Setup Mobile Application Environment

ENABLE LOCATION SERVICES TO THE APPLICATION

Date	15 November 2022
Team ID	PNT2022TMID39536
Project Name	CONTAINMENT ZONE ALERTING APPLICATION

Create the location request and set the parameters are shown in this code:

```
  protected void createLocationRequest() {
  LocationRequest locationRequest = LocationRequest.create();
  locationRequest.setInterval(10000);
  locationRequest.setFastestInterval(5000);
  locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
}
```

Get current location settings:

```
LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
    .addLocationRequest(locationRequest);
```

Next check whether the current location settings are satisfied:

```
LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()

SettingsClient client = LocationServices.getSettingsClient(this);
Task<LocationSettingsResponse> task = client.checkLocationSettings(builder.build());
```

Prompt the user to change location settings:

```
task.addOnSuccessListener(this, new OnSuccessListener<LocationSettingsResponse>() {
   @Override
   public void onSuccess(LocationSettingsResponse locationSettingsResponse) {
      // All location settings are satisfied. The client can initialize
      // location requests here.
      // ...
   }
});

task.addOnFailureListener(this, new OnFailureListener() {
   @Override
   public void onFailure(@NonNull Exception e) {
      if (e instanceof ResolvableApiException) {
         // Location settings are not satisfied, but this can be fixed
         // by showing the user a dialog.
         try {
            // Show the dialog by calling startResolutionForResult(),
            // and check the result in onActivityResult().
            ResolvableApiException resolvable = (ResolvableApiException) e;
            resolvable.startResolutionForResult(MainActivity.this,
```

```
                    REQUEST_CHECK_SETTINGS);
            } catch (IntentSender.SendIntentException sendEx) {
            }
        }
    }
});
```

Make a location request:

```
@Override
protected void onResume() {
    super.onResume();
    if (requestingLocationUpdates) {
        startLocationUpdates();
    }
}

private void startLocationUpdates() {
    fusedLocationClient.requestLocationUpdates(locationRequest,
            locationCallback,
            Looper.getMainLooper());
}
```

Define the location update callback:

```
private LocationCallback locationCallback;

// ...

@Override
protected void onCreate(Bundle savedInstanceState) {
```

```
    // ...

    locationCallback = new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult locationResult) {
            if (locationResult == null) {
                return;
            }
            for (Location location : locationResult.getLocations()) {
                // Update UI with location data
                // ...
            }
        }
    };
}
```

Save the state of the activity:

```
@Override
protected void onResume() {
    super.onResume();
    if (requestingLocationUpdates) {
        startLocationUpdates();
    }
}
```

Background location access :

Use the following checklist to identify potential location access logic in the background:

- In your app's manifest, check for the ACCESS_COARSE_LOCATION permission and the ACCESS_FINE_LOCATION permission. Verify that your app requires these location permissions.
- If your app targets Android 10 (API level 29) or higher, also check for the ACCESS_BACKGROUND_LOCATION permission. Verify that your app has a feature that requires it.
- Look for use of location access APIs, such as the Fused Location Provider API, Geofencing API, or LocationManager API, within your code such as in the following constructs:
- Background services
- JobIntentService objects
- WorkManager or JobScheduler tasks
- AlarmManager operations
- Pending intents that are invoked from an app widget
- If your app uses an SDK or library that accesses location, this access is attributed to your app. To determine whether an SDK or library needs location access, consult the library's documentation.