

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

Введение	3
1 Обзор приложений в данной области	4
1.1 Spyzie	4
1.2 Жизнь 360	4
1.3 GPS Phone Tracker	5
1.4 Glympse	6
1.5 Где мой Droid	7
2 Алгоритм получение координат	8
3 Постановка задачи	9
4 Текущие результаты	10
5 Приложение	11
5.1 Java-код тестового приложения	11
5.2 XML-разметка тестового приложения	21
Библиографический список использованной литературы	24

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.

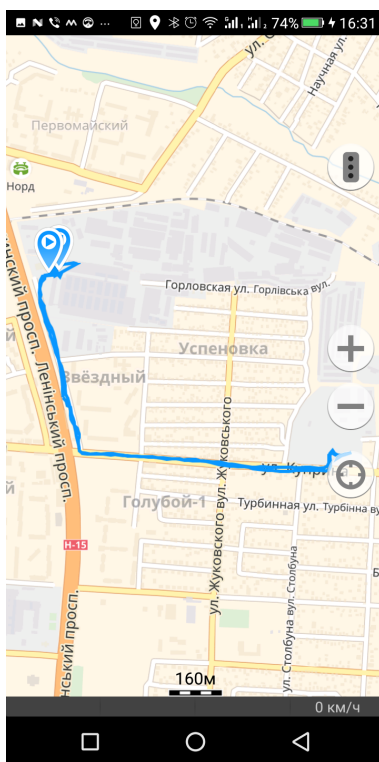


Рис. 1: Пример трекера

1 Обзор приложений в данной области

Для создания приложения необходимо проанализировать уже имеющиеся решения в данной области. Проанализировав основные приложения, можно понять, что надо обязательно реализовать для своего приложения, а что напротив надо устранить.

Проанализированные приложения хорошо выполняют свой функционал. Однако многие из них интенсивно расходуют заряд батареи мобильного устройства.

1.1 Spyzie

Spyzie является полным устройством слежения приложения, которое позволит вам получить доступ к существенной информации, относящейся к устройству в одном месте. Очень проста в использовании, имеет веб-интерфейс для пользователя панель управления, которая может быть доступна практически на любом устройстве. Приложение может быть использовано для получения обновления местоположения в режиме реального времени и позволяет доступ важных данных (например, сообщения, фотографии, заметки и многое другое) на устройстве[2].

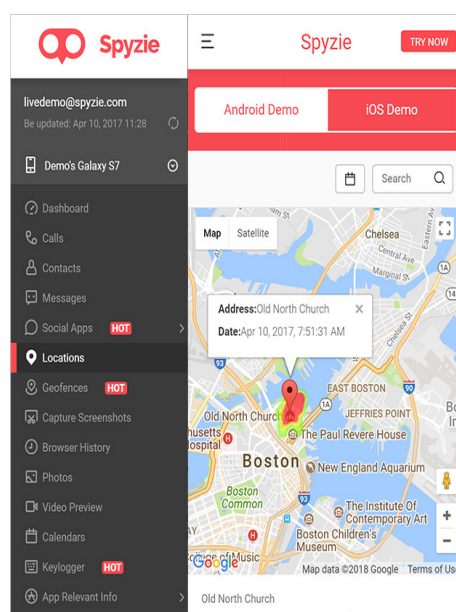


Рис. 2: Spyzie

1.2 Жизнь 360

Жизнь 360 является полной семьёй отслеживания приложения, которое поставляется с большим количеством дополнительных функций. Приложение позволяет легко добавить

круги для вашей семьи и друзей, чтобы знать свои последние места. С его помощью можно также можете получить журнал своих прошлых мест, а также. Это один из лучших Android приложений GPS трекер, который поставляется с дополнительной поддержкой вождения. Он может обнаружить сбой, отправить экстренное сообщение, и проанализировать ваш шаблон вождения[2].



Рис. 3: Жизнь 360

1.3 GPS Phone Tracker

GPS Phone Tracker является одним из старейших и наиболее широко используется следящий Android устройства приложений там. То, что делает его одним из лучших следящих приложений Android GPS являются его легко возможности подключения, точные результаты и бесшовные использования. Можно использовать приложение, чтобы получить точное местоположение своих друзей и семьи. Она также имеет функцию отслеживания устройства, чтобы получить обновление в режиме реального времени для потерянного телефона Android[2].

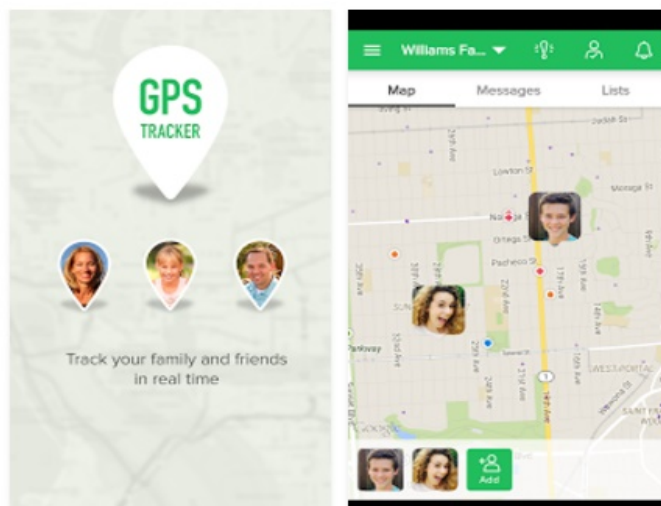


Рис. 4: GPS Phone Tracker

1.4 Glympse

Glympse является универсальным GPS трекером для Android, который будет отслеживать местонахождение без вторжения в вашу частную жизнь. Он может быть использован для отслеживания доставки, запросить местоположение ваших друзей или сообщить своей семье и коллегам о своём местонахождении. Приложение также может быть использовано для отслеживания устройства, чтобы защитить его от кражи[2].

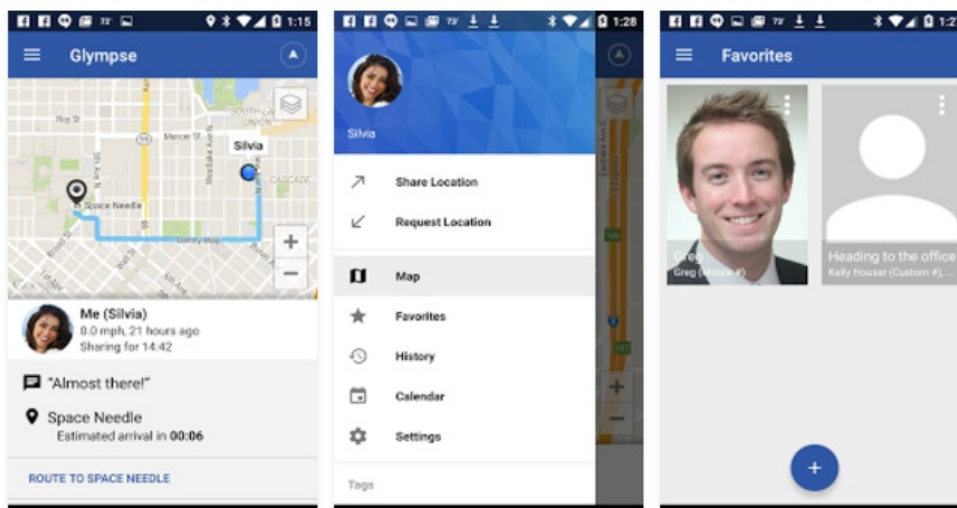


Рис. 5: Glympse

1.5 Где мой Droid

Данное приложение предоставляет надёжный способ, чтобы найти телефон удалённо. GPS трекер для Android уже используется миллионами людей во всем мире, чтобы получить в режиме реального времени и точное местоположение своего устройства. Он имеет отличную функцию защиты от кражи с пассивными глодаданных и гео-ограждением собственности[2].

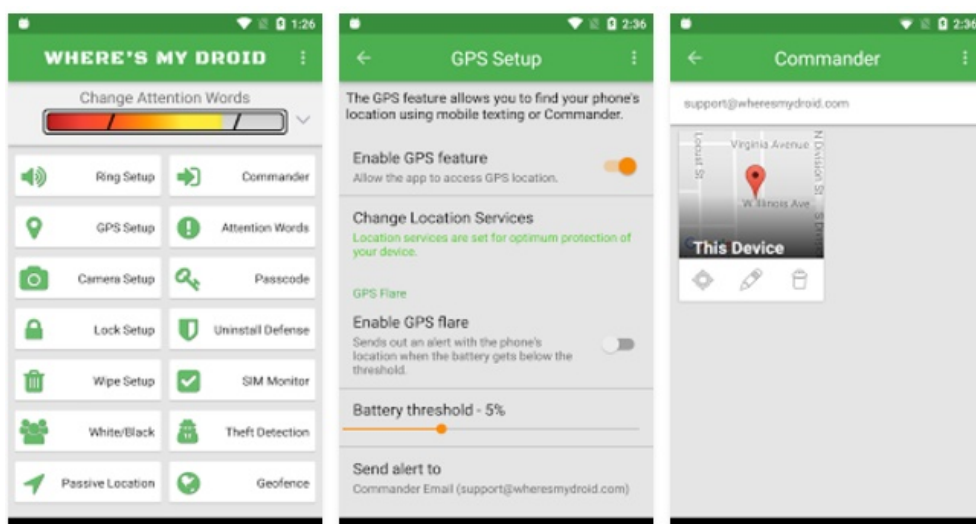


Рис. 6: Где мой Droid

2 Алгоритм получение координат

Приложение запрашивает у android-устройства разрешение на получение данных о местоположение пользователя. Если оно получено, то запускается опрос координат с определенным интервалом. С помощью обработчиков событий можно получить новые координаты, как только они изменились. Разработчик может выбирать частоту обновления и точность местоположения. Они должны быть такими, чтобы приложение было энергоэффективным, но при этом погрешность координат не была слишком большой. Данные о координатах устройство получает с помощью GPS-модуля или информации о подключении к Интернету[3].

3 Постановка задачи

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.

4 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.

5 Приложение

5.1 Java-код тестового приложения

```
1 package com.beginerdranch.android.myapplication;
2
3
4 import android.Manifest;
5 import android.app.Activity;
6 import android.content.pm.PackageManager;
7 import android.location.Location;
8 import android.os.Bundle;
9 import android.support.v4.app.ActivityCompat;
10 import android.util.Log;
11 import android.view.View;
12 import android.widget.Button;
13 import android.widget.TextView;
14
15 import com.google.android.gms.common.ConnectionResult;
16 import com.google.android.gms.common.GooglePlayServicesUtil;
17 import com.google.android.gms.common.api.GoogleApiClient;
18 import com.google.android.gms.common.api.PendingResult;
19 import com.google.android.gms.common.api.Status;
20 import com.google.android.gms.location.LocationListener;
21 import com.google.android.gms.location.LocationRequest;
22 import com.google.android.gms.location.LocationServices;
23
24 import java.text.DateFormat;
25 import java.util.Date;
26
27 public class LocationActivity extends Activity implements
28     LocationListener,
29     GoogleApiClient.ConnectionCallbacks,
30     GoogleApiClient.OnConnectionFailedListener {
31
32     private static final String TAG = "LocationActivity";
33     private static final long INTERVAL = 1000 * 10;
34     private static final long FASTEST_INTERVAL = 1000 * 5;
```

```

35     Button btnFusedLocation;
36     TextView tvLocation;
37     LocationRequest mLocationRequest;
38     GoogleApiClient mGoogleApiClient;
39     Location mCurrentLocation;
40     String mLastUpdateTime;
41
42     protected void createLocationRequest() {
43         mLocationRequest = new LocationRequest();
44         mLocationRequest.setInterval(INTERVAL);
45         mLocationRequest.setFastestInterval(FATEST_INTERVAL);
46         mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY
47             );
48     }
49
50     @Override
51     protected void onCreate(Bundle savedInstanceState) {
52         super.onCreate(savedInstanceState);
53         Log.d(TAG, "onCreate .....");
54         //show error dialog if GoolglePlayServices not available
55         if (!isGooglePlayServicesAvailable()) {
56             finish();
57         }
58
59         createLocationRequest();
60         mGoogleApiClient = new GoogleApiClient.Builder(this)
61             .addApi(LocationServices.API)
62             .addConnectionCallbacks(this)
63             .addOnConnectionFailedListener(this)
64             .build();
65
66         setContentView(R.layout.activity_location);
67         tvLocation = (TextView) findViewById(R.id.tvLocation);
68
69         btnFusedLocation = (Button) findViewById(R.id.btnShowLocation);
70         btnFusedLocation.setOnClickListener(new View.OnClickListener() {
71             @Override
72             public void onClick(View arg0) {

```

```

71         updateUI();
72     }
73 });
74
75 }
76
77 @Override
78 public void onStart() {
79     super.onStart();
80     Log.d(TAG, "onStart fired .....");
81     mGoogleApiClient.connect();
82 }
83
84 @Override
85 public void onStop() {
86     super.onStop();
87     Log.d(TAG, "onStop fired .....");
88     mGoogleApiClient.disconnect();
89     Log.d(TAG, "isConnected .....: " + mGoogleApiClient.
        isConnected());
90 }
91
92 private boolean isGooglePlayServicesAvailable() {
93     int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(
        this);
94     if (ConnectionResult.SUCCESS == status) {
95         return true;
96     } else {
97         GooglePlayServicesUtil.getErrorDialog(status, this, 0).show();
98         return false;
99     }
100 }
101
102 @Override
103 public void onConnected(Bundle bundle) {
104     Log.d(TAG, "onConnected - isConnected .....: " +
        mGoogleApiClient.isConnected());

```

```

105     startLocationUpdates();
106 }
107
108 protected void startLocationUpdates() {
109     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.
110         ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
111         && ActivityCompat.checkSelfPermission(this, Manifest.permission
112         .ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
113         // TODO: Consider calling
114         //     ActivityCompat#requestPermissions
115         // here to request the missing permissions, and then overriding
116         // public void onRequestPermissionsResult(int requestCode,
117         //     String[] permissions,
118         //                                     int[] grantResults)
119         // to handle the case where the user grants the permission. See
120         // the documentation
121         // for ActivityCompat#requestPermissions for more details.
122         return;
123     }
124     PendingResult<Status> pendingResult = LocationServices.
125         FusedLocationApi.requestLocationUpdates(
126         mGoogleApiClient, mLocationRequest, this);
127     Log.d(TAG, "Location update started .....: ");
128 }
129
130 @Override
131 public void onConnectionSuspended(int i) {
132 }
133
134 @Override
135 public void onConnectionFailed(ConnectionResult connectionResult) {
136     Log.d(TAG, "Connection failed: " + connectionResult.toString());
137 }
138
139 @Override
140 public void onLocationChanged(Location location) {

```

```

136     Log.d(TAG, "Firing onLocationChanged
        .....");
137     mCurrentLocation = location;
138     mLastUpdateTime = DateFormat.getInstance().format(new Date());
139     updateUI();
140 }
141
142 private void updateUI() {
143     Log.d(TAG, "UI update initiated .....");
144     if (null != mCurrentLocation) {
145         String lat = String.valueOf(mCurrentLocation.getLatitude());
146         String lng = String.valueOf(mCurrentLocation.getLongitude());
147         tvLocation.setText("At Time: " + mLastUpdateTime + "\n" +
148             "Latitude: " + lat + "\n" +
149             "Longitude: " + lng + "\n" +
150             "Accuracy: " + mCurrentLocation.getAccuracy() + "\n" +
151             "Provider: " + mCurrentLocation.getProvider());
152     } else {
153         Log.d(TAG, "location is null .....");
154     }
155 }
156
157 @Override
158 protected void onPause() {
159     super.onPause();
160     stopLocationUpdates();
161 }
162
163 protected void stopLocationUpdates() {
164     LocationServices.FusedLocationApi.removeLocationUpdates(
165         mGoogleApiClient, this);
166     Log.d(TAG, "Location update stopped .....");
167 }
168
169 @Override
170 public void onResume() {
171     super.onResume();

```

```

172     if (mGoogleApiClient.isConnected()) {
173         startLocationUpdates();
174         Log.d(TAG, "Location update resumed .....");
175     }
176 }
177 }

```

5.2 XML-разметка тестового приложения

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://
    schemas.android.com/apk/res/android"
3  xmlns:app="http://schemas.android.com/apk/res-auto"
4  xmlns:tools="http://schemas.android.com/tools"
5  android:layout_width="match_parent"
6  android:layout_height="match_parent"
7  tools:context=".LocationActivity">
8
9  <TextView
10     android:id="@+id/textView"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:layout_centerHorizontal="true"
14     android:layout_marginTop="45dp"
15     android:text="@string/locationTxt"
16     app:layout_constraintStart_toStartOf="parent"
17     app:layout_constraintTop_toTopOf="parent" />
18
19  <Button
20     android:id="@+id/btnShowLocation"
21     style="?android:attr/buttonStyleSmall"
22     android:layout_width="fill_parent"
23     android:layout_height="wrap_content"
24     android:layout_below="@+id/textView"
25     android:layout_centerHorizontal="true"
26     android:layout_marginTop="270dp"
27     android:background="#ffff1a7c"
28     android:text="Show Location"

```



```
29     android:textColor="#ffffffff"
30     app:layout_constraintStart_toStartOf="parent"
31     app:layout_constraintTop_toBottomOf="@+id/tvLocation" />
32
33 <TextView
34     android:id="@+id/tvLocation"
35     android:layout_width="fill_parent"
36     android:layout_height="wrap_content"
37     android:layout_alignParentBottom="true"
38     android:layout_marginTop="18dp"
39     app:layout_constraintStart_toStartOf="parent"
40     app:layout_constraintTop_toBottomOf="@+id/textView" />
41 </android.support.constraint.ConstraintLayout>
```

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 - Загл. с экрана.
2. 5 лучших GPS приложения Tracker для Android в 2018 году [Электронный ресурс] : [сайт] - Электрон. дан. - Режим доступа:<http://global.spyzie.biz/ru/android-tracker/best-gps-tracker-apps-for-android.html> - Загл. с экрана.
3. Android Developers [Электронный ресурс] : [сайт] - Электрон. дан. - Режим доступа:<https://developer.android.com/> - Загл. с экрана.
4. Google Play [Электронный ресурс] : [сайт] - Электрон. дан. - Режим доступа:<https://play.google.com> - Загл. с экрана.
5. Филлипс Б., Стюарт К., Марсианко К. *Android программирование для профессионалов* М.: Питер. 2017 — 687 с.