

# Einsatzplanung und Tracking von alpinen Rettungsaktionen mit Hilfe GPS

## Studienarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Herbstsemester 2017

Autoren:	Lukas Oberholzer & David Riederer
Betreuer:	Prof. Dr. Farhad Mehta
Projektpartner:	Namics AG, St. Gallen & Beutler Coaching, Herisau

## ABSTRACT

Die Alpine Rettung Schweiz (ARS) leistet terrestrische Einsätze für in Not geratene und hilfsbedürftige Menschen im alpinen, voralpinen und schwer zugänglichen Gebiet der Schweiz und dem angrenzenden Ausland. Die ARS ist mit verschiedenen Problemstellungen während eines Einsatzes konfrontiert.

Wie so oft kann es vorkommen, dass eine eigenständige Alarmierung nicht mehr möglich ist. In diesem Fall löst eine Vermisstenmeldung von Angehörigen die Alpine Rettung aus. Die Meldung lautet zum Beispiel, dass die Person sein Fahrzeug bei der Seilbahn auf der Schwägalp parkiert hat und vor hatte zu Fuss auf den Säntis zu wandern. Nun gibt es nicht nur einen Weg auf den Säntis und das Suchgebiet erstreckt sich über ein grosses und unwegsames Gelände. Aufgrund der Grösse müssen einige Suchtrupps, Suchhunde bis zu mehreren Helikoptern aufgeboden werden. Der zeitliche Aspekt ist dabei eine sehr wichtige Komponente, dass die Person lebend geborgen werden kann.

Ziel dieser Arbeit war es, eine bestehende Java Spring Boot Webapplikation und Android Tracking-App weiterzuentwickeln, sowie eine Mobile-App mit NativeScript zur Unterstützung der Retter während eines Einsatzes zu implementieren. Die vorhandene Webapplikation wurde um Funktionen wie «Einsatzplanung durchführen», «Suchaufträge planen» und «Live-Vorschau eines laufenden Suchauftrages» ergänzt. Bei der Android Tracking-App wird neu auch der Akkustand ausgelesen und GPS-Daten werden erst ab 25m Abweichung an die Webapplikation übermittelt, damit Streusignale nicht die Übersichtlichkeit auf der Karte vermindern. Ferner wurde ein unzuverlässiger Service für Standortupdates ausgetauscht und durch Google Play Services ersetzt. Der Fokus dieser App wurde hauptsächlich auf einen niedrigen Akkuverbrauch gelegt, da für in Not geratene Menschen jede zusätzliche Minute Akku lebensrettend sein kann. Durch den Einsatz des Cross-Plattform Frameworks NativeScript ist die Retter-App für Android und iOS nutzbar. Sie unterstützt die Retter während eines Einsatzes durch Anzeigen eines vom Einsatzleiter zugewiesenen Suchauftrages auf der Swisstopo Karte – die detaillierteste und genaueste Karte für die Schweiz.

Als Programmiersprachen wurden Java, JavaScript und TypeScript eingesetzt. Bootstrap – ein HTML, CSS und JavaScript Framework – und jQuery – eine JavaScript Bibliothek – wurden für die Gestaltung des Frontends der Webapplikation verwendet. Die Benutzeroberfläche der Retter-App orientiert sich an Googles Material Design Richtlinien und sorgt deshalb für schnelle Vertrautheit und Erlernbarkeit. Bei der Tracking-App ist das GUI sehr simpel gehalten und bietet nur wenige Einstellungsmöglichkeiten. Die Idee dahinter ist, dass die App von den Benutzern installiert wird und danach «vergessen» geht. Durch den geringen Akkuverbrauch fällt die App den Benutzern nicht negativ auf und im Notfall können die an die Webapplikation übermittelten Standortdaten lebensrettend sein, weil der Einsatzleiter bei der Einsatzplanung auf aktuelle Standortdaten zurückgreifen kann.

Das Projekt wurde in vier Phasen aufgeteilt, wobei in jeder Phase ein agiles Vorgehen gewählt wurde. So wurden 3 Wochen für Elaboration, 9 Wochen für Construction und 2 Wochen für Transition aufgewendet. Die Phase Inception wurde bereits vor Semesterbeginn abgeschlossen. Während der Construction Phase wurde nach jedem Sprint à 2 Wochen ein neuer Release der Webapplikation auf dem Produktiv-System veröffentlicht. Dadurch wurden Verbesserungsvorschläge und allfällige Fehler schnell entdeckt und konnten in die Planung des darauffolgenden Sprints miteinbezogen werden.

# MANAGEMENT SUMMARY

## Ausgangslage

Die Alpine Rettung Schweiz (ARS) leistet terrestrische Einsätze für in Not geratene und hilfsbedürftige Menschen im alpinen, voralpinen und schwer zugänglichen Gebiet der Schweiz und dem angrenzenden Ausland. Die Alpine Rettung Schweiz (ARS) ist mit verschiedenen Problemstellungen während eines Einsatzes konfrontiert. Die Ausgangslage dieser Studienarbeit beruht auf folgenden aktuellen Herausforderungen:

## Planung

- Die Planung und Einteilung der Suchtrupps benötigt viel Zeit und kann zu Missverständnissen führen.
- Die Weg- und/oder Reviersuche muss manuell vom Retter von einer Karte gemessen und im GPS von Hand erfasst werden.
- Nicht alle Retter verfügen über ein GPS-Gerät und nicht alle Retter beherrschen die Handhabung mit einem GPS-Gerät

## Auswertung

- Ein GPS-Track, wo die Suchtrupps wirklich waren, würde eine höhere Sicherheit zur weiteren Planung des Einsatzes führen.

## Prävention

- Eine Alarmierung mit der Rega-App ist vielfach durch den Vermissten oder Verunfallten nicht mehr auszuführen. Dies aufgrund einer technischen Gegebenheit wie Netzabdeckung, schwacher Akku oder durch fehlende menschliche Möglichkeit zu einer Aktivität z.B. Person liegt nach Sturz in Ohnmacht.

Mit moderner Technologie soll einerseits die Planung und die Auswertung eines Einsatzes vereinfacht und effizienter gestaltet werden. Andererseits soll präventiv durch ständiges GPS-Tracking eine vermisste Person schneller gefunden werden. Die Grundidee ist, dass eine Person sich beim Rettungssystem registriert und sich mit diesem System tracken lassen kann. Im Ernstfall, kann die Alpine Rettung auf die GPS Daten zugreifen und kann so die letzten Trackingdaten des Vermissten einsehen und so das Suchgebiet wesentlich eingrenzen, was sowohl viel Geld, als auch wertvolle Zeit einspart.

## Vorgehen

Das Projekt wurde in vier Phasen aufgeteilt, wobei in jeder Phase ein agiles Vorgehen gewählt wurde. So wurden 3 Wochen für Elaboration, 9 Wochen für Construction und 2 Wochen für Transition aufgewendet. Die Phase Inception wurde bereits vor Semesterbeginn abgeschlossen. Während der Construction Phase wurde nach jedem Sprint à 2 Wochen ein neuer Release der Webapplikation auf dem Produktiv-System veröffentlicht. Dadurch wurden Verbesserungsvorschläge und allfällige Fehler schnell entdeckt und konnten in die Planung des darauffolgenden Sprints miteinbezogen werden.

## Technologien

Die Webapplikation für die Einsatzplanung wurde mit Java Spring Boot umgesetzt. Das Backend basiert dementsprechend auf Java und das Frontend auf HTML, CSS, Bootstrap, JavaScript und jQuery. Die Mobile-App der Retter für Android und iOS wurde mit dem Cross-Plattform Framework NativeScript entwickelt, welche 100% gemeinsamer Code aufweist. Die Tracker-App für Android auf Java Basis wurde um Google Play Services für die Lokalisierung, sowie um einige andere Features erweitert.

## Ergebnisse

Die Webapplikation «trs-cockpit» unterstützt den Einsatzleiter bei der Planung, Durchführung und Auswertung von Rettungseinsätzen. Beim Erstellen eines Einsatzes sieht der Einsatzleiter auf einer Karte, wo seine Retter verteilt sind und kann die nahe am Einsatzort befindlichen Retter schneller aufbieten. Ist die vermisste Person mit der Tracking-App ausgestattet, kann zusätzlich der Track dieser Person in den Einsatz mit einbezogen werden. Hat die in Not geratene Person z.B. einen Notruf per REGA-App abgesetzt, so kann zusätzlich die letzte bekannte Position als Einsatzpunkt in der Webapplikation hinterlegt werden. Dies hilft zusätzlich, das Suchgebiet einzuschränken. Ein Einsatz wird in Suchaufträge unterteilt, welche entweder eine Weg- oder Reversuche darstellen. Diese werden auf der Karte eingezeichnet und Rettern zugewiesen.

Die Retter sind mit der Retter-App «trs-rescuer» ausgestattet. Auf dieser sehen sie ihre aktuelle Position, sowie die vom Einsatzleiter zugeteilten Suchaufträge. Öffnet der Retter einen Suchauftrag, so wird er zuerst über die Details informiert und sieht anschliessend die Weg- oder Reversuche auf der Karte. Befindet er sich im entsprechenden Suchgebiet, so kann er den Suchauftrag starten und die Route absuchen. Währenddessen sieht der Einsatzleiter in der Webapplikation, wo sich seine Retter aufhalten und kann ihre zurückgelegte Strecke nachverfolgen.

Das Tracking wird durch die Tracker-App «trs-tracker» – eine sehr stromsparende und im Hintergrund ausgeführte App – sichergestellt. Alle Einsatzleiter und Retter sind mit dieser ausgestattet. Die App sendet in einem konfigurierbaren Intervall jeweils die aktuelle Position mitsamt dem Akkustand an die API der Webapplikation. Ist kein mobiles Netzwerk oder WLAN verfügbar, so versucht die App nach Ablauf des nächsten Intervalls die verbleibenden Daten hochzuladen. Möchte man das Tracking vorübergehend deaktivieren, so kann dies in der App ausgeschaltet werden.

## Ausblick

Die drei entwickelten Applikationen werden an den Auftraggeber übergeben und dort als Pilot-Projekt eingesetzt. Wunschziel ist es, dass jede/r Berggänger/in mit der Tracking-App ausgestattet ist und im Notfall dank des Trackings wertvolle Zeit bei Suchaktionen eingespart werden kann. Die Retter werden im Umgang mit der Retter-App geschult und können bestenfalls auf zusätzliche Hilfsgeräte wie GPS-Uhren verzichten.

## ERKLÄRUNG DER EIGENSTÄNDIGKEIT

Hiermit erklären wir,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben
- dass wir keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt haben

Rapperswil, 22.12.2017

Lukas Oberholzer

A handwritten signature in blue ink, appearing to be 'L. Oberholzer', with a long horizontal stroke extending to the right.

David Riederer

A handwritten signature in blue ink, appearing to be 'D. Riederer', written in a cursive style.

# AUFGABENSTELLUNG STUDIENARBEIT

## „Einsatzplanung und Tracking von alpinen Rettungsaktionen mit Hilfe GPS“ HS 2017

### 1 AUFTRAGGEBER UND BETREUER

- *Auftraggeber:* Beutler Coaching & Namics AG
- *Ansprechpartner Auftraggeber:*
  - Heinz Beutler, Beutler Coaching & Namics AG, [heinz@beutler-coaching.ch](mailto:heinz@beutler-coaching.ch)
  - Beat Helfenberger, Namics AG, [beat.helfenberger@namics.com](mailto:beat.helfenberger@namics.com)
- *Betreuer:* Prof. Dr. Farhad Mehta

### 2 STUDIERENDE

Diese Arbeit wird als Studienarbeit an der Abteilung Informatik durchgeführt von

- Lukas Oberholzer
- David Riederer

### 3 AUSGANGSLAGE

Die Alpine Rettung Schweiz (ARS) leistet terrestrische Einsätze für in Not geratene und hilfsbedürftige Menschen im alpinen, voralpinen und schwer zugänglichen Gebiet der Schweiz und dem angrenzenden Ausland. Die Alpine Rettung Schweiz (ARS) ist mit verschiedenen Problemstellungen während eines Einsatzes konfrontiert. Die Ausgangslage dieser Studienarbeit beruht auf folgende aktuellen Herausforderungen:

#### Planung

- Die Planung und Einteilung der Suchtrupps benötigt viel Zeit und kann zu Missverständnissen führen.
- Die Weg- und/oder Reviersuche muss manuell vom Retter von einer Karte gemessen und im GPS von Hand erfasst werden.
- Nicht alle Retter verfügen über ein GPS-Gerät und nicht alle Retter beherrschen die Handhabung mit einem GPS-Gerät

#### Auswertung

- Ein GPS-Track, wo die Suchtrupps wirklich waren, würde eine höhere Sicherheit zur weiteren Planung des Einsatzes führen.

#### Prävention

- Eine Alarmierung mit der Rega-App ist vielfach durch den Vermissten oder Verunfallten nicht mehr auszuführen. Dies aufgrund einer technischen Gegebenheit wie Netzabdeckung, schwacher Akku oder durch fehlende menschliche Möglichkeit zu einer Aktivität z.B. Person liegt nach Sturz in Ohnmacht.

Mit moderner Technologie soll einerseits die Planung und die Auswertung eines Einsatzes vereinfacht und effizienter gestaltet werden. Andererseits soll präventiv durch ständiges GPS-Tracking eine vermisste Person schneller gefunden werden.

Wie so oft kann es vorkommen, dass eine eigenständige Alarmierung nicht mehr möglich ist. In diesem Fall löst eine Vermisstenmeldung von Angehörigen die Alpine Rettung aus. Die Meldung lautet zum Beispiel, dass die Person sein Fahrzeug bei der Seilbahn auf der Schwägalp parkiert hat und vor hatte zu Fuss auf den Säntis zu wandern. Nun gibt es nicht nur einen Weg auf den Säntis und das Suchgebiet erstreckt sich über ein grosses und unwegsames Gelände. Aufgrund der Grösse müssen einige Suchtrupps, Suchhunde bis zu mehreren Helikoptern aufgeboden werden. Der zeitliche Aspekt ist dabei eine sehr wichtige Komponente, dass die Person lebend geborgen werden kann. Ebenfalls verursacht die Rettung immense Suchkosten und soll anhand von GPS Tracking reduziert werden. Die Grundidee ist, dass eine Person sich beim Rettungssystem registriert und sich mit diesem System tracken lassen kann. Im Ernstfall, kann die Alpine Rettung auf die GPS Daten zugreifen und kann so die letzten Trackingdaten des Vermissten einsehen und so das Suchgebiet wesentlich eingrenzen.

#### **4 BESCHREIBUNG DER AUFGABE**

Es soll eine Webapplikation und eine Mobile-App entwickelt werden, welches die oben beschriebenen Punkte in der alpinen Rettung verbessert. Am Ende der Arbeit erwarten die Ansprechpartner eine funktionsfähige Applikation. Die Entwicklung und deren Umfang definiert sich in einem agilen Softwareentwicklungsvorgehen.

Damit die Ansprechpartner, nach Abschluss der Studienarbeit, die Applikation weiterentwickeln können, gelten folgende Rahmenbedingungen:

- Mobile OS-Unterstützung von Apple und Android. Die Mobile-App soll auch offline Funktionen aufweisen.
- Die Webapplikation für das Backendsystem soll mit dem Bootstrap Framework Responsive umgesetzt werden.
- Zu unterstützende Browser sind die Standard-Browser der Betriebssysteme Windows, iOS und Android.
- Die Backendentwicklung basiert auf Basis Java und dem Spring Framework
- Die Webapplikation wird in einem Tomcat Application Server deployed.
- Das Betriebssystem für das Backend ist Linux.
- Eine benötigte Datenhaltung soll in MongoDB, MySQL oder MariaDB umgesetzt werden. Eine alternative Datenhaltung wie PostgreSQL kann zu Beginn des Projekts analysiert werden.
- Ein git-Sourcenverwaltung wird von den Ansprechpartnern zur Verfügung gestellt.
- Zielsystem für das Backend wird von den Ansprechpartnern installiert.

Die Auftraggeber Namics AG und Beutler Coaching dürfen die Resultate dieser Arbeit frei nutzen und weiterentwickeln. Die Nutzungsrechte können durch den Auftraggeber weiter vergeben werden.

Folgende Minimalziele werden von den Auftraggebern erwartet:

- Eine im Hintergrund ständig laufende Mobile Tracking App für Android und iOS mit geringem Akkuverbrauch. Fehlt der Netzeempfang, so werden die Trackingdaten zu einem späteren Zeitpunkt gesendet damit das Bewegungsprofil möglichst vollständig ist. Das Intervall für die Standortübermittlung kann in der App definiert werden.
- In einer Webapplikation kann ein Einsatzleiter ein Revier oder einen Weg definieren und diesen einem Suchtrupp zuweisen. Der Einsatzleiter hat dann einen Überblick über alle aktiven Suchtrupps und kann auch Änderungen an deren Routen vornehmen.
- Der Suchtrupp sieht die definierte Route oder das abgesteckte Revier, sowie die Spur des Vermissten in ihrer Mobile App und werden ebenfalls fortlaufend getrackt. Wird Ihre Route geändert, wird dies in ihrer App aktualisiert.

## 5 ZUR DURCHFÜHRUNG

Mit dem Betreuer finden wöchentliche Besprechungen statt. Besprechungen mit dem Auftraggeber sind von den Studierenden nach Bedarf zu initialisieren.

Alle Besprechungen sind von den Studierenden mit einer Traktandenliste vorzubereiten, die Besprechung ist durch die Studierenden zu leiten und die Ergebnisse sind in einem Protokoll festzuhalten, das den Betreuern und dem Auftraggeber per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsergebnisse in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsergebnisse erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Dokumentation.

## 6 DOKUMENTATION

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen (siehe <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html>). Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Alle Resultate sind vollständig auf CD/DVD in 3 Exemplaren abzugeben. Der Bericht ist ausgedruckt in doppelter Ausführung abzugeben.

## 7 TERMINE

Siehe Terminplan auf <https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html>.



## 8 ARBEITSUMFANG

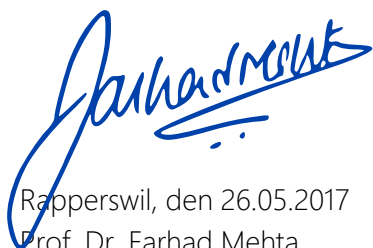
Eine erfolgreiche Studienarbeit zählt 8 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von 30 Stunden budgetiert. Die verwendete Arbeitszeit muss erfasst und dokumentiert werden.

## 9 BEURTEILUNG

Für die Beurteilung ist der HSR-Betreuer verantwortlich. Die Benotung erfolgt gemäss folgender Tabelle.

Gewichtspunkt	Gewicht
1. Organisation, Durchführung	1/5
2. Berichte	1/5
3. Inhalt	3/5

Im Übrigen gelten die Bestimmungen der Abteilung Informatik für Studienarbeiten (siehe <https://www.hsr.ch/Ablaeufe-und-Regelungen-Studie.7479.0.html>).



Rapperswil, den 26.05.2017  
Prof. Dr. Farhad Mehta

# INHALTSVERZEICHNIS

Abstract .....	I
Management Summary .....	II
Erklärung der Eigenständigkeit .....	IV
Aufgabenstellung Studienarbeit .....	V
1 Analyse .....	1
1.1 Ausgangslage .....	1
1.1.1 Anwendungsumfeld .....	1
1.1.2 Bestehende Applikation .....	1
1.1.3 Webapplikation .....	1
1.1.4 Tracking-App .....	3
1.2 Domainanalyse .....	4
1.3 Use Cases .....	5
1.3.1 Use Case Diagramm .....	5
1.3.2 Aktoren & Stakeholder .....	6
1.3.3 Kurzbeschreibung der Use Cases .....	6
1.4 Functional Requirements .....	8
1.5 Non-Functional Requirements .....	8
1.6 Qualitätsmassnahmen .....	9
1.6.1 Qualitätsmerkmale .....	9
1.6.2 Systemtest Spezifikation .....	10
2 Projektmanagement .....	11
2.1 Projektplan .....	11
2.2 Phasen .....	11
2.3 Meilensteine .....	11
2.4 Risikomanagement .....	12
3 Konzeption und Design .....	14
3.1 Webapplikation .....	14
3.1.1 Java Spring Boot .....	14
3.1.2 MySQL-Datenbank .....	14
3.2 Retter-App .....	15
3.3 Tracking-App .....	16
4 UI Konzept .....	17
4.1 Mockups .....	17

4.2	Webapplikation.....	18
4.3	Retter-App .....	21
4.4	Tracking-App.....	21
5	Ergebnisse .....	22
5.1	Deployment .....	22
5.2	Webapplikation.....	23
5.2.1	Screenshots.....	23
5.2.2	Code-Metriken.....	28
5.3	Retter-App .....	29
5.3.1	Screenshots.....	29
5.3.2	Code-Metriken.....	30
5.4	Tracking-App.....	31
5.4.1	Screenshots.....	31
5.4.2	Code-Metriken.....	32
5.5	Zielerreichung .....	32
5.6	Schlussfolgerung .....	33
5.6.1	Projektverlauf .....	33
5.6.2	Fazit zur Webapplikation .....	34
5.6.3	Fazit zur Tracking-App .....	34
5.6.4	Fazit zur Retter-App.....	35
6	Ausblick .....	36
6.1	Sicherheit .....	36
6.2	Aktualisieren von Einsätzen und Suchaufträgen .....	36
6.3	Registrierung .....	36
6.4	Anonyme Tracker .....	37
6.5	Drohneneinsätze .....	37
	Abbildungsverzeichnis .....	38
	Tabellenverzeichnis .....	39
A	Anhang.....	41
A.1	Use Case Detail .....	41
A.1.1	UC01: Person tracken.....	41
A.1.2	UC02: Track anzeigen.....	41
A.1.3	UC03: Person als Retter definieren .....	42
A.1.4	UC04: Retter anzeigen .....	42

A.1.5	UC05: Einsatz definieren .....	43
A.1.6	UC06: Suchauftrag definieren .....	43
A.1.7	UC07: Suchauftrag abholen .....	44
A.1.8	UC08: Suchauftrag starten .....	44
A.1.9	UC09: Suchauftrag abschliessen .....	45
A.1.10	UC10: Suchauftrag und Track vergleichen .....	45
A.1.11	UC11: Retter laufendem Einsatz hinzufügen .....	46
A.1.12	UC12: Retter laufendem Suchauftrag hinzufügen .....	46
A.1.13	UC13: Suchauftrag abbrechen .....	47
A.1.14	UC14: Einsatz abschliessen .....	47
A.1.15	UC15: Person registrieren .....	48
A.1.16	UC16: Retter authentifizieren .....	48
A.2	Systemtest trs-cockpit .....	49
A.2.1	ST1: Authentifizieren .....	49
A.2.2	ST2: Track anzeigen .....	49
A.2.3	ST3: Profil bearbeiten .....	49
A.2.4	ST4: GPS-Daten löschen .....	49
A.2.5	ST5: Einsatz erstellen .....	50
A.2.6	ST6: Suchauftrag erstellen .....	50
A.2.7	ST7: Suchauftrag-Details anzeigen .....	50
A.3	Systemtest trs-rescuer .....	51
A.3.1	ST8: Starten der App .....	51
A.3.2	ST9: Authentifizieren .....	51
A.3.3	ST10: Suchaufträge anzeigen .....	51
A.3.4	ST11: Suchauftrag durchführen .....	52
A.4	Systemtest trs-tracker .....	53
A.4.1	ST12: Erster Start .....	53
A.4.2	ST13: Intervall konfigurieren .....	53
A.5	Projektmanagement Detail .....	54
A.5.1	Gant Diagramm .....	54
A.6	Risikomanagement Detail .....	56
A.6.1	Risiko-Analyse in der Elaboration-Phase .....	56
A.6.2	Risiko-Analyse in der Construction-Phase .....	56

# 1 ANALYSE

## 1.1 Ausgangslage

### 1.1.1 Anwendungsumfeld

Die Alpine Rettung Schweiz (ARS) leistet terrestrische Einsätze für in Not geratene und hilfsbedürftige Menschen im alpinen, voralpinen und schwer zugänglichen Gebiet der Schweiz und dem angrenzenden Ausland. Die Alpine Rettung Schweiz (ARS) ist mit verschiedenen Problemstellungen während eines Einsatzes konfrontiert.

Wie so oft kann es vorkommen, dass eine eigenständige Alarmierung nicht mehr möglich ist. In diesem Fall löst eine Vermisstenmeldung von Angehörigen die Alpine Rettung aus. Die Meldung lautet zum Beispiel, dass die Person sein Fahrzeug bei der Seilbahn auf der Schwägalp parkiert hat und vor hatte zu Fuss auf den Säntis zu wandern. Nun gibt es nicht nur einen Weg auf den Säntis und das Suchgebiet erstreckt sich über ein grosses und unwegsames Gelände. Aufgrund der Grösse müssen einige Suchtrupps, Suchhunde bis zu mehreren Helikoptern aufgeboden werden. Der zeitliche Aspekt ist dabei eine sehr wichtige Komponente, dass die Person lebend geborgen werden kann. Ebenfalls verursacht die Rettung immense Suchkosten und soll anhand von GPS Tracking reduziert werden. Die Grundidee ist, dass eine Person sich beim Rettungssystem registriert und sich mit diesem System tracken lassen kann. Im Ernstfall, kann die Alpine Rettung auf die GPS Daten zugreifen und kann so die letzten Trackingdaten des Vermissten einsehen und so das Suchgebiet wesentlich eingrenzen.

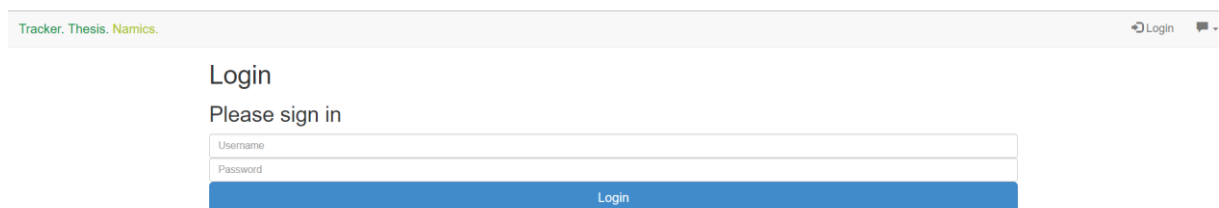
### 1.1.2 Bestehende Applikation

Die Webapplikation wurde ursprünglich als Bachelor-Arbeit entwickelt. Diese bot eine Übersicht über alle getrackten Benutzer inklusiv ihrer letzten Position und Track jeder Person. Die ganze Darstellung wurde auf einer Karte von Google Maps realisiert. Die Datenhaltung erfolgte in einer MongoDB und die API für Tracking-Daten war bereits implementiert. Zusätzlich existierte auch ein Entwurf der Tracker-App. Diesen haben wir um zusätzliche Funktionen wie Akkustand auslesen ergänzt und den unzuverlässigen Service für Standortdaten durch Google Play Services ausgetauscht.

### 1.1.3 Webapplikation

#### Login

Das Login erfolgt mittels Benutzername und Passwort.



Tracker. Thesis. Namics. Login

Login

Please sign in

Username

Password

Login

Abbildung 1.1: trs-cockpit Login

## Übersicht

Nach erfolgreichem Login werden alle erfassten Benutzer angezeigt. Es kann der Track und die letzten Positionen aller Benutzer auf Google Maps angezeigt werden.

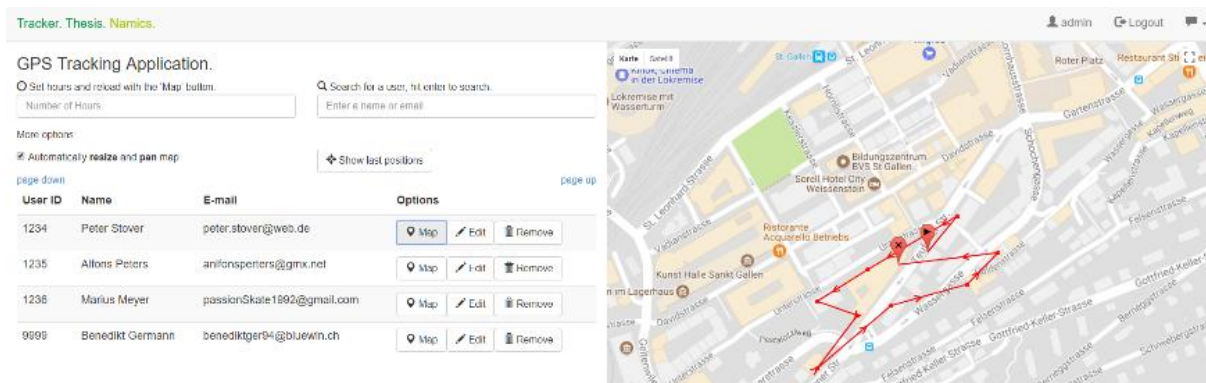


Abbildung 1.2: trs-cockpit Übersicht

## Benutzerprofil

Beim Profil kann der Name und die E-Mail-Adresse angepasst werden.



Abbildung 1.3: trs-cockpit Benutzerprofil

## Standortdaten

Der Administrator kann die Standortdaten jedes Benutzers ansehen.

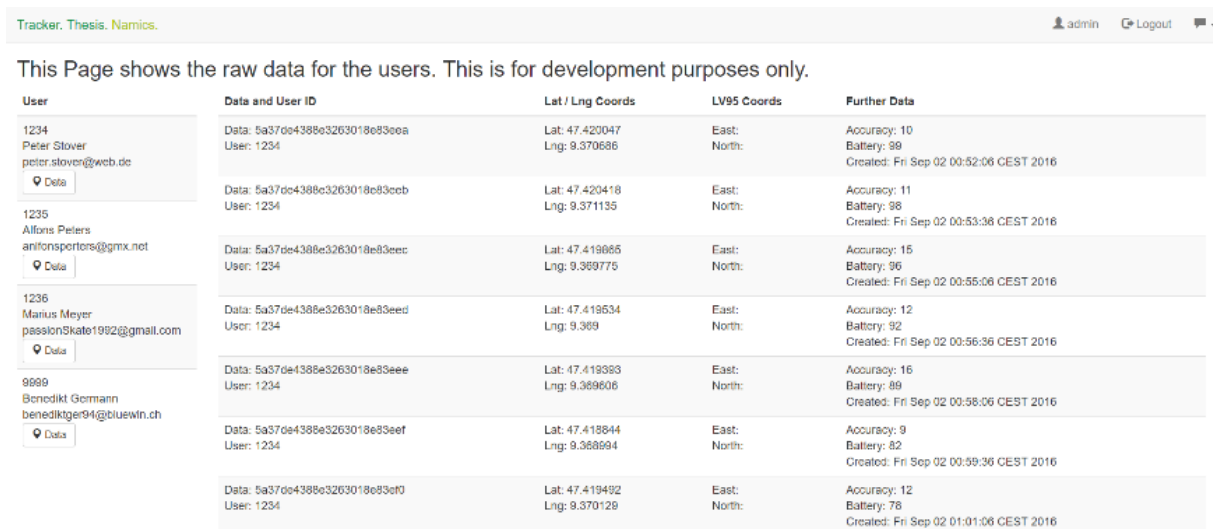
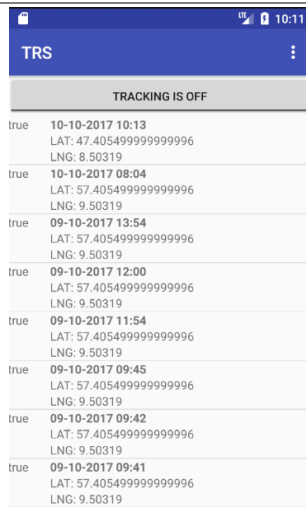


Abbildung 1.4: trs-cockpit Standortdaten

## 1.1.4 Tracking-App

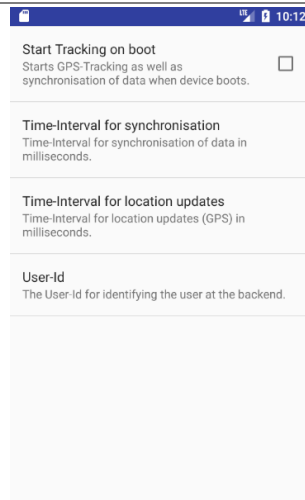
### Übersicht



TRS	
TRACKING IS OFF	
true	10-10-2017 10:13 LAT: 47.405499999999996 LNG: 8.50319
true	10-10-2017 08:04 LAT: 57.405499999999996 LNG: 9.50319
true	09-10-2017 13:54 LAT: 57.405499999999996 LNG: 9.50319
true	09-10-2017 12:00 LAT: 57.405499999999996 LNG: 9.50319
true	09-10-2017 11:54 LAT: 57.405499999999996 LNG: 9.50319
true	09-10-2017 09:45 LAT: 57.405499999999996 LNG: 9.50319
true	09-10-2017 09:42 LAT: 57.405499999999996 LNG: 9.50319
true	09-10-2017 09:41 LAT: 57.405499999999996 LNG: 9.50319

Abbildung 1.5: Übersicht

### Einstellungen



Start Tracking on boot  
Starts GPS-Tracking as well as synchronisation of data when device boots. ☐

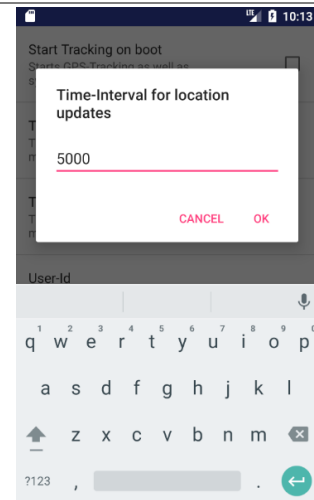
Time-Interval for synchronisation  
Time-Interval for synchronisation of data in milliseconds.

Time-Interval for location updates  
Time-Interval for location updates (GPS) in milliseconds.

User-Id  
The User-Id for identifying the user at the backend.

Abbildung 1.6: Einstellungen

### Zeitintervall



Start Tracking on boot  
Starts GPS-Tracking as well as synchronisation of data when device boots. ☐

Time-Interval for location updates  
5000

CANCEL OK

User-Id

Abbildung 1.7: Zeitintervall

Zeigt, ob der erfasste Standort bereits auf den Server übertragen wurde. Zusätzlich werden ein Zeitstempel und die Koordinaten im WGS84-Format angezeigt.

Die Einstellungen sind auf Englisch. Es können Intervalle für Standortupdates und Synchronisation eingestellt werden. Die User-ID wird verwendet, um die zu übertragenden Daten einem Benutzer auf dem Server zuzuweisen.

Die Intervalle müssen in Millisekunden definiert werden. Damit das neue Intervall übernommen wird, muss das Tracking einmal aus- und wieder eingeschaltet werden.

Tabelle 1.1: trs-tracker: Ausgangslage

## 1.2 Domainanalyse

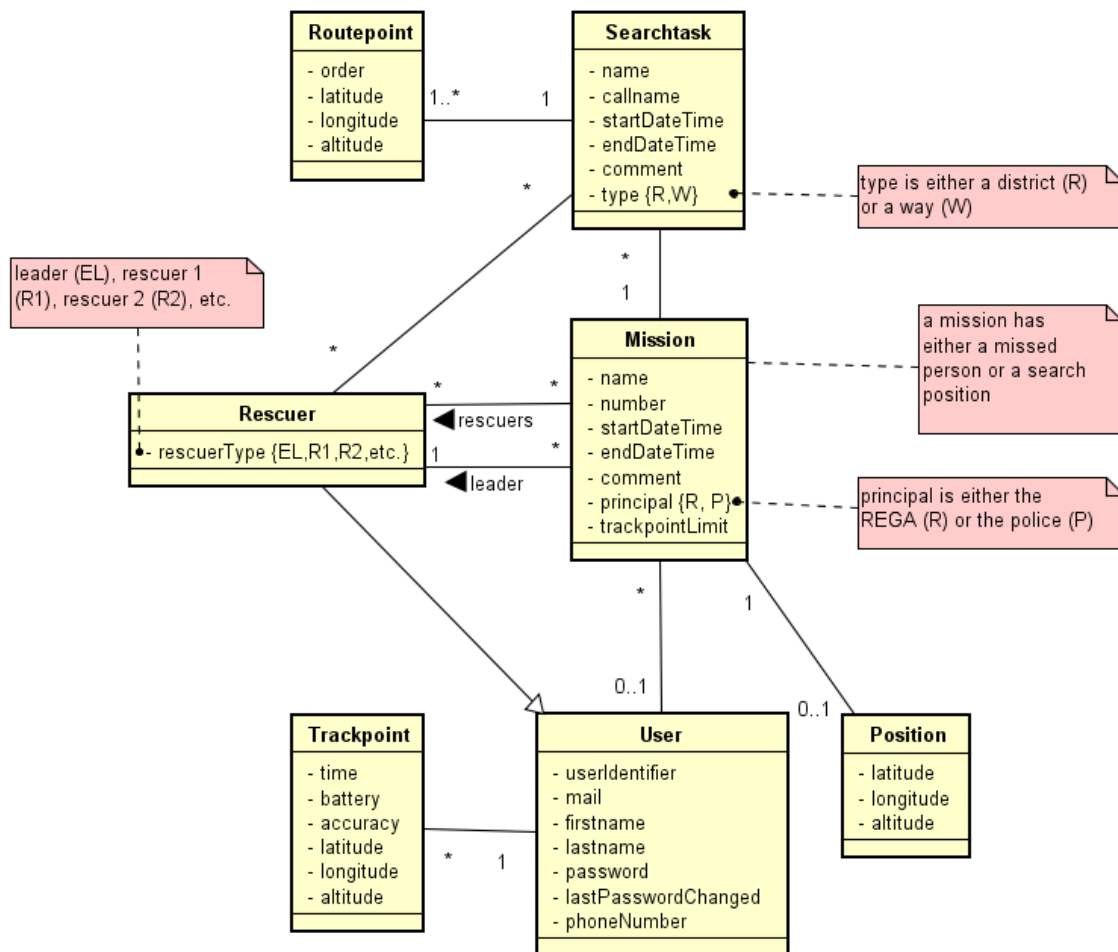


Abbildung 1.8: Domainmodell

### Kurzbeschreibung der Konzepte

<b>Routepoint</b>	Mehrere auf der Karte für die Weg- oder Reviersuche eingezeichnete Koordinaten.
<b>Searchtask</b>	Ein Suchauftrag ist entweder eine Weg- oder Reviersuche und wird Rettern zugewiesen.
<b>Mission</b>	Ein Einsatz umfasst die komplette Suchaktion und umfasst mehrere Suchaufträge. Der Einsatz wird geschlossen, sobald die vermisste Person gefunden wurde oder die Suche eingestellt wird.
<b>Rescuer</b>	Es gibt verschiedene Retter-Typen. Diese Spezialisierung hilft dem Einsatzleiter bei der Einsatzplanung und je nach Typ haben diese auch mehr oder weniger Zugriffsrechte im System.
<b>Position</b>	Definiert den Einsatzpunkt. Wo wurde die vermisste Person zuletzt gesehen oder wo wird diese vermutet.
<b>User</b>	Die getrackte/vermisste Person. Kann auch ein Retter sein.
<b>Trackpoint</b>	Von der Tracking-App der Benutzer an das System übermittelte Standortdaten. Dienen als Planungselement für Einsätze und Suchaktionen.

Tabelle 1.2: Kurzbeschreibung der Konzepte



## 1.3 Use Cases

### 1.3.1 Use Case Diagramm

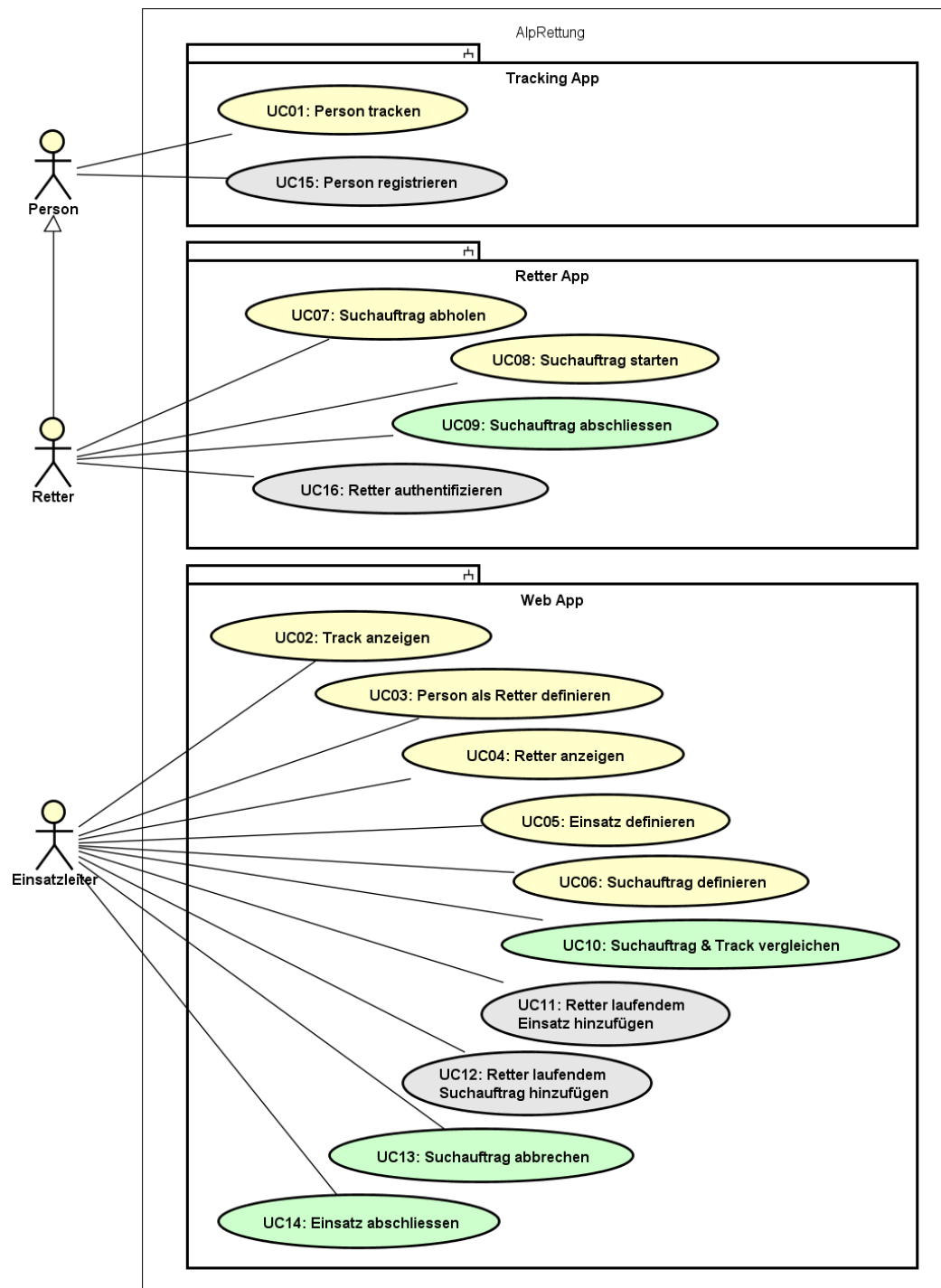


Abbildung 1.9: Use Case Diagramm

Erklärung zu den Farben: Gelb: 1. Priorität / Grün: 2. Priorität / Grau: 3. Priorität

### 1.3.2 Akteure & Stakeholder

Einsatzleiter	Person mit mehrjähriger Erfahrung bei alpinen Rettungsaktionen
Retter	Mitglied der Alpinen Rettung Schweiz
Person	Meist Wanderer oder Kletterer

*Tabelle 1.3: Akteure & Stakeholder*

### 1.3.3 Kurzbeschreibung der Use Cases

#### UC01: Person tracken

Die GPS Daten einer Person werden in einem konfigurierbaren Intervall an den Server gesendet.

#### UC02: Track anzeigen

Der Einsatzleiter selektiert eine Person und sieht deren Track in einem bestimmten Zeitintervall.

#### UC03: Person als Retter definieren

Eine getrackte Person kann als ein Retter definiert werden.

#### UC04: Retter anzeigen

Der Einsatzleiter sieht alle Retter auf der Karte, wenn er einen neuen Einsatz erstellt.

#### UC05: Einsatz definieren

Bei einer Vermisstenmeldung wird ein neuer Einsatz mit Name, Einsatznummer, Startzeitpunkt und den beteiligten Rettern definiert.

#### UC06: Suchauftrag definieren

Der Einsatzleiter definiert in einem Einsatz eine Revier- oder Wegsuche auf der Karte mit einem Namen und Rufnamen. Der Suchauftrag wird einem oder mehreren Rettern zugewiesen.

#### UC07: Suchauftrag abholen

Der Retter kann in der Retter-App einen Suchauftrag abholen und sieht die vorgegebene Weg- oder Reviersuche. Zusätzlich wird die letzte an den Server gesendete Position angezeigt sowie der Track der vermissten Person, falls dieser verfügbar ist.

#### UC08: Suchauftrag starten

Sobald der Retter am Startpunkt seines Suchauftrages angekommen ist, bestätigt er, dass sein Suchauftrag nun offiziell beginnt.

#### **UC09: Suchauftrag abschliessen**

Sobald die Retter mit dem Suchauftrag fertig sind, schliessen sie den Suchauftrag mit einer Bemerkung ab.

#### **UC10: Suchauftrag und Track vergleichen**

Der Einsatzleiter sieht die definierte Weg- oder Reviersuche und zusätzlich den aktuellen Track aller an diesem Suchauftrag beteiligten Retter.

#### **UC11: Retter laufendem Einsatz hinzufügen**

Einem Einsatz können auch zu einem späteren Zeitpunkt noch weitere Retter hinzugefügt werden.

#### **UC12: Retter laufendem Suchauftrag hinzufügen**

Einem Suchauftrag können auch zu einem späteren Zeitpunkt weitere Retter hinzugefügt werden.

#### **UC13: Suchauftrag abbrechen**

Der Einsatzleiter kann einen Suchauftrag aus verschiedenen Gründen abbrechen.

#### **UC14: Einsatz abschliessen**

Ist ein Einsatz zu Ende, wird dieser als abgeschlossen definiert.

#### **UC15: Person registrieren**

Eine Person installiert die Tracking-App und muss sich mit seiner E-Mail-Adresse identifizieren.

#### **UC16: Retter authentifizieren**

Ein Retter besitzt ein Login (Email + Passwort) und loggt sich damit in der Retter-App ein.

## 1.4 Functional Requirements

FR-Nr.	Beschreibung
1	Es wird in allen drei Applikationen das von den Auftraggebern zur Verfügung gestellte Logo verwendet.
2	Alle User Interfaces werden auf Deutsch eingestellt.
3	Jede Funktion soll bestätigt werden, damit in der Hektik des Einsatzes nichts aus- versehen etwas gelöscht oder verändert wird.

*Tabelle 1.4: Funktionale Anforderungen*

## 1.5 Non-Functional Requirements

NFR-Nr.	Beschreibung
1	Die Tracking App muss sehr akkusparend laufen (Akkuverbrauch 1-2%)
2	Die Webapplikation soll schlussendlich tausende gleichzeitige Anfragen der Tracking App handhaben können
3	Die Webapplikation soll schlussendlich 100-200 gleichzeitige Verbindungen der Retter App verwalten können
4	Die Webapplikation soll schlussendlich 10 gleichzeitige Verbindungen über das Frontend verwalten können
5	Serverstandort ist in der Schweiz und Hostler ist zertifiziert <ul style="list-style-type: none"><li>▪ SLA verspricht 99.5% Verfügbarkeit</li><li>▪ 2 Cluster an einem Standort inkl. Load Balancer</li><li>▪ Backup alle 4 Stunden garantiert</li></ul>

*Tabelle 1.5: Nicht-funktionale Anforderungen*

## **1.6 Qualitätsmassnahmen**

### **1.6.1 Qualitätsmerkmale**

#### **Zuverlässigkeit**

Die Tracker-App kann ohne bestehende Internetverbindung die Standortdaten auslesen und diese lokal in einer SQLite Datenbank zwischenspeichern bis wieder eine Internetverbindung vorhanden ist. Besteht wieder eine Internetverbindung, so werden alle noch nicht synchronisierten Daten an die API des Servers übertragen und lokal aus der SQLite Datenbank entfernt. Die Retter-App zeigt die Karte bis zu einem gewissen Zoom-Faktor auch ohne kontinuierliche Internetverbindung an. Dies ist eine sehr hilfreiche Funktion, wenn man bedenkt, dass der Netzempfang in den Bergen teilweise stark eingeschränkt ist.

#### **Verständlichkeit**

Für den User soll durch entsprechendes Feedback immer klar sein, ob eine Aktion erfolgreich war oder nicht. Die Navigation in der Webapplikation und der Retter-App wird möglichst einfach gehalten und es wird darauf geachtet, dass keine View überladen wirkt. Wenn in einer View keine Daten verfügbar sind, dann wird der Benutzer mittels Feedback darauf hingewiesen. Es werden viele gebräuchliche Symbole verwendet, um Interaktionen auszulösen.

#### **Erlernbarkeit**

Die Bedienelemente in den mobilen Apps entsprechen den Material Design Richtlinien von Google. Der Benutzer findet eine vertraute Oberfläche vor – das Design orientiert sich an vielen anderen beliebten Apps - und erlernt somit schnell den Umgang mit der Applikation. Beispielsweise den Navigation Drawer aus der Retter-App kennen viele Benutzer aus der Gmail App.

#### **Bedienbarkeit**

Die Verschachtelung von Inhalten wurde möglichst flach gehalten. So sind wichtige Funktionen schnell und ohne Umweg erreichbar. Wichtige und oft benutzte Elemente nehmen mehr Platz ein oder wurden durch spezielles Design hervorgehoben. Bei der Verwaltung von Einsätzen und Suchaufträgen in der Webapplikation ist das Unser Interface fast identisch und erleichtert dadurch die Bedienbarkeit.

## 1.6.2 Systemtest Spezifikation

Damit die Applikationen nebst der Unit Tests auch entsprechend getestet werden können, wurden Systemtests definiert, die mindestens alle Use-Cases der ersten Priorität, sowie die Functional Requirements abdecken.

### trs-cockpit

Systemtest	Use Cases	Functional Requirements
ST1: Authentifizieren	UC16	
ST2: Track anzeigen	UC02	
ST3: Profil bearbeiten	UC03	
ST4: GPS-Daten löschen		
ST5: Einsatz erstellen	UC04, UC05	
ST6: Suchauftrag erstellen	UC06	
ST7: Suchauftrag-Details anzeigen	UC10	

*Tabelle 1.6: Systemtests trs-cockpit*

### trs-rescuer

Systemtest	Use Cases	Functional Requirements
ST8: Starten der App		FR1
ST9: Authentifizieren	UC16	
ST10: Suchaufträge anzeigen	UC07	
ST11: Suchauftrag durchführen	UC08, UC09	FR3

*Tabelle 1.7: Systemtests trs-rescuer*

### trs-tracker

Systemtest	Use Cases	Functional Requirements
ST12: Erster Start	UC01, UC15	FR1
ST13: Intervall konfigurieren	UC15	

*Tabelle 1.8: Systemtests trs-tracker*

## 2 PROJEKTMANAGEMENT

### 2.1 Projektplan

Das Projekt beinhaltet sechs Meilensteine, die über drei Phasen verteilt sind.

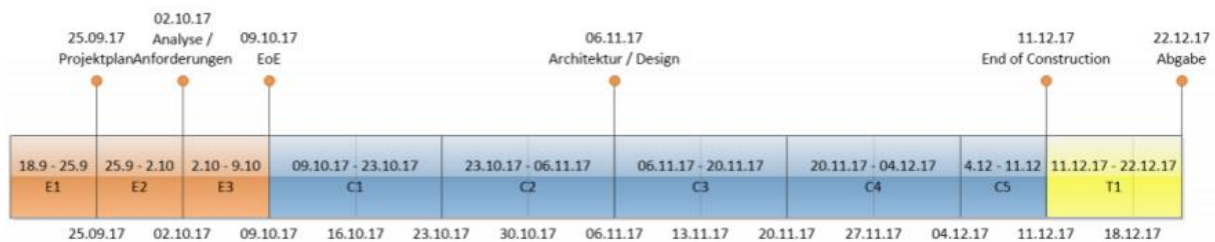


Abbildung 2.1: Projektplan

### 2.2 Phasen

Phase	Beschreibung	Dauer
Inception	Erstellung Aufgabenstellung	erledigt
Elaboration	Erstellung Projektplan, Definition Meilensteine, Erstellung Prototyp, Einarbeitung, Anforderungen, Architektur	3 Wochen
Construction	Programmierung der Mobile-Apps und der Web-App, Erstellung der Datenbank, Testing	9 Wochen
Transition	Projektauswertung, Statistiken, Soll-Ist-Vergleich, Abschluss der Dokumentation	2 Wochen

Tabelle 2.1: Projektphasen

### 2.3 Meilensteine

Meilenstein	Beschreibung	Datum
MS 1: Projektplan	Abgabe Projektplan	25.09.2017
MS 2: Analyse / Anforderungen	Funktionale und nichtfunktionale Anforderungen definiert, Use Cases brief und casual	02.10.2017
MS 3: Ende Elaboration	Erster Architektur-Prototyp ist fertig, Technologien und Arbeitsabläufe sind definiert	09.10.2017
MS 4: Architektur / Design	Die Softwarearchitektur ist komplett definiert und mehrheitlich bereits implementiert.	06.11.2017
MS 5: Ende Construction	Alle geplanten Features programmiert und getestet	11.12.2017
MS 6: Abgabe	Abschluss der Dokumentation	22.12.2017

Tabelle 2.2: Meilensteine

## 2.4 Risikomanagement

### R1: Code Datenverlust

<b>Beschreibung</b>	Der Code auf Git geht verloren
<b>Massnahme</b>	Anlegen von lokalen Arbeitskopien
<b>Verhalten bei Eintritt</b>	Lokale Arbeitskopien der Teammitglieder werden zusammengeführt und ins Git Repository hochgeladen

*Tabelle 2.3: Risiko Code Datenverlust*

### R2: Dokumentenverlust

<b>Beschreibung</b>	Dokumente in OneDrive gehen verloren
<b>Massnahme</b>	Wöchentlich lokale Kopien und Versionierung der Dokumente
<b>Verhalten bei Eintritt</b>	Zurückspielen der lokalen Datensicherung

*Tabelle 2.4: Risiko Dokumentenverlust*

### R3: Ausfall Redmine

<b>Beschreibung</b>	Redmine ist nicht erreichbar
<b>Massnahme</b>	Keine Experimente auf dem System mit der Redmine-Installation
<b>Verhalten bei Eintritt</b>	Server neustarten, Problem analysieren und innert nützlicher Frist beheben

*Tabelle 2.5: Risiko Ausfall Redmine*

### R4: Datenverlust Redmine

<b>Beschreibung</b>	Redmine-Datenbank beschädigt oder inkonsistent
<b>Massnahme</b>	Wöchentliche CSV-Exports aller wichtigen Redmine-Komponenten
<b>Verhalten bei Eintritt</b>	Je nach Fehler den Server neu aufsetzen, Redmine neu installieren, neusten CSV-Export in Redmine importieren

*Tabelle 2.6: Risiko Datenverlust Redmine*

### R5: Technologie Einarbeitung

<b>Beschreibung</b>	Zu wenig Erfahrung mit den eingesetzten Technologien und Frameworks
<b>Massnahme</b>	Einarbeitung in Themen mit Tutorials und kleineren Testprojekten
<b>Verhalten bei Eintritt</b>	Kontakt zu Personen suchen, die mit entsprechender Technologie vertraut sind

*Tabelle 2.7: Risiko Technologie Einarbeitung*



#### R6: Ungenügende Kommunikation

<b>Beschreibung</b>	Kommunikation mit Kunde / Betreuer erweist sich als ungenügend oder schwierig
<b>Massnahme</b>	Regelmässige Planung von Sitzungen
<b>Verhalten bei Eintritt</b>	Planung der Sitzungen überarbeiten, Kontakt mit Kunde / Betreuer aufnehmen

*Tabelle 2.8: Risiko Ungenügende Kommunikation*

#### R7: Unterschätzte Komplexität

<b>Beschreibung</b>	Teammitglied braucht viel mehr Zeit als geplant für ein Arbeitspaket
<b>Massnahme</b>	Arbeitspakete mit Zeitreserven planen, Backlog regelmässig kontrollieren
<b>Verhalten bei Eintritt</b>	Arbeitspakete in kleinere Teilpakete aufteilen oder schlimmstenfalls auf Feature verzichten

*Tabelle 2.9: Risiko Unterschätzte Komplexität*

#### R8: Ausfall Entwickler-PC

<b>Beschreibung</b>	PC eines Teammitglieds funktioniert nicht mehr und verhindert die Weiterarbeit am Projekt
<b>Massnahme</b>	Versionsverwaltungs-Werkzeug, Dokumentationen für die Installation wichtiger Komponenten, allgemeine Regeln im Umgang mit PCs beachten
<b>Verhalten bei Eintritt</b>	PC neu installieren oder Ersatz besorgen, Entwicklerumgebung gemäss Dokumentation einrichten und alle aktuellen Versionen des Projektes ins System importieren

*Tabelle 2.10: Risiko Ausfall Entwickler-PC*

## 3 KONZEPTION UND DESIGN

### 3.1 Webapplikation

Die Webapplikation stellt alle Applikationsdaten zur Verfügung und verwaltet diese. Zusätzlich führt es die Authentifizierung von Benutzern mittels JSON Web Token (JWT) durch. Ausserdem bietet die Webapplikation eine API um Standortdaten der Tracker-App zu empfangen und in der Datenbank abzulegen. Die Applikationsdaten werden in einer MySQL-Datenbank persistiert, welche über die Java Persistence API (JPA) mit der Webapplikation verbunden ist.

#### 3.1.1 Java Spring Boot

Durch die Vorarbeit eines Studenten, der bereits einen Teil der Webapplikation auf Basis von Spring Boot als Bachelorarbeit implementiert hat, wurde mit dieser Technologie fortgefahren. Zudem wurde in der Aufgabenstellung definiert, dass die Webapplikation weiterentwickelt werden soll und nicht von Grund auf etwas Neues implementiert wird.

#### 3.1.2 MySQL-Datenbank

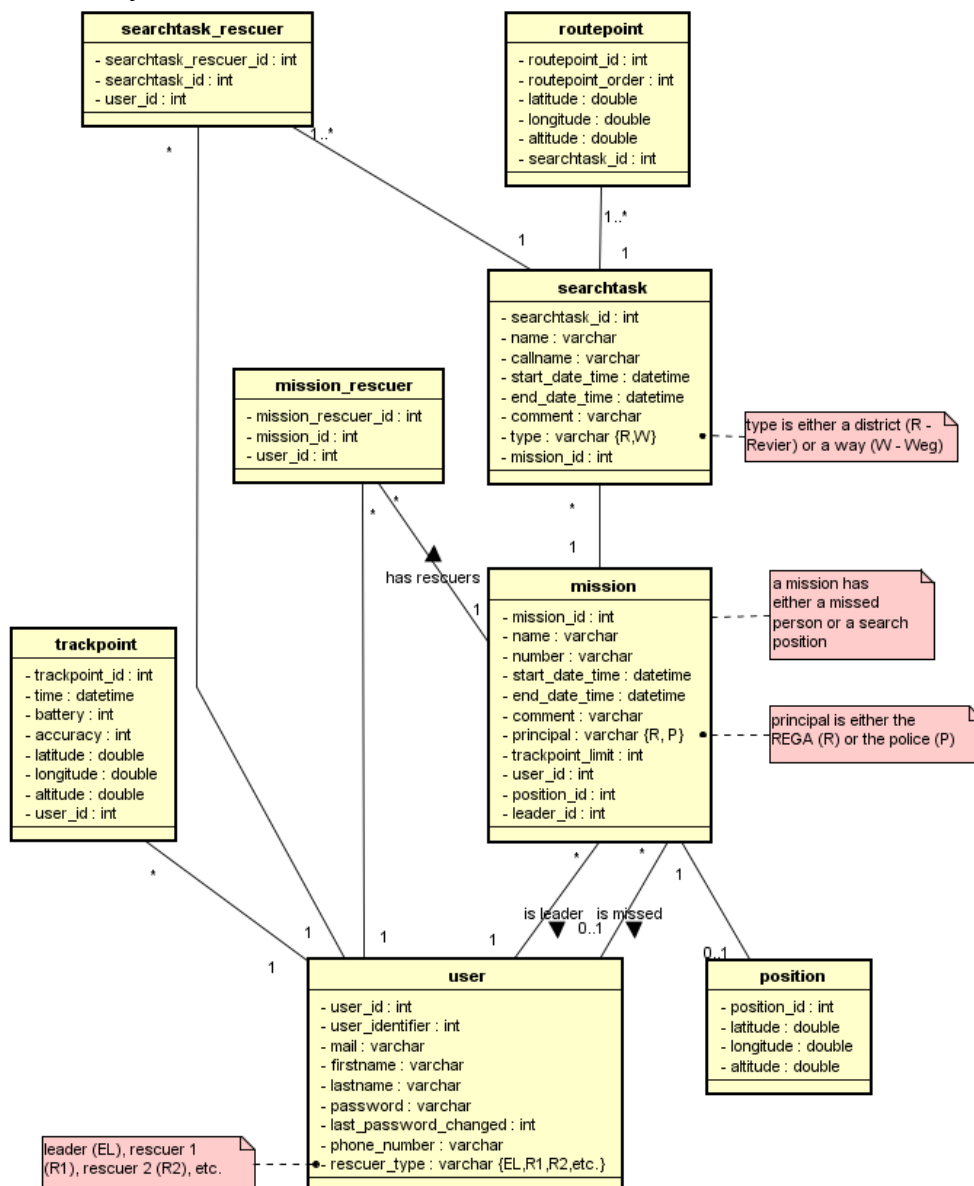


Abbildung 3.1: Datenbankmodell

## 3.2 Retter-App

Die zu einsetzende Technologie wurde durch die vorgegebene Aufgabenstellung mit NativeScript als Cross-Plattform Framework bereits definiert.

### NativeScript

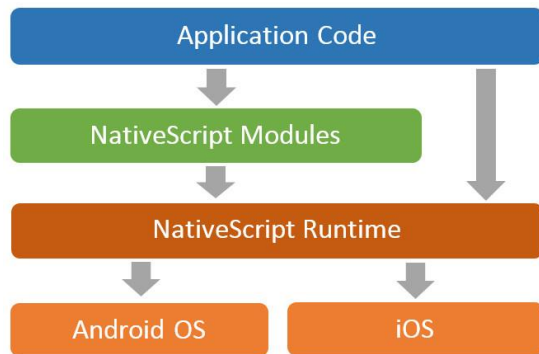


Abbildung 3.2: NativeScript Funktionsweise

<sup>1</sup> Das NativeScript Framework erlaubt es, Android und iOS Apps in TypeScript oder JavaScript zu entwickeln. Alle Elemente die implementiert werden, können beim Kompilieren für beide Betriebssysteme verwendet werden. NativeScript produziert Apps, welche die gleichen APIs verwenden, wie Apps die in Android Studio oder Xcode entwickelt wurden.

Allgemeine Plugins können über den Paketmanager «npm» installiert werden. Spezielle Plugins, die weitere Systemfunktionen erfordern, können mittels «tns

plugin» installiert werden. Projekte werden über die Kommandozeile erstellt, angepasst und kompiliert.

Die Benutzeroberflächen werden durch plattformunabhängige XML-Dateien erstellt. Mit NativeScript werden diese anschliessend in ihre nativen Elemente auf der jeweiligen Plattform übersetzt. NativeScript wird in einer Node.js Laufzeitumgebung ausgeführt. Eine Installation von Node.js ist deshalb zwingend nötig.

<sup>1</sup> Bildquelle: <https://1.f.ix.de/scale/geometry/700/q75/developer/imgs/06/1/8/6/2/7/0/8/tamhan-n-1-ee415fdd38deaed8.png>

### 3.3 Tracking-App

Die Tracking-App auf Basis von Android wurde grösstenteils bereits von einem Mitarbeiter der Namics AG entwickelt. Unsere Hauptaufgabe bestand darin, diese zu testen und anschliessend mögliche Verbesserungen einfließen zu lassen. Die ermittelten Standortdaten werden als einzelnes JSON-Objekt oder als Array von JSON-Objekten an die API gesendet. Auf dem Request muss der Header «Content-Type: application/json» gesetzt sein und der Body hat folgende Struktur:

```
[
  {
    "battery":80,
    "coords":{
      "accuracy":20,
      "lat":47.002568,
      "lng":9.507347
    },
    "timestamp":1511946077506,
    "user":" admin@tracking-system.ch"
  },
  {
    "battery":79,
    "coords":{
      "accuracy":12,
      "lat":47.002534,
      "lng":9.507219
    },
    "timestamp":1511947366907,
    "user":" admin@tracking-system.ch"
  }
]
```

Wenn in der Webapplikation bereits ein Benutzer mit der im JSON vorhandenen E-Mail-Adresse existiert, dann werden die GPS-Positionen dem entsprechenden Benutzer hinzugefügt. Andernfalls wird anhand der im JSON vorhandenen E-Mail-Adresse ein neuer Benutzer erstellt. Um im Notfall dem Einsatzleiter die Benutzersuche zu erleichtern, ist es wichtig, dass die Benutzer der Tracking-App eine gültige und eindeutig zuweisbare E-Mail-Adresse angeben. Zum jetzigen Zeitpunkt kann nicht verhindert werden, dass mehrere Benutzer dieselbe Adresse angeben. Die Tracking-App überprüft nicht, ob bereits ein anderer Benutzer sich mit der angegebenen E-Mail-Adresse identifiziert hat. Dazu müsste zuerst die ganze Registration definiert und implementiert werden.

## 4 UI KONZEPT

### 4.1 Mockups

#### Webapplikation

Durch Weiterverwendung des vorhandenen Designs wurden keine Mockups erstellt.

#### Retter-App

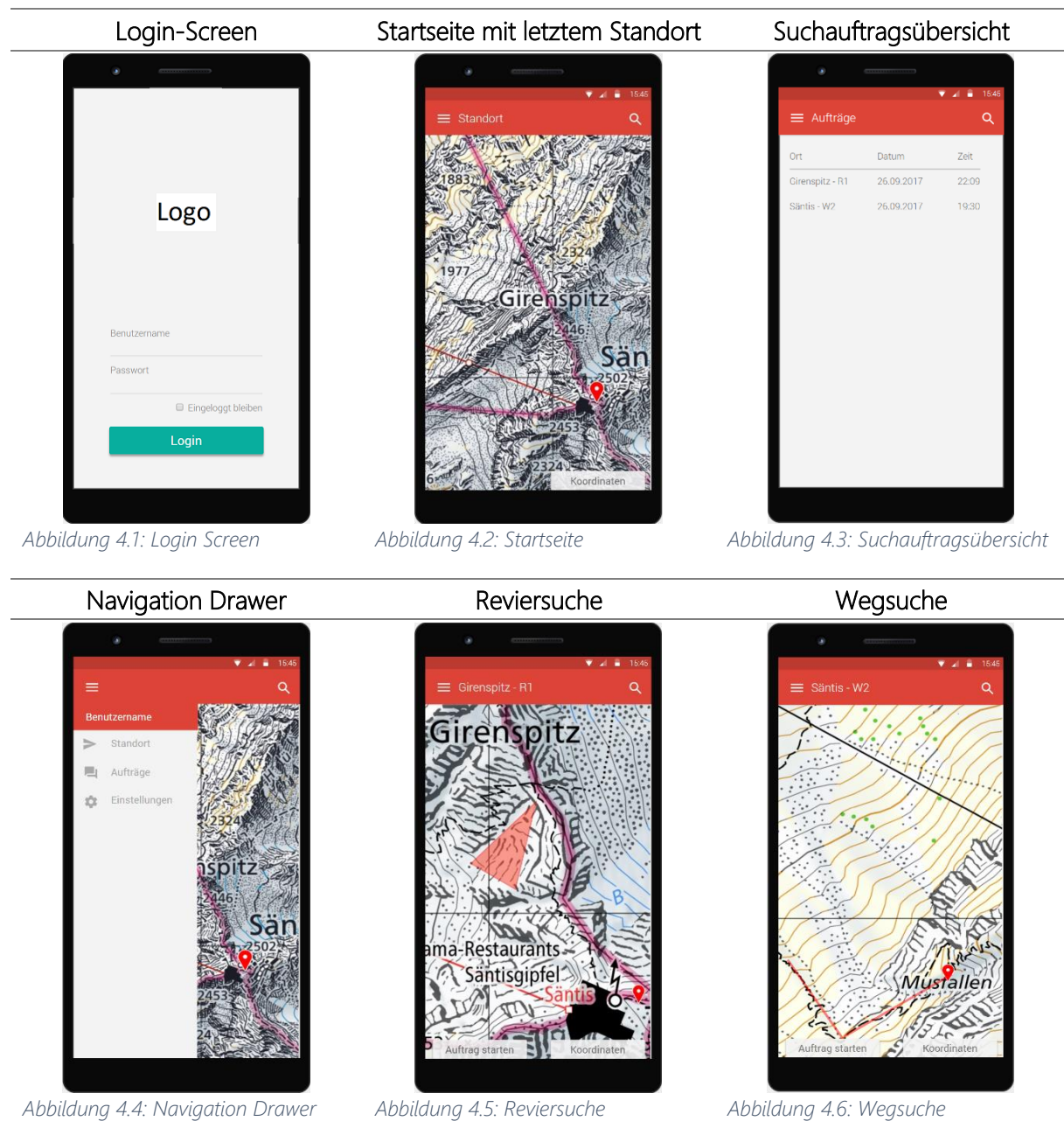


Tabelle 4.1: UI Mockups Retter-App

#### Tracking-App

Das vorhandene Design wurde belassen, da die Funktionalität im Vordergrund steht und nicht das Optische.

## 4.2 Webapplikation

### Allgemein

Durch Verwendung des Bootstraps Frameworks wird das bestmögliche Layout für jedes Gerät garantiert. Bei mobilen Geräten werden weniger wichtige Details ausgeblendet, damit die Seiten nicht überladen wirken.

### Login

Es können sich nur Personen anmelden, die gültige Zugangsdaten haben. Eine Registration ist derzeit nicht möglich, da die Tracking-Daten besonders schützenswert sind. Die Zugangsdaten werden von einem Einsatzleiter erfasst und an die Retter weitergegeben. Die Retter können die Zugangsdaten später in ihrem Profil verwalten und gemäss ihren Wünschen anpassen.

### Startseite

Zeigt eine Übersicht aller mit der Tracking-App ausgestatteten Benutzer, vorausgesetzt man hat die entsprechenden-Rechte. Andernfalls sieht man nur den eigenen Benutzer und hat keinen Zugriff auf die GPS-Daten anderer Personen. Bei jedem Benutzer hat man die Möglichkeit, den Track mit den letzten 100 Positionen anzuzeigen, das Profil zu bearbeiten oder den Benutzer zu löschen. Wenn jemand ein besonders kurzes Intervall für die Synchronisation der GPS-Daten in der Tracking-App eingestellt hat, so kann der Zeitraum für den Track auch auf Minuten oder Stunden heruntersetzt werden. Standardmässig ist der Zeitraum auf Tage eingestellt, da dies in den meisten Fällen reicht. Mit zunehmender Anzahl Benutzer wird die Liste dementsprechend grösser und unübersichtlicher. Eine Suche erlaubt das schnelle Auffinden eines spezifischen Benutzers oder es können Filter auf die verschiedenen Retter-Typen angewendet werden. Der Button «Letzte Positionen anzeigen» setzt für jeden in der Liste angezeigten Benutzer einen Marker auf der Karte. Durch Klicken auf den Marker werden in einem Popover zusätzliche nützliche Details angezeigt. Möchte der Einsatzleiter wissen, wo sich seine Retter befinden, kann er einen Filter auf den entsprechenden Retter-Typ setzen und mit Bestätigen durch «Letzte Positionen anzeigen» werden diese auf der Karte dargestellt. Der Marker ist gelb, wenn es sich um einen Retter handelt und rot, falls es sich um einen normalen Benutzer handelt. Durch diese Massnahme wird eine Verwechslung von Rettern und Benutzern ausgeschlossen. Alle Daten werden standardmässig auf der Swisstopo Karte angezeigt. Diese Karte ist für die Schweiz die aktuell genaueste mit wichtigen Details wie Höhenlinien, Schuttkegeln und Felsabsätzen. Alternativ kann auch auf Google Maps gewechselt werden. Dies ist dann sinnvoll, wenn die API GPS-Daten aus dem Ausland empfängt. Google Maps kann weltweit GPS-Positionen anzeigen, die Swisstopo Karte hingegen nur innerhalb der Schweiz und dem nahen Ausland.

## Einsätze

Auf den ersten Blick sind die offenen Einsätze mit den wichtigsten Details sichtbar. Die Seite wurde schlicht gehalten, damit nicht vom Wesentlichen abgelenkt wird. Abgeschlossene Einsätze können über ein Dropdown eingeblendet werden und zeigen die gleichen Details wie offene Einsätze. Sind keine offenen Einsätze vorhanden, wird der Benutzer mit der entsprechenden Meldung darüber informiert – gleiches gilt bei den offenen Einsätzen. Ein neuer Einsatz kann über das entsprechende Dropdown erstellt werden. Auf der Karte werden die letzten Positionen aller Retter mit einem Marker und dem Namen des Retters dargestellt. Dadurch kann der Einsatzleiter die an einem Einsatzort nahegelegensten Retter bereits dem Einsatz zuweisen. Durch Klicken auf die Karte wird ein Einsatzpunkt erstellt. Sind dem Einsatzleiter die Koordinaten der vermissten Person bekannt, kann er die Koordinaten im LV03 oder WGS84 Format eingeben und ein entsprechender Einsatzpunkt wird auf der Karte gesetzt. Über ein Google API wird anhand der Koordinaten automatisch die Höhe ausgerechnet und dargestellt. Ist die vermisste Person im System erfasst, kann diese über ein Feld – mit typeahead für die effiziente Suche – ausgewählt werden. Dadurch wird zusätzlich der Track dieser Person auf der Karte angezeigt. Auch hier kann wieder über einen Filter der Zeitraum für den Track eingeschränkt werden. Ferner können der Karte Layer für Wanderwege und Wildruhezonen hinzugefügt werden. Der Liste für die Retter ist eine fixe Höhe zugewiesen, damit diese bei einer grossen Anzahl Retter nicht zu viel Platz einnimmt und dadurch unübersichtlich wird. Auch hier erlaubt ein Suchfeld wieder die schnelle und effiziente Suche nach einem Retter, der dem Einsatz zugewiesen werden soll.

## Suchaufträge

Die Seite ist gleich aufgebaut wie bei den Einsätzen mit dem Unterschied, dass automatisch der zuletzt erstellte offene Einsatz in der Karte angezeigt wird. Der Benutzer findet daher einen vertrauten Seitenaufbau vor und die Erlernbarkeit wird erleichtert. Auf der Karte werden alle dem Einsatz zugewiesenen Retter dargestellt. Zusätzlich ist der Einsatzpunkt und / oder der vorher definierte Track der vermissten Person inkl. rotem Marker für die letzte Position sichtbar. Sind dem Einsatz bereits Suchaufträge zugewiesen, dann erscheinen diese Ebenfalls auf der Karte. Die Farbe der vorhandenen Suchaufträge wurde so gewählt, dass sie sich gut von anderen Elemente auf der Karte wie Autobahnen, Hauptstrassen und Landesgrenzen unterscheidet. Möchte der Einsatzleiter einen neuen Suchauftrag erstellen, geschieht dies über den entsprechenden Dropdown-Eintrag. Die Karte wird automatisch verkleinert und es wird auf den relevanten Bereich gezoomt. Je nach getroffener Auswahl kann in der Karte eine Weg- oder Reviersuche eingezeichnet werden. Wie bereits beim Erstellen eines Einsatzes können wieder die Gleichen zusätzlichen Layer eingeblendet werden. Dies ist besonders dann hilfreich, wenn man Kenntnis davon hat, dass die vermisste Person auf offiziellen Wanderwegen unterwegs war. Die Liste der Retter enthält alle noch nicht einem Suchauftrag zugewiesenen Retter. Damit der Einsatzleiter auch weitere Suchaufträge erstellen kann, wenn bereits alle Retter einem Suchauftrag zugewiesen sind, kann er entweder einen Suchauftrag ohne Retter erstellen oder dem Einsatz noch nicht zugewiesene Retter hinzufügen. Über einen Link auf dem Namen oder Button bei den offenen und abgeschlossenen Suchaufträgen wird man zu den Details des Suchauftrags weitergeleitet.

## Suchauftrag Details

Hier ist die Karte besonders wichtig und nimmt dementsprechend viel Platz ein. Es wird automatisch in den wichtigen Bereich, sprich die Weg- oder Reversuche gezoomt. Der Einsatzleiter kann praktisch in Echtzeit die Suche verfolgen. Nicht ganz in Echtzeit, da der in der Tracking-App konfigurierte Intervall massgebend ist. Haben die Retter in der Tracking-App das Standardintervall von 5 Minuten aktiviert, dann ist ihr Track 5 Minuten verzögert sichtbar bis die nächste Synchronisation stattfindet. Unter Umständen hat das Gerät eines Retters keine Internetverbindung und demzufolge kann der Track auch älter als 5 Minuten sein.

## Profil

Wie bei vielen anderen Seiten kann auch hier der Benutzer sein Profil verwalten. Die Wichtigsten Angaben sind hier die E-Mail-Adresse und das Passwort, weil diese für das Login in der Retter-App nötig sind. Nützlich ist auf jeden Fall auch die Angabe einer gültigen Telefonnummer, falls Einsatzleiter und Retter telefonisch in Kontakt treten müssen. Der Einsatzleiter sieht die Telefonnummer beim Erstellen eines neuen Einsatzes oder Suchauftrages. Diese ist als Telefon-Link hinterlegt, damit der Einsatzleiter direkt von der Webapplikation Kontakt aufnehmen kann – vorausgesetzt es ist ein Programm für Telefonanrufe wie z.B. Skype installiert.

## GPS-Daten

Mit zunehmender Nutzungsdauer der Tracking-App werden immer mehr GPS-Positionen im System erfasst. Jeder Nutzer mit einem Login hat die Möglichkeit seine GPS-Daten eigenständig zu löschen. Die Idee dahinter ist, dass alte GPS-Positionen mit der Zeit nicht mehr relevant sind und somit nicht unnötig Speicherplatz besetzt wird. Zudem spielt auch der Datenschutz eine Rolle. Deshalb kann der Benutzer einzelne GPS-Positionen löschen oder gleich alle. Für jede von der Tracking-App erfasste Position werden die Koordinaten im WGS84 Format, Höhe, Akku, Genauigkeit und Zeitstempel angezeigt.

## Berechtigungen

Wie bereits weiter oben erwähnt, gibt es für die Benutzer unterschiedliche Berechtigungen. Diese äussern sich darin, dass gewisse Elemente für Benutzer mit wenig Rechten nicht sichtbar sind. Dadurch wird verhindert, dass aus Versehen Daten gelöscht werden oder sensible Daten missbraucht werden. Versierte Nutzer im Umgang mit der Webapplikation wie der Einsatzleiter haben natürlich alle Rechte und brauchen diese auch, um sinnvoll mit der Einsatzplanung umgehen zu können.



## 4.3 Retter-App

### Standort

Nach erfolgreichem Login wird die aktuelle Position auf der Swisstopo Karte dargestellt. Diese Funktion ist auch hilfreich, wenn man in den Bergen unterwegs ist und wissen möchte, wo man sich gerade befindet. Es wird die letzte von der Tracking-App an den Server übermittelte Position angezeigt. Der Benutzer hat jedoch die Möglichkeit, über den Floating Action Button mit dem GPS-Symbol die aktuelle Position auszulesen. Diese wird in einer Alert-Box in WGS84- und LV03-Koordinaten inkl. Genauigkeit angezeigt und direkt auch die Abweichung in Meter vom letzten übermittelten Standort berechnet. Gleichzeitig wird der neue Standort an die Webapplikation übermittelt und über einen Websocket wird daraufhin die angezeigte Position aktualisiert.

### Aufträge

Zeigt eine Liste mit allen Suchaufträgen, die dem Retter vom Einsatzleiter zugewiesen wurden. Die Liste ist unterteilt in laufende und abgeschlossene Suchaufträge damit das Wesentliche sofort sichtbar ist. Durch Auswahl eines Auftrages werden die zugehörigen Details in einer Alert-Box dargestellt und der Retter kann den Auftrag öffnen oder zurück zur Auftragsübersicht gelangen.

### Suchauftrag

Hat der Retter einen Suchauftrag geöffnet, dann wird die vom Einsatzleiter in der Webapplikation erfasste Weg- oder Reversuche angezeigt. Zusätzlich ist wieder die eigene letzte Synchronisierte Position sichtbar und falls vorhanden der Track der in Not geratenen Person oder andernfalls ein Einsatzpunkt. Der Einsatzpunkt ist diejenige Position, an der die Person vermutet wird oder von welcher die Person einen Notruf an die Polizei oder REGA abgesetzt hat.

## 4.4 Tracking-App

### Übermittelte Positionen

Nach dem Starten der App werden in einer Liste alle erfassten GPS-Positionen angezeigt. Das «true» oder «false» in der linken Spalte zeigt an, ob die Position bereits an die API der Webapplikation übermittelt wurde. Ferner werden in der rechten Spalte zu jeder Position der Zeitstempel, die Koordinaten im WGS84 Format und der Akkustand dargestellt.

### Einstellungen

Die Einstellungen sind sehr minimalistisch gehalten und leicht verständlich. So kann nur das automatische Starten der App nach einem Neustart des Smartphones aktiviert, das Zeitintervall für Standortupdates und die E-Mail-Adresse geändert werden. Die E-Mail-Adresse dient zur Identifikation des Benutzers in der Webapplikation.

## 5 ERGEBNISSE

### 5.1 Deployment

Die Webapplikation wird bei der aspectra AG (ISO 27001 zertifiziert) in einem Schweizer Rechenzentrum gehostet. Für jedes produktive System wird mindestens ein Ersatzsystem bereitgestellt. Dadurch wird eine optimale Verfügbarkeit garantiert.

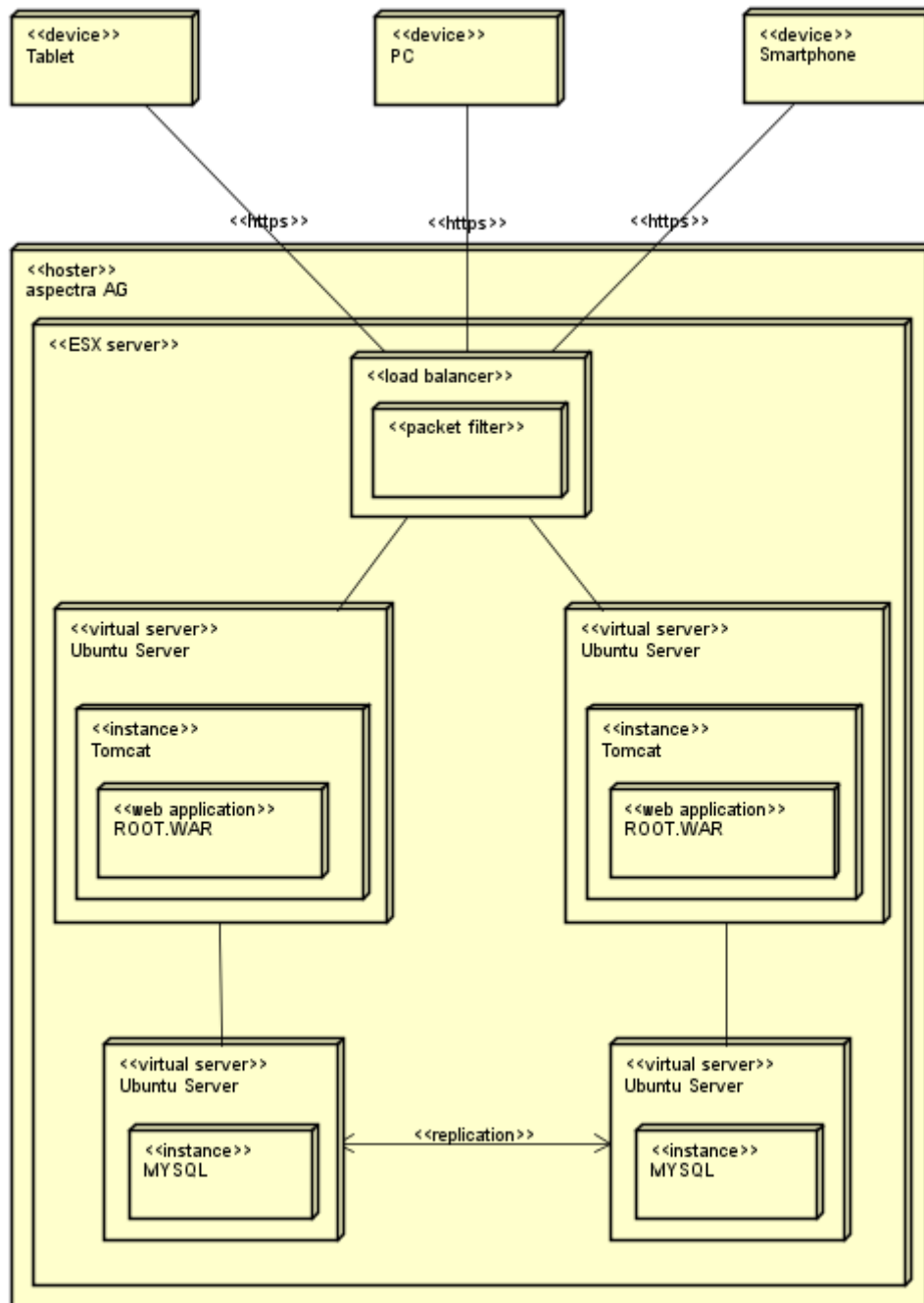


Abbildung 5.1: Deployment Diagramm

## 5.2 Webapplikation

### 5.2.1 Screenshots

#### Login

Das Login erfolgt mit E-Mail-Adresse und Passwort. Alle Anfragen auf <https://cockpit.tracking-system.ch/> werden automatisch auf /Login weitergeleitet, sofern man noch nicht angemeldet ist.

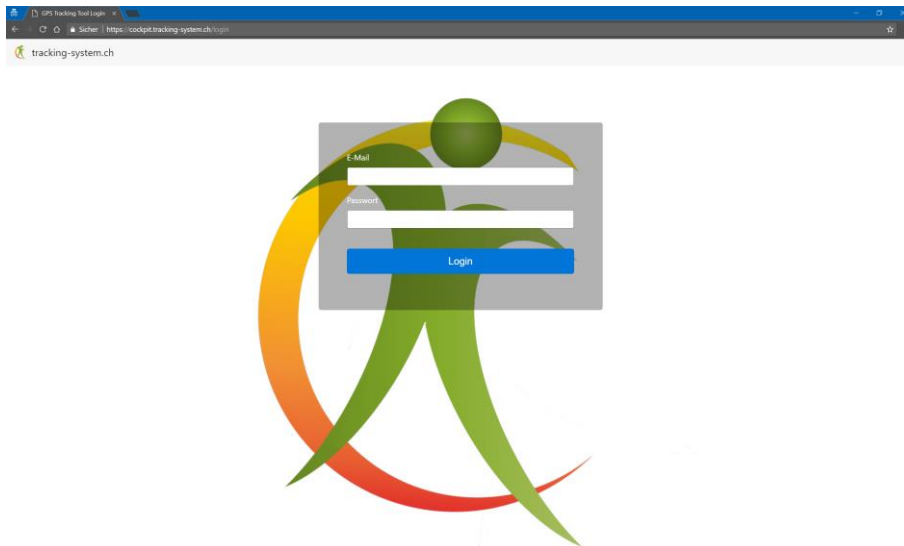


Abbildung 5.2: trs-cockpit Login

#### Übersicht

Nach erfolgreichem Login erfolgt eine Weiterleitung auf die Übersichtsseite. Hier werden alle im Tracking-System erfassten Benutzer angezeigt. Es können die letzten Positionen und Tracks aller Benutzer auf der Swisstopo Karte oder auf Google Maps dargestellt werden. Je nach Berechtigungen sieht man alle Benutzer oder nur den Eigenen.

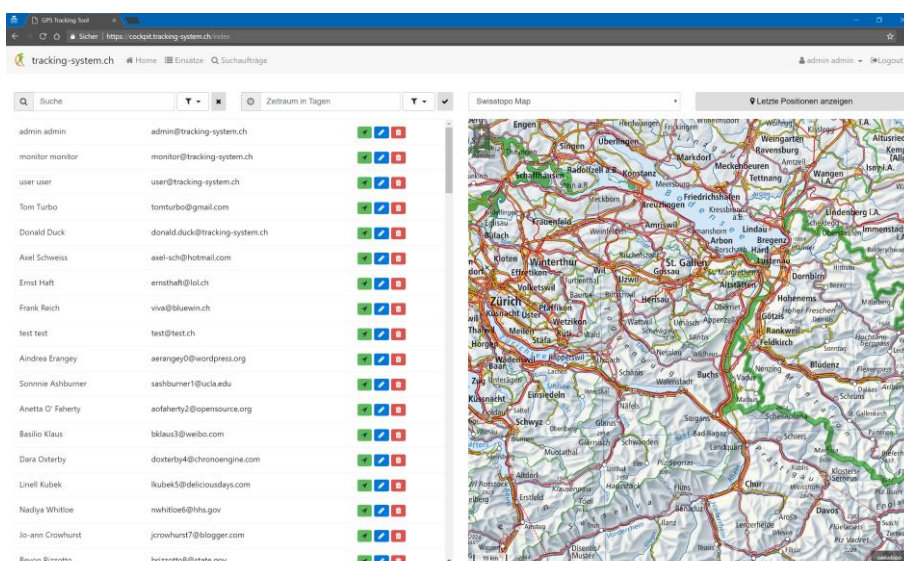


Abbildung 5.3: trs-cockpit Übersicht

## Benutzerprofil

Die wichtigsten Angaben sind die E-Mail-Adresse und die Telefonnummer. Die Mail-Adresse wird verwendet, um die von der Tracker-App empfangenen Standortdaten dem richtigen Benutzer zuzuweisen. Die Telefonnummer ist hilfreich, wenn der Einsatzleiter telefonisch Kontakt mit einer in Not geratenen / vermissten Person oder einem Retter aufnehmen möchte.

The screenshot shows the 'Profil von: David Riederer' page in the trs-cockpit. It contains a form with the following fields: VORNAME (David), NACHNAME (Riederer), E-MAIL (drieder@tr.ch), TELEFON (+41 00 000 00 00), and RETTER-TYP (EL). There is a 'PASSWORT ÄNDERN' link and two buttons at the bottom: 'Änderungen speichern' (green) and 'Abbrechen' (red).

Abbildung 5.4: trs-cockpit Benutzerprofil

## Standortdaten

Mit Administratorrechten kann man die Standortdaten jedes Benutzers sehen und löschen. Ohne die entsprechenden Rechte ist dies nur bei den eigenen Standortdaten der Fall. Die Daten sind jeweils nach Zeitstempel sortiert.

The screenshot shows the 'Tracking-Daten von: David Riederer' page. On the left is a search table with columns for name, email, and action buttons. On the right is a detailed view of tracking data for David Riederer, showing latitude, longitude, height, battery level, accuracy, and timestamp for each data point.

Nutzer	E-Mail	Daten anzeigen	Daten löschen
admin admin	admin@tracking-system.ch	[Green button]	[Red button]
monitor monitor	monitor@tracking-system.ch	[Green button]	[Red button]
user user	user@tracking-system.ch	[Green button]	[Red button]
Tom Turbo	tomturbo@gmail.com	[Green button]	[Red button]
Donald Duck	donald.duck@tracking-system.ch	[Green button]	[Red button]
Axel Schweiss	axel-sch@hotmail.com	[Green button]	[Red button]
Ernst Huft	ernsthuft@iol.ch	[Green button]	[Red button]
Frank Reich	vliva@bluewin.ch	[Green button]	[Red button]
test test	test@test.ch	[Green button]	[Red button]

Latitude	Longitude	Höhe	Akku	Genauigkeit	Zeitstempel	Aktion
46.9807971	9.5096793	1196.17 m ü. M.	31 %	18 m	2017-12-18 22:27:10.0	[Red button]
46.968683	9.5420851	687.88 m ü. M.	44 %	20 m	2017-12-17 20:23:33.0	[Red button]
46.9686937	9.5421246	686.83 m ü. M.	0 %	19 m	2017-12-17 17:46:53.0	[Red button]
46.9087347	9.5420397	689.47 m ü. M.	68 %	82 m	2017-12-17 14:36:25.0	[Red button]
46.98205	9.542266	513.86 m ü. M.	69 %	2500 m	2017-12-17 14:24:08.0	[Red button]
47.0077178	9.5021668	503.64 m ü. M.	69 %	1600 m	2017-12-17 14:19:13.0	[Red button]
46.9902249	9.4951495	820.41 m ü. M.	69 %	800 m	2017-12-17 14:14:33.0	[Red button]

Abbildung 5.5: trs-cockpit Standortdaten

## Einsatz definieren

Auf der Swisstopo Karte werden die letzten Positionen der Retter angezeigt. Auf der Karte können ausserdem zusätzliche Layer wie Wanderwege und Wildruhezonen eingeblendet werden. Bei der Zuweisung der Retter stehen jeweils alle im System erfassten Retter zur Auswahl. Durch Klicken auf die Karte wird ein Einsatzpunkt erstellt. Über eine Google-API wird automatisch die Höhe des Einsatzpunktes berechnet. Die Koordinaten des Einsatzpunktes werden jeweils im LV03- und WGS84-Format angezeigt. Ist die vermisste Person im System erfasst, so wird durch entsprechende Auswahl ihr Track dargestellt. Der Zeitraum des Tracks kann limitiert werden auf Minuten, Stunden oder Tage.

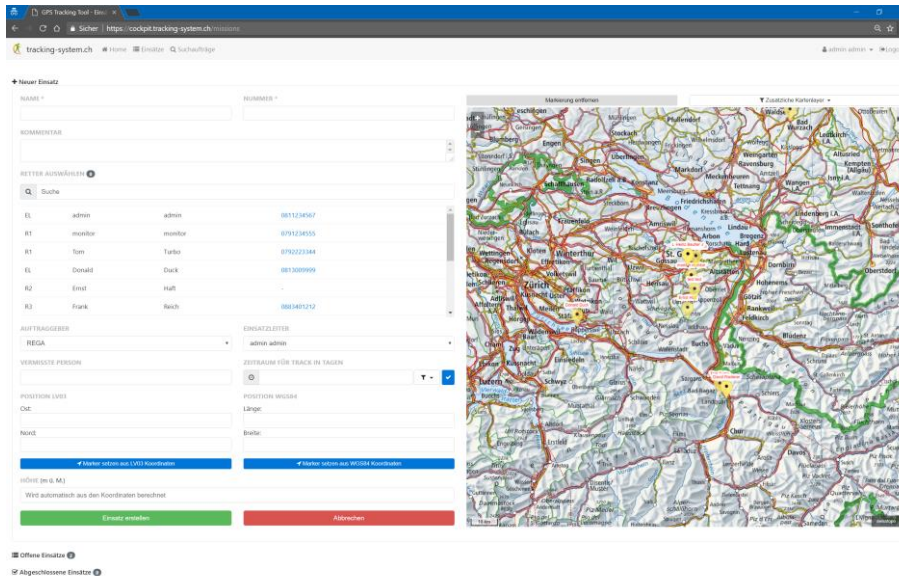


Abbildung 5.6: trs-cockpit Einsatz definieren

## Suchauftragsübersicht

Einem Einsatz können mehrere Suchaufträge zugewiesen werden. Bereits definierte Suchaufträge werden auf der Karte eingezeichnet und es wird automatisch in den relevanten Bereich gezoomt. Auf untenstehendem Bild sind bereits zwei Suchaufträge eingezeichnet. Auch der im Einsatz definierte Einsatzpunkt ist sichtbar.

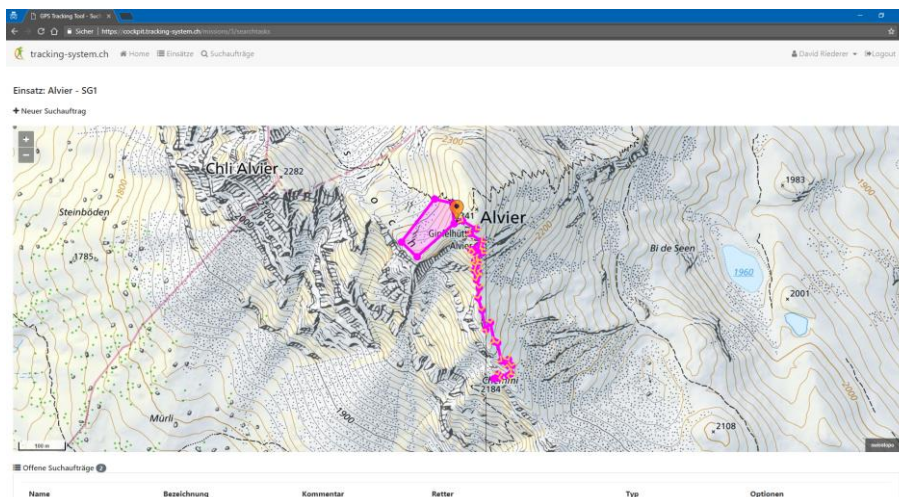


Abbildung 5.7: trs-cockpit Suchauftragsübersicht mit Einsatzpunkt



Ist die in Not geratene / vermisste Person im System erfasst, dann wird ergänzend ihr Track angezeigt. Durch den vorhandenen Track und Prognosen kann der Einsatzleiter besser abschätzen, in welchem Gebiet sich die Person aufhalten könnte. Ist die Person beispielsweise immer dem Wanderweg gefolgt, dann ist es sehr wahrscheinlich, dass auch der weitere Wegverlauf auf Wanderwegen stattgefunden hat.

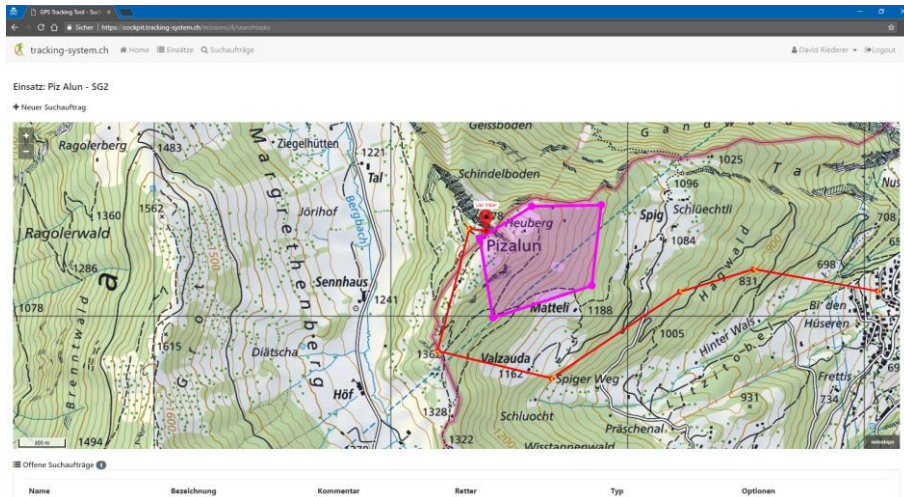


Abbildung 5.8: trs-cockpit Suchauftragsübersicht mit Track der vermissten Person

## Suchauftrag definieren

Bei einem neuen Suchauftrag werden auf der Karte alle bereits vorhandenen Suchaufträge eingezeichnet. Somit wird verhindert, dass sich mehrere Suchgebiete überlappen und dadurch Ressourcen suboptimal eingesetzt werden. Bei der Zuweisung der Retter stehen alle dem Einsatz zugewiesenen Retter abzüglich den bereits in einem Suchauftrag befindlichen Retter zur Auswahl. Ein Retter kann daher nicht gleichzeitig zwei Suchaufträgen zugewiesen werden, wenn es sich um denselben Einsatz handelt. Ebenfalls sichtbar sind die Details der offenen Suchaufträge. Retter können jederzeit einem offenen Suchauftrag hinzugefügt oder daraus entfernt werden. Das Abschiessen, Anzeigen und Löschen von Suchaufträgen ist über die Buttons unter «Optionen» möglich.

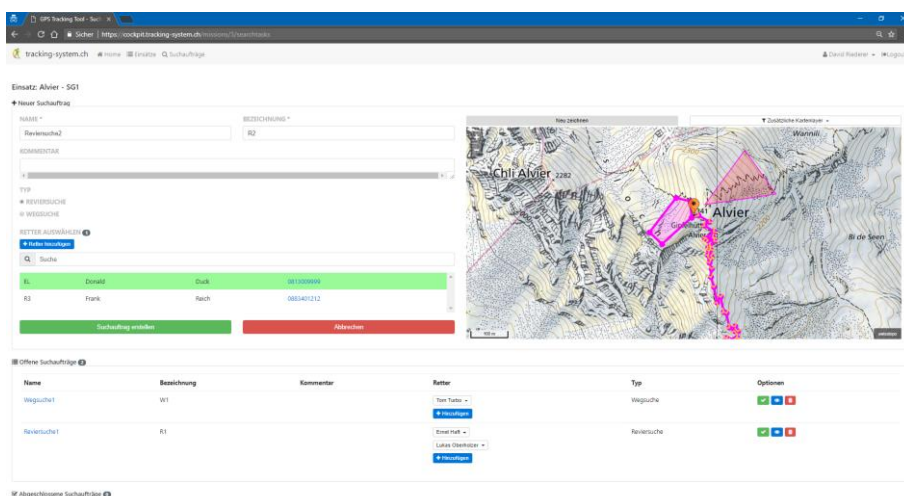


Abbildung 5.9: trs-cockpit Suchauftrag definieren

## Suchauftrag-Details anzeigen

Sobald ein dem Suchauftrag zugewiesener Retter den Auftrag in der Retter-App gestartet hat, kann der Einsatzleiter den Track der Retter mitverfolgen. Er sieht genau, wo die Retter bereits waren und kann bei der Planung eines neuen Suchauftrages diese Informationen miteinbeziehen. Im Hintergrund wird jeder Retter bei einem Websocket registriert und sobald auf neue Daten empfangen werden, wird der Track auf der Karte um den neuen Standort erweitert.

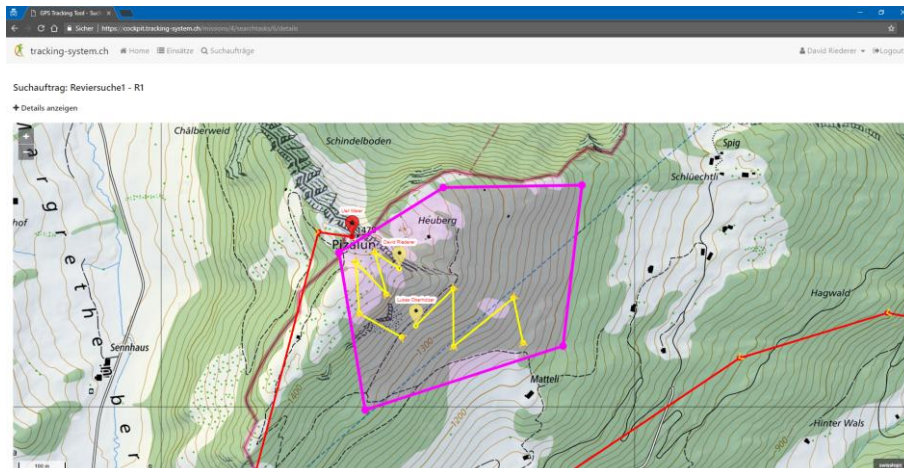


Abbildung 5.10: trs-cockpit Suchauftrag-Details

Im Reiter «Details anzeigen» sind weitere Informationen zum Suchauftrag verfügbar. Wenn die Retter den Suchauftrag über die App abschliessen, können sie einen Kommentar verfassen. Unter Umständen kann dieser für den Einsatzleiter für die weitere Planung von Nutzen sein.

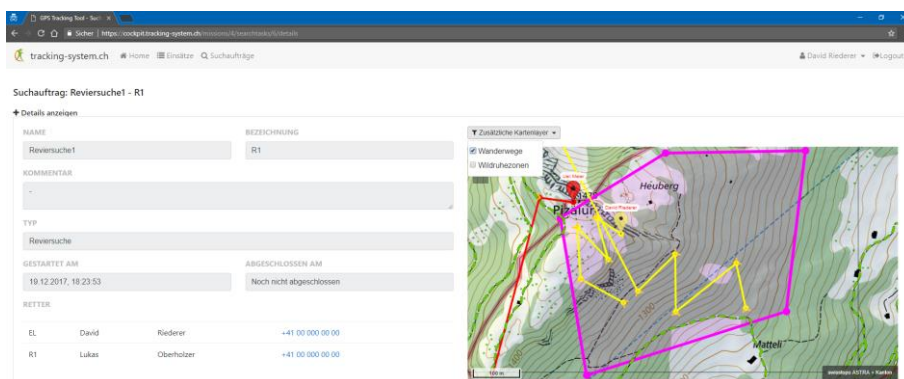


Abbildung 5.11: trs-cockpit Suchauftrag-Details anzeigen

## Bestätigung von Aktionen

Möchte der Benutzer eine Aktion durchführen, dann muss er diese meist noch bestätigen. Dadurch wird verhindert, dass aus Versehen unbeabsichtigte Interaktionen durchgeführt werden, die beispielsweise einen Einfluss auf offene Einsätze oder Suchaufträge haben.

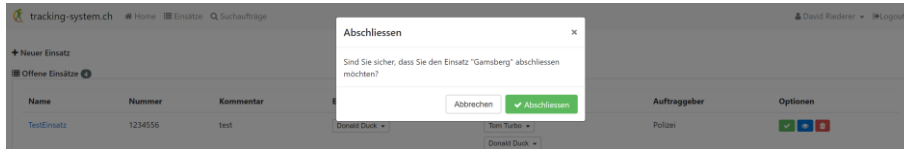


Abbildung 5.12: trs-cockpit Bestätigung von Aktionen

### 5.2.2 Code-Metriken

Die Code-Metriken wurden mit dem IntelliJ Plugin «MetricsReloaded» analysiert und folgendermassen ausgewertet. Die Test Coverage ist mit nur 50% aller Zeilen relativ tief. Dies ist darauf zurückzuführen, dass getter und setter Methoden, sowie simple Servicemethoden, die nur eine Funktion aus dem Repository aufrufen und zurückgeben, nicht getestet wurden.

		Vorhanden	Aktuell	Differenz
Non-comment Lines of Code	Java	1'081	3'706	2'625
	JavaScript	610	2'481	1'871
	Java Server Page	579	2'059	1'480
	SQL	0	1'174	1'174
	Diverses	354	392	38
	<b>Total</b>	<b>2'624</b>	<b>9'812</b>	<b>7'188</b>
Number of Classes	Product classes	23	47	24
	Test classes	2	9	7
	<b>Total</b>	<b>25</b>	<b>56</b>	<b>31</b>
Test Coverage	Classes	35%	63%	+28%
	Lines	26%	50%	+24%

Tabelle 5.1: Code-Metriken Webapplikation



## 5.3 Retter-App

### 5.3.1 Screenshots

Login-Screen



Abbildung 5.13: Login Screen

Startseite mit letztem Standort



Abbildung 5.14: Standort

Info über aktuellen Standort

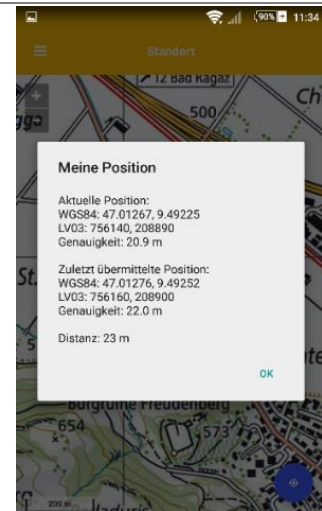


Abbildung 5.15: Suchaufträge

Das Login erfolgt anhand der E-Mail-Adresse und Passwort. Bei Bestätigung von «Eingeloggt bleiben?», wird das Login beim nächsten App-Start automatisch ausgeführt.

Nach erfolgreicher Authentifizierung wird auf der Karte der letzte an den Server gesendete Standort angezeigt.

Durch Klicken auf den Floating Action Button unten Rechts, wird der aktuelle Standort ausgelesen und zum Server übermittelt.

Navigation Drawer

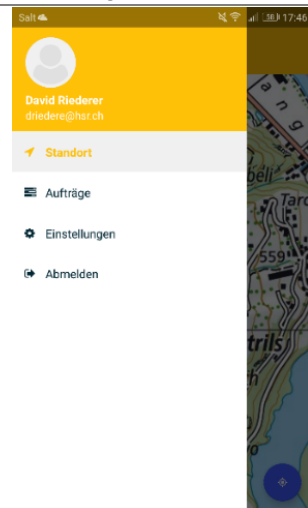


Abbildung 5.16: Navigation Drawer

Suchauftragsübersicht

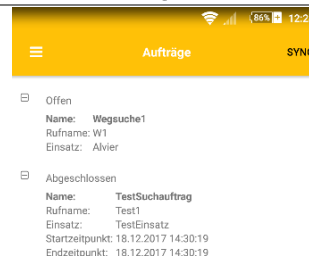


Abbildung 5.17: Reversuche

Suchauftrags-Details

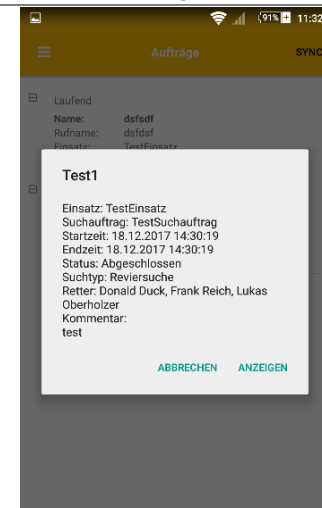
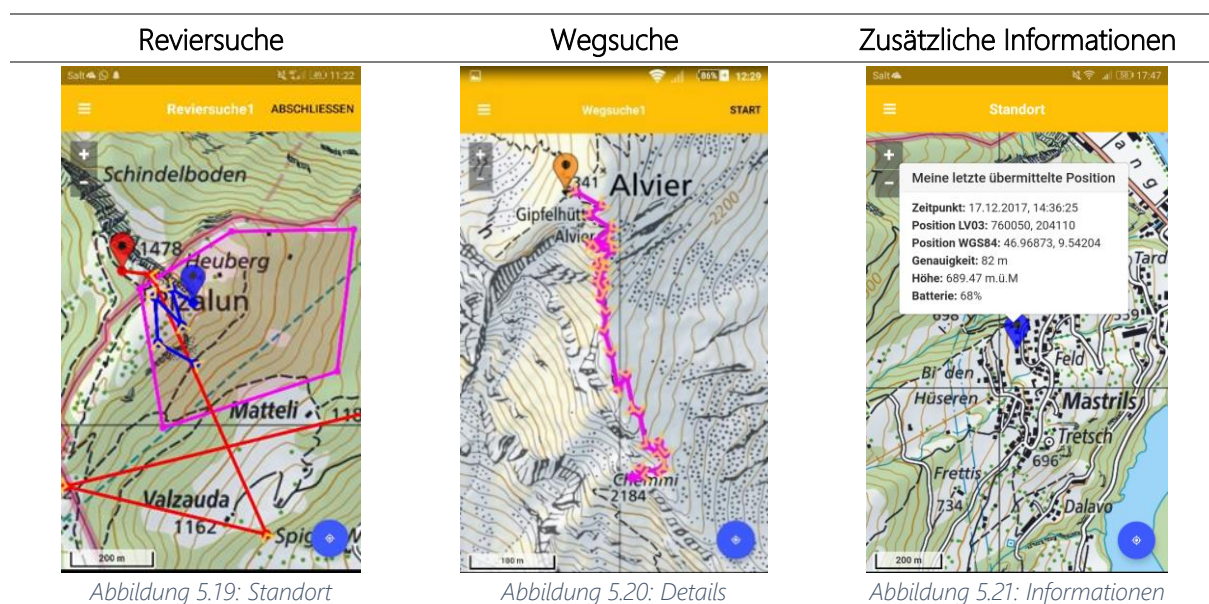


Abbildung 5.18: Wegsuche

Der Navigation Drawer wurde gemäss Mockup umgesetzt. Die Navigation ist schnell erreichbar und die wichtigen Elemente sind darin vorhanden.

Suchaufträge können über das «+» oder «-»-Zeichen ein- und ausgeblendet werden. Dies ist hilfreich, wenn die Liste mit zunehmender Nutzung länger wird.

Bei Auswahl eines Suchauftrages erscheinen die Details. Durch «Anzeigen» wird die Weg- oder Reversuche geladen.



Beispiel einer vom Einsatzleiter definierten Reversuche. Er kann genau nachvollziehen wo im Suchgebiet er bereits war und wohin er noch muss.

Beispiel einer vom Einsatzleiter definierten Wegsuche.

Klickt der Benutzer auf den blauen Marker mit der letzten Position, dann werden in einem Popover weitere Informationen angezeigt.

Tabelle 5.2: trs-rescuer Screenshots

Das Design aus den Mockups konnte sehr gut umgesetzt werden. Zusätzlich wurden noch neue nützliche Features implementiert. Dazu gehören das Übermitteln der aktuellen Position und Poppers für weitere hilfreiche Informationen.

### 5.3.2 Code-Metriken

Die Code-Metriken wurden mit dem Webstorm Plugin «MetricsReloaded» analysiert und folgendermassen ausgewertet. Die Test Coverage konnte nicht ausgewertet werden, da die Tests mittels «tns test android» ausgeführt werden und anschliessend kein Report generiert wird, sondern nur angezeigt wird, welche Tests erfolgreich und welche nicht erfolgreich waren.

		Aktuell
Non-comment Lines of Code	TypeScript	1'543
	XML	415
	SCSS	183
	Diverses	218
	Total	2'359
Number of Classes	Product classes	23
	Test classes	5
	Total	28

Tabelle 5.3: Code-Metriken Retter-App

## 5.4 Tracking-App

### 5.4.1 Screenshots

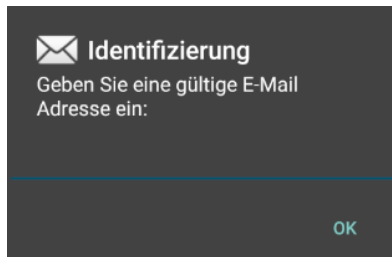


Abbildung 5.22: E-Mail Identifizierung

Beim ersten Start der App wird man aufgefordert, eine E-Mail-Adresse anzugeben. Diese wird verwendet, um die Standortdaten einem Benutzer auf dem Server zuzuweisen. Handelt es sich um eine noch nicht bekannte E-Mail-Adresse auf dem Server, dann wird ein neuer Benutzer mit der soeben angegebenen Mail-Adresse erstellt.

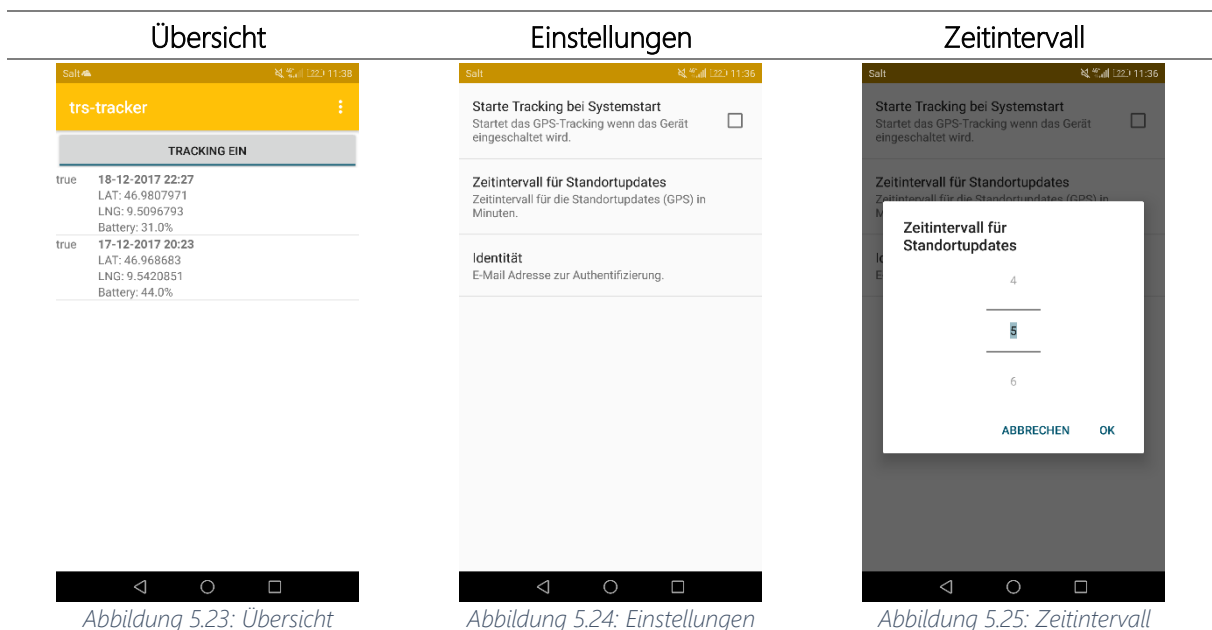


Abbildung 5.23: Übersicht

Abbildung 5.24: Einstellungen

Abbildung 5.25: Zeitintervall

Die Farben sind an die Rettungsorganisation angepasst. Neu wird auch der Akkuladestand ausgelesen und an den Server übermittelt. Trackt der Service einen neuen Standort, so erscheint dieser ohne einen Reload in der Liste.

Die Einstellungen sind auf Deutsch. Es kann nur noch ein Intervall konfiguriert werden. Dies führt zu weniger Verwirrung beim Benutzer.

Das einzustellende Intervall wird für das Auslesen der Standortdaten und die Synchronisation verwendet. Mit einem modifizierten NumberPicker kann das Intervall in Minuten einfach und unkompliziert ausgewählt werden.

Tabelle 5.4: trs-tracker Screenshots

### 5.4.2 Code-Metriken

Die Code-Metriken wurden mit dem Android Studio Plugin «MetricsReloaded» analysiert und folgendermassen ausgewertet.

		Vorhanden	Aktuell	Differenz
Non-comment Lines of Code	Java	979	1'193	214
	XML	175	196	21
	Diverses	53	83	30
	<b>Total</b>	<b>1'207</b>	<b>1'456</b>	<b>265</b>
Number of Classes	Product classes	23	28	5
	<b>Total</b>	<b>23</b>	<b>28</b>	<b>5</b>

Tabelle 5.5: Code-Metriken Tracking-App

## 5.5 Zielerreichung

Die während der Studienarbeit entwickelten Applikationen entsprechen dem aktuellen Stand der Technik und erfüllen alle vom Kunden erwarteten Anforderungen.

Folgende Punkte wurden erreicht:

- Die Retter-App – entwickelt mit NativeScript – unterstützte die Betriebssysteme Android und iOS. Kartenausschnitte werden bis zu einem gewissen Zoom-Faktor vorgeladen und daher kann die App auch ohne ständige Internetverbindung verwendet werden.
- Die Webapplikation verwendet das Bootstrap-Framework und ist daher auf allen Geräten sehr benutzerfreundlich umgesetzt. Zudem werden alle gängigen Browser unterstützt. Die vorhandene Basis (Spring Framework) wurde weiterentwickelt und gemäss den Wünschen der Auftraggeber angepasst.
- Die vorhandene Datenhaltung in MongoDB wurde abgelöst und durch MySQL ersetzt.
- Die Zuverlässigkeit von Standortupdates in der Tracking-App wurde verbessert und dies bei gleichbleibendem Akkuverbrauch. Das User Interface wurde auf Deutsch umgestellt, die Einstellungen sind benutzerfreundlicher und entsprechen den Google Material Design Kriterien.

Dem Auftraggeber kann eine gute Basis für die Weiterentwicklung der Applikationen übergeben werden.

## 5.6 Schlussfolgerung

### 5.6.1 Projektverlauf

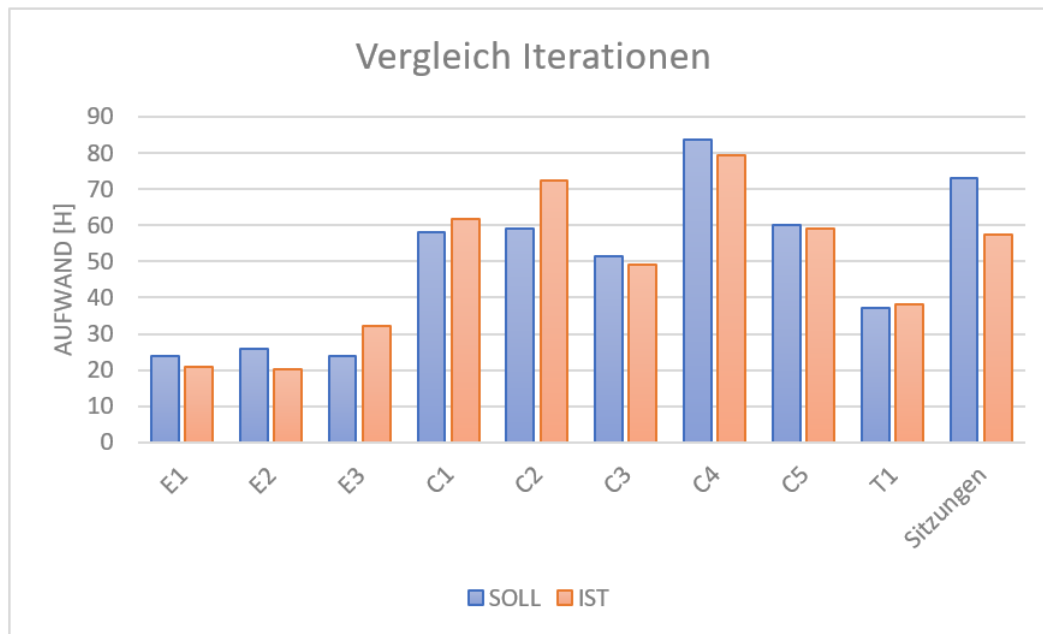


Abbildung 5.26: Projektverlauf

Mit einem Totalaufwand von 490.5 Stunden liegt das Projekt im erwarteten Rahmen einer Studienarbeit. Da die Kommunikation und Zusammenarbeit mit den Auftraggebern so reibungslos funktionierte, mussten neben den Meetings in St. Gallen keine anderen grösseren Sitzungen einberufen werden. Daher liegt der Aufwand an Sitzungen etwas tiefer als erwartet. Die gewonnene Zeit konnte daher in die Entwicklung weiterer Features in der Construction-Phase investiert werden.

Dass der zeitliche Aufwand in den Construction-Iterationen höher ist als in den Elaboration-Iterationen hat in erster Linie damit zu tun, dass in der Elaboration-Phase nur einwöchige Iterationen durchgeführt wurden, anschliessend bis auf «C5» zweiwöchige. Rechnet man aber den wöchentlichen Aufwand, so liegt dieser in der Elaboration-Phase dennoch leicht tiefer. Dies war so beabsichtigt. Zu Beginn war es wichtig, alle Anforderungen präzise mit dem Auftraggeber zu definieren, was am effizientesten in einem wöchentlichen Meeting stattgefunden hat. Daraufhin konnten wir dann in der Construction-Phase selbstständiger die besprochenen Anforderungen umsetzen, wobei in den immer noch wöchentlichen Sitzungen Anforderungen auch agil besprochen und allenfalls leicht angepasst werden konnten.

## 5.6.2 Fazit zur Webapplikation

Die Weiterentwicklung der bestehenden Webapplikation auf Basis von Java Spring Boot war anspruchsvoll aber hat auch sehr viel Spass gemacht. Am Anfang brauchte es eine gewisse Zeit, bis wir uns an diese Technologie gewöhnt hatten. Wir hatten noch keine Erfahrungen damit und es war alles neu für uns. Bei Herausforderungen findet man im Internet viele Hilfestellungen und Lösungsansätze. Auch bei den Mitarbeitern der Namics konnten wir jederzeit nachfragen und es wurde uns gerne geholfen. Die Umstellung der Datenbank von MongoDB zu MySQL hat uns am Anfang einige Schwierigkeiten bereitet, weil durch die Flyway<sup>1</sup>-Integration und Anpassungen an den SQL-Skripten jeweils die Checksummen in der Datenbank nicht mehr korrekt waren und eine Migration von neuen Daten verhindert wurde. Durch eine saubere Versionierung der SQL-Skripte haben wir dieses Problem dann gelöst. Für die Übertragung von Objekten zu Datenbankeinträgen haben wir die Java Persistence API (JPA)<sup>2</sup> verwendet, welche wir im Modul Datenbanksysteme 2 kennengelernt hatten. Bei der Erstellung der JSP-Seiten wurden wir mit der JavaServer Pages Standard Tag Library (JSTL) konfrontiert. Der sichere Umgang mit dieser Library war uns schnell vertraut und hat deshalb kaum Probleme bereitet. Für das Design des Frontend haben wir bewährte Techniken wie Bootstrap und jQuery eingesetzt. Diese Frameworks waren uns aus privaten Projekten bereits bestens bekannt. Gegen Ende des Projektes haben wir alle Aktualisierungen von Positionen und Tracks mit Hilfe von Websockets implementiert, damit die Seite nicht ständig manuell aktualisiert werden muss. Besondere Vorsicht war dort bei der korrekten Benutzung von subscribe und unsubscribe auf die richtigen Benutzer-IDs geboten, um nicht falsche Daten anzuzeigen.

Abschliessend kann gesagt werden, dass wir viele neue und spannende Web-Technologien kennengelernt haben und diese sicher auch in zukünftigen Projekten einsetzen werden.

## 5.6.3 Fazit zur Tracking-App

Die Tracking-App war bereits mehrheitlich implementiert als wir mit der Studienarbeit begonnen haben. Zum Start des Projektes haben wir den Fokus auf die Webapplikation und die Retter-App gelegt. Erst gegen Mitte haben wir mit dem aktiven Testing der Tracking-App angefangen. Wir haben gemeinsam eine Wanderung auf den Alvier unternommen und uns mit der App tracken lassen. Auf dem Gipfel angekommen haben wir festgestellt, dass die Standortupdates nicht regelmässig, sprich nicht immer im definierten Zeitintervall an den Server gesendet werden. Wieder zu Hause angekommen haben wir uns diesem Problem angenommen. Durch ausführliches Debugging war schnell klar, welcher Service nicht zuverlässig funktionierte. Recherchen im Internet haben unsere Feststellung bezüglich des Service bestätigt. Daraufhin wurde dieser durch Google Play Services ersetzt und siehe da, alles funktionierte wie es sollte. Da die App möglichst benutzerfreundlich sein sollte, wurden auch am User Interface noch einige Änderungen vorgenommen - wir verweisen an dieser Stelle auf die Kapitel 1.1.4 und 5.4.1.

---

<sup>1</sup> <https://flywaydb.org/>

<sup>2</sup> <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

#### 5.6.4 Fazit zur Retter-App

Bei der Retter-App hatten wir Mühe ein passendes Plugin für https-Verbindungen zu finden. Auf unserem HSR-Testserver war kein gültiges SSL-Zertifikat installiert und dadurch erwies sich das Testing der App als relativ mühsam. Die eingesetzten Plugins haben alle aus Sicherheitsgründen eine Verbindung zum Server verweigert und es gab keine Möglichkeit das nicht vertrauenswürdige Zertifikat zu ignorieren. Die Verbindung zum Produktiv-System hat jeweils wunschgemäss geklappt, weil dort ein gültiges SSL-Zertifikat vorhanden ist. Es blieb uns nichts anderes übrig, als ein selbst signiertes Zertifikat zu erstellen und dieses im Emulator und auf unseren Smartphones zu installieren. Durch diese Massnahme hat anschliessend alles wie gewünscht funktioniert. Ohne diese Aktion hätten wir immer 2 Wochen auf das Deployment des neusten Release auf dem Produktiv-System warten müssen, um neue Features zu testen.

Die grösste Herausforderung war das Einbinden der Swisstopo Karte in die Retter-App. Die API für das Kartematerial benötigt einen lokalen Webserver, um die Karte darzustellen und Layer zu verwalten. Da es keine NativeScript Plugins gibt, um lokal einen Webserver zu starten, mussten wir den Umweg über einen Webserver machen. Dabei wurde der gleiche Webserver wie für die Webapplikation benutzt. Die ganze Verwaltung der in der Retter-App angezeigten Karte geschieht nun auf dem Webserver und die JSP-Seite mit dem Kartematerial – geschützt durch JSON Web Token (JWT) – wurde in der App in eine angepasste WebView eingebunden. Dadurch kommt leider eine zusätzliche Round Trip Time (RTT) dazu aber dafür ist die App akkuschonender geworden. Ein lokaler Webserver in der App hätte sicher einige Ressourcen in Anspruch genommen mit entsprechendem Einfluss auf die Akkulaufzeit.

NativeScript ist auf jeden Fall eine spannende und noch relativ neue Technologie mit viel Potenzial. Wir sind gespannt, ob sich NativeScript neben anderen grossen Namen wie Xamarin in der Cross-Plattform Entwicklung durchsetzen wird. Mit zunehmender Bekanntheit und einer wachsenden Community ist dies jedoch durchaus möglich.

## **6 AUSBLICK**

### **6.1 Sicherheit**

Der Server für die Webapplikation ist mit einem gültigen SSL-Zertifikat ausgestattet. Die gesamte Kommunikation zwischen Retter-App und Webapplikation findet somit verschlüsselt statt. Damit keine sensiblen Daten wie Passwörter serialisiert werden, wurden in den Klassen entsprechende Annotationen gesetzt. Standortdaten sind sensible Daten und müssen besonders gut geschützt werden. Die Gefahr von Hackerangriffen kann nie ganz ausgeschlossen werden und daher wäre ein Server-Hardening sinnvoll. Leider haben wir keinen direkten Zugriff auf den Webserver – dieser wird extern bei der aspectra AG gehostet – und deshalb hat man hier nur beschränkte Möglichkeiten.

Betreffend Ausfallsicherheit ist man bereits bestens abgesichert. Ein Loadbalancer verteilt die Anfragen auf zwei unabhängige Webserver. Sollte ein Webserver ausfallen, kann der noch aktive Webserver alle Anfragen bearbeiten. Auch von der Datenbank werden ständig Backups gemacht. Im Normalfall können die Daten bis auf Transaktionsebene wiederhergestellt werden. Im Worst-Case gehen höchstens die letzten 4 Stunden verloren.

### **6.2 Aktualisieren von Einsätzen und Suchaufträgen**

Derzeit können Einsätze und Suchaufträge nicht aktualisiert werden. Wird beispielsweise der Einsatzleiter von den Rettern über aktuelle Geschehnisse aus dem Suchgebiet informiert und entscheidet sich dadurch für eine Anpassung des Suchauftrages, so muss er den offenen Suchauftrag zuerst löschen oder abschliessen. Danach muss er einen neuen Suchauftrag erstellen und kann die erhaltenen Informationen in die Planung miteinbeziehen. In so einer Situation wäre es hilfreich, wenn der Suchauftrag editierbar wäre. Zusätzlicher Aufwand wie das Zuweisen von Rettern und Einzeichnen des neuen Suchgebiets würde wegfallen.

### **6.3 Registrierung**

Aktuell kann nur ein Administrator die Logins für andere Benutzer freischalten, indem er bei ihnen im Profil ein Passwort setzt. Es wäre wünschenswert, wenn die Benutzer sich mit einer E-Mail-Adresse und Passwort registrieren könnten – wie man es von anderen Diensten kennt – und der Admin dadurch weniger Aufwand beim Einrichten der Logins hat. Zudem müsste der Administrator dem Benutzer das Passwort nicht auf eventuell unsicheren Kanälen übertragen.

Sobald die Registration abgeschlossen ist, können sich die Benutzer an der Webapplikation anmelden. Dort haben sie Zugriff auf ihre eigenen Standortdaten und können diese selbstständig verwalten. Durch eine Exportfunktion könnten die Daten in andere Dienste integriert werden und würden so auch Auswertungen und Statistiken z.B. über absolvierte Wanderungen erlauben.



## 6.4 Anonyme Tracker

Eine andere Idee wäre, dass man an den Tal- und Bergstationen von Bergbahnen anonyme Tracker mitnehmen kann. Dabei hat jeder Tracker eine Identifikationsnummer und sendet in einem bestimmten Zeitintervall die Standortdaten an die Webapplikation. Bei Mitnahme des Trackers meldet der Wanderer wie lange er ungefähr unterwegs ist. Dies wird im System vermerkt und wenn der Tracker bis Ablauf der geplanten Rückkehrzeit nicht retourniert wurde, dann wird ein Alarm ausgelöst. Dank der Aufzeichnung des Tracks kann dadurch sehr schnell und gezielt nach der Person gesucht werden. Möchte der Wanderer die Frist verlängern, so muss er telefonisch Kontakt mit der Tal- oder Bergstation aufnehmen und dann wird die neue geplante Rückkehrzeit im System vermerkt.

## 6.5 Drohneneinsätze

Drohnen nehmen in unserer Gesellschaft einen immer wichtigeren Stellenwert ein. Sei dies ein Hobbypilot, welcher schöne Naturaufnahmen machen möchte oder Amazon, welche Pakete mittels Drohnen ausliefern, um nur zwei Beispiele zu nennen. Die Technologien werden immer ausgereifter und dies führt zu längeren Akkulaufzeiten, mehr Reichweite, besseren Sensoren und diversen anderen Verbesserungen. Eine Idee wäre, dass Drohnen autonom ein für sie definiertes Suchgebiet überfliegen und filmen. Vor allem wenn Suchaktionen für Menschen zu gefährlich sind, sei dies wegen Lawinengefahr oder Steinschlag, kann ein Drohneneinsatz wertvolle Informationen liefern. Ist die Drohne mit zusätzlichem Equipment wie RECCO oder LVS ausgestattet, könnte diese bei Empfang eines Signals das Suchgebiet selbständig einkreuzen und eine vermisste Person genaustens lokalisieren.

## ABBILDUNGSVERZEICHNIS

Abbildung 1.1: trs-cockpit Login .....	1
Abbildung 1.2: trs-cockpit Übersicht .....	2
Abbildung 1.3: trs-cockpit Benutzerprofil .....	2
Abbildung 1.4: trs-cockpit Standortdaten .....	2
Abbildung 1.5: Übersicht.....	3
Abbildung 1.6: Einstellungen.....	3
Abbildung 1.7: Zeitintervall.....	3
Abbildung 1.8: Domainmodell.....	4
Abbildung 1.9: Use Case Diagramm.....	5
Abbildung 2.1: Projektplan .....	11
Abbildung 3.1: Datenbankmodell .....	14
Abbildung 3.2: NativeScript Funktionsweise.....	15
Abbildung 4.1: Login Screen .....	17
Abbildung 4.2: Startseite .....	17
Abbildung 4.3: Suchauftragsübersicht.....	17
Abbildung 4.4: Navigation Drawer .....	17
Abbildung 4.5: Reversuche .....	17
Abbildung 4.6: Wegsuche .....	17
Abbildung 5.1: Deployment Diagramm .....	22
Abbildung 5.2: trs-cockpit Login.....	23
Abbildung 5.3: trs-cockpit Übersicht.....	23
Abbildung 5.4: trs-cockpit Benutzerprofil .....	24
Abbildung 5.5: trs-cockpit Standortdaten.....	24
Abbildung 5.6: trs-cockpit Einsatz definieren.....	25
Abbildung 5.7: trs-cockpit Suchauftragsübersicht mit Einsatzpunkt.....	25
Abbildung 5.8: trs-cockpit Suchauftragsübersicht mit Track der vermissten Person .....	26
Abbildung 5.9: trs-cockpit Suchauftrag definieren.....	26
Abbildung 5.10: trs-cockpit Suchauftrag-Details.....	27
Abbildung 5.11: trs-cockpit Suchauftrag-Details anzeigen .....	27
Abbildung 5.12: trs-cockpit Bestätigung von Aktionen .....	28
Abbildung 5.13: Login Screen .....	29
Abbildung 5.14: Standort.....	29
Abbildung 5.15: Suchaufträge.....	29
Abbildung 5.16: Navigation Drawer .....	29
Abbildung 5.17: Reversuche .....	29
Abbildung 5.18: Wegsuche .....	29
Abbildung 5.19: Standort.....	30
Abbildung 5.20: Details.....	30
Abbildung 5.21: Informationen.....	30
Abbildung 5.22: E-Mail Identifizierung .....	31
Abbildung 5.23: Übersicht.....	31
Abbildung 5.24: Einstellungen.....	31
Abbildung 5.25: Zeitintervall.....	31
Abbildung 5.26: Projektverlauf .....	33
Abbildung A.1: Gantt-Diagramm .....	55

## TABELLENVERZEICHNIS

Tabelle 1.1: trs-tracker: Ausgangslage .....	3
Tabelle 1.2: Kurzbeschreibung der Konzepte .....	4
Tabelle 1.3: Aktoren & Stakeholder .....	6
Tabelle 1.4: Funktionale Anforderungen.....	8
Tabelle 1.5: Nicht-funktionale Anforderungen .....	8
Tabelle 1.6: Systemtests trs-cockpit .....	10
Tabelle 1.7: Systemtests trs-rescuer .....	10
Tabelle 1.8: Systemtests trs-tracker .....	10
Tabelle 2.1: Projektphasen .....	11
Tabelle 2.2: Meilensteine .....	11
Tabelle 2.3: Risiko Code Datenverlust.....	12
Tabelle 2.4: Risiko Dokumentenverlust.....	12
Tabelle 2.5: Risiko Ausfall Redmine .....	12
Tabelle 2.6: Risiko Datenverlust Redmine .....	12
Tabelle 2.7: Risiko Technologie Einarbeitung .....	12
Tabelle 2.8: Risiko Ungenügende Kommunikation.....	13
Tabelle 2.9: Risiko Unterschätzte Komplexität.....	13
Tabelle 2.10: Risiko Ausfall Entwickler-PC.....	13
Tabelle 4.1: UI Mockups Retter-App.....	17
Tabelle 5.1: Code-Metriken Webapplikation.....	28
Tabelle 5.2: trs-rescuer Screenshots.....	30
Tabelle 5.3: Code-Metriken Retter-App .....	30
Tabelle 5.4: trs-tracker Screenshots .....	31
Tabelle 5.5: Code-Metriken Tracking-App.....	32
Tabelle A.1: UC01: Person tracken.....	41
Tabelle A.2: UC02: Track anzeigen .....	41
Tabelle A.3: UC03: Person als Retter definieren .....	42
Tabelle A.4: UC04: Retter anzeigen .....	42
Tabelle A.5: UC05: Einsatz definieren .....	43
Tabelle A.6: UC06: Suchauftrag definieren .....	43
Tabelle A.7: UC07: Suchauftrag abholen.....	44
Tabelle A.8: UC08: Suchauftrag starten.....	44
Tabelle A.9: UC09: Suchauftrag abschliessen.....	45
Tabelle A.10: UC10: Suchauftrag und Track vergleichen .....	45
Tabelle A.11: UC11: Retter laufendem Einsatz hinzufügen.....	46
Tabelle A.12: UC12: Retter laufendem Suchauftrag hinzufügen .....	46
Tabelle A.13: UC13: Suchauftrag abbrechen.....	47
Tabelle A.14: UC14: Einsatz abschliessen .....	47
Tabelle A.15: UC15: Person registrieren .....	48
Tabelle A.16: UC16: Retter authentifizieren .....	48
Tabelle A.17: ST1: Authentifizieren .....	49
Tabelle A.18: ST2: Track anzeigen.....	49
Tabelle A.19: ST3: Profil bearbeiten .....	49
Tabelle A.20: ST4: GPS Daten löschen.....	49
Tabelle A.21: ST5: Einsatz erstellen .....	50

Tabelle A.22: ST6: Suchauftrag erstellen .....	50
Tabelle A.23: ST7: Suchauftrag-Details anzeigen .....	50
Tabelle A.24: ST8: Starten der App .....	51
Tabelle A.25: ST9: Authentifizieren .....	51
Tabelle A.26: ST10: Suchaufträge anzeigen .....	51
Tabelle A.27: ST11: Suchauftrag durchführen .....	52
Tabelle A.28: ST12: Erster Start .....	53
Tabelle A.29: ST13: Intervall konfigurieren .....	53
Tabelle A.30: Risiko-Analyse in der Elaboration-Phase .....	56
Tabelle A.31: Risiko-Analyse in der Construction-Phase .....	56

## A ANHANG

### A.1 Use Case Detail

#### A.1.1 UC01: Person tracken

Description	Die GPS Daten einer Person werden in einem zeitlichen Intervall an den Server gesendet.
Primary Actor	Benutzer
Trigger	Ein Benutzer möchte sich von der App tracken lassen
Stakeholder and Interests	Benutzer: Will bei einem Notfall möglichst schnell von Rettungskräften gefunden werden
Priority	1
Preconditions	GPS Signal vorhanden und mobile Daten aktiviert
Postconditions	GPS-Standort wurde auf den Server übermittelt
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Benutzer installiert die App</li> <li>2. Die App läuft immer im Hintergrund und sendet GPS Daten im konfigurierten Zeitintervall</li> </ol>
Extensions	2a) Kann die App die Daten nicht zum Server senden oder sind keine GPS Daten vorhanden, so werden die Informationen zwischengespeichert und zu einem späteren Zeitpunkt gesendet.
Frequency of Occurrence	Je nach konfiguriertem Intervall in der App

Tabelle A.1: UC01: Person tracken

#### A.1.2 UC02: Track anzeigen

Description	Der Einsatzleiter selektiert eine Person und sieht deren Spur in einem bestimmten Zeitintervall.
Primary Actor	Einsatzleiter
Trigger	Der Einsatzleiter möchte die aufgezeichneten Standortdaten einer verunfallten/vermissten Person anzeigen
Stakeholder and Interests	Einsatzleiter: Will Einsatz planen mit Hilfe der verfügbaren Standortdaten der vermissten Person und verschiedenem Kartenmaterial (Swisstopo, OSM, Google Maps)
Priority	1
Preconditions	Benutzer hat Tracking-App installiert und es sind Standortdaten vorhanden
Postconditions	Standortverlauf der Person wird auf der Karte angezeigt
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Einsatzleiter selektiert Person</li> <li>2. WebApp zeigt Standortverlauf der Person auf der ausgewählten Karte an.</li> </ol>
Extensions	2a) Das Zeitintervall des anzuzeigenden Tracks kann angepasst werden

Tabelle A.2: UC02: Track anzeigen

### A.1.3 UC03: Person als Retter definieren

Description	Eine getrackte Person kann als ein Retter definiert werden.
Primary Actor	Einsatzleiter
Trigger	Auf der Karte sollen «normal» getrackte Personen und Retter unterschieden werden können und später einem Einsatz zugeordnet werden können.
Stakeholder and Interests	Einsatzleiter: Muss Benutzer und Retter unterscheiden können.
Priority	1
Preconditions	Der Retter lässt sich mit der Tracking App tracken
Postconditions	Der Retter ist als Retter im System definiert
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Der Einsatzleiter editiert eine Person</li> <li>2. Definiert in den Eigenschaften die Person als Retter</li> </ol>
Extensions	2a) Weist dem Retter einen Benutzernamen und Passwort zu.

Tabelle A.3: UC03: Person als Retter definieren

### A.1.4 UC04: Retter anzeigen

Description	Der Einsatzleiter sieht alle Retter auf der Karte. Diese werden als «Frei», «in einem Einsatz» oder «in einem Suchauftrag» angezeigt.
Primary Actor	Einsatzleiter
Trigger	Der Einsatzleiter möchte sehen, welche Rettungskräfte sich in der Nähe einer verunfallten Person befinden und aktuell verfügbar sind.
Stakeholder and Interests	Einsatzleiter: Will Einsatz der Rettungskräfte planen
Priority	1
Preconditions	<ul style="list-style-type: none"> <li>▪ Retter sind in der WebApp als Rettungskraft identifiziert</li> <li>▪ Tracking App ist bei allen Rettern installiert</li> </ul>
Postconditions	Die Standorte der einzelnen Rettungskräfte werden auf der Karte angezeigt
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Einsatzleiter öffnet Web-App</li> <li>2. Web-App zeigt auf der Karte den Standort aller Retter an</li> </ol>
Extensions	2a) Es soll nach den 3 Status «Frei», «in einem Einsatz» oder «in einem Suchauftrag» gefiltert werden können.

Tabelle A.4: UC04: Retter anzeigen

### A.1.5 UC05: Einsatz definieren

<b>Description</b>	Bei einer Vermisstmeldung wird ein neuer Einsatz mit Name, Einsatznummer, Startzeitpunkt und den beteiligten Rettern definiert.
<b>Primary Actor</b>	Einsatzleiter
<b>Trigger</b>	Alle Suchaufträge zu einer Person laufen unter demselben Einsatz. (Es können mehrere Einsätze unabhängig voneinander an verschiedenen Orten laufen.)
<b>Stakeholder and Interests</b>	Einsatzleiter: Koordiniert seinen Einsatz
<b>Priority</b>	1
<b>Preconditions</b>	Vermisstmeldung mit Standort und/oder Personendaten eingegangen
<b>Postconditions</b>	Es wurde ein Einsatz im System erstellt
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Einsatzleiter öffnet Web-App</li> <li>2. Einsatzleiter wählt gesuchte Person aus falls vorhanden oder gibt Koordinaten ein</li> <li>3. Web-App zeigt den Standort und falls vorhanden den Track der gesuchten Person an</li> <li>4. Einsatzleiter erstellt einen neuen Einsatz <ol style="list-style-type: none"> <li>a. Namen und Einsatznummer definieren</li> <li>b. Startzeitpunkt definieren</li> <li>c. Alle beteiligten Retter dem Einsatz zuteilen</li> </ol> </li> </ol>

Tabelle A.5: UC05: Einsatz definieren

### A.1.6 UC06: Suchauftrag definieren

<b>Description</b>	Der Einsatzleiter definiert in einem Einsatz eine Revier- oder Wegsuche mit einem Namen (R1, W1) sowie einem Rufnamen (Suchtrupp W1) und weist diese einem oder mehreren Rettern zu.
<b>Primary Actor</b>	Einsatzleiter
<b>Trigger</b>	Der Einsatzleiter möchte mit Hilfe der Rettungskräfte nach der verunfallten/vermissten Person suchen
<b>Stakeholder and Interests</b>	Einsatzleiter: Will den Rettungskräften ein Zielgebiet für die Suchaktion vorgeben
<b>Priority</b>	1
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>▪ Retter hat Tracking-App installiert</li> <li>▪ Einsatz wurde erstellt</li> </ul>
<b>Postconditions</b>	Suchauftrag steht für die Retter zum «abholen» bereit.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Einsatzleiter wählt einen Einsatz aus</li> <li>2. Einsatzleiter definiert mit dem Auswahl-Werkzeug eine Route oder ein Revier auf der Karte</li> <li>3. Einsatzleiter weist den Suchauftrag einem oder mehreren Rettern, welche dem Einsatz zugeteilt sind, zu</li> </ol>
<b>Extensions</b>	3.a) Aus allen Rettern auswählen und automatisch dem Einsatz hinzufügen

Tabelle A.6: UC06: Suchauftrag definieren

### A.1.7 UC07: Suchauftrag abholen

<b>Description</b>	Der Retter kann seinen Auftrag abholen und sieht nun in seiner App die für ihn definierte Route/Revier sowie die Spur der vermissten Person.
<b>Primary Actor</b>	Retter
<b>Trigger</b>	Der Retter möchte mit Hilfe der definierten Route nach der Person suchen
<b>Stakeholder and Interests</b>	Retter: Will genaue Standortdaten für eine effiziente Suche
<b>Priority</b>	1
<b>Preconditions</b>	Dem Retter wurde ein Suchauftrag zugewiesen
<b>Postconditions</b>	Die vom Einsatzleiter definierte Route/Revier inkl. Spur der vermissten Person wird auf der Karte angezeigt
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Rettungskraft öffnet Such-App</li> <li>2. App aktualisiert die Auftragsliste</li> <li>3. Alle verfügbaren Aufträge für den entsprechenden Retter werden angezeigt</li> <li>4. Rettungskraft wählt einen Auftrag aus</li> <li>5. App zeigt die Route/Revier, den eigenen Standort, sowie Spur der Person auf der Karte an</li> </ol>

Tabelle A.7: UC07: Suchauftrag abholen

### A.1.8 UC08: Suchauftrag starten

<b>Description</b>	Sobald der Retter am Startpunkt seines Suchauftrages angekommen ist, bestätigt er, dass sein Suchauftrag nun offiziell beginnt.
<b>Primary Actor</b>	Retter
<b>Trigger</b>	Der Retter lädt den Auftrag womöglich bereits in der Basis. Somit gehört der Weg bis zu seiner Route noch nicht zur eigentlichen Suche.
<b>Stakeholder and Interests</b>	Einsatzleiter: Übersichtshalber will er später den Track der Retter nur innerhalb der Dauer des entsprechenden Suchauftrages analysieren.
<b>Priority</b>	1
<b>Preconditions</b>	Suchauftrag erstellt und vom Retter abgeholt
<b>Postconditions</b>	Der Start Zeitstempel für den Suchauftrag wurde gesetzt.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Sobald die Retter am Auftragsort angekommen sind, startet einer der Gruppe den Auftrag</li> </ol>

Tabelle A.8: UC08: Suchauftrag starten



### A.1.9 UC09: Suchauftrag abschliessen

<b>Description</b>	Sobald die Retter mit dem Suchauftrag durch sind, schliessen sie den Suchauftrag mit einer Bemerkung ab.
<b>Primary Actor</b>	Retter
<b>Trigger</b>	Sind die Retter mit einem Suchauftrag fertig, so sind diese für einen neuen Suchauftrag bereit.
<b>Stakeholder and Interests</b>	Einsatzleiter: Hat Überblick über laufende Suchaufträge
<b>Priority</b>	2
<b>Preconditions</b>	Suchauftrag wurde gestartet
<b>Postconditions</b>	Der Suchauftrag ist als geschlossen definiert
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Sobald das Trupp durch ist oder die Suche zu gefährlich wird, meldet es den Auftrag als fertig</li> <li>2. Dabei wird eine Bemerkung hinterlegt</li> </ol>

*Tabelle A.9: UC09: Suchauftrag abschliessen*

### A.1.10 UC10: Suchauftrag und Track vergleichen

<b>Description</b>	Der Einsatzleiter sieht die definierte Route/Revier sowie den aktuellen Track der Retter.
<b>Primary Actor</b>	Einsatzleiter
<b>Trigger</b>	Der Einsatzleiter möchte die Abweichungen bezüglich definiertem Suchauftrag und effektiver Suchroute der Rettungskräfte sehen
<b>Stakeholder and Interests</b>	Einsatzleiter: Will nachvollziehen können, wo die Retter gesucht haben.
<b>Priority</b>	2
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>▪ Suchauftrag wurde definiert</li> <li>▪ Suchauftrag wurde abgeholt und gestartet</li> </ul>
<b>Postconditions</b>	Suchauftrag sowie realer Track der Retter werden auf der Karte dargestellt
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Einsatzleiter wählt Einsatz aus</li> <li>2. Einsatzleiter wählt Suchauftrag aus</li> <li>3. Web-App zeigt Route/Revier, sowie alle Tracks der beteiligten Retter des Auftrages seit Beginn des Suchauftrages an</li> </ol>

*Tabelle A.10: UC10: Suchauftrag und Track vergleichen*

### A.1.11 UC11: Retter laufendem Einsatz hinzufügen

Description	Einem Einsatz können auch zu einem späteren Zeitpunkt noch weitere Retter hinzugefügt werden.
Primary Actor	Einsatzleiter
Trigger	Es sind nicht unbedingt alle Retter von Anfang an dabei und somit kann nicht von Anfang genau gesagt werden, welche Retter alle an dem Einsatz teilnehmen.
Stakeholder and Interests	Einsatzleiter: Kann während des Einsatzes weitere Retter dem Einsatz hinzufügen.
Priority	3
Preconditions	Einsatz wurde definiert
Postconditions	Der Retter wurde dem Einsatz hinzugefügt
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Einsatzleiter wählt Einsatz aus</li> <li>2. Einsatzleiter fügt Retter dem Einsatz hinzu</li> </ol>

*Tabelle A.11: UC11: Retter laufendem Einsatz hinzufügen*

### A.1.12 UC12: Retter laufendem Suchauftrag hinzufügen

Description	Einem Suchauftrag können auch zu einem späteren Zeitpunkt noch weitere Retter hinzugefügt werden.
Primary Actor	Einsatzleiter
Trigger	Gestaltet sich die Suche in einem gewissen Gebiet als schwierig können diesem Suchauftrag weitere Retter hinzugefügt werden
Stakeholder and Interests	Einsatzleiter: Kann während des Einsatzes weitere Retter dem Suchauftrag hinzufügen.
Priority	3
Preconditions	Suchauftrag wurde definiert
Postconditions	<ul style="list-style-type: none"> <li>▪ Retter wurde dem Suchauftrag hinzugefügt</li> <li>▪ Retter kann seinen Suchauftrag abholen</li> </ul>
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Einsatzleiter wählt Suchauftrag aus</li> <li>2. Einsatzleiter fügt Retter dem Suchauftrag hinzu</li> </ol>

*Tabelle A.12: UC12: Retter laufendem Suchauftrag hinzufügen*

#### A.1.13 UC13: Suchauftrag abbrechen

Description	Der Einsatzleiter kann einen Suchauftrag aus diversen Gründen abbrechen.
Primary Actor	Einsatzleiter
Trigger	Merkt der Einsatzleiter, dass ein Suchauftrag in die falsche Richtung geht, kann er diesen abbrechen und einen neuen Suchauftrag definieren.
Stakeholder and Interests	Einsatzleiter: Kann seine Retter einem effizienteren Suchauftrag zuweisen
Priority	2
Preconditions	Suchauftrag wurde gestartet
Postconditions	Der Suchauftrag ist als geschlossen definiert
Main Success Scenario	<ol style="list-style-type: none"><li>1. Einsatzleiter wählt einen Suchauftrag aus</li><li>2. Einsatzleiter bricht diesen ab</li><li>3. Einsatzleiter hinterlegt eine Bemerkung</li></ol>

*Tabelle A.13: UC13: Suchauftrag abbrechen*

#### A.1.14 UC14: Einsatz abschliessen

Description	Ist ein Einsatz zu Ende, wird dieser als abgeschlossen definiert.
Primary Actor	Einsatzleiter
Trigger	Wurde die vermisste Person gefunden oder die Suche wurde eingestellt, so wird der Einsatz geschlossen und abgelegt.
Stakeholder and Interests	Einsatzleiter: Hat Überblick über laufende Einsätze
Priority	2
Preconditions	Einsatz wurde definiert
Postconditions	Der Einsatz ist als geschlossen definiert
Main Success Scenario	<ol style="list-style-type: none"><li>1. Einsatzleiter wählt einen Einsatz aus</li><li>2. Einsatzleiter schliesst diesen ab</li><li>3. Einsatzleiter hinterlegt eine Bemerkung</li></ol>

*Tabelle A.14: UC14: Einsatz abschliessen*

### A.1.15 UC15: Person registrieren

Description	Eine Person installiert die Tracking-App und muss sich mit seiner Mailadresse registrieren.
Primary Actor	Benutzer
Trigger	Wird ein Benutzer vermisst, so kann er mindestens durch die Mailadresse im Backend identifiziert werden
Stakeholder and Interests	Benutzer: Möchte schnellstmöglich identifiziert werden
Priority	3
Preconditions	Benutzer hat die Tracking-App installiert
Postconditions	Benutzer ist registriert
Main Success Scenario	<ol style="list-style-type: none"><li>1. Benutzer öffnet App</li><li>2. Benutzer registriert sich mit E-Mail Adresse</li><li>3. Benutzer wird automatisch angemeldet und getrackt</li></ol>

Tabelle A.15: UC15: Person registrieren

### A.1.16 UC16: Retter authentifizieren

Description	Ein Retter besitzt ein Login (Username + Passwort) und loggt sich damit in der Retter-App ein.
Primary Actor	Retter
Trigger	Wird eine Suchaktion gestartet, so kann der Retter an dieser teilnehmen
Stakeholder and Interests	Retter: Möchte bei der Suchaktion helfen
Priority	3
Preconditions	<ul style="list-style-type: none"><li>▪ Retter hat die Retter-App installiert</li><li>▪ Ist als Retter definiert und besitzt Login-Daten</li></ul>
Postconditions	Retter ist eingeloggt und über laufende Suchaktionen informiert
Main Success Scenario	<ol style="list-style-type: none"><li>1. Retter meldet sich mit Zugangsdaten an</li><li>2. Retter wird authentifiziert</li><li>3. Retter ist angemeldet</li></ol>

Tabelle A.16: UC16: Retter authentifizieren

## A.2 Systemtest trs-cockpit

### A.2.1 ST1: Authentifizieren

Testziel	Der Benutzer kann sich mit dem Login an der WebApp authentifizieren
Vorbedingungen	<ul style="list-style-type: none"><li>Der Benutzer verfügt über gültige Zugangsdaten, die von einem Einsatzleiter erfasst wurden</li></ul>
Durchführung	<ol style="list-style-type: none"><li>Öffnen der Webapplikation</li><li>E-Mail-Adresse und Passwort eingeben</li></ol>
Erwartetes Ergebnis	Benutzer ist authentifiziert und wird auf Startseite weitergeleitet

Tabelle A.17: ST1: Authentifizieren

### A.2.2 ST2: Track anzeigen

Testziel	Der Benutzer kann seinen eigenen Track auf der Karte anzeigen
Vorbedingungen	<ul style="list-style-type: none"><li>Der Benutzer ist authentifiziert</li><li>Der Benutzer hat die Tracking-App aktiviert und bereits Standortdaten an die API übermittelt</li></ul>
Durchführung	<ol style="list-style-type: none"><li>Der Benutzer klickt auf den Button mit dem Pfeil-Symbol neben seinem Eintrag in der Tabelle</li></ol>
Erwartetes Ergebnis	Der Track des Benutzers wird auf der Karte dargestellt

Tabelle A.18: ST2: Track anzeigen

### A.2.3 ST3: Profil bearbeiten

Testziel	Der Benutzer kann sein Profil bearbeiten
Vorbedingungen	<ul style="list-style-type: none"><li>Der Benutzer ist authentifiziert</li></ul>
Durchführung	<ol style="list-style-type: none"><li>Der Benutzer klickt auf seinen Benutzernamen</li><li>Der Benutzer wählt den Navigationspunkt «Profil bearbeiten» aus</li></ol>
Erwartetes Ergebnis	Das Profil des Benutzers wird angezeigt und er kann an seinem Profil Änderungen vornehmen

Tabelle A.19: ST3: Profil bearbeiten

### A.2.4 ST4: GPS-Daten löschen

Testziel	Der Benutzer kann einzelne oder alle eigenen GPS-Daten löschen
Vorbedingungen	<ul style="list-style-type: none"><li>Der Benutzer ist authentifiziert</li></ul>
Durchführung	<ol style="list-style-type: none"><li>Der Benutzer klickt auf seinen Benutzernamen</li><li>Der Benutzer wählt den Navigationspunkt «GPS-Daten» aus</li><li>Der Benutzer klickt auf den Button "alle Daten löschen" neben seinem Eintrag in der Liste oder den Button mit dem Papierkorb neben einem Eintrag in der Liste mit allen GPS-Positionen</li></ol>
Erwartetes Ergebnis	Der Benutzer hat alle seine GPS-Daten gelöscht oder nur eine einzelne GPS-Position

Tabelle A.20: ST4: GPS Daten löschen

### A.2.5 ST5: Einsatz erstellen

Testziel	Der Einsatzleiter kann einen Einsatz erstellen.
Vorbedingungen	<ul style="list-style-type: none"><li>▪ Der Benutzer ist mit Einsatzleiterberechtigungen authentifiziert</li></ul>
Durchführung	<ol style="list-style-type: none"><li>1. Der Benutzer wählt den Menüpunkt «Einsätze» aus</li><li>2. Der Benutzer klappt den Reiter «Neuer Einsatz» auf</li><li>3. Der Benutzer füllt alle Felder mit den vorhandenen Informationen aus</li><li>4. Der Benutzer klickt «Einsatz erstellen»</li></ol>
Erwartetes Ergebnis	Beim Erstellen sieht der Benutzer alle Retter auf der Karte. Wählt er die vermisste Person aus, sieht er auch deren Track im definierten Zeitintervall. Definiert er einen Einsatzpunkt, so erscheint dieser auch als Marker auf der Karte.

Tabelle A.21: ST5: Einsatz erstellen

### A.2.6 ST6: Suchauftrag erstellen

Testziel	Der Einsatzleiter kann einen Suchauftrag erstellen.
Vorbedingungen	<ul style="list-style-type: none"><li>▪ Der Benutzer ist mit Einsatzleiterberechtigungen authentifiziert</li><li>▪ Es wurde bereits ein Einsatz erstellt.</li></ul>
Durchführung	<ol style="list-style-type: none"><li>1. Der Benutzer wählt den Menüpunkt «Einsätze» aus</li><li>2. Der Benutzer wählt einen Einsatz aus</li><li>3. Der Benutzer klappt den Reiter «Neuer Suchauftrag» auf</li><li>4. Der Benutzer füllt die Felder aus und zeichnet ein Revier oder einen Weg auf der Karte ein</li><li>5. Der Benutzer klickt «Suchauftrag erstellen»</li></ol>
Erwartetes Ergebnis	Beim Erstellen sieht der Benutzer alle dem Einsatz zugeordneten Retter, sowie bereits erstellte Suchaufträge auf der Karte.

Tabelle A.22: ST6: Suchauftrag erstellen

### A.2.7 ST7: Suchauftrag-Details anzeigen

Testziel	Der Einsatzleiter kann einen Suchauftrag mitverfolgen
Vorbedingungen	<ul style="list-style-type: none"><li>▪ Es wurde ein Suchauftrag erstellt</li><li>▪ Dem Suchauftrag sind Retter zugewiesen</li><li>▪ Die Retter haben den Suchauftrag in der Retter-App gestartet</li></ul>
Durchführung	<ol style="list-style-type: none"><li>1. Der Benutzer wählt den Navigationspunkt «Suchaufträge» aus</li><li>2. Der Benutzer klickt auf den Namen eines Suchauftrages oder in den Optionen auf den Button mit dem Auge-Symbol bei den offenen Suchaufträgen</li></ol>
Erwartetes Ergebnis	Der Benutzer sieht den Track der Retter auf der Karte, sowie den Track und / oder den Einsatzpunkt.

Tabelle A.23: ST7: Suchauftrag-Details anzeigen

## A.3 Systemtest trs-rescuer

### A.3.1 ST8: Starten der App

Testziel	Der Applikationsstart wird wie folgt ausgeführt.
Vorbedingungen	-
Durchführung	1. Starten der App
Erwartetes Ergebnis	Beim Start wird der Splash Screen mit dem trs Logo angezeigt.

Tabelle A.24: ST8: Starten der App

### A.3.2 ST9: Authentifizieren

Testziel	Der Retter kann sich in der App anmelden.
Vorbedingungen	<ul style="list-style-type: none"><li>▪ Der Retter besitzt ein Login.</li><li>▪ Das Gerät hat eine aktive Internetverbindung.</li></ul>
Durchführung	<ol style="list-style-type: none"><li>1. Der Retter gibt seine Mail-Adresse und sein Passwort ein</li><li>2. Der Retter wählt ob seine Login-Daten gespeichert werden sollen.</li><li>3. Der Retter drückt «Anmelden».</li></ol>
Erwartetes Ergebnis	Ist die Authentifizierung erfolgreich, wird der Retter in den internen Bereich weitergeleitet. Wurde entschieden, dass die Daten gespeichert werden sollen, müssen beim nächsten Applikationsstart keine Eingaben mehr getätigt werden.

Tabelle A.25: ST9: Authentifizieren

### A.3.3 ST10: Suchaufträge anzeigen

Testziel	Der Retter hat eine Übersicht über alle ihm zugeordneten Suchaufträge.
Vorbedingungen	<ul style="list-style-type: none"><li>▪ Der Retter ist authentifiziert</li><li>▪ Dem Retter wurde mindestens ein Suchauftrag zugeteilt</li></ul>
Durchführung	<ol style="list-style-type: none"><li>1. Der Retter wählt in der Navigation den Punkt «Suchaufträge» aus</li><li>2. Der Retter wählt einen Suchauftrag aus.</li></ol>
Erwartetes Ergebnis	Es erscheint eine Liste mit allen Suchaufträgen, unterteilt ob diese offen, laufend oder abgeschlossen sind. Klickt er auf einen Suchauftrag, erscheint eine Meldung mit weiteren Informationen zum Suchauftrag.

Tabelle A.26: ST10: Suchaufträge anzeigen

### A.3.4 ST11: Suchauftrag durchführen

Testziel	Der Retter kann einen Suchauftrag öffnen und durchführen, d.h. starten, abschliessen und allenfalls erneut öffnen.
Vorbedingungen	▪ ST3 erfolgreich durchgeführt.
Durchführung	<ol style="list-style-type: none"><li>1. Der Retter wählt im Popup «Anzeigen»</li><li>2. Der Retter startet den Suchauftrag mit Klick auf «Starten»</li><li>3. Der Retter beendet den Suchauftrag mit Klick auf «Abschliessen» und gibt einen Kommentar dazu ab.</li><li>4. Der Retter kann den Suchauftrag mit Klick auf «Erneut öffnen» erneut öffnen, wenn der Suchauftrag doch noch nicht ganz abgeschlossen ist.</li></ol>
Erwartetes Ergebnis	Der Suchauftrag wird auf der Karte mit dem Track der vermissten Person, dem Einsatzpunkt und dem aktuellen Track des Retters angezeigt. Anschliessend können alle obigen Aktionen erfolgreich durchgeführt werden und der Status passt sich im Backend dementsprechend an.

*Tabelle A.27: ST11: Suchauftrag durchführen*



## A.4 Systemtest trs-tracker

### A.4.1 ST12: Erster Start

Testziel	Der Benutzer kann die App starten und benutzen.
Vorbedingungen	-
Durchführung	<ol style="list-style-type: none"><li>1. Starten der App</li><li>2. Eingabe einer Mail-Adresse</li><li>3. Starten des Trackings durch Klicken auf den Statusbutton «Tracking aus»</li></ol>
Erwartetes Ergebnis	Beim Start erscheint der Splash Screen mit dem trs Logo. Anschliessend erscheint ein Eingabefeld, dass nach einer Mail-Adresse fragt. Gibt man eine vom Format gültige Adresse ein, so erscheint die Log Übersicht. Wird das Tracking gestartet, wird bereits der erste Trackingpunkt erstellt.

Tabelle A.28: ST12: Erster Start

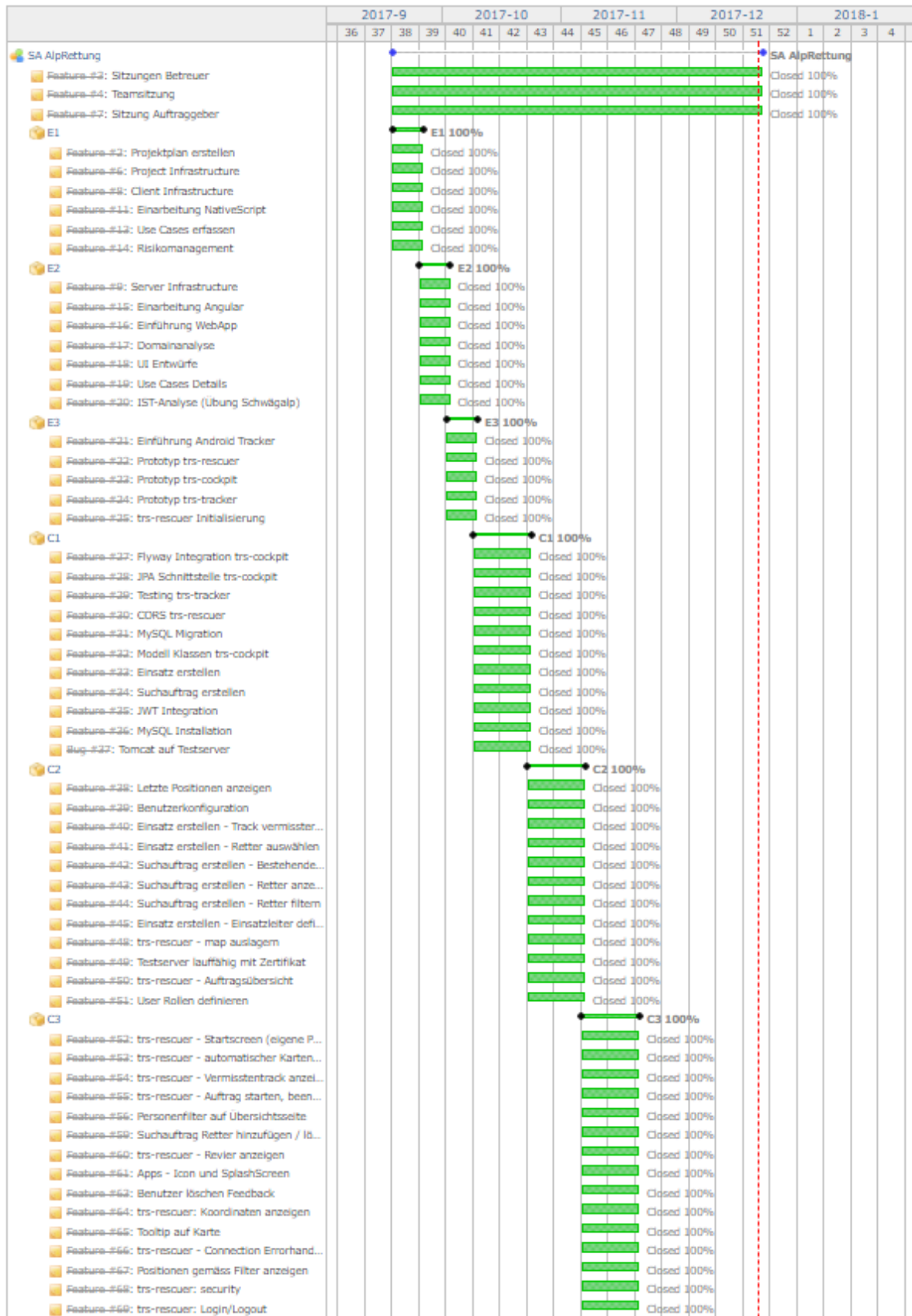
### A.4.2 ST13: Intervall konfigurieren

Testziel	Der Benutzer kann das Intervall konfigurieren.
Vorbedingungen	<ol style="list-style-type: none"><li>4. Mail Adresse hinterlegt</li></ol>
Durchführung	<ul style="list-style-type: none"><li>▪ Aufrufen der Einstellungen übers Menü</li><li>▪ Menüpunkt «Zeitintervall für Standortupdates» auswählen</li><li>▪ Beliebigen Zeitintervall einstellen und bestätigen</li></ul>
Erwartetes Ergebnis	Das Zeitintervall kann in Minuten ausgewählt werden. Läuft das Tracking momentan, wird beim Bestätigen der Service neu gestartet und das neue Intervall angewendet.

Tabelle A.29: ST13: Intervall konfigurieren

## A.5 Projektmanagement Detail

### A.5.1 Gant Diagramm



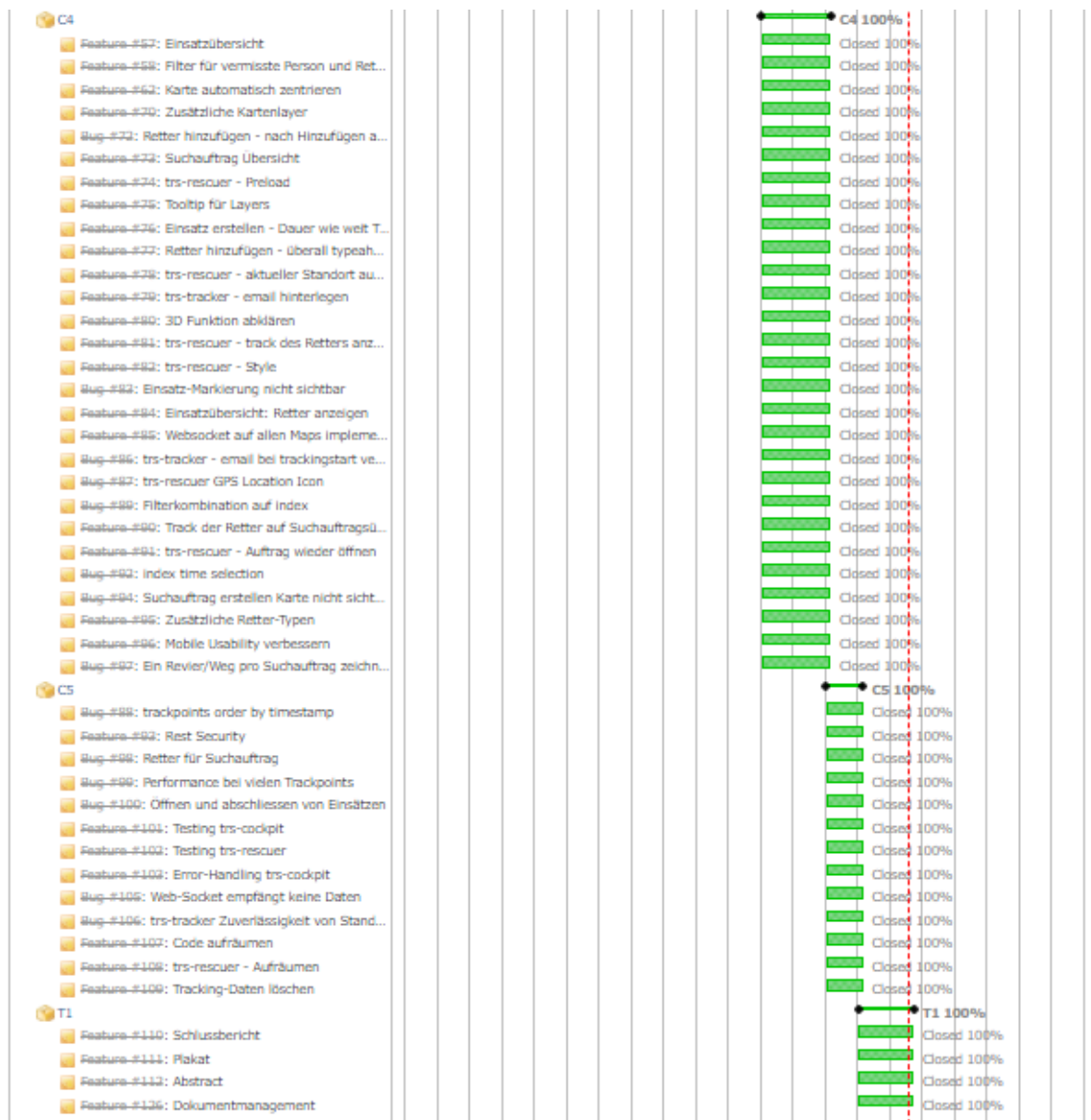


Abbildung A.1: Gantt-Diagramm

## A.6 Risikomanagement Detail

### A.6.1 Risiko-Analyse in der Elaboration-Phase

Risiko	Max. Schaden [h]	Eintrittswahrscheinlichkeit in %	Gewichteter Schaden [h]
R1	16	5	0.8
R2	20	5	1
R3	8	15	1.2
R4	4	10	0.4
R5	32	30	9.6
R6	8	10	0.8
R7	32	35	11.2
R8	16	10	1.6
<b>Total</b>	<b>136</b>		<b>26.6</b>

*Tabelle A.30: Risiko-Analyse in der Elaboration-Phase*

### A.6.2 Risiko-Analyse in der Construction-Phase

Risiko	Max. Schaden [h]	Eintrittswahrscheinlichkeit in %	Gewichteter Schaden [h]
R1	16	5	0.8
R2	10	5	0.5
R3	8	5	0.4
R4	4	5	0.2
R5	16	15	2.4
R6	8	5	0.4
R7	16	15	2.4
R8	16	10	1.6
<b>Total</b>	<b>94</b>		<b>8.7</b>

*Tabelle A.31: Risiko-Analyse in der Construction-Phase*