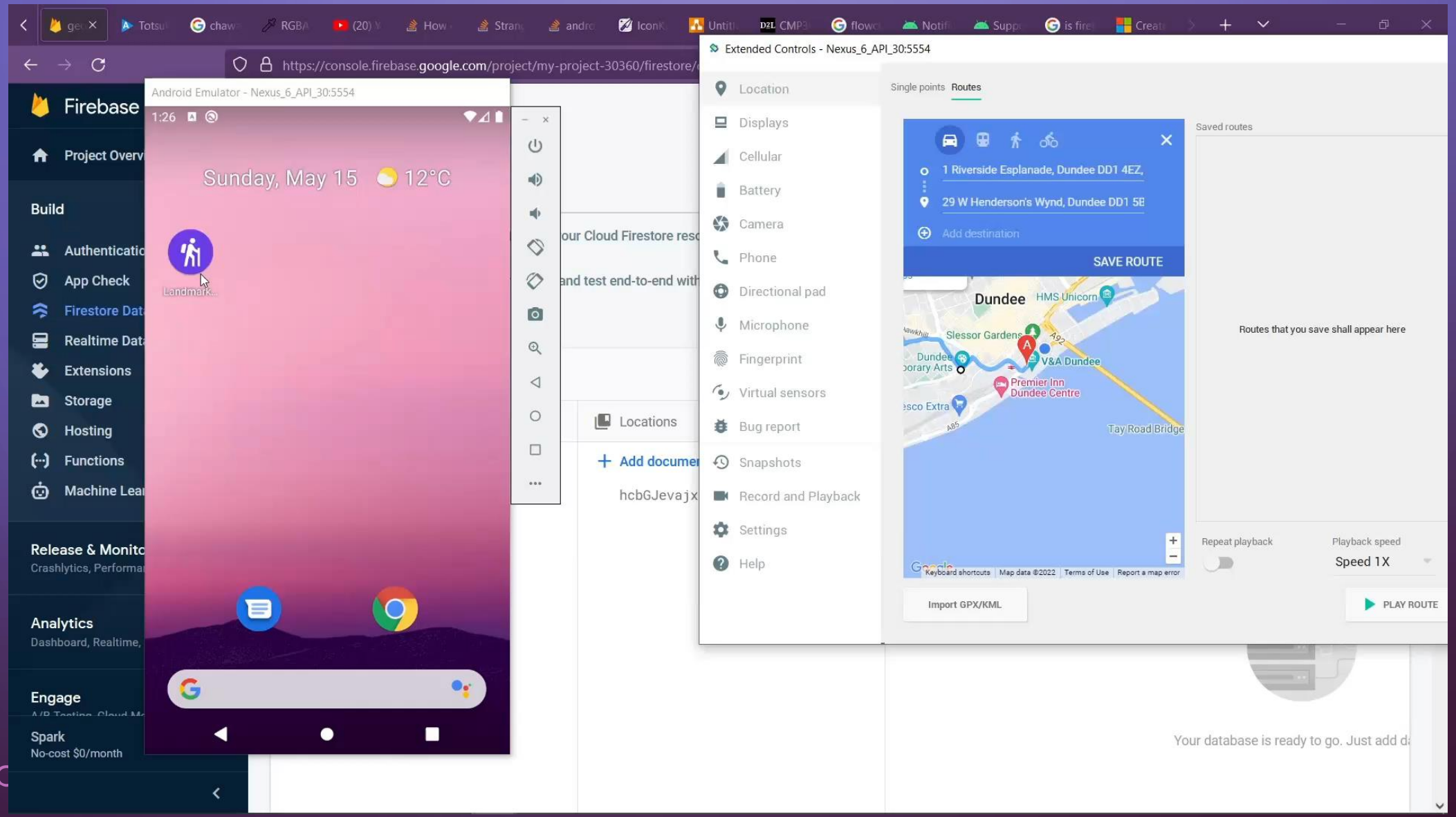
LANDMARK IN YOUR AREA!

MARTIN GEORGIEV – STUDENT №1901560

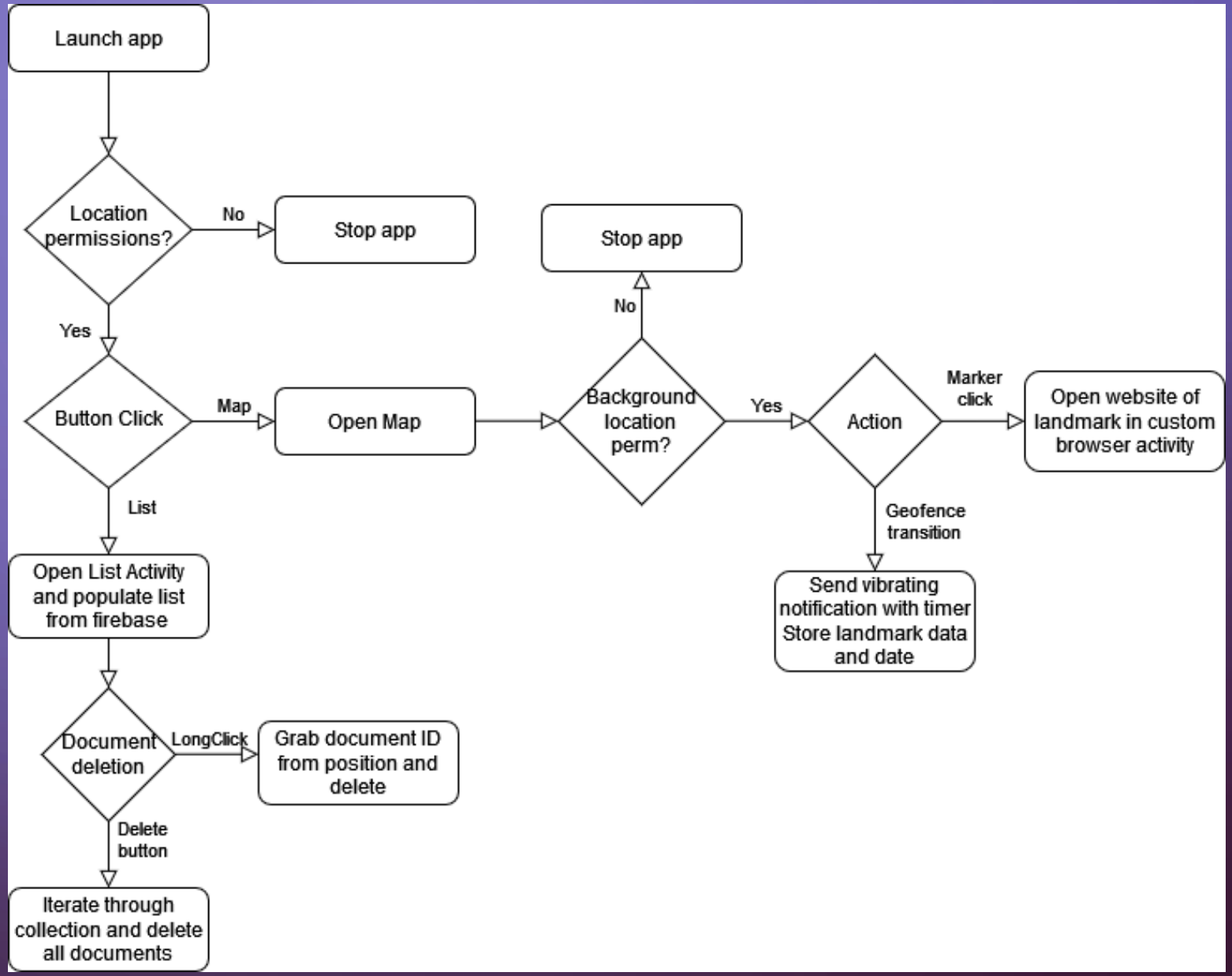
SYNOPSIS

- Video demo
- Flow diagram
- Code
  - Geofences/Map
  - Notifications
  - Location History
- Overview and critical analysis
- References

VIDEO DEMO

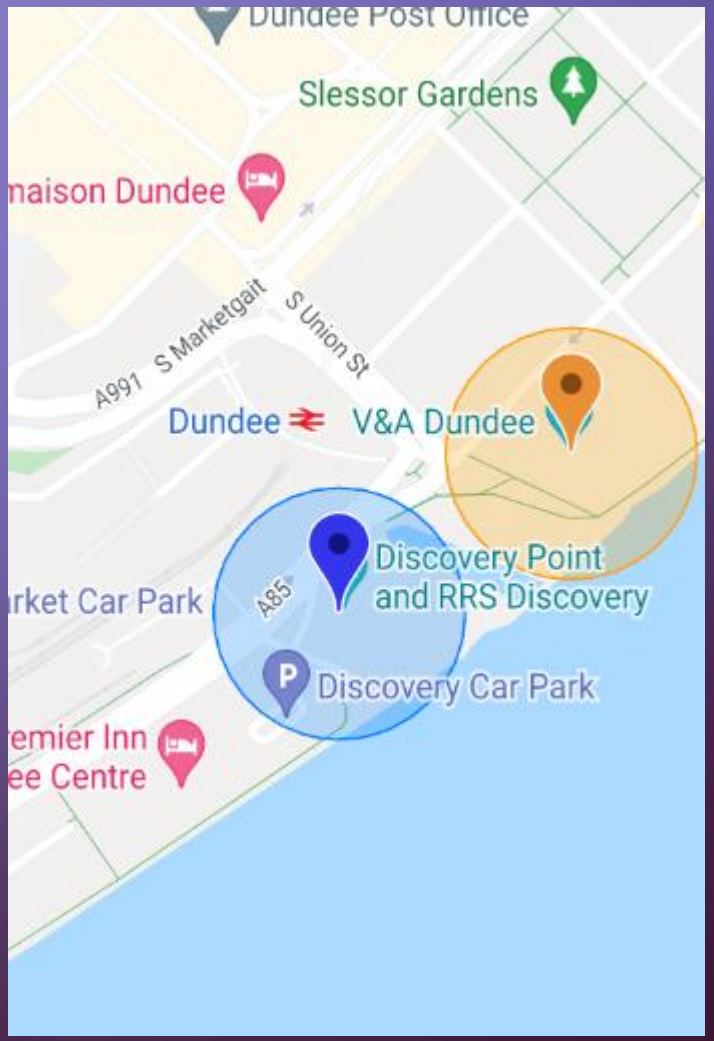

Extended Controls - Nexus_6_API_30:5554
Location
Displays
Cellular
Battery
Camera
Phone
Directional pad
Microphone
Fingerprint
Virtual sensors
Bug report
Snapshots
Record and Playback
Settings
Help
Single points
Routes
1 Riverside Esplanade, Dundee DD1 4EZ,
29 W Henderson's Wynd, Dundee DD1 5E
Add destination
SAVE ROUTE
Saved routes
Routes that you save shall appear here
Dundee
HMS Unicorn
Slessor Gardens
V&A Dundee
Premier Inn Dundee Centre
Tay Road Bridge
Repeat playback
Playback speed
Speed 1X
Import GPX/KML
PLAY ROUTE
Android Emulator - Nexus_6_API_30:5554
1:26
Sunday, May 15
12°C
Firebase
Build
Authentication
App Check
Firestore Database
Realtime Database
Extensions
Storage
Hosting
Functions
Machine Learning
Release & Monitor
Analytics
Engage
Spark
No-cost $0/month
Locations
Your database is ready to go.

FLOW DIAGRAM



CODE EXAMPLES – GEOFENCES

- Implemented with three class files:
  - MapsActivity
  - GeofenceHelper
  - GeofenceBroadcastReceiver
- What does each do?
- Code examples



MAPS ACTIVITY

- Uses Google Maps API
- Draws the geofences, adds markers, zooms onto user's current location
- Uses the GeofenceHelper to send data to the BroadcastReceiver

MAPS ACTIVITY – ADDMARKER

```
// Method which adds a marker on the map
private void addMarker(LatLng latLng, String ID){
    MarkerOptions markerOptions = new MarkerOptions().position(latLng);
    // Switch statement taking the geofence ID and adding the appropriate title and colour for the landmark marker.
    switch (ID) {
        case GEOFENCE_MCMANUS:
            markerOptions.title("The McManus Art and Museum Gallery");
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));
            break;
        case GEOFENCE_VA:
            markerOptions.title("V&A Dundee");
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ORANGE));
            break;
        case GEOFENCE_VERDANT:
            markerOptions.title("Verdant Works");
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_CYAN));
            break;
        case GEOFENCE_DISCOVERY:
            markerOptions.title("Discovery Point and RSS Discovery");
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE));
            break;
        case GEOFENCE_ARTS:
```

MAPS ACTIVITY – ADDCIRCLE

```
// Adds a geofence circle based on ID - specifies radius, center location and ARGB colours
private void addCircle (LatLng latLng, float radius, String ID){
    CircleOptions circleOptions = new CircleOptions();
    switch (ID) {
        case GEOFENCE_MCMANUS:
            circleOptions.center(latLng);
            circleOptions.radius(radius);
            circleOptions.strokeColor(Color.argb( alpha: 255,  red: 255,  green: 0,  blue: 0));
            circleOptions.fillColor(Color.argb( alpha: 64, red: 255, green: 0, blue: 0));
            circleOptions.strokeWidth(4);
            break;
        case GEOFENCE_VA:
            circleOptions.center(latLng);
            circleOptions.radius(radius);
            circleOptions.strokeColor(Color.argb( alpha: 255,  red: 255,  green: 162,  blue: 0));
            circleOptions.fillColor(Color.argb( alpha: 64, red: 255, green: 162, blue: 0));
            circleOptions.strokeWidth(4);
            break;
        case GEOFENCE_VERDANT:
```

MAPSACTIVITY – CREATING THE GEOFENCE

```java
// Method used to create the geofence by taking the location and the geofence ID
private void createGeoFence (LatLng latLng, String ID) {
    //mMap.clear();
    addMarker(latLng, ID);
    addCircle(latLng, GEOFENCE_RADIUS, ID);
    addGeofence(latLng, GEOFENCE_RADIUS, ID);
}

/* Method which makes use of a helper to add geofences based on radius, location and ID. It is used in createGeoFence.
   It also monitors the user transition with the helper. Additionally, it sends a pending intent to the BroadcastReceiver
   class which handles the transition types. */
@SuppressLint("MissingPermission")
private void addGeofence(LatLng latLng, float radius, String ID){
    Geofence geofence = geofenceHelper.getGeofence(ID , latLng, radius, transitionTypes: Geofence.GEOFENCE_TRANSITION_ENTER | Geofence.GEOFEI
    GeofencingRequest geofencingRequest = geofenceHelper.getGeofencingRequest(geofence);
    PendingIntent pendingIntent = geofenceHelper.getPendingIntent();
    geofencingClient.addGeofences(geofencingRequest, pendingIntent)
            .addOnSuccessListener(unused -> Log.d(TAG, msg: "Geofence Added." + ID))
            .addOnFailureListener(e -> {
                String errorMessage = geofenceHelper.getErrorString(e);
                Log.e(TAG, msg: "onFailure" + errorMessage);
            });

}
```

GEOFENCEHELPER

```
// Geofencing request method which builds the fences and sets the initial trigger type
public GeofencingRequest getGeofencingRequest(Geofence geoFence){
    return new GeofencingRequest.Builder()
            .addGeofence(geoFence)
            .setInitialTrigger(GeofencingRequest.INITIAL_TRIGGER_ENTER)
            .build();

}

/* Method which gets the geofence data, transition types and loitering delay
   (time spent in/within/out to trigger a transition) */
public Geofence getGeofence (String ID, LatLng latLng, float radius, int transitionTypes) {
    geofenceID = ID;
    return new Geofence.Builder()
            .setCircularRegion(latLng.latitude, latLng.longitude, radius)
            .setRequestId(ID)
            .setTransitionTypes(transitionTypes)
            .setLoiteringDelay(5000)
            .setExpirationDuration(Geofence.NEVER_EXPIRE)
            .build();

}
```

GEOFENCEBROADCASTRECEIVER

```
// Switch case monitoring the transition types
switch (transitionType) {
    // Action based on transition Enter type
    case Geofence.GEOFENCE_TRANSITION_ENTER:
        /* Send notification with the notificationHandler object with the landmark name,
           date of entry and a timer showing the time spent inside the area. */
        notificationHandler.sendDwellNotification( title: "Landmark nearby!",  body: "You have entered " + geofenceList.get(0).getRequestId
        timeEntered = currentTime;
        addLocationEnterToFirebase(geofenceList.get(0).getRequestId(), currentTime);
        break;
    case Geofence.GEOFENCE_TRANSITION_DWELL:
        // Optional functionalities for staying inside the geofence
        break;
    // Action based on transition Leave type
    case Geofence.GEOFENCE_TRANSITION_EXIT:
        /* Send notification with the notificationHandler object with the landmark name,
           and date of leave. Pressing it opens the visited locations list.*/
        notificationHandler.sendLeaveNotification( title: "Warning!",  body: "You have left " + geofenceList.get(0).getRequestId() + "'s ar
        timeLeft = currentTime;
        addLocationLeaveToFirebase(geofenceList.get(0).getRequestId(), currentTime);
        break;
}
```

CODE EXAMPLES – NOTIFICATIONS REVOICE

- A single handler class
- Constructs channels based on the device's API
- Two different functions to create different notifications based on activity

NOTIFICATIONHANDLER – CREATECHANNELS

```java
// Grab API version and build channels
@RequiresApi(api = Build.VERSION_CODES.O)
private void createChannels() {
    NotificationChannel notificationChannel = new NotificationChannel(CHANNEL_ID, CHANNEL_NAME, NotificationManager.IMPORTANCE_HIGH);
    notificationChannel.enableLights(true);
    notificationChannel.enableVibration(true);
    notificationChannel.setDescription("this is the description of the channel.");
    notificationChannel.setLightColor(Color.RED);
    notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);
    NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    manager.createNotificationChannel(notificationChannel);
}
```

NOTIFICATIONHANDLER - NOTIFICATIONS

```java
// Sends notification upon geofence entry
public void sendDwellNotification(String title, String body, Class activityName) {

    Intent intent = new Intent( packageContext: this, activityName);
    PendingIntent pendingIntent = PendingIntent.getActivity( context: this,  requestCode: 267, intent, PendingIntent.FLAG_UPDATE_CURRENT);

    Notification notification = new NotificationCompat.Builder( context: this, CHANNEL_ID)
            // Sets icon, priority ,ticker for time spent and data obtained from the GeofenceBroadcastReceiver
            .setSmallIcon(android.R.drawable.ic_lock_idle_alarm)
            .setTicker("Ticker Text")
            .setWhen(System.currentTimeMillis())  // the time stamp, you will probably use System.currentTimeMillis() for most scenarios
            .setUsesChronometer(true)
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .setStyle(new NotificationCompat.BigTextStyle().setSummaryText("Duration: ").setBigContentTitle(title).bigText(body))
            .setContentIntent(pendingIntent)
            .setAutoCancel(true)
            .build();

    NotificationManagerCompat.from(this).notify(new Random().nextInt(), notification);


}
```

CODE EXAMPLES – LOCATION HISTORY

- Implemented with three classes:
  - FirestoreLocation
  - LocationAdapter
  - LocationListActivity
- Obtains data from firestore and displays it
- Allows user to manipulate the data

LOCATIONADAPTER

```
public View getView(int position, View convertView, ViewGroup parent) {
    // Get the locations data for this position
    FirestoreLocation loc = getItem(position);

    // Check if an existing view is being reused, otherwise inflate the view
    if (convertView == null) {
        convertView = LayoutInflater.from(getContext()).inflate(R.layout.location_layout, parent,  attachToRoot: false);
    }
    // Lookup views

    TextView display_location = (TextView) convertView.findViewById(R.id.display_location);
    TextView display_enter = (TextView) convertView.findViewById(R.id.display_entry);
    TextView display_leave = (TextView) convertView.findViewById(R.id.display_leave);

    // Add the data to the template view
    display_location.setText("Landmark: " + loc.getLocationValue());
    if (loc.getEnterValue() != null ) {
        display_enter.setText("Enter Date: " + loc.getEnterValue().toString());
        display_leave.setVisibility(View.GONE);
    } else if (loc.getLeaveValue() != null ){
        display_enter.setVisibility(View.GONE);
        display_leave.setText("Leave Date: " + loc.getLeaveValue().toString());
    }

    // Return the completed view to render on screen
    return convertView;
}
```

LOCATIONLISTACTIVITY - LONGCLICKDELETE

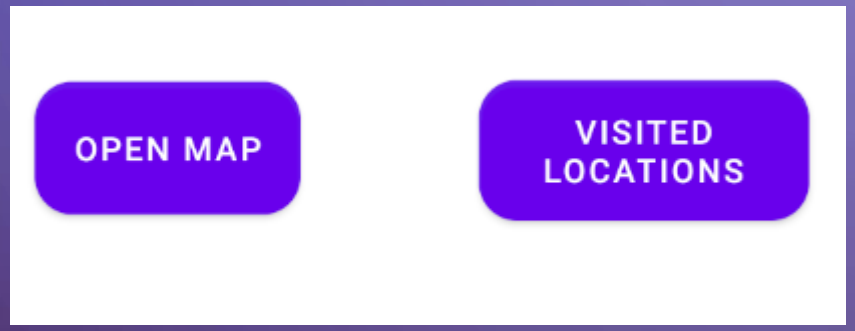
```
// OnLongClick listener for data deletion
listView.setOnItemLongClickListener((parent, view, position, id) -> {
    // Collection path and document ID based on list position
    DocumentReference docRef = db.document( documentPath: "Locations/" + documentIDs.get(position));

    // Deletion alert dialogue
    new AlertDialog.Builder( context: LocationListActivity.this)
            .setTitle("Delete entry")
            .setMessage("Are you sure you want to delete this entry?")

            // Listener forcing the dialog to take action before dismissing itself
            .setPositiveButton(android.R.string.yes, (dialog, which) -> docRef.delete().addOnSuccessListener(aVoid -> {

                // Remove document and refresh activity (visual big within emulator)
                documentIDs.remove(position);
                finish();
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                startActivity(getIntent());
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                listView.invalidate();
            }).addOnFailureListener(e -> Log.e(Utils.TAG, msg: "Failed to delete file!", e)))

            // Null listener closing the dialogue without actions
            .setNegativeButton(android.R.string.no, listener: null)
            // Sets alert icons and shows the dialogue
            .setIcon(android.R.drawable.ic_dialog_alert)
            .show();
```

LOCATIONLISTACTIVITY – DELETE BUTTON

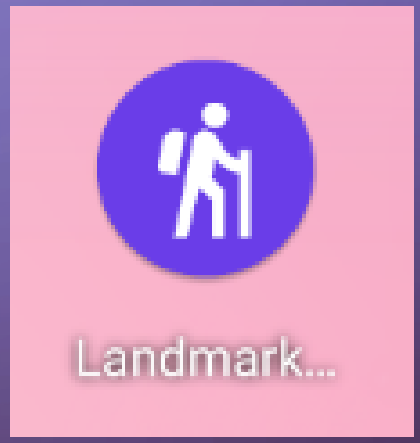
```
// Listener forcing the dialog to take action before dismissing itself
.setPositiveButton(android.R.string.yes, (dialog, which) -> FirebaseFirestore.getInstance().collection( collectionPath: "Locations")
        // Get Collection
        .get() Task<QuerySnapshot>
        .addOnSuccessListener(queryDocumentSnapshots -> {
            // Batch used to mass-delete all documents within the collection
            WriteBatch batch = FirebaseFirestore.getInstance().batch();
            List<DocumentSnapshot> snapshotList = queryDocumentSnapshots.getDocuments();

            for (DocumentSnapshot snapshot:snapshotList) {
                batch.delete(snapshot.getReference());
            }
            // Commit delete action
            batch.commit()
                    .addOnSuccessListener(unused -> Log.d(Utils.TAG,  msg: "Documents Deleted"))
                    .addOnFailureListener(e -> Log.e(Utils.TAG,  msg: "Failure to delete documents: ", e));
            // Clear adapter and notify to clear the list view
            adapter.clear();
            adapter.notifyDataSetChanged();
        })
        .addOnFailureListener(e -> {

        }))
```

COSMETIC FEATURES

- Round buttons



- Custom icon (IconKitchen)



OVERVIEW AND CRITICAL ANALYSIS

- Why this application? – useful for tourists
- Why those features? – more pleasant user experience
- Orientation handling, UI, usability and data persistence

OVERVIEW AND CRITICAL ANALYSIS

- Compatibility – device size and OS version
- Technical analysis – battery life, resources, multi-threading, etc.
- Security – permissions, data storage
- Future work

REFERENCES

- Android Developers. (2005 - Present Day). *Notifications Overview*. Available: https://developer.android.com/guide/topics/ui/notifiers/notifications. Last accessed 25th Mar 2022.
- Android Developers. (2005 - Present Day). *Create and monitor geofences*. Available: https://developer.android.com/training/location/geofencing. Last accessed 12th Apr 2022.
- Android Developers. (2005 - Present Day). *Firebase Documentation*. Available: https://firebase.google.com/docs. Last accessed 20th Apr 2022.
- yoursTruly. (2020). *Locations and Maps*. Available: https://www.youtube.com/watch?v=rNYaEFl6Fms&list=PLdHg5T0SNpN3GBUmpGqjiKGMcBaRT2A-m. Last accessed 26th Apr 2022.
- Android Developers. (2005 - Present Day). *Create and manage notification channels*. Available: https://developer.android.com/training/notify-user/channels. Last accessed 2nd May 2022.
- Android Developers. (2022). *Manage Data*. Available: https://firebase.google.com/docs/firestore/manage-data/transactions#batched-writes. Last accessed 4th May 2022.
- Mentzel, W. (2015). *Rounded button in android*. Available: https://stackoverflow.com/questions/9334618/rounded-button-in-android. Last accessed 5th May 2022.
- Nurik, R. (2021). *IconKitchen*. Available: https://icon.kitchen/. Last accessed 6th May 2022.

THANK YOU FOR YOUR ATTENTION