tgm
Die Schule der Technik

DIPLOMARBEIT

NAV AR

Ausgeführt in Zuge der Reife und Diplomprüfung
Ausbildungszweig Systemtechnik

unter der Leitung von
Prof. Dr. Helmut Vana
Abteilung für Informationstechnologie

eingereicht am Technologischen Gewerbemuseum Wien
Höhere Technische Lehr- und Versuchsanstalt
Wexstrasse 19-23, A-1200 Wien

von
Dominik George, 5AHITT
Thomas Pokorny, 5AHITT
Michael Fegerl, 5AHITT
Aleksey Korabach, 5AHITT

Wien, im April 2014

Abteilungsvorständin:	Prof. Dipl.-Ing. Grete Kugler
Tag der Reifeprüfung:	xx. xx xxxx
Prüfungsvorsitzender:	Univ.–Prof. Dipl.-Ing. Dr.techn. xxx
Erster Gutachter:	Dipl.-Ing.(FH) Mag. Dr.techn. Gotti Koppi
Zweiter Gutachter:	Prof. Dr.techn. Wenn Vorhanden

Vorwort

Diese Arbeit wurde im Jahr 2012 im Zuge unserer Ausbildung in der Abteilung für Informationstechnologie am tgm, HTBLVA Wien 20, durchgeführt.

Dankesworte

Wien, im Oktober 2012 Name, Name, Name, Name

Abstract

This is the english abstract.

Kurzfassung

Deutsche Kurzfassung kommt hierher

Contents

List of Figures

Listings

CHAPTER 1

Introduction

> "In the long history of humankind those who learned to collaborate and improvise most effectively have prevailed"
>
> (Charles Darwin)

Jetzt fügen[1] wir etwas aus dem Intenet ein, das im Kapitel 1.1 kommt[2]:

Figure 1.1: Da TGM Logo als Vektorgrafik, Schrift in schwarz, pfeil in blau und mehr mag ich nciht mehr schrieben.,.,.

Wie schon klar war, bklalaödsafdsaöladsf dsafmkdskjdsajhdsaf bal-. [6]
Wir alle finden [7–9].
Anmeldung zur Projektwoche am Turnersee [9] Als Erziehungsberechtigte(r) meines Sohnes (meiner Tochter)/ als Eigenberechtigte(r) gebe ich die Zustimmung, dass die Hin- und Rückreise zum oder vom Kursort mit Autobussen erfolgt. Ebenso gebe ich die Zustimmung, dass diese(r) von einer eventuellen Ausgehm?glichkeit vom Heim in den Ortsbereich bei aufgehobener Aufsichtspflicht der begleitenden Lehrpersonen Gebrauch machen darf. Ich nehme zur Kenntnis, dass für die gesamte Dauer einer Schulveranstaltung Alkoholverbot besteht. Das Rauchen

[1] asdsdakjadslhksfdjdf1
[2] sdkjhkahfklsd2

in der Unterkunft ist nicht erlaubt [7]. Ich erkl?re mich einverstanden, dass mein Sohn (meine Tochter)/ich bei groben Verst??en gegen die Anordnung der Aufsichtspersonen und gegen die Heimordnung auf meine Kosten mit einer Begleitperson nach Hause geschickt oder von einem Erziehungsberechtigten/Elternteil vom Kursort abgeholt wird/werde. In dringenden F?llen bin ich mit der Heranziehung eines Arztes und notwendigen ?rztlichen Veranlassungen am Veranstaltungsort einverstanden. Bei Nicht-Teilnahme besteht die Pflicht zum Ersatzunterricht. Für alle sich dabei ergebenden Ereignisse und Folgen übernehme ich als Erziehungsberechtigter die volle Verantwortung. Ein Betrag von 160 liegt bei.
Anmeldung zur Projektwoche am Turnersee Als Erziehungsberechtigte(r) meines Sohnes (meiner Tochter)/ als Eigenberechtigte(r) gebe ich die Zustimmung, dass die Hin- und Rückreise zum oder vom Kursort mit Autobussen erfolgt. Ebenso gebe ich die Zustimmung, dass diese(r) von einer eventuellen Ausgehm?glichkeit vom Heim in den Ortsbereich bei aufgehobener Aufsichtspflicht der begleitenden Lehrpersonen Gebrauch machen darf. Ich nehme zur Kenntnis, dass für die gesamte Dauer einer Schulveranstaltung Alkoholverbot besteht. Das Rauchen in der Unterkunft ist nicht erlaubt. Ich erkl?re mich einverstanden, dass mein Sohn (meine Tochter)/ich bei groben Verst??en gegen die Anordnung der Aufsichtspersonen und gegen die Heimordnung auf meine Kosten mit einer Begleitperson nach Hause geschickt oder von einem Erziehungsberechtigten/Elternteil vom Kursort abgeholt wird/werde.

- In dringenden F?llen bin ich mit der Heranziehung eines Arztes und notwendigen ?rztlichen Veranlassungen am Veranstaltungsort einverstanden. Bei Nicht-Teilnahme besteht die Pflicht zum Ersatzunterricht.

- Für alle sich dabei ergebenden Ereignisse und Folgen übernehme ich als Erziehungsberechtigter die volle Verantwortung. Ein Betrag von 160 liegt bei.

Anmeldung zur Projektwoche am Turnersee Als Erziehungsberechtigte(r) meines Sohnes (meiner Tochter)/ als Eigenberechtigte(r) gebe ich die Zustimmung, dass die Hin- und Rückreise zum oder vom Kursort mit Autobussen erfolgt. Ebenso gebe ich die Zustimmung, dass diese(r) von einer eventuellen Ausgehm?glichkeit vom Heim in den Ortsbereich bei aufgehobener Aufsichtspflicht der begleitenden Lehrpersonen Gebrauch machen darf. Ich nehme zur Kenntnis, dass für die gesamte Dauer einer Schulveranstaltung Alkoholverbot besteht. Das Rauchen in der Unterkunft ist nicht erlaubt. Ich erkl?re mich einverstanden, dass mein Sohn (meine Tochter)/ich bei groben Verst??en gegen die Anordnung der Aufsichtspersonen und gegen die Heimordnung auf meine Kosten mit einer Begleitperson nach Hause geschickt oder von einem Erziehungsberechtigten/Elternteil vom Kursort abgeholt wird/werde. In dringenden F?llen bin ich mit der Heranziehung eines Arztes und notwendigen ?rztlichen Veranlassungen am Veranstaltungsort einverstanden. Bei Nicht-Teilnahme besteht die Pflicht zum Ersatzunterricht. Für alle sich dabei ergebenden Ereignisse und Folgen übernehme ich als Erziehungsberechtigter die volle Verantwortung. Ein Betrag von 160 liegt bei.

Anmeldung zur Projektwoche am Turnersee Als Erziehungsberechtigte(r) meines Sohnes (meiner Tochter)/ als Eigenberechtigte(r) gebe ich die Zustimmung, dass die Hin- und Rückreise zum oder vom Kursort mit Autobussen erfolgt. Ebenso gebe ich die Zustimmung, dass diese(r) von einer eventuellen Ausgehm?glichkeit vom Heim in den Ortsbereich bei aufgehobener Aufsichtspflicht der begleitenden Lehrpersonen Gebrauch machen darf. Ich nehme zur Kenntnis, dass für die gesamte Dauer einer Schulveranstaltung Alkoholverbot besteht. Das Rauchen in der Unterkunft ist nicht erlaubt. Ich erkl?re mich einverstanden, dass mein Sohn (meine Tochter)/ich bei groben Verst??en gegen die Anordnung der Aufsichtspersonen und gegen die Heimordnung auf meine Kosten mit einer Begleitperson nach Hause geschickt oder von einem Erziehungsberechtigten/Elternteil vom Kursort abgeholt wird/werde. In dringenden F?llen bin ich mit der Heranziehung eines Arztes und notwendigen ?rztlichen Veranlassungen am Veranstaltungsort einverstanden. Bei Nicht-Teilnahme besteht die Pflicht zum Ersatzunterricht. Für alle sich dabei ergebenden Ereignisse und Folgen übernehme ich als Erziehungsberechtigter die volle Verantwortung. Ein Betrag von 160 liegt bei.
Anmeldung zur Projektwoche am Turnersee Als Erziehungsberechtigte(r) meines Sohnes (meiner Tochter)/ als Eigenberechtigte(r) gebe ich die Zustimmung, dass die Hin- und Rückreise zum oder vom Kursort mit Autobussen erfolgt. Ebenso gebe ich die Zustimmung, dass diese(r) von einer eventuellen Ausgehm?glichkeit vom Heim in den Ortsbereich bei aufgehobener Aufsichtspflicht der begleitenden Lehrpersonen Gebrauch machen darf. Ich nehme zur Kenntnis, dass für die gesamte Dauer einer Schulveranstaltung Alkoholverbot besteht. Das Rauchen in der Unterkunft ist nicht erlaubt. Ich erkl?re mich einverstanden, dass mein Sohn (meine Tochter)/ich bei groben Verst??en gegen die Anordnung der Aufsichtspersonen und gegen die Heimordnung auf meine Kosten mit einer Begleitperson nach

Hause geschickt oder von einem Erziehungsberechtigten/Elternteil vom Kursort abgeholt wird/werde. In dringenden F?llen bin ich mit der Heranziehung eines Arztes und notwendigen ?rztlichen Veranlassungen am Veranstaltungsort einverstanden. Bei Nicht-Teilnahme besteht die Pflicht zum Ersatzunterricht. Für alle sich dabei ergebenden Ereignisse und Folgen übernehme ich als Erziehungsberechtigter die volle Verantwortung. Ein Betrag von 160 liegt bei.

1.1 Background and Motivation

1.1.1 Background

1.1.2 Motivation

1.2 Objectives of this Thesis

1.3 Methodology for the Developement

1.4 Thesis Outline

CHAPTER 2

Theoretical Basics

2.1 General

Today everybody can create a App for any Platforms. In this world, most of the pupils uses a Smartphone. So in that case this work describes from which modules or elements the App NAVAR created of. Furthermore it shows how to use the Modules in General and how to deploy them. However for this project we used three programming languages. For the user Interface we used JQuery Mobile Framework. This app is just only for Android supported Smartphones. The reason is the SDK that we are using for the app, it has been just developed for Android and IOS. A programming language is an Interface between Human and Machine, in this project Human and Smartphone. A person can understand the programing language and a machine or smartphone can process it.

The next point is, in this project we are using Javascript,HTML5 and CSS with the Framework JQuery Mobile for graphic display. The main problem was to combine the user interface with the logic implementation. Further more the connection between the smartphone and the Navision Server was also not so easy to implement.

In this chapter it describes the programming languages and the Frameworks, that was used in this project:

1. Java
2. JavaScript,HTML5,CSS
3. jQuery Mobile

4. C#

2.2 Augmented Reality

2.2.1 What is Augmented Reality ?

Augmented Reality(AR) is a type of virtual environment. So it means, that aims to duplicate the worlds environment in the computer. The idea of AR is that it combines the scene of the real world with virtual scene generated by the device. Furthermore the virtual scene augments the scene with additional Information. So the virtual scene which is designed to enhance the users sensory perception of the virtual World, they are seeing or interacting with. The virtual scenes which are generated by computer are sensory inputs such as sounds, videos, graphics or GPS data. It replaces real world with a simulated one.

A good example for Augmented Reality is the sport games which will be live broadcasted and on the TV you can see the scores or other information. This shows that, Augmentation is conventionally in real-time and in semantic context with environmental elements. The advanced AR technology helps to surround the real world of the user with information, so it becomes interactive and digitally manipulable.[1]

2.2.2 Augmented reality vs Virtual Reality

'Augmented reality (AR) and virtual reality (VR) are fields in which the lines of distinction are kind of blurred. To put it another way, you can think of VR as the precursor to AR, with some parts overlapping in both. The main difference between the two technologies is that VR does not use a camera feed. All the things displayed in VR are either animations or prerecorded bits of film. '[10]

2.2.3 Companies for Augmented Reality

There are three Companies, which produce Software or programs to use the Augmented Reality Technology for smart phones or Google Glasses. The name of the first company is Metaio GmbH. It is a German corporation. This company was founded in 2003. It offers Augmented Reality for industrial and automotive sectors for product design and factory planning. This concern already created few Apps for Smartphones. They are working with other companies together to create Applications for mobile phones, such as an app for an E-manual for Audi. This app starts the camera and scans the car components such as the steering of the car and tells the user which functionality the steering has. Furthermore the firm is very young and with this technology it has a big developmental push. It has a big development platform for creating applications with AR. So it sells products such as SDKs and other programs to create our own AR Applications

for smartphones and Desktop PCs.

The second Company is a spin-off company from Swiss Federal Institute of Technology(ETH) in Zurich, and name of the product is Kooaba. It was founded in November 2006 and the mission of this product is to unlock the information captured in images using sophisticated image recognition technology. This firm is a competitor to Metaio ltd.

The last popular company is Google. They created the Google glasses and another project called tango. For the project tango they created a specified mobile phone with two back cameras.

Project Tango is an attempt to create a mobile device unlike like any other, a mobile device that shares our sense of space and movement, that understands and perceives the world the same way we do.

They have been collaborating with universities, research labs, and industrial partners who share this passion spanning 9 countries around the world to concentrate the past 10 years of research in robotics and computer vision into a unique mobile phone. We now have prototypes ready to put into the hands of eager development partners that can help us imagine the possibilities and to transform those ideas into reality.[3]

Augmented Reality is used by Smartphones and Google Glasses. Google glasses is a very good product for the AR Technology.

Figure 2.1: [1]

2.2.4 Google Glasses

The new innovation from google was really exciting. Google glass is a new gadget for the whole world. Furthermore it combines the reality with virtual components . This project's launch event was in 2012 .Google Glass is the name for a type of wearable computer. It was created by the Goggles Project team Glass. It provides Augmented Reality for users by visually connecting them to an Android-run heads up display that offers many of the features of an Android smartphone. With this device the user can connect to Goggle's key cloud features such as maps, calendar ,Gmail, Google and Google Places. Google hopes to have the gadget in the market in the near future. So they expect the technology to cost about as much as a smartphone.

Figure 2.2: [1]

Google Glass has 7 main functionalities[6]:

The first one ,it doesn't need an extension of a smartphone or tablet. This gadget has it's own hardware such as in mobile phones. It can perform itself various day to day tasks, without moving the hands of a user.

The second function is that it can record a video or it will take picture, if the user just give an oral command. So in that case the user never have to touch a Button or the hardware. The photos and videos will be stored on the 4GB flash memory of the device and it can also be shared on social networking websites or emailed.

The third function is , it shows the user text messages as well as emails that the user receives. Via voice commands the user can reply to the text message or email.

The fourth function is about googling with this device. If the person like to find a lot of information , the user just have to ask a question and Google glass will pull the answer from the internet. For example ,the end-user can ask when the st. stephen's cathedral was built or to give a few pictures of the church. The answers or the pictures will be provide on the small screen in form of the users eye.

The next feature is to show maps. Probably lots of people uses Google Maps, so Google Maps are integrated into Glass. The user will be able to chart the course of the journey or lookup locations. It is possible to do establishments via voice commands.

The fifth feature is live video sharing. Google Glass has the ability to show the world what the user of the device sees → live! A good example is , when the user is attending a family function and users childes school play or a concert, he/she can share the feed with her/his friends or family members in real-time. So he/she can make them a part of the experience.

Google Glasses next features is , it has Google Now integrated. Google Now is a digital voice assistant. It will keep track of the daily habits , such as when the user leave for office of the route that the user take. Google Now will give the user a alternate routes if there is a traffic on the way or it gives weather updates periodically and it has among various other functions.

The last function from Google Glass is, translation from a language to another. The user have to ask Google Glass to translate a phrase or sentence from one language to another and it will speak that out.

Figure 2.3: [1]

[7]

2.2.5 Usage of Augmented Reality[8]

2.2.5.1 Military and Law Enforcement

The military and law enforcement agencies uses AR Technology for full simulators which are designed to help in training. For Example, a wide screen inside a room or a vehicle on which various scenarios is presented, and the trainee must decide the best course of action.

Some advanced Special Forces teams have basic AR goggles that , along with the land in sight, display information such as altitude, angle of viewing, light intensity, and so on.AR technology also used by specialized night vision glasses. This device can display location and other information. The most of the unmanned vehicles in the military branches uses also AR technology as well. These vehicles, especially the aerial ones, can be thousands of kilometres away from their operators. The next point is that the vehicles have one or more cameras mounted on their exterior, which transmit video to their operator. This vehicle are equipped with several sensors as well. There is a sensor which sends data to the operator along with the video. This data is the processed and augmented over the video. The operators System with complex algorithms picks out the mark building or objects of interest. This kind of information will be displayed as an overlay on the video.

2.2.5.2 Vehicles

Nowadays AR technology started to be implemented in vehicles. Often there are multiple screens in the vehicle, each showing particular direction. So just think about there is only one screen and multiple cameras, the vehicle will either switch the feed automatically or have the option for the user to switch between the cameras. The exterior of the vehicle has the ability to control the several cameras. The images from the camera will be shown on the screen and it is overlayed with useful data such as small map, compass, direction arrows, alternate routes, weather forecast and much more. This kind of technology is currently most visible in airplanes and trains at the moment. Some smart cars has the same ability ,but they are in test phase . The Submarines and ships are using this technology as well. The important thing is that Space Shuttles had this kind of AR technology also.

It is possible to create apps which implement a sort of hybrid way on the Android platform. The reason is that the most Android devices seem to bee lacking in features that normal vehicles have, the same kind of features are not achieved. On the other hand, apps can be written for the help to navigate by using the

GPS to get to the right location. With the right API it is possible to write a APP to use the accelerometer to help with acquiring the speed of the vehicle. Android device provides the AR power and the vehicle provides the vehicle part.

A example of a smart car:

Figure 2.4: [1]

[9]

2.2.5.3 Medical

AR technology is quite popular in the medical field. This technology enables to becoming more common these days for surgeries. With AR the error rate are smaller in Surgery branch. The reason is that the computer provides valuable inputs on the surgery and uses the information to control robots to perform some or all of the surgery. Furthermore , the computer can often provide alternative ways and instructions on what can be done to improve the surgery in real time. Augmented Reality stream , along with other data, these data can be sent per remote to doctors, who can view the information of the patient as if the patient were in front of them.

In the medical field ,there are other medical apps of AR technology. It is possible to use AR machines to monitor a large number of patients and make sure that their vital signs are under observation at all times.

This kind of AR technology can never be implemented on a Android smartphone. The main reason is , it is to expensive. To create such a app we need a team of very good developers, a team of highly skilled and experienced doctors and a large amount of money.

2.2.5.4 Trail Rooms

AR technology are widespread in several shops. The reason is , why some shops uses AR is to create a virtual trial room. The idea of virtual trial room is that the user stands in front of a screen with a camera mounted somewhere. So the user will see himself displayed on the screen. The next point is the user uses an input devices such as a mouse or a keyboard to select any of the available clothing options. In the background the computer use a algorithm to augment that item onto the users image and display it on the screen. The user can turn to view himself from all angles.

2.2.5.5 Tourism

The Tourism branch is also using the AR technology. Around the World, there are a lot of famous spots. So the organized tours now offer a head-mounted AR system that displays information about the current site and its buildings when the user look at it. Furthermore the tourist can rebuild buildings, cities , landscapes and terrains as they existed in the past with the AR technology. The Tourism AR provide icons or markers for famous monuments. Tourism AR has the ability to find parks, restaurants, hotels and other tourist related sites and attractions in an unfamiliar city. These applications are not limited to historical places.

2.2.5.6 Architecture

In this the world there a lot of camera-equipped machines that can generate a blueprint form an existing structure or display a virtual structure from the blueprints on the proceed site of constructions. These functionality helps to design and check buildings. Augmented reality technology provides the functionality to simulate natural disaster conditions. So it can show how the building structure will react under that kind of pressure.

2.2.5.7 Education

In Educational Institutes is AR technology very useful. Children or students can learn through AR. AR act in this field as add-ons to the textbook material or as a virtual, 3d textbook in itself. Furthermore the AR give the ability for the student to ẗeliveëvents as they are known to have happened, while never leaving their class.

2.2.5.8 Art

Augmented Reality helps to create paintings, models and other forms of art. The technology helps to try out a particular design, before actually putting it down in ink or craving it out of stone. It is also able to paint something virtually to see how they turn out and the artist can repaint as often he wants until he is satisfied . Then he can put down on the canvas finally.

2.2.5.9 Translation

AR technology can be used for translate text from multiple languages all over the world. AR feature OCR and either have an entire cross-language dictionary on the device or it can translate the language over the Internet. Few companies are producing apps with this ability. For this function we have to use a ready-made optical character recognition(OCR) library to convert the images from the camera to text. The idea of OCR is it extract the text from image and put compare it with the translation dictionary or it can be translated through the internet. The translated result will be shown on the display.

2.2.5.10 Weather Forecasting

Most of the weather forecast app are augmented. The Data for the weather will be recorded and while the recording the green backdrop serves as a marker. If the recording is finished, a computer is used to add the map and position to match the forecasters actions. AR are used by transmitting the forecast live to the viewers .

2.2.6 Future of Augmented Reality[10]

Augmented Reality is a growing up technology. It has amazing abilities, but few of the abilities can't be implemented right now due to limitations in hardware and algorithms.

2.2.6.1 Virtual Experiences

In the future the AR technology could have a system ,which could transform from the current location into something completely different. A good example is , just imagine in the future you can live through movies by wearing such a system and seeing the movie happen around. Probably this technology could convert the house of a user into a medieval castle or into the international space station. Furthermore with the combination of smell-emitting technology and the aural AR , it could make the environment lifelike and feel completely real. In addition to this ,it is capable to add a emulation of the sense of touch with a body suit. That will make it absolutely and undeniably real.

2.2.6.2 Holograms

The following point is that AR allows the user to have a live direct or indirect of the world. That could enable users to have holograms in front of them. These holograms could be interactive or merely descriptive. For instance somebody is calling you and a hologram of these person appears in front of you. So we see AR could have this ability.

2.2.6.3 Video Conferencing

In the future , multiple people will appear in the same conference room if a video feed of a conference room is transmitted to them with the AR technology. The idea is that the people could use the webcam to äppearïn the seat of the room, along with the members.

This idea could probably help people who are not able to attend the meeting ,because they are thousands of kilometres away. So this futuristic Video Conferencing could solve this problem. Furthermore for this implementation we need a high-speed internet and the person which participating the conference have to stay exactly in the same place, if not then the algorithm have to positioning him again and these need a big amount of the data streaming.

2.2.6.4 Movies

This technology can be used to play entire movies. The idea is that the theatre could be replaced with the background of the movie or the theatre could be replaced with the actors only. The first method is that the actors could be augmented onto the background and in other way the background could be augmented behind the actors. The second method would reduce the costs of the shooting. These methods could provide more realistic and fun movies.

2.2.6.5 Gesture Control

AR could be used for many gesture controls such as eye dialing. It should track the eye movement from the user and should select the right the appropriate number key. If they key has been selected , the user could blink to press that number and then proceed to select the next key. This kind of algorithm could be implemented to control music players, mobile apps, computers and other form of technology.

To create app with this algorithm requires a few things:

First of all it needs a front camera with a reasonable resolution. The second thing is that the algorithm has to be well written to detect fine eye movements and to convert it to the right information. This algorithm has to filter other movements.

2.2.7 Summary

So we see that AR is a developing technology. The basic requirements for the technology is back and front camera , GPS, accelerometer and compass. Most of the requirements are fulfilled by almost all Android devices on the market. Now is great time to create AR apps, because in these the competition is very low and it is good to start business with it. Augmented Reality are quite popular in many fields such as Military, Medical and Education.

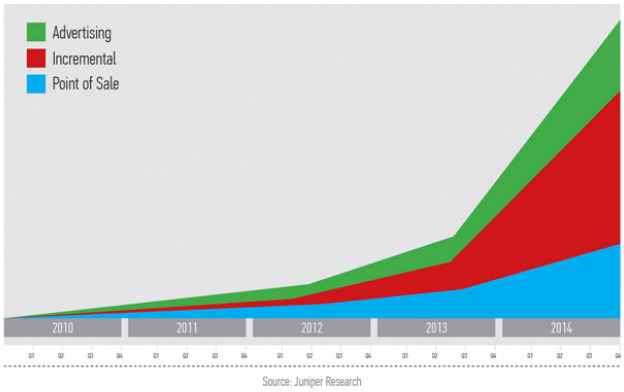

Figure 2.5: [1]

[11]
The Graph is showing the Advertising and Point of Sale rate of AR. We can see the graph is increasing year to year. The Result of the Graph is that the features of the technology is increasing.

2.3 Java

2.3.1 What is Java

Java is a computing platform and object oriented programming language first released by Sun Microsystems in 1995. Oracle has bought Java in 2010. [10]

The Java platform consists of the Java application programming interfaces (APIs) and the Java virtual machine (JVM).

Java is class-bases and object oriented. It is intended to let application developers'write once, run anywhere' meaning that code that runs on one platform does not need to be recompiled to run on another. Java programs are compiled to byte-code. this code can run on any JVM regardless of the real computer architecture. [11]

Java is next to C/C++ one of the most popular programming languages. [12] The language also has a similar syntax to C and C++.

2.3.2 Class Based & Object Oriented

Class-based object-oriented languages, such as Java , are founded on the concept of two distinct entities: classes and instances. [?]

1. **Class:** A class is a blueprint or prototype from which objects are created. [13] In class-based languages, you define a class in a separate class definition. In that definition you can specify special methods, called constructors, to create instances of the class. A constructor method can specify initial values for the instance's properties and perform other processing appropriate at creation time. [1] In Java the **new** Operator is with a call of the constructor method is used to make a new instance of a class.

2. **Instance:** An instance or object is the instantiation of a class that is one of its members. Software objects are often used to model the real-world objects

3. **Interface:** An interface is a collection of empty methods. When a class implements an interface, in java with the keyword **implements**, it has to implement all methods of the interface. A class describes the attributes and behaviors of an object. An interface contains behaviors that a class implements.

4. **Subclasses:** In a class-based language, you create a hierarchy of classes through the class definitions. [1] The subclass, in Java the keyword **extends** is used, provides all functionalities of the super class an can add new ones ore modify the existing properties.

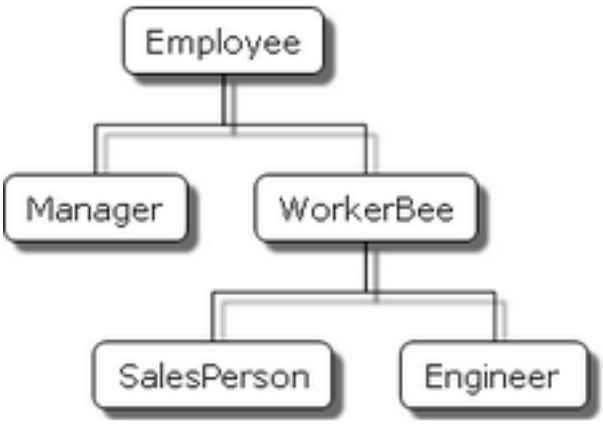

Figure 2.6: [1]

As you can see in this example *Engineer* is an *Employee*. But *Manager* which also is an employee has not the same properties.

5. **Abstract Class:** An abstract class is a class that can't be instantiated. It's only purpose is for other classes to extend. Abstract classes are similar

to Interfaces but an abstract class, in contrast, provides more structure. It usually defines some default implementations and provides some tools useful for a full implementation. [14]

6. **Package:** A package is a namespace for organizing classes and interfaces. Packages make large software projects easier to manage. [13]

2.3.3 Design Patterns

Design patterns are proven solutions approaches to specific problems. A design pattern is not a framework! They are based on the base principles of object orientated design.

1. Program to an interface not an implementation
2. Favor object composition over inheritance.

2.3.4 Performance

Programs written in Java have the reputation of being slower than other languages. However in the last 10 years the JVM execution speed increased dramatically. In six separate web performance benchmarks, Java frameworks took 22 out of the 24 top-four positions. The JVM has been optimized that much that Java code is now running nearly as fast as C++ code. [15]

2.3.5 JVM

The Java virtual machine is what makes Java a platform independent programming language. A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. Therefore, the JVM runs on all kinds of hardware to execute the Java Bytecode without changing the Java execution code. Java developers do not need to know how the JVM exactly works. However a deeper knowledge of the JVM helps understanding how JAVA works and can be helpful to solve various problems. [2]

Features of JVM:

1. **Stack-based virtual machine:** Most computer architectures such as Intel x86 Architecture and ARM Architecture are based on registers. Whereas the JVM is stack based. [2] That means that the VM doest need to know the operand addresses, it only calls the Stack-Pointer which points to the current instruction. [16]
2. **Symbolic reference:** All data types except for primitives are referred to through a symbolic reference.

3. **Garbage collection:** The garbage collector frees the memory from objects that are not in use any more. [17]

4. **Guarantees platform independence by clearly defining the primitive data type:** In other more tradition languages like C or C++ primitive data types have different sizes according to the System. In Java the JVM defines a fixed size for primitives.

[2]

2.3.6 Java bytecode

The Java bytecode is the result of a compiled Java source-code. It is a middle-language between Java and the machine code. [2]

2.3.7 Java Code Execution Process

The Java code execution process is shown in the following figure.

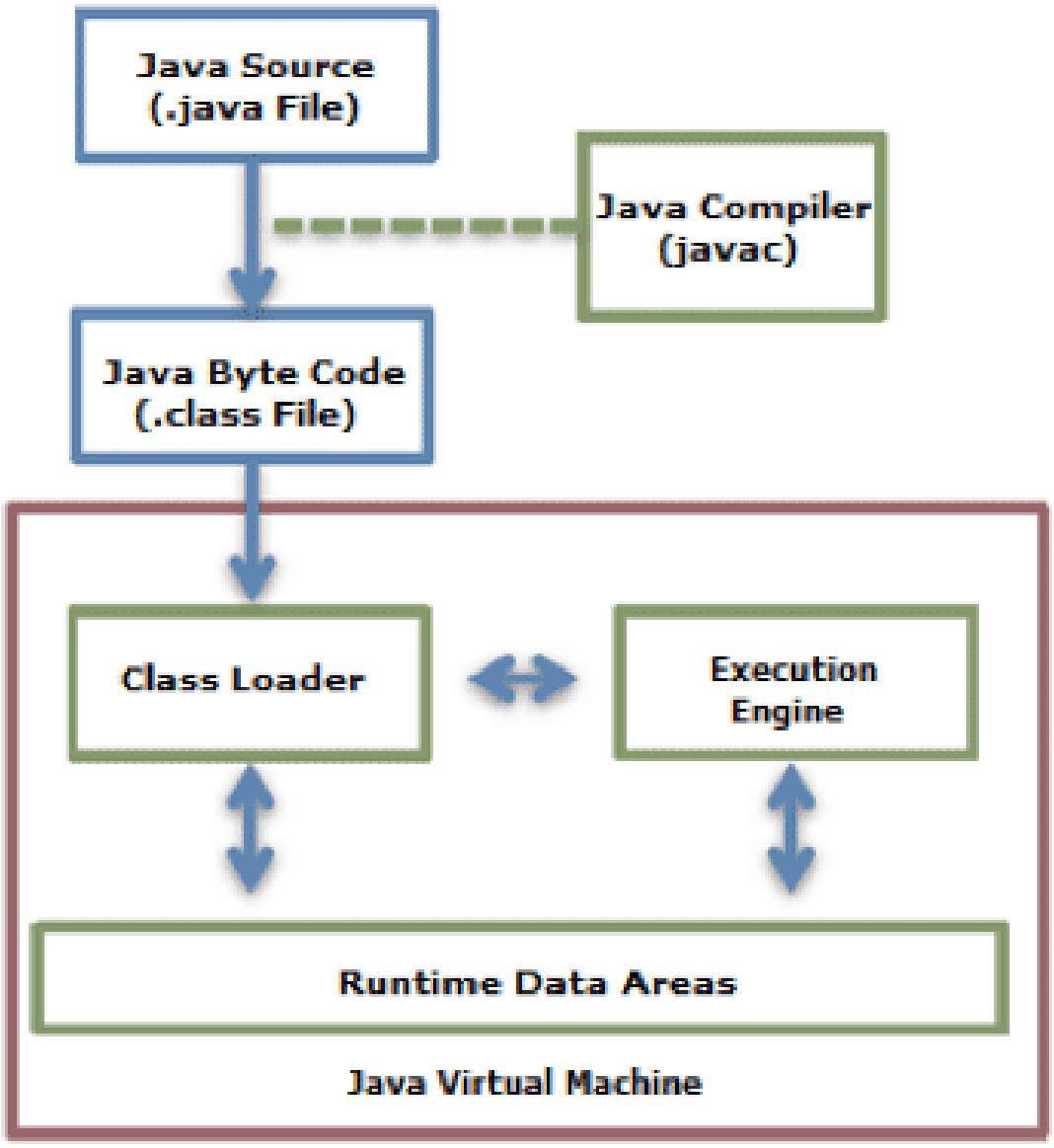

Figure 2.7: java code execution process [2]

2.3.7.1 Class Loader

The Java Class Loader loads and links a class when it refers to a class the first time at runtime. Every class loader has its own namespace that stores the loaded classes. [2]

2.3.7.2 Runtime Data Areas

The JVM Runtime Data Areas is the Memory assigned to a program when it runs on the OS. They can be divided into six areas: the Pc Register, JVM Stack, Native Method Stack, Heap, Method Area, and the Runtime Constant Pool. The first three are created for a single thread the other areas are shared by all threads.

1. **PC register:** One **program counter** register exists for one thread. It gets created when the thread starts. Pc register has the address of the JVM instruction that is executed now. [2]

2. **JVM Stack:** Each thread has a private JVM Stack, created the same time as the thread. A Java Virtual Machine stack stores frames. Frames are used to store data and results, new frames are created each time a method is invoked. It gets destroyed when its method invocation completes, whether that completion is normal or abrupt (it throws an uncaught exception). [18]

3. **Native Method Stack:** A stack for native code written in a other language than Java. It is a stack used to execute C od C++ Methods. [2].

4. **Heap:** The JVM Heap is a data area that is shared among all Java Threads. The heap is created on virtual machine start up. Its a space that stores all class instances Arrays and Variables. If a program requires more heap space than aviable the Java Virtual Machine throws an **OutOfMemoryError** [18]

5. . **Method area:** The method area is shared by all threads, created when the JVM starts. It stores runtime constant pool, field and method information, static variable, and method bytecode for each of the classes and interfaces read by the JVM. Unlike in the heap the garbage collection in the method area is optional for each JVM version. [2]

6. **Runtime constant pool:** The Runtime pool is a part of the Native Method stack and gets created when a class or interface gets created. Its the run-time representation of the **constant pool** table in a class file. This constant pool table contains several constants [19]

   For example:

   Listing 2.1: Java example Code

```
1 System.out.println("Hello, world!");
```

   Generated byte-code:

   Listing 2.2: JVM bytecode

```
1 0:   getstatic      #2;
2 3:   ldc      #3;
3 5:   invokevirtual    #4;
```

#n indicates that this is a reference to the constant pool. 2 is a symbolic reference to *System.out*, #3 is the *Hello, world!* string.#4 references to the *PrintStream.println(String)* method. [20]

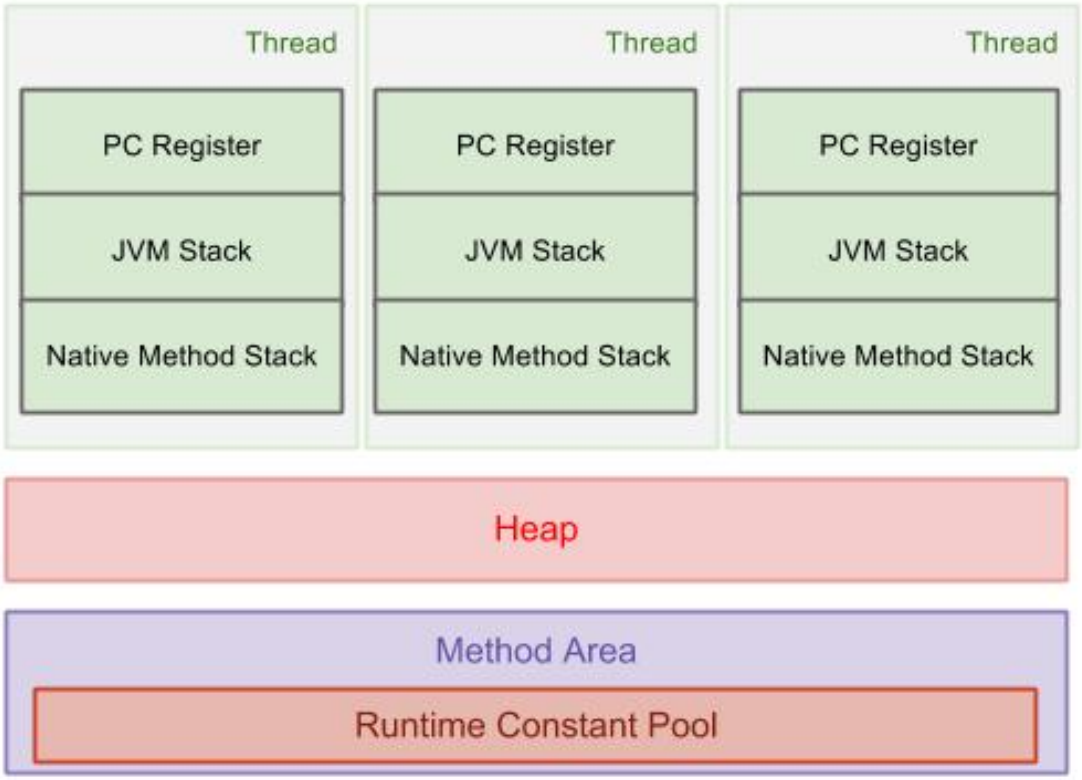

Figure 2.8: Java Run-time Data Areas [3]

2.3.7.3 Execution Engine

The bytecode that is assigned to the runtime data areas in the JVM loaded from the class loader is executed by the execution engine. The execution engine reads the Java Bytecode in the unit of instructions. It is like a real CPU executing the machine commands one by one. Each command consists of 1 Operation Code byte and an additional Operand code. The execution engine gets one OpCode and execute task with the Operand, and then executes the next OpCode. [2]

2.3.8 .JAR File

A JAR **(Java ARchive)**is a file that contains the class, image, sound, etc. files for a Java application or applet gathered into a single file and possibly compressed. [21]

2.3.8.1 Executable JAR

Its also possible to create a executable .Jar files. It behaves similar to a .exe file in Windows. It can be executed with a double click when Java is installed on the system.

CHAPTER 3

Integrated Development Environment

3.1 Andorid SDK Eclipse

3.2 JetBrains WebStorm

The application's logic had to be created with a programming language called JavaScript. Because of that, the project group had to find a development environment that's best suited for this language. JetBrains WebStorm 7.0.3 was best fit for all future tasks and should be the environment in which JavaScript had been developed.

However, not only JavaScript, but also HTML as well as CSS could be developed with this IDE. All information about this product can be found on Jet Brains homepage. [22]

3.2.1 Overview

JetBrains WebStorm is a professional JavaScript IDE that supports a wide range of modern technologies related to JavaScript programming language, HTML and CSS, and provides the complete experience for productive Web development.

WebStorm offers developers an intelligent code editor that truly supports the structure of code written in JavaScript, HTML or CSS, as well as their modern successors. It features the best-of-breed coding assistance for a whole set of cutting-edge web technologies, including code completion, refactorings, code

formatting, on-the-fly error prevention, and much more.

WebStorm is also great for developing Node.js applications. Together with integrated instruments for testing, debugging and code analysis and integration with various VCS, WebStorm is an essential tool for powerful and productive web development.

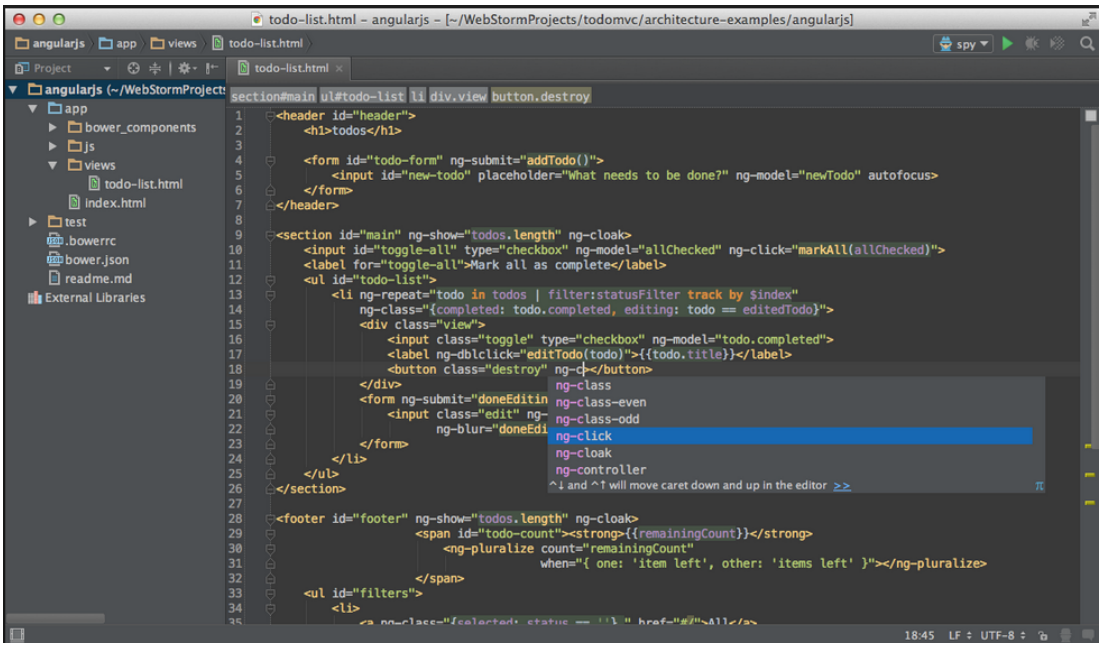

Figure 3.1: WebStorm Interface

3.2.2 Features

- Intelligent JavaScript, HTML, and CSS editor with syntax highlighting, code completion, configurable formatting configuration, refactorings, on-the-fly error detection and support of language mixtures.
- Support for a wide range of technologies: TypeScript, CoffeeScript, Dart, LESS, Sass, Stylus, Compass, EJS, Handlebars, Mustache, Web Components, Jade, Emmet, and many more.
- Productivity-boosting Live Edit feature: See the changes in the browser immediately without reloading the page.
- JavaScript debugger for Chrome and Firefox, with breakpoints, stepping, frames view and watchers. Full-featured debugging of TypeScript, CoffeeScript and Dart with sourcemaps.

- File Watchers for automatic compilation/transpilation of higher-level languages like TypeScript, CoffeeScript, LESS, Sass, and Stylus.
- A debugger for Node.js applications with the latest features of V8 Debugger Protocol.
- Intelligent code inspections, one-click quick-fix suggestions, JSHint, JSLint, and Google Closure Linter.
- JavaScript unit testing with integrated JSTestDriver or Karma test runner with code coverage.
- Built-in HTTP Server, REST Client, Terminal and Node.js package manager npm.
- New Project Wizard with well-known project templates like Twitter Bootstrap and Node.js Express App.
- Integration with Version Control Systems including Git, Subversion, Mercurial, CVS, Perforce, and GitHub.
- Easily configurable FTP/FTPS/SFTP deployment.
- Integration with various issue trackers.

3.3 Visual Studio

CHAPTER 4

Logic Implementation with JavaScript

The logic of this app is divided into two parts. First part is JavaScript logic, that is responsible for the functionalities of each HTML site, more precisely, the dynamic response to the user. Second part is Java Android logic, which is responsible for the main function called car tracking and all other functionalities that could not be accomplished with help of JS.

Altogether there are 10 HTML sites and each one of them has some functionalities that had to be implemented with JS or Android Java. A simple example of a functionality is pressing a button. This button triggers a function inside the JS.

However, JavaScript and Android Java did not provide everything that has been needed for the project application. That's the reason why project members had to use several other web frameworks like jQuery or phonegap.js. This was necessary to accomplish the main goal of powerful, user-friendly mobile application. The usage of web frameworks will be explained in further chapters.

4.1 Start Menu

At the beginning of the application a start menu is displayed. In this menu the user can use functions that had been especially developed for him. Help, car tracking, favourite list, about and exit are the functions in the start menu.

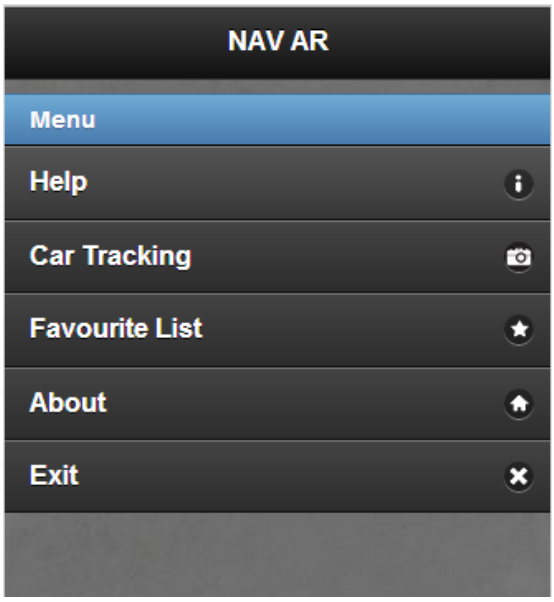

Figure 4.1: Start menu

Each one of these buttons have their logic that is implemented in **index.html**. The listing 4.1 shows the functionality behind each button.

```
1 <li data-icon="info"><a href="help.html" rel="external">Help</a></li>
2 <li data-icon="camera"><a href="#" onclick="trackClick();">Car Tracking
3 <li data-icon="star"><a href="myfavourite.html" rel="external">Favouri
4 <li data-icon="home" ><a href="about.html" rel="external">About</a></li
5 <li data-icon="delete"><a href="#" onclick="turnOff();">Exit</a></li>
```

Listing 4.1: start menu source code

Functions *trackClick()* and *turnOff()* were implemented in Android Java and are described in chapter....

```
1 function trackClick() {
2     MyTracking.performClick();
3 }
4
5 function turnOff(){
6         Exit.exitClick();
7 }
```

Listing 4.2: JavaScript functions

The button **help** forwards the user to the help display *help.html*. The same functionality features **about** and **favourite list**, except they link to another display.

Exit button invokes the function *turnOff()* which calls another Android Java implemented function. *Exit.exitClick()* ends the application. Illustrated in listing 4.2.

Car tracking calls a function *trackClick()*. This method starts the main function.

4.2 Help

The help display provides only two major options: back button and the link to the self created tutorial video.

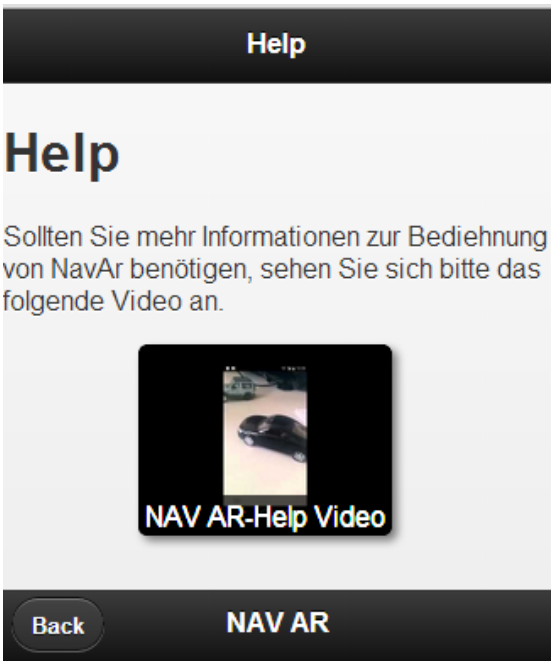

Figure 4.2:

The back button leads to the main menu. Shown in listing 4.3.

```
1 <a class="ui-btn-left" href="index.html" rel="external">Back</a>
```

Listing 4.3: Back button

If the user touches the picture **HAV AR-Help Video**, he will be linked to a specific how-to YouTube video. This video serves as a simple help to understand how the application works. It shows how to use the application's main functions and more.

```
1 <ul id="Gallery" class="youtube-videogallery" align="center">
2 <li><a href="https://www.youtube.com/watch?v=6U4oT5AbAsg&feature=youtu
3 </ul>
```

Listing 4.4: Help video

4.3 Start Menu

The most important function of the whole mobile applications is **car tracking**. This function is executed by *MyTracking.peformClick()*. More in chapter 4.1)Start Menu.

After a car was successfully tracked, the user is linked to a new display called **index.html**, which is the start menu. It provides the user with additional options. Options that deliver technical information as well as review about the tracked car and more other useful functions.

There is also a possibly to add the tracked car to users car collection named **the favourite list**. Out of there he can select one specific vehicle to use the start menu options, like picture or videos gallery.

JavaScript functions had to be created for each of this options. These functions are described in chapter 4.3.1)JavaScript Functions.

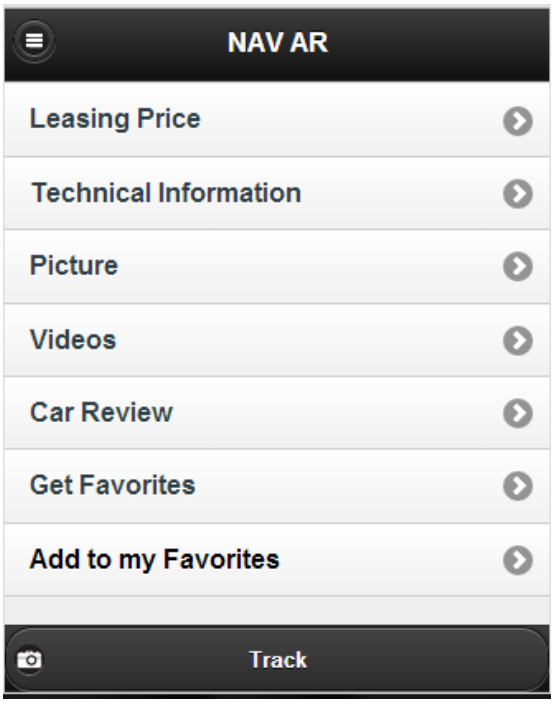

Figure 4.3:

4.3.1 JavaScript Functions

4.3.1.1 Start timer

When the site is loaded a timer automatically starts. Specific functions stop the timer and sends the time to the Navision server.

```
1 function startTime(){
2         var d = new Date();
3         timestart = d.getTime();
4 }
```

Listing 4.5: Start timer function

4.3.1.2 End timer

This function stops the timer which had been started with the method *startTime()* and safes the time with the method *timestampsave()*. The timer was used to get the time how long a user has selected a specific car and used certain start menu options.

```
1 function endtime(){
```

```
2           var d = new Date();
3           var endtime = d.getTime();
4           timestampsave(timestart,endtime);
5       }
```

Listing 4.6: End timer function

4.3.1.3 Save the time

This function saves start time and end time of the timer, into the Navision server. The start time and end time are the input parameters.

To save the time into the server *timestampsave()* needs several other information like email and id of the tracked car. More about sending information to the server and to the connectivity between application and server read the chapter ...

time1.....start time time2.....end time

```
1  function timestampsave(time1,time2){
2           var stime       =       time1;
3           var endtime     =       time2;
4           var emailan =   sessionStorage.getItem('email');
5           var fid         =       sessionStorage.getItem('id');
6
7
8           $(document).ready(function () {
9               $.ajax({
10                  type: "GET",
11                  url: "http://ar-tgm-navax.cloudapp.net:9090/rest/insert
12                  async: false,
13                  dataType: 'JSONP',
14                  success: function(data){
15                      //do your stuff with the JSON data
16                      var test=data;
17                      console.log(test);
18                  }
19              });
20          });
21      }
```

Listing 4.7: Save time function

4.3.1.4 Set parameters

There are two parameters that have to be saved into the session storage to establish a connection with the Navision server. That were id of the tracked car and the email address of the user.

```
1 function setParam(){
2         window.sessionStorage.setItem('id', myVariable);
3         window.sessionStorage.setItem('email',email);
4 }
```

Listing 4.8: Set parameter function

4.3.1.5 Start car tracking

This function starts to track a car. For more explanation refer to chapter .....

```
1 function trackClick() {
2         MyTracking.performClick();
3     }
```

Listing 4.9: Car tracking function

4.3.1.6 Review

This method is implemented with Java Android. More about in chapter...

```
1 function reviewClick(){
2         Review.performClick();
3     }
```

Listing 4.10: Review function

4.3.1.7 Turn off

This function is implemented with Java Adroid and has been documented in 4.1)Start Menu.

```
1 function turnOff(){
2         Exit.performClick();
3     }
```

Listing 4.11: Turn off funtion

4.3.1.8 Home

This function returns the user back to the start menu and is implemented with Java Android.

```
function home(){
        Home.performClick();
    }
```

Listing 4.12: Home function

4.3.1.9 Read car name

This method returns the name of the car that had been tracked threw a car specific id. Each transport has its own unique id. This id is predefined and set after the tracking was successful. Later it is stored in session storage.
So the input parameter *cname* is that specific id of the tracked or selected car. The name of the car is stored in the Navision server. A request had to be send to receive the name. More about Connectivity in chapter .....

```
function readcarname(cname){
        var test = '';
        $(document).ready(function () {
            $.ajax({
                type: "GET",
                url: "http://ar-tgm-navax.cloudapp.net:9090/rest/getKF
                async: false,
                dataType: 'JSONP',
                success: function(data){
                    test=data.split(';');
                    globalcarname=test[0];
                                        document.getElementById("add_f
                }
            });

        });
    }
```

Listing 4.13: Read car name function

4.3.1.10 Save email

As the name says, *saveEmail()* saves the email of the user. The information about users email was already stored in session storage through the function *setParam()*. Later this email is lend to the Navision server. More about connection

between app and server in chapter .....

```
function saveEmail(){
        var value3 = sessionStorage.getItem('email');
        $(document).ready(function () {
            $.ajax({
                type: "GET",
                url: "http://ar-tgm-navax.cloudapp.net:9090/rest/insertAr
                async: false,
                dataType: 'JSONP',
                success: function(data){
                    //do your stuff with the JSON data
                    var test=data;
                    console.log(test);
                }
            });
        });
```

Listing 4.14: Save email function

4.3.1.11 Save car

This function saves the id and the name of the tracked vehicle. This method is used for adding new cars to users car collection. In this function a feature called local storage that provides HTML5 for its users, was used. The function can be split into four phases.

Phase one checks if the input parameter *name* is not empty. If it is empty user receives information about it, otherwise it processes with the other phases.

```
if(name!=null){
.....
}else{
alert("Function is loading due to slow internet connection.");
}
```

Listing 4.15: Phase one

Phase two is the search phase. It searches for unique local storage place threw specific name *(favorites ,fcarna)* and inspects if the storage with the name exists. If it doesn't exists an empty array is put inside the two local storages, else nothing happens.

```
ififif ((localStorage.getItem("favorites") === null) && (localStorage.getIt
```

```
2           var names = [];
3           localStorage["favorites"] = JSON.stringify(names);
4           localStorage["fcarname"] = JSON.stringify(names);
5   }
```

Listing 4.16: Phase two

In phase three variables *storedIds* and *storedNames* are filled with information inside the local storage *favorites* and *fcarname*.

```
1   var storedIds = JSON.parse(localStorage["favorites"]);
2   var storedNames = JSON.parse(localStorage["fcarname"]);
```

Listing 4.17: Phase three

Phase four checks if the car exists in the local storage. If it does the user receives information that this car already exists in the favourite list, else the id and car name is saved into the local storage.

```
1   if (storedIds.indexOf(id) > -1) {
2           Notifier.error('Car already exists.');
3   }else{
4           storedIds.push(id);
5           storedNames.push(name);
6           localStorage["favorites"] = JSON.stringify(storedIds);
7           localStorage["fcarname"] = JSON.stringify(storedNames);
8           Notifier.success('Car has been added.');
9   }
```

Listing 4.18: Phase four

The listing 4.19 shows the hole function with its four phases.

```
1   function LocalStorageWriteId(id,name){
2                   if(name!=null){
3                           if ((localStorage.getItem("favorites") === nul
4                                   var names = [];
5                                   localStorage["favorites"] = JSON.string
6                                   localStorage["fcarname"] = JSON.stringi
7                           }
8
9                           var storedIds = JSON.parse(localStorage["favor
10                          var storedNames = JSON.parse(localStorage["fcar
11
12                          if (storedIds.indexOf(id) > -1) {
13                                  Notifier.error('Car already exists.');
```

```
14                          }else{
15                                  storedIds.push(id);
16                                  storedNames.push(name);
17                                  localStorage["favorites"] = JSON.stringif
18                                  localStorage["fcarname"] = JSON.stringify
19                                  Notifier.success('Car has been added.');
20                          }
21                  }else{
22                          alert("Function is loading due to slow internet c
23                  }
24      }
```

Listing 4.19: Save car function

4.3.2 Used Functionalities

Several new technologies were used to create the start menu. In this chapter are all these technologies described. Some are linked to other chapters where they have already been explained.

4.3.2.1 Session Storage

Moreover ,the function called Sessionstorage has a big importance in this project. The Sessionstorage saves the value not persist, that means if the App is closed or has been ended so the value will be wiped of . In the next Session or if the App has been started , there will be then a new Sessionstorage. In this case the Project NAVAR uses the Sessionstorage to save the ID from the car which has been tracked. Sessionstorage allows to save a large amount of key/value pairs and lots of text. This feature is impossible to do it via cookie. This kind of functionality uses a protocol to save the Data. This protocol checks if the key and value is a string, but if not it convert them to a string. Furthermore if a key was already present, its entry has to be removed and the new one will be appended. The sessionStorage has it own methods for specific functionality. First one is method is used to tell how many key/pair the SessionStorage contains. This method is same function ,which tells the length of an Array.

4.3.2.2 Local Storage

HTML5 provides us with a new feature called Web Storage. In other words, with it web pages can store data locally within the user's browser or mobile application.

Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used ONLY when

asked for. It is also possible to store large amounts of data, without affecting the website's performance. [23]

The data is stored in name/value pairs, and a web page can only access data stored by itself. Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server. [23]

HTML5 Web Storage provides two new objects for storing data on the client:

1. window.localStorage - stores data with no expiration date [23]
2. code.sessionStorage - stores data for one session (data is lost when the tab is closed) [23]

method/attribute	args	returns
setItem	String key, String value	
getItem	String key	String value
removeItem	String key	
clear		
key	int index	String key
length		int length

Figure 4.4: Methods and attributes of local storage [4]

Here is an example of setItem and getItem in local storage.

```
1 var foo = localStorage.getItem("bar");
2 // ...
3 localStorage.setItem("bar", foo);
```

Listing 4.20: setItem example (Adapted from [5])

In these application not a string but an array is stored inside the local storage. Here is an example how we put an empty array into a local storage.

```
1 var names = [];
2 localStorage["favorites"] = JSON.stringify(names);
```

Listing 4.21: array into local storage

Here an example how we received the array form local storage.

```
1 var storedIds = JSON.parse(localStorage["favorites"]);
```

Listing 4.22: start timer function

4.3.2.3 AJAX

More about AJAX in chapter .....

4.3.3 GUI

The GUI provides the user with 8 operations and a slide panel. Each operation is a button which invokes self created functions that are described in chapter 4.3.1)Created Functions.

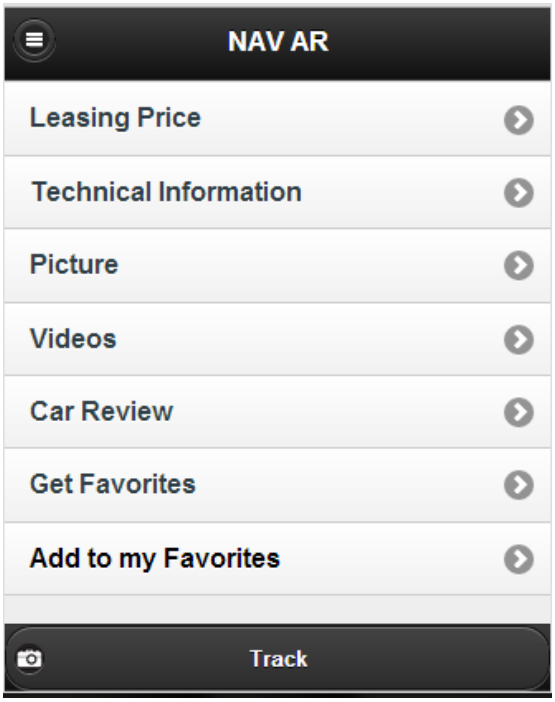

Figure 4.5: GUI of the main menu

```
1 <ul data-role="listview" >
2             <li><a href="leasingprice.html" rel="external">Leasing Pric
3             <li><a href="technicalinfo.html" rel="external">Technical
4             <li><a href="slide.html"  data-transition="slide" rel="ext
5             <li><a href="video.html" rel="external">Videos</a></li>
6             <li><a href="#" rel="#"  onclick="reviewClick();">Car Revie
7             <li><a href="myfavourite.html"rel="external" >Get Favorites
8                         <li><a id="add_favorite" onclick="LocalStorageV
9 </ul>
```

Listing 4.23: GUI source code

4.3.3.1 Leasing Price

When the user presses on button **leasing price** he will be linked to the html page *leasingprice.html* where he receives informations about the specific vehicle.

More about functions that had been used in leasing price in chapter .....

Figure 4.6: Leasing Price

4.3.3.2 Technical Information

By calling **technical information** facts about a specific car are presented. To create it several new features had to be used. Phone Gallary and Dynamic Selection of Colour. Informations about those are featured in chapters ....

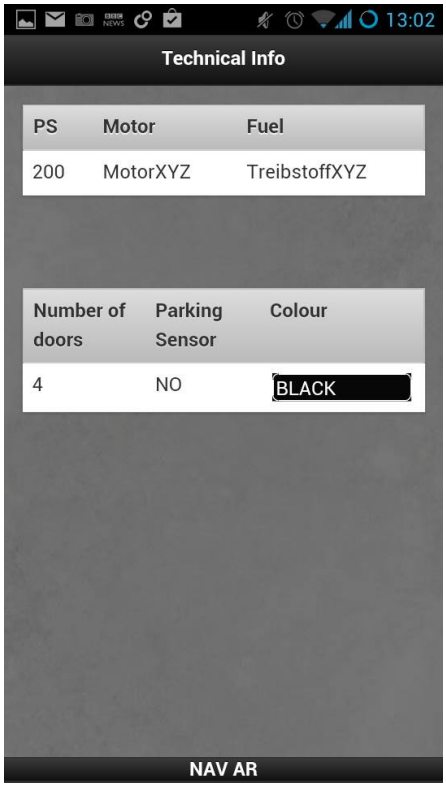

Figure 4.7: Technical Information

4.3.3.3 Pictures

Has freshest pictures of the specific car. Feature called Photo Gallery in chapter ..... was used to create this option.

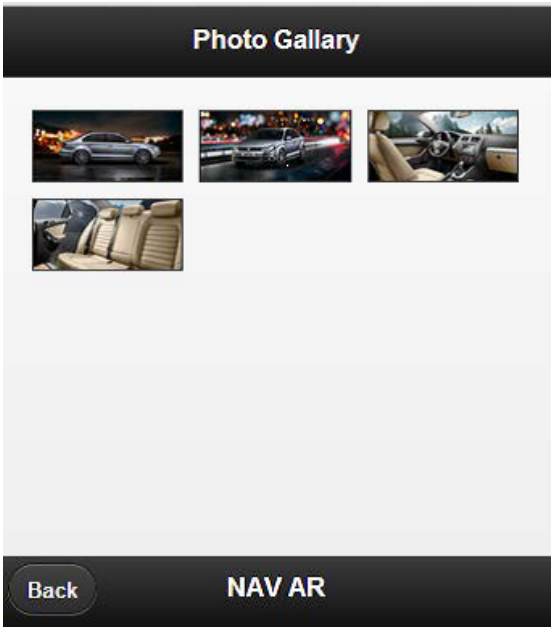

Figure 4.8: Pictures

4.3.3.4 Videos

This option provides the user with videos about the selected car from favourite list or fresh tracked one. More about video gallery in chapter.....

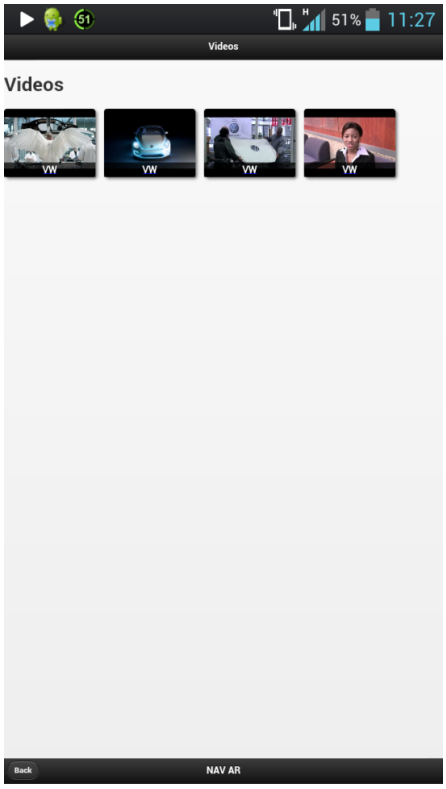

Figure 4.9: Video Gallery

4.3.3.5 Review

Operation review invokes a self created method called *reviewClick()*. Description to thisit is in chapter 4.3.1 Created Methods, Review. Basically review links the user to a new display where he can read review about the specific car.

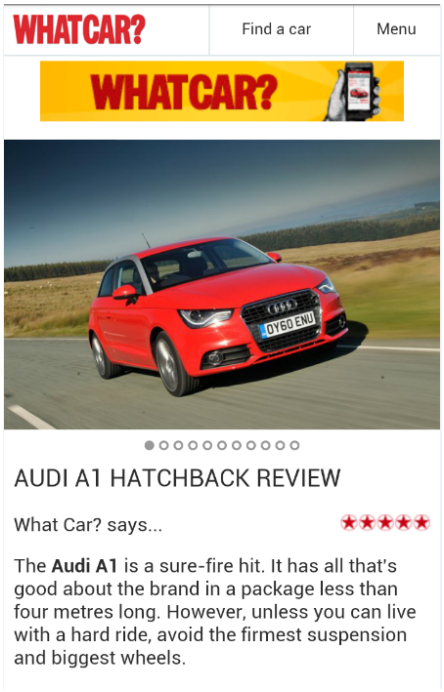

Figure 4.10:

4.3.3.6 Get Favourites

This operations links the user to his favourite cars which he saved with the option **add to my favourite**. Information about the favourite list in chapter 4.4)My Favourites.

Figure 4.11: Favourite list

4.3.3.7 Add to my Favourites

The button **Add to my Favourites** trigger the function *LocalStorageWriteId()*. It saves the id and the name of the tracked car into users favourites. For the first parameter it takes the tracked car id from the session storage. For the second parameter the global variable *globalcarname*.

```
1 <li><a id="add_favorite" onclick="LocalStorageWriteId(sessionStorage.ge
```

Listing 4.24: add favourite sorce code

Before the user can add the vehicle to his favourites he has to wait several seconds. In this time the request is send to the server for information about the car threw its id. If the user wants to access the operation in its loading time, the application denies him the access and informs him about the loading time.

Figure 4.12: Not ready function

The operations colour changes from red to black when the function is loaded.

Figure 4.13: Ready function

4.3.4 Slide Panel

In the upper left corner of the display exists a small button that calls the slide panel to open. More about slide panel itself in chapter .....

After opening the slide panel more options are available.

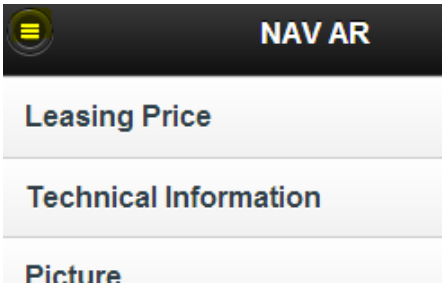

Figure 4.14: Slide panel

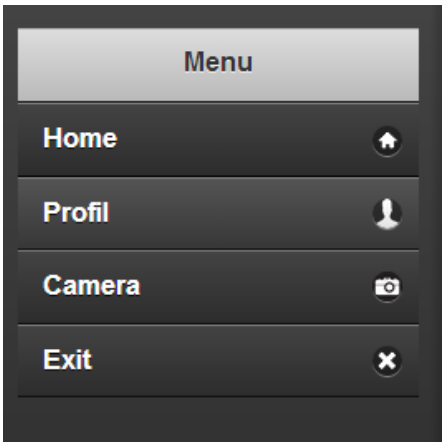

Figure 4.15: Options of slide panel

Here has the user four new options. He can return to the start menu with the display **home** or he can accesses his profile with **profil**. Also he can start to track a new car with **camera**. If the user doesn't want use the mobile application any more, he can close it the button **exit**.

```
1 <ul data-role="listview" data-theme="a" >
2 <li data-icon="home" ><a href="#" onclick="endtime();home();">Home</a></
3  <li data-icon="profil"><a href="profile.html" rel="external" >Profil</a>
4 <li data-icon="camera" ><a href="#" onclick="endtime();trackClick();">Ca
5 <li data-icon="delete"><a href="#" onclick="endtime();turnOff();">Exit</a
6 </ul>
```

Listing 4.25: Source code of slide panel options

The **home** button not only returns the user to the start menu but also ends the timer that has been started after a car was tracked. In addition, this timer is send straight to the Navision server.

Display **profil** calls to another display, in which the user can see his profile data.

Camera function ends the timer and starts the tracking function. Exit ends the application.

4.4 My Favourites

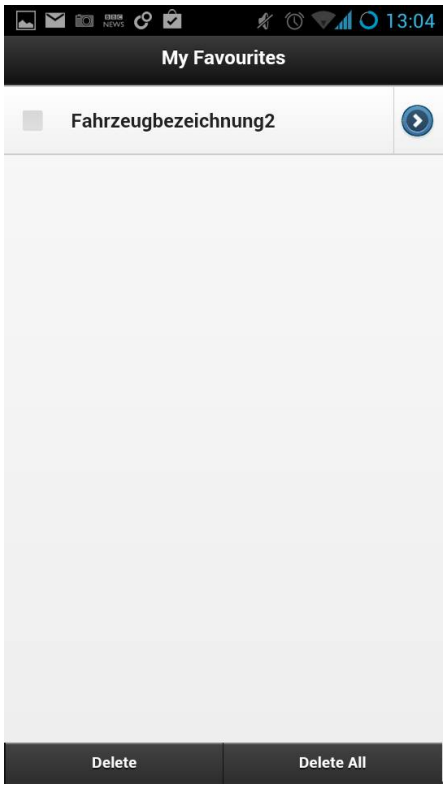

Figure 4.16: My Favourites display

Inside the favourites are cars had been added threw the operation **add to my favourites**. The favourite cars can be selected or deleted. Removing the cars from favourites is possible by selecting the specific vehicle or removing all of the cars.

4.4.1 Loading of favourite cars

Before the functions of the favourite list can be used, the list entries(cars) have to be initialized. This is done automatically when the site is loaded. Each line that is written inside document.ready starts after the document is ready. This is where the favourite cars are initialized.

```
$(document).ready(function () {
.
.
});
```

Listing 4.26: source for document is ready

First, all car names are loaded from local storage into an array called *storedCarNames*. So this array is filled with vehicle names which user added to his favourites.

```
var storedCarNames = JSON.parse(localStorage["fcarname"]);
```

Listing 4.27: Array with favourite cars

Now the filling of the cars into a list begins. The loop goes so long as the number of cars in the array. In this loop a car name is put into the *listItem1* which is just a panel shown in figure 4.17.

Next *listItem1* is put into another list. This list is where all panels(favourite cars) are stored. Each new vehicle is put into the list.

```
for (var i = 0; i < storedCarNames.length; i++)
        {
            var key = storedCarNames[i];

            listItem1 = '<li><a style="padding-top: 0px;padding-bottom: 0
                    '<label style="border-top-width: 0px;margin-top: 0px;
                    '<fieldset data-role="controlgroup" >' +
                '<input id="' + i + '" name="SelectedSensors[0].Value" ty
                '<input id="SelectedSensors_0__Id" name="SelectedSensors
                '<label for="SelectedSensors_0__Value" style="border-top-
                '<label  style="padding:10px 0px 0px 10px;">' +
                key +
                '</label>' +
                '</label>' +
                ' </fieldset>' +
            '</label>' +
            '</a><a onclick="EventHandler();" href="mainmenue.html" rel="
            '</li>';

            $('#liste').append(listItem1);
        }
```

Listing 4.28: Adding list items into the list

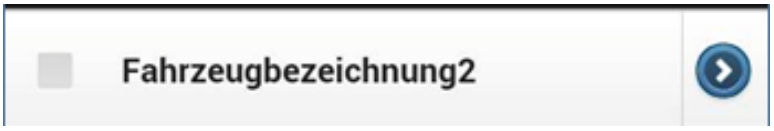

Figure 4.17: A list item

Last but not least the list that has to be refreshed so the list items are displayed.

```
1 $('#liste').listview('refresh').trigger('create');
```

Listing 4.29: Refreshing the list

4.4.2 JS Functions

4.4.2.1 Delete favourite car

This method deletes selected car with help of check box. If no car is selected, nothing happens by clicking on the button.

```
1  function deleteF(){
2
3          Array.prototype.clean = function(deleteValue) {
4                  for (var i = 0; i < this.length; i++) {
5                  if (this[i] == deleteValue) {
6                          this.splice(i, 1);
7                                  i--;
8                          }
9                  }
10                 return this;
11         };
12
13         var storedNames = JSON.parse(localStorage["favorites"]); //the
14     var storedCarNames = JSON.parse(localStorage["fcarname"]); //the na
15         var lengthof=0;
16
17         for(var s=0;s<storedNames.length;s++){
18                 if(document.getElementById(s).checked){
19                         delete storedNames[s];
20             delete storedCarNames[s];
21                         lengthof++;
22                 }
23         }
```

```
24          if(lengthof!=0){
25                  Notifier.success('Cars deleted.');
26
27                  storedNames.clean(undefined);
28                  storedCarNames.clean(undefined);
29
30                  localStorage["favorites"] = JSON.stringify(storedNames);
31                  localStorage["fcarname"] = JSON.stringify(storedCarNames
32          }
33          window.location.reload();
34  }
```

Listing 4.30: Delete function

4.4.2.2 Delete all favourite cars

Removes all favourite cars without selecting them.

```
1  function deleteAll(){
2      Notifier.success('All cars have been deleted.');
3          localStorage.clear();
4      window.location.reload();
5  }
```

Listing 4.31: Delete all function

4.4.2.3 Select favourite car

Each car inside the favourite list can be selected. After the car is selected, the user is linked to the start menu.

```
1  function EventHandler() {
2          Notifier.success('Car is selected.');
3          var id = this.id;
4
5          var storedNames = JSON.parse(localStorage["favorites"]);
6
7          for(var i=0; i<storedNames.length;i++){
8                  if(id==i){
9                          window.sessionStorage.setItem('id', storedNames[i
10                 }
11         }
12 }
```

Listing 4.32: Select car function

4.5 About

The **about** display has no logic and no self made functions except one the back button which functionality you have learnt in 4.2)Help chapter.

Figure 4.18: About display

CHAPTER 5

Implementation in Android Java and Metaio Tracking

5.1 Android Platform

We choose Android because it is the most popular mobile platform and we already had experience in developing apps for android.

5.1.1 Android Operation System

Android is an operating system based on Linux with a Java programming interface.

Android is currently primarily developed by Google.

Android allows background processing, provides a rich user interface library, support s 2-D and 3-D graphics using the OpenGL libraries, access to the file system and provides an embedd ed SQLite database. [24]

5.1.2 Android user interface components

The most important user interface components ion android are:

1. **Activity:** An Activity is the visible UI of an Android application. It contains so called *widgets* for example buttons text-fields labels etc. to build a user interface. [24]

2. **Fragments:** Fragments are components which run in the context of an Activity. Fragments however make it easier to use build UIs for different sized devices. [24]

3. **Views and layout manager:** Views are user interface widgets, e.g. buttons or text fields. They have attributes which can be used to configure their appearance and behavior. (colour, size, onClick action, ..) [24]

4. **Layout XML:** The user interface for Activities is typcally defined via XML files (layout files). [24] But in this project we used HTML 5 and JQuery to make sure that the app could easily be ported to an other mobile platform just by re-writing the tracking logic to the specific system.

5. **Android Webview:** The WebView class is an extension of Android's View class that allows to display web pages as a part of the activity layout. [25] As we already mentioned we choose this solution over plain Java-Android to achieve more platform independents.

5.1.3 Develop an Android Application

First of all the **Android SDK** is needed. The Android Software Development Kit contains the necessary tools to create, compile and package Android application. [24]

A compiled Android app is a **.APK** file which can be installed on a Android device.

The SDK also provides the **Android debug bridge** (adb).A tool which allows to connect to an virtual or real Android device. [24] We used this tool to test and debug our application.

5.1.4 Android Manifest

Every application must have an AndroidManifest.xml file in its root directory. The file describes essential information about the app. [26]. For instance all activities, the app name, starting activity, size, etc.

It also contains al list of security permissions. We had to add the camera, internet, read_owner_data and get_accounts permissions to implement all functions of our application.

Listing 5.1: extracts from our AndroidManifest

```
1 <uses-permission android:name="android.permission.CAMERA" />
2 <uses-permission android:name="android.permission.INTERNET" />
3  <uses-permission android:name="android.permission.READ_OWNER_DATA" />
4 <uses-permission android:name="android.permission.GET_ACCOUNTS" />
5  ...
6 <!-- define an activity -->
```

```
<activity
        android:name=".TrackLogic"
</activity>
```

5.1.5 Creating an activity

Every Android activity has to extend the **Activity** class and override the **onCreate** Method. onCreate gets executed when the activity is initialized.

5.1.6 Android web-view

We used the Activity sub-class **DroidGap** from **phonegap** to build activities. With this class it is easy to load webpages as view.

Listing 5.2: extracts from our source code

```
public class MainMenue extends DroidGap {
        ...
        public void onCreate(Bundle savedInstanceState)
        {
                ..
                super.loadUrl("file:///android_asset/www/mainmenue.html"
                ...
        }
}
```

The method **loadUrl** loads any web-page, doesn't matter if the URL is internal or external. It is also possible to pass an additional int time out parameter.

5.1.7 Java JavaScript Communication

In the section above we mentioned that we used html pages to create the view of our application. Because of that we had to find a way to pass variables through, the in Java written activity, to the web-view. We accomplished that by using JavaScript. Again the activity has to extend the **DriodGap** class.

First javascript must be enabled:

```
super.appView.getSettings().setJavaScriptEnabled(true);
```

Now its possible to pass variables to the html view:

```
int a=1;
super.loadUrl("javascript:{var myVariable=\""+a+"\";}");
```

5.1.7.1 JavaScript Android Interface

With such an interface it is possible to call activity methods form JavaScript. We needed them to react on button clicks from inside of the html view.

First we had to write a JavaScript function:

```
<script language="Javascript">
function trackClick() {
        MyTracking.performClick();
    }
</script>
..
<!-- set onCLick Action -->
<a href="#" onclick="trackClick();">Camera</a>
```

Then add a JavaScript interface to the activity:

```
super.addJavascriptInterface(new Object()
{
        public void performClick()
        {
                //react to button click
        }
},"MyTracking");
```

5.2 Working with the Metaio SDK

We used the Metaio SDK track and identify 3D or 2D Objects.

5.2.1 Who is Metaio

> "Metaio is the worldwide leader in Augmented Reality research and technology. Serving over 80,000 developers with over 1,000 apps for enterprise, marketing, retail, publishing and industrial cases, Metaio's AR software reaches over 30 Million consumers across the world."
>
> (Metaio GmbH)

5.2.2 Other Augmented Reality SDK's

There are many other augmented reality technologies on the market:

1. **Qualcomm Vuforia**: The Vuforia platform is mostly used for image recognition.

2. **Total Immersion D'Fusion**: D'Fusion is the world?s most widely-used commercial Augmented Reality solution. Unfortunately there are no free versions.

3. **Wikitude**: Wikitude is a powerful augmented reality sdk.However the cheapest SDK version costs 99$.

4. **String**: String only recognises framed images. So this SDK was unusable for our application.

[27]
In the End we choose Metaio SDK not only because our client recommended it but also because Metaio is a very powerful SDK. Unlike like most other technologies it has a free version and can track 3D objects.

Metaio also has a very big user community with a help-desk and many tutorials.

5.2.3 Metaio Toolbox

The metaio Toolbox is an application used to create or edit 3D tracking maps of all textured objects in your surrounding.The created Maps than can be used in the Metaio SDK. We used the Toolbox-App to create 3D Maps of the cars our application is tracking.

The Toolbox also allows you to play or edit AREL scenes. The AREL scene created in the metaio Creator can be directly played in the Toolbox. The geometries transformations can be edited in the Toolbox.

Furthermore, the Toolbox also has camera calibration function that allows you to determine camera parameters of your device. The Metaio Toolbox is a simple APP which is a available for Android and IOS. The APP can simply be downloaded and installed from Apple App-Store or Google Play. [28]
Process of creating a Map:

1. Download and install Metaio Toolbox.

2. Open The App.

3. Tab on *3D* Maps and than on *new Map*. (It is also possible to edit existing map)

4. After that the camera opens and a Object can be tracked.

The red dots in the figure show the tracking points, the *FEATURES* count shows how many tracking points have been placed. The more points the better can the object later be tracked by an application.

Figure 5.1: Metaio Tool Box

5.2.4 AREL (Augmented Reality Experience Language)

AREL (Augmented Reality Experience Language) is a JavaScript binding of the metaio SDK's API in combination with a static XML content definition
With Areal Scenes its possible to create a script with all tracking object and their behaviour, that scene than can be run by any metaio SDK. That's how you are able to create one platform independent Augmented Reality experience with AREL instead of using platform specific programming languages.
AREL consists of the following parts:

A XML Part which defines the content that should be loaded (3D-Models,Maps) and their size, position, transformation, etc.

The HTML 5 layer, this part provides graphical user interface and interacts using the JavaScript bridge with the metaio SDK. The AREL JavaScript bridge is a javascript library that allows to communicate with the Metaio SDK. All callbacks from the SDK are forwarded to the JavaScript Logic. [28]

However AREL scenes where not use in the project because they only offered a limited size of Actions. For instance we needed to get the ID of the tracked object to determine which car had been recognized. It turned out that this is a much more difficult process in AREL so we wrote the whole logic in Java using the Metaio SDK.

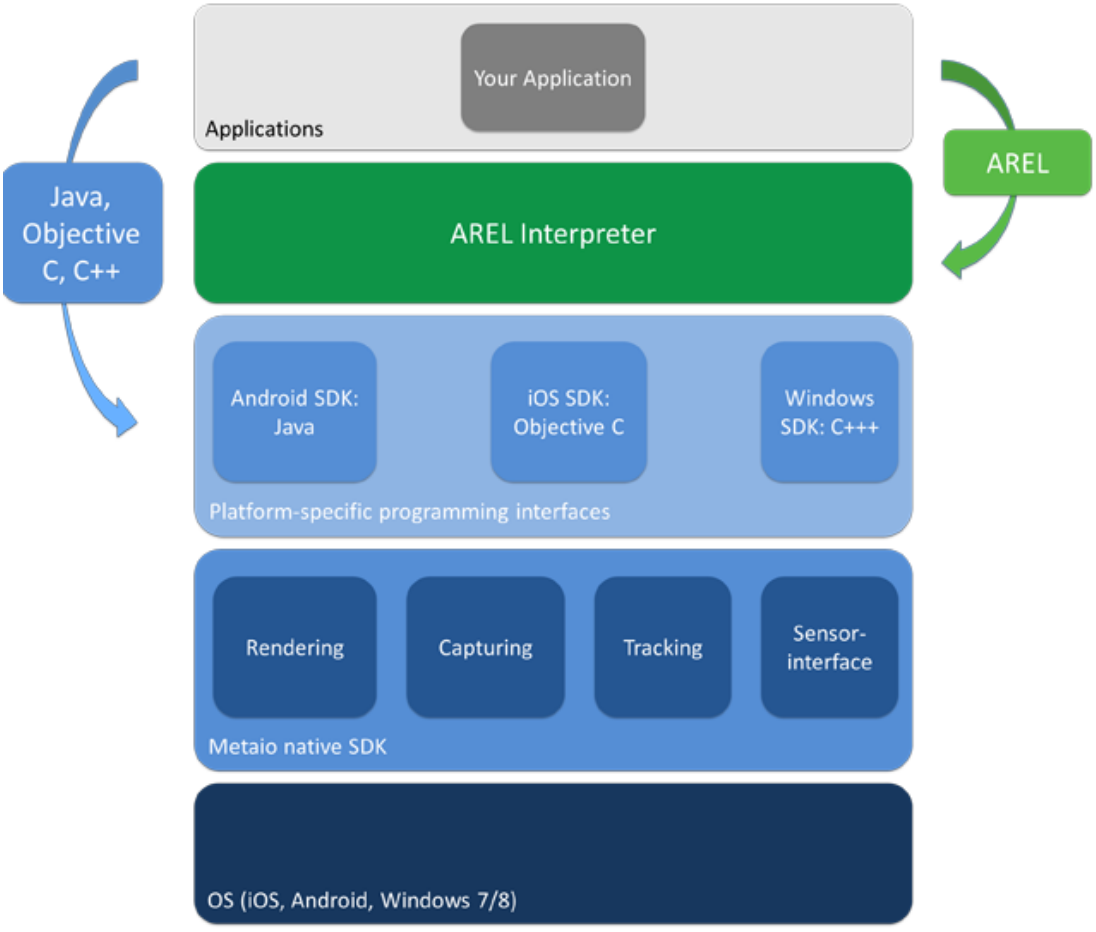

Figure 5.2: AREL

5.2.5 Tracking XML

The Tracking XML file is an XML file which can be created by the Metaio Creater or written by hand. It contains all of the tracking data. (3D Maps, etc. ..)

Listing 5.3: Tracking XML example

```
1  <?xml version='1.0' encoding='UTF-8'?>
2  <TrackingData>
3          <Sensors>
4                  <Sensor subtype='ML3D' type='FeatureBasedSensorSource'>
5                          <SensorID>TrackingObject 1</SensorID>
6                          <Parameters>
7                                  <featureorientationassignment>regular</
8                          </Parameters>
9                          <SensorCOS>
10                                 <SensorCosID>08921d8412c4de43d04eefc13c
11                                 <parameters>
12                                         <numextensiblefeatures>0</nume
13                                         <mintriangulationangle>6</mint
14                                         <map>08921d8412c4de43d04eefc13c
15                                         <MinMatches>15</MinMatches>
16                                         <DesiredMatchesRatioExtensible>
17                                         <NumExtensibleFeatures>250</Nu
18                                 </parameters>
19                         </SensorCOS>
20         </Sensors>
21 </TrackingData>
```

All tracking Objects have to be added into the **Sensors** element. Every new object needs a **SensorCosID**, the name of the object. The **<Parameters>** element describes the 3D Map for the object. The map has to be in the same folder like the tracking xml. Other parameters are for instance the **<MinMathes>** element, it describes how many points have to match so that the object gets recognised. Metaio has not really got a documentation for the Tracking XML therefore we choose to let the Metaio Creator create the xml file.

5.2.6 Extracting the Tracking XML in Android

In order to use xml file with the Metaio SDK, the tracking xml has to be placed in the **assets** folder of the android project.

The files than have to be extracted by the *AssetsManager* to make them accessible to the metaio SDK. This has to bee done in an Android *AsyncTask*.

The AsyncTask enables proper and easy use of the UI thread. This class allows to perform background operations and publish results on the UI thread. AsyncTasks should ideally be used for short operations (a few seconds at the most.) For threads that need to be running for long periods of time, it is recommended you use the various APIs provided by the java.util.concurrent pacakge such as *Executor*, *ThreadPoolExecutor* and *FutureTask*. [29]

Listing 5.4: Extracting Assets

```
1  private class AssetsExtracter extends AsyncTask<Integer, Integer, Boolean
2  {
3          @Override
4          protected Boolean doInBackground(..
5          {
6                  try
7                  {
8                          // Extract all assets and overwrite existing file
9                          AssetsManager.extractAllAssets(...);
10                 }
11                 catch (IOException e)
12                 {
13                         //Error Messages
14                 }
15
16                 return true;
17         }
18
19         Override
20         protected void onPostExecute(Boolean result)
21         {
22                 // Asset Path to the Tracking xml
23                 final String arelConfigFilePath =
24                   AssetsManager.getAssetPath("Tracking.xml");
25                 //Starting new Activity
26                 Intent intent = new Intent(getApplicationContext(),
27                   TrackLogic.class);
28                 intent.putExtra(getPackageName()+".AREL_SCENE",
29                   arelConfigFilePath);
30                 startActivity(intent);
31
32                 finish();
33           }
34 }
```

The AsyncTask gets executed like this:

Listing 5.5: executing AsyncTask

```
1 AssetsExtracter asyncTask=new AssetsExtracter();
2 //executing the task
3 asyncTask.execute(0);
```

Android first runs the *doInBackground* method, in this case the method extract all Assets using the Metaio AssetsManager: *AssetsManager.extractAllAssets(...);*. After this method ends successfully the *onPostExecute* method gets called.

onPostExecute extracts the path to the tracking.xml and than starts the TrackingLogic class. This Activity is where the tracking happens.

5.2.6.1 Loading the Tracking XML

The TrackingLogic class in our project loads the tracking xml using the Metaio SDK:

Listing 5.6: executing AsyncTask

```
1 String filepath = AssetsManager.getAssetPath("Tracking.xml");
2 ...
3 // set the tracking configuration
4 metaioSDK.setTrackingConfiguration(filepath);
```

5.3 Track Objects

In order to run the camera and start the tracking process we wrote a new Activity which has to extend the Metaio **ARViewActivity**. Important methods provided by this class:

1. **loadContents():**This Method load the tracking xml.
2. **onTouch(View v, MotionEvent event):** By overwriting this method you can set what happens when the user touches the screen.
3. **onGeometryTouched(final IGeometry geometry):** Metaio provides the possibility to draw 3D and 2D Models on the screen while tracking with the camera on, the method determines what happens when this geometry gets touched
4. **onTrackingEvent(TrackingValuesVector trackingValues):**
   One of the most important methods.onTrackingEvent describes what happens when an object gets tracked successfully. We overwrote this method to get the ID of the tracked Object. After the object hast been tracked the Main Menue activity with all specific car informations gets started.

Listing 5.7: where the magic happens

```
1  public void onTrackingEvent(TrackingValuesVector trackingValues)
2  {
3   //Check through all maps
4   for (int i=0; i<trackingValues.size(); i++)
5   {
6    final TrackingValues v = trackingValues.get(i);
7    if (v.isTrackingState())
8    {
9          Intent inte=new Intent(getApplicationContext(),MainMenue.class);
10         //get id of tracked object
11         inte.putExtra("id",""+v.getCoordinateSystemID());
12         startActivity(inte);
13         //close activity
14         finish();
15   }
16
17  }
18 }
```

CHAPTER 6

Design Concept

6.1 Video Gallery

The video Gallery in our mobile-application gives the user the possibility to watch test reviews and commercial videos of the tracked car.

For implementing the YouTube car video gallery we used **jquery** and the jquery plug-in **jquery.youtubevideogallery**. The Design and CSS files were taken from Jack Moore's plug-ins. [30]. His great work makes it possible to dynamically scale the size and position of the video-boxes so that they perfectly fit on every device screen!

As we can se in the figure on the next page the position and size automatically adjust to the 3 different screen sizes.

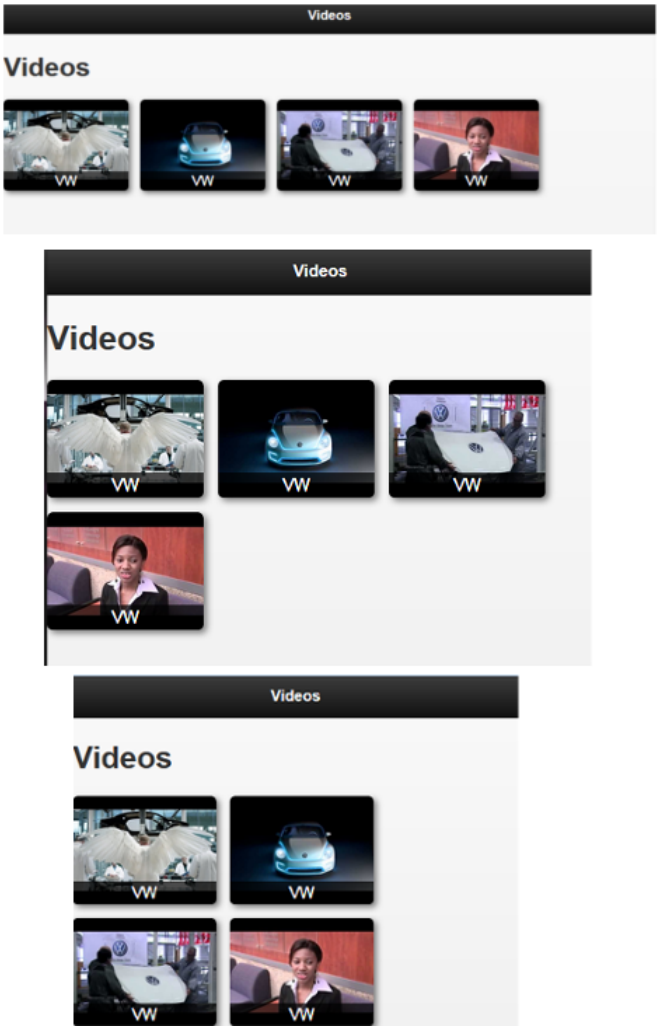

Figure 6.1: dynamically fit to device screen

6.1.1 Implementation

We wrote a JavaScript script that appends an array of different YoutTube video url to an **ul** tag which uses the **youtube-videogallery** class.

Listing 6.1: extracts from the video gallery src

```
1  <ul id="Gallery" class="youtube-videogallery">
2          <script>
3          ...
4          //append video urls
5          for (var i=0; i<videos.length; i++){
6                  $("#Gallery").append(videos[i]);
7          }
8          </script>
9  </ul>
10 <!-- Use jack Mores .js and css filess to make the gallery -->
11 <script>
12     $(document).ready(function(){
13         $("ul.youtube-videogallery").youtubeVideoGallery( {plugin:'colorb
14     });
15 </script>
```

6.2 hallo

my name is NASDAQ property

APPENDIX A

Appendix

Bibliography

[1] Mozilla. (2014) Java object model. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Details_of_the_Object_Model#Class-Based_vs._Prototype-Based_Language

[2] S. H. Park. (2012) Understanding jvm internals. [Online]. Available: http://www.cubrid.org/blog/dev-platform/understanding-jvm-internals/

[3] R. Wang. (2013) java run time data aarea 2. [Online]. Available: http://java.dzone.com/articles/jvm-run-time-data-areas

[4] P. Fox. (May 28 2012) The localstorage api. [Online]. Available: http://www.drdobbs.com/web-development/the-localstorage-api/240000682

[5] M. Pilgrim. (2014) The past, present & future of local storage for web applications. [Online]. Available: http://diveintohtml5.info/storage.html

[6] lol.

[7] G. Koppensteiner, R. Hametner, R. Paris, A. Passani, and M. Merdan, "Knowledge driven mobile robots applied in the disassembly domain," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, dec. 2011, pp. 52 –56.

[8] M. Merdan, "Knowledge-based multi-agent architecture applied in the assembly domain," Ph.D. dissertation, Vienna University of Technology, 2009.

[9] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE V2*. Wiley, 2007. [Online]. Available: http://www.amazon.ca

[10] Oracle. (2014) What is java. [Online]. Available: https://www.java.com/en/download/faq/whatis_java.xml

[11] [Online]. Available: http://en.wikipedia.org/wiki/Java_%28programming_language%29

[12] langpop. (2014) prgramm languages popularity. [Online]. Available: http://www.langpop.com/

[13] Oracle. (2014) Java object oriented programming. [Online]. Available: http://docs.oracle.com/javase/tutorial/java/concepts/

[14] D. Desert. (2012) Java abstract vs interface. [Online]. Available: http://stackoverflow.com/questions/10040069/abstract-class-vs-interface-in-java

[15] B. Evans. (2013) java performance. [Online]. Available: http://www.infoq.com/articles/9_Fallacies_Java_Performance

[16] K. Stern. (2012) Stack based cpu vs register based. [Online]. Available: http://markfaction.wordpress.com/2012/07/15/stack-based-vs-register-based-virtual-machine-architecture-and-the-dalvik-vm/

[17] [Online]. Available: http://stackoverflow.com/questions/3798424/what-is-the-garbage-collector-in-java

[18] OracleInc. (2014) Java jvm oracle. [Online]. Available: http://docs.oracle.com/javase/specs/jvms/se7/html/jvms-2.html

[19] J. papers. (2013) Java jvm run-time data areas. [Online]. Available: http://javapapers.com/core-java/java-jvm-run-time-data-areas/

[20] Axtavt. (2012) Java constant pool. [Online]. Available: http://stackoverflow.com/questions/10209952/java-constant-pool

[21] M. Rouse. (2005) Java jar archive. [Online]. Available: http://searchdatamanagement.techtarget.com/definition/JAR-file

[22] WebStorm. (2014) Webstorm - the smartest javascjava ide. [Online]. Available: http://www.jetbrains.com/webstorm/

[23] w3school. (2014) Html5 web storage. [Online]. Available: http://www.w3schools.com/html/html5_webstorage.asp

[24] L. Vogel. (2013) Android development tutorial. [Online]. Available: http://www.cs.virginia.edu/~cs201/labs/Vogella-Android-Development-Tutorial.pdf

[25] Google. (2014) Andoird web view. [Online]. Available: http://developer.android.com/guide/webapps/webview.html

[26] Google. (2014) Android manifest. [Online]. Available: http://developer.android.com/guide/topics/manifest/manifest-intro.html

[27] newdigimarketing. (2014) Augmented reality sdks. [Online]. Available: http://augmentedrealitynews.org/ar-sdk/top-5-augmented-reality-sdks/

[28] Metaio. (2014) Metaio toolbox. [Online]. Available: https://dev.metaio.com/sdk/toolbox/

[29] Google. (2014) Andriod async task. [Online]. Available: http://developer.android.com/reference/android/os/AsyncTask.html

[30] J. Moore.

Erklärung

Hiermit erklären wir, dass die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Oktober 2012

Name1

Name2

Name3

Name4