

Pflichtenheft PSE - VINJAB: VINJAB Is Not Just A Boardcomputer

Vladimir Bykovski Jonas Haas David Grajzel Nicolas Schreiber
Valentin Springsklee Yimeng Zhu

30. Januar 2016



Inhaltsverzeichnis

1 Einleitung	1
2 Zielbestimmung	2
2.1 Musskriterien	2
2.2 Wunschkriterien	2
2.3 Abgrenzungskriterien	3
3 Produkteinsatz	4
3.1 Anwendungsbereiche	4
3.2 Zielgruppen	4
3.3 Betriebsbedingungen	4
4 Produktumgebung	5
4.1 Software	5
4.2 Hardware	6
4.3 Systemarchitektur	7
5 Produktfunktionen	8
5.1 Endgerät mit Server verbinden	8
5.2 GUI-Instrumente auswählen	8
5.3 Datenhaltung	9
5.4 Einparkhilfe	11
5.5 Karte und POI	12
6 Produktdaten	14
6.1 Physische Sensordaten	14
6.2 Einstellungen und errechnete Daten	15
7 Produktleistungen	16
8 Graphische Benutzeroberfläche	17
8.1 Dashboard	17
8.2 Grid	18
8.3 Scrolling	19
8.4 Größen	20
8.5 Dashboard mit Statistik	21
8.6 Tabellen	22
8.7 Einstellungen	23

8.8 Karte	24
8.9 Rückfahrkamera	25
9 Globale Testfälle	26
10 Testszenarien	29
10.1 Allgemeiner Funktionstest	29
10.2 Remote Test	30
11 Qualitätsbestimmung	31
12 Glossar	32
13 Anhang	33
13.1 Quellen und Credit	33
13.1.1 Quellen	33
13.1.2 Credit	33

1 Einleitung

Zur Verbesserung des Fahrens besitzen moderne PKWs eine Vielzahl an Sensoren, die praktisch alle für das Führen eines Fahrzeugs relevanten Informationen liefern. Boardcomputer, die mittlerweile serienmäßig in allen modernen Autos verbaut sind, sind für die Darstellung dieser Daten mittlerweile unverzichtbar. Die Verarbeitung dieser Daten ist jedoch zumeist nur rudimentär, die Darstellung nur wenig personalisierbar. Eine weitläufige Speicherung der Daten und deren erneute Darstellung ist zumeist nicht vorgesehen. VINJAB bietet eine Client-Server-basierten Boardcomputerfunktion, die auf beliebigen, über Netzwerk verbundenen Computern wie Laptops, Tablets und Smartphones angezeigt werden kann. Insbesondere bietet VINJAB die Möglichkeit, den Boardcomputer auch aus großer räumlicher Distanz über das Internet anzuzeigen, sowie Fahrtdaten zu analysieren, auszuwerten und in Statistiken anzuzeigen. Als Schnittstelle zum Auto wird der mittlerweile in fast jedem PKW vorhandene OBD2-Anschluss verwendet. VINJAB ist webbasiert, leichtgewichtig, und einfach zu erweitern. Außerdem wird eine API angeboten, die den Zugriff auf Funktionen wie das Verwalten, Speichern und Verteilen von Daten ermöglicht.

2 Zielbestimmung

2.1 Musskriterien

/MK010/ Datenquelle

Die Software soll mit Sensordaten aus einem PKW arbeiten.

/MK020/ Daten verwalten

Die Daten sollen in einer zentralen Datenbank verwaltet werden.

/MK030/ Daten darstellen

Die Daten sollen im Internetbrowser auf einem Endgerät visuell dargestellt werden.

/MK040/ Informationen auswählen

Der Nutzer soll wählen können, welche Informationen dargestellt werden.

/MK050/ Aggregierte Funktionen

Neben den Daten der Sensoren sollen virtuelle Sensoren realisiert sowie aggregierte Funktionen berechnet und dargestellt werden.

/MK060/ Karte

Der Nutzer kann sich eine Karte der Umgebung anzeigen lassen, auf der gegebenenfalls Points of Interest angezeigt werden.

/MK070/ Einparkhilfe

Der Nutzer kann eine Einparkhilfe anzeigen lassen, welche Hindernisse visualisiert und die aktuelle Fahrspur anzeigt.

2.2 Wunschkriterien

/WK010/ Lokalisierung

Die Software speichert GPS-Daten vom Endgerät.

/WK020/ Sensordaten vom Endgerät

Die Software verwaltet Sensordaten von den verbundenen Endgeräten, wie zum Beispiel den Beschleunigungssensor.

/WK030/ Streckenbezug

Daten und aggregierte Funktionen sind Streckenbezogen verfügbar.

2.3 Abgrenzungskriterien

/AK010/ Weitere Sensoren

Für andere Sensoren werden keine Treiber bereitgestellt.

/AK020/ Unidirektional

Die Software liest und interpretiert die Daten nur und sendet keine Daten an Quellen zurück.

/AK030/ Bildauswertung

Kamerabilder werden von der Software nicht interpretiert.

/AK040/ Android

Die Visualisierung der Daten erfolgt nur im Browser, wir stellen keine gesonderte App zur Verfügung.

/AK050/ Flotte

Der Zugriff auf verschiedene Fahrzeuge wird nicht verwaltet.

/AK060/ Diagnose

Die Datenspeicherung ersetzt nicht den Fehlerspeicher des Steuergeräts im Auto.

/AK070/ Remote Access

Für entferntes Monitoring wird keine maximale Latenz garantiert.

/AK080/ Navigation

Die Anzeige der aktuellen Position und der umgebenden POI erfolgt über Drittsoftware und kann nicht als Navigationssystem benutzt werden.

3 Produkteinsatz

Das Produkt dient zur Speicherung und Distribution von Daten mehrerer verteilter Sensoren eines Autos. Dies beinhaltet Informationen wie die Geschwindigkeit, Motor-temperatur, Raddrehzahl und ähnliches. Damit bietet es die Möglichkeit, diese Daten auf einem Smartphone, Laptop oder anderem Computer mit Bildschirm anzuzeigen.

3.1 Anwendungsbereiche

- Dieses Produkt dient zum Speichern und Anzeigen von in einem PKW verfügbaren Sensordaten.

3.2 Zielgruppen

- Private Autofahrer mit oder ohne tieferen technischen Kenntnissen von PKWs.

3.3 Betriebsbedingungen

- Der PKW muss über einen OBD2-Anschluss verfügen.
- Das Produkt muss Strom aus dem Auto beziehen können.
- Für Remote-Zugriff muss eine Netzwerkverbindung vorhanden sein.
- Das System, auf dem der Server läuft, muss über korrekte Uhrzeit und Datum verfügen.

4 Produktumgebung

Das Rahmenwerk und die Module werden in TypeScript¹ programmiert. Der Server wird in node.js² laufen. Als Datenbanksystem wird LevelUp³ verwendet.

- Eine Client-Server-Architektur nach dem Thin-Client-Konzept
- Auf dem Server läuft der Teil der Software, der die Sensordaten empfängt, einpflegt und über WebRTC an die Endgeräte versendet.
- Auf der Clientseite laufen verschiedene Applikationen neben VINJAB. Diese Anwendungen sollen die Software aber nicht beeinflussen und auch nicht von ihr beeinflusst werden.

Testumgebung

- Testfahrzeug: Opel Astra H 1.6 Baujahr 2005
- Testfahrzeug: ITK-Engineering Modellauto
Ein Modellauto der Firma ITK-Engineering AG, angetrieben über einen Elektromotor. Alle Daten, die hier verfügbar sind, liegen über CAN an.
- Raspberry Pi 2 Model B
- ASUS BT400 Bluetooth USB Adapter
- UMTS-Stick
- OBD2-Bluetooth-Adapter ELM327

4.1 Software

- **Serverseite:**
Ein Debian-Linux-basierter Server, auf dem eine LevelDB-Datenbank läuft.
- **Clientseite:**
Web-Browser:
 - Google Chrome Version 46

¹<http://www.typescriptlang.org>

²<https://nodejs.org/>

³<https://github.com/Level/levelup>

- Safari Version 9
- Firefox Version 42
- Google Chrome Android Version 14.10

4.2 Hardware

- **Serverseite:**

Kleincomputer mit:

- Broadcom BCM2836 Arm7 Quad Core Processor (900MHz)
- 1GB RAM
- 4 x USB 2 ports
- Bluetooth-Funktionalität

Can-Modul

- **Clientseite:**

Standardrechner (min. 1 GHz und 2 GB RAM)

Android Smartphone (min. 1 GHz und 1 GB RAM)

4.3 Systemarchitektur

Das System ist modular aufgebaut. Es gibt pro System genau einen Server. Endgeräte können eine Netzwerkverbindung mit dem Server herstellen. Dabei kann die Verbindung entweder direkt via Wi-Fi im Auto oder entfernt (/FA150/) über das Internet hergestellt werden. (/FA010/, /FA020/) Das soll in der Funktionalität keinen Unterschied machen. Diese Verbindung beruht auf WebRTC⁴ und soll den Datenkanal dieses Protokolls verwenden, um Sensordaten zu übertragen. Des Weiteren ist eine Rückfahrkamera per USB mit dem Server verbunden. Die Kamerabilder werden über den Media Kanal (WebRTC) auf das Endgerät übertragen. Das mit dem Lenkrad markierte Endgerät stellt das Gerät des Fahrers dar. Personenbezogene Daten werden nach Fahrer gespeichert. (/FA030/, /FA040/)

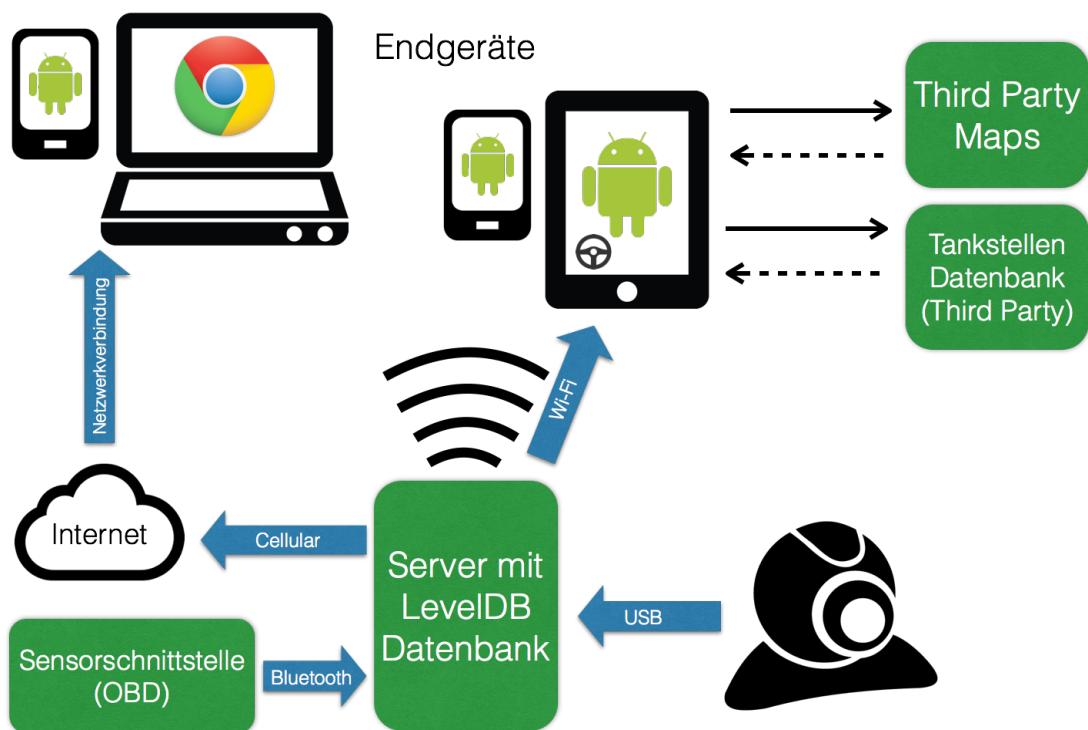


Abbildung 4.1: Blockbild der Systemarchitektur

⁴<http://www.webrtc.org>

5 Produktfunktionen

5.1 Endgerät mit Server verbinden

/FA010/ Erstmals anmelden

Ziel: Der Benutzer soll sich im System anmelden können, wobei sich nur maximal eine Person als Fahrer anmelden kann.

Vorbedingung: Das Endgerät war noch nicht mit dem Server verbunden.

Nachbedingung: Es wird ein Account für den Nutzer dieses Endgeräts angelegt.

Auslösendes Ereignis: Es wird eine Netzwerkverbindung zwischen Server und Endgerät aufgebaut.

/FA020/ Erneut anmelden

Vorbedingung: Das Endgerät war bereits mit dem Server verbunden und ist zum aktuellen Zeitpunkt getrennt.

Nachbedingung: Das Gerät ist angemeldet und der entsprechende Account ist ausgewählt.

Auslösendes Ereignis: Es wird eine Netzwerkverbindung zwischen Server und Endgerät aufgebaut.

/FA030/ Nutzer als Fahrer identifizieren

Ziel: Genau ein Benutzer wird vom System als Fahrer erkannt.

Vorbedingung: Kein Endgerät ist mit dem Server verbunden.

Nachbedingung: Der Benutzer dieses Geräts wird als Fahrer identifiziert.

Auslösendes Ereignis: Der Benutzer meldet sich als Fahrer an.

/FA040/ Fahrer unter Nutzern identifizieren

Ziel: Höchstens ein Benutzer, der sich mit einem anderen Endgerät mit dem Server verbindet, wird vom System als Fahrer erkannt.

Vorbedingung: Es ist mindestens ein Endgerät mit dem Server verbunden.

Nachbedingung: Der Benutzer eines anderen Endgeräts wird als Fahrer identifiziert.

Auslösendes Ereignis: Der Benutzer verbindet ein weiteres Endgerät mit dem Server und meldet sich als Fahrer an.

5.2 GUI-Instrumente auswählen

Der Nutzer soll GUI-Instrumente anzeigen und ausblenden können. Zu den GUI-Instrumenten gehören sowohl Dashes mit Echtzeitinformationen als auch Statistiken. GUI-Elemente

können nur zu im Fahrzeug verfügbaren Daten eingeblendet werden. Einige GUI-Elemente werden immer angezeigt.

Vorbedingung: Der Server läuft und es sind Endgeräte mit ihm verbunden.

/FA050/ **Instrumente anzeigen**

Vorbedingung: -

Nachbedingung: Das Instrument wird auf dem Bildschirm angezeigt und zeigt die aktuellen Daten an. Falls keine aktuellen Daten vorhanden sind, steht es in neutraler Position.

Auslösendes Ereignis: Der Nutzer wählt in den Einstellungen auf dem Endgerät ein Instrument aus.

/FA060/ **Instrumente verfügbar machen**

Ziel: Es werden nur Instrumente verfügbar gemacht, zu denen Daten im Auto verfügbar sind.

Vorbedingung: Die Information, welche Daten im Fahrzeug verfügbar sind, ist vorhanden.

Nachbedingung: Nur diejenigen Instrumente, deren Daten verfügbar sind, werden angezeigt.

/FA070/ **Dashboard nicht überladen**

Ziel: Das Dashboard soll nicht überladen wirken.

Auslösendes Ereignis: Der Nutzer wählt in den Einstellungen ein zusätzliches Instrument aus, welches nicht mehr passen würde.

Nachbedingung: Das Dashboard wird nach rechts erweitert.

Vorbedingung: Es werden Dashes auf dem Dashboard angezeigt.

/FA080/ **Instrumente ausblenden**

Ziel: Der Nutzer kann bestimmte Anzeigeelemente ausblenden.

Vorbedingung: Es werden Dashes auf dem Dashboard angezeigt.

Nachbedingung: Das Instrument wird nicht mehr angezeigt.

Auslösendes Ereignis: Der Benutzer wählt im Einstellungspanel ein Dash ab.

5.3 Datenhaltung

/FA090/ **Protokoll führen**

Ziel: Es werden eingehende Daten gespeichert.

Vorbedingung: Bluetooth-Verbindung mit OBD2-Adapter ist hergestellt.

Nachbedingung: Die Daten, die von der Bluetooth-Schnittstelle kommen, befinden sich in der Datenbank.

Auslösendes Ereignis: Von der Bluetooth-Schnittstelle treffen Daten ein.

/FA100/ **Vergangene Daten aufrufen**

Ziel: Es werden Daten von vergangenen Zeitpunkten dargestellt.

Vorbedingung: Es wurde bereits eine Fahrt getätigt.

Nachbedingung: Die Daten werden auf dem Endgerät angezeigt.
Auslösendes Ereignis: Der Nutzer fordert Daten aus der Vergangenheit an.

/FA110/ **Maximale Kapazität einstellen**

Ziel: Es soll eine neue maximale Kapazität der Datenbank gesetzt werden.
Vorbedingung: keine weiteren Vorbedingungen.
Nachbedingung: Die neue gespeicherte maximale Kapazität der Datenbank ist der vom Nutzer eingegebene Wert.
Auslösendes Ereignis: Der Benutzer versucht, eine neue maximale Kapazität der Datenbank zu setzen.

/FA120/ **Daten automatisch löschen**

Ziel: Um Überfüllung der Datenbank zu vermeiden, werden Daten gelöscht.
Vorbedingung: keine weiteren Vorbedingungen.
Nachbedingung: Die Datenbank hat weniger Einträge als ihre maximale Kapazität.
Die ältesten Einträge wurden gelöscht.
Auslösendes Ereignis: Die definierte maximale Kapazität der Datenbank ist erreicht.

/FA130W/ **Smartphone Utilization**

Vorbedingung: Es existiert eine Verbindung mit einem Endgerät. Die Sensoren des Endgeräts sind aktiviert.
Nachbedingung: Die Sensordaten vom Endgerät werden auf den Server übertragen.
Die Daten werden auf dem Server gespeichert und von der Software bearbeitet.
Auslösendes Ereignis: Auf dem Endgerät sind neue Sensordaten verfügbar.

/FA140W/ **Position tracken**

Ziel: Ein Bewegungsprofil erstellen, das mit den Sensorwerten in der Datenbank abgelegt wird.
Vorbedingung: Mindestens ein Endgerät ist mit dem Server verbunden. Genau ein Nutzer ist als Fahrer identifiziert. Das Endgerät des Fahrers verfügt über Standortinformationen. Smartphone-Sensordaten können an den Server übermittelt werden.
(/FA130W/)

/FA150/ **Remote Access erzeugen**

Ziel: Der Benutzer kann die Software auf einem entfernten Endgerät benutzen.
Vorbedingungen: Das System läuft und hat eine Verbindung zum Internet.
Nachbedingung: Die Funktionen GUI-Konfiguration, Fahrinformation, Vergangene Daten, Karte und Einparkhilfe stehen auf dem Endsystem zur Verfügung.
Auslösendes Ereignis: Der Benutzer verbindet ein entferntes Endgerät über Eingabe einer URL im Browser mit dem Server.

/FA160/ **Fahrinformationen darstellen**

Ziel: Darstellung der aktuellen Sensorwerte auf dem Endgerät

Vorbedingung: Das System läuft und es werden durchgängig neue Daten zur Datenbank hinzugefügt. Es existiert eine Verbindung mit mindestens einem Endgerät. Dieses Gerät besitzt eine Konfiguration des GUI.

Nachbedingung: Das Endgerät zeigt diese Daten entsprechend der aktuellen GUI-Konfiguration an. Die Datenbank bleibt unverändert.

Auslösendes Ereignis: Das Endgerät empfängt Sensordaten vom Server.

/FA170/ Aggregierte Funktionen berechnen

Ziel: Berechnung der aggregierten Funktionen.

Vorbedingung: Fahrinformation nach /FA160/ ist vorhanden. Gewünschte aggregierte Funktionen sind ausgewählt.

Nachbedingungen: Das Endgerät zeigt die Daten im UI an.

Ablauf: Die Software berechnet die gewünschten aggregierten Funktionen. Sie schickt die Informationen an das Endgerät. Das Endgerät zeigt die berechneten Daten entsprechend der GUI-Konfiguration an.

Auslösendes Ereignis: Es kommen zur Berechnung der aggregierten Funktionen benötigte Sensorwerte am Server an.

Aggregierte Funktionen sind:

- Treibstoffverbrauch: Die Software berechnet über den Tankfüllstand und die Zeit den aktuellen Treibstoffverbrauch.
- Durchschnittsverbrauch: Die Software berechnet über den Tankfüllstand und die Zeit den Durchschnittsverbrauch über die letzten 100km
- Restkilometer: Die Software berechnet eine Schätzung der Restkilometer abhängig vom aktuellen Treibstoffverbrauch.
- Strecke: Die Software berechnet die zurückgelegte Strecke über die Geschwindigkeit und die Zeit.
- Durchschnittsgeschwindigkeit

5.4 Einparkhilfe

/FA180/ Einparken vereinfachen

Ziel: Vereinfachen des rückwärts, beziehungsweise rückwärts-seitwärts Einparkens durch Anzeige einer Rückfahrkamera und Einblenden einer Visualisierung der möglichen Fahrstrecke.

Vorbedingung: Es ist eine Rückfahrkamera vorhanden.

Nachbedingung: Das Bild der Rückfahrkamera wird angezeigt und regelmäßig aktualisiert. Die Visualisierung der Fahrbahn wird korrekt dargestellt und regelmäßig aktualisiert.

Auslösendes Ereignis: Der Benutzer fordert das System auf, die Kamera einzuschalten.

/FA190/ Ultraschallsensoren visualisieren

Ziel: Die Distanzwerte der Ultraschallsensoren visualisieren.

Vorbedingung: Es sind Ultraschallsensoren vorhanden.

Nachbedingung: Die korrekten Messwerte werden angezeigt.

Auslösendes Ereignis: Der Benutzer möchte die Messwerte anzeigen lassen.

/FA200/ **Einparkhilfe beenden**

Ziel: Die Einparkhilfe beenden.

Vorbedingung: Auf dem Bildschirm wird die Einparkhilfe angezeigt.

Nachbedingung (Erfolg): Die Einparkhilfe wird nicht mehr angezeigt.

Auslösendes Ereignis: Der Benutzer fordert das System auf, die Einparkhilfe zu beenden. Alternativ: Das Fahrzeug fährt schneller als 10 km/h vorwärts.

5.5 Karte und POI

/FA210/ **Karte anzeigen**

Ziel: Die Karte der Umgebung anzeigen

Vorbedingung: Das Endgerät des Fahrers hat eine Verbindung zum Netzwerk. Weiterhin kennt es den aktuellen Standort.

Nachbedingung: Die Karte der Umgebung wird angezeigt.

Auslösendes Ereignis: Der Benutzer fordert die Karte an.

/FA220/ **Tankstellen auflisten**

Ziel: Die Tankstellen in der Nähe finden

Vorbedingung: Das Endgerät des Fahrers hat eine Verbindung zur Netzwerk. Die Lokalisierungsinformationen stehen zur Verfügung.

Nachbedingung: Eine Liste, auf der die Tankstellen in der Nähe mit jeweiligem Preis bzw. jeweiliger Entfernung stehen, wird erzeugt und angezeigt. Auf der Karte werden die Tankstellen in der Nähe gekennzeichnet. Der angezeigte Preis ist der des ausgewählten Treibstoffs.

Auslösendes Ereignis: Der Tankfüllstand sinkt unter den festgelegten Mindeststand und ein Benutzer öffnet die Karte..

/FA230/ **Werkstätten anzeigen**

Ziel: Das OBD-System meldet ein Warnsignal. Daraufhin werden die umliegenden Werkstätten auf der Karte angezeigt.

Vorbedingung: Das Endgerät des Fahrers hat eine Verbindung zum Netzwerk. Weiterhin kennt es den aktuellen Standort.

Nachbedingung: Eine Liste der Werkstätten in der Nähe mit jeweiliger Entfernung wird erzeugt und angezeigt. Auf der Karte werden die Werkstätten in der Nähe gekennzeichnet.

Auslösendes Ereignis: Es liegt ein kritischer Sensorwert vor (außer Tankfüllstand). Ein Benutzer öffnet die Karte.

/FA240W/ **Route zu POI berechnen**

Ziel: Der Fahrer wird über die Route zur gewünschten Tankstelle oder Werkstatt informiert.

Vorbedingung: Die POI in der Nähe sind angezeigt. Das Endgerät des Fahrers hat eine Verbindung zum Netzwerk.

Nachbedingung: Die Route zum gewünschten POI wird auf der Karte angezeigt.
Auslösendes Ereignis: Der Fahrer wählt einen POI aus.

6 Produktdaten

6.1 Physische Sensordaten

Es sind zeitgestempelte und fahrergestempelte (/FA030/) Messwerte der folgenden Sensoren zu speichern, sofern diese prinzipiell im PKW vorhanden sind und von der Software abgerufen werden können:

- /PD010/ Temperatur der Motorkühlflüssigkeit
- /PD020/ Kraftstoffdruck
- /PD030/ Motordrehzahl
- /PD040/ Geschwindigkeit des Fahrzeugs
- /PD050/ Lenkraddrehung
- /PD060/ Ansauglufttemperatur
- /PD070/ Laufzeit des Motors seit letztem Start
- /PD080/ Tankfüllstand
- /PD090/ Status der Abgasrückführung
- /PD100/ Status des Kontrollsystems für Einspritzdruck
- /PD110/ Status des Kontrollsystems für Kraftstoffdruck
- /PD120/ Gasdruck des Verdampfers
- /PD130/ Katalysatortemperatur
- /PD140/ Position der Drosselklappe
- /PD150/ Position des Gaspedals
- /PD160/ Außentemperatur
- /PD170/ Motordrehmoment
- /PD180/ Abgastemperatur
- /PD190/ Abgasdruck

/PD200/ Distanzwerte der Ultraschallsensoren

/PD210/ Durchschnittsverbrauch (temporär)

/PD220/ Zurückgelegte Strecke (temporär)

/PD230/ Durchschnittsgeschwindigkeit (temporär)

6.2 Einstellungen und errechnete Daten

Es sind die folgenden Daten aus den Einstellungen sowie errechnete Daten zu speichern:

/PD240/ Endgerät-Signaturen

/PD250/ Anzeigeeinstellungen

/PD260/ Präferierter Kraftstoff (Super E5, Diesel, etc.)

/PD270/ Anzahl der Einzelfahrten

/PD280/ Präferierte maximale Datenbankkapazität

/PD290/ Bluetooth-Verbindungsdaten mit dem OBD2-Adapter (temporär)

/PD300/ Netzwerk-Verbindungsdaten aller verbundenen Geräte (temporär)

/PD310/ Aktuell anzuzeigende Sensordaten auf den Endgeräten (temporär)

/PD320/ Aktuell als Fahrer eingetragener Benutzer (temporär)

7 Produktleistungen

/PL010/ Latenz von /FA100/

Die Funktion /FA100/ soll maximal 300 ms dauern. Wenn die Daten nach dieser Zeit noch nicht angezeigt wurden soll ein Wartesymbol angezeigt werden.

/PL020/ Auswirkung von /FA120/

Die Funktion /FA120/ darf keine Auswirkungen auf die sonstige Performanz des Systems haben.

/PL030/ Latenz von /FA160/

Die Funktion /FA160/ darf höchstens 100 ms benötigen.

/PL040/ Bildrate der Rückfahrkamera

Das Kamerabild sowie die dargestellte Fahrbahn sollen im zeitlichen Abstand von maximal 100 ms aktualisiert werden.

/PL050/ Aktualisierungsrate Ultraschallsensoren

Die Aktualisierung von /FA190/ soll im zeitlichen Abstand von maximal 100 ms stattfinden.

/PL060/ Multiple Access

Die Verbindung von bis zu vier Personen gleichzeitig ist möglich.

8 Graphische Benutzeroberfläche

Auf den Endgeräten wird eine Graphische Benutzeroberfläche angezeigt. Diese soll sich je nach Endgerät nur im verfügbaren Platz entsprechend der Größe der Displays unterscheiden, nicht aber in der Funktionalität.

Der Nutzer soll GUI-Instrumente anzeigen und ausblenden können. Zu den GUI-Instrumenten gehören sowohl Dashes mit Echtzeitinformation als auch Graphen, die Aufzeichnungen von Daten in einem längeren Zeitraum visualisieren.

8.1 Dashboard

Das Dashboard besteht aus verschiedenen Anzeigeelementen, sogenannten Dashes, und oben einer Leiste, auf der verschiedene Informationen dargestellt werden. Links wird, beim Fahrer, ein Lenkrad angezeigt. Mittig ist der Einstellungsbutton, über den man in die Einstellungsansicht wechseln kann. Rechts wird die Uhrzeit angezeigt. Diese Leiste ist immer zu sehen, die Elemente können sich jedoch, je nach aktueller Anzeige, ändern.



Abbildung 8.1: Verschiedene Dash-Anzeigeelemente.

8.2 Grid

Das Dashboard ist als Gitter (Grid) aufgebaut. Die Anzahl der Zeilen und Spalten hängt von der Bildschirmgröße des Endgeräts ab. Hier wird also z.B. auf einem Smartphone ein 8x4-Gitter angezeigt während auf einem Computerbildschirm z.B. ein 16x9-Gitter angezeigt wird.

Welche Dashes im allgemeinen angezeigt werden, kann über die Einstellungsansicht konfiguriert werden. Wenn man in dieser weitere Anzeigeelemente hinzufügt, werden diese in die noch verfügbaren Rechtecke platziert. Zur Not werden auf der rechten Bildschirmseite noch weitere Rechtecke erzeugt, die durch scrolling erreicht werden können.

Durch klicken auf ein Elementes lässt sich das Element verschieben. Durch einen Doppelklick auf ein Element werden zwei Schaltflächen angezeigt. Eine um das Dash zu entfernen, das andere um die Größe des Dashes zu verändern. Die Größe bleibt jedoch proportional und im Rahmen des Gitters. Hier werden dann z.B. aus 1x1-Anzeigen 2x2-Anzeigen, oder aus 1x2-Dashes 2x4-Anzeigen. In einem 8x4-Gitter sind jedoch keine 5x5-Anzeigen möglich.

Durch das Verschieben werden andere Anzeigeelemente auch passend verschoben. Bei dem folgenden Bild z.B., wenn man das Dash in der Mitte nach links verschiebt, werden die links platzierten Anzeigen automatisch nach rechts verschoben.



Abbildung 8.2: Verschiedene Dash-Anzeigeelemente mit eingeblendeten Grid-Grenzen.

8.3 Scrolling

Wenn die Rechtecke im Anzeigefenster für die gewünschten Dashes nicht ausreichen, ist es möglich die komplette Anzeige horizontal zu scrollen. Hier werden rechts neue Rechtecke hinzugefügt, falls versucht wird ein Dash an den rechten Rand der Anzeige zu schieben. Beim scrollen bleibt das Dashboard so stehen, dass keine abgeschnittenen Rechtecke angezeigt werden.



Abbildung 8.3: Ein Dashboard während dem Scrollen.

8.4 Größen

Wie bereits weiter oben beschrieben gibt es verschiedene Anzeigeelemente, die auch in verschiedenen Größen konfiguriert werden können. Hier ein Beispiel in dem in der Mitte eine einfache Geschwindigkeitsanzeige als 4x4-Element angezeigt wird. Es gibt jedoch nicht nur quadratische Elemente, wie man links unten sehen kann. Hier gibt es z.B. eine 2x1-Anzeige des Tankfüllstands und der Wassertemperatur. Außerdem gibt es neben grafischen Anzeigen auch textuelle Dashes wie z.B. in der oberen rechten Ecke zu sehen ist.



Abbildung 8.4: Verschiedene Dash-Anzeigeelemente mit Grid-Größen

8.5 Dashboard mit Statistik

Weitere mögliche Anzeigeelemente neben Live-Anzeigen sind Graphen, auf denen Statistiken eines Wertes angezeigt werden können. In diesem Beispiel kann man z.B. die Geschwindigkeit über einen bestimmten Zeitraum, sowie den Kraftstoffverbrauch in zwei Graphen sehen. Zusätzlich sind noch weitere Dashes angezeigt.

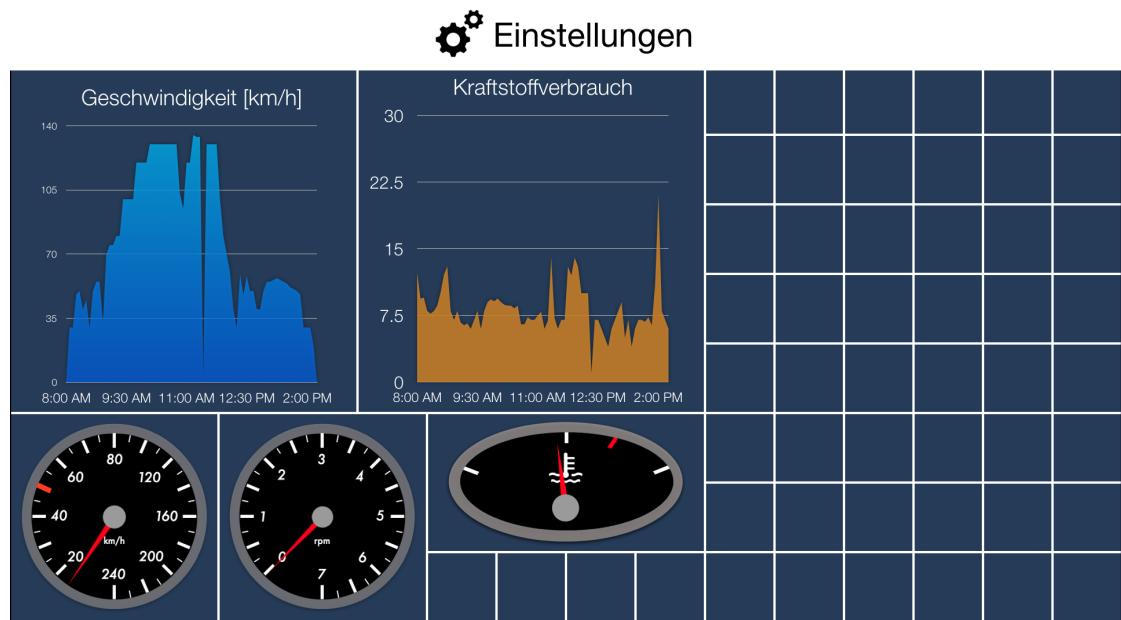


Abbildung 8.5: Statistiken mit anderen Anzeigeelementen.

8.6 Tabellen

Die Tabelle kann in verschiedenen Anzeigen genutzt werden. Die Tabellengröße, also die Anzahl der Spalten und Zeilen, ist variabel. In der Tabelle kann man scrollen und ihre Zellen können mit verschiedenen Elementen, wie z.B. Bildern oder Text, gefüllt werden. Über die Auswahl einer Zelle kann eine Funktion aufgerufen werden.

Genutzt wird eine Tabelle z.B. in der Karte.

The screenshot shows a car's infotainment system interface. At the top left are icons for a steering wheel and a square. In the center is a gear icon labeled "Einstellungen". On the right is the time "20:55". Below the gear icon is a table with the following rows:

	Kühlmitteltermperatur [°F]	Zeigt die Kühlmitteltermperatur im klassischen Rundinstrument-Design (Halbkreis)
	Öldruckanzeige	Zeigt den Öldruck im klassischen Rundinstrument-Design (Halbkreis)
	Momentanverbrauch [l/100km]	Zeigt den ermittelten Momentanverbrauch als Zahl in Liter pro 100 km
	Momentanverbrauch [mpg]	Zeigt den ermittelten Momentanverbrauch als Zahl in Miles per Gallon
	Außentemperatur °C	Zeigt die Außentemperatur als Zahl in Grad Celsius
	Außentemperatur °F	Zeigt die Außentemperatur als Zahl in Grad Fahrenheit
	Geschwindigkeit [km/h]	Zeigt die momentane Geschwindigkeit als Zahl in der Einheit Kilometer pro Stunde
	Geschwindigkeit [mph]	Zeigt die momentane Geschwindigkeit als Zahl in der Einheit Meilen pro Stunde
	Motordrehzahl [rpm]	Zeigt die momentane Motordrehzahl als Zahl in der Einheit Kilometer pro Stunde
	Geschwindigkeit über Zeit [km/h]	Zeigt die aufgezeichneten Geschwindigkeitsdaten über die Zeit als Diagramm
	Geschwindigkeit über Zeit [mph]	Zeigt die aufgezeichneten Geschwindigkeitsdaten über die Zeit als Diagramm
	Verbrauch	Zeigt die aufgezeichneten Verbrauchsdaten über die Zeit als Diaframm

Abbildung 8.6: Beispiel: Tabelle mit den verschiedenen verfügbaren Informationen.

Das Verlassen der Tabelle ist möglich über die mittig in der obigen Leiste befindliche Schaltfläche mit dem Namen der vorherigen Anzeige.

8.7 Einstellungen

Hier werden zuerst alle Einstellungskategorien in der Tabelle angezeigt, diese sind die Anzeigeelemente und Allgemeine Konfigurationen, wie der niedrige Tankfüllstand. Bei Auswahl einer Kategorie wird die Anzeige der gewählten Kategorie geöffnet. Hier werden in einer Tabelle die verschiedenen Konfigurationsmöglichkeiten dargestellt. Diese lassen sich auch wieder über einen Auswahl eines Tabellenelements beeinflussen. In dem folgenden Beispiel wird das Einstellungsmenü dargestellt.

In den gesamten Einstellungen kann man sich oben links über das Kästchen mit dem Haken als Fahrer ausweisen.

Vorgenommene Einstellungen werden erst bei Verlassen der Einstellungen aktiv.

 <input checked="" type="checkbox"/>	 Dashboard	20:55
	Dashes Auswahl der verschiedenen Anzeigeelemente auf dem Dashboard	
	Kraftstoff Auswahl der verschiedenen Anzeigeelemente auf dem Dashboard	
	Tankwarnung Bei welchem Tankfüllstand wird gewarnt	
	...	

Abbildung 8.7: Beispiel: Tabelle mit den verschiedenen verfügbaren Informationen.

8.8 Karte

Eine weitere Ansicht ist die Karte. Hier wird links eine Tabelle genutzt, um z.B. Werkstätten oder Tankstellen in der Nähe vorzuschlagen. Rechts werden diese dann in einer Karte zusammen mit dem aktuellen Standort angezeigt. Bei Auswahl eines Vorschlags wird auf der Karte die relative Position zum Auto angezeigt.

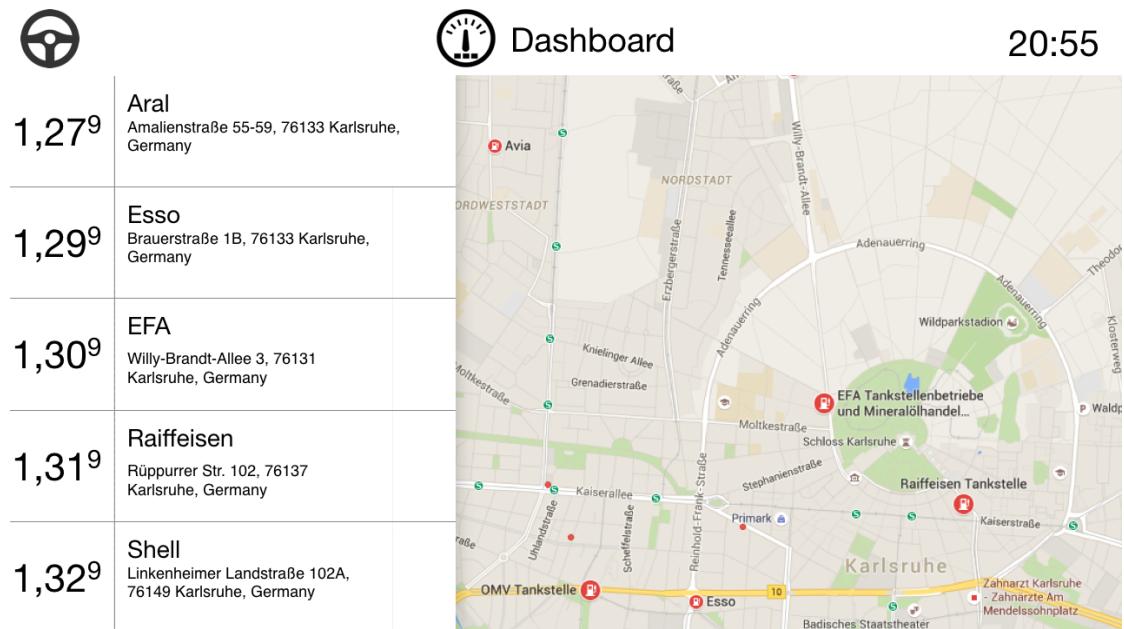


Abbildung 8.8: Eine Kartenanzeige mit einer Tabelle

8.9 Rückfahrkamera

Die Rückfahr-Ansicht zeigt dem Nutzer, falls verfügbar, die aktuelle Fahrbahn über eine Kurve auf dem Kamerabild sowie die Ultraschalldaten visualisiert an der passenden unteren Seite des Kamerabildes. Erreichbar ist diese Ansicht durch eine Schaltfläche ähnlich dem Karten-Button.



Abbildung 8.9: Kamerabild mit eingeblendeter Fahrbahn und Abstandswerten.

9 Globale Testfälle

Globale Vorbedingungen:

- Die Zündung ist an.
- Der Server läuft.

/GT010/ Erstanmeldung

Ablauf: Ein Endgerät verbindet sich zum ersten Mal mit dem Server. Es wird überprüft ob die Signatur des Endgeräts in die Datenbank hinzugefügt wurde.

/GT020/ Erneute Anmeldung

Vorbedingung: Das Endgerät hatte sich bereits einmal mit dem Server verbunden, ist aktuell jedoch getrennt.

Ablauf: Das Endgerät verbindet sich mit dem Server. Es wird überprüft ob automatisch die Daten der passenden Signatur geladen werden.

/GT030/ Fahreranmeldung

Ablauf: Ein Endgerät verbindet sich mit dem Server. Es wird als Gerät des Fahrers identifiziert. Es wird überprüft, ob die anschließend gespeicherten Daten mit der Signatur des neuen Fahrers abgelegt werden.

/GT040/ Fahrerwechsel

Ablauf: Ein Endgerät verbindet sich mit dem Server. Es wird als Gerät des Fahrers identifiziert. Es verbindet sich ein zweites Endgerät und meldet sich als Fahrer an. Es wird überprüft, ob die anschließend gespeicherten Daten mit der Signatur des neuen Fahrers und nur der Signatur des neuen Fahrers abgelegt werden.

/GT050/ Instrumente anzeigen

Vorbedingung: Ein Endgerät ist mit dem Server verbunden.

Ablauf: Alle verfügbaren Instrumente werden auf dem Endgerät ein- und ausgetauscht und kombiniert. Es wird auf korrekte Auflistung der Dashes im Bezug auf die vorhandenen Sensoren geprüft.

/GT060/ Aggregierte Funktionen

Vorbedingung: Ein Endgerät ist mit dem Server verbunden.

Ablauf: Es werden vom Server verschiedene aggregierte Funktionen berechnet. Diese werden dann mit händisch berechneten Werten verglichen.

/GT070/ Smartphone Utilization

Vorbedingung: Die Sensoren des Endgeräts sind aktiviert.

Ablauf: Es werden die Sensorwerte vom Endgerät auf dem Server gesendet. Dann wird überprüft ob der Server die Daten empfangen und gespeichert hat, und ob diese von Software bearbeitet werden.

/GT080/ **Fahrinformationen darstellen**

Ablauf: Es werden neue Sensordaten zur Datenbank hinzugefügt. Dann werden diese auf dem Endgerät angezeigt.

/GT090/ **Protokoll führen und Daten anzeigen**

Vorbedingung: Das Auto fährt.

Ablauf: Das Auto fährt und erzeugt Daten. Die Daten werden im Endgerät auf allen Instrumenten live und auf Abruf angezeigt. Es wird auf Konsistenz und ausreichend kleine Latenz geprüft. Datenausfälle werden simuliert. Die Datenbank wird auf korrekte Einträge überprüft.

/GT100/ **Maximale Kapazität prüfen**

Vorbedingung: keine weiteren Vorbedingungen erforderlich

Ablauf: Die Default-Kapazität auf dem Endgerät wird geprüft. Es wird eine neue Default-Kapazität gesetzt. Mit dem Auto werden Daten erzeugt, bis die Datenbank bis zum Grenzwert gefüllt ist. Es wird überprüft, ob das Datenlimit eingehalten wurde und ob, nur wie spezifiziert, die ältesten Daten gelöscht wurden. Die Datenbank wird auf Konsistenz und korrektes Löschen aller Daten geprüft.

/GT110/ **Remote Access**

Vorbedingung: Keine weiteren Vorbedingungen erforderlich.

Ablauf: Ein entferntes Endgerät wird mit dem Server verbunden. Anzeigen und Ausblenden von Instrumenten und Anzeige von Daten wird getestet und die Anmeldung überprüft.

/GT120/ **Einparken vereinfachen**

Vorbedingung: Der Nutzer ruft die Einparkhilfe-Funktion auf.

Ablauf: Das Bild der Rückfahrkamera mit der aktuellen Fahrspur wird angezeigt. Der Fahrer bewegt das Auto. Es wird geprüft, ob das Bild regelmäßig aktualisiert wird.

/GT130/ **Ultraschallsensoren visualisieren**

Vorbedingung: Der Nutzer ruft die Einparkhilfe-Funktion auf.

Ablauf: Das Bild der Kamera mit den Distanzwerten der Ultraschallsensoren (/FA190/) wird angezeigt.

/GT140/ **Einparkhilfe beenden**

Vorbedingung: Einparkhilfe ist angezeigt.

Ablauf: Der Fahrer fährt mit entsprechender Geschwindigkeit (/FA200/) vorwärts. Es wird überprüft, ob die Einparkhilfe ausgeblendet wird.

/GT150/ **Karte anzeigen**

Vorbedingung: Ein Endgerät mit Lokalisierungsinformationen ist mit dem Server

verbunden.

Ablauf: Der Benutzer fordert den Kartendienst an. Es wird überprüft, ob die Umgebungskarte angezeigt wird.

/GT160/ **Tankstellen auflisten**

Vorbedingung: Ein Endgerät, das über den aktuellen Standort verfügt, ist mit dem Server verbunden.

Ablauf: Der Fahrer fährt mit dem Auto bei niedrigem Tankfüllstand. Es wird überprüft, ob eine Liste der Tankstellen in der Umgebung angezeigt und die Tankstellen auf der Karte markiert werden.

10 Testszenarien

10.1 Allgemeiner Funktionstest

- 01 Bob, ein Diplominformatiker der Schwäbischen Hochschule für Informationstechnologie setzt sich in seinen Mittelklassewagen.
- 02 Er schaltet die Zündung an und verbindet sein Smartphone mit dem Server.
- 03 Bob öffnet zum ersten mal die Software im Browser seines Smartphones. (/GT010/)
- 04 Er identifiziert sich als Fahrer (/GT030/)
- 05 und startet den Motor, sobald er seine Dashes ausgewählt hat und die Dashes angezeigt werden. (/GT050/)
- 06 Ihm werden nun die aktuellen Sensorwerte angezeigt. (/GT080/)
- 07 Weil Bob versucht spritsparend zu fahren, lässt sich er sich unter anderem die Durchschnittsgeschwindigkeit anzeigen. (/GT090/)
- 08 Während der Fahrt sinkt der Tankfüllstand unter den definierten Wert.
- 09 Darauf öffnet Bob mit dem Druck auf das Tank-Icon die Karte, (/GT150/)
- 10 wählt die nächste Tankstelle aus und bekommt diese dann auf der Karte angezeigt. Er tankt und setzt sich wieder in das Auto. (/GT160/)
- 11 Bob schaltet die Zündung wieder an, das Smartphone verbindet sich mit dem Netzwerk und der Diplominformatiker startet die Software. (/GT020/)
- 12 Da vor ihm noch ein Auto steht, muss Bob erst einmal etwas nach hinten fahren. Dazu wechselt er in den Rückfahrkameramodus der Software und sieht über die Kamera, wie weit er noch nach hinten fahren kann. (/GT120/, /GT130/)
- 13 Nachdem er die Tankstelle verlassen hat und wieder schneller fährt beendet sich die Einparkhilfe automatisch. (/GT140/) Irgendwann kommt Bob dann an seiner Hochschule an, schaltet den Motor ab und verlässt sein Auto.
- 14 Auf der Fahrt nach Hause, bei der Bob sich wieder mit dem Server, holt er unterwegs seine Schwester von der Arbeit ab. Da sie erst vor kurzem ihren Führerschein gemacht hat, lässt er sie den Rest der Strecke fahren. Sie verbindet ihr Smartphone mit dem Server und meldet sich als neuen Fahrer an. (/GT040/)

- 15 Bob, der auf seinem eigenen Smartphone die Fahrtdaten beobachtet, schaut sich den durchschnittlichen Benzinverbrauch an, um das Fahrverhalten seiner Schwester beurteilen zu können. (/GT060/)

10.2 Remote Test

- 01 Alice, Bobs Frau traut ihm nicht ganz und hat sich, nachdem sie von diesem neuen Gerät gehört hat, sofort mit Bobs Auto verbunden. (/GT020/)
- 02 Während Bob gerade auf dem Weg zu seiner Hochschule ist, schaut sich Alice die Daten der vergangenen Fahrt an. (/GT110/)
- 03 Um Bob länger ausspionieren zu können, setzt sie die maximale Kapazität der Aufzeichnungen auf den Höchstwert, da die Daten der vorletzten Fahrt bereits gelöscht waren. (/GT100/)

11 Qualitätsbestimmung

Produktqualität	Sehr gut	Gut	Normal	Irrelevant
Funktionalität		X		
Angemessenheit	X			
Richtigkeit			X	
Interoperabilität	X			
Sicherheit				X
Zuverlässigkeit		X		
Reife	X			
Fehlertoleranz	X			
Wiederherstellbarkeit				X
Benutzbarkeit	X			
Verständlichkeit	X			
Erlernbarkeit	X			
Bedienbarkeit	X			
Effizienz			X	
Zeitverhalten		X		
Verbrauchsverhalten			X	
Änderbarkeit		X		
Analysierbarkeit		X		
Modifizierbarkeit	X			
Stabilität		X		
Prüfbarkeit		X		
Übertragbarkeit			X	
Installierbarkeit			X	
Konformität			X	
Austauschbarkeit		X		

12 Glossar

- **Benutzer, Nutzer, User:** Jeder Nutzer der Software
- **Durchschnittsverbrauch:** Errechneter durchschnittlicher Verbrauch an Kraftstoff pro 100 Kilometer.
- **Dash:** Anzeigeelement, virtuelle Armatur
- **Dashboard:** Gesamtanzeige der Elemente; virtuelles Armaturenbrett
- **Endgerät:** Tablet, Smartphone, Laptop oder Desktop-PC.
- **Fahrer:** Der Benutzer, der das Fahrzeug zur Zeit fährt.
- **Fahrinformationen:** Die Gesamtheit aller für den aktuellen Fahrer während der Fahrt interessanten Sensordaten.
- **GUI, UI:** (Graphical) User Interface: Die grafische Benutzerschnittstelle der Software.
- **kritischer Sensorwert:** Sensorwert, der schwere Auswirkungen auf das zukünftige Verhalten oder die Funktion des Fahrzeugs haben kann.
- **POI:** Point of Interest; Ort in der Nähe, der (Aufgrund bestimmter Ereignisse) für den Benutzer interessant ist.
- **Rahmenwerk:** Softwaretechnisches Gerüst, in dem komponentenbasiert Funktionen hinzugefügt werden können.
- **Remote Access:** Zugriff von einem Endgerät, das sich nicht in unmittelbarer Umgebung des Fahrzeugs befindet.
- **Server:** Dienstleister im Client-Server Modell.
- **System:** Die Gesamtheit der im Pflichtenheft beschriebenen Hardware und Software.

13 Anhang

13.1 Quellen und Credit

13.1.1 Quellen

- <http://simpleicon.com>
- <https://icons8.com>

13.1.2 Credit

AndroidTM platform und ChromeTM browser sowie deren Logos sind eingetragene Handelsnamen von Google Inc.

Versionshistorie

Version	Datum	Autor(en)	Änderungen
1.0	29.11.2015	V. Bykovski , J. Haas , D. Grajzel , N. Schreiber , V. Springsklee , Y. Zhu	-
1.1	30.01.2016	J. Haas	Betriebsbedingung hinzugefügt: Server-System muss das aktuelle Datum kennen.