**package** com.example.stuart.drivercoach;  
  
**import** android.Manifest;  
**import** android.app.Activity;  
**import** android.app.ProgressDialog;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.location.LocationListener;  
**import** android.location.LocationManager;  
**import** android.net.Uri;  
**import** android.os.AsyncTask;  
**import** android.os.Bundle;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.appindexing.Action;  
**import** com.google.android.gms.appindexing.AppIndex;  
**import** com.google.android.gms.appindexing.Thing;  
**import** com.google.android.gms.common.api.GoogleApiClient;  
  
**import** java.sql.Connection;  
**import** java.sql.Statement;  
  
*/\*\*  
 \* Created by Stuart on 22/03/2019.  
 \*/***public class** Location **extends** Activity **implements** LocationListener {  
  
 ProgressDialog **progressDialog**;  
 ConnectionClass **connectionClass**;  
 Button **finish**;  
 **private long curTime**;  
 **private long lastUpdate** = 0;  
 **private float lastSpeed** = 0;  
 **private float counter** = 0;  
  
 **private float distance**;  
 **private float acceleration**;  
  
 **private float TotalSpeed**;  
 **private float TotalDistance**;  
 **private float TotalAcceleration**;  
 **private float TotalTime**;  
  
 **private float AvgSpeed**;  
 **private float AvgAcceleration**;  
 **private float speed**;  
 **int gpsEnabled** = 0;  
  
  
 */\*\*  
 \* ATTENTION: This was auto-generated to implement the App Indexing API.  
 \* See https://g.co/AppIndexing/AndroidStudio for more information.  
 \*/* **private** GoogleApiClient **client**;  
 LocationManager **lm**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***location***);  
  
  
 **client** = **new** GoogleApiClient.Builder(**this**).addApi(AppIndex.***API***).build();  
 **connectionClass** = **new** ConnectionClass();  
  
 **progressDialog** = **new** ProgressDialog(**this**);  
 **finish** = (Button) findViewById(R.id.***finish***);  
  
 **finish**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
  
  
 Location.Dofinish dofinish = **new** Location.Dofinish();  
 dofinish.execute(Float.*toString*(**AvgSpeed**), *//takes the average values of the accelerometer to the database.* Float.*toString*(**AvgAcceleration**),  
 Float.*toString*(**TotalDistance**),  
 Float.*toString*(**counter**),  
 Float.*toString*(**TotalTime**));  
  
 }  
 });  
  
 **gpsEnabled** = 1;  
  
  
 **lm** = (LocationManager) **this**.getSystemService(Context.***LOCATION\_SERVICE***);  
  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***ACCESS\_COARSE\_LOCATION***,  
 Manifest.permission.***ACCESS\_FINE\_LOCATION***, Manifest.permission.***INTERNET***}  
 , 10);  
 Log.*i*(**"GPS"**, **"Permissions error for GPS"**);  
 **return**;  
 }  
 **lm**.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, **this**);  
 Log.*i*(**"GPS"**, **"Permissions ok for GPS"**);  
 **this**.onLocationChanged(**null**);  
  
 }  
  
  
 **public void** onRequestPermissionsResult(**int** requestCode,  
 String[] permissions,  
 **int**[] grantResults) {  
  
 **lm**.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, **this**);  
  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
  
 } **else** {  
 Log.*i*(**"GPS"**, **"Permissions ok for GPS"**);  
 **this**.onLocationChanged(**null**);  
  
 *// ATTENTION: This was auto-generated to implement the App Indexing API.  
 // See https://g.co/AppIndexing/AndroidStudio for more information.* }  
 }  
  
 @Override  
 **public void** onLocationChanged(android.location.Location location) {  
 TextView txt = (TextView) **this**.findViewById(R.id.***textView***);  
  
  
 **if** (location == **null**) {  
 txt.setText(**"-.- m/s"**);  
 } **else** {  
  
  
 **speed** = location.getSpeed();  
  
  
 */\* we need to make sure we only sample a subset of the data  
 we get from the device's accelerometer. We store the system's  
 current time (in milliseconds) store it in curTime and check  
 whether more than 100 milliseconds have passed since the last  
 time onSensorChanged was invoked.  
 \* \*/* **curTime** = System.*currentTimeMillis*();  
  
  
 **if** ((**curTime** - **lastUpdate**) > 100)  
  
 {  
 **counter** = **counter** + 1;  
 **TotalTime** = (**TotalTime** + 100)/1000;  
 **lastUpdate** = **curTime**;  
 **distance** = **speed** / 10;  
 **acceleration** = (**speed** - **lastSpeed**) \* 10;  
 **lastSpeed** = **speed**;  
  
  
  
 **TotalSpeed** = **TotalSpeed** + **speed**;  
 **TotalDistance** = **TotalDistance** + **distance**;  
 **TotalAcceleration** = **TotalAcceleration** + **acceleration**;  
  
  
 **AvgAcceleration** = **TotalAcceleration** / **counter**; *//Counter determines the number of times a a timestamp was recorded.* **AvgSpeed** = **TotalSpeed** / **counter**;  
 }  
  
 *//StringBuilder sb = new StringBuilder().append("Speed:").append(speed).append("\n");  
 //txt.setText(sb.toString());* }  
  
 StringBuilder sb = **new** StringBuilder().append(**"Speed:"**).append(**speed**).append(**"\n"**);  
 sb.append(**"Acceleration:"**).append(**acceleration**).append(**"\n"**);  
 sb.append(**"Distance:"**).append(**TotalDistance**).append(**"\n"**);  
 *//sb.append("Time:").append(curTime/1000).append("\n");* txt.setText(sb.toString());  
  
  
  
 */\* Intent intent = new Intent(Location.this, Accelerometer.class); //This allows you to carry over values from one activity to another.  
  
 intent.putExtra("speed", speed);  
  
 startActivity(intent);  
 \*/* }  
  
  
  
  
 @Override  
 **public void** onStatusChanged (String s,**int** i, Bundle bundle){  
  
 }  
  
 @Override  
 **public void** onProviderEnabled (String s){  
  
 }  
  
 @Override  
 **public void** onProviderDisabled (String s){  
  
 }  
  
  
 */\*\*  
 \* ATTENTION: This was auto-generated to implement the App Indexing API.  
 \* See https://g.co/AppIndexing/AndroidStudio for more information.  
 \*/* **public** Action getIndexApiAction() {  
 Thing object = **new** Thing.Builder()  
 .setName(**"Location Page"**) *//* ***TODO: Define a title for the content shown.*** *//* ***TODO: Make sure this auto-generated URL is correct.*** .setUrl(Uri.*parse*(**"http://[ENTER-YOUR-URL-HERE]"**))  
 .build();  
 **return new** Action.Builder(Action.***TYPE\_VIEW***)  
 .setObject(object)  
 .setActionStatus(Action.***STATUS\_TYPE\_COMPLETED***)  
 .build();  
 }  
  
 @Override  
 **public void** onStart() {  
 **super**.onStart();  
  
 *// ATTENTION: This was auto-generated to implement the App Indexing API.  
 // See https://g.co/AppIndexing/AndroidStudio for more information.* **client**.connect();  
 AppIndex.***AppIndexApi***.start(**client**, getIndexApiAction());  
 }  
  
 @Override  
 **public void** onStop() {  
 **super**.onStop();  
  
 *// ATTENTION: This was auto-generated to implement the App Indexing API.  
 // See https://g.co/AppIndexing/AndroidStudio for more information.* AppIndex.***AppIndexApi***.end(**client**, getIndexApiAction());  
 **client**.disconnect();  
 }  
  
 **public class** Dofinish **extends** AsyncTask<String, String, String> {  
  
  
 String **z** = **""**;  
  
  
 **boolean isSuccess** = **false**;  
  
 @Override  
 **protected void** onPreExecute() {  
 **progressDialog**.setMessage(**"Loading..."**);  
 **progressDialog**.show();  
 }  
  
 @Override  
 **protected** String doInBackground(String... params) {  
  
  
 **try** {  
 Connection con = **connectionClass**.CONN();  
 **if** (con == **null**) {  
 **z** = **"Upload Failed: Please check your internet connection"**;  
 } **else** {  
  
 *//String query = "insert into trips values('" + xstr + "','" + ystr + "','" + zstr + "','" + anglestr + "','" + speedstr + "')";* String query = **"insert into trips values(NULL,'"** + params[0] + **"','"** + params[1] + **"','"** + params[2] + **"','"** + params[3] + **"','"** + params[4] + **"')"**;  
  
 Statement stmt = con.createStatement();  
 stmt.executeUpdate(query);  
  
 **z** = **"Trip Finished: Upload Complete"**;  
 **isSuccess** = **true**;  
  
  
 }  
 } **catch** (Exception ex) {  
 **isSuccess** = **false**;  
 **z** = **"Exceptions"** + ex;  
 }  
  
 **return z**;  
 }  
  
  
 @Override  
 **protected void** onPostExecute(String s) {  
  
 Toast.*makeText*(getBaseContext(), **""** + **z**, Toast.***LENGTH\_LONG***).show();  
  
  
 **if** (**isSuccess**) {  
  
 Intent intent = **new** Intent(Location.**this**, TrackRecord.**class**);  
  
 startActivity(intent);  
 }  
  
  
 **progressDialog**.hide();  
  
 }  
  
 }  
  
 }  
*//}*