# Einleitung

## Ziel

Das Ziel dieses Abschnitts dient für eine detaillierte Beschreibung des Anwendungssystem. Es wird den Zweck, die Feature, die Interfaces sowie die Funktionen und die Einschränkungen der Software erklären.

## Umfang

Das System soll ein mobiles Hardwaresystem sowie eine Mobilanwendung beinhalten. Ein Nutzer soll in der Lage sein, auf Bewegung des Hardwaresystems eine Benachrichtigung an sein Smartphone zu erhalten, auf welchem die Mobilanwendung installiert ist. Zudem soll das Hardwaresystem seinen aktuellen Standort übertragen, damit der Nutzer diesen in der Anwendung anzeigen lassen kann.

## Einschränkungen der Funktionalitäten

Da diese Anwendung als eine konzeptionelle Umsetzung der LoRaWAN-Technologie erarbeitet wird, ist der Fokus auf die grundsätzliche Funktionalität der Anzeige von Standortdaten gesetzt. Aufgrund dessen werden Funktionalitäten wie mögliche Anpassungsmöglichkeiten der Anwendung an den Benutzern vorerst nicht beachtet. Darunter fallen unter anderem Änderungsmöglichkeiten des Anwendungsdesigns oder Änderungen von Anmeldungsdaten.

Nicht Beachtung wie Registrierung von HWS im Netzwerk/Datenbank

## Definitionen, Akronyme, Abkürzungen

|  |  |
| --- | --- |
| Begriff | Definition |
| Benutzer | Jemand, der mit der Mobilanwendung interagiert |
| GNSS | Global Navigation Satellite System, Sammelbegriff für verschiedene Arten von Satellitnavigationssystemen wie GPS, GLONASS und Galileo (Quelle?) |
| Hardwaresystem | Ein mobiles System aus einem Mikrocontroller und den dazugehörigen Peripheriegeräten für die Standorterfassung des Mikrocontrollers. |
| http | Hypertext Transfer Protocol, Übertragung von Daten auf der Anwendungs-schicht über ein Rechnernetz (Quelle?) |
| ISM-Band | Industriel, Scientific and Medical Band, lizenzfreie Frequenzbänder für industrielle, wissenschaftliche und medizinische Anwendungen |
| Join- Accept | Antwort in Form einer MAC-Nachricht eines LoRaWAN-Netzwerkservers auf ein akzeptiertes Join-Request eines Endgerätes (LoRaWAN 1.0.3 S.34f) |
| Join- Request | MAC-Nachricht gesendet von einem Endgerät zur Anfrage um den Beitritt eines LoRaWAN-Netzwerkes (LoRaWAN 1.0.3 S.34f) |
| LoRa | Long Range, Modulation mit Spektrumsspreizung, für eine Übertragung mit großer Reichweite und geringem Stromverbrauch. (https://www.semtech.com/lora/what-is-lora 09.11) |
| LoRa Alliance | Offener, gemeinnütziger Verband für die Regelung und Weiterentwicklung vom LoRaWAN-Standard (https://lora-alliance.org/about-lora-alliance/ 09.11) |
| LoRaWAN | Long Range Wide Area Network, LPWA-Netzwerkprotokoll designt für einen drahtlosen Kontakt von Endgeräten. (https://lora-alliance.org/about-lorawan/ 09.11) |
| LPWAN | Low-Power Wide-Area-Network, Klasse von Netzwerkprotokollen zur Verbindung von Niedrigenergiegeräten |
| MAC-Nachricht | Definiert die Struktur des PHYPayloads einer LoRa-Nachricht (LoRaWAN 1.0.3 S.15f) |
| SPI | Serial Peripheral Interface, synchroner serieller Bus, welcher Verbindung von Mikrocontrollern und Peripheriegeräten ermöglicht (Quelle?) |
| SQL | Structured Query Language, deklarative Datenbanksprache für die Definition, Abfrage und Manipulation von Daten (Quelle?) |
| UART | Universal Asynchronous Receiver/Transceiver, definiert Protokoll für Datenaustausch zwischen zwei Geräten (Quelle?) |
| UBX | Protokoll für die Kommunikation mit u-blox 8/u-blox M8 Receiver[5 S.1] |

## Referenzen

[1] Heltec LoRa Node 151, https://heltec.org/project/lora-node-151/, Zugriff 09.11.2021.

[2] LoRa-Alliance: RP002-1.0.3 LoRaWAN Regional Parameter, https://lora-alliance.org/resource\_hub/rp2-1-0-3-lorawan-regional-parameters/, Zugriff 08.11.2021.

[3] Mateksys: GPS Module SAM-M8Q, http://www.mateksys.com/?portfolio=sam-m8q, Zugriff 09.11.2021.

[4] Semtech: Semtech SX1276, https://www.semtech.com/products/wireless-rf/lora-core/sx1276, Zugriff: 09.11.2021.

[5] u-blox: u-blox8/u-blox M8 Receiver description Including protocol specification, https://www.u-blox.com/en/ubx-viewer/view/u-blox8-M8\_ReceiverDescrProtSpec\_UBX-13003221?url=https%3A%2F%2Fwww.u-blox.com%2Fsites%2Fdefault%2Ffiles%2Fproducts%2Fdocuments%2Fu-blox8-M8\_ReceiverDescrProtSpec\_UBX-13003221.pdf, Zugriff 09.11.2021.

## Übersicht

Zuerst wird eine allgemeine Übersicht über das gesamte System gegeben. Hierbei wird geklärt, wie das System aufgebaut ist und welche Grundfunktionen dieses System enthält. Zu dem werden noch die Einschränkungen, Annahmen und Abhängigkeiten vom System vorgeführt. Daraufhin werden alle funktionalen und qualitativen Anforderungen an das System gestellt. Begonnen wird dies mit der detaillierten Beschreibung aller In- und Output des Systems sowie der Software-, Hardware- und Kommunikationsinterfaces. Zudem werden Prototypen der Benutzeroberfläche dargestellt.

# Allgemeine Beschreibung

## Produktperspektive

Das System wird aus vier Bestandteilen bestehen: ein mobiles Hardwaresystem, eine Mobilanwendung, ein Webserver und den Dienstleistungen. Das Hardwaresystem wird genutzt, um auf die Bewegung eines Beschleunigungsmessers die aktuelle Position mittels eines GNSS-Moduls zu bestimmen. Die Mobilanwendung wird für die Darstellung dieser Standorte und deren Informationen auf einer Karte und für die Benachrichtigung des Nutzers genutzt. Um Dienste wie die Benachrichtigung, die Routenplanung oder der Weiterreichung von LoRaWAN-Nachrichten zu ermöglichen, werden außenstehende Dienstleistungen verwendet. Der Webserver wird den Zugriff auf eine Datenbank organisieren, um Funktionalitäten wie die Speicherung von Standorten zu ermöglichen.

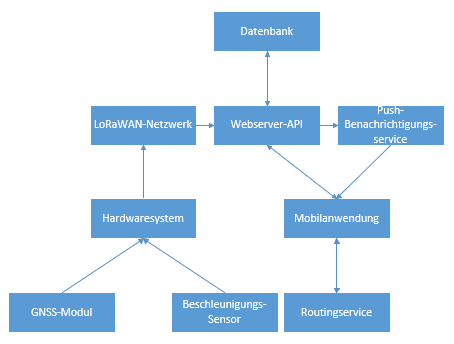


Abbildung Blockdiagramm des Gesamtsystems

## Produktfunktionen

Damit die Bewegung und die Standortdaten vom Hardwaresystem erfasst werden können, muss dieses mit jeweils einem Beschleunigungsmodul und einem GNSS-Modul kommunizieren.   
Um die Standortdaten erreichbar zu machen, muss sich das Hardwaresystem in ein LoRaWAN-Netzwerk einloggen, damit die erfassten Daten an dieses übertragen werden können.

Damit die Daten nun für die Verwendung innerhalb der Anwendung gespeichert werden können, sendet das LoRaWAN-Netzwerk diese an den Webserver, welches die Daten in einer Datenbank speichert.

Der Webserver wird für die Kommunikation mit einer Datenbank ein Interface anbieten, welches unter anderem Ermöglichen wird, die erhaltenen Standortdaten zu speichern. Darüber hinaus wird der Webserver mit Hilfe eines Benachrichtigungsdienst die Mobilanwendung über den Erhalt eines Standortes informieren.

Die Mobilanwendung wird mit dem Webserver kommunizieren, um unter anderem die Standorte eines Hardwaresystems zu ermitteln. Diese Standorte können nun in einer Kartenansicht markiert werden. Um diese Karte anzeigen zu können, wird die Anwendung eine Kartenansicht-Software einbauen, welche die Karten an den erhaltenen Standorten bereitstellt und diese detailliert anzeigt. Für eine Routenplanung zwischen Standorten wird die Anwendung mit einem Routingservice kommunizieren.

## Einschränkungen

Das Hardwaresystem ist durch das Vorhandensein eines Auslastungsgrades für die Verwendung des öffentlich zugänglichen ISM-Bandes eingeschränkt. Der Auslastungsgrad ist pro Region im Regionalen Parameter Dokument der LoRa-Alliance definiert[2] und beträgt für das ISM-Band in Deutschland unter einem Prozent. Das heißt, wenn die Übertragung einer Nachricht eine Sekunde dauert, dann muss das sendende Gerät mindestens 99 Sekunden warten, bevor es wieder senden darf, da sonst fehlende Funktionen auftreten können.

Die Internetverbindung ist zudem auch eine Einschränkung, da die Mobilanwendung Daten aus der Datenbank über dem Webserver und somit über das Internet bezieht, ist eine vorhandene Internetverbindung notwendig.

## Annahmen und Abhängigkeiten

Eine Annahme ist, dass die Mobilanwendung immer auf einem Smartphone mit ausreichender Leistung betrieben wird. Falls das Smartphone mal nicht genügend Hardwareressourcen zur Verfügung stellt, zum Beispiel wenn der Benutzer diese bereits für andere Anwendung verwendet, dann können Szenarios auftreten, in welchem die Mobilanwendung nicht so wie geplant oder überhaupt funktioniert.

Eine weitere Annahme ist, dass die LoRaWAN-Abdeckung vom verwendeten LoRaWAN-Netzwerk ausreichend ausgebaut ist. Falls das Hardwaresystem eine LoRa-Nachricht in einem Gebiet sendet, welches nicht abgedeckt ist, wird diese Nachricht nicht im LoRaWAN-Netzwerk ankommen und wird verfallen.

# Spezifische Anforderungen

## Externe Interface-Anforderungen

### User Interface

Falls der Benutzer beim Öffnen der Mobilanwendung noch nicht angemeldet ist, dann soll der Benutzer das Anmeldebildschirm sehen, siehe Abbildung 2. Falls der Benutzer noch nicht registriert ist, dann soll er dazu die Möglichkeit im Anmeldefenster bekommen.

Falls der Benutzer beim Öffnen der Mobilanwendung angemeldet ist oder sich erfolgreich angemeldet hat, soll der Benutzer das Hauptmenü auffinden, siehe Abbildung 3. Hier soll der Benutzer die Möglichkeit besitzen, eine Standortsuche zu betreiben, seine registrierten Hardwaresysteme oder sein Profil einzusehen.

In Abbildung 4 ist die Eingabemaske der Standortsuche angegeben. In dieser soll der Benutzer Parametern für die Standortsuche nach bestimmten Hardwaresystem durchführen können. Zu diesen Parametern soll die Auswahl eines Hardwaresystems und die Eingabe von Daten-, Breiten- und Längengradintervallen gehören.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Abbildung Anmeldebildschirm | Abbildung Hauptmenü | Abbildung Standortsuche |

Wenn ein Benutzer nach Standorten sucht, dann soll das Ergebnis dieser Suche in einer Liste wiedergegeben werden, siehe Abbildung 5. Ein Header mit Sortieroptionen soll dem Benutzer erlauben, die Ergebnisliste nach der Zeit, dem Breiten- oder dem Längengrad zu sortieren. Jeder Listeneintrag soll Informationen über den Hardwarenamen, den Standort in Graden, die Erfassungsuhrzeit und das Erfassungsdatum beinhalten.

In der Kartenansicht soll jeder Standort eine Markierung auf der Karte repräsentieren, siehe Abbildung 6.

Im Benutzerprofil soll der Benutzer sein Benutzername einsehen und sich von der Anwendung abmelden können, siehe Abbildung 7. Nach dem Abmelden soll der Anwender wieder zum Anmeldebildschirm gelangen.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Abbildung Ergebnisliste | Abbildung Ergebniskarte | Abbildung Benutzerprofil |

In der Übersicht des Benutzers registrierten Hardwaresysteme sollen diese in einer Liste mit einem Header mit Sortieroptionen für den Namen und der Zeit der letzten Übertragung angegeben werden, siehe Abbildung 8. Hier soll der Benutzer die Möglichkeit besitzen, über einen Button neue Hardwaresysteme für sich zu registrieren.

Für diese Registrierung sollen alle vorhandenen Hardwaresystem im gesamten System in einer Liste ausgegeben werden, siehe Abbildung 9. Es soll zudem eine Filterungsoption angeboten werden, um die Liste nach einem spezifischen Namen zu filtern. Der Benutzer soll durch das Antippen eines Listeneintrages diesen für sich registrieren können.

|  |  |
| --- | --- |
|  |  |
| Abbildung Übersicht benutzerregistrierte Hardwaresysteme | Abbildung Neue Hardwaresystemregistrierung |

### Hardware-Interface

Das Hardwaresystem soll auf einem Heltec LoRaNode 151[1] laufen, welches mit dem Beschleunigungsmesser MMA7361 und dem GNSS-Modul SAM-M8Q von Mateksys[3] kommuniziert. Das LoRaNode 151 beinhaltet die Möglichkeit zur LoRa-Kommunikation mit Hilfe des eingebauten LoRa-Chip SX1276[4].

Die Mobilanwendung soll auf dem Android-Betriebssystem laufen, welches Zugriff auf eine Internetverbindung anbieten muss.

### Software-Interface

* Das Hardwaresystem soll mit dem Beschleunigungsmesser kommunizieren, um eine Bewegung zu ermitteln.
* Das Hardwaresystem soll mit dem GNSS-Modul kommunizieren, um den aktuellen Standort zu erfahren.
* Das Hardwaresystem soll mit dem LoRaWAN-Netzwerk kommunizieren, um die Standorte bei einer Bewegung zu übertragen.
* Das LoRaWAN-Netzwerk soll mit dem Webserver kommunizieren, um die Standortdaten von Hardwaresystemen weiterzuleiten.
* Der Webserver soll mit der einem Benachrichtigungsdienst kommunizieren, um Benachrichtigungen an Smartphones mit installierter Mobilanwendung zu senden.
* Die Mobilanwendung soll mit dem Webserver kommunizieren, um die benötigten Daten für die Anwendung zu ermitteln.
* Die Anwendung soll mit einem Routingdienst kommunizieren, um die Route zwischen Standorten zu erfahren.

### Kommunikation-Interface

Das Hardwaresystem soll für die Kommunikation mit dem GNSS-Modul das UART-Protokoll und für dessen Nachrichten das UBX-Protokoll[5], für die Kommunikation mit dem Beschleunigungsmesser eine analoge Verbindung, für die Kommunikation mit dem LoRa-Chip die SPI-Schnittstelle und für die Kommunikation mit dem LoRaWAN-Netzwerk die Modulationstechnik LoRa verwenden.

Die Kommunikation über das Internet soll über das http-Protokoll stattfinden.

## Funktionale Anforderungen

### Teilsystem 1 – Hardwaresystem

#### Funktionale Anforderung 1.1

|  |  |
| --- | --- |
| Name | Erfassung einer Bewegung |
| Beschreibung | Ein Hardwaresystem soll mit einem Beschleunigungsmesser kommunizieren, um eine Bewegung des Systems erfassen und erkennen zu können. |
| Eingabe | Ist Sensor in Bewegung: boolean |
| Verarbeitungsschritte | Messe den Ausgang zur Angabe, ob sich der Bewegungsmesser in Bewegung befindet.  Falls dies der Fall ist, gebe wahr zurück und deaktiviere den Beschleunigungssensor.  Falls nicht, dann gebe falsch zurück. |
| Ausgabe | Es wird entweder wahr oder falsch ausgegeben. |

#### Funktionale Anforderung 1.2

|  |  |
| --- | --- |
| Name | Aus Tiefschlaf erwachen |
| Beschreibung | Das Hardwaresystem soll auf die Erfassung einer Bewegung aus dem Tiefschlaf des Gerätes erwachen. |
| Eingabe | Bewegung erkannt: boolean |
| Verarbeitungsschritte | Erwecke das System aus dem Tiefschlaf. |
| Ausgabe | Es wird das System aktiviert vorgefunden. |

#### Funktionale Anforderung 1.3

|  |  |
| --- | --- |
| Name | Aufbau Verbindung zu LoRaWAN-Netzwerk |
| Beschreibung | Das System soll beim erstmaligen Hochfahren eine Verbindung zum LoRaWAN-Netzwerk aufbauen. |
| Eingabe | Ist angemeldet: boolean |
| Verarbeitungsschritte | Prüfe, ob das System angemeldet ist.  Falls nicht, dann sende eine Join-Request-Nachricht an das LoRaWAN-Netzwerk und erwarte eine Join-Accept-Nachricht.  Falls keine Nachricht erhalten wurde, gehe für einen kurzen Moment in den Tiefschlaf.  Falls eine Join-Accept-Nachricht erhalten wurde, dann setze „ist angemeldet“ auf wahr.  Wiederhole die Funktion solange, bis „ist angemeldet“ wahr ist. |
| Ausgabe | Es wird „ist angemeldet“ zurückgegeben. |

#### Funktionale Anforderung 1.4

|  |  |
| --- | --- |
| Name | Erfassung des Standortes |
| Beschreibung | Das Hardwaresystem soll mit einem GNSS-Modul kommunizieren, um die aktuelle Position des Systems zu erfassen. |
| Eingabe | - |
| Verarbeitungsschritte | Sende eine Standortanforderung an das GNSS-Modul und warte auf eine Antwort.  Parse die Antwort.  Falls die Antwort keinen Standort erhält, wiederhole die Standortanforderung.  Falls die Antwort einen Standort enthält, dann ermittele den Längen- und Breitengrad und gebe diese zurück. |
| Ausgabe | Es wird die Längen- und Breitengrade des Standortes zurückgegeben. |

#### Funktionale Anforderung 1.5

|  |  |
| --- | --- |
| Name | Aufbau Payload einer LoRaWAN-Nachricht |
| Beschreibung | Das Hardwaresystem soll den Payload einer LoRaWAN-Nachricht mit dem Längen- und Breitengrad des Standortes befüllen können. |
| Eingabe | Längengrad: long  Breitengrad: long |
| Verarbeitungsschritte | Entferne die niedrigsten acht Bits der beiden Grade.  Erstelle eine sechs-Byte große Variable.  Füge die drei Bytes des Längengrades in die oberen drei Bytes der Variable ein  Füge die drei Bytes des Breitengrades in die unteren drei Bytes der Variable ein.  Gebe die Variable zurück. |
| Ausgabe | Es wird eine sechs-Byte große Variable gefüllt mit dem Längen- und Breitengrad des Standortes zurückgegeben. |

#### Funktionale Anforderung 1.6

|  |  |
| --- | --- |
| Name | Übertragung einer LoRa-Nachricht |
| Beschreibung | Das Hardwaresystem soll eine LoRa-Nachricht versenden können. |
| Eingabe | Payload: Variable |
| Verarbeitungsschritte | Füge die Payload in das Gerüst einer LoRa-Nachricht ein.  Sende die Nachricht mittels der Modulationstechnik LoRa. |
| Ausgabe | Es wird nichts zurückgegeben. |

#### Funktionale Anforderung 1.7

|  |  |
| --- | --- |
| Name | Versetzung in Tiefschlaf |
| Beschreibung | Das Hardwaresystem soll nach dem Senden einer LoRa-Nachricht in den Tiefschlaf übergehen. |
| Eingabe | - |
| Verarbeitungsschritte | Deaktiviere alle Peripheriegeräte bis auf den Beschleunigungssensor.  Setze ein Aufwachevent bei der Erfassung einer Bewegung.  Setze das Hardwaresystem in den Tiefschlaf. |
| Ausgabe | Es wird nichts ausgegeben. |

### Teilsystem 2 – LoRaWAN-Netzwerk

#### Funktionale Anforderung 2.1

|  |  |
| --- | --- |
| Name | Dekodierung Payload der erhaltenen Nachricht |
| Beschreibung | Das LoRaWAN-Netzwerk soll die Payload einer LoRaWAN-Nachricht dekodieren und den Längen- und Breitengrad aus der Payload beziehen können. |
| Eingabe | Payload: Variable |
| Verarbeitungsschritte | Dekodiere die Payload.  Speichere die oberen drei Bytes der Payload in einer Variablen ab und füge acht Nullen als die niedrigsten Bits hinzu. Nenne diese Variable Längengrad.  Speichere die unteren drei Bytes der Payload in einer Variablen ab und füge acht Nullen als die niedrigsten Bits hinzu. Nenne diese Variable Breitengrad. |
| Ausgabe | Es wird der Längen- und Breitengrad zurückgegeben. |

#### Funktionale Anforderung 2.2

|  |  |
| --- | --- |
| Name | Übertragung dekodierter Payload an Webserver |
| Beschreibung | Das LoRaWAN-Netzwerk soll die beiden Grade, die aktuelle Uhrzeit und das aktuelle Datum in einer Nachricht an den Webserver senden können. |
| Eingabe | Längengrad: long  Breitengrad: long  Aktuelle Uhrzeit und Datum: time |
| Verarbeitungsschritte | Füge den Längen- und Breitengrad sowie die aktuelle Uhrzeit und Datum in eine Nachricht.  Sende die Nachricht an den Webserver. |
| Ausgabe | Es wird eine Nachricht mit dem Längen- und Breitengrad sowie mit der aktuellen Uhrzeit und Datum als Inhalt ausgegeben. |

### Teilsystem 3 – Webserver

#### Funktionale Anforderung 3.1

|  |  |
| --- | --- |
| Name | Speicherung eines Standortes |
| Beschreibung | Der Webserver soll auf Erhalt einer Standort-Nachricht vom LoRaWAN-Netzwerk den Standort des entsprechenden Hardwaresystem in der Datenbank speichern. |
| Eingabe | Name des Hardwaresystem: String  Breitengrad: float  Längengrad: float  Uhrzeit und Datum: time |
| Verarbeitungsschritte | Prüfe, ob die Eingabedaten vollständig sind.  Falls nicht, dann verwerfe die Nachricht  Falls schon, dann führe ein SQL-Insert-Befehl mit den Eingabedaten durch. |
| Ausgabe | Die Eingabedaten sind in der Datenbank gespeichert. |

#### Funktionale Anforderung 3.2

|  |  |
| --- | --- |
| Name | Benachrichtigung Benutzer auf Erhalt eines Standortes |
| Beschreibung | Der Webserver soll auf Erhalt eines Standortes alle Benutzer, die mit dem Hardwaresystem verknüpft sind, über den Erhalt eines Standortupdates benachrichtigen. |
| Eingabe | Hardwaresystemname: String  Breitengrad: float  Längengrad: float  Datum und Uhrzeit: time |
| Verarbeitungsschritte | Ermittle alle Benutzer, die mit dem Hardwaresystem verknüpft sind.  Sende eine Benachrichtigungsnachricht mit allen Benutzern und den Daten des Standortes vom Hardwaresystem an ein Benachrichtigungsdienst. |
| Ausgabe | Es wird eine Benachrichtigungsnachricht an ein Benachrichtigungsdienst gesendet. |

#### Funktionale Anforderung 3.3

|  |  |
| --- | --- |
| Name | Registrierung eines Benutzers |
| Beschreibung | Der Webserver soll auf Erhalt einer Registrierungsnachricht von der Mobilanwendung einen neuen Benutzer registrieren, indem dessen Benutzername und Passwort in der Datenbank gespeichert werden. |
| Eingabe | Benutzername: String  Benutzerpasswort: String |
| Verarbeitungsschritte | Prüfe, ob der Benutzername frei ist.  Falls nicht, dann gebe als Antwortnachricht zurück, dass der Benutzername vergeben ist  Falls schon, dann führe ein SQL-insert-Befehl aus und speichere den Benutzername und das -passwort. Gebe als Antwortnachricht zurück, dass die Registrierung erfolgreich ist. |
| Ausgabe | Entweder Speicherung der Eingabedaten und Rückgabe einer Antwort-nachricht mit Erfolg der Registrierung oder Rückgabe einer Antwort-nachricht mit fehlgeschlagener Registrierung aufgrund des bereits vergebenen Benutzernamen. |

#### Funktionale Anforderung 3.4

|  |  |
| --- | --- |
| Name | Anmeldung eines Nutzers |
| Beschreibung | Der Webserver soll auf Erhalt einer Anmeldenachricht von der Mobilanwendung die Anmeldung eines Benutzers überprüfen. |
| Eingabe | Benutzername: String  Benutzerpasswort: String |
| Verarbeitungsschritte | Prüfe, ob der Benutzername registriert ist.  Falls nicht, dann gebe als Antwortnachricht zurück, dass die Anmeldung nicht erfolgreich ist.  Falls schon, dann prüfe, ob das Passwort zum Benutzernamen passt.  Falls nicht, dann gebe als Antwortnachricht zurück, dass die Anmeldung nicht erfolgreich ist.  Falls schon, dann gebe als Antwortnachricht zurück, dass die Anmeldung erfolgreich ist. |
| Ausgabe | Entweder ist die Rückgabe einer Antwortnachricht mit erfolgreicher Anmeldung oder die Rückgabe einer Antwortnachricht mit fehlgeschlagener Anmeldung vorhanden. |

#### Funktionale Anforderung 3.5

|  |  |
| --- | --- |
| Name | Rückgabe Standortsuche eines Hardwaresystem |
| Beschreibung | Der Webserver soll auf Erhalt einer Standortsuchnachricht alle Standorte eines Hardwaresystems innerhalb gegebener Parameter in einer Antwort zurückgeben. |
| Eingabe | Hardwaresystemname: String  Untere Grenze Zeit: Time  Obere Grenze Zeit: Time  Untere Grenze Breitengrad: Float  Obere Grenze Breitengrad: Float  Untere Grenze Längengrad: Float  Obere Grenze Längengrad: Float |
| Verarbeitungsschritte | Beginne einen SQL-Read-Befehl mit der Bedingung des Hardwaresystem und der Eingrenzung der Zeit.  Prüfe, ob die untere Grenze des Breitengrades angegeben ist.  Falls nicht, dann überspringe die Bedingung des Breitengrades.  Falls schon, dann füge dem SQL-Befehl die Bedingung hinzu, dass der Breitengrad eines Standortes zwischen der oberen und unteren Grenze liegt.  Prüfe, ob die untere Grenze des Längengrades angegeben ist.  Falls nicht, dann überspringe die Bedingung des Längengrades.  Falls schon, dann füge dem SQL-Befehl die Bedingung hinzu, dass der Breitengrad eines Standortes zwischen der oberen und unteren Grenze liegt.  Führe den SQL-Befehl aus.  Gebe als Antwortnachricht alle Einträge der Suche zurück. |
| Ausgabe | Eine Antwortnachricht mit allen Einträgen der Standorte wird zurück-gegeben. |

#### Funktionale Anforderung 3.6

|  |  |
| --- | --- |
| Name | Rückgabe aller registrierten Hardwaresysteme eines Benutzers |
| Beschreibung | Der Webserver soll auf einer Hardwaresystemregistrierungs-übersichtnachricht alle registrierten Hardwaresysteme eines Benutzers in einer Antwort zurückgeben. |
| Eingabe | Benutzername: String |
| Verarbeitungsschritte | Führe einen SQL-Read-Befehl mit dem Benutzernamen aus  Gebe als Antwortnachricht alle erhaltenen Einträge zurück. |
| Ausgabe | Es wird eine Antwortnachricht mit allen Hardwaresysteme eines Benutzers zurückgegeben. |

#### Funktionale Anforderung 3.7

|  |  |
| --- | --- |
| Name | Verknüpfung eines Hardwaresystems an einem Benutzer |
| Beschreibung | Der Webserver soll bei Erhalt einer Verknüpfungsnachricht ein Hardware-system in Verbindung mit einem Benutzer speichern. |
| Eingabe | Benutzername: String  Hardwaresystemnachricht: String |
| Verarbeitungsschritte | Prüfe, ob die Verknüpfung frei ist.  Falls nicht, dann gebe als Antwortnachricht zurück, dass die Verknüpfung fehlgeschlagen ist.  Falls schon, dann führe einen SQL-Insert-Befehl mit dem erhaltenen Benutzer und Hardwaresystem durch und gebe als Antwort-nachricht zurück, dass die Verknüpfung erfolgreich ist. |
| Ausgabe | Es wird entweder eine Antwortnachricht über die erfolgreiche Verknüpfung oder eine Antwortnachricht über eine fehlgeschlagene Verknüpfung zurückgegeben. |

### Teilsystem 4 – Mobilanwendung

#### Funktionale Anforderung 4.1

|  |  |
| --- | --- |
| Name | Benutzerregistrierung |
| Beschreibung | Der Benutzer soll sich über die Mobilanwendung registrieren können. Der Benutzer muss dafür einen Benutzernamen und Passwort angeben. |
| Eingabe | Benutzername: String  Benutzerpasswort: String |
| Verarbeitungsschritte | Prüfe, ob das Eingabefeld für den Benutzername und den Benutzerpasswort leer sind.  Falls mindestens ein Eingabefeld leer ist, dann wird eine entsprechende Fehlermeldung ausgegeben.  Falls kein Eingabefeld leer ist, dann sende eine Registrierungsnachricht an den Webserver und erwarte eine Antwortnachricht.  Falls die Antwortnachricht beinhaltet, dass die Registrierung fehlgeschlagen ist, dann gebe eine entsprechende Fehlermeldung aus.  Falls die Antwortnachricht beinhaltet, dass die Registrierung erfolgreich ist, dann gebe eine entsprechende Meldung aus und wechsele zum Anmeldebildschirm. |
| Ausgabe | Eine Meldung über eine erfolgreiche Registrierung wird angezeigt und es wird zum Anmeldebildschirm übergeführt oder es wird eine Fehlermeldung angezeigt. |

#### Funktionale Anforderung 4.2

|  |  |
| --- | --- |
| Name | Benutzeranmeldung |
| Beschreibung | Wenn der Benutzer registriert ist, dann soll er sich in der Anwendung anmelden können. Der Benutzer muss dafür sein Benutzernamen und Passwort angeben. Die Anmeldedaten sollen gespeichert werden und in der Zukunft soll der Benutzer automatisch angemeldet werden. |
| Eingabe | Benutzername: String  Benutzerpasswort: String |
| Verarbeitungsschritte | Prüfe, ob das Eingabefeld für den Benutzername und den Benutzerpasswort leer sind.  Falls mindestens ein Eingabefeld leer ist, dann wird eine entsprechende Fehlermeldung ausgegeben.  Falls kein Eingabefeld leer ist, dann sende eine Anmeldenachricht mit dem Benutzernamen und -Passwort an den Webserver und erwarte eine Nachricht.  Falls die Antwortnachricht beinhaltet, dass die Anmeldung fehlgeschlagen ist, dann gebe eine entsprechende Fehlermeldung aus.  Falls die Antwortnachricht beinhaltet, dass die Anmeldung erfolgreich ist, dann speichere die Anmeldedaten und wechsele zum Hauptmenü-Bildschirm. |
| Ausgabe | Es wird zum Hauptmenü-Bildschirm übergangen oder eine Fehlermeldung angezeigt. |

#### Funktionale Anforderung 4.3

|  |  |
| --- | --- |
| Name | Benutzerprofil |
| Beschreibung | Der angemeldete Benutzer soll sein Benutzerprofil betrachten können. |
| Eingabe | - |
| Verarbeitungsschritte | Übergehe zum Benutzerprofilbildschirm. |
| Ausgabe | Es wird zum Benutzerprofilbildschirm übergegangen. |

#### Funktionale Anforderung 4.4

|  |  |
| --- | --- |
| Name | Benutzerabmeldung |
| Beschreibung | Der Benutzer das angemeldete Profil abmelden können. Die gespeicherten Anmeldedaten sollen gelöscht werden. |
| Eingabe | - |
| Verarbeitungsschritte | Die gespeicherten Anmeldedaten werden gelöscht.  Übergehe zum Anmeldebildschirm.  Der Zugang zu dem vorherigen Bildschirm über die „Zurück“-Taste wird verhindert. |
| Ausgabe | Es wird zum Anmeldebildschirm übergangen. |

#### Funktionale Anforderung 4.5

|  |  |
| --- | --- |
| Name | Suche nach Standorten |
| Beschreibung | Der Benutzer soll nach den vergangenen Standorten eines Hardwaresystems mit verschiedenen Suchoptionen suchen können. Zu diesen Suchoptionen soll die Auswahl eines Hardwaresystem, ein Zeit-, Breitengrad- und Längengradintervall gehören. Ein Benutzer soll dazu in der Lage sein, mehrere Suchoptionen in einer Suche zu verwenden. |
| Eingabe | Hardwaresystemname: String  Untere Grenze Zeit: time  Obere Grenze Zeit: time  Untere Grenze Breitengrad: float  Obere Grenze Breitengrad: float  Untere Grenze Längengrad: float  Obere Grenze Längengrad: float |
| Verarbeitungsschritte | Prüfe, ob ein Hardwaresystemname ausgewählt wurde.  Falls nicht, dann gebe eine entsprechende Fehlermeldung aus.  Falls schon, dann übergebe den ausgewählten Hardwaresystemname an die Suche  Prüfe, ob beide Eingabefelder für das Zeitintervall gefüllt sind.  Falls nicht, dann prüfe, ob eins gefüllt ist.  Falls schon, dann übergebe das ausgewählte Datum als obere und untere Grenze an die Suche.  Falls nicht, dann gebe eine entsprechende Fehlermeldung aus.  Falls schon, dann übergeben beide Grenzen an die Suche.  Prüfe, ob beide Eingabefelder für das Breitengradintervall gefüllt sind.  Falls schon, dann übergebe das ausgewählte Breitengradintervall an die Suche.  Falls nicht, dann prüfe, ob beide Eingabefelder leer sind.  Falls schon, dann über gebe kein Breitengradintervall an die Suche.  Falls nicht, dann gebe eine entsprechende Fehlermeldung aus.  Prüfe, ob beide Eingabefelder für das Längengradintervall gefüllt sind.  Falls schon, dann übergebe das ausgewählte Längengradintervall an die Suche.  Falls nicht, dann prüfe, ob beide Eingabefelder leer sind.  Falls schon, dann über gebe kein Längengradintervall an die Suche.  Falls nicht, dann gebe eine entsprechende Fehlermeldung aus.  Sende eine Standortsuchnachricht mit den erhaltenen Parametern an den Webserver und erwarte die Antwortnachricht.  Auf Erhalt der Antwortnachricht übergehe in das Listenansichtbildschirm. |
| Ausgabe | Es wird zum Listenansichtbildschirm übergangen oder eine Fehlermeldung wird ausgegeben. |

#### Funktionale Anforderung 4.6

|  |  |
| --- | --- |
| Name | Suchergebnis in einer Liste |
| Beschreibung | Der Benutzer soll das Suchergebnis in einer Liste ansehen können. Jedes Element in der Liste soll einem der vergangenen Standorte des Hardwaresystems darstellen. Zudem soll jedes Element den Standort in Breiten- und Längengrad sowie das Datum und die Uhrzeit der Erfassung des Standortes angeben. Liste soll nach dem Datum und der Uhrzeit vorsortiert werden. Es soll ein Header mit Sortieroptionen und ein Button für die Auswahl von Einträgen für eine Routenplanung angeboten werden. |
| Eingabe | Liste Standorteinträge: List<Standort> |
| Verarbeitungsschritte | Prüfe, ob mindestens ein Standorteintrag vorhanden ist.  Falls schon, dann gebe alle Einträge innerhalb der Liste wieder.  Falls nicht, dann gebe eine entsprechende Meldung an und zeige eine leere Liste an. |
| Ausgabe | Es wird eine ausgefüllte Liste mit Standorteinträgen ausgegeben oder es wird eine Meldung über keine vorhandenen Standorte und eine leere Liste angegeben. |

#### Funktionale Anforderung 4.7

|  |  |
| --- | --- |
| Name | Sortierung Standortliste |
| Beschreibung | Wenn der Benutzer eine Standortliste betrachtet, dann soll er das Resultat nach den Breitengraden, Längengraden und der Zeit der Erfassung sortieren können. |
| Eingabe | - |
| Verarbeitungsschritte | Prüfe, welche Sortieroption gewählt wurde.  Falls die Breitengradoption gewählt wurde, dann sortiere die Einträge der Liste vom kleinsten Breitengrad zum größten Breitengrad.  Falls die Längengradoption gewählt wurde, dann sortiere die Einträge der Liste vom kleinsten Längengrad zum größten Längengrad.  Falls die Zeitoption gewählt wurde, dann sortiere die Einträge der Liste vom jüngsten Zeitpunkt zum ältesten Zeitpunkt.  Falls eine Option mehrfach hintereinander genutzt wird, dann kehre die vorhandene Sortierung um. |
| Ausgabe | Es wird eine nach der ausgewählten Sortieroption sortierte Liste ausgegeben. |

#### Funktionale Anforderung 4.8

|  |  |
| --- | --- |
| Name | Suchergebnis in einer Kartenansicht |
| Beschreibung | Der Benutzer soll das Suchergebnis in einer Kartenansicht einsehen können. Auf der Karte soll jeder Standorteintrag als eine spezifische Markierung vorhanden sein und die Sicht soll im Zentrum aller Einträge sein. Es soll ein Button für die Auswahl von Einträgen für eine Routenplanung angeboten werden. |
| Eingabe | Liste Standorteinträge: List<Standort> |
| Verarbeitungsschritte | Prüfe, ob mindestens ein Standorteintrag vorhanden ist.  Falls schon, dann gebe alle Einträge auf der Karte als Markierung wieder.  Falls nicht, dann gebe eine entsprechende Meldung an und zeige eine leere Karte an. |
| Ausgabe | Es wird eine Karte mit Standortmarkierungen ausgegeben oder es wird eine Meldung über keine vorhandenen Standorte und eine leere Karte angegeben. |

#### Funktionale Anforderung 4.9

|  |  |
| --- | --- |
| Name | Ansichtsänderung zwischen Liste und Kartenansicht |
| Beschreibung | Der Benutzer soll zwischen der Listen- und Kartenansicht wechseln können. |
| Eingabe | - |
| Verarbeitungsschritte | Falls in der Listenansicht, übergehe zur Kartenansicht.  Falls in der Kartenansicht, übergehe zur Listenansicht. |
| Ausgabe | Es wird in die entsprechend andere Ansicht gewechselt. |

#### Funktionale Anforderung 4.10

|  |  |
| --- | --- |
| Name | Routenplanung |
| Beschreibung | Wenn der Benutzer in der Listenansicht oder in der Kartenansicht ist, dann soll der Benutzer in einen Modus zum Routen planen übergehen können und Standortseinträge in einer frei auswählbaren Reihenfolge markieren können. Es soll ein Button vorhanden sein, um die Routenplanung der Reihenfolge entsprechend durchzuführen. |
| Eingabe | - |
| Verarbeitungsschritte | Falls ein nicht ausgewählter Eintrag markiert wird, dann füge einen Haken an den Eintrag an.  Falls ein markierter Eintrag gewählt wird, dann entferne den Haken. |
| Ausgabe | Entsprechend der Ansicht wird entweder eine Liste mit Standorteinträgen mit eventuellem Haken oder eine Karte mit eventuell mit Haken versehende Markierungen ausgeben. |

#### Funktionale Anforderung 4.11

|  |  |
| --- | --- |
| Name | Routenergebnis |
| Beschreibung | Nach dem Drücken des Routenplanungsbutton soll das System eine Route zwischen den gewählten Standorten ermitteln und diese in einer Kartenansicht darstellen können. |
| Eingabe | Gewählte Standorte: List<Standort> |
| Verarbeitungsschritte | Prüfe, ob mindestens zwei Standorte markiert sind.  Falls nicht, dann gebe eine entsprechende Fehlermeldung aus.  Falls schon, dann erfasse alle markierten Einträge innerhalb einer Liste.  Sende eine Routennachricht mit allen Einträgen an einen Routendienst und erwarte eine Antwort.  Prüfe den Inhalt der Antwort.  Falls keine Route vorhanden ist, dann gebe eine entsprechende Fehlermeldung aus.  Falls eine Route vorhanden ist, dann erstelle eine neue Kartenansicht mit den markierten Standorten und der Route zwischen diesen. |
| Ausgabe | Entweder werden Fehlermeldungen über eine fehlgeschlagene Routenplanung ausgegeben oder eine neue Kartenansicht mit den ausgewählten Standorten und der Route ausgegeben. |

#### Funktionale Anforderung 4.12

|  |  |
| --- | --- |
| Name | Übersicht registrierter Hardwaresysteme |
| Beschreibung | Der Benutzer soll eine Übersicht in Form einer Liste über all seine registrierten Hardwaresystemen einsehen können. Es soll zudem ein Header mit Sortieroptionen für den Namen und den letzten Zeitpunkt eines Hardwaresystems, eine Möglichkeit zur Filterung nach einem Namen geben sowie ein Button für die Übersicht aller vorhandenen Hardwaresystemen vorhanden sein. |
| Eingabe | - |
| Verarbeitungsschritte | Sende eine Hardwaresystemregistrierungsübersichtnachricht an den Webserver und erwarte eine Antwort.  Prüfe, ob Einträge in der Antwort vorhanden sind.  Falls mindestens ein registriertes Hardwaresystem vorhanden ist, dann füge alle als individuelle Einträge in eine Liste ein und zeige diese an.  Falls keine Einträge vorhanden sind, dann gebe eine Meldung über die Abwesenheit von registrierten Hardwaresystem mit dem Verweis auf wie man ein Hardwaresystem registriert und eine leere Liste an. |
| Ausgabe | Es wird eine Liste mit Hardwaresystemeinträgen oder eine Meldung und eine leere Liste ausgegeben. |

#### Funktionale Anforderung 4.13

|  |  |
| --- | --- |
| Name | Sortierung Übersicht Hardwaresysteme |
| Beschreibung | Wenn der Benutzer seine Hardwaresystemliste betrachtet, dann soll er das Resultat nach den Namen und den Zeitpunkt der letzten Erfassung sortieren können. |
| Eingabe | - |
| Verarbeitungsschritte | Prüfe, welche Sortieroption gewählt wurde.  Falls die Namenoption gewählt wurde, dann sortiere die Einträge der Liste alphabetisch nach dem Namen der Hardwaresysteme.  Falls die Zeitoption gewählt wurde, dann sortiere die Einträge der Liste vom jüngsten Zeitpunkt zum ältesten Zeitpunkt.  Falls eine Option mehrfach hintereinander genutzt wird, dann kehre die vorhandene Sortierung um. |
| Ausgabe | Es wird eine nach der ausgewählten Sortieroption sortierte Liste ausgegeben. |

#### Funktionale Anforderung 4.14

|  |  |
| --- | --- |
| Name | Filterung Liste neuer HWS |
| Beschreibung | Wenn der Benutzer seine Hardwaresystemliste betrachtet, dann soll er die Liste nach den Hardwaresystemnamen filtern können. |
| Eingabe | Filtereingabe: String |
| Verarbeitungsschritte | Prüfe, ob das Eingabefeld vom Filter befüllt ist.  Falls schon, dann filtere die Einträge der vorhandenen Liste nach deren Hardwaresystemnamen, die teilweise mit der Filtereingabe übereinstimmen.  Falls nicht, dann erhalte die vorhandene Liste. |
| Ausgabe | Es wird die vorhandene Liste oder eine Liste mit Einträgen gefiltert nach der Filtereingabe ausgegeben. |

#### Funktionale Anforderung 4.15

|  |  |
| --- | --- |
| Name | Übersicht aller Hardwaresysteme |
| Beschreibung | Der Benutzer soll eine Übersicht über alle Hardwaresysteme im System in Form einer Liste erhalten. In der Liste sollen alle Hardwaresysteme nur mit deren Namen sowie nach deren Namen in alphabetischer Reihenfolge sortiert angegeben werden. Es soll zudem die Möglichkeit für eine Filterung nach den Namen der Hardwaresysteme geben. |
| Eingabe | - |
| Verarbeitungsschritte | Sende eine Hardwaresystemübersichtnachricht und erwarte eine Antwort.  Erfasse alle Hardwaresysteme aus der Antwort in einer Liste.  Sortiere die Liste alphabetisch.  Gebe die sortierte Liste aus. |
| Ausgabe | Es wird eine nach dem Alphabet sortierte Liste der Namen aller Hardwaresysteme ausgegeben. |

#### Funktionale Anforderung 4.16

|  |  |
| --- | --- |
| Name | Auswahl eines neuen Hardwaresystem |
| Beschreibung | Wenn der Benutzer die Liste der Namen aller Hardwaresysteme anschaut, dann soll er auf einen Eintrag auswählen können, um dieses Hardwaresystem für seinem Profil zu registrieren. Wenn eine Wahl getroffen wurde, dann soll ein Dialog geöffnet werden, in dem der Benutzer zur Eingabe des Passwortes vom Hardwaresystems aufgefordert wird. |
| Eingabe | Hardwaresystemname: String |
| Verarbeitungsschritte | Öffne auf Klick eines Eintrages ein Dialog mit Namen des Hardwaresystems, ein Eingabefeld für das Passwort und ein Button. |
| Ausgabe | Es wird ein Dialog zur Registrierung eines ausgewählten Hardwaresystems ausgegeben. Bei korrekter Eingabe wird zur Übersicht der registrierten Hardwaresysteme gewechselt. Ansonsten wird der Dialog geschlossen. |

#### Funktionale Anforderung 4.17

|  |  |
| --- | --- |
| Name | Registrierung eines neuen Hardwaresystems |
| Beschreibung | Im Dialogfenster für die Registrierung eines Hardwaresystem soll der Benutzer nach Eingabe eines zum Hardwaresystem passendes Passwort das Hardwaresystem für sich registrieren können. |
| Eingabe | Benutzername: String  Hardwaresystemname: String  Hardwaresystempasswort: String  Eingegebenes Passwort: String |
| Verarbeitungsschritte | Prüfe, ob das Passwortfeld leer ist.  Falls dem so ist, dann gebe eine entsprechende Fehlermeldung aus.  Falls nicht, dann prüfe, ob das eingegebene Passwort gleich dem Hardwaresystempasswort ist.  Falls nicht, dann gebe eine entsprechende Fehlermeldung aus.  Falls schon, dann sende eine Verknüpfungsnachricht, welche den Benutzernamen und den Hardwaresystemnamen enthält, an den Webserver und erwarte eine Antwort.  Falls die Antwort den Erfolg der Verknüpfung enthält, dann gebe eine entsprechende Meldung aus und wechsele zur Übersicht aller registrierten Hardwaresysteme.  Falls die Antwort den Fehlschlag über die Verknüpfung enthält, dann gebe eine entsprechende Fehlermeldung aus. |
| Ausgabe | Entweder werden Fehlermeldungen oder eine Meldung über den Erfolg der Verknüpfung mit anschließendem Wechsel zur Hardwaresystemübersicht ausgegeben. |

#### Funktionale Anforderung 4.18

|  |  |
| --- | --- |
| Name | Push-Benachrichtigung |
| Beschreibung | Der Benutzer der Anwendung soll eine Benachrichtigung bei Erhalt einer Standortübertragung eines seiner registrierten Hardwaresysteme bekommen. |
| Eingabe | Benachrichtigung: Standort |
| Verarbeitungsschritte | Bei Erhalt eines Standorteintrages verfasst die Mobilanwendung eine Benachrichtigung mit dem Namen und den Standort des Hardwaresystems. |
| Ausgabe | Auf dem Mobiltelefon wird eine Benachrichtigung über den Erhalt eines Standortes eines Hardwaresystem angegeben. |

## Performance-Anforderungen

Der Stromverbrauch des Hardwaresystem soll unter der Verwendung von stromsparenden Modi so gering wie möglich gehalten werden. Zudem sollen die Antwortzeiten des Webserver unter der Verwendung von industrieüblichen Methoden minimiert werden.

## Zuverlässigkeitsanforderung

Es sind keine Zuverlässigkeitsanforderungen für dieses System definiert. Dennoch sollen Feature-Tests durchgeführt werden.

## Verfügbarkeitsanforderung

Es soll ein angemessener Aufwand verbracht werden, um zu versichern, dass das System mindestens 95% der Zeit verfügbar ist.

## Sicherheitsanforderung

Das System muss industrieübliche Methoden für eine sichere Softwareentwicklung folgen. Der Webserver soll Methoden für die Vermeidung von Cyberangriffen implementieren.

## Wartbarkeitsanforderung

Die Architektur, das Design, die Implementation und die Dokumentation müssen die Wartungskosten der Anwendung minimieren.