bauoll

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
| Projektdokumentation | |
| Leistungsbeurteilung LB 1 | |
|  |  |
|  |  |
| Modul | 431 |
| Eingereicht von | Gruppe A |
| Projektthema | Temperatursensoren an der gibb IET |
| Eingereicht bei | Lehrperson |
| Datum | 29. August 2023 |

Dokumentinformation

|  |  |
| --- | --- |
| Projektleiter | Levyn Schneider |
| Projektmitglied  Projektmitglied  Diktator | |  |  | | --- | --- | | Meer  Schär | David  Josia | | Navagan | Navaajanan | |  |  | |  |  | |

Änderungsverzeichnis:

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Bemerkung / Änderung | Autor |
| 29.08.2023 | 1 | Dokumentation erstellt | Urs Dummermuth |
| 29.08.2023 | 1.1 | Variantenunterscheid, Rollen | Navaajanan Navagan |
| 29.08.2023 | 1.2 | Projektauftrag eingetragen | Levyn Schneider |
| 29.08.2023 | 1.3 | Soll/Ist-Analyse | Josia Schär |
| 29.08.2023 | 1.4 | Projektstrukturplan | David Meer |
| 29.08.2023 | 1.5 | Lösungsvariante eingefügt | Levyn Schneider |
| 05.09.2023 | 1.6 | Ziele | Navaajanan Navagan |
| 05.09.2023 | 1.4.1 | Projektstrukturplan | David Meer, Levyn Schneider |
| 05.09.2023 | 1.7 | Zeitplan | Josia Schär |
| 05.09.2023 | 1.8 | Materialliste | David Meer |
| 05.09.2023 | 1.9 | Blackbox | Navaajanan Navagan |
| 05.09.2023 | 1.10 | Whitebox | Levyn Schneider |
| 12.09.2023 | 2.1 | Bauplan/Installation | Navaajanan Navagan |
| 12.09.2023 | 2.2 | Whitebox, Blackbox und Komponentendiagram | Levyn Schneider |
| 19.09.2023 | 2.3 | Domain/Testkonzept | NN |
| 19.09.2023 | 2.4 | Webseite | LS |
| 19.09.2023 | 2.5 | Materiellste | JS |
| 19.09.2023 | 2.6 | Webdesign erstellt | JS/DM |
| 17.10.2023 | 3.1 | Arbeitsjournal/Dokumentation | NN |
| 17.10.2023 | 3.2 | Impressum/Bilder für Dokumentation | DM |
| ￼17.10.2023 | ￼3.3 | ￼Zeitplan aktualisiert/ Kahoot frage | ￼JS |
| ￼17.10.2023 | ￼3.4 | ￼Testscript/ Sensoren mit Hotspot von NN erfolgreich getestet | ￼LS |
| 24.10.2023 | 3.5 | Arbeitsjournal | NN |
| 24.10.2023 | 3.6 |  |  |
| 24.10.2023 | 3.7 |  |  |
| 24.10.2023 | 3.8 |  |  |

Inhaltsverzeichnis

[1 Studie (Projektauftrag) 16](#_Toc395153595)

[1.1 Ausgangslage 16](#_Toc91904624)

[1.2 SOLL-/IST-Analyse 16](#_Toc146579701)

[1.3 Verfeinerte Ziele mit Anforderungen & Rahmenbedingungen 17](#_Toc610575196)

[1.4 Variantenentscheid 18](#_Toc1184266766)

[1.5 Projektstrukturplan (allenfalls Printscreen aus Tool) 18](#_Toc781985814)

[1.6 Zeitplan 19](#_Toc1029702113)

[1.7 Aufwand & Kosten 19](#_Toc1994297544)

[2 Zusammenfassung 21](#_Toc1793094925)

[3 Initialisierungsphase 22](#_Toc1060768860)

[3.1 Ziele **Fehler! Textmarke nicht definiert.**](#_Toc1797107533)

[3.2 Rahmenbedingungen: 22](#_Toc2106815677)

[3.3 Anforderungen an das System: 22](#_Toc1418577855)

[3.4 Organisation 22](#_Toc1559646308)

[3.4.1 Rollen 22](#_Toc1552367814)

[3.5 Zeitplan 23](#_Toc1677756597)

[3.6 Arbeitspakete 23](#_Toc1584143594)

[3.7 Lösungsvarianten 23](#_Toc796428497)

[4 Konzeptphase 24](#_Toc1296320657)

[4.1 Blackbox 24](#_Toc727338061)

[4.2 Whitebox 24](#_Toc1189657159)

[4.2.1 Komponentendiagramm 25](#_Toc46255796)

[4.3 Bauplan / Installation 25](#_Toc1354229951)

[4.4 Materialliste 27](#_Toc1773827573)

[4.5Testkonzept 27](#_Toc1020322889)

[4.6Testprotokoll 29](#_Toc912988225)

[5 Realisierungsphase 29](#_Toc539074344)

[5.1 Entwicklung / Aufbau der Webseite 29](#_Toc430985992)

[5.2 Administration 29](#_Toc1105363900)

[5.3 Benutzerverwaltung 29](#_Toc1235132105)

[5.4 Testen 29](#_Toc3192838)

[6 Einführungsphase 31](#_Toc1126210779)

[6.1 Präsentation 31](#_Toc460398963)

[6.2 Schulung 31](#_Toc1275362327)

[6.3 Abnahme und Einführung 31](#_Toc631821083)

[7 Arbeitsjournal / Pendenzen 32](#_Toc651048831)

[8 Reflexion 34](#_Toc940154331)

[8.1 Projektmitglied 1: Funktion X 34](#_Toc2112500040)

[8.2 Projektmitglied 2 35](#_Toc1632839625)

[8.3 Projektmitglied 3 35](#_Toc896878697)

[8.4 Projektmitglied 4 35](#_Toc1166023922)

[9 Anhang 36](#_Toc1737519980)

[9.1 Abbildungen 36](#_Toc1321335678)

[9.2 Quellenverzeichnis 36](#_Toc2108870486)

Vorwort

Dokumentation "temperatursensoren-gibb.ch"

Dies ist die Dokumentation der Website "temperatursensoren-gibb.ch", entwickelt von Levyn Schneider mit Hilfe von David Meer.   
  
Website

Die Website ist basierend auf Next.js und React.js Framework. Auf der Hauptseite (Landingpage) werden alle Sensoren, welche aktuell im Einsatz sind aufgelistet. Dies funktioniert mithilfe der Datenbank, welche später erwähnt wird.

Ein Bild, das Text, Screenshot, Software, Computer enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Computer enthält.

Automatisch generierte Beschreibung

Auf den Einzelnen Räumen (Sensoren), wird die Temperatur und Feuchtigkeit genauer angezeigt mit zusätzlichen Informationen, wann z.B. die letzte Messung stattgefunden hat.

Ein Bild, das Text, Screenshot, Software, Betriebssystem enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Karte Menü enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Computer enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia enthält.

Automatisch generierte Beschreibung

Datenbank

Als Datenbank wurde wie beschlossen Supabase verwendet. Supabase ist eine JavaScript Datenbank welche einfache Funktionalität in Websites ermöglicht. Die Datenbank Struktur innerhalb der Datenbank ist mit zwei Tabellen aufgebaut: rooms und data.

In der "rooms" Tabelle werden alle Sensoren abgespeichert mit dem Namen des Raums, welcher der Sensor sich befindet und welche Etage. Die Website benutzt wie vielleicht schon bemerkt die einzelnen Namen der Sensoren / Räume, um zu validieren, dass der Link z.B. /raum/ie-ie215 korrekt ist.

Ein Bild, das Text, Screenshot, Schwarz enthält.

Automatisch generierte Beschreibung

In der "data" Tabelle werden alle Messungen der einzelnen Sensoren gespeichert. Die Sensoren messen jede Halbestunde. Da es aktuell zwei Sensoren gibt und die Messungen 7 tagelang bleiben, gibt es gut mal bis zu 672 einzelne Messungen in der Tabelle.   
  
Die Tabelle ist aufgebaut mit folgenden Attributen: timestamp, room\_id, temperature und humidity. Timestamp wird genutzt, um zu speichern, wann die Messung stattgefunden hat.  
room\_id wird als Fremdschlüssel verwendet um zum Sensor / Raum zu verlinken. Dann temperature und humidity sind die eigentlichen Messdaten des Sensors.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Raspberry Pi

Das Raspberry Pi hat die Funktion den Sensoren jede Halbestunde eine Anfrage zu senden für die Temperatur und Luftfeuchtigkeit. Das Raspberry Pi ist auf einem einfachen Linux Distro konfiguriert mit einem Python File welche ununterbrochen läuft. Um dem RPI, Selbstständigkeit zu vergeben, wird das Python File sofort nach Start ausgeführt, somit muss man das File nicht selbst starten.

Das Python File wurde speziell so entwickelt, dass es jede Halbestunde in der "echten Zeit" ausführt und nicht, ab dann wann der RPI gestartet wurde, resp. das File. Das Python File reguliert auch das Löschen der Daten nach 7 Tagen in der Datenbank.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung



Ein Bild, das Text, Software, Multimedia-Software, Computer enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Abkürzungsverzeichnis

|  |  |
| --- | --- |
| RPI | Raspberry PI |
| GPL | **GNU General Public License** |
| ￼TS | ￼**Temperatursensoren** |

# Studie (Projektauftrag)

## Ausgangslage

Wir haben bereits zwei Temperatursensoren für einen Prototypen bestellt, welche zwischen 27. August und 4. September ankommen. Ebenfalls haben wir ein gewissen Grundwissen in Skripten, Netzwerktechniken und Webentwicklung. Im Internet haben wir gewisse Inspirationen gefunden, wie wir das Projekt umsetzten könnten.   
  
Die Idee entstand aus einem Chat-GPT Vorschlag für eine IOT-Wetterstation. Der Grundbaustein für diese Idee entstand aus einem Witz, da die Gibb für ihre warme Zimmertemperaturen bekannt ist.

## SOLL-/IST-Analyse

› Material / Ressourcen  
 ***Ist:***Wir haben 2 Temperaturmessgeräte bestellt. Raspberry Pi von Levyn Schneider, denn wir für das Projekt benutzen können.

***Soll:*** *Unser Temperatur Messgeräte müssen noch geliefert werden. Levyn muss sein Raspberry Pi von zu Hause mitbringen.*

› Skills  
 ***Ist:***Wir haben einige Kenntnisse über das Programmieren von Webseiten, Kenntnisse über verschiedene Designs und wir wissen, wie man die Temperatursensoren mit dem Netzwerk verbindet.

***Soll:*** *Wir müssen das Knowhow über das Programmieren und Design noch verbessern.*

› Interessen

***Ist:*** Wir möchten uns im Webseiten programmieren verbessern. Wir möchten den andern Lernenden helfen sich auf der Raumtemperatur vorzubereiten. Wir möchten unser Teamwork Fähigkeit verbessern.

***Soll*:** Wir müssen auf YT oder anderen Informationsquellen, Infos zum Programmieren holen. Wir müssen als Gruppe zusammenarbeiten und nicht als vier Individuen in einer Gruppe.

› Bestehende Elemente aus dem Umfeld

***Ist***: im Moment ist es heiss in der IET-Abteilung und weshalb wir uns auf die Raumtemperatur vorbereiten wollen.

***Soll:*** Wir möchten anderen Lernenden helfen sich auf dieausgeprägtenTemperaturen vorzubereiten.

## Ziele mit Anforderungen & Rahmenbedingungen

**Ziel 1:**

Die Lernenden und die Lehrer von der Gibb IET sollen von unserer Webseite herauslesen können, was die aktuellen Raumtemperaturen sind. Diese werden durch Implementierung eines zuverlässigen Temperaturüberwachungssystem auf die Webseite dargestellt.

**Anforderung zum Ziel 1:**

Dass die Lernenden von der Website profitieren können, indem sie sich auf den Raum vorbereiten können (passende Kleidung, evtl. Ventilator, genügend Flüssigkeit).

**Ziel 2:**

Eine Webseite, welche aktuelle Temperaturen und Luftfeuchtigkeit von den Räumen anzeigt.

**Anforderung zum Ziel 2:**

Website, um die aktuellen Temperaturen und Luftfeuchtigkeit von den Räumen and der Gibb-IET anzuzeigen.

**Ziel 3:**

Die Daten auf der Webseite protokolieren, um im späteren Zeitpunkt anzuzeigen, wie die Temperatur verläuft, um Massnahmen für die kommenden Jahren zu unternehmen. Die Daten werden grafisch aufgezeigt.

**Anforderung zum Ziel 3**:

Protokolieren der Daten, um anzuzeigen wie die Temperatur verläuft, um Massnahmen für die kommenden Jahren zu unternehmen.

## Variantenentscheid

**Variante 1:** Wir kaufen Temperaturmessgeräte ein und bauen diese in allen IET-Zimmern ein. Danach erstellen wir einen Teamskanal. In diesem Teamskanal sind die Temperaturen der einzelnen Zimmern ersichtlich. Die Lehrer und die Lernenden können im Teamskanal die eizelnen Zimmern auswählen und sehen dort die Temperaturen für das ausgewählte Zimmer.

Vorteile:

* Wir benötigen weniger Arbeitszeit
* Der gesamte Auftrag ist weniger aufwendig

Nachteile:

* Für die User ist die Verwendung von Webseiten eventuell einfacher.

**Variante 2:** Wir kaufen Temperaturmessgeräte ein und bauen diese in allen IET-Zimmern ein. Des Weiteren erstellen wir eine eigene Webseite, wo wir auch die Temperaturen von den vergangen Tagen lesen können. Dies soll, denn Lehrpersonen wie auch den Studierenden aufzeigen, dass es immer wärmen oder kälter wird.

Vorteile:

* Wir lernen somit eine Webseite Programmieren.

Nachteile:

* Wir lernen nicht, wie wir ein Temperatur Messgerät bauen können.

**Variante 3:** Wir arbeiten mit eigenen Temperaturmessgeräte, dazu benötigen wir ein Microcontroller (Rasperry Pi), Breadbord, ein Jumper-Kabel, Display und natürlich eine Stromversorgungsquelle. Diese Daten werden wir via Webseite, allen zur Verfügung stellen.

Vorteile:

* Wir arbeiten hier mit mehr Hardware.

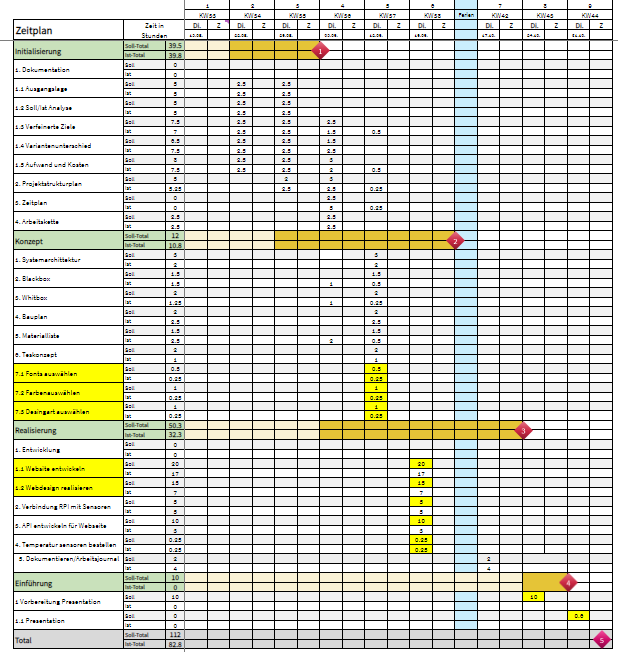
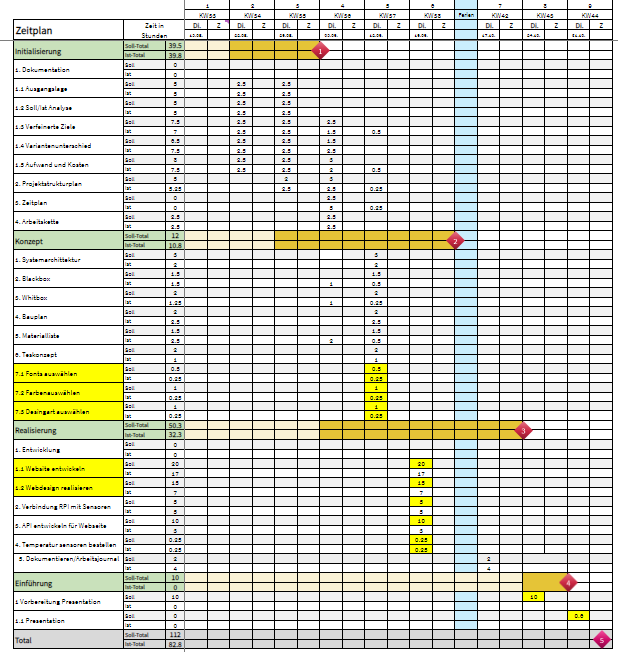
Nachteile:

* Die Teile für die Temperaturmessegeräte kosten mehr Geld als eine fertige zu kaufen.
* Wir brauchen viel Zeit, um die Temperaturmessgeräte herzustellen, weshalb wir nicht genügend Zeit haben, um die Webseite selber zu programmieren.

## Projektstrukturplan (allenfalls Printscreen aus Tool)

<https://miro.com/app/board/uXjVMoLtpAk=/?share_link_id=285771068323>

## Zeitplan



## Aufwand & Kosten

An diesem Projekt arbeiten 4 Personen. Der Aufwand schätzen wir auf 35 Arbeitsstunden insgesamt für ungefähr alle 4 Personen. Der Stundenansatz berechnen wir auf CHF 120 pro Person.  
  
CHF 36.90 für zwei Temperaturmessgeräte als Prototyp + 35h pro Person CHF 120.  
Total: CHF 16’836.90

* externe kosten:

**Produktkosten:** 150 Fr  
**Transportkosten:** 0 Fr**.**  
**Lizenz- und Zertifizierungskosten:** aktuell 0 Fr.  
**Marketingausgaben:** 0 Fr.  
**Versicherungskosten:**   
**Steuern und Abgaben:**

* interne kosten:

**Personalkosten:** An diesem Projekt arbeiten 4 Personen. Der Aufwand schätzen wir auf 35 Arbeitsstunden insgesamt für ungefähr alle 4 Personen. Der Stundenansatz berechnen wir auf CHF 120 pro Person. Das entspricht 16'800 Fr  
**Materialkosten:** 36.90 Fr.  
**Versandkosten**: 0 Fr   
**Verwaltungskosten:**  
**Supportkosten**: Der Preis für die IT-Servicestunde liegt bei 60 Fr pro Stunde  
**Sonstige Kosten:** aktuell keine Vorhanden

# Zusammenfassung

# Initialisierungsphase

## Rahmenbedingungen:

1. Unsere Temperaturmessgeräte müssen so deponiert werden, das unbefugte unsere Messgeräte verstellen/fälschen.
2. Wir müssen sicherstellen das unbefugte keinen Zugriff auf unsere Datenbank oder auf Raspberry Pi haben.
3. Wir können die Systemdefekte/Fehler nicht berücksichtigen

## Anforderungen an das System:

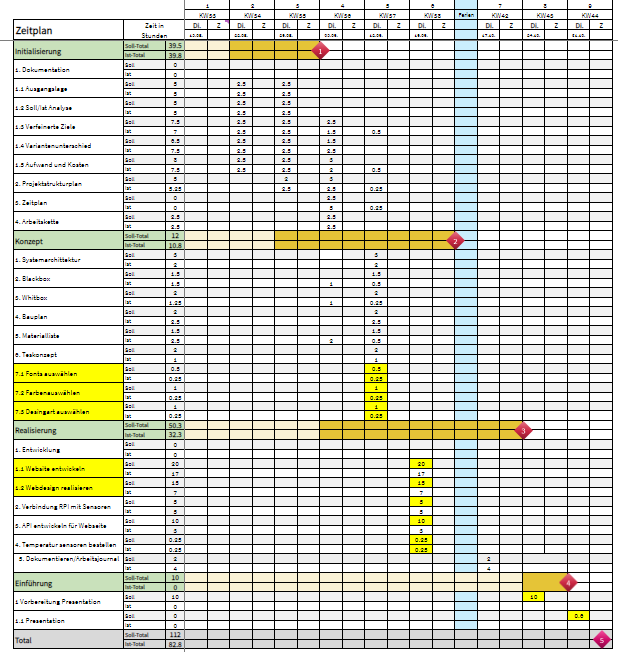
* Das System sollte so konfiguriert sein, dass bei Fehlfunktionen oder Sensoren ausfällen eine Benachrichtigung auf der Webseite anzeigt.
* Die Temperaturdaten müssen mit einer Genauigkeit von mindestens +-0,5° erfasst und auf der Webseite angezeigt werde.

## Organisation

### Rollen

|  |
| --- |
| Auftraggeber:   * Simon Witter   Projektleiter:   * Levyn Schneider   Projektmitglieder:   * David Meer, Josia Schär   Dokumentierer:   * Navaajanan Navagan |

## Zeitplan



## Arbeitspakete

Unter diesem Link findet man unsere Arbeitspakete:   
<https://miro.com/app/board/uXjVMoLtpAk=/?share_link_id=285771068323>

## Lösungsvarianten

Die Lösung, die wir gewählt haben, ist, dass wir einsatzbereite Temperaturmessgeräte kaufen und bauen diese in allen IET-Zimmern ein. Des Weiteren erstellen wir eine eigene Webseite, wo wir die aktuellen Temperaturen an der gibb IET sehen aber auch die Temperaturen von den vergangen Tagen lesen können. Dies soll, denn Lehrpersonen wie auch den Studierenden aufzeigen, wie die aktuelle Temperatur ist oder wie auch dass es immer wärmer oder kälter wird.

# Konzeptphase

## Blackbox

Unsere Webseite soll einfach zum Bedienen sein. Auf der Webseite sollten die Lernenden die einzelnen Räume auswählen können und dort die aktuelle Raumtemperatur lesen können.

In der Navigationsleiste kann man zu jedem Raum, an der gibb-IET die Temperatur Verläufe ansehen.

Unsere Temperaturen werden wir voraussichtlich auf den Schränken platzieren, damit sie weniger Aufmerksamkeit erregen.  
  
Die Schriftarten der Website sind bestehenden aus: Poppins, für Title Elemente und Nunito Sans für diversere kleinere Texte.  
  
Das Farbdesign der Website sind bestehenden aus:  
Primary: #F7FAFF  
Secondary: #CDE8FF  
Accent: #C9FFBE  
Text: #202020  
  
Die Designart soll aus einem minimalistischen, räumlichen, aber auch übersichtlichem Design bestehen.  
  
Mockup:

Ein Bild, das Screenshot, Text, Software, Webseite enthält.

Automatisch generierte Beschreibung

## Whitebox

Das System ist aufgebaut auf mehreren erweiterbaren Temperatursensoren. Die Temperatursensoren werden über die Tuya "SmartLife App" initialisiert und mit dem Netzwerk verbunden.   
  
Die Temperatursensordaten werden über das Raspberry PI mit Hilfe des CLI-Betriebssystems "Home Assistent" über eine API auf die Website übermittelt.   
  
Das Frontend der Website ist auf React basiert und das Backend basierend auf Node.js.   
  
Der Temperaturverlauf wird stündlich auf die "Supabase" Datenbank geschrieben. Die Daten werden nach einer Woche wieder gelöscht, um Überlastung und Speicher zu sparen.  
  
Bei einem Request von einem User ruft die Website die aktuellen Daten von der Datenbank ab.

### Komponentendiagramm

|  |  |
| --- | --- |
| Name | Funktion |
| Navigation | Rendered die Navigation mit den einzelnen Punkten |
| Footer | Rendered den Footer |
| Sensor | Eine Komponente, welcher für jeden Sensor die Daten abruft und eine Seite generiert |
| Räume | Komponente für die Räume. |
| Logik | Logik Komponente für generelle Logik |

## Bauplan / Installation

**Module;**

**Temperatursensoren**: Dieses Modul umfasst die Hardwarekomponenten, welche in den Klassenzimmern installiert werden, um die Raumtemperatur zu erfassen.

**Raspberry P**i: Das Raspberry Pi dient als Blackbox, um Daten von den Temperatursensoren zu sammeln und diese in die Datenbank zu übertragen.

**Datenbank**: Die Datenbank dient damit die Temperaturdaten, welche vom Raspberry Pi gesammelt werden und auf der Webseite anzuzeigen werden.

**Webseite**: Das Webseitenmodul dient für die Benutzer, um die aktuellen Rauntemperaturen anzuzeigen.

**Domain**: Dieses Modul umfasst die Bereitstellung einer Webseite. Damit User und Benutzer der Webseite den entsprechenden Zugriff haben.

**externe Schnittstellen;**

**Lehrperson**: Der Lehrperson ist eine externe Schnittstelle, weil er das Projekt Genehmigen oder nicht genehmigen kann.

**Die IET-Räume (Eventuell Schulleiter)**: Der Schulleiter muss und eine Genehmigung erteilen, um die Temperatursensoren in den Klassenzimmern zu teilen.

**Lehrkräfte** **und** **Schüler**: Lehrerpersonen, Schüler, & Hauswärter/innen sind die Endnutzer der Website, welche die Raumtemperaturen anzuzeigen.

**Temu**: Temu ist Onlineshop, welche uns die Temperatursensoren zuliefert, bei einer Bestellung.

**Webhosting-Anbieter**: Die Website wird auf einem Webhosting-Server gehostet, der eine externe Schnittstelle darstellt. Voraussichtlich benutzen wir Netlify als Webhoster.

**Domain**: DNS-Server sind externe Schnittstellen, weil sie die Zuordnung von Domainnamen zu IP-Adressen bereitstellen. Diese sind für den Zugriff der Domain im Internet entscheidend.

**Interne Schnittstellen;**

**Kommunikationsschnittstelle zwischen Temperatursensoren, Raspberry Pi:** Die erfassten Daten von den Sensoren werden durch Raspberry P auf die Webseite übertragen.

**Supabase**: Die erfassten Temperaturdaten werden in der Supabase gespeichert.

**Webanwendung**: Die Schnittstelle zwischen dem Überwachungssystem und der Website ermöglicht die Übertragung und Anzeige der Temperaturdaten auf der Website.

**Systemgrenze;**   
Wir können die allgemeine Stromversorgung und die Netzwerkinfrastrukturen der Schule nicht beeinflussen.

Wir können nicht sicherstellen das unsere Temperaturgeräte manipuliert werden.

Das Webhosting über Netlify, bietet eine Uptime von 99.98%. Allerdings gibt es keine Garantie beim Starter oder Pro Plan.

Unsere Datenbank, Supabase, hat ein Uptime von 99.9%.

**Domain**: Da wir eine Datenbank (Supabase) verwenden, besteht eine Schnittstelle zwischen dem Webserver und dem Datenbankserver.

**Software;**Informationen finden Sie hier:~ 4.2.1 Komponentendiagram

**Hardware;**Informationen finden Sie hier:~ 4.1 Blackbox  
~ 4.2 Whitebox

## 4.4 Materialliste

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Anzahl** | **Name** | **Link** | **Lizenz** | **Preis** |  | **Bestellt/Geliefert** |
| **2** | Tuya WiFi Temperatur Feuchtigkeitssensor | <https://bit.ly/temperatursensor_gibb> | - | 18.99 CHF pro Stück |  | Geliefert |
| **1** | Raspberry PI 2 | https://bit.ly/raspberrypi2-gibb | - | 40 CHF |  | Schon vorhanden |
| **-** | SmartLife | [bit.ly/smartlifegibb](https://apps.apple.com/ch/app/smart-life-smart-living/id1115101477) | GNU General Public License | - |  |  |
| **-** | Home Assistent | <https://www.home-assistant.io/> | Apache License (free and open source) | - |  |  |
| **-** | LocalTuya | <https://github.com/rospogrigio/localtuya> | GNU General Public License v.3.0 | - |  |  |
| **-** | VSCode | <https://code.visualstudio.com/> | Standard MIT license | - |  |  |
| **-** | Node.js | <https://nodejs.org/de> | Permissive MIT license | - |  |  |
| **-** | React JS | <https://react.dev/> | MIT Licensed | - |  |  |

## 4.5Testkonzept

|  |  |  |  |
| --- | --- | --- | --- |
| Testfall Nummer | Name | Vorgehen | Erwartetes Resultat |
| 1 | Aufruf | Der Benutzer ruft die Website auf. | Die Website öffnet sich. |
| 2 | Raum Sub Page | Der Benutzer klickt auf einen Raum. | Der Raum öffnet sich auf der Website und die Temperatur wird angezeigt. |
| 3 | Sensorverlauf Analytik | Der Benutzer benötigt, die Temperatur-daten der vergangenen Wochen und öffnet die Webseite. | Es werden ihm/ihr die aktuellen Daten anzeigt und oben in der Mitte kann man mit einem Klick auf den Pfeil die letzten Daten einsehen. |
| 4 | Clouddaten | Wir öffnen die Tuya Webseite. | Wir können die Sensoren- Daten von der Cloud ansehen |
| 5 | Sensor hinzufügen | Sensor in der App "Smartlife" hinzufügen. Verbindung hinzufügen in der Tuya IOT Cloud. | In der API-Rückgabe sollte der neue Sensor inkl. die Alten vorhanden sein. |
| 6 | Sensor löschen | Sensor von der App "Smartlife" entfernen. Sensor von der Tuya IOT Cloud löschen. | In der API-Rückgabe sollte der Sensor nicht mehr vorhanden sein, und gelöscht sein in der Datenbank. |

## 4.6Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| Testfall Nummer | Name | Fazit |  |
| 1 | Aufruf | Dies hat erfolgreich funktioniert. Die Testperson kam wie erwartet auf unsere Webseite. |  |
| 2 | Raum Sub Page | Der Raum öffnete sich und man konnte die Temperatur lesen. |  |
| 3 | Sensorverlauf Analytik | Die Testperson konnte die Daten von der letzten Woche nachsehen. |  |
| 4 | Clouddaten | Ein Interner Testperson konnte bestätigen, die Sensoren- Daten von der Cloud ansehen |  |
| 5 | Sensor hinzufügen | Dieser Test war erfolgreich |  |
| 6 | Sensor löschen | Sensor löschen war erfolgreich |  |

# Realisierungsphase

## Entwicklung / Aufbau der Webseite

## Administration

## Benutzerverwaltung

## Testen

Checkliste

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Einführungsphase

## Präsentation

## Schulung

## Abnahme und Einführung

# Arbeitsjournal / Pendenzen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| KW | Zeit | Kürzel | Zielsetzung / Tätigkeit | Erfüllungsgrad Positives / Negatives | Reflexion/Fazit |
| 38 | 12:15 | LS | Das Ziel ist es, ein Webdesign und einer Webseite zu erstellen.  Mein Tagesziel ist es eine Webseite mit next und react zu kreieren. | Ich habe mein Ziel erreicht. Dabei hatte ich Probleme bei der Internetverbindung mit den TS. Dies habe ich anschliessend mit dem Hotspot-verbindung gelöst. Was mir auch Schwierigkeiten verursacht hatte, war die Datenbankverbindung zwischen TS und der Webseite. Das Problem dabei war, dass ich kein Recht zum Zugriff auf die Datenbank hatte, dann habe ich mir die Richtlinien verleiht und konnte dann weiterarbeiten. | Ich habe viel Zeit damit verbracht auf die Datenbankverbindung zuzugreifen, ohne mir den Zugriff zu gewähren. Das nächste Mal werde ich das schon von Anfang an machen. Ich war auch froh, meiner Gruppe über den aktuellen Stand zu erklären. |
| 38 | 12:00 | DM | Das Ziel ist es, ein Webdesign und eine Webseite zu erstellen.  Mein Ziel für heute ist es mit JS eine simples und schönes Webdesign zu erstellen. | Das Webdesign haben wir zuerst auf wix.com versucht zu erstellen und waren damit unzufrieden, wegen des eingeschränkten Designfreiheit und wir haben gelesen, dass bei Wix.com erstellte Webseiten, Werbung von Wix.com erhalten, was unprofessionell ist. Wir haben und dann für figma.com entschieden und dies war einfach zu bedienen und wir konnten damit unseren Webdesign erstellen. | Das nächste Mal werden wir mehr Zeit in die Vorarbeit investieren. Da wir die Webseite erstellt haben, war es für uns klar, wo wir die TS-Daten sieht. Wir mussten aber die Webseite Benutzer freundlich und einfach gestalten.  Im Großen und Ganzen bin ich froh, dass das Webdesign mir meinen Teamkameraden gefallen hat. |
| 38 | 12:15 | JS | Das Ziel ist es, ein Webdesign und eine Webseite zu erstellen. | Wie oben bei DM schon erwähnt, haben wir bei figma.com weiterarbeiten können. DS und ich, wir beide waren uns schnell einig, dass wir 2 Grafen: 1 für Temperatur und 1 für die Luftfeuchtigkeit erstellen. Die Farben haben wir auch neutral gestalten. Problem dabei hatten wir nicht. Wir haben unser Design an LS und NN gezeigt und sie waren damit auch zufrieden. | Ich bin froh, dass unser Webdesign einfach und freundlich darstellen konnten. Für weitere Webseiten werde ich auf figma.com zurückgreifen. |
| 38 | 12:15 | NN | Angeschaut wer wann Ferien hat, und besprochen wie man erreichbar ist, wenn es brennt.  Mein Tagesziel für heute ist es, den Arbeitsjournal fertig zu schreiben. Und den anderen so gut wie möglich zu helfen. | Wir haben alle in verschiedenen KW-Ferien/ wir sind nicht alle gezeitigt erreichbar.  LS habe ich geholfen seine Probleme zu analysieren wie z.B. Datenbankverbindung und habe mit ihm nach möglichen Lösungen gesucht. LS und Ich, wir waren auch sehr zufrieden mit dem simplen Design von DM und JS. | Via WhatsApp sind wir am besten erreichbar.  Ich hatte die Möglichkeit in beide arbeiten reinzuschauen und mir Notizen zu machen für das Dokumentieren. Das nächste Mal beginne ich früher mit dem Arbeitsjournal damit ich am Ende nur noch die jeweilige Reflexion schreiben kann. |
| 42 | 12:00 | LS | Mein Ziel für heute ist es, die Webseite fertig zu coden. | Ich habe die Webseite mit JS fertig gecodet und habe mit DM das Impressum fertig geschrieben. Ich bin sehr überrascht von meinen Ergebnissen, weil sie sehr positiv ausgefallen sind. | Ich bin zufrieden mit meiner Arbeit heute. Ich konnte heute durchgehend problemlos fertig coden. |
| 42 | 12:00 | DM | Mein Tagesziel für heute ist es, das Impressum fertig zu schreiben. | Ich habe mich informiert, wie man ein Impressum schreibt und das habe ich auch geschrieben. | Ich bin auch sehr zufrieden, da mein Impressum mir gelungen ist und alle vom Team einverstanden sind. |
| 42 | 12:00 | JS | Ich möchte heure die Webseite fertig gecodet haben mit LS. | Ich habe mit LS die Webseite fertig gecodet und den Zeitplan aktualisiert. | Ich bin auch zufrieden. Ich habe gemerkt, dass ich noch einiges zum Lernen habe, was das coden angeht. In der Zukunft werde ich mehr daran arbeiten, um im Team mehr helfen zu können. |
| ￼42 | 12:00 | NN | Ich muss heute die Dokumentation fertig schreiben. | Ich habe mich informiert, wer was macht und damit die Dokumentation fertig geschrieben, da ich der Dokumentierer. | Ich bin froh, dass ich das Team nicht viel unterbrechen musste, um nachzufragen an was sie arbeiten und wie es aktuell läuft. Ich konnte mir die Infos gut merken und das hier notieren. |

# Reflexion

«Das Argumentieren besteht aus neun Teilen: Sie werden für einen bestimmten Zweck argumentieren (1), Sie werden versuchen, ein Problem zu lösen (2), zumindest eines, Sie werden Informationen verwenden, die Sie von irgendwoher bekommen haben (3), Sie werden diese Informationen mit Konzepten und Ideen interpretieren (4). Sie werden zu einigen Schlussfolgerungen kommen (5). Diese Schlussfolgerungen werden Auswirkungen haben (6). Die Schlussfolgerungen werden auf Annahmen beruhen (7), die Sie mit Ihrem Standpunkt begründen werden (8). Unabhängig davon, ob Sie es erkennen oder nicht, haben Sie in Ihrem Kopf eine Rechtfertigung (9) für Ihre Schlussfolgerungen. Warum die Schlussfolgerungen für Sie richtig erscheinen.» (Paul, 2022)

## Projektmitglied 1: Funktion X

Welche Arbeiten habe ich für die Projektgruppe übernommen und realisiert?

Was ist mir gut gelungen?

Wenn es Schwierig wurde, wie habe ich die Probleme gelöst?

Wie habe ich mich beholfen? Welche Hilfestellung habe von wem erhalten?

Reflexion, Fazit, Massnahme

## Projektmitglied 2

## Projektmitglied 3

## Projektmitglied 4

# Anhang

## Abbildungen

Abb. 1 Beschriftung

## Quellenverzeichnis

Cohnen, T. (13. 03 2013). Placemat (Platzdeckchen-Methode). Rheinland Pfalz.

Diepenhorst, H. (11. 01 2020). *Teamentwicklung Lab*. Von https://teamentwicklung-lab.de/tuckman-phasenmodell abgerufen

Hübscher, H., Petersen, H.-J., Rathgeber, C., Richter, K., & Scharf, D. D. (2015). *IT-Handbuch* (9 Ausg.). Braunschweig: Westerman.

Oefner, M. (2013). *In 20 Schritten zum Redeprofi* (1 Ausg.). Zürich: Verlag SKV AG.

Paul, D. R. (04. 08 2022). Write: How to Teach Students to Write Well. Von https://youtu.be/YDlrN3DfZ\_M abgerufen

*Universität Leipzig Schreibportal - Zitationsregeln*. (03. 08 2020). Von https://home.uni-leipzig.de/schreibportal/zitationsregeln/ abgerufen

*WikipediA - Ablauforganisation*. (05. 08 2020). Von https://de.wikipedia.org/wiki/Ablauforganisation abgerufen

*WikipediA - Projektmanagement*. (04. 08 2020). Von Projektmanagement: https://de.wikipedia.org/wiki/Projektmanagement abgerufen

*WikipediA - Projektorganisation*. (04. 08 2020). Von https://de.wikipedia.org/wiki/Projektorganisation abgerufen