

Project report within the scope of the course 188.939 *Bachelorarbeit für Informatik und Wirtschaftsinformatik* submitted to the Interactive Media Systems Group,
Vienna University of Technology.

GPS Child Finder

Android Application

Daniela Blum

Matr.-No.: 1025570

Program code: 033 532

Supervisor: Ao.Univ.Prof. Mag. Dr. Horst Eidenberger

October 3, 2014

Table of Contents

1	Project Definition.....	3
1.1	Task.....	3
1.2	Use Cases.....	3
1.2.1	Start application (StartingActivity.class).....	3
1.2.2	Required services dialog.....	4
1.2.3	Show map (MainActivity.class, Map.class).....	5
1.2.4	Show settings (SettingsActivity.class).....	7
1.2.5	Show help (HelpActivity.class).....	9
1.3	Time Schedule	9
2	Implementation	11
2.1	System Environment	11
2.1.1	GPS-Bluetooth-Device	11
2.2	Libraries	12
2.3	Implementation Features.....	12
2.3.1	Calculation of the maps bounds.....	12
2.3.2	Get GPS coordinates from the GPS-Bluetooth-device	12
2.4	Known Bugs and Limitations	13
2.4.1	Devices.....	13
2.4.2	GPS-Bluetooth-device.....	13
2.4.3	Bluetooth range.....	13
2.4.4	Compass mode sensitivity.....	13
2.4.5	Precision	13
2.5	Potential and Further Development	13
3	Availability.....	14
3.1	Source Code.....	14
3.2	Documentation.....	14
3.3	Languages	14
4	References.....	14

1 Project Definition

1.1 Task

The task of this project is to develop an application for Android mobile phones: "GPS Child Finder". The application should show the location of the users child on a map. It comprises a GPS-based location device that sends the GPS coordinates of its current location via Bluetooth to the mobile phone. This device should be carried by the child. The application now shows the location and the direction of the users child (based on the current orientation of the device) and the distance between its location and the location of the user.

The application is intended to be used on large playgrounds or something like that, to answer the recurring question: "*Where is my child?*"

1.2 Use Cases

The following passages are describing all possible use cases that can occur when the user interacts with the application.

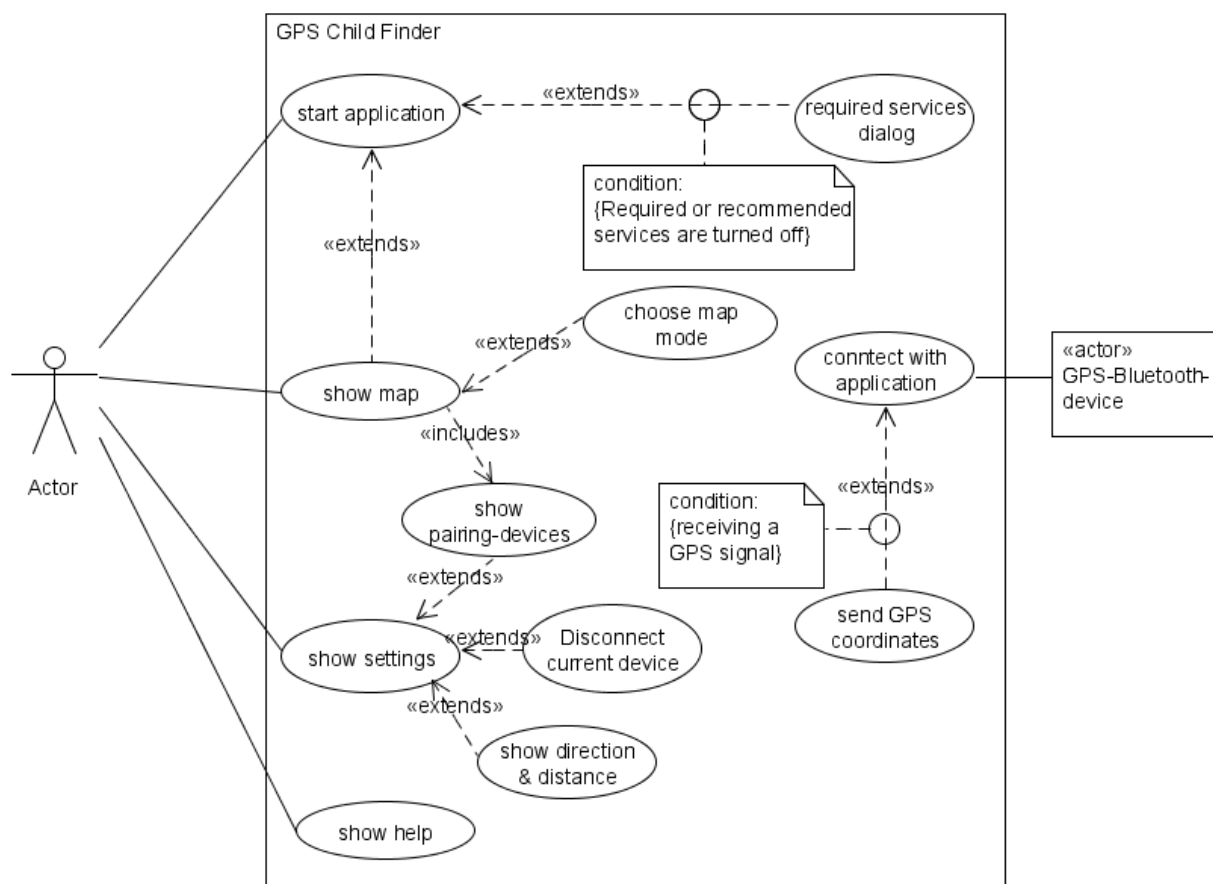


Figure 1: Use case diagram

1.2.1 Start application (StartingActivity.class)

When the application is started, the first thing the application does is to check if the required/recommended services are turn on or off. If one of them is turned off, the application shows the required services dialog.

1.2.2 Required services dialog

This dialog is shown when the application is started. But only if one of the required or recommended services is turn off. It's possible to go to the phones settings menu (to turn the services on) or to cancel the dialog. Canceling while one of the required services isn't activated exits the application. That's because the GPS Child Finder serves no purpose without them.

The following services are required to run the application (Figure 2):

- *Bluetooth services*
are needed because a GPS-Bluetooth-device (which shows the location of the users child on the map) should be connected to the phone.
- *Internet connection*
is necessary to display the embedded Google Map.
- *Location access*
is required to gain information about the current location of the user and show it on the map.

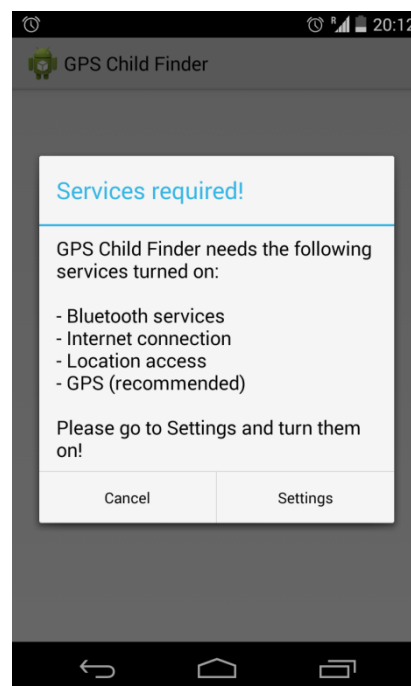


Figure 2: Required services – dialog

The *Google Play Services API* [1] must be installed on the phone to run specific features of the application properly. The application shows the right information when the API isn't available.

Moreover, *GPS* is not required but recommended to get a better precision concerning the location of the mobile phone. The user has the option to ignore the suggestion, if only this one isn't turned on (Figure 3).

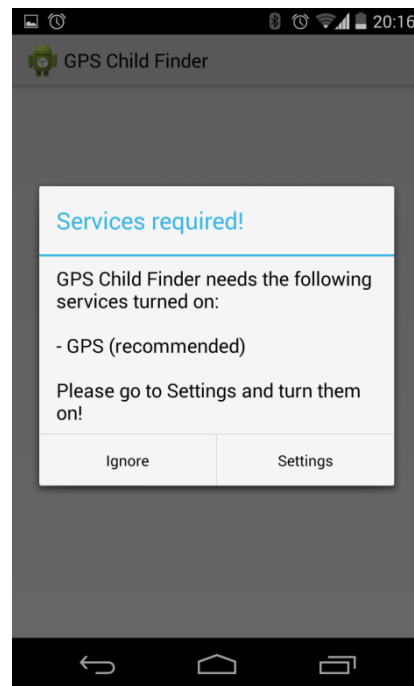


Figure 3: Recommended Service - dialog with ignore button

1.2.3 Show map (MainActivity.class, Map.class)

Only after the required and – optional – the recommended services are turned on, the main screen is displayed. It contains of an action bar with several menu items and an embedded Google Map with a zooming function (zooming can also be done via touchscreen) and two buttons, which are connected to the location icons on the map (Figure 4). If no GPS-Bluetooth-device is connected with the application yet, the dialog with the Bluetooth pairing-devices is displayed in the foreground. This dialog is explained later.

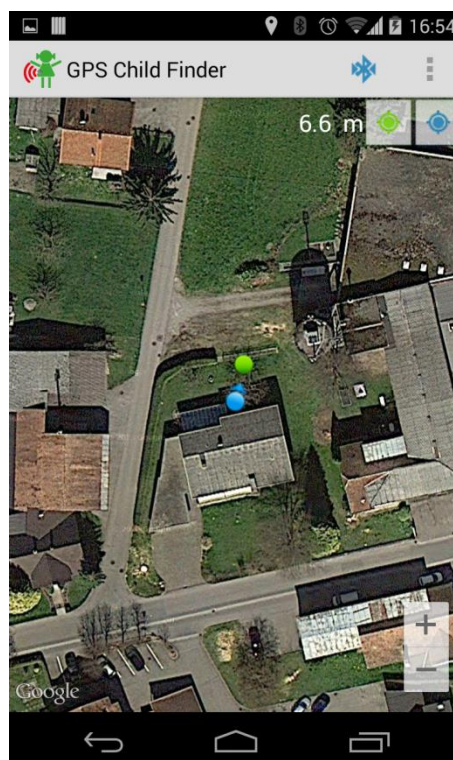


Figure 4: Main screen with the embedded Google Maps

Action bar

The action bar contains the menu. Switching between map modes (satellite, normal and terrain mode), going to the settings menu or calling the help function can be done here (Figure 5).

It also includes a Bluetooth icon on the left of the drop-down menu, which turns blue if a Bluetooth-device is connected. Otherwise, it's colored grey.

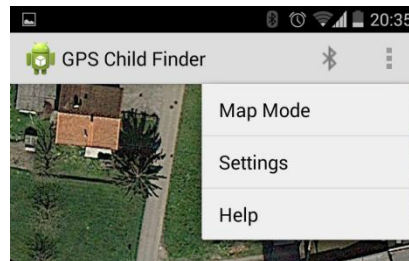


Figure 5: Action bar and manu on the main screen

If the user has a phone with a menu-button, the drop-down-menu (the three dots) is not shown on the action bar. Consequently, the menu items can be accessed over the menu button.

Buttons and Icons

- **My location (Where am I?)**



The blue pin on the map shows where the user/the Android mobile phone is on the map. The pin is only displayed when the location information can be obtained.



The blue button, which is located on the upper right corner of the map, is connected with the blue pin on the map – therefore with the location of the user. The user should press this button to find/focus his current location. It also triggers the compass mode (meaning that the view is oriented in the direction in which the user is facing).

A disabled button means that the app is unable to gain location information. Reasons for that could be that the location access or the internet was turned off or the Google Play Services were uninstalled.

- **Device location (Where is my child?)**



The green pin on the map shows where the child/the GPS-Bluetooth-device is located. The pin is only displayed when a device is connected and is able to get a GPS Signal.



The green button, which is located on the left of the blue button in the upper right corner of the map, is connected with the green pin on the map. The user should press this button to find/focus the location of his child.

A disabled button indicates that the app is unable to gain location information from the device. That means either that no Bluetooth-device is connected or that the device can't get any GPS signal by now.

Direction and Distance

- **Direction**



The little arrow on the blue pin shows the direction of the users child or rather the location of the GPS-Bluetooth-device on the map. The conditions are that a device should be connected and functioning.

- **Distance**

1.0 m On the buttons left it displays the direction in meters to the child who carries device. But only if the device is connected and functioning.

Choose map mode

The menu drop-down also contains an item which the user can choose his desired map mode. It can be chosen between three different types: Satellite, normal and terrain mode.

1.2.4 Show settings (SettingsActivity.class)

The user can switch from the main screen to the settings menu, where he can perform various settings according to the application (Figure 6).

The menu offers two main categories: “Bluetooth” and “Other” with certain sub items which are explained hereafter. Besides, a question mark is shown at the upper right corner to forward to the help screen.

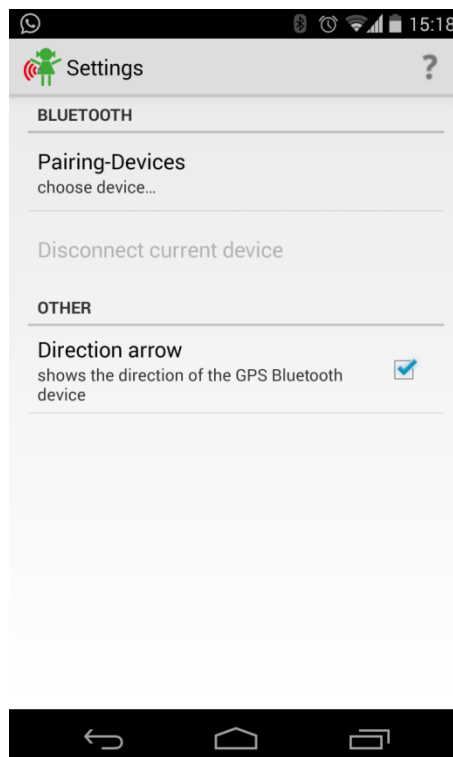


Figure 6: Settings menu

Show/Choose pairing-devices

The first one is “**Pairing-Devices**” to show the list of the currently – with the mobile phone paired – devices (Figure 7). The user can choose one of them to connect with the application. GPS Child Finder doesn’t make any difference between the types of the devices and shows all of them in a list view,

because there is no specific type for a GPS transferring device. However, only the ones which support and send GPS input to the Android phone can be connected.

The user can add a new device if the GPS-Bluetooth-device for his child isn't paired with the phone at this time. The "Add new device" button redirects directly to the phones Bluetooth settings menu.

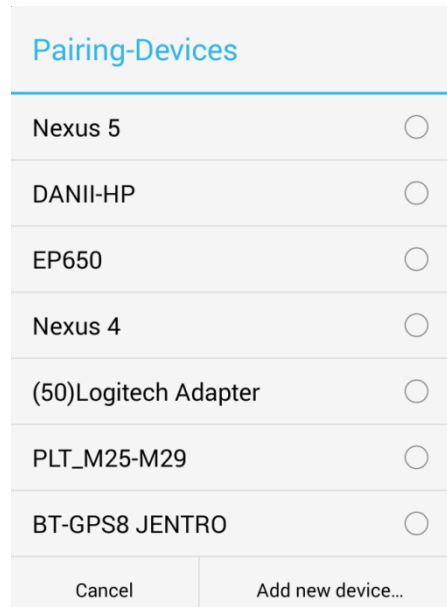


Figure 7: List of pairing-devices

If the user selects a device a progress dialog is shown with further instructions (Figure 8). If the connection was successful, the Bluetooth icon on the main menu should be shaded blue and the picked device should be displayed under the "Paired-Devices" point in the settings menu. Now the application waits for the GPS-Bluetooth-device to send the GPS coordinates it's receiving, so that the green icon can be shown on the map. If no connection can be set up, the application notifies the user with a message – a toast with information and further instructions.

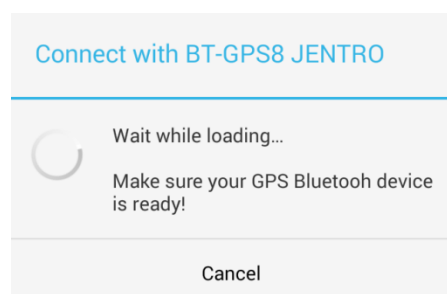


Figure 8: Progress dialog when the phone tries to connect with a Bluetooth-device

Disconnect current device

"Disconnecting current device", which is currently connected with the application, can also be done here. It's depending on the connection status: If a device is connected, the option is enabled; otherwise it's disabled and shaded grey.

Show direction and distance

The last point is the checkbox **"Direction arrow"**, where the user is able to activate or disable the direction arrow on the blue pin. Furthermore, the distance is shown with it or not.

1.2.5 Show help (HelpActivity.class)

The help screen can be accessed from the main screen – in the menu – or from the settings menu by pressing the question mark icon. It should help the user to have a short overview about the GPS Child Finder (Figure 9).

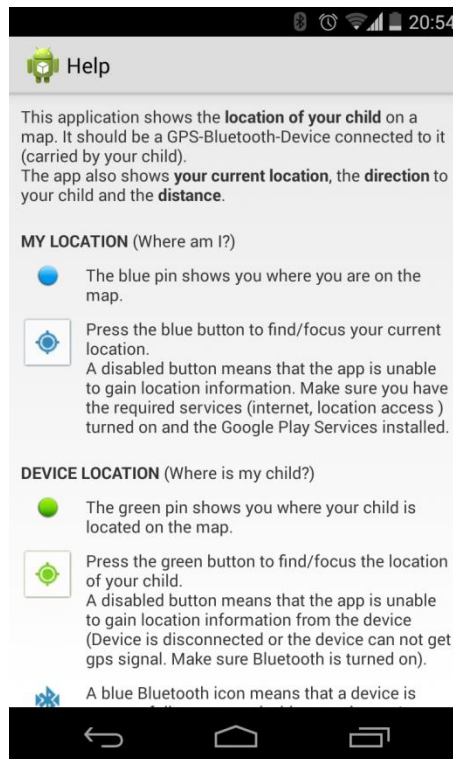


Figure 9: Help screen

1.3 Time Schedule

<i>Start date</i>	<i>Task</i>	<i>Time (in h)</i>
01.04.2014	Research (GPS-Bluetooth-device)	2
02.04.2014	Research (GPS-Bluetooth-device)	3
03.04.2014	Project planning	0,5
02.05.2014	Project setup (project hosting)	0,5
04.05.2014	Testing	0,5
05.05.2014	Project setup (Eclipse, Android SDK, test device, SVN)	11
05.05.2014	Research (Android Workspace)	2
07.05.2014	Project planning	2,5
13.05.2014	Programming (action bar, buttons, settings menu)	2
14.05.2014	Project setup (changing project hosting: bitbucket.org, git)	1
14.05.2014	Research (Google Maps API, OSMdroid), Programming	2,75
15.05.2014	Programming	3
16.05.2014	Research (Google Maps API, OSMdroid), Programming	6

17.05.2014	Programming	7,5
18.05.2014	Programming, UI Design	5,75
26.05.2014	Programming	7
27.05.2014	Programming	2
30.05.2014	Programming	2,75
31.05.2014	Research (Bluetooth), Programming	2,5
01.06.2014	Programming	6,50
02.06.2014	Programming	6,5
05.06.2014	Research (GPS input: NMEA-Sentences)	0,5
06.06.2014	Programming	8
10.06.2014	Programming	4
13.06.2014	Research (maps compass mode)	1,5
14.06.2014	Programming	4,75
15.06.2014	Programming	6,25
13.07.2014	Programming, UI design, Refactoring	2,5
14.07.2014	Programming	7,25
15.07.2014	Programming	10,25
16.07.2014	Programming, UI design & feedback	8,50
19.07.2014	Programming	1
20.07.2014	Programming, UI design	2,75
22.07.2014	Documentation	0,5
23.07.2014	Programming	4,5
04.08.2014	Programming	0,5
08.08.2014	Programming, UI design & feedback	5
10.08.2014	Programming	1,75
11.08.2024	Programming, Refactoring	0,5
12.08.2014	Refactoring	2
14.08.2014	Refactoring	3
17.08.2014	Programming	2,5
18.08.2014	Research (map, show direction arrow and distance), Programming	4
19.08.2014	Programming	6,5
21.08.2014	Programming	3,25
22.08.2014	Programming	3
23.08.2014	Programming	4,75
26.08.2014	Programming	4

27.08.2014	Programming, Languages (english, german), Exception-Handling	10
28.08.2014	UI design & feedback	0,25
29.08.2014	Programming	3,5
01.09.2014	Programming	7,5
02.09.2014	Programming	4
03.09.2014	Programming	2
04.09.2014	Programming, Testing, Exception Handling	5,5
09.09.2014	Programming, UI design	6,25
10.09.2014	Programming, UI design	2
11.09.2014	Documentation	4,5
13.09.2014	Documentation	0,5
14.09.2014	Refactoring, Documentation	10
15.09.2014	Programming, Documentation	5
24.09.2014	Programming, Documentation	1,5
25.09.2014	Documentation	1
26.09.2014	Programming	2
27.09.2014	Programming	6
28.09.2014	Programming, Documentation	1
30.09.2014	Testing, Documentation	6
01.10.2014	Programming, Testing, Documentation	9
02.10.2014	Documentation	7
03.10.2014	Documentation	1,75
	<i>Total</i>	<i>276,5</i>

2 Implementation

2.1 System Environment

This Android project was developed under Windows 7 (64 bit) in Java 7 with the Eclipse IDE [2] and the help of the Android SDK [3] and ADT (Android Development Tools) plugin for Eclipse [4]. The Android ADT contains all necessary components to develop an Android application.

The application was tested on various Android mobile phones, mostly on a Google Nexus 4 with Android version 4.4.4. The application is designed to run on Android devices 4.0 and higher.

2.1.1 GPS-Bluetooth-Device

GPS Child Finder comprises a GPS-Bluetooth-device, which should be carried by the child of the user to show its location. The application was developed and tested with a **BT-GPS-8 Jentro device** [5]. To add this device to the phones Bluetooth paired devices, the required pin is 0000 (like usually).

GPS Child Finder should also work with any other Bluetooth-device which sends NMEA sentences to the phone it's connected with. It's important to know that there is no guarantee for that, because the application isn't a matured software at this point.

2.2 Libraries

- *Google Play Services API(google-play-services_lib - Library Project)*
Provides Google-powered features such as Google Maps and more. [1]
- *Android Support v4 (android-support-v4.jar)*
Assists with development of applications for android API level 4 or later. [6]
- *SLF4J Android (slf4j-android-1.6.1-RC1.jar)*
Logging Framework for the application. [7]

2.3 Implementation Features

2.3.1 Calculation of the maps bounds

The new bounds of the map are calculated when the user wants to focus on a certain location like the location of himself or the location of the child/the GPS-Bluetooth-device by clicking on the specific button. The purpose of this calculation is to make both – the child's location and the location of the user – visible on the map when one of them is focused on. So the user doesn't have to search around or zoom in or out for seeing the two location icons properly.

The method is held by the **BoundsCalculator.class** and the main task is done in the **calculateNewBounds()** method.

The method simply checks, if the location (otherLocation), which the user hasn't focused on – but should still be visible on the map – is inside the current bounds of the focused location (latLngBounds). If that's NOT the case, the zoom level of the map will be decreased (zoomLevel) or in other words: it's zooming out on the map until both locations are visible:

```
while(!(map.getProjection().getVisibleRegion().latLngBounds)
        .contains(otherLocation)) {
    zoomLevel -= 1;
}
```

2.3.2 Get GPS coordinates from the GPS-Bluetooth-device

If a GPS-Bluetooth-device is successfully connected with the phone, the application begins to read the NMEA-sentences which the device sends via Bluetooth. These NMEA-Sentences contain information about the location of the device. The ones that contain the GPS coordinates start with "\$GPGGA" (Global Positioning System Fix Data). These are composed of the time, latitude, longitude, the number of satellites in view, altitude and other data (See [8] for more information).

The **BluetoothDataManager.class** contains the **readNMEASentences()** method. This reads the input of the device from a StringBuffer and calls the **getCoordinates()** method.

Example for extracting the coordinates from the received sentence:

```
$GPGGA,143019, 4811.558,N, 01622.656,E,1,08,0.9,545.4,M,46.9,M,,*47

4811.558,N          Latitude 48 degree 11.558' North
01622.656,E          Longitude 16 degree 22.656' East
```

The `getCoordinates()` method splits the sentence, extracts the coordinates and converts them into decimal coordinates to forward it to the map and position the icon of the device on the right location.

2.4 Known Bugs and Limitations

2.4.1 Devices

GPS Child Finder can only be used on Android Smartphones with version 4.0 (Ice Cream Sandwich) or higher which have the Google Play Service API installed, have internet connection and support Bluetooth.

2.4.2 Precision

GPS Child Finder is depending on a good Internet Connection and/or a GPS Signal, so that the location can be shown as precisely as possible. Additionally, the GPS-Bluetooth-device needs a strong GPS signal to send exact location information to the mobile phone of the user. Inaccuracy concerning the locations of the user and the child cannot be ruled out

2.4.3 GPS-Bluetooth-device

Like written before, GPS Child Finder should also work with any other Bluetooth-device (not only with the GPS-BT Jentro 8) which sends NMEA sentences to the phone it's connected with. It's important to know that there is no guarantee for that, because the application isn't a matured software at this point.

2.4.4 Bluetooth range

Another point is the Bluetooth range. The most GPS-Bluetooth-devices have a range of 10 meters maximum (like the one this application was tested with). These are the Class 2 Bluetooth-devices. There is also a Class 1, which guarantees a range of 100 meters, but these devices are very expensive and therefore not suitable for this kind of project.

2.4.5 Compass mode sensitivity

If the compass mode is activated (when the user focuses on his location) the application reacts to hardware sensor events from the mobile phone. Some mobile phones can be very sensitive concerning rotation, shaking or other movements a user does with his phone. The result could be a jittering of the Google Map.

2.5 Potential and Further Development

- Make GPS Child Finder suitable for larger distances. The following further developments could be made:
 - Expanding the Bluetooth range, e.g. with a Bluetooth-Class 1-device
 - Change the location information transfer from Bluetooth to WLAN by using a GPS-device with a SIM card.
- Tracking the route the child is going
- Make it possible to connect more GPS-Devices to the users mobile phone for every child he has. On the one hand it should be possible to select a child to track, on the other hand the map should show all children on one screen.

3 Availability

3.1 Source Code

The source code of this project is available on Bitbucket and can be cloned from the following git repository <https://bitbucket.org/lxdanii/gps-child-finder>.

3.2 Documentation

The documentation of the source code is also available on the Bitbucket repository named above in the folder GPSCildFinder/doc.

3.3 Languages

GPS Child Finder is available in two languages: English and German. The application chooses the appropriate language according to the local settings of the mobile phone at run time.

4 References

- [1] Google Play Services API <https://developer.android.com/google/play-services/index.html>
- [2] Eclipse IDE <http://eclipse.org/downloads/>
- [3] Android SDK <http://developer.android.com/sdk/index.html>
- [4] Android ADT Plugin <http://developer.android.com/sdk/index.html>
- [5] BT-GPS-Jentro 8 http://www.handytechnik.de/assets/files/bedienungsanleitung/Jentro%20activepilot%20GPS%20Maus/89150_Jentro-bt-gps-8_UM.pdf
- [6] Android Support v4 <http://developer.android.com/tools/support-library/index.html>
- [7] SLF4J Android <http://www.slf4j.org/>
- [8] NMEA Sentences <http://aprs.gids.nl/nmea/>