

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГБОУ «ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Промежуточный отчет о научно-исследовательской работе

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ПЕРСОНАЛИЗИРОВАННЫЙ  
ТРЕКЕР ПОЛЬЗОВАТЕЛЯ

Выполнил:

студент 2 курса группы 22207 В. В. Клименко

---

*подпись*

Научный руководитель:

к.ф.-м.н., преподаватель В. М. Димитров

Оценка руководителя:

---

*подпись*

Представлен на кафедру

« \_\_\_\_ » \_\_\_\_\_ 2018 г.

---

*подпись принявшего работу*

# Содержание

<b>Введение</b>	<b>2</b>
<b>1 Разработка тестового приложения</b>	<b>2</b>
1.1 Java-код приложения . . . . .	2
1.2 XML-разметка приложения . . . . .	8
<b>2 Постановка задачи</b>	<b>9</b>
2.1 Создание трекера пользователя . . . . .	9
<b>3 Текущие результаты</b>	<b>11</b>
<b>Библиографический список использованной литературы</b>	<b>11</b>

# Введение

Трекер - это программа позволяющая отслеживать путь пользователя и выводить различную информацию о том, каким образом он перемещался.

Сегодня такое приложение необходимо тем, кто занимается туризмом и спортом. Ведь это очень удобно, чтобы человек имел статистику о том, с какой скоростью, где и сколько он прошёл. Однако пользователь вынужден включать и выключать запись своих передвижений, что уменьшает удобность использования. Целью данной работы является создание трекера, который работал бы в фоновом режиме, то есть постоянно вёл запись перемещений без участия пользователя.

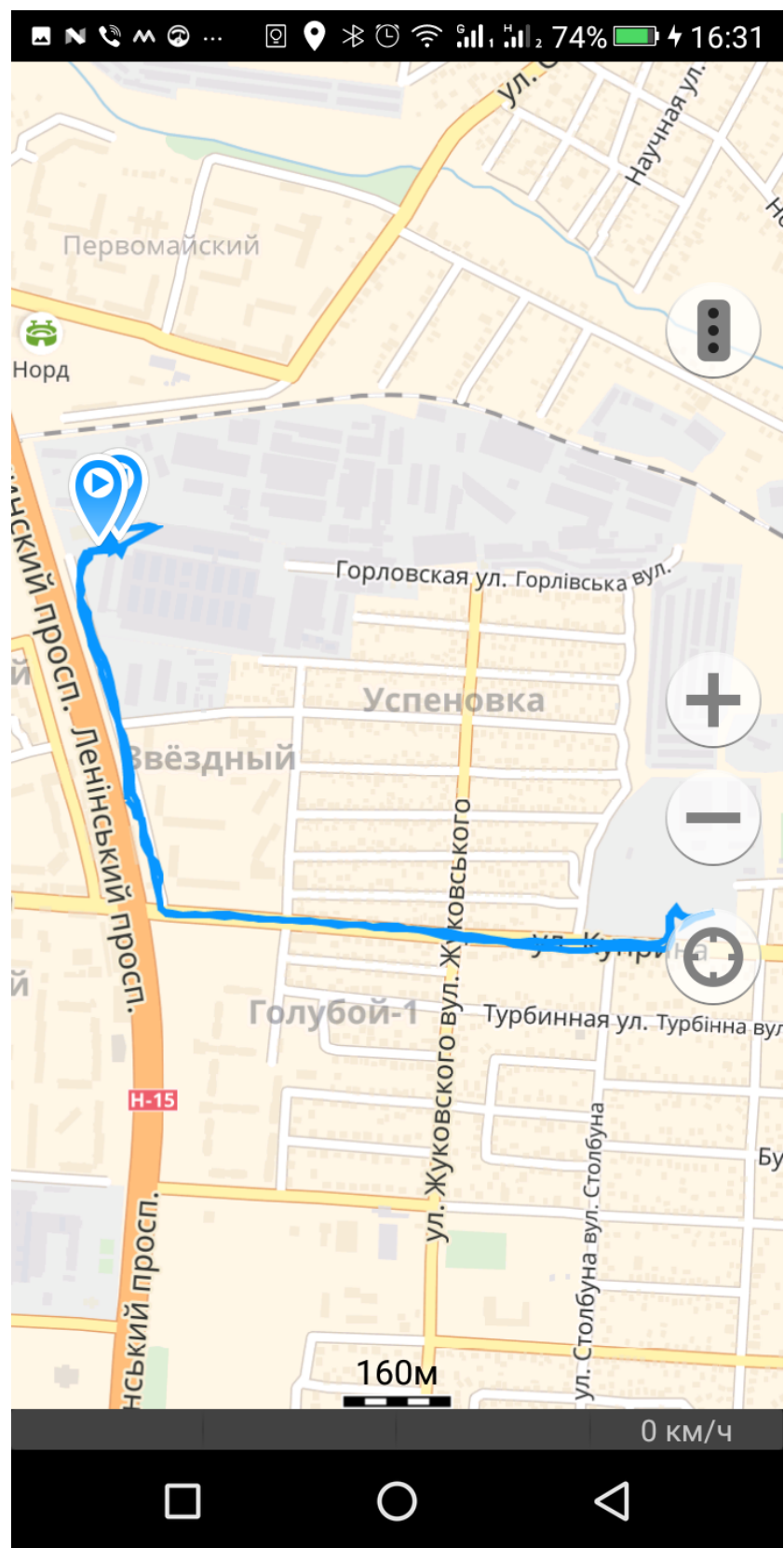


Рис. 1: Пример трекера

# 1 Разработка тестового приложения

## 1.1 Java-код приложения

Ниже приведён java-код приложения.

---

```
package com.beginerdranch.android.myapplication;

import android.Manifest;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.location.Location;
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 com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;

import java.text.DateFormat;
import java.util.Date;

public class LocationActivity extends Activity implements
    LocationListener,
    GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener {

    private static final String TAG = "LocationActivity";
    private static final long INTERVAL = 1000 * 10;
    private static final long FASTEST_INTERVAL = 1000 * 5;
    Button btnFusedLocation;
```

```

TextView tvLocation;
LocationRequest mLocationRequest;
GoogleApiClient mGoogleApiClient;
Location mCurrentLocation;
String mLastUpdateTime;

protected void createLocationRequest() {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(INTERVAL);
    mLocationRequest.setFastestInterval(FATEST_INTERVAL);
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
;
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //show error dialog if GoolglePlayServices not available
    if (!isGooglePlayServicesAvailable()) {
        finish();
    }
    createLocationRequest();
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addApi(LocationServices.API)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .build();

    setContentView(R.layout.activity_location);
    tvLocation = (TextView) findViewById(R.id.tvLocation);

    btnFusedLocation = (Button) findViewById(R.id.btnShowLocation);
    btnFusedLocation.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            updateUI();
        }
    });
}

```

```

@Override
public void onStart() {
    super.onStart();
    mGoogleApiClient.connect();
}

@Override
public void onStop() {
    super.onStop();
    mGoogleApiClient.disconnect();
}

private boolean isGooglePlayServicesAvailable() {
    int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(
this);
    if (ConnectionResult.SUCCESS == status) {
        return true;
    } else {
        GooglePlayServicesUtil.getErrorDialog(status, this, 0).show();
        return false;
    }
}

@Override
public void onConnected(Bundle bundle) {
    startLocationUpdates();
}

protected void startLocationUpdates() {
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.
PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.
PERMISSION_GRANTED) {
        // TODO: Consider calling
        //     ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //     public void onRequestPermissionsResult(int requestCode, String
[] permissions,

```

```

        //                                                    int[] grantResults)
        // to handle the case where the user grants the permission. See the
documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }

    PendingResult<Status> pendingResult = LocationServices.
FusedLocationApi.requestLocationUpdates(
    mGoogleApiClient, mLocationRequest, this);
}

@Override
public void onConnectionSuspended(int i) {

}

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {

}

@Override
public void onLocationChanged(Location location) {
    mCurrentLocation = location;
    mLastUpdateTime = DateFormat.getTimeInstance().format(new Date());
    updateUI();
}

private void updateUI() {
    if (null != mCurrentLocation) {
        String lat = String.valueOf(mCurrentLocation.getLatitude());
        String lng = String.valueOf(mCurrentLocation.getLongitude());
        tvLocation.setText("At Time: " + mLastUpdateTime + "\n" +
            "Latitude: " + lat + "\n" +
            "Longitude: " + lng + "\n" +
            "Accuracy: " + mCurrentLocation.getAccuracy() + "\n" +
            "Provider: " + mCurrentLocation.getProvider());
    } else {
    }
}
}

```



```

@Override
protected void onPause() {
    super.onPause();
    stopLocationUpdates();
}

protected void stopLocationUpdates() {
    LocationServices.FusedLocationApi.removeLocationUpdates(
        mGoogleApiClient, this);
}

@Override
public void onResume() {
    super.onResume();
    if (mGoogleApiClient.isConnected()) {
        startLocationUpdates();
    }
}
}

```

---

Листинг 1: Основной код приложения из LocationActivity.java

## 1.2 XML-разметка приложения

Ниже приведена XML-разметка приложения.

---

```

<?xml version="1.0" encoding="utf-8"?>
    <android.support.constraint.ConstraintLayout xmlns:android="http://
schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".LocationActivity">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"

```

```

        android:layout_marginTop="45dp"
        android:text="@string/locationTxt"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btnShowLocation"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="270dp"
    android:background="#ffff1a7c"
    android:text="Show Location"
    android:textColor="#ffffffff"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvLocation" />

<TextView
    android:id="@+id/tvLocation"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginTop="18dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
</android.support.constraint.ConstraintLayout>

```

---

Листинг 2: Разметка приложения из activity\_location.xml

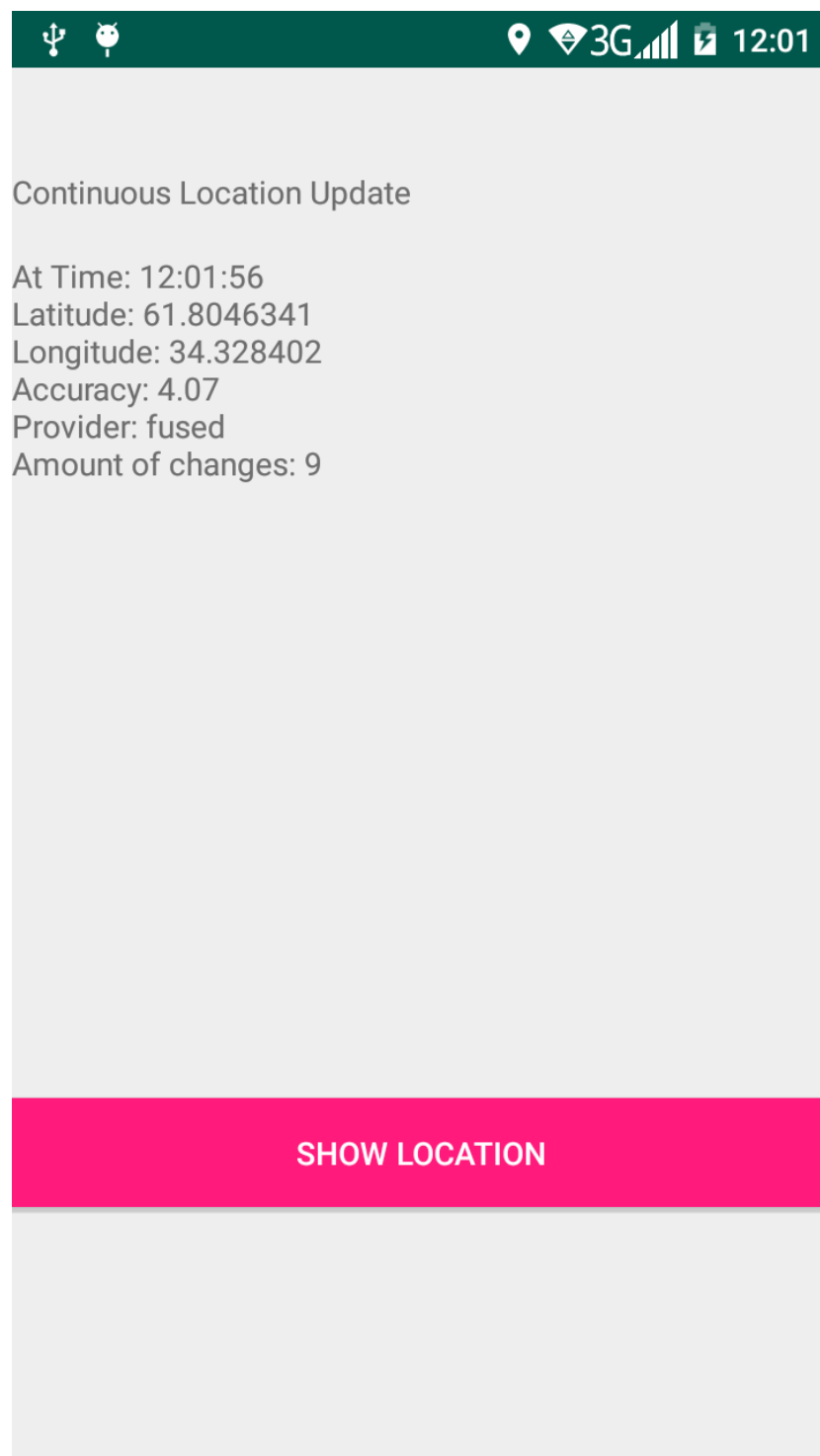


Рис. 2: Внешний вид тестового приложения при работе

## 2 Постановка задачи

### 2.1 Создание трекера пользователя

Такая серьёзная задача требует знаний и умений в области разработки приложений под платформу Android. Прежде всего требуется разработать тестовый вариант приложения, который позволит оценить расход батареи мобильного устройства. Оно постоянно работать и опрашивать местоположение пользователя. Статистика расхода батареи позволит сделать вывод о возможности создания энергоэффективного трекера.

Для достижения поставленной цели необходимо решить задачи:

1. Изучить основные принципы разработки мобильного приложения
2. Установить и настроить инструменты разработки под платформу Android
3. Изучить основные технологии создания приложений под платформу Android
4. Изучить технологии отслеживания местоположения
5. Изучить технологии работы приложения в фоновом режиме
6. Написать тестовое приложение
7. Написать основное приложение

### 3 Текущие результаты

На данный момент получены следующие результаты:

1. Изучены базовые принципы разработки мобильных приложений
2. Установлены и настроены инструменты разработки под Android
3. Изучены основные технологии разработки под платформу Android
4. Изучены технологии отслеживания местоположения
5. Написано тестовое приложение без работы в фоне

## Список литературы

1. Геотрекер - GPS трекер - Всё о пройденном пути [Электронный ресурс] : [сайт] - Электрон. дан. - Режим доступа:[http://helpix.ru/appinion/201806/2160-geotreker\\_-\\_gps\\_treker-vsjo\\_o\\_projdennom\\_puti.html](http://helpix.ru/appinion/201806/2160-geotreker_-_gps_treker-vsjo_o_projdennom_puti.html) - Загл. с экрана.
2. Android Developers [Электронный ресурс] : [сайт] - Электрон. дан. - Режим доступа:<https://developer.android.com/> - Загл. с экрана.
3. Филлипс Б., Стюарт К., Марсианко К. *Android программирование для профессионалов* М.: Питер. 2017 — 687 с.