0.1 Produktübersicht

Was ist der Frost-Server? Weboberfläche zum einfachen Abspeichern von Sensordaten im FROST Server

0.2 Musskriterien

1. Import von CSV-/XSLX-Dateien blalb
2. konvertierung von sensordaten von csv und xlsx in ISO format nach Vorgabe der SensorThings API (Format der Daten auf Sever)
3. Konfiguration der Formate der Sensordaten
4. erstellen, abspeichern, auslesen von Favoriten-konfigurationen
5. Datentyptransformationen
6. Verarbeiten von Sonderwerten (Magic Numbers)
7. Import von Fremdquellen wie anderen webseiten
8. Weitergabe von Fehlermeldungen an Nutzer, falls Import schief läuft

0.3 Wunschkriterien

1. Sprachauswahl DE/ENG
2. Bereitstellen einer vorauswahl (default)
3. überprüfung auf Duplikate
4. Erkennen und Rückgabe von Tipp- und Sinnfehlern bzw. Vorschläge für weiteres Vorgehen
5. Rückgabe nicht bearbeiteter Datensätze in neuer CSV-Datei
6. Docker-Container des Server-Programmes
7. Auswahl komplexer Transformationen im Webinterface (Aggregationen (Zusammenlegen von Daten), Summe, Min/Max, Durchschnitt...)
8. automatisierter regelmäßiger Download von entfernter Adresse

9. Import kompletter Datensätze von anderem FROST-Server
10. automatisierte Erkennung des Formats
11. Erweiterungsmöglichkeit der Software für besondere Formate

0.4 Abgrenzungskriterien

keine Verwaltung bestehender Sensordaten auf dem Server. Produkt dient nur zum Import von Sensordaten.

0.5 Anwendungsbereiche

Überll wo Daten im CSV-Format verarbeitet werden müssen:

- Verarbeiten von neuen Sensordaten im CSV-Format
- Integration von bestehenden Sensordaten in den FROST-Server

0.6 Zielgruppen

2 Zeilgruppen:

- Besitzer von Sensoren; Privatpersonen
- Personen ohne technisches Vorwissen bspw. Beamte
- Personen mit technischem know-how

0.7 Betriebsbedingungen

Unterschieden wird zwischen Endgerät und Server.
Die Betriebsbedingungen beschreiben die minimalen Anforderungen die die Software zur Ausführung benötigt.
Das Ziel-Endgerät ist ein Computer/Tablet, wobei jedes Gerät mit einer Internetanbindung zum Programmserver, einem Webbrowser und aktiviertem JavaScript ausreichend ist.

Der Programmserver benötigt nur eine Anbindung an den FROST-Server und das Endgerät und ein Java JRE.

- Browser
- JavaScript
- Verbindung zum Programmserver
- Verbindung zum FROST-Server
- Java JRE
- Anbindung an Endgerät (zB Internet)

0.8 Produktumgebung

Die Produktumgebung ist eine Instanz unter der die Software läuft. Diese erfüllt mindestens die Betriebsbedingungen.

- head-less Server
- Linux/Windows Server
- Java JRE
- SensorThings API

Die Software soll plattformunabhängig auf einem Server mit minimaler Rechen- und Speicherkapazität laufen. Des weiteren soll das Programm head-less ununterbrochen laufen. Der Server stellt das Java JRE zur Verfügung, auf dem die Software läuft. Die Software benötigt nur noch eine Verbindung zu einer FROST-Server-Instanz, welche ihm vom Nutzer in der Konfiguration gegeben wird.

0.9 Funktionale Anforderungen

- Hilfe-Button: Durch Klick wird die README-Datei im Browser aufgerufen
- Speichern-Button: Bei Klick auf diesen Button wird der Upload der Daten in die Datastreams/MultiDatastreams auf den FROST-Server angestoßen

- Radio-Boxen "Datei auswählen/"Web-Adresse angeben": Durch Auswahl einer der Boxen wird entweder die Textbox zur Eingabe des Weblinks der entfernten Datei freigeschaltet oder der "DurchsuchenButton zur Auswahl einer lokalen

- Durchsuchen: Durch Klick auf den "DurchsuchenButton öffnet sich der Datei-auswählen-Systemdialog, bei dem man die Datei auswählen kann. Im Systemdialog wird das Dateiformat auf entweder CSV oder XLS/XSLX begrenzt. Nachdem die Datei im Systemdialog ausgewählt wurde, wird sie auf die Größe überprüft und bei erfolgreicher Prüfung hochgeladen.

- Textbox SZiel": Hier wird die Adresse des FROST-Servers angegeben

- Button Äuswählen": Durch Klick auf diesen Button wird die im Textfeld SZielängegebene Adresse überprüft und durch einen Haken und im Log bestätigt. (Es sollte überprüft werden ob eine Verbindung hergestellt werden kann,ob eventuell Zugriffsrechte benötigt werden und ob die Zieladresse überhaupt zu einem FROST-Server führt)

- Konfiguration durchsuchen/laden: Durch Klick auf "Durchsuchen"öffnet sich der Datei-auswählen-Systemdialog in dem man die CFG-Konfigurationsdatei auswählen kann. Die Konfiguration wird geprüft und anschließend geladen.

- Konfiguration speichern: Durch Klick auf den Button SSpeichern"wird der Download der aktuellen Konfiguration als CFG-Datei gestartet

- Reset-Button: Bei Klick auf den Reset-Button wird die gesamte Konfiguration (Datums-/Zeitformat, Datastreams, Ziel) zurückgesetzt

- Thing- / id-Dropdown-Menü: Ein existierendes Thing kann anhand des Namens oder der ID aus einer Liste ausgewählt werden

- Button "Neu": Existiert ein Thing noch nicht so kann es durch den Button "neuërstellt werden. Es öffnet sich eine Dialogbox, die es ermöglicht ein neues Thing zu erstellen

- Textfelder SSpalte": Hier kann durch Angabe der Spaltennummern eingestellt werden in welchen Spalten das Datum/die Uhrzeit zu finden ist

- Textfelder "Format": Hier wird per RegEx-String eingestellt, in welchem Format das Datum in den Spalten vorliegt
- Button "+": Hier können mehr Spalten, die das Datum beinhalten, ausgewählt werden
- "NameDropdown-Menü: Hier kann der (Multi-)Datastream über seinen Namen oder seiner ID aus einer Liste ausgewählt werden
- Spalten-Textfelder: Hier kann zu der vorangehenden Einheit die Spalte, die den Wert enthält, angegeben werden
- Button "Neu": Durch Klick auf den Button öffnet sich ein Dialogfenster, welches das Erstellen eines neuen (Multi-)Datastreams ermöglicht
- Button "+": Durch klick auf den Button "+"wird eine neue Box von Feldern erstellt, in der man Spalten auf Werte eines Datastreams abbilden kann

0.10 Produktdaten

Die gesicherten Daten der Software sind minimal, konkret heißt das es wird nur ein Log-File gesichert. Die Konfiguration wird dem Nutzer per CFG-Datei zum Download bereitgestellt und nicht Serverseitig gespeichert.
Sollte das Kriterium [Wunsch: Automatischer Download] mit aufgenommen werden, so muss die Adresse der entfernten Datei und des zu nutzenden FROST-Servers auf dem Server gespeichert werden.

- Log-Datei
- Abspeichern der Konfigurationen
- Speichern der Domain des FROST-Servers auf dem Server
- Speichern der Domain der entfernten CSV-Adresse auf dem Server

0.11 Produktleistungen

Die Software soll sämtliche Aktionen der Nutzer in einem Log-File speichern. Die Eingaben sollen möglichst direkt verarbeitet und die Reaktionszeit möglichst gering sein. Bei langsamen Operationen soll der Nutzer Feedback

über eine Fortschrittsleiste und über das Log-Fenster bekommen.
Die Daten müssen korrekt an den FROST-Server übermittelt werden, dazu gehört auch Robustheit der Anwendung gegenüber falschen Angaben. Bei falschen Angaben oder sonstigen Fehlern soll der Nutzer mittels einer Fehlermeldung benachrichtigt werden.

- Logging
- Statusmeldungen an den Nutzer
- Schnelle Reaktionszeit auf Benutzer-Eingaben
- korrekte Übertragung der Daten
- Robustheit gegenüber falschen Formatangeben (Fehlermeldungen)

0.12 Nichtfunktionale Anforderungen

Die Software ist Open-Source und benötigt einen erreichbaren FROST-Server.Sie basiert auf der SensorThings-API welche wie der FROST-Server ebenfalls Open-Source ist. Das Programm wird in Java geschrieben und hängt dementsprechend von einer Java-Umgebung ab. Um Sicherheit zu gewährleisten sollte das Programm robust gegenüber schädlichen Anfragen sein, vor allem zu große oder zu viele Anfragen sollten dementsprechend abgelehnt werden um eine Überlastung des Servers zu vermeiden. Außerdem sollten die daten HTTPS-/SSL-Verschlüsselt werden um Abfangen der daten zu verhindern.

(einzuhaltende Gesetze, Normen, Urheber- und Markenrechte, Sicherheitsanforderungen, Plattformabhängigkeiten)

- Genutzt wird der FROST-Server und die SensorThings API (beides Open-Source)
- Die Software ist Open-Source (GPLv3)
- Die Software ist abhängig von der Java-Plattform
- Robustheit gegenüber zu großen bzw. zu vielen Anfragen
- HTTPS-/SSL-Verschlüsselung

0.13 Qualitätsanforderungen

Das Programm sollte über lange Zeiträume auch ohne Unterbrechungen zuverlässig ausgeführt werden können, solange die Produktumgebung dies zulässt. Die Benutzbarkeit hat höchste Priorität, auch Nutzer ohne oder wenig technischem Vorwissen sollten in der Lage sein die Software zu bedienen. Außerdem soll es möglichst einfach sein die Software schnell und unkompliziert zu ändern oder zu erweitern. Eine einfache Portierung ist gegeben, da die Software auf Java basiert und damit Platformunabhängig ist. Es wird lediglich eine Produktumgebung vorausgesetzt die die Betriebsbedingungen erfüllt.

(In diesem Kapitel wird den Qualitätsmerkmalen Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit je eine Qualitätsstufe aus sehr gut, gut, normal und nicht relevant zugeordnet.)

- Die Software sollte zuverlässig auch über längere Zeiträume ohne Unterbrechung laufen sofern es die Produktumgebung zulässt
- Die Benutzbarkeit hat höchste Priorität, es soll auch Nutzern mit wenig Computerkentnissen möglich sein die Software zu bedienen
- Es soll einfach möglich sein, Änderungen und Erweiterungen an der Software vorzunehmen
- Die Software ist einfach zu übertragen, da sie auf der plattformunabhängigen Programmiersprache Java basiert, es wird nur eine Produktumgebung vorausgesetzt, die lediglich die Betreibsbedingungen erfüllt

0.14 Globale Testfälle und Szenarien / Anwendungsfälle

- to be added after the GUI design

0.15 Zeitplanung

Beispielhafte Terminplanung:

- 07.05. - 27.05.: Pflichtenheft
- 28.05. - 24.06.: Entwurfsphase
- 25.06. - 22.07.: Implementierung

- 23.07. - 12.08.: Beispiel-Urlaubstermin (2 Wochen)
- 13.08. - 02.09.: Qualitätssicherung
- 03.09. - 09.09.: Terminfenster interne Abnahme
- 10.09. - 17.09.: Terminfenster Abschlusspräsentation

Fragen:

- Anfragen an server für POST, PATCH, PUT,...
- Standard der Daten (wir abgespeichert?), Beispiel CSV Dateien,...
- welche Formate sind möglich?
- was sind komplexere Tranformationen?
- was passiert mit falschen Daten? (13. Monat)

Glossar:

- FROST
- SensorThings