



Abschlussprüfung Sommer 2017

Fachinformatiker für Anwendungsentwicklung  
Dokumentation zur betrieblichen Projektarbeit

## CrewOverwatch

Anwendung zum Anzeigen der Standorte von Gruppenmitgliedern auf mobilen  
Endgeräten

Abgabetermin: 24.04.2017

**Prüfungsbewerber:**

Dennis Rauscher

Lärchenweg 17

57339 Erndtebrück



**Ausbildungsbetrieb:**

NEO 7EVEN GmbH

Spandauer Straße 40

57072 Siegen

## Inhaltsverzeichnis

<b>Abbildungsverzeichnis .....</b>	<b>IV</b>
<b>Tabellenverzeichnis .....</b>	<b>V</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VI</b>
<b>Quellen.....</b>	<b>VII</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Projektbeschreibung .....	1
1.2 Projektziel.....	1
1.3 Projektumfeld .....	1
1.4 Projektbegründung.....	1
1.5 Projektschnittstellen .....	2
1.6 Projektabgrenzung .....	2
<b>2 Projektplanung .....</b>	<b>2</b>
2.1 Projektphasen .....	2
2.2 Ressourcenplanung.....	2
2.3 Entwicklungsprozess .....	2
<b>3 Analysephase .....</b>	<b>3</b>
3.1 Ist-Analyse .....	3
3.2.1 „Make or Buy“-Entscheidung.....	3
3.2.2 Projektkosten .....	3
3.2.3 Amortisationsdauer und Finanzierung.....	4
3.3 Nicht-monetäre Vorteile .....	5
3.4 Anwendungsfälle .....	5
3.5 Lastenheft/Fachkonzept .....	5
<b>4 Entwurfsphase.....</b>	<b>5</b>
4.1 Zielplattform .....	5
4.2 Architekturdesign .....	5
4.3 Entwurf der Benutzeroberfläche .....	6
4.4 Datenmodell.....	6
4.5 Geschäftslogik .....	6
4.6 Pflichtenheft .....	6
<b>5. Implementierungsphase .....</b>	<b>6</b>
5.1 Iterationsplanung .....	6
5.2 Implementierung der Datenstrukturen.....	7
5.3 Implementierung der Geschäftslogik.....	7
5.4 Implementierung der Benutzeroberfläche .....	8
<b>6 Abnahme- und Einführungsphase .....</b>	<b>9</b>
6.1 Abnahme durch den Fachbereich .....	9

---

6.2 Deployment und Einführung .....	9
<b>7 Dokumentation .....</b>	<b>9</b>
<b>8 Fazit.....</b>	<b>10</b>
8.1 Soll-/Ist-Vergleich.....	10
8.2 Gelerntes .....	10
8.3 Zukunft des Projekts .....	10
<b>A Anhang.....</b>	<b>i</b>
A.1 Detaillierte Zeitplanung .....	i
A.2 Verwendete Ressourcen .....	ii
Personal .....	ii
Hardware.....	ii
Software .....	ii
A.3 Use-Case-Diagramm.....	iii
A.4 Lastenheft (Auszug) .....	iv
Anforderungen .....	iv
A.5 Nutzwertanalyse zur Auswahl der Zielplattform.....	v
A.6 Nutzwertanalyse zur Auswahl der Programmiersprache.....	v
A.7 Pflichtenheft (Auszug) .....	vi
A.8 Screenshot GPS-Fehlermeldung .....	vi
A.9 Entwurf der Benutzeroberfläche .....	vii
A.10 Screenshot Hauptmenü .....	viii
A.11 Screenshot Gruppenerstellung/-einstellungen.....	viii
A.12 Screenshot Gruppenbeitritt .....	ix
A.13 Screenshot Listenansicht .....	ix
A.14 Ausschnitt aus der Kundendokumentation .....	x
A.15 Ausschnitt aus der Entwicklerdokumentation .....	xi

---

## Abbildungsverzeichnis

---

Abbildung 1: Use-Case-Diagramm .....	iii
Abbildung 2: Screenshot GPS-Fehlermeldung .....	vi
Abbildung 3: Screenshot Hauptmenü .....	viii
Abbildung 4: Screenshot Gruppenerstellung/-einstellungen .....	viii
Abbildung 5: Screenshot Gruppenbeitritt .....	ix
Abbildung 6: Screenshot Listenansicht .....	ix

## Tabellenverzeichnis

---

Tabelle 1: Grobe Zeitplanung .....	2
Tabelle 2: Kostenaufstellung .....	4
Tabelle 3: Vorkalkulation .....	4
Tabelle 4: Soll-/Ist-Vergleich .....	10

## Abkürzungsverzeichnis

---

API	
Application Programming Interface	7
CSS	
Cascading Style Sheets	8
DOM	
Document Object Model	8
GPS	
Global Positioning System	5
HTML	
Hypertext Markup Language	8
IHK	
Industrie- und Handelskammer	10
IP	
Internet Protocol	9
JSON	
JavaScript Object Notation	7
MIT	
Massachusetts Institute of Technology	2
N7	
NEO 7EVEN GmbH	1
PKW	
Personenkraftwagen	1
REST	
Representational State Transfer	7
SASS	
Syntactically Awesome Stylesheets	8

## Quellen

---

VueJS – The Progressive JavaScript Framework

<https://vuejs.org/>

[Stand: 10.04.2017]

SocketIO – How to write a chat application

<https://socket.io>

[Stand: 10.04.2017]

AngularJS – Superheroic JavaScript MVW Framework

<https://vuejs.org/>

[Stand: 09.04.2017]

SCSS – CSS with superpowers

<http://sass-lang.com/>

[Stand: 12.04.2017]

JavaScript Object Notation (JSON)

[https://de.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://de.wikipedia.org/wiki/JavaScript_Object_Notation)

[Stand: 12.04.2017]

Begriff „Echtzeit“

<https://de.wikipedia.org/wiki/Echtzeit>

[Stand: 14.04.2017]

Listenverkaufspreis errechnen

<https://www.controllingportal.de/Fachinfo/Kostenrechnung/Selbstkostenkalkulation.html>

[Stand: 14.04.2017]

Auth0 – How it works

<https://auth0.com/how-it-works>

[Stand: 14.04.2017]

### 1 Einleitung

---

Die folgende Projektdokumentation schildert den Ablauf des IHK-Abschlussprojektes, welches der Autor im Rahmen seiner Ausbildung zum Fachinformatiker Fachrichtung Anwendungsentwicklung durchgeführt hat.

Ausbildungsbetrieb ist die NEO 7EVEN GmbH (N7), ein Kleinunternehmen mit Standort in Siegen. Zurzeit beschäftigt N7 20 Mitarbeiter<sup>1</sup>, wird aber den Betrieb zum 31.03.2017 einstellen.<sup>2</sup>

N7 hat sich auf die Entwicklung von Softwareprodukten zur Erstellung von Katalogen spezialisiert. Ergänzend werden Kundenprojekte aller Art realisiert.

#### 1.1 Projektbeschreibung

Der Kunde für dieses Projekt ist der Erndtebrücker Handballclub 1978 e.V. (EHC), ein Handballverein in Siegen-Wittgenstein, welcher jede Saison sowohl Heimspiele als auch Auswärtsspiele absolvieren muss.

Das bedeutet für die Spieler, dass vor jedem Auswärtsspiel kleine Fahrgemeinschaften gebildet werden müssen da der EHC aus finanziellen Gründen keinen Vereinsbus für die Anfahrten stellen kann. Der Konvoi fährt dann zwar gleichzeitig in Erndtebrück los, jedoch verlieren sich die einzelnen PKWs oft auf dem Weg zum Ziel. Wenn sich die PKWs verloren haben, müssen die vorderen PKWs des Konvois anhalten und auf die zurückgebliebenen PKWs warten oder sie via Handy kontaktieren.

Der EHC möchte deshalb auf eine Software zurückgreifen, um die Abstände zwischen den einzelnen Fahrzeugen der Teammitglieder im Konvoi bestimmen zu können.

#### 1.2 Projektziel

Ziel des Projekts ist die Bestimmung der Abstände zwischen den Fahrzeugen innerhalb des Konvois, sowie die Bereitstellung der Abstände für die einzelnen Nutzer. Das Produkt soll von mobilen Endgeräten verwendbar sein, eine Verwirklichung als App oder Webseite ist daher naheliegend. Die Anzeige der einzelnen Teilnehmer eines Konvois sowie die Entfernung der einzelnen Fahrzeuge zueinander soll als Liste ausgegeben werden. Es soll die Möglichkeit geben Gruppen zu erstellen, zu betreten, zu verlassen und das Ziel der Gruppe festzulegen. Durch das Produkt sollen die Anfahrten im Konvoi bei Auswärtsspielen vereinfacht werden.

#### 1.3 Projektumfeld

Auftraggeber ist der EHC welcher momentan mit drei Mannschaften in verschiedenen Handballligen aktiv ist. Jede der drei Mannschaften bestreitet auch Auswärtsspiele, wodurch fast jedes Wochenende, während einer laufenden Saison (zwischen Oktober und Mai), ein Auswärtsspiel ist. An diesen Spieltagen soll die Software angewandt werden. Jeder der Spieler und Trainer besitzt ein Smartphone worüber die Software aufgerufen werden kann.

Um den Anforderungen des EHCs gerecht zu werden, ist eine intensive Rücksprache sowie Kommunikation mit dem Verein zwingend erforderlich.

#### 1.4 Projektbegründung

Aufgrund des nicht vorhandenen Mannschaftsbusses des Vereins müssen die Spieler sich selbst organisieren und kleine Fahrgruppen vor jedem Auswärtsspiel bilden. Oft verlieren sich die einzelnen PKWs auf dem Weg zum Ziel. Passiert das müssen die vorderen PKWs des Konvois

---

<sup>1</sup> Stand 01.03.2017

<sup>2</sup> Grund ist die Firmenschließung



## 2 Projektplanung

anhalten und auf die zurückgebliebenen PKWs warten oder sie via Handy kontaktieren. Das kostet Zeit und belastet die Spieler zusätzlich.

Aufgrund dieses Problems wurde NEO 7EVEN damit beauftragt eine Lösung zu erstellen, welche auf mobile Endgeräte angepasst ist und die Entfernungen der Fahrzeuge zueinander anzeigen kann.

### 1.5 Projektschnittstellen

Um die Distanz zwischen den PKWs korrekt berechnen zu können, wird die Google Maps API verwendet. Diese kann anhand der Breiten- und Längengrade den Standort der Fahrzeuge feststellen und die zu fahrende Distanz zwischen den PKWs ausgeben. Als Schnittstelle zwischen Client und Server wird SocketIO für eine Echtzeitverbindung verwendet.

### 1.6 Projektabgrenzung

Da der Projektumfang begrenzt ist, soll eine Anzeige des Standortes auf einer Karte o.ä. nicht Bestandteil des Abschlussprojektes sein.

## 2 Projektplanung

### 2.1 Projektphasen

Insgesamt 70 Stunden standen dem Autoren zur Verfügung. Bei Projektbeginn wurden diese auf verschiedene Phasen verteilt, die während der Softwareentwicklung durchlaufen werden. Es wurde zunächst eine grobe Zeitplanung erstellt welche die Zeiten für die Hauptphasen plant, festgehalten in Tabelle 1: [Grobe Zeitplanung](#). Des Weiteren können diese Phasen in kleinere Unterpunkte zerlegt werden. Eine detaillierte Übersicht befindet sich im Anhang [A.1: Detaillierte Zeitplanung](#) auf Seite [i](#).

Projektphase	Geplante Zeit
Analysephase	5 h
Entwurfsphase	2 h
Implementierungsphase	40 h
Abnahme und Veröffentlichung	5 h
Erstellen der Dokumentation	18 h
Gesamt	70 h

Tabelle 1: Grobe Zeitplanung

### 2.2 Ressourcenplanung

Alle Ressourcen die in diesem Projekt verwendet wurden sind im Anhang [A.2: Verwendete Ressourcen](#) auf Seite [ii](#) aufgelistet. Dabei wurden sowohl Hard- und Softwareressourcen als auch das Personal aufgelistet. Bei der Auswahl der Software wurde darauf geachtet, dass sie kostenfrei zur Verfügung steht (z.B. MIT-Lizenz) oder N7 die Lizenzen bereits besitzt. So sollen die anfallenden Projektkosten möglichst gering gehalten werden.

### 2.3 Entwicklungsprozess

Zunächst musste ein Entwicklungsprozess gefunden werden, welcher die Vorgehensweise definiert, nach der die Umsetzung erfolgen soll. Für dieses Projekt wurde vom Autor ein agiler Entwicklungsprozess ausgewählt. Dabei soll in Anlehnung an das Vorgehensmodell Scrum

## 3 Analysephase

---

gearbeitet werden. Iteratives Durchlaufen der Projektphasen und Rücksprachen mit den Kunden sind die Hauptmerkmale, die diesen Entwicklungsprozess auszeichnen.

Mit kurzen Iterationszyklen wird eine flexible Verwirklichung der Anforderungen ermöglicht, so können dem Verein relativ früh Resultate geliefert werden. Deshalb wurde bei der Projektplanung in Abschnitt 2.1 (Projektphasen) für die Entwurfsphase vergleichsweise wenig Zeit eingeplant, die Teile dieser Phase ergeben sich erst im Laufe der Entwicklung.

Durch stetige Absprache mit dem Verein und dem daraus erhaltenen Feedback kann die Software an die Ansprüche des Kunden angepasst werden und nachträgliche Wünsche können berücksichtigt werden. Nebenbei kann der Verein sich bereits mit der Software vertraut machen, das spart Zeit bei der Projektabnahme und -einführung. Deshalb wurde für die Phase der Projektplanung ebenfalls relativ wenig Zeit einkalkuliert.

## 3 Analysephase

---

### 3.1 Ist-Analyse

Wie bereits im Abschnitt 1.1 (Projektbeschreibung) beschrieben wurde, hat der Verein momentan noch keine Software oder ein bestimmtes System um den Abstand der Fahrer innerhalb des Konvois zu bestimmen. Das heißt, ist ein Fahrzeug des Konvois zurück geblieben müssen die anderen PKWs anhalten und einer der Spieler kontaktiert den vermissten Fahrer. Das kostet Zeit und gegebenenfalls auch Geld wegen der Telefonate.

#### 3.2.1 „Make or Buy“-Entscheidung

Es gibt kein Softwareprodukt welches den Bedarf des EHC abdeckt, der Bedarf ist zwar von einigen Vereinen gegeben, wird allerdings nicht vom Markt gesättigt. Bisher wurde meist das Produkt „Google Maps<sup>3</sup>“ genutzt um den eigenen Standort und die Entfernung zum Ziel zu bestimmen, dort kann jedoch nicht der Standort der anderen Fahrer des Konvois eingesehen werden.

#### 3.2.2 Projektkosten

In diesem Abschnitt sollen die Kosten, welche für das Projekt anfallen berechnet werden. Dabei werden sowohl Personalkosten als auch die Kosten genutzter Ressourcen (Hard- und Software, Arbeitsplatz etc.) berechnet. Da Personalkosten nicht herausgegeben werden dürfen, ist die Berechnung auf Stundensätze von 8€ für einen Auszubildenden und 23€ für einen Mitarbeiter festgelegt. Für die Nutzung der Ressourcen wurde ein pauschaler Stundensatz von 12€ genommen. Es wurden verschiedene externe Bibliotheken verwendet welche kostenlos<sup>4</sup> zur Verfügung stehen und alle Funktionen bieten die gebraucht werden.

---

<sup>3</sup> <https://www.google.de/maps>

<sup>4</sup> Der Stand vom 29.03.17: <https://auth0.com/pricing>

## 3 Analysephase

Die anfallenden Gesamtkosten und die jeweiligen Teilkosten für die einzelnen Vorgänge des Projekts lassen sich aus der Tabelle 2: [Kostenaufstellung](#) entnehmen.

Vorgang	Mitarbeiter	Zeit	Personal	Ressourcen	Gesamt
Entwicklungskosten	1 x Auszubildender	70 h	560,00€	840,00€	1.400,00€
Fachgespräch	2 x Mitarbeiter	5 h	230,00€	120,00€	350,00€
Code-Review	1 x Mitarbeiter	3 h	69,00€	36,00€	105,00€
Abnahme	2 x Mitarbeiter	1 h	46,00€	24,00€	70,00€
			<b>Projektkosten gesamt</b>		<b>1.925,00€</b>

Tabelle 2: Kostenaufstellung

### 3.2.3 Amortisationsdauer und Finanzierung

Nun muss bestimmt werden ab welchem Zeitpunkt sich das Projekt amortisiert. Mit dem Amortisationswert kann bestimmt werden, ob die Verwirklichung eines Projekts auf Dauer aus wirtschaftlicher Sicht rentabel für die Firma ist. Dieser Wert wird berechnet indem man die Anschaffungskosten durch die Einnahmen, die durch das Produkt durchschnittlich pro Jahr eingenommen werden, dividiert.

Das Produkt wird dem Kunden mit 2.476,65€ in Rechnung gestellt, dieser Betrag ergibt sich aus dem Listenverkaufspreis der Tabelle 3: [Vorkalkulation](#). In diesem Betrag sind bereits ein Gewinnzuschlag von 192,50€, ein mögliches Kundenskonto von maximal 5% und ein möglicher Kundenrabatt von maximal 10% berücksichtigt.

Beschreibung	Betrag
Selbstkosten	1.925,00€
+ Gewinnzuschlag (10%)	192,50€
= Barverkaufspreis	2.117,50€
+ Kundenskonto (5%)	111,45€
= Zielverkaufspreis	2.228,95€
+ Kundenrabatt (10%)	247,66€
<b>= Listenverkaufspreis</b>	<b>2.476,61€</b>

Tabelle 3: Vorkalkulation

Dadurch ergibt sich, dass das Produkt ab dem ersten Verkauf amortisiert ist. Ohne Inanspruchnahme des Kundenskontos und des Kundenrabatts bringt das Produkt bereits einen Gewinn von 551,61€<sup>5</sup> ein. Ein kleiner Mannschaftsbus für den EHC schlägt bereits mit ca.

<sup>5</sup> Gewinnzuschlag + Kundenskonto + Kundenrabatt

## 4 Entwurfsphase

---

87.000€<sup>6</sup> zu buche. Somit kann das Projekt aus wirtschaftlicher Sicht für beide Seiten als sinnvoll eingestuft werden.

### 3.3 Nicht-monetäre Vorteile

Die Realisierung des Projekts ist bereits durch die Wirtschaftlichkeitsanalyse ausreichend gerechtfertigt, deshalb wird an dieser Stelle auf eine detaillierte Analyse nicht-monetärer Vorteile verzichtet.

Nicht-monetäre Vorteile des Projekts wären z.B. Kundendaten über Standorte, Kundenprofildaten, neue Kenntnisse über GPS-Technologien sowie eine Steigerung der Popularität der Firma.

### 3.4 Anwendungsfälle

Im Laufe der Analysephase ist ein Use-Case-Diagramm erstellt worden, welches alle Anwendungsfälle nach dem Login darstellt. Es befindet sich im Anhang [A.3: Use-Case-Diagramm](#) auf Seite [iii](#). Alle aus der Sicht des Endanwenders benötigten Funktionen werden dort abgebildet.

### 3.5 Lastenheft/Fachkonzept

Es wurde in der Analysephase in Zusammenarbeit mit dem EHC erstellt und umfasst alle Anforderungen des Auftraggebers an die Anwendung. Ein Auszug aus dem Lastenheft befindet sich im Anhang [A.4: Lastenheft \(Auszug\)](#) auf Seite [iv](#).

## 4 Entwurfsphase

---

### 4.1 Zielplattform

In Abschnitt 1.2 (Projektziel) wurde bereits erwähnt, dass das Abschlussprojekt für mobile Endgeräte erstellt werden soll, nach einer Nutzwertanalyse wurde beschlossen, dass das Produkt als Webseite verwirklicht werden soll. Die Nutzwertanalyse ist im Anhang [A.5: Nutzwertanalyse zur Auswahl der Zielplattform](#) auf Seite [v](#) einzusehen. Dort befinden sich auch die einzelnen Kriterien mit deren Gewichtung und den Bewertungen für die verschiedenen Plattformen. Für die Gewichtung und die Bewertung werden Werte zwischen 1 und 5 verwendet. Die Gewichtung wird jeweils mit der Bewertung multipliziert. Der Quotient aus der gewichteten Gesamtbewertung und der Summe der Gewichtungen ergibt den Nutzwert der jeweiligen Plattform. Der geringste Wert, welcher erreicht werden könnte, wäre 1, der höchste 5.

Auch die Programmiersprache anhand welcher das Projekt verwirklicht werden soll, wurde mit einer Nutzwertanalyse bestimmt. Im Anhang [A.6 Nutzwertanalyse zur Auswahl der Programmiersprache](#) auf Seite [v](#) kann diese und die Kriterien, anhand deren die Analyse durchgeführt wurde, eingesehen werden. Das Verfahren bleibt dabei gleich. Aufgrund dieser Bewertung fiel die Entscheidung zugunsten der Programmiersprache JavaScript.

### 4.2 Architekturdesign

Da das Projekt komplett von Anfang an entwickelt wird muss sowohl der Server als auch der Client implementiert werden. Bei der Planung wurden die Entscheidungen aus [4.1 Zielplattform](#) bereits berücksichtigt.

Der Client wird auf Basis der etablierten Webentwicklungssprachen HTML, CSS und JavaScript umgesetzt. Um einen möglichst übersichtlichen Programmcode zu erhalten und Fehlern durch falsche Verlinkungen von Variablen etc. vorzubeugen, wurde VueJS als JavaScript Framework herangezogen. VueJS wird sich um das automatische Anpassen der angezeigten Daten

---

<sup>6</sup> Nachweis siehe: [http://www.trucksout24.de/search/ger/detail.asp?vehicletype\\_id=6&vehicle\\_id=14829242](http://www.trucksout24.de/search/ger/detail.asp?vehicletype_id=6&vehicle_id=14829242)

## 5. Implementierungsphase

---

kümmern, sobald die entsprechenden Variablen geändert werden. Alternative Frameworks wären AngularJS und BackboneJS, welche beide die gleiche Funktionalität wie VueJS bieten, jedoch auf größere Projekte ausgelegt sind.

Damit sich die Nutzer authentifizieren können wurde eine Erweiterung namens Auth0 herangezogen. Diese Erweiterung regelt die Anmeldung des Nutzers und liefert anschließend die dazugehörigen Daten zurück.

Für die stetigen Updates der Distanz der Fahrzeuge zueinander muss eine WebSocket-Verbindung hergestellt werden über welche die Daten dann in Echtzeit übertragen werden. Dies wird mit SocketIO erreicht. SocketIO regelt den Datenfluss zwischen Server und Client, ohne unnötige Anfragen zu versenden.

Der Server wird mit NodeJS als Webserver auf Basis von JavaScript erstellt, der Webserver speichert die Daten nur während der Laufzeit, es wird somit keine Datenbank benötigt.

### 4.3 Entwurf der Benutzeroberfläche

Um das Produkt so anwenderfreundlich wie möglich zu gestalten, wurde eine klar strukturierte und vor allem einfache Benutzeroberfläche entwickelt. Mit Hilfe von Designs wurden die Ansichten der Applikation jeweils vordefiniert, dabei war der EHC als Kunde intensiv mit in die Entwurfsphase einbezogen.

Wichtig für den EHC ist, dass man auf einen Blick alle Fahrer sehen kann und eine klare Übersicht über die Fahrer in einem Konvoi und deren Entfernung zueinander hat. Um das zu erreichen ist eine Ansicht als Liste vereinbart worden, in welcher alle Mitglieder des aktuellen Konvois angezeigt werden.

### 4.4 Datenmodell

Da keine Datenbank für dieses Projekt benötigt wird, ist an dieser Stelle auf ein Entity-Relationship-Model (ER-Model) verzichtet worden. Die Anmeldedaten werden vom Anbieter Auth0 verwaltet und gespeichert.

### 4.5 Geschäftslogik

Aufgrund der testgetriebenen<sup>7</sup> Entwicklung ergeben sich die zu implementierenden Datenstrukturen erst während der Implementierungsphase. Deshalb ist eine Vorbestimmung der benötigten Daten und die Erstellung eines Diagramms zur Bestimmung des Datenflusses im Voraus nicht möglich.

### 4.6 Pflichtenheft

Das Pflichtenheft soll als Leitfaden zur Realisierung des Projekts dienen und beschreibt wie und womit das Projekt verwirklicht werden soll.

Im Anhang [A.7 Pflichtenheft](#) auf Seite [vi](#) befindet sich ein Auszug aus dem Pflichtenheft, welches der Autor auf Basis des Lastenhefts (vgl. dazu in Abschnitt 3.5 Lastenheft/Fachkonzept) und den Erkenntnissen der verschiedenen Analysen erstellt hat.

## 5. Implementierungsphase

---

### 5.1 Iterationsplanung

Zunächst wurde ein Iterationsplan erstellt in dem die einzelnen Iterationsschritte und deren Reihenfolge definiert wurden. Während einer Iteration wurde jeweils eine bestimmte

---

<sup>7</sup> Testgetrieben heißt: es werden für Teile der Software Tests geschrieben, welche nach der Implementation dieses Teils nicht fehlschlagen sollen.

## 5. Implementierungsphase

---

Funktionalität implementiert, welche dann am Ende der Iteration dem EHC präsentiert wurde. Diese Vorgehensweise entspricht dem im Abschnitt 2.3 (Entwicklungsprozess) beschriebenen Prinzip der agilen Softwareentwicklung.

### 5.2 Implementierung der Datenstrukturen

Da die Daten bei diesem Produkt im Laufzeitspeicher des Servers gespeichert werden und nicht in einer Datenbank, gibt es keine Datenstrukturen welche vordefiniert werden können. Viel mehr ergeben sich Datenstrukturen während der Implementierung der Geschäftslogik aufgrund der testgetriebenen Entwicklung. Solche Datenstrukturen sind Variablen welche der NodeJS-Server während der Laufzeit speichert und verwaltet z.B. eine Liste der Benutzer in einer Gruppe.

### 5.3 Implementierung der Geschäftslogik

Die Implementierung der Geschäftslogik stellt die Kernaufgabe des gesamten Projekts dar, deshalb wird diese hier genauer erläutert. Eine geeignete Entwicklungsumgebung ist von großer Bedeutung, wenn das Projekt möglichst komfortabel durchgeführt werden soll. Der Autor hat für das Projekt die Entwicklungsumgebung Atom verwendet, da diese Entwicklungsumgebung kostenlos<sup>8</sup> und auf Webentwicklung abgestimmt ist.

Es wurde mit der Entwicklung des Servers begonnen, da der Server (welcher auf NodeJS basiert) sich um die Verarbeitung und Verteilung der Daten kümmert. Der Server soll die Daten der Nutzer verwalten und ihnen dann in Echtzeit antworten, wenn es neue Daten gibt (z.B. wenn ein anderer Nutzer sich bewegt). Dies lässt sich am besten mit SocketIO umsetzen, einer Erweiterung für NodeJS, die es ermöglicht den Datenfluss zwischen Server und Client zu regeln und kostenlos<sup>9</sup> ist. Der NodeJS-Server kümmert sich jedoch nicht um die Berechnungen der Entfernungen der einzelnen Teilnehmer zueinander, sondern dient lediglich der Gruppenbildung, Zwischenspeicherung der Daten und der Datenverteilung nach einer Änderung. Die Berechnung<sup>10</sup> wurde auf die Google Maps API verlagert und durch den Client aufgerufen. Da die Entfernungen jeweils von der Position des Clients aus zu allen anderen Clients berechnet werden sollen, wurde dieser Lösungsweg gewählt.

Die eigentliche Berechnung der Daten geschieht somit beim Client mit JavaScript, welches die Daten via SocketIO-Anbindung annimmt und sie dann an die Google Maps API sendet um die Entfernungen zu den anderen Teilnehmern im Konvoi zu erhalten. Diese Änderungen werden dann sofort<sup>11</sup> bei allen anderen Clients angezeigt.

Die Google Maps API verfügt über eine REST (Representational State Transfer) API und wird via JavaScript mit den entsprechenden Parametern aufgerufen (Positionsdaten, Berechnungsoption), die API gibt dann das Ergebnis mit Standortinformationen und der Distanz zum Ziel in einer JSON (JavaScript Object Notation) Ausgabe zurück. Das Ergebnis kann nun mit Hilfe von JavaScript aus der JSON Ausgabe entnommen und weiterverwendet werden. Die Google Maps API bietet 2500 dieser Berechnungen kostenlos<sup>12</sup> pro Tag an, diese sollten reichen um den Bedarf für den EHC zu decken.

---

<sup>8</sup> Nachweis siehe: <https://atom.io/>

<sup>9</sup> Nachweis siehe: <https://socket.io/>

<sup>10</sup> Berechnung der Distanz zwischen den Fahrzeugen

<sup>11</sup> Sofort heißt ohne Verzögerungen abgesehen von der Latenzzeit und den Berechnungszeiten

<sup>12</sup> Nachweis siehe: <https://developers.google.com/maps/documentation/geocoding/usage-limits>



## 5. Implementierungsphase

---

Um diese Berechnungen sofort im HTML/CSS<sup>13</sup>-Client anzuzeigen wurde die Bibliothek VueJS verwendet. Diese „verbindet“<sup>14</sup> JavaScript Variablen mit dem entsprechenden DOM-Element<sup>15</sup> (Document Object Model) in HTML und ändert das DOM-Element sobald die Variable sich geändert hat. Die Bibliothek AngularJS funktioniert ähnlich wie VueJS, ist allerdings auf größere Projekte ausgelegt, weshalb die kostenlose<sup>16</sup> Bibliothek VueJS verwendet wurde. Um ein ähnliches Ergebnis ohne die Bibliothek zu erreichen, wäre eine eigene Lösung dieses „binding“-Vorgangs von Nöten welche den Quellcode unter Umständen unsauberer und unübersichtlicher erscheinen lässt.

Nachdem nun die Verarbeitung und Berechnung der Daten fertiggestellt waren, musste noch eine Identifikation für den Kunden eingebaut werden, um die einzelnen Nutzer voneinander unterscheiden zu können. Da Anmeldeformulare (Logins) oft zu Sicherheitsproblemen führen können und eine Datenbank benötigt wird, welche Extrakosten und Aufwand mit sich bringt, wurde auf eine eigene Implementation verzichtet und stattdessen die externe Lösung Auth0 herangezogen. Auth0 ist kostenlos<sup>17</sup> und sorgt mit einer einfachen API für eine simple Implementierung eines Anmeldevorgangs, welcher nach erfolgreichem durchlaufen Infos wie Namen, EMail etc. an den Server weiter gibt. Auf Basis dieser Daten kann eine eindeutige Identifikation erfolgen.

Sobald ein neuer Teilnehmer einer Gruppe beitrifft wird die Position von seinem Handy via GPS ermittelt und die Positionsdaten an den Server übermittelt, welcher dann an alle Mitglieder des Konvois ein Update sendet, damit diese die neuen Daten zur Berechnung der Entfernung verwenden werden können. Probleme können hier auftreten, wenn das GPS des Handys nicht für den Browser freigegeben ist. Um diese Probleme zu umgehen überprüft die Anwendung nach dem Login ob die GPS-Daten zur Verfügung stehen. Falls nicht wird das Problem dem Benutzer erläutert und ihm eine entsprechende Fehlerseite angezeigt, ein Screenshot ist im Anhang [A.8 Screenshot GPS-Fehlermeldung](#) auf Seite [vi](#) zu sehen.

Sobald sich die Positionsdaten eines Teilnehmers ändern, werden diese erneut an den Server übermittelt. Infolgedessen sendet der Server ein Update der Daten an die Clients. Dieser Vorgang wird ebenfalls durchlaufen, wenn der Gruppenleiter das Ziel ändert oder ein Teilnehmer die Gruppe verlässt. Alle Clients sind somit bei einer bestehenden Verbindung zum Server immer auf dem aktuellen Stand. Bei all diesen Übermittlungen wurde SocketIO verwendet, um die Anfragen in Echtzeit<sup>18</sup> zu senden und zu empfangen.

### 5.4 Implementierung der Benutzeroberfläche

Bei der Implementierung der Benutzeroberfläche wurde lediglich mit SASS (Syntactically Awesome Stylesheets) und HTML gearbeitet, SASS wird von einem entsprechenden Compiler wieder in CSS umgewandelt, es sorgt dafür das der Quellcode für die CSS-Dateien übersichtlicher und anpassbarer wird indem SASS u.a. die Definition von Variablen erlaubt. Im Anhang [A.9 Entwurf der Benutzeroberfläche](#) auf Seite [vii](#) ist ein erster Entwurf der Oberfläche einzusehen.

---

<sup>13</sup> Die Programmierung erfolgte in SASS welches automatisch in CSS kompiliert wird.

<sup>14</sup> Im englischen wird oft „binding“ verwendet um diesen Vorgang zu umschreiben

<sup>15</sup> Vgl. [https://www.w3schools.com/jsref/dom\\_obj\\_all.asp](https://www.w3schools.com/jsref/dom_obj_all.asp)

<sup>16</sup> Nachweis siehe: <https://vuejs.org/>

<sup>17</sup> Kostenlos bis 7000 aktive Benutzer - Nachweis siehe: <https://auth0.com/pricing>

<sup>18</sup> Nachweis: <https://de.wikipedia.org/wiki/Echtzeit>

## 6 Abnahme- und Einführungsphase

---

Bei dem Entwurf wurde vor allem darauf geachtet, dass die gesamte Applikation für den Nutzer einfach und intuitiv zu bedienen ist und die relevanten Daten schnell ablesbar sind. Durch den Fokus auf ein Design für mobile Endgeräte sind diese Punkte besonders wichtig, da der Anzeigebildschirm kleiner ist als bei einer Desktopanwendung. Das lässt sich vor allem in dem schlicht und konstant gestalteten Menü als auch in der Listenansicht der Fahrer eines Konvois erkennen. Wären diese Elemente nicht so übersichtlich gestaltet worden, könnte der Nutzer die Informationen nur umständlich oder falsch entnehmen. Die Listenansicht zeigt dem Nutzer so auf einen Blick u. a. wo das erste und wo das hintere Fahrzeug ist. Auch die Entfernung zum gewählten Ziel ist auf einen Blick zu erkennen.

Die farbliche Gestaltung des Clients sollte modern und schlicht wirken, deshalb hat sich der Autor für die in Anhang [A.9 Entwurf der Benutzeroberfläche](#) dargestellte Farbkombination entschieden, welche im gesamte Produkt Verwendung fand.

Screenshots der Menüs sind ab Anhang [A.10](#) bis [A.12](#) auf Seite [viii](#) und [ix](#) zu finden und Screenshots der Listenansicht, in welcher die einzelnen Distanzen angezeigt werden, sind im Anhang [A.13 Screenshot Listenansicht](#) auf Seite [ix](#) zu finden.

## 6 Abnahme- und Einführungsphase

---

### 6.1 Abnahme durch den Fachbereich

Die Endabnahme erfolgte nach der Fertigstellung. Aufgrund der agilen Entwicklungsweise wurde die Software allerdings schon mehrfach mit dem EHC getestet und besprochen. Dadurch war der Kunde bereits bei der Abnahme mit der Funktionsweise und Oberfläche der Applikation vertraut. Desweiteren konnten frühzeitig Probleme und Anregungen behandelt werden, so dass bei der Endabnahme keine Probleme oder Hindernisse aufkamen und der Einführung der Anwendung nichts mehr im Weg stand. Außerdem wurden Code-Reviews<sup>19</sup> mit anderen Entwicklern zur Qualitätssicherung durchgeführt.

### 6.2 Deployment und Einführung

Die Veröffentlichung des Projekts wurde mit unserem firmeninternen Server verwirklicht. Via IP-Adresse und Port kann man den Webserver erreichen. Diese IP-Adresse und der dazugehörige Port wurden zum Testen an andere Entwickler und an den Kunden weitergegeben. Das Produkt erfordert somit allerdings eine konstante Verbindung zum Internet via Webbrowser. Beim Webbrowser lässt sich die IP-Adresse einfach als Zieladresse eingeben. In Zukunft wäre eine Lösung via Domain denkbar, um die Erreichbarkeit und das Merken der Adresse zu erleichtern. Bei Benutzerschulungen wurde den Spielern des EHC der Umgang mit dem Webbrowser und der Applikation eingehend gezeigt, sowie die IP-Adresse und der dazugehörige Port zum Zugriff mitgeteilt.

## 7 Dokumentation

---

Die gesamte Dokumentation besteht aus zwei Teilen: der Projektdokumentation und der Kundendokumentation. In der Projektdokumentation wird vom Autoren beschrieben, welche Phasen während der Umsetzung des Projekts durchlaufen wurden. Die Kundendokumentation soll dem Benutzer den Umgang mit dem Produkt näher bringen und ihm Einzelheiten erklären.

---

<sup>19</sup> „Code-Review“ ist nach Scrum die Bezeichnung für ein Meeting in welchem der Code geprüft und überarbeitet wird. Dies geschieht meist in Zusammenarbeit mit anderen Entwicklern



## 8 Fazit

Ein Auszug aus dieser Dokumentation befindet sich im Anhang [A.14 Ausschnitt aus der Benutzerdokumentation](#) auf Seite x.

Auf eine Entwicklerdokumentation wurde verzichtet da bei NodeJS und JavaScript nicht mit komplexen Klassenstrukturen gearbeitet wurde, stattdessen wurde der Programmcode sorgfältig durch Kommentare dokumentiert um zukünftigen Entwicklern den Einstieg zu erleichtern, Nachweise sind im Anhang [A.15 Ausschnitt aus der Entwicklerdokumentation](#) auf Seite xi einzusehen.

## 8 Fazit

### 8.1 Soll-/Ist-Vergleich

Wenn man das Produkt welches bei diesem IHK-Abschlussprojekt entstanden ist mit den Anforderungen gemäß des Pflichtenhefts vergleicht kann man festhalten das alle Anforderungen erfüllt wurden. Zu Beginn des Projekts im Abschnitt 2.1 (Projektphasen) wurde ein Projektplan erstellt welcher auch eingehalten werden konnte. Der Autor hat in der Tabelle 4: [Soll-/Ist-Vergleich](#) die benötigten mit den geplanten Zeiten für die einzelnen Projektphasen gegenübergestellt. Es wurde nur leicht vom ursprünglichen Zeitplan abgewichen. Die Abweichungen kann man aus der Spalte „Differenz“ in der Tabelle 4: [Soll-/Ist-Vergleich entnehmen](#). Die Differenzen welche sich ergaben konnten miteinander kompensiert werden, somit ist das Projekt in dem von der IHK festgelegten Zeitrahmen von 70 Stunden umgesetzt worden.

Projektphase	Soll	Ist	Differenz
Analysephase	5 h	5 h	0h
Entwurfsphase	2 h	3,5 h	+ 1,5 h
Implementierungsphase	40 h	40,5 h	+ 0,5 h
Abnahme und Veröffentlichung	5 h	5 h	0h
Erstellen der Dokumentation	18 h	16 h	- 2 h
Gesamt	70 h	70 h	0 h

Tabelle 4: Soll-/Ist-Vergleich

### 8.2 Gelerntes

Der Autor konnte bei der Planung und Durchführung dieses Projekts viele für die Zukunft wertvolle Erfahrungen sammeln. Besonders im Bereich der Entwicklung von Webanwendungen für mobile Endgeräte und im Umgang und der Planung mit Kunden konnte viel dazugelernt werden. Vor allem die stetige Absprache mit dem EHC während der testgetriebenen Entwicklung vermittelte wertvolle Kommunikationsfähigkeiten und förderte das planungstechnische Talent enorm. Auch das Zusammenspiel von Google Maps API, VueJS und SocketIO mit GPS-Daten hat viele wertvolle Erfahrungen mit sich gebracht.

### 8.3 Zukunft des Projekts

Trotz der erfolgreichen Realisierung aller Anforderungen des Lastenhefts gibt es dennoch einige zusätzliche Vorschläge des EHC und von anderen Entwicklern wie das Projekt verbessert werden könnte. Durch die Modularität der Anwendung ist eine gute Wertbarkeit und Erweiterbarkeit sichergestellt und lässt durchaus Raum für solche Anfragen.

## A Anhang

### A.1 Detaillierte Zeitplanung

<b>Analysephase</b>		<b>5 h</b>
Ist-Analyse	2 h	
Nutzwertanalysen und Wirtschaftlichkeitsanalyse	2 h	
Erstellung des Lastenhefts mit dem Kunden	1 h	
<b>Entwurfsphase</b>		<b>4 h</b>
Zielplattform und Programmiersprache festlegen	0,5 h	
Entwurf der Benutzeroberfläche (UI)	0,5 h	
Planung der Geschäftslogik	1,5 h	
Erstellung des Pflichtenhefts	1,5 h	
<b>Implementierungsphase</b>		<b>40 h</b>
Anlegen der Ordnerstrukturen	1 h	
Vorbereitung der Entwicklungsumgebung	2 h	
Implementierung des Servers	25 h	
Implementierung der Benutzeroberfläche	10 h	
Anbindung an die Schnittstellen (Google Maps API)	2 h	
<b>Abnahme und Einführung</b>		<b>5 h</b>
Abnahme durch den Fachbereich	1 h	
Abnahme durch den Kunden	2 h	
Einführung in die Anwendung der Applikation	2 h	
<b>Erstellen der Dokumentation</b>		<b>16 h</b>
Erstellung der Projektdokumentation	14 h	
Erstellung der Benutzerdokumentation	2 h	
<b>Gesamt</b>		<b>70h</b>

## A.2 Verwendete Ressourcen

### Personal

- Mitarbeiter aus dem Bereich Entwicklung: Review des Codes und Erstabnahme
- Entwickler (Autor) – Umsetzung des Projekts

### Hardware

- Büroarbeitsplatz mit zwei Bildschirmen und einem PC

### Software

- Windows 10 als Betriebssystem
- Atom IDE als Entwicklungsumgebung für HTML, CSS und JavaScript
- Git zur Versionierung
- NodeJS zum ausführen des NodeJS Servers
- NPM für die Installation von zusätzlichen Paketen für das Projekt
- SocketIO für eine Echtzeitverbindung
- Auth0 für den Umgang mit Logins
- GoogleAPI für Distanzberechnungen

## A.3 Use-Case-Diagramm

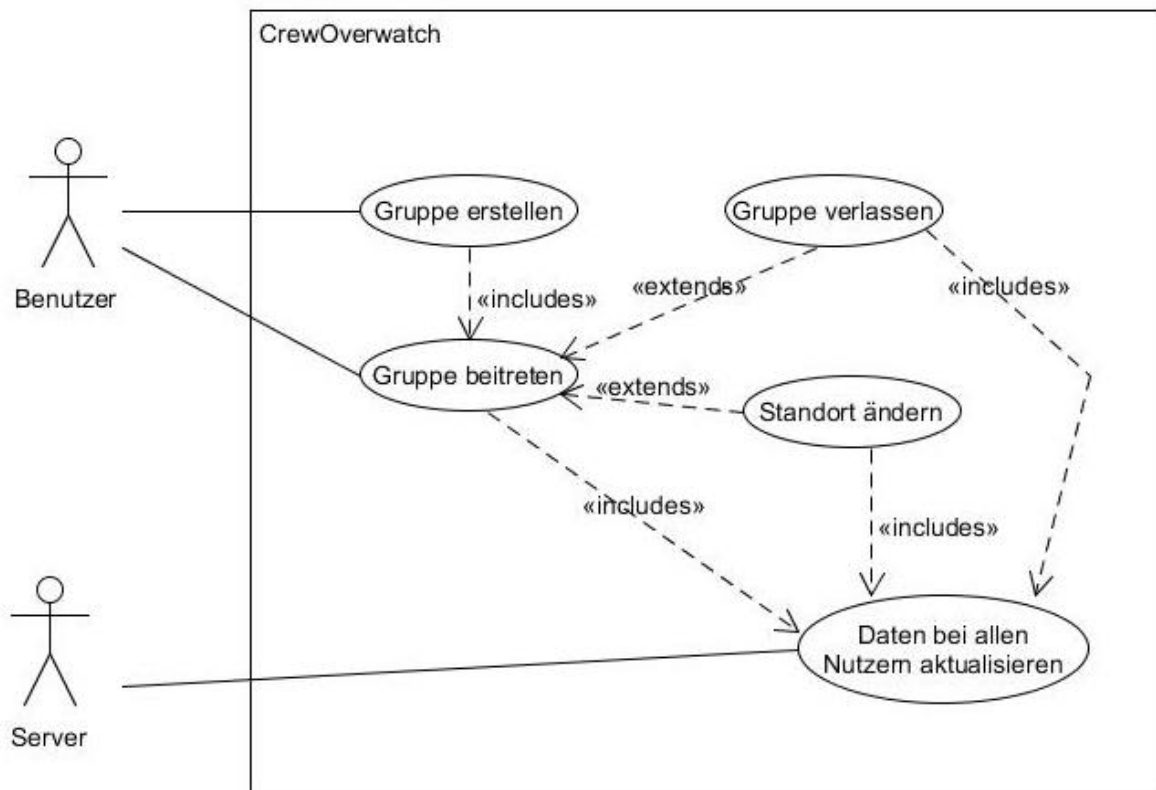


Abbildung 1: Use-Case-Diagramm

## A.4 Lastenheft (Auszug)

Im hier dargestellten Auszug aus dem Lastenheft werden die Anforderungen an das zu entwickelnde Produkt definiert.

[ ... ]

### Anforderungen

Von dem Produkt müssen folgende Anforderungen erfüllt werden:

- Das Produkt soll für mobile Endgeräte optimiert sein.
- Das Produkt muss die Möglichkeit einer Anmeldefunktion bieten.
- Das Produkt soll eine Funktion zum Erstellen und Verwalten einer Gruppe bereitstellen.
- In der Verwaltung der Gruppe soll es möglich sein die Gruppe zu verlassen und ein neues Ziel festzulegen
- Das Produkt muss den Mitgliedern eines Konvois erlauben Gruppen beizutreten und sie zu verlassen.
- Das Produkt soll in der Lage sein die Standorte der Mitglieder eines Konvois zu bestimmen und deren Entfernung zueinander und zum Ziel zu bestimmen.

[ ... ]

## A.5 Nutzwertanalyse zur Auswahl der Zielplattform

Eigenschaft	Gewichtung	Android App (bew. <sup>20</sup> )	Website (bew.)	Android App (gew. <sup>21</sup> )	Webseite (gew.)
Kenntnisstand (Entwickler)	4	2	5	8	20
Performance	3	4	2	12	6
Benutzeroberfläche	5	2	5	10	25
Fehleranfälligkeit	2	4	3	8	6
Betriebssystem Unabhängigkeit	5	1	5	5	25
Kosten	2	4	5	8	10
Gesamt:	25			59	108
Nutzwert:				<b>2,36</b>	<b>4,32</b>

## A.6 Nutzwertanalyse zur Auswahl der Programmiersprache

Eigenschaft	Gewichtung	C# (bew.)	Java (bew.)	Java-Script (bew.)	C# (gew.)	Java (gew.)	JavaScript (gew.)
Kenntnisstand (Entwickler)	5	1	2	5	5	10	25
Verfügbarkeit der benötigten Bibliotheken	3	2	2	4	6	6	12
Fehleranfälligkeit	2	5	4	2	10	8	4
Betriebssystem Unabhängigkeit	3	2	3	5	6	9	15
Kosten	2	2	4	5	4	10	8
Gesamt:	15				31	43	64
Nutzwert:					<b>~2.07</b>	<b>~2.87</b>	<b>~4.27</b>

<sup>20</sup> bew. = bewertet<sup>21</sup> gew. = gewichtet

## A.7 Pflichtenheft (Auszug)

[ ... ]

Umsetzung der Anforderungen ist wie folgt geplant:

- Die Optimierung für mobile Endgeräte erfolgt via CSS (SASS) als Webseite.
- Die Anmeldung im Produkt erfolgt über den Anbieter Auth0.
- Es werden Funktionen zum Erstellen, Beitreten und Verwalten einer Gruppe geben.
- Es wird möglich sein über die Einstellungen einer Gruppe dessen Ziel festzulegen.
- Es wird möglich sein über die Einstellungen einer Gruppe diese zu verlassen.

[ ... ]

## A.8 Screenshot GPS-Fehlermeldung

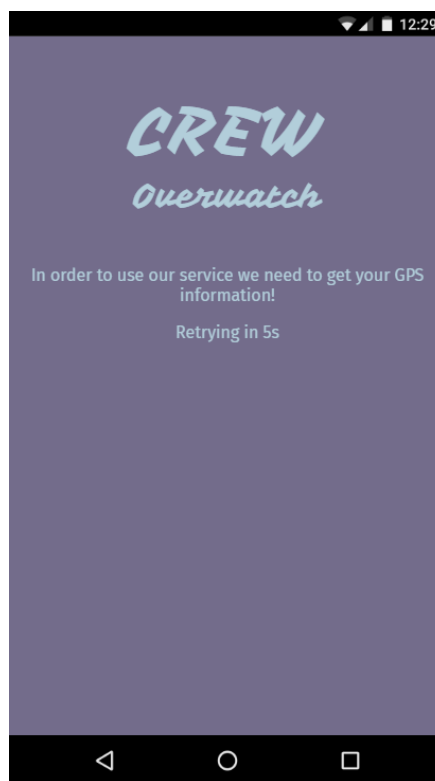






Abbildung 2: Screenshot GPS-Fehlermeldung

## A.9 Entwurf der Benutzeroberfläche

CrewID: 989898		
Ziel: Musterort		
	Max Mustermann	1,5km
	Musterstraße 13, Musterstadt	
	Laura Musterfrau	-
	Musterstraße 13, Musterstadt	
	Maik Mustermann	-1,2km
	Musterstraße 13, Musterstadt	
		



## A.10 Screenshot Hauptmenü

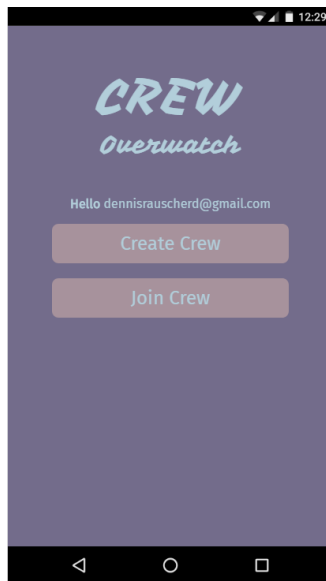


Abbildung 3: Screenshot Hauptmenü

## A.11 Screenshot Gruppenerstellung/-einstellungen

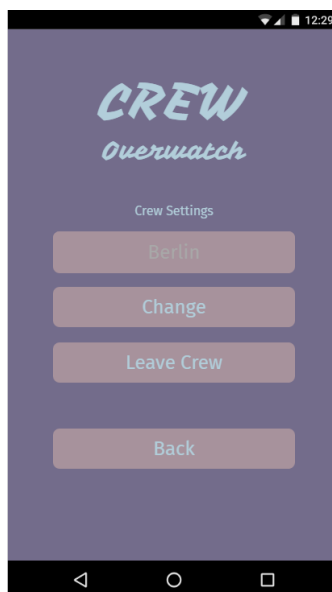


Abbildung 4: Screenshot Gruppenerstellung/-einstellungen

## A.12 Screenshot Gruppenbeitritt

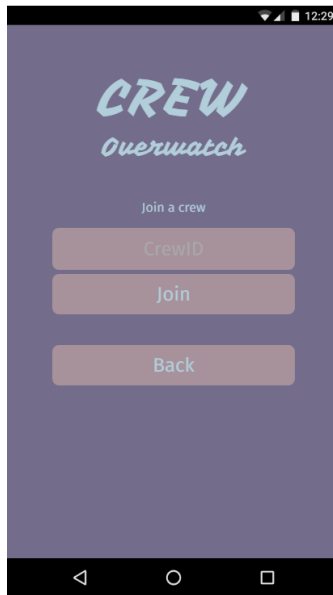


Abbildung 5: Screenshot Gruppenbeitritt

## A.13 Screenshot Listenansicht

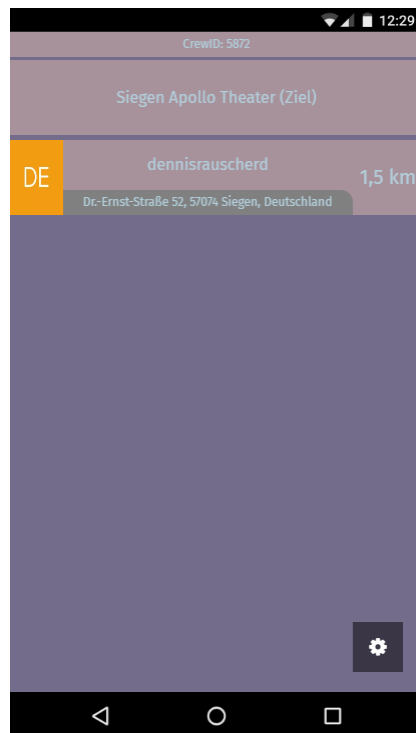


Abbildung 6: Screenshot Listenansicht

#### A.14 Ausschnitt aus der Kundendokumentation



Abbildung 5 – Crew Einstellungen

## A.15 Ausschnitt aus der Entwicklerdokumentation

```
2
3  var lastPage; //Zuletzt besuchte Seite
4  var nextPageAfterError; //Nächste Seite falls ein Fehler auftritt
5  var userToken = localStorage.getItem('access_Token'); //der UserToken für Auth0
6  var gpsAccepted = false; //Definiert ob bereits GPS-Standorte abgerufen werden können
7
8  //wird nach dem Laden der Webseite aufgerufen
9  $(function() {
10     //setLoading(true);
11     navigate_to('home');
12 });
13
14 //Eine Funktion um den Ladebildschirm anzeigen zu lassen
15 //isLoading : boolean
16 function setLoading(isLoading)
17 {
18     if(isLoading)
19     {
20         $('#loadingScreen').addClass('active');
21     }
22     else
23     {
24         $('#loadingScreen').removeClass('active');
25     }
26 }
27
28 //Eine Funktion um den Fehlerbildschirm anzeigen zu lassen
29 //msg : string, next : string
30 function showError(msg, next)
31 {
32     nextPageAfterError = next;
33
34     $('#errorMsg').html(msg);
35     navigate_to('errorScreen');
36 }
```