



Florian Haid

Mobile Managementlösung für ein umfassendes echtzeitnahes Lagebild zur optimierten Koordination von Sicherheitskräften im Rahmen von unterschiedlichen Einsatzszenarien

Bachelorarbeit

to achieve the university degree of

Bachelor of Science

Bachelor's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn. Roman Kern

Institute of Interactive Systems and Data Science

Head: Univ.-Prof.Dr. Stefanie Lindstaedt

Graz, May 2020

This document is set in Palatino, compiled with pdfL^AT_EX2e and Biber.

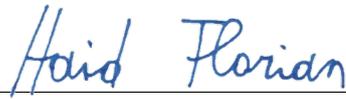
The L^AT_EX template from Karl Voit is based on KOMA script and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Graz, 15.05.2020

Datum



Unterschrift

Abstrakt

Bei Großveranstaltungen entsteht ein sehr hoher Managementaufwand um die Sicherheit für alle Besucher gewährleisten zu können. Es sind nämlich nicht nur private Sicherheitskräfte im Einsatz, sondern oft auch Polizei, Rettung, oder auch die Feuerwehr. Aus diesem Grund ist es sehr wichtig, dass alle beteiligten Organisationen effizient und ohne organisatorische Probleme zusammenarbeiten können. Bei Notfällen, kann es durch schlechten Informationsaustausch schnell zu einer kritischen Situation kommen. Um dieses Problem zu beheben, wurde eine Managementlösung entwickelt mit der es möglich ist den Informationsaustausch zu optimieren, so dass alle Beteiligten schnellen und einfachen Zugriff auf alle Informationen haben. Das System besteht aus einer Web-Anwendung für die Einsatzzentrale, so wie einer Android-Applikation für alle mobilen Einheiten. Da es mit diesem Managementsystem nun möglich ist, dass alle Organisationen, das selbe System verwenden, können die Informationen direkt an alle zuständigen Organisationen ohne Umwege gesendet werden. Durch die Verwendung einer eigenen Android-Applikation verfügen außerdem auch alle mobilen Einsatzkräfte über die notwendigen Informationen und nicht mehr nur die Einsatzzentrale. Somit können durch den optimierten Informationsaustausch zwischen allen beteiligten Organisationen, kritische Situation effizient und ohne organisatorische Problem gelöst werden. Dieses Projekt ist zwar lediglich ein Prototyp, aber es zeigt bereits sehr gut, was alles möglich ist, und wie es eingesetzt werden kann.

Inhaltsverzeichnis

Abstrakt	iv
1 Einleitung	1
2 Grundlagen	3
2.1 Smartphone	3
2.2 Android	3
2.2.1 Verbreitung	3
2.2.2 Versionen	5
2.2.3 Activity-Lebenszyklus	6
2.3 Firebase Cloud Messaging	9
3 Verwandte Arbeiten	10
4 Anwendungsbereiche	16
5 Methodik	19
5.1 Weboberfläche	20
5.1.1 Konzept	20
5.1.2 Implementierung	23
5.2 Android Applikation	29
5.2.1 Konzept	29
5.2.2 Implementierung	30
6 Evaluierung	42
6.1 User-Test	42
6.2 Änderungen durch User-Test	43
7 Schlussfolgerung	48
7.1 Zukünftige Arbeit	49

Inhaltsverzeichnis

Literatur

50

Abbildungsverzeichnis

2.1	Smartphone-Nutzung in Österreich	4
2.2	Verteilung der Smartphone-Betriebssysteme in Österreich . .	4
2.3	Android-Activity-Lebenszyklus	8
3.1	Monitoring-Software für die einfache Lokalisierung von aktiven Einheiten bei Großveranstaltungen	11
3.2	Monitoring-Software zur Visualisierung der geschätzten Personenanzahl als Heatmap	13
5.1	Übersicht aller Komponenten	20
5.2	Weboberfläche - Programmübersicht	22
5.3	Weboberfläche - Kartenansicht	24
5.4	Weboberfläche - Nachrichtenansicht	27
5.5	Weboberfläche - Neue Nachricht	28
5.6	Weboberfläche - Livestream-Ansicht	29
5.7	Android-Applikation Flussdiagramm	30
5.8	Android-Applikation Registrierung	31
5.9	Android-Applikation Freizeitmodus	31
5.10	Android Fragment Lebenszyklus	33
5.11	Android-Applikation Status-Fragment	34
5.12	Android-Applikation Map-Fragment	34
5.13	Android-Applikation Inbox	37
5.14	Android-Applikation Outbox	37
5.15	Android-Applikation Neue Nachricht	39
5.16	Android-Applikation Video-Fragment	39
6.1	Android-Applikation - Neue Benutzerverwaltung	44
6.2	Android-Applikation - Neue Chatübersicht	45
6.3	Android-Applikation - Neuer Chat	45

Abbildungsverzeichnis

6.4	Android-Applikation - Neues Flussdiagramm	46
-----	---	----

Tabellenverzeichnis

2.1	Andorid-Versionen Überblick	5
2.2	Andorid-Versionen Verbreitung	6
5.1	Weboberfläche - Verwendete Bibliotheken	25
5.2	Android-API - Unterstützte Geräte	32

1 Einleitung

Bei Großveranstaltungen kommt es leider immer wieder zu Tragödien, die aus heutiger Sicht vielleicht verhindert werden hätten können. Als es zum Beispiel 1985 zu einem Brand im englischen Fußballstadion in Bradford kam, rannten die Zuschauer auf verschlossene Ausgänge zu, wodurch 56 Menschen ums Leben kamen und weitere 200 verletzt wurden. Auch 2010 kam es zu zwei tragischen Massenpaniken in denen sehr viele Menschen starben. Ein Vorfall davon ereignete sich in Deutschland bei der Loveparade in Duisburg, wo es in einem Tunnel zu einer Massenpanik kam und 21 Menschen starben. Die andere Tragödie geschah nur ein paar Monate später in der kambodschanischen Hauptstadt Phnom Penh, wo auf einer Fußgängerbrücke zur Insel Koh Pich eine Massenpanik entstand und mehrere hundert Personen umgekommen, sowie über 700 Personen verletzt worden sind¹.

Damit solche tragischen Momente nicht wieder geschehen, war das Ziel der Arbeit die Entwicklung einer mobilen Managementlösung für ein umfassendes Lagebild zur Koordination von Sicherheitskräften. In vielen sicherheitsrelevanten Bereichen ist es wichtig die Informationen für alle verschiedenen Einheiten verfügbar und effizient darstellen zu können. Außerdem müssen sie in der Lage sein sich untereinander Nachrichten versenden zu können um einen schnellen Informationsaustausch zu gewährleisten. Alle diese Anforderungen werden beispielsweise in der Security bei Großveranstaltungen oder auch bei der Polizei, sowie beim Bundesheer benötigt.

Da das System aus mehreren Komponenten aufgebaut ist, kann es optimal für die Einsatzzentrale, sowie für alle mobilen Einheiten eingesetzt werden. Es besteht aus einer Serverkomponente, in der alle Informationen

¹<https://www.news.at/a/die-ungluecke-25-jahren-unfaelle-gross-events-stadien-273976>
(Zuletzt aufgerufen am: 10.05.2020)

1 Einleitung

gespeichert sind und an die dementsprechenden Einheiten weiterleitet. Neben dem Server gibt es eine Weboberfläche, die für die Einsatzzentrale konzipiert wurde und somit alle Informationen übersichtlich darstellt. Für alle mobilen Einheiten gibt es eine eigene Android Applikation, damit auch die mobilen Einsatzkräfte auf alle relevanten Informationen zugreifen und gegebenenfalls Nachrichten an andere Einheiten versenden können.

Auf einer Kartenansicht werden alle Einheiten auf ihrer aktuellen Position dargestellt, wodurch es für den Einsatzleiter um einiges leichter ist sich ein umfassendes Lagebild über die aktuelle Situation zu verschaffen. Neben den Einheiten können auch Alarmmeldungen von angeschlossenen Sensorsystemen visuell dargestellt werden, um schnell darauf reagieren zu können.

Mit diesem System sind alle Einheiten besser miteinander vernetzt und haben immer einen Überblick auf die aktuelle Position der anderen Einheiten um so besser auf die Situation reagieren zu können. Durch die Kartenansicht kann der Einsatzleiter die Einsatzkräfte auch effizienter positionieren und sie an aktuelle Ereignisse anpassen ohne den Überblick zu verlieren.

2 Grundlagen

2.1 Smartphone

Das Smartphone hat heutzutage einen sehr großen Stellenwert und ist im alltäglichen Leben nicht mehr wegzudenken. Laut Statista¹ benutzen bereits 97% der Menschen in Österreich ein Smartphone, sowie 63% ein Tablet, wie man in der Abbildung 2.1 sehen kann².

Aufgrund dieser beeindruckenden Zahl ist es wichtig, dass man ebenfalls eine Handy-Applikation entwickelt um alle Zielgruppen abzudecken.

2.2 Android

2.2.1 Verbreitung

Für Smartphones gibt es eigene Betriebssysteme von denen Android³ und iOS⁴ die am meist verbreitetsten sind. Bei einer weiteren Statistik von Statista in Abbildung 2.2 sieht man die Verteilung der Betriebssysteme in Österreich. Hier erkennt man deutlich, dass aktuell Android der Marktführer mit einem Anteil von über 64% ist⁵.

¹<https://de.statista.com> (Zuletzt aufgerufen am: 14.01.2020)

²<https://de.statista.com/statistik/daten/studie/668880/umfrage/smartphone-und-tablet-nutzer-in-oesterreich/> (Zuletzt aufgerufen am: 14.01.2020)

³https://www.android.com/intl/de_de/ (Zuletzt aufgerufen am: 14.01.2020)

⁴<https://www.apple.com/at/ios/ios-13/> (Zuletzt aufgerufen am: 14.01.2020)

⁵<https://de.statista.com/statistik/daten/studie/297049/umfrage/bevorzugte-betriebssysteme-fuer-mobile-endgeraete-in-oesterreich/> (Zuletzt aufgerufen am: 14.01.2020)

2 Grundlagen

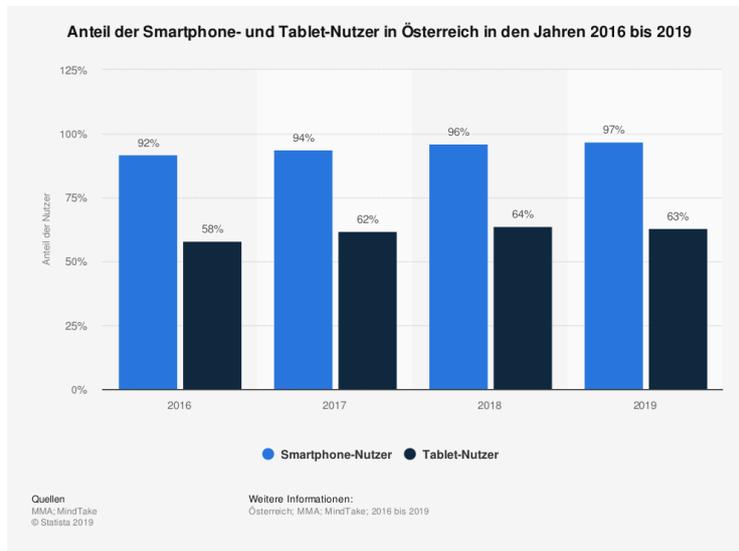


Abbildung 2.1: Die Anzahl der Smartphone-Nutzer in Österreich liegt bereits bei 97%².

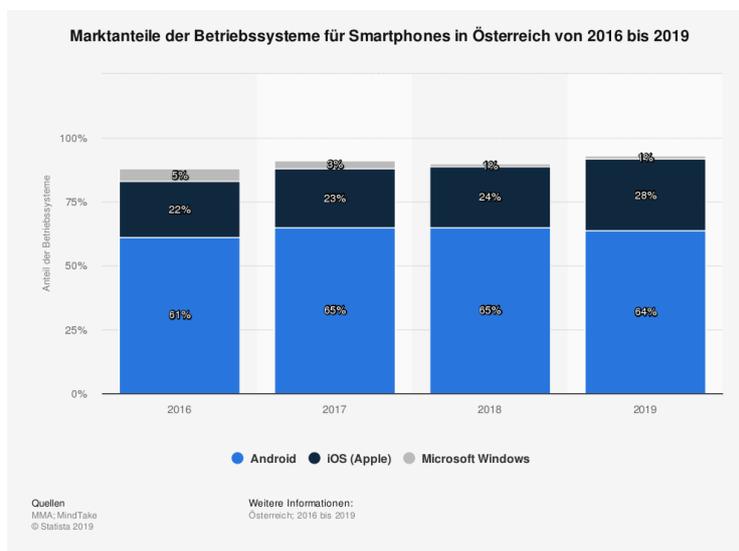


Abbildung 2.2: Verteilung der Smartphone-Betriebssysteme in Österreich mit einem Android-Marktanteil von über 64%⁵.

2 Grundlagen

Name	Version	API	Veröffentlichung
Base	1.0-1.1	1-2	23. September 2008
Cupcake	1.5	3	27. April 2009
Donut	1.6	4	15. September 2009
Éclair	2.0-2.1	5-7	26. Oktober 2009
Froyo	2.2-2.2.3	8	20. Mai 2010
Gingerbread	2.3-2.3.7	9-10	6. Dezember 2010
Honeycomb	3.0-3-2-6	11-13	22. Februar 2011
Ice Cream Sandwich	4.0-4.0.4	14-15	18. Oktober 2011
Jelly Bean	4.1-4.3.1	16-18	9. Juli 2012
KitKat	4.4-4.4.4	19-20	31. Oktober 2013
Lollipop	5.0-5.1.1	21-22	12. November 2014
Marshmallow	6.0-6.0.1	23	5. Oktober 2015
Nougat	7.0-7.1.2	24-25	22. August 2016
Oreo	8.0-8.1	26-27	21. August 2017
Pie	9.0	28	6. August 2018
Android 10	10.0	29	3. September 2019

Tabelle 2.1: Alle Android-Versionen im Überblick⁶

2.2.2 Versionen

Android gibt es bereits seit September 2008. Aus diesem Grund gibt es auch schon sehr viele Versionen, die alle in Tabelle 2.1 aufgelistet sind. Angefangen mit der Version 1.5 bis inklusive Version 9.0, bekam jede Version einen Namen einer Süßigkeit im Alphabet aufsteigend. Die Version 1.5 hieß "Cupcake", 1.6 hieß "Donut" und so weiter bis zur Version 9.0 mit dem Namen "Pie". Bei der aktuellsten Version 10.0, entschied man sich gegen diese Tradition und wird daher nur Android 10 genannt⁶.

In der nächsten Tabelle 2.2 sieht man die Verbreitung aller Versionen mit Stand vom 7. Mai 2019. Wie man erkennen kann verwenden rund 60% noch immer eine Version die bereits drei Jahre alt ist, obwohl jährlich eine neue

⁶https://en.wikipedia.org/wiki/Android_version_history (Zuletzt aufgerufen am: 15.01.2020)

2 Grundlagen

Name	Version	API	Verbreitung
Gingerbread	2.3-2.3.7	9-10	0.3%
Ice Cream Sandwich	4.0-4.0.4	14-15	0.3%
Jelly Bean	4.1-4.3.1	16-18	3.2%
KitKat	4.4-4.4.4	19-20	6.9%
Lollipop	5.0-5.1.1	21-22	14.5%
Marshmallow	6.0-6.0.1	23	16.9%
Nougat	7.0-7.1.2	24-25	19.2%
Oreo	8.0-8.1	26-27	28.3%
Pie	9.0	28	10.4%

Tabelle 2.2: Die Verbreitung aller Android-Versionen (Stand: 7. Mai 2019)⁷

Version zur Verfügung gestellt wird. Ebenfalls verwenden nur 10% aller Android-Nutzer die aktuellste Version⁷.

Aus diesem Grund muss man als Entwickler darauf achten, dass die Applikation dann auch für einen Großteil der Nutzer funktioniert. Denn jede API-Version (Application Programming Interface) bringt neue Funktionen mit sich. Will man nun auch ältere Versionen noch unterstützen muss man versuchen diese neuen Funktionen zu umgehen, beziehungsweise bei der Implementation verschiedene Lösungswege in Betracht ziehen.

2.2.3 Activity-Lebenszyklus

Bei Android muss man auch den Lebenszyklus von Activities beachten, da es dahingehend eigene Funktionen gibt die in einer bestimmten Reihenfolge aufgerufen werden. Denn genauen Ablauf sieht man in der Abbildung 2.3.

Activities in Android sind jene Klassen, die auf dem Bildschirm angezeigt werden. Diese Activities werden vom System über sogenannte „activity stacks“ gemanagt. Sobald eine neue Activity gestartet wird kommt sie oben auf den aktiven Stack und ist für den Nutzer sichtbar. Die vorherige Activity

⁷<https://developer.android.com/about/dashboards> (Zuletzt aufgerufen am: 15.01.2020)

2 Grundlagen

befindet sich unterhalb im Stack und ist erst nach Beenden der aktiven Activity wieder sichtbar⁸.

Man kann die Activity in drei Lebenszyklen einteilen:

- **Gesamte Lebenszeit**
Dieser Abschnitt beschreibt die gesamte Lebensdauer der Activity. Sie beginnt beim ersten Aufruf von *onCreate()* und endet mit dem Aufruf von *onDestroy()*. In diesen zwei Funktionen kann man alle Member-Variablen initialisieren beziehungsweise wieder freigeben oder stoppen.
- **Sichtbare Lebenszeit**
Wenn die Activity sichtbar ist, erreicht sie diesen Abschnitt, der mit der Funktion *onStart()* eingeleitet und mit *onStop()* wieder verlassen wird. In diesem Zeitraum ist die Activity sichtbar, jedoch nicht zwangsläufig im Vordergrund. Es besteht die Möglichkeit, dass eine andere Activity, die nicht den ganzen Bildschirm ausfüllt, im Vordergrund ist und mit dem Benutzer interagiert. In diesem Lebensabschnitt können zum Beispiel alle Ressourcen verwaltet werden die notwendig sind um die Activity anzeigen zu können.
- **Vordergrund Lebenszeit**
In diesem Abschnitt ist die Activity im Vordergrund und interagiert mit dem Benutzer. Dieser wird durch den Aufruf von *onResume()* gestartet und endet mit *onPause()*. Da dieser Abschnitt relativ oft aufgerufen werden kann, sollte man in diesen Funktionen keinen rechenaufwendigen Code haben.

Sobald eine Activity nicht sichtbar für den User ist kann sie vom System beendet werden und startet bei erneutem Aufruf mit *onCreate()* und muss somit wieder neu initialisiert werden⁸.

⁸<https://developer.android.com/reference/android/app/Activity> (Zuletzt aufgerufen am: 15.01.2020)

2 Grundlagen

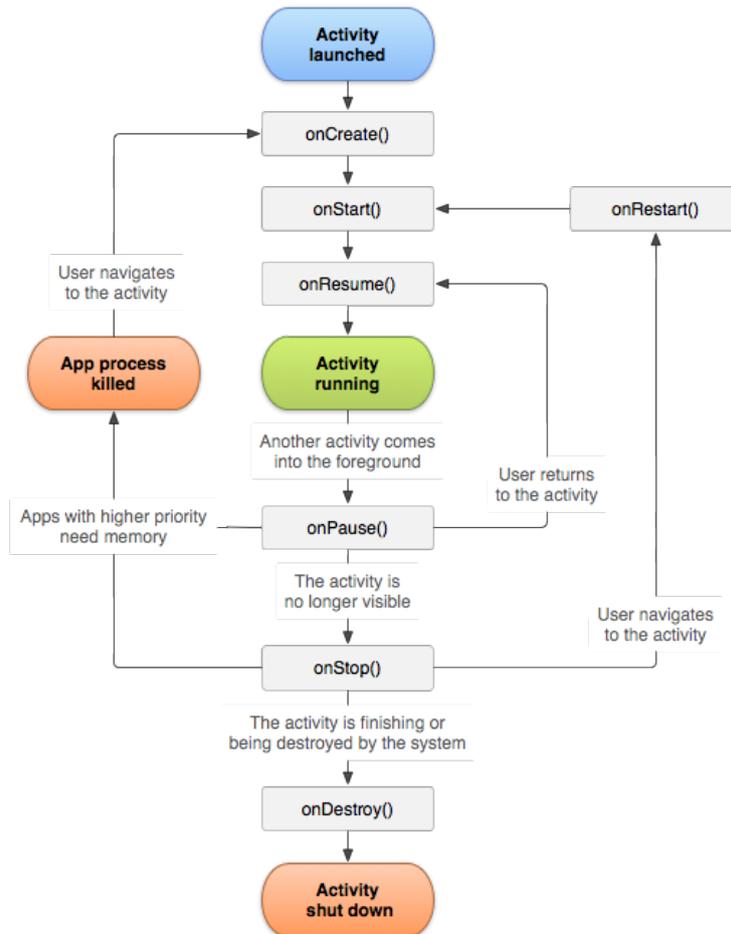


Abbildung 2.3: Android-Activity-Lebenszyklus⁸

2.3 Firebase Cloud Messaging

Firestore Cloud Messaging (FCM) ist eine Cross-Plattform-Lösung zum Senden und Empfangen von Nachrichten. Mit diesem Service ist es möglich Nachrichten zu empfangen und Benachrichtigungen anzuzeigen ohne regelmäßig beim Server abfragen zu müssen ob es neue Nachrichten gibt. Der Dienst kann sowohl für die Android Applikation als auch für die Weboberfläche verwendet werden⁹.

⁹<https://firebase.google.com/docs/cloud-messaging> (Zuletzt aufgerufen am: 16.01.2020)

3 Verwandte Arbeiten

Da ich so gut wie keine wissenschaftliche Literatur über bestehende Systeme einer ähnlichen Managementlösung gefunden habe, gehe ich in diesem Kapitel etwas genauer auf verwandte Arbeiten ein, die sich mit den Schwerpunkten dieser Arbeit befassen.

Folgende Arbeit befasst sich mit der Abstimmung des Arbeitsablaufs und der Entscheidungsfindung in der Grenzsicherheit mit unterstützenden Informations- und Kommunikationssystemen. Die Arbeit beschränkt sich auf die Grenzsicherheit, jedoch gehen auch hier einige wichtige Punkte für eine effiziente Managementlösung hervor. Damit die Integration mehrere Systeme in eine Umgebung effektiv ist, müssen zwei Aspekte beachtet werden. Einerseits muss die Durchführung bestimmter Operationen sowie der Informationsaustausch koordiniert und nahtlos möglich sein. Andererseits müssen die Nutzer des Systems ihren Fokus auf die eigentliche Arbeit und nicht auf die Benutzung der Anwendung legen können. Der Fokus kann durch die vielen Informationssystemen zwischen den unterschiedlichen Beteiligten (Private Sicherheitskraft, Polizei, Rettung) gestört werden. Deshalb ist es vom Vorteil, wenn man ein einheitliches System verwendet. Das System muss nicht nur bei der Erfassung und dem Austausch von Daten helfen, sondern auch eine intuitive Analyse unterstützen [May+11].

Laut Wu und Zhang entsteht die Schwierigkeit der Entscheidungsfindung im Management dann, wenn mehrere verschiedene Organisationen daran beteiligt sind. Dann reicht die Übermittlung der Informationen über E-Mail oder gemeinsam genutzte Datenbanken nicht aus, da es kein gegenseitiges Verständnis garantiert. Daher sind sie für kartenbasierte Informationssysteme, die alle Beteiligten verwenden können und auf denen alle Informationen dargestellt werden, um so nicht mehr durch mehrere System abgelenkt zu werden [WZ09].

3 Verwandte Arbeiten



Abbildung 3.1: Monitoring-Software für die einfache Lokalisierung von aktiven Einheiten bei Großveranstaltungen [Köf+18a]

3 Verwandte Arbeiten

Die Arbeit von Köfler, Pammer-Schindler, Almer und Schnabel, war sehr relevant für diese Arbeit, da sie die Grundprobleme von aktuellen Managementlösungen bei Veranstaltungen in Österreich genau in Augenschein nimmt. Daraus geht hervor, dass es typischerweise keine geteilte Infrastruktur zwischen den verschiedenen Organisationen, die daran beteiligt sind, gibt. Einzig und allein die Radiokommunikation wird manchmal geteilt. Während sich der private Sicherheitsdienst über Funk beziehungsweise über einen Nachrichtendienst austauscht, verwenden offizielle Einsatzkräfte wie Bundesheer, Polizei, Feuerwehr oder auch die Rettung TETRA (Terrestrial Trunked Radio). Dieses System speichert automatisch die Position und die Kommunikation in einer Protokolldatei ab. In einer Einsatzzentrale wurde außerdem eine Landkarte aus Papier verwendet, auf der alle Einheiten mit der zugewiesenen ID und deren Name eingezeichnet waren. Der Sicherheitsdienst verwendete auch eine eigene Software, die die aktuelle GPS-Position der Einheiten anzeigte, jedoch wurde sie so gut wie nicht verwendet, sondern kam nur bei Notfällen zum Einsatz. Alle Vorfälle wurden von Einheiten nur mündlich weitergegeben und der Operator in der Einsatzzentrale musste diese erst in vordefinierten Formularen eintragen und der Position der Einheit zuordnen. Dadurch ist es recht kompliziert die Informationen mit den anderen Organisationen zu teilen. Deshalb ist es wichtig eine immer aktuelle Anzeige über alle Informationen, so wie deren Position zu haben, auf die alle beteiligten Organisationen Zugriff haben, um den Austausch so schnell und einfach wie möglich realisieren zu können. Eine erste Version der Software ist in Abbildung 3.1 zu sehen. Darauf basierend konnte das Grundgerüst dieser Arbeit entwickelt werden [Köf+18a].

Frings, Engelbach und Remondes haben ebenfalls ein Informations- und Kooperationsportal entwickelt. Dieses System ist web-basierend und dient als Unterstützung für die beteiligten Organisationen bei Großveranstaltungen. Dieses Projekt spezialisiert sich jedoch mehr in Richtung der Planung und Nachbereitung von Veranstaltungen. Insbesondere sollen hier alle Daten auf einer leicht zugänglichen, aber geschützten Seite hinterlegt sein. Dadurch können alle Beteiligten mit den jeweiligen Zugriffsrechten auf Dokumente wie Einsatzpläne, raumbezogene Informationen oder auch Unwetterwarnungen zugreifen [FER10].

Das Projekt „e-Triage“ beschäftigt sich damit, dass das Kommandocenter, sowie die Einsatzkräfte die Informationen über die Art und Anzahl der

3 Verwandte Arbeiten



Abbildung 3.2: Monitoring-Software zur Visualisierung der geschätzten Personenanzahl als Heatmap [Köf+18b]

Verletzten, so schnell wie möglich bekommt, damit jede verletzte Person eine optimale Versorgung erhält. Damit die Informationsübertragung so gut als möglich funktioniert, hat dieses Projekt fast die gleichen Anforderungen wie diese Arbeit.

1. Das System muss skalierbar und von allen beteiligten Organisationen benutzbar sein.
2. Das System muss dynamisch sein. Da die Einsatzkräfte nicht immer durchgehend daran arbeiten und auch Pausen haben, müssen sie sich ab- beziehungsweise wieder anmelden können.
3. Die Benutzeroberfläche muss einfach zu bedienen sein. Denn es gibt sehr viele unterschiedliche Benutzer und auch in einer Stresssituation müssen sie in der Lage sein, das System bedienen zu können.
4. Die Kommunikation muss sicher sein. Es muss gewährleistet sein, dass nur autorisierte Personen das System verwenden können, da auch persönliche Daten übermittelt werden.

Das satellitengestützte Kommunikationssystem arbeitet mit terrestrischen Funkzellen, passenden Endgeräten mit einer eigenen Anwendungssoftware für die Einsatzkräfte vor Ort und einer automatisch synchronisierten Datenbank [GD10].

Ein weitere Arbeit, worauf diese hier aufgebaut ist, ist das Paper von Köf-ler, Almer, Schnabel, Perko und Pammer-Schindler. Sie befasst sich mit

3 Verwandte Arbeiten

einem umfassenden Monitoring von Massenbewegungen um kritische Situationen bei Großveranstaltungen vorhersehen zu können. Während in der Arbeit [Köf+18a] die Grundprobleme von aktuellen Managementlösungen bei Veranstaltungen betrachtet und eine Software die diese Probleme beheben soll entwickelt wurde, bezieht sich diese Arbeit auf die schnelle und effiziente Detektion von Massenbildungen um rechtzeitig eingreifen zu können. Dafür wird ein Management- und Führungsunterstützungssystem verwendet um ein echtzeitnahes Lagebild zu ermöglichen und zeitkritische Entscheidungsprozesse zu unterstützen. Das System besteht aus einer digitalen Karte auf der georeferenzierte Daten verfügbar sind. Als Datenquellen werden Luftbilder, stationäre Kameras, sowie die Smartphones, auf denen eine eigene App installiert ist, um Daten an die Leitstelle senden und Befehle empfangen zu können, hergenommen. Auf der interaktiven Karte, zu sehen in Abbildung 3.2 wird die geschätzte Personenanzahl und Personendichte als Heatmap dargestellt. Durch das Kontextmenü besteht auch die Möglichkeit bestimmte Aktionen durchzuführen, wie zum Beispiel eine Einheit dorthin zu schicken oder, dass man nach einer gewissen Zeit wieder auf die Situation hingewiesen wird. Zusätzlich zu dieser Software wird auch die Software von der oben beschriebenen Arbeit verwendet [Köf+18b].

Die Befehlssoftware „ILIAS-HE“, ebenfalls ein System für softwareunterstütztes Informations- und Kommunikationsmanagement, wird von der hessischen Polizei und von anderen globalen Organisationen, die sich mit Sicherheitsaufgaben beschäftigen, eingesetzt. Das System bietet einen verbesserten Belegfluss, optimierte Recherchemöglichkeiten und Hilfen zu Lagebilddarstellungen. Außerdem werden alle Informationen frei zugänglich in einem Informationspool bereitgestellt. Ein wichtiger Punkt hierbei ist, dass die Software den Führungsstab nur unterstützen und keine Entscheidungen treffen soll. Dieses System kann nicht nur auf jedem Computer verwendet werden, sondern unterstützt auch den Zugriff von mobilen Endgeräten [Heio9].

Folgende Arbeit befasst sich mit einem neuen Ansatz für Anwendungen für Notfallsituationen. Dabei wurde ein formales semantisches Domänenmodell in ein ereignisgesteuertes Entscheidungsunterstützungssystem integriert. Die zugrunde liegenden Datenbanken liefern nur begrenzte Informationen, doch mit diesem Modell können sich wertvolle Erkenntnisse über den aktuellen Kontext ableiten lassen, um so eine intelligente Argumen-

3 Verwandte Arbeiten

tation herzuleiten. Es wird gezeigt wie man damit Notfallereignisse und ein dazugehöriges Szenario erstellen und wie man den Zustand eines Feuerwehrmannes während so einer Situation überwachen kann. Eingehende Echtzeitdaten werden vom Modell übernommen, von denen dann das ereignisgesteuerte System, bei Notwendigkeit Alarme an die dementsprechenden Einheiten sendet. Die Anwendung die dafür entwickelt wurde spezialisiert sich auf Bergungseinsätze für die Feuerwehr. Damit die gesuchten Personen schnell und sicher für die Einsatzkraft gefunden werden können und der Einsatzleiter immer einen Überblick über den Gesundheitszustand und den Verlauf seines Teams überwachen kann, um bei kritischen Situationen einschreiten zu können [HOT13].

4 Anwendungsbereiche

Ein wichtiger Punkt in vielen sicherheitsrelevanten Bereichen ist die schnelle und unkomplizierte Verknüpfung aller beteiligten Personen und auch sonst relevanten Informationen. Diese Informationen wie zum Beispiel Sensordaten oder Textnachrichten müssen intern ausgewertet und dann an die zuständigen Rollen weitergeleitet werden.

Ein geeigneter Anwendungsfall wäre beispielsweise für die Security bei Konzerten oder anderen Veranstaltungen. Hier ist es ebenfalls wichtig, dass alle Einheiten untereinander kommunizieren können und der Einsatzleiter einen Überblick über alle Einheiten hat und schnell Kommandos verteilen kann.

Bei Großveranstaltungen bei denen neben der Security auch andere Organisationen wie zum Beispiel Polizei, Rettung oder Feuerwehr vor Ort sind, ist es mit Hilfe des Projektes um einiges effizienter die Informationen an alle zuständigen Organisationen zu verteilen. Somit kann es zu keinem Informationsverlust kommen, wie es sonst der Fall sein könnte, wenn erst der Einsatzleiter die Notdienste verständigen muss, obwohl er nicht direkt vor Ort ist.

Als weitere Informationsquelle neben den Einheiten können auch Sensormasten eingesetzt werden, die bei Detektion bestimmter Ereignisse einen Alarm an die Leitstelle sendet, die den wiederum verifiziert und bei Bedarf Einheiten in der Nähe verständigt. Zusätzlich stellt sie auch einen Livestream ihres aktuellen Sichtfeldes zur Verfügung.

Das System kann auch bei polizeilichen Einsätzen sowie beim Grenzschutz eingesetzt werden, da hier nahezu die gleichen Anforderungen sind, wie bei der Security.

4 Anwendungsbereiche

Daraus ergeben sich verschiedene Rollen mit unterschiedlichen Aufgaben und andere Informationen benötigen:

- Mobile Einheiten (Sicherheitsdienst, Polizeistreife, Soldaten (Bundesheer))
 - Alle mobilen Einheiten haben die Aufgabe Sichtungen an die Leitstelle zu melden und bei kritischen Situation einzugreifen
 - Sie müssen in der Lage sein alle benötigten Informationen an die Leitstelle zu senden und von der wiederum Einsatzbefehle zu erhalten
- Leitstellen (Einsatzleiter, Polizeikommando, Bundesheerkommando)
 - Die Aufgabe der Leitstelle besteht darin alle aktiven Einheiten zu überwachen und ihnen Befehle zu erteilen, bzw. alle Meldungen der Einheiten zu analysieren und gegebenenfalls weitere Schritte einzuleiten
 - Dafür benötigt sie alle aktuellen Positionen der Einheiten um die Situation besser einschätzen zu können und um bei Problemen genau zu wissen wo sie sich befindet um Hilfe zu dieser Einheit senden zu können
- Hilfsorganisationen (Rettung, Feuerwehr, Polizei)
 - Die Hilfsorganisationen müssen abrufbereit sein, um bei etwaigen Notfallsituationen schnell eingreifen zu können
 - Hier reicht es aus die situationsbedingten Informationen von der Leitstelle oder den mobilen Einheiten zu empfangen

Aus diesen Anwendungsfällen und den unterschiedlichen Rollen haben wir daraus folgende Voraussetzungen abgeleitet:

- Funktionale Anforderungen
 - Weboberfläche für den Leitstand um alle Informationen übersichtlich darstellen zu können
 - Android Applikation für alle mobilen Einheiten um auch unterwegs alle Informationen abrufen zu können
 - Kartenansicht inklusive der Positionen aller Einheiten
 - Möglichkeit Nachrichten/Anweisungen versenden und empfangen zu können (Mobile Einheiten \Leftrightarrow Leitstand)

4 Anwendungsbereiche

- Unterstützung zur Anzeige des Livestreams der Sensormasten
 - Benutzerverwaltung und Zuordnung zu bestimmten Rollen
- Nicht-funktionale Anforderungen
 - Zugriffsbeschränkungen damit nur autorisierte Personen auf die Software zugreifen können
 - Passwörter verschlüsselt speichern
 - Sichere Verbindung zum Datenserver
 - Benutzerfreundliche Oberfläche sowohl bei der Weboberfläche als auch der mobilen Applikation
 - Ressourcenschonend um auch in Gebieten mit schlechter Internetanbindung eingesetzt werden zu können

5 Methodik

Das gesamte System besteht aus drei Komponenten:

- Zentraler Server
- Weboberfläche
- Android-Applikation

Eine Übersicht aller Komponenten ist in Abbildung 5.1¹ ersichtlich. Zusätzlich zum zentralen Server, der Weboberfläche und der Android-Applikation wird hier noch der Sensormast als eigene Komponente dargestellt. Dieser ist ebenfalls ins System integriert und liefert Analyseergebnisse sowie Livebilder von seinem Standort. Diese Daten können dann in der Weboberfläche und in der mobile Applikation angezeigt werden.

Der Server ist in C# implementiert und kommuniziert über WCF². Er wird dann auf einem Server gehostet um von überall mit Internetzugang darauf zugreifen zu können. Auf dem Server werden alle Daten verwaltet und zusätzlich ist er für das Weiterleiten der Nachrichten zuständig. Da der Server nicht Teil dieser Arbeit war, werde ich nicht weiter auf die genaue Implementation eingehen.

Die Weboberfläche ist für die Einsatzzentrale beziehungsweise für den Leitstand gedacht. Hier werden alle Informationen über die aktiven Einheiten übersichtlich dargestellt. Falls vom Sensormasten ein Alarm ausgelöst wird, wird die Position der erkannten Detektion ebenfalls hier angezeigt und kann vom Einsatzleiter verifiziert werden. Außerdem kann man den

¹Icons erstellt von Eucalyp, Freepik, Pixelmeetup, Kiranshastry, Monkik, Roundicons von <https://www.flaticon.com>

²<https://docs.microsoft.com/de-de/dotnet/framework/wcf/whats-wcf> (Zuletzt aufgerufen am: 17.01.2020)

5 Methodik

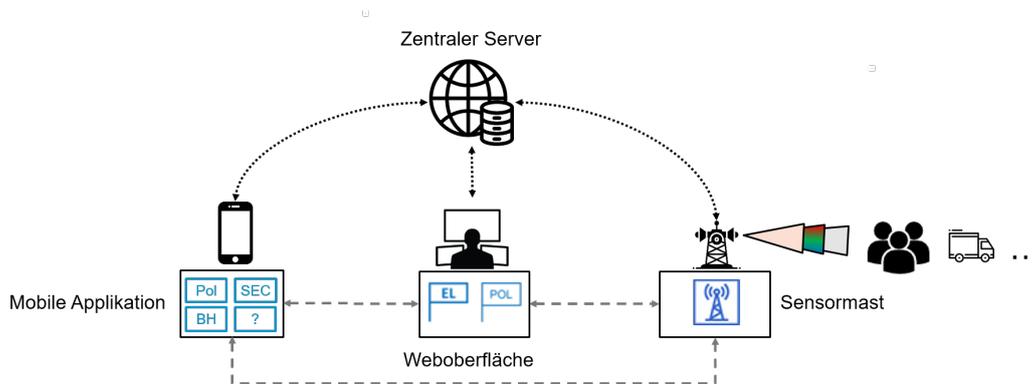


Abbildung 5.1: Übersicht aller Komponenten und deren Verbindungen zueinander¹

Einheiten Kommandos senden und den Livestream der Kameras auf den Sensormasten ansehen.

Damit auch die mobilen Einheiten Zugriff auf alle Informationen haben gibt es eine eigene Android Applikation. Sie bietet fast die selben Funktionen wie auch die Weboberfläche, nur dass die Alarmmeldungen hier nicht angezeigt werden, da sie erst von der Leitstelle verifiziert werden müssen.

Durch diese drei Komponenten ist das System sehr gut miteinander verschmolzen. Egal ob man die Weboberfläche oder die Android Applikation verwendet, sind überall alle Informationen vom Server abrufbar.

5.1 Weboberfläche

5.1.1 Konzept

Die Weboberfläche ist rein für die Einsatzzentrale konzipiert und ist damit nur für die Verwendung auf einem Desktop-Bildschirm angepasst. Für alle mobilen Nutzer gibt es daher eine eigene Android-Applikation.

Die Weboberfläche wurde mit Hilfe von Javascript, HTML (Hypertext Markup Language) und CSS (Cascading Style Sheets) entwickelt. Diese Komponenten sind die typischen Programmiersprachen, die man verwendet,

wenn man eine Webseite erstellen will. Um die Webseite zu hosten wird die Plattform Node.js verwendet ³.

Node.js

Node.js ist eine serverseitige Plattform, die sich sehr gut dazu eignet einen Webserver zu realisieren. Da die Scripts auf dem Server laufen, kann man relativ einfach dynamische Webseiten erstellen, bevor sie zum Client gesendet werden.

Um die Performance zu erhöhen wird eine ereignisgesteuerte Architektur, sowie asynchrones I/O (Input/Output) verwendet. Bei dieser Architektur wird alles durch bestimmte Ereignisse gesteuert. Diese können entweder Benutzereingaben oder auch interne Ereignisse, wie zum Beispiel der Empfang neuer Daten, sein. Dadurch lässt sich die Webseite dynamisch aufbauen, sowie aktualisieren. Bei der Verwendung von asynchroner I/O werden die im Vergleich zu den anderen recht langsamen Operationen in eigene Threads ausgelagert um effizienter zu arbeiten.

Neben den bereits genannten Vorteilen hat Node.js mit dem Paketmanager npm ebenfalls die größte Ansammlung von Open Source-Bibliotheken. Diese Bibliotheken werden von der Community zur Verfügung gestellt und können bei einer Vielzahl von Problemen helfen und somit die Entwicklung um einiges vereinfachen ⁴.

Codestruktur

Wie man in Abbildung 5.2 sehen kann, besteht die Weboberfläche im Grunde aus zwei Komponenten:

- Server
- Frontend

³<https://nodejs.org/de/> (Zuletzt aufgerufen am: 20.01.2020)

⁴<https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/> (Zuletzt aufgerufen am: 20.01.2020)

5 Methodik

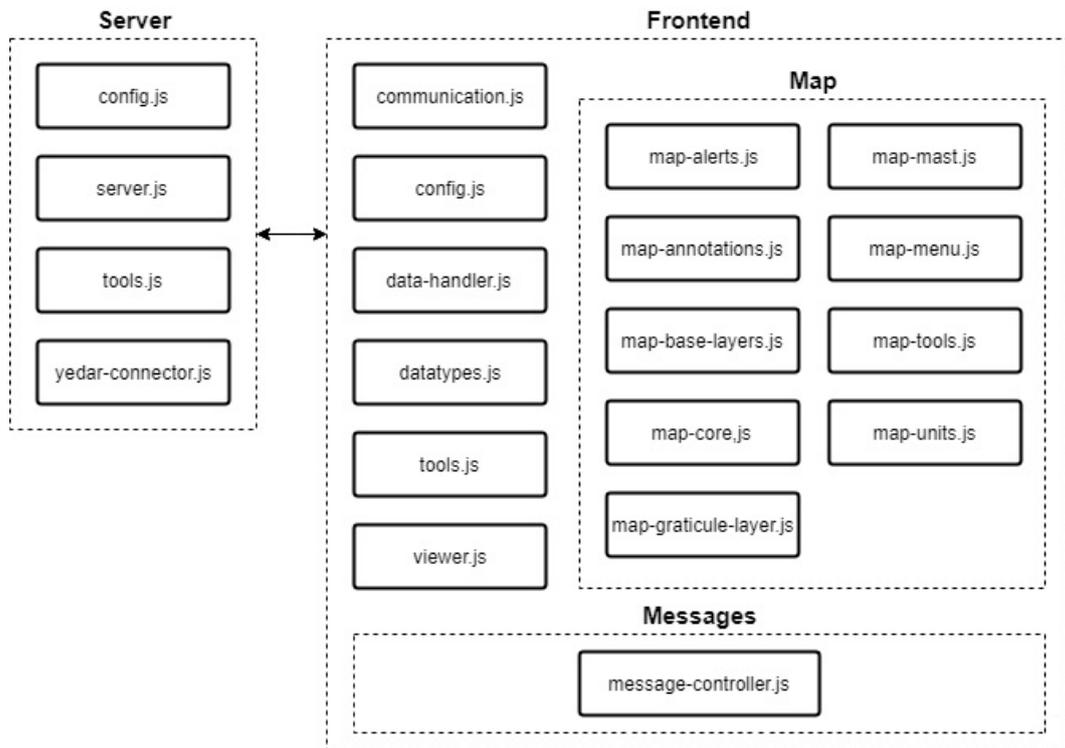


Abbildung 5.2: Weboberfläche - Programmübersicht; Die Weboberfläche besteht aus einem Server-Part und einem Frontend-Part. Diese sind wiederum in mehrere Files aufgeteilt, so dass man beim Frontend auch die zwei Unterkategorien für die Karten- und die Nachrichtenansicht veranschaulichen kann.

Die Serverkomponente ist dafür zuständig, dass der Webserver läuft und man auf das Frontend zugreifen kann. Außerdem ist sie für die Authentifizierung der User, sowie allen Anfragen die Zugriff auf bestimmte Dateien benötigen oder etwas mit dem Livestream zu tun haben, zuständig.

Das Frontend ist für die eigentliche Webseite, die dem Nutzer dann auch dargestellt wird, verantwortlich. Hier gibt es mehrere Dateien, die für die allgemeinen Bereich zuständig sind, sowie die zwei Unterbereiche für die Karten- und die Nachrichtenansicht. Neben den dargestellten Javascript-Dateien gibt es hier noch einige HTML- und CSS-Files, die für die visuelle Darstellung benötigt werden.

5.1.2 Implementierung

Die Weboberfläche wird wie bereits erwähnt mittels Node.js gehostet. Deswegen möchte ich hier ein kurzes Beispiel zeigen, wie eine solche Implementierung aussehen kann.

```
1 const http = require('http');
2 const express = require('express');
3
4 var app = express();
5
6 http.createServer(app).listen(80);
7
8 app.get('/', function(req, res) {
9   res.sendFile('index.html');
10 });
11
12 app.get('/hello', function(req, res) {
13   res.send('Hello World!');
14 });
```

Listing 5.1: Beispiel von einem einfachen Node.js-Server

Wie man in diesem Beispiel sehen kann, ist es sehr einfach einen Webservice mit Node.js zu starten. Mithilfe von *require* kann man die benötigten Packages einbinden. In diesem Fall zum Beispiel die Pakete *http* und *express*. In Zeile 6 wird der Server gestartet und dem Port 80 zugewiesen. Mit den unteren zwei Funktionen *app.get(...)* kann man bestimmen was passieren soll, wenn

5 Methodik

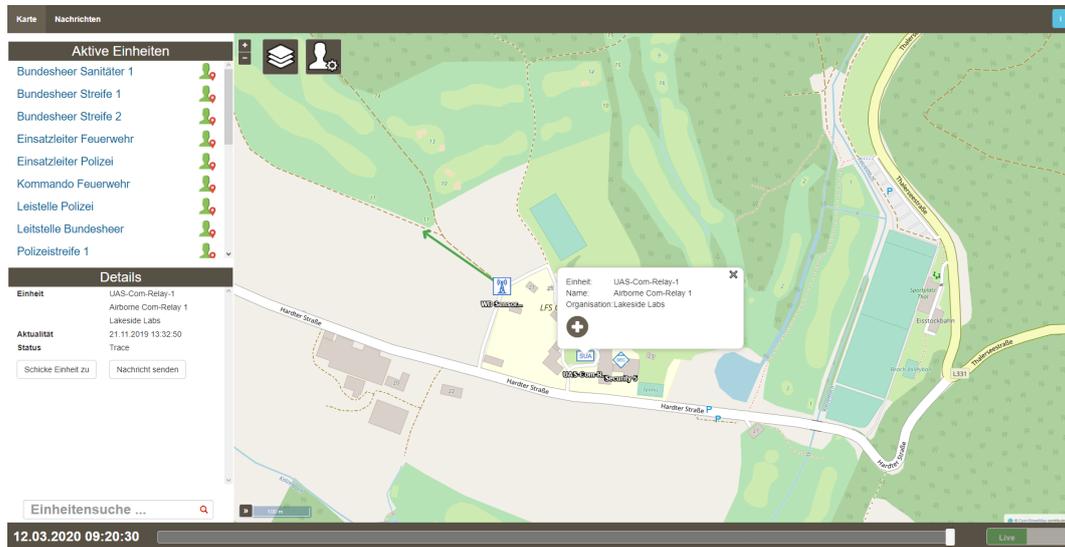


Abbildung 5.3: Weboberfläche - Kartenansicht; Übersicht über alle aktiven Einheiten in einer Liste und auf der Karte an ihrer aktuellen Position

der User die entsprechende Seite aufruft. Mit *sendFile(...)* oder auch *send(...)*, ist es dann möglich dem Nutzer vordefinierte HTML-Files oder auch einfach nur einen Text als Antwort zu senden.

Da insgesamt sehr viele Bibliotheken⁵ verwendet werden, habe ich sie in der Tabelle 5.1 zusammengefasst.

Sobald man die Weboberfläche aufruft, wird man nach Benutzernamen und Passwort gefragt. Damit wird gewährleistet, dass nur berechtigte Personen die Webseite verwenden können. Hat man sich erfolgreich angemeldet, gelangt man zur Startseite, die man in Abbildung 5.3 sehen kann.

Im linken Bereich der Seite sieht man oben, in einer Liste alle Einheiten, die derzeit aktiv sind. Mit einem Klick auf das Symbol neben dem Namen, wird die gewählte Person direkt in der Kartenansicht daneben zentriert um sie zu lokalisieren. Bei einem Klick auf den Namen werden im unteren Bereich alle Details zur Einheit dargestellt. Zusätzlich besteht hier auch die Möglichkeit der Einheit eine Nachricht zu senden oder sie an einen bestimmten Ort

⁵Alle genannten Bibliotheken sind unter <https://www.npmjs.com/package/<packagename>> abrufbar. (Zuletzt aufgerufen am: 03.02.2020)

5 Methodik

Bibliothek	Version
bootstrap	3.3.6
bootstrap-slider	10.0.0
bootstrap-switch	3.3.4
fselect	1.0.1
jquery	2.2.3
jquery-ui	1.12.1
jsgrid	1.5.3
ol	4.3.2
js-cookie	2.2.1
moment	2.18.1
split.js	1.3.5
basic-auth	2.0.1
cookie-parser	1.4.4
cors	2.8.5
express	4.16.2
express-http-proxy	1.5.1
express-logging	1.1.1
express-request-proxy	2.2.2
fast-xml-parser	3.12.20
http-forward	0.1.3
node-fetch	1.7.3
redis	2.8.0

Tabelle 5.1: Weboberfläche - Verwendete Bibliotheken inklusive der verwendeten Versionsnummer⁵

zu schicken. Falls es sehr viele aktive Einheiten gibt, ist auch eine Suche integriert bei der man ebenfalls zur aktuellen Position der eingegebenen Einheit gebracht wird. Der grüne Pfeil in der Ansicht symbolisiert die Richtung in die der Sensormast ausgerichtet ist, wodurch das Sichtfeld des Masten besser eingeschätzt werden kann.

Kartenansicht

In der Kartenansicht, die in Abbildung 5.3 zu sehen ist und den Großteil der Webseite ausmacht, werden auch alle aktiven Einheiten inklusive ihrer Position angezeigt. Alle Einheiten werden mit einem taktischen Symbol, das die unterschiedlichen Einheitentypen repräsentiert, dargestellt. Da es bei zu vielen Einheiten unübersichtlich werden kann, besteht die Möglichkeit die Einheiten zu filtern. Neben dem filtern der Einheiten, ist es auch möglich das Kartenlayout der Karte zu ändern um die Darstellung je nach Situation anpassen zu können.

Wenn man auf eine Einheit in der Kartenansicht anklickt, werden ebenfalls die Details in einem Popup, so wie im linken unteren Bereich angezeigt. Bei einem Rechtsklick öffnet sich das Kontextmenü, in dem man auch Nachrichten an die Einheiten versenden, oder die Einheit an eine bestimmte Position schicken kann.

Auf der Karte werden auch alle Alarmmeldungen vom Sensormasten an der detektierten Position angezeigt und man bekommt eine eigene Benachrichtigung, falls eine Alarmmeldung reinkommt. Damit kann der Benutzer schnell auf den Alarm reagieren, ihn verifizieren in dem er sich das Livebild der Kamera auf dem Sensormasten, die den Alarm detektiert hat, ansieht und dann gegebenenfalls direkt eine Einheit dorthin schicken.

Um vergangene Einsätze zu analysieren, gibt es einen eigenen Replay-Modus. Dabei kann man sich die Daten von einem bestimmten Zeitpunkt noch einmal anschauen, in dem man die Zeitleiste am unteren Rand der Webseite an den gewünschten Zeitpunkt setzt und auf Start klickt. Danach werden auf der Karte alle aktiven Einheiten an der Position dargestellt, an der sie zu diesem Zeitpunkt waren. Alarme, die zu diesem Zeitpunkt reingekommen sind, werden ebenfalls dargestellt. Mit einem Klick auf

5 Methodik

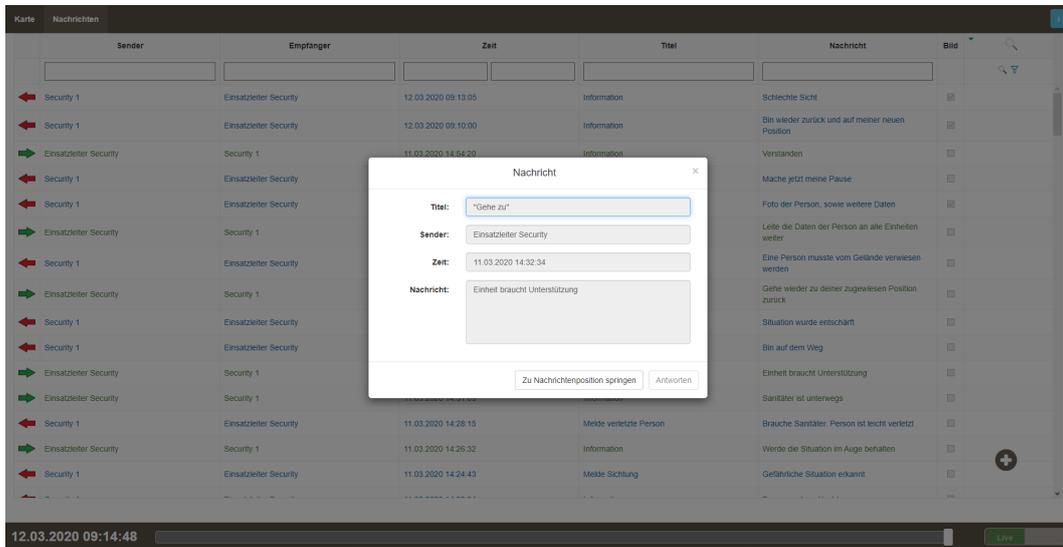


Abbildung 5.4: Weboberfläche - Nachrichtenansicht; Tabelle mit allen eingehenden und ausgehenden Nachrichten, sowie der Möglichkeit auf Nachrichten zu antworten oder zur Position zu springen von der aus die Nachricht versendet wurde.

Replay gelangt man dann wieder zurück zum Livemodus, in dem alle aktuellen Daten angezeigt werden.

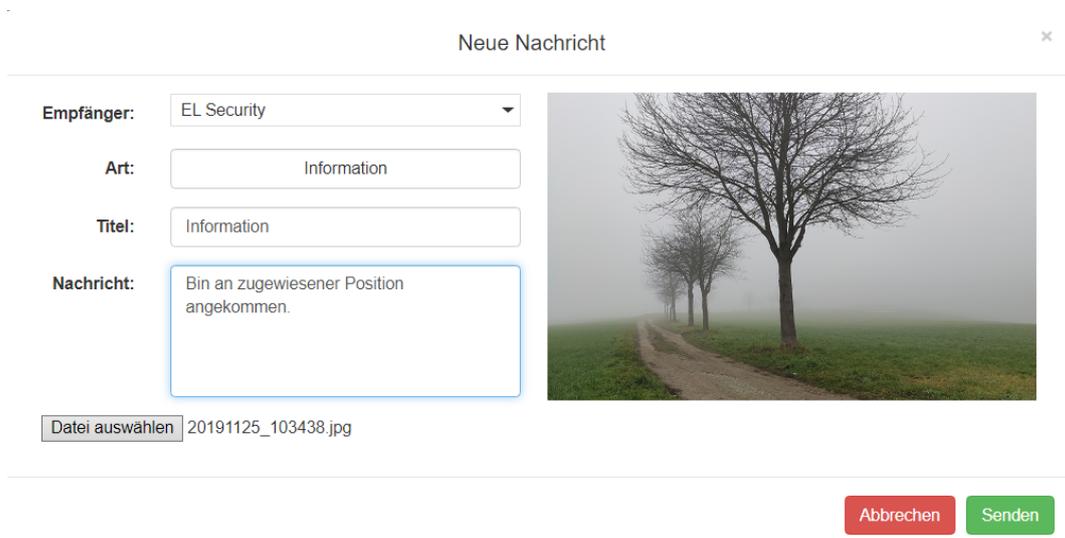
Nachrichtenansicht

In der Nachrichtenansicht werden alle eingehenden sowie ausgehenden Nachrichten in einer Tabelle (siehe Abbildung 5.4) dargestellt.

Um deutlich zu erkennen ob es eine eingehende oder eine ausgehende Nachricht ist, werden die ausgehenden Nachrichten mit einem grünen Pfeil und in grüner Schrift dargestellt und die eingehenden Nachrichten in blauer Schrift mit einem roten Pfeil in die andere Richtung.

Die Nachrichten können nach verschiedenen Eigenschaften sortiert und auch gefiltert werden, um schnell die gewünschte Nachricht finden zu können. Diese Filterung und auch Sortierung wird vollständig clientseitig ausgeführt. Vom Server bekommt man nur eine Liste mit allen Nachrichten

5 Methodik



Neue Nachricht

Empfänger: EL Security

Art: Information

Titel: Information

Nachricht: Bin an zugewiesener Position angekommen.

Datei auswählen 20191125_103438.jpg

Abbrechen Senden

Abbildung 5.5: Weboberfläche - Neue Nachricht; Neben dem Empfänger und dem Nachrichteninhalt, kann auch die Art der Nachricht sowie der Titel gesetzt werden. Gegebenenfalls kann auch ein Bild angehängt werden.

die von oder an die angemeldete Einheit versendet wurden. Per Klick auf eine Nachricht werden in einer Dialogbox alle Details zur Nachricht, sowie angehängte Bilder, angezeigt und man kann zur Position springen von der aus die Nachricht versendet wurde, oder direkt auf die Nachricht antworten.

Außerdem kann man hier auch eine neue Nachricht an einen oder mehrere Empfänger senden. Dabei öffnet sich eine Dialogbox, in der man alle Informationen eintragen, sowie Bilder hinzufügen kann (siehe Abbildung 5.5).

Videoansicht

Um den Livestream der auf den Sensormasten montierten PTZ-Kamera (Pan, Tilt, Zoom) darzustellen, wird der Benutzer auf eine weitere Seite weitergeleitet. Auf dieser Seite kann man nicht nur den Livestream der Kamera betrachten, sondern sie auch bewegen beziehungsweise die Zoomstufe anpassen. Die Webseite ist in Abbildung 5.6 dargestellt.

5 Methodik



Abbildung 5.6: Weboberfläche - Livestream-Ansicht; In dieser Ansicht kann der Livestream des Sensormasten betrachtet werden. Außerdem kann die Kamera frei bewegt und auch die Zoom-Stufe eingestellt werden.

5.2 Android Applikation

5.2.1 Konzept

Wie bereits genannt ist die Android Applikation für alle mobilen Einheiten gedacht. Um auch hier alle Informationen anzeigen zu können wurde das Layout dementsprechend angepasst. Android basiert grundlegend auf Java und ist somit auch eine objektorientierte Programmiersprache. Zusätzlich zu den normalen Klassen, gibt es hier auch sogenannte Activities. Eine Activity wird dann verwendet, wenn man dem User etwas auf dem Display anzeigen will. Sobald auf dem Smartphone ein neues Fenster angezeigt wird, startet im Hintergrund eine neue Activity. In Abbildung 5.7 sieht man eine Übersicht aller Activities, die es in dieser Applikation gibt und wie sie zusammenhängen.

Damit so viele Nutzer wie möglich die Applikation installieren können, wurde versucht die minimale API-Version so gering wie möglich zu halten. Die älteste API-Version die von dieser App unterstützt wird ist API 16.

5 Methodik

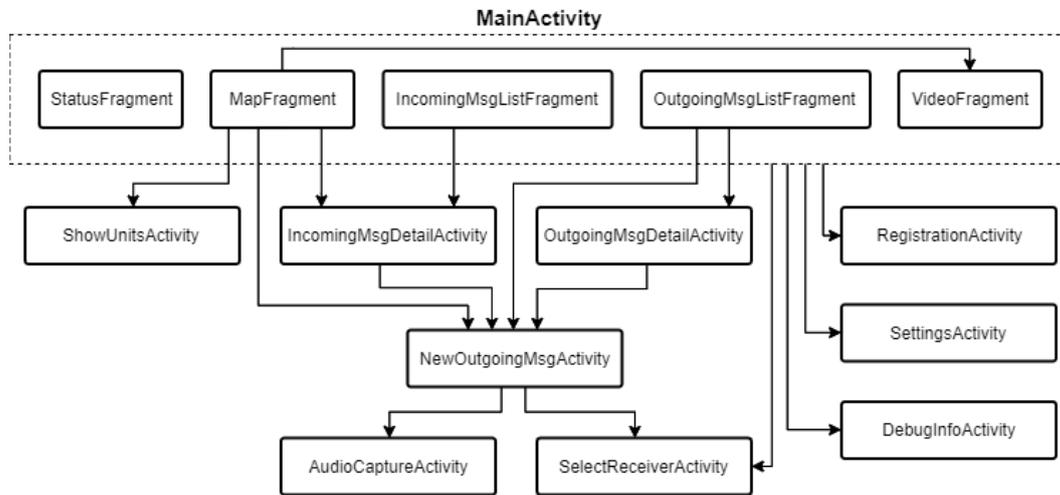


Abbildung 5.7: Android-Applikation Flussdiagramm aller Activities

Diese API-Version entspricht der Versionsnummer „4.1“ mit dem Namen *Jelly Bean* und wurde bereits im Juni 2012 veröffentlicht. Da diese Version wie man in der Tabelle 5.2 sehen kann, 99,4% aller Geräte unterstützt haben wir uns für diese Version entschieden⁶.

5.2.2 Implementierung

Damit man die Applikation vollständig nutzen kann, muss man sich beim ersten Programmstart zuerst registrieren, da der Server das Gerät sonst nicht identifizieren kann. Ist man einmal registriert, bleibt man so lange registriert, bis die Applikation deinstalliert wird. Es müssen mehrere Daten angegeben werden wie zum Beispiel Benutzername, Anzeigename, Passwort, den Typ der Einheit und weitere optionale Felder die nicht verpflichtend ausgefüllt werden müssen. In Abbildung 5.8 kann man das Registrierungsformular sehen.

Nach diesen Schritten befindet man sich zuerst im sogenannten „Freizeitmodus“ (siehe Abbildung 5.9). In diesem Modus wird die aktuelle Position

⁶<https://developer.android.com/about/dashboards> (Zuletzt aufgerufen am: 15.01.2020)

5 Methodik

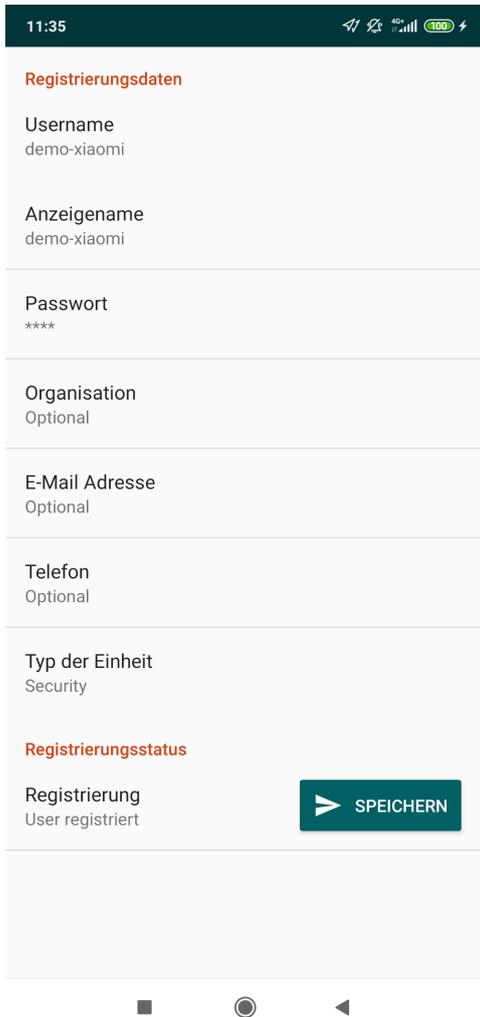


Abbildung 5.8: Android-Applikation Registrierung

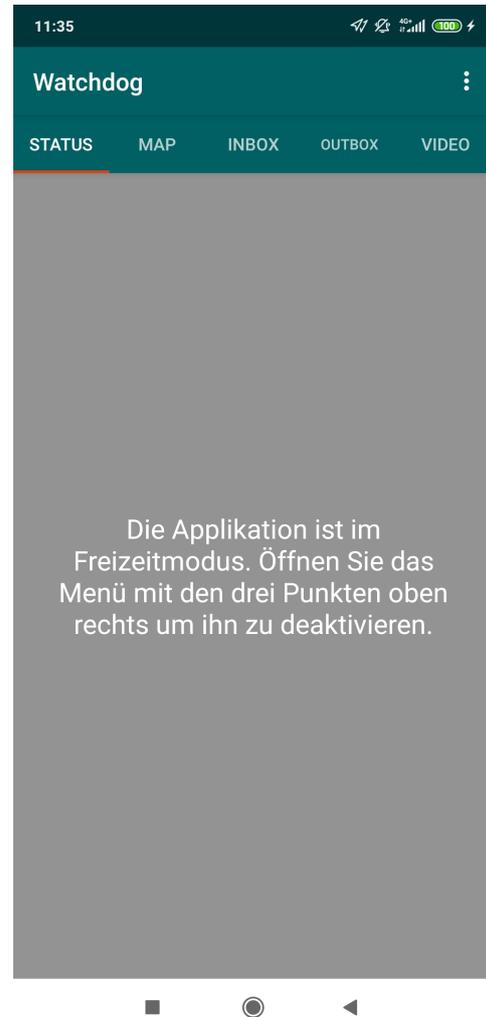


Abbildung 5.9: Android-Applikation Freizeitmodus

API	Name	Version	Unterstützte Geräte
28+	Pie	9.0+	10.4%
26+	Oreo	8.0+	38.7%
24+	Nougat	7.0+	57.9%
23+	Marshmallow	6.0+	74.8%
21+	Lollipop	5.0+	89.3%
19+	KitKat	4.4+	96.2%
16+	Jelly Bean	4.1+	99.4%

Tabelle 5.2: Android-API - Unterstützte Geräte; Übersicht mit welcher API wann wie viele der derzeit verwendeten Geräte unterstützt. (Stand: 7. Mai 2019)⁶

des Geräts nicht dauerhaft an den Server gesendet und man ist nicht aktiv. Das heißt man ist für die anderen Einheiten nicht sichtbar. Im Menü rechts oben kann man den Freizeitmodus beliebig aktivieren/deaktivieren. Nur im deaktivierten Freizeitmodus, hat man Zugriff auf alle Funktionen der Applikation.

Die MainActivity ist in fünf Fragmente aufgeteilt die jeweils eine unterschiedliche Funktion erfüllen. Ein Fragment ist so ähnlich wie eine untergeordnete Activity und kann man auch als Modul ansehen. Damit lassen sich mehrere getrennte Module in einer Activity darstellen. Diese Fragmente haben ihren eigenen Lebenszyklus, wie man in Abbildung 5.10 sehen kann⁷.

Da die Fragmente den Hauptteil der Applikation ausmachen werde ich auf sie jeweils etwas genauer eingehen.

Status-Fragment

In diesem Fragment kann man seinen aktuellen Status an den Server senden. Der Status kann folgende Werte haben:

- Im Einsatz

⁷<https://developer.android.com/guide/components/fragments> (Zuletzt aufgerufen am: 07.04.2020)

5 Methodik

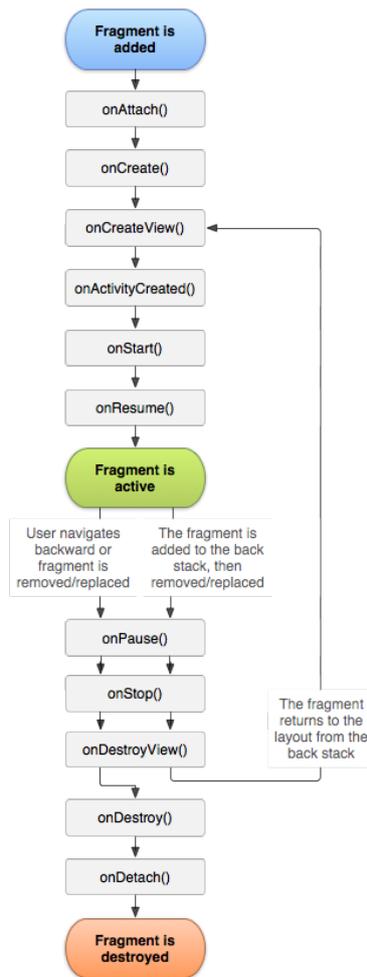


Abbildung 5.10: Android Fragment Lebenszyklus⁷

5 Methodik

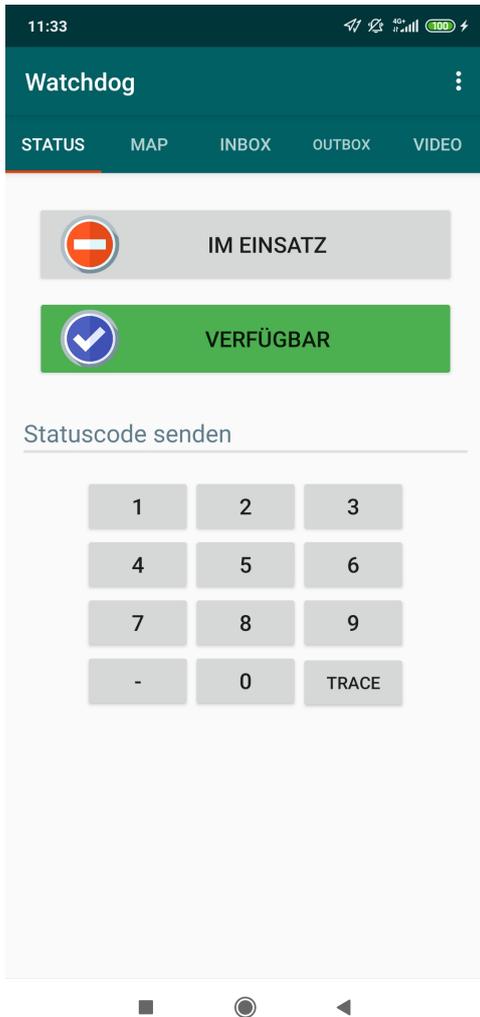


Abbildung 5.11: Android-Applikation Status-Fragment

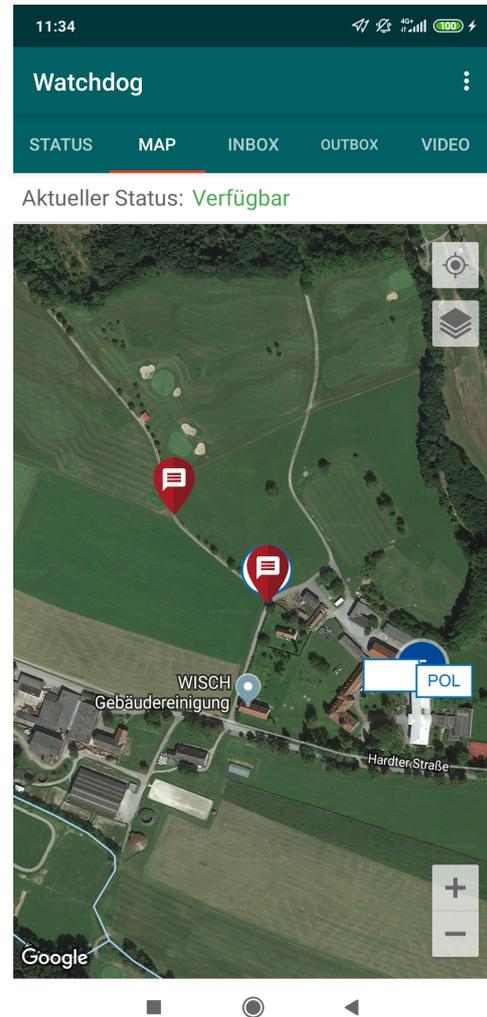


Abbildung 5.12: Android-Applikation Map-Fragment

- Verfügbar
- Statuscode

Der Statuscode ist eine Zahl von 0 bis 9 oder „trace“. Dieser Code wird oft bei Sicherheitsdiensten eingesetzt. Da sonst nichts Relevantes in diesem Fragment passiert, können wir zum Map-Fragment übergehen.

Map-Fragment

In diesem Fragment wird eine Karte angezeigt, die mit Hilfe der Google Maps SDK⁸ mit der Version 16.1.0 implementiert wurde. Da es sich nur um einen Prototyp handelt kann diese SDK verwendet werden. Ansonsten wäre eine andere SDK sicher besser, da hier vertrauliche und sicherheitsrelevante Daten angezeigt werden.

Auf dieser Karte werden die aktuellen Positionen aller aktiven Einheiten dargestellt. Die dargestellten Einheiten können auch nach Einheitentyp gefiltert werden, um sich mehr Übersicht verschaffen zu können. Neben den Einheiten werden hier auch alle eingegangenen Nachrichten an der Position dargestellt von der aus sie versendet wurden.

Durch die vielen Marker die somit auf der Karte angezeigt werden können, wurde ebenfalls die Utility-Bibliothek⁹ von der Google Maps SDK mit der Version 0.4.4 verwendet. Diese Bibliothek ermöglicht das Clustern von Marker, wodurch die Kartenansicht noch übersichtlicher gestaltet werden konnte.

Bei Klick auf die Marker werden in einem Popup zusätzliche Information, in Abhängigkeit des Markertyps, darüber angezeigt. Bei den Einheiten wären das der Name der Einheit sowie deren Organisation. Bei Nachrichten hingegen ist es die Überschrift der eingegangenen Nachricht.

Bei einem weiteren Klick auf das Popup, können dann weitere Aktionen durchgeführt werden:

⁸<https://developers.google.com/maps/documentation/android-sdk/intro> (Zuletzt aufgerufen am: 22.01.2020)

⁹<https://developers.google.com/maps/documentation/android-sdk/utility> (Zuletzt aufgerufen am: 22.01.2020)

5 Methodik

- Mobile Einheit / Einsatzzentrale ⇒ Kann man eine Nachricht an die ausgewählte Einheit senden
- Masten / Sensorsystem ⇒ Kann der Livestream des ausgewählten Masten angezeigt werden
- Nachricht ⇒ Kann die Detailansicht der ausgewählten Nachricht geöffnet werden

Will man einer anderen Einheit einen „Gehe zu“-Befehl senden, kann man dies ebenfalls machen in dem man lange auf den gewünschten Zielort klickt.

Inbox-Fragment/Outbox-Fragment

Diese zwei Fragmente stellen alle eingegangen beziehungsweise versendeten Nachrichten in einer Liste dar. Die Listeneinträge wurden versucht so informativ wie möglich zu gestalten. Man sieht alle wichtigen Informationen zur Nachricht auf einen Blick, sowie den aktuellen Status der Nachricht. Sollte eine versendete Nachricht nicht erfolgreich zugestellt werden, wird ein Fehler angezeigt und sie kann mit nur einem Klick erneut versendet werden. Falls man die ganze Nachricht lesen will, muss man nur die gewünschte Nachricht auswählen um dann zu einer Detailansicht zu kommen.

In der Outbox kann man außerdem eine neue Nachricht an einen oder mehrere Empfänger senden. Eine Nachricht besteht aus mehreren Informationen:

- Nachrichtentyp
Damit kann grob definiert werden um was für eine Nachricht es handelt, damit der Empfänger schnell erkennen kann um was es geht um dementsprechend zu handeln.
- Überschrift
In der Überschrift kann noch einmal kurz zusammengefasst werden um was es in der Nachricht geht. Im Normalfall reicht es aus wenn hier das selbe steht wie in der Meldung, weshalb es bereits dementsprechend vor ausgefüllt wird.
- Nachrichtentext
Hier wird die gesamte Nachricht, mit allen Informationen, eingetragen.

5 Methodik

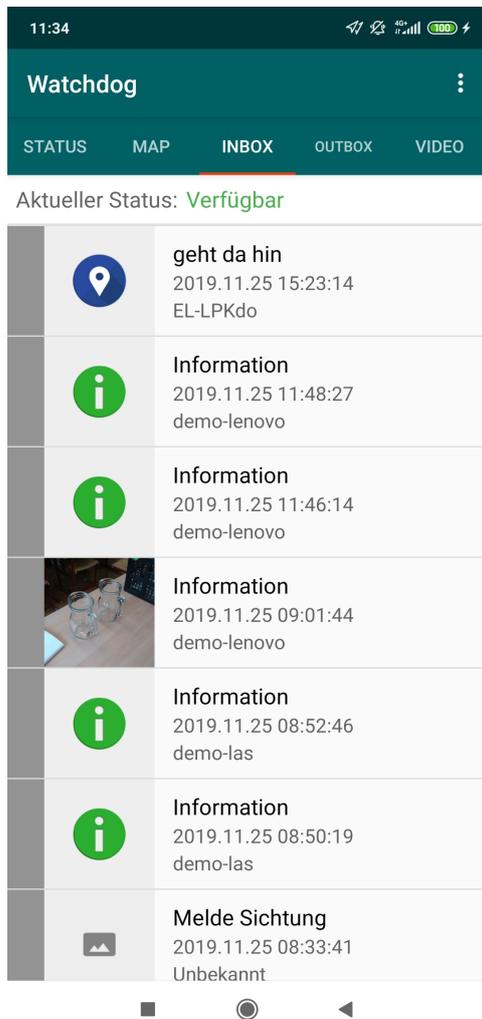


Abbildung 5.13: Android-Applikation
Inbox

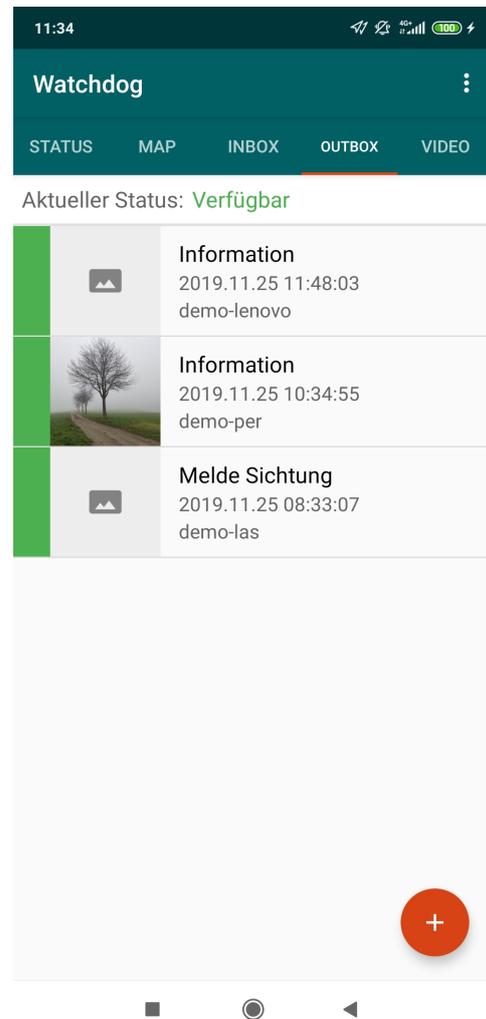


Abbildung 5.14: Android-Applikation
Outbox

- Anhänge
Zusätzlich zum Text kann optional auch noch eine Audioaufnahme und/oder ein Foto mit versendet werden. Das kann oft sehr nützlich sein, damit der Empfänger die Situation besser einschätzen kann.

Video-Fragment

In diesem Fragment kann der Videostream der Sensormasten angesehen werden. Auf dem Sensormasten ist eine PTZ-Kamera verbaut. PTZ steht wie bereits erwähnt für Pan, Tilt, Zoom. Das heißt, dass die Kamera in alle Richtungen gedreht sowie das Bild vergrößert werden kann. Dadurch ist es möglich, dass auch die mobilen Einheiten vor Ort sich ein Bild vom Blickfeld des Masten aus machen können um Ereignisse zu erkennen die von der Position am Boden vielleicht nicht sichtbar sind. Falls man etwas genauer betrachten will, oder es sich aus dem Blickfeld der Kamera bewegt, ist es von der App aus möglich die Kamera mitzubewegen oder auch näher ran zu zoomen. Um zu verhindern, dass mehrere Nutzer gleichzeitig die Kamera ansteuern, muss man zuerst eine Anfrage an die Einsatzzentrale senden, damit diese dann die Steuerung für eine gewisse Zeit freigibt.

Menü

Das Menü kann von der MainActivity aus immer geöffnet werden. Im Menü hat man folgenden Möglichkeiten:

- Einstellungen
Grundeinstellungen vornehmen, wie die verschiedenen Sendeintervalle oder auch die versendete Bildgröße
- Debug-Menü
Hier werden zusätzliche Informationen angezeigt die für das Debuggen nützlich sein können. Außerdem kann man hier die Adresse vom Server einstellen. Diese Option ist nur für Entwickler gedacht und sollte für den Endnutzer nicht relevant sein.
- Favoriten

5 Methodik

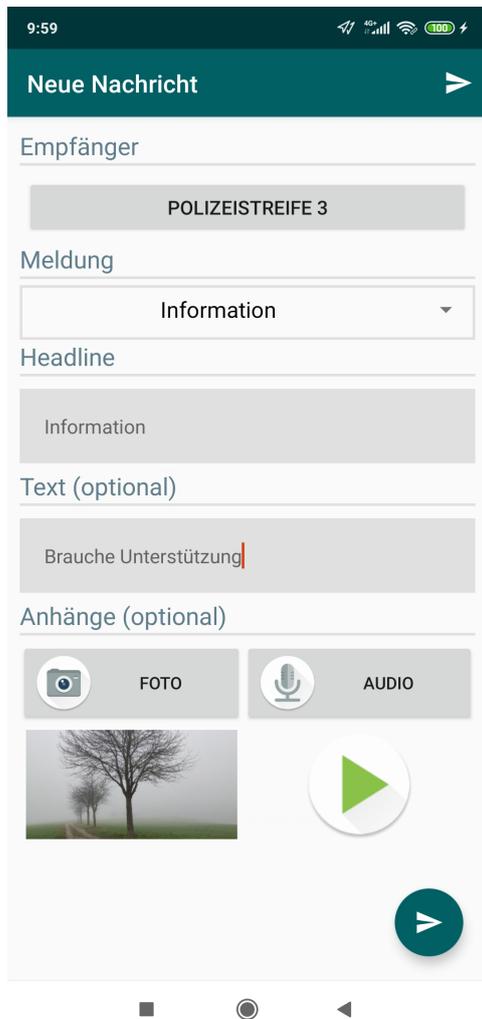


Abbildung 5.15: Android-Applikation
Neue Nachricht



Abbildung 5.16: Android-Applikation
Video-Fragment

Um bei der Empfängerauswahl nicht lange nach dem Kontakt suchen zu müssen, kann man hier seine persönlichen Favoriten festlegen.

- Registrieren
In diesem Fenster werden alle Daten, mit denen das Gerät registriert wurde, angezeigt. Man kann auch noch Änderungen an allen Daten (bis auf den Benutzernamen) vornehmen.
- Beenden
Um die Applikation zu beenden, so dass sie nicht weiter im Hintergrund aktiv ist und keine Positionsdaten an den Server gesendet werden.

Allgemein

Für die Kommunikation zwischen der Android App und dem Server wird die Bibliothek OkHttp¹⁰ mit der Version 3.10.0 verwendet. Diese Bibliothek ist sehr hilfreich um HTTP/HTTPS-Anfragen zu versenden und die Antworten zu empfangen.

Alle Daten die vom Server empfangen werden, werden auch lokal auf dem Smartphone gespeichert. Dafür werden eine SQLite-Datenbank sowie die SharedPreferences verwendet. Wenn man in Android extern auf statische Variablen zugreifen will, kann es aufgrund des Activity-Lebenszyklus passieren, dass dort nicht die erwarteten Daten gespeichert sind. Aus diesem Grund muss man eine andere Möglichkeit finden, um auf die Daten zugreifen und sie laufend aktualisieren zu können.

Ich habe mich hier für die Funktion `onSharedPreferenceChanged`¹¹ entschieden. Dabei setzte ich die Activities die immer die aktuellsten Daten anzeigen sollen auf einen `OnSharedPreferenceChangeListener`, wodurch sie informiert werden, sobald sich die Daten geändert haben. Diese Funktion wird auch aufgerufen, wenn sich die Activity im Hintergrund befindet und kann so die Daten immer aktuell halten.

¹⁰<https://square.github.io/okhttp/> (Zuletzt aufgerufen am: 24.01.2020)

¹¹<https://developer.android.com/reference/android/content/SharedPreferences.OnSharedPreferenceChangeListener> (Zuletzt aufgerufen am: 24.01.2020)

5 Methodik

Damit man die App nachträglich auch noch Debuggen kann, wird die Bibliothek `logback-android`¹² mit der Version 1.1.1-6 verwendet mit der es möglich ist einfach Log-Files zu erstellen.

Eine weitere Bibliothek, die verwendet wird, ist `Picasso`¹³ mit der Version 2.5.2. Damit ist es möglich die Bilder speicherplatzeffizienter und im Hintergrund zu laden, ohne dass das Programm dabei blockiert wird.

Um nicht ständig abfragen zu müssen ob es neue Nachrichten gibt, wird die Bibliothek `Firebase Cloud Messaging`¹⁴ mit der Version 17.4.0 verwendet. Bei diesem System wird man benachrichtigt, falls man eine neue Nachricht bekommen hat.

¹²<https://tony19.github.io/logback-android/> (Zuletzt aufgerufen am: 24.01.2020)

¹³<https://square.github.io/picasso/> (Zuletzt aufgerufen am: 24.01.2020)

¹⁴<https://firebase.google.com/docs/cloud-messaging> (Zuletzt aufgerufen am: 24.01.2020)

6 Evaluierung

6.1 User-Test

Das gesamte System wurde bei einer Demonstration in Thal bei Graz getestet. Dabei konnten noch mehrere Probleme bezüglich Nutzerfreundlichkeit festgestellt werden. Ein Problem war zum Beispiel die Nutzerverwaltung. Zu diesem Zeitpunkt war es nur möglich sich einmalig über die Android Applikation zu registrieren und dann die Daten auf dem Gerät zu ändern. Mit diesem Account konnte man sich dann auch in der Weboberfläche anmelden, jedoch war es nicht möglich sich mit dem Account auf einem weiteren Android Gerät anzumelden.

Ein Grund für diese Umsetzung war der FCM Token. Dieser Token ist dafür zuständig, damit das Android Gerät Nachrichten empfangen und eindeutig zugeordnet werden kann. Dieser Token wird standardmäßig aber nur einmalig bei der Installation gesetzt und kann nachträglich nicht auf einen bestimmten Token, zum Beispiel einen bereits registrierten Token, gesetzt werden. Deshalb waren ebenfalls der User und das Gerät direkt verknüpft, da es zu keinen Situationen kommen konnte, an denen ein User zwei aktive Geräte hatte.

Ein weiteres Problem war die unübersichtliche Nachrichtenansicht. Da die Nachrichten in der Android App, ähnlich wie bei Email, mit Posteingang und Postausgang aufgeteilt waren, konnte man keinen Nachrichtenverlauf erkennen. Aus diesem Grund war es schwer den Nachrichtenverkehr zwischen zwei Personen nachträglich nachvollziehen zu können. In der Weboberfläche werden die Nachrichten ebenfalls nur in einer Tabelle dargestellt, weshalb es auch hier nicht so einfach ist den Verlauf nachzuverfolgen.

6.2 Änderungen durch User-Test

Aufgrund der aufgetretenen Problem wurde eine neue bessere Benutzerverwaltung für die Android Applikation implementiert. Jetzt ist es möglich sich auf einem Gerät an- und auch wieder abzumelden, sowie mehrere Geräte mit dem selben User zu verknüpfen. Dafür wurde der User (identifiziert sich mit Benutzername und Passwort) vom Gerät getrennt, damit man sich auch ohne Gerät registrieren beziehungsweise anmelden kann. Das Gerät wird erst in einem weiteren Schritt registriert und wird dann mit dem angemeldeten User verknüpft. Außerdem besteht auch die Möglichkeit das Passwort, so wie alle Daten vom Gerät nachträglich zu ändern. In folgender Abbildung 6.1 sieht man die neue verbesserte Benutzerverwaltung.

Neben der Benutzerverwaltung wurde auch der Chat komplett überarbeitet. Aus der vorher Email-orientierten Ansicht mit Posteingang und Postgang wurde ein richtiger Chat, der jetzt ähnlich aufgebaut ist wie WhatsApp¹. Es gibt ein Chat-Fragment in dem eine Liste mit allen Kontakte mit denen man geschrieben hat dargestellt wird. In dieser Ansicht sieht man bereits einige wichtige Informationen über die letzte Nachricht mit dem jeweiligen Kontakt. Neben dem Titel und dem Zeitstempel wird auch der letzte Status der Nachricht angezeigt, sowie die Anzahl an ungelesenen Nachrichten im Chat, wie in Abbildung 6.2 zu sehen ist.

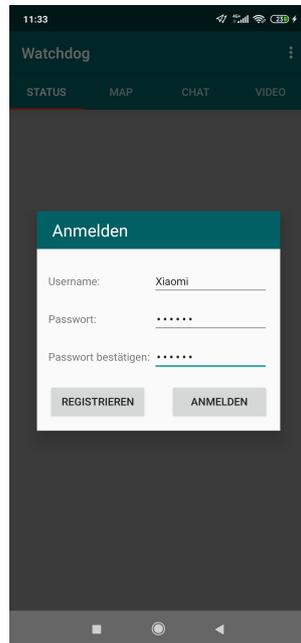
Wählt man dann einen bestimmten Kontakt aus, sieht man alle Nachrichten mit dieser Person in zeitlicher Reihenfolge. Neben der kompletten Nachricht werden hier auch Bilder sowie „Gehe-zu“-Befehle optisch dargestellt. Um schnell kurze Nachrichten verfassen zu können, besteht nun auch die Möglichkeit eine einfache Textnachricht, ohne viel Einstellungen, als Nachricht mit dem Typ Information zu versenden. Der neue Chat wird in Abbildung 6.3 dargestellt.

Aus den vorangegangenen Änderungen hat sich auch ein neues Klassendiagramm ergeben, wie man in folgender Abbildung 6.4 sehen kann.

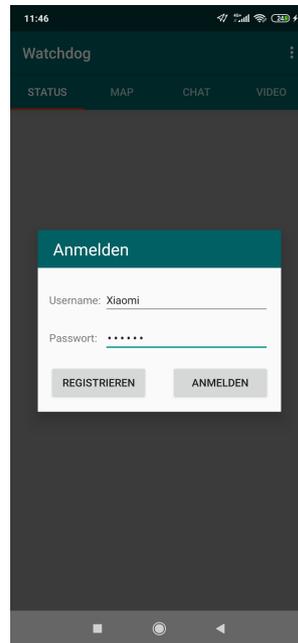
Bei der Weboberfläche traten die selben Probleme, bezüglich Benutzerverwaltung und der Nachrichtenübersicht auf, jedoch war es aus zeitlichen

¹<https://www.whatsapp.com> (Zuletzt aufgerufen am: 04.02.2020)

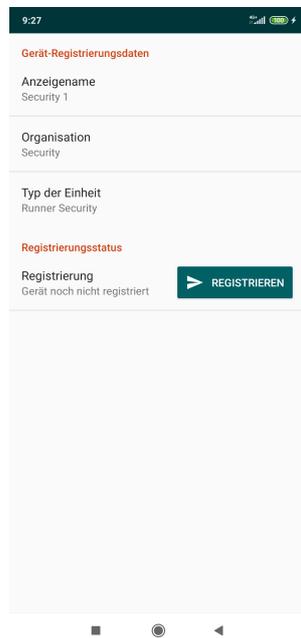
6 Evaluierung



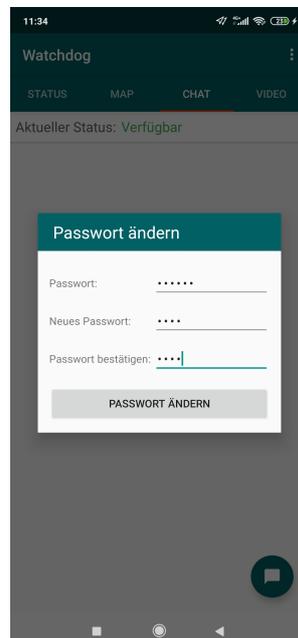
(a) Registrieren



(b) Anmelden



(c) Gerät registrieren



(d) Passwort ändern

Abbildung 6.1: Android-Applikation - Neue Benutzerverwaltung

6 Evaluierung

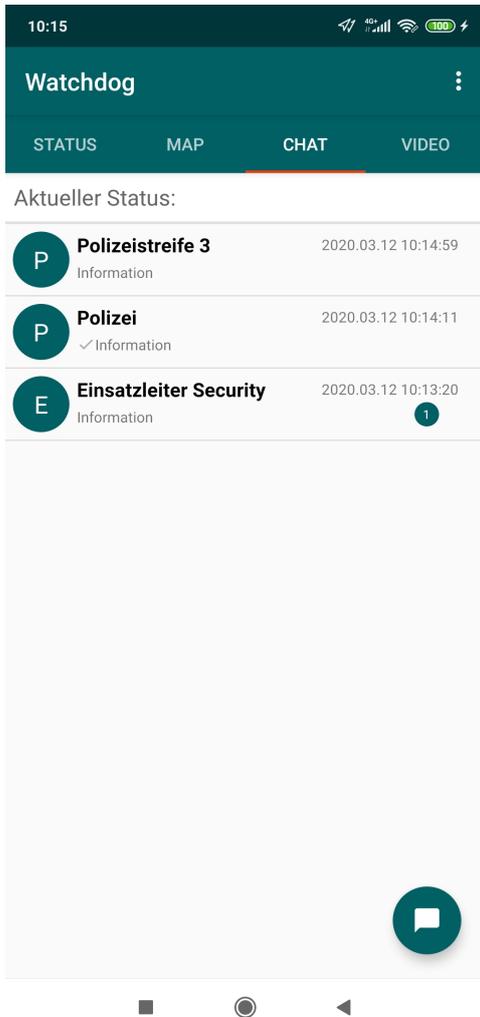


Abbildung 6.2: Android-Applikation -
Neue Chatübersicht

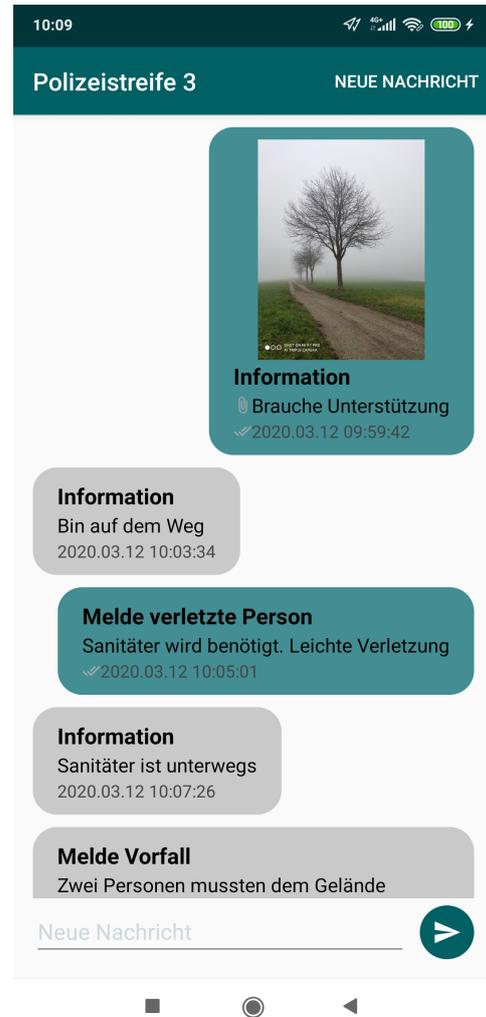


Abbildung 6.3: Android-Applikation -
Neuer Chat

6 Evaluierung

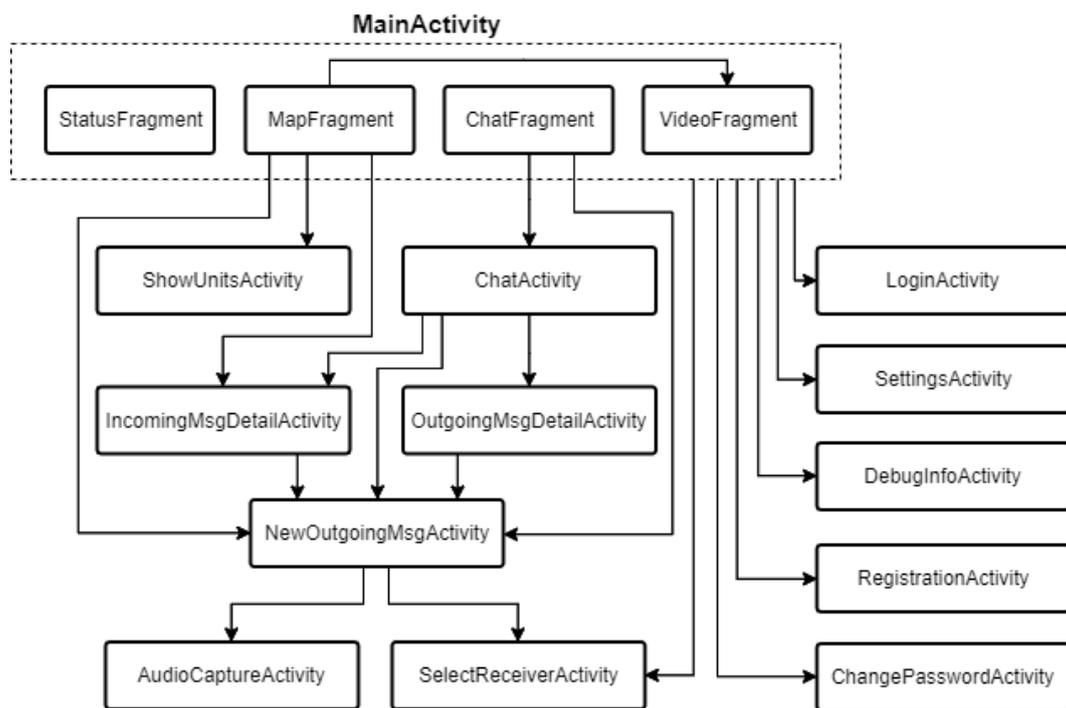


Abbildung 6.4: Android-Applikation - Neues Flussdiagramm aller Activities

6 Evaluierung

Gründen nicht mehr möglich diese Änderungen ebenfalls vorzunehmen. Allerdings konnte eine andere Verbesserung im Bezug auf die Sicherheit vorgenommen werden. Bisher war die Weboberfläche nur über HTTP erreichbar, doch durch die Verwendung eines gültigen Zertifikats, das wir ebenfalls bei der Serverkomponente verwenden, konnte die Weboberfläche auch auf HTTPS umgestellt werden, um so eine sichere Verbindung zu gewährleisten.

7 Schlussfolgerung

Ziel dieser Arbeit war die Entwicklung einer mobilen Managementlösung für ein umfassendes echtzeitnahes Lagebild zur optimierten Koordination von Sicherheitskräften. Für die effiziente Verwendung des Systems wurde eine eigene Weboberfläche für die Einsatzzentrale entwickelt. Mithilfe dieser Oberfläche ist es möglich sich einen genauen Überblick über alle Positionen und Informationen aller aktiven beteiligten Einsatzkräften zu verschaffen und diese zu koordinieren. Es können Nachrichten beziehungsweise Befehle an die Einheiten versendet, sowie empfangen werden. Auf der interaktiven Karte werden neben den Einheiten auch Alarmer von Sensormasten visuell dargestellt.

Zusätzlich zur Weboberfläche wurde auch eine mobile Applikation für Android entwickelt, die von allen mobilen Einheiten eingesetzt werden kann. In dieser Applikation werden ebenfalls die Positionen aller Einheiten auf einer Karte dargestellt, um sich auch im Einsatz einen Überblick über die aktuelle Lage verschaffen zu können. Außerdem wird das versenden von Nachrichten inklusive Mediendateien unterstützt. Durch die Möglichkeit auch Bilder und Audioaufnahmen versenden zu können, kann die Situation von der Einsatzzentrale besser beurteilt werden ohne vor Ort sein zu müssen.

Aufgrund dessen, dass es immer mehr Großveranstaltungen gibt, bei denen mehrere verschiedene Organisationen beteiligt sind, ist dieses System genau dafür entwickelt worden. Diese Managementlösung, kann sowohl von privaten Sicherheitsdiensten, als auch von staatlichen Organisationen wie Polizei, Rettung, oder auch Feuerwehr, verwendet werden um so einen effizienten und optimalen Informationsaustausch zwischen allen Beteiligten gewährleisten zu können.

7.1 Zukünftige Arbeit

Wie bereits erwähnt, gibt es für die Weboberfläche noch ein paar Verbesserungsmöglichkeiten. Darunter wäre eine verbesserte Benutzerverwaltung, die ähnlich aufgebaut ist wie bei der Android Applikation und eine übersichtlichere Nachrichtenansicht um den Nachrichtenverkehr zwischen Personen besser nachvollziehen zu können. Ein weiterer Punkt wäre, die Nachrichten in der Weboberfläche ebenfalls über FCM zu empfangen, um nicht weiter in regelmäßigen Abständen abfragen zu müssen ob es neue Daten gibt.

Eine Möglichkeit dies umzusetzen, besteht darin die Firebase Realtime Database¹ zu verwenden. Bei dieser Datenbank werden alle verbundenen Geräte immer auf den aktuellen Stand gebracht und erhalten so ohne Abfragen alle aktuellen Daten. Ein weiterer Vorteil ist, dass sie in allen Komponenten, also im Server, in der Android Applikation und in der Weboberfläche, eingesetzt werden kann. Somit sind immer alle auf dem aktuellen Stand und müssen keine Daten mehr abfragen.

¹<https://firebase.google.com/docs/database/> (Zuletzt aufgerufen am: 06.02.2020)

Literatur

- [FER10] Sandra Frings, Wolf Engelbach und David Remondes. »Informations- und Kooperationsportal als Unterstützungssystem der Behörden und Organisationen mit Sicherheitsaufgaben für Großveranstaltungen.« In: Jan. 2010, S. 819–824 (siehe S. 12).
- [GD10] Thomas Greiner und Anton Donner. »Data Management in Mass Casualty Incidents: The e-Triage Project«. In: Sep. 2010, S. 200–206. ISBN: 978-3-88579-270-3 (siehe S. 13).
- [Hei09] Rudi Heimann. »Entscheidungsfindung in polizeilichen Einsatzlagen - Softwareunterstütztes Informations- und Kommunikationsmanagement«. In: *GI Jahrestagung*. 2009 (siehe S. 14).
- [HOT13] Anna Hristoskova, Femke Ongenae und Filip Turck. »Semantic reasoning for intelligent emergency response applications«. In: Juli 2013, S. 547–554. DOI: 10.1109/INDIN.2013.6622943 (siehe S. 15).
- [Köf+18a] Armin Köfler u. a. »Decision making support in security forces command centers at open air music festivals: Localization of resources and sharing information«. In: *it - Information Technology* 60 (Aug. 2018), S. 229–238. DOI: 10.1515/itit-2018-0005 (siehe S. 11, 12, 14).
- [Köf+18b] Armin Köfler u. a. »Echtzeitnahe Entscheidungsunterstützung im Sicherheitsmanagement von Großveranstaltungen«. In: *AGIT - Journal für Angewandte Geoinformatik 4-2018* (Juli 2018), S. 132–137. DOI: 10.14627/537647016 (siehe S. 13, 14).
- [May+11] Robin E. Mays u. a. »Aligning border security workflow and decision making with supporting information and communication systems«. In: *ICIS 2011*. 2011 (siehe S. 10).

Literatur

- [WZ09] Anna Wu und Xiaolong Zhang. »Supporting collaborative sensemaking in map-based emergency management and planning«. In: Mai 2009, S. 395–396. DOI: 10.1145/1531674.1531741 (siehe S. 10).