Projektdokumentation Optikerkette SchönesGlas

Projektarbeit

Ausbildung Fachinformatik

Fachrichtung Anwendungsentwicklung

Elektronikschule Tettnang

von

Malte Blumenthal, Benjamin Schick

Abgabedatum:	24.04.2023
Bearbeitungszeitraum:	24.03.2023 - 24.04.2023
Klasse:	EFI222

Inhaltsverzeichnis

1 Projektplanung

1.1 Ausganssituation

1.1.1 Projektziele

1. Mobiles ESP8622
   Als Station für die Aufzeichnung der Daten soll ein Microcontroller verwendet werden. Vorgabe der Stadt Tettnang war ein ESP 8266, welches über eine mobile Stromversorgung betrieben werden soll. Als Sensoren für das Aufzeichnen der Daten sollen ein DHT22 und ein BMP 180 verwendet werden.

2. DHT22
   Der DHT22 Sensor wird für die Aufzeichnung von Luftfeuchtigkeit und Temperatur verwendet.

3. BMP180
   Für die Aufzeichnung der Höhe über Normal Null und Luftdruck wird ein BMP180 Sensor verwendet.

4. Raspberry Pi 4
   Der Webserver braucht Hardware um betrieben zu werden. Da das Verkehrsvolumen auf dem Server als relativ gering eingeschätzt wird, sollte ein Raspberry Pi 4 als Hardwareinfrastruktur ausreichen.

5. Webserver
   Für das Bereitstellen von Visualisierungen der Daten und Aufzeichnung der Daten in einem Datenbank System wird ein Webserver als Interface benötigt. Dafür wird in diesem Projekt NodeJS verwendet.

6. Frontend
   Für die Darstellung der Daten soll eine Website zur Verfügung bereitgestellt werden. Diese soll vorerst eine einzelne Seite sein, die nur eine Tabelle die die Höhe über Normal Null auf einer Horizontalen Zeitachse darstellt.

7. Backend
   Das Backend soll das Topic im MQTT-Broker abonnieren, und dann alle Updates in die Datenbank speichern. Außerdem soll das Backend die Daten aus der Datenbank auslesen und für die Frontendimplementierung bereitstellen.

8. Datenbankserver
   Der Datenbankserver soll auch auf dem Raspberry Pi laufen und die Daten, die von den Sensoren ausgeliefert wurden, permanent abspeichern. Als Datenbanksystem soll MariaDB, eine SQL -Implementierung, verwendet werden.

9. Kommunikation mit MQTT-Broker
   Die Daten müssen zwischen den verschiedenen Geräten ausgetauscht werden. Für die Kommunikation wird im Vorfeld ein öffentlicher MQTT-Broker verwendet. In diesem Projekt wird voraussichtlich mit HiveMQ gearbeitet. Optimal wäre das Hosten eines privaten MQTT-Brokers, aber dies liegt außerhalb des Projekterahmens.

10. Webserver und Broker
    Der Webserver soll sich als Abonnent am entsprechenden Topic des MQTT-Brokers anmelden, um die regelmäßigen Updates des Topics und somit neue Daten zu erhalten.

1.1.2 Teilaufgaben

1. Festlegung Projektziele
2. Festlegung Teilaufgaben
3. Beschreibung Projektumfeld
4. Beschreibung Projektschnittstellen
5. Personalplanung
6. Sachmittelplanung
7. Terminplanung
8. Ablaufplan
9. Kostenplanung
10. Definition Technologiestack
11. Estellen Netzwerkplan
12. Einrichten Raspberry Pi
13. Programmierung ESP
14. Programmierung Backend
15. Programmierung Frontend
16. Testen der Infrastruktur mittels Testfahrt

17. Dokumentation des Pilotprojekts

1.1.3 Projektumfeld, Projektschnittstellen

Tabelle 1.1: Projektumfeld

Auftraggeber	Elektronikschule Tettnang im Auftrag der Stadt Tettnang
Auftragnehmer	Schüler der Klasses EFI222
Räumliches Umfeld	Klassenzimmer an der Elektronikschule Tettnang
Ansprechpartner	Herr Rauschmaier Frau Wattenbach Klassenkameraden
Einstieg	Anfrage der Stadt Tettnang
Ausstieg	Übergabe des Prototypen

1.2 Ressourcen- und Ablaufplanung

1.2.1 Terminplanung

ID	Name	Start Date	End Date	Duration	Resources	Color
1	Projektplanung	May 06, 2024	May 06, 2024	0.75 days	Benjamin Schick,Malte Blumenthal	
3	Erstellung Gantt Diagramm	May 06, 2024	May 06, 2024	1 hour	Malte Blumenthal	
5	Erstellen einer Materialliste	May 06, 2024	May 06, 2024	2 hours	Malte Blumenthal	
4	Erstellen eines Netzwerkplans	May 06, 2024	May 06, 2024	2 hours	Malte Blumenthal	
12	Teammeeting Projektplanung	May 06, 2024	May 06, 2024	20 minutes	Malte Blumenthal	
13	Erstellen Abnahmeprotokoll	May 06, 2024	May 06, 2024	2 hours	Benjamin Schick	
14	Definition Ziele	May 06, 2024	May 06, 2024	2 hours	Benjamin Schick	
21	Definition Technologiestack	May 06, 2024	May 06, 2024	0.25 days	Benjamin Schick	
22	Projektplanung Abgeschlossen	May 06, 2024	May 06, 2024	0 minutes	Benjamin Schick,Malte Blumenthal	
2	Umsetzung	May 06, 2024	May 08, 2024	1.37 days	Benjamin Schick,Malte Blumenthal,ESP8266 Microcontroller,Raspberry Pi 4	
6	Einrichten Raspberry Pi	May 06, 2024	May 07, 2024	2 hours	Benjamin Schick,Raspberry Pi 4	
7	Programmierung ESP	May 06, 2024	May 07, 2024	0.5 days	Benjamin Schick,ESP8266 Microcontroller	
15	Erfassung von Höhendaten	May 06, 2024	May 07, 2024	2 hours	Benjamin Schick,ESP8266 Microcontroller	
16	Verbreitung von Höhendaten an MQTT	May 07, 2024	May 07, 2024	2 hours	Benjamin Schick,ESP8266 Microcontroller	
10	Programmierung Frontend	May 07, 2024	May 08, 2024	0.62 days	Benjamin Schick	
8	Planung Datenbank	May 06, 2024	May 06, 2024	1 hour	Malte Blumenthal	
11	Programmierung Backend	May 06, 2024	May 07, 2024	0.87 days	Malte Blumenthal	
9	Einrichtung Datenbank	May 06, 2024	May 07, 2024	1 hour	Malte Blumenthal	
17	Speicherung der Höhendaten in einer Datenbank	May 07, 2024	May 07, 2024	0.75 days	Malte Blumenthal,Raspberry Pi 4	
25	Teammeeting Projektfortschritt	May 07, 2024	May 07, 2024	20 minutes	Malte Blumenthal	
23	Test und Projektabschluss	May 08, 2024	May 08, 2024	0.87 days	Benjamin Schick,ESP8266 Microcontroller,Malte Blumenthal,Raspberry Pi 4	
18	Testen der Infrastruktur mittels einer Testfahrt	May 08, 2024	May 08, 2024	1 hour	Benjamin Schick,ESP8266 Microcontroller,Malte Blumenthal,Raspberry Pi 4	
24	Teammeeting Projektabschluss	May 08, 2024	May 08, 2024	20 minutes	Malte Blumenthal	
19	Abnahme	May 08, 2024	May 08, 2024	0 minutes	Benjamin Schick,ESP8266 Microcontroller,Malte Blumenthal,Raspberry Pi 4	
20	Dokumentation des Pilotprojekts	May 08, 2024	May 08, 2024	0.71 days	Benjamin Schick,Malte Blumenthal	
26	Projekt Abgeschlossen	May 08, 2024	May 08, 2024	0 days		

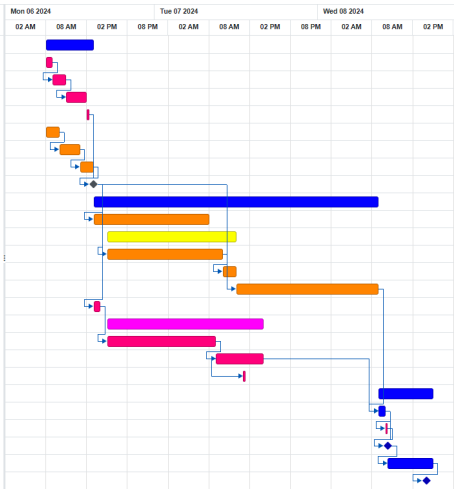

Abbildung 1.1: Terminplanung Tabellarische Ansicht

In dieser Zeitplanung sind keine Puffer berücksichtigt worden, da die gesamte Projektzeit sehr knapp bemessen ist.

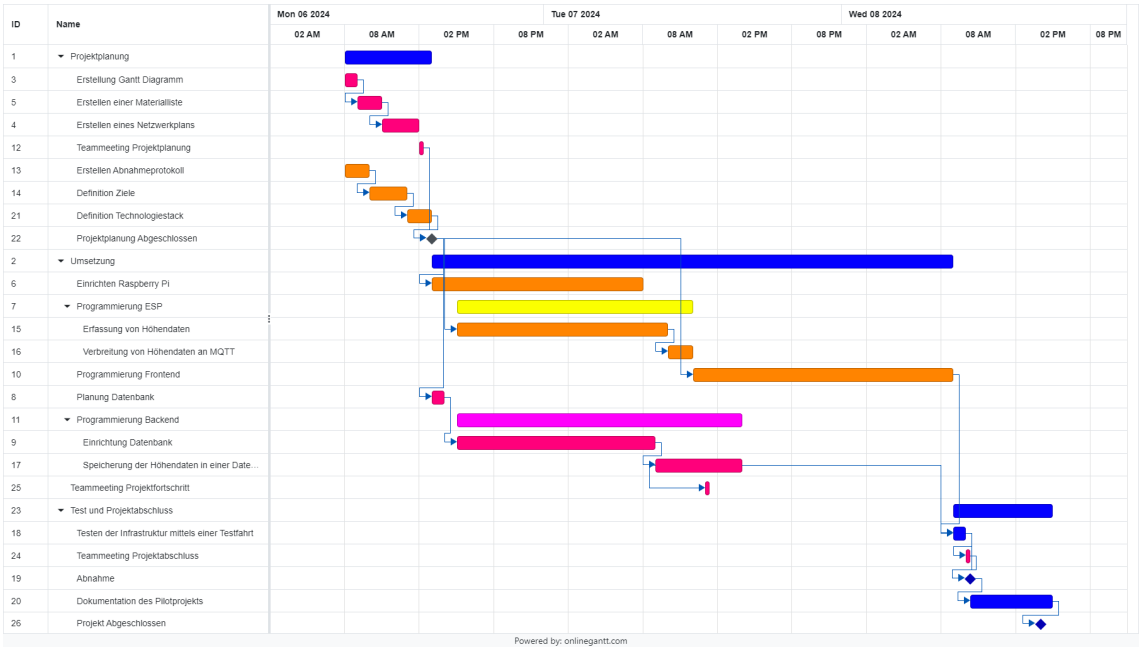

Abbildung 1.2: Terminplanung Gantt-Diagramm

1.2.2 Personalplanung

In dieser Zeitplanung sind keine Puffer berücksichtigt worden, da die gesamte Projektzeit sehr knapp bemessen ist.

1.2.3 Sachmittelplanung

Tabelle 1.2: Sachmittelpanung Hardware

Hardware	**Menge**	**Beschreibung**	**Verfügbarkeit**
Entwickler PC	2x	Hardware benötigt für die Entwicklung der Software und Dokumentation des Projekts	Ist Vorhanden
MQTT-Broker (öffentlich)	1x	MQTT-Server der öffentlich zugänglich ist	Muss beschafft werden
Raspberry Pi 4	1x	Hardware, welche die Rolle des Servers übernimmt. Ist Host des Webservers und der Datenbank	Muss beschafft werden
ESP8266	1x	Microcontroller, welcher die Messungen durchführen soll	Muss beschafft werden
DHT22	1x	Modul für ESP, Zeichnet Temperatur und Luftfeuchtigkeit auf	Muss beschafft werden
BMP180	1x	Modul für ESP, Zeichnet Höhe über NN und Luftdruck auf	Muss beschafft werden

Tabelle 1.3: Sachmittelplanung Software

Software	Beschreibung	Verfügbarkeit
Visual Studio Code	Entwicklungsumgebung für alle Softwareanwendungen, die benötigt werden	Freeware
MariaDB	Datenbanksystem, das für das Speichern der Messdaten verwendet wird	Freeware
MySQL	Datenbanksprache auf der MariaDB aufbaut	Freeware
PlatformIO-Tools	Tools und Bibliotheken für das Programmieren verschiedener Microcontroller	Freeware
NodeJS	Webserver für das Hosten von Webanwendungen	Freeware
UbuntuServer 22.04	Betriebssystem für den Server	Freeware
C++	Programmiersprache für ESP8266	Freeware
MQTT	IoT Netzwerkprotokoll	Freeware

1.2.4 Kostenplanung

Tabelle 1.4: text

Artikel	Menge	Preis
ESP8622	1x	8,00 €
Netzteil ESP8622	1x	5,00 €
Raspberry Pi 4	1x	50,00 €
Netzteil Raspberry Pi 4	1x	5,00 €
DHT22 Sensor	1x	10,00 €
BMP180 Sensor	1x	5,00 €
Mobile Stromquelle	1x	15,00 €
		98,00 €
Lohn Entwickler	48x	6,00 €
Betriebskosten PC	2x	10,00 €
		308,00 €
Gesamtpreis		416,00 €

Somit belaufen sich die geplanten gesamten Projektkosten auf **416,00€**.

1.2.5 Qualitätsplanung

- Qualitätssicherung des Codes mithilfe von GitHub und Versionierung
- Regelmäßiger Self-Test der Hardware
- Fehlermanagement im Code
- SCRUM

2 Soft- und Netzwerktechnik

In diesem Teil wird beschrieben, welche Technologien für die Entwicklung der Internetpräsenz des Kunden verwendet worden sind. Des weiteren wird hier thematisiert welche Entscheidungen während der Entwicklung getroffen werden mussten.

2.1 Softwaretechnik (Kundenwebsite)

2.1.1 Technologiestack

Nodejs

Nodejs ist ein sehr ausgereiftes Framework für Webapplikationen. Es hat eine sehr große Userbase, und ist auch in der Industrie weit verbreitet. Es wird unter anderem von Firmen wie LinkedIn, PayPal und eBay für ihre Internetpräsenz verwendet. Auch die Organisation NASA benutzt Nodejs in ihrem Technologiestack.[10]
In diesem Projekt wird auch der Nodejs interne Webserver verwendet, der obwohl er nicht optimal ist, für den aktuellen Projektstand vollkommen ausreichend.

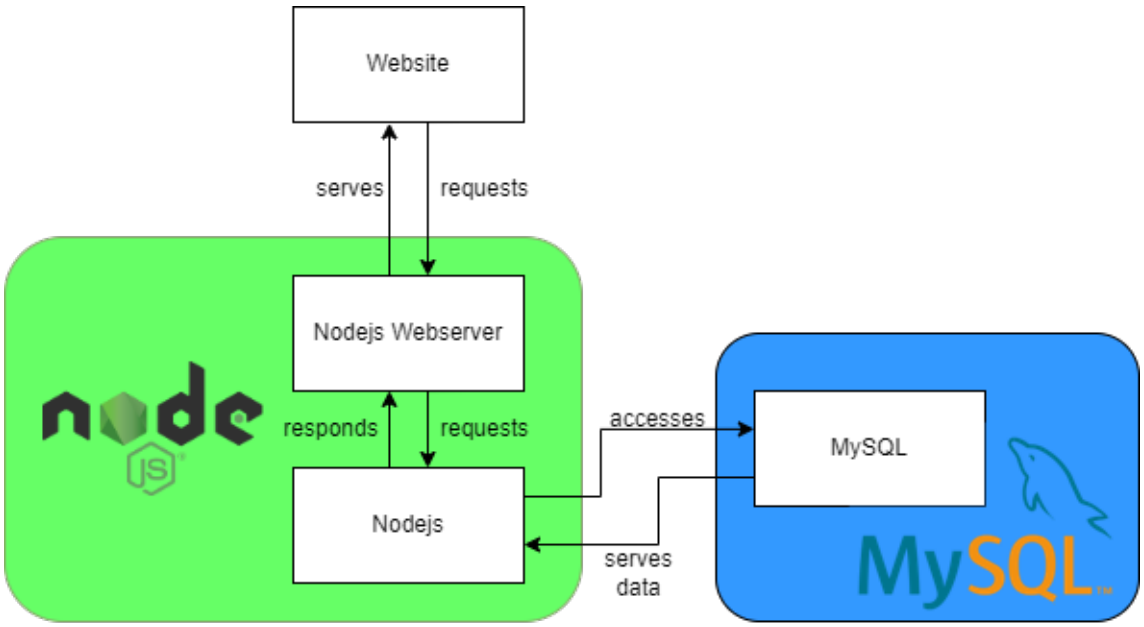

Abbildung 2.1: Technologiestack

MySQL

Für die Datenspeicherung und das Accountmanagement wird eine Datenbank benötigt. MySQL stellt dafür eine weit verbreitete und einfach zu integrierende Lösung dar. Wie aus dem Namen hervorgeht ist MySQL eine Datenbank(-sprache) aus der SQL Familie. Die Webanwendungen von Amazon, Twitter, Netflix und Udemy beispielsweise, arbeiten zumindest teilweise mit MySQL. [9]

HTML - Hyper Text Markup Language

Hyper Text Markup Language oder HTML ist die Programmiersprache mit der Websites geschrieben werden. Das HTML Dokument definiert die generelle Struktur der Seiten und deren Inhalt.

CSS - Cascading Syle Sheet

CSS (Cascading Style Sheets) werden genutzt um HTML Dokumente in eine ästhetisch ansprechende Form zu bringen.

Javascript

Javascript ist eine Programmiersprache, die verwendet wird um Webapplikationen responsiv zu gestalten. Das heißt Anpassungen am Inhalt der Website in Echtzeit durchzuführen. „As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries.“ [11]

EJS - Embedded Javascript Templating

Wie aus dem Namen hervorgeht ist Embedded Javascript Templating (kurz EJS) eine Form Websites in Templates aufzutrennen, also eine Website in die wichtigen Komponenten zu zerlegen. Dies hilft sowohl mit der Usability und der Recognizeability der Website, als auch mit dem Layouting, da die verschiedenen Routen teilweise oder komplett mit den gleichen Komponenten aufgebaut werden. Manchmal unterscheidet sich lediglich der Inhalt der Routen.

Bootstrap

Die in diesem Projekt verwendete Bibliothek Bootstrap ist eine Bibliothek, die sowohl CSS als auch Javascript Funktionalitäten kombiniert um das Design des Frontends zu erleichtern.

2.1.2 Qualitätsmanagement

Die Qualität der Anwendung ist durch ein Git Repository, das auf der Website GitHub angelegt wurde, gewährleistet worden. Git Repositories helfen beim Entwickeln von Anwendungen jeder Art, da das Repository, wenn es gut gepflegt wird, den Entwicklern hilft alte Zustände von Software wiederherzustellen oder Änderungen nachzuvollziehen. Dies Hilft vor allem bei der Fehlersuche, und gewährleistet, dass es, sobald die Software einmal funktioniert hat, immer eine funktionsfähige Version der Software verfügbar ist. Git ermöglicht es außerdem herauszufinden wer welchen Code entwickelt bzw. geändert hat.

2.1.3 Installation

MySQL

Für das Nutzersystem wird das Datenbanksystem MySQL benötigt. Um MySQL auf Linux zu installieren sollten die folgenden Schritte befolgt werden:

1. sudo apt-update

2. sudo apt install mysql-server

3. sudo systemctl start mysql.service

Dann muss die Passwortauthentifizierung für MySQL aktiviert werden da der Webserver sonst keine Daten aus der Datenbank lesen oder Daten in die Datenbank schreiben kann.

1. sudo mysql

2. ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';

3. exit

4. sudo mysql_secure_installation

5. mysql -u root -p

6. ALTER USER 'root'@'localhost' IDENTIFIED WITH auth_socket;

Aktuell läuft das System über den root Benutzer des Datenbanksystems, was nicht optimal ist, da es einfach wäre die Datenbank anzugreifen. [4]

Nodejs

Die Website an sich ist geschrieben mit und wird gehostet von NodeJS. Das heißt auf dem Server muss auch NodeJS installiert werden. [5]

1. sudo apt update

2. sudo apt install nodejs

3. node -v

4. sudo apt install npm

Website Relevante Daten

Um die Website später Hosten zu können müssen die Dateien im src/ Ordner des Repositories auf dem Linux Server liegen. Der einfachste Weg sollte ein standard Git Clone

Befehl sein.

1. git clone ‚link-to-repository-here‘

Git sollte bereits teil der Ubuntu Distribution sein.
Daraufhin sollte in den src/ Ordner des Repositories navigiert werden.

2. Cd repositoryName/src

3. npm install

Jetzt muss die Datenbank initialisiert werden:

4. mysql -u root -p < initDatabase.sql

Damit sollte die Installation abgeschlossen sein und der Server kann über das Ausführen des Befehls „node server.js“ gestartet werden.

2.1.4 Website

Die Website besteht wie den Spezifikationen zu entnehmen war aus mehreren einzelnen Seiten. Diese Seiten sind Home oder in diesem Projekt Index, Products,Services, Contact und About. Eine Anforderung, die bewusst geändert wurde ist die Index-Seite. Diese ist in die About-Seite eingebunden worden. Diese Seiten sind vom Kunden gefordert worden und mit firmenspezifischen Informationen gefüllt worden.
Um es Benutzern zu ermöglichen ein Konto anzulegen und dieses zu verwalten wurden die Seiten Register, Login und Account angelegt. Wenn ein User nicht angemeldet ist wird dieser auf der Login-Seite, die von jeder anderen Seite aus der Sidebar erreichbar ist, aufgefordert sich anzumelden. Sollte der Kunde noch kein Konto besitzen kann er von hier auf die Register-Seite gelangen. Die Account-Seite ist nur erreichbar, wenn sich ein Nutzer erfolgreich angemeldet hat. Ein Link zu dieser Seite und ein Logout Button ersetzen dann auch den Login-Eintrag in der Sidebar.

2.1.5 Datenbank

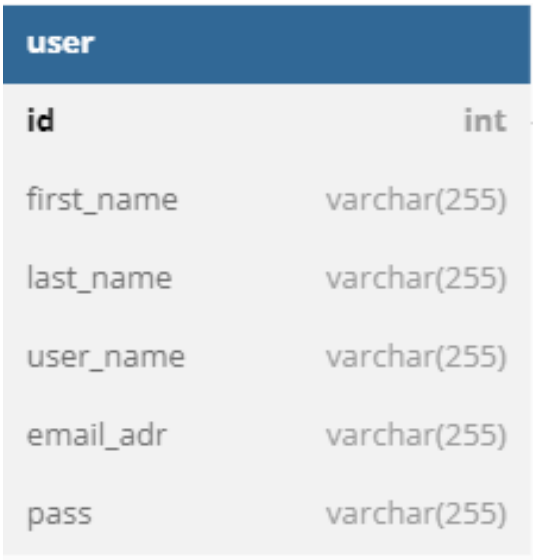

Abbildung 2.2: Datenbankschema

Die Datenbank, die das Accountmanagement übernimmt, ist aktuell sehr simpel gestaltet. Sie speichert den Vornamen, den Nachnamen und die E-Mail Adresse des Benutzers ab. Außerdem speichert sie den Loginnamen oder Benutzernamen und das Passwort des Benutzers.

2.2 Netzwerkkonfiguration

Im Folgenden werden die verwendeten Netzwerkgeräte mit deren jeweiligen Softwarekonfigurationen beschrieben und die daraus resultierende Funktionalität erläutert. Vorwiegend wird hierbei jedoch auf die praktische Umsetzung eingegangen, die theoretisch geplanten Netzwerkgeräte und deren entsprechende Konfigurationen werden kurz erläutert, jedoch wird nicht tiefer darauf eingegangen.

2.2.1 Praktische Hardwarekonfiguration

In der Praxis sollten ein Cisco-Router (Modell unbekannt, zwei GigabitEthernet-Ports) und ebenfalls ein Cisco-Switch entsprechend verkabelt und konfiguriert werden. Ebenfalls war eine IP-Konfiguration von zwei verschiedenen Endgeräten durchzuführen. Zusätzlich wurde noch ein Access-Point eingerichtet, sodass auf das Netzwerk auch kabellos zugegriffen werden kann. Die Verkabelung der einzelnen Komponenten wurde entsprechend 2.3 durchgeführt.

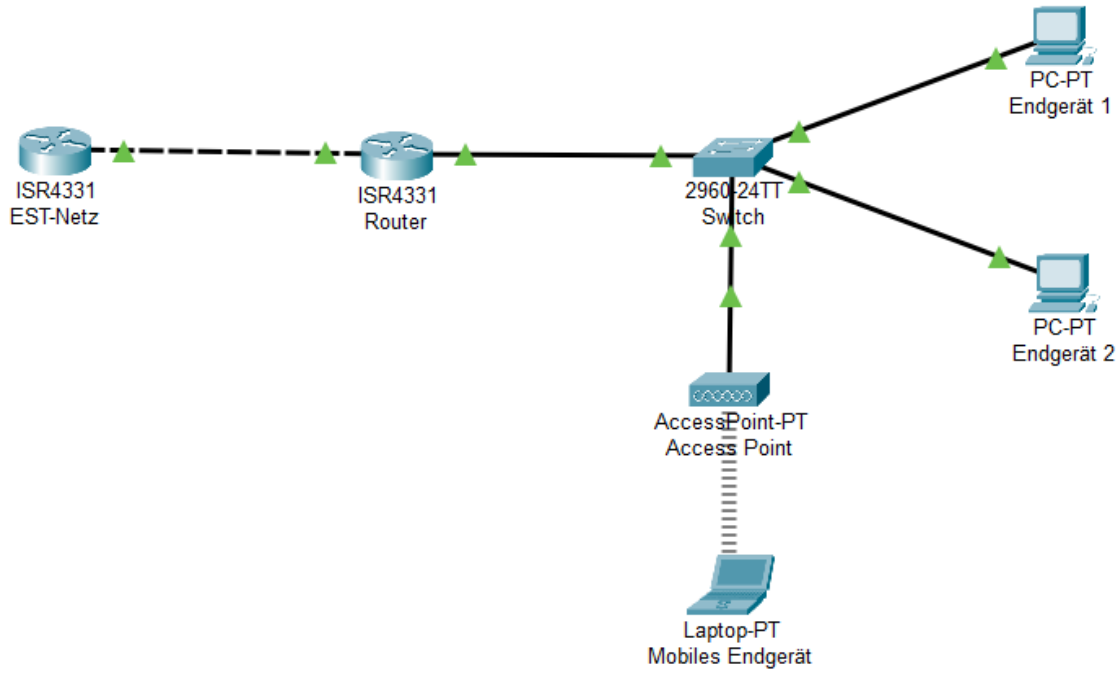

Abbildung 2.3: Netzplan für die praktische Umsetzung

2.2.2 Theoretische Hardwarekonfiguration

In der Theorie gab es so viele Filialen wie Schülerteams, welche ebenfalls untereinander vernetzt werden sollten. Hierfür wurde wie 2.4 zu entnehmen ist, das gegebene Netz in 16 Subnetze unterteilt und neun davon auch tatsächlich benutzt. Diese Subnetze wurden dann pro Team erneut unterteilt um die Abteilungen entsprechend darzustellen 2.6.

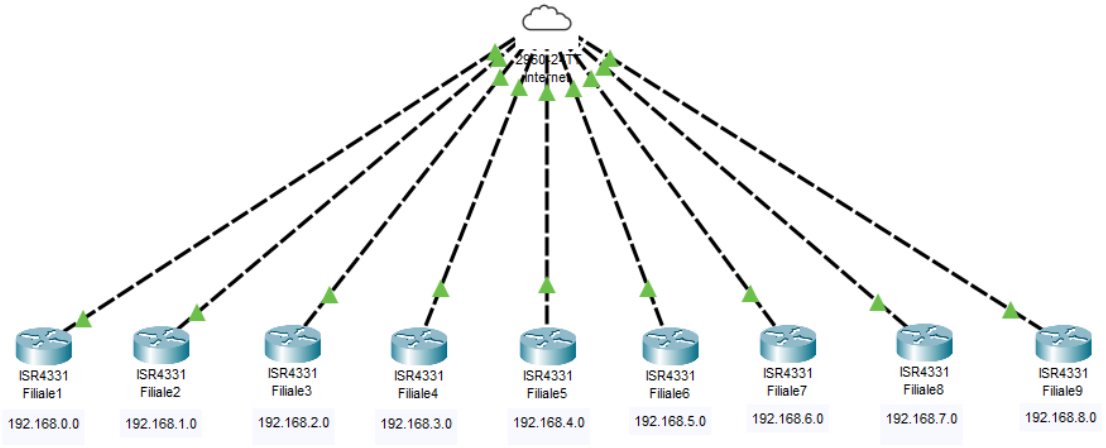

Abbildung 2.4: Darstellung aller verwendeten Subnetze der Filialen

Die theoretische Verkabelung unterscheidet sich im Wesentlichen davon vom praktischen Aufbau, dass hier alle Abteilungen und alle Endgeräte berücksichtigt wurden und sich somit mehr Geräte im Netz befinden und entsprechend auch eine größere Anzahl an Geräten konfiguriert werden muss. In der theoretischen Netzwerkplanung wurde der Access-Point jedoch noch nicht berücksichtigt. Der theoretische Aufbau mit allen Netzwerk- und Endgeräten, wie er in einer Filiale vorliegen würde, wird in 2.5 dargestellt.

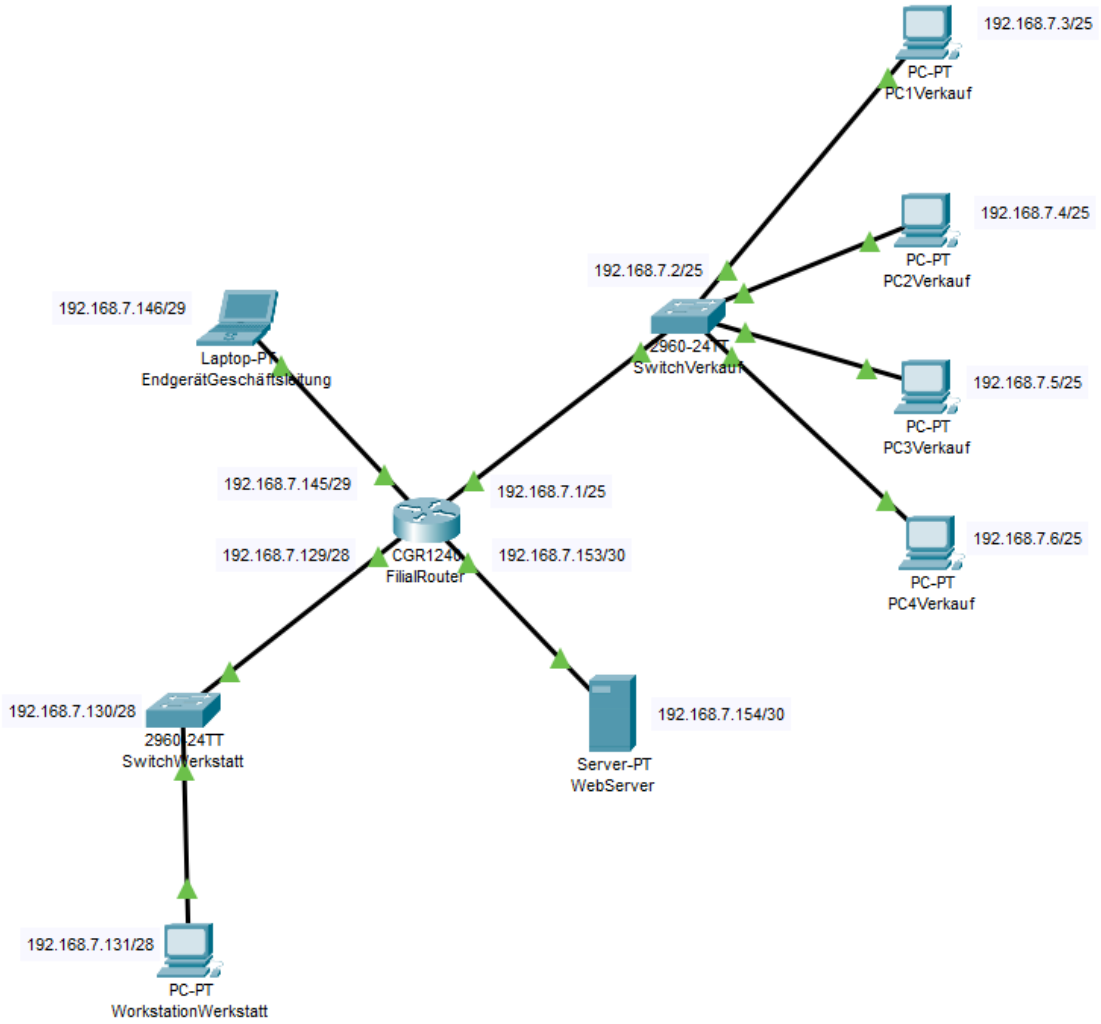

Abbildung 2.5: Darstellung der Subnetze und Geräte in der Filiale

2.2.3 Gerätekonfiguration

Router-Konfiguration

```
Hauptnetz: 192.168.7.0/24


Subnetz Verkauf:
    Netzadresse:    192.168.7.0/25
    Kleinster Host: 192.168.7.1
    Größter Host:   192.168.7.126
    Broadcast:      192.168.7.127
    Subnetzmaske:   255.255.255.128

Routerinterface Verkauf:            192.168.7.1
VlanSwitchVerkauf:                  192.168.7.2
PC1Verkauf:                         192.168.7.3
PC2Verkauf:                         192.168.7.4
PC3Verkauf:                         192.168.7.5
PC4Verkauf:                         192.168.7.6


Subnetz Werkstatt:
    Netzadresse:    192.168.7.128/28
    Kleinster Host: 192.168.7.129
    GrößterHost:    192.168.7.142
    Broadcast:      192.168.7.143
    Subnetzmaske:   255.255.255.240

Routerinterface Werkstatt:          192.168.7.129
VlanSwitchWerkstatt:                192.168.7.130
WorkstationWerkstatt:               192.168.7.131


Subnetz Geschäftsleitung:
    Netzadresse:    192.168.7.144/29
    Kleinster Host: 192.168.7.145
    Größter Host:   192.168.7.150
    Broadcast:      192.168.7.151
    Subnetzmaske:   255.255.255.248

Routerinterface Geschäftsleitung:   192.168.7.145
EndgerätGeschäftsleitung:           192.168.7.146

Subnetz Server:
    Netzadresse:    192.168.7.152/30
    Kleinster Host: 192.168.7.153
    Größter Host:   192.168.7.154
    Broadcast:      192.168.7.155
    Subnetzmaske:   255.255.255.252

Routerinterface Server: 192.168.7.153
Server:         192.168.7.154
```

Abbildung 2.6: Subnetze mit zugeordneten IPs

Bei der Router-Konfiguration wurde die IP-Addresskonfiguration am GigabitEthernet-Port 0/0/0 mit DHCP durchgeführt und als Default-Gateway der Standardgateway der EST gesetzt. Dem GigabitEthernet-Port 0/0/1 wurde eine statische IP-Adresse zugewiesen. Hierbei wurde die erste Host-IP innerhalb eines zuvor berechneten Subnetzes 2.6 verwendet, um sich zumindest an die IP-Konfiguration des theoretischen Netzwerkes zu halten. Ebenso wurde eine Banner-Nachricht sowie Passwörter für den Konsolen-n Fern-

und Konfigurationszugang vergeben.

Switch-Konfiguration

Am Switch wurde ein VLAN-Interface eingerichtet,ebenfalls eine Banner-Nachricht und Passwörter für den Konsolen- und VLAN-Zugriff vergeben. An den beiden Kabelgebundenen Endgeräten wurde eine IP-Konfiguration die ebenfalls schonim Vorhinein geplant wurde, eingerichtet und die Firewall-Regeln wurden aktualisiert, um auf den Webserver, der auf einem der beiden Geräte gehostet wurde, zuzugreifen.

AP-Konfiguration

Dem Access-Point wurde eine statische IP innerhalb des vorbestimmten Netzes zugewiesen und das Netzwerk mit WPA2 verschlüsselt. Da der Router nur mit statischen IP-Adressen konfiguriert ist, ist auch eine Wireless-Verbindung nur dann möglich, wenn die IP-Adresse statisch eingestellt wird und sich innerhalb des Subnetzes des Access-Points befindet.

Theoretische Gerätekonfiguration

Die Konfigurationen der Geräte unterscheiden sich im Wesentlichen nicht von denen, die auch in der Praxis durchgeführt wurden. Es mussten nur zwei Switches statt einem und am Router vier anstelle von zwei Ports konfiguriert werden. Die IP-Konfiguration war jedoch trotzdem dieselbe, die schon in 2.6 dargestellt wurde.

3 Ausblick

Mit der Abnahme des Projektes sind die Ziele des Kunden für den Rahmen des Projektes vollständig erfüllt worden. Es bestehen trotzdem noch Möglichkeiten das System zu erweitern. Ein Punkt wäre das Layout der Website. Die Website beinhaltet zwar das Impressum, aber dies ist aber auf der „About“- oder „Über Uns“-Seite doch sehr versteckt. Man könnte das Impressum beispielsweise in einen Footer verschieben den man auf allen Seiten sehen kann. Ein weiterer Punkt wäre ein Shopsystem. Dafür müssten die nötigen Anpassungen auf den relevanten Seiten getroffen werden, aber auch die Datenbank müsste für diesen Schritt sehr erweitert werden. Noch dazu kommt, dass das Usermanagement momentan sehr rudimentär implementiert ist, was in einer zukünftigen Version etwa aussehen könnte wie in Abbildung 3.1.

Der dritte Punkt wäre die Verschlüsselung der Kundendaten. Aktuell wird lediglich das Passwort mit der Methode md5 im Backend, also von Nodejs, gehashed. Das Ziel war ursprünglich alle vom Nutzer eingegebenen Daten im Frontend, also Clientside, zu hashen und zu salten, damit ein potentieller Man-in-the-middle-Angriff sehr viel weniger gefährlich für die Nutzer des Systems und die Optikerkette sind. Man würde durch diese Maßnahme außerdem die Privatsphäre der Nutzer schützen, da keiner der Administratoren die Daten der Nutzer dann willkürlich aus der Datenbank auslesen kann.

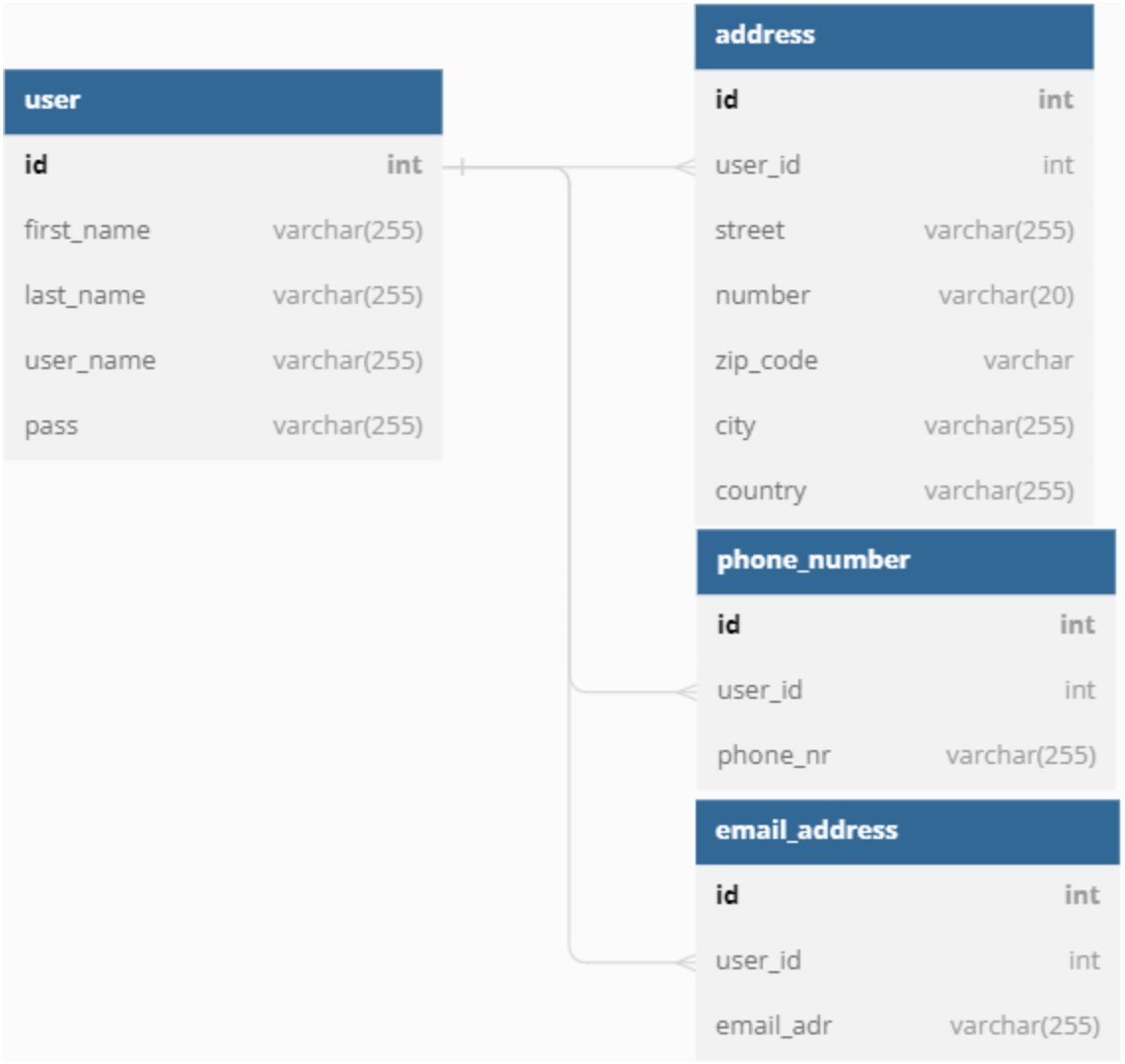

Abbildung 3.1: Benutzerspeicherung in der Datenbank

Literatur

[1] Malte Blumenthal. *Diagramm erstellt mit*. URL: https://dbdiagram.io. (accessed: 18.04.2023, 14:56).

[2] Cherry. *Cherry Desktop Sets*. URL: https://www.cherry.de/desktop-sets. (accessed: 24.03.2023, 13:28).

[3] cisco. *Cisco Packet Tracer*. URL: https://www.netacad.com/courses/packet-tracer. (accessed: 24.03.2023, 08:34).

[4] digitalocean. *Install MySQL*. URL: https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04. (accessed: 22.04.2023, 09:59).

[5] digitalocean. *Install Nodejs*. URL: https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04. (accessed: 22.04.2023, 10:12).

[6] Lenovo. *Lenovo*. URL: https://www.lenovo.com/de/de/. (accessed: 24.03.2023, 12:56).

[7] LG. *LG Monitor*. URL: https://www.lg.com/de/monitore. (accessed: 24.03.2023, 13:10).

[8] Notepad++. *Notepad++*. URL: https://notepad-plus-plus.org. (accessed: 24.03.2023, 08:17).

[9] stackshare. *MySQL*. URL: https://stackshare.io/mysql. (accessed: 18.04.2023, 12:17).

[10] trio. *Nodejs*. URL: https://www.trio.dev/blog/companies-use-node-js. (accessed: 18.04.2023, 10:27).

[11] wikipedia. *Javascript*. URL: `https://en.wikipedia.org/wiki/JavaScript`. (accessed: 18.04.2023, 14:56).

[12] wikipedia. *MySQL Logo*. URL: `https://de.wikipedia.org/wiki/Datei:MySQL_logo.svg`. (accessed: 18.04.2023, 09:21).

[13] wikipedia. *Nodejs Logo*. URL: `https://de.wikipedia.org/wiki/Datei:Node.js_logo.svg`. (accessed: 18.04.2023, 08:51).

Abbildungsverzeichnis

Erklärung

Wir versichern hiermit, dass wir unsere Dokumentation zur Projektarbeit Projektarbeit mit dem Thema:

Projektdokumentation Optikerkette SchönesGlas

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Tettnang, den 13. Mai 2024

Malte Blumenthal, Benjamin Schick