ManageFacilityView
- TAG: String
- facilityImageView: ImageView
- facilityNameTextView: TextView
- facilityLocationTextView: TextView
- addEditFacilityButton: Button
- deleteFacilityButton: Button
- controller: ManageFacilityController
- currentFacility: Facility
+ onCreate(savedInstanceState:Bundle): void
+ onResume(): void
+ loadFacility(): void
+ updateUIWithFacility(facility:Facility): void
+ clearFacilityUI(): void
+ confirmDeleteFacility(facility:Facility): void
+ deleteFacility(facility:Facility): void

BrowseImagesController
- storage: FirebaseStorage
+ BrowseImagesController()
+ fetchImages(folders:String[], onImageFetched: Consumer<StorageImage>, onComplete: BiConsumer<Boolean,String>)

HomePageController
- context: Context
- events: List<Event>
+ HomePageController(context:Context, events: List<Event>)
+ getView(position:int, convertView:View, parent: ViewGroup): View

BrowseFacilityController
- db: FirebaseFirestore
+ BrowseFacilitiesController()
+ fetchFacilities(onSuccess:Consumer<List<Facility>>, onFailure: Consumer<String>)

EntrantsListController
- repository: EntrantListRepository
- repoList: HashMap<String, String>
- waitlistFragment: WaitingListView
- attendeeFragment: AttendeesFragment
- entrantList: EntrantList
+ EntrantListController()
+ fetchEntrantList(eventId:String, status: String, callback: EntrantListRepository.FetchEntrantListCallback)
+ drawAttendee(eventId:String; redraw: boolean, context: Context)

ManageFacilityController
- TAG: String
- repository: FacilityRepository
+ ManageFacilityController(context:Context)
+ loadFacility(listener:ManageFacilitiesListener, deviceId: String)
+ saveFacility(name: String, location: String, imageUrl: Uri, listener: ManageFacilitiesListener, deviceId: String)
+ deleteFacility(ManageFacilitiesListener, deviceId: String)

AddFacilityController
- repository: FacilityRepository
+ AddFacilityController()
+ saveFacility(name:String, location:String, imageUri:String, listener: AddFacilityListener, deviceId: String)
+ proceedToSave(name: String, location: String, imageUrl: String, listener: AddFacilityListener, deviceId: String)
+ loadFacility(listener:AddFacilityListener, deviceId: String)

BrowseEventController
- db: FirebaseFirestore
+ BrowseEventsController()
+ fetchEvents(onSuccess:Consumer<List<Event>>, onFailure: Consumer<String>)
+ deleteEvent(event: Event, onSuccess: Runnable, onFailure: Consumer<String>)

EditProfileController
- repository: UserRepository
+ EditProfileController()
+ loadUserProfile(listener:EditProfileListener)
+ saveUserProfile(user: User, imageUri: Uri, listener: EditProfileListener)
+ deleteProfileImage(listener:EditProfileListener)

FacilityRepository
- db: FirebaseFirestore
- storage: FirebaseStorage
+ FacilityRepository()
+ uploadImage(imageUri: Uri, deviceId:String, callback: UploadImageCallback)
+ saveFacility(facility:Facility, deviceId: String, callback: FirestoreCallback)
+ loadFacility(deviceId:String, callback: LoadFacilityCallback)
+ deleteFacility(deviceId:String, callback: FirestoreCallback)
+ getStorageReference(imageUrl: String): StorageReference

User Repository
- db: FirebaseFirestore
- storage: FirebaseStorage
- auth: FirebaseAuth
+ facilityRepository()
+ uploadImage(imageUri: Uri, deviceId: String, callback: UploadImageCallback)
+ saveFacility(facility:Facility, deviceId: String, callback: FirestoreCallback)
+ loadFacility(deviceId:String, callback: LoadFacilityCallback)
+ deleteFacility(deviceId:String, callback: FirestoreCallback)
+ getStorageReference(imageUrl: String): StorageReference

EntrantListRepository
- entrantList: EntrantList
- db: FirebaseFirestore
- auth: FirebaseAuth
+ EntrantListRepository()
+ getEntrantsList(eventId:String, status: String, callback: FirestoreCallback<Long>)
+ getAttendeeListSize(eventId:String, callback: Callback<Long>)
+ sampleAttendees(eventId:String, size: int, context: Context)
+ updateAttendee(eventId:String, attendees: ArrayList<Attendee>, context:Context)
+ sendNotification(userId:String, context: Context)
+ updateAttendeeListCount(eventId:String, size: int)

HomeRepository
- db: FirebaseFirestore
- deviceId: String
+ HomeRepository(context:Context)
+ fetchWaitlistEvents(FetchEventsCallback)
+ fetchSelectedEvents(FetchEventsCallback)
+ removeFromWaitlist(eventName:String)
+ updateEventStatus(eventName:String, newStatus: String)

Facility
- ImageUrl: String
- location: String
- name: String
- id: String
+ Facility(imageUrl: String, location: String, name: String, id: String)
+ getImageUrl(): String
+ setImageUrl(imageUrl:String): void
+ getLocation(): String
+ setLocation(location:String): void
+ getName(): String
+ setName(name: String): void
+ getId(): String
+ setId(id:String): void

StorageImage
- name: String
- path: String
- url: String
+ StorageImage()
+ StorageImage(name: String, path: String, url: String)
+ getName(): String
+ getPath(): String
+ getUrl(): String

User
- userID: String
- name: String
- profileImageUrl: String
- email: String
- dob: String
- phone: String
- country: String
- isAdmin: boolean
- notificationsPerm: boolean
- isOrganizer: boolean
- eventsAttending: Map<String, EventAttendance>
- events: List<Event>
+ User()
+ User(userID: String, name: String, profileImageUrl: String, email: String, dob: String, phone: String, country: String, isAdmin: boolean)
+ getUserID(): String
+ setUserID(id:String)
+ getName(): String
+ setName(name:String): String
+ getProfileImageUrl(): String
+ setProfileImageUrl(profileImageUrl:String)
+ getEmail(): String
+ setEmail(email:String)
+ getDob(): String
+ setDob(dob:String)
+ getPhone(): String
+ setPhone(phone:String)
+ getCountry(): String
+ setCountry(country:String)
+ isAdmin(): boolean
+ setAdmin(isAdmin:boolean)
+ isNotificationsPerm(): boolean
+ setNotificationsPerm(notificationsPerm:boolean)
+ isOrganizer(): boolean
+ setOrganizer(isOrganizer:boolean)
+ getEvents(): List<Event>
+ setEvents(events:List<Event>)
+ getEventsAttending(): Map<String, EventAttendance>
+ setEventsAttending(eventsAttendingMap<String, EventAttendance>)
+ addOrUpdateEventAttendance(String, attendance: EventAttendance)
+ removeEventAttendance(eventId:String)

AdminProfileActivity
- profileImageView: CircleImageView
- editProfileImageButton: ImageButton
- removeProfileImageButton: ImageButton
- backButton: ImageButton
- nameField: EditText
- emailField: EditText
- dobField: EditText
- phoneField: EditText
- countrySpinner: Spinner
- saveChangesButton: Button
- buttonBrowseUsers: Button
- recyclerBrowseUsers: RecyclerView
- buttonBrowseEvents: Button
- buttonBrowseFacilities: Button
- buttonBrowseImages: Button
- buttonBrowseQR: Button
- imageUri: Uri
- isAdmin: boolean
- isOrganizer: boolean
- notificationsPerm: boolean
- myEventsButton: Button
- imagePickerLauncher: ActivityResultLauncher<Intent>
+ onCreate(savedInstanceState:Bundle): void
+ initializeUI()
+ setListeners()
+ openFileChooser()
+ loadUserProfile()
+ saveProfileData()
+ uploadProfileImage(userRef: DocumentReference)
+ deleteProfileImage()
+ showDatePicker()
+ initializeDefaultFields()
+ generateDefaultAvatar(name:String): void
+ BrowseUsers()
+ BrowseEvents()
+ BrowseFacilities()
+ BrowseImages()
+ BrowseQR()
+ openMyEvents()

Attendee
- userId: String
- userName: String
- userEmail: String
- status: String
- latitude: Double
- longitude: Double
+ Attendee()
+ Attendee(userId: String, userName: String, userEmail: String, status: String, latitude: Double, longitude: Double)
+ getUserId(): String
+ setUserId(userId:String): void
+ getUserName(): String
+ setUserName(userName:String): void
+ getUserEmail(): String
+ setUserEmail(userEmail:String): void
+ getStatus(): String
+ setStatus(status:String): void
+ getLatitude(): Double
+ setLatitude(latitude:Double): void
+ getLongitude(): Double
+ setLongitude(longitude:Double): void

EntrantList
- attendees: List<Attendee>
- capacity: int
- sampleSize: int
+ EntrantList(capacity:int, sampleSize: int)
+ getUsers(): List<Attendee>
+ setUser(user: Attendee): void
+ getCapacity(): int
+ setCapacity(capacity:int): void
+ getSampleSize(): int
+ setSampleSize(sampleSize:int): void
+ sampleAttendees(size:int): ArrayList<Attendee>

Event
- eventId: String
- eventName: String
- eventDate: String
- eventDateTime: String
- description: String
- maxAttendees: int
- maxWaitlist: Integer
- currentWaitlist: Integer
- geolocationEnabled: boolean
- qrCodeLink: String
- currentAttendees: int
- organizerId: String
- facility: Facility
+ Event()
+ Event(eventId: String, eventName: String, ...)
+ getEventID(): String
+ setEventId(eventId: String): void
+ getEventName(): String
+ setEventName(eventName:String): void
+ getDate(): String
+ setDate(eventDate:String): void
+ getDateTime(): String
+ setDateTime(eventDateTime:String): void
+ getDescription(): String
+ setDescription(description:String): void
+ getMaxAttendees(): int
+ setMaxAttendees(maxAttendees:int): void
+ getMaxWaitlist(): Integer
+ setMaxWaitlist(maxWaitlist:Integer): void
+ getCurrentWaitlist(): Integer
+ setCurrentWaitlist(currentWaitlist:Integer): void
+ isGeolocationEnabled(): boolean
+ setGeolocationEnabled(geolocation:boolean): void
+ getQrCodeLink(): String
+ setQrCodeLink(qrCodeLink:String): void
+ getPosterUrl(): String
+ setPosterUrl(posterUrl:String): void
+ getCurrentAttendees(): int
+ setCurrentAttendees(currentAttendees:int): void
+ getOrganizerId(): String
+ setOrganizerId(organizerId:String): void
+ getFacility(): Facility
+ setFacility(facility:Facility): void

AttendeeAdapter
- attendeeList: List<Attendee>
- context: Context
- status: String
- eventId: String
- db: FirebaseFirestore
+ AttendeesAdapter(attendeeList: List<Attendee>, context: Context, status: String, eventId: String)
+ onCreateViewHolder(parent: ViewGroup, viewType: int): AttendeeViewHolder
+ onBindViewHolder(holder: AttendeeViewHolder, position:int): void
+ getItemCount(): int
+ cancelAttendee(attendee:Attendee, position:int): void

AvatarUtil
+ generateAvatar(letter: String, size:int, context: Context): Bitmap
+ getColorForLetter(letter: String): int

BrowseUsersView
- context: Context
- userRecyclerView: RecyclerView
- userAdapter: UserAdapter
- controller: BrowseUsersController
+ BrowseUsersView(context:Context, recyclerView: RecyclerView, controller: BrowseUsersController, currentUserId: String)
+ setToolbar(toolbar: Toolbar, onBackPressed: Runnable)
+ loadUsers(): void

BrowseImagesView
- context: Context
- imagesRecyclerView: RecyclerView
- progressBar: ProgressBar
- imageAdapter: ImageAdapter
- controller: BrowseImagesController
+ BrowseImagesView(context:Context, recyclerView: RecyclerView, progressBar: ProgressBar, controller: BrowseImagesController, numberOfColumns: int)
+ setToolbar(toolbar: Toolbar, onBackPressed: Runnable): void
+ loadImages(folders: String[]): void

BrowseEventsView
- context: Context
- eventRecyclerView: RecyclerView
- eventAdapterAdmin: EventAdapterAdmin
- controller: BrowseEventsController
+ BrowseEventsView(context:Context, recyclerView: RecyclerView, controller: BrowseEventsController)
+ setToolbar(toolbar: Toolbar, onBackPressed: Runnable): void
+ loadEvents(): void
+ deleteEvent(event: Event): void

AttendeesFragment
- TAG: String
- ARG_EVENT_ID: String
- controller: EntrantListController
- recyclerViewSelected: RecyclerView
- recyclerViewConfirmed: RecyclerView
- recyclerViewCancelled: RecyclerView
- selectedAdapter: AttendeesAdapter
- confirmedAdapter: AttendeesAdapter
- cancelledAdapter: AttendeesAdapter
- noneConfirmed: TextView
- noneSelected: TextView
- noneCancelled: TextView
- drawButton: Button
- selectedList: List<Attendee>
- confirmedList: List<Attendee>
- cancelledList: List<Attendee>
- selected: String
- confirmed: String
- cancelled: String
- eventId: String
- db: FirebaseFirestore
+ newInstance(eventId: String): AttendeesFragment
+ onCreate(savedInstanceState:Bundle): void
+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState:Bundle): View
+ fetchAttendees(): void
+ updateEmptyListMessages(): void

AddFacilityView
- TAG: String
- facilityImageView: ImageView
- facilityNameField: EditText
- facilityLocationField: EditText
- facilityImageUri: Uri
- uploadFacilityImageButton: Button
- saveFacilityButton: Button
- controller: AddFacilityController
- imagePickerLauncher: ActivityResultLauncher<Intent>
+ onCreate(savedInstanceState:Bundle): void
+ openImagePicker(): void
+ saveFacilityDetails(): void
+ onActivityResult(requestCode:int, resultCode: int, data: Intent): void
+ onFacilityLoaded(facility:Facility): void
+ onFacilityLoadFailed(e:Exception): void
+ onFacilitySavedSuccessfully(): void
+ onFacilitySaveFailed(e:Exception): void
+ onImageUploaded(imageUrl:String): void
+ onImageUploadFailed(e:Exception): void

BrowseFacilitiesView
- context: Context
- facilityRecyclerView: RecyclerView
- facilityAdapter: FacilityAdapterAdmin
- controller: BrowseFacilitiesController
+ BrowseFacilitiesView(context:Context, recyclerView: RecyclerView, controller: BrowseFacilitiesController)
+ setToolbar(toolbar: Toolbar, onBackPressed: Runnable): void
+ loadFacilities(): void

BaseActivity
- db: FirebaseFirestore
retrieveDeviceId(): String
getUserName(callback: UserNameCallback): void
getUserType(callback: UserTypeCallback): void

BrowseImagesActivity
- view: BrowseImagesView
+ onCreate(savedInstanceState: Bundle): void
+ onOptionsItemSelected(item: MenuItem): boolean

BrowseUsersActivity
- view: BrowseUsersView
+ onCreate(savedInstanceState: Bundle): void
+ onOptionsItemSelected(item: MenuItem): boolean

EntrantProfileView
- nameField: EditText
- dobField: EditText
- emailField: EditText
- phoneField: EditText
- profileImage: ImageView
- saveButton: Button
- datePickerDialog: DatePickerDialog
- calendar: Calendar
- imageUri: Uri
- entrantId: String
+ onCreate(savedInstanceState:Bundle): void
+ loadUserProfile(): void
+ updateProfile(): void
+ validateDOB(String):boolean
+ validateDOB(String):boolean
+ onProfileLoadFailed(String): void
+ onSaveButtonClick(): void

HomeView
- selectedEventsListView: ListView
- selectedEventsAdapter: SelectedEventsAdapter
- selectedEventsList: List<Event>
- waitlistEventsAdapter: HomePageController
- selectedEvents: List<Event>
- waitlistEvents: List<Event>
+ onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState:Bundle?): View
+ fetchWaitlistEvents(): void
+ fetchSelectedEvents(): void
+ updateWaitlistEvents(fetchedWaitlistEventsList:List<Event>): void
+ updateSelectedEvents(fetchedSelectedEvents:List<Event>): void

BrowseEventsActivity
- view: BrowseEventsVIEW
+ onCreate(savedInstanceState: Bundle): void
+ onOptionsItemSelected(item: MenuItem): boolean

BrowseFacilitiesActivity
- view: BrowseFacilitiesView
+ onCreate(savedInstanceState: Bundle): void
+ onOptionsItemSelected(item: MenuItem): boolean

NotificationService
- BASE_NOTIFICATION_ID: int
+ sendNotification(user:UserProfile, Context, title:String, description:String): void
+ optOutOfNotifications(user:UserProfile): void
+ optInToNotifications(context:Context, user: UserProfile): void
+ updateChosenFromWaitingListPreference(user: UserProfile, preference: boolean): void
+ updateNotChosenFromWaitingListPreference(user: UserProfile, preference: boolean): void

NotificationUtils
- CHANNEL_ID: String
+ getChannelId(): String
+ createNotificationChannel(context: Context): void

OrganizationNotificationActivity
- recyclerView: RecyclerView
- eventAdapter: OrganizerEventAdapter
- db: FirebaseFirestore
- deviceId: String
+ onCreate(savedInstanceState:Bundle): void
+ fetchOrganizerEvents(): void
+ setToolbar(toolbar: Toolbar, onBackPressed: Runnable): void
+ onOptionsItemSelected(item:MenuItem): boolean

WaitingList
- waitingList: List<UserProfile>
- random: Random
+ WaitingList()
+ addWaiter(user: UserProfile, capacity:int): boolean
+ removeWaiter(user: UserProfile): boolean
+ sampleAttendees(numAttendees:int): List<UserProfile>
+ getWaitingList(): List<UserProfile>
+ clearWaitingList(): void
+ getAllAttendees():List<UserProfile>

EntrantsMapFragment
- mapView: MapView
- googleMap: GoogleMap
- db: FirebaseFirestore
- TAG: String
+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState:Bundle): View
+ onMapReady(map: GoogleMap): void
- fetchEntrantsLocations(): void
+ onResume(): void
+ onStart(): void
+ onStop(): void
+ onPause(): void
+ onLowMemory(): void
+ onDestroy(): void
+ onSaveInstanceState(outState:Bundle): void

EventDetailsActivity
- eventNameTextView: TextView
- dateTextView: TextView
- timeTextView: TextView
- descriptionTextView: TextView
- posterImageView: ImageView
- db: FirebaseFirestore
- eventId: String
+ OnCreate(savedInstanceState:Bundle): void
- loadEventDetails()

Details Fragment
- eventId: String
- db: FirebaseFirestore
- storage: FirebaseStorage
- storageReference: StorageReference
- eventNameTextView: TextView
- dateTextView: TextView
- timeTextView: TextView
- descriptionTextView: TextView
- maxAttendeesTextView: TextView
- maxWaitlistTextView: TextView
- geolocationTextView: TextView
- posterImageView: ImageView
- qrCodeImageView: ImageView
- progressBar: ProgressBar
- imagePickerLauncher: ActivityResultLauncher<Intent>
- detailsUpdatePosterButtonButton
+ newInstance(eventId:String): DetailsFragment
+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
+ fetchEventDetails(): void
+ generateQRCode(qrLink:String): void
+ showUpdatePosterConfirmationDialog(): void
+ openImagePicker(): void
+ uploadNewPoster(imageUri:Uri): void
+ updatePosterUrlInFirestore(posterUrl: String): void

GroupEntrantsActivity
- textViewEntrants: TextView
- buttonSendNotification: Button
- groupType : String
+ onCreate(savedInstanceState:Bundle): void
+ loadEntrantsList(groupType:String)
+ sendNotificationToGroup(): void

CreateEventActivity
- eventNameEditText: EditText
- dateEditText: EditText
- timeEditText: EditText
- descriptionEditText: EditText
- maxAttendeesEditText: EditText
- maxWaitlistEditText: EditText
- geolocationCheckbox: CheckBox
- saveButton: Button
- db: FirebaseFirestore
- eventsRef: CollectionReference
+ onCreate(savedInstanceState:Bundle): void
+ validateInputs(): boolean
+ saveEvent(): void

EventWaitlistActivity
- eventId: String
- eventName: String
- expandableListView: ExpandableListView
- adapter: WaitlistExpandableListAdapter
- sendNotificationButton: Button
- db: FirebaseFirestore
- listGroupTitles: List<String>
- listData: HashMap<String, List<String>>
+ onCreate(savedInstanceState:Bundle): void
+ fetchWaitlistData(): void
+ sendNotificationToSelectedGroups(): void
+ sendNotificationToUser(userId: String, title: String, message: String): void

ImagePagerAdapter
- context: Context
- imageList: List<StorageImage>
- storage: FirebaseStorage
+ ImageAdapter(context: Context, imageList: List<StorageImage>)
+ onCreateViewHolder(parent: ViewGroup, viewType: int): ImageViewHolder
+ onBindViewHolder(holder: ImageViewHolder, position:int): void
+ getItemCount(): int
+ getImageUrl(position:int): StorageImage
+ deleteImage(image: StorageImage, position: int): void

ManageQrLinkActivity
- adapter: QRLinkAdapter
- db: FirebaseFirestore
- eventList: List<Event>
+ onCreate(savedInstanceState:Bundle): void
+ loadEventsWithQrHash(): void
+ deleteQrHash(eventId: String): void

QrLinkAdapter
- eventList: List<Event>
- deleteQrHashListener: OnDeleteQrHashListener
+ onCreateViewHolder(parent: ViewGroup, viewType: int): ViewHolder
+ onBindViewHolder(holder: ViewHolder, position:int): void
+ getItemCount(): int

QRSacannerActivity
- previewView: PreviewView
- flashToggleButton: ImageButton
- isFlashOn: boolean
- isScanning: boolean
- cameraProviderFuture: ListenableFuture<ProcessCameraProvider>
- cameraExecutor: ExecutorService
- barcodeScanner: BarcodeScanner
- camera: Camera
+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
+ startCamera(): void
+ bindCameraUseCases(cameraProvider: ProcessCameraProvider): void
+ processImageProxy(imageProxy: ImageProxy): void
+ handleScannedData(data: String): void
+ toggleFlash(): void
+ onResume(): void
+ onDestroy()

WaitlistExpandableListAdapter
- TAG: String (static) - context: Context
- listGroupTitles: List<String>
- listData: HashMap<String, List<String>>
- selectedGroups: Set<String>
+ WaitlistExpandableListAdapter(context: Context, listGroupTitles: List<String>, listData: HashMap<String, List<String>>)
+ getChildrenCount(groupPosition:int): int
+ getGroup(groupPosition:int): Object
+ getChild(groupPosition:int, childPosition:int): Object
+ getGroupId(groupPosition:int): long
+ getChildId(groupPosition:int, childPosition:int): long + hasStableIds(): boolean
+ getGroupView(groupPosition:int, isExpanded: boolean, convertView: View, parent: ViewGroup): View
+ getChildView(groupPosition:int, childPosition:int, isLastChild: boolean, convertView: View, parent: ViewGroup): View
+ isChildSelectable(groupPosition:int, childPosition:int): boolean
+ getSelectedGroups(): List<String>
+ capitalize(text: String): String

EventDetailsActivity
- eventId: String
- progressBar: ProgressBar
- buttonDetails: Button
- buttonWaitlist: Button
- buttonEntrants: Button
- buttonLocations: Button
+ onCreate(savedInstanceState:Bundle): void
+ loadFragment(fragment: Fragment, tag: String): void
+ refreshAttendees(): void

EventPageAdapter
- eventId: String
+ createFragment(position:int): Fragment
+ getItemCount(): int

HomeFragment
- eventsListView: RecyclerView
- db: FirebaseFirestore - eventList: List<Event>
+ OrgEventRepo(context:Context, eventDataList: ArrayList<Event>)
+ getView(position:int, convertView:View, parent: ViewGroup): void
+ updateEventDataList(eventDataList: ArrayList<Event>): void

OrgEventRepo
- context: Context
- eventDataList: ArrayList<Event>
+ OrgEventRepo(context:Context, eventDataList: ArrayList<Event>)
+ getView(position:int, convertView:View, parent: ViewGroup): void
+ updateEventDataList(eventDataList: ArrayList<Event>):void

WaitlistFragment
- ARG_EVENT_ID: String (constant)
- eventId: String
+ WaitlistFragment() : constructor
+ newInstance(eventId:String): WaitlistFragment
+ onCreate(savedInstanceState:Bundle): void
+ onCreateView(inflater:LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View

UserAdapter
- context: Context
- userList: List<User>
- db: FirebaseFirestore
- currentUserId: String
+ UserAdapter(context:Context, userList: List<User>, currentUserId: String)
+ onCreateViewHolder(parent: ViewGroup, viewType: int): ViewHolder
+ onBindViewHolder(holder: ViewHolder, position:int): void
+ getItemCount(): int

MainActivity
- mAuth: FirebaseAuth
- bottomNavigationView: BottomNavigationView
- userManager: UserManager
+ onCreate(savedInstanceState:Bundle): void
+ signInAnonymously(): void

LocationsFragment
- eventId: String
- mapView: MapView
- db: FirebaseFirestore
+ newInstance(eventId:String): LocationsFragment
+ onCreateView(savedInstanceState:Bundle):
+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
+ updateMapMarkers(): void
+ adjustMapView(documents: List<DocumentSnapshot>): void

UserProfile
- id: int
- name: String
- receiveNotifications:boolean
- chosenFromWaitingList: boolean
- notChosenFromWaitingList: boolean
- user: UserProfile
+ UserProfile()
+ getId(): int
+ getName(): String
+ isRecievingNotifications(): boolean
+ setReceiveNotifications(receiveNotifications:boolean): void
+ isChosenFromWaitingList(): boolean
+ setChosenFromWaitingList(chosenFromWaitingList: boolean): void
+ isNotChosenFromWaitingList(): boolean
+ setNotifyNotChosenFromWaitingList(notChosenFromWaitingList: boolean): void
+ equals(o: Object): boolean
+ hashCode(): int

UserManager
- TAG: String (static final)
- db: FirebaseFirestore
- deviceId: String
+ UserManager(context:Context)
+ retrieveDeviceId(context:Context): String
+ fetchUserRole(context:Context, callback: RoleCallback): void
+ navigateToProfile(context:Context, role: String): void
+ promoteToOrganizer(context: Context): void