tgm
Die Schule der Technik

DIPLOMARBEIT

NAVAR

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
Aliaksei Korabach, 5AHITT

Wien, im Mai 2014

Abteilungsvorständin:	Prof. Dr. Günther ZANDRA
Tag der Reifeprüfung:	11.06.2014-12.06.2014
Prüfungsvorsitzende:	MinR Mag. Sabine NIEMEYER
Erster Gutachter:	Prof. Dr. Helmut VANA
Zweiter Gutachter:	Prof. Mag. Hans BRABENETZ

Projectteam

The following page shows,which teammember worked on which part of the diploma thesis.

Michael Fegerl

- Theoretical Basics
  - AJAX
  - Windows Azure
  - MS SQL Server
  - MS Dynamics NAV
- IDE
  - Visual studio
- Streaming
- Future enhancements and possibilities

Dominik George

- Introduction
  - Augmented Reality
- Theoretical Basics
  - JavaScript,HTML5,CSS
  - jQuery Mobile
- Logic Implementation with JavaScript
  - Features
- Design Concept

Aliaksei Korabach

- IDE
  - Webstorm
- Logic Implementation with JavaScript
  - JavaScript,HTML5,CSS
  - Display
  - Implemented Function
  - Features

Thomas Pokorny

- Kurzfassung
- Abstract
- Introduction
  - General
- IDE
  - Android SDK Eclipse
- Implementation in Android Java and Metaio Tracking
- Design Concept
  - Video Gallery

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 2014

Michael Fegerl

Dominik George

Aliaksei Korabach

Thomas Pokorny

Abstract

Augmented reality is a future oriented technology which allows to integrate digital information onto the view of the real-world. The potential uses for this technology are nearly endless, from video games, designers and retailers to health-care companies. Even if AR is a comparatively young technology their are already various different Applications which made use of it.

The purpose of the presented diploma thesis is to describe the usage of this technology in an Android mobile application. The project of developing the app was made in cooperation with our client 4relation Consulting GmbH. Our client recommended to use the Metaio SDK as AR-technology. Metaio is the worldwide leader in Augmented Reality research and technology with over 80 000 developers.

The main function of the mobile application is to track cars (tracking is the process of recognizing objects with a camera). After the successful recognition the user is able to view diverse information of the auto-mobile. These information get loaded from a NAV database in which they are stored. The client is going to show the application together with the NAV database connection in his business presentations

Kurzfassung

Das Thema der vorliegenden Diplomarbeit ist die Nutzung der Technik Augmented Reality beim Einsatz einer mobilen Applikation für den Auftraggeber 4relation Consulting GmbH. Augmented Reality ist ein besonders aktuelles Thema, wie durch die im Moment bevorstehenden Ankündigungen der generellen Verfügbarkeit von e.g. Google Glasses dokumentiert wird. Durch die gesteigerte Leistungsfähigkeit der für Mobile Phones und verwandte Geräte, wie Smartphones, Tablets, etc., eingesetzten modernen Prozessoren scheint die Zeit reif für diese fortgeschrittenen Anwendungen.
Der Auftraggeber empfahl uns die Metaio SDK als Augmented Reality Technologie zu verwenden. Diese SDK bietet die Möglichkeiten sowohl 2D als auch 3D Objekte zu erkennen. Dies wird in unserer Android Applikation (App) implementiert. Ziel der App ist es durch das Trackingsystem verschiedene Fahrzeugmodelle zu erkennen und dem User Informationen zu diesen anzuzeigen. Diese Informationen werden in einer NAV Datenbank abgespeichert. Nach erfolgreicher Fahrzeug-Erkennung werden die entsprechenden Daten heruntergeladen und am Display des jeweiligen Gerätes angezeigt.
Um solch eine Applikation umzusetzen werden Wissen und Technologien aus verschiedensten Teilen der Informatik benötigt. Unser Auftraggeber wird diese mobile Applikation zusammen mit in zentralen MS-Dynamics-NAV Servern gespeicherten Informationen ihren Kunden vorstellen. Dieses Anwendungsszenario wurde im Rahmen dieser Diplomarbeit entwickelt und implementiert.

Contents

List of Figures

Listings

Chapter 1

Introduction

1.1 General

Today everybody can create an App for any Platform. In this world, most of the pupils use Smartphones. So in that case this work describes from which modules or elements the App NAVAR is created of. Furthermore it shows how to use the Modules in General and how to deploy them. However for this project we used three programing languages. For the user Interface we used JQuery Mobile Framework. This Application has been developed for the Android platform. We wrote this application in Android because the client recommended it. The SDK for the Tracking system is implemented in this app and it is developed for Android and IOS.

The following point is, that the project uses Javascript,HTML5 ,CSS and the Framework JQuery Mobile for the user interface. The main problem was to combine the user interface with the tracking logic implementation. Further more the connection between the smartphone and the Navision Server was not so easy to implement. In this chapter it describes the programing languages and the Framework, that was have used for this project:

- Java
- JavaScript,HTML5,CSS
- jQuery Mobile
- C#

1.2 Augmented Reality

1.2.1 What is Augmented Reality ?

Augmented Reality(AR) is a type of virtual environment. It aims to duplicate the world's environment into 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. [9]

A good example for Augmented Reality are the sport games which will be live broadcasted. 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. [9]

1.2.2 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]

1.2.3 Companies for Augmented Reality

There are three major 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 company already created 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 of the mobile device and scans the car components, such as the steering of the car. Then it 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 the Swiss Federal Institute of Technology(ETH) in Zurich and the name of the product is Kooaba. It was founded in November 2006 . The mission of this product is to unlock the information which has been captured in images in using the sophisticated image recognition technology. This firm is a competitor to Metaio GmbH.

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. [11]

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. [11]

Augmented Reality is used by Smartphones and Google Glasses. Google glasses is one of the Gadgets which is used for the AR Technology. [4]

Figure 1.1: AR Example [1]

1.2.4 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 projects launch event was in 2012 .Google Glass is the name for a type of wearable computer. It was created by the Google's 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 Google'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. [2]

Figure 1.2: Google Glasses [2]

Google Glass has 7 main functionalities [12]:

The first one is, 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 tasks, without moving the hands of a user.

The second function is, it can record a video or it will take pictures, if the user just gave an oral command. So in that case the user never has 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 to googling with this device. If the person likes 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 such as the users child's 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 feature 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 and which route the user takes. 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.

[7]

Figure 1.3: Google Glasses Ability [3]

1.2.5 Usage of Augmented Reality

1.2.5.1 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, displays 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 operator's 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. [10]

1.2.5.2 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 have 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 an 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. [10]

A example of a smart car: [9]

Figure 1.4: Smartcar [4]

1.2.5.3 Medical

AR technology is quite popular in the medical field. This technology enables to become more common these days for surgeries. With AR the error rate are smaller in Surgery branches. 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, has data which 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. [10]

1.2.5.4 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 user's image and display it on the screen. The user can turn to view himself from all angles. [10]

1.2.5.5 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. [10]

1.2.5.6 Architecture

In this 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. [10]

1.2.5.7 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 *relive* events as they are known to have happened, while never leaving their class. [10]

1.2.5.8 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 it down on the canvas finally. [10]

1.2.5.9 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. [10]

1.2.5.10 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 forecaster's actions. AR are used by transmitting the forecast live to the viewers . [10]

1.2.6 Future of Augmented Reality

Augmented Reality is a growing up technology. It has amazing abilities, but few of them can't be implemented right now due to limitations in hardware and algorithms. [10]

1.2.6.1 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. [10]

1.2.6.2 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. [10]

1.2.6.3 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 *appear* in the seat of the room, along with the members. [10]

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 data streaming. [10]

1.2.6.4 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 into 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. [10]

1.2.6.5 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 the 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. [10]

To create an 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 written well to detect fine eye movements and to convert it to the right information. This algorithm has to filter other movements.

1.2.7 Summary

So we see that AR is a developing technology. The basic requirements for the technology for smarthphones is, that it has a back and front camera , GPS, accelerometer and compass. Most of the requirements are fulfilled by almost all Android devices on the market. Now it 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 is quite popular in many fields such as Military, Medical and Education.

Chapter 2

Theoretical Basics

2.1 Java

2.1.1 What is Java

Java is a computing platform and object oriented programming language first released by Sun Microsystems in 1995. Oracle has bought Sun in 2010. [13]

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 developer's 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. [14]

Java is next to C/C++ one of the most popular programming languages. [15] The language also has a similar syntax to C and C++.

2.1.2 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. [16] 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. [5] 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. [5] 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.

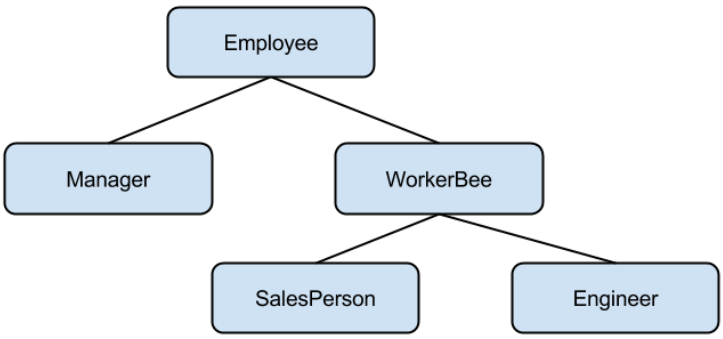

Figure 2.1: Java Subclasses

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. [17]

6. **Package:** A package is a namespace for organizing classes and interfaces. Packages make large software projects easier to manage. [16]

2.1.3 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.

2.1.4 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. [18]

2.1.5 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. [6]

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. [6] That means that the VM doest need to know the operand addresses, it only calls the Stack-Pointer which points to the current instruction. [19]

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. [20]

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.

[6]

2.1.6 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. [6]

2.1.7 Java Code Execution Process

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

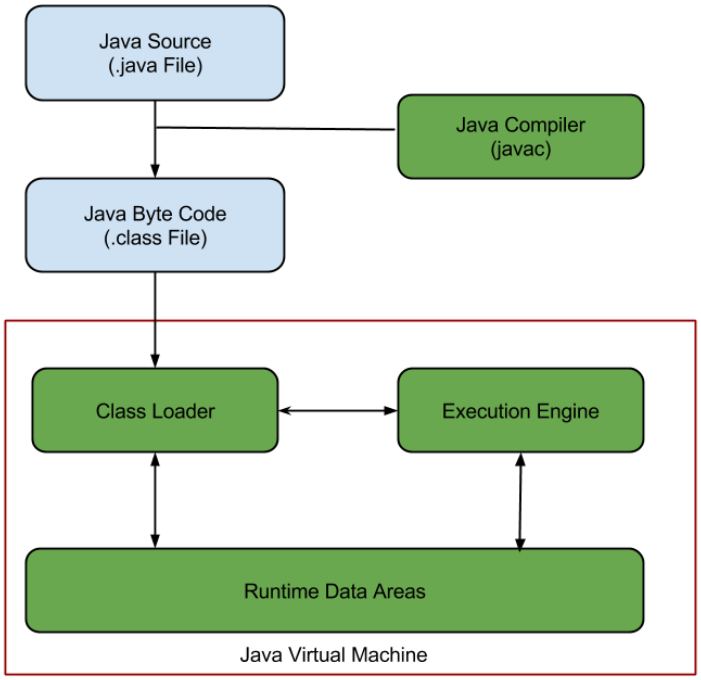

Figure 2.2: java code execution process

2.1.7.1 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. [6]

2.1.7.2 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. [6]

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). [21]

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. [6].

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** [21]

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. [6]

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 [22]

   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. [23]

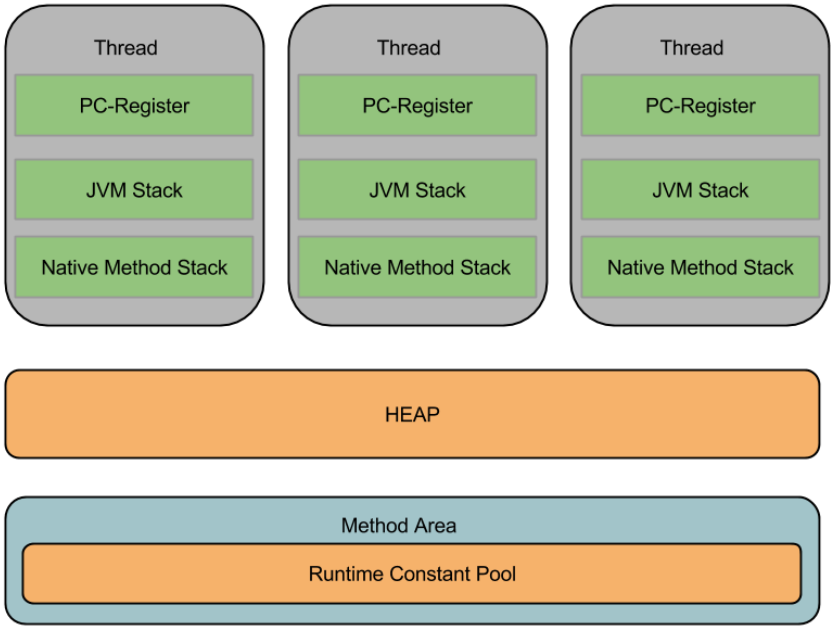

Figure 2.3: Java Run-time Data Areas

2.1.7.3 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. [6]

2.1.8 .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. [24]

2.1.8.1 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.

2.2 JavaScript, HTML5,CSS3

2.2.1 JavaScript

Web programing uses JavaScript. It is used to make the web page interactive. JavaScript is very usefull to change the content dynamically of a web page. It is also a expansion for HTML5 and CSS. JavaScript was developed in 1995 by Brendan Erich. JavaScript describes a dynamic typed, object oriented and classless scripting language. [25]

2.2.2 HTML5

Html5 is the latest standard version of HTML. HTML has also previous version such as HTML 4.01, came in 1999. In 1999 the internet has changed significantly.HTML5 was created to replace both HTML 4, XHTML and the HTML DOM Level 2. HTML5 is designed such that nobody has to use plugins. It is capable of: animations to graphics, music to movies, and it can also be used to build complicated web applications. The big benefit from HTML5 is ,it being cross-platform. It can be used for designing apps for PC, Tablet, Smartphone and TV.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new features of HTML5 is to play video and audio in a easier way. The next ability is to draw graphics. With HTML5 , web applications can be developed with helpful elements such as:

- Local data storage
- Local file access
- Local SQL database
- Application cache
- Javascript workers
- XHTMLHttpRequest 2

Furthermore HTML5 is able to use CSS3. [26]

2.2.3 CSS3

CSS3 is the latest version of CSS. The benefit of CSS3 is, it is completely backwards-compatible with earlier version of CSS. Moreover CSS3 has been split into 'modules'. However it contains old CSS specification, which has been split into smaller pieces .

The new Modules that has been added and which are most important in CSS3 are:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations

- Animations
- Multiple Column Layout
- User Interfaces

[27]
The listed modules are used in this project in combination with jQuery Mobile. JQuery will be described in the following chapter.

2.3 jQuery Mobile

jQuery Mobile is touch-optimized web framework. This framework was created by the JQuery Foundation. It is completely open source. This framework is also one of the most popular mobile frameworks. That's the reason why this project NAVAR uses jQuery Mobile framework. This framework was created by Jasper de Groot, Alexander Schmitz, Anne-Gaelle Colom, Gabriel Schulhof.

jQuery Mobile is a HTML5 based user interface system, it is designed to create responsive web sites and apps that are accessible on all smartphones, tablets and desktop devices. The framework is build on jQuery and jQuery UI foundation. It offers also AJAX navigation with page transitions, touch events and it has other interesting components and features. This framework is a lightweight code, so it has a flexible ,easily theme able design.

jQuery Mobile also updates their framework versions. To built a theme with the framework is not so difficult. The user interface has the components of jQuery Mobile. The components of jQuery will be explained in Chapter Design concept.

2.4 AJAX

AJAX (Asynchronous JavaScript and XML) is a method which is widely used in web development. Ajax uses several technologies like: JavaScript, CSS, XML and XMLHttpRequest. [8]

2.4.1 XML

"Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere." [28]

2.4.2 XMLHttpRequest

XMLHttpRequest is used in Ajax for exchanging data with a server. The main advantages of this request is that it can be used in the background for sending data to the server, updating a page without reloading it, request and receive data from the server after the page is loaded. [29]

2.4.3 How does AJAX work?

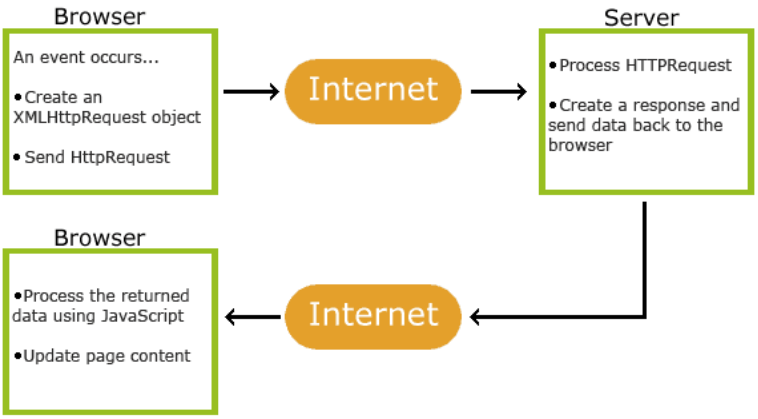

Figure 2.4: AJAX overview [8]

As described in the graphic, whenever a program needs to send data from the browser to the server it creates an XMLHTTPRequest object. This request is then accepted or denied by the server and a response will be sent back to the browser where it can be displayed or processed.

2.5 Windows Azure

"Azure is an open and flexible cloud platform that enables you to quickly build, deploy and manage applications across a global network of Microsoft-managed datacenters. You can build applications using any language, tool or framework. And you can integrate your public cloud applications with your existing IT environment" [30]

2.5.1 Overview

The typical Windows Azure environment consists of many components such as the Azure Service Platform, which can be several virtual machines with services. A software application which communicate with Azure over XML and other technologies. Several end users which access the Azure cloud over a browser and developers who access the cloud directly.

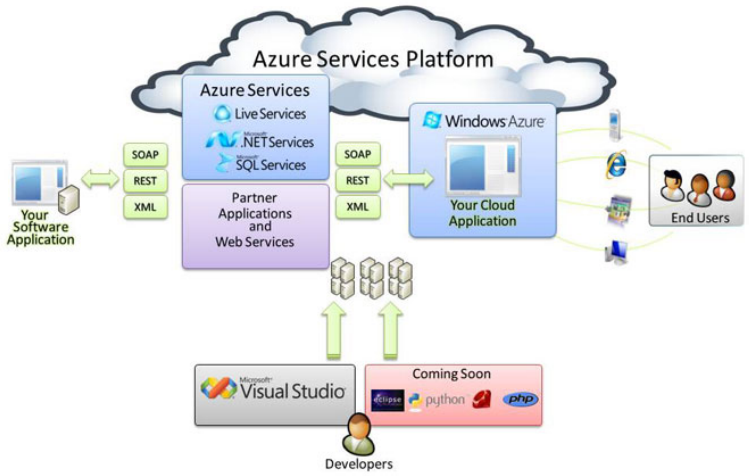

Figure 2.5: Azure overview
[31]

2.6 Microsoft SQL Server

Microsoft SQL Server is a relation database software for saving,modifying and providing data. Microsoft SQL Server can be deployed on a desktop server as well as in the cloud.It provides a edition for high-end performanc, critical applications and a lighter version for normal applications/data. [32]

There are five main releases of MS SQL Server:

- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- Microsoft SQL Server 2012
- Microsoft SQL Server 2014

[32]

2.6.1 Program Overview

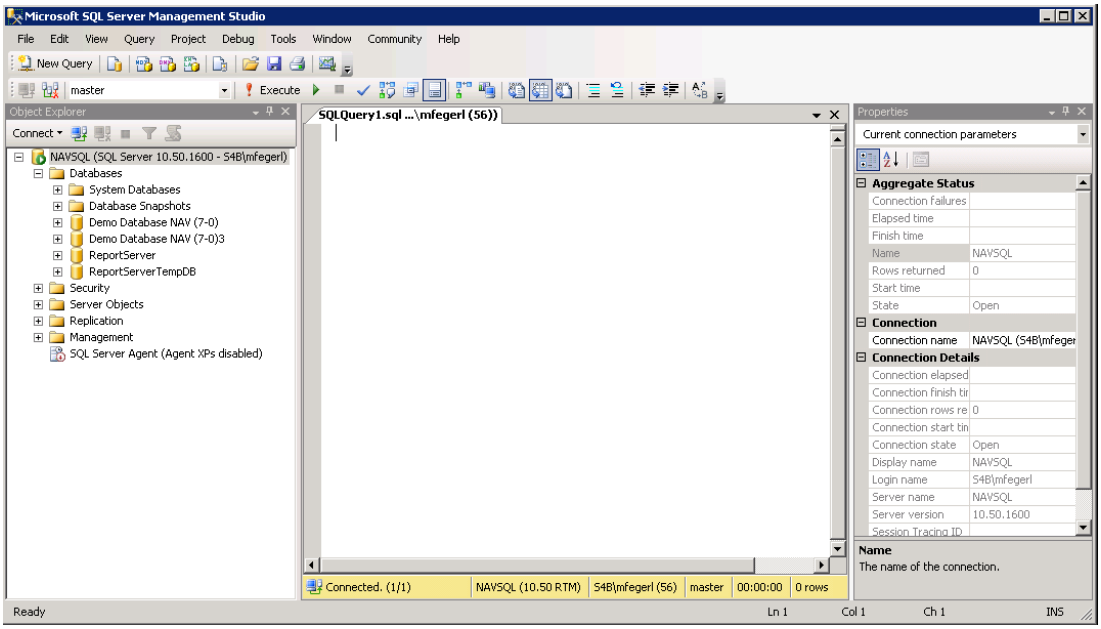

Figure 2.6: SQl Server 2008 overview

The figure shows the graphical user interface of Microsoft SQL Server Management Studio. It is used to maintain and configure SQL databases. In the left panel of the program the available databases are shown.The panel in the middle is a simple editor for database queries.On the right side the properties of the selected datbase are displayed.

2.7 Microsoft Dynamics NAV

Microsoft Dynamics NAV is an ERP(Enterprise Ressource Planing) software for small and medium sized corporations. It is a highly adaptable software which provides functionalities for managing a whole business. Such as sales, shipping, financing, project management, supply chain management, business intelligence,reporting and other services. The look and feel of the application is based on Microsoft Office to provide a simple entry point to the product if you are familiar with Microsoft Office. Microsoft NAV can be either installed on local servers as a 3-tier ,2-tier or 1-tier implementation as well as in the cloud to provide the best solution for a business. [33]

2.7.1 Program Overview

The following figure shows the graphical user interface of Microsoft NAV, the RoleTailored client(RTC).It shows the RTC for the role sales manager with the demo Database "CRONUS International Ltd." from Microsoft.

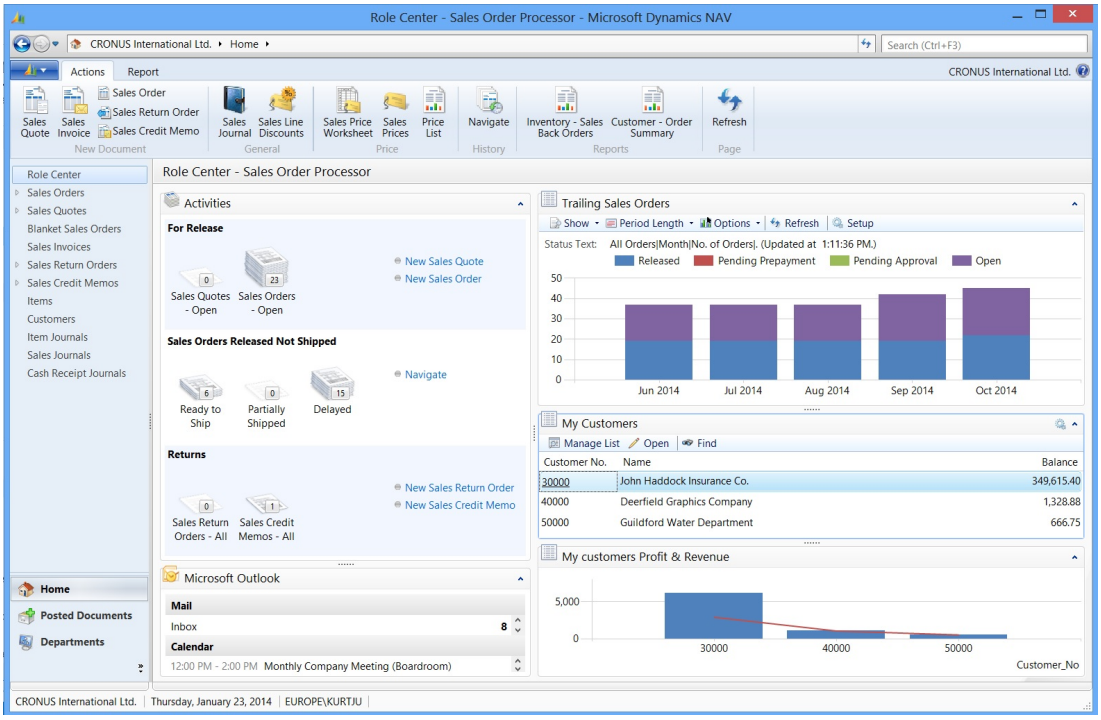

Figure 2.7: Program overview
[31]

2.7.2 Data structure

2.7.2.1 Tables

Microsoft Dynamics NAV saves data into a Microsoft SQL Server database.The databases consist of several tables, which can be created, edited and deleted. Table's are the basic modules of the database and are fundamental.It provides the functionality to modify,delete and display data on the run.

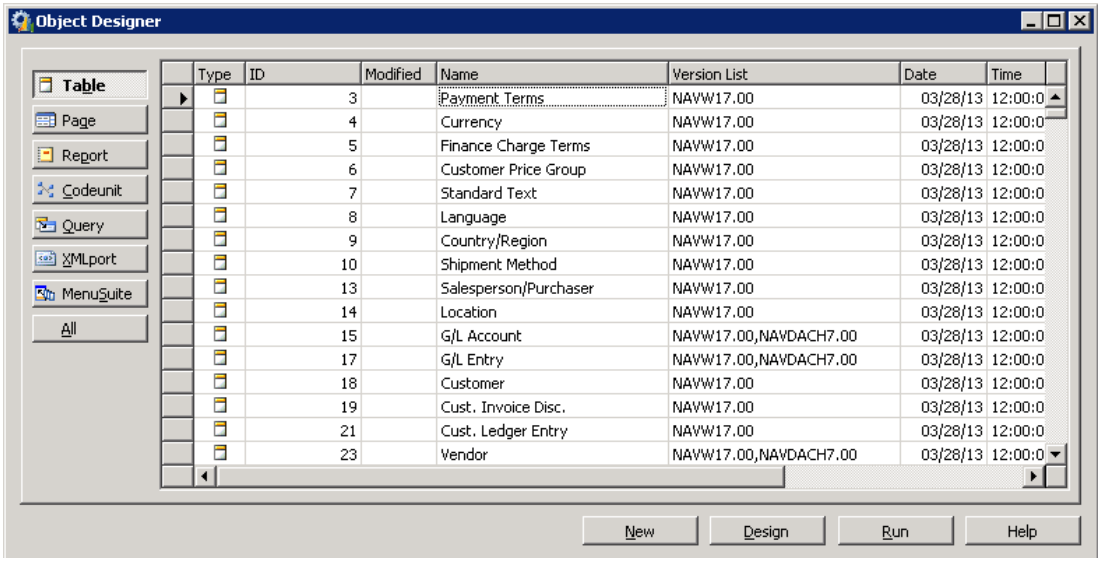

Figure 2.8: Table example

2.7.2.2 Pages

A page is a XML object which consists of several properties and code. It is used to display, structure and organize data. It can either be accessed via a client and displayed graphically or trough a web service.In this project the pages are used over a web service by the C# application. The usage of the pages are explained in the chapter "Streaming".

The following figure shows an example page with a list of customers.

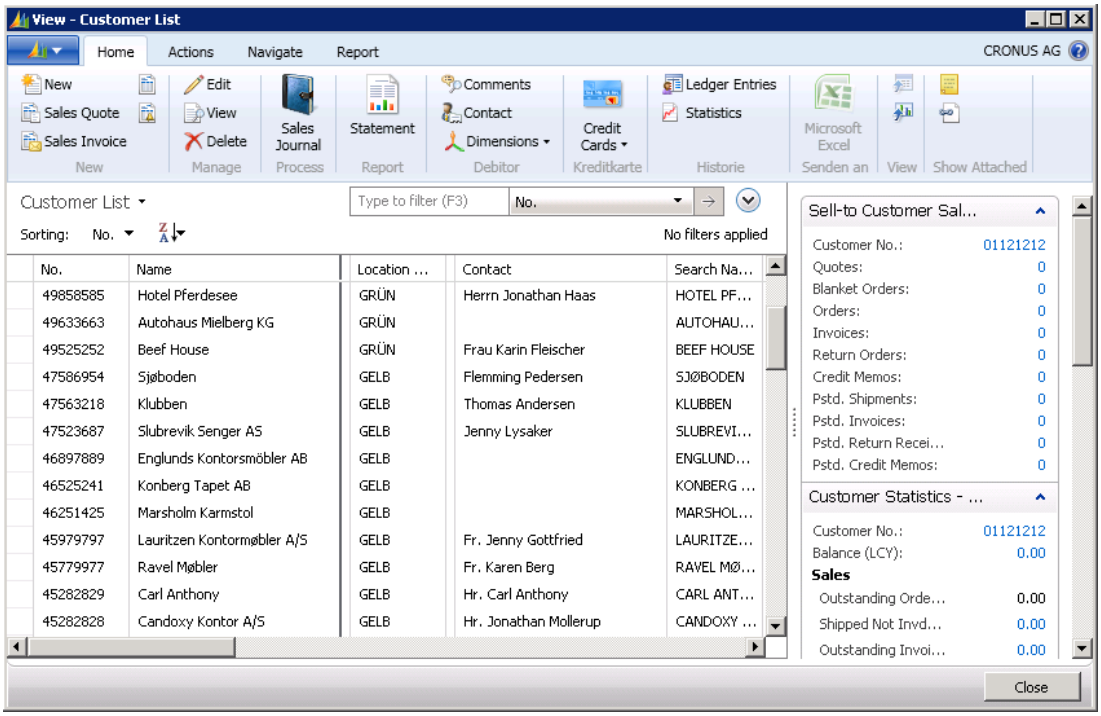

Figure 2.9: customer page example

This page displays the content of the table customers. It consist of several attributes such as the name and the telephone number of a specific person.The page provides the functionality to create, delete or modify the customers within the RTC.

Chapter 3

IDE

3.1 Andorid SDK Eclipse

Eclipse is an integrated development environment (IDE) mostly written in Java developed by the Eclipse Foundation.
Eclipse can be used as development environment for almost every common programming language.

3.1.1 Eclipse Workbench

The Eclipse GUI contains four different sections as one can see in the figure bellow

1. Overview of all Project inside the workspace
2. Text-editor with functions like: syntax-highlighting, auto-complete, source-generation.
3. The outline contains attributes, methods and classes of the selected file inside the editor.
4. The last sections shows the console-output

3.1.2 Android SDK

Their is an Android ADT (Adnroid Development Tools) plug-in available for Eclipse. ADT makes it easy to set up new Android projects, create an application UI, add packages based on the Android Framework API, debug applications using the Android SDK tools, and export signed .apk files in order to distribute applications.

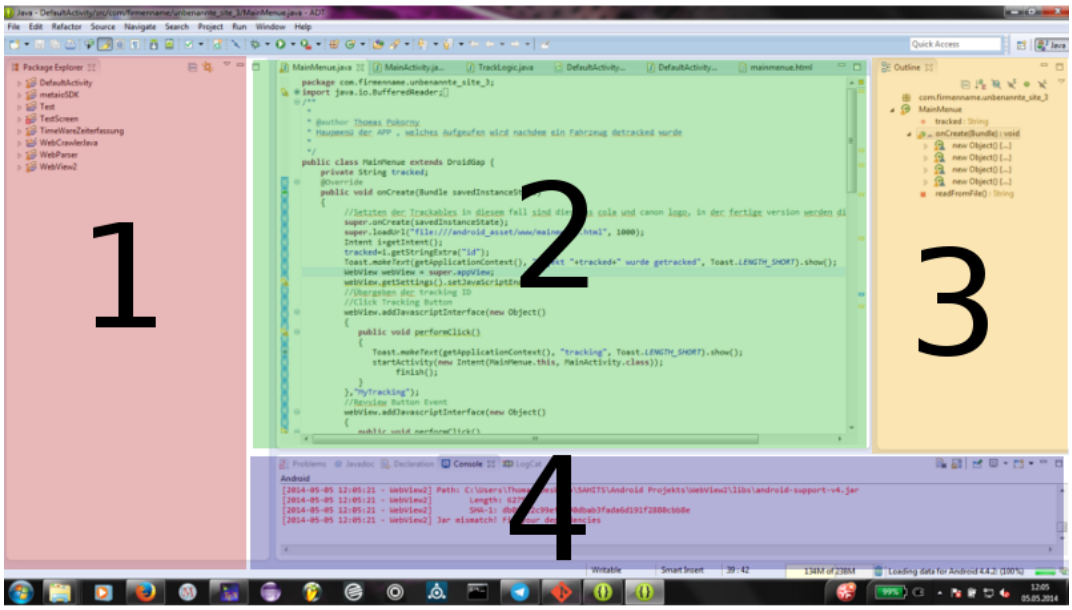

Figure 3.1: Eclipse Workbench

3.2 JetBrains WebStorm

The applications 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. [34]

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." [34]

"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." [34]

"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." [34]

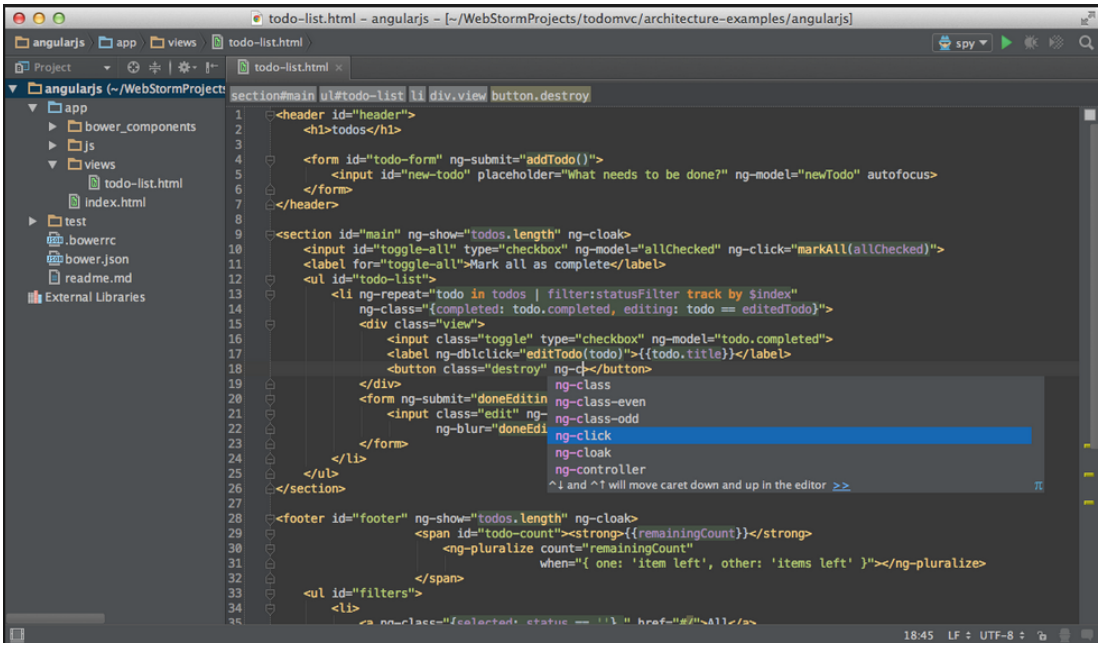

Figure 3.2: WebStorm Interface

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.* [34]

- *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.* [34]
- *JavaScript debugger for Chrome and Firefox, with breakpoints, stepping, frames view and watchers. Full-featured debugging of TypeScript, CoffeeScript and Dart with sourcemaps.* [34]
- *File Watchers for automatic compilation/transpilation of higher-level languages like TypeScript, CoffeeScript, LESS, Sass, and Stylus.* [34]
- *A debugger for Node.js applications with the latest features of V8 Debugger Protocol.* [34]
- *Intelligent code inspections, one-click quick-fix suggestions, JSHint, JSLint, and Google Closure Linter.* [34]
- *JavaScript unit testing with integrated JSTestDriver or Karma test runner with code coverage.* [34]

3.3 Visual Studio

Visual Studio is a development environment for creating applications for Microsoft platforms and beyond. It provides the programming languages C++, C#, HTML and JavaScript. The newest available version of Visual Studio is Visual Studio 2013. [35]

3.3.1 Visual Studio 2013

Visual Studio 2013 provides four commercial versions: Ultimate, Premium,Pro and Test Pro. The difference between these versions can be found on the official Visual Studio page : http://www.visualstudio.com/products/compare-visual-studio-products-vs

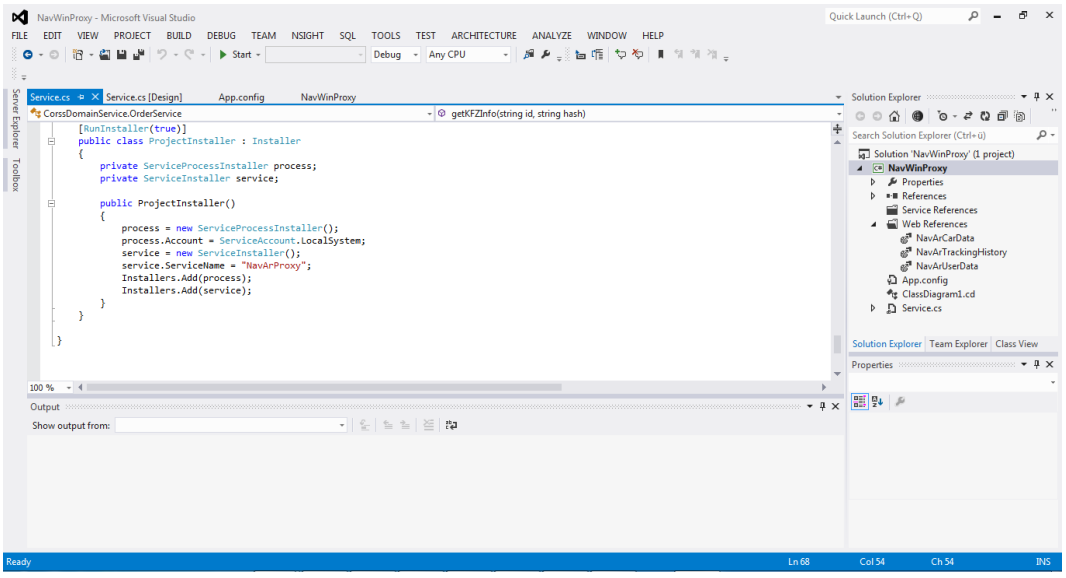

Figure 3.3: Visualstudio development environment [8]

Chapter 4

Implementation in 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 a powerful, user-friendly mobile application. The usage of web frameworks will be explained in following chapters.

4.1 Display

This chapter describes the functionality behind each display of mobile application.

4.1.1 Start Menu

The Start menu is displayed after the application was started. It provides user with several functions like help, car tracking, favourite list, about and exit. Basically it is the first thing that user sees and from there he navigates threw the whole application.

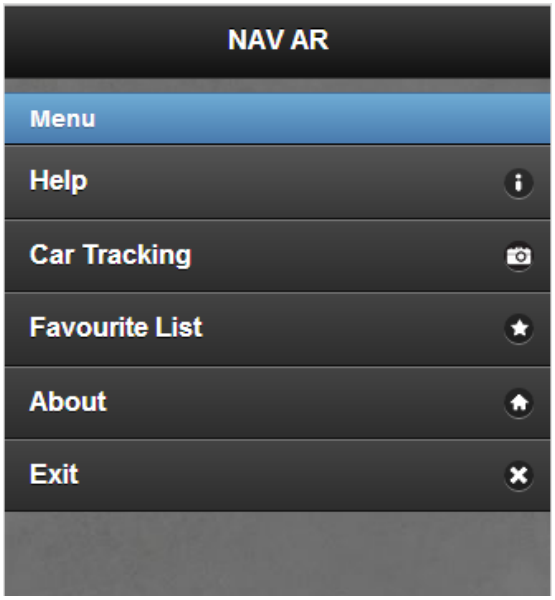

Figure 4.1: Start menu

Each one of these buttons have their logic that is implemented in **index.html**.

The listing 4.1 shows the functionality behind each button.

```
<li data-icon="info">
  <a href="help.html" rel="external">
    Help
  </a>
</li>
<li data-icon="camera">
  <a href="#" onclick="trackClick();">
    Car Tracking
  </a>
</li>
<li data-icon="star">
  <a href="myfavourite.html" rel="external">
    Favourite List
  </a>
</li>
<li data-icon="home">
  <a href="about.html" rel="external">
    About
  </a>
</li>
<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 5)"Implementation in Android Java and Metaio Tracking".

```
function trackClick() {
    MyTracking.performClick();
}

function turnOff(){
        Exit.exitClick();
}
```

Listing 4.2: JavaScript functions in start menu

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.

4.1.2 Help

The help display provides only two major options: back button and the link to a tutorial video. This tutorial was created by the project members and is an YouTube video.

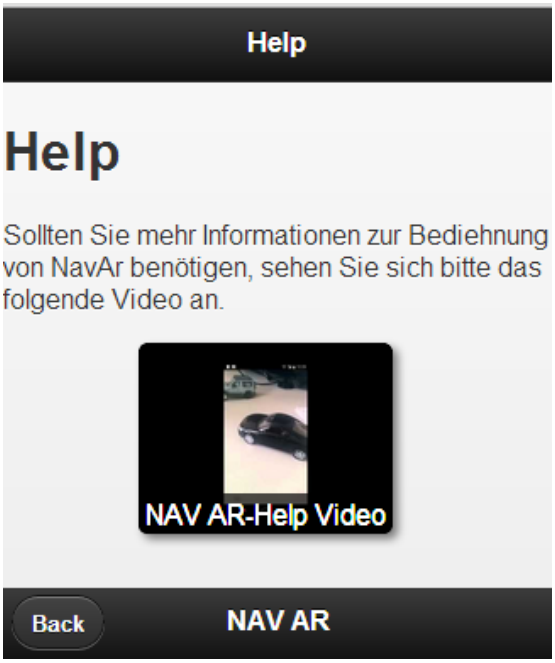

Figure 4.2: Help display

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

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

Listing 4.3: Back button

If the user touches the picture **NAV 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 <a href="https://www.youtube.com/watch?v=6U4oT5AbAsre">
2   NAV AR-Help Video
3 </a>
```

Listing 4.4: Help video

4.1.3 Main 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".

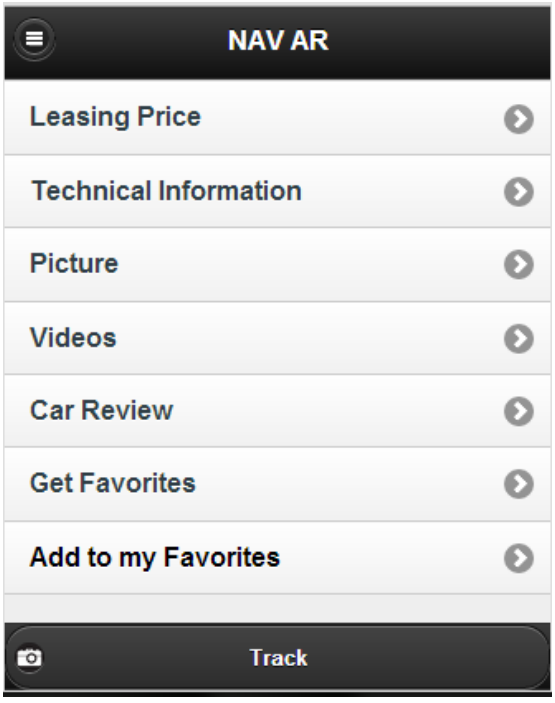

Figure 4.3: Main Menu

Listing 4.23 shows the function behind each button. Some buttons only link to another, other invoke specific functions like *LocalStorageWriteId()*.

```
<li>
  <a href="leasingprice.html" rel="external">
    Leasing Price
  </a>
</li>
<li>
  <a href="technicalinfo.html" rel="external">
    Technical Information
  </a>
</li>
<li>
  <a href="slide.html" rel="external">
    Picture
  </a>
</li>
<li>
  <a href="video.html" rel="external">
    Videos
  </a>
</li>
<li>
  <a href="#" rel="#" onclick="reviewClick();">
    Car Review
  </a>
</li>
<li>
  <a href="myfavourite.html"rel="external">
    Get Favorites
  </a>
</li>
<li>
  <a id="add_favorite"
  onclick="LocalStorageWriteId
  (sessionStorage.getItem('id'),globalcarname);"
    style="color:red" rel="external">
      Add to my Favorites
  </a>
</li>
```

Listing 4.5: Main menu source code

4.1.3.1 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.

Figure 4.4: Leasing Price

4.1.3.2 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 4.6)"Photo Gallery" and 4.8)"Dynamic Selection of Colours".

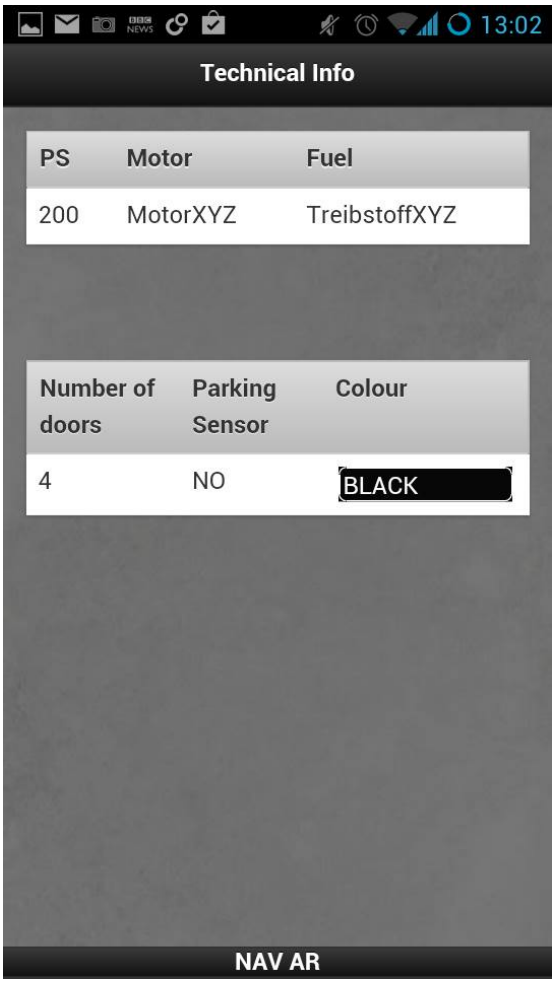

Figure 4.5: Technical Information

4.1.3.3 Pictures

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

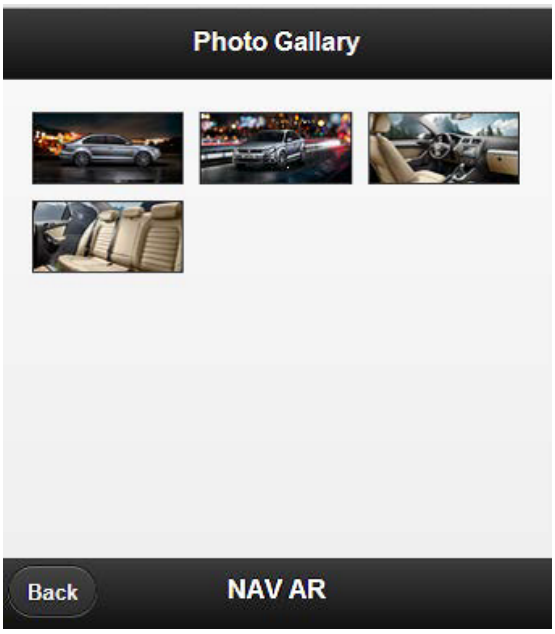

Figure 4.6: Pictures

4.1.3.4 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 4.7: Video Gallery

4.1.3.5 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.

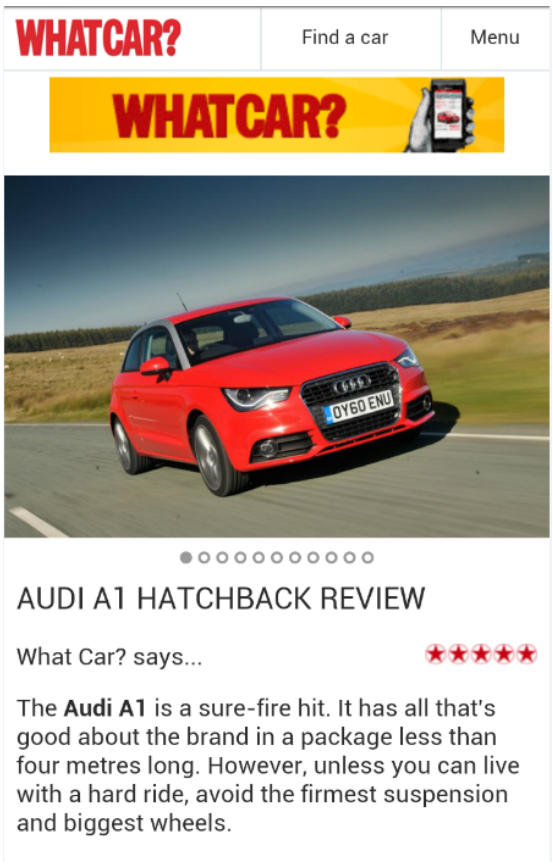

Figure 4.8: Review

4.1.3.6 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.

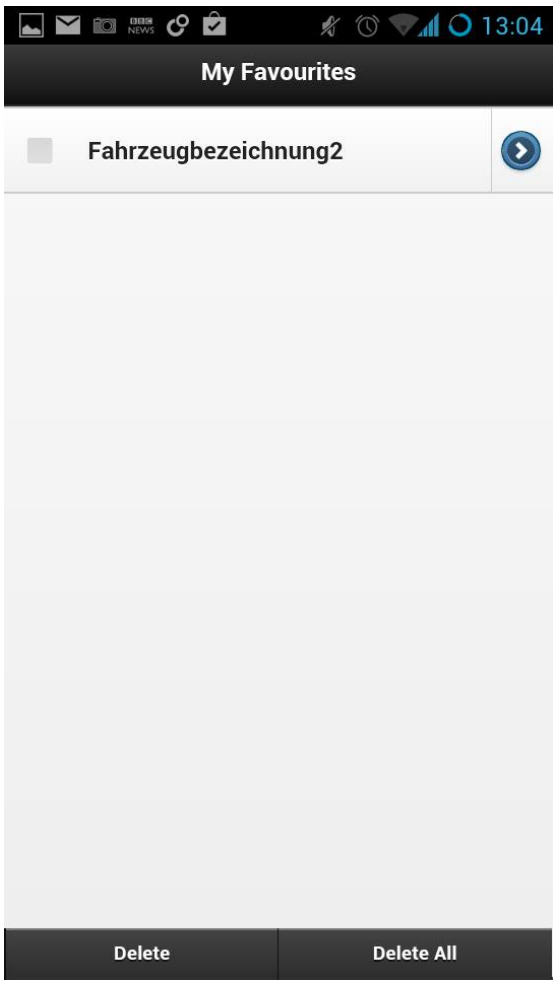

Figure 4.9: Favourite list

4.1.3.7 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>
2   <a id="add_favorite"onclick="LocalStorageWriteId
3 __(sessionStorage.getItem('id'),globalcarname);"
4   style="color:red" rel="external">
5     Add to my Favorites
6    </a>
7 </li>
```

Listing 4.6: 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 4.10: Not ready function

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

Figure 4.11: Ready function

4.1.4 My Favourites

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.

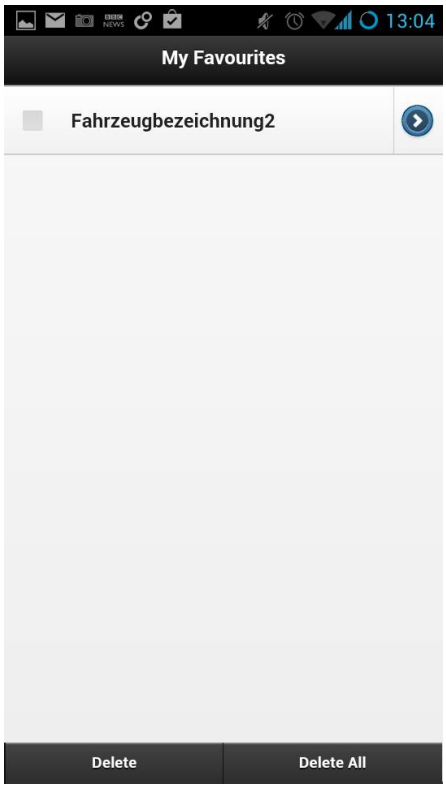

Figure 4.12: My Favourites display

4.1.4.1 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 4.7: 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 4.8: 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 = '"specific_list_item"';

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

Listing 4.9: Adding list items into the list

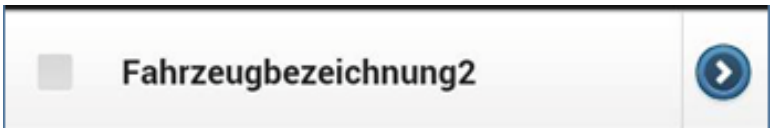

Figure 4.13: A list item

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

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

Listing 4.10: Refreshing the list

4.1.5 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 4.14: About display

4.2 Implemented Function

4.2.1 Start timer

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

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

Listing 4.11: Start timer function

4.2.2 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 4.12: End timer function

4.2.3 Save the time

This function saves start time and end time of the timer, into the NAV 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 7.0.3.4)"Communication from mobile device to the C App".

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

```
function timestampsave(time1,time2){
  var stime     = time1;
  var endtime = time2;
  var emailan = sessionStorage.getItem('email');
  var fid = sessionStorage.getItem('id');

  $(document).ready(function () {
     $.ajax({
        type: "GET",
        url: "URL",
        async: false,
        dataType: 'JSONP',
        success: function(data){
        //do your stuff with the JSON data
          var test=data;
          console.log(test);
        }
     });
  });
}
```

Listing 4.13: Save time function

4.2.4 Set parameters

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

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

Listing 4.14: Set parameter function

4.2.5 Start car tracking

This function starts to track a car. For more explanation refer to chapter 5)'Implementation in Android Java and Metaio Tracking'.

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

Listing 4.15: Car tracking function

4.2.6 Review

This method is implemented with Java Android. More about in chapter 5)'Implementation in Android Java and Metaio Tracking'.

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

Listing 4.16: Review function

4.2.7 Turn off

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

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

Listing 4.17: Turn off funtion

4.2.8 Home

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

```
1 function home(){
2         Home.performClick();
3     }
```

Listing 4.18: Home function

4.2.9 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 NAV server. A request had to be send to receive the name. More about Connectivity in chapter 7)"Streaming".

```
function readcarname(cname){
  var test = '';
  $(document).ready(function () {
    $.ajax({
      type: "GET",
        url:"URL",
        async: false,
        dataType: 'JSONP',
        success: function(data){
          test=data.split(';');
          globalcarname=test[0];
          document.getElementById("add_favorite").
          style.color="black";
        }
    });
  });
}
```

Listing 4.19: Read car name function

4.2.10 Save email

As the name says, *saveEmail()* saves the email of a user. The information about users email was already stored in session storage through the function *setParam()*. More in chapter 5.4)"Get Email Account from an Android device".

Later this email is send to the NAV server. More about connection between app and server in chapter 7)"Streaming".

```
function saveEmail(){
  var value3 = sessionStorage.getItem('email');
  $(document).ready(function () {
     $.ajax({
```

```
5       type: "GET",
6       url: "URL",
7         async: false,
8         dataType: 'JSONP',
9         success: function(data){
10        //do your stuff with the JSON data
11          var test=data;
12          console.log(test);
13        }
14      });
15    });
16  }
```

Listing 4.20: Save email function

4.2.11 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.

```
1  if(name!=null){
2  .....
3  }else{
4    alert("Function is loading.");
5  }
```

Listing 4.21: 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.

```
if ((localStorage.getItem("favorites") === null) &&
 (localStorage.getItem("fcarname") == null)) {
    var names = [];
    localStorage["favorites"] = JSON.stringify(names);
    localStorage["fcarname"] = JSON.stringify(names);
}
```

Listing 4.22: Phase two

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

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

Listing 4.23: 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.

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

Listing 4.24: Phase four

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

```
function LocalStorageWriteId(id,name){
 if(name!=null){
   if((localStorage.getItem("favorites")===null)&&
    (localStorage.getItem("fcarname")==null)){
          var names = [];
```

```
6           localStorage["favorites"] = JSON.stringify(names);
7           localStorage["fcarname"] = JSON.stringify(names);
8     }
9     var storedIds = JSON.parse(localStorage["favorites"]);
10    var storedNames = JSON.parse(localStorage["fcarname"]);
11    if(storedIds.indexOf(id)>-1){
12     Notifier.error('Car already exists.');
13    }else{
14     storedIds.push(id);
15     storedNames.push(name);
16     localStorage["favorites"] = JSON.stringify(storedIds);
17     localStorage["fcarname"] = JSON.stringify(storedNames);
18     Notifier.success('Car has been added.');
19    }
20    }else{
21     alert("Function is loading.");
22    }
23 }
```

Listing 4.25: Save car function

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

```
18      delete storedNames[s];
19      delete storedCarNames[s];
20          lengthof++;
21    }
22  }
23  if(lengthof!=0){
24    Notifier.success('Cars deleted.');
25    storedNames.clean(undefined);
26    storedCarNames.clean(undefined);
27    localStorage["favorites"]=JSON.stringify(storedNames);
28    localStorage["fcarname"]=JSON.stringify(storedCarNames);
29  }
30   window.location.reload();
31  }
```

Listing 4.26: Delete function

4.2.13 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 4.27: Delete all function

4.2.14 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    var storedNames = JSON.parse(localStorage["favorites"]);
5
6    for(var i=0; i<storedNames.length;i++){
7     if(id==i){
8      window.sessionStorage.setItem('id', storedNames[i]);
9     }
```

```
10     }
11 }
```

Listing 4.28: Select car function

4.3 Features

Several new technologies were used to create the start menu. This chapter describes all those technologies. Some are linked to other chapters where they have already been explained.

4.3.1 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. [?]

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. [?]

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

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

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

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

Listing 4.29: setItem example (Adapted from [?])

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.

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 4.15: Methods and attributes of local storage [?]

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

Listing 4.30: array into local storage

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

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

Listing 4.31: start timer function

4.3.2 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 in chapter 6.1.0.4)Slide Panel.

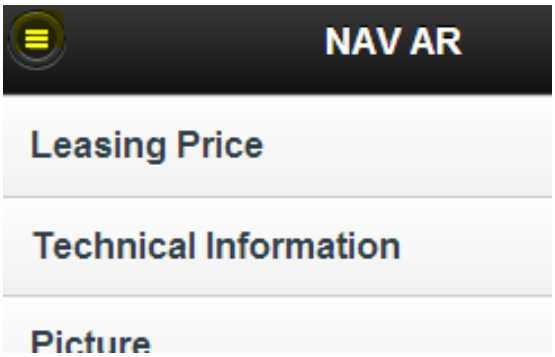

Figure 4.16: Slide panel

After opening the slide panel more options are available.

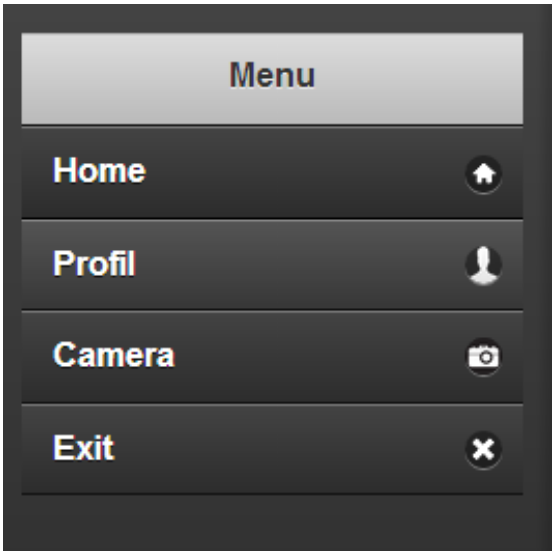

Figure 4.17: Options of slide panel

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**.

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

Listing 4.32: 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 NAV 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.

4.3.3 Photo Gallery

In this subchapter it describes the functionality of the photo gallery . The photo gallery is a feature of this project NAVAR. This feature shows the picture of the car which has been tracked by the NAVAR App. The Photo gallery is a plugin from the photo swipe webpage. The Logic of the Photo swipe is implemented in a javaScript Library from photo swipe→ klass.min.js. The functionality of the Design is defined in a css file→ photoswipe.css. In this project it combined the Library form the photo swipe with the Library from jQuery Mobile. Furthermore the Table shows a Code snippet how to use the Library for this project:

Source Code	Description

(function(window, $, PhotoSwipe){
$(document).ready(function(){
var options = {};
$(#Gallery a").photoSwipe(options);
});
}(window, window.jQuery,
window.Code.PhotoSwipe));
	This Function invokes few methods from the Photo Swipe Library.
	This part of the Code shows, that there is a Gallery tag with an id in which the pictures are saved for the swipe effect.

Figure 4.18: Photoswipe

Furthermore the pictures are saved on a server and in the Gallery we saved the URL of these pictures. The URL's are saved in an Array which has the id Gallery. For each car there is an Array with URL's of the pictures. Besides the photo swipe has the function to set a automatic Diashow.

4.3.4 Sessionstorage

Moreover the function called Sessionstorage is one of the big functionalities in this project. The Sessionstorage saves the value not persist, it means if the App is closed or has been ended the value will be persistant . In the next Session or if the App has been started , there will be 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 via cookie. This kind of functionality uses a protocol to save the Data. This protocol checks if the key and the value are 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 method is used to tell how many key/pair the SessionStorage contains. This method has the same function ,which tells the length of an Array, HTMLCollection ...

Source Code	Description
`sessionStroage.length;`	In this case it will return the value 0 ,because the Sessionstorage has not been defined with key/pair
`sessionStorage.setItem("key", "value");`	This code snippet defines a key and a value for the Sessionstorage. If someone invokes the method length for the Sessionstorage,then the size will be one.

Figure 4.19: Sessionstorage

The second Method of Sessionstorage is called *setItem(key:string,data:string)*. This method stores a specified key and the data. But if the key has been already stored and it uses the same key it will be overwritten. →Example for setItem: **sessionStorage.setItem('testkey','testvalue')**. The third function is to get Data from the a specified key ,which has been already set. This method accepts any sort of string ,which has been used as key and returns the associated string as value or null if the key has not been stored before. Example:

Source Code	Description
`sessionStorage.getItem("test");`	In this line it will return null, because this key has not been stored
`sessionStorage.setItem("test", "NAV");`	In this line it define the key test with a value.
`sessionStorage.getItem("test");`	This code snippet will return the value NAV.

Figure 4.20: Sessionstorage

The last function is how to remove the key if it has no need for the Session. Example:

Source Code	Description
`sessionStorage.getItem("test");`	This line returns NAV.
`sessionStorage.removeItem("test");`	This code line deletes the value of test.
`sessionStorage.getItem("test");`	This code snippet will return null, because there is nothing stored in this key.

Figure 4.21: Sessionstorage

4.3.5 Dynamic Selection of Colour

This product has the feature to select the colour of the car . In the Technical Information the user has the opportunity to chose the colour of the Car, which has been tracked. The following code in the Figure shows how to create a dynamic selector with JavaScript:

Source Code	Description
`var s1= $('<select class="ui-select" name="mySelect" id="mySelect" onchange="CheckSelect()" data-native-menu="false" />');`	In the first paragraph of the code snippet, it defines in JavaScript variable with a html Selector tag with attributes.
`for(j=0; j< col.length; j++) { $('<option />', {value: j, text: col[j]}).appendTo(s1); }`	In the loop the Selector becomes the items to select.
`s1.appendTo(cell[k]);` `row[c].appendChild(cell[k]);`	If the Items has been added to the selector the selector will be added to the Information Table of the Technical Information page.

Figure 4.22: Selector descritpion

The following picture shows how the Technical Info table looks like:

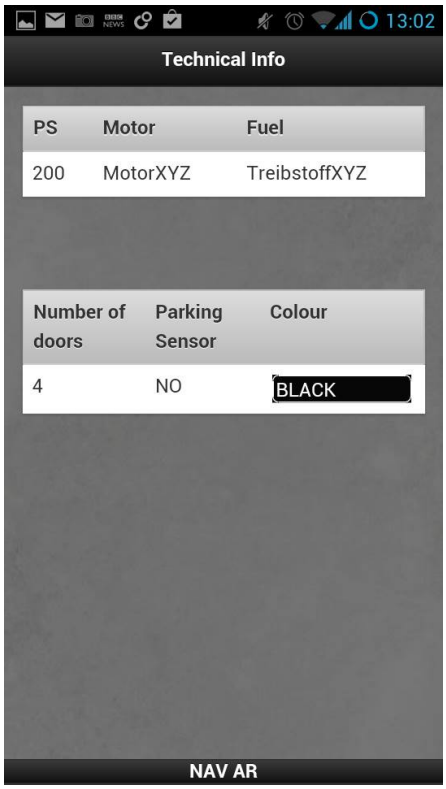

Figure 4.23: Table

The second picture shows how the dynamic List of items from the Selector looks like:

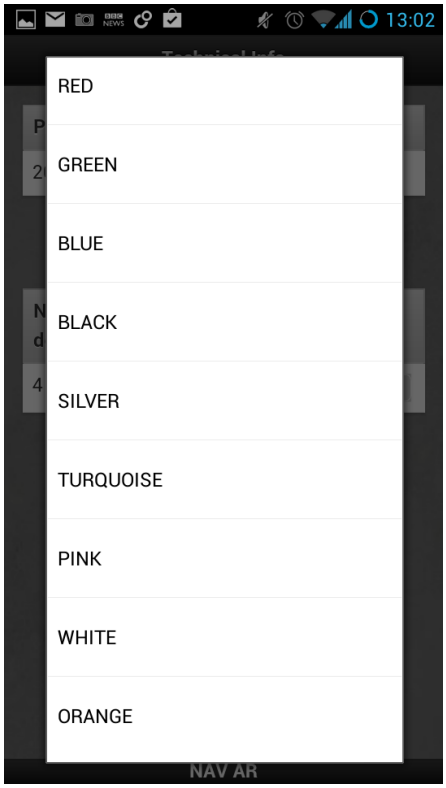

Figure 4.24: Selector

Furthermore there is a logic implemented which checks if the Colour is white or Black , that changes the colour of the font. The following code snippet shows how to write it:

Source Code	Description

function CheckSelect(){
var n =
document.getElementById('mySelect').value;
 	
if(col[n]=="WHITE"){
$('#mySelect').css("color","#070707");
$('#mySelect').css("backgroundColor",
""+hex[n]);
}else{
 	
$('#mySelect').css("color","white");
$('#mySelect').css("backgroundColor",
""+hex[n]);
}
 	
}
	The JavaScript variable has the value of the Selector.
	
	In this line it has a Condition , if the colour is White then it turn the font colour Black and if it is another colour then it set the font colour black.

Figure 4.25: Condition

Chapter 5

Implementation in Android

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 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. [36]

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. [36]

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. [36]

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, ..) [36]

4. **Layout XML:** The user interface for Activities is typcally defined via XML files (layout files). [36] 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. [37] 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. [36]

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. [36] 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. [38]. 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.

```
1 <uses-permission android:name=
2         "android.permission.CAMERA" />
3 <uses-permission android:name=
4         "android.permission.INTERNET" />
5  <uses-permission android:name=
6         "android.permission.READ_OWNER_DATA" />
7 <uses-permission android:name=
```

```
8           "android.permission.GET_ACCOUNTS" />
9   ...
10  <!-- define an activity -->
11  <activity
12          android:name=".TrackLogic"
13  </activity>
```

Listing 5.1: extracts from our AndroidManifest

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.

```
1  public class MainMenue extends DroidGap {
2          ...
3          public void onCreate(Bundle savedInstanceState)
4          {
5                  ..
6                  super.loadUrl(
7                  "file:///android_asset/file.html", 10);
8                  ...
9          }
10 }
```

Listing 5.2: extracts from our source code

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);
```

Listing 5.3: ??????

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");
```

Listing 5.4: ?????

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.

[39]
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. [40]
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.

Figure 5.1: Metaio Tool Box

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.

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:

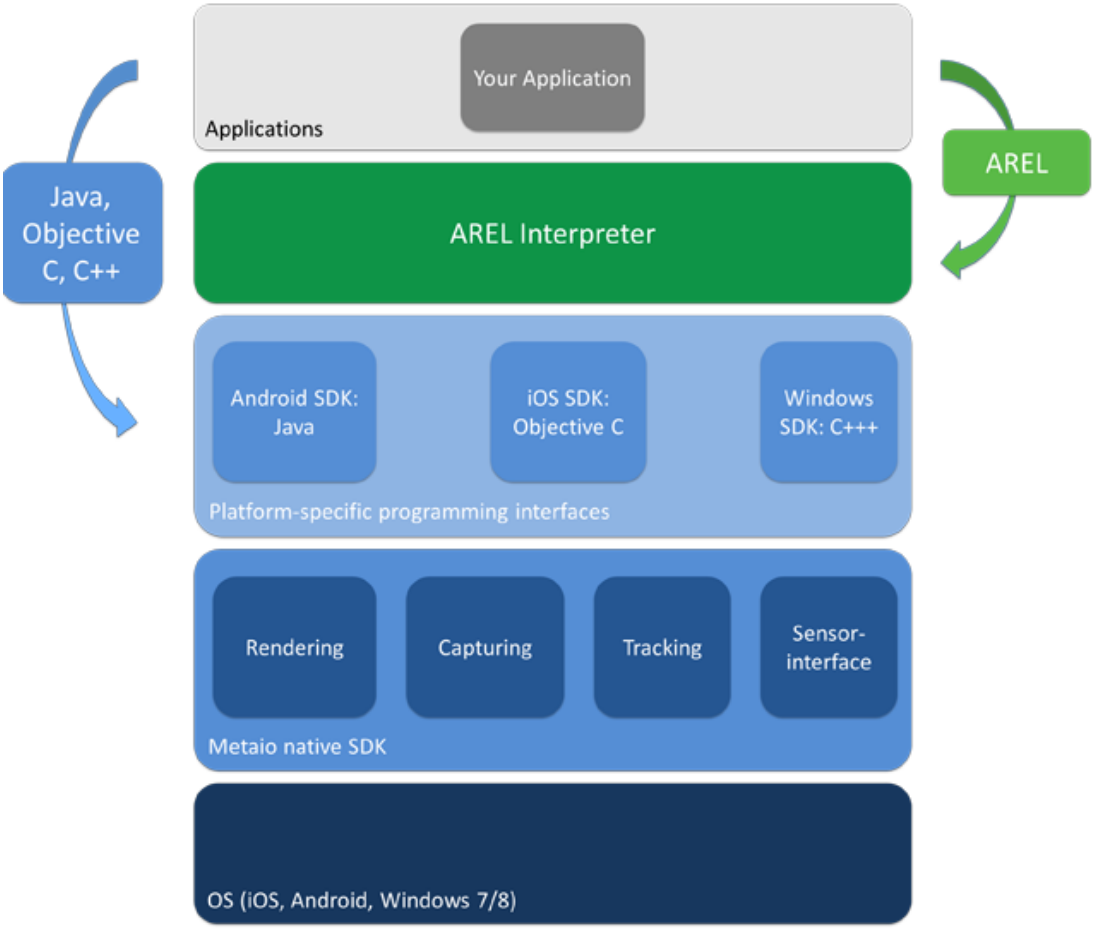

Figure 5.2: AREL

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. [40]

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.

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. ..)

```
1  <?xml version='1.0' encoding='UTF-8'?>
2  <TrackingData>
3          <Sensors>
4          <Sensor subtype='ML3D' type=
5                  'FeatureBasedSensorSource'>
6                  <SensorID>TrackingObject 1</SensorID>
7                  <Parameters>
8                  <featureorientationassignment>regular
9                  </featureorientationassignment>
10                 </Parameters>
11                 <SensorCOS>
12                 <SensorCosID>
13                 08921d8412c4de43d04eefc13c28ee3b
14                 </SensorCosID>
15                 parameters>
16                  <numextensiblefeatures>
17                          0
18                  </numextensiblefeatures>
19                  <mintriangulationangle>
20                          6
21                  </mintriangulationangle>
22                  <map>
23                  08921d8412c4de43d04eefc13c28ee3b.f3b
24                  </map>
```

```
25                      <MinMatches>15</MinMatches>
26                      <DesiredMatchesRatioExtensible>0.35
27                      </DesiredMatchesRatioExtensible>
28                      <NumExtensibleFeatures>250
29                      </NumExtensibleFeatures>
30                    </parameters>
31                    </SensorCOS>
32          </Sensors>
33 </TrackingData>
```

Listing 5.5: Tracking XML example

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*. [41]

```
1  private class AssetsExtracter extends AsyncTask
2  {
3          @Override
4          protected Boolean doInBackground(..
5          {
6                  try
7                  {
8                          // Extract all assets
9                          AssetsManager.
10                                 extractAllAssets(...);
```

```
11                  }
12                  catch (IOException e)
13                  {
14                          //Error Messages
15                  }
16
17                  return true;
18          }
19
20          Override
21          protected void onPostExecute(Boolean result)
22          {
23                  // Asset Path to the Tracking xml
24                  final String arelConfigFilePath =
25                    AssetsManager.
26                    getAssetPath("Tracking.xml");
27                  //Starting new Activity
28                  Intent intent =
29                  new Intent(getApplicationContext(),
30                   TrackLogic.class);
31                  intent.
32                  putExtra(getPackageName()+".AREL_SCENE",
33                    arelConfigFilePath);
34                  startActivity(intent);
35
36                  finish();
37            }
38  }
```

Listing 5.6: Extracting Assets

The AsyncTask gets executed like this:

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

Listing 5.7: executing AsyncTask

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:

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

Listing 5.8: executing AsyncTask

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.

```
public void onTrackingEvent(TrackingValuesVector tV)
{
 //Check through all maps
 for (int i=0; i<tV.size(); i++)
 {
  final TrackingValues v = tV.get(i);
  if (v.isTrackingState())
  {
        Intent inte=
        new Intent(getApplicationContext(),Menue.class);
```

```
11          //get id of tracked object
12          inte.putExtra("id",""+v.getCoordinateSystemID());
13          startActivity(inte);
14          //close activity
15          finish();
16    }
17
18  }
19 }
```

Listing 5.9: Tracking Event

5.4 Get Email Account from an Android device

This product has a small feature, it saves the email address from the smartphone and this email Address will be sent to the Navision Server. In this paragraph it describes how the project team implemented the feature in this Application. Description & Codesnippet: This method will be invoked by the method getmail. This method will be explained in the next row. The getAccount() method filters the Google Accounts . This happened through the Method getAccountsByType('com.google');. Furthermore if the first object of the Account Array is bigger then the value of Account it will be returned.

```
1  private static Account
2          getAccount(AccountManager accountManager) {
3 Account[] accounts =
4          accountManager.getAccountsByType("com.google");
5 Account account;
6 if (accounts.length > 0) {
7          account = accounts[0];
8 } else {
9        account = null;
10 }
11    return account;
12 }
```

Listing 5.10: Accounts

In this Method it returns the Account object as String value. This happens through the Accountmanager and this method invokes the getAccount method. With the account Object it checks if it null if not then it returns the email address as String object.

```
1  static String getEmail(Context context) {
2  AccountManager accountManager =
3  AccountManager.get(context);
4  Account account = getAccount(accountManager);
5  if (account == null) {
6          return null;
7  } else {
8      return account.name;
9          }
10 }
```

Listing 5.11: Email

The Email Address will be saved in a file , so the application have to read the email address from the file.

Chapter 6

Design Concept

6.1 Design concept with JQuery Mobile Framework

In this chapter it will be explained how the project NAVAR used the jQuery Mobile Framework,CSS3 and HTML5 to create the user Interface of the APP NAVAR. This chapter will describe how to use the components of the JQuery mobile framework. The reason why the project NAVAR uses Javascript,HTML5, CSS3 and jQuery Mobile for the user interface is, that the design can be used for other platforms. This pictures shows how the app is built:

The blue arrow shows the SDK of Metaio. The Metaio SDK will be explained in the chapter Implementation. Moreover, if Metaio GmbH creates a SDK for

another platform, so it should be necessary to change the SDK and Java Components of this project NAVAR(green arrow). NAVAR was built with modularity. There is then no need to change something in the user interface because it can be used for every mobile platform.

6.1.0.2 jQuery Mobile Page

The Basic of jQuery Mobile is to create a blank page and then to define the components such as Button or Listview, etc, which defines the interaction for the user interface. The blank page is a HTML File. It has the same basic html tag structure. The main benefit of the jQuery Mobile Framework is, there are attributes for each component or HTML tag such as:

- data-theme → to Define the colour of a component
- data-role → defines the components in HTML such as Button,Listview or Link
- data-icon → defines which icon should be used for the HTML component.

There are more data- attributes which are defined in this jQuery Mobile API: http://demos.jquerymobile.com/1.2.0/docs/api/data-attributes.html

The following Codesnippet will show the Source Code looks for a jQuery Mobile page:

```
<html>
<head>
    <title>NAV-AR</title>
    <link href="css/jquery.mobile-1.3.2.min.css"
    rel="stylesheet" type="text/css"/>
    <script src="js/jquery-1.9.1.min.js"
    type="text/javascript">
    </script>
    <script src="js/jquery.mobile-1.3.2.min.js"
    type="text/javascript">
    </script>
    <script type="text/javascript" src="js/notifier.js">
    </script>
<body>
<div data-role="header">
</div>
<div data-theme="a" data-role="footer"
data-position="fixed" data-id="footer">
    <a class="ui-btn-left" href="index.html"
    rel="external">Back</a>
</div>
</body>
</html>
```

Listing 6.1: jQuery Page

Listing 2.1 shows how to add the Library into a HTML and how to create Header, Footer from jQuery Mobile Framework. Furthermore how to use the data-role Attribute, it is often defined in HTML container Tag called div.

6.1.0.3 Mainmenu

In this segment it will be explained how to use the Listview to create a Menu with navigation form. The usage of Lists are for data display, navigation, result lists, and data entry. First of all ,to create a Listview, the html file have to include jQuery Mobile library→Code:

```
<script src="js/jquery.mobile-1.3.2.min.js"
 type="text/javascript"></script>
```

Listing 6.2: Input Library

It is possible to create a dynamical Listview, but for the dynamic function it needs JavaScript. For the menu of the APP the list view is static. This Figure shows the syntax for the Listview:

Source Code

```
<ul data-role="listview" data-theme="a" >

      <li data-role="list-divider" role="heading"
class="ui-li ui-li-divider ui-bar-b ui-first-
child">Menu</li>

<li data-icon="info" ><a  href="help.html"
rel="external">Help</a></li>

<li data-icon="camera"><a href="#"
onclick="trackClick();">Car Tracking</a></li>
</ul>
```

Figure

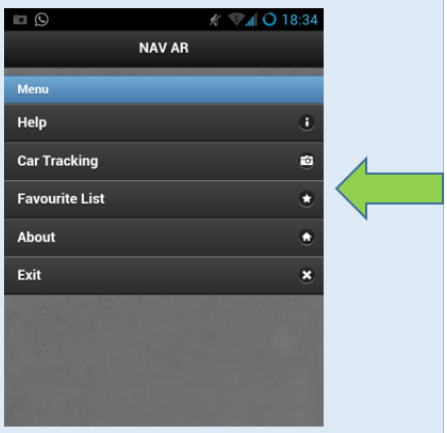

Figure 6.1: Menu Page

Jquery Mobile uses the HTMLTags to create a component of jQuery mobile and to define the data-role. Data-role is an attribute which can be used if the JQuery Mobile Library is included(Blue arrow). For this project the List items(Green Arrow) are linked to another html Files or it invokes a Function.

6.1.0.4 Slide Panel

The Slide Panel is one of the big functionalities of the jQuery Mobile Framework. With the Slide Panel it is possible to make it easy to create menus, collapsible columns, drawers, inspectors panes and more. The Panel have to be defined in a jQuery Mobile Page. Furthermore the Panel can not be placed outside of the page. The Figure shows ,how the panel looks like with the code snippet of this project NAVAR. The main point is to a create panel and to define the attribute data-role to a panel(Blue arrow.)

Afterwards the project has List Items in the panel for a menu. The user can navigate easily through the user interface with this feature. [42]

Source Code

```
<div data-role="panel" id="mypanel"

data-position="left" data-

display="reveal" data-theme="a">

<ul data-role="controlgroup"
                data-theme="d">

<div data-role="header"  data-theme="d">

<h1>Menu</h1>

 </div>

<ul data-role="listview" data-theme="a" >

<li data-icon="home" ><a href="#"
onclick="endtime();home();">Home</a></li>

                <li data-icon="profil">
            </ul>
        </ul>
    </div>
```

Figure

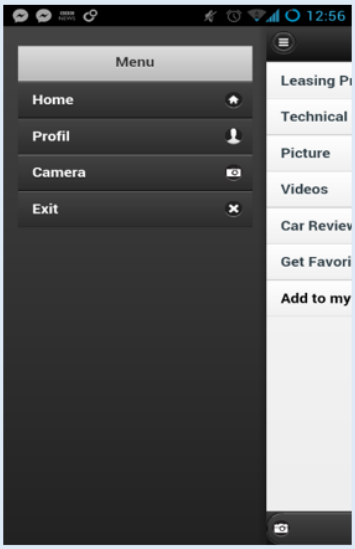

Figure 6.2: Slide Panel

The Panel has to defined in the header of jQuery Mobile page or as Button function. This code snippet shows how to do it. First all the Panel has to been defined with a id to get reference of it. Codesnippet:

```
<div data-role="header">
        <h1>NAV AR</h1>

         <a href="#mypanel" data-
          icon="bars" data-
        iconpos="notext">Menu</a>

    </div>
s
```

Listing 6.3: Panel definition in Header

6.1.0.5 jQuery Mobile Table

JQuery Mobile Framework has the feature to create a table. In this project tables are very important, because the Information from the Navision Server will be list in a table. The following code snippet will show how it works :

Source Code

```
<table data-role="table" id="info" data-

mode="columntoggle" class="ui-body-d ui-shadow table-

stroke ui-responsive"  data-column-popup-theme="a">

              <thead>

                  <tr class="ui-bar-d">

                      <th data-priority="1">PS</th>
                      <th data-priority="1">Motor</th>
                      <th data-priority="1">Fuel</th>

                   </tr>

                </thead>

</table>
```

Figure

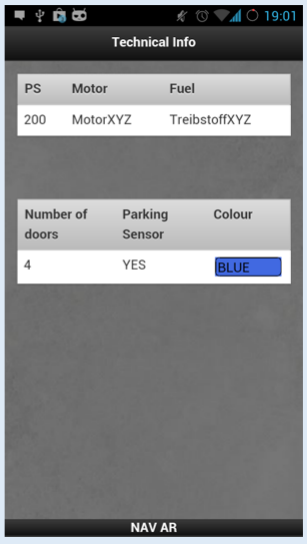

Figure 6.3: Table

shows the attribute data-role should have bee defined as 'table'(Blue Arrow).With the other attributes such as: Data–mode, data–column–popup–theme. With these attributes it is possible to define the appearance of the table.

6.2 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. [43]. 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 one can see in the figure on the next page the position and size automatically adjust to the 3 different screen sizes.

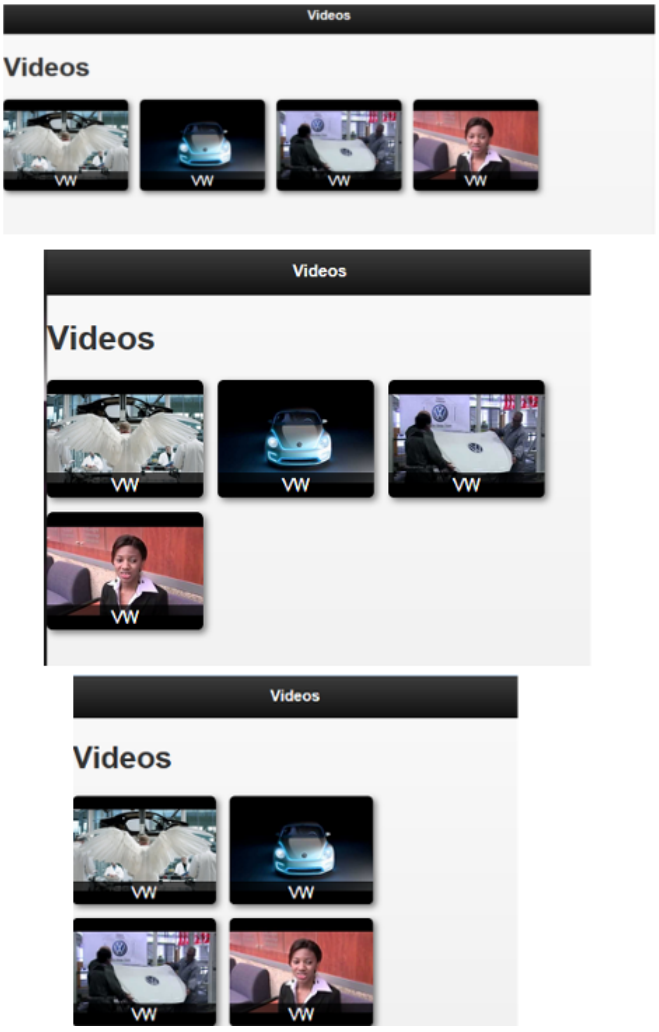

Figure 6.4: dynamically fit to device screen

6.2.1 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.

```
<ul id="Gallery" class="youtube-videogallery">
        <script>
        ...
        //append video urls
        for (var i=0; i<videos.length; i++){
                $("#Gallery").append(videos[i]);
        }
        </script>
</ul>
<!-- Use jack Mores .js for the gallery -->
<script>
    $(document).ready(function(){
        $("ul.youtube-videogallery").
        youtubeVideoGallery(
        {plugin:'colorbox',assetFolder:'../'} );
    });
</script>
```

Listing 6.4: extracts from the video gallery src

Chapter 7

Streaming

This chapter explains the network structure of the whole project and how the connections between the servers and the app are done.

7.0.2 General network

The following figure shows the detailed network set-up and application flow of the project.

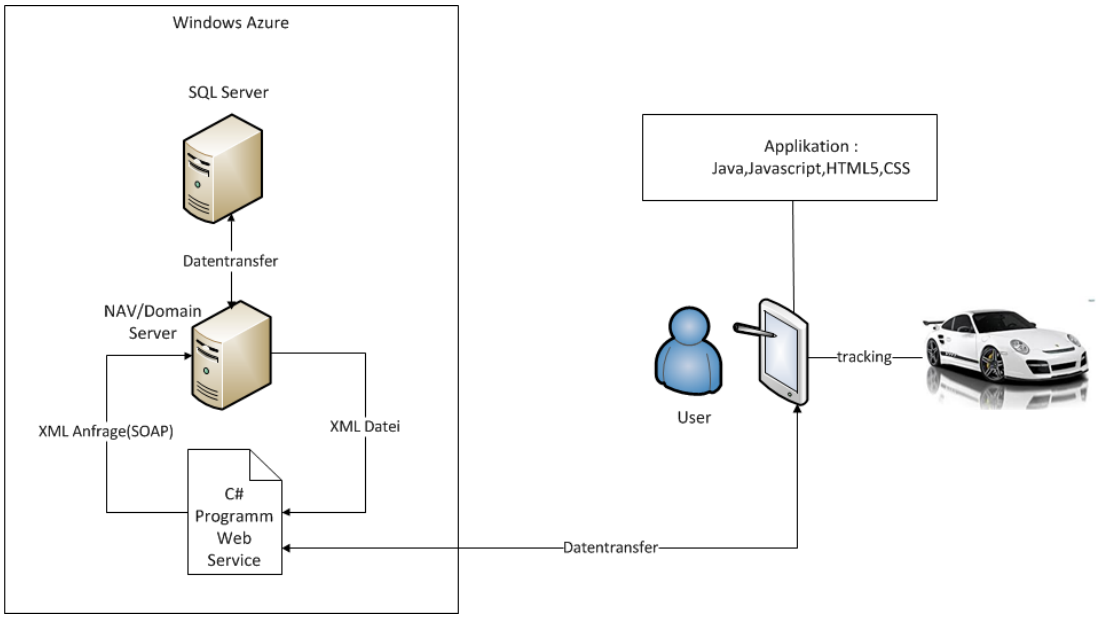

Figure 7.1: network overview

The front end consists of an application which runs on a mobile device and is operated by a user. On the back end is a Microsoft Windows Azure platform

which contains the following two server instances. The server with Microsoft SQL in the top of the graphic is for saving and processing the data. The bottom server in the graphic, the main server on which runs Microsoft Navision 2013(NAV) and a domain for communicating with the SQL server. Both of these server operation systems are windows server 2008 R2. The C# application is installed as a windows service and is used to communicate between back end and front end.

7.0.3 C# Application

The application consists of two important main parts. The first part is the communication between the Navision server and the C# program. To solve this problem it uses a Navision web service to access and receive the needed data. The Navision web service will be explained in the following sub chapter "Navision Server". When the set up of the NAV web service is done it can be simply added as a web reference in Visual Studio to access it in the code.

The application has three web references

1. NavArCarData, which is used to get all information about a tracked car.
2. NavArTrackingHistory , which is used to save a history of tracked cars.
3. NavArUserData, which is used to save data from the user.

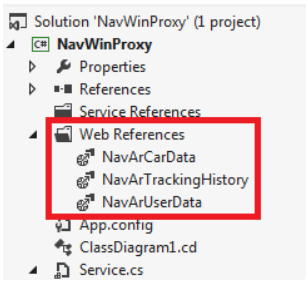

Figure 7.2: web reference

The web references can be accessed via the code in Visual Studio.

7.0.3.1 Read from the web service

```
1  //use the created web reference
2  using NavWinProxy.NavArCarData;
3  //Create a service object
4  NavArCarDataPage_Service service = new
       NavArCarDataPage_Service();
5  //set user credentials which have access rights to the
       Navision server
6  service.Credentials = System.Net.CredentialCache.
       DefaultCredentials;
7  //create a new Object for storing the data
8  NavArCarDataPage carData = new NavArCarDataPage();
9  //read the data which a specific id
10 carData = service.Read('1');
11 //check if the id exists
12 if (carData != null){
13     //access data and store data
14     string accessData=carData.Feld1;
15 }
```

Listing 7.1: Example reading webservice

In this code example the web reference can simply accessed with the using statement and a service object. After that the credentials are set with "System.Net.CredentialCache.DefaultCredentials". These are the default credentials from the user who runs the program. After that a NavArCarDataPage object is created for storing data from the web service. Then the data can easily read with service object with service.Read(String ID) method.

7.0.3.2 Write to the web service

```
1  public string insertProfileData(string androidEmail,
       string profileEmail, string firstName, string lastName
       , string hash)
2          {
3          //create the service object
4          NavArUserDataPage_Service service = new
               NavArUserDataPage_Service();
5          //set user credentials which have access rights to
               the Navision server
6          service.Credentials = System.Net.CredentialCache.
               DefaultCredentials;
7          //read the user data from a specific google mail
               address
8          NavArUserDataPage userData = service.Read(
               androidEmail);
9          //check if the email exis
10         if (userData != null)
11             {
12             //set the profile data
13             userData.Feld2 = profileEmail;
14             userData.Feld3 = firstName;
15             userData.Feld4 = lastName;
16             //update the navision database with the new
                   data
17             service.Update(ref userData);
18             }
19             else return "fail";
20                 return "success";
21             }
22             else return "fail";
23         }
```

Listing 7.2: Example writing to web service

The beginning of the example code is similar to reading from the web service.At line number 8 profile data are read with a google email adress and is checked if it exist. If it exist the attributes for are profile email,first name and last name is set and it get updated with .Update(Object to Update) method.The shown example method is simply used for inserting/updating profile data from the application in the Navsion Server

7.0.3.3 Provide a web service

The other important part of this application is to provide a web service for communication with the mobile devices. To provide the functionality the C# program uses ”System.Service.Model.web” which contains several methods for implementing a web service. In the following code example an operation contract is made to publish specific methods so that the mobile devices can access these methods.

```
1 [ServiceContract]
2     public interface IOrderService
3     {
4         [OperationContract]
5         [WebGet(UriTemplate = "getKFZInfo/{id}/{hash}",
              ResponseFormat =
6         WebMessageFormat.Json, RequestFormat =
              WebMessageFormat.Json)]
7         string getKFZInfo(string id, string hash);
8     }
```

Listing 7.3: ????

This example consist of one method ”getKFZInfo” with two parameters id and hash. It returns the car information with the given id if the hash is correct. The hash value is used to prevent access from other unwanted programs. The method could be access via the web url ”serveradress/getKFZInfo/id/hashvalue”. An invocation with the web browser would look like this: 127.0.0.1/getKFZInfo/5/ac3rf229f1fb88a8719e5f6d29443545

7.0.3.4 Communication from mobile device to the C# App

The mobile app access the web service of the C# application within JavaScript. Ajax is used for providing the communication between the user and the server. In the following code example an Ajax request to the server is shown.

```
1  function readcarname(cname){
2          $(document).ready(function () {
3              $.ajax({
4                  type: "GET",
5                  url: "http://tgm.cloudapp.net:9090/rest/
                       getKFZInfo/"
6                  +cname+"/ac73f229f1fb88a8719e5f6d295bee45?
                       callback=?"
7                  ,async: false,
8                  dataType: 'JSONP',
9                  success: function(data){
10                     globalcarname= data.split(';')[0];
11                 }
12             });
13         });
14     }
```

Listing 7.4: readcarname example

This example consist of one function readcarname(cname) which sends an Ajax request with a given parameter to the server. The first part of the url in the request represents the azure domain: http://tgm.cloudapp.net:9090 where the C# application is hosted and the second part is the method which should be called. The dataType is JSONP so that it is possible to transfer and receive data over different domains. The success method in the Ajax request will be triggered after the server sends the data back. In this example the response data from the server is stored in the variable ”globalcarname”.

7.0.4 Navision Server

For storing and receiving the important data for the mobile application a Navision Server 2013 is used.

7.0.4.1 Table structure

The table structure contains three tables for storing

1. NavArTrackingHistory
2. NavArCarData
3. NavArUserData.

The NavArTrackingHistory table is used to save the tracking history of a user to the database. It has six columns for saving the tracking history of a user.

Field No.	Field Name	Data Type	Length
1	ID	Integer	
2	Feld1	Text	250
3	Feld2	Text	250
4	Feld3	Text	250
5	Feld4	Text	250
6	Feld5	Text	250

Figure 7.3: NavArTtrackingHistory table

The following figure shows an example entry. This entry consist of a unique ID, email address, ID of the tracked car, start time, end time and duration.

ID	Feld1	Feld2	Feld3	Feld4	Feld5
30000	dummi.test@gmail.com	2	2/30/2015 7:04:07 AM	2/30/2015 7:04:42 PM	00:00:35

Figure 7.4: Example table entry

The NavArCarData table stores all necessary information about the available cars which can be tracked.

Field No.	Field Name	Data Type	Length
1	ID	Text	30
2	Feld1	Text	250
3	Feld2	Decimal	
4	Feld3	Decimal	
5	Feld4	Decimal	
6	Feld5	Integer	
7	Feld6	Text	250
8	Feld7	Text	250
9	Feld8	Integer	
10	Feld9	Boolean	
11	Feld10	Text	250

Figure 7.5: NavArCarData table

The following figure of a entry consists of a unique car ID, car name, car price, leasing car price, monthly leasing price, horse power(hp), engine , motor fuel, number of doors, parking sensors and color.

ID	Feld1	Feld2	Feld3	Feld4	F...	Feld6	Feld7	Feld8	Feld9	Feld10
0	Fahrzeug...	75,000.00	100,000.00	8,333.33	200	MotorXYZ	TreibstoffXYZ	4	☑	RED.#ff0000

Figure 7.6: Example table entry

The NavArUserData table has four columns which are used for saving the google email address,a custom email address, a name and a surname.

Field No.	Field Name	Data Type	Length
1	Feld1	Text	250
2	Feld2	Text	250
3	Feld3	Text	250
4	Feld4	Text	250

Figure 7.7: NavArUserData table

Feld1	Feld2	Feld3	Feld4
test.test@gmail.com	test2.test@gmail.com	Michael	Miller

Figure 7.8: Example table entry

To provide access to these three tables a page for each table was created with the same structure as the tables.

7.0.5 Web service

The communication between the C# application and the Navision server was achieved by a web service. The web service functionality is built in the Navision server and can be used to publish pages.A page was published for each of the mentioned tables of the previous chapter.

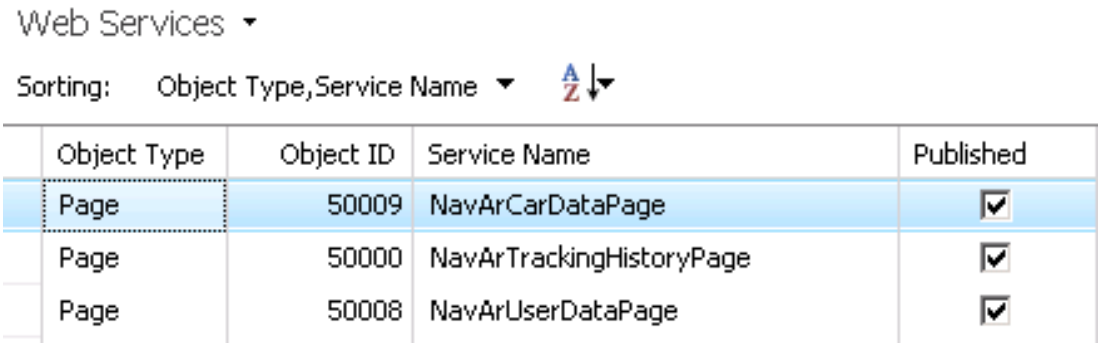

Figure 7.9: published pages

These pages can be accessed via the web service from other programs. The web service also provides methods for creating, reading, deleting and updating data. An example for reading from and writing to a web service can be found in the chapter Streaming C# Application.

Chapter 8

Future enhancements and possibilities

The developed application can provide several possibilites for the future.

8.0.6 Support for other platforms

The Metaio SDK is available for several platforms.
It currently supports:

- Android
- iOS
- Windows(PC)
- Unity

[?]

As described in chapter 6 ”Design Concept” the application was implemented to be platform independent as possible.To run the application on another platform it is only necessary to swap the dependent Java code which access the Metaio SDK.For example it is possible to change the dependent code with C code to provide an app for the platform iOS.

8.0.7 Support for Google Glass

The Metaio SDK provides support for Google Glass if the operation system is Android with the version 4.4.2 . [?]
If the mobile app will be adapted to Google Glass in the future the user controls need to be updated. It is necessary because the current application uses the touchscreen of the mobile phone for user input and Google Glass uses speech recognition. Speech recognition is a technology to translate spoken words into text. More information about Google Glass can be found in chapter 1 "Google Glasses".

8.0.8 Real estate

The application currently is used to track cars.However Metaio also provides the functionality to track objects with the size of a building. Another enhancement for the future could be to track buildings instead of a car.In this scenario the application is for the real estate market.So every person who walks by a property can simply put there mobile phone out and track the building to get information. Such as the price, number of rooms and so on.

Chapter 9

Resume

The project NAVAR was a great success. Not only the implementation of the project but also the improvement of our knowledge and our skills to work as a team.

Since the start of the project we were driven to do the best we can to make a outstanding project for ourselves, our client 4relation Consulting GmbH. and our school Technische Gewerbe Museum(TGM).We improved our skills in five programming languages, which are HTML, CSS, Java , Javascript and C#. The team acquired know how in new technologies like Windows Azure, Microsoft Dynamics Navsion and augmented reality. We learned to coordinate with our project team and the client. Every team member had his own responsibilities, task and deadline when something should be done or implemented. This increased our self-reliance and time management.

Our project had also setbacks but we never lost track and continued to work eagerly. With our mental focus and the goal to succeed we were able to hit every milestone on target without delays.With the cooperation with our client we were introduced to an unknown world, the business world.

This unique experiences and challenges could never be achieved by typical school lessons. So we are proud and thankful that we had the opportunity to do such an amazing project. As we now look back at the start of the project we are amazed what a small group of students can create in such a short period of time.

Bibliography

[1] Arpad. [Online]. Available: http://www.emilyedu.info/2014/03/

[2] C. Janssen. Glassesbild und text. [Online]. Available: http://www.digitaltrends.com/mobile/entitled-acting-tech-nerd-kicks-google-glass-ban-controversy/http://www.techopedia.com/definition/28524/google-glass

[3] [Online]. Available: http://www.glassappsource.com/google-glass-features/secret-features-google-glass.html

[4] [Online]. Available: https://blogs.lt.vt.edu/jdschmitt2/augmented-reality-what-will-the-future-hold/

[5] 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

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

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

[8] w3schools. (2014) Ajax introduction. online. W3Schools. [Online]. Available: http://www.w3schools.com/ajax/ajax_intro.asp

[9] b2bsales@quinstreet.com. Augmented reality definition. [Online]. Available: http://www.webopedia.com/TERM/A/Augmented_Reality.html

[10] R. Sood, Pro Android Augmented Reality, S. Anglin, Ed. Paul Manning, 2012.

[11] Google. (2012) Project tango. [Online]. Available: https://www.google.com/atap/projecttango/

[12] T. T. O. INDIA. [Online]. Available: http://timesofindia.indiatimes.com/tech/slideshow/googleglass/Translate/itslideshow/18609226.cms

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

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

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

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

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

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

[19] 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/

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

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

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

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

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

[25] About.com. [Online]. Available: http://javascript.about.com/od/reference/p/javascript.htm

[26] W3Schools. [Online]. Available: http://www.w3schools.com/html/html5_intro.asp

[27] W3Schools. [Online]. Available: www.w3schools.com/css/css3_intro.asp

[28] w3. (2013, 10) Extensible markup language (xml). online. W3C. [Online]. Available: http://www.w3.org/XML/

[29] w3schools. (2014) The xmlhttprequest object. online. W3Schools. [Online]. Available: http://www.w3schools.com/xml/xml_http.asp

[30] Microsoft. (2014) What is azure? [Online]. Available: 30.04.2014

[31] system center. (2014) Azure overview. [Online]. Available: http://www.system-center.fr/wp-content/uploads/2012/10/how_it_works_slide_3-Azure.jpg

[32] Microsoft. (2014) Sql server. [Online]. Available: https://www.microsoft.com/sql

[33] Microsoft. (2014) Microsoft dynamics nav. [Online]. Available: http://www.microsoft.com/en-us/dynamics/erp-nav-overview.aspx

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

[35] M. us/products/visual-studio-online-overview vs.aspx. (2014) Any app, any team. [Online]. Available: http://www.visualstudio.com/en-us/dn469161

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

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

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

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

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

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

[42] jQueryMobile. [Online]. Available: http://demos.jquerymobile.com/1.3.0-beta.1/docs/panels/

[43] J. Moore.