

Hochschule Merseburg (FH)
University of Applied Sciences

Fachbereich: Ingenieur-
und Naturwissenschaften

Masterarbeit

zum Thema

Darstellung der GSM Abdeckung basierend auf Messdaten von IoT-Devices

Zur Erlangung des akademischen Grades eines
Master of Engineering (M.Eng)

Vorgelegt von

Aymen Ghdamsi (Matr. -Nr. 22471)

Studiengang: Master Informatik und Kommunikationssysteme

Im Wintersemester 2018/2019

Begutachtet von

Herr Prof. Dr. phil. Dr. rer. Nat. habil. Michael Schenke

Herr Dr. Rupprecht Gabriel

Sperrvermerk

Diese Masterarbeit basiert auf internen und vertraulichen Daten der Firma CargoGuard. Diese Arbeit darf Dritten, mit Ausnahme der betreuenden Dozenten, befugten Mitgliedern des Prüfungsausschusses und dem Prüfungsamt, ohne ausdrückliche Zustimmung des Unternehmens und des Verfassers nicht zugänglich gemacht werden.

Eine Vervielfältigung und Veröffentlichung der Arbeit ohne ausdrückliche Genehmigung – auch in Auszügen – ist nicht erlaubt.

Eidesstattliche Erklärung

Ich versichere eidesstattlich durch eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Merseburg, im März 2019

Aymen Ghdamsi

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt und motiviert haben.

Zuerst gebührt mein Dank Prof. Dr. Michael Schenke, der meine Masterarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen, seine Geduld und seine Verfügbarkeit möchte ich mich herzlich bedanken.

Ich bedanke mich bei dem Geschäftsführer der CargoGuard Firma, Dr. Rupprecht Gabriel, um mir die Möglichkeit zu geben, eine gute Erfahrung in Bereich IT zu machen.

Ebenfalls möchte ich mich bei meinem Kollege Benjamin Kissmann bedanken, die mir mit viel Geduld, Interesse und Hilfsbereitschaft zur Seite standen. Bedanken möchte ich mich für die zahlreichen interessanten Debatten und Ideen, die maßgeblich dazu beigetragen haben, dass diese Masterarbeit in dieser Form vorliegt.

Abschließend möchte ich mich bei meinen Eltern, meine Schwester und meiner Verlobten Wejden Bouzidi bedanken, die mir mein Studium durch ihre moralische und finanzielle Unterstützung ermöglicht haben.

Kurzzusammenfassung

Das Internet of Things bezieht sich auf Milliarden von physischen Geräten auf der ganzen Welt, die jetzt mit dem Internet verbunden sind, und Daten sammeln und gemeinsam nutzen. Das Ziel der vorliegenden Masterarbeit ist es, die GSM-Abdeckung grafisch auf einer Karte basierend auf Messdaten von einigen IoT-Devices darzustellen.

Die GSM-Abdeckung kann in verschiedene Formen auf der Karte dargestellt werden. In dieser Arbeit werden mehreren Ideen diskutiert, um ein gutes Ergebnis am Ende der Arbeit zu erhalten.

Die Messdaten dieser IoT-Devices könnten für mich große Probleme mitbringen und das Aussehen der Abdeckung auf der Karte verfälschen, weil die Daten auf dem Server fehlerhaft abgespeichert werden. Deshalb werden in diesem Projekt verschiedene GSM-Abdeckung Karten dargestellt, um die beste, die unserem Zweck entspricht, zu wählen.

Abstract

The Internet of Things refers to billions of physical devices around the world in this moment connected to the Internet, collecting and sharing data. So, my objective of this master thesis was to present a GSM coverage map based on measurement data from some of this IoT devices.

Since the GSM coverage map can be presented in different forms, several ideas are discussed to get a good result at the end of the work.

The measurement data of this IoT device can create a big problem for me to get a good coverage look on the map, because IoT-Device share randomly the data to the server, therefore in this project different GSM coverage map are presented, to choose the best one, that suits our purpose.

Inhaltsverzeichnis

1	Einleitung.....	4
1.1	CargoGuard Firma.....	4
1.1.1	CargoGuard Lösungen	5
1.1.2	CargoCop Beschreibung	6
1.2	Motivation zur Aufgabenstellung.....	7
1.3	Anforderung und Ziel der Arbeit.....	7
1.4	Aufbau der Arbeit.....	8
2	Grundlagen.....	10
2.1	Internet of Things	10
2.1.1	Was ist Internet of Things	10
2.1.2	Architektur	11
2.1.3	Bereichen des IoT	14
2.2	Leaflet.js Bibliothek	15
2.2.1	Was ist leaflet.js?	15
2.2.2	Warum Leaflet?.....	16
2.2.3	Was braucht Man als Hilfsmittel	17
2.3	Web-Entwicklung.....	18
2.3.1	HTML- oder XHTML-Sprache.....	18
2.3.2	CSS – Cascading Style Sheets	19
2.3.3	Angular.....	21
2.3.4	Angular 2+	22

2.3.5	Node.js	23
2.3.6	Rest Api Server mit C sharp Programmierungssprache.....	24
2.3.7	Web Application Architecture	26
2.4	Datenerhalten.....	26
2.4.1	Microsoft SQL Server	27
2.5	Zusammenfassung	30
3	Projektentwurf.....	31
3.1	Identifikation von Akteuren	31
3.2	Entwicklung von Anwendungsfalldiagrammen	32
3.3	Dynamische Diagramme	34
3.3.1	Sequenzdiagramm	34
3.3.2	Sequenzdiagramm der All Units Option	35
3.3.3	Sequenzdiagramm der Unit Option.....	36
3.3.4	Aktivitätsdiagramm.....	37
3.4	Klassendiagramme	39
3.5	Zusammenfassung	40
4	Realisierung.....	42
4.1	Datenbank-Seite	42
4.1.1	Die Daten	42
4.1.2	Vorbereitung der Daten.....	44
4.1.3	Verarbeitung der Daten	47
4.2	Verbindung des Rest Api Servers- und Client Seite	51

4.3	Ausführung der Anwendung	54
4.3.1	Erste Versuch mit der GSM Abdeckung.....	59
4.3.2	Das finale Ergebnis der GSM Abdeckung	61
4.3.3	Die Benutzeroberfläche der Anwendung	63
4.4	Zusammenfassung	70
5	Fazit.....	71

1 Einleitung

Dieses Projekt befasst sich mit dem Bereich Internet of Things „IoT“, eine der größte IT-Technologien, die es heute gibt und welche sich in den letzten zehn bis zwanzig Jahre am meisten entwickelt hat.

Das Internet of Things kann in allen Bereichen gefunden werden, wie smart Cities, e-Health, Industrie, Logistik, smart Grid, Automation, Umgebung und Transport, die Domäne, die mich interessiert und in der ich mein Projekt entwickelt habe.

IoT bezieht sich auf eine Art vom Netzwerk, um auf der Grundlage festgelegter Protokolle alles über das Erkennen von Informationen mit dem Internet zu verbinden, um Informationsaustausch und Kommunikation durchzuführen, um intelligentes Erkennen, Positionieren, Verfolgen, Überwachen und Verwalten zu erreichen.

Dies wird mit Hilfe von mehreren Geräten, die miteinander verbunden sind, realisiert. Im Jahre 2018 lag die Anzahl der angeschlossenen Geräte bei über elf Milliarden - und das beinhaltet nicht Computer und Telefone. Deswegen wird IoT weiterhin ein heißes Thema in den nächsten Jahren sein.

Für diese Arbeit wurden solche Geräte wie die oben genannten für das Messen von Daten verwendet.

Im nächsten Schritt beschreibe ich diese Geräte und gebe einen kurzen Überblick über die Firma CargoGuard.

1.1 CargoGuard GmbH

Die Firma existiert seit 2007 und bietet IoT-basierte Sicherheits- und Lieferketten-Transparenzlösung als Logistikdienstleister an:

- ✓ One-Stop-Shop-Lösungen zum Lokalisieren und Routen, Überwachen und Sichern, Analysieren und Verwalten von Transporten weltweit und jederzeit.
- ✓ Höchste digitale Sicherheit und Zuverlässigkeit durch eigene Entwicklung von Hard- und Software.

- ✓ Echtzeitüberwachung mehrerer Umgebungsbedingungen
- ✓ Wettbewerbsfähige Systemkosten sowie flexible Geschäftsmodelle machen CargoGuard für globale Kunden attraktiv.



Abbildung 1.1: CargoGuard Logo

1.1.1 CargoGuard Lösungen

In diesem Schritt gebe ich einen kurzen Überblick über die verschiedenen Lösungen der Firma CargoGuard und im Anschluss beschreibe ich detailliert das Modul CargoCop, das für dieses Projekt genutzt wird.

1.1.1.1 CargoCop

- Fest und mobil
- GPS / GSM-Tracker
- Mehrere integrierte Sensoren und Aktoren
- Peripherie-IoT-Sensoren für mehrere Umgebungsbedingungen

1.1.1.2 CargoShield

- Verfolgte und überwachte Sicherheitsboxen
- IoT-Sperren und IoT-basierter kontrollierter Zugriff
- Intrusion Sensing und Ink-Technologie

1.1.1.3 CargoControl

- Plattform zur Steuerung und Verwaltung der digitalen Logistik
- Cloud-basierter Sicherheitsserver und Webportal
- Zeit-, orts- oder umgebungsbedingte Regel-Engine

1.1.2 CargoCop Beschreibung

Ist ein robuster, leichter Tracker für raue Logistikumgebungen, und besitzt deshalb wasserdichte, vergossene Elektronik und ein stoßfestes Gehäuse. Für die einfache Installation gibt es verschiedene Optionen wie die Anbringung mit Klebeband oder Magnet.

Der Tracker integriert Sensoren für mehrere Umgebungsbedingungen: wie z.B Temperatur, Licht, Beschleunigung / Schock, Feuchtigkeit und hat keine Antennen an der Außenseite des Anhängers.

Im Bereich des Energieverbrauchs ist das Modul CargoCop für den geringen Stromverbrauch ausgelegt.



Abbildung 1.2: CargoCop

1.1.2.1 CargoCop-Peripheriegeräte

- ✓ Mit den IoT-Datenloggern von CargoGuard wird jede Sendung zu einem in Echtzeit überwachten IoT-Element, das sich auf den Sicherheits- und Umweltstatus bezieht.
- ✓ Echtzeitüberwachung und -kontrolle der Sendung.
- ✓ Es ist kabellos und über ISM Band verbunden

- ✓ Permanente Aufzeichnung von Temperatur und Luftfeuchtigkeit (alle sieben Sekunden).
- ✓ Bei Bedarf können andere Sensoren mit diesem Gerät aufgenommen werden (z. B. Schock und PIR).

1.1.2.2 CargoCop Optionen

- Flexible Batteriekonzepte
 - AA-Standard Alkaline, 11 oder 22 Wh
 - AA-Standard-Lithium, 16 oder 32 Wh, arbeitet unter -30°C
 - Wiederaufladbarer Li-Ion-Akku mit Qi-Ladung, 19 oder 37 Wh, moderne, hoch aufladbare, mehrfach wiederaufladbare Stromquelle.
- Zusätzliche integrierte Sensoren: Luftdruck, berührungsloser NFC-Näherungssensor (z. B. berührungslose Erfassung des Türstatus)
- Mehrere drahtlose Schnittstellen zum Anschluss von Peripheriegeräten
 - Temperatur, Luftfeuchtigkeit, Sirene, PIR-Sensor, Lastsensor usw.

1.2 Motivation zur Aufgabenstellung

Als Aufgabe bei der Firma CargoGuard bearbeite ich gerade ein Frontend Projekt, bei dem die Anzeige von Geo-Position und das Routing den Hauptteil einnimmt. Es war ein ganz neues Thema für mich aber ich finde es sehr interessant und man kann es in verschiedenen Bereichen und Richtung nutzen, um neue Projekte zu entwickeln.

Deswegen haben wir entschieden, dass ich meine Masterarbeit in dieser Richtung mit anderen Anforderungen weiterentwickle.

1.3 Anforderung und Ziel der Arbeit

Der am häufigsten auftretende Alarm bei der Firma CargoGuard ist der so genannte Tracking interval Failure Alarm. Dieser entsteht, wenn keine Verbindung zum Server aufgebaut werden kann, da z.B. das Mobilfunknetz nicht funktioniert.

Deswegen ist mein Ziel des Projektes eine Darstellung der GSM-Abdeckung auf einer Karte aus den gemessenen Daten dieser IoT-Geräte.

Damit könnte man, die Signalstärke abhängig von dem Ort anzeigen sehen.

In dieser Arbeit nutzte ich mehrere Technologien, um dieses Ziel zu erreichen und werde diese im nächsten Kapitel beschreiben.

1.4 Aufbau der Arbeit

Nach der Einleitung werden die verschieden genutzten Technologien in dem theoretischen Teil, den Grundlagen, erläutern.

In diesem Kapitel findet man einen Überblick über das Thema IoT, die genutzten Methoden im Projekt, die Programmiersprachen, die Struktur der Arbeit, Diagrammen, usw.

Im nächsten Kapitel wird die Arbeit mit dem Konzept, Realisierung, Test usw. in einem Praktische Teil präsentiert.

Im letzten Kapitel wird eine Zusammenfassung der Arbeit, sowie eine Diskussion und die mögliche zukünftige Entwicklung in dieser Richtung kurz erläutert.

Teil 1

Theoretische Planung und Vorbereitung

2 Grundlagen

In diesem Kapitel werde ich die verschiedenen Technologien, die ich in meinem Projekt genutzt habe, erläutern.

Der GPS Tracker ist ein wichtiger Teil in meiner Arbeit und gehört zur IoT Technologie. Deswegen kann ich nicht über andere Grundlagen schreiben, bevor ich die IoT Technologie mit ihrer Definition, Struktur und Bereichen erkläre.

2.1 Internet of Things

2.1.1 Was ist Internet of Things

Das IoT kann als eine Sammlung von Geräten klassifiziert werden, die mit Sensoren, Aktoren und Prozessoren ausgestattet sind, die miteinander kommunizieren, um einen sinnvollen Zweck zu erfüllen.

Sie wird auch als Wechselwirkung zwischen der physischen und der digitalen Ebene definiert. Diese Wechselwirkung erfolgt mit Hilfe verschiedener Sensoren, die an allen denkbaren Objekten angebracht sind [1].

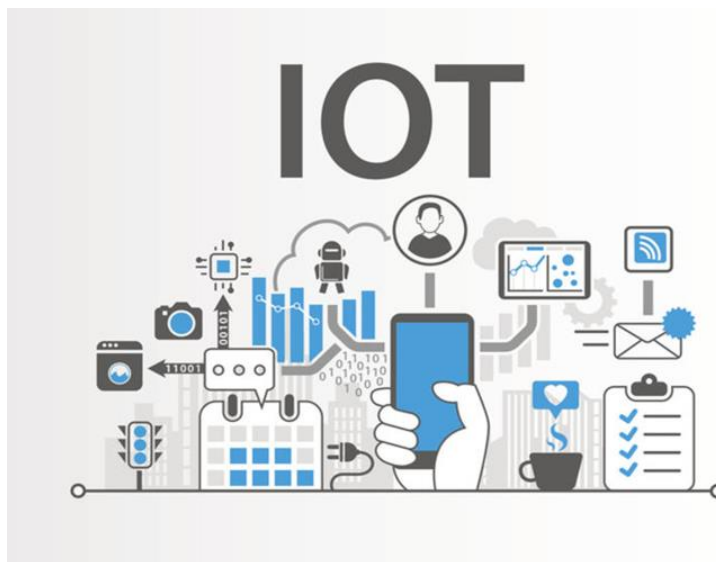


Abbildung 2.1: IoT

Quelle : <https://www.zdnet.com/article/the-future-of-enterprise-iot-2-factors-to-watch/>

2.1.2 Architektur

Die IOT-Architektur besteht aus verschiedenen Schichten von Technologien, die IoT unterstützen.

Sie soll veranschaulichen, wie verschiedene Technologien miteinander zusammenhängen, und die Skalierbarkeit, Modularität und Konfiguration von IOT-Bereitstellungen in verschiedenen Szenarien kommunizieren.

Diese Beschreibungsteil ist sehr wichtig, so kann man verstehen, wie ich den GPS Tracker genutzt habe und mit welchen Funktionen.

Die IoT Struktur besteht aus den folgenden Teilen:

- Sensors/ Aktuators (drahtlos/verdrahtet).
- Internet Gateways und Networks
- Management Service Layer
- Data Center / Cloud
- Application Layer

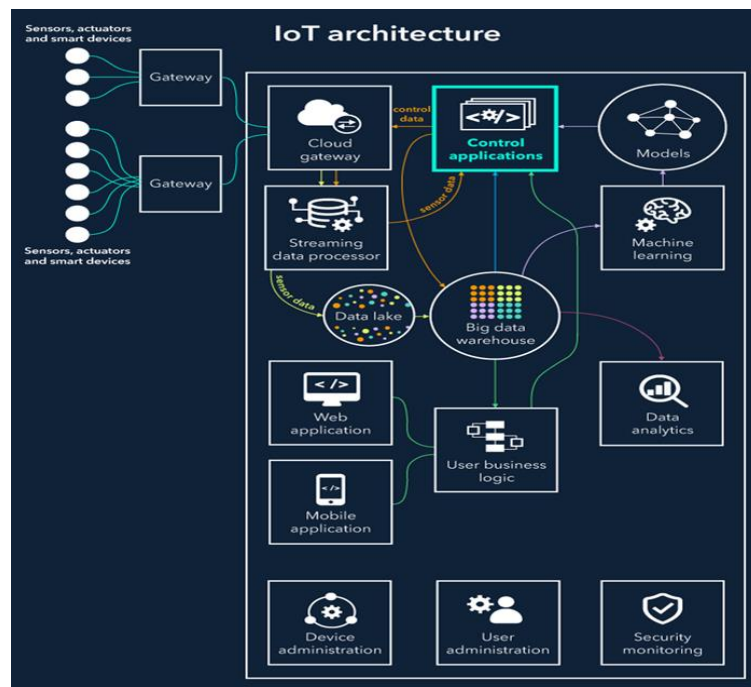


Abbildung 2.2: IoT-Architektur

Quelle : <https://www.scnsoft.com/blog/iot-architecture-in-a-nutshell-and-how-it-works>

Nachfolgende beschreibe ich die Funktionalität jeder Schicht in Details.

2.1.2.1 Sensors/ Aktuators

Die Sensoren sind die erste physikalische Schicht in der Architektur. Sie erfassen und sammeln die Daten und Messungen wie Temperatur, Luftqualität, Geschwindigkeit, Feuchtigkeit, Druck, Strömung, Bewegung und Elektrizität usw. aus der Umgebung.

Seine Arbeit umfasst die Erfassung physikalischer Parameter oder die Identifizierung anderer intelligenter Objekte.

Die Sensoren können überall sein, als Beispiel: die Temperatur Sensoren, Parkplatz Sensoren.

Dann kommen die Aktoren, die eine Veränderung der Umgebung beeinflussen können.

Als Beispiel: Ein Sensor erkennt, dass sich die Temperaturen geändert haben und das Licht niedrig ist. Daher schaltet ein Aktor automatisch die Straßenbeleuchtung ein.

2.1.2.2 Internet Gateways Layer und Networks

2.1.2.2.1 Internet Gateways

Die Gateways sammeln die Daten von Sensor knoten und übertragen die an die Infrastruktur.

Die Daten, die von den Sensoren kommen, müssen vorbereitet werden, bevor sie in die endgültige Verarbeitungsphase gelangen. Grundsätzlich müssen die Daten, die in analoger Form empfangen werden, aggregiert und in digitale Form konvertiert werden, und diese Schicht tut dies mit Hilfe eines Internet-Gateways, das sie zur weiteren Verarbeitung über WLANs ^[9] oder andere Netzwerke weiterleitet.

Diese Gateways können mit zusätzlichen Funktionalitäten wie Analyse, Schutz vor Malware und Datenverwaltung aufgebaut werden.

2.1.2.2.2 Networks

Mit diesen winzigen Sensoren wird ein enormes Datenvolumen erzeugt. Dies erfordert eine robuste und leistungsstarke Infrastruktur für kabelgebundene oder drahtlose Netzwerke als Transportmedium.

Aktuelle Netzwerke, die oft mit sehr unterschiedlichen Protokollen verbunden sind, wurden verwendet, um M2M-Netzwerke (Machine-to-Machine) -Netzwerke und ihre Anwendungen zu unterstützen.

Da die Nachfrage nach einem breiteren Spektrum an IoT-Diensten und -Anwendungen wie schnellen Transaktionsdiensten, kontextabhängigen Anwendungen usw. besteht, sind mehrere Netzwerke mit verschiedenen Technologien und Zugriffsprotokolle erforderlich, um in einer heterogenen Konfiguration miteinander zu arbeiten.

2.1.2.3 Management Service Layer

Datenverwaltung ist die Fähigkeit, den Dateninformationsfluss zu verwalten.

Mit der Datenverwaltung in der Verwaltungsdienstschicht können Informationen abgerufen, integriert und gesteuert werden. Anwendungen mit höheren Schichten können vor der Notwendigkeit, unnötige Daten zu verarbeiten, abgeschirmt werden und das Risiko der Offenlegung der Datenquelle durch den Datenschutz verringern.

Datenfilterungstechniken wie die Anonymisierung von Daten, Datenintegration und Datensynchronisierung werden verwendet, um die Details der Informationen auszublenden, während sie nur die wesentlichen Informationen bereitstellen, die für die jeweiligen Anwendungen verwendet werden können.

Mithilfe der Datenabstraktion können Informationen extrahiert werden, um eine allgemeine Geschäftsansicht der Daten bereitzustellen, um eine größere Flexibilität und Wiederverwendung in verschiedenen Domänen zu erreichen [2].

2.1.2.4 Data Center / Cloud

Wo die Daten werden auf starken IT-Systemen analysiert, verwaltet und gespeichert. Diese Daten sind die, die mehr verarbeitet werden müssen, und das Feedback muss nicht sofort erfolgen.

Ein Rechenzentrum oder ein Cloud-basiertes System eignet sich perfekt für diesen Zweck. Hier dauert das Ergebnis einige Zeit, aber das, was zur Verfügung steht, ist detaillierter und kann mit anderen Daten kombiniert werden, um tiefere Einblicke zu erhalten.

2.1.2.5 Application Layer

Dieser Layer stellt die anwendungsspezifischen Dienste für den Benutzer bereit.

Die IoT-Anwendung deckt „intelligente“ Umgebungen in folgenden Bereichen ab: Transport, Gebäude, Stadt, Lifestyle, Einzelhandel, Landwirtschaft, Fabrik, Lieferkette, Notfall, Gesundheitswesen, Benutzerinteraktion, Kultur und Tourismus, Umwelt und Energie [3].

2.1.3 Bereichen des IoT

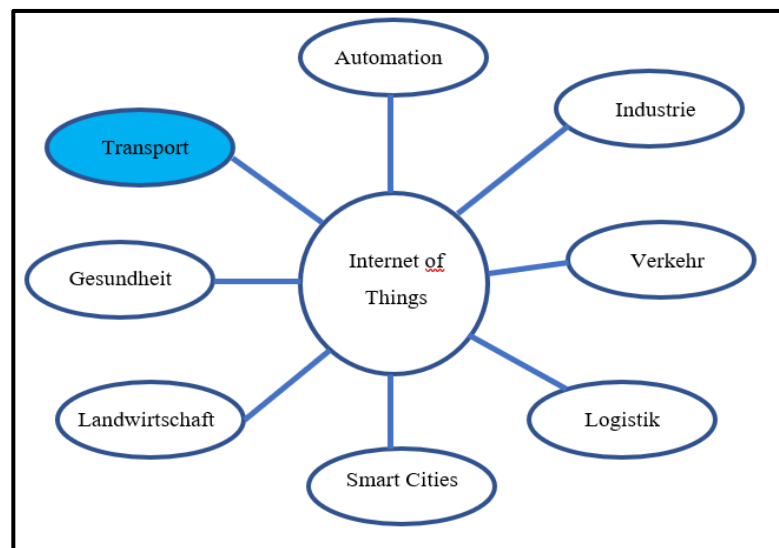


Abbildung 2.3: IoT-Bereichen

Wie in der Abbildung 2.3 gezeigt ist, findet man derzeit IoT überall und in alle Domänen; Smart Cities, Transport, e-Health, Industrie, Logistik, Verkehr,

Automation, Landwirtschaft usw. aber was mich interessiert ist die Domäne Transport, in der ich mein Projekt entwickelt habe.

Verschieden Technologie könnten in diesem Bereich verwendet werden, zum Beispiel bei der Flottenverfolgen:

Die Fahrzeugflottenverfolgungssysteme verwenden GPS-Technologie, um die Standorte der Fahrzeuge in Echtzeit zu verfolgen.

Die Fahrzeugstandorte und Routendaten können analysiert werden, um Engpässe in der Lieferkette, z.B. Verkehrsstaus auf Routen, Zuweisungen und Generierung alternativer Routen, sowie die Optimierung der Lieferkette zu ermitteln.

Ein anderes Beispiel, bei dem diese Technologie verwendet wird ist die Sendungsüberwachung:

Sie hilft bei der Versandüberwachung von Transportsystemen und kann auch die Überwachung der Bedingungen in Containern ermöglichen [4].

2.2 Leaflet.js Bibliothek

2.2.1 Was ist leaflet.js?

Leaflet.js ist eine open Source JavaScript Bibliothek, welche macht die Bereitstellung von Maps auf einer Website sehr einfach. Open Source das heißt, dass der Code von jedem genutzt werden kann und jeder kann auch mit Verbesserung der Code zum Projekt beisteuern.

Um diese Mapping bereitzustellen, ist eine spärliche JavaScript Datei verwendet, die mit der Website geladen wird und bietet eine Reihe von Funktionen, mit denen man eine Karte anzeigen kann. Heutzutage gibt es moderne Desktop Browser, die auf Desktop und mobilen Plattformen unterstützen werden, sodass die Karte überall bereitstellen können.

Ein paar Beispielen von Firma, die bereits diese Leaflet verwenden; craigslist, Flickr, foursquare, CloudMade, CartoDB, Meetup, WSJ.

Leaflet bietet auch verschiedene Funktionalität, um Markierungen, Popups, Überlagerungslinien und formen- hinzufügen.

Eine der wesentlichen Stärken von Leaflet ist die Fähigkeit, die Funktionalität des Skripts mit Plugins von Third-Partys zu erweitern. Es gibt über 80 separate Plugins, mit denen können einer Heatmap, das Animieren von Markern, das Laden von CSV-Dateien von Daten, das Zeichnen komplexer Formen, das Messen von Entfernungen, das Bearbeiten von Ebenen und das Anzeigen von Koordinaten ermöglichen [5].

2.2.2 Warum Leaflet?

2.2.2.1 Mein prinzipieller Grund

Zu viele interaktive Mapping-Plattformen existieren, als Beispiele; Google Maps API, Mapbox.js, Leaflet, Open Street Maps, usw. aber was ist den Unterschied zwischen diesen verschiedenen Plattformen und warum habe ich Leaflet gewählt.

Niemand kann bestreiten, dass Google Maps API definitiv der Große Boss für Interaktives Mapping online ist. Es ist in der Regel die erste Plattform für solche Anwendungen aber ein großes Problem mitkommt und verhindert mich, um diese Google API zu nutzen.

Google Maps API ist leider nicht open source. Um es zu nutzen, braucht man ein Schlüssel und muss selbstverständlich auch für diese Schlüssel was zu bezahlen.

im Gegensatz zu Leaflet, ist ein Open Source Plugin mit vielen verfügbaren Mapping für JavaScript, über das viele Plugins und Addons zur Verfügung stehen.

2.2.2.2 Unterschiede zwischen Karten

2.2.2.2.1 Karte-Dienste

Obwohl Google viele eigene Dienste, wie Verkehr, geolocation, Transit usw., die von anderswo Google API einfach erhalten können, bietet, stützt Leaflet sich dabei auf Third-Party Dienste wie (wie Nominatim oder andere).

2.2.2.2.2 Dokumentation

Leaflet enthält eine standardisierte JavaScript-Plugin-Dokumentation sowie online Ressourcen in Web also man kann einfacher die gesuchte Information finden. Die Entwickler finden die Leaflet Dokumentation besser als Google.

2.2.2.2.3 Der Code

Beide Google Maps API und Leaflet sind in Bezug auf die Codequalität herausragend.

Beide Karten können benutzerdefinierte Markierungen erstellen, GeoJSONs hinzufügen, Layer-Interaktion, Popups und vieles mehr. Nur bei sehr großen oder sehr spezifischen Implementierungen werden die Unterschiede deutlicher.

- Leaflet ist für die Entwickler zugänglich, da das Plugin Open Source ist.
- Beide Maps sind schnell und ansprechend
- Leaflet ist Open-Source bedeutet, dass sie täglich von einer Welt von Entwicklern geprüft wird und nicht durch die Nutzungsbedingungen von Google eingeschränkt wird [6].

2.2.2.2.4 Karte Schnittstelle und Schichten

Google zwingt Nutzer zur Verwendung der standardmäßigen Google-Basisebene, obwohl diese anpassbar ist.

Im Gegensatz dazu ist Leaflet für die Verwendung mit mehreren Arten von Basisebenen ausgelegt. Es ist nicht erforderlich, Logos oder Attributionen aufzubewahren. Es gibt keine Leaflet-Kartographen, die Daten sammeln, wie beim Google. Daher ist Leaflet von Third-Partys abhängig, um den Benutzern verschiedene Kartenebenen zur Verfügung zu stellen [6].

2.2.3 Was braucht Man als Hilfsmittel

- Hypertext Markup Language (HTML)
- JavaScript oder TypeScript
- Cascading Style Sheets (CSS)
- Web Servers

Ich beschreibe einzelne Objekt in Details im nächsten schritt

2.3 Web-Entwicklung

Webprogrammierung ist Computerprogrammierung, mit der Sie Websites bearbeiten können. Es kann verschiedene Formen annehmen: von der einfachen statischen Seite bis zur dynamischen Seite mit Verbindung zu einer Datenbank.

2.3.1 HTML- oder XHTML-Sprache

Jede Webseite enthält eine Basis von HTML oder XHTML. Es ist eine Auszeichnungssprache, die im Wesentlichen die Struktur der Webseite (Titel, Tabellen, Absätze usw.) definiert.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Seitentitel</title>
  </head>
  <body>
    <h1>Dies ist eine Überschrift</h1>
    <p>Dies ist ein Absatz.</p>
  </body>
</html>
```

Listing 1: Ein Beispiel eines HTML-Skript

Dieses Skript hat eine bestimmte Struktur, wie oben im Bild gezeigt ist. Die erste Zeile deklariert die Art des Dokumentes; <!DOCTYPE html> und <html>.

Jede HTML Seite hat ein <head> und <body>;

- „head“ kommt mit einem Titel für das Tabs in Web Browser
- „body“ ist der Inhalt von einen HTML Seite, der auch mit verschiedener Struktur kommt. Als Beispiel:
 - <h1>: die große von Schriften, 1 ist die größte zahle bis 6 die kleinste
 - <p>: Anfang zum schreiben

- Es gibt auch anderen Strukturen innerhalb des `<body>` wie `<div>`, `<table>`, usw. jede hat seine eigene Funktion.



Abbildung 2.4: HTML-Seite

Die Abbildung 2.4 erklärt, wie ein HTML-Code auf einen Web-Browser kompiliert wird, der Code wird erst als eine .html Datei gespeichert und mit irgendeinem webBrowser geöffnet.

Nr.1 im Bild ist `<head>`: Seitentitel

Nr.2 ist `<h1>` im Code

Nr.3 Ist `<p>` im Code

2. und 3. sind innerhalb `<body>` und jede Struktur soll in HTML Code geschlossen werden z.B `<body>` mit `</body>` oder `<p>` mit `</p>`

2.3.2 CSS – Cascading Style Sheets

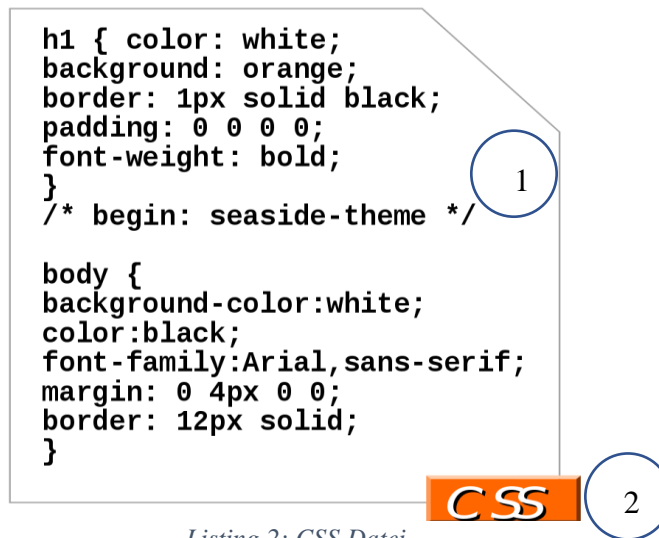
Cascading Style Sheets (CSS) ist eine Stylesheet-Sprache, mit dem die Darstellung eines Dokuments beschrieben wird, das in einer Markup-Sprache wie HTML geschrieben ist. Es ist neben HTML, JavaScript und TypeScript eine den Eckpfeilern des World Wide Web.

CSS wurde entwickelt, um die Trennung von Darstellung und Inhalt, einschließlich Layout, Farben und Schriftarten zu ermöglichen. Diese Trennung kann Inhalt Zugänglichkeit verbessern, mehr Flexibilität und Kontrolle in der Beschreibung der Präsentationseigenschaften bieten, mehr Web-Seiten ermöglicht die Formatierung zu

teilen, indem Sie die entsprechende CSS in einer separaten CSS-Datei spezifiziert, und die Komplexität und die Wiederholung des strukturellen Gehalt reduzieren [7].

```
h1 { color: white;
background: orange;
border: 1px solid black;
padding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
border: 12px solid;
}
```



Listing 2: CSS Datei

Listing 2, Nr.1 zeigt, wie die Anweisungen in einer CSS Datei gespeichert sind und Nr. 2 präsentiert das Ergebnis.

Wie ich im HTML Teil erklärt habe, (h1) ist die Größe von Schriften aber hat auch als Einstellung Farben, Hintergrund, Rand, Schriftgewicht usw.

- Color: white; definiert die Schriftfarbe Weiß
- Background: orange; der Hintergrund Orange wird dargestellt
- Border: 1px solid black; ist die Größe, Art und Farbe von Grenzen
- Font-weight: bold; ist der Art von Schriften

Beim body-tag, der beschreibt den Inhalt einer Website und wo div und andere Absatz existieren, gibt es auch Einstellung für Schriften, wie Farben, Art, Rand, Marge usw.

- Font-family: Arial, sans-serif; Art von Schriften
- Color: black; Farbe von Schriften

Nr.2 ist das Ergebnis der Kombination aus HTML Datei und den dargestellte CSS-Anweisungen. Im Folgenden sind Beispiele aufgezeigt, wie eine CSS Datei in einer HTML Datei eingebunden wird.

```
<div id="map">
```

Listing 4: ID in HTML

```
#map {  
    width:100px  
}
```

Listing 3: ID in CSS

Wenn man einen bestimmten Tag ändern möchte, kann diese Tag einen ID zugewiesen werden und dann im CSS als Bezeichnung nutzen. Dafür wird ein Doppelkreuz vor die ID geschrieben wie es im Listing 3 gezeigt ist.

Die ID von einem Tag wird über das Attribut (id) vergeben „Listing 4“.

Man kann auch eine externe Stylesheet-Datei einbinden. Diese wird in der Head von HTML Datei geschrieben.

```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

Listing 5: External Stylesheet

2.3.3 Angular

Angular ist ein Front-end web Framework und kommt mit verschiedenen Versionen. Erste Version von Angular 1.0 die nennt AngularJS basierend auf JavaScript Programmierung Sprache und das Angular 2+ bis angular 7 basierend auf TypeScript Sprache [8].

2.3.3.1 AngularJS oder Angular 1.x

Wie erwähnt AngularJS ist ein Front-end web Framework basierend auf JavaScript Sprache, das wird von Google, eine Community von Einzelpersonen und Unternehmen betrieben, um viele der Herausforderungen zu lösen, die bei der Entwicklung von Single-Page-Webanwendungen auftreten.

Es zielt, die Entwicklung und das Testen von solchen Anwendungen zu vereinfachen, indem ein Rahmen für clientseitige Model View Controller (MVC) und Model View Viewmodel (MVVM) -Architekturen, zusammen mit Komponenten, die üblicherweise verwendet in Rich-Internet-Anwendungen [9].

Wie funktioniert? - es liest am ersten die Seite Hypertext Markup Language (HTML), in die extra benutzerdefinierte HTML-Attribute eingebettet sind und dann führt diese Attribute als Anweisungen aus, um Eingabe- oder Ausgabeteile der Seite an ein Modell zu binden, das durch JavaScript-Variablen dargestellt wird.

In einem vorherigen Teil habe ich die HTML Sprache schon erklärt, in diesem Teil gebe ich einen kurzen Überblick über JavaScript so wird eine vollständige Idee über AngularJS erstellt.

2.3.3.1.1 JavaScript

Ist eine interpretierte High-Level-Programmiersprache, die der ECMAScript-Spezifikation entspricht.

JavaScript ist neben HTML und CSS eine der drei Kerntechnologien des World Wide Web und bietet mehrere Möglichkeiten zur Datenverarbeitung, von einfacher Anwendung wie eine Hallo Welt Anzeige bis zu komplexer Website [10].

Das Unterschied zwischen JavaScript und andere Programmiersprache wie C++/C, ist, dass der Code nicht kompiliert, sondern auf einen Browser ausgeführt wird.

Zu viele Framework unterstützt JavaScript neben AngularJS, nenne ich, jAbfrage Masonry, KineticJS usw.

2.3.4 Angular 2+

Angular 2+ bis heutzutage Version 7 kommt ohne JS, da es auf TypeScript basiert, aber wird auch von denselben Unternehmen als AngularJS betrieben

Man kann AngularJS und Angular mit dem Syntax unterscheiden.

Angular hat kein Konzept für "Scope" oder Controller, sondern verwendet eine Hierarchie von Komponenten als primäres architektonisches Merkmal.

Es hat eine andere Ausdruckssyntax, die sich auf "[]" für die Eigenschaftsbindung und "()" für die Ereignisbindung konzentriert.

Verfügt über eine eigene Suite moderner UI-Komponenten, die für das Web, Mobile und Desktop geeignet sind und als Angular Material bezeichnet werden.

2.3.4.1 TypeScript

TypeScript ist eine Open-Source-Programmiersprache, die von Microsoft entwickelt und verwaltet wird. Es wurde für die Entwicklung großer Anwendungen und Transpile nach JavaScript entwickelt.

Da TypeScript eine Obermenge von JavaScript ist, sind vorhandene JavaScript-Programme ebenfalls gültige TypeScript-Programme. TypeScript kann verwendet werden, um JavaScript-Anwendungen für die Ausführung auf Client- und Server-Seite (Node.js) zu entwickeln.

Die vorherige Teilen Leaflet Bibliothek sowie Web-Entwicklung mit den Untertiteln HTML, CSS und Angular beschreiben theoretisch die Methoden und Programmierungssprache, wie mein Ergebnis gezeigt werden kann. Aber um diese Daten zu zeigen braucht man eine Verbindung zwischen den Client und Datenbank. Dies wird durch eine Rest API Server mit verschiedene Programmierungssprache wie, Node.JS oder C sharp realisiert [11].

2.3.5 Node.js

Node.js wurde ursprünglich von Ryan Dahl im Jahr 2009 geschrieben, als Open-Source-Plattform JavaScript- run-time environment, in der JavaScript-Code außerhalb eines Browsers ausgeführt wird.



Abbildung 2.5: NodeJS Logo

Quelle: <https://nodejs.org/en/about/resources/>

Die Kernfunktionalität von Node.js befindet sich in einer JavaScript-Bibliothek. Die in C ++ geschriebenen Node.js-Bindungen verbinden diese Technologien miteinander und mit dem Betriebssystem.

Unten den nativen Modulen von Node.js findet man das http Modul, das die Entwicklung eines HTTP-Servers ermöglicht. Es ist daher möglich, mit Web-Servern wie Apache oder Nginx beim Bereitstellen von Sites und Webanwendung, die mit NodeJS entwickelt wurden.

Die Architektur vom von JavaScript ermöglicht eine gleichzeitige Verwendung von beispielsweise client-verbinding oder Datenbank zugriffen.

Dies hat als Vorteil die Geschwindigkeitsleistung gleichzeitiger die Aufrechterhaltung der Effizienz. Mit einem solchen Tool kann man -beispielsweise- E-Mails zehn Mal schneller als PHP senden, mit weniger erfolglosen Operationen. Ein anderer Vorteil dafür ist den Anruf von Bibliotheken in der gewünschten Zeit, was zu einer optimalen Nutzung der Ressourcen führt.

Das node.js-Paket- npm ist das größte Ökosystem von Open-Source-Bibliotheken der Welt.

Node.js verwendet ein ereignisgesteuertes, nicht blockierendes I / O Modell, das es leicht und effizient macht. I/O bezieht sich auf Eingabe / Ausgabe. Dies kann vom Lesen / Schreiben lokaler Dateien bis zum Anlegen einer HTTP-Anforderung an eine API reichen.

Das I/O hat als Nachteil das Benötigen von viele Zeit und es blockiert somit andere Funktionen [12].

2.3.6 Rest API Server mit C sharp Programmierungssprache

2.3.6.1 C sharp

Dies hat als Symbole C# aber wird C sharp ausgesprochen, ist eine objektorientierte Programmiersprache, die seit 2002 von Microsoft vermarktet wird und auf dem Microsoft .NET-Plattform entwickelt werden soll.

C# abgeleitet von C und C ++ und wird zur Entwicklung von Webanwendungen sowie von Desktopanwendungen, Webdiensten, Befehlen, Widgets oder Klassenbibliotheken verwendet.

Wie früher erwähnt, C # soll auf der .NET-Plattform entwickelt werden. Das Hauptkern dieser Technologie ist das .NET-Framework und besteht aus:

- Die ASP.NET- und Winforms-Umgebungen, in denen Webanwendungen für die .NET-Plattform ausgeführt werden
- Eine Klassenbibliothek, die es ermöglicht:
 - Dateien bearbeiten
 - Tabellen oder Baumstrukturen manipulieren
 - Zugang zum Internet
 - Grafische Schnittstellen erstellen
 - Zugriff auf Datenbank
- Die Common Language Runtime (CLR) ist die Laufzeit, die von den .NET-Plattformsprachen (C #, Visual Basic .NET, J # usw.) verwendet wird. Die von der CLR bereitgestellten Dienste, Starten und Ausführen von Programmen, Speicherwiederherstellung und Ausnahmebehandlung [13].

Was ist .NET? .NET ist ein Software-Framework, das von Microsoft entworfen und entwickelt wurde. Die erste Version von .Net Framework war 1.0 aus dem Jahr 2002. Es handelt sich um eine virtuelle Maschine zum Kompilieren und Ausführen von Programmen, die in verschiedenen Sprachen wie C #, VB.Net usw. geschrieben sind.

.NET Framework unterstützt mehr als 60 Programmiersprachen, in denen elf Programmiersprachen von Microsoft entworfen und entwickelt werden [14].

2.3.7 Web Application Architecture

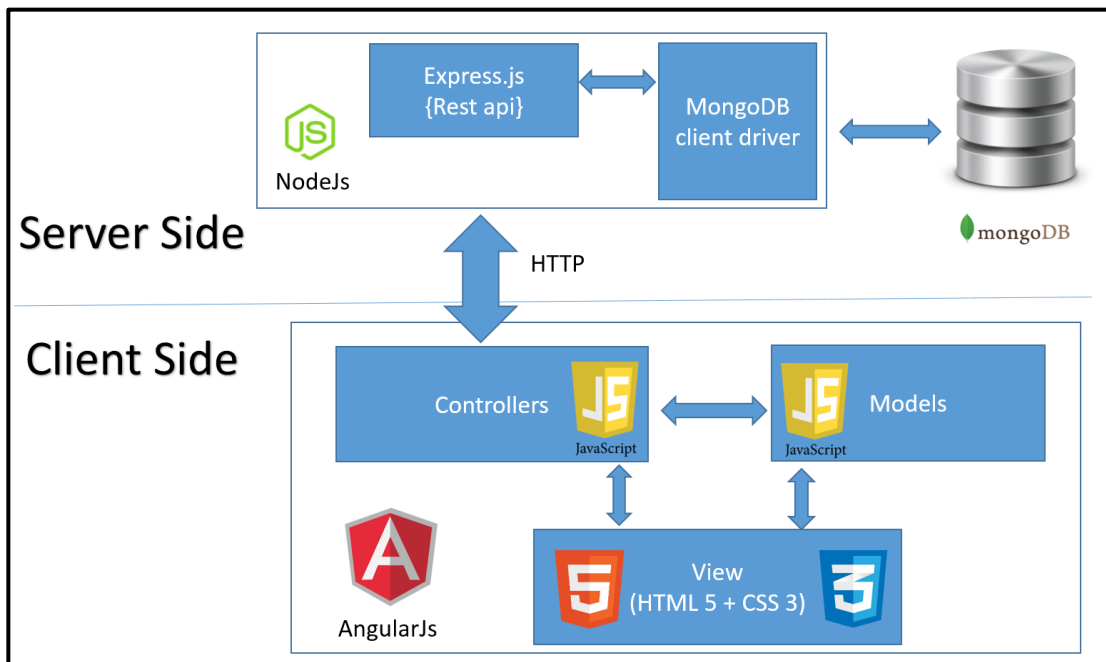


Abbildung 2.6: Web-Anwendung-Architektur

Quelle : <https://thingoftheday.blogspot.com/2016/01/crud-application-using-angularjs-nodejs.html>

Die Abbildung 2.6 erklärt wie man eine Webanwendung implementieren kann, mit Hilfe einer Rest-API, Datenbank und Angular.

Wie das Bild zeigt, besteht eine Webanwendung aus zwei Teilen, Server-Seite und Client-Seite. Die beiden sind über einem HTTP-Modul verbunden.

Der Rest-API-Sever kann mit NodeJS oder mit C sharp entwickelt werden.

Ein HTTP-Modul ist eine Baugruppe, die bei jeder Anforderung aufgerufen wird, die an eine Anwendung gesendet wird. Sie werden als Teil der ASP.NET-Anforderungspipeline aufgerufen und haben während der gesamten Anforderung einen Zugriff auf Lebenszykluseignisse. Mit solchen Modulen kann man eingehende und ausgehende Anforderungen prüfen und je nach Anforderung Maßnahmen ergreifen.

2.4 Datenerhalten

Die Daten, die von den GPS Trackers gewonnen wurden, sind auf einer Microsoft SQL Datenbank gespeichert.

2.4.1 Microsoft SQL Server

2.4.1.1 Definition der MS SQL Server

Microsoft SQL Server ist ein relationales Datenbankverwaltungssystem, das von Microsoft entwickelt wurde. Als Datenbankserver handelt es sich um ein Softwareprodukt mit der primären Funktion zum Speichern und Abrufen von Daten, die von anderen Softwareanwendungen angefordert werden. Diese können entweder auf demselben Computer oder auf einem anderen Computer in einem Netzwerk (einschließlich Internet) ausgeführt werden [15].

Was ist ein relationales Datenbankverwaltungssystem? Ein relationales Datenbankverwaltungssystem (RDBMS) ist ein Datenbankverwaltungssystem (DBMS), das auf dem relationalen Datenmodell basiert [16].

Relationales Datenmodell? Das relationale Modell (RM) für das Datenbankmanagement ist ein Ansatz zum Verwalten von Daten mithilfe einer Struktur und Sprache, die der Prädikatslogik erster Ordnung entspricht, wobei alle Daten in Form von Tupeln dargestellt werden, die in Beziehungen gruppiert sind. Eine Datenbank, die im Hinblick auf das relationale Modell organisiert ist, ist eine relationale Datenbank [17].

2.4.1.2 Architektur der Microsoft SQL Server

Ein MS SQL Server besteht aus folgenden Teilen:

- Client: Wo die Anfrage initiiert wurde.
- Abfrage: SQL Abfrage, die Hochsprache ist.
- Logical Units: Schlüsselwörter, Ausdrücke und Operatoren usw.
- N / W-Pakete: Netzwerkbezogener Code.
- Protokolle: In SQL Server gibt es 4 Protokolle.
 - Shared Memory: für lokale Verbindungen und zur Fehlerbehebung
 - Named pipes: für Verbindungen, die sich in einer LAN-Verbindung befinden.
 - TCP / IP: für Verbindungen, die sich in WAN-Konnektivität befinden.

- VIA-Virtual Interface Adapter: erfordert vom Hersteller hergestellte spezielle Hardware und wird auch von der SQL 2012-Version nicht mehr unterstützt.
- Server: Wo SQL Services installiert wurden und sich Datenbanken befinden.
- Relational Engine: Hier wird die eigentliche Ausführung ausgeführt. Es enthält Abfrage Parser, Abfrage Optimizer und Abfrage Executor.
 - Abfrage Parser: Dies überprüft die Syntax der Abfrage und konvertiert die Abfrage in Maschinsprache.
 - Abfrage Optimizer: Der Ausführungsplan wird als Ausgabe vorbereitet, indem Abfrage, Statistiken und Algerier-Baum als Eingabe verwendet werden.
 - Execution Plan: Es ist wie eine Roadmap, die die Reihenfolge aller Schritte enthält, die im Rahmen der Abfrageausführung ausgeführt werden sollen.
 - Abfrage Executor: Hier wird die Abfrage mit Hilfe des Ausführungsplans Schritt für Schritt ausgeführt, und auch die Speicher-Engine wird kontaktiert.
- Storage Engine: ist für das Speichern und Abrufen von Daten im Speichersystem (Festplatte, SAN usw.), Datenmanipulation, Sperren und Verwalten von Transaktionen verantwortlich.
- SQL OS: Dies liegt zwischen der Hostmaschine (Windows-Betriebssystem) und SQL Server. Alle auf der Datenbank-Engine ausgeführten Aktivitäten werden von SQL OS übernommen. SQL OS stellt verschiedene Betriebssystemdienste bereit, z. B. Speicherverwaltungsabkommen mit Pufferpool, Protokollpuffer und Deadlock-Erkennung unter Verwendung der Blockierungs- und Sperrstruktur [18].

2.4.1.3 SQL Abfrage

SQL ist eine domänenspezifische Sprache, die bei der Programmierung verwendet wird und zum Verwalten von Daten in einem relationalen Datenbankverwaltungssystem entwickelt wurde.

Es ist eine Sprache, die ausschließlich für den Zugriff auf Datenbanken entwickelt wurde, und ist die beliebteste von ihnen.

Mit SQL kann man Tabellen erstellen, Daten ändern, auf Daten zurückgreifen, an denen wir interessiert sind, z. B. wenn man herausfinden würde, welche neue Tracker in der letzten Woche beigetreten ist oder welche Tracker ein bestimmtes Abzeichen hat.

2.4.1.3.1 Themen in SQL

SQL Abfrage Typen [19]:

- ✓ SELECT: Abruf von Datensätze aus einer Tabelle
- ✓ SELECT TOP: Abruf von Datensätze aus einer Tabelle und Begrenzung die Ergebnisse
- ✓ INSERT: Einfügen von Datensätze in eine Tabelle
- ✓ UPDATE: Aktualisierung von Datensätze in einer Tabelle
- ✓ DELETE: Löschen von Datensätze aus einer Tabelle
- ✓ TRUNCATE TABLE: Löschen Alle Datensätze aus einer Tabelle (kein Rollback)
- ✓ Usw.

SQL Joins:

- ✓ JOIN-Tabellen: Innere und äußere Verbindungen

SQL-Klauseln:

- ✓ FROM: Auflisten von Tabellen und Verknüpfungsinformationen
- ✓ WHERE: Ergebnisse filtern
- ✓ GROUP BY: Gruppieren eine oder mehrere Spalten

SQL Funktion:

- ✓ COUNT: Zurückgabe der Anzahl eines Ausdrucks
- ✓ MIN: Zurückgabe der min eines Ausdrucks
- ✓ MAX: Zurückgabe des Maximums eines Ausdrucks zurück
- ✓ AVG: Zurückgabe den Durchschnitt eines Ausdrucks zurück

SQL Bedingung:

- ✓ AND: 2 oder mehr zu erfüllenden Bedingungen
- ✓ IN: Alternative zu mehreren ODER-Bedingungen
- ✓ NOT: Negiert eine Bedingung
- ✓ IS NULL: für NULL-Wert
- ✓ IS NOT NULL: Test auf NOT NULL-Wert

SQL-Tabellen und -Ansichten:

- ✓ CREATE TABLE: Eine Tabelle Erstellen

Ich habe nur ein paar SQL Typen, Funktionen Bedingungen, Klauseln erklärt, die ich in meiner Arbeit genutzt habe

2.5 Zusammenfassung

In diesem Kapitel habe ich die genutzte Software, Hardware und Methoden theoretisch erklärt, sodass die Grundlagen für die Realisierung gelegt sind.

Im nächsten Kapitel werde ich die Funktionalität der Anwendung mit Hilfe von Diagrammen erläutern.

3 Projektentwurf

Dieses Kapitel befasst sich mit der Entwicklung des Designs des Projekts. Das umfasst die Funktion des Systems, und wie es aus Sicht der Benutzer reagieren sollte. Die Vorgehensweise, die ich in meinem Design verfolgen werde, wird durch das folgende Flussdiagramm angegeben:

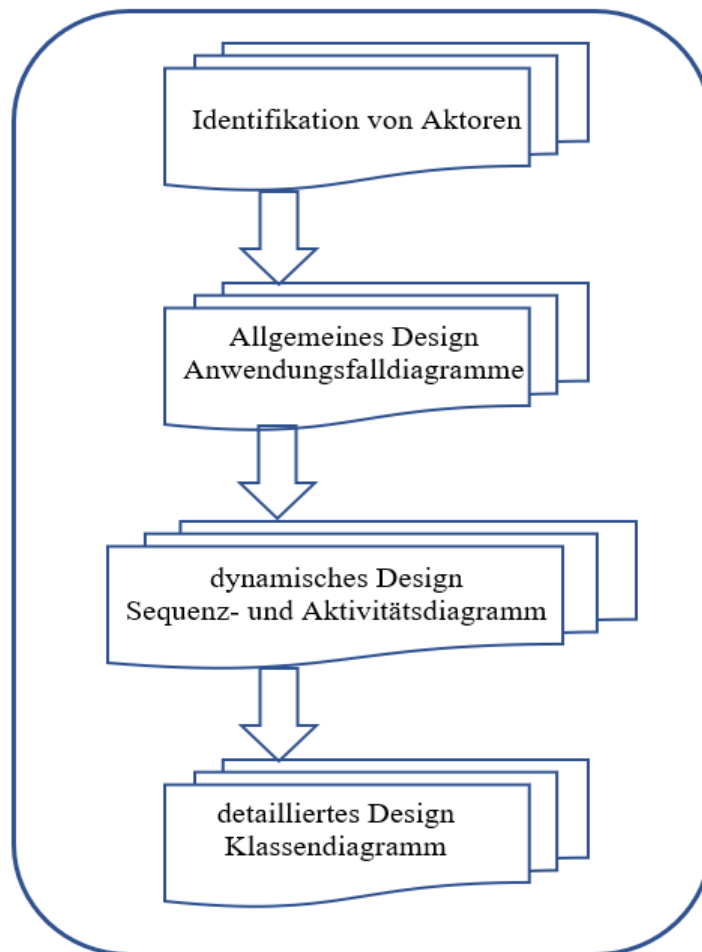


Abbildung 3.1: Flussdiagramm des Designs des Projektes

3.1 Identifikation von Akteuren

Die Hauptakteure sind die Benutzer des Systems; Ein Akteur repräsentiert somit eine kohärente Rolle gegenüber einem System. Für meine Projekt existiert nur einzelne Akteur, der Benutzer meiner Anwendung.

3.2 Entwicklung von Anwendungsfalldiagrammen

Ein Anwendungsfalldiagramm ist ein Akteurs Diagramm, das die Bedürfnisse der Benutzer in Bezug auf das System darstellen soll. Tatsächlich gibt jeder Anwendungsfall eine Aktionssequenz, einschließlich Varianten, an, die die Entität mit den Akteuren der Entität realisiert. Im Folgenden präsentiere ich die Anwendungsfalldiagramme meiner Anwendung.

3.2.1.1 Vorläufiges Anwendungsfalldiagramm

Dieses Diagramm soll die Hauptaufgaben meiner Anwendung umsetzen.

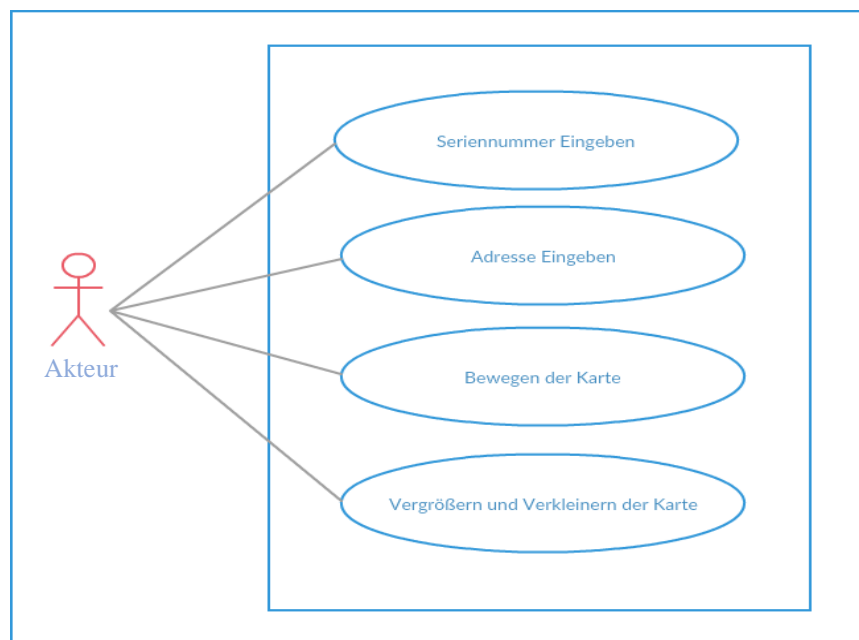


Abbildung 3.2: Vorläufiges Anwendungsfalldiagramm

Wie das Diagramm zeigt, kann einer Akteur verschiedenen Aktionen ausführen.

- ✓ Seriennummer eingeben: der Akteur kann einen Seriennummer von einem Modul eingeben, um seine Performance der Signalstärke auf der Karte zu zeigen.
- ✓ Adresse eingeben: der Akteur kann auch eine Adresse auf der Karte suchen
- ✓ Bewegen, Vergrößern und Verkleinern der Karte: der Akteur kann die Karte navigieren um verschieden Ergebnisse zu sehen.

3.2.1.2 Detailliertes Anwendungsfalldiagramm

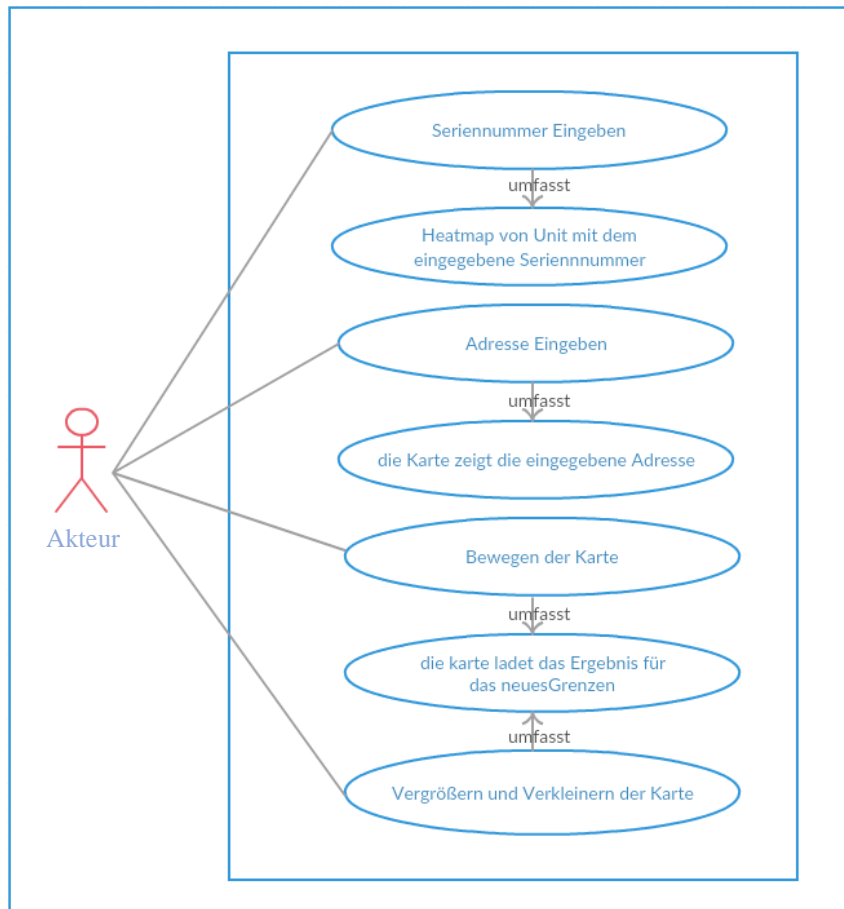


Abbildung 3.3: Detailliertes Anwendungsfalldiagramm

Beschreibung:

- Seriennummer Eingeben: die Karte kann verschiedene Resultate zeigen. Das Ergebnis aller Modulen oder von einem Modul nach der Seriennummer.
- Adresse Eingeben: diese Bedingung hilft den Benutzer, einen Ort zu finden und die Performance der Signalstärke an diesem Ort zu sehen, wenn die Karte mit Daten geladen ist.
- Bewegen, Verkleinern und Vergrößern der Karte: da die Daten zu groß sind, kann die Karte nicht alle Ergebnisse gleichzeitig zeigen, und zeigt deswegen nur das Resultat innerhalb des sichtbaren Kartenausschnitt. Nach jeder Navigation wird die Karte mit anderem Resultat neugeladen.

3.3 Dynamische Diagramme

Dies ist eine detaillierte Erklärung eines Anwendungsfalldiagramms, das sich auf die chronologische Reihenfolge von Nachrichten konzentriert, die von einer Gruppe von Objekten gesendet und empfangen werden.

3.3.1 Sequenzdiagramme

In Bezug auf meine Anwendung ist das allgemein Sequenzdiagramm wie folgt:

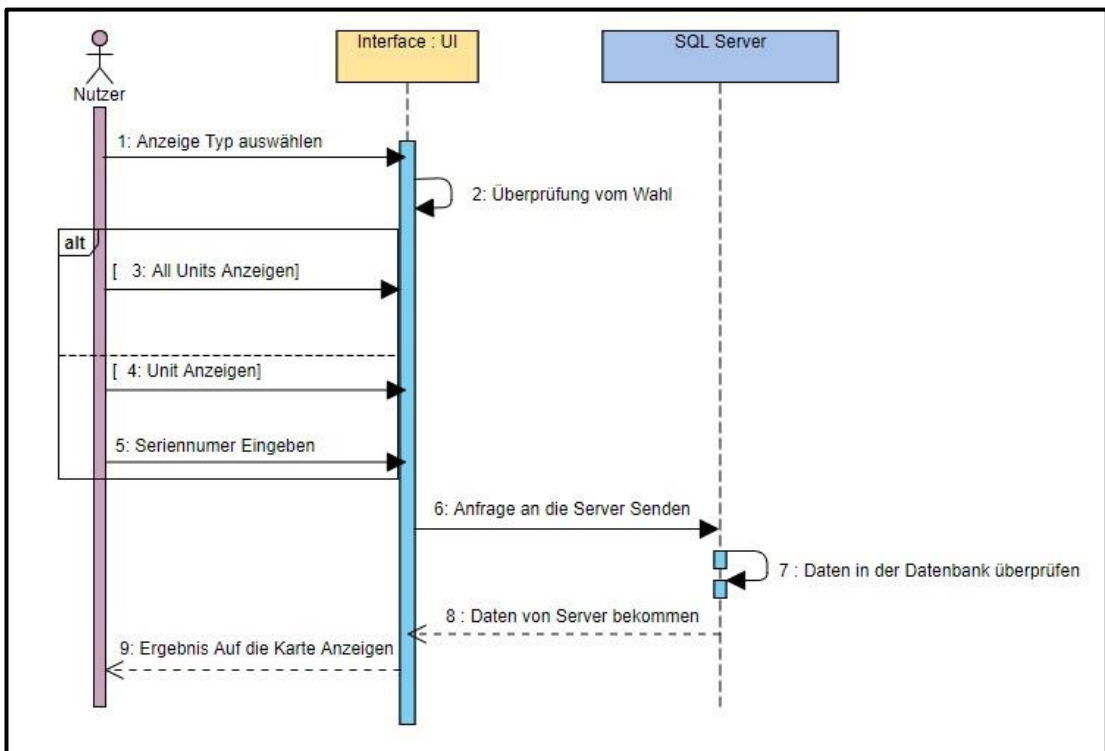


Abbildung 3.4: Allgemeines Sequenzdiagramm

Beschreibung:

In diesem Diagramm sind drei Teilen erkenntlich: Nutzer, Interface, SQL Server

Der Ablauf im Sequenzdiagramm fängt beim Nutzer an und geht über Interface zum SQL Teil und wieder zurück.

Das Menü in der Anwendung besitzt zwei verschiedene Option und eine von beiden muss gewählt werden. Im Diagramm ist das als „alt“-Block abgebildet.

Die zwei Optionen sind:

- Wenn **All Units** gewählt ist, wird eine Anfrage zu dem Server gesendet um das Ergebnis für alle Module auf der Karte anzuzeigen.
- Wenn **nur eine Unit** gewählt ist, dann muss zusätzlich eine Seriennummer eingegeben werden, um eine Anfrage mit dieser Seriennummer zum Server zu schicken.

Der SQL Server führt dann die Anfragen und sendet eine Antwort mit den gefundenen Daten zum User Interface. Diese Daten werden nun auf der Karte angezeigt.

3.3.2 Sequenzdiagramm der All Units Option

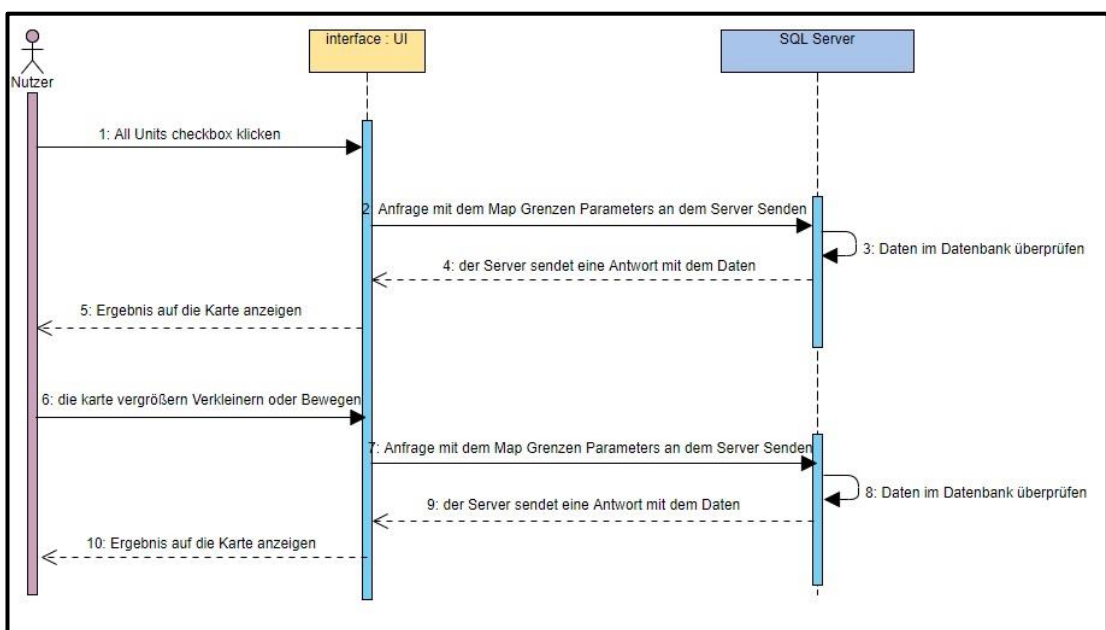


Abbildung 3.5: Sequenzdiagramm der All Units Option

Beschreibung:

Wie beim ersten Sequenzdiagramm erwähnt, hat die Anwendung zwei Optionen, eine für All Units und die anderen für eine Unit. Die Abbildung 3.5 erklärt den Ablauf der ersten Option.

Ich möchte in dieser kurzen Beschreibung die Beschränkung der Daten auf den sichtbaren Bereich der Karte erklären. Der Kartenausschnitt hat immer vier begrenzende Ecken und diese ändern sich mit jeglicher Bewegung oder Zoom der Karte. Da die Daten zu groß sind, schicke ich die Werte diese vier Ecken als

Parameters zum Server. So werden nur Positionen innerhalb dieser Grenzen ausgewertet.

Mit jeder Bewegung, Vergrößern oder Verkleinern der Karte ändern sich die Parameter und dieselbe Prozedur wird wiederholt.

3.3.3 Sequenzdiagramm der Unit Option

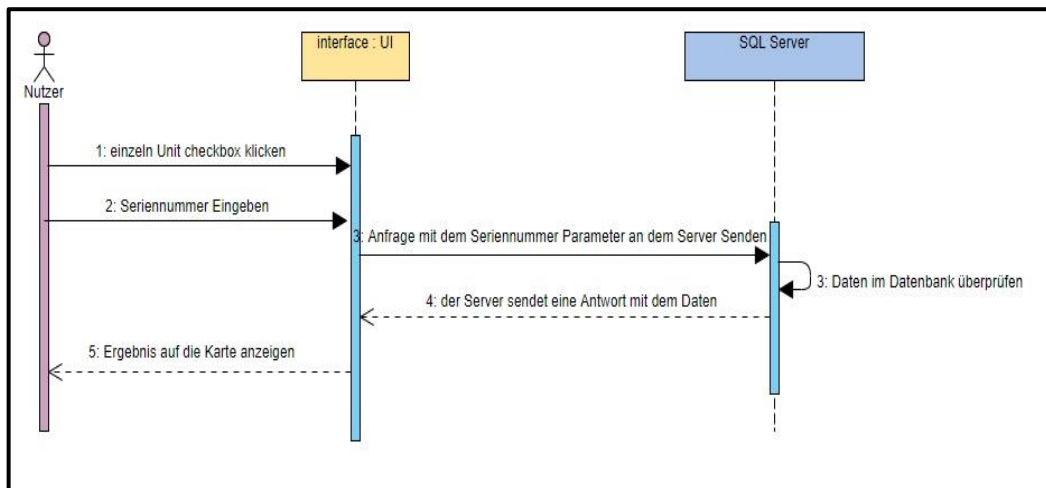


Abbildung 3.6: Sequenzdiagramm der einzelnen Unit Option

Beschreibung:

In der Abbildung 3.6 ist das Sequenzdiagramm der zweiten Option zu sehen. Hier kann ein einzelnes Modul ausgewählt werden, um die individuelle Performance zu sehen.

Jedes Modul hat eine eindeutige Seriennummer und diese muss als Parameter in der Anfrage zum Server gesendet werden, um die Daten und Positionen dieses Modules auf dem Server auszuwerten.

3.3.4 Aktivitätsdiagramme

Dieses Diagramm wird verwendet, um ein internes Verhalten eines Anwendungsfalls oder einer Unterstützungsoperation zu modellieren. Das habe ich in Form einer Folge von Aktionen realisiert.

3.3.4.1 All Unit Aktivitätsdiagramm

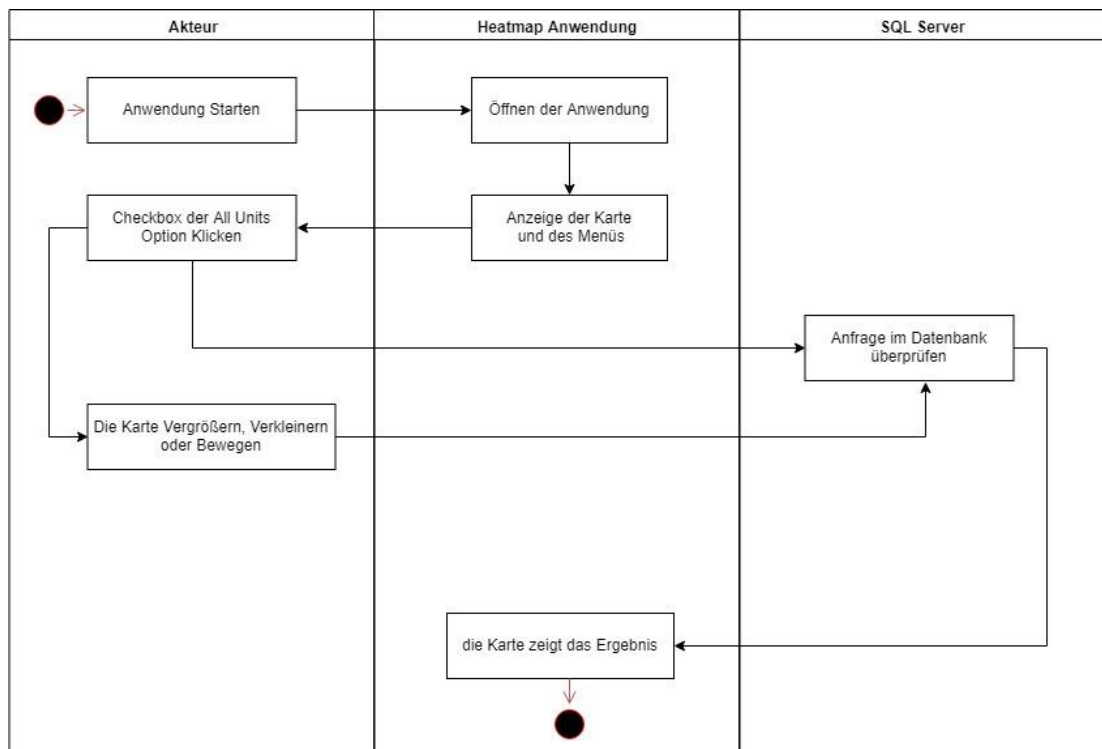


Abbildung 3.7: Aktivitätsdiagramm von All Units Option

Beschreibung:

Diese Aktivitätsdiagramm ist auch auf 3 Teilen geteilt:

- ✓ Was der Akteur in der Anwendung machen kann
- ✓ Was bei der Anwendung passiert werden kann
- ✓ Was im SQL Server passiert

Prozedur:

- Der Akteur macht die Anwendung auf
- Eine Karte und ein Menü werden angezeigt
- Der Akteur klickt auf die erste Checkbox der ersten Option All Units
- Die Anfrage wird im Server überprüft
- Das Ergebnis Auf die Karte anzeigen
- Dann für jede Aktion auf die Karte, andere Ergebnis wird gezeigt abhängig mit dem Grenzen der karte

3.3.4.2 Einzelne Unit Aktivitätsdiagramm

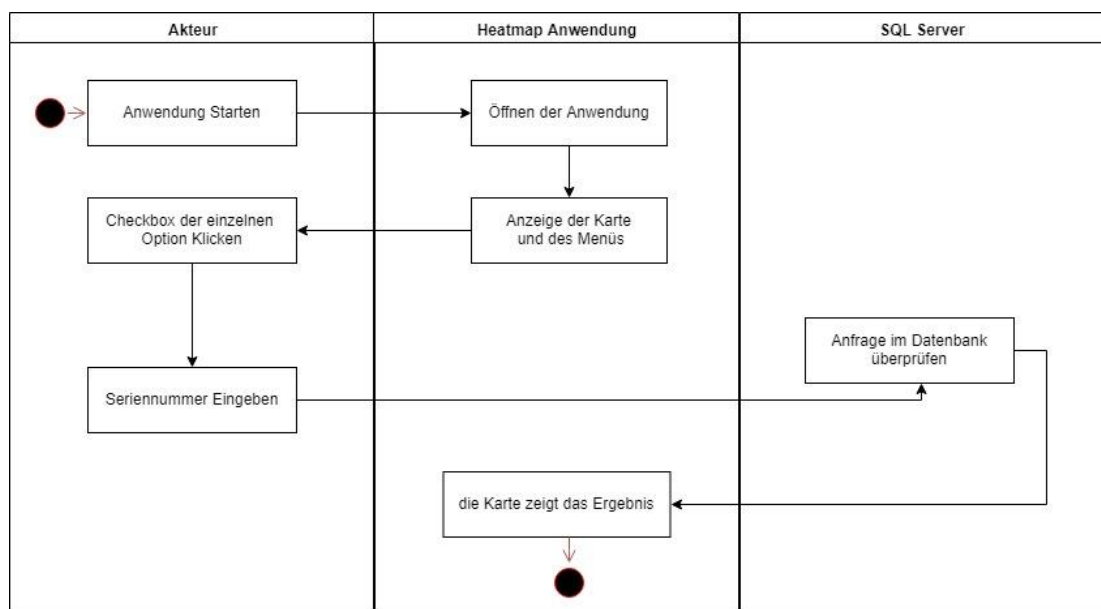


Abbildung 3.8: Einzelne Unit Aktivitätsdiagramm

Prozedur:

Dies zeigt wie die Prozedur der zweiten Option Einzelne Unit läuft.

- Der Akteur macht die Anwendung auf
- Eine Karte und ein Menü werden angezeigt
- Der Akteur klickt auf die Checkbox der Option einzelne Unit
- Der Akteur schreibe einen Seriennummer
- Das Ergebnis Auf die Karte anzeigen
- Für jeden neuen Seriennummer wird, die wiederholt

3.4 Klassendiagramm

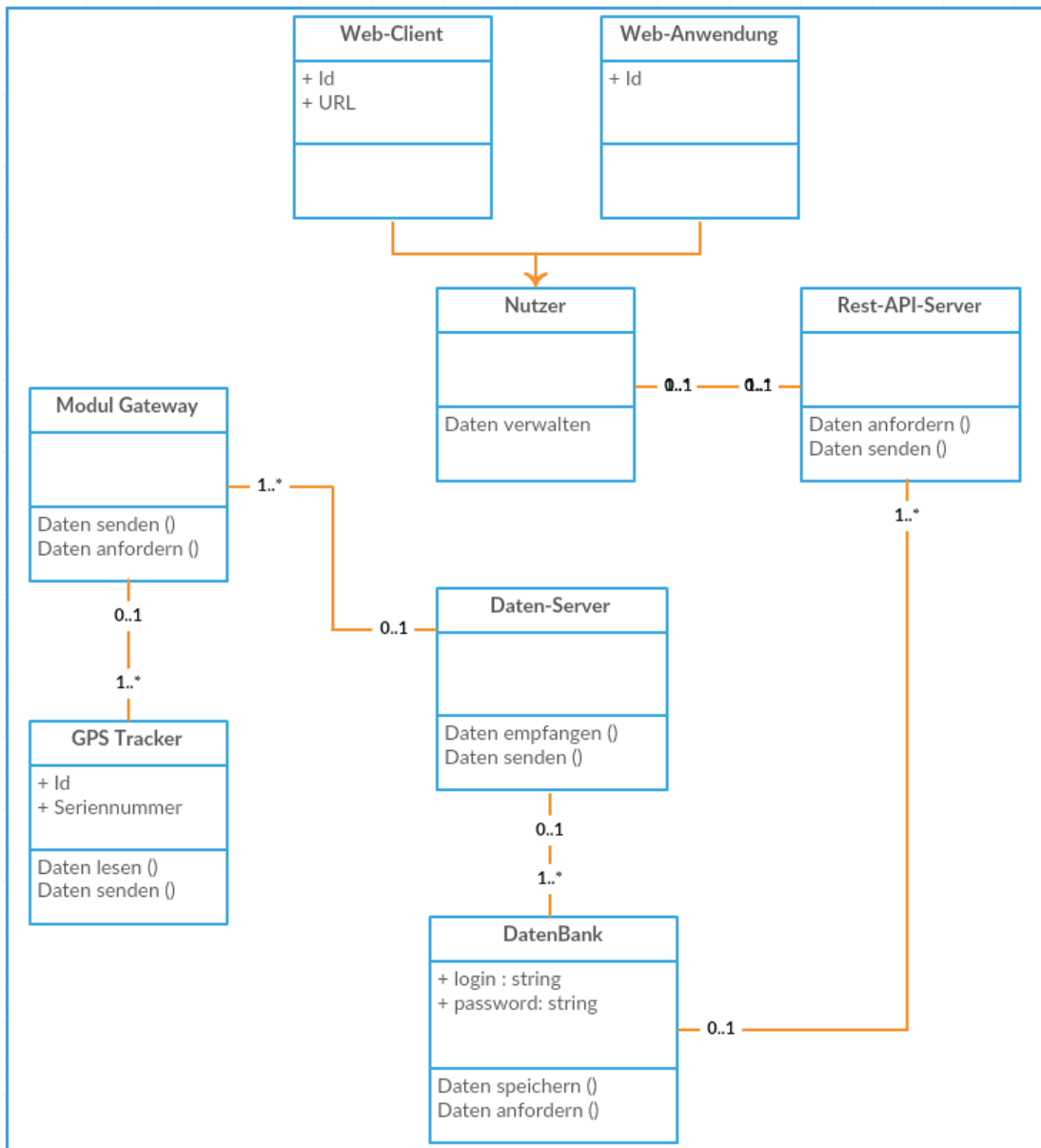


Abbildung 3.9: Klassendiagramm

Diese Abbildung zeigt die gesamte Betriebszyklus. Fängt mit dem Lesen von Daten im Modul an bis die Verwendung dieser Daten auf die Web-Anwendung

3.5 Zusammenfassung

In diesem Kapitel habe ich das Design meiner Anwendung entworfen und detailliert die Anwendungsfalldiagramme, Sequenzdiagramme, Aktivitätsdiagramme und Klassendiagramm beschrieben. Daher kann ich im nächsten Kapitel mit der Realisierung beginnen.

Das nächste Kapitel wird meine Anwendung und die vorherigen Diagramme detailliert erklären.

Teil 2

Entwicklung der Anwendung

4 Realisierung

Wie ich im theoretischen Teil geklärt habe, besteht eine Web-Anwendung aus zwei Teilen; Server-Seite und Client-Seite.

In diesem Kapitel beschreibe ich die gesamte Realisierung des Projekts von der Aufbereitung der Daten in der Datenbank auf der Server-Seite bis zur Anzeige der Daten auf der Karte auf der Client-Seite

4.1 Datenbank

Die Daten, die von den GPS Trackers gewonnen worden, sind der wichtigste Teil in meiner Arbeit.

Mein Projekt basierend auf diesen Daten, die in einer SQL Datenbank gespeichert sind. Leider befindet sich die Daten auf dem Server nicht in einer, für dieses Projekt, brauchbaren Form und müssen erst aufbereitet werden.

4.1.1 Die Daten

Ich möchte in diesem Schritt einen Überblick über die Daten geben, bevor ich über ihre Verarbeitung schreibe.

Unsere GPS Trackers können mit allem verfügbaren GSM Netze verbinden, und kann damit die Signalstärke von jedem Netz erkennen und zum Server übertragen.

Dabei überträgt auch seine GPS Position als Breiten- und Längengrad.

Jeder GPS Tracker hat eine Seriennummer, ID und Name in der Datenbank.

Eine Zusammenfassung über diese Informationen:

- Breitengrad: In der Geographie ist der Breitengrad eine geographische Koordinate, die die Nord-Süd-Position eines Punktes auf der Erdoberfläche angibt. Latitude ist ein Winkel (unten definiert), der von 0° am Äquator bis zu 90° (Nord oder Süd) an den Polen reicht.

- Längengrad: Längengrad ist das Maß östlich oder westlich des Nullmeridians. Der Längengrad wird durch imaginäre Linien gemessen, die vertikal (auf und ab) um die Erde verlaufen und sich am Nord- und Südpol treffen. Diese Linien werden als Meridiane bezeichnet. Jeder Meridian misst einen Längengrad.
- Seriennummer: eindeutige Identifikationsnummer für jedes Modul.
- GSM Signalstärke: Die Signalstärke wird nur durch einen Wert definiert: RSSI: Received Signal Strength Indicator; RSSI ist ein negativer Wert und je näher an 0, desto stärker das Signal.
 - ≥ -70 dBm: Sehr gut: Starkes Signal bei maximalen Datengeschwindigkeiten.
 - -70 dBm bis -85 dBm: gut: Starkes Signal bei guten Datengeschwindigkeiten
 - -86 dBm bis -100 dBm: angemessen: angemessen aber nützliche, schnelle und zuverlässige Datengeschwindigkeiten erreicht werden, aber geringfügige Daten mit Aussetzern sind möglich.
 - < -100 dBm: Schlecht: Die Leistung wird drastisch sinken
 - -110 dBm: Kein Signal: Verbindung trennen [20].

Da die Qualität der Signalstärke ist für uns Interessant und sehr wichtig in der Verarbeitung der Daten, muss ich die verschiedenen Faktoren, die die Signalstärke und -qualität beeinflussen, erläutern.

- ✓ Turm last
- ✓ Nähe zum Mobilfunkmast
- ✓ Signal durch einen zellularen Repeater
- ✓ Konkurrierende Signale
- ✓ Körperliche Barrieren (Berge, Gebäude, Züge usw.)
- ✓ Wetter
- ID und Name: auf den Server steht ein Name und ID für jeden GPS Tracker: dies ist sehr wichtig so kann ich die Tabellen zusammen verbinden, weil die Informationen in verschiedenen Tabellen verteilt sind.

Am nächsten Schritt beschreibe ich, wie die Informationen genutzt werden.

4.1.2 Vorbereitung der Daten

Die obengenannte Information ist auf dem SQL Server gespeichert. Deswegen muss ich erst die Software „Microsoft SQL Server Management Studio“ auf dem Rechner installieren, um diese Daten zu erreichen.



Abbildung 4.1:
SQL Server Logo

Nach der Installation kann man jetzt mit dem Server kommunizieren. (Abb. 4.2)

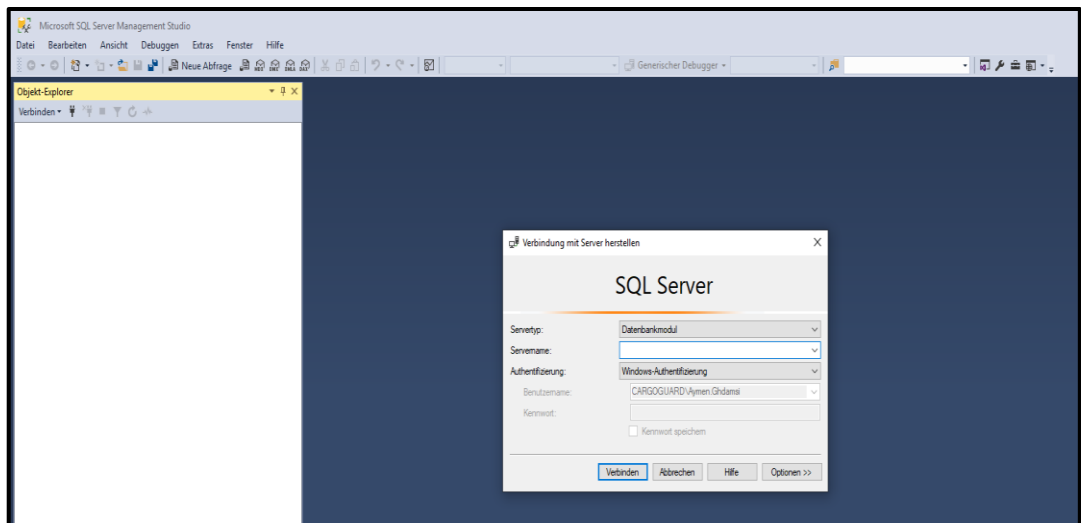


Abbildung 4.2: Verbindung zum SQL Server

Wie oben im Bild dargestellt, kann eine Verbindung zum Server über die Eingabe des Server Name, Benutzername und hergestellt werden.

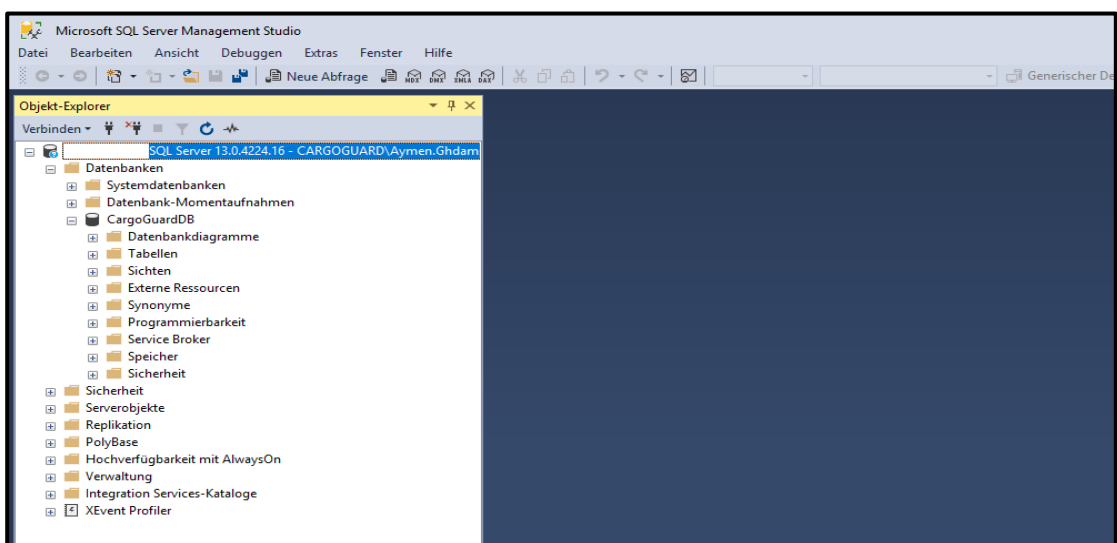


Abbildung 4.3: SQL Datenbank

Nach der Verbindung kommen wir zu unserer Datenbank (Abb. 4.3), wo die benötigte Tabellen in CargoGuardDB dargestellt werden.

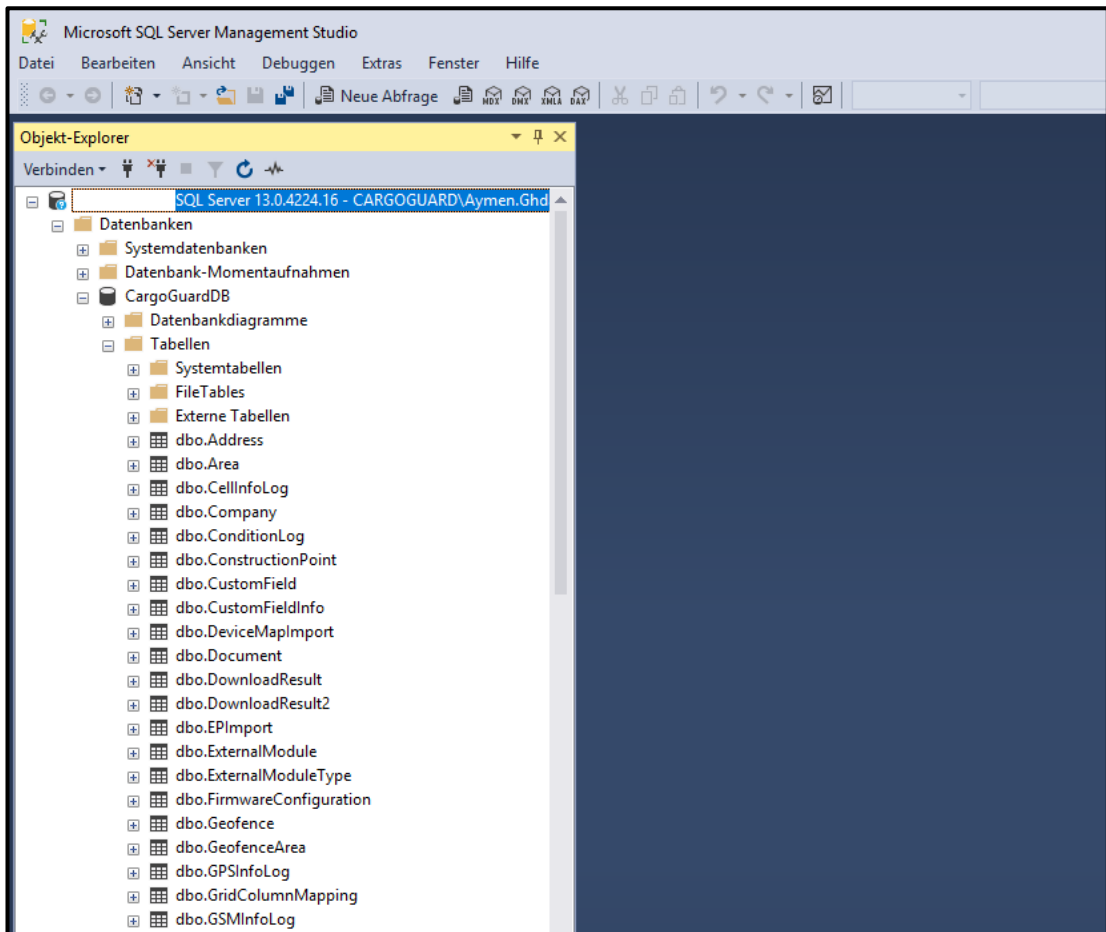


Abbildung 4.4: Die Tabellen in der Datenbank

Die Abbildung 4.4 zeigt die Tabellen in unserer Datenbank. Die benötigten Daten für meine Anwendung sind in verschiedenen Tabellen verteilt.

Ich kann die Daten in dieser Tabelle mit SQL Abfragen anschauen. SQL Abfrage habe ich in dem theoretischen Teil erklärt, welchen Funktionen, Methoden usw. existieren, die man in verschiedenen Fall nutzen kann.

```

SELECT TOP (100) p.Latitude, p.Longitude, cells.CellId, cells.SignalStrength
FROM Position p
LEFT JOIN CellInfoLog cells ON p.GSMInfoId = cells.GSMInfoLogId
WHERE p.PositionSource = 'GPS'
AND cells.Id IS NOT NULL
AND CellId != '0'

```

Listing 6: Erste SQL Abfrage

Beschreibung des Obengenannten Codes:

Meine benötigte Daten sind in zwei Tabelle verteilt;

- Position: wo Latitude, Longitude, GSMInfos Id und PositionSource als Spalten gespeichert werden
- CellInfosLog: wo Id, CellId, SignalStrength und GSMInfosLogId dort auch als Spalten gespeichert werden.

Wie funktioniert der Code? Ich wähle Latitude und Longitude aus der Position Tabelle und CellId und SignalStrength aus der CellInfoLog Tabelle und verbinde die beiden zusammen, wenn die GSMInfosId von der Position Tabelle gleich der GSMInfosLogId in der CellInfosLog Tabelle ist.

Dabei werden nur Resultaten gezeigt, bei denen, die PositionSource = “GPS“ und Id und CellId in der CellInfoLog Tabelle nicht null sind.

TOP(100) bedeutet, dass nur die ersten 100 Zeilen gezeigt werden, so kann ich sehen und überprüfen, ob der Befehl richtig geschrieben ist.

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the 'Objekt-Explorer' showing a server connection to '10.174.20.246 (SQL Server 13.0.4224.16 - CARGOGUARD\Aymen.Ghdam)'. The main window displays the SQL query from Listing 6. Below the query, the 'Ergebnisse' tab is active, showing a table with 10 rows and 4 columns: Latitude, Longitude, CellId, and SignalStrength.

	Latitude	Longitude	CellId	SignalStrength
1	53.5251998901367	10.0978002548218	36654	-95
2	53.5251998901367	10.0978002548218	11864	-82
3	53.5251998901367	10.0978002548218	5740	-96
4	53.5251998901367	10.0978002548218	36608	-93
5	53.5256004333496	10.09850025177	36654	-96
6	53.5256004333496	10.09850025177	5740	-96
7	53.5256004333496	10.09850025177	36607	-81
8	53.5256004333496	10.09850025177	36608	-94
9	48.1866989135742	11.6070003509521	31641	-85
10	48.1866989135742	11.6070003509521	10433	-66

Abbildung 4.5: Das Ergebnis der SQL Abfrage

Die erhaltenen Daten habe ich in eine neue Tabelle eingefügt, um diese auf der Client-Seite zu nutzen. Aber mit den Daten haben wir leider ein richtiges Problem.

Die über 26 Millionen Datenzeilen waren nicht zugeordnet und enthielten viele falsche Werte.

Während dem Erstellen der GSM-Abdeckung haben wir diese Fehler entdeckt. Zu ein Tracking bekamen wir immer verschiedene Signalstärkenwerte. Zum anderen gab es Positionen mit schlechter Signalstärke zwischen mehreren Positionen mit sehr guter Signalstärke. Das macht keinen Sinn, ein paar Bilder dafür werde ich in der Client-Seite Teil zeigen.

Aus diesem Grund habe ich entschieden, dass ich die Bedingung für die Abfrage ändere.

4.1.3 Verarbeitung der Daten

```
Select p.Id, p.Latitude, p.Longitude, cell.CellId, cell.SignalStrength, p.UnitId
from Position p
cross apply
  (SELECT CellId, SignalStrength
   FROM CellInfoLog
   WHERE CellId != 0 AND GsmInfoLogId = p.GSMInfoId
   Order BY SignalStrength DESC
  ) as cell
where p.PositionSource = 'GPS'
AND p.Latitude is not null
AND p.Longitude is not null
And cell.SignalStrength is not null
```

Listing 7: Die benutzte SQL Abfrage

Beschreibung der neuen SQL Abfrage:

Mit diesem Problem bekam ich eine neue Anforderung: Ich mache auch eine GSM Abdeckung für jedes Modul anhand seiner Seriennummer, deswegen erweitern wir die Tabelle mit der UnitId.

Mit dieser Variablen kann ich meine Tabelle mit anderen zusammen verbinden, um diese Seriennummer Werte zu kriegen.

Wie läuft die neuen SQL Abfrage? Ich nehme Latitude, Longitude and UnitId aus der Tabelle „Position“ dann CellId und SignalStrength aus der Tabelle „CellInfoLog“ wenn CellId nicht null ist und GSMInfoLogId gleich GSMInfoId aus der Tabelle

„Position“ ist. Am Ende wende ich den CROSS APPLY Operator an. Diese gibt nur die Zeilen des linken Tabellenausdrucks (in seiner endgültigen Ausgabe) „Position“ zurück, wenn er mit dem rechten Tabellenausdruck „CellInfoLog“ übereinstimmt.

Und dies ist möglich nur mit einer allgemeinen Bedingung, wenn PositionSource = „GPS“ und Latitude, Longitude, SignalStrength sind nicht null.

Die endgültigen Daten musste ich in einer neuen Tabelle sammeln.

4.1.3.1 Erstellung der Tabelle

Beim SQL Server kann man mit SQL Abfrage Tabellen erstellen und Daten drin einfügen.

```
CREATE TABLE [dbo].[GsmSignalStrength] (  
    [ID] [uniqueidentifier] NOT NULL,  
    [Latitude] [float] NOT NULL,  
    [Longitude] [float] NOT NULL,  
    [CellId] [int] NOT NULL,  
    [SignalStrength] [int] NOT NULL,  
    [UnitID] [uniqueidentifier] NULL  
    ) ON [PRIMARY]
```

Listing 8: Die SQL Abfrage zur Erstellung der Tabelle

Kurz Erklärung:

Eine Tabelle mit dem Name GSMSignalStrength wird gestellt und hat als Spalten (Abb. 4.6):

- ID: als eindeutige Kennung, die nicht null sein muss
- Latitude, Longitude als float: sind beide wichtig für die Position auf der Karte und dürfen auch nicht null sein.
- SignalStrength: Messwerte von Signalstärke und nicht null
- CellId: Kennzeichnung von jeder Zelle der Signalstärke

- UnitID eine eindeutige Kennung auch, kann null sein und mit denen kann ich zu anderen Tabellen verbinden.

Nach der Ausführung dieser Abfrage wird meine Tabelle mit den anderen vorhandenen Tabellen erscheint. (Abb. 4.6)

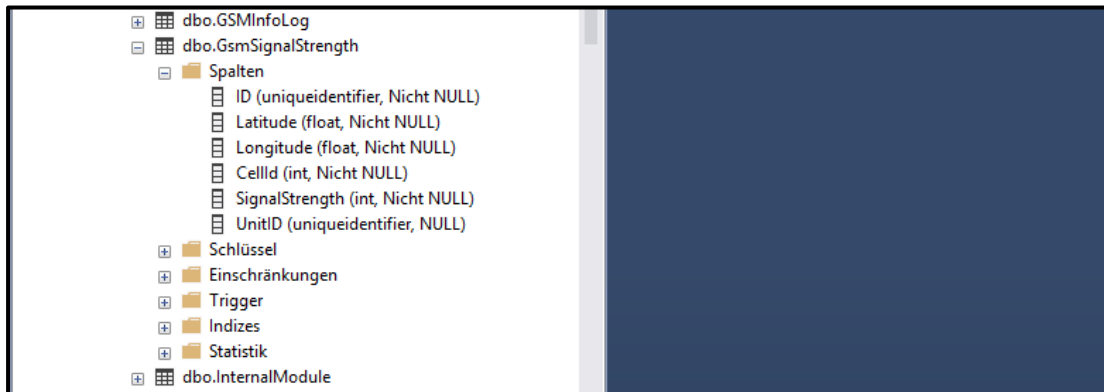


Abbildung 4.6: Meine Tabelle

Aber bis jetzt meine Tabelle ist noch leer und deshalb kommt den nächsten Schritt die Ausfüllung von Tabelle.

4.1.3.2 Ausfüllung der Tabelle

```
Insert into GsmSignalStrength(Id, Latitude, Longitude, CellId, SignalStrength, UnitId)
```

Listing 9: Insert SQL Abfrage

Mit diesem SQL Abfrage „Insert Into“ kann ich die Datensätze in meiner Tabelle „GsmSignalStrength“ für jeden Spalten einfügen.

Die Datensätze habe ich schon im vorherigen Schritt beschrieben.

```
Select p.Id, p.Latitude, p.Longitude, cell.CellId, cell.SignalStrength, p.UnitId
from Position p
cross apply
  (SELECT CellId, SignalStrength
   FROM CellInfoLog
   WHERE CellId != 0 AND GsmInfoLogId = p.GSMInfoId
   Order BY SignalStrength DESC
  ) as cell
where p.PositionSource = 'GPS'
AND p.Latitude is not null
AND p.Longitude is not null
And cell.SignalStrength is not null
```

Listing 10: Die SQL Abfrage der neuen Tabelle

4.1.3.3 Anzeige der Tabelle

Die Anzeige der Tabelle ist sehr einfach. Ich schreibe einen SQL Abfrage sowie:

```
SELECT TOP (10) CellId, Latitude, Longitude, SignalStrength, UnitId
from GsmSignalStrength
```

Listing 11: Die SQL Abfrage der Anzeige der Tabelle

Ich wähle nur zehn Zeilen mit CellId, Latitude, Longitude, SignalStrength und UnitId als Spalten aus meiner Tabelle GsmSignalStrength.

Da die Daten zu groß sind, habe ich nur zehn gewählt, um die Tabelle zu überblicken und das Ergebnis zu überprüfen. Aber in meinem Projekt habe ich alle verfügbaren Daten genutzt.

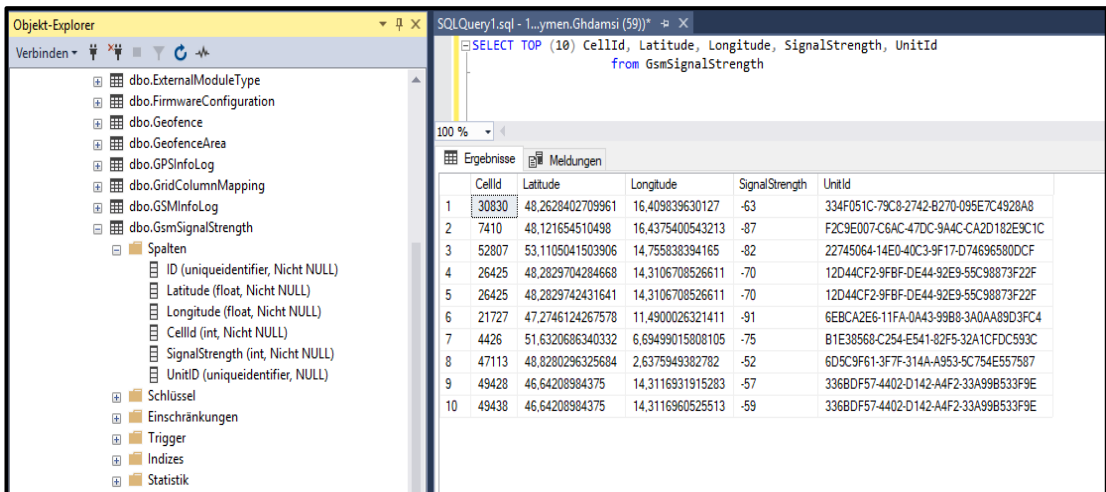


Abbildung 4.7: Die neue Tabelle

Abbildung 4.7 zeigt drei verschiedenen Teilen:

- Linke Seite: wo meine Tabelle mit Spalten steht
- Rechte Seite: wo man das SQL Abfrage schreibt
- Rechuntere Seite: wo das Ergebnis nach der Ausführung des SQL Abfrage gezeigt wird.

4.2 Verbindung des Rest API Servers- und Client Seite

Um die Daten, die in dem SQL Server gespeichert sind, auf den Client-Seite zu nutzen, brauche ich einen Rest-API-Server. Dies habe ich im theoretischen Teil ein bisschen erklärt aber ich beschreibe hier kurz seine Aufgabe in meinem Projekt.

Eine RESTful-API - auch als RESTful-Webdienst bezeichnet - basiert auf der REST-Technologie (Representational State Transfer), einem Architekturstil und einem Kommunikationsansatz, der häufig bei der Entwicklung von Webdiensten verwendet wird.

Eine API für eine Website ist eine Anwendungsprogrammchnittstelle, mit der zwei Softwareprogramme miteinander kommunizieren können und die HTTP-Anforderungen an GET-, PUT-, POST- und DELETE-Daten verwendet.

Das heißt, ich muss einen Rest Server programmieren, um die Daten zwischen die Datenbank und meine Client-Seite, die Angular ist, zu übertragen.

Was braucht den Server dafür? - das SQL Datenbank URL für eine SQL Verbindung und die SQL Anfragen für den Client-Seite, um das Ergebnis auf die Karte zu zeigen.

```
new SqlConnection("Persist Security Info=False;Integrated Security=true; Initial Catalog = CargoGuardDB; Server = 10.123.12.456")
```

Listing 12: SQL Verbindung

Die Listing 12 zeigt, wie eine Verbindung zur unseren CargoGuard Datenbank herstellt, man muss den Name der Datenbank „CargoGuardDB“ und das URL vom Server, wo die DB liegt, bestimmen.

Danach ist eine SQL Anfrage erforderlich.

```
using (var connection = new SqlConnection("Persist Security Info=False;Integrated Security=true;
Initial Catalog = CargoGuardDB; Server = 10.174.20.246"))
{
    await connection.OpenAsync();

    var cmd = new SqlCommand("SELECT p.Latitude, p.Longitude, p.UnitID, p.SignalStrength, n.SerialNumber " +
        "FROM GsmSignalStrength p", connection);
```

Listing 13: Verbindung zur SQL Datenbank

Wie kann man jetzt die Daten auf der Client-Seite vom Server erhalten?

Jeden Server hat ein URL mit einem Port, dies steht in dem launchSettings.json Datei in der Einstellung vom Server und ist änderbar.

Mein URL lautet <http://localhost:5000>; localhost heißt, dass der Server auf lokalen Computer über den Port 5000 gehostet wird.

Kommen wir jetzt zur Client-Seite, in meinem Projekt ist das die Angular.

Wie beim theoretischen Teil geklärt wurde, ist das HTTP-Modul für die Verbindung zwischen Server-Seite und Client-Seite verantwortlich. Deshalb müssen wir dieses HTTP-Modul in Angular zuerst importieren und dann nutzen:

- Im ersten Schritt schreiben wir diese Zeile in den Code: `import { HttpClient } from '@angular/common/http';`
- Im zweiten Schritt deklarieren wir das **HttpClient** Modul im Konstruktor: `constructor(public http: HttpClient)`
- Am Ende binden wir das Modul auf den Server <http://localhost:5000> über diese Methode: `„this.http.get(http://localhost:5000).subscribe((result: any) => { })“`
Mit der GET-Methode werden Daten von einer angegebenen Ressource angefordert.
Mit der Subscribe Funktion können wir unsere Daten kriegen, wenn die Verbindung erfolgreich abgeschlossen ist.

Es kann aber selbstverständlich sein, dass wir Fehler vom Server kriegen, wenn etwas nicht stimmt. (Abb. 4.8)

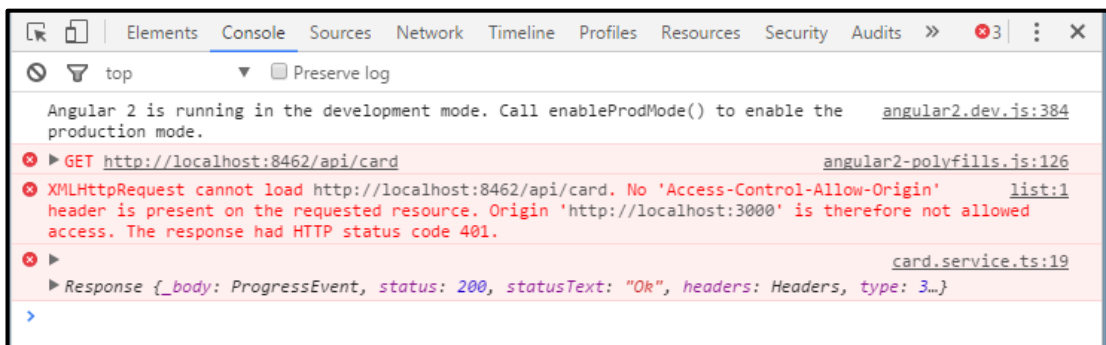


Abbildung 4.8: Ein Beispiel eines Fehlers

Eine solche Fehlermeldung habe ich in meinem Projekt mehr Male bekommen bevor ich es richtig konfiguriert.

Diese „Access Control Allow Origin“-Fehler bedeutet, dass die aufgerufene Methode mit dem angegebenen Ursprung nicht genutzt werden darf. Das kann mit etwas Code auf der Server-Seite behoben werden.

```
services.AddCors(o => o.AddPolicy("MyPolicy", builder =>
    {
        builder.AllowAnyHeader();
        builder.AllowAnyMethod();
        builder.AllowAnyOrigin();
    }));
```

Listing 14: CORS Code

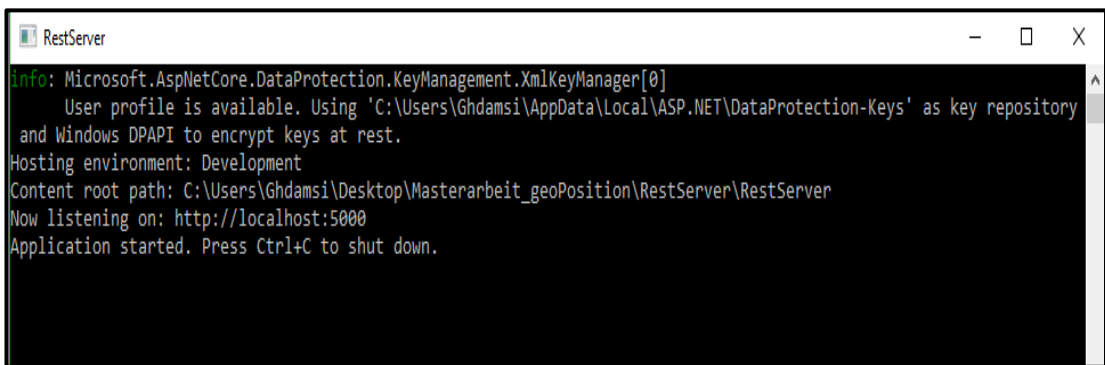
Mit diesem Code konfigurieren wir unseren Server Service mit einer Methode, so dass alle Header, Methode oder Origin erlaubt sind.

4.3 Ausführung der Anwendung

Dieser Teil kümmert sich um die graphische Realisierung der Anwendung.

Als ersten Schritt möchte ich zeigen, wie ich meine Anwendung starten kann.

Zum Anfang führe ich meinen Server Code aus, dann wird eine Microsoft Konsole erscheinen, wo zu sehen ist, dass der Server gestartet ist.



```
RestServer
Info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using 'C:\Users\Ghdamsi\AppData\Local\ASP.NET\DataProtection-Keys' as key repository
      and Windows DPAPI to encrypt keys at rest.
Hosting environment: Development
Content root path: C:\Users\Ghdamsi\Desktop\Masterarbeit_geoPosition\RestServer\RestServer
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
```

Abbildung 4.9: Rest-Server Konsole

Danach muss ich die Angular Plattform auch starten. Dafür muss ich mit der rechten Maustasten und Umschalt-Taste auf den Ordner, in dem das angular Projekt liegt, klicken, und ein Power-Shell Fenster öffnen. So bekomme ich auch eine Microsoft Konsole, in die ich den Befehl schreibe, um die Angular Plattform auszuführen.

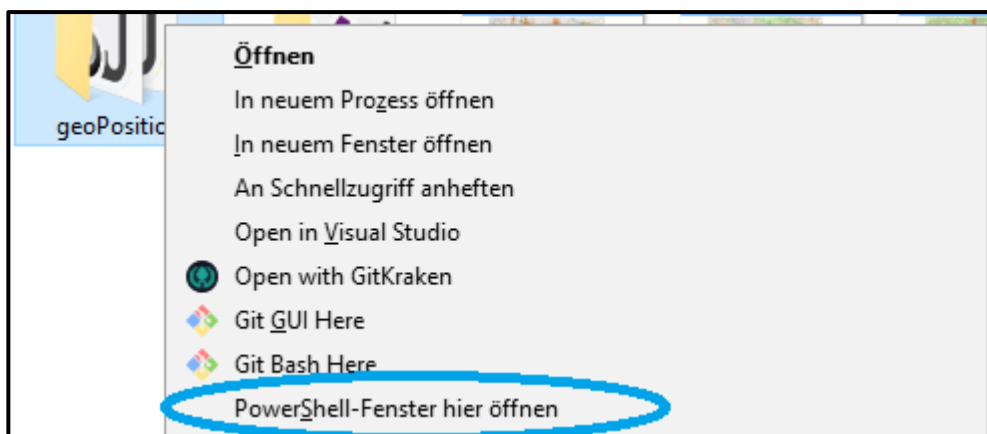


Abbildung 4.10: PowerShell öffnen

```
ng
PS C:\Users\Ghdamsi\Desktop\Masterarbeit_geoPosition\geoPosition> ng serve
** Angular Live Development Server is listening on localhost:4200, open your browser on
http://localhost:4200/ **

Date: 2019-03-07T22:55:58.365Z
Hash: ab6fdefc919cdf5ed54
Time: 10446ms
chunk {main} main.js, main.js.map (main) 29.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 223 kB [initial] [rendered]

chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 147 kB [rendered]
chunk {styles} styles.js, styles.js.map (styles) 793 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.83 MB [initial] [rendered]
   [wdm]: Compiled successfully.
```

Abbildung 4.11: Angular Starten

Der Befehl, um die Angular Plattform zu öffnen ist; **ng serve**.

Wie auf der Server-Seite hat angular auch eine URL. In meinem Projekt ist das <http://localhost:4200/>. Das heißt, dass die angular auf dem lokalen Port 4200 gehostet wird. Diese Parameter sind auch änderbar, man kann den Port ändern und auch die URL Adresse.

Wenn „Compiled successfully“ geschrieben ist, dann ist alles in Ordnung und wir können unsere Anwendung in dem Browser sehen.

Wenn das nicht der Fall ist, kriegen wir eine Fehlermeldung mit der Art des Fehlers. Dann muss man die Fehler im Code korrigieren.

Im letzten Schritt schreiben wir die URL der Anwendung in irgendeinen Browser; Google Chrome, Microsoft Edge, Firefox usw.

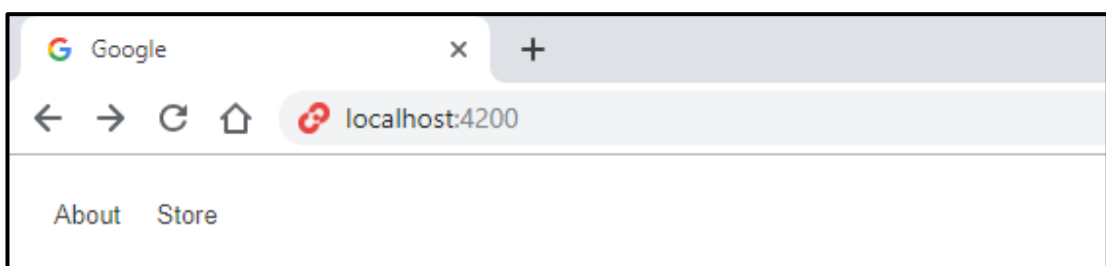


Abbildung 4.12: Die URL der Anwendung

Nach dem Abruf dieser URL bekommen wir die Schnittstelle unserer Anwendung zu sehen.

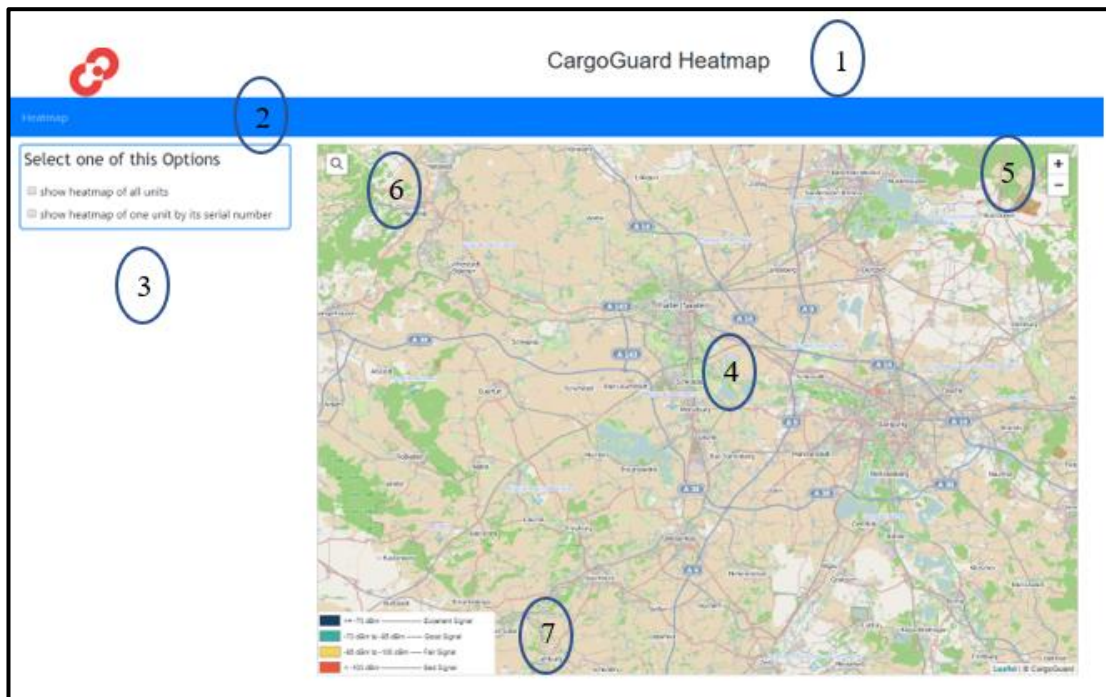


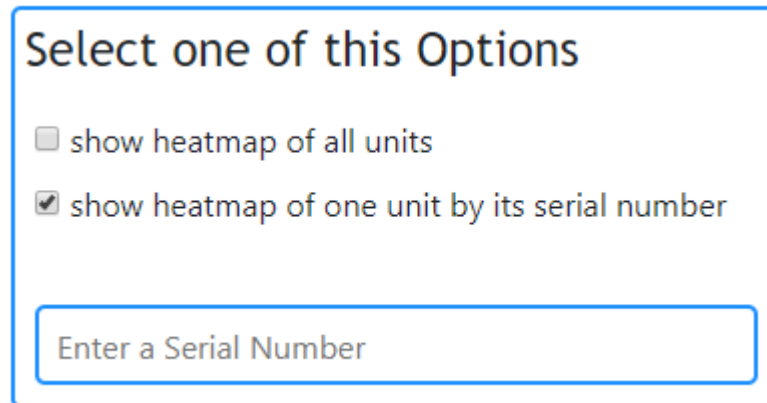
Abbildung 4.13: Die Schnittstelle der Anwendung

In diesem Schritt werde ich die Schnittstelle meiner Anwendung detailliert beschreiben. Ich habe ein paar Zahlen über die Abbildung hinzugefügt, damit die Beschreibung einfacher wird.

1. ist die Kopfzeile meiner Anwendung, wo das Logo und der Name der Firma; CargoGuard; sowie Anwendungsname; Heatmap.
2. Ist ein Navbar, wo man verschiedene Komponenten stellen kann, in meinem Fall ist nur eine Komponente, die ich als Heatmap genannt habe.
3. Ist ein kleines Menü mit zwei Optionen:
 - Show heatmap for all units
 - Show heatmap for one unit using its Serial Number

Am ersten ist der Eingabefeld von Seriennummer nicht gezeigt aber wenn man die zweite Option des Menüs wählt, scheint dies Feld ein. (Abb. 4.14)

Dies verschwindet immer, wenn seine Checkbox nicht geprüft.



Select one of this Options

show heatmap of all units

show heatmap of one unit by its serial number

Enter a Serial Number

Abbildung 4.14: Das Eingabefeld der zweiten Option

4. Ist die Karte, die von Leaflet JS gebaut ist. Leaflet JS eine Open-Source-JavaScript-Bibliothek für interaktive Karten. Kartenkacheln werden standardmäßig über OpenStreetMap oder MapQuest (mit einem App-Schlüssel) bereitgestellt.

In meinem Fall nutze ich die Karte über OpenStreetMap. Eine detaillierte Beschreibung befindet sich in Theoretische Teil.

5. Sind Standard Tasten bei der Leaflet Karte; die (+) Taste ist für das Vergrößern der Karte und die (-) Taste ist für das Verkleinern der Karte.
6. Ist eine Taste für die Suche einer Adresse, wenn man daraufklickt, wird eine Suchfeld gezeigt. (Abb. 4.15)

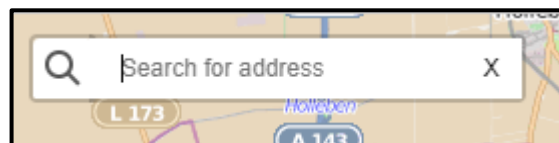


Abbildung 4.15: Das Adresssuchfeld

Dies ist eine intelligente Suche, man braucht nicht die ganze Adresse zu schreiben. Ein richtiges Wort reicht, um alle Möglichkeiten, die zu diesem Wort gehört, zu finden. (Abb. 4.16)

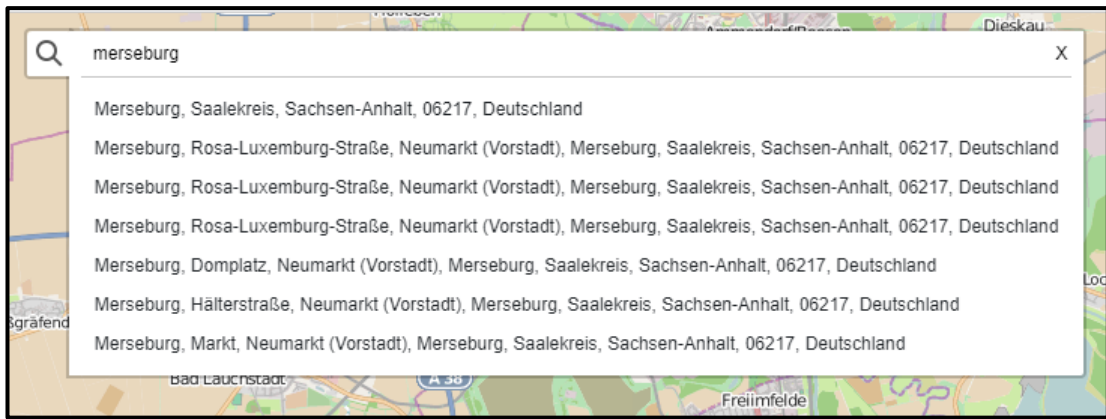



Abbildung 4.16: Ein Beispiel einer Suche

Die Abbildung 4.16 ist ein Beispiel für die Suche einer Adresse, hier habe ich Merseburg als ein Beispiel Wort genommen und wie das oben genannten Bild zeigt, kommt viele Möglichkeiten zu diesem Wort, welche man wählen kann.

Wir können die (x) Taste am Ende des Suchfeldes nutzen, um die angegeben Adresse zu löschen und die  Taste, um das Suchfeld zu verstecken oder zu zeigen.

7. Die GSM Abdeckung wird mit verschiedenen Farben auf die Karte gezeigt, deswegen habe ich eine Legende auf die Karte gestellt so man kann verstehen, welche Rolle die Farben spielen.

Bei der Legende gibt es 4 diverse Farben.

- Die blaue ist für eine exzellent Signalstärke bis -70 dBm
- Die grüne ist für eine gute Signalstärke, die mit Wert von -70 dBm bis -85 dBm kommt.
- Die gelbe ist für eine fair Signalstärke, die vom Wert -86 dBm bis -100 dBm ist.
- Die Rote ist für die schlechte Signalstärke, die über -100 dBm ist.

In der vorherigen Beschreibung habe ich die verschiedenen Komponenten der GUI ohne die gezeigte GSM Abdeckung charakterisieren.

Im nächsten Schritt konzentriere ich auf das Ergebnis auf die Karte bzw. die GSM Abdeckung.

Um ein gutes Ergebnis zu kriegen, habe ich verschiedene Methoden probiert. wie in der Vorbereitung der Daten Teil erwähnt, hatte ich ein Problem mit meinem ersten Versuch, weil die erste SQL Abfrage nicht die richtige für eine gute Resultat war. In den Absatz erkläre ich was war diese Abfrage und wie ist auf die Karte gezeigt wird.

4.3.1 Erste Versuch mit der GSM Abdeckung

4.3.1.1 SQL Abfrage

```
using (var connection = new SqlConnection("Persist Security Info=False;
Integrated Security=true; Initial Catalog = CargoGuardDB; Server = 10.123.12.456"))
{
    await connection.OpenAsync();

    var cmd = new SqlCommand("SELECT p.Latitude, p.Longitude, p.SignalStrength " +
        "FROM GsmSignalStrength p ", connection);
```

Listing 15: Erste SQL Abfrage auf der Rest-Server

Für die erste Abfrage habe ich nur die Daten von der Breite, Länge und Signalstärke von meiner Tabelle von dem SQL Server abgerufen.

Auf der angular-Seite habe ich die leaflet heatmap Bibliothek genutzt. Mit dieser habe ich die verschiedenen Positionen als Punkte auf der Karte dargestellt. Die Messwerte der Signalstärke sind mit verschiedenen Farben repräsentiert.

Im ersten Versuch habe ich nur zwei Farbe genutzt. Blau für die gute Signalstärke (bis -100 dBm) und Rot für die schlechte Signalstärke (ab -100 dBm).

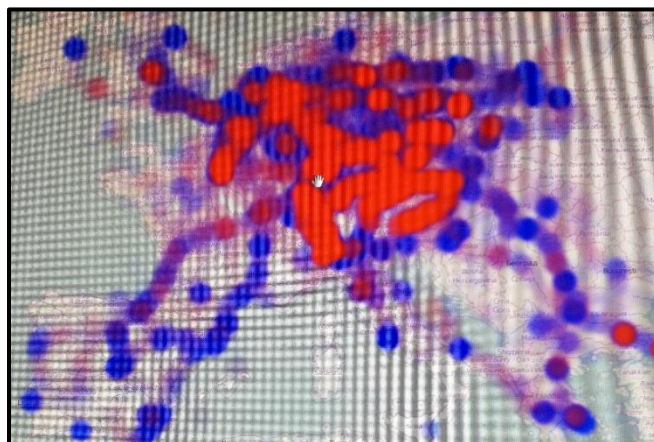


Abbildung 4.17: Das Erst Ergebnis mit zwei Farben

In der Abbildung 4.17 merkt man, dass die vorherrschende Farbe auf der Karte rot ist. Das heißt, dass die Signalstärke an den meisten Orten schlecht ist.

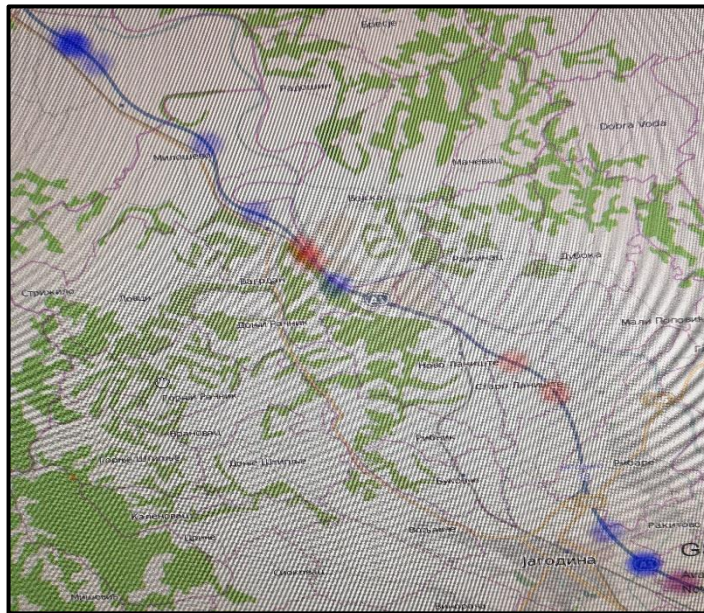


Abbildung 4.18: Ein Vergrößern der Punkte auf die Karte

Man merkt auch in der Abbildung 4.18, dass die Position sind übereinander liegen, wenn ein Punkt dunkel ist. Wie In der Abbildung 4.18 gezeigt ist, kann es auch sein, dass jede einzelne Position verschieden Signalstärke hat.

Deshalb habe ich entschieden, mehrere Farben zu nutzen, statt zwei vier. Die gute Signalstärke sind beim nächsten Versuch auf drei Menge mit drei Farben geteilt. Blau für Exzellent, Grüne für gut und gelb für ausreichend.

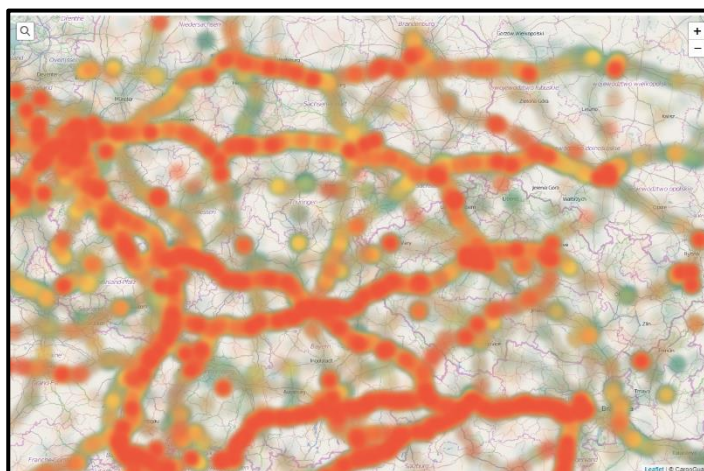


Abbildung 4.19: Die GSM-Abdeckung für das zweite Versuch

Die Abbildung 4.19 zeigt das neue Ergebnis des zweiten Versuchs. Jetzt ist das Aussehen der GSM-Abdeckung ja besser aber das Problem von falschen und schlechten Punkten besteht weiterhin. Wir merken, dass die Punkte mit verschiedenen Signalstärken überlappen.

Aus diesem Grund haben wir entschieden, die Daten erst aufzubereiten, bevor wir sie beim Benutzer anzeigen.

4.3.2 Das finale Ergebnis der GSM-Abdeckung

Mit den bisherigen gezeigten Versuchen haben wir ein schlechtes Ergebnis erhalten. Deshalb hatten wir eine neue Idee; die gezeigten Punkte auf der Karte sollten in Rechtecke verändert werden. Dies hilft, die Streuung der Punkte mit verschiedene Signalstärken gesammelt darzustellen.

Wie habe ich das realisiert?

Ich nehme ein Beispiel zur besseren Erklärung. Jede Position auf der Karte hat eine Breite und Länge, wie z.B. Merseburg Zentrum hat als Breite 51.357208 und 11.992270 Länge; wir merken, dass jede Werte mehr als fünf Dezimalstellen hat.

Um eine bestimmte Rechteck Umgebung, in der diese Punkt liegt, zu kriegen, löschen wir einfach drei oder vier Dezimalstellen, um erste Ecke zu bauen und dann für die zweite Ecke wir schreiben einfach die nächste dezimale Zahl. In der Abbildung 4.20 ist eine kurze Erklärung dafür.

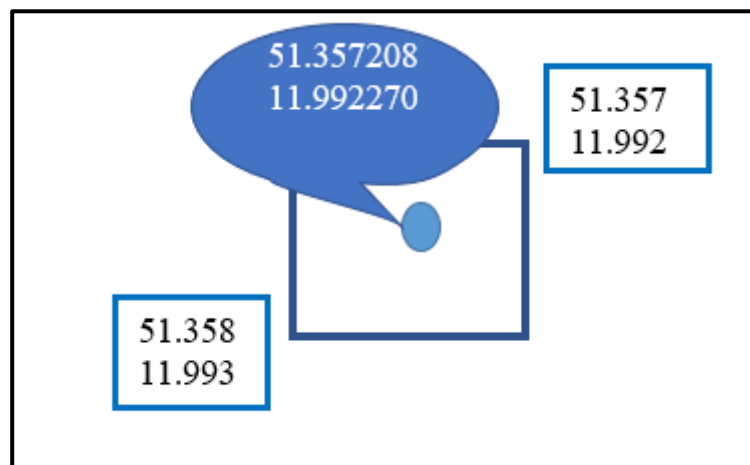


Abbildung 4.20: Das Aufbau der Rechtecken

Für die Signalstärke rechnen wir der Mittelwert der Signalstärke aller Punkten in diesem Rechteck so kriegen wir immer nur einen Wert für jede Rechteck.

Um diese Idee, in meinem Projekt zu realisieren, brauche ich einen SQL Abfrage auf dem Server zu schreiben.

```
SELECT AVG(p.SignalStrength) as SignalStrength, Floor(p.Latitude*100)/100 as Latitude,
Floor(p.Longitude*100)/100 as Longitude,
p.UnitID as UnitID, count(*) as Count
FROM GsmSignalStrength p
Group by FLOOR(p.Latitude * 100), FLOOR(p.Longitude * 100), p.UnitID
order by FLOOR(p.Latitude * 100) ASC, FLOOR(p.Longitude * 100) ASC, p.UnitID ASC
```

Listing 16: Die SQL Abfrage des letzten Versuchs

Beschreibung:

Mit der AVG Funktion kriegen wir den Mittelwert der Signalstärke und der Floor Funktion. Diese hat als Rolle die Zurückgabe des größten ganzzahligen Wertes, der gleich oder kleiner als der Breite und Länge.

(Latitude*100)/100 um zwei dezimal zu kriegen.

Nun fasse ich die gesamte Abfrage zusammen. Ich wähle die Mittelwerte von der Signalstärke und Floor Funktion von Latitude und Longitude aus meiner Tabelle, und als Bedingungen:

- Ich gruppere diese Variablen nach Floor Latitude, Floor Longitude und UnitId
- Ich sortiere dieselben Variablen aufsteigende nach Floor Latitude, Longitude und UnitId

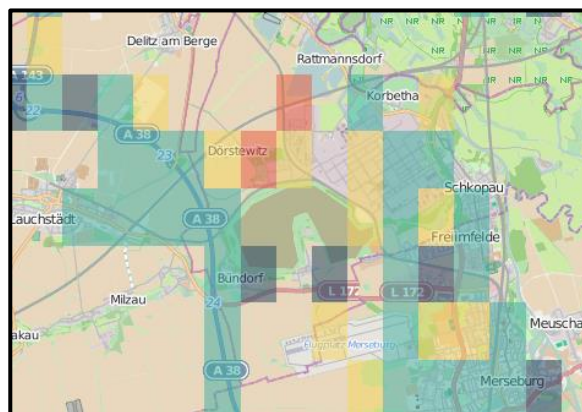


Abbildung 4.21: Das beste Resultat

Wir merken jetzt in der Abbildung 4.21, dass das Resultat als Rechtecken repräsentiert. Das sieht viel besser und richtiger als die andere Versuchen aus.

4.3.3 Die Benutzeroberfläche der Anwendung

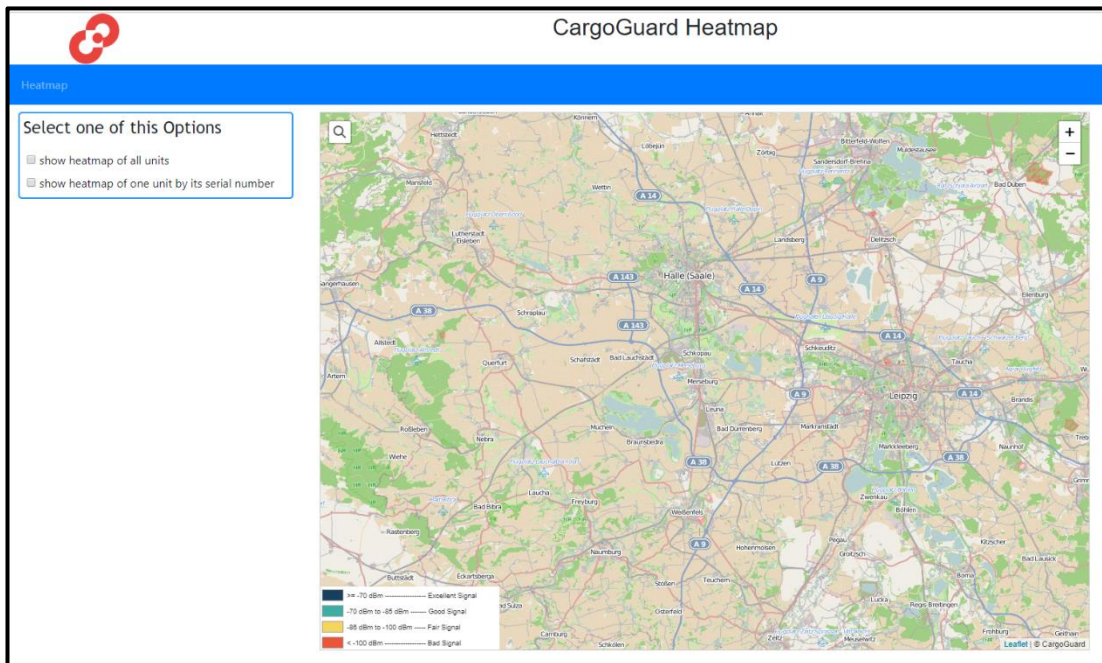


Abbildung 4.22: Die Anwendung auf dem ersten blick

Nach dem Abruf des URL: Localhost:4200 bekommen wir diese Benutzeroberfläche.

Die Zoomstufe auf die Karte ist Manuel eingestellt, das heißt, wir schreiben die Zoomstufe Standard nur einmal in dem Code. Wenn die Komponente jedes Mal neugeladen wird, wird die Karte mit dieser Standard Zoomstufe gezeigt.

Nach diesem Schritt hat der Benutzer die Möglichkeit zwischen die beiden Optionen zu wählen und auf einer von beiden Checkbox zu klicken.

4.3.3.1 Erste Option: Show heatmap of all units

Die erhalten Daten von dem Server sind ca. Millionen und das kann ein großes Problem für die Anwendung machen.

Wenn ich alle diese Daten in einmal auf die Karte abrufe, hängt die Anwendung und wird nicht mehr funktioniert. Deswegen habe ich eine Methode geschrieben, die uns hilft, um diesem Problem zu lösen.

Die Lösung ist, dass ich die Daten nur in einem bestimmten Kartenausschnitt lade. Die Karte ist immer wie ein Rechteck gezeigt, das heißt hat jede Karte vier Punkte, die ein Grenzen bauen.

Aber mit der Leaflet Bibliothek kann ich zwei Punkte, Nordöstliche und Südwestliche, von vier nutzen, um diese Grenzen zu bauen. Jeder Punkt von den beiden hat eine Breite und Länge.

Diese Positionen können immer automatisch gekriegt werden, wenn der Benutzer auf der Karte navigieren oder die Karte verkleinert und vergrößert.

Jedes Mal wird dieser Vorgang ausgeführt, werden die Werten (Nordost Breite, Nordost Länge, Südwest Breite, Südwest Länge) dieser Positionen zum Server geschickt. Diese werden in der SQL Abfrage als Bedingung genutzt.

```
this.http.get(this.moreResizedUrl + "?northLat=" + this.northLat +  
"&southLat=" + this.southLat + "&northLng=" + this.northLng + "&southLng=" + this.southLng
```

```
public async Task<ActionResult<IEnumerable<Model>>> AllUnitsAsync  
(double northLat, double southLat, double northLng, double southLng)
```

„This.moreResizedUrl“ in der http-Abfrage ist die URL der Rest-Server.

Die werten werden erste vom Client-Seite angular zum Server geschickt, wie der obene Code zeigt.

Im nächsten Schritt werden die Variablen in der Funktionsparameters abgerufen, wie dieser Code zeigt:

```
cmd.Parameters.AddWithValue("@northLat", northLat);  
cmd.Parameters.AddWithValue("@southLat", southLat);  
cmd.Parameters.AddWithValue("@northLng", northLng);  
cmd.Parameters.AddWithValue("@southLng", southLng);
```

Jede Werte wird in einer Variablen gespeichert, damit kann man in der SQL Abfrage Bedingung nutzen.

```

SELECT AVG(p.SignalStrength) as SignalStrength,
Floor(p.Latitude*100)/100 as Latitude, Floor(p.Longitude*100)/100 as Longitude, count(*) as Count " +
  FROM GsmSignalStrength p
  WHERE p.Latitude < @northLat AND p.Latitude > @southLat
  AND p.Longitude < @northLng AND p.Longitude > @southLng
  Group by FLOOR(p.Latitude * 100), FLOOR(p.Longitude * 100)
  order by FLOOR(p.Latitude * 100) ASC, FLOOR(p.Longitude * 100) ASC);

```

Listing 17: Die SQL Abfrage der ersten Option

Diese SQL Abfrage habe ich vorher erklärt. Nun beschreibe ich die Bedingung des Kartenausschnittes.

Mit dieser Bedingung nehme ich aus der Tabelle nur:

- Latitude kleiner als Nordöstlicher Breite und größer als südwestlicher Breite.
- Longitude kleiner als Nordöstlicher Länge und größer als Südwestlicher Länge.

Jedes Mal werden diesen Werten geändert, wird dies SQL Abfrage automatische geändert.

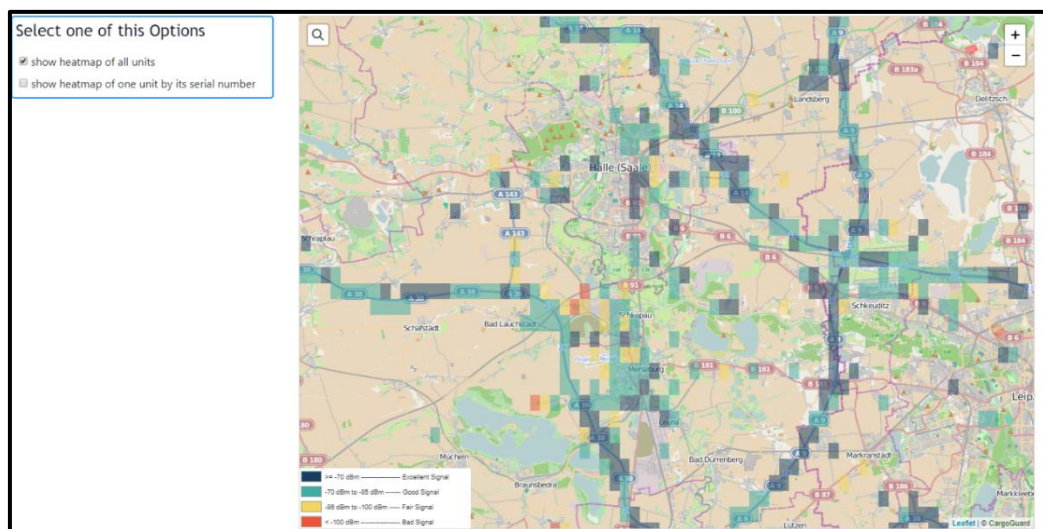


Abbildung 4.23: Das Ergebnis der ersten Option

Die Abb. 4.23 zeigt das Ergebnis nach der Anwendung des Kartenausschnitt Filter. Die Karte lädt andere Daten, wenn der Benutzer die Karte in andere Richtung bewegt.

Jetzt kommen wir zu anderem Problem; das Verkleinern und Vergrößern der Karte

In der Abbildung 4.23 kann man merken, dass die Rechtecke fast klein sind. wenn der Nutzer die Karte verkleinert, und die Zoomstufe fünf erreicht, werden die Rechtecken fast verschwendet.

Deshalb machen wir eine Kleine Änderung bei unsere SQL Abfrage abhängig mit der Zoomstufe.

```
SELECT AVG(p.SignalStrength) as SignalStrength,  
Floor(p.Latitude*10)/10 as Latitude, Floor(p.Longitude*10)/10 as Longitude, count(*) as Count " +  
FROM GsmSignalStrength p  
WHERE p.Latitude < @northLat AND p.Latitude > @southLat  
AND p.Longitude < @northLng AND p.Longitude > @southLng  
Group by FLOOR(p.Latitude * 10), FLOOR(p.Longitude * 10)  
order by FLOOR(p.Latitude * 10) ASC, FLOOR(p.Longitude * 10) ASC);
```

Mit diesem Code kriegen wir die Breite und Länge nur mit einer Dezimalstelle. Bei angular addiere ich einfach 0.1 werden zu den erhaltenen Daten der Positionen aus dem Server.

z.B.: (51.357208, 11.992270) die Position von Merseburg Zentrum. In meinem Fall mit der Floor Methode, die ich vorher erklärt habe, bekomme ich (51.4, 12.0). Bei Client-Seite wird die neue Position so (51.4 + 0.1, 12.0 + 0.1) gerechnet.

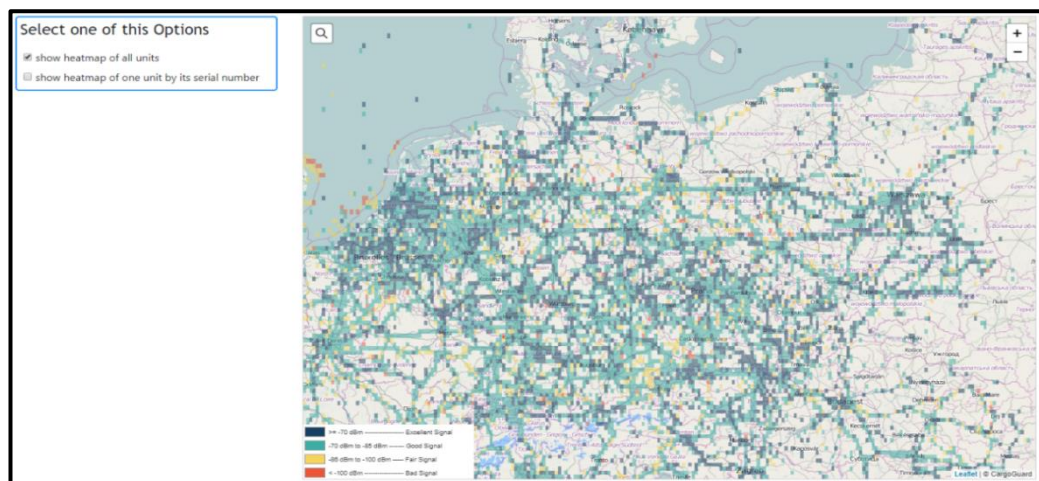


Abbildung 4.24: Das Ergebnis mit einer niedrigen Zoomstufe

Mit diesen Methoden werden die Rechtecke größer als die andere gezeigt so kann man die beim niedrige Zoomstufe merken.

In der Abbildung 4.24 können wir sehen, dass wir immer die Abdeckung auch bei der niedrigen Zoomstufe sehen können. Für jede Zoomstufe Intervall habe ich eine neue Abfrage bis die letzten niedrigen Zoomwerte abgerufen. (Abbildung 4.25).

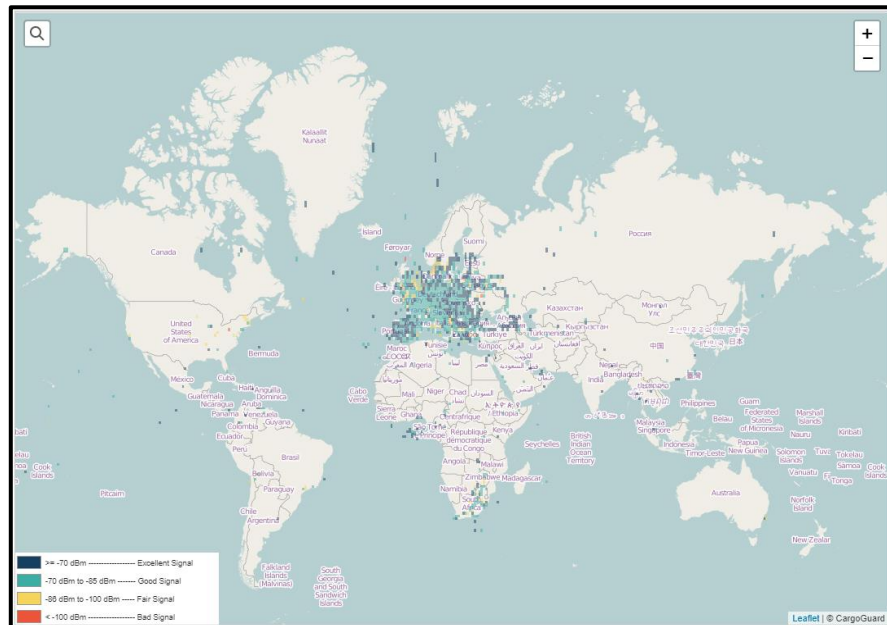


Abbildung 4.25: Das Ergebnis mit einer sehr niedrigen Zoomstufe

Die signalstärke erhalten wir immer mit dieselben (Mittelwert) Methode von SQL.

4.3.3.2 Zweite Option: Show heatmap of one unit

Diese Option ist für die Anzeige der GSM-Abdeckung für jeden GPS Tracker. Da die Daten von jedem Modul nicht zu groß sind, habe ich der Filter der Kartenausschnitt nicht angewandt.

Diese Option braucht den Seriennummer jedes GPS Trackers. In der Anwendung gibt es einen Eingabefeld, in dem den Seriennummer geschrieben werden kann.

Select one of this Options

show heatmap of all units

show heatmap of one unit by its serial number

1006013

Abbildung 4.26: Das Eingaben der Seriennummer

Diese Eingabefeld wird erscheint nur, wenn wir die zweite Option wählen. Man schreibt einen bestimmten Seriennummer und dies wird zum Server geschickt, um mit der SQL Abfrage zu nutzen.

Wie im vorherigen Schritt wird diese Variable mit der URL beim Angular-Seite nach dem Rest-Server geschickt.

```
this.http.get(this.UnitLessResizedApiUrl + "?serialNumber=" + this.serial)
```

This.UnitlessResizedApiUrl in HTTP-Abfrage ist die URL des Rest-Servers und this.serial ist der Werte von der Seriennummer. Wir schicken diese Abfrage von angular und erhalten die im Rest-Server mit dieser unten genannten Methode.

```
public async Task<ActionResult<IEnumerable<Model>>> LessResizedUnitAsync(string serialNumber)
cmd.Parameters.AddWithValue("@serialNumber", serialNumber);
```

Der Variable SerialNumber ist als Parameter in der Funktion, in der unsere SQL Abfrage liegt.

```
SELECT AVG(p.SignalStrength) as SignalStrength, Floor(p.Latitude*10)/10 as Latitude,
Floor(p.Longitude*10)/10 as Longitude,
p.UnitID as UnitID, n.SerialNumber as SerialNumber, count(*) as Count
FROM GsmSignalStrength p " +
LEFT JOIN Unit n ON p.UnitID = n.Id
WHERE n.SerialNumber = @serialNumber
Group by FLOOR(p.Latitude * 10), FLOOR(p.Longitude * 10), p.UnitID, n.SerialNumber
order by FLOOR(p.Latitude * 10) ASC, FLOOR(p.Longitude * 10) ASC, p.UnitID ASC, n.SerialNumber ASC
```

In dieser Abfrage benutze ich die LEFT JOIN Klausel, um meine Tabelle GSMSignalStrength und die Tabelle Unit, in der die Seriennummern liegen, zu verbinden.

Die Verbindung zwischen die beiden Tabellen wird über die UnitId Spalte in meiner Tabelle und die Id Spalte in der Unit Tabelle realisiert. Wir kriegen die Daten nur Wenn: - Die werten beider Spalten gleich sind

- Der bekommende Variable SerialNumber ist gleich als eine von den Seriennummern in der Unit Tabelle.

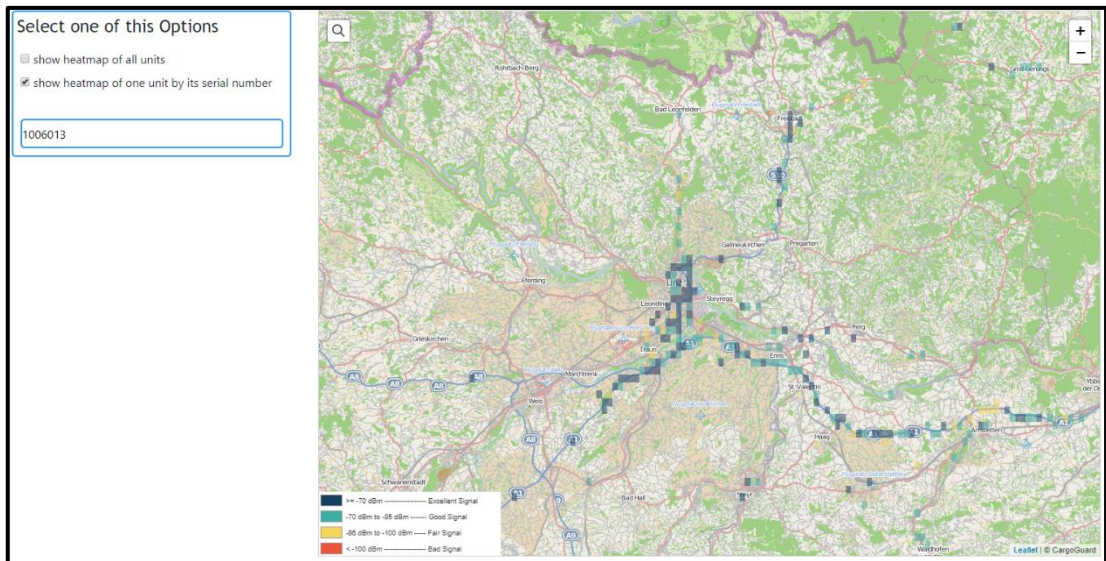


Abbildung 4.27: Die GSM-Abdeckung eines Moduls

Bei dieser Option steht auch dasselbe Problem mit der Zoomstufe, deshalb habe ich dieselbe Methode von erster Option angewandt.

Die Untengenannten Bilder zeigen die verschiedenen Ergebnisse der GSM-Abdeckung mit verschiedenen Zoomstufe für den gleichen GPS Tracker.

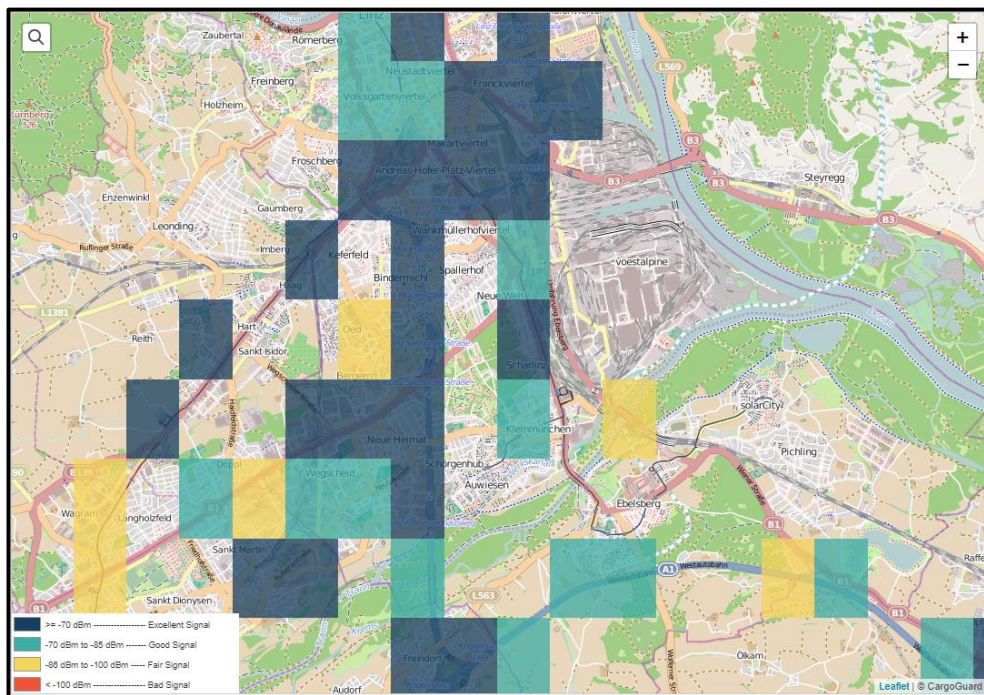


Abbildung 4.28: Die GSM-Abdeckung nach dem Vergrößern der Karte

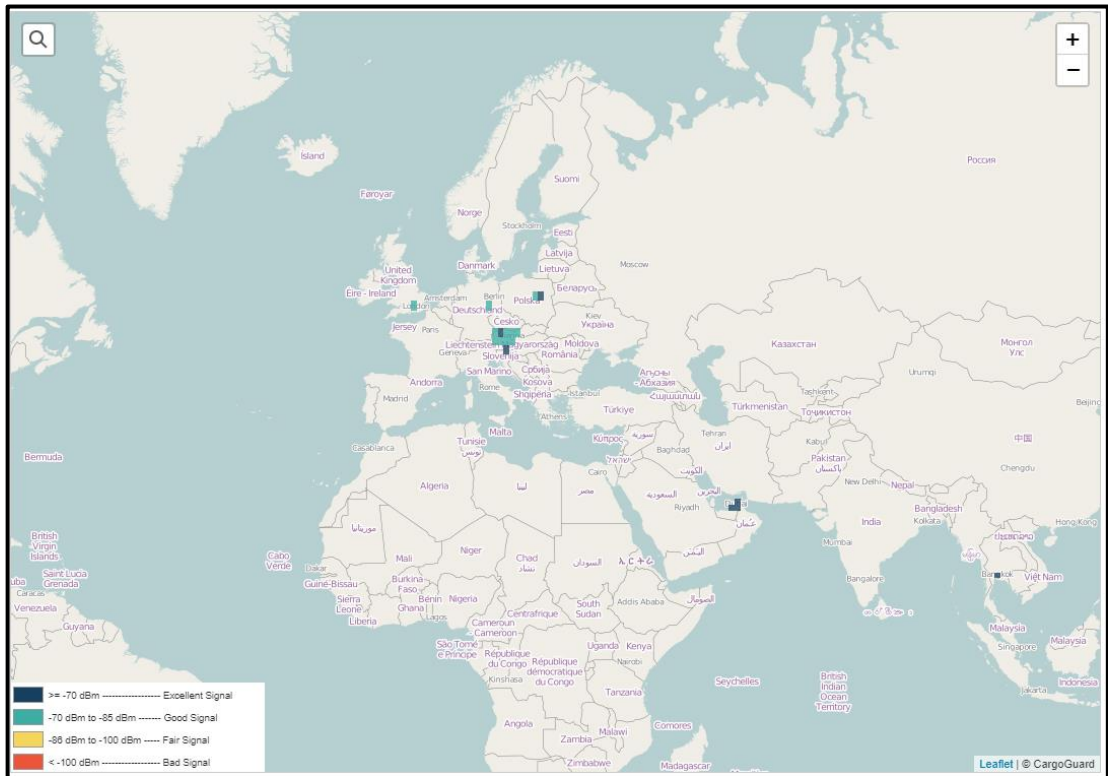


Abbildung 4.29: Die GSM-Abdeckung nach dem Verkleinern der Karte

4.4 Zusammenfassung

In diesem Kapitel habe ich im ersten Teil die betreffenden Probleme in der Realisierung erläutern und wie ich Schritt für Schritt an der richtigen Lösung dafür bekommen habe.

Danach habe ich erklärt, wie ich an dem finalen Ergebnis mit ein paar Bilder der Anwendung gekommen.

5 Fazit

Da ein Teil meines Projekts basierend auf die Internet of Things Technologie, wollte ich bemerken, dass dieses Projekt zu verschiedenen Bereichen gehört werden kann. Im Transport IoT-Bereich, welche unserem Fall ist und auch beim Mobilfunk-Feld, da den zweiten Teil der Arbeit die GSM Abdeckung repräsentiert.

Nach dem Einleitung Kapitel und im ersten Teil versuchte ich alle benutzte Methode, Software Hardware usw. theoretisch erklären sowie beim Projektentwurf Kapitel, wo die Verschiedene Diagrammen dargestellt werden.

Als erste Idee in dieser Arbeit war eine GSM-Abdeckung als Heatmap zu darstellen aber die bestehenden Daten in der SQL Datenbank haben mir leider nicht richtig geholfen, weil die Daten in mehreren Tabellen fehlerhaft abgespeichert werden.

Deshalb im Lauf dieses Projekts wurde verschiedene Methoden, SQL Abfragen sowie GSM-Abdeckung Formen gesucht, um ein gutes finales Ergebnis zu zeigen.

Eine Verarbeitung von Daten war sehr wichtig in meiner Arbeit, um den falschen Werten von sowie der Positionen zu vermeiden.

In diesem Projekt kann man selbstverständlich noch andere Ideen weiter entwickeln. Verschiedene Anwendung können mit dem bisherigen Ergebnis realisieren. Ich nenne:

- Die Daten weiter analysieren und ein Algorithmus selbst schreiben, das uns helfen kann, um die schlechten Orte wie Tunnel oder Zwischen Bergen, wo man keine Signalstärke finden kann, auf die Karte zu darstellen.
- Dieses Projekt basierend nur auf die Signalstärke Daten, die mit GPS Daten zusammenkommen, aber mit dem Fall wo es keine Messdaten von GPS gibt, kann man leider keine GSM-Abdeckung Karte darstellen. Deswegen ein Algorithmus, das die Messung von Signalstärke unabhängig von GPS Position macht, kann helfen, um eine Weltweite GSM-Abdeckung Karte zu darstellen.

Abbildungsverzeichnis

Abbildung 1.1: CargoGuard Logo	5
Abbildung 1.2: CargoCop	6
Abbildung 2.1: IoT	10
Abbildung 2.2: IoT-Architektur	11
Abbildung 2.3: IoT-Bereichen	14
Abbildung 2.4: HTML-Seite	19
Abbildung 2.5: NodeJS Logo.....	23
Abbildung 2.6: Web-Anwendung-Architektur	26
Abbildung 3.1: Flussdiagramm des Designs des Projektes	31
Abbildung 3.2: Vorläufiges Anwendungsfalldiagramm	32
Abbildung 3.3: Detailliertes Anwendungsfalldiagramm	33
Abbildung 3.4: Allgemeines Sequenzdiagramm	34
Abbildung 3.5: Sequenzdiagramm der All Units Option.....	35
Abbildung 3.6: Sequenzdiagramm der einzelnen Unit Option	36
Abbildung 3.7: Aktivitätsdiagramm von All Units Option.....	37
Abbildung 3.8: Einzelne Unit Aktivitätsdiagramm	38
Abbildung 3.9: Klassendiagramm.....	39
Abbildung 4.1: SQL Server Logo	44
Abbildung 4.2: Verbindung zum SQL Server.....	44
Abbildung 4.3: SQL Datenbank.....	44
Abbildung 4.4: Die Tabellen in der Datenbank	45
Abbildung 4.5: Das Ergebnis der SQL Abfrage	46
Abbildung 4.6: Meine Tabelle	49
Abbildung 4.7: Die neue Tabelle	50
Abbildung 4.8: Ein Beispiel eines Fehlers	53
Abbildung 4.9: Rest-Server Konsole	54
Abbildung 4.10: PowerShell öffnen.....	54

Abbildung 4.11: Angular Starten	55
Abbildung 4.12: Die URL der Anwendung	55
Abbildung 4.13: Die Schnittstelle der Anwendung	56
Abbildung 4.14: Das Eingabefeld der zweiten Option	57
Abbildung 4.15: Das Adresssuchfeld.....	57
Abbildung 4.16: Ein Beispiel einer Suche	58
Abbildung 4.17: Das Erst Ergebnis mit zwei Farben.....	59
Abbildung 4.18: Ein Vergrößern der Punkte auf die Karte	60
Abbildung 4.19: Die GSM-Abdeckung für das zweite Versuch	60
Abbildung 4.20: Das Aufbau der Rechtecken.....	61
Abbildung 4.21: Das beste Resultat	62
Abbildung 4.22: Die Anwendung auf dem ersten blick.....	63
Abbildung 4.23: Das Ergebnis der ersten Option	65
Abbildung 4.24: Das Ergebnis mit einer niedrigen Zoomstufe	66
Abbildung 4.25: Das Ergebnis mit einer sehr niedrigen Zoomstufe.....	67
Abbildung 4.26: Das Eingaben der Seriennummer.....	67
Abbildung 4.27: Die GSM-Abdeckung eines Moduls.....	69
Abbildung 4.28: Die GSM-Abdeckung nach dem Vergrößern der Karte	69
Abbildung 4.29: Die GSM-Abdeckung nach dem Verkleinern der Karte.....	70

Listingverzeichnis

Listing 1: HTML-Skript Beispiel.....	18
Listing 2: CSS Datei	20
Listing 4: ID in CSS.....	21
Listing 3: ID in HTML.....	21
Listing 5: External Stylesheet	21
Listing 6: Die erste SQL Abfrage	46
Listing 7: Die benutzte SQL Abfrage	47
Listing 8: Die SQL Abfrage zur Erstellung der Tabelle	48
Listing 9: Insert SQL Abfrage.....	49
Listing 10: Die SQL Abfrage der neuen Tabelle	49
Listing 11: Die SQL Abfrage der Anzeige der Tabelle	50
Listing 12: SQL Verbindung.....	51
Listing 13: Verbindung zur SQL Datenbank	51
Listing 14: CORS Code	53
Listing 15: Die erste SQL Abfrage auf dem Rest-Server	59
Listing 16: Die SQL Abfrage des letzten Versuchs	62
Listing 17: Die SQL Abfrage der ersten Option	65

Abkürzungsverzeichnis

GSM	Global System for Mobile communications
IoT	Internet of Things
IT	Information Technology
PIR	Passive infrared sensor
Wh	Kilowatt hour
NFC	Near-field communication
GPS	Global Positioning System
WLAN	wireless local area network
API	Application programming interface
HTML	Hyper Text Markup Language
XHTML	EXtensible HyperText Markup Language
PHP	Hypertext Preprocessor
SQL	Structured Query Language
URL	Uniform Resource Locator
dBm	Decibels milliwatts
Abb.	Abbildung

Literaturverzeichnis

[1] Margaret Rouse, internet of things (IoT), (online)

<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>

[2] JR Fuller, Practice Director, IoT, Wipro Limited, (online)

<https://techbeacon.com/enterprise-it/4-stages-iot-architecture>

[3] Priya, Application Layer Protocols for IOT : IOT Part 11, (online)

<https://www.engineersgarage.com/Articles/IoT-Application-Layer-Protocols>

[4] Margaret Rouse, internet of things (IoT), (online)

<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>

[5] Malcolm Maclean, Leaflet Tips and Tricks (online)

<https://leanpub.com/leaflet-tips-and-tricks/read>

[6] Victor Gerard Temprano, Google Maps API or Leaflet: What's Best for your Project? (online)

<https://www.codementor.io/victorgerardtemprano/google-maps-api-or-leaflet--what-s-best-for-your-project-faaev60vm>

[7] MDN web docs, CSS: Cascading Style Sheets, (online)

<https://developer.mozilla.org/en-US/docs/Web/CSS>

[8] Google, Angular,(online)

<https://angular.io/>

[9] Google, AngularJS, (online)

<https://angularjs.org/>

- [10] w3schools.com, JavaScript Tutorial, (online)
<https://www.w3schools.com/js/>
- [11] Wikipedia, TypeScript, (online)
<https://en.wikipedia.org/wiki/TypeScript>
- [12] Wikipedia, NodeJS, (online)
<https://en.wikipedia.org/wiki/Node.js>
- [13] Microsoft, C# programming guide, (online)
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
- [14] Wikipedia, .NET_Framework, (online)
https://en.wikipedia.org/wiki/.NET_Framework
- [15] Microsoft, SQL-Server, (online)
<https://www.microsoft.com/en-us/sql-server/default.aspx>
- [16] Wikipedia, Relation database managment system, (online)
https://en.wikipedia.org/wiki/Relational_database_management_system
- [17] Wikipedia, Relational model, (online)
https://en.wikipedia.org/wiki/Relational_model
- [18] TutorialsPoint, MS SQL Server – Architecture, (online)
https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_architecture.htm
- [19] TechOnTheNet, SQL Tutorial, (online)
<https://www.techonthenet.com/sql/index.php>
- [20] Teltonika, Wiki Knowledge Base, (online)
https://wiki.teltonika.lt/view/Mobile_Signal_Strength_Recommendations