

Diplomarbeit

Name



AUTOR: IRENA BALA

Inhaltsverzeichnis

1	Umsetzung - Irena Bala	1
1.1	Allgemeine Beschreibungen	1
1.1.1	Chatbot	2
1.1.2	Server	2
1.1.3	Technologien	3
1.2	Structured Software Design	8
1.3	Konfiguration von Raspberry PI Server	9
1.4	Datenbank	10
1.5	SSL Verschlüsselung	11
1.6	Anzeige	11
1.7	Wetterdaten	11
1.8	Chatbot	12
1.8.1	Chatbot Einrichtung:	12
1.8.2	Einrichtung des Chatbots in RaspberryPI	13
1.8.3	Grundlagen für die Umsetzung von Chatbot	13
1.8.4	Funktionalitäten von Chatbot	13
1.8.5	Unregistrierte Benutzer:	13
1.8.6	Konkrete Beispiele mit Bildern	16
1.9	Probleme, Herausforderungen und deren Lösung	20
1.10	Qualitätssicherung, Controlling	21
1.11	Ergebnisse - Irena Bala	22
1.11.1	Implementierung	22
1.12	Handbuch für die Bedienung	22
1.12.1	Beschreibung der Bedienung als User	22
1.12.2	Beschreibung der Bedienung als Administrator	23
1.13	Evaluierung und Resümee	23
1.13.1	Planung vs Realisierung	23
1.13.2	Wertschöpfung und Lessons Learned	24

Kapitel 1

Umsetzung - Irena Bala

1.1 Allgemeine Beschreibungen

Bei dieser Diplomarbeit wurde darauf abgezielt, ein intelligentes System zu entwickeln, dass die täglichen Aufgaben des Menschen erleichtert und so viel wie möglich automatisch gesteuert wird. Um dieses Ziel zu erreichen wurden bestimmte Komponenten im System implementiert. Diese wurden mit Absicht so ausgewählt, um die zukünftige Erweiterung und Umsetzung des Projekts auf vielen Anwendungsgebiete zu erlauben. Ein weiterer Zweck besteht also darin, das System zu vervollständigen und sein Entwicklungsgemeinde zu vergrößern.

Diese Komponente wurden in unterschiedlichen Bereichen unterteilt. Der erste Bereich ist die Datenbank. Die Datenbank ist der wesentliche Bestandteil des Systems, weil die Basis für die Speicherung der benötigten Daten bildet. In die Datenbank wurden alle von den verwendeten APIs gekommene Daten gespeichert. Außerdem wurden dort auch die Schuldaten gelegen. Diese Daten sind ein separater Teil des Projekts, die in den folgenden Kapiteln genauer erklärt werden.

Die Darstellung an Bildschirm von den gespeicherten Daten wurde durch eine Admin-Webseite realisiert. Es wurden viele Layouts für die Anzeige (Bildschirm) entworfen, damit die Informationen auf unterschiedlichen Weisen dargestellt werden.

Das Infotainment System funktioniert wie die meisten Systeme nach dem Server- Client Prinzip. Sowohl der Server, als auch der Client werden näher betrachtet, denn sie die Hauptbereiche des ganzen Systems sind. Ein weiterer interessanter Bereich ist der Chatbot. Chatbot wurde deswegen implementiert, weil es die Interaktion des Menschen mit dem System ermöglicht und dadurch wurde angenommen, dass die Umsetzung dieses Komponentes das Interesse des Menschen an dem System erhöhen wird.

Dies war eine allgemeine Beschreibung von den bis jetzige erreichte Ergebnisse. Entsprechend der jeweiligen Arbeitsaufteilung werden in den folgenden Kapiteln einige von den oben genannten Aspekten näher erläutert.

1.1.1 Chatbot

In diesem Unterkapitel wird eine Einführung in Chatbot gemacht und dessen Umsetzung erklärt.

Einführung in Chatbot

Chatbot ist ein sehr wichtiger Komponent von der künstlichen Intelligenz. Es bietet eine Kommunikationsschnittstelle zwischen Menschen und technischen Systeme. Chatbot empfängt Anweisungen in Textform von den Menschen und überträgt diese so, dass die Systeme diesen Anweisungen entsprechen. Basierend auf was Chatbots anbieten, werden als sehr schlaue Komponente angesehen, die immer mehr implementiert werden.

Umsetzung von Chatbot

Chatbot wurde bei dieser Diplomarbeit so implementiert, dass es den Schülern die Möglichkeit gibt, aufgenommene Bilder zum Chatbot zu schicken und dann diese Bilder werden automatisch auf dem Bildschirm angezeigt. Das ist auch die wesentliche Funktionalität von Chatbot bei diesem Projekt.

Um den Chatbot zu implementieren, haben viele kleine Prozesse stattgefunden, die als weitere oder zusätzliche Funktionen angesehen werden können.

Zuerst wurde Telegram Bot API als eine Schnittstelle für die Chatbot-Implementierung ausgewählt. Durch diese API können neue Bots erstellt, bearbeitet und verändert werden. Telegram Bot API funktioniert gleich wie die anderen Kommunikationsapplikation z.B Whatsapp. Der wesentliche Unterschied ist, dass bei dieser Applikation wird nicht nur die Möglichkeit mit anderen Menschen zu chatten angeboten, sondern auch mit Chatbots. Jedes Bot, das erstellt wird, bekommt ein Token, dass eindeutig für das Bot ist, wie die Telefonnummer für uns eindeutig ist.

Die komplette Funktionalität des Bots wurde in RaspberryPI Server, mithilfe der Python Programmiersprache programmiert.

Je nachdem ob die Person, die mit dem Bot chatten will, als ein normaler Benutzer oder ein Administrator in der Datenbank definiert ist, werden ihm verschiedene Funktionen im Zusammenhang mit dem Bot zur Verfügung gestellt. Die Nachrichten, die zu dem Bot geschickt werden, werden nach Inhalt überprüft. Basierend auf Inhalt der Nachrichten, wird der Bot auf verschiedene Weisen reagieren.

Die Art der Umsetzung bzw. Realisierung aller diesen Funktionen kann im Unterkapitel 2.1 gelesen werden.

1.1.2 Server

Das Infotainment System wie die anderen technischen Systeme, funktioniert nach dem Client-Server Prinzip. Das bedeutet, es gibt einen Server und einen Client, die miteinander kommunizieren. Der Client ist in diesem Projekt die Anzeige bzw. das Bildschirm, dass die von dem Server bekommene Daten darstellt.

Im Server liegen aber alle benötigten Informationen für die Darstellung. Diese Informationen sind auf der Datenbank gespeichert, die auf dem Server stattfindet.

Der Server beinhaltet auch die grundlegenden Skripts für die Chatbot-Implementierung und für die Programmierung der Admin-Webseite. Im Server wurde auch das SSL Zertifikat für eine sichere Datenübertragung erstellt.

Die Kommunikation zwischen dem Client und dem Server wird dann aufgebaut, wenn Daten von dem Server ausgewählt und zum Client geschickt werden. Der Server ist der grundlegende Teil des Projekts. Es enthält alle benötigten Ressourcen für die vollständige Umsetzung des Systems.

1.1.3 Technologien

In diesem Unterkapitel werden die verwendeten Technologien und Software-Ressourcen beschrieben. Es werden die grundlegenden Theorien, die hinter diesen Technologien stehen, im Detail erläutert. Dazu werden auch die Gründe für die Auswahl der Software-Ressourcen erklärt. Die unterliegende Tabelle listet die verwendeten Technologien auf und daneben steht auch eine kurze Beschreibung für jede Technologie.

Name	Beschreibung
Apache HTTP Server	Webserver
MySQL	Relationales Datenbanksystem
PHP	Serverseitige Programmiersprache
JavaScript	Programmiersprache zur dynamischen Veränderung von Webseiten
Python	Objektorientierte/ prozedurale Programmiersprache
HTML	Auszeichnungssprache zur Erstellung von Inhalten bei Webseiten
CSS	Methode, zur Entkopplung von Designanweisungen einer HTML Datei
Wetter API	Schnittstelle zur Aufnahme von Wetterdaten aus großen Wettervorhersage-Datenbanken
JSON	strukturiertes Dateiformat
Telegram API	Schnittstelle zur Implementierung von Chatbot
Raspberry PI	Minirechner, der für Scripting, Linux Programmierung geeignet ist
SSL	Methode zur verschlüsselten Datenübertragung zwischen Browser und Server

Tabelle 1.1: Technologien

Was ist Apache HTTP Server?

Der Apache HTTP¹ Server ist ein weltweit verbreiteter Web Server. Dieser Server ist Open Source, das bedeutet, dass es keine Lizenz gekauft werden soll, um es zu verwenden. Es ist kompatibel auf allen kohärenteren Betriebssystemen, beispielsweise Linux, Windows, Mac OS und andere. Es bietet viele Versionen an, die zu unterschiedlichen Anwendungsgebiete passen und verbesserte Eigenschaften bereitstellen. Durch dieses Webserver können Webseiten erstellt werden. Die Erstellung der Webseiten erfolgt über serverseitige Scriptsprachen, die von dem Server selbst nicht unterstützt werden. Sie werden als Zusatzfunktionen angehängt.

Der Apache HTTP Server bietet viele Funktionalitäten an, die seine Entwicklungsumgebung vergrößern. Die wichtigste davon ist die Möglichkeit der Integration eines SSL ²Zertifikats. Das ermöglicht die Übertragung der Daten in einer verschlüsselten

¹Hypertext Transfer Protocol

²Secure Socket Layer

Form. Die detaillierte Funktionsweise eines SSL Zertifikats wird in den Unterkapiteln beschrieben. [2]

Funktionsweise von Apache HTTP Server

„Obwohl Apache als Webserver bezeichnet wird, handelt es sich nicht um einen physischen Server. Apache ist eine Software, die auf einem Server ausgeführt wird. Seine Aufgabe ist es, eine Verbindung zwischen einem physischen Server mit den gespeicherten Webseiten und den Browsern der Internetuser herzustellen.

Wenn ein User eine URL in seinen Webbrowser eingibt, sendet der Browser eine HTTP oder HTTPS ³ Anforderung an den Server, auf dem die Webseite gespeichert ist.“ [2]

Was ist MySQL?

MySQL ist ein weitverbreitetes relationales Datenbanksystem. Die Datenbanksysteme werden allgemein zur Datenspeicherung und Datenverwaltung verwendet. Ein wichtiges Kriterium für die Datenspeicherung ist die Performanz. Diese Anforderung wird durch MySQL optimal erfüllt. Das ist auch der Grund, warum dieses Datenbanksystem so populär und bekannt ist. Die von MySQL für die Abarbeitung, Verwaltung und Systematisierung von Daten verwendete Sprache ist SQL⁴. MySQL ist auch eine Open Source Software, die in meisten Fällen in Verbindung mit serverseitigen Scriptsprachen wie PHP, vorkommt. [3]

Funktionsweise von MySQL

Das MySQL Datenbanksystem wird sehr häufig implementiert. Es gibt viele Unternehmen und Institutionen, die ihre Daten über eine gewisse Zeit speichern wollen. Das MySQL Datenbanksystem, das die Daten beinhaltet, wird als ein Server vorgesehen. Jeder, der versucht, Zugriff auf diese Daten zu haben, wird als ein Client vorgesehen. Der Server kann die erforderliche Zugänglichkeit erlauben oder nicht. Das hängt von den Clientrechten ab. Die Daten können von den Clients selektiert, bearbeitet oder gelöscht werden. Diese Ereignisse erfolgen durch SQL-Abfragen. Die SQL-Abfragen werden mithilfe der SQL Datenbanksprache erstellt. [3]

Was ist PHP?

PHP⁵ ist eine serverseitige Programmiersprache. Das bedeutet, dass diese Sprache, um die vom Server auszuführenden Ereignissen zu programmieren, verwendet wird. PHP ist eine sehr verbreitete Programmiersprache, die am meisten zur Erstellung und Programmierung von Webseiten verwendet wird. Eigentlich ist PHP sehr flexibel, denn es einen großen Schnittstellenansatz anbietet. Diese Programmiersprache kann auch im Zusammenhang mit Datenbanken genutzt werden. [4]

³Hypertext Transfer Protocol Secure

⁴Structured Query Language

⁵Parallel History Project

Funktionsweise von PHP

Hier kommt das Client-Server Prinzip wieder vor. Der Webbrowser ist der Client und der Webserver ist der Server. Der mit PHP programmiertes Skript wird zum Webserver geschickt, danach erfolgt die Rückgabe einer HTML-Datei als Antwort zum Webbrowser, der in diesem Fall als Client betrachtet wird. [4]

Was ist JavaScript?

JavaScript ist eine Programmiersprache, die am meisten zur Erstellung von dynamischen Funktionalitäten bei Webseiten, verwendet wird. Die JavaScript Programmiersprache hat in der Vergangenheit nur eine beschränkte Anzahl von Funktionen angeboten, aber heutzutage bietet sie eine Vielzahl von Einsatzmöglichkeiten. [5]

Mögliche Funktionen von JavaScript

„JavaScript wurde entwickelt, um dynamische HTML-Seiten per Webbrowser anzuzeigen. Die Verarbeitung von JavaScript erfolgt meist clientseitig direkt durch den Webbrowser.

Mit Hilfe der Skriptsprache JavaScript lassen sich viele dynamische Funktionen realisieren. Hier sind einige Beispiele für die Verwendung von JavaScript:

- dynamische Veränderung von Webseiten – zum Beispiel für die Anzeige eines formatierten und aktualisierten Datums
- Prüfung von in Formularen eingegebenen Daten auf Plausibilität
- Anzeige von Laufschriften oder Bannern
- Öffnen und Anzeigen von Dialogfenstern
- Aktualisieren von Daten einer Webseite ohne neu laden im Browser
- Unterstützung der Eingabe von Daten durch den User
- Veränderung von Texten oder Grafiken durch den Mauszeiger” [5]

Was ist Python?

Python ist eine objektorientierte Programmiersprache, aber kann auch in prozedurale Programmierung verwendet werden. Sie wurde ausschließlich zum Zweck der einfach einprägsamen Syntax entwickelt. Andererseits haben die Entwickler der Systematisierung des Codes große Bedeutung beigemessen. Wegen dieser angewandten Eigenschaften kann Python in die Gruppe der leichten Programmiersprachen aufgenommen werden. Diese Programmiersprache wird viel verwendet, aber was die Anwendungsumgebung besonders erhöht, ist die Möglichkeit andere Module anzuhängen. Es ist auch eine Open Source Software, der von den Programmierern verwendet, verändert, angepasst bzw. bearbeitet werden kann. Es wird meistens für komplexe Aufgaben verwendet werden, deswegen wird es als eine Hochsprache betrachtet. [6]

Merkmale von Python

- Einfach einprägsame Syntax
- Objektorientierte und prozedurale Programmiersprache
- Open Source
- Hohes Niveau Programmiersprache
- Leicht veränderbare Programmiersprache[6]

Was ist HTML?

HTML⁶ ist keine Programmiersprache, die wird für die Erstellung von Inhalten bei Webseiten verwendet. Diese Inhalte können Texte, Bilder oder andere Komponente sein. HTML wird als eine Auszeichnungssprache betrachtet. Sie ist nicht nur für die Erstellung von Webseite-Inhalten zuständig, sondern auch für ihr Design. Diese Sprache liegt mithilfe von bestimmten Tags die Struktur einer Webseite fest. Im Tag werden die Inhalte gespeichert. Es gibt bestimmte Tags für verschiedene Layout-Elemente.[7]

Was ist CSS?

CSS⁷ wird im Zusammenhang mit HTML verwendet. Diese Methode wird unten genauer betrachtet.

„CSS steht für Cascading-Style-Sheets und ist eine Möglichkeit für HTML-Dokumente, den Inhalt einer Seite von den Designanweisungen der einzelnen Elemente, wie zum Beispiel Überschriften, Zitaten) zu entkoppeln.“ [11]

Was ist Raspberry PI?

Der Raspberry PI ist ein Minirechner, die zur Linux Programmierung, Shell Scripting und Realisierung von technischen Projekten verwendet wird. Es braucht eine Tastatur, eine Maus, ein Netzteil, VGA⁸ und HDMI⁹-VGA Konverter, damit es genutzt werden kann. Die Konfiguration von einem Raspberry PI erfolgt durch eine SD-Karte. Diese SD¹⁰-Karte beinhaltet das Image, wo das Betriebssystem liegt. Ein Raspberry PI kann in Zusammenhang mit vielen anderen Komponenten verwendet werden.[9]

⁶HyperText Markup Language

⁷Cascading Style Sheets

⁸Video Graphics Array

⁹High-Definition Multimedia Interface

¹⁰Secure Digital Memory Card

Was ist SSL?

SSL steht für Secure Socket Layer und ist für die verschlüsselte Übertragung der Daten vom Browser zum Server verantwortlich. Die Verbindung zwischen dem Server und dem Browser erfolgt durch das HTTPS Protokoll. Das ist ein Kommunikationsprotokoll, das eine verschlüsselte Datenübertragung ermöglicht. Heutzutage wird TLS¹¹ am meisten verwendet, der der neueste und modernste Standard von SSL ist.[1]

SSL-Verschlüsselung

Um eine verschlüsselte Verbindung zwischen einem Browser und einem Server aufzubauen, werden SSL – Zertifikaten integriert. Mittels ein SSL Zertifikats wird die Authentizität einer Webseite überprüft. Das SSL Zertifikat wird von einer Zertifizierungsstelle, erzeugt. Diese Zertifizierungsstelle heißt CA¹² und erfordert einige Daten von dem Antragsteller, die für die Erstellung des Zertifikats notwendig sind. Als nächstes, erzeugt der Antragsteller für die Entschlüsselung und Verschlüsselung der zwischenübertragenen Daten ein öffentlicher, -und ein privater Schlüssel. Je grösser die Länge des Schlüssels ist, desto sicherer und besser ist. Meistens werden Schlüssel mit einer Länge von 256 Bit verwendet.[1]

Was ist Telegram Bot API?

Telegram Bot API¹³ ist eine Schnittstelle, die die Implementierung von Chatbot ermöglicht. Es bietet verschiedene Funktionen an, nämlich die Einrichtung, Erstellung und die Verarbeitung von Bots. Diese Funktionen sind in der eigenen Dokumentation von Telegram Bot API klar beschrieben. [12]

Was ist Wetter API?

„Wetter APIs sind Schnittstellen, die die Verbindung zu einer großen Wettervorhersage-Datenbank und die Aufnahme benötigter Daten ermöglichen.“ [10]

Was ist JSON?

„JSON¹⁴ bietet einen einfachen Standard für die strukturierte Kodierung von Daten in Form von menschenlesbarem Text. Dies bietet Vorteile bei einer automatisierten Weiterverarbeitung, macht sie aber auch einer manuellen Inspektion und Überarbeitung besser zugänglich.“

[8]

In der untenstehenden Tabelle sind alle Technologien zusammen mit dem Bereich wo sie gehören ersichtlich.

¹¹Transport Layer Security

¹²Certificate Authority

¹³Application Programming Interface

¹⁴JavaScript Object Notation

Bereich	Technologie
Datenbank	Apache HTTP Server, MySQL, PHP
Anzeige	HTML, CSS, JavaScript
Server	RaspberryPI, SSL
Wetterdaten	Wetter API, JSON
Chatbot	Telegram API, Python

Tabelle 1.2: Bereiche und Technologien

1.2 Structured Software Design

Die Abbildung 1.1 stellt den Software Structured Design von dem Server dar. Durch diese Methode wird die komplette Funktionalität des Servers bei diesem System geplant. Alle Module, die miteinander verbinden sind die Komponente des Systems, die Teil des Servers sind. Der Pfeil bei der unteren Abbildung, das Init genannt ist zeigt die Einschaltung des Servers. Nach der Servereinschaltung wird ein Request zu allen Komponenten geschickt, dass sie auch eingeschalten werden sollen. Diese Komponente sind der Chatbot, die Datenbank und die APIs. Nachdem die Komponenten eingeschalten sind, schicken sie eine Statusmeldung zum Server, wie in der Abbildung 1.1 ersichtlich ist.

Nachdem der Chatbot eingeschalten wird und mit der Arbeit angefangen hat, wird er Daten produzieren, die zuerst überprüft werden sollen, ob sie im richtigen Format sind oder nicht. Falls der Chatbot Bilder kriegt, wird ein Query abgebaut und die Daten werden in die Datenbank gespeichert. Falls nicht, wird alles im Log Modul gespeichert. Der Log Modul hat dieselbe Funktionalität für alle Komponente. Es speichert einfach alle Daten und alle Ergebnisse. Dasselbe passiert mit der APIs auch. Die WetterAPI und Website API speichern ihre Ergebnisse im Log. Wenn sie Daten zurückliefern, werde sie zuerst validiert und dann auf die Datenbank gespeichert.

Am Ende erfolgt eine Anfrage von dem RaspberryPI Server. Er erfordert die gesammelten Daten aller Komponenten. Sie werden zuerst validiert dann von Log genommen und danach zum Server geschickt.

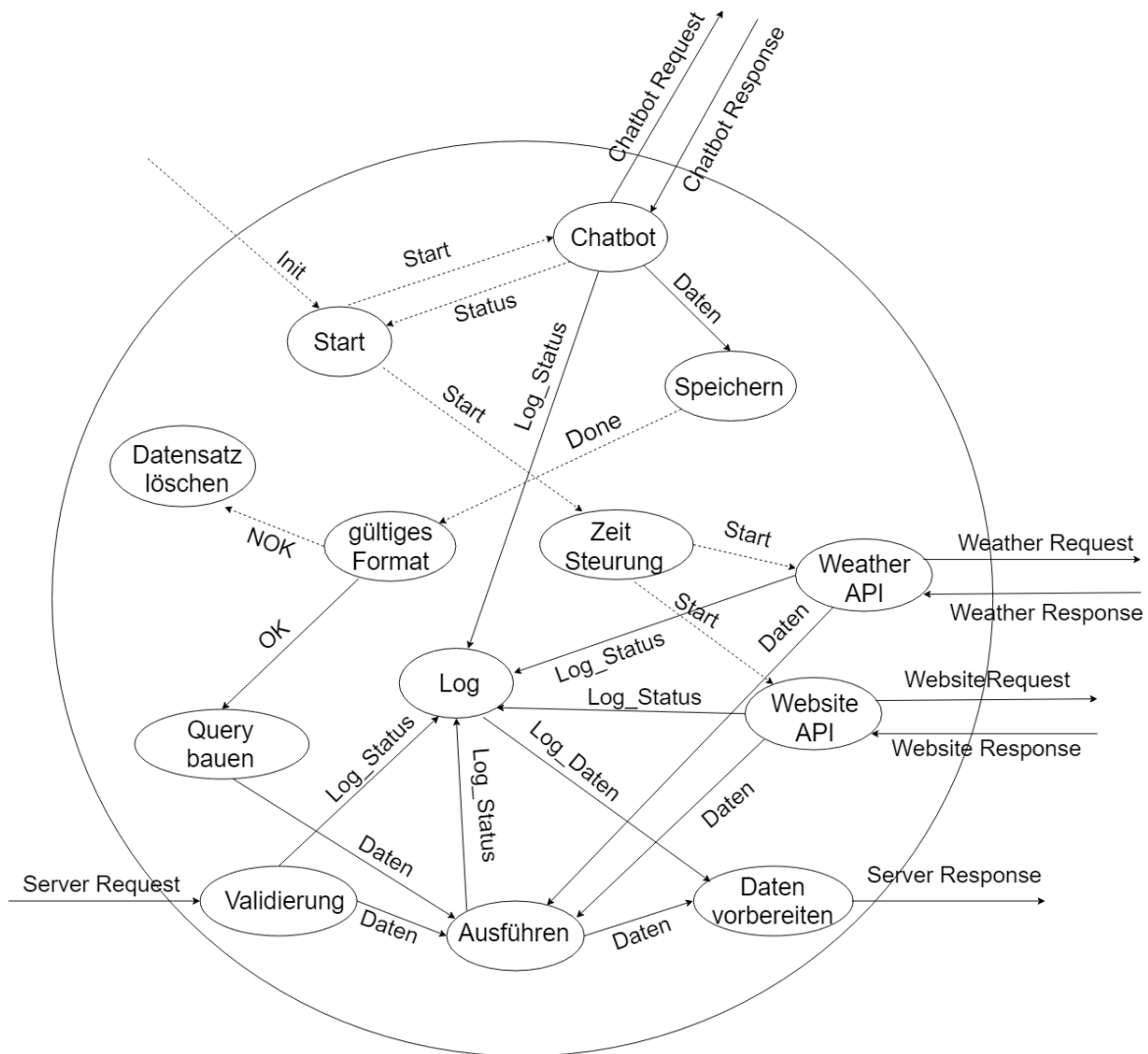


Abbildung 1.1: Structured Software Design

1.3 Konfiguration von Raspberry PI Server

- Zuerst wurde ein Image daraufgespielt, die das Betriebssystem von Raspberry PI beinhaltet
- Danach wurden die folgenden gebrauchten Paketen installiert: git, vim, apache2, python-pip, telepot, php php-mbstring, mariadb-server php-mysql, phpmyadmin
- In der Konfigurationsdatei wurde die IP-Adresse von dem Raspberry PI angelegt
- Der nächste Schritt war die Erstellung der Datenbank und der dazugehörigen Tabellen
- Dann wurden die Benutzer angelegt und die Rechte vergeben
- Als letztens wurde SSH¹⁵ aktiviert, damit eine sichere Verbindung zu diesem Server von einem externen Gerät ermöglicht werden kann

¹⁵Secure Shell

1.4 Datenbank

- Zuerst wurde ein ER¹⁶ Diagramm mit Papier gezeichnet. Das Ziel war die richtige Erstellung der benötigten Tabellen. Die Tabellen wurden mit den Spalten und ihren Datentypen erstellt. Es wurden auch die Kardinalitäten dazwischen gezeichnet. Die erstellten Tabellen sind:
 - Unterricht
 - Fehlende Lehrer
 - Wetterdaten
 - Chatbot Users
 - User Rolle im Chatbot
 - Chatbot Bilder
- Basierend auf das ER Diagramm wurden dann die Tabellen mit phpmyadmin erstellt
- Dann sind für alle Tabellen mit MySQL Workbench die entsprechenden gespeicherten Prozeduren erstellt

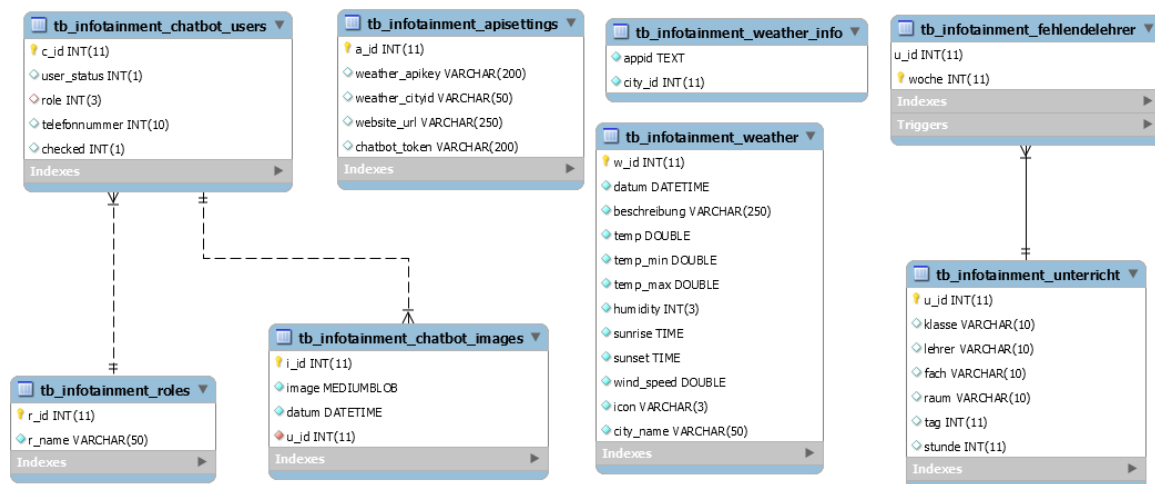


Abbildung 1.2: ERD

¹⁶Entity Relationship

1.5 SSL Verschlüsselung

Es wurde ein SSL Zertifikat für den Apache HTTP Server eingerichtet

- Zuerst wurde das SSL Modul für Apache aktiviert
- Danach wurde das SSL Zertifikat erstellt.
- Nach der Erstellung des SSL Zertifikats werden einige Eingaben geschickt, die erfüllt werden sollen.
- Danach wurde die Datei `/etc/apache2/sites-available/default-ssl.conf` geöffnet.
- Unter der Zeile, wo SSL Engine On steht, wurden die erstellten Zertifikatdateien zugefügt
- Danach wurde der Virtuallhost mit SSL aktiviert.
- Es wurde ein System reboot gemacht und der Apache Server noch einmal gestartet [13]

1.6 Anzeige

Bei der Anzeige sind die Wetterdaten, Kalenderinformationen, Stundenplan Tabelle, Supplierplan Tabelle dargestellt. Die Wetterdaten wurden aus dem API Call gelesen, in die Datenbank gespeichert und dann durch ein PHP Skript von der Datenbank abgelesen. Die Formatierung für die Anzeige wurde mit JavaScript gemacht. Die Kalenderinformationen, sind Informationen, die selbst geschrieben werden. Diese Informationen sollten so angepasst, dass sie in unterschiedlichen Teilbereichen des Layouts hinzugefügt werden können und auch von den anderen lesbar sind.

Die Supplierplan Tabelle und die Stundenplan Tabelle wurden aus der Datenbank selektiert und dann mit PHP dargestellt. Die Formatierung wurde mit JavaScript gemacht. Bei den Tabellen, wo viele Datensätze gab, wurde die Tabelle in zwei Seiten aufgeteilt, damit es richtig im Layout der Anzeige angepasst werden kann. Auf die Abbildung 1.3 ist die Darstellung der Wetterdaten auf dem Bildschirm ersichtlich.

1.7 Wetterdaten

- Als erster Schritt erfolgte die Registrierung bei openweathermap.org
- Danach wurde ein API Key bekannt gegeben, mit dem die API Call gemacht werden kann
- Die Antwort von API Call wird im JSON-Format gegeben
- Danach erfolgt die Speicherung der bestimmten Parameter, die von dem API CALL kommen, in der Datenbank

- Als letztens werden mithilfe von PHP die in der Datenbank gespeicherten Parameter aufgerufen und auf dem Bildschirm angezeigt

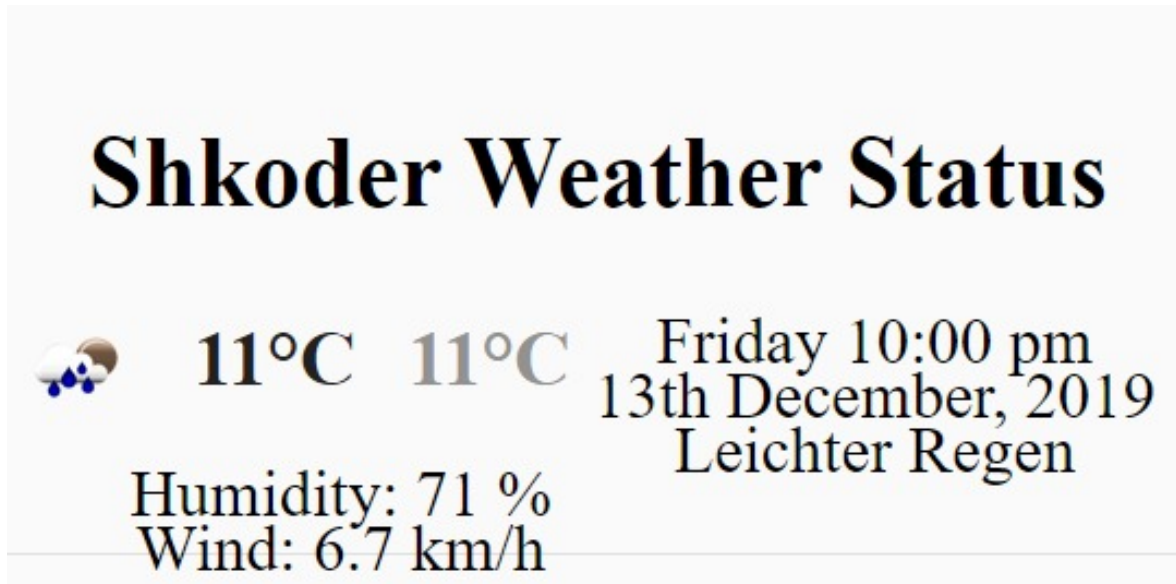


Abbildung 1.3: Wetterdaten

1.8 Chatbot

In diesem Unterkapitel wird der Chatbot erklärt und dessen Umsetzung auch.

1.8.1 Chatbot Einrichtung:

- Zuerst wurde die Telegram Applikation auf dem Handy heruntergeladen
- Als nächstes wurde es nach einem Benutzer mit der Name Botfather gesucht. Der Botfather ist der Verwalter von Bots und ist für die Einrichtung des Chatbots zuständig.
- Danach wurde zum Botfather eine Anfrage geschickt, um einen neuen Bot zu erstellen.
- Der Botfather fragte danach nach, wie das neue Bot genannt wird und was für ein Benutzername es haben wird.
- Danach wurde von dem Botfather das Token des neuen Bots bekannt gegeben, mit dem der Bot weiterentwickelt werden kann.
- Nach der Durchführung von diesen Schritten, ist der Bot fertig erstellt geworden. Es könnte dann mit dem angegebenen Benutzernamen durchgesucht werden und der entsprechende Chat damit geöffnet werden.

1.8.2 Einrichtung des Chatbots in RaspberryPI

Die Entwicklung und die Programmierung von Chatbot wurde im RaspberryPI Server gemacht. Damit der RaspberryPI mit dem Telegram Bot API verbunden werden kann, sollten im Raspberry PI zwei Paketen installiert werden. Diese Paketen sind telepot und python-pip. Telepot ist die Pakete, die die Verbindung zum Telegram Bot API erstellt und mit einem Python-Framework funktioniert.[12]

1.8.3 Grundlagen für die Umsetzung von Chatbot

- Zuerst wurde ein Python Skript erstellt.
- Danach wurden die benötigten Modulen für den Python Interpreter importiert. Der wichtigste davon ist der, der die Schnittstelle zum Telegram Bot API bildet.
- Der Token, den wir von dem Botfather bekommen haben, wurde in einer Variable gespeichert. Mit diesem Token ist der Zugriff zu dem erstellten Bot möglich. Mit den dargestellten Funktionen sollte die Verbindung zu dem Infotainment Bot erfolgen und gleichzeitig sollte auch getestet ob diese Verbindung überhaupt funktioniert und ob Informationen von dem Bot zurückkommen.
- Dann erfolgt die Verbindung zu der Datenbank. Um die Verbindung mit der Datenbank zu ermöglichen, sollte zuerst die MySQL Bibliothek importiert. Das wurde bei dem zweiten Schritt gemacht.

1.8.4 Funktionalitäten von Chatbot

Je nachdem ob die Person, die eine Nachricht zu dem Infotainment Bot schickt, ein Administrator, ein normaler Benutzer oder ein unregistrierter Benutzer ist, werden ihm verschiedene Funktionalitäten angeboten.

1.8.5 Unregistrierte Benutzer:

Wenn eine Person zum ersten Mal eine Nachricht zum Infotainment Bot schickt, ist er noch nicht in der Datenbank registriert. Der Bot wird ihn fragen ob er registrieren will oder nicht. Die Möglichkeit zur Registrierung erfolgt durch zwei Buttonen, die im Bot integriert werden. Falls der unregistrierte Benutzer Ja drückt, bittet er ihn die Telefonnummer einzugeben. Durch diese Telefonnummer wird dann der Benutzer in der Datenbank gespeichert. Falls nein, wird ihm nur eine Nachricht vom Chatbot zurückgeschickt.

Die Telefonnummer, die von dem unregistrierten Benutzer eingegeben wird soll nach Inhalt überprüft werden. Wenn der Benutzer z.B Text eingibt, wird das nicht genehmigt und nicht gespeichert. Falls eine Zahl ist, soll es 10 Ziffern lang sein und mit einem vorgelegten Prefix anfangen. Ansonsten wird es nicht genehmigt und der unregistrierter Benutzer kriegt wieder Möglichkeiten vom Chatbot die Nummer richtig zu schreiben. Die Informationen über den unregistrierten Benutzer wird der Administrator bekommen. Er soll dann die Genehmigung geben, ob diese Person schon eine Interaktion mit dem Chatbot haben darf oder nicht.

Normale Benutzer:

Die Daten über die normalen Benutzer sind in der Datenbank gespeichert. Diese Benutzer sind von dem Administrator genehmigt. Die Nachrichten, die diese normalen Benutzer zum Chatbot schicken werden nach Inhalt überprüft. Falls sie ein Bild schicken, wird dieses Bild auf die Datenbank gespeichert und auf dem Bildschirm angezeigt. Wenn aber der Benutzer eine Nachricht schickt, die kein Bild ist, wird es keine Interaktion mit dem Bot geben, weil dem Bot interessieren die anderen Nachrichtformate nicht.

Administrator:

Dem Administrator werden andere Möglichkeiten zur Verfügung gestellt. Er ist in der Lage die normalen und die unregistrierten Benutzer zu sehen. Die werden von der Datenbank selektiert und durch den Bot dargestellt. Dort kann er die Genehmigung für die unregistrierten Benutzer geben. Er kann auch die normalen Benutzer blockieren, wenn sie einmal ein unpassendes Bild geschickt haben.

Umsetzung der Funktionalitäten:

- Zuerst wurde ein Query geschrieben, um zu überprüfen ob die Person, die eine Nachricht zum Bot geschrieben hat, schon in der Datenbank registriert ist oder nicht.

```
1 query=("SELECT c_id, role, user_status, checked, telefonnummer
2 from tb_infotainment_system_chatbot_users
3 where c_id = %s") %(int(char_id))
4 count=curs.execute(query)
5 if count > 0:
6     user=curs.fetchone()
```

Listing 1.1: Select Query für die registrierten Benutzer in die Datenbank

- Falls diese Person registriert ist, wird überprüft ob er ein Administrator oder ein normaler Benutzer ist. Falls er ein Administrator ist, werden die Nachrichten, die er zum Chatbot sendet analysiert und die entsprechenden Ergebnisse zurückgeschickt.
- Wenn der Administrator /users zum Bot schreibt, bedeutet dass er die normalen Benutzer sehen will. Es wird ein Query geschrieben, die diese Benutzer aus der Datenbank selektiert. An dem Administrator wird nur die ID des Benutzers geschickt.


```
1 if command == '/users/':
2     curs.execute('Select c_id, role, user_status from
3     tb_infotainment_chatbot_users')
4     variable1=curs.fetchall()
5     for users in variable1:
6         bot.sendMessage(chat_id, users[1])
```

Listing 1.2: Darstellung der registrierten Benutzer

- Wenn der Administrator /SeeUnregisteredUsers zum Chatbot schreibt, bedeutet dass er die unregistrierten Benutzer sehen will. Ihm werden dann die ID und die Telefonnummer von diesen Benutzer zurueckgeschickt.

```
1 if command == '/see_unregistered_users':
2     curs.execute("Select c_id, telefonnummer,
3     (row_number() over (order by c_id)
4     from tb_infotainment_chatbot_users where checked=0")
5     unconfirmed=curs.fetchall()
6     users=""
7     for user in unconfirmed:
8         users+=(str(user[2])+" "+str(user[0])+
9         " "+str(user[1])+"\n")
10    bot.sendMessage(chat_id, users)
```

Listing 1.3: Darstellung der unregistrierten Benutzer

- Wenn der Administrator /DoNotAccept und eine bestimmte Chat ID zum Chatbot schreibt, bedeutet, dass der einen unregistrierten Benutzer nicht genehmigen will.

```
1 DoNotAccept=command.split()
2 if DoNotAccept[0] == '/DoNotAccept':
3     confirmationquery=
4     ("update tb_infotainment_chatbot_users
5     set checked=2 where c_id = %s") %(int(DoNotAccept[1]))
6     curs.execute(confirmationquery)
7     conn.commit()
```

Listing 1.4: Genehmigungen für die neue Registrierungen

- Der Administrator kann bestimmte Benutzer blockieren, wenn er '/Block' und eine bestimmte Chat ID zum Chatbot schreibt.

```
1 block=command.split()
2 if block[0] == '/block':
3     updatequery = ("update tb_infotainment_chatbot_users
4     set user_status='0' where c_id = %s") %(int(block[1]))
5     curs.execute(updatequery)
6     conn.commit()
```

Listing 1.5: Blockierung der Benutzer

- Falls ein normaler Benutzer die oberen Befehle probiert, wird der Chatbot ihn eine Nachricht zurückschicken, dass er nicht der Administrator ist.
- Durch diese Funktion wird überprüft, ob die Person, die den Button Ja gedrückt hat, registriert ist oder nicht. Falls nein bittet Chatbot ihn, die Telefonnummer einzuschreiben.

```
1 def on_callback_query(msg):
2     query_id, from_id, query_data=
3     telepot.glance(msg, flavor='callback_query')
4     bot.answerCallbackQuery(query_id, text='getIt')
5     if query_data=='press1':
6         bot.sendMessage(from_id, text="ok")
7         query2=("Select c_id from tb_infotainment_chatbot_users
8                 where c_id=%s") %(int(from_id))
9         count1=curs.execute(query2)
10    if count1 > 0:
11        bot.sendMessage(from_id, text="Please write your
12        phone number below")
```

Listing 1.6: Überprüfung der Eingabe der Benutzer

- Danach erfolgt die Überprüfung ob die Telefonnummer richtig eingegeben ist oder nicht. Falls ja, wird diese Person in die Datenbank hinzugefügt. Der wartet aber auf die Genehmigung des Administrators.

```
1 query3=("insert into tb_infotainment_chatbot_users" \
2 "(c_id, user_status, role, telefonnummer, checked)"
3 "VALUES(%s, %s, %s, %s, %s)")
4 execute=(chat_id,1,555,reply,0)
5 curs.execute(query3,execute)
6 conn.commit()
```

Listing 1.7: Genehmigung des Administrators

1.8.6 Konkrete Beispiele mit Bildern

In den folgenden Bildern werden die Funktionalitäten des Administrators beim Chatbot dargestellt.

- Funktion des Administrators, die Benutzer anzuschauen. Chatbot schickt die Chat ID des Benutzers zurück, die für jeden Benutzer eindeutig ist, und auch die Telefonnummer. Diese Daten sind in der Datenbank gespeichert und werden davon selektiert.

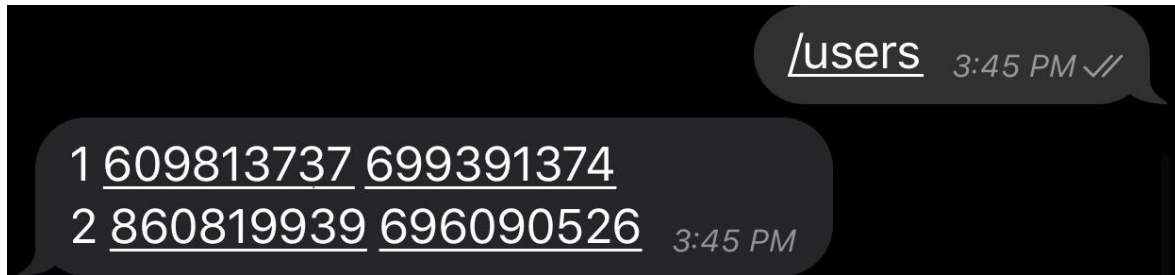


Abbildung 1.4: Auflistung der Chatbot-Benutzer

- Funktion des Administrators, die unregistrierten Benutzer anzuschauen. Chatbot schickt die Chat ID des Benutzers zurück, die für jeden Benutzer eindeutig ist, und auch die Telefonnummer. Diese Daten sind in der Datenbank gespeichert und werden davon selektiert.

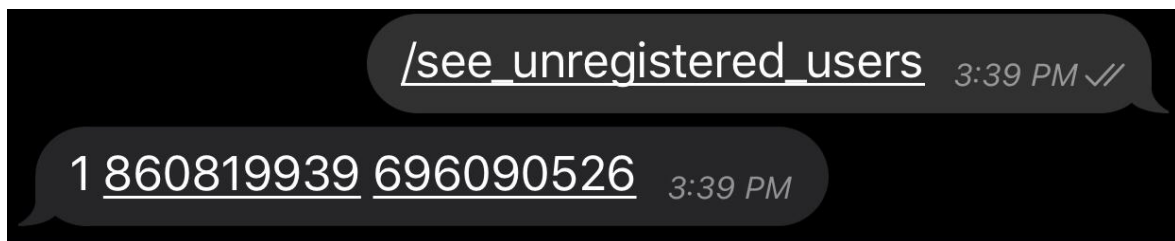


Abbildung 1.5: Auflistung der unregistrierten Benutzer

- Funktion des Administrators, die Benutzer zu blockieren. Der Administrator schreibt das einfach das Befehl /block und gibt auch die Chat-ID des Benutzers dazu. Chatbot schickt eine Nachricht zurück.

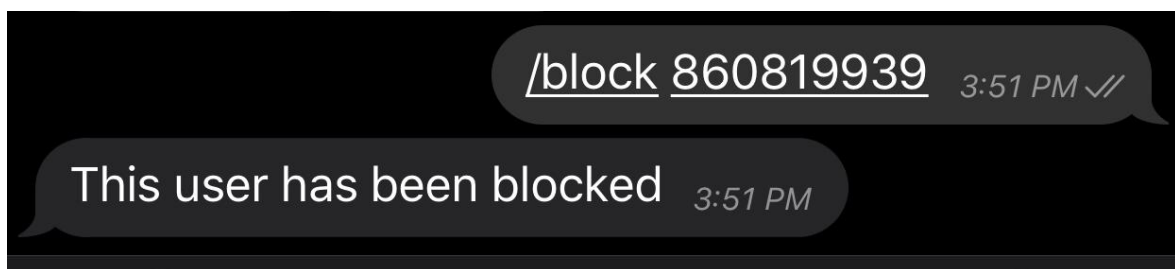


Abbildung 1.6: Benutzer Blockierung

In den folgenden Bildern werden die Funktionalitäten des unregistrierten Benutzer beim Chatbot dargestellt.

- Chatbot gibt den Benutzer die Möglichkeit, beim Chatbot zu registrieren. Der Benutzer hat zwei Buttons zur Wahl.

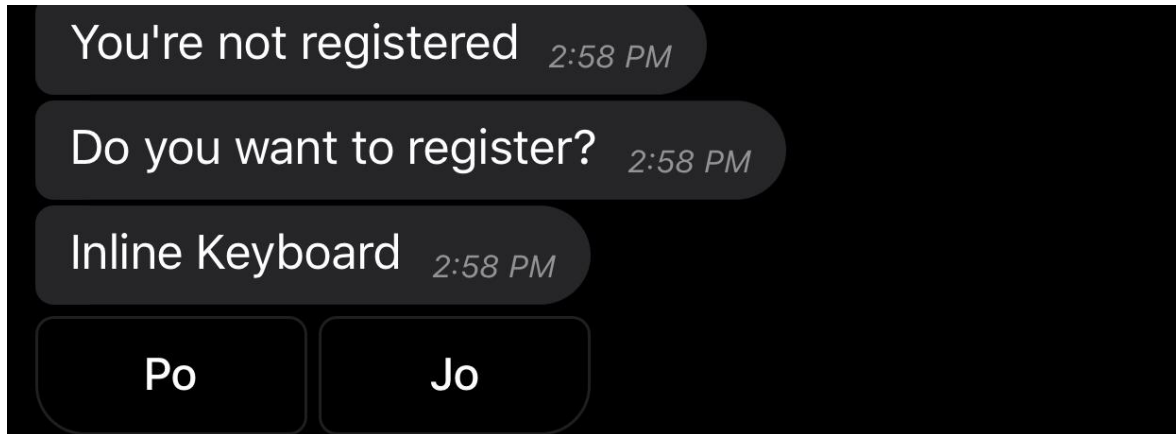


Abbildung 1.7: Registrierung beim Chatbot

- Wenn der Benutzer Ja klickt, bitten der Chatbot ihn, seine Telefonnummer einzugeben. In diesem Fall hat der Benutzer nur 2 Zahlen eingegeben. Das ist falsch, weil die Telefonnummer 10 Ziffern lang sein soll.

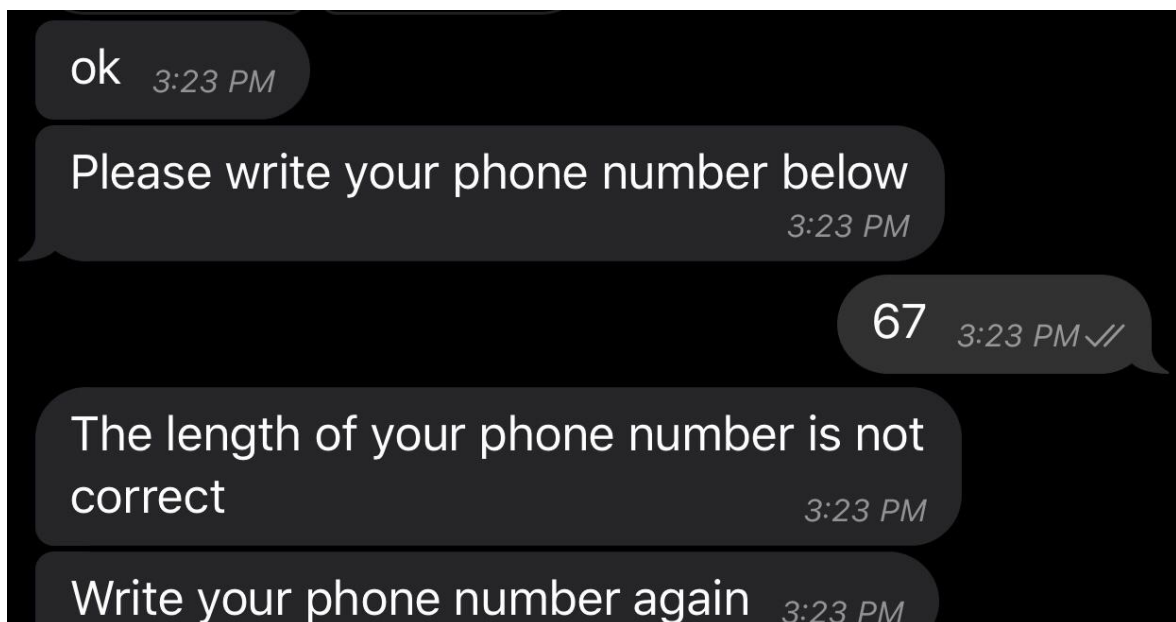


Abbildung 1.8: Falsche Eingabe des Telefonnummers beim Chatbot

- Falls der Benutzer statt eine Telefonnummer, ein Text schickt, wird der Chatbot ihm sagen, dass diese Eingabe kein korrektes Format für ein Telefonnummer ist.

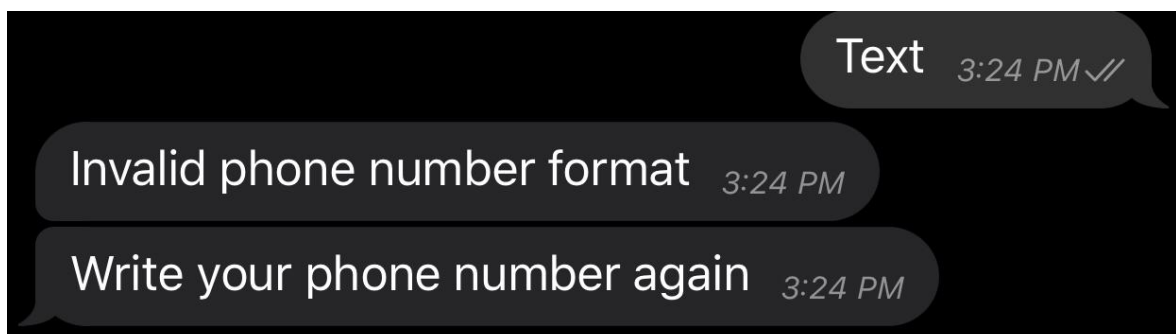


Abbildung 1.9: Falsches Format des Telefonnummers beim Chatbot

- Die Telefonnummer soll zu dem Prefix anpassen. Wenn nicht, erkennt der Chatbot das und schickt eine Fehlermeldung zurück.

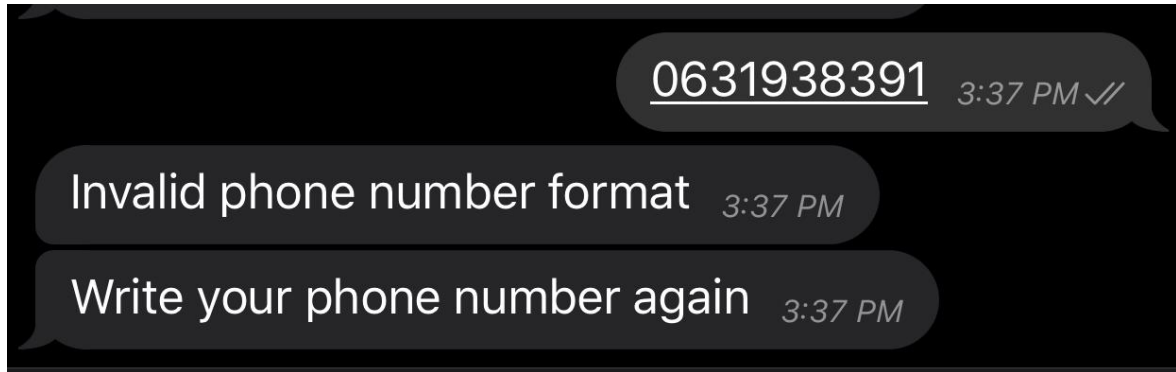


Abbildung 1.10: Eingabe einer falschen Telefonnummer beim Chatbot

- Falls der Benutzer die Telefonnummer richtig eingegeben hat, wird ihm eine Nachricht geschickt, dass seine Daten jetzt gespeichert sind. Der Benutzer soll auf die Genehmigung des Administrators warten.

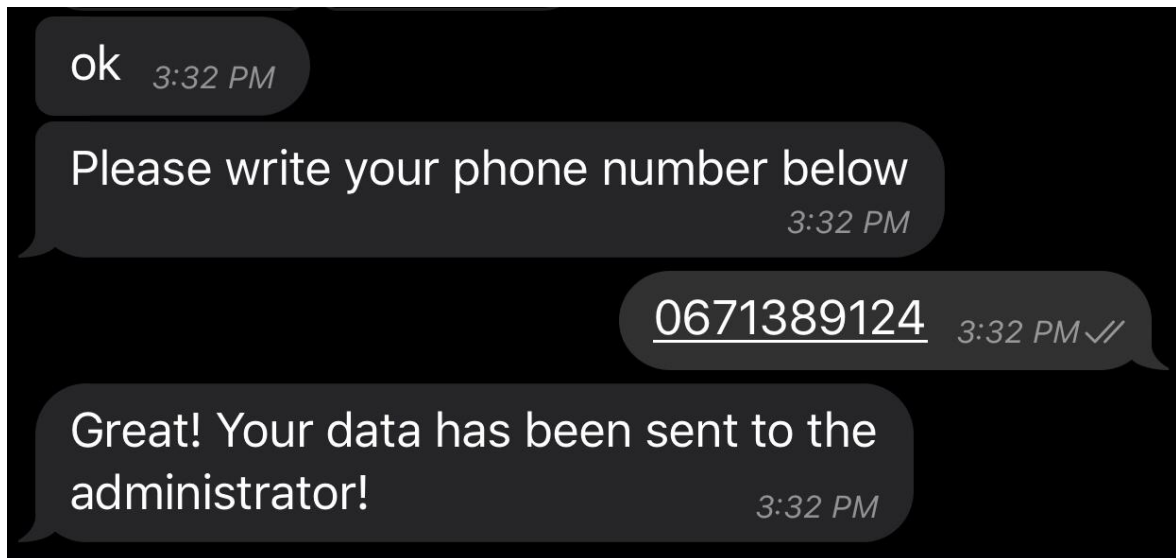


Abbildung 1.11: Korrekte Eingabe der Telefonnummer beim Chatbot

- Falls der Benutzer versucht, nachdem er einmal die Telefonnummer eingegeben hat, eine andere Telefonnummer einzugeben, wird der Chatbot ihm eine Nachricht schicken, dass er auf die Genehmigung des Administrators warten soll. Die andere Nachricht, die später kommen werden nicht berücksichtigt bis der Administrator die Genehmigung gegeben hat.

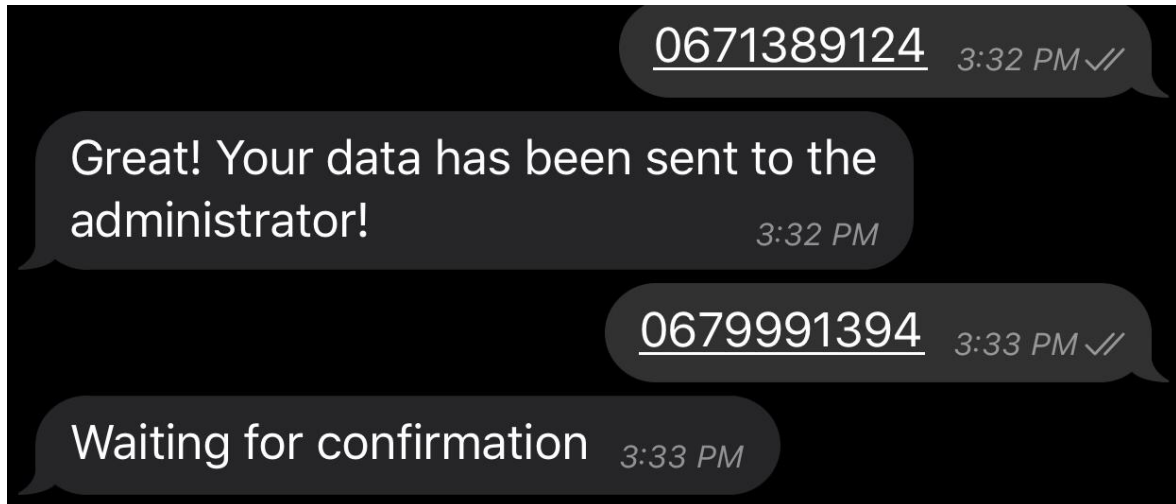


Abbildung 1.12: Das Warten der Benutzer auf die Administrator Bestätigung
In dem folgenden Bild wird die Interaktion eines blockierten Benutzer mit dem Chatbot dargestellt.

- Wenn ein Benutzer von dem Administrator blockiert ist und trotzdem versucht eine Nachricht zum Chatbot zu schicken, wird der Chatbot nur eine default Nachricht schicken, falls die vom Benutzer gesendete Nachricht im Textformat ist. Falls dieser Benutzer versucht, ein Bild zu schicken, wird der Chatbot ihm sagen, dass seine Bilder nicht mehr auf dem Bildschirm angezeigt werden.

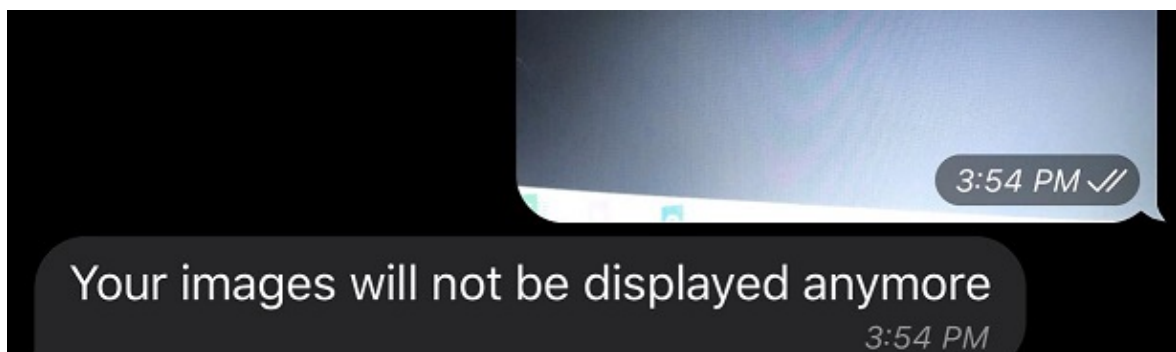


Abbildung 1.13: Nachricht von einem blockierten Benutzer

1.9 Probleme, Herausforderungen und deren Lösung

Während der Umsetzung des Projekts gab es einige Probleme und Herausforderungen, die aber am Ende gelöst wurden.

APIs

Die APIs haben einige Herausforderungen gebracht. Zuerst sollten die richtigen APIs gefunden werden. Danach sollte nachgeschaut, wie sie genau funktionieren. Es sollte

genau analysiert, auf welchem Format die API die Daten zurückliefert, damit diese Daten an dem Bildschirm angezeigt werden konnten.

Anzeige

Wie in den oberen Kapiteln erklärt wurde, war die Design Vorbereitung für die Anzeige nicht leicht. Jedes Layout wurde in unterschiedlichen Teilbereichen unterteilt und in jedem Bereich sollten verschiedene Informationen dargestellt werden. Aber die Formatierung war sehr problematisch. Einige Teilbereiche des Layouts waren sehr groß, und die Informationen passten dort wunderbar. Es gab aber andere Sektoren, wo die Information sehr schwer hinpasste. Das wurde mit JavaScript gelöst. Beispielsweise sollten die Wetterdaten in allen Sektoren hinpassen, und für jeden Sektor wurde die Formatierung der Wetterdaten anders vorbereitet, ansonsten funktionierte es nicht, und die Informationen wurden irgendwo auf dem Bildschirm angezeigt. Diese Aufgabe hat sehr viel Zeit gebraucht. Das sollte nicht für die Wetterdaten, sondern auch für die Kalenderinformationen.

Die Supplierplan Tabelle hat eine andere Herausforderung gebracht. Wenn innerhalb eines Tages viele Lehrer in dieser Tabelle eingetragen werden, wird die Tabelle zu groß und ist es schwer es anzupassen. Die gefundene Lösung hier ist die Tabelle auf 2 Seiten zu teilen, wenn sie sehr groß wird. Das wurde mit JavaScript gemacht.

Chatbot

Chatbot war auch ein neues Konzept. Die Herausforderungen im Bezug auf Chatbot waren nur am Anfang, weil es sollte die API gewählt werden und auch die Einrichtung richtig gemacht.

Datenbank

Die Datenbank war problematisch, weil es sollte genau überlegt werden, was für Tabellen gebraucht sind und wie sie verwendet werden.

Die richtige Erstellung der Prozeduren war auch ein Thema. Es sollte eine Prozedur erstellt werden, die dem Administrator bei der Webseite mit einer Drop Down Liste alle Lehrer anzeigte, die nicht fehlen, die bereit zum supplyieren sind und kein Unterricht haben. Diese Lehrer wurden dann als Supplierer eingetragen. Aber die richtige Kombination der Daten und die Erstellung dieser Prozedur war wirklich komplex und hat viel Zeit gebraucht, bis es richtig gelöst wurde.

1.10 Qualitätssicherung, Controlling

Qualität wurde gesichert, denn die Hauptanforderung des Auftraggebers vollständig erfüllt worden ist. Diese Anforderung war die Entwicklung eines Systems, die die wichtigsten Ankündigungen des Tages auf dem Bildschirm in einer digitalisierten Form darstellt. Es sollten auch die Schuldaten angezeigt werden.

Zusätzlich zu den Hauptfunktionen wurden auch andere Punkte implementiert, wie Chatbot oder die Darstellung von Wetterdaten auf dem Bildschirm. Chatbot ist ein wichtiges Komponent für die Qualitätsicherung, weil es viele Funktionen anbietet.

1.11 Ergebnisse - Irena Bala

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst.

1.11.1 Implementierung

Die wichtigsten Ergebnisse der Arbeit sind:

- Die Implementierung von einem Hauptkomponent der künstlichen Intelligenz wie Chatbot, die das entwickelte System interessanter macht.
- Design Vorbereitung für die Anzeige, dass dort die Informationen auf unterschiedliche Weise dargestellt werden können
- Anlegung einer Datenbank, die die Basis für die Speicherung aller Informationen ist
- Verwendung von APIs, die die Möglichkeit anbieten Zugriff zu verschiedenen Daten zu haben und als Schnittstelle dienen für die Implementierung von der Komponenten
- Einrichtung und Anlegung eines Servers, dass auch der Hauptteil des Systems ist und die Basis für die Zusammensetzung aller Komponenten anbietet
- Integration des Systems, damit alle Komponente mit einander verbinden können

1.12 Handbuch für die Bedienung

In diesem Kapitel ist das Handbuch für die Bedienung von Chatbot beschrieben.

1.12.1 Beschreibung der Bedienung als User

Die Bedienung von Chatbot ist eigentlich sehr leicht. Ein normaler Benutzer ist in der Lage, Bilder zum Chatbot zu schicken und die danach in die Datenbank gespeichert und automatisch auf dem Bildschirm angezeigt.

Zuerst soll im Handy Telegram Applikation heruntergeladen werden. Danach soll nach dem Benutzernamen Infotainment gesucht werden. Der Chat mit dem Infotainment Bot wird geöffnet. Jeder Benutzer, der eine Interaktion mit Chatbot haben will, soll zuerst registrieren.

Sobald der Chatbot eine Nachricht von jemand, der für das erste Mal an ihm etwas schickt, wird der Chatbot ihm fragen, ob er registrieren will oder nicht. Die Wahl kommt in Form von 2 Buttons, die die Antworten Ja und Nein beinhalten.

Für die Registrierung soll Ja geklickt werden und dann die Telefonnummer für die Registrierung soll auch eingegeben werden.

Danach soll gewartet werden, bis der Administrator die Genehmigung für die neue Registrierung gegeben hat. Sobald der Administrator diese Genehmigung gegeben hat, wird dieser Benutzer eine Nachricht bekommen und nur dann kann er Bilder zum Chatbot schicken, die an dem Bildschirm angezeigt werden.

1.12.2 Beschreibung der Bedienung als Administrator

Der Administrator von Chatbot hat andere Funktionalitäten im Vergleich mit einem Benutzer.

- Wenn die Nachricht /users zum Chatbot geschickt wird, wird dem Administrator eine Liste mit allen Chatbot Benutzer zurückgeschickt.
- Wenn die Nachricht /SeeUnregisteredUsers zum Chatbot geschickt wird, wird dem Administrator eine Liste mit allen unregistrierten Benutzer zurückgeschickt.
- Der Administrator kann die Benutzer blockieren durch die folgende Eingabe: /block und die Chat ID von dem Benutzer
- Wenn der Administrator eine Registrierung genehmigen will, soll die folgende Eingabe zu dem Chatbot geschickt werden: /Accept und die Chat ID von dem Benutzer
- Wenn der Administrator eine Registrierung nicht genehmigen will, soll die folgende Eingabe zu dem Chatbot geschickt werden: /DoNotAccept und die Chat ID von dem Benutzer

1.13 Evaluierung und Resümee

1.13.1 Planung vs Realisierung

Beim Chatbot sind die Anforderungen erfüllt. Die grundlegende Funktionalität, also Bilderaufnahme und deren Darstellung an dem Bildschirm durch Chatbot wurde vollständig gemacht. Zusätzlich sind aber viele neue Funktionen gekommen, wobei die Interaktion des Menschen mit dem Chatbot erhöht wurde.

Diese Funktionen sind:

- Anlegung der User-Rechte (Administrator, Benutzer)
- Blockierung bestimmter Benutzer
- Registrierung beim Chatbot

Die Anforderungen bei der Anzeige sind auch erfüllt geworden. Dort können verschiedene Informationen angezeigt werden. Für die Anzeige gibt es auch verschiedene Layouts, wie es schon vorher geplant war.

Datenbank wurde anders geplant, aber während der Arbeit wurden immer mehr neue Änderungen gemacht. Es sind mehrere Tabellen erstellt, als geplant war, wo die Informationen gespeichert wurden. Eine Tabelle davon ist die Tabelle für die Chatbot Benutzer. Die war früher nicht geplant, weil es wurde auch die Anlegung von mehreren User-Rechte beim Chatbot nicht geplant. Das bedeutet, dass alle Komponente des Systems mit einander im Zusammenhang sind und wenn ein davon ändert bringt diese Änderung Konsequenzen auf den anderen Komponenten. Deswegen wurden diese Punkte mit der Zeit verbessert.

1.13.2 Wertschöpfung und Lessons Learned

Es wurden die Wissen in folgenden Bereichen erweitert:

- HTML
- CSS
- PHP
- JavaScript
- Einrichtung von Datenbanken
- Konfiguration von Systemen
- Programmierung in Python
- Verwendung von APIs

Abbildungsverzeichnis

1.1	Structured Software Design	9
1.2	ERD	10
1.3	Wetterdaten	12
1.4	Auflistung der Chatbot-Benutzer	17
1.5	Auflistung der unregistrierten Benutzer	17
1.6	Benutzer Blockierung	17
1.7	Registrierung beim Chatbot	18
1.8	Falsche Eingabe des Telefonnummers beim Chatbot	18
1.9	Falsches Format des Telefonnummers beim Chatbot	18
1.10	Eingabe einer falschen Telefonnummer beim Chatbot	19
1.11	Korrekte Eingabe der Telefonnummer beim Chatbot	19
1.12	Das Warten der Benutzer auf die Administrator Bestätigung	20
1.13	Nachricht von einem blockierten Benutzer	20

Tabellenverzeichnis

1.1	Technologien	3
1.2	Bereiche und Technologien	8

Literatur

Aus dem Netz

- [1] 2daygeek. URL: <https://www.2daygeek.com/install-ssl-tls-certificate-on-apache-web-server/> (besucht am 12.10.2019).
- [2] Biteno. URL: <https://www.biteno.com/was-ist-apache/> (besucht am 01.10.2019).
- [3] Big Data Insider. URL: <https://www.bigdata-insider.de/was-ist-mysql-a-614184/> (besucht am 07.10.2019).
- [4] Dev Insider. URL: <https://www.dev-insider.de/was-ist-php-a-578773/> (besucht am 10.10.2019).
- [5] Dev Insider. URL: <https://www.dev-insider.de/was-ist-javascript-a-586580/> (besucht am 10.10.2019).
- [6] Dev Insider. URL: <https://www.dev-insider.de/was-ist-python-a-843060/> (besucht am 12.10.2019).
- [7] Dev Insider. URL: https://praxistipps.chip.de/was-ist-html-verstaendlich-erklaert_40979 (besucht am 11.10.2019).
- [8] Dev Insider. URL: <https://www.dev-insider.de/was-ist-json-a-702243/> (besucht am 20.10.2019).
- [9] Elektronik Kompendium. URL: <https://www.elektronik-kompendium.de/sites/com/1904221.htm> (besucht am 13.10.2019).
- [10] reDim. URL: <https://www.redim.de/schnittstellenprogrammierung> (besucht am 18.10.2019).
- [11] Sistrix. URL: <https://www.sistrix.de/frag-sistrix/css-cascading-style-sheets/> (besucht am 12.10.2019).
- [12] Computer Woche. URL: <https://core.telegram.org/> (besucht am 17.10.2019).
- [13] Computer Woche. URL: <https://www.tecchannel.de/a/owncloud-9-unter-ubuntu-server-16-04-lts-installieren,3277807,2> (besucht am 16.10.2019).