package com.example.containmentzone\_alert.activities;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import android.Manifest;

import android.app.Activity;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Build;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.Switch;

import android.widget.Toast;

import com.example.containmentzone\_alert.R;

import com.example.containmentzone\_alert.extras.Constants;

import com.example.containmentzone\_alert.extras.LocationFetch;

import com.example.containmentzone\_alert.extras.LogTags;

import com.example.containmentzone\_alert.extras.Notifications;

import com.example.covid\_19alertapp.containmentzone\_alert.Permissions;

import com.example.containmentzone\_alert.services.BackgroundLocationTracker;

public class TrackingprogressActivity extends AppCompatActivity {

/\*

settings (currently only contains location on/off)

\*/

Button home\_btn;

Switch notification\_switch;

private static boolean switch\_status;

// for location permission

private Permissions permissions;

private static final String[] permissionStrings = {

Manifest.permission.ACCESS\_FINE\_LOCATION,

Manifest.permission.ACCESS\_BACKGROUND\_LOCATION,

Manifest.permission.ACCESS\_WIFI\_STATE

};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_tracker\_settings);

home\_btn= findViewById(R.id.home\_button\_settings);

//start notification channel(do this is MainActivity

Notifications.createNotificationChannel(this);

notification\_switch = findViewById(R.id.notification\_switch);

home\_btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

notification\_switch.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

save\_preferences(notification\_switch.isChecked());

if(notification\_switch.isChecked())

{

try {

LocationFetch.checkDeviceLocationSettings(TrackerSettingsActivity.this);

if(LocationFetch.isLocationEnabled) {

// location is enabled

// start tracker service

Log.d(LogTags.Location\_TAG, "onClick: location found enabled");

// start BackgroundLocationTracker

startTrackerService();

}

else{

// location is not enabled

Log.d(LogTags.Location\_TAG, "onClick: location found disabled");

notification\_switch.setChecked(false);

Toast.makeText(getApplicationContext(), "Turn on location or press again please", Toast.LENGTH\_LONG)

.show();

save\_preferences(false);

}

}catch (Exception e){

// set switch off

notification\_switch.setChecked(false);

// set shared preferences false

save\_preferences(false);

// most probable reason for error is permission not granted

promptPermissions();

Log.d(LogTags.TrackerSettings\_TAG, "onClick: error starting background location service! permission taken?");

}

}

else

{

try {

// stop location tracker

stopService(new Intent(getApplicationContext(),BackgroundLocationTracker.class));

}catch (Exception e){

Log.d(LogTags.TrackerSettings\_TAG, "onClick: error occured!");

}

}

}

});

loadData();

updateViews();

}

private void startTrackerService(){

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

startForegroundService(new Intent(getApplicationContext(), BackgroundLocationTracker.class));

Log.d(LogTags.TrackerSettings\_TAG, "onClick: newer version phones foreground service stared");

} else

startService(new Intent(getApplicationContext(), BackgroundLocationTracker.class));

}

private void promptPermissions() {

permissions = new Permissions(this, permissionStrings, Constants.PERMISSION\_CODE);

if(!permissions.checkPermissions())

permissions.askPermissions();

}

public void save\_preferences(boolean state)

{

SharedPreferences sharedPreferences =

getSharedPreferences(Constants.LOCATION\_SETTINGS\_SHARED\_PREFERENCES, MODE\_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putBoolean(Constants.location\_tracker\_state,state);

editor.apply();

}

public void loadData()

{

SharedPreferences sharedPreferences =

getSharedPreferences(Constants.LOCATION\_SETTINGS\_SHARED\_PREFERENCES, MODE\_PRIVATE);

switch\_status = sharedPreferences.getBoolean(Constants.location\_tracker\_state,false);

updateViews();

}

public void updateViews()

{

notification\_switch.setChecked(switch\_status);

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data){

super.onActivityResult(requestCode, resultCode, data);

switch (requestCode){

case Constants.LOCATION\_CHECK\_CODE:

// user input from the dialogbox showed after checkLocation()

if(Activity.RESULT\_OK == resultCode){

// user picked yes

Log.d(LogTags.Location\_TAG, "onActivityResult: user picked yes. starting background location tracker");

startTrackerService();

// save settings preferences

save\_preferences(true);

// set LocationFetch boolean

LocationFetch.isLocationEnabled = true;

//set the settings switch UI to true

notification\_switch.setChecked(true);

}

else if(Activity.RESULT\_CANCELED == resultCode){

// user picked no

Log.d(LogTags.Location\_TAG, "onActivityResult: user picked no. setting boolean and preference to false");

save\_preferences(false);

LocationFetch.isLocationEnabled = false;

}

break;

}

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

//resolve unresolved permissions

switch (requestCode){

case Constants.PERMISSION\_CODE:

try {

this.permissions.resolvePermissions(permissions, grantResults);

}catch (Exception e){

Log.d(LogTags.Permissions\_TAG, "onRequestPermissionsResult: "+e.getMessage());

}

break;

     }

    }

}