

```
package com.example.covid_19alertapp.activities;

import androidx.annotation.NonNull;

import androidx.fragment.app.FragmentActivity;

import android.content.Context;

import android.content.Intent;

import android.location.Location;

import android.location.LocationManager;

import android.net.wifi.WifiManager;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import com.example.covid_19alertapp.R;

import com.example.covid_19alertapp.extras.AddressReceiver;

import com.example.covid_19alertapp.extras.Internet;

import com.example.covid_19alertapp.extras.LogTags;

import com.google.android.gms.common.api.Status;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.Marker;
```

```
import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.libraries.places.api.Places;

import com.google.android.libraries.places.api.model.Place;

import com.google.android.libraries.places.api.model.TypeFilter;

import com.google.android.libraries.places.api.net.PlacesClient;

import com.google.android.libraries.places.widget.AutoCompleteFragment;

import com.google.android.libraries.places.widget.AutoCompleteSupportFragment;

import com.google.android.libraries.places.widget.listener.PlaceSelectionListener;

import java.util.Arrays;

public class AddressPickerMapsActivity extends FragmentActivity implements

    OnMapReadyCallback,

    GoogleMap.OnMyLocationButtonClickListener,

    GoogleMap.OnMyLocationClickListener,

    GoogleMap.OnMapLongClickListener {

    private GoogleMap mMap;

    private Button confirmButton;

    private Marker homeMarker = null;

    // home address location

    Location pickedLocation;

    // places api client

    PlacesClient placesClient;
```

```
@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_address_picker_maps);

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.

    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map);

    mapFragment.getMapAsync(this);


    if(!Internet.isInternetAvailable(this)) {

        // no internet, map not visible


        Toast.makeText(this, "No internet! Failed to load map.", Toast.LENGTH_LONG)
            .show();


        TextView textView = findViewById(R.id.userHelperText);

        textView.setText(getString(R.string.map_no_internet_text));

    }


    initPlacesApi();


    confirmButton = findViewById(R.id.confirm_button);

}
```

```

private void initPlacesApi() {

    Places.initialize(getApplicationContext(), getString(R.string.google_maps_key));

    placesClient = Places.createClient(this);

    // initialize fragment

    AutocompleteSupportFragment autocompleteFragment =
        (AutocompleteSupportFragment)
        getSupportFragmentManager().findFragmentById(R.id.autocomplete_fragment);

    // specify place type (find out more)

    autocompleteFragment

        .setPlaceFields(Arrays.asList(Place.Field.NAME, Place.Field.LAT_LNG))

        .setCountries("BD")

        .setTypeFilter(TypeFilter.GEOCODE);

    // place selection listener

    autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {

        @Override

        public void onPlaceSelected(@NonNull Place place) {

            // move camera to place

            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(place.getLatLng(), 16.0f));

            Log.d(LogTags.Map_TAG, "onPlaceSelected: place selected = "+place.getName()+"
            "+place.getLatLng());

```

```
}
```

```
@Override
```

```
public void onError(@NonNull Status status) {
```

```
    Toast.makeText(AddressPickerMapsActivity.this, "please try again", Toast.LENGTH_LONG)
```

```
        .show();
```

```
    Log.d(LogTags.Map_TAG, "onError: place selection error = "+status.toString());
```

```
}
```

```
});
```

```
}
```

```
@Override
```

```
public void onMapReady(GoogleMap googleMap) {
```

```
    mMap = googleMap;
```

```
    // Add a marker in Dhaka and move the camera
```

```
    LatLng dhaka = new LatLng(23.7805733, 90.2792376);
```

```
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(dhaka, 10.0f));
```

```

// check if all are needed

mMap.setMyLocationEnabled(true);

mMap.getUiSettings().setMyLocationButtonEnabled(true);

mMap.setOnMyLocationClickListener(this);

mMap.setOnMyLocationButtonClickListener(this);

mMap.setOnMapLongClickListener(this);


Log.d(LogTags.Map_TAG, "onMapReady: map ready");
}


@Override

public void onMapLongClick(LatLng latLng) {

    /*

    location selected by long press on map

    ask user to confirm

    */

    Log.d(LogTags.Map_TAG, "onMapLongClick: marker at = "+latLng.toString());


    pickedLocation = new Location(getLocalClassName());

    pickedLocation.setLatitude(latLng.latitude);

    pickedLocation.setLongitude(latLng.longitude);


    if(homeMarker!=null){

```

```
        homeMarker.remove();  
    }  
}
```

```
homeMarker = mMap.addMarker(new MarkerOptions().position(latLng).title("Home"));
```

```
Toast.makeText(  
    this,  
    "press 'Confirm' to confirm or select another",  
    Toast.LENGTH_LONG  
).show();
```

```
confirmButton.setEnabled(true);  
}
```

```
@Override
```

```
public boolean onMyLocationButtonClick() {  
    /*  
    notify user if location and/or wifi is inactive  
    */  
}
```

```
String toastText = "";  
  
if(!wifiEnabled() && !locationEnabled())  
    toastText = "Turn On both WiFi & Location";  
else if(!locationEnabled())  
    toastText = "Turn On Location";
```

```

else if(!wifiEnabled())

    toastText = "Turn On WiFi";

if(!toastText.equals(""))

    Toast.makeText(this

        , toastText + " to show your location"

        , Toast.LENGTH_LONG)

        .show();

return false;
}

@Override

public void onMyLocationClick(@NonNull Location location) {

    if(location.getAccuracy()>150)

        Toast.makeText(

            this,

            "Location Accuracy is LOW. press again please!" + location, Toast.LENGTH_SHORT

        ).show();

}

public boolean wifiEnabled(){

    WifiManager wifi = (WifiManager) getApplicationContext()

```



```
        .getSystemService(Context.WIFI_SERVICE);

        return wifi.isWifiEnabled();
    }
}
```

```
public boolean locationEnabled(){

    LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

    return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) &&

        locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

}
```

```
public void confirmClicked(View view) {

    /*

    take this location and set it as home address

    */
```

```
Log.d(LogTags.Map_TAG, "confirmClicked: location taken = "+pickedLocation.toString());
```

```
Toast.makeText(this, "Your home location was saved!", Toast.LENGTH_SHORT)

    .show();
```

```
// send data to parent activity
```

```
Intent resultIntent = new Intent();
```

```
resultIntent.putExtra("latitude-longitude",
```

```
    pickedLocation.getLatitude()+"", "+pickedLocation.getLongitude());
```

```
setResult(RESULT_OK, resultIntent);
```

```
finish();
```

```
}
```

```
}
```