

Software zur Temperaturüberwachung

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Änderung | Bearbeiter |
| 1.0 | 03.04.2020 | Erstellung | M-Nour, Florian |
|  |  |  |  |

Änderungshistorie

Inhaltsverzeichnis

[1. User Story 3](#_Toc36794783)

[1.1 Akzeptanzkriterien: 3](#_Toc36794784)

[2. Anwendungsfälle/Anforderunge (Übersicht): 4](#_Toc36794785)

[2.1 Nicht unterstützte Anwendungsfälle 4](#_Toc36794786)

[2.2 Allgemeine Anforderungen (nichtfunktional) 4](#_Toc36794787)

[2.3 Funktionale Anforderungen 4](#_Toc36794788)

[3. Analyse 4](#_Toc36794789)

[3.1 Aktuelle Umsetzung 4](#_Toc36794790)

[3.2 Ablauf (IPO) 5](#_Toc36794791)

[3.2.1 Input: 5](#_Toc36794792)

[3.2.2 Prozess: 5](#_Toc36794793)

[3.2.3 Output: 5](#_Toc36794794)

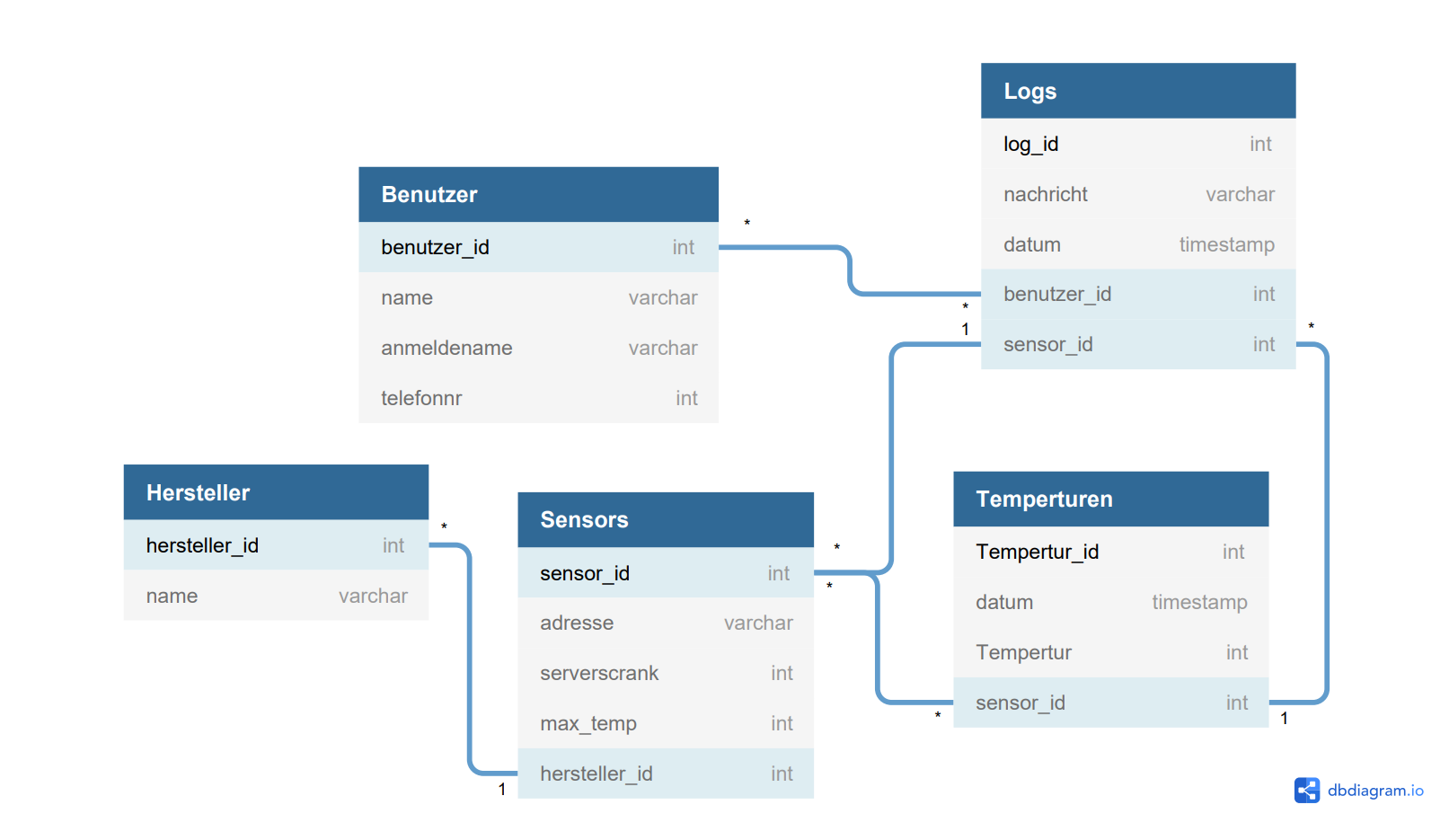
[3.3 Funktionelle Anforderungen auf Basis der Analyse 5](#_Toc36794795)

# User Story

Ihre Firma entwickelt und erweitert verschiedene Softwareprodukte für unterschiedliche Kunden. Ihr Chef lädt alle Auszubildenden zu einem Gespräch ein, in dem er seine Überlegungen zur Modernisierung der Räumlichkeiten und Technik in Richtung „SmartCompany“ vorstellt. Sie sollen bei der Modernisierung mitarbeiten und während Ihrer Ausbildung selbständig Hard- und Software für dieses System planen, entwickeln und umsetzen. Die angestrebten Lösungen sollen keine fertigen industriellen Lösungen sein. Es stehen Eigenentwicklung und der damit verbundene Kenntniszuwachs im Vordergrund. Um sich in diese Problematik einzuarbeiten, gibt Ihnen Ihr Chef eine erste Aufgabe zur Optimierung der Temperaturüberwachung der Servertechnik.

## Akzeptanzkriterien:

In der Firma befinden sich drei eingebaute 19"Serverschränke. Die Raumtemperatur wird zum jetzigen Zeitpunkt durch eine zentrale Temperaturmessung und Klimaanlage konstant gehalten. Durch überhöhte Temperatur einzelner Geräte kam es in letzter Zeit zu vereinzelten Technikausfällen. Aus diesem Grund soll jeder Serverschrank mit einem Temperatursensor ausgestattet werden. Die Messwerte der Sensoren sollen in einer relationalen Datenbank gespeichert, über verschiedensten Abfragen zur Verfügung gestellt und das Überschreiten einer Maximaltemperatur eines Sensors überwacht werden. Erstellen Sie eine Applikation zur Darstellung, Auswertung und Administration der Temperaturdaten. Aus einem vorangegangenen Projekt existiert folgendes Datenbankschema:



# Anwendungsfälle/Anforderunge (Übersicht):

* mehrere Sensoren liefern in einem bestimmten Zeitabstand Temperaturwerte
* die Temperaturen werden mit dem aktuellen Zeitwert gespeichert
* jeder Sensor besitzt eine spezifische Sensoradresse und befindet sich in einem Serverschrank
* zu jedem Sensor soll der Hersteller aus einer vorgegebenen Liste gespeichert werden
* um das Überschreiten einer Maximaltemperatur zu überwachen, wird jedem Sensor eine spezifische Maximaltemperatur zugeordnet
* die Änderung der Maximaltemperatur dürfen nur eingetragene Benutzer übernehmen - die Benutzer werden mit Name, Anmeldename und Telefonnummer gespeichert
* die Änderungen der Maximaltemperatur eines Sensors soll mit der Sensornummer, dem Benutzer und dem Datum separat in einem log-Bereich protokolliert werden

## Nicht unterstützte Anwendungsfälle

* Keine eigentlich GUI(Grafische Benutzeroberfläche) sondern ein API über Browser

## Allgemeine Anforderungen (nichtfunktional)

* Keine Allgemeinen Anforderungen vorhanden
* Performance
* SicherheitZuverläsigkeit(Nur wenn es nicht eingehalten werden kann)
* Wartbarkeit
* ergonomische Gestaltung der Benutzerschnittstelle

## Funktionale Anforderungen

* Benutzer kann
* Solange diese von anderen Funktionen nicht benötigt werden, sollen alle mit den ausgefilterten verknüpften Elementen wie: „definitions“, „resposnes“ etc. in der Zieldatei nicht übernommen werden
* Alle ausgefilterten Dateien müssen im Json-Format auszugeben sein, für die Nutzung in ReDoc.
* Muss per Konsole auszugeben sein
* Python und diverse packages müssen intalliert sein
* Bereits vorhandene Dateien werden überschrieben

# Analyse

## Aktuelle Umsetzung

* Die Umsetzung erfolgt in Django Framework (Python)

## Ablauf (IPO)

### Input:

* Benutzr kann eingeben von oder ändern oder löschen werten von die datenbank per API
* Benutzer kann Zeigen werten mit viele Sortierung und Filterung oder Suchung Möglischkeiten

### Prozess:

* Die Applikation besteht aus folgenden Schritten:

1. Benutzer oder admin anmelden
2. Wählen eine von die angezeigete Tabellen
3. Nach auswahlen steht die Moglischkeit hinzufügen, editeren, löschen von einen wert

### Output:

* + Rückgabe die neue werte
  + Protokolliert die änderung mit eineة nachricht in Logs tabelle falls, es ist verbunden mit

## Funktionelle Anforderungen auf Basis der Analyse

* Noch keien funktionellen Anforderungen auf Basis der Analyse bekannt.

# Startanweisungen

zum Ausführen benötigt man das Python 3.6.5 und einen MySQL Server wie von XAMPP.

Dann sollten man die Pakete aus der Datei requirements.txt installieren

In Kommandozeile wie folgendes:

.

Legen die Datei requirements.txt in dem Verzeichnis ab, in dem der Befehl ausgeführt wird. Wenn es sich in einem anderen Verzeichnis befindet, geben Sie den Pfad an.

$ pip install -r ./requirements.txt

Erstellen einen Datenbank mit die name „tempverwaltung“

Dann in dem Verzeichnis wo die Datei managy.py stehet führen die Folgende Kommando:

$ Python3 manage.py migrate

$ Python3 manage.py runserver

Jetzt die Applikation läuft direkt auf localhost: <http://127.0.0.1:8000/>