FusedLocationProviderClient
+ onGetEventID(eventID : String)
+ onGetVerified(verified : boolean)
+ onGetEventsAndevents : List<Event>)
- getLastLocation()
- switchPage()
onCreate(savedInstanceState : Bundle)
<interface>
OnMapReadyCallback
EventAttendeesInfoActivity
eventAttBackButton: ImageButton
dataList: ArrayList<AttendeesStatsClass>
- checkInList: ListView
- registeredList: ListView
- controller: FirestoreController
- map: GoogleMap
fc: FirestoreController
colors: int[]
startingPoints: float[]
gradient: Gradient
+ onGetAttendants(List<Attendance>)
+ onGetRegisteredAttendants(List<Registration>)
+ onGetEvent(Event)
+ onGetLatLngs(List<LatLng>)
# onCreate(Bundle)
- addHeatMap(List<LatLng>)
- onMapReady(GoogleMap)
shareFragment
- ARG_PARAM1: String
- eventID: String
- event: Event
controller: FirestoreController
shareQR: ImageView
checkInQR: ImageView
shareBM: Bitmap
checkInBM: Bitmap
checkInCode: String
shareCode: String
+ shareFragment()
+ onGetQRID(String, String)
+ onGetCheckinBitmap(Bitmap)
+ onGetShareBitmap(Bitmap)
+ onGetShareQRPath(String)
+ onGetCheckinQRPath(String)
+ onGetEvent(Event)
+ newInstance(String): shareFragment
+ onCreate(Bundle)
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
+ onViewCreated(View, Bundle)
+ share(Bitmap, Boolean)
FirestoreController
- instance: FirestoreController
- dbRef: FirebaseFirestore
- usersRef: CollectionReference
- eventsRef: CollectionReference
- eventAttendanceRef: CollectionReference
- eventRegistrationRef: CollectionReference
- qrRef: CollectionReference
- qrImagesStorageLocation: String
+ FirestoreController()
+ getInstance(): FirestoreController
+ getNotifications(EventID):String, FirestoreCallbackListener)
+ createNotification(Notification, FirestoreCallbackListener)
+ deleteImageByPath(String, FirestoreCallbackListener)
+ getImageBitmap(String, FirestoreCallbackListener)
+ getBitmapByQRCode(qrCode: QrCode, jobId: String, callbackListener: FirestoreCallbackListener)
+ getBitmapByQRID(qrId: String, jobId: String, callbackListener: FirestoreCallbackListener)
+ getQRIDByEventID(eventID: String, eventID: String, checkIn: Boolean, callbackListener: FirestoreCallbackListener)
+ deleteUserByUserID(userID: String, callbackListener: FirestoreCallbackListener)
+ updateEvent(Event event, callbackListener: FirestoreCallbackListener)
+ deleteEventByEventID(eventID: String, callbackListener: FirestoreCallbackListener)
+ createEvent(Event event, callbackListener: FirestoreCallbackListener)
+ updateQRCode(qrCode: QrCode, callbackListener: FirestoreCallbackListener)
+ updateQRCodeEventIDbyQRCodeID(String, String)
+ updateQRCode(qrId: String, callbackListener: FirestoreCallbackListener)
+ getEventsByOwnerID(userID: String, callbackListener: FirestoreCallbackListener)
+ getUserByID(userID: String, callbackListener: FirestoreCallbackListener)
+ getPosterByEventEvent(event: Event, callbackListener: FirestoreCallbackListener)
+ getEventsByEventQRID(qrID: String, callbackListener: FirestoreCallbackListener)
+ getEventAttendanceByEventIDUserID(eventID: String, callbackListener: FirestoreCallbackListener)
+ getEventAttendanceByEventID(eventID: String, callbackListener: FirestoreCallbackListener)
+ getAdminEvents(callbackListener: FirestoreCallbackListener)
+ getAdminProfiles(callbackListener: FirestoreCallbackListener)
+ getEventsByUserID(userID: String, callbackListener: FirestoreCallbackListener)
+ getEventsListFromEventIDs(eventIDS: List<String>, callbackListener: FirestoreCallbackListener)
+ updateAttendance(eventID: String, userID: String, callbackListener: FirestoreCallbackListener)
+ getRegisteredAttendees(eventID: String, callbackListener: FirestoreCallbackListener)
+ addEvent(event: Event)
+ addAttendance(attendance: Attendance)
+ addRegistration(registration: Registration)
+ createUserDjangoCompleteListener: OnCompleteListener<DocumentReference>)
+ uploadImage(image: Uri, reference: StorageReference)
+ uploadImageBitmap(image: ImageView, sRefference: StorageReference)
+ uploadImageBitmap(bitmap: Bitmap, sRefference: StorageReference)
+ getImageByPath(posterPath: String, context: Context, poster: ImageView)
+ deleteFile(filePath: String)
+ getUserByID(userID: String, callbackListener: FirestoreCallbackListener)
+ getPosterBy(Event event: Event, callbackListener: FirestoreCallbackListener)
+ getEventsBy(ownerID: String, callbackListener: FirestoreCallbackListener)
+ getEventsBy(Date(year: int, month: int, day: int, callbackListener: FirestoreCallbackListener)
+ getEventsBy(Date(year: int, month: int, day: int, callbackListener: FirestoreCallbackListener)
+ getEventsBy(QrCode: QrCode, callbackListener: FirestoreCallbackListener)
+ updateQRCode(qrCode: QrCode, callbackListener: FirestoreCallbackListener)
+ updateQRCodeEventIDbyQRCodeID(String, String)
+ updateQRCode(qrId: String, callbackListener: FirestoreCallbackListener)
+ getEventsByOwnerID(userID: String, callbackListener: FirestoreCallbackListener)
+ getUserByID(userID: String, callbackListener: FirestoreCallbackListener)
+ getPosterBy(Event event: Event, callbackListener: FirestoreCallbackListener)
+ getEventsBy(QRID(qrID: String, callbackListener: FirestoreCallbackListener)
+ getEventAttendanceByEventIDUserID(eventID: String, callbackListener: FirestoreCallbackListener)
+ getEventAttendanceBy(EventID(eventID: String, callbackListener: FirestoreCallbackListener)
+ getAdminEvents(callbackListener: FirestoreCallbackListener)
+ getAdminProfiles(callbackListener: FirestoreCallbackListener)
+ getEventsBy(UserID(userID: String, callbackListener: FirestoreCallbackListener)
+ getEventsListFromEventIDs(eventIDS: List<String>, callbackListener: FirestoreCallbackListener)
+ updateAttendance(eventID: String, userID: String, callbackListener: FirestoreCallbackListener)
+ getRegisteredAttendees(eventID: String, callbackListener: FirestoreCallbackListener)
+ addEvent(event: Event)
+ addAttendance(attendance: Attendance)
+ addRegistration(registration: Registration)
+ createUserDjangoCompleteListener: OnCompleteListener<DocumentReference>)
+ uploadImage(image: Uri, reference: StorageReference)
+ uploadImageBitmap(image: ImageView, sRefference: StorageReference)
+ uploadImageBitmap(bitmap: Bitmap, sRefference: StorageReference)
+ getImageByPath(posterPath: String, context: Context, poster: ImageView)
+ deleteFile(filePath: String)
AttendeesRegisteredEvent
backgroundOverlay : View
announcementsList : ListView
backBtn : ImageButton
poster : ImageView
eventName : TextView
eventDate : TextView
eventLocation : TextView
eventDetails : TextView
displayEvent : Event
dateTextView : TextView
locationTextView : TextView
descriptionTextView : TextView
controller : FirestoreController
+ onGetEvent(event : Event)
# onCreate(savedInstanceState : Bundle)
+ setFields()
+ getProperDateFormatting(date : String) : String
+ showPopupFragment()
+ hidePopupFragment()
AttendeeUpdateActivity
userID : String
USER_GEOPOINT_TAG : String
selectedImagesUri : Uri
textDrawable : RectDrawable
fusedLocationProviderClient : FusedLocationProviderClient
location : Location
fc : FirestoreController
lc : LocalStorageController
profilePictureTextView : TextView
usernameTextView : TextView
userID:View : TextView
editFirstName : EditText
editLastName : EditText
editEmail : EditText
editPhoneNumber : EditText
geolocationCheck : CheckBox
+ onGetUserUser : User)
# onCreate(savedInstanceState : Bundle)
+ updateUserInformation(userID : String)
OrganizerEventListActivity
listView: ListView
arrayList: ArrayList<Integer>
orgEventListBackBtn: ImageButton
createEventButton: FloatingActionButton
controller: FirestoreController
eventAdapter: EventAdapter
+ onGetEventsLists(List<Event>)
# onCreate(Bundle)
AttendeeBrowseEventsActivity
listView : ListView
events : List<Event>
attBrowseEventsBackBtn : ImageButton
controller : FirestoreController
eventAdapter : EventAdapter
+ onGetEvents(events : List<Event>)
# onCreate(savedInstanceState : Bundle)
AttendeeEventDetails
attEventBackBtton : ImageButton
poster : ImageView
eventName : TextView
eventDate : TextView
eventLocation : TextView
eventDetails : TextView
registerButton : Button
registrationOpen : Boolean
registrationMax : int
currentlyRegistered : int
displayEvent : Event
controller : FirestoreController
+ onGetEvent(displayEvent : Event)
+ onGetRegistrationInfo(objects : Object[])
# onCreate(savedInstanceState : Bundle)
+ setFields()
+ getProperDateFormatting(date : String) : String
+ registrationBtnClick()
AttendeeRegisteredAdapter
context : Context
userList : List<Registration>
inflater : LayoutInflater
+ AttendeeRegisteredAdapter(context : Context, userList : List<Registration>)
+ getCount() : int
+ getItem(position : int) : Object
+ getItemId(position : int) : long
+ getView(position : int, convertView : View, parent : ViewGroup) : View
AttendeeCheckInAdapter
context : Context
userList : List<Attendance>
inflater : LayoutInflater
+ AttendeeCheckInAdapter(context : Context, userList : List<Attendance>)
+ getCount() : int
+ getItem(position : int) : Object
+ getItemId(position : int) : long
+ getView(position : int, convertView : View, parent : ViewGroup) : View
EventAdapter
context : Context
eventList : List<Event>
inflater : LayoutInflater
+ EventAdapter(context : Context, eventList : List<Event>)
+ getCount() : int
+ getItem(position : int) : Event
+ getItemId(position : int) : long
+ getView(position : int, convertView : View, parent : ViewGroup) : View
+ getProperDateFormatting(date : String) : String
Adapts
Adapts
Adapts
Adapts
Used by
Used by
Used by
Used by
Used by
Attendance
- eventID: String
- timesCheckedIn: int
- userID: String
+ Attendance()
+ Attendance(DocumentSnapshot documentSnapshot)
+ Attendance(String eventID, int timesCheckedIn, String userID)
+ getEventID(): String
+ setEventID(String eventID)
+ getTimesCheckedIn(): int
+ setTimesCheckedIn(int timesCheckedIn)
+ getUserID(): String
+ setUserID(String userID)
Event
- eventName: String
- eventLocation: String
- eventDescription: String
- eventDate: String
- eventTime: String
- signUpLimit: int
- currentlySignedUp: int
- signUpLimitBool: Boolean
- geolocation: Boolean
- reUseQR: Boolean
- newQR: Boolean
- posterRef: String
- shareQRId: String
- checkInQRId: String
- ownerID: String
- eventID: String
+ Event(String, String, String, String, String, int, int, Boolean, Boolean, Boolean, Boolean, String, String, String, String, String, String)
+ Event()
+ Event(DocumentSnapshot)
+ getEventName(): String
+ setEventName(String)
+ getEventLocation(): String
+ setEventLocation(String)
+ getEventDescription(): String
+ setEventDescription(String)
+ getEventDate(): String
+ setEventDate(String)
+ getEventTime(): String
+ setEventTime(String)
+ getSignUpLimit(): int
+ setSignUpLimit(int)
+ setCurrentlySignedUp(int)
+ getCurrentlySignedUp(): int
+ getSignUpLimitBool(): Boolean
+ setSignUpLimitBool(Boolean)
+ getGeolocation(): Boolean
+ setGeolocation(Boolean)
+ getReUseQR(): Boolean
+ setReUseQR(Boolean)
+ getNewQR(): Boolean
+ setNewQR(Boolean)
+ getPosterRef(): String
+ setPosterRef(String)
+ getShareQRId(): String
+ setShareQRId(String)
+ getCheckInQRId(): String
+ setCheckInQRId(String)
+ getOwnerID(): String
+ setOwnerID(String)
+ getEventID(): String
+ setEventID(String)
Notification
- eventID: String
- title: String
- description: String
+ Notification(String, String, String)
+ Notification()
+ getEventID(): String
+ setEventID(String)
+ getTitle(): String
+ setTitle(String)
+ getDescription(): String
+ setDescription(String)
Registration
- eventID: String
- userID: String
+ Registration()
+ Registration(DocumentSnapshot)
+ Registration(String, String)
+ getEventID(): String
+ setEventID(String)
+ getUserID()
+ setUserID(String)
NotificationAdapter
- context: Context
- notificationList: List<Notification>
- inflater: LayoutInflater
+ NotificationAdapter(Context, List<Notification>)
+ getCount(): int
+ getItem(int): Object
+ getItemId(int): long
+ getView(int, View, ViewGroup): View
QrCode
- qrId: String
- checkIn: boolean
- eventID: String
- image: String
+ QrCode()
+ QrCode(boolean, String, String)
+ isCheckedIn(): boolean
+ setCheckIn(boolean)
+ getEventID(): String
+ setEventID(String)
+ getImage(): String
+ setImage(String)
+ getQrId(): String
+ setQrId(String)
User
- email : String
- firstName : String
- lastName : String
- id : String
- phoneNumber : String
- geolocation : Boolean
- latlong : GeoPoint
- wantNotifications : Boolean
- fcmToken : String
- admin: Boolean
+ User()
+ User(email : String, firstName : String, lastName : String, id : String, phoneNumber : String, geolocation : Boolean, latlong : GeoPoint, wantNotifications : Boolean, fcmToken : String)
+ getEmail() : String
+ setEmail(email : String)
+ getFirstName() : String
+ setFirstName(firstName : String)
+ getLastName() : String
+ setLastName(lastName : String)
+ getId() : String
+ setId(id : String)
+ getPhoneNumber() : String
+ setPhoneNumber(phoneNumber : String)
+ getGeolocation() : Boolean
+ setGeolocation(geolocation : Boolean)
+ getLatlong() : GeoPoint
+ setLatlong(latlong : GeoPoint)
+ getWantNotifications() : Boolean
+ setWantNotifications(wantNotifications : Boolean)
+ getFcmToken() : String
+ setFcmToken(fcmToken : String)
+ getAdmin() : Boolean
+ setAdmin(boolean)
RALLY
UP