

Lokalisierung von Dokumenten und Tracking des Workflows mit Hilfe von IoT-Geräten

STUDIENARBEIT

des Studienganges Informatik
an der Dualen Hochschule Baden-Württemberg Karlsruhe

Max Heidinger
2361793

Pascal Riesinger
7586259

| | |
|----------------------|-------------------------|
| Abgabedatum | 18.05.2020 |
| Bearbeitungszeitraum | 01.10.2019 - 18.05.2020 |
| Kurs | TINF17B1 |
| Betreuer | Prof. Dr. Marcus Strand |

Eidesstattliche Erklärung

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

Lokalisierung von Dokumenten und Tracking des Workflows mit Hilfe von IoT-Geräten

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, den 18. März 2020

Max Heidinger, Pascal Riesinger

Inhaltsverzeichnis

| | |
|---|-----------|
| Abkürzungsverzeichnis | IV |
| Abbildungsverzeichnis | V |
| Listings | VI |
| 1 Einführung | 1 |
| 1.1 Idee des Paper-Tracker | 1 |
| 2 Grundlagen | 2 |
| 2.1 Projektumfeld | 2 |
| 2.2 Funktionsweise WLAN | 2 |
| 2.2.1 SSID | 2 |
| 2.2.2 BSSID | 2 |
| 2.2.3 Signalstärke RSSI | 2 |
| 2.3 Beispiel eines Workflows | 2 |
| 3 Analyse | 4 |
| 3.1 Ist-Analyse | 4 |
| 3.2 Soll-Analyse | 4 |
| 3.3 Anforderungsanalyse | 4 |
| 4 Entwurf | 7 |
| 4.1 Systemarchitektur | 7 |
| 4.2 Komponentenarchitektur | 9 |
| 4.2.1 Tracking mit IoT-Hardware | 9 |
| 4.2.2 Backend-Anwendung | 10 |
| 4.2.3 App für Mobilgeräte | 12 |
| 5 Implementierung und Validierung | 13 |
| 5.1 Backend-Server | 13 |
| 5.1.1 Verwendete Technologien | 13 |
| 5.1.2 Architektur | 13 |
| 5.1.3 Analyse-Algorithmus | 13 |
| 5.1.4 REST-Schnittstelle | 13 |
| 5.1.5 COAP-Schnittstelle | 13 |
| 5.2 Firmware | 13 |
| 5.2.1 Verwendete Technologien | 13 |

| | | |
|----------|--|------------|
| 5.3 | App | 13 |
| 5.3.1 | Verwendete Technologien | 13 |
| 5.4 | Bekannte Probleme | 13 |
| 5.4.1 | Ungenauigkeit des Tracking | 13 |
| 5.4.2 | UI-Details in der App | 13 |
| 6 | Validierung | 14 |
| 6.1 | Genauigkeitsmessung des Tracking | 14 |
| 6.1.1 | Beschreibung | 14 |
| 6.1.2 | Durchführung | 14 |
| 6.1.3 | Ergebnisse | 14 |
| 6.2 | Nutzerumfrage zur UX der App | 14 |
| 6.2.1 | Beschreibung | 14 |
| 6.2.2 | Durchführung | 14 |
| 6.2.3 | Ergebnisse | 14 |
| 6.2.4 | Gesamtergebniss der Validierung | 14 |
| 7 | Ausblick und Weiterentwicklungen | 15 |
| 7.1 | Audiovisuelle Signale | 15 |
| 7.2 | Verbessertes Gehäuse | 15 |
| 7.3 | Tracking von Smartphones | 15 |
| 8 | Zusammenfassung | 16 |
| | Glossar | 17 |
| | Literatur | VII |

Abkürzungsverzeichnis

AMQP Advanced Message Queuing Protocol. 9

AP Access Point. 9

CoAP Constrained Application Protocoll. 9, 10

DHBW Duale Hochschule Baden-Württemberg. V, 1–3

HTTP Hypertext Transfer Protocol. 9

IoT Internet of Things. 9

MQTT Message Queuing Telemetry Transport. 9

WLAN Wireless Local Area Network. 9

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 2.1 | Beschaffungs-Genehmigungsverfahren der Duale Hochschule Baden-Württemberg (DHBW) | 3 |
| 4.1 | Architektur des Paper-Tracker | 8 |
| 4.2 | UML-Diagramm für die Backend-Anwendung | 11 |

Listings

1 Einführung

An der DHBW in Karlsruhe basieren, wie auch noch in vielen anderen Organisationen, viele Prozesse auf Papier. Dies bedeutet, dass für zum Beispiel die Genehmigung für einen Einkauf, ein oder mehrere Blätter Papier von unterschiedlichen Personen unterzeichnet werden müssen. Dafür wird das Blatt durch die einzelnen Büros getragen, bis letztendlich der Antragssteller es wieder bekommt.

Auch für simple Prozesse kann diese einige Zeit in Anspruch nehmen. Während dieser ganzen Zeit hat der Antragsteller leider keine Einsicht, bei welcher Person sich die Papiere befinden. Damit kann er auch nicht versuchen den Prozess zu beschleunigen, ohne Schritt für Schritt bei allen Personen des Prozesses nachzufragen. Zusätzlich kann er im Nachhinein nicht herausfinden, weshalb der Prozess länger als erwartet gebraucht hat.

Um dieses Problem anzugehen, wurde der „Paper-Tracker“ entwickelt. Dieser soll eine Transparenz in alle auf Papier basierten Prozesse bringen und zusätzlich Daten liefern, um die Prozesse selbst zu optimieren.

1.1 Idee des Paper-Tracker

Die Idee des Paper-Trackers basiert darauf, einen möglichst kleinen Tracker basierend auf einem Mikrocontroller an den Papieren eines Prozesses zu befestigen. Dieser soll eine Lokalisierung der Papiere ermöglichen. Auf diese Lokalisierung soll der Antragsteller eines Prozesses über eine App Zugriff bekommen. Weiter soll auch in der App der komplette Prozess oder Workflow selbst verfolgt werden. Somit kann zum Beispiel auch eine Notifizierung zuständiger Personen für einen Schritt in einem Workflow erfolgen.

2 Grundlagen

2.1 Projektumfeld

Das Projekt wird im Umfeld der DHBW Karlsruhe durchgeführt.

2.2 Funktionsweise WLAN

2.2.1 SSID

2.2.2 BSSID

2.2.3 Signalstärke RSSI

2.3 Beispiel eines Workflows

In folgender Abbildung 2.1 ist ein Beispiel für einen Workflow an der DHBW Karlsruhe dargestellt. Es ist das „Beschaffungs-Genehmigungsverfahren“, das benötigt wird, um eine Bestellung für die DHBW durchzuführen.

3 Analyse

In den folgenden Abschnitten werden der Ist-Zustand, aus welchem sich die Problemstellung ergibt, der Soll-Zustand, sowie die gestellten Anforderungen beschrieben.

3.1 Ist-Analyse

Es besteht das Problem, dass der aktuelle Status eines auf Papier basierten Prozesses nicht nachvollzogen werden kann. Für den Antragsteller eines Prozesses ist es nicht möglich herauszufinden, bei welcher Person oder in welchem Raum sich ein Dokument zu einem bestimmten Zeitpunkt befindet. Zusätzlich kann, sobald der Prozess beendet ist, nicht herausgefunden werden, an welcher Stelle im Prozess zum Beispiel eine besonders hohe Wartezeit stattgefunden hat.

3.2 Soll-Analyse

Ziel der Entwicklung und damit dieser Studienarbeit ist das Erarbeiten einer Lösung, mit welcher die aktuelle Position eines Dokuments und der aktuelle Fortschritt des Prozesses sichtbar gemacht werden kann. Es soll ein möglichst unauffälliges Gerät sein und eine intuitive Bedienung ermöglichen. Über aktuelle und abgeschlossene Prozesse sollen Daten analysiert werden können, um Möglichkeiten zur Prozessoptimierungen erkennen zu können.

3.3 Anforderungsanalyse

Im Folgenden werden die gestellten Anforderungen in funktionale und nichtfunktionale Anforderungen unterteilt. Dabei beschreiben funktionale Anforderungen konkrete Funktionen, die die entwickelte Lösung bieten muss, nichtfunktionale Anforderungen hingegen zeigen Rahmenbedingungen, sowie Anforderungen an die technische Umsetzung auf.

Evtl. noch eine Anforderungsanalyse mit anderen Leuten durchführen

Desweiteren werden den Anforderungen Prioritäten zugewiesen, wobei ein Wert von 0 die höchste Priorität, ein Wert von 3 die niedrigste Priorität beschreibt, sodass die Lösung ohne Erfüllung der Anforderungen mit Priorität 0 nicht einsetzbar ist und Priorität 3 optionale Funktionalitäten beschreibt, die nicht zum Betrieb notwendig sind.

| Name | Beschreibung | Priorität |
|------|---|-----------|
| F-1 | Der Standort von Dokumenten kann raumgenau verfolgt werden | 0 |
| F-2 | Der aktuelle Status des mit dem Dokument verbundenen Workflows kann nachvollzogen werden | 0 |
| F-3 | Neue Workflows können vom Nutzer definiert werden | 2 |
| F-4 | Neue Tracker können dem System vom Nutzer hinzugefügt werden | 1 |
| F-5 | Der Raumplan kann vom Nutzer eingelernt werden | 3 |
| F-6 | Nutzer werden vom System benachrichtigt, wenn die Akkuladung eines Trackers unter einen gegebenen Prozentsatz fällt oder die Batterie ausgetauscht werden muss | 2 |
| F-7 | Einzelne Schritte von Workflows können optional sein und übersprungen werden | 1 |
| F-8 | Für einen Schritt im Workflow können mehrere Orte hinterlegt werden | 1 |
| NF-1 | Die Laufzeit der Tracker im Batteriebetrieb beläuft sich auf mindestens eine Woche | 2 |
| NF-2 | Die Tracker sollen möglichst unscheinbar und daher klein und leicht sein | 1 |
| NF-3 | Die Daten können in einer Applikation für Mobiltelefone abgefragt werden. Dabei ist besonders die Verwendbarkeit unter dem Betriebssystem Android wichtig, die Unterstützung von iOS ist optional | 0 |
| NF-4 | Die Anwendung muss stabil laufen und sollte sich nicht unerwartet beenden | 1 |
| NF-5 | Die Lösung sollte skalierbar hinsichtlich der Anzahl der Tracker und der Größe des Tracking-Bereiches sein | 2 |
| NF-6 | Zur optimalen Nachvollziehbarkeit von Workflows sollten deren auch nach Abschluss nicht gelöscht werden | 2 |
| NF-7 | Die im System gespeicherten Daten sollen in einem maschinenlesbaren Format exportiert werden können, um Auswertungen über die Effizienz der Workflows durchführen zu können | 2 |

Tabelle 3.1: Funktionale und nichtfunktionale Anforderungen

4 Entwurf

4.1 Systemarchitektur

In Abbildung 4.1 ist eine grobe Übersicht über die verschiedenen Komponenten des Projektes gegeben. Es besteht aus drei Komponenten:

- Hardware-Tracker
- Backend-Anwendung
- App

Der Hardware-Tracker ist die physische Komponente, die direkt an ein zu trackendes Papier angeheftet wird. Er wird regelmäßig Daten erfassen und diese zur Lokalisierung an die Backend-Anwendung senden.

Die Backend-Anwendung dient als zentrales Element der Architektur. Sie bekommt Daten des Trackers zugeschickt, wertet diese aus und speichert sie. Auch das Management der Räume und Workflows wird von der Anwendung übernommen.

Nutzer verwenden zur Kommunikation mit dem Backend eine Smartphone-App. Sie kommuniziert ebenfalls mit der Backend-Anwendung und fragt von dieser Informationen an oder löst Aktionen aus.

Die Entwürfe der drei Komponenten werden in den nachfolgenden Abschnitten im Detail erläutert.

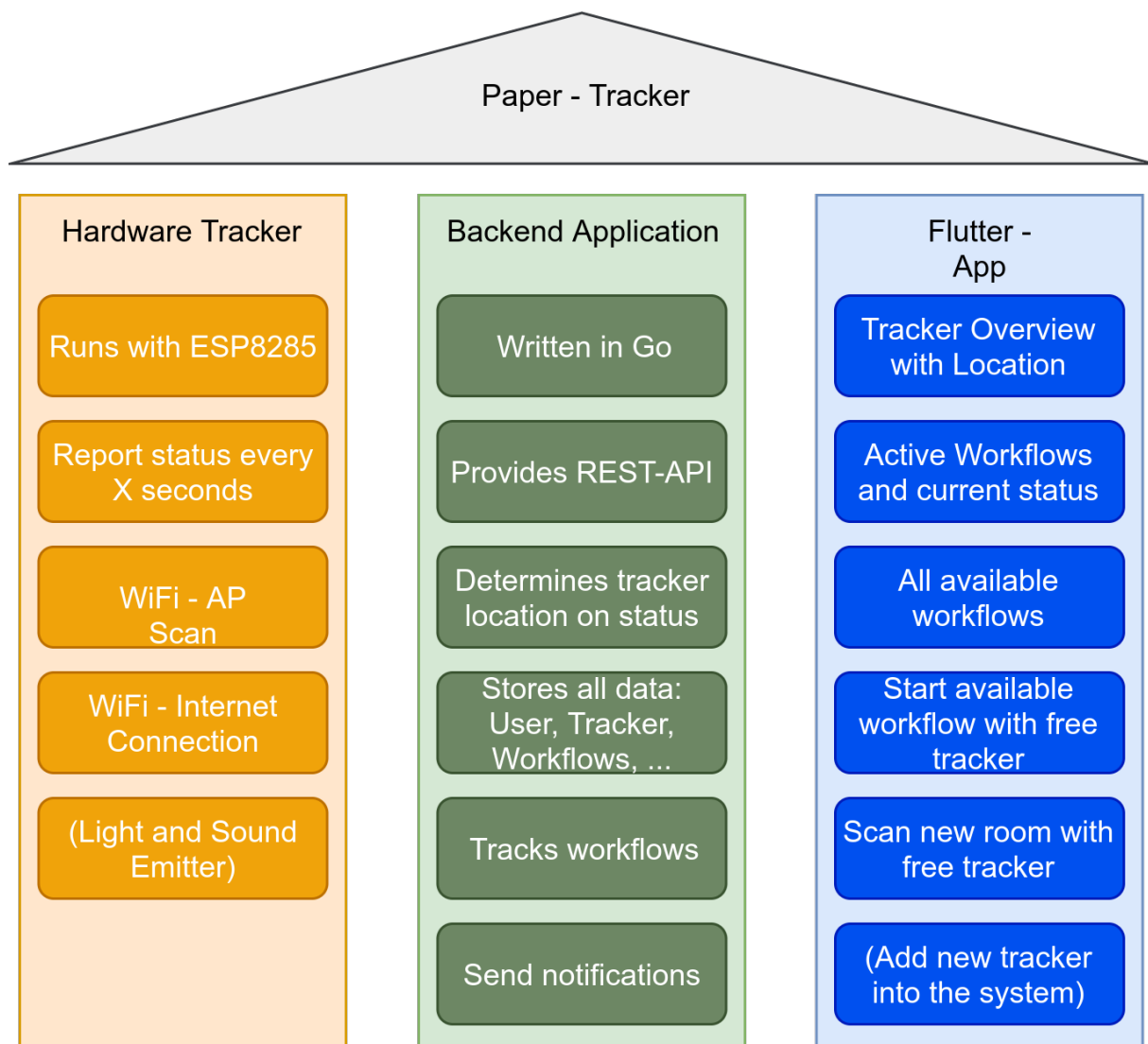


Abbildung 4.1: Architektur des Paper-Tracker

4.2 Komponentenarchitektur

In den folgenden Abschnitten wird detaillierter auf die einzelnen Komponenten des in Abschnitt 4.1 beschriebenen Systems eingegangen. Dabei werden die Komponenten in derselben Reihenfolge beschrieben, in welcher die Daten im Gesamtsystem fließen.

4.2.1 Tracking mit IoT-Hardware

Zur Erfassung der Standortdaten der Dokumente werden, wie in Abschnitt 3.2 beschrieben, Hardware-Tracker eingesetzt. Diese verwenden jedoch keine absolute Positionierungstechnologie wie beispielsweise GPS, da das hierfür benötigte Satellitensignal in Gebäuden zu schwach, und somit die Positionierung fehlschlagen kann.

Für die Lokalisierung von Geräten innerhalb eines Gebäudes gibt es daher mehrere Ansätze. Eine Möglichkeit ist es, ein Mesh-Netzwerk aus den Tracker-Geräten aufzubauen. Ist die Position einiger weniger Knoten des Meshes bekannt, kann durch die Signalstärke zu umliegenden Knoten oder über die sogenannte Time-of-Arrival bestimmt werden, wo sich ein Knoten relativ zu anderen Knoten und damit auch zu den Knoten mit bekannter Position befindet. Mit dieser Methode kann eine Genauigkeit von bis zu einem Meter bei der Lokalisierung erzielt werden. (vgl. [1])

Eine weitere Möglichkeit ist, auf bestehende Infrastruktur zurückzugreifen. Wireless Local Area Network (WLAN) ist heutzutage in praktisch jedem Gebäude verfügbar. In Gebäuden, in welchen es mehrere Access Points (APs) gibt, kann die Position dadurch bestimmt werden, dass der Tracker auswertet, von welchen APs er ein Signal empfängt und wie stark der Empfang zum jeweiligen AP ist. Da die Positionsdaten ohnehin zur Auswertung an das Backend übertragen werden müssen, wofür eine WLAN-Verbindung notwendig ist, wird diese Methode eingesetzt.

Um die Daten über die Signalstärke an das Backend zu übermitteln wird das Protokoll Constrained Application Protocol (CoAP) eingesetzt. Hierbei handelt es sich um ein spezialisiertes Protokoll für die Verwendung in Internet of Things (IoT)-Hardware, welches dafür optimiert wurde möglichst wenig Overhead zu haben. Im IoT-Bereich haben sich einige Protokolle durchsetzen können. Dazu zählen vor allem Hypertext Transfer Protocol (HTTP), Message Queuing Telemetry Transport (MQTT), Advanced Message Queuing

Es wäre schön, Glossar-Referenzen zu kennzeichnen (oft mit Glos, oder Gls)

Quelle für die Genauigkeit von GPS in Gebäuden

Ist das hier eher Analyse?

Glossar: Mesh-Netzwerk

Time-of-Arrival erläutern

Glossar: Overhead

Protocol (AMQP) und CoAP. Da die Tracker Batteriebetrieben sind, ist für diesen Anwendungsfall besonders die Performance des eingesetzten Protokolles wichtig. Je weniger Overhead das Protokoll aufweist, desto weniger Instruktionen müssen vom Prozessor des Trackers durchgeführt werden, was zu einer längeren Laufzeit führt. Werden die Protokolle anhand dieses Aspektes verglichen, erweist sich CoAP als effizientestes Protokoll. (vgl. [2], [3])

Quellen verwenden, die in diesem Paper zitiert werden

4.2.2 Backend-Anwendung

Da die Backend-Anwendung die zentrale Komponente des Systems ist und auch die Daten verwaltet, werden für sie die Daten-Klassen modelliert. Das resultierende UML-Diagramm ist in Abbildung 4.2 abgebildet.

Die zentrale Klasse des Diagramms bildet die „Tracker“-Klasse. Sie modelliert einen Hardware-Tracker. Als Attribut besitzt sie eine Referenz auf einen „Room“.

Ein „Room“ ist ein physikalischer Raum, der von einem Tracker erkannt werden soll. Neben einem Identifizierer und einem „Label“ (Name) für den Raum werden zusätzlich mehrere „BSSIDTrackingData“ gespeichert. Diese beinhalten für einen spezifischen „Access Point“ einige Messdaten zu der gemessenen Signalstärke. Über diese kann über eine Antwort des Trackers mit „ScanResults“ die Position des Tracker bestimmt werden.

Damit der Tracker eine Antwort sendet, wird diese mit Hilfe eines Kommandos angefragt. Diese Kommandos werden jeweils in einer Instanz der „Command“-Klasse abgebildet. Pro Tracker können somit mehrere Kommandos abgespeichert werden. Die Kommandos besitzen einen spezifischen Typ der Enumeration „CommandType“, der die Aktion des Kommandos bestimmt. Auch wird pro Kommando eine „SleepTimeSec“ festgelegt, die bestimmt wie lange der Tracker sich abschalten darf.

Spezielle Antworten der Tracker auf Kommandos und auch des Servers auf Anfragen der App sind in dem Unterpaket „communication“ modelliert. Dies ist zum Beispiel eine Fehler-Antwort und eine Basisklasse für Antworten des Trackers. Diese „TrackerResponse“-Klasse besitzt auch ein Feld für den aktuellen Batteriestand des Trackers, um diesen überwachen zu können.

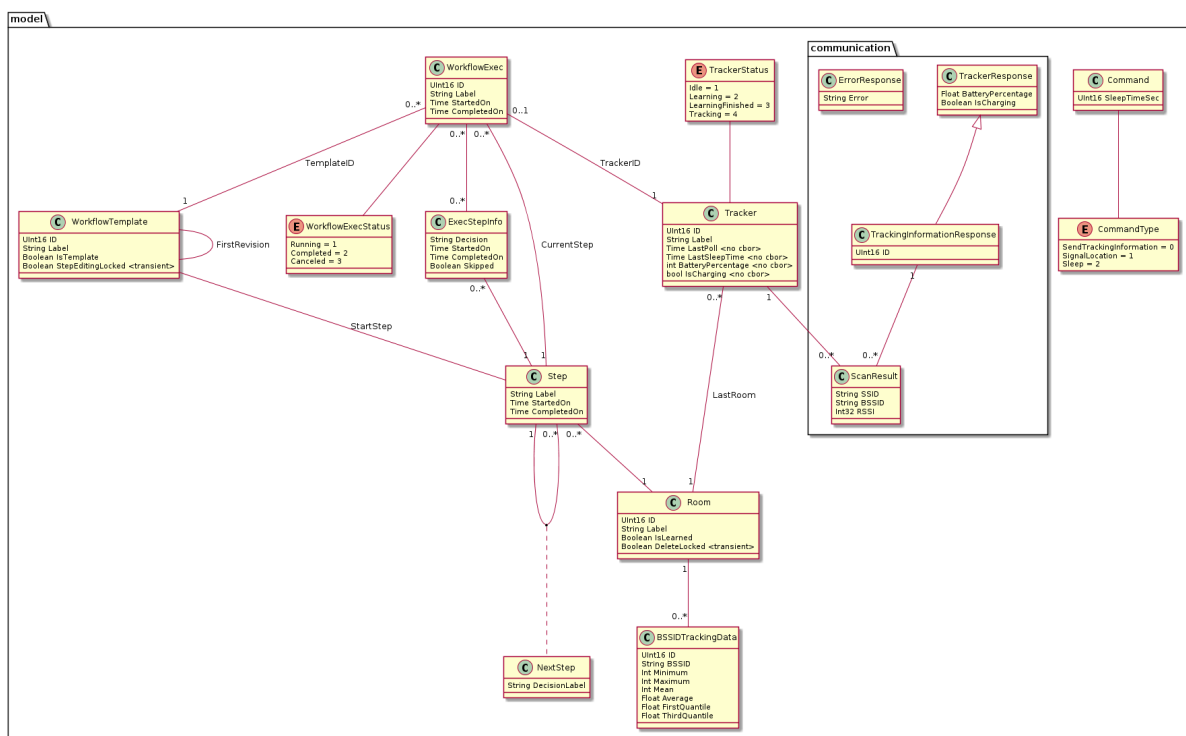


Abbildung 4.2: UML-Diagramm für die Backend-Anwendung

4.2.3 App für Mobilgeräte

5 Implementierung und Validierung

5.1 Backend-Server

5.1.1 Verwendete Technologien

5.1.2 Architektur

5.1.3 Analyse-Algorithmus

5.1.4 REST-Schnittstelle

5.1.5 COAP-Schnittstelle

5.2 Firmware

5.2.1 Verwendete Technologien

5.3 App

5.3.1 Verwendete Technologien

5.4 Bekannte Probleme

5.4.1 Ungenauigkeit des Tracking

5.4.2 UI-Details in der App

6 Validierung

6.1 Genauigkeitsmessung des Tracking

6.1.1 Beschreibung

6.1.2 Durchführung

6.1.3 Ergebnisse

6.2 Nutzerumfrage zur UX der App

6.2.1 Beschreibung

6.2.2 Durchführung

6.2.3 Ergebnisse

6.2.4 Gesamtergebniss der Validierung

7 Ausblick und Weiterentwicklungen

7.1 Audiovisuelle Signale

7.2 Verbessertes Gehäuse

7.3 Tracking von Smartphones

Es ist wünschenswert, dass eine Person, die einen Raum betritt, in welchem ein Dokument auf deren Bearbeitung wartet, darüber informiert wird. Zu diesem Zweck können Smartphones in das Tracking-System mit aufgenommen werden.

8 Zusammenfassung

Glossar

GPS Das Global Positioning System (GPS) ist ein von den Vereinigten Staaten von Amerika entwickeltes globales Navigationssystem, welches im zivilen, kommerziellen, wissenschaftlichen und militärischen Bereich Anwendung findet, um die Position von GPS-Empfängern zu bestimmen. (vgl. [4]). 9

Literatur

- [1] Patwari, N. u. a. „Relative location estimation in wireless sensor networks“. In: *IEEE Transactions on Signal Processing* 51.8 (08/2003), S. 2137–2148 (siehe S. 9).
- [2] Dizdarević, J. u. a. „A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration“. In: *ACM Computing Surveys* 51.6 (01/2019), S. 1–29 (siehe S. 10).
- [3] Naik, N. „Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP“. In: *2017 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 10/2017 (siehe S. 10).
- [4] Department Of Transportation, F. A. A. *Global Positioning System Wide Area Augmentation System (WAAS) Performance Standard*. 10/2008. URL: <https://web.archive.org/web/20170427033332/http://www.gps.gov/technical/ps/2008-WAAS-performance-standard.pdf> (besucht am 30.10.2019) (siehe S. 17).

Todo list

| | |
|---|----|
| <input type="checkbox"/> Evtl. noch eine Anforderungsanalyse mit anderen Leuten durchführen | 4 |
| <input type="checkbox"/> Es wäre schön, Glossar-Referenzen zu kennzeichnen (oft mit Glos, oder Gls) . . | 9 |
| <input type="checkbox"/> Quelle für die Genauigkeit von GPS in Gebäuden | 9 |
| <input type="checkbox"/> Ist das hier eher Analyse? | 9 |
| <input type="checkbox"/> Glossar: Mesh-Netzwerk | 9 |
| <input type="checkbox"/> Time-of-Arrival erläutern | 9 |
| <input type="checkbox"/> Glossar: Overhead | 9 |
| <input type="checkbox"/> Quellen verwenden, die in diesem Paper zitiert werden | 10 |