SUNLIME
IT SERVICES

Driver's Logbook

Helena Adam, Claudio Knapp, Laura Rössl, Paul Zwölfer

March 17, 2016

Abstract

Every day people drive to work in their own, or their company's car. In order to get tax refunds, the company has to keep careful records of the driven kilometers. This procedure takes a lot of time, especially when the recording is done manually. Therefore our partner company suggested to automatically collect the needed data on an external device.

Our project consists of a device placed in the car. This hardware consists of a Raspberry Pi 2 (RPi2) and a Global Positioning System (GPS) module. It tracks the driven distances and computes a total of the kilometers. This information is inserted into a Database (DB). If there is no internet connection while the car is in motion, the data are saved on the device and uploaded as soon as possible once there is an internet connection available. Afterwards, the user can log into his user account on our website or mobile application to see the driven tracks. The user also can enter routes manually using the app or through the website. The software provides for a separation between private and business usage of the car. A possibility to edit the tracked ways is also given. It's also possible to view it on the phone or tablet via a mobile application.

In summary, we have developed a system consisting of hardware and software to log tracks via GPS data while driving.

Acknowledgements

We want to thank a lot of people. On the one hand, those people, who made it possible for us to attend the Höhere Technische Bundeslehranstalt (HTBLA) Kaindorf. On the other hand, those who contributed valuable help to create a successful diploma thesis.

We are full of thankfulness to all employees of our partner company, Sunlime IT Services GmbH, who supported us in our work. Special thanks go to our employer DI Dominik Fuchshofer, who gave us the chance to do our diploma thesis with his company. As contact person he always stood at our side whenever we needed assistance. Without his tireless input, patience and understanding it would not have been possible to complete our diploma thesis successfully.

Furthermore, we would like to thank DI Dr. Wolfgang Pölzleitner and DI Florian Schreiber, who took over the role supervising teachers. When we had problems or other blockades, they always were there to help us finding a solution and make the best out of it. Through their cheerful character and their enthusiastic interest in our work, we even enjoyed the meetings and project follow-ups.

Then, we want to thank our head teacher Mrs. Mag. Michaela Primig, who supported us during our career at HTBLA Kaindorf. Sometimes it was not that easy for her to understand our attitudes, but then, she never lost hope that we would finally cope.

Last but not least, our deep thankfulness needs to be expressed to our parents. They supported us during our HTBLA career and provided motivational help whenever needed. Finally special thanks go to Paul's dad, who helped us a lot with our diploma thesis.

Project Members

Helena Adam

Helena took over the role of the team leader of this project. She was our connection person to our two supervising teachers DI Dr. Wolfgang Pölzleitner and DI Florian Schreiber, and our partner company Sunlime IT Services GmbH. Since she had already worked there for an internship, she had great connections to the employees.

She managed the task of comparing competitive products we worked on together. Additionally, she documented held meetings and guided the project management in general.

Claudio Knapp

Our hardware expert Claudio cared about our hardware prototype, the RPi2. He was involved in the analysis of competing products and find out if there is a already a similar offer. He also researched on necessary modules to extend the functionalities of our product. After he found out how we could install them properly, he got to the work. These added modules were for example the GPS module, the Uninterruptible Power Supply (UPS) or the antenna. With his enthusiastic way, working on the thesis never got boring.

Laura Rössl

The role of the database expert was taken over by Laura, who set up and maintained our DB structure. She kept up with all the software construction changes and therefore updated the Entity Relationship Diagram (ERD) in the required way. Laura also designed other diagrams such as the software diagrams or the use case diagrams.

She was also participated in the research of different competitors and the analysis of the demand of the market.

Moreover, she did a great job creating the wireframing with the tool balsamiq. This was essential for the future organization of the user interface in cooperation with the company.

Paul Zwölfer

Pauls responsibility was the programming of the software for the working prototype. He provided a lot of technical information about the software, modules and the different APIs. Due to his engagement and never ending ambition, the development of the software went very well.

He also contributed documentation of competitors, supported Laura with designing the wireframing and maintaining the DB.

Together with Claudio, he started to test the software right from the first working state. This helped to avoid problems and provided a well conceived program structure.

Statutory Declaration

With this I declare that all work presented in this diploma thesis is my own mindwork. I have used only the referred sources and aids and all parts which have been either literally or in general manner from other sources have been indicated accordingly.

Furthermore, the secondary technical school HTBLA Kaindorf/Sulm is privileged to use any of my work included in this diploma thesis for educational or research purposes, as long as special attention is paid to data security and competitional regulations.

Helena Adam

Claudio Knapp

Laura Rössl

Paul Zwölfer

Contents

Abstract 2

Statutory Declaration 6

Introduction 14

1 Project Management 18

1.1 Scrum . . . 18

1.1.1 Description of Scrum . . . 18

1.1.2 Why we have chosen Scrum . . . 18

1.2 Project Organization Chart . . . 19

1.3 Milestones . . . 20

1.4 Working Hours . . . 21

2 Software Requirements 22

2.1 Initially . . . 22

2.1.1 Data Transmission . . . 22

2.1.2 Data Design . . . 22

2.1.3 Webserver . . . 22

2.2 Additional Requirements . . . 22

2.2.1 REST . . . 22

2.2.2 MySQL Database . . . 23

3 Competitor Analysis **24**
3.1 Introduction 24
3.2 Miles 26
3.3 OsmAnd 27
3.4 TOMTOM 28
3.5 Fahrtenbuch (von Stefan Meyer) 29
3.6 TOUR 30
3.7 Mileage Logbook 31
3.8 Fahrtenbuch (myLogbook) 32
3.9 MyLog GPS Fahrtenbuch Kosten 33
3.10 Carpanion 34
3.11 Fahrtenbuch Mileage Book 35
3.12 Trucker Logbook 36
3.13 Logbuch App 37
3.14 Drives Fahrtenbuch 38
3.15 Abax Triplog 39
3.16 GPS Log Book 40
3.17 GPS Log Book Live 41
3.18 Gurtam 42
3.19 Maxtech 43
3.20 Tractive 45
3.21 Summary 46

4 Prototype **48**
4.1 RPi2 Hardware 48
4.1.1 GPS Module 48
4.1.2 GPS Antenna 48
4.1.3 UMTS Stick 49
4.1.4 UPS 49

4.1.5 OS . . . 51
4.1.5.1 Raspbian Wheezy . . . 51
4.1.5.2 RASPBIAN JESSIE LITE . . . 60
4.2 Software Description . . . 69
4.2.1 Functions . . . 69
4.2.2 GPS API . . . 69
4.2.3 Tape API . . . 69
4.2.4 Save Thread . . . 69
4.2.5 API Thread . . . 70
4.2.6 PHP (REST) . . . 70
4.3 Code Structure . . . 71
4.3.1 Classes . . . 71
4.3.1.1 GPS.java . . . 71
4.3.1.2 BL.java . . . 71
4.3.1.3 SaveDataThread.java . . . 72
4.3.1.4 Data Manager.java . . . 72
4.3.1.5 JContainer.java . . . 72
4.3.1.6 JPoint.java . . . 72
4.3.1.7 GPSConfiguration.java . . . 72
4.3.1.8 GsonConverter.java . . . 73
4.3.2 Class Diagram . . . 73
4.4 Problems . . . 74
4.4.1 Uploading into Database . . . 74
4.4.2 Offline Saving . . . 74
4.5 Functional Testing . . . 74
4.5.1 Database Connection Error . . . 74
4.5.2 Inaccuracy of the GPS Coordinates . . . 75
4.5.3 No GPS Signal . . . 76
4.6 Future Work and Ideas . . . 77

5 Database **79**
5.1 Entity Relationship Diagram . . . 79
5.2 Entities . . . 80
5.2.1 Customer . . . 80
5.2.2 Vehicle . . . 81
5.2.3 Address . . . 81
5.2.4 TrackingHardware . . . 81
5.2.5 Licence . . . 82
5.2.6 Track . . . 82
5.2.7 Point . . . 82
5.3 Data Definition Language . . . 82
5.4 Data Manipulation Language . . . 85
5.5 Problems . . . 86

6 Wireframing **87**
6.1 Smartphone . . . 87
6.1.1 Login . . . 87
6.1.2 Menu . . . 87
6.1.3 Tracks . . . 87
6.1.4 Settings . . . 88
6.1.5 Statistics . . . 88
6.2 Tablet . . . 89
6.2.1 Login . . . 89
6.2.2 Menu . . . 89
6.2.3 Tracks . . . 89
6.2.4 Settings . . . 89
6.2.5 Rename Device . . . 90
6.2.6 Register Car . . . 90
6.2.7 Statistics . . . 90

6.3 Web Portal 91
6.3.1 Home 91
6.3.2 Login 91
6.3.3 Settings 91
6.3.3.1 Register Car 91
6.3.3.2 Rename Device 91
6.3.4 Track 91
6.3.4.1 Edit 92
6.3.4.2 Create 92
6.3.5 Statistics 92
6.3.6 Menu 92

7 Optional Uses **94**
7.1 Taxi 94
7.2 Car Renting 94
7.3 Train Controller 94
7.4 Mowing Machine 95
7.5 Helicopter 95
7.6 Pet Tracking 95

8 User Manual **97**
8.1 General 97
8.2 Installation Regulations 97
8.3 LED 97
8.4 Warnings 98

9 Tools and Technologies **100**
9.1 Java 100
9.2 TAPE API 100
9.3 GPS API 100

9.4 RASPBIAN 101
9.5 GIT 101
9.6 Google Drive 101
9.7 PHP 103
9.8 BalsamiQ 103
9.9 MySQL 103
9.10 Netbeans 104
9.11 Notepad++ 104
9.12 LaTeX and Texmaker 104
9.13 draw.io 105
9.14 XAMPP 105
9.15 phpMyAdmin 105
9.16 Hangouts 106

Glossary **107**

Bibliography **110**

Foreword from the Employer

Kilometre allowance and the documentation of driven kilometres for the company are important todos for every self employed entrepreneur. The fact is that writing every business car ride into an excel sheet or manually into a car log book is really annoying. It would be much more convenient if a device in my business car logs every trip and I am able to define my car rides via a web portal afterwards. This was the idea behind "Log book" which we wanted to test drive with our friends at HTBLA Kaindorf/Sulm. Next to the hardware prototype it was also important to get an idea for a possible DB solution. To understand the project better we provided a design proposal next to the hardware to sum up the first section of the project.

But more important than the final outcome of the project is to see how young people are able to come up with new and interesting ideas and dive into project related subtasks which have to be accomplished. It is also great to transfer some of the know-how we have worked out and to see how these ideas are adopted and developed for the project.

I am sure that every project member is able to take something from this diploma thesis and use it for their own future business life. I wish Helena, Laura, Claudio and Paul the very best for the final exams. You are going to rock the stage.

Best,
Dominik

DI Dominik Fuchshofer, BSc
CEO/CTO at Sunlime IT Services GmbH
February 2016

Introduction

Our project leader Helena Adam got in contact with the company Sunlime IT Services GmbH during previous internships. This is why we got the chance to work in cooperation with the company on our diploma-thesis. The Chief Executive Officer (CEO) of Sunlime, DI Dominik Fuchshofer, gave us actively support when necessary and was on hand for help and advice. He had the idea to create an electronic driver's logbook. If you use your private car for companys purpose, you have to note down driven trips for financial reasons. Using a company car for private and business trips, until now, you also had to note down the distances you drove.

One year has passed since we started working on our diploma thesis and finally we have completed. The main tasks of our project were the competitor analysis, the design of the database, the wireframing, the project management and the working prototype. The software we created runs on this prototype, collects GPS data and uploads it in our online DB. Our partner company provided the web server for the web portal and all technical components we needed.

With the help of our supervising teachers and our employer, and useful tutorials on the internet, we were able to create a working prototype.

State of the Art

Until now, the company has to track the driven distance by hand. They note down different parameters in an EXCEL sheet. There is a separate sheet for each month of the year. These parameters include the date, tachometer kilometers at the beginning and the end of the trip, how much kilometers are driven for the companys or private use, the route they took, and the reason for driving.

This method is very time consuming and has to be done for every company related trip. Non company related trips must be excluded.

Our functionalities

The RPi2 powers up when the driver starts up the cars ignition. After that, the autostart routine starts our software, including the process of collecting GPS data, saving and uploading it.

When the software started up, it checks if there is some GPS data from previous tracks to be uploaded. If that is the case, this data will be read from the save.jPoint file into the queue. After that, the save thread uploads a maximum of 50 points to the Representational State Transfer (REST) service.

Then the GPS data of the current track gets saved via the API thread into the queue and then synchronized with the save.jPoint file.

Reading and writing data simultaneously is possible due to the use of a blocking queue. This special type of queue manages the access of the list so that no errors occur.
Successfully uploaded points are removed from the list and therefore from the synchronized backup file.

This runs through as long as the RPi2 is supplied with power.

Role of the Company

In this part of our diploma-thesis it is listed up what our employer, Sunlime IT Services GmbH, contributes to the work of the electronic logbook. Following areas are included:

- Hardware
- Data transfer
- Design requirements
- Content requirements

The company provides the RPi2 with all the components like GPS Module or antenna to create a working hardware prototype for the logbook.

Secondly, at the data transfer, they will define when and over which portal the data has to be sent. Moreover, it should be apparent which user is transferring the data, so it can be dedicated online. It is also important to consider the user data when designing the DB. Furthermore, a feature list has to be defined and integrated. The web server for the portal is also provided and maintained by the company.

An external service provider creates the design from the existing wireframing and Sunlime IT Services implements the technical aspects to it. The external service provider should also design a Logo for the product.

Thirdly, the company has to choose a "Mobile-first" approach to react to the display measures of each end device.

Then, the logbook should also work on smartphones. Therefore, Adobe PhoneGap will help to transact the program. The mobile application has to be designed one time only.

Finally, the company is responsible for the security on the servers. It should not be possible to have access on a server as third person. Passwords are only allowed to get saved encrypted.

Goals

The goal of our diploma-thesis is to build a working hardware prototype. A RPi2 with a GPS module has to be placed in a car. The prototype tracks GPS data and transfers it to the REST service. REST is responsible for uploading data into the DB.

In case there is no internet connection, we store the GPS data locally. The program structure is described in our Software description section.

Information about the user and the tracks have to be stored in the DB. We also created and designed an ERD to illustrate the structure of the DB. Furthermore, the Data Definition Language (DDL) and the Data Manipulation Language (DML) resulted out of the ERD.

The wireframing is the Front-End representation of our diploma thesis. It displays the user interface on our web portal and mobile application. The result of it is a user-friendly design.

To get a unique product, the market needs to get analysed. Here, the competitors should get detected and compared in different attributes. For instance: costs, supported platforms, GPS, hardware etc. With this information, we create a diploma-thesis that differs from other products. Furthermore, other possible uses of the hardware prototype should get noted down.

Last but not least, the project management of the work packages has to be defined appropriate and documented.

1 Project Management

1.1 Scrum

1.1.1 Description of Scrum

Scrum is an iterative and incremental agile process used to develop software. The team had to organize itself by taking the tasks from the scrum board.

At the beginning of developing the product, the project goals got divided into sprints. To have an overview over the whole project, the so called scrum meetings were held. During these scrum meetings, done tasks and what's left to do, got discussed. Those discussions can be held daily and help to see the process of the project.

Three different roles are represented in scrum. Firstly, the development team. They are responsible for the programming part of the software. Secondly, the product owner. He provides the user stories of the team. Moreover, to keep the customer satisfied, he makes sure that the software includes everything the customer requested. Last but not least, the Scrum master. Disturbances can delay the milestones deadlines. To prohibit this, he makes sure that the development team can work efficient without the project managers or the customers interference.

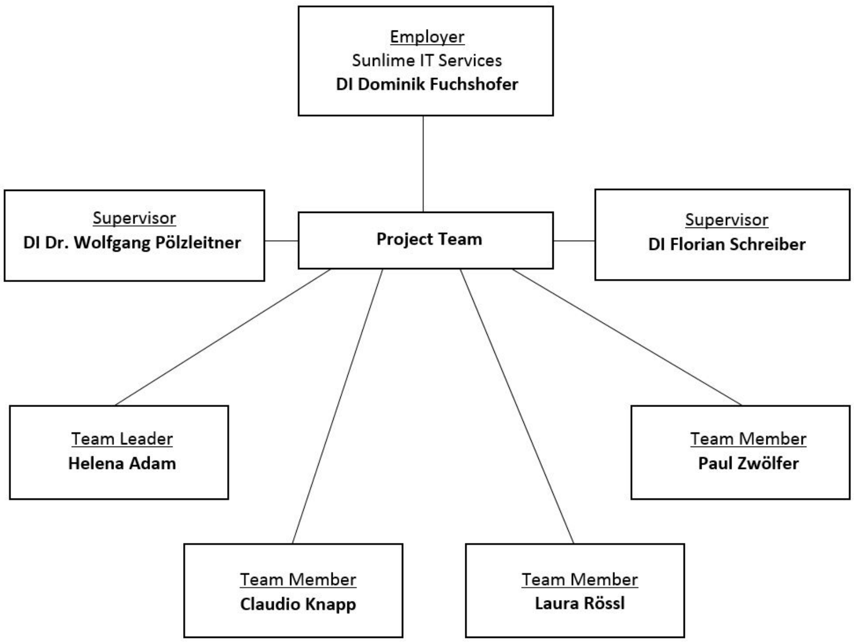
1.1.2 Why we have chosen Scrum

As already mentioned above, scrum is agile because of the sprints. Basically, it means that there is no problem in changing or adding sprints in the current tasks. We have already had a good experience with scrum in the 3rd grade so we chose it.

1.2 Project Organization Chart

The following diagram shows the structure of the organization. It pictures who was involved, what their position in the project was and their relation to each other.

Our contact person for our diploma-thesis was DI Dominik Fuchshofer. He is the CEO of the company Sunlime IT Services GmbH, we created our work for. Our two supervisors DI Dr. Wolfgang Pölzleitner and DI Florian Schreiber took care of the progress we made. The members of our project team are Helena Adam, Claudio Knapp, Laura Rössl and Paul Zwölfer. Everyone communicated with the employer on its own, over Hangouts.



1.3 Milestones

Milestones are tools for the project management to schedule the project. The present work gets divided into some parts. Then, it gets defined until when a part has to be done. This tool is used to avoid bigger delays in this project.

In the chart below it is visible into which parts we have divided our diploma-thesis and the date until it was due.

Milestones	**Deadline**
Competitor analysis	July 10, 2015
Wireframing	July 31, 2015
Designing of data	November 24, 2015
Functioning software	January 12, 2016
Hardware prototype	February 16, 2016
Bug fixing	February 23, 2016
Documentation	March 17, 2016

1.4 Working Hours

Here you can see a listing of our working hours and how much time we spent on different tasks. You can see in the following table that we spent the most hours on documentation of the diploma thesis. The main reason for this is the technology LaTeX we used.

Type	**Hours**
Administration	194,05
Designing	7
Data design	43
Documentation	225,7
Programming	179,75
Research	85
Raspberry Pi	140
Testing	18
UML Diagrams	14
Wireframing	81

2 Software Requirements

2.1 Initially

2.1.1 Data Transmission

It has to be clear when a transmission takes place and which data it contains. The user who is getting tracked must be recorded for the DB too.

2.1.2 Data Design

Also the information about the users have to be stored in the DB. Furthermore, a list of features has to be defined and integrated. This serves to adapting extensions of products from the competitive analysis.

2.1.3 Webserver

The webserver is provided and maintained by the company Sunlime IT Services GmbH. The server is located in a data center in Vienna and is tethered with 10Gigabit per second (Gbit/s).

2.2 Additional Requirements

2.2.1 REST

We switched to the technology REST to save data in the DB. Our REST Server is written in PHP Hypertext Preprocessor (PHP) and uploads data.

2.2.2 MySQL Database

Since our partner company's main products are homepages, they already had a running My Sequential Query Language (MySQL) DB. So they created a new workspace for us.

3 Competitor Analysis

3.1 Introduction

In this part of our diploma thesis we give you some information about our current competitors we are sharing our project idea with.

One important factor to create an outstanding product and to lead the market is to know our competitors. Like it is said "Keep your friends close, but your enemies closer". So we were supposed to do some researches about similar products. They are listed on the next few pages. We try to sum up the most important facts and a short description of the products competing.

The competitor analysis showed us what we have to look at and what customers exactly need.

Compared Products

- Miles
- OsmAnd
- TOMTOM
- Fahrtenbuch (von Stefan Meyer)
- TOUR
- Mileage Logbook
- Fahrtenbuch (myLogbook)

- MyLog GPS Fahrtenbuch Kosten
- Carpanion
- Fahrtenbuch Mileage Book
- Trucker Logbook
- Logbuch App
- Drives Fahrtenbuch
- Abax Triplog
- GPS Log Book
- GPS Log Book Live
- Gurtam
- Maxtech
- Tractive

3.2 Miles

Costs €39,99 once

Supported Platforms The application Miles is compatible for iPhone Operating System (iOS) 8.

GPS No GPS connection necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking Data can only be entered manually.

Internet Connection Internet connection necessary.

Export It's possible to export data to a Portable Document Format (PDF) or Comma-separated values (CSV) File.

Other Features No other features included.

References: [23]

3.3 OsmAnd

Costs €6,99 once

Supported Platforms For Android and iOS smartphones and tablets. Also runs on a wide array of Linux-based systems.

GPS Navigation- and tracking system in one.

Business/Private mode Only private mode.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking automatic

Internet Connection No Internet connection necessary.

Export It's possible to export data to a PDF or CSV File.

Other Features download offline Maps for tracking and navigation

References: [27]

3.4 TOMTOM

Costs €69.41 once

Supported Platforms For Android (2.3.3 or higher) and iOS (5.0 or higher).

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking No information

Internet Connection No information **Export** It's possible to export data to a PDF or CSV File.

Other Features No other features included.

References: [5]

3.5 Fahrtenbuch (von Stefan Meyer)

Costs €5,99 once

Supported Platforms iOS

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App for iPhone and Apple Watch

Manual/Automatic Tracking manual and automatic

Internet Connection Internet connection is necessary.

Export It's possible to export data to a PDF or CSV File.

Other Features No other features included.

References: [8]

3.6 TOUR

Costs Cost free. with features €8,99 + €5 to transfer the data to the computer

Supported Platforms iOS

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking automatic

Internet Connection Internet connection is necessary.

Export It's possible to export a Tax Office conform file.

Other Features Trips can be combined afterwards

References: [30]

3.7 Mileage Logbook

Costs free

Supported Platforms Android

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking automatic

Internet Connection No internet connection is necessary.

Export It's possible to export data to a Hyper Text Markup Language (HTML), Excel or CSV File.

Other Features No other features included.

References: [22]

3.8 Fahrtenbuch (myLogbook)

Costs €4,98 once

Supported Platforms Android

GPS GPS connection is necessary.

Business/Private mode Only business mode.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking automatic

Internet Connection No internet connection is necessary.

Export It's possible to export data to a HTML, Extensible Markup Language (XML), INtex, Euro-Fahrtenbuch, Fahrtenbuch Express, WISO Fahrtenbuch or CSV File.

Other Features No other features included.

References: [7]

3.9 MyLog GPS Fahrtenbuch Kosten

Costs free

Supported Platforms Android

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking automatic

Internet Connection No internet connection is necessary.

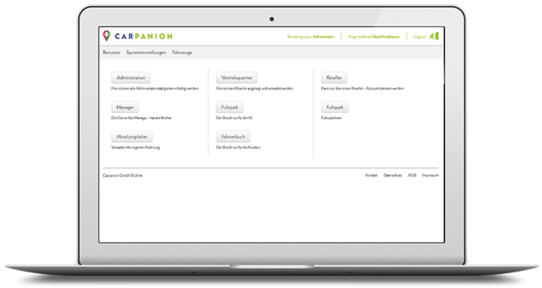
Export It's possible to export data to a PDF File.

Other Features No other features included.

References: [24]

3.10 Carpanion

Costs free



Supported Platforms Android, iOS, Windows

GPS GPS connection is necessary.

Business/Private mode Only private mode.

Hardware No Hardware. App is only used on smartphones.

Access Forms Web Form and App

Manual/Automatic Tracking manual

Internet Connection Internet connection necessary.

Export It's possible to export data to a PDF or CSV File.

Other Features You can take pictures of your fuel receipts and save them together with your refueling stops.

References: [3]

3.11 Fahrtenbuch Mileage Book

Costs €3,95 monthly (30 days testversion)

Supported Platforms Android, iOS, Windows

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms Web form, App

Manual/Automatic Tracking automatic and manual

Internet Connection No internet connection necessary.

Export It's possible to export data to a PDF, XML, Excel or CSV File.

Other Features No other features included.

References: [6]

3.12 Trucker Logbook

Costs free

Supported Platforms Android, iOS

GPS No information

Business/Private mode Only business mode.

Hardware No Hardware. App is only used on smartphones.

Access Forms Web form, App

Manual/Automatic Tracking manual

Internet Connection No information

Export No information

Other Features No other features included.

References: [32]

3.13 Logbuch App

Costs €5,99 once

Supported Platforms iOS

GPS GPS connection is necessary.

Business/Private mode Only private mode.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking No information

Internet Connection No information

Export It's possible to export data to a PDF File, Excel or CSV File.

Other Features Weather

References: [19]

3.14 Drives Fahrtenbuch

Costs Cost free, but there's also a chargeable version where private routes get tracked automatically and you can register more than one vehicle.

Supported Platforms iOS

GPS No information

Business/Private mode You can split your private tracking from your business tracking.

Hardware No Hardware. App is only used on smartphones.

Access Forms App

Manual/Automatic Tracking The manual tracking version is cost free, but at the chargeable version the automatic tracking is possible and you can split your private and business tracking.

Internet Connection No information

Export It's possible to export data to a PDF or CSV File.

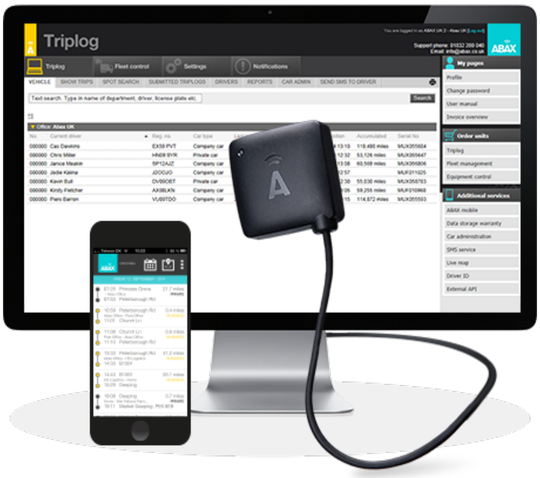
Other Features

- You can use DB entries as pattern for future tracks
- Statistics
- Backups
- In the chargeable version of the app, you can register more than one vehicle

References: [4]

3.15 Abax Triplog

Costs €35,29 monthly & €249,00 Hardware



Supported Platforms iOS

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

HardwareThe hardware is very robust and waterproof.

Access Forms Web form, App

Manual/Automatic Trackingautomatic

Internet Connection Internet connection necessary.

ExportNo information

Other Features

- Contract period of 3 years
- Lifetime Warranty

References: [1]

3.16 GPS Log Book

Costs €117,11 + €26,33 per year

Supported Platforms Web interface

GPS GPS connection is necessary.

Business/Private mode Only private mode.

HardwareThe product consists of a plug in for the car.

Access Forms Web form

Manual/Automatic Trackingautomatic, manual

Internet Connection Internet connection is necessary, but data doesn't get transmitted to the DB.

ExportIt's possible to export data to a PDF or CSV File.

Other Features

- 5 years data retention
- Google Maps integrated
- 4MegaByte (MB) log data on the device

The GPS Log Book device plugs into a cigarette lighter. You have to copy the data manually from the GPS Log Book on you PC.

References: [10]

3.17 GPS Log Book Live

Costs€416,69 + €18,16 monthly + Installation €89,87

Supported Platforms Web interface

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

HardwareThe hardware is very robust and waterproof.

Access Forms Web form

Manual/Automatic Trackingautomatic + manual

Internet Connection Internet connection necessary. Data gets automatically transmitted to the DB.

ExportIt's possible to export data to a PDF or CSV File.

Other Features

- 5 years data retention
- Google Maps integrated
- Live

References: [11]

3.18 Gurtam

Costs No information

Supported Platforms Every platform that supports Google Chrome 20+, Firefox 15+, Safari 5+, IE 9+, Opera 10+

GPS GPS connection is not necessary.

Business/Private mode You can split your private tracking from your business tracking.

HardwareNo Hardware. App is only used on smartphones.

Access Forms Web form

Manual/Automatic Trackingmanual

Internet Connection No internet connection is necessary.

ExportIt's possible to export data to different Files.

Other Features

- specifying the beginning and the end of the trip; choosing type (business/personal trip);
- specifying initial and final position, duration, mileage data, etc.;
- manual adding columns containing user notes;
- logging all the changes;
- printing or exporting to file;
- setting Costs per kilometer or per litre (in case of personal trip it helps to calculate the total Costs).

References: [14]

3.19 Maxtech

Costs €189 once service fee (including the service, SIM card and subscription) is roughly €20 per month (prices excl. VAT)

Supported Platforms No information given

GPS GPS connection is necessary.

Business/Private mode You can split your private tracking from your business tracking.

HardwareHardware is included (picture on the right)

Access Forms Web form

Manual/Automatic Trackingautomatic and manual

Internet Connection Yes, data gets transmitted to a server

ExportIt is possible to export data to a PDF or CSV File.

Other Features

- An automatic logbook remembers to collect all the drive data so you will no longer forget any travels.
- data is available anywhere and anytime
- track your car when the driver is someone else and it also helps if your car is stolen – a tracking device will send the location of your car when it moves.
- if no internet is available, the drive data gets stored on an internal buffer memory, that can save up to 2000 km

Vehicle tracking devices can be installed in three different ways:

- Cigarette lighter plug
- OBD connector
- Fixed mounting to car battery (10V-30V)

References: [21]

3.20 Tractive

Costs€79.99 once + €4.99 or €7.49 (Premium) per month

Supported Platforms Android and iOS

GPS GPS connection is necessary.

Business/Private mode -

HardwareHardware is included (picture on the right)

Access Forms App

Manual/Automatic Tracking automatic

Internet Connection Yes, data gets transmitted on the smartphone of the user.

Export-

Other Features

- Sensors for:
  - Movement
  - Speed-Up
  - Temperature
  - Brightness
- Waterproof
- Alarm, if the pet leaves a certain area

References:[31]

3.21 Summary

Most of the products only run on very few or only on one platform. There are different versions of software (pay/free) that divide the user experience in either good or bad way. Our idea is a single software with an extra hardware, web form and a mobile application. This mobile application runs on iOS, Android and also on Windows phones. These main functionalities should distinguish us from the rest of the market.

In the analysis above you can see some of the most important competitor products. In our opinion, the largest competitors are Mileage Book and Fahrtenbuch (von Stefan Meyer). These products have many features and a very low or rather no price. However, their product only includes a mobile application.

On the hardware side, GPS Log Book Live and the company maxtech with their product Automatic Driver's Logbook are our biggest competitors. Although their product is highly innovative, there is no mobile application included.

Another competitor product is the product of the company Tractive. It has functions like helping you not to lose your pet, tracking its activity and analyzing this data throughout the day.

In conclusion, there are hardly any products consisting of a hardware and a mobile application. We only found two competitors who include both. On the one hand it is Tractive that's original intention is to track your own pets. On the other hand it is Abax Triplog. But this product only runs on iOS so we have an advantage. So as you can see, our offer of an hardware and a mobile application together will distinguish our product from the competitors.

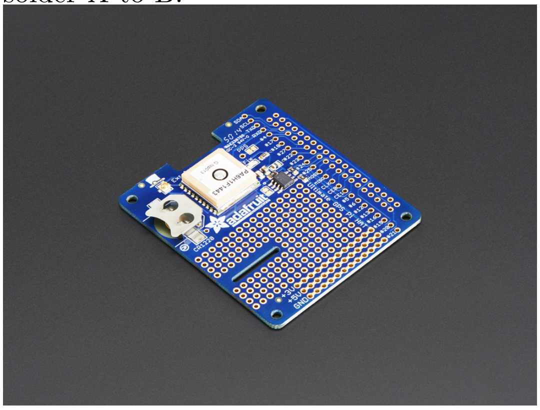
Name of Product	**Costs**	**Supported Plattform**	**includes Hardware**
Miles	€39,99	IOS 8	No
OsmAnd	€6,99	IOS, Tablets, Linux	No
TOMTOM	€69,41	Android and IOS	No
Fahrtenbuch (Stefan Meyer)	€5,99	IOS	No
TOUR	free	IOS	No
Mileage Logbook	free	Android	No
Fahrtenbuch (myLogbook)	€4,98	Android	No
MyLog GPS Fahrtenbuch	free	Android	No
Carpanion	free	Android, IOS, Windows	No
Fahrtenbuch Mileage Book	€3,95 per Month	Android, IOS, Windows	No
Trucker Logbook	free	Android, IOS	No
Logbuch App	€5,99	IOS	No
Drivers Fahrtenbuch	free	IOS	No
Abax Triplog	€35,29 per Month + €249 Hardware	IOS	Yes
GPS Log Book	€117,11 + €26,33 per Year	Web interface	Yes
GPS Log Book Live	€416,69 + €18,16 per Month + €89,87		
Installation	Web interface	Yes	
Gurtam	No information	Web interface	No
Maxtech	€189	No information	Yes
Tractive	€79,99 once + €4,99 or €7,99 (Premium)per Month	Andoid, IOS	Yes

4 Prototype

4.1 RPi2 Hardware

4.1.1 GPS Module

The GPS module had to be soldered to the General Purpose Input Output (GPIO) pins. We didn't trust our soldering skills and tried to avoid damage to the module. But it turned out that we needed a maximum of reliability for the connection and decided to solder A to B.



4.1.2 GPS Antenna

We only used the GPS antenna for testing. Our problem was that it took really long to get a GPS signal every time we changed the location. With the antenna it worked much quicker. In the final product an antenna would not be necessary because the gps board remembers the last position and a position update is faster by orders of magnitude.

4.1.3 UMTS Stick

The Universal Mobile Telecommunications System (UMTS) stick is necessary for transmitting the data to the DB.

4.1.4 UPS

If the engine of the car is switched off, the power supply of the RPi2 is interrupted. When the RPi2 gets power from the engine again, it has to do a system check and repair all files. Our final product will use an UPS. This UPS supplies our RPi2 with power even if the engine of the car is turned off. This eliminates the problem of data loss.

4.1.5 OS

4.1.5.1 Raspbian Wheezy

When we started our diploma thesis the most common operation system was and still is Raspbian Wheezy. At the beginning we downloaded the image and wrote the .img file on the microSD Card. You always have to connect the RPi2 to a screen to start Raspbian Wheezy. There you have to configure things like the keyboard layout. You can also do that later with the sudo raspi-config command. After that you have to reboot the RPi2. Since we wanted to have a remote connection, we installed a Tight Virtual Network Computing (TightVNC) Server on the RPi2. For this, we used the

```
sudo apt-get install tightvncserver
```

command.
To connect the RPi2 via a TightVNC Client, we had to start the server with the

```
tightvncserver
```

command. Then the Server started on the Port 5901. For the connection establishment, we had to write the Internet Protocol (IP) and the Port 5901 into the TightVNC Client. After that, we started to configure everything for the use of the GPS module.

GPS

First we had to configure the communication between the RPi2 and the Universal Asynchronous Receiver Transmitter (UART) pins on it. These UART pins are GPIO 14 and 15.
To stop the sending of the debug information, we had to edit the following file /boot/cmdline.txt with the

```
sudo nano /boot/cmdline.txt
```

command.
There was one line within this text file and there the following text needed to be deleted

```
console=ttyAMA0,115200 kgdboc=ttyAMA0,115200
```

Now it looked like this:

```
/dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

The RPi2 sends all terminal output over the external serial. To disable this behaviour, we had to edit the /etc/inittab file. We did this with the

```
sudo nano /etc/inittab
```

command.
There we had to comment out the following line:

```
#Spawn a getty on Raspberry Pi serial line
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

and after we did that, it looked like:

```
#Spawn a getty on Raspberry Pi serial line
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

An interesting thing was, that this file did not exist on the Raspbian Jessie Lite image. There we had to change nothing.
The next step was to ensure if the Raspbian was up-to-date. At the beginning we did that with the

```
sudo apt-get update
```

and the

```
sudo apt-get dist-upgrade
```

command. Then we had to restart the RPi2. We waited, but the RPi2 did not boot. So to find out what's going on, we connected the RPi2 to a screen to look at the output. It showed us something like that all CPUs stopped and at the end there was a line that looked like this:

```
---[ end Kernel panic - not syncing: VFS: Unable to mount root
   fs on unknown-block(0,0)
```

So we looked up for another option and found out that we should use the

```
sudo rpi-update
```

command before rebooting. Searching for a fitting solution, we found many possibilities we had to try until we found the right one for our problem. So we started the whole process of ensuring if the Raspbian was up-to-date all over again. Actually we did this many times, for every possible solution we found.
So after we used the

```
sudo rpi-update
```

command and rebooted the RPi2, we could finally continue.
Now we had to download and configure the required packages. We had to use the pps-tools and the libcap-dev. We did this with the

```
sudo apt-get install pps-tools
```

and the

```
sudo apt-get install libcap-dev
```

commands.
After that we also had to install the GPS Daemon (GPSD) with the following command:

```
sudo apt-get install gpsd gpsd-clients python-gps
```

The GPSD presents us the data over a small server.
Before we could start the GPSD, we had to kill all the running GPSD's. We did that with the

```
sudo killall gpsd
```

command. Finally we could start it and use it with the

```
sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock -G
```

command.

GPS Data

We had to test if the GPS was working. There are many command line interfaces presenting you the data so you can see something.
There ist the

```
sudo cgps -s
```

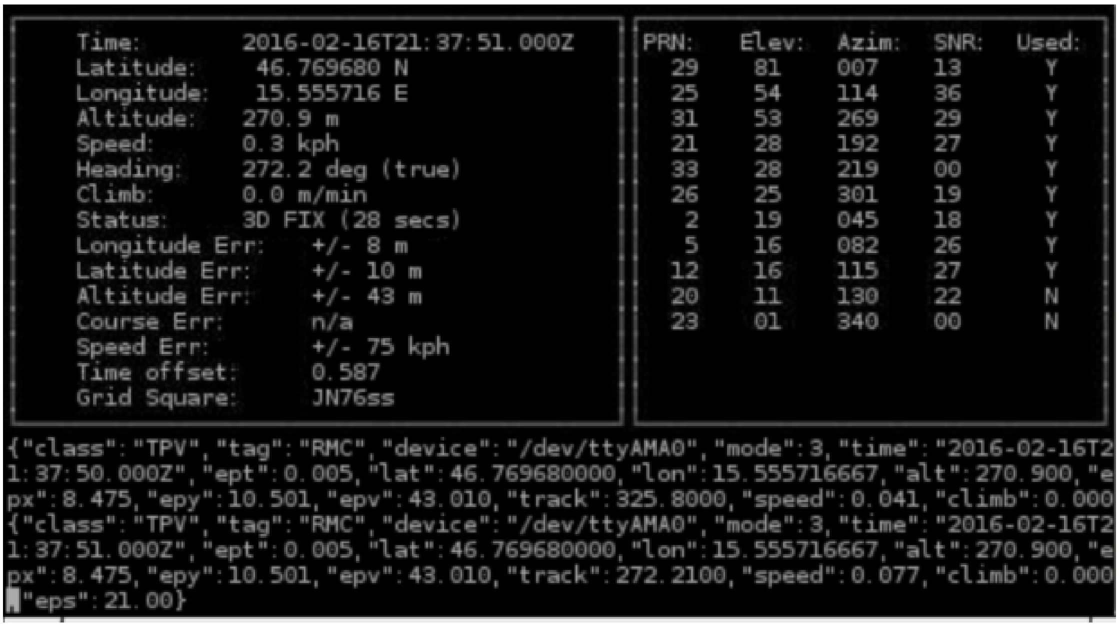
command that shows following output:

```
Time:          2016-02-16T21:20:00.000Z    PRN:   Elev:   Azim:   SNR:   Used:
Latitude:       46.769710 N                  29     83      302     13       Y
Longitude:      15.555699 E                  25     61      105     29       Y
Altitude:      269.1 m                       31     52      283     27       Y
Speed:         0.6 kph                      124     35      171     00       Y
Heading:       184.2 deg (true)               2     25      048     28       Y
Climb:         0.0 m/min                     12     23      110     25       Y
Status:        3D FIX (9 secs)               21     20      192     31       Y
Longitude Err:    +/- 8 m                    26     18      298     31       Y
Latitude Err:     +/- 16 m                    5     12      089     23       Y
Altitude Err:     +/- 17 m                   20     05      135     34       N
Course Err:       n/a                        14     03      230     00       N
Speed Err:        +/- 117 kph
Time offset:      0.605
Grid Square:      JN76ss
```

then the

```
sudo xgps
```

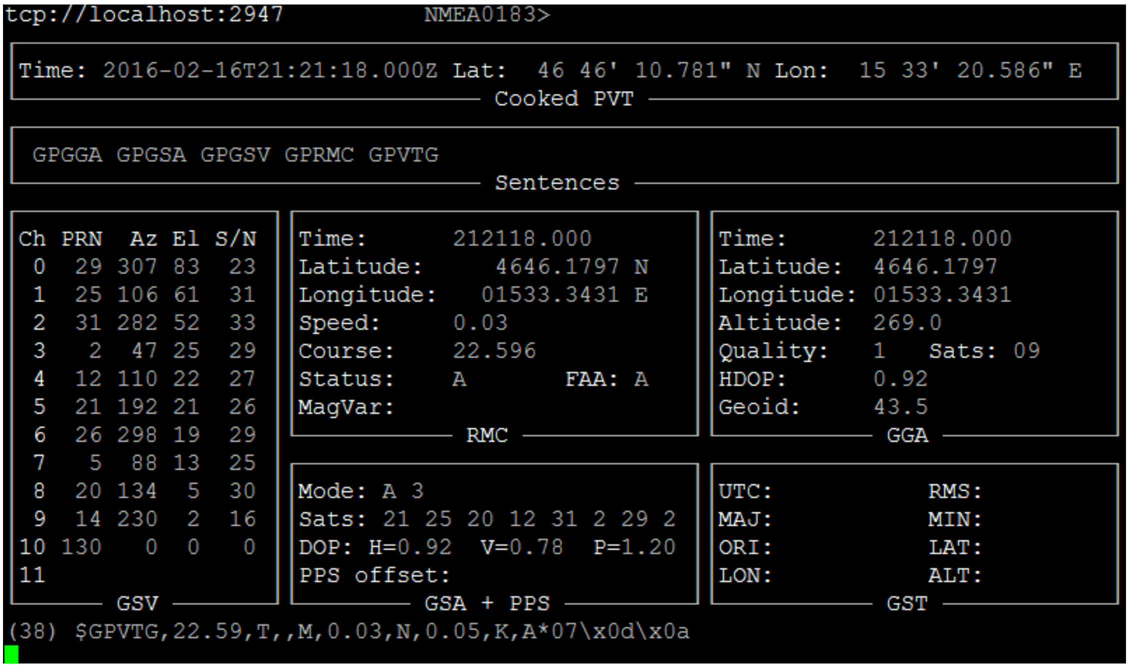
command where you can see this:



and at last the

```
sudo gpsmon
```

command to see following informations:



Then we wanted to save the GPS Data on the RPi2. We did this with the

```
gpspipe -r | grep '^\$G' | tee test.nmea
```

command. But since we could not use the National Marine Electronics Association (NMEA) format, we had to transform it into something like GPS Exchange Format (GPX) or Keyhole Markup Language (KML). For that we used the GPSBabel command line program. GPSBabel converts GPS data into other formats and saves it into a file. But firstly we had to install GPSBabel with the

```
sudo apt-get install gpsbabel
```

command. We converted it into both, KML and GPX and for that we had to use the two following commands. To get a KML file, we used:

```
gpsbabel -i nmea -f test.nmea -o kml -F test.kml
```

To get a GPX file, we used:

```
gpsbabel -i nmea -f test3.nmea -o gpx -F test3.gpx
```

Program

After we applied these commands, we started programming. We all know the programming language JAVA the best, so we decided to use it for our software.

We looked for an Application Programming Interface (API) for JAVA and found one and a small test program with it.
But we needed JAVA on the RPi2 to run a JAVA program. We wanted to use JAVA 8, but there was no real JAVA 8 Java Development Kit (JDK) for the RPi2, so we had to use JAVA 7. We installed it with the following command:

```
sudo apt-get install oracle-java7-jdk
```

You can read later everything about the program.

Auto boot

While developing we started our program mostly from our own notebooks, but in the final product we wanted it to start automatically after booting for further use. For the final product everything has to start after booting anyway. So we edited the /etc/rc.local file with the

```
sudo nano /etc/rc.local
```

command. At the last position we added the line

```
/home/pi/autostart.sh
```

This means, that the autostart.sh, that was created by us, will be called upon when the RPi2 starts. Then we created the autostart.sh with the

```
nano autostart.sh
```

command. In in this file we wrote:

```
#!/bin/sh

sudo killall gpsd
sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock -G
```

To make the script executable we had to use the following command:

```
chmod +x autostart.sh
```

Problems

Here we are going to explain some appeared problems we did not mention before.
Since we had a Wireless Local Area Network (WLAN) stick, we wanted the RPi2 to be connected with a Router. We found out, that the /etc/network/interfaces file must be edited. This was done with the

```
sudo nano /etc/network/interfaces
```

command.
At the beginning it looked like this:

```
auto lo
iface lo inet loopback

auto eth0
allow-hotplug eth0
iface eth0 inet dhcp

auto wlan0
allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

auto wlan1
allow-hotplug wlan1
iface wlan1 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Then we changed the WLAN interface.

```
auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ssid "linksys"
wpa-psk "raspberry"
```

This worked out quite well, but if the RPi2 started and there was no WLAN, the RPi2 did not accept Local Area Network (LAN) too.

4.1.5.2 RASPBIAN JESSIE LITE

Jessie Lite is a command line Operating System (OS). Our final product will run on this OS, because it's starting much faster than Raspbian Wheezy. Additionally, Raspbian Wheezy is no longer updated since 2015 and isn't able to be downloaded from the official site since February 2016.

Getting started

At the beginning we downloaded the Raspbian Jessie Lite image and wrote the .img file on the microSD Card. There was the first problem. The old Raspbian Wheezy image set the partition size to the maximum while booting. With Jessie Lite we had to do that on our own.

The next problem was that the ethernet interfaces IP was set to manual. Wheezy had set it to auto. So, if we want to have internet access from the beginning, we had to edit the interface file (/etc/network/interfaces) before booting the RPi2. Since we did it after the boot, too, we are going to describe how to do that later and there you can also see what we changed in this file.

The next step was to edit the boot file(/boot/cmdline.txt). When this was carried out, the GPSD received data from the pins. We changed the same at the ethernet interface later. In this file was no difference when we did it.

Here we had to remove

```
console=ttyAMA0,115200
```

Then it looked like this:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=$
```

After these applied changes we inserted the microSD Card into the RPi2 and plugged it in.

GPS Configuration

If we would not configure the ethernet interfaces IP to manual, we have to connect the RPi2 to a screen. Then we edited the interface file(/etc/network/interfaces). This can be done with the

```
sudo nano /etc/network/interfaces
```

command.
In this file there is a line iface eth0 inet manual and instead of this line we had to write:

```
auto eth0
allow-hotplug eth0
iface eth0 inet auto
```

Then we rebooted the RPi2 and accessed it over the LAN.
If we had not done it before, we would have edited the boot file(/boot/cmdline.txt) with the following command:

```
sudo nano /boot/cmdline.txt
```

There we removed console=ttyAMA0,115200 from the first line.
After that we checked if everything was up to date with the following two commands:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

Next we had to install different programs to get the GPS data.
First we installed pps tools with the

```
sudo apt-get install pps-tools
```

command, the libcap dev with the

```
sudo apt-get install libcap-dev
```

command, and then the GPSD with the

```
sudo apt-get install gpsd gpsd-clients python-gps
```

command.
Since our program is written in JAVA, we also had to install JAVA with the

```
sudo apt-get install oracle-java7-jdk
```

command.
Some other changes we had to apply on the Jessie Lite image, were editing the GPSD file. We edited it with this command:

```
sudo nano /etc/default/gpsd
```

Below you can see the whole file with our changes.

```
# Default settings for the gpsd init script and the hotplug wrapper.

# Start the gpsd daemon automatically at boot time
START_DAEMON="true"

# Use USB hotplugging to add new USB devices automatically to the daemon
USBAUTO="false"

# Devices gpsd should collect to at boot time.
# They need to be read/writeable, either by user gpsd or the group dialout.
DEVICES="/dev/ttyAMA0"

# Other options you want to pass to gpsd
GPSD_OPTIONS=""

# Additionally:
GPSD_SOCKET="/var/run/gpsd.sock"
```

Copy program on RPi2

Now we had to copy the program on the RPi2. We wanted to do that via Universal Serial Bus (USB) stick. But there was the next problem. With the old Wheezy image we just connected the USB stick. With the Jessie Lite image we had to mount the stick. We created a new folder called usb in the /mnt/ directory.

```
sudo mkdir usb
```

Then we had to mount the USB stick. We did this with the

```
sudo mount /dev/sda1 /mnt/usb/ -o uid=1000
```

command. If we wanted to unplug the USB stick we had to use the

```
sudo umount /mnt/usb
```

command.
Then had to copy the .tar.gz file to /opt with the

```
sudo cp gps.tar.gz /opt
```

command. To extract it, we used the

```
sudo tar xzvf gps.tar.gz
```

command.

Autostart(as Daemon)

We created a gps_rest.service script and put it into a folder near our program called systemd. Our script looks like this:

```
[Unit]
Description=GPS Rest Daemon PaClLaHell
[Service]
```

```
ExecStart=/sbin/start-stop-daemon --start --quiet --make-pidfile
--pidfile /var/run/gps_rest.pid --background --user pi --chuid pi
--chdir /opt/gps --exec /usr/bin/java -- -jar GPS_REST.jar
ExecStop=/sbin/start-stop-daemon --stop --quiet --pidfile
/var/run/gps_rest.pid --user pi --chuid pi --exec /usr/bin/java
Type=forking
[Install]
WantedBy=multi-user.target
```

Then in /etc/systemd/system/multi-user.target.wants we create a like too the gps_rest.service (*sudo ln -s /opt/gps/systemd/gps_rest.service*). After that we rebooted the RPi2 and started the GPS_REST.jar as a daemon. To start and stop the daemon, we used following commands:

```
sudo systemctl start gps_rest.service
sudo systemctl stop gps_rest.service
```

And to get information about the daemon we used this command:

```
sudo systemctl status gps_rest.service
```

UMTS Stick

Since the UMTS stick is not the newest, it was not that easy to make it working with the RPi2.

Configure the USB config

At first we had to configure the RPi2 so it would accept the stick as a modem.
We created a file in /opt/gps/udev and there we had to name this file 70-usb-modeswitch.rules.
In this file we had to add this line:

```
ACTION=="add", SUBSYSTEM=="usb",
ATTRS{idVendor}=="0e8d",
```

```
ATTRS{idProduct}=="0002",
RUN+="/usr/sbin/usb_modeswitch -v 0e8d -p 0002 -M
'55534243123456780000000000000006f00103000000000000000000000000000'"
```

Then we had to create a link to the right position, that was /opt/gps/udev/rules/. This has been done with the

```
sudo ln -s /opt/gps/udev/rules/70-usb-modeswitch.rules
```

command.
After that the following command showed us if the ID from the stick was right.

```
lsusb
```

Configuration of the provider

To make configurations, we had to change to the super user state, with the

```
sudo su
```

command.
After that we had to create two files with information for ppp.
We created the first file called hot_internet in /etc/chatscripts/.
In this file we had to put these lines:

```
TIMEOUT 10
ABORT 'BUSY'
ABORT 'NO ANSWER'
ABORT 'ERROR'
ABORT 'NO CARRIER'

'' 'ATZ'
'OK' 'ATE1'
'OK' 'AT+CGDCONT=1,"IP","webaut","0.0.0.0",0,0'
```

```
'OK' 'ATDT*99#'
'CONNECT' '\c'
```

The next file was also called hot_internet and created in /etc/ppp/peers/.
This file looked like this:

```
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/hot_internet"
debug
/dev/ttyUSB0
115200
defaultroute
replacedefaultroute
noipdefault
usepeerdns
crtscts
lock
local

# Redial and interval
persist
holdoff 5

# No compression
novj
novjccomp
nopcomp
nodeflate
```

```
# PAP authentication

# LCP echo messages settings
lcp-echo-failure 4
lcp-echo-interval 65535
```

After that we could start it with

```
pon hot_internet
```

and stop it with

```
poff hot_internet
```

Auto boot PPP

When pon/poff worked, we had to configure the /etc/network/interfaces and here we had to add the ppp interface as you can see in the following code lines:

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

```
allow-hotplug ppp0
auto ppp0
iface ppp0 inet ppp
provider hot_internet

allow-hotplug wlan0
iface wlan0 inet manualrp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

References: [16], [34], [20], [26], [37], [35], [15], [17], [28], [33], [9], [2], [12], [18], [29]

4.2 Software Description

The software of our RPi2 consists of two threads and a blocking FileObjectQueue. We get the GPS data from the API and upload it via REST in the DB.

4.2.1 Functions

First of all a logging property file and a gps property file are loaded. The logging property file contains logging information like the saving location and the maximum number of logging files. The gps property file contains the raspberry id, the server Uniform Resource Locator (URL) and the gps module IP.

4.2.2 GPS API

The API (Gpsd4java) manages the connection with the GPS daemon, that covers receiving data, parsing this information into an useful format and supplying it to the "API Thread". The GPS data we use consists of the timestamp, coordinateX, coordinateX-Error, coordinateY, coordinateYError, acceleration, altitude and altitudeError.

4.2.3 Tape API

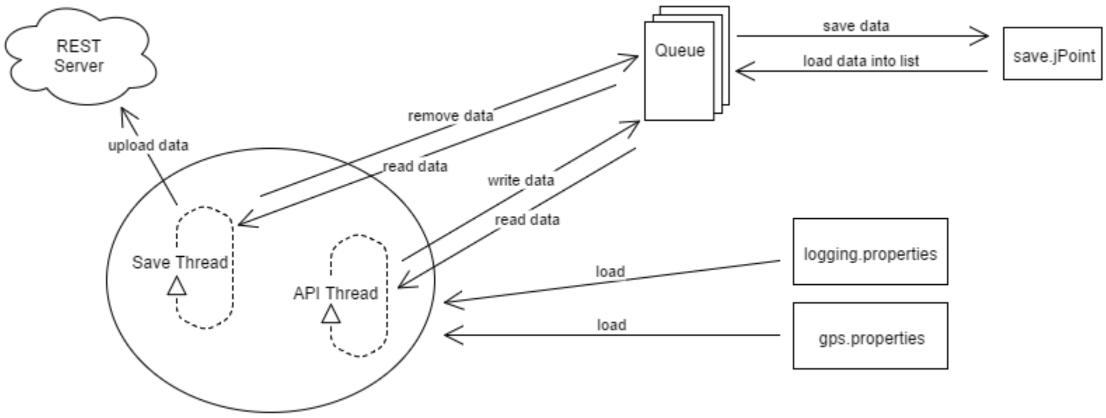
We implemented this API, because of the problem with saving data into an offline file. This API provides us a FileObjectQueue where we can put and remove our points and this FileObjectQueue is automatically saved in a file.

4.2.4 Save Thread

The Save thread reads point data from the FileObjectQueue and sends the data to the REST server. When the tracks are written in the database, the Save Thread removes these points from the FileObjectQueue.

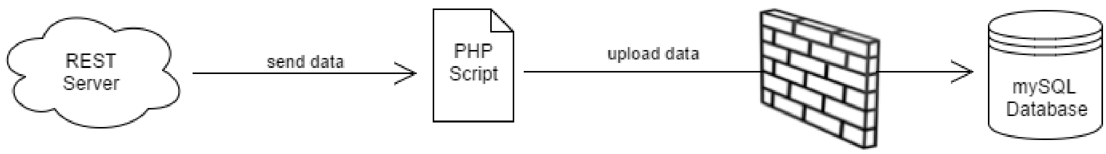
4.2.5 API Thread

When the GPS API measures GPS data, the API Thread gets it from the API. The API Thread adds the information about the track and loads it into the FileObjectQueue.



4.2.6 PHP (REST)

This part of our software is responsible for the upload into the database. The REST server gets the data formatted as JavaScript Object Notation (JSON).



4.3 Code Structure

4.3.1 Classes

4.3.1.1 GPS.java

Constructor

In the GPS.java class, the gpsModuleAddress variable is initialized and gets it is value from the GPSConfiguration.java class.

startGpsdClient

In the startGpsdClient method, the GPS API is initialized and the validate method from the BL.java class is called every time we get data from the GPS API.

4.3.1.2 BL.java

Constructor

In this class the FileObjectQueue is created and initialized by the save.jPoint file and the GsonConverter.java class. Then a new track is created and at last the SaveDataThread object is initialized and started.

Validate

Checks if the data from the GPS API is a Number if not it will be set to 0 except the latitude and the longitude. If they are Not a Number (NaN) it will do nothing. Then it adds the data from the GPS API to the FileObjectQueue.

ProcessData

This method peeks max 50 and min 15 points from the FileObjectQueue and puts them into a JContainer. Then it sends these points to the DataManager.java class to upload them via the uploadContainer method.

4.3.1.3 SaveDataThread.java

This is an intern Class in the Business Logic (BL) class that extends Thread.

Run

This override method calls the ProcessData method and then waits 1 second.

4.3.1.4 Data Manager.java

Constructor

Gets the server URL and puts it on the urlString variable.

UploadContainer

Uploads the JContainer it got, to the Server via REST

4.3.1.5 JContainer.java

This is a beans class that contains attributes, getter and setter methods for the respective attribute and a toString method of the JContainer.

4.3.1.6 JPoint.java

This is a beans class that contains attributes, getter and setter methods for the respective attribute and a toString method of the JPoint.

4.3.1.7 GPSConfiguration.java

At the beginning this class loads the logging properties.

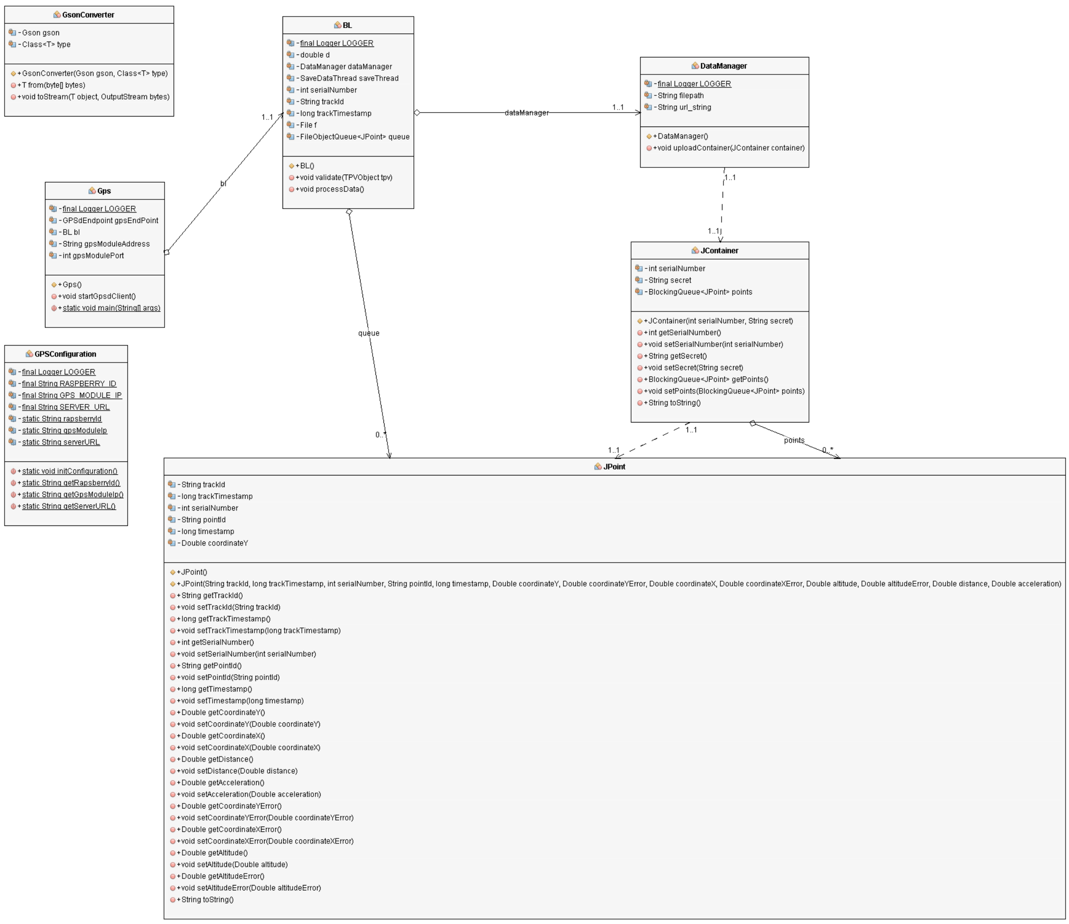
InitConfiguration

Reads the properties for the program from a file and sets these values on the specified variables. If a value is not in the file, it sets a default value. And at the end it closes the file.

4.3.1.8 GsonConverter.java

This class is a converter class that has implemented a FileObjectQueue Converter. In the method from, the bytes from the file are converted, so that they can be saved on the FileObjectQueue. In the toStream method, it is the other way round, so the FileObjectQueue is saved on a file.

4.3.2 Class Diagram



4.4 Problems

4.4.1 Uploading into Database

When we started with this project, we uploaded tracked data directly into the database over a standard java database connection. During our work on this project, we learned the technology hibernate in school so we reconstructed our program and used hibernate. Later our employer told us, that we should use REST to upload data for a higher security level and fewer connections to the database and therefore less overhead. We wrote the REST PHP script on our own.

4.4.2 Offline Saving

When the engine stopped while saving data offline, we had a loss of points, because the whole save file we used was overwritten. So we created a copy of this file, removed the data that was already in the database and deleted the original file. Then we renamed the copy to the original file name. Nevertheless, there was still a problem. On the one hand, it was possible that no data was lost, but on the other hand everything could be lost if the power was cut off between deleting the original file and renaming the copy. This is the reason why we use Tape API now.

4.5 Functional Testing

To ensure functionality of the hardware prototype and the software, we had to do several tests. They reach from logical problems to practical testing on the road.

4.5.1 Database Connection Error

While developing our software, the error handling when loosing database connection was a very important issue. We had to think about different solutions for different code structure because we changed the database connection type.

We decided on the connection type called REST. When using the connection that relies on a working connection, there had to be a backup plan if exactly this connection is not working anymore.

When does an connection error occur:

- internet not working
- database not working/running

We decided on saving points, that cannot be uploaded, into a JSON (save.jPoint). This solution proved to be the most fitting idea we thought of. It perfectly handled the described problems, reliably managed points and the saving mechanism worked phenomenal.

The same concept is used when the database is offline or not working properly.

4.5.2 Inaccuracy of the GPS Coordinates

After the first tests, where we tried to get GPS signal, we noticed the inaccuracy of the signal. This inaccuracy gets influenced by several factors. Therefore we had to find out how precise our GPS coordinates are and how much of an imprecision (part of the GPS data we get from our module) there is.

We evaluated driven tracks and measured how much distance there is between the actual location and the GPS coordinates received. This difference was about 5 metres, so no problem at all.

Due to the fact that we receive how much of an inaccuracy there is in the GPS data, our partner company can easily extract this information and compensate it in the visualization process. To do that even better, we also saved the deviation and the altitude.

4.5.3 No GPS Signal

When handling the GPS inaccuracy, we had to take into account that there could be no signal at all. Therefore we had to decide on a solution when receiving this data and how to handle and save it.

We tried different solutions, but finally decided on not saving them. It is the most efficient way concerning storage, both offline and online.

Meaning of not saving data when having no signal:

- when there is no GPS signal, there is often also no internet connection
- this leads to offline stored “null” points
  - not saving these null points is the best solution
- data extraction for our partner company when displaying the data on maps will be easier

References: [25], [13]

4.6 Future Work and Ideas

Due to our limitation in time, we cannot infinitely improve our program. Therefore we sum up our future ideas on what could be implemented or improved. Of course, we only write about the things that came up to our minds, and therefore this is just a small list.

Security between RPi2 and REST service

As we found out during the implementation process, REST supports an encryption process. After consulting the partner company, we decided on not using the encryption on the prototype.

Access via SSH over UMTS

It should be possible to access the RPi2 over the internet via Secure Shell (SSH) for remote maintenance.

Critical error reporting

If a severe problem happens, RPi2 should be able to report that problem so that it can be fixed.

Mobile Application/Webform

Using the wireframing of the webform and mobile devices, it is possible to create an access interface that can be accessed via a web browser or a mobile application.

Putting the GPS Points on the street

As shown in the Functional Testing part, the GPS data is rather inaccurate. Therefore our partner company has to interpolate and/or use the inaccuracy we add to the uploaded data. This task was not possible not accomplish in the given time, but could be implemented afterwards.

Compensating the inaccuracy will help to get an exact distance measurement and provide a better user experience.

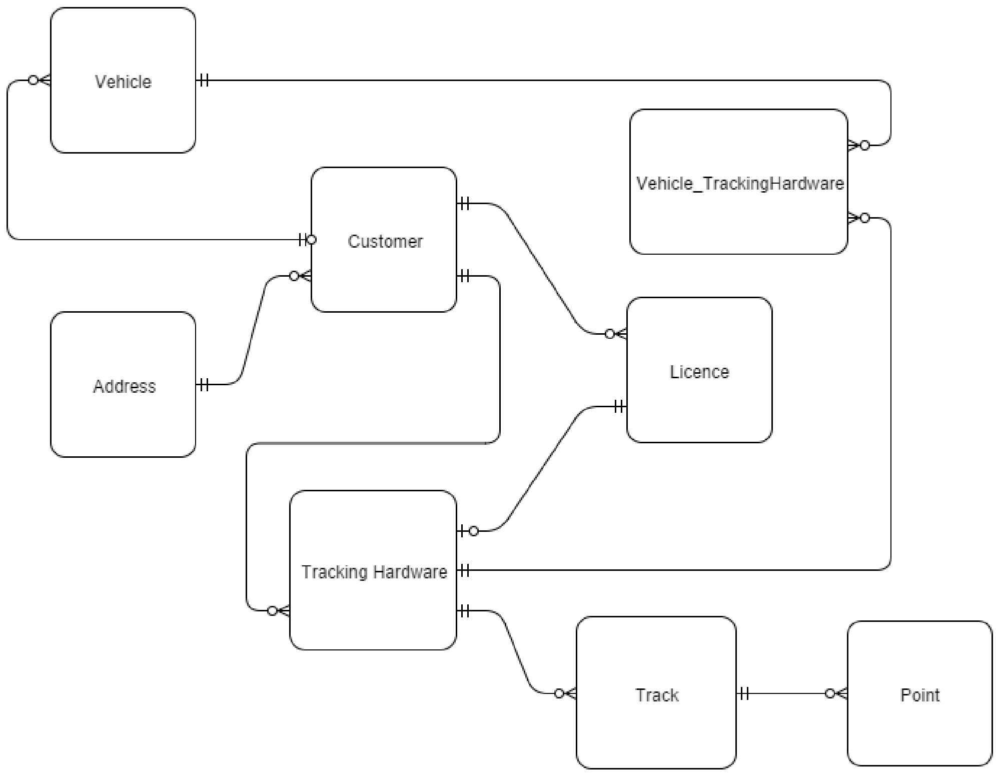
Optimize Http overhead

If there are many points on the RPi2 and the connection is established, the connection should only be closed after sending all the Points.

5 Database

One of our four main parts of our diploma thesis was the data design and DB. We started with creating an entity relationship diagram. This ERD was the base of our DB. Because of constant changes in our program structure, we had to change the DB structure too.

5.1 Entity Relationship Diagram



Customer(svnr, firstname, lastname, dateofbirth, addressIDFK)
Vehicle(vehicleID, svnrFK)
Address(addressID, street, zipcode)
TrackingHardware(serialNumber, svnrFK, description, licenceIDFK)
Licence(licenceID, svnrFK, dateofactivation)
Track(id, serialNumberFK, timestamp)
Point(id, timestamp, (track_id,track_serialNumber)FK ,coordinateX, coordinateXError, coordinateY, coordinateYError, acceleration, altitude, altitudeError, distance)
Vehicle_TrackingHardware(serialNumberFK, vehicleIDFK)

5.2 Entities

5.2.1 Customer

The entity Customer has the social security number as the primary key. Additional attributes are the firstname, the lastname and the date of birth of the person the account belongs to. As you can see in the ERD, every customer has one reported address and an address can belong to zero or more customers. Due to this, the entity customer contains a foreign key with the id from the address. A customer can only register once, so every customer can have zero or one AdministrationAccount, depending on if he is already registered. Clearly, an account can only belong to one customer so it is well defined who is the owner of the device. There can be a customer who drives more than one vehicle so if he changes to another car, of course he does not have to buy a new device. He can use it for zero or more vehicles. The vehicle is only reported for one customer. This is because if the car is used by more than one person they will not have more than one account. Even if a customer usually only needs one device, it is possible for him to buy zero or more devices. And clearly he does not have to create a new administration account for each of them. But the other way around, every tracking hardware belongs to only one customer.

5.2.2 Vehicle

Every Vehicle has its own unique vehicle id to identify it. This id is also used in our DB to identify different vehicles. The vehicle is only reported for one customer. This is because if the car is used by more than one person they will not have more than one account. In one vehicle you can place zero or more tracking hardwares and one tracking hardware can be used in zero or more vehicles too, so this is called a “n to m relationship”. We splitted this relationship to an extra entity called Vehicle_TrackingHardware. This entity contains the two primary keys of the first two tables as primary and foreign key at once.

5.2.3 Address

The entity Address has the unique address id for primary key to identify. It also contains the street name and the zipcode of each address. As you can see in the diagram, one address can belong to zero or more customers but one customer can only be reported to one address.

5.2.4 TrackingHardware

In our DB, every TrackingHardware is identified by its unique serial number. An additional attribute in this table is the description of the device. One hardware can only belong to one customer and one customer can own zero or more tracking hardwares. So we had to create a foreign key called svnr of the social security number of the Customer. Because of the already mentioned “n to m relationship” between TrackingHardware and Vehicle, you can see an extra entity called Vehicle_TrackingHardware. With every purchase on a tracking hardware, you acquire an associated Licence. So this table includes the primary key of the Licence called licenceID as a foreign key. A Licence can belong to zero or one device. If the RPi2 is not purchased yet, the licence is not used. One TrackingHardware has already tracked zero or more Tracks and one Track always belongs to exactly one device.

5.2.5 Licence

The table Licence consists of its own unique licenceID. The svnr is the connection to the Customer Table and is a foreign key in this table. The Licence belongs to one Customer and the Customer can have zero and more Licences. The dateofactivation attribute is, as the names says, the date when the licence was activated on the web portal with the associated device.

5.2.6 Track

Every Track belongs to one TrackingHardware. This Track contains all the Points which belong to this Track. The identifier for the Track is a Universally Unique Identifier (UUID), generated by the program on the RPi2 and the serialNumber from the TrackingHardware. The table also has a timestamp generated at the start of the program on the RPi2.

5.2.7 Point

The greatest part of the information is stored in the Point table. There the identifier is also a UUID, generated by the program on the RPi2. The primary key is compound of the id, and the primary key from the Track table which is also the foreign key in the Point Table. The identifier from Track is composed of the tack_id and the serialNumber. The other attributes are the coordinates, the variance of the coordinates, the acceleration, the altitude and the variance of the altitude. The last thing is the distance, which is empty at the time being. Every Point belongs to one Track and every Track contains zero or more points.

5.3 Data Definition Language

```
CREATE TABLE Address(
addressID INT(6) AUTO_INCREMENT PRIMARY KEY,
```

```sql
street VARCHAR(30),
zipcode INT(6)
);
CREATE TABLE Customer(
svnr INT(6) PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50) NOT NULL,
dateofbirth DATE NOT NULL,
addressID INT(6),
FOREIGN KEY fk_Customer(addressID)
REFERENCES Address(addressID)
);

CREATE TABLE Vehicle(
vehicleID INT(8) AUTO_INCREMENT PRIMARY KEY,
svnr INT(6),
FOREIGN KEY fk_Vehicle(svnr)
REFERENCES Customer(svnr)
);

CREATE TABLE Licence(
licenceID INT(8) AUTO_INCREMENT PRIMARY KEY,
svnr INT(6),
dateofactivation DATE NOT NULL,
FOREIGN KEY fk_account (accountID)
REFERENCES AdministrationAccount(accountID)
);
```

```
CREATE TABLE TrackingHardware(
serialNumber INT(8) AUTO_INCREMENT PRIMARY KEY,
svnr INT(6),
description VARCHAR(100),
licenceID INT(8),
FOREIGN KEY fk_trackHardw(svnr)
REFERENCES Customer(svnr),
FOREIGN KEY fk_licence(licenceID)
REFERENCES Licence(licenceID)
);

CREATE TABLE VehicleTrackingHardware(
serialNumber INT(8),
vehicleID INT(8),
timestamp DATE NOT NULL,
PRIMARY KEY (serialNumber, vehicleID),
FOREIGN KEY fk_vth_serialNumber(serialNumber)
REFERENCES TrackingHardware(serialNumber),
FOREIGN KEY fk_vth_vehicleID(vehicleID)
REFERENCES Vehicle(vehicleID)
);

CREATE TABLE Track(
id VARCHAR(255),
serialNumber INT(8),
timestamp BIGINT NOT NULL,
PRIMARY KEY(id, serialNumber),
FOREIGN KEY fk_track_serial(serialNumber)
REFERENCES TrackingHardware(serialNumber)
```

```
);

CREATE TABLE Point(
id VARCHAR(255),
timestamp BIGINT NOT NULL,
track_serialNumber INT(8),
track_id VARCHAR(255),
coordinateY DOUBLE NOT NULL,
coordinateYError DOUBLE NOT NULL,
coordinateX DOUBLE NOT NULL,
coordinateXError DOUBLE NOT NULL,
acceleration DOUBLE NOT NULL,
altitude DOUBLE NOT NULL,
altitudeError DOUBLE NOT NULL,
distance DOUBLE,
PRIMARY KEY(id, timestamp, track_serialNumber,track_id),
FOREIGN KEY fk_point_track(track_serialNumber, track_id)
REFERENCES Track(serialNumber, id)
);
```

5.4 Data Manipulation Language

```
INSERT INTO Address(addressID, street, zipcode)
VALUES(1,'Blumenweg',8430);
INSERT INTO Customer(svnr, firstname, lastname, email, dateofbirth, addressID)
VALUES(4523, 'Karl', 'Huber', 'karl.huber@gmail.com',
DATE_FORMAT('1970-12-09','%Y-%m-%d'),1);
INSERT INTO Vehicle(vehicleID, svnr)
VALUES(832,4523);
```

```
INSERT INTO Licence(licenceID, svnr, dateofactivation)
VALUES(832495,4523,DATE_FORMAT('2016-01-20','%Y-%m-%d'));
INSERT INTO TrackingHardware(serialNumber, description, licenceID)
VALUES(1234, 'Karlis Raspi', 832495);
INSERT INTO VehicleTrackingHardware(serialNumber, vehicleID, timestamp)
VALUES(1234, 832, DATE_FORMAT('2016-01-24','%Y-%m-%d'));
INSERT INTO Track ('id', 'serialNumber', 'timestamp')
VALUES('018f8ff5-395b-42d2-b2f6-5c976292cd5e', 1234, 1455535245172);
INSERT INTO Point ('id', 'timestamp', 'track_serialNumber', 'track_id',
'coordinateY', 'coordinateYError', 'coordinateX', 'coordinateXError',
'acceleration', 'altitude', 'altitudeError', 'distance')
VALUES('007dae03-c915-4aae-baf9-2b7c42a41e43', 1455535245172, 1234,
'1f114f58-fd0c-4da6-a1b6-571ae9e79748', 46.78029, 14.815, 15.54794, 10.697,
0.103, 277.3, 18.17, 0),
('dedc843e-e29b-4a0d-b32d-20cb718088ea', 1455535282394, 1234,
'1f114f58-fd0c-4da6-a1b6-571ae9e79748', 46.780401667, 14.815, 15.547526667,
10.697, 7.562, 279, 16.56, 0);
```

5.5 Problems

While we tried to insert testing data into our DB, we noticed that it wasn't possible to save the timestamp as milliseconds. After researches, we discovered that the problem was the version of our MySQL DB (version 5.5.46). Due to this problem we had to change the datatype of our timestamp into bigint.

6 Wireframing

6.1 Smartphone

The design for the smartphone consists of four different pages.

6.1.1 Login

The login dialogue is the first page you see. There are two input fields for the username and the password. Below these input fields there is the login button to confirm and login. Then there is a radio button to stay logged in.

The link if the password has been forgotten and a button to register and create a new user account are the last things on this page.

6.1.2 Menu

The menu pops up from the left edge as you can see on the picture. It can be chosen between viewing the tracks, customize settings, view statistics and logout.

6.1.3 Tracks

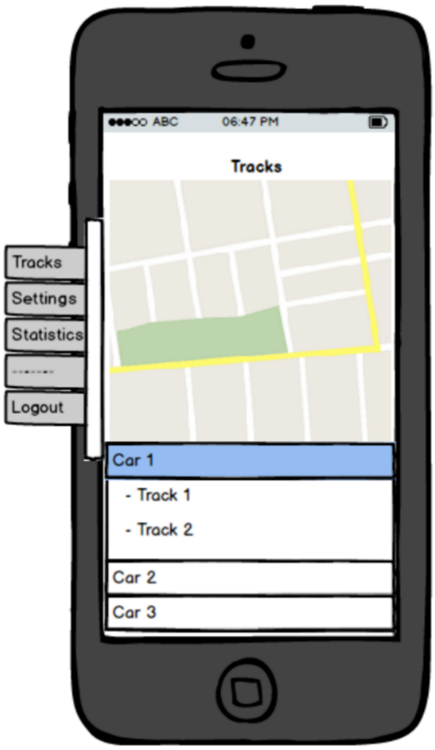
On the tracks page a map shows the last track by default but on the bottom any available track can be chosen to view it.

6.1.4 Settings

Under settings, basically changes like edit the mail address or change the password of the account can be applied, but also if the default track type should be private or public.

6.1.5 Statistics

It's possible to view a graph where informations of driven tracks are shown. Every registered car of this account can be chosen to view it's graph and additionally, it's selectable if the displayed data was tracked the last week, month or year.



6.2 Tablet

The wireframing we created for the Tablet design is nearly similar to the smartphone design. Except of view extensions such as for renaming the device and editing tracks more precise.

6.2.1 Login

The login dialogue is the first page you see. There are two input fields for the username and the password. Below these input fields there is the login button to confirm and login. Then there is a radio button to stay logged in. The link if the password has been forgotten and a button to register and create a new user account are the last things on this page.

6.2.2 Menu

The menu pops up from the left edge as you can see on the picture. It can be chosen between viewing the tracks, customize settings, view statistics, rename the device, register a car and logout.

6.2.3 Tracks

On the tracks page a map shows the last track by default. A track can be selected and there is the option to edit or merge a route.

6.2.4 Settings

Under settings, basically changes like edit the mail address or change the password of the account can be applied, but also if the default track type should be private or public.

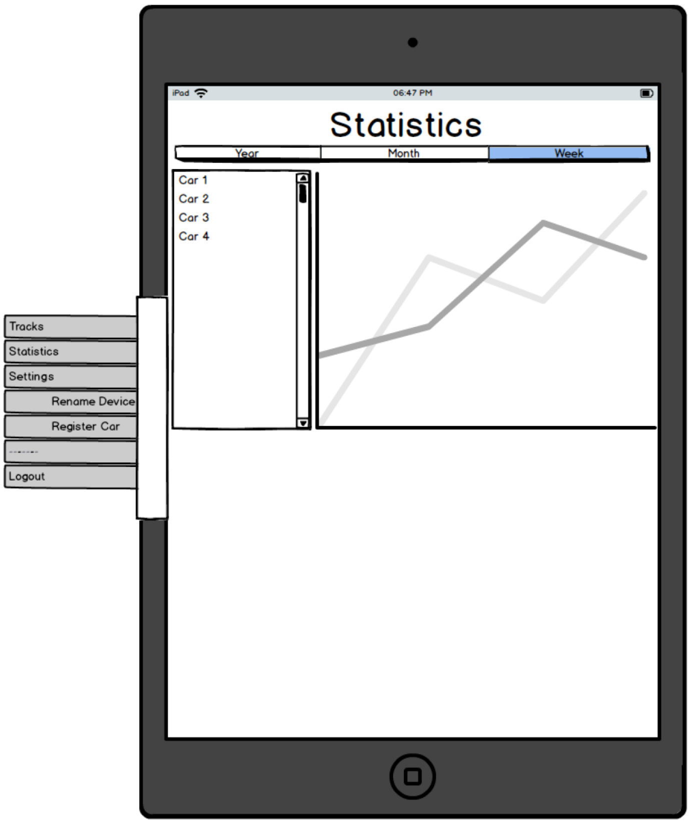
6.2.5 Rename Device

Every device has a name and to know better which device is which, it is possible to give the devices names.

6.2.6 Register Car

6.2.7 Statistics

It's possible to view a graph where informations of driven tracks are shown. Every registered car of this account can be chosen to view it's graph and additionally, it's selectable if the displayed data was tracked the last week, month or year.



6.3 Web Portal

6.3.1 Home

Before the user is logged in, information about the product is shown on this page.

After the login, all tracks are shown and it is possible to change the tracks from private to public.

6.3.2 Login

The login dialogue is the first page you see. There are two input fields for the username and the password. Below these input fields there is the login button to confirm and login. Then there is a radio button to stay logged in. The link if the password has been forgotten and a button to register and create a new user account are the last things on this page.

6.3.3 Settings

Under settings, basically changes like edit the mail address or change the password of the account can be applied, but also if the default track type should be private or public.

6.3.3.1 Register Car

6.3.3.2 Rename Device

Every device has a name and to know better which device is which, it is possible to give the devices names.

6.3.4 Track

On the tracks page a map shows the last track by default. It can be chosen between the cars and the tracks of the respective cars can be viewed.

6.3.4.1 Edit

If a track is selected, there is the possibility to edit it. If the route isn't complete or must be changed in any way, it can be done over the web portal. Should it happen that one track is splitted up into two or more tracks because of complications, you can merge them into one whole route.

6.3.4.2 Create

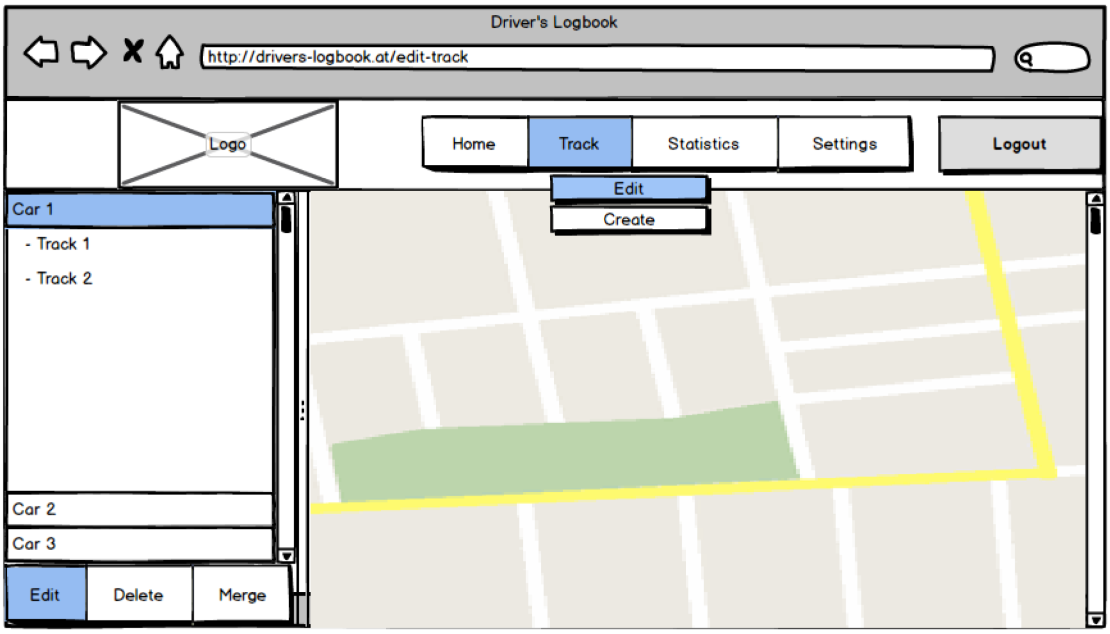
Here a new track can be created manually over the web portal and the route of the track can be chosen. It's possible to set the start and end position of the new track and changing the route by pulling the line to the desired way.

6.3.5 Statistics

It's possible to view a graph where informations of driven tracks are shown. Every registered car of this account can be chosen to view it's graph and additionally, it's selectable if the displayed data was tracked the last week, month or year.

6.3.6 Menu

The menu from the homepage contains home, tracks, statistics, settings and login. The menu item tracks contains subitems called edit and create. The subitems of settings are register car and rename device.


Driver's Logbook
http://drivers-logbook.at/edit-track
Logo
Home
Track
Statistics
Settings
Logout
Edit
Create
Car 1
- Track 1
- Track 2
Car 2
Car 3
Edit
Delete
Merge

7 Optional Uses

The basic concept of our product was to track driven distances of a car you use for private and business purposes. Nevertheless, we thought about other possible uses for this product.

7.1 Taxi

One of our ideas for optional uses was for a taxi company. Our product could be placed in the taxi, the taxi driver presses a button if a new customer enters the car and the device calculates the price of the taxi ride based on the distance, the velocity and the fix basic costs.

7.2 Car Renting

A car renting company usually gets paid for renting their cars per day. Our product could help them to control the use of their cars. Our product could track the distances the customer took so the company could calculate a suitable price.

7.3 Train Controller

We thought about the optional use as a train controller for checking the current location of a train, check for complications and preventing delays. It would be also possible planning ahead to avoid incidents and more delays.

7.4 Mowing Machine

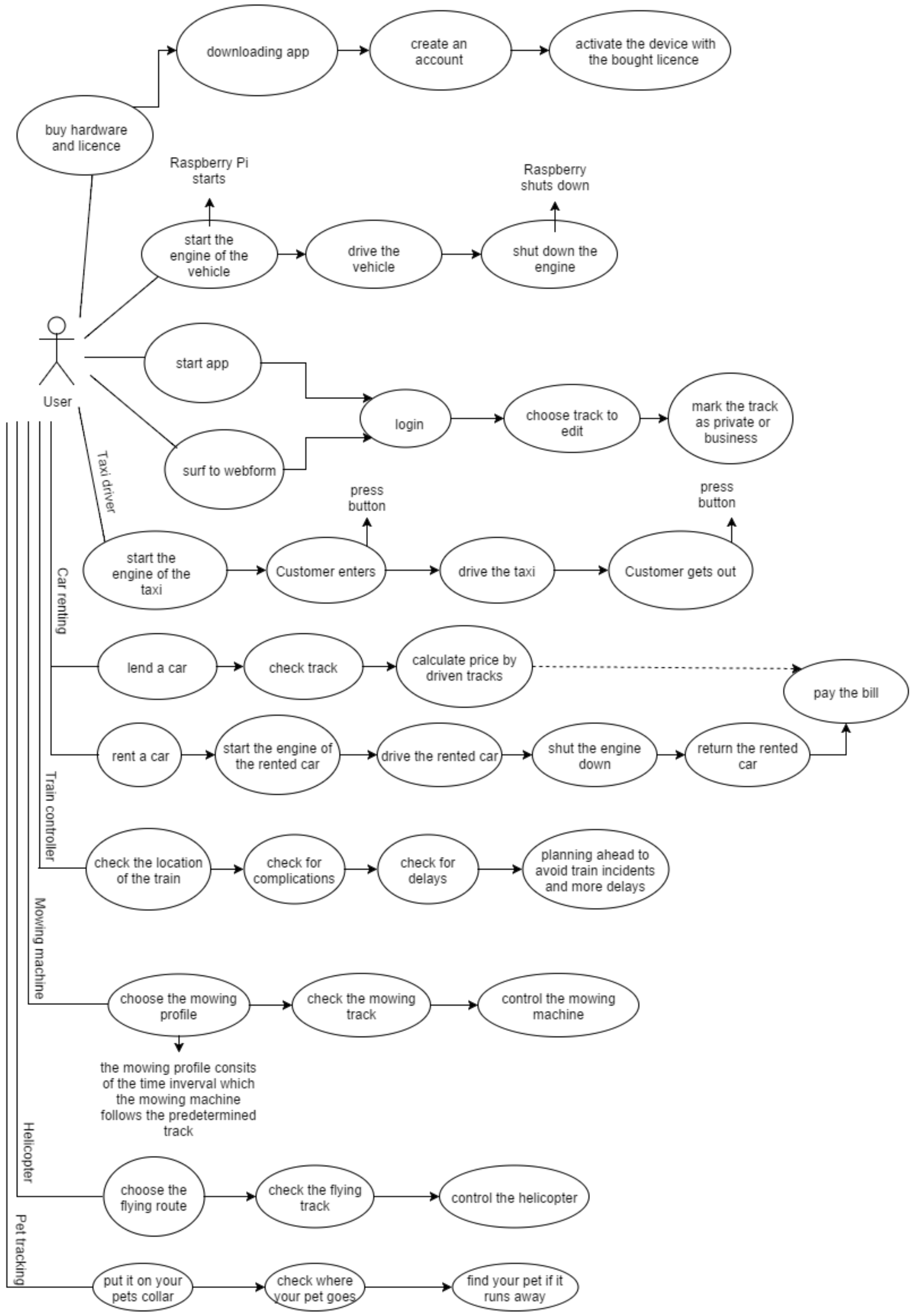
There are already mowing machines that can drive around in your garden on their own and they are already able to discover the shape of your ground. Using our product, you could extend the conventional device so you can choose between predetermined tracks that your mowing machine could take. It would also be possible to control your mowing machine over a mobile application to change the direction or stop mowing.

7.5 Helicopter

Another optional use for our product could be a controlling device for a helicopter. Choosing a flying track or checking the position and the track of the helicopter could be possible.

7.6 Pet Tracking

Another possibility for our product is to power the device with an battery pack and use it to track your pet. For example if you have a dog or a cat, you can put it on their collar for example and watch where it is and where it goes.


downloading app
create an account
activate the device with the bought licence
buy hardware and licence
Raspberry Pi starts
Raspberry shuts down
start the engine of the vehicle
drive the vehicle
shut down the engine
User
start app
login
choose track to edit
mark the track as private or business
surf to webform
Taxi driver
press button
press button
start the engine of the taxi
Customer enters
drive the taxi
Customer gets out
Car renting
lend a car
check track
calculate price by driven tracks
pay the bill
rent a car
start the engine of the rented car
drive the rented car
shut the engine down
return the rented car
Train controller
check the location of the train
check for complications
check for delays
planning ahead to avoid train incidents and more delays
Mowing machine
choose the mowing profile
check the mowing track
control the mowing machine
the mowing profile consits of the time inverval which the mowing machine follows the predetermined track
Helicopter
choose the flying route
check the flying track
control the helicopter
Pet tracking
put it on your pets collar
check where your pet goes
find your pet if it runs away

8 User Manual

8.1 General

First of all, thank you for buying our product. With your purchase on our product, you acquired a licence. Now you can register on our web portal and create your own account. Using your purchased licence, you can activate the product and connect it with your user account. You can also transfer the product with the related licence to another user account. Over the web portal, you can log in and view tracks, visualize them and eventually edit them.

8.2 Installation Regulations

The product has to be placed in the car and connected with the cigarette lighter receptacle to get power. When starting the engine of the vehicle, the device powers itself up and starts searching for a GPS signal.

8.3 LED

When you connect the product to a power source you will see some Light-emitting Diode (LED)s light up. There are three LEDs. A red one on top of the RPi2, which is for GPS module and two on the opposite side of the USB ports that are red and green. When the red LED next to the green one is on, the RPi2 gets power. The green LED on the RPi2 lights up when the device boots and prepares itself for tracking. When it is ready and it receives data it flickers quick. And if the red LED on the RPi2 GPS module blinks once per second, it has a GPS signal

8.4 Warnings

- This product shall only be connected to an external power supply rated at 5V dc, and a minimum current of 500-700 milliampere (mA) for model A and 700-1200mA for model B. Any external power supply used with the RPi2 shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should not be overclocked as this may make certain components very hot.
- This product should be operated in a well ventilated environment and the case should not be covered.
- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- The connection of unapproved devices to the GPIO connector may affect compliance or result in damage to the unit and invalidate the warranty.
- All peripherals used with the RPi2 should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors, and mice used in conjunction with the RPi2.
- Where peripherals are connected that do not include the cable or connector, the used cable or connector has to offer adequate insulation and operation in order that the requirements of the relevant performance and safety requirements are met.

References: [36]

To avoid malfunction or damage to your device please observe following:

- Do not expose it to water, moisture or place on a conductive surface whilst in operation.
- Do not expose it to heat from any source; the RPi2 is designed for reliable operation at normal ambient room temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling the printed circuit board while it is powered. Only handle by the edges to minimise the risk of electrostatic discharge damage.
- The RPi2 is not designed to be powered from a USB port of the other connected equipment. If this is attempted, it may malfunction.

9 Tools and Technologies

9.1 Java

We decided on using JAVA, because it runs on every platform, is easy to implement and our partner company could use it on Android devices too. Also the fact that every member of the group got 5 years of programming experience in this language shows that it is the most fitting choice for us.

9.2 TAPE API

We implemented this API, because of the problem with saving data into an offline file. This API provides us a FileObjectQueue where we can put and remove our points. This FileObjectQueue is automatically saved in a file.

9.3 GPS API

This API was rather hard to find because there are few possible ways of handling the GPS data from an RPi2 in JAVA. Nevertheless, we read through the excellent documentation and found ways to use this API in an easy way.

9.4 RASPBIAN

As we tried out different things in linux on the Raspbian Wheezy during our NVSU lessons, it showed its potential. Another point why we used it, was that it was constructed to work perfectly with the RPi2. Therefore, our first choice was this image for our RPi2. Later we found out, that Raspbian Wheezy was not supported any more and so we had to switch to Raspbian Jessie Lite. Raspbian Jessie Lite is only a command line OS, so it is perfect for our project.

9.5 GIT

When working in a group, a good collaboration tool is needed. Our choice was GIT because we only made negative experiences.

Unfortunately, we had problems at the beginning of our software development. Everytime we tried to "PULL" committed changes, we encountered a merge conflict. This conflict somehow followed us through the whole development phase.

At the end, we also used GIT for LaTeX cooperation. Due to the usage of the GIT console, it was a very pleasant experience.

9.6 Google Drive

At the beginning and almost through the whole project, we used Google Drive for document management, time reporting and organization.

The main Google products we have used were Google Docs and Google Tables.

Collaborating was the main reason why we used especially this technologies. It worked trouble-free and had option to restore older versions of documents, which came in handy from time to time.

9.7 PHP

Originally it stood for Personal Home Page, but now for recursive backronym. It was designed for web development, but also used as programming language.

The usage of PHP is based on REST. It's functioning kind of like a server, which connects to the DB, uploads the GPS data and disconnects.

9.8 BalsamiQ

We used this tool to provide our wireframing plans and mock ups, since it is a rapid wireframing tool. Mock-ups make it simple to generate ideas for designs and to pick the best one. It was provided by our employer because they have made a very good experience when using it.

9.9 MySQL

Our employers standard as database technology was MySQL. The open-source Relational Database Management System (RDBMS) is one of the most used client-server model RDBMS. It is a popular choice of a database for use in a web application in general. Due to our experience we got in our DBI lessons, we encountered no problems.

9.10 Netbeans

The programming Integrated Development Environment (IDE) of our choice was Netbeans. It is a software development platfrom written in Java and that is also the programming language we used at our work. Over 4 years of experience in the same working environment, it payed off when working on a new, more complex set of problems on our own. These problems reached from logical errors to migrations of completely new programme structures.

9.11 Notepad++

This program is a text editor for Windows that supports several languages. Since it is a portable app, you can do your work at all places as well. It provides features like tabbed documentation document interface or a FTP (File Transfer Protocol) Browser. We used it to write the code for the web interface as we created PHP files. Furthermore, we could look in the scripts of our database to check the DDL (Data Definition Language) or the DML (Data Manipulation Language) of our database.

9.12 LaTeX and Texmaker

We used the word processor and document markup language LaTex to make our diploma-thesis look more professional. In this program it is simpler to format the output and give it a nice shape compared to other document preparation systems. To create existing files of our diploma-thesis, we needed the tool Texmaker as well. This program is a cross-platform LaTex editor that integrates many tools needed to develop documents with LaTex. Without it, we would had had to recreate the whole work.

9.13 draw.io

It is provided by Google to draw different kinds of Unified Modeling Language (UML) diagrams, flowcharts, process diagrams and so on. Because it is on drive, sharing and working on diagrams with the other users gets pretty simple and the day time does not matter anymore. So, as we were designing our database, we used it to create the ERD model for our database with all its entities and relations. Moreover, it got used to draw the Use Case and any other kind of diagrams we needed.

9.14 XAMPP

XAMPP makes it easy to create a local web server, since everything you need is setup for making a web server like a server application, database and PHP as scripting language. It is an easy to install Apache distribution containing MySQL, PHP, Perl and some other features. This means it works equally well on Linux, Mac and Windows. XAMPP is downloaded in a zip file. There is only little or no configuration required to make up the web server. It is regularly updated to incorporate the latest release of Apache, MariaDB, PHP and Perl. Furthermore, it contains modules like OpenSSL, phpMyAdmin or WordPress.
Over XAMPP it was possible to start pgAdmin and the SQL database for our diploma-thesis.

9.15 phpMyAdmin

To handle the administration of MySQL and our database with the use of a web browser, we used phpMyAdmin. It is an open source tool written in PHP and can perform various tasks such as creating, modifying and deleting all around database. Furthermore, you can execute Structured Query Language (SQL) statements with it and manage users and permissions.

9.16 Hangouts

Because it is really hard to meet all the time in person during weekends or holidays, we used Hangout to talk and discuss our diploma-thesis. It made the communication a lot easier for us and helped us to productively do our work. Meetings could also be held simple at any time of the day.

Glossary

API Application Programming Interface. 57, 69, 70, 100

BL Business Logic. 72

CEO Chief Executive Officer. 14, 19

CSV Comma-separated values. 26–29, 31, 32, 34, 35, 37, 38, 40, 41, 43

DB Database. 2, 5, 13, 14, 16, 17, 22, 23, 38, 40, 41, 49, 69, 79, 81, 86, 103

DDL Data Definition Language. 17

DML Data Manipulation Language. 17

ERD Entity Relationship Diagram. 5, 17, 79, 80, 105

Gbit/s Gigabit per second. 22

GPIO General Purpose Input Output. 48, 51, 98

GPS Global Positioning System. 2, 4, 14–17, 26, 28–31, 33–35, 37, 39–43, 45, 48, 51, 54, 56, 61, 69, 70, 75–77, 97, 98, 100, 103

GPSD GPS Daemon. 53, 60–62

GPX GPS Exchange Format. 56

HTBLA Höhere Technische Bundeslehranstalt. 3, 6, 13

HTML Hyper Text Markup Language. 31, 32

IDE Integrated Development Environment. 104

iOS iPhone Operating System. 26–30, 34–39, 45, 46

IP Internet Protocol. 51, 60, 61, 69

JDK Java Development Kit. 57

JSON JavaScript Object Notation. 70, 75

KML Keyhole Markup Language. 56

LAN Local Area Network. 59, 61

LED Light-emitting Diode. 97, 98

mA milliampere. 98

MB MegaByte. 40

MySQL My Sequential Query Language. 23, 86, 103, 105, 106

NaN Not a Number. 71

NMEA National Marine Electronics Association. 56

OS Operating System. 60, 101

PDF Portable Document Format. 26–29, 33–35, 37, 38, 40, 41, 43

PHP PHP Hypertext Preprocessor. 22, 74, 103–106

RDBMS Relational Database Management System. 103

REST Representational State Transfer. 15, 17, 22, 69, 70, 72, 74, 75, 77, 103

RPi2 Raspberry Pi 2. 2, 4, 15–17, 49, 51–53, 56–61, 63, 64, 69, 77, 78, 81, 82, 97–101

SQL Structured Query Language. 106

SSH Secure Shell. 77

TightVNC Tight Virtual Network Computing. 51

UART Universal Asynchronous Receiver Transmitter. 51

UML Unified Modeling Language. 105

UMTS Universal Mobile Telecommunications System. 49, 64

UPS Uninterruptible Power Supply. 4, 49

URL Uniform Resource Locator. 69, 72

USB Universal Serial Bus. 63, 97, 99

UUID Universally Unique Identifier. 82

WLAN Wireless Local Area Network. 58, 59

XML Extensible Markup Language. 32, 35

Bibliography

[1] *Abax Triplog*. 2016. URL: http://abax.co.uk/electronic-triplog/ (visited on 05/01/2015).

[2] Jonathan Adams. *Raspberry Pi GPS and PPS NTP Server*. 2016. URL: https://jonathan-adams.co.uk/raspberry-pi-gps-and-pps-ntp-server/ (visited on 05/01/2015).

[3] *Carpanion*. 2016. URL: http://www.carpanion.eu/de/elektronisches-fahrtenbuch/fuhrparkmanagement/ (visited on 05/01/2015).

[4] *Drives Fahrtenbuch*. 2016. URL: http://gratis-app.com/de/apps/fahrtenbuch-app-iphone (visited on 05/01/2015).

[5] *Electronic Logbook from TOMTOM*. http://business.tomtom.com/de_de/landingpages/fahrtenbuch/c/ ; http://business.tomtom.com/de_de/landingpages/mobile-solutions/b/. 2016. (Visited on 05/01/2015).

[6] *Fahrtenbuch Mileage Book*. https://play.google.com/store/apps/details?id=com.AppzStudio.MileageBook.Android ; http://www.mileagebook.net/. 2016. (Visited on 05/01/2015).

[7] *Fahrtenbuch (myLogbook)*. http://android.pdassi.de/125071/Fahrtenbuch_myLogbook.html ; https://play.google.com/store/apps/details?id=com.Mag.myLogBook&hl=de. 2016. (Visited on 05/01/2015).

[8] *Fahrtenbuch (von Stefan Meyer)*. 2016. URL: https://itunes.apple.com/at/app/fahrtenbuch/id286070473?mt=8 (visited on 05/01/2015).

[9] Daniel Foreman. *RPi - How to install Tight VNC Server*. 2016. URL: https://www.youtube.com/watch?v=yUk-T8fx6CQ (visited on 05/01/2015).

[10] *GPS Log Book*. 2016. URL: https://www.gpslogbook.co.nz/PublicPages/Home.aspx (visited on 05/01/2015).

[11] *GPS Log Book LIVE!* 2016. URL: https://www.gpslogbook.co.nz/PublicPages/Home.aspx (visited on 05/01/2015).

[12] *GPSBabel Documentation*. 2016. URL: http://www.gpsbabel.org/htmldoc-1.4.4/gpsbabel-1.4.4.pdf (visited on 05/01/2015).

[13] *Gpsd4java*. 2016. URL: http://taimos.github.io/GPSd4Java/ (visited on 05/01/2015).

[14] *Gurtam Wialon Driving Logbook*. https://en.wikipedia.org/wiki/Wialon ; http://gurtam.com/en/gps_tracking/gps_business/telematics_soft.html. 2016. (Visited on 05/01/2015).

[15] *How To Configure a Linux Service to Start Automatically After a Crash or Reboot*. 2016. URL: https://www.digitalocean.com/community/tutorials/how-to-configure-a-linux-service-to-start-automatically-after-a-crash-or-reboot-part-1-practical-examples (visited on 02/15/2016).

[16] *Installing a GPS Daemon (gpsd)*. 2016. URL: https://learn.adafruit.com/adafruit-ultimate-gps-hat-for-raspberry-pi/use-gpsd (visited on 05/01/2015).

[17] Ali Erdinç Köroğlu. *Alternative Way to Daemonize Java Applications on Systemd (CentOS7/RHEL7)*. 2016. URL: http://ae.koroglu.org/alternative-way-to-daemonize-java-applications-on-systemd-centos7rhel7/ (visited on 02/15/2016).

[18] smos linux. *SMOS Linux 1.3 - How to make autostart script (using shell/putty)*. 2016. URL: https://www.youtube.com/watch?v=Nq5opAqxaIQ (visited on 05/01/2015).

[19] *Logbuch App*. 2016. URL: http://www.logbook-app.com/logbuch-app.html (visited on 05/01/2015).

[20] Marko Martinović. *Manage GSM mobile broadband connections without Network Manager (Ubuntu, Linux Mint, Debian)*. 2016. URL: http://www.techytalk.info/manage-gsm-mobile-broadband-connections-without-network-manager-ubuntu-mint-debian/ (visited on 02/16/2016).

[21] *Maxtech*. 2016. URL: http://www.maxtech.com/index.php/en/fleet-management/track-my-fleet/automatic-drivers-logbook (visited on 05/01/2015).

[22] *Mileage Logbook*. 2016. URL: http://beste-apps.chip.de/android/app/fahrtenbuch-dienstfahrten-per-app-aufzeichnen,eu.whistlerconsulting.apps.android.mlb/ (visited on 05/01/2015).

[23] *Miles*. 2016. URL: http://www.milesapp.at/Miles/fahrtenbuch.html (visited on 05/01/2015).

[24] *MyLog GPS Fahrtenbuch Costs*. 2016. URL: https://play.google.com/store/apps/details?id=sk.i2m.mylog&hl=de (visited on 05/01/2015).

[25] NetBeansVideos. *Setting Up a GitHub Repository Using NetBeans IDE*. 2016. URL: https://www.youtube.com/watch?v=Ih9MFMuInjI (visited on 05/01/2015).

[26] oliver_hot. *SB-Modem-Stick Medion S4222 und Telekommunikationsanbieter Hofer Telekom HoT*. 2016. URL: http://www.linuxmintusers.de/index.php?topic=24461.0 (visited on 02/16/2016).

[27] *OsmAnd*. 2016. URL: http://wiki.openstreetmap.org/wiki/OsmAnd (visited on 05/01/2015).

[28] *RAPBERRY PI hompage*. 2016. URL: https://www.raspberrypi.org/ (visited on 05/01/2015).

[29] Robert Schanze. *Raspberry Pi: WLAN einrichten – So geht's*. 2016. URL: http://www.giga.de/zubehoer/raspberry-pi/tipps/raspberry-pi-wlan-einrichten-so-geht-s/ (visited on 05/01/2015).

[30] *TOUR*. http://de.tourapp.io/fahrtenbuch-app ; http://futurezone.at/apps/tour-das-vollautomatische-fahrtenbuch-fuers-smartphone/92.489.909. 2016. (Visited on 05/01/2015).

[31] *Tractive*. 2016. URL: https://tractive.com/de/gps (visited on 05/01/2015).

[32] *Trucker Logbook*. https://play.google.com/store/apps/details?id=com.keeptruckin.android&hl=de ; https://keeptruckin.com/. 2016. (Visited on 05/01/2015).

[33] *Tutorial: Raspberry Pi - Tastaturlayout ändern*. 2016. URL: https://www.youtube.com/watch?v=Bg1Hz97y2Vk (visited on 05/01/2015).

[34] *USB ModeSwitch*. 2016. URL: https://wiki.ubuntuusers.de/USB_ModeSwitch/ (visited on 02/16/2016).

[35] *USB Surfstick am Raspberry Pi verwenden – Mobiles Internet*. 2016. URL: https://raspberry.tips/raspberrypi-tutorials/usb-surfstick-am-raspberry-pi-verwenden-mobiles-internet/ (visited on 02/16/2016).

[36] *Warnings*. 2016. URL: http://www.farnell.com/datasheets/1835724.pdf (visited on 02/02/2016).

[37] Mark A. Ziesemer. *Configuring Persistent PPP under Ubuntu*. 2016. URL: http://blogger.ziesemer.com/2008/10/persistent-ppp-ubuntu.html (visited on 02/16/2016).