

# **Dokumentation Guestbook**

Semester Projekt Softwareentwurf und Anwendung verteilter Systeme

Florian Geiselhart

Internet der Dinge 3

Sommersemester 2021

Tom Bürkle

# Inhaltsangabe

A) Guestbook, eine Plattform für soziale Interaktion in der Echten Welt, System-Design-Konzept.

B) System Bestandteile Technologische Betrachtung

1) Sniffer (Sensor)

- Technologien Hardware / Software
- Energie Versorgung
- Kommunikation
- Inbetriebnahme

2) Backend (Repo)

- Technologien
- Technisches Fluss-Diagramm

3) Datenbank

- Technologien

4) Frontend (**Repo**)

- Technologien

C) Ausblick - Skalierbarkeit

- Speicherbedarf DB
- Langzeitbetrieb
- Ausfallsicherheit

## A) Guestbook eine Plattform für soziale Interaktion im echten Leben, System-Design Konzept.

Die Ziel der Plattform ist es Nutzern in einer bestimmten Umgebung, in der mehrere Veranstaltungen gleichzeitig stattfinden, oder mehrere Event Locations, wie Bars Sportplätze, Parks oder auch Steges auf Festivals gleichzeitig einladen, auf zu zeigen wie viele Bekannte oder enge Freunde sich dort befinden um so eine Entscheidung fällen zu können ob man lieber zu einer Location gehen möchte auf der sich viele, wenige oder keine Bekannten aufhalten. Hierbei ist es nicht möglich zu sehen wer genau sich dort aufhält, lediglich die Anzahl der Personen werden nach ihrer Beziehung die man zu ihnen hat angezeigt.

Im MVP werden die Funktionen der Mobilen Nutzung ohne Netzanschluss und die Datenverarbeitung der Beziehungen zum Nutzer vernachlässigt. So ist für den MVP die Hardware nicht mit einem Akku ausgestattet. Und die Daten der Profilabhängigkeiten werden nicht gefiltert, die einzige Information die der Nutzer im Frontend erhält ist die Anzahl aller Gäste einer Veranstaltung.

Der Fokus des MVP liegt darauf die Hardware Komponente im Zusammenspiel mit den weiteren System Komponenten zu Testen, ob ein Nutzer von der Hardware erkannt wird, wie diese Information im Backend korrekt verarbeitet wird, und dann weiter an das Frontend gegeben und hier fehlerfrei ausgegeben wird (Durchstich). Weitere Services und Technologien sind im Ansatz für einen Funktionsprototypen auch vorhanden um einen Belastungstest vor zu bereiten.

## B) System Bestandteile Technologische Betrachtung

### 1) Sniffer (Sensor)

Pro Event muss ein Sensor in Betrieb genommen werden. Der Sensor hat im fertigen System die Aufgabe die Location via GPS zu ermitteln um eine Fehleingabe aus zu schließen sollte eine Falsche Adresse für das Event Eingegeben werden. Die Adresse die Beim „Event Erstellen“ nur noch um als Test ausgegeben zu werden, Die Direktionen und die Anzeige des Events auf der Karte werden mit den GPS Daten verarbeitet. Hier sollte für einen Voll umfassenden Prototypen ein Kontrolle service die eingegebene Adresse mit den Gps Daten vergleichen um bei zu großen Abweichungen einen Fehler aus zu geben. Ebenso sollte der Sniffer mit seiner Macadresse in form eines QR-Codes und in Textform auf seinem Gehäuse bedruckt sein um die Inbetriebnahme beim erstellen eines Events zu vereinfachen, oder es Manuell ein zu geben, hierfür ist ein Weiterer Service nötig welche in der Sniffer Collection die Einträge einem Nutzer zuweist, sobald dieser ein Event mit der Passenden Macadresse eingibt, und einen service welcher diese Eingabe mit Hilfe eines QR-Code Scanns übernimmt.

## - Technologien Hardware / Software

Der sniffer für den MVP besteht aus einem ESP8266 Einem Netzteil und einem USB kabel (Hardware). Dem sniffer.ino Sketch welcher das Programm des sniffers darstellt Siehe (Software). Und darin eingebundenen Wifi Manager zur Inbetriebnahme (Software Service).

## - Energie Versorgung

Für den MVP wird der Snigger über das Netzteil an einer Steckdose permanent mit Strom versort. Für einen Vollumfassenden Prototypen wäre für die Mobile Nutzung einen Powerbank mit Ladesteuerung nötig um den Microcontroller zu versorgen und eine Laufzeit von mindestens 12 Stunden zu gewährleisten. In dem Falle müsste noch ein Ein/Aus Schalter eingebaut werden.

## - Kommunikation

Der Sniffer Kommuniziert über eine Http Post Api mit dem Backend. Die Verbindung hierfür wird im Code hergestellt, für einen Vollumfassenden Prototypen müsste diese Verbindung gesichert werden durch einen Verschlüsselung Service. Für eine Mobile Nutzung müsste außerdem Netzwerk von einem dritten gerät ein WiFi Client geöffnet werden.

## - Inbetriebnahme

Bei der Inbetriebnahme gehen wir für einen Prototypen davon aus dass bei der Herstellung der sniffen die Macadresse auf dem Gehäuse abgedruckt ist in den genannten Versionen, die Firmware bereits auf den Microcontroller geflashed und getestet wurde und der Akku mindestens halb voll geladen wurde.

Die Inbetriebnahme erfolgt dann über den Wifi Manager, dafür ist der Sniffen zunächst ein zu Schalten (Im MVP ans Netz an zu schließen), dann mit dem Handy bei der WLAN suche den Client „**AutoConnect AP**“ aus zu wählen. Anschließend kann im Browser Dann mit dem selben gerät ein Dialog Fenster geöffnet werden unter der Adresse: „**192.168.4.1**“, hier kann dann unter dem ersten Menüpunkt „**Configure WiFi**“ die SSID und das Passwort für das Gewünschte Netzwerk ein zu geben in dem der Sniffen die Verbindung zum Web aufbauen soll. Nach einer Minute sollte der Sniffen dann mit im gewünschten Netzwerk angemeldet sein und die ersten Daten im Backend ankommen.

## B) System Bestandteile Technologische Betrachtung

### 2) Backend

Das Backend wird für eine Vollumfassenden Prototypen von einem Online Service (Serverfarm) gehostet. Es Verarbeitet Sämtliche Daten die von der Hardware dem Frontend und der Datenbank ein und aus gehen, und ist Schnittstelle um Informationen über die Komponenten aus zu lesen. Hierfür müssen auch in den Komponenten Status Protokolle angefertigt und an das Backend gesendet werden.

Für den MVP hat das backende die Aufgabe die gefundenen Sensor Daten an zu nehmen, in den Datenbank Collections nach den passenden Einträgen zu suchen und diese Ab zu gleichen, dann die Erhaltenen Daten Neu an zu ordnen und Einträge Wie Events um die Anzahl der Gäste zu erweitern. Ebenso die Informationen aus dem Frontende, zB. Beim Erstellen eines Profils oder dem Anlegen eines Events, correct und den jeweiligen Datenbank Collections Ab zu legen.

#### - Technologien

Node.js:	Node ist eine Plattform übergreifende Entwicklungsumgebung Für JavaScript Programme („Backends“ und scripts).
Mongoose:	Datenbank Manager, Mongoose ist ein Plugin zu Verwaltung einer Datenbank auf Mongodb
Axios:	Axios ist ein HTTP-Client über den das Backend und das Frontend Kommunizieren über die jeweiligen API Endpoints.
Express:	Express ist ein Framework für Node.js, um Anwendungen zu erstellen.
body-parserr:	Body-Parker ermöglicht die Verwendung von Middleware, für die Kommunikation.
CORS:	Cross-Origin Resource Sharing Ist ein Protokoll um Ressourcen übergreifende Kommunikation von Plattformen zu ermöglichen ohne dass die vom Browser vorgeschrieben Eigenschaften für die Herstellung der Verbindung dafür verwendet werden müssen. Somit kann man das Backend und das Frontend auf verschiedenen Servern einfach Via deren URL gesichert verbinden.
bcrypt:	Library die Passwörter verschlüsselt, und bei anfrage die Originale vergleicht, ohne dass der admin zugriff hat.
jsonwebtoken:	Library um eine verschlüsselte Form eines Nutzers zu authentifizieren.

Sämtliche dieser Technologien müssen vor der Lokalen Inbetriebnahme des Backends installiert sein.

#### - Technisches Fluss-Diagramm

liegt im Backend Repository unter [GuesbookProzess.jpg](#)

## B) System Bestandteile Technologische Betrachtung

### 3) Datenbank

Die für den MVP wird der Service der MongoDB Atlas suit verwendet. Diese hat ausreichend Kapazität um einen Umfangreichen Test mit mehreren hundertern Usern und tausenden Anfragen zu verarbeiten. Hier können allerdings nur einfach Datentypen wie String Float Number usw. Abgelegt werden.

### 4) Frontend

Das Frontend ist für den MVP vor allem auf die Event Funktion Reduziert, Medien Werden hier noch nicht berücksichtigt.

#### - Technologien

Vue.js	Vue.js ist ein übergreifendes Framework und Development umgebung zfür Web-Anwendungen.
vuex	vuex ist ein Datenspeicher mit dem Komponenten Global Daten und States austauschen können.
bootstrap:	Boosträp ist ein Framework komponenten Styles und CSS Klassen.
axios:	Axios Erstellt einen HTTP-Client über den das Frontend mit dem Backende über die entsprechenden APIs kommuniziert.
MapBox:	Mapbox ist ein Service zum stylen und erstellen von Karten, mit den dazugehörigen Funktionen.

### C) Ausblick - Skalierbarkeit

#### - **Speicherbedarf DB**

Sobald aber sämtliche Funktionen wie geteilte Medien mit implementiert werden sollte, müsste man auf einen Service wie Google Cloud Storage wechseln der große Datenmengen verwalten kann.

#### - **Langzeitbetrieb**

Seed-Dateien die in der Datenbank für den Prototypen wichtig sind um Funktionen zu testen und weitere unnütze Daten müssten aus dem System entfernt sein. Ebenso sollte automatische gesetzte Intervalle für die Dauer von Veranstaltungen funktionieren und die.

#### - **Ausfallsicherheit**

Ein Dienstleister müsste die Stabilität des Systems stets überprüfen.

Hierzu gehört die Erreichbarkeit des Systems auf globaler Ebene zugriffsspitzen auf das Front und Backend.

Für Großevents bei dem Tausende User auf einem Event anwesend sind sollte ein angepasster Sniffer entwickelt werden, mit mehreren Einheiten um den Ausfall eines einzelnen abfangen zu können.