

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	Layout	12
1.7	Anzeige	13
1.8	Offline Betrieb	15
1.8.1	Konkrete Beispiele	16
1.9	Wetterdaten	19
1.10	Chatbot	20
1.10.1	Chatbot Einrichtung	20
1.10.2	Einrichtung des Chatbots in RaspberryPI	20
1.10.3	Grundlagen für die Umsetzung von Chatbot	21
1.10.4	Funktionalitäten von Chatbot	21
1.10.5	Umsetzung der Funktionalitäten	22
1.10.6	Konkrete Beispiele	27
1.11	Probleme, Herausforderungen und deren Lösung	37
1.12	Qualitätssicherung, Controlling	38
1.13	Ergebnisse - Irena Bala	38
1.13.1	Implementierung	38
1.14	Handbuch für die Bedienung	39
1.14.1	Beschreibung der Bedienung als User	39
1.14.2	Beschreibung der Bedienung als Administrator	39
1.15	Evaluierung und Resümee	40
1.15.1	Planung vs Realisierung	40

1.15.2 Wertschöpfung und Lessons Learned	41
--	----

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 auch 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 es die Basis für die Speicherung der benötigten Daten bildet. In die Datenbank wurden alle von den verwendeten APIs empfangene Daten gespeichert. Außerdem wurden dort auch die Schuldaten hinterlegt. Diese Daten sind ein separater Teil des Projekts, die in den folgenden Kapiteln genauer erklärt werden.

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

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 sind die Hauptbereiche des ganzen Systems. 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 dieser Komponente das Interesse des Menschen an dem System erhöhen wird.

Dies war eine allgemeine Beschreibung von den bisherigen erreichten Ergebnissen. 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 gegeben und dessen Umsetzung erklärt.

Einführung in Chatbot

Chatbot ist eine sehr wichtige Komponente im Bereich der künstlichen Intelligenz. Es bietet eine Kommunikationsschnittstelle zwischen Menschen und technischen Systemen an. Chatbot empfängt Anweisungen in Textform von Menschen und überträgt diese so, dass die Systeme diesen Anweisungen ausführt und eine Antwort zurückgibt. Basierend auf was Chatbots anbietet, werden diese als sehr schlaue Komponente angesehen, die immer häufiger 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 diese dann 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 Kommunikationsapplikationen, wie z.B. Whatsapp. Der wesentliche Unterschied ist, dass bei dieser Applikation nicht nur die Möglichkeit angeboten wird, mit anderen Menschen zu chatten, sondern auch mit Chatbots. Jeder Bot, der erstellt wird, bekommt ein Token, dass eindeutig für den Bot ist, wie die Telefonnummer, die für uns eindeutig ist.

Die komplette Funktionalität des Bots wurde in RaspberryPI Server, mithilfe der Programmiersprache Python 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 dieser Funktionen kann im Unterkapitel 4.10.5 gelesen werden.

1.1.2 Server

Das Infotainment System funktioniert wie viele andere technische Systeme auch 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. der Bildschirm, in dem die von dem Server erhaltene Daten dargestellt werden.

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

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 folgende Tabelle listet die verwendeten Technologien auf und daneben steht auch eine kurze Beschreibung für jede dieser Technologien.

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 Client Browser und Server

Tabelle 1.1: Technologien

Was ist Apache HTTP Server?

Der Apache HTTP¹ Server ist ein weltweit verbreitender Webserver. Dieser Server ist Open Source, das bedeutet, dass dafür keine Lizenz gekauft werden muss, um es zu verwenden. Es ist kompatibel auf allen Betriebssystemen, beispielsweise Linux, Windows, Mac OS und andere. Es bietet viele Versionen an, die zu unterschiedlichen Anwendungsgebiete passen und verbesserte Eigenschaften bereitstellen. Durch diesen 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

¹Hypertext Transfer Protocol

SSL² Zertifikats. Das ermöglicht die Übertragung der Daten in einer verschlüsselten Form. Die detaillierte Funktionsweise eines SSL Zertifikats wird in dem Unterkapitel 4.5 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 HTTPS³ Anforderung an den Server, auf dem die Webseite gespeichert ist.“ [2]

Was ist MySQL?

MySQL ist ein weitverbreitetes relationales Datenbanksystem. Ein relationales Datenbanksystem dient für die Speicherung der Daten in verschiedenen Tabellen in einem bestimmten Format. Die Datenbanksysteme werden allgemein zur Datenspeicherung und Datenverwaltung verwendet. Ein wichtiges Kriterium für die Datenspeicherung ist die Performance. 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 angesehen. Jeder, der versucht, Zugriff auf diese Daten zu bekommen, wird als ein Client angesehen. 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 zur Programmierung von der vom Server auszuführenden Ereignissen verwendet wird. PHP

²Secure Socket Layer

³Hypertext Transfer Protocol Secure

⁴Structured Query Language

⁵Parallel History Project

ist eine sehr verbreitete Programmiersprache, die am meisten zur Erstellung und Programmierung von Webseiten verwendet wird. Eigentlich ist PHP sehr flexibel, denn es bietet einen großen Schnittstellenansatz an. Diese Programmiersprache kann auch im Zusammenhang mit Datenbanken genutzt werden. [4]

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 als 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 Benutzerfreundlichkeit besonders erhöht, ist die Möglichkeit andere Module anzuhängen. Es ist auch eine Open Source Software, die von den Programmierern verwendet, verändert, angepasst bzw. bearbeitet werden kann. Es wird meistens für komplexe Aufgaben verwendet, deswegen wird es als eine Hochsprache betrachtet. [6]

Merkmale von Python

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

Was ist HTML?

HTML⁶ ist keine Programmiersprache, sondern wird für die Erstellung von Inhalten bei Webseiten verwendet. Diese Inhalte können Texte, Bilder oder andere Komponenten sein. HTML wird als eine Auszeichnungssprache angesehen. Sie ist nicht nur für die Erstellung von Inhalten von Webseiten zuständig, sondern auch für ihr Design. Diese Sprache legt 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, Zitate) zu entkoppeln.“ [11]

⁶HyperText Markup Language

⁷Cascading Style Sheets

Was ist Raspberry PI?

Der Raspberry PI ist ein SBC⁸-Minirechner, der zur Linux Programmierung, Shell Scripting und Realisierung von technischen Projekten verwendet wird. Es braucht eine Tastatur, eine Maus, einen Netzteil, VGA⁹-Kabel und HDMI¹⁰-VGA Konverter, damit es benutzt werden kann. Die Konfiguration von einem Raspberry PI erfolgt durch eine SD¹¹-Karte. Diese SD-Karte beinhaltet das Image, wo das Betriebssystem liegt. Ein Raspberry PI kann in Zusammenhang mit vielen anderen Komponenten verwendet werden.[9]

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 TLS¹² am meisten verwendet, da es das neueste und modernste Standard von SSL ist. [1]

SSL-Verschlüsselung

Um eine verschlüsselte Verbindung zwischen einem Browser und einem Server aufzubauen, werden SSL – Zertifikate integriert. Mittels eines SSL Zertifikats wird die Authentizität einer Webseite überprüft. Das SSL Zertifikat wird von einer Zertifizierungsstelle, erzeugt. Diese Zertifizierungsstelle heißt CA¹³ 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 übertragenen Daten einen öffentlichen und einen privaten Schlüssel. Je grösser die Länge des Schlüssels ist, desto sicherer und besser ist die Verschlüsselung. Meistens werden Schlüssel mit einer Länge von 256 Bit verwendet. [1]

Was ist Telegram Bot API?

Telegram Bot API¹⁴ ist eine Schnittstelle, die die Chatbot-Implementierung ermöglicht. Es bietet verschiedene Funktionen an, nämlich die Einrichtung, Erstellung und die Verarbeitung von Bots. Die genommenen Funktionen sind in der Dokumentation vom Telegram Bot API klar beschrieben. [12]

⁸Single Board Computer

⁹Video Graphics Array

¹⁰High-Definition Multimedia Interface

¹¹Secure Digital Memory Card

¹²Transport Layer Security

¹³Certificate Authority

¹⁴Application Programming Interface

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?

„JSON¹⁵ 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.

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 Structed Software Design von dem Server dar. Durch diese Methode wird die komplette Funktionalität des Servers dieses Systems geplant. Alle Module, die miteinander verbunden sind, sind die Komponenten des Systems, also Teil des Servers. Zwischen den Modulen erfolgt die Datenübermittlung. Der Pfeil auf Abbildung1.1, Init genannt, zeigt das Hochfahren des Servers. Nach der Servereinschaltung wird ein Request zu allen Komponenten geschickt, damit sie auch eingeschaltet 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 SQL-Query erstellt und die Daten werden in die Datenbank gespeichert. Falls nicht, wird alles im Log Modul gespeichert. Der Log Modul hat dieselbe Funktionalität für alle Komponenten. Es speichert einfach alle Daten und alle Ergebnisse. Dasselbe passiert mit der APIs auch. Die Wetter API und Website API speichern ihre Ergebnisse im Log. Wenn sie

¹⁵JavaScript Object Notation

Daten zurückliefern, werden 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.

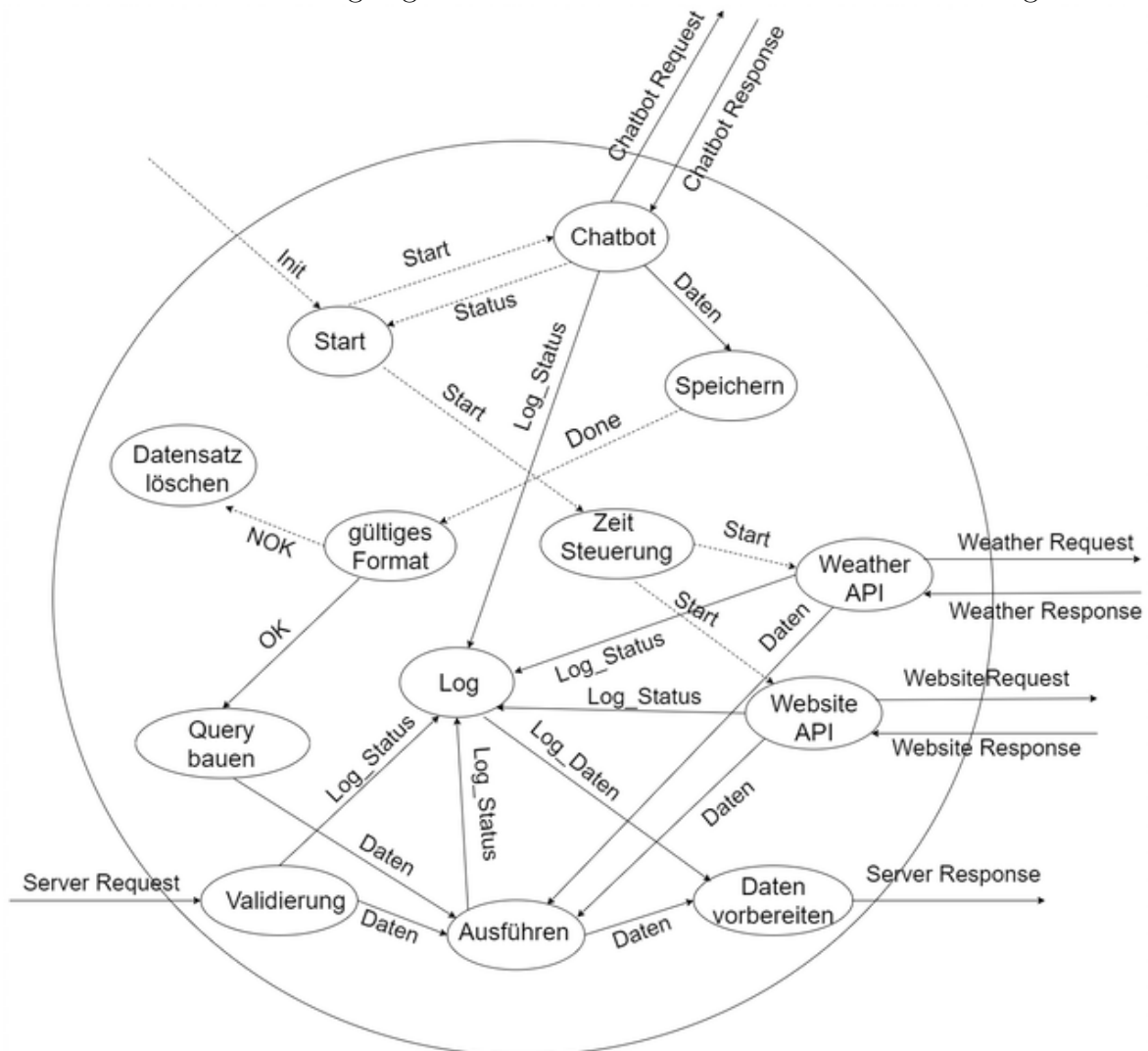


Abbildung 1.1: Structured Software Design vom Server

1.3 Konfiguration von Raspberry PI Server

Folgende Schritte sind für die Konfiguration vom RaspberryPI notwendig:

- Zuerst wurde ein Image auf RaspberryPI gespielt, 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 und 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 Datenbankrechte vergeben.
- Danach wurde SSH¹⁶ aktiviert, damit eine sichere Verbindung zu diesem Server von einem externen Gerät ermöglicht werden kann.
- Als letztens erfolgte die SSL Verschlüsselung.

1.4 Datenbank

Zuerst wurde ein ER¹⁷ Diagramm auf 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
- Chatbot Multilanguage
- Stunden
- Wetterdaten Info
- API Settings
- Kalendarinformationen

Basierend auf das ER Diagramm wurden dann die Tabellen mit phpmyadmin erstellt. Dann wurden für alle Tabellen mit MySQL Workbench die entsprechenden gespeicherten Prozeduren erstellt. Für die Anzeige von den leeren Klassen wurde eine gespeicherte Prozedur erstellt, die als Parameter den Tag und die Stunden nimmt. Danach wurden alle Räume aus der Tabelle Unterricht selektiert, die leer sind, indem es kein Unterricht stattfindet.

¹⁶Secure Shell

¹⁷Entity Relationship

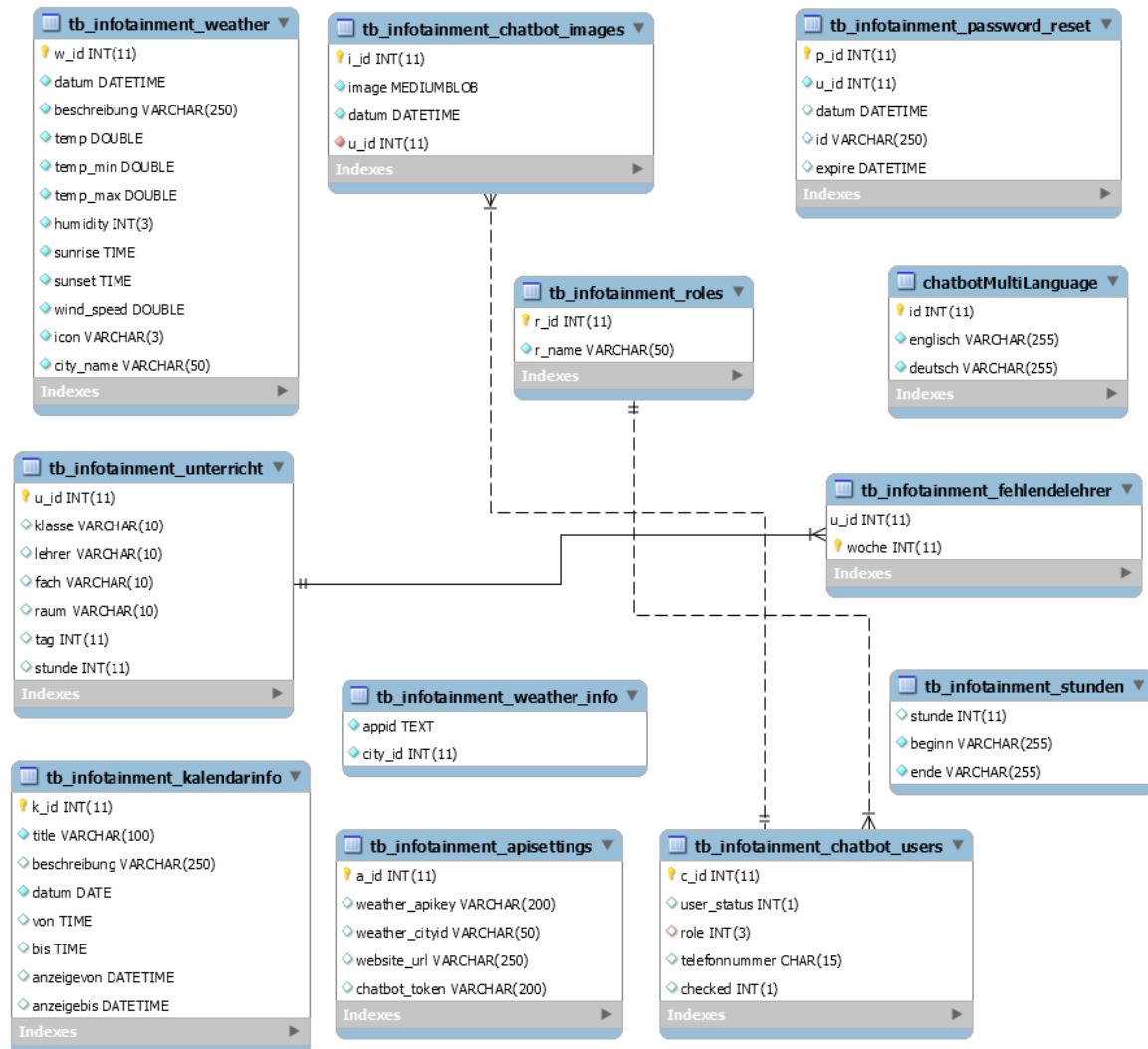


Abbildung 1.2: ERD

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]

Das Ziel bei der SSL Verschlüsselung war es, eine sichere Datenübertragung zwischen Server und Browser(Client) zu ermöglichen. Auf Abbildung 1.3 ist die SSL Zertifikate ersichtlich. Diese ist eine selbst erstellende Zertifikate. Es wurde nicht von CA¹⁸ generiert, deswegen wird es angezeigt, dass diese Zertifikate nicht vertrauenswürdig ist.

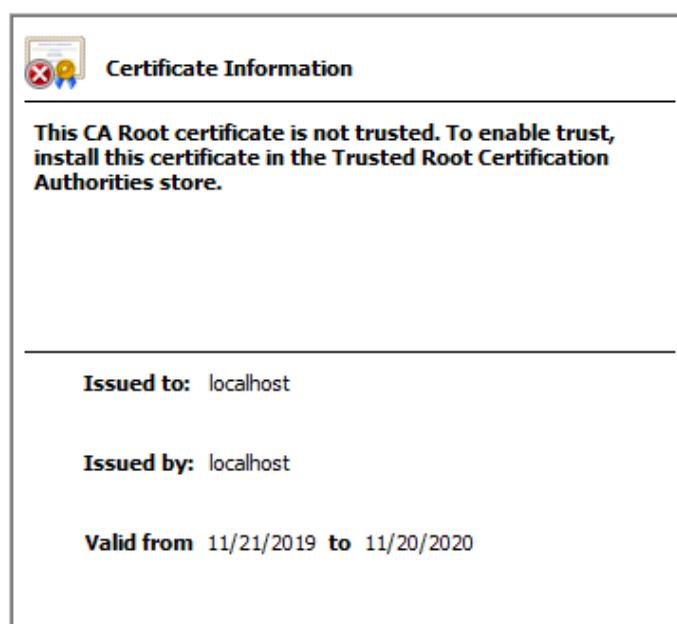


Abbildung 1.3: SSL Zertifikat

1.6 Layout

Es werden verschiedene Layouts für die Anzeige zur Verfügung gestellt. Diese Layouts wurden mit CSS formatiert. Auf Abbildung 1.4 ist Layout 2 ersichtlich, wobei die Anzeige in vier Teilbereichen geteilt wird. Der kleinste Bereich, der oben steht, ist für die Darstellung von Uhr und Datum geeignet. In dem großen Bereich werden z.B. der Stundenplan oder Supplierplan dargestellt. Die anderen Bereiche dienen für die Darstellung von der letzten Post der Webseite der Schule und für die Wetterdaten. Auf Abbildung 1.5 ist Layout 4 ersichtlich. In diesem Fall wird die Anzeige in Vollbildmodus dargestellt. Hier können z.B. die Chatbot-Bilder angezeigt werden.

¹⁸Certificate Authorities



Abbildung 1.4: Layout 2

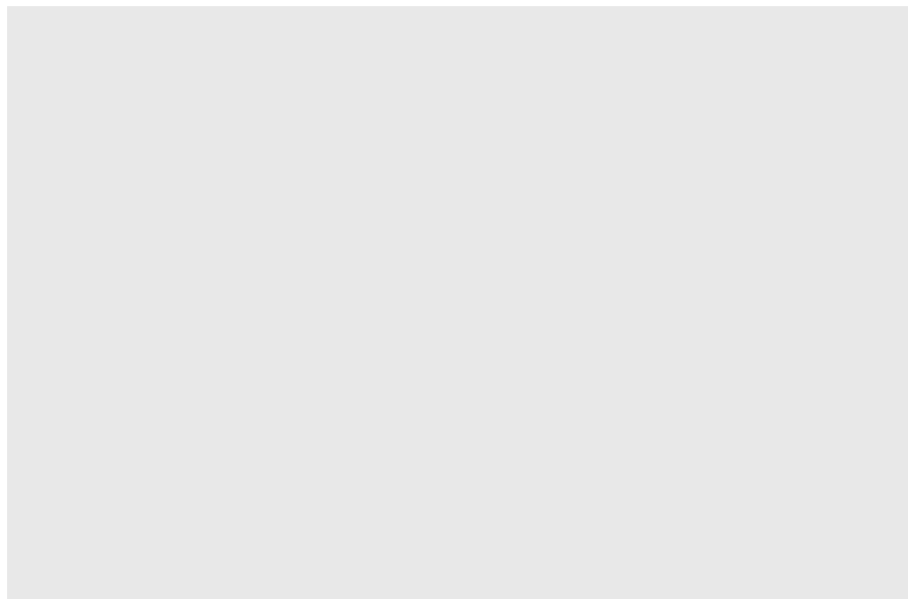


Abbildung 1.5: Layout 4

1.7 Anzeige

Bei der Anzeige sind die Wetterdaten, Kalenderinformationen, Stundenplan und die leeren Klassen dargestellt. Die Wetterdaten wurden aus dem API Call gelesen, in der Datenbank gespeichert und dann durch ein PHP Skript von der Datenbank abgelesen und angezeigt. Auf die Abbildung 1.15 ist die Darstellung der Wetterdaten auf dem Bildschirm ersichtlich.

Die Kalenderinformationen sind selbst vom Administrator geschriebene Informationen. Der Administrator kann bei der Webseite die Kalenderinformationen hinzufügen, wobei er einen Titel, eine Beschreibung, das Datum, an diesem ein bestimmtes Ereignis eintritt, und die Start, -Endezeit schreibt. Diese Daten werden danach in der Kalenderinformationen Tabelle in der Datenbank gespeichert. Für die Anzeige werden die Datensätze aus der Datenbank selektiert, später werden sie formatiert und dargestellt. Die Schnittstelle mit der Datenbank bietet die PHP Programmiersprache, während CSS für die Formatierung der Daten gebraucht wurde. Die Kalenderinformationen sind auf Abbildung 1.6 ersichtlich.

Um den Stundenplan darzustellen, wurden die Daten von der Tabelle Unterricht mittels eines PHP Skriptes selektiert. Die herausgenommenen Informationen wurden danach für die Darstellung vorbereitet. Beim Stundenplan sollten die Klassen, Gruppen, Räume und Fächer dargestellt. Bei der Darstellung wurde die HTML rowspan Attribut für die Einteilung der Klassen in Gruppen verwendet. Danach wurden die Einträge für jede Gruppe erfüllt. Der Stundenplan ist auf Abbildung 1.8 ersichtlich. Die leeren Klassen sind durch die Erstellung einer gespeicherten Prozedur von der Unterricht Tabelle herausgenommen. Diese Prozedur bekommt als Parameter den Tag und die Stunde anhand derer die leeren Klassen gefunden werden sollen. Die Formatierung von den leeren Klassen wurde, wie bei den anderen Tabellen auch, mit CSS gemacht. Die leeren Klassen sind auf Abbildung 1.7 ersichtlich.

Title/Titulli	Beschreibung/Pershkrimi	Datum/data	Beginnt / Fillon	Endet / Mbaron
Olympiade	Die Olympiade findet heute statt.	2020-02-17	12:10:00	13:10:00
Sporttag	Der Sporttag findet heute statt.	2019-11-25	12:15:00	13:10:00

Abbildung 1.6: Kalenderinformationen

K21, BIB, K24, HLAB, S-LAB, HW1, PR1, HW2, K04, GYM

Abbildung 1.7: Leere Klassen für Montag, die zweite Stunde

Klasse	1			2			3			4			5			6			7			8			9			10				
	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum	Fach	Lehrer	Raum		
6a	D	ABN	K03	AM	PRL	K03	BSPK	ZHN	BSPK	BSPK	ZHN	BSPK	ME	SHM	BIB	D	HOM	K11														
	D	HOM	K15				BSPM	GJA	BSPM	BSPM	GJA	BSPM				E	GIC	K03														
6b	GG-AL	TAF	K04	D	OFM	K15	BCH	CEG	NWL	AL	SHB	K04	SYT	KAF	PR5	TEW	KAF	PR5														
				D	ABN	K04							E	NED	K04	D	ABN	K04														
6c	AL	SHB	K05	SBK	DUE	K05	BSPK	ZHN	BSPK	BSPK	ZHN	BSPK	AM	FIO	K05	GG-AL	TAF	K05														
							BSPM	GJA	BSPM	BSPM	GJA	BSPM																				
7a	AM	FIO	K01	BCH	CEG	NWL	D	ABN	K01	D	ABN	K01	BSPM	GJA	BSPM	AL	SHB	K01	AL	SHB	K01	ME	DUE	BIB								
							D	PEN	K05	SBK	PEN	K03	BSPK	ZHN	BSPK																	
7b	E	PEN	K27	D	HOM	K27	AM	SAL	K02	Ph	SAL	K02	AL	SHB	K02	SYT	KAE	SW2	SYT	KAE	SW2											
	E	NED	K02	E	NED	K02										D	ABN	K02														
8a	D	AIT	K11	D	AIT	K11	D	SCR	K11	D	DUE	K11	AM	PRL	K11	D	DUE	K13	Ph	PRL	NWL	BSPK	ZHN	BSPK	BSPK	ZHN	BSPK					
	SYT	KAE	PR1	SYT	KAE	PR1	SYT	KAF	PR5	SYT	E	GIC	PR5			E	NED	K11				BSPM	GJA	BSPM								
8b	Ph	PRL	K12	AL	MEI	K12	AM	FIO	K12	E	SYT	SCG	SW2	D	HOM	K23	H-AL	TAF	K12	ME	DUE	BIB	BSPM	GJA	BSPM	BSPM	GJA	BSPM				
														D	BAS	K12																
8c	BCH	CEG	NWL	D	LIB	K13	D	LIB	K13	AL	MEI	K13	AL	MEI	K13	BSPK	ZHN	BSPK	BSPK	ZHN	BSPK											
				SYT	SCG	SW2	SYT	SCG	SW2							BSPM	GJA	BSPM														
9a	D	HEK	PR5	D	HEK	PR5	SYT	ABW	K26	AL	KAB	K26	AL	KAB	K26	MED2	BUB	CLAB	MED2	BUB	CLAB	CPH	BUB	HW1	CPH	BUB	HW1					
	D	SCR	K26	D	SCR	K26										CPS	PLE	PR3	CPS	PLE	PR3	CPH	ALB	PR5								
9b	AL	KAB	K14	AL	KAB	K14	D	AIT	K14	E	NED	K14	K	SCG	K14	ITSS	ABW	K14	AM	SAL	K14	Ph	SAL	K14	H-AL	NEG	K14					
							CPS	PLE	PR1	CPS	ABW	PR1																				
9e	E	GIC	PR4	E	GIC	PR4	D	HOM	K15	D	HOM	K15	AM	SAL	K15	K	SCG	K15	AL	KAB	K15	GG-AL	NEG	K15	Ph	SAL	K15					
	SYT	KAW	HW1	SYT	KAW	HW1	D	OFM	K25	D	OFM	K25																				
2a	NWTK	NUF	K28	AM	SAL	K28	CPS	MAM	PR4	D	SCR	K28	SYT	ABW	HLAB	SYT	ABW	HLAB	MED2	FAJ	PR1	MED2	FAJ	PR1								
							E	GIC	K28	CPS	MAM	PR4	D	OFM	K28	D	OFM	K28	CPH	KAW	HW2											
2b	AM	SAL	K22	NWTK	NUF	K22	CPH	KAW	HW2	CPH	KAW	HW2	CPH	KAW	HW2	CPH	KAW	HW2	SYT	KAF	HLAB	SYT	KAF	HLAB								
							D	HEK	K22	D	HEK	K22	D	HEK	K22	D	HEK	K22	D	HEK	K22	MED2	OFF	MLAB								
3a	AL	MEI	K23	SEW	MAM	K23	SYT	STD	K23	E	LIB	K23	MED2	SCC	PR2	MED2	SCC	PR2														
													MED2	OFF	MLAB	MED2	OFF	MLAB														
3b	SEW	MAM	K25	SYT	ABW	K25	MED2	SCC	PR2	MED2	SCC	PR2	INSY	FAJ	K25	AM	PRL	K25	AL	MEI	K25	AL	MEI	K25	Ph	FAJ	NWL					
							MED2	OFF	MLAB	MED2	OFF	MLAB																				
4a	INSY	SCC	PR2	SEW	KUA	PR3	SEW	KUA	PR3	NWTK	KUA	PR3	D	SCR	K27	D	SCR	K27	WIR_3	SCG	K27	AM	KAE	K27								
	ARM	STD	SYTLAB	ARM	STD	SYTLAB	NWTK	BUB	CLAB	NWTK	BUB	CLAB																				
4b	SEW	KUA	PR3	E	PEN	K21	AL	MEI	K21	AM	KAE	K21	NWTK	KUA	PR3	AM	SAL	K21	MTEC	SCC	PR2	MTEC	SCC	PR2								
5a	E	LIB	K24	INSY	SCC	K24	H-AL	TAF	K24	SVS	NUF	S-LAB	SVS	NUF	S-LAB	SEW	KUA	PR3	D	OFM	K24	D	OFM	K24								
										III	STD	SYTLAB	III	STD	SYTLAB	III	STD	SYTLAB														
5b	MTEC	OFF	MLAB	MTEC	OFF	MLAB	AL	KAB	K27	AM	PRL	K24	H-AL	TAF	K24	AM	KAE	K24	QM2	ALB	K26	WIR_2	SCG	SW2	WIR_2	SCG	SW2					

Abbildung 1.8: Stundenplan

1.8 Offline Betrieb

Der Offline-Betrieb ist aus mehreren Gründen für das Infotainment-System sehr wichtig. Durch den Offline-Betrieb werden die Displays, die als Clients dienen, bei Verbindungsausfall nicht vollständig ausgeschaltet, sondern weiterhin Informationen anzeigen. Da die Displays die Daten kontinuierlich vom Server empfangen, würde ohne Strom die Verbindung zum Server getrennt und die Bildschirme ausgeschaltet. Daher war die Implementierung von Offline-Betrieb äußerst notwendig. Der Offline-Betrieb dient zur Synchronisierung von Daten zwischen Client und Server. Es ermöglicht ein kontinuierliches Backup. Wenn kein Strom vorhanden ist, zeigen die Bildschirme weiterhin die letzten Daten an, die sie vom Server empfangen haben. Die Datenübertragung erfolgt durch Aufrufen eines Skripts - mithilfe von crontab Befehl - - alle 5 Minuten. Crontab dient dazu, ein bestimmtes Ereignis regelmäßig auszuführen. Der Client erhält also automatisch alle Änderungen, die auf dem Server auftreten. Das Skript zum Synchronisieren von Daten wurde manuell mit der PHP-Programmiersprache erstellt.

- Es wurde ein PHP-Skript für die Implementierung von Offline Betrieb erstellt.
- Zuerst wurde ein Array mit Tabellennamen, die ebenfalls an den Client übergeben werden müssen, erstellt. Diese sind die Namen der Tabellen, die im Server liegen.
- Es wurde dann eine Verbindung mit dem Server gebaut.
- Es wurde ein SQL-Query gebaut, wo alle Daten von den Displays, oder Clients selektiert wurden.

- Es wurde eine Verbindung mit jedem Display gemacht.
- Nachdem die Verbindung mit den Displays gemacht wurde, wurden die Datensätze von den Server-Tabellen, deren Namen im ersten Array sind, selektiert.
- Danach wurden die Spalten von diesen Tabellen mithilfe von dem Befehl Show Columns aufgelistet. Auf listing 1.1 ist die Auflistung der Spalten ersichtlich.

```
1 $st="SHOW COLUMNS FROM ".$tables[$i].";";
2 $pdo = $con->prepare($st);
3 $pdo->execute();
4 $res = $pdo->fetchAll(PDO::FETCH_ASSOC);
```

Listing 1.1: Implementierung von Offline Betrieb

- Es wurden dann die Tabellen, die im Client und im Server sind, verglichen. Wenn im Client fehlende Tabellen gibt, wurde dann ein Query gebaut, wo alle Tabellen die im Server sind, aber im Client fehlen, erstellt werden und die Spalten auch. Auf listing 1.2 ist die Erstellung der fehlenden Tabellen im Client ersichtlich.

```
1 try{
2     $stmt="CREATE TABLE IF NOT EXISTS ".$tables[$i]. "(";
3     for($k=0; $k<sizeof($res);$k++) {
4         if($k!=sizeof($res)-1){
5             $stmt.=' '.$res[$k]['Field'].' '.$res[$k]['Type'].' ,';
6         }else{
7             $stmt.=' '.$res[$k]['Field'].' '.$res[$k]['Type'].' )';
8         }
9     }
10    $pdo = $Connection->prepare($stmt);
11    $pdo->execute();
12 }catch(PDOException $e){
13     echo "<br> Create failed:". $e->getMessage();
```

Listing 1.2: Offline Betrieb-Erstellung der fehlenden Tabellen im Client

- Mithilfe der Funktion arraydiff in Python, wurde ein Vergleich zwischen den Tabellen und deren Datensätze im Client und im Server gemacht. Falls diese Funktion nichts zurückgibt, bedeutet es, dass der Client schon auf dem neuesten Stand ist. Falls nicht, wird alles was im Server ist, aber im Client nicht, im Client hinzugefügt.

1.8.1 Konkrete Beispiele

- In der folgenden Abbildung 1.9 ist phpMyAdmin von einem Display mit fehlenden Tabellen ersichtlich. Die Infotainment Datenbank existiert schon, aber die Tabellen und deren Datensätze fehlen. Es wurde ein Select-Query aufgebaut, wobei alle Tabellen von Infotainment System Datenbank selektiert worden sind, aber es kommt eine leere Ergebnismenge zurück. Das bedeutet, dass momentan die Datenbank leer ist.

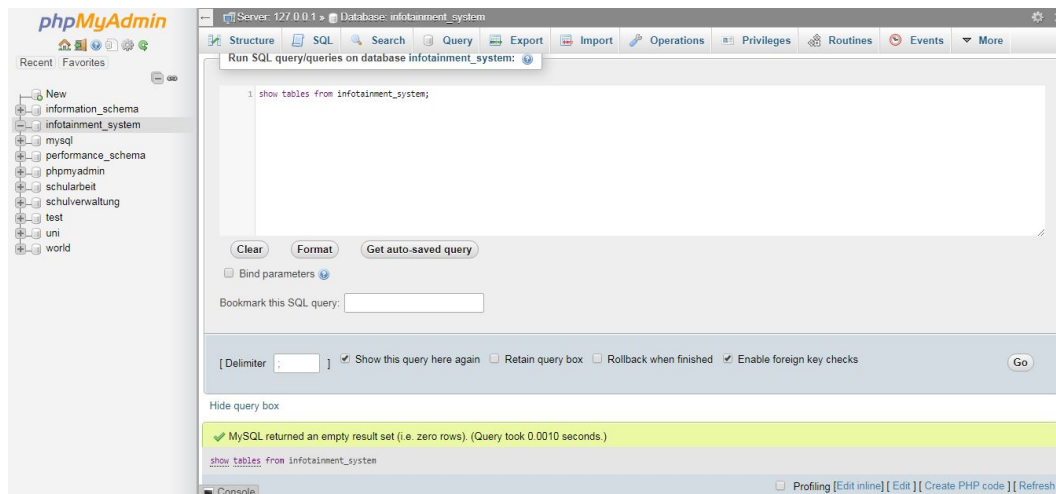


Abbildung 1.9: Display mit fehlenden Tabellen

- Dieses Display/Client soll die Daten vom Server übernehmen, deswegen wurde das Offline Betrieb Skript ausgeführt, damit dieses Display den gleichen Zustand wie der Server bekommt. Auf Abbildung 1.10 ist die Ausführung von Offline Betrieb Skript ersichtlich.

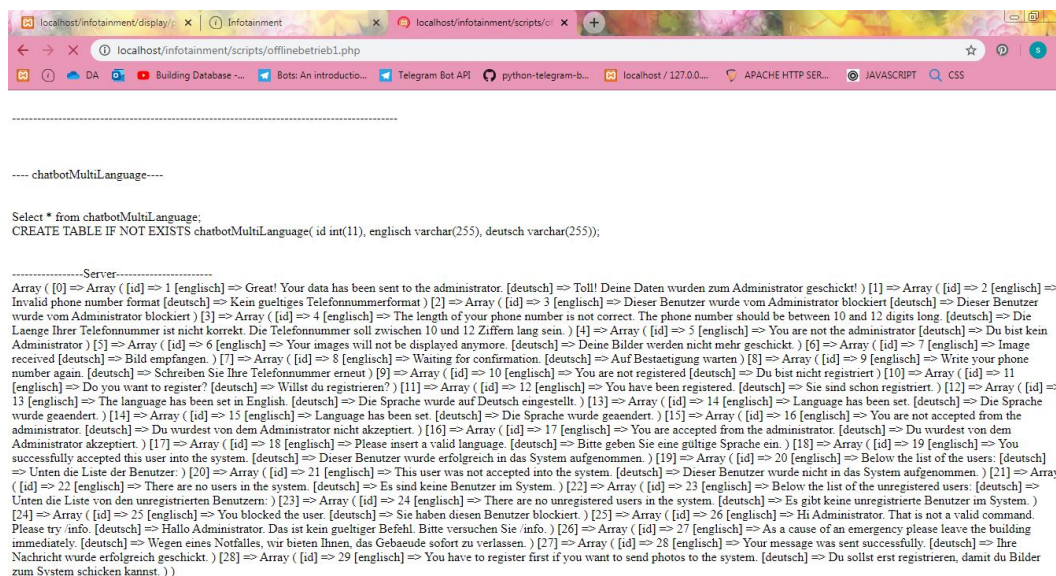


Abbildung 1.10: Ausführung Offline Betrieb Skript

- Nachdem dieses Skript ausgeführt wird, werden die Tabellen, die in der Datenbank von Server liegen, im Client nacheinander erstellt. In der folgenden Abbildung 1.11 ist die Erstellung von den ersten zehn Tabellen ersichtlich.

Table	Action	Rows	Type	Collation	Size	Overhead
chatbotmultilanguage	Browse Structure Search Insert Empty Drop	29	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_apisettings	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_chatbot_users	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_display	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_fehlendelehrer	Browse Structure Search Insert Empty Drop	44	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_kalendario	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_klasse	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_language	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_layout	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	3.5 MiB	-
tb_infotainment_layout_sections	Browse Structure Search Insert Empty Drop	13	InnoDB	latin1_swedish_ci	16 KiB	-
10 tables	Sum	109	InnoDB	latin1_swedish_ci	3.7 MiB	0.8

Abbildung 1.11: Erstellung der Tabellen im Client

- Auf Abbildung 1.12 ist die Erstellung von 16 Tabellen im Client ersichtlich. Das Skript wird weiter ausgeführt.

Table	Action	Rows	Type	Collation	Size	Overhead
chatbotmultilanguage	Browse Structure Search Insert Empty Drop	29	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_apisettings	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_chatbot_users	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_display	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_fehlendelehrer	Browse Structure Search Insert Empty Drop	44	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_kalendario	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_klasse	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_language	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_layout	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	3.5 MiB	-
tb_infotainment_layout_sections	Browse Structure Search Insert Empty Drop	13	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_location	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_roles	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_roles	Browse Structure Search Insert Empty Drop	32	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_roles	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_roles	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16 KiB	-
tb_infotainment_roles	Browse Structure Search Insert Empty Drop	721	InnoDB	latin1_swedish_ci	64 KiB	-
16 tables	Sum	881	InnoDB	latin1_swedish_ci	3.8 MiB	0.8

Abbildung 1.12: Weitere Erstellung der Tabellen im Client

- Auf Abbildung 1.13 ist die vollständige Erstellung von 18 Tabellen im Client ersichtlich. Also, das Display/der Client hat alle Tabellen vom Server bekommen.

The screenshot shows the phpMyAdmin interface for the 'infotainment_system' database. The left sidebar lists the database structure, and the main panel displays a table of 18 tables. Each table entry includes its name, a star icon, and a set of action icons (Browse, Structure, Search, Insert, Empty, Drop). The table summary at the bottom indicates 18 tables, a total of 1,679 rows, and a total size of 3.9 MiB.

Table	Action	Rows	Type	Collation	Size	Overhead
chatbotmultilanguage	Star	29	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_apisettings	Star	1	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_chatbot_users	Star	3	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_display	Star	4	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_fehlendelehrer	Star	44	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_kalendarinfo	Star	2	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_klasse	Star	6	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_language	Star	6	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_layout	Star	5	InnoDB	latin1_swedish_ci	3.5 M18	-
tb_infotainment_layout_sections	Star	13	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_location	Star	2	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_roles	Star	2	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_supplieren	Star	32	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_timetable	Star	6	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_timetable_layout	Star	9	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_unterricht	Star	1,699	InnoDB	latin1_swedish_ci	96 K18	-
tb_infotainment_users	Star	6	InnoDB	latin1_swedish_ci	16 K18	-
tb_infotainment_weather	Star	414	InnoDB	latin1_swedish_ci	64 K18	-
Sum		1,679	InnoDB	latin1_swedish_ci	3.9 M18	0 B

Abbildung 1.13: Vollständige Erstellung der Tabellen im Client

- Auf Abbildung 1.14 ist das Hinzufügen von Datensätzen zu den Tabellen ersichtlich. In diesem Fall wurde die Tabelle Kalenderinformationen als Beispiel genommen. Es werden also nicht nur die Tabellen im Client erstellt, sondern auch die entsprechenden Datensätze zu den Tabellen hinzugefügt.

The screenshot shows the phpMyAdmin interface for the 'infotainment_system' database, specifically for the 'tb_infotainment_kalendarinfo' table. The 'Insert' tab is active, and a message indicates that the current selection does not contain a unique column. The table structure is shown with columns: k_id, title, beschreibung, datum, von, bis, anzeigevon, and anzeigebis. Two rows of data are displayed.

k_id	title	beschreibung	datum	von	bis	anzeigevon	anzeigebis
2	Olympiade	Die Olympiade findet heute statt.	2020-02-17	12:10:00	13:10:00	2020-02-17 11:25:00	2020-03-02 16:20:00
3	Sporttag	Der Sporttag findet heute statt.	2019-11-25	12:15:00	13:10:00	2020-02-17 11:20:00	2020-02-17 15:20:00

Abbildung 1.14: Hinzufügen von Datensätzen zu den Tabellen

1.9 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 letztes werden mithilfe von PHP die in der Datenbank gespeicherten Parameter aufgerufen und auf dem Bildschirm angezeigt.

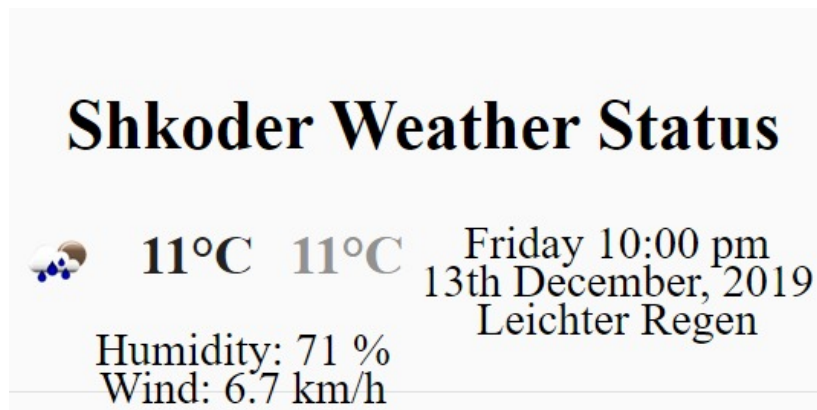


Abbildung 1.15: Wetterdaten

1.10 Chatbot

In diesem Unterkapitel wird der Chatbot und dessen Umsetzung erklärt.

1.10.1 Chatbot Einrichtung

- Zuerst wurde die Applikation Telegram auf dem Handy heruntergeladen.
- Als nächstes wurde 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 mit dem Befehl /newbot geschickt, um einen neuen Bot zu erstellen.
- Der Botfather fragte danach, wie das neue Bot genannt wird und was für ein Benutzername es haben wird.
- Anschliessend wurde von dem Botfather das Token des neuen Bots bekannt gegeben, mit dem der Bot weiterentwickelt werden kann.
- Nach der Durchführung dieser Schritten, ist der Bot fertig erstellt geworden. Es könnte dann mit dem angegebenen Benutzernamen durchsucht werden und der entsprechende Chat damit geöffnet werden.

1.10.2 Einrichtung des Chatbots in RaspberryPI

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

1.10.3 Grundlagen für die Umsetzung von Chatbot

Als erstens muss ein Python Skript erstellt werden. Danach wurden die benötigten Module 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 auf dem erstellten Bot möglich. Mit den dargestellten Funktionen sollte die Verbindung zu dem Infotainment Bot erfolgen und gleichzeitig sollte auch getestet werden, 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 werden. Das wurde im zweiten Schritt gemacht.

1.10.4 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.

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 Buttons, 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, ("Registrierung nicht durchgeführt, Interaktion nicht möglich").

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 es eine Zahl ist, soll es zwischen 10 und 12 Ziffern lang sein und mit einem Prefix anfangen. Ansonsten wird es nicht genehmigt und der unregistrierter Benutzer bekommt wieder die Möglichkeit vom Chatbot die Nummer richtig zu schreiben.

Der Infotainment Bot bietet dem unregistrierten Benutzer die Möglichkeit, die Sprache zu ändern. Dieser Benutzer kann zwischen zwei Sprachen wählen, nämlich zwischen Englisch und Deutsch. Basierend auf der ausgewählten Sprache, ermöglicht der Infotainment Bot die Interaktion mit dem Benutzer in dieser Sprache. Daher werden alle Nachrichten in der ausgewählten Sprache an Benutzer geschickt. Momentan stellt der Infotainment Bot nur diese zwei Sprachen zur Verfügung. Das System ist aber skalierbar, was bedeutet, dass es leicht erweitert und adaptiert werden kann. Das heißt, es können dem Benutzer auch andere Sprachen angeboten werden.

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 nach der Registrierung in der Datenbank gespeichert. Diese Benutzer sind vom 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 die anderen Nachrichtformate nicht interessieren. Außerdem hat der normale Benutzer, die Möglichkeit, die Sprache auszuwählen. Der Infotainment Bot stellt die Sprachen Englisch und Deutsch zur Verfügung. Je nachdem was man wählt, wird die Interaktion zwischen Infotainment Bot und den Benutzer in der ausgewählten Sprache gemacht.

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. Der Administrator kann im Falle eines Notfalls eine Nachricht zu allen Benutzern schicken, um sie zu benachrichtigen.

Wie alle anderen Benutzer, kann der Administrator die Sprache auswählen, in der die Interaktion mit dem Infotainment Bot stattfinden soll.

1.10.5 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. Auf listing 1.3 ist das Select Query für die registrierten Benutzer in die Datenbank ersichtlich.

```
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 1.3: Select Query für die registrierten Benutzer in die Datenbank

- Falls diese Person registriert ist, wird überprüft welche Rechte er hat. 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 alle Benutzer sehen will. Es wird ein Query geschrieben, die alle Benutzer aus der Datenbank selektiert. An dem Administrator wird die ID der Benutzer und deren Telefonnummer geschickt. Wenn es keine Benutzer gibt, wird ihm eine Nachricht zurückgeschickt, dass es momentan im System keine Benutzer gibt. Auf listing 1.4 ist die Umsetzung der Darstellung aller Benutzer ersichtlich.

```
1 if command == '/users':
2     userscommand=('SELECT c_id, telefonnummer, (row_number() over (order
3         by c_id)), role, checked,
4     user_status from tb_infotainment_chatbot_users')
5     count4=curs.execute(userscommand)
6     variable1=curs.fetchall()
7     if count4>0:
8         users2=""
9         for users in variable1:
10             users2+=(str(users[2])+"|"+" userid: "+str(users[0])+"telefon:
11                 "+str(users[1])+"\n")
12             bot.sendMessage(chat_id, txt[19][language])
13             bot.sendMessage(chat_id, users2)
14     else:
15         bot.sendMessage(chat_id, txt[21][language])
```

Listing 1.4: Darstellung aller Benutzer

- Wenn der Administrator /SeeUnregisteredUsers zum Chatbot schreibt, bedeutet es dass er die unregistrierten Benutzer sehen will. Ihm werden dann die ID und die Telefonnummer von diesen Benutzern zurückgeschickt. Wenn es keine unregistrierte Benutzer im System gibt, wird dem Administrator eine Nachricht geschickt: Es gibt keine unregistrierte Benutzer. Auf 1.5 ist die Umsetzung der Darstellung der unregistrierten Benutzer ersichtlich.

```
1 if command == '/see_unregistered_users':
2     unregisteredcommand=("Select c_id, telefonnummer,
3         (row_number() over (order by c_id)
4     from tb_infotainment_chatbot_users where checked=0")
5     count5=curs.execute(unregisteredcommand)
6     unconfirmed=curs.fetchall()
7     if count5>0:
8         users=""
9         for user in unconfirmed:
10             users+=(str(user[2])+"|"+" userid: "+str(user[0])+"
11                 "telefon: "+str(user[1])+"\n")
12             bot.sendMessage(chat_id, txt[22][language])
13             bot.sendMessage(chat_id, users)
14     else:
15         bot.sendMessage(chat_id, txt[23][language])
```

Listing 1.5: Darstellung der unregistrierten Benutzer

- Wenn der Administrator /DoNotAccept und eine bestimmte Chat ID zum Chatbot schreibt, bedeutet, dass der einen unregistrierten Benutzer nicht genehmigen wird. Das bedeutet, dass alle von diesem Benutzer empfangenen Nachrichten werden vom ChatBot nicht berücksichtigt. Wenn der Benutzer, der vom Administrator nicht akzeptiert wurde, versucht, Interaktion mit dem Infotainment Bot zu haben, wird ihm eine Nachricht geschickt, dass die Interaktion nicht möglich ist, weil er nicht genehmigt wurde. Auf listing 1.6 ist die Umsetzung von diesem Fall ersichtlich.

```
1 if usermng[0]== '/DoNotAccept':
2 donotacceptquery=("update tb_infotainment_chatbot_users
3 set checked=2 where c_id = %s") %(int(usermng[1]))
4 curs.execute(donotacceptquery)
5 conn.commit()
6 bot.sendMessage(usermng[1],txt[15][language])
7 bot.sendMessage(chat_id, txt[20][language])
```

Listing 1.6: Genehmigungen für die neue Registrierungen

- Wenn der Administrator /Accept und eine bestimmte Chat ID zum Chatbot schreibt, bedeutet, dass der einen unregistrierten Benutzer genehmigen will. Das bedeutet, dass dieser Benutzer hat den Status von einem normalen Benutzer gekriegt. Dieser Benutzer bekommt auch eine Nachricht vom Chatbot, dass er vom Administrator akzeptiert wurde und dadurch ist die Interaktion mit dem Infotainment Bot möglich. Auf listing 1.7 ist die Umsetzung der Genehmigungen des Administrators für die neue Registrierungen ersichtlich.

```
1 if usermng[0]== '/Accept':
2 acceptquery=("update tb_infotainment_chatbot_users
3 set checked=1 where c_id = %s") %(int(usermng[1]))
4 curs.execute(acceptquery)
5 conn.commit()
6 bot.sendMessage(usermng[1],txt[16][language])
7 bot.sendMessage(chat_id, txt[18][language])
```

Listing 1.7: Genehmigungen für die neue Registrierungen

- Der Administrator kann bestimmte Benutzer blockieren, wenn er '/Block' und eine bestimmte Chat ID zum Chatbot schreibt. Wenn die Benutzer, die von dem Administrator blockiert wurden, versuchen, Interaktion mit dem Infotainment Bot zu haben, wird ihnen eine Nachricht geschickt, dass sie blockiert sind und die Interaktion nicht möglich ist. Auf 1.8 ist die Umsetzung der Benutzer-Blockierung ersichtlich.

```
1 if usermng[0] == '/block':
2 updatequery = ("update tb_infotainment_chatbot_users
3 set user_status='1' where c_id = %s") %(int(usermng[1]))
4 curs.execute(updatequery)
5 conn.commit()
6 bot.sendMessage(chat_id, txt[24][language])
```

Listing 1.8: Blockierung der Benutzer

- Im Falle eines Notfalles, kann der Administrator an alle Benutzer eine Nachricht schicken, um sie vorher zu warnen. Auf listing 1.9 ist die Umsetzung der Notfall Nachrichten ersichtlich.

```
1 if command=='/notfall':
2 querynotfall=("select c_id, user_status from tb_infotainment_chatbot_
   users where role!='777'")
3 curs.execute(querynotfall)
4 variablenotfall=curs.fetchall()
5 for not in variablenotfall:
6     idd=not[0]
7     bot.sendMessage(str(idd),txt[26][language])
8 bot.sendMessage(chat_id, txt[27][language])
```

Listing 1.9: Notfall Nachricht

- Wie vorher erwähnt, kann der Administrator auch die Sprache ändern. Danach folgt die Interaktion mit dem Chatbot in der ausgewählten Sprache. Auf listing 1.10 ist die Umsetzung der Spracheinstellungen ersichtlich.

```
1 if usermng[0]=='/setLanguage':
2     global language
3     language=1
4     if usermng[1]=='EN':
5         language=1
6         bot.sendMessage(chat_id, txt[12][language])
7     elif usermng[1]=='DE':
8         language=2
9         bot.sendMessage(chat_id, txt[12][language])
10    else:
11        bot.sendMessage(chat_id, txt[17][language])
```

Listing 1.10: Spracheinstellungen

- Wenn der Administrator eine Nachricht zum Chatbot schickt, dass keine der oberen Befehle ist, dann schickt der Infotainment Bot eine Nachricht zurück, die besagt, dass die von ihm gesendete Nachricht falsch ist, und weist ihn an, zum Menü zu gehen, in dem er alle verfügbaren Optionen sehen kann. Die Menü wird dann angezeigt, wenn der Benutzer /info eingibt. Auf listing 1.11 ist die Erstellung von dem Hauptmenü für den Administrator ersichtlich.

```
1 if command=='/info':
2     if language==1:
3         bot.sendMessage(chat_id, defaultadminmsgen)
4     else:
5         bot.sendMessage(chat_id, defaultadminmsgde)
```

Listing 1.11: Hauptmenü für den Administrator

- Falls ein normaler Benutzer die oberen Befehle probiert, wird er vom Chatbot eine Nachricht erhalten, 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 einzutragen. Auf listing 1.12 ist die Überprüfung der Eingabe der Benutzer ersichtlich.

```
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         query2=("Select c_id from tb_infotainment_chatbot_users
7                 where c_id=%s") %(int(from_id))
8         count1=curs.execute(query2)
9         if count1 > 0:
10            bot.sendMessage(from_id, txt[11][language])
11        else:
12            global button
13            button="pressed"]
14            if language==1:
15                bot.sendMessage(from_id, text="Please write your phone
16                                number below...")
17            elif language==2:
18                bot.sendMessage(from_id, text="Bitte schreiben Sie
19                                Telefonnummer...")
20        else:
21            bot.sendMessage(from_id, text='Registrierung nicht moeglich')
```

Listing 1.12: 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. Dieser wartet aber auf die Genehmigung des Administrators. Auf listing 1.13 ist das Hinzufügen eines neuen Benutzers zur Datenbank.

```
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 1.13: Hinzufügen eines neuen Benutzers zur Datenbank

- Chatbot wird in zwei Sprachen angeboten, nämlich in Deutsch und in Englisch. Die Umsetzung wurde so gemacht:

Zuerst wurde eine Tabelle in der Datenbank erstellt. Diese Tabelle hat drei Spalten: id, englisch und deutsch. In den letzten Spalten wurden alle Nachrichten, die Chatbot zu den Benutzern schickt, in den jeweiligen Sprachen gespeichert. Diese Tabelle ist auf Abbildung 1.16 ersichtlich. Es wurde in dem Python-Skript, wo Chatbot programmiert wurde, eine Variable erstellt, die mit dem Wert 1 initialisiert ist. Solange diese Variable dass Wert 1 hat, werden zu den Benutzern nur Nachrichten auf Englisch geschickt, weil dieser Wert, bedeutet, dass nur die Datensätze von der ersten Spalte der Tabelle genommen werden. Die erste Spalte der Tabelle ist die Spalte Englisch. Wenn der Benutzer die Sprache auf Deutsch umstellt, dann wird die Variable mit dem Wert 2 initialisiert. Es werden dann nur Nachrichten auf Deutsch zu den Benutzern geschickt.

id	englisch	deutsch
1	Great! Your data has been sent to the administr...	Toll! Deine Daten wurden zum Administrator ges...
2	Invalid phone number format	Kein gueltiges Telefonnummerformat
3	Dieser Benutzer wurde vom Administrator blockiert	Dieser Benutzer wurde vom Administrator blockiert
4	The length of your phone number is not correct...	Die Laenge Ihrer Telefonnummer ist nicht korrek...
5	You are not the administrator	Du bist kein Administrator
6	Your images will not be displayed anymore.	Deine Bilder werden nicht mehr geschickt.
7	Image received	Bild empfangen.
8	Waiting for confirmation.	Auf Bestaetigung warten
9	Write your phone number again.	Schreiben Sie Ihre Telefonnummer erneut
10	You are not registered	Du bist nicht registriert

Abbildung 1.16: Chatbot Multilanguage

1.10.6 Konkrete Beispiele

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

- Auf Abbildung 1.17 ist das Hauptmenü für den Administrator ersichtlich. Dieses Menü enthält alle Funktionalitäten, die dem Administrator zur Verfügung gestellt sind. Der Administrator kann die Benutzer anschauen, die Benutzer akzeptieren, die Benutzer blockieren, Notfall Nachrichten schicken und die Sprache umstellen.

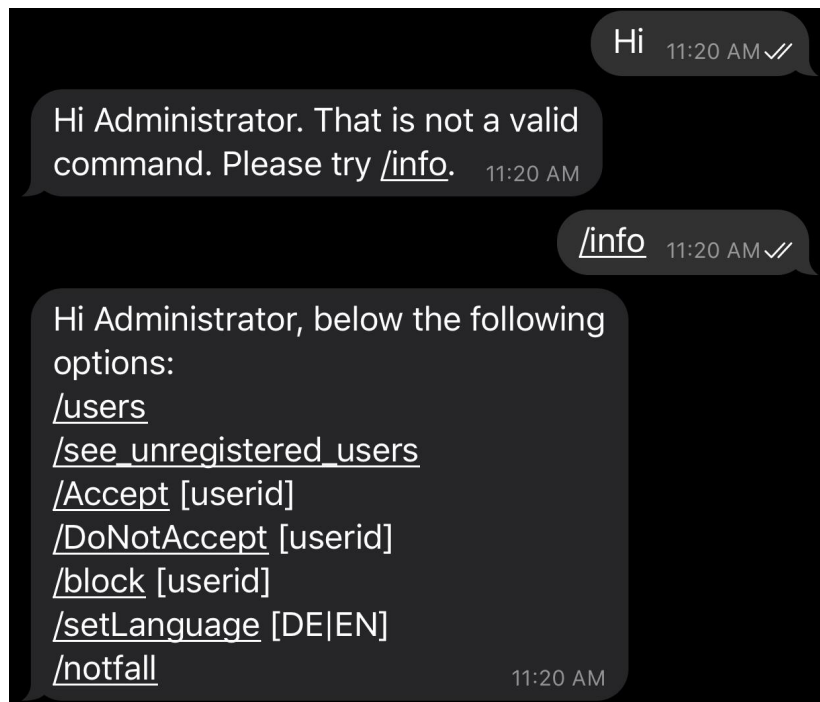


Abbildung 1.17: Hauptmenü für den Administrator

- 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. Diese Funktionalität ist auf Abbildung 1.18 ersichtlich.

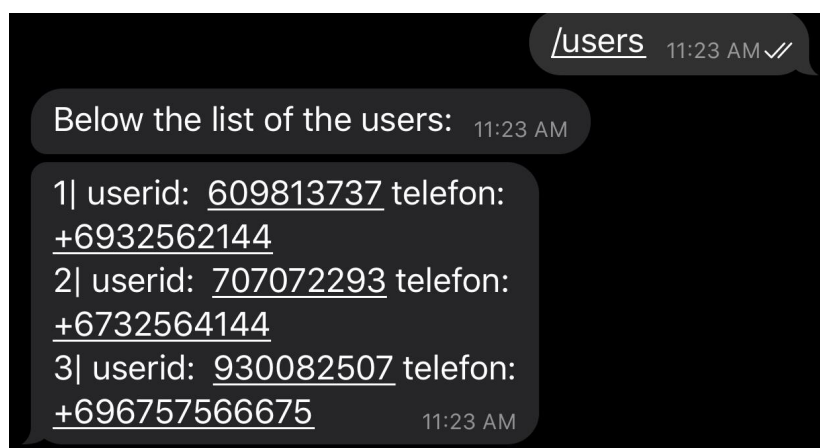


Abbildung 1.18: Auflistung der Chatbot-Benutzer

- Funktion des Administrators, die unregistrierten Benutzer anzuschauen: Chatbot schickt die eindeutige 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. Diese Funktion ist auf Abbildung 1.19 ersichtlich.

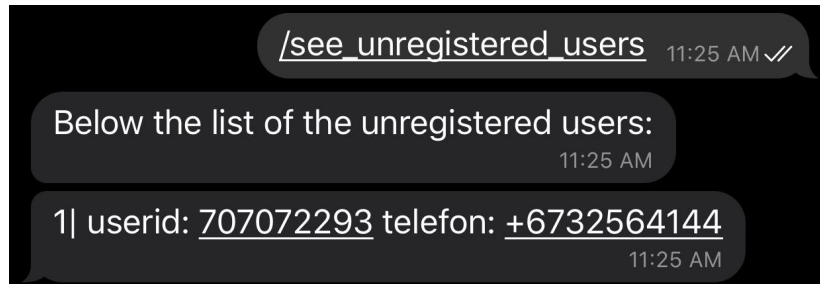


Abbildung 1.19: Auflistung der unregistrierten Benutzer

- Falls es keine unregistrierte Benutzer gibt, schickt der Infotainment Bot eine Nachricht zum Administrator, dass keine unregistrierte Benutzer im System gibt. Dieser Fall ist auf Abbildung 1.20 ersichtlich.

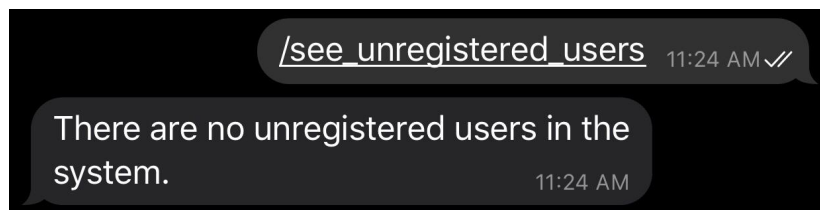


Abbildung 1.20: Keine Unregistrierte Benutzer

- Der Administrator hat die Möglichkeit, die Benutzer zu genehmigen. Er kriegt von Chatbot eine Nachricht zurück, wenn die Genehmigung erfolgreich war. Der Benutzer, der von dem Administrator akzeptiert wurde, wird von Chatbot als ein normaler Benutzer angesehen. Diese Funktion ist auf Abbildung 1.21 ersichtlich.

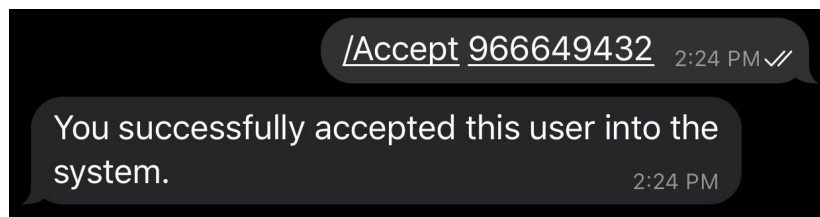


Abbildung 1.21: Genehmigung des Administrators

- Der Administrator hat die Möglichkeit, die Benutzer nicht zu akzeptieren. Er kriegt von Chatbot eine Nachricht zurück, wenn die Ablehnung erfolgreich war. Der Benutzer, der von dem Administrator nicht akzeptiert wurde, kann keine Interaktion mit dem Chatbot machen. Diese Funktion ist auf Abbildung 1.22 ersichtlich.

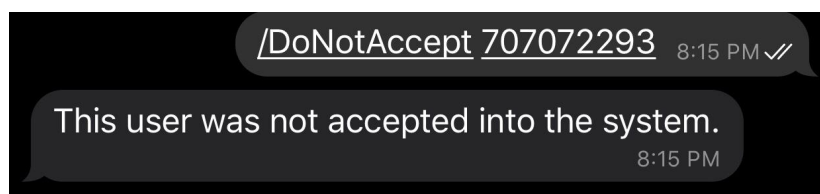


Abbildung 1.22: Ablehnung des Administrators

- Funktion des Administrators, eine Notfallwarnmeldung an alle Benutzern zu schicken. Der Administrator erhält dann eine Bestätigungsnachricht vom Chatbot. Diese Funktion ist auf Abbildung 1.23 ersichtlich.

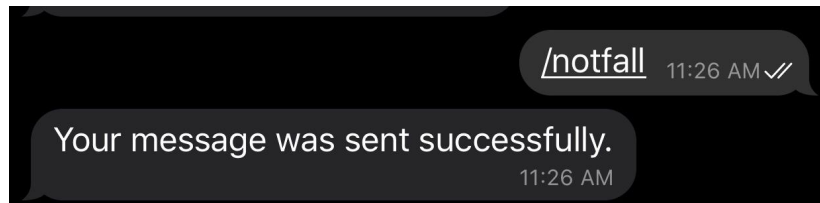


Abbildung 1.23: Notfallwarnmeldung

- Funktion des Administrators, die Benutzer zu blockieren: Der Administrator schreibt einfach das Befehl `/block` und die Chat-ID des Benutzers. Chatbot schickt eine Nachricht zurück, dass dieser Benutzer geblockt wurde. Diese Funktion ist auf Abbildung 1.24 ersichtlich.

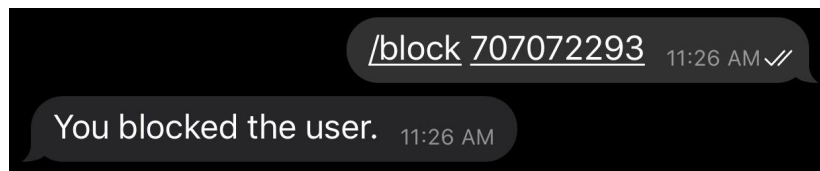


Abbildung 1.24: Benutzer Blockierung

- Funktion des Administrators, die Sprache zu ändern. In der folgenden Abbildung 1.25 wurde die Sprache von dem Administrator auf Deutsch geändert. Er kriegt eine Bestätigungsnachricht vom Chatbot, dass die Sprache auf Deutsch eingestellt wurde.

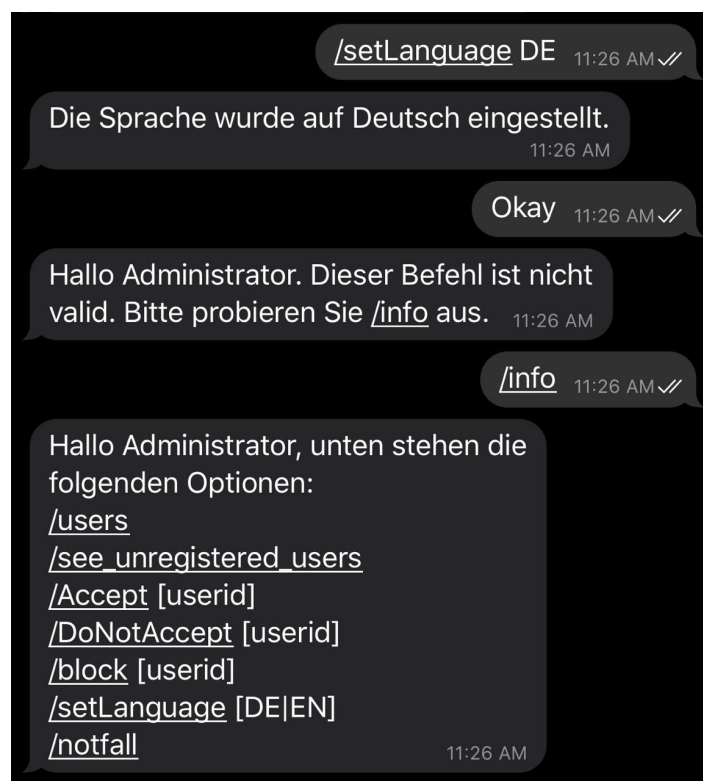


Abbildung 1.25: Spracheinstellungen

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

- Hauptmenü für den unregistrierten Benutzer

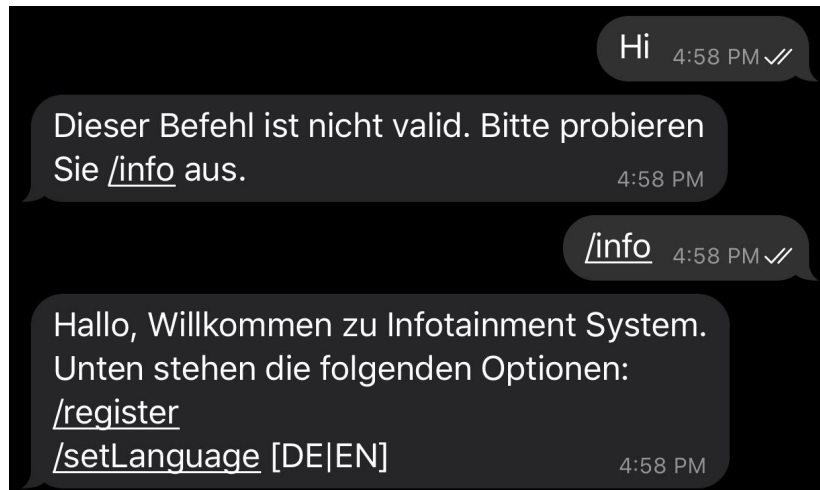


Abbildung 1.26: Hauptmenü für den unregistrierten Benutzer

- Chatbot gibt den Benutzer die Möglichkeit, sich beim Chatbot zu registrieren. Der Benutzer hat zwei Buttons zur Wahl. In diesem Fall hat der Benutzer auf "Ja" gedrückt. Der Benutzer soll die Telefonnummer eingeben. Der Chatbot schickt eine Nachricht, wo es erklärt wird, in welcher Format die Telefonnummer geschrieben werden soll.

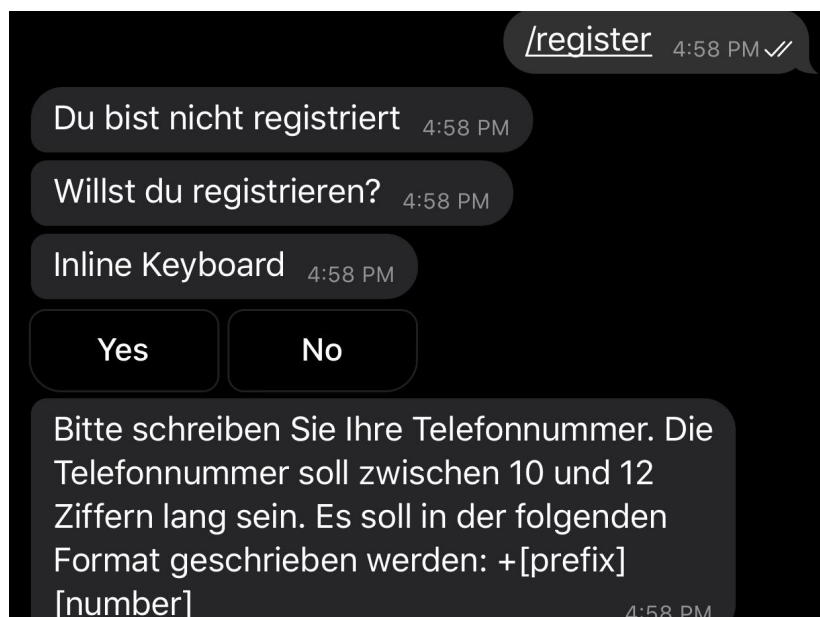


Abbildung 1.27: Registrierung beim Chatbot

- Wenn der Benutzer, statt die Telefonnummer Text eingibt, bekommt er eine Fehlermeldung von Chatbot.

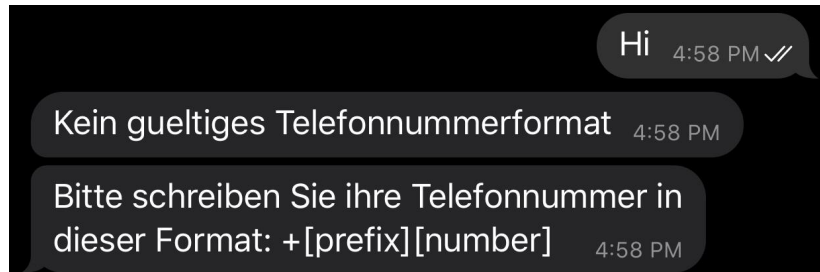


Abbildung 1.28: Falsche Eingabe des Telefonnummers

- Wenn die Länge der Telefonnummer nicht korrekt ist, bekommt der Benutzer eine Fehlermeldung von Chatbot.

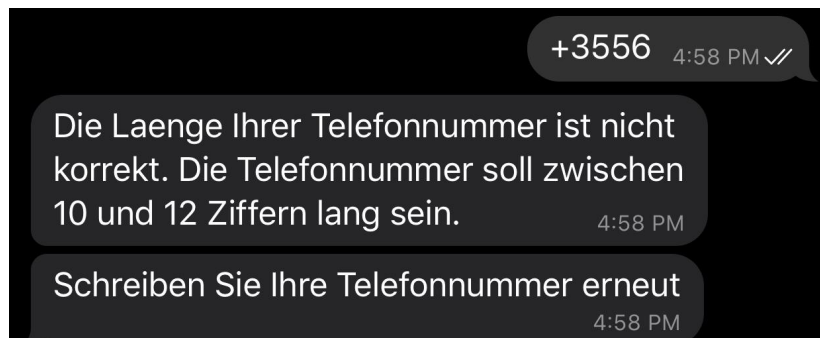


Abbildung 1.29: Telefonnummer zu kurz

- Wenn die Telefonnummer von dem Benutzer korrekt eingegeben wird, bekommt er eine Meldung von Chatbot und er kann keine Interaktion mit dem Chatbot haben, bis er die Genehmigung von Administrator kriegt.

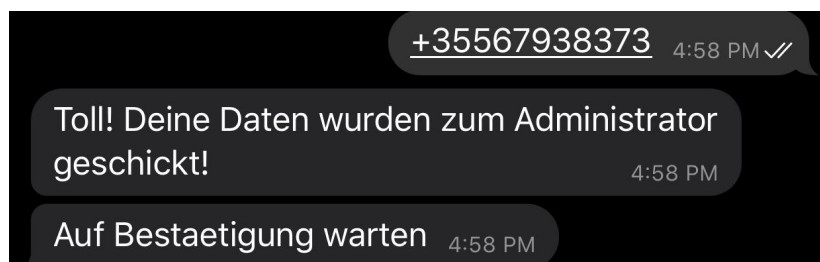


Abbildung 1.30: Korrekte Eingabe des Telefonnummers

- 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 anderen Nachrichten, die später kommen, werden nicht berücksichtigt bis der Administrator die Genehmigung gegeben hat.

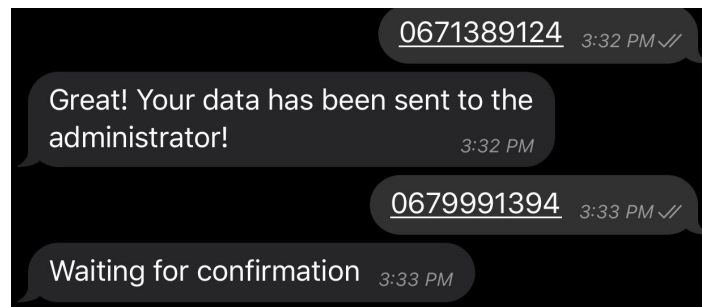


Abbildung 1.31: Das Warten der Benutzer auf die Bestätigung des Administrators

- Falls der Benutzer auf "Nein" drückt, schickt der Chatbot ihm eine Meldung zurück, dass die Interaktion mit dem Chatbot ohne Registrierung nicht möglich ist.

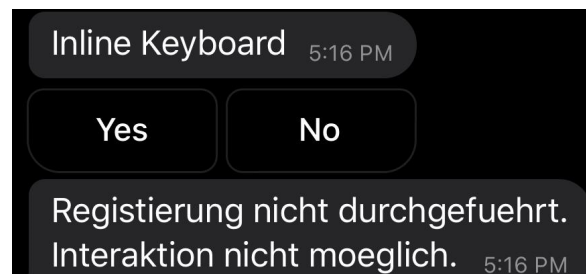


Abbildung 1.32: Keine Registrierung

- Funktion von einem unregistrierten Benutzer, die Sprache zu ändern.

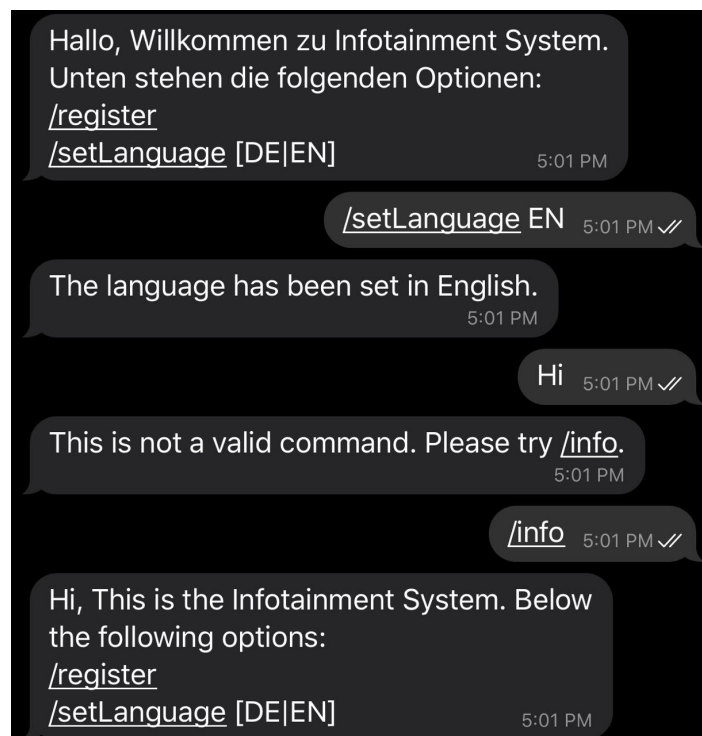


Abbildung 1.33: Spracheinstellungen für unregistrierte Benutzer

In den folgenden Abbildungen wird die Interaktion eines normalen Benutzers mit dem Chatbot dargestellt.

- Hauptmenü für einen normalen Benutzer

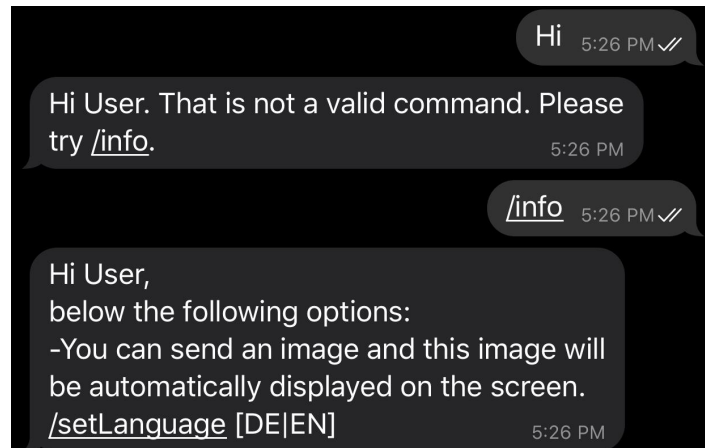


Abbildung 1.34: Hauptmenü für einen normalen Benutzer

- Bilderfunktionalität

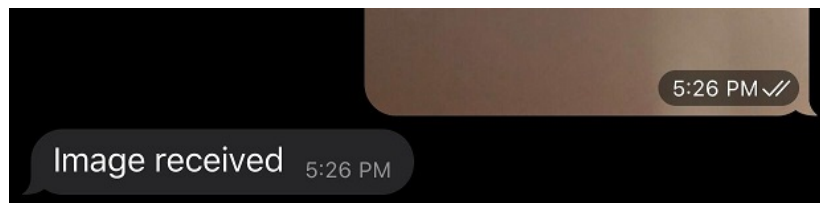


Abbildung 1.35: Bilderfunktionalität

- Spracheinstellungen für einen normalen Benutzer

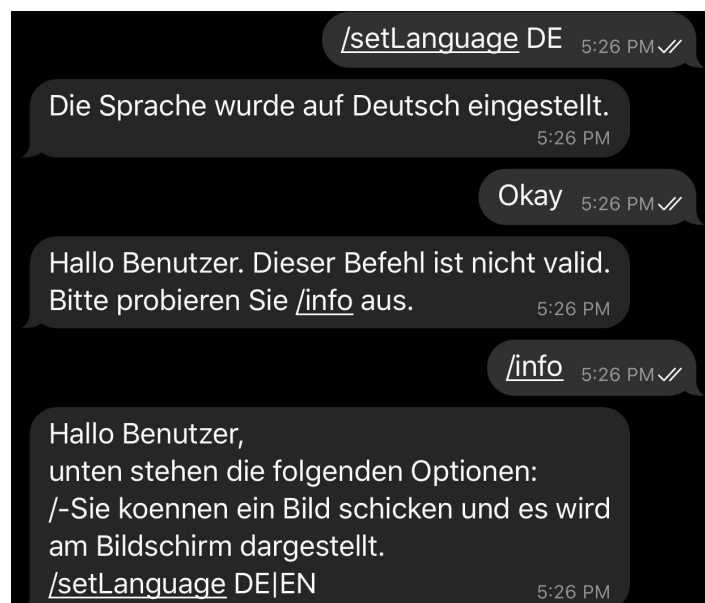


Abbildung 1.36: Spracheinstellungen für einen normalen Benutzer

- Wenn der Administrator eine Notfallwarnmeldung zu den Benutzern schickt, kriegen die Benutzer die folgende Nachricht von Chatbot.



Abbildung 1.37: Notfallwarnmeldung vom Administrator

In der folgenden Abbildung 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 ihm eine Meldung schicken, dass die von diesem Benutzer erhaltene Nachrichten nicht berücksichtigt werden, weil er blockiert ist.

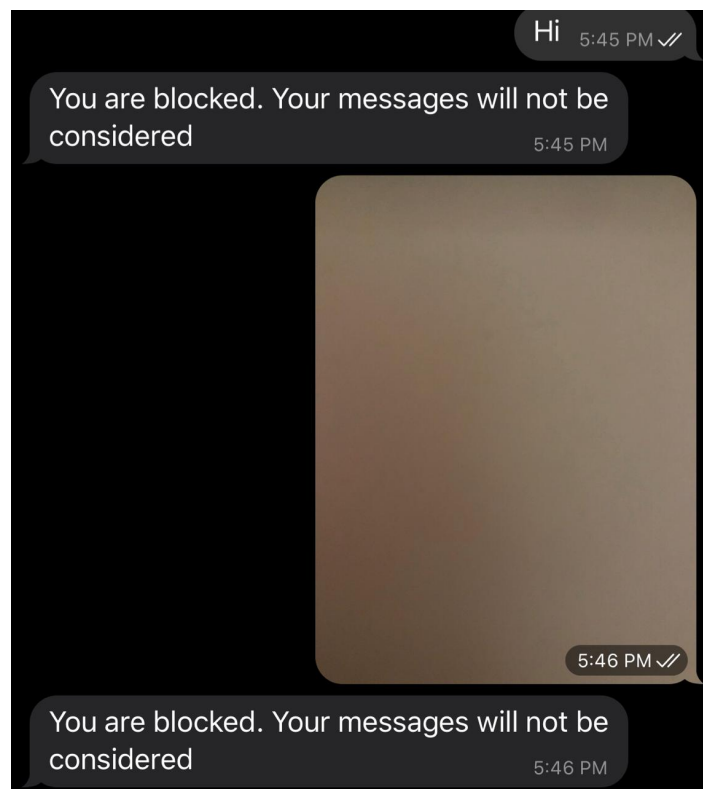


Abbildung 1.38: Nachricht von einem blockierten Benutzer

In der folgenden Abbildung wird die Interaktion eines nicht akzeptierten Benutzer mit dem Chatbot dargestellt.

- Wenn ein Benutzer von dem Administrator nicht akzeptiert ist und trotzdem versucht eine Nachricht zum Chatbot zu schicken, wird der Chatbot ihm eine Meldung schicken, dass die von diesem Benutzer erhaltene Nachrichten nicht berücksichtigt werden, weil er noch nicht akzeptiert ist.

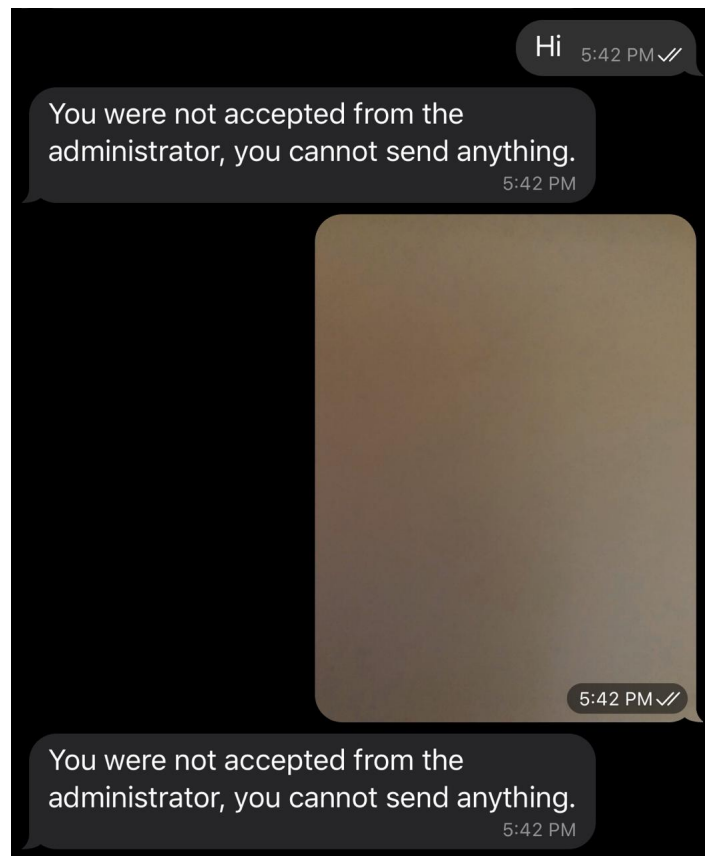


Abbildung 1.39: Nicht akzeptierter Benutzer

In der folgenden Abbildung wird die Interaktion eines akzeptierten Benutzer mit dem Chatbot dargestellt.

- Wenn ein Benutzer von dem Administrator akzeptiert wird, kriegt er den Status einer normalen Benutzer.

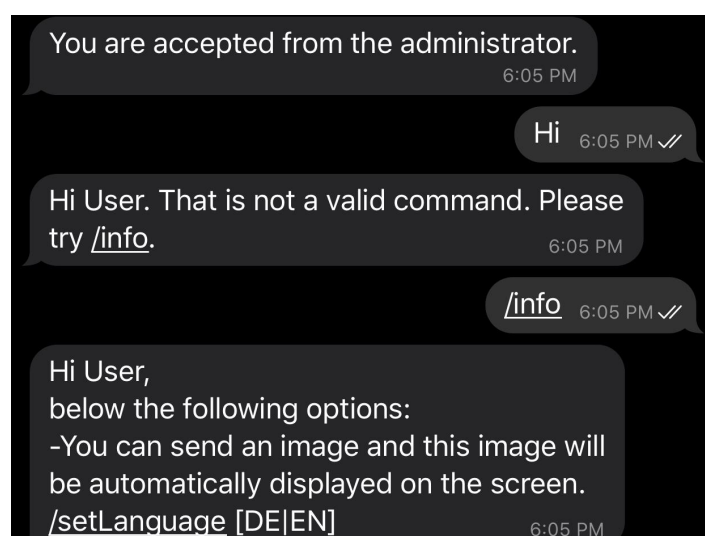


Abbildung 1.40: Akzeptierter Benutzer

1.11 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 werden, wie diese genau funktionieren. Es sollte genau analysiert werden, auf welchem Format die API die Daten zurückliefert, damit diese Daten an dem Bildschirm angezeigt werden konnten. Nach vielen Versuchen wurden die richtigen APIs gefunden.

Anzeige

Die Design Vorbereitung hat viele Probleme gebracht. Die Darstellung von den Daten auf die Anzeige war nicht einfach. Die Darstellung wurde mit der Programmiersprache PHP und HTML durchgeführt, wobei die Kenntnisse beider Sprachen kombiniert werden mussten. Das Anpassen der Daten auf dem Bildschirm war ein weiteres Problem, da einige Tabellen zu viele Daten enthielten und einige weniger. Diese mussten aber am Bildschirm passen. Eine große Herausforderung war die Darstellung von dem Stundenplan. Wenn der Unterricht in zwei Gruppen geteilt war, sollte die Klasse in zwei Zellen geteilt sein, wobei für jede Gruppe die Einträge(Fach, Lehrer, Raum) dargestellt werden sollten. Das war eine schwierige und aufwendige Aufgabe. Es wurden Beispiele im Internet angeschaut und analysiert, um diese Aufgabe zu lösen.

Offline Betrieb

Die Implementierung von Offline Betrieb war die größte Herausforderung. Zuerst sollte das Prinzip und der Weg der Programmierung genau überlegt werden. Am Ende hat es funktioniert, aber das einzige Problem war bei der Tabelle Chatbot Bilder. Dort wurden die Bilder mit einem bestimmten Format, nämlich mit blob Datentyp gespeichert, deswegen gab es Probleme bei der Übertragung der Daten von dieser Tabelle vom Server zum Client. Deswegen wurde entschieden, die Tabelle Chatbot Bilder nicht vom Server zum Client zu übertragen, da es Probleme gab. Die Daten anderer Tabellen konnten alle übertragen werden.

Chatbot

Chatbot war auch ein neues Konzept. Die Herausforderungen im Bezug auf Chatbot waren nur am Beginn. Es sollte die richtige API gewählt werden und auch die Einrichtung musste passen. Der Chatbot sollte kontinuierlich getestet werden, weil es könnte sein, dass der Chatbot die falschen Nachrichten zu den Benutzern schickt und die Funktionalitäten stimmten dann nicht ganz. Für die Lösung der Probleme beim Chatbot habe ich viele Tutorials geschaut und viele Beispiele im Internet nachgeschaut.

Datenbank

Die Datenbank war problematisch, weil es sollte genau überlegt werden, welche Tabellen notwendig sind und wie sie verwendet werden.

Die richtige Erstellung der Prozeduren war auch ein Thema. Es sollte eine Prozedur erstellt werden, die dem Administrator der Webseite mit einer Drop Down Liste alle Lehrer anzeigt, die nicht fehlen, die bereit zum supplieren sind und kein Unterricht haben. Diese Lehrer wurden dann als Supplierer eingetragen. Aber die richtige Kombination der Daten und die Erstellung dieser Prozedur war sehr komplex und hat viel Zeit im Anspruch genommen, bis eine Lösung gefunden wurde.

1.12 Qualitätssicherung, Controlling

Qualität wurde sicher gestellt, damit die Hauptanforderungen des Auftraggebers vollständig erfüllt werden. 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, die Darstellung von Wetterdaten auf dem Bildschirm oder die Implementierung von Offline Betrieb. Chatbot ist eine wichtige Komponente für die Qualitätssicherung, weil es viele Funktionen anbietet. Die wichtigste Funktionalität des Chatbots ist die Interaktion mit dem Benutzer. Da es sonst keine Interaktion der Benutzer mit dem Bildschirm, wo die Informationen dargestellt werden, gibt, wurde mittels Chatbot diese Funktion angeboten. Um sicherzustellen, dass es keine Fehler auftreten wurden alle Komponenten sehr oft getestet.

1.13 Ergebnisse - Irena Bala

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst.

1.13.1 Implementierung

Die wichtigsten Ergebnisse der Arbeit sind:

- Die Implementierung von einer Hauptkomponente der künstlichen Intelligenz wie Chatbot, die das entwickelte System interessanter macht. Diese Komponente wird mehrsprachig angeboten.
- Die Anzeige der leeren Klassen, Stundenplan, Kalenderinformationen und Wetterdaten.
- Anlegen einer Datenbank, die als Basis für die Speicherung aller Informationen dient.
- Verwendung von APIs, die die Möglichkeit anbieten, Zugriff auf verschiedene Daten zu haben und als Schnittstelle dienen für die Implementierung der Komponenten.

- Einrichtung und Anlegung eines Servers, dass auch der Hauptteil des Systems ist und als Basis für die Zusammensetzung aller Komponenten dient.
- Integration des Systems, damit alle Komponenten miteinander verbunden werden können.
- Implementierung von Offline Betrieb.
- SSL-Verschlüsselung für eine sichere Datenübertragung.
- Einrichtung des Systems. Dieser Punkt erfolgt ganz am Ende. Es geht hier um die LAN¹⁹ Verbindung zum System. Es wird dann die Verbindung zur Datenbank überprüft.

1.14 Handbuch für die Bedienung

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

1.14.1 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 die zuerst in die Datenbank gespeichert und automatisch auf dem Bildschirm angezeigt werden.

Zuerst soll am Handy die Applikation Telegram 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 sich zuerst registrieren.

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

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

Danach wird gewartet, bis der Administrator die Genehmigung für die neue Registrierung gegeben hat. Sobald der Administrator diese Genehmigung gegeben hat, wird dieser Benutzer benachrichtigt und nur dann kann er Bilder zum Chatbot schicken, die an dem Bildschirm angezeigt werden. Die Benutzer haben die Möglichkeit, selbst die Sprache auszuwählen, in der die Interaktion mit dem Chatbot erfolgen soll.

1.14.2 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.

¹⁹Local Area Network

- 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, sollte die folgende Eingabe zum Chatbot geschickt werden: /Accept und die Chat ID von dem Benutzer
- Wenn der Administrator eine Registrierung nicht genehmigen will, soll die folgende Eingabe schicken: /DoNotAccept und die Chat ID von dem Benutzer
- Wenn der Administrator eine Notfallwarnmeldung zu den Benutzer schicken will, soll er die folgende Eingabe schicken: /notfall
- Wenn der Administrator die Sprache ändern will, soll er die folgende Eingabe schicken: /setLanguage[DE/EN]
- Wenn der Administrator das Hauptmenü sehen will, soll er /info eingeben.

1.15 Evaluierung und Resümee

1.15.1 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:

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

Die Anforderungen bei der Anzeige sind auch erfüllt worden. Dort können verschiedene Informationen angezeigt werden. Für die Anzeige gibt es auch verschiedene Layouts. Die Datenbank wurde anders als geplant, aufgebaut. Während der Arbeit wurden viele Adaptierungen gemacht. Es sind mehrer Tabellen als geplant erstellt worden. Eine Tabelle davon ist die Tabelle für die Chatbot Benutzer. Diese war früher nicht geplant, weil die Anlegung von mehreren User-Rechte beim Chatbot nicht geplant war. Außerdem wurden auch Views und gespeicherte Prozeduren erstellt, z.B. für den Stundenplan

und die leeren Klassen, was nicht im Voraus geplant war. Beim Offline Betrieb, wurde gedacht, dass es eine automatische Methode für die Datensynchronisation zwischen Server und Client gibt, aber dass passte so nicht und alles wurde manuell implementiert und programmiert. Bei der Implementierung von Offline Betrieb lag auch die große Herausforderung, weil die Programmierung dieser Funktionalität sehr aufwendig war. Das bedeutet, dass alle Komponenten des Systems verknüpft sind und wenn eine davon geändert wurde, hatte diese Konsequenzen auf andere Komponenten. Es gab viele Änderungen auf diese Art und Weise, um die Aufgaben zu lösen. Aber am Ende sind die Ziele erfolgreich abgeschlossen worden.

1.15.2 Wertschöpfung und Lessons Learned

Es wurde das Wissen in den folgenden Bereichen erweitert:

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

HTML, CSS, PHP, Javascript wurden für die Anzeige sehr oft benötigt. Sie sind wichtig für die Darstellung der Daten. PHP bietet eine Schnittstelle zur Datenbank und kann zusammen mit HTML verwendet werden. Deswegen war diese Programmiersprache sehr hilfreich. Die Konfiguration vom Raspberry PI Server hat das Wissen in allgemeine Konfiguration von Systemen erweitert. Programmierung in Python ist am meisten vorgekommen, weil der ganze Chatbot wurde mit Python programmiert. Es wurden für die Darstellung der Wetterdaten APIs verwendet, daher wurde das Wissen auch in diesem Bereich erweitert. Diese Diplomarbeit hat nicht nur geholfen, die technischen Fähigkeiten bzw. Kompetenzen zu entwickeln, sondern auch in Bezug auf organisatorischen Teil, Teamwork, Kommunikation und Engagement. Es gab während der Umsetzung häufig Probleme, aber durch Ausdauer, Entschlossenheit und Zusammenarbeit wurden die beabsichtigten Ziele erreicht.

Abbildungsverzeichnis

1.1	Structured Software Design vom Server	9
1.2	ERD	11
1.3	SSL Zertifikat	12
1.4	Layout 2	13
1.5	Layout 4	13
1.6	Kalenderinformationen	14
1.7	Leere Klassen für Montag, die zweite Stunde	14
1.8	Stundenplan	15
1.9	Display mit fehlenden Tabellen	17
1.10	Ausführung Offline Betrieb Skript	17
1.11	Erstellung der Tabellen im Client	18
1.12	Weitere Erstellung der Tabellen im Client	18
1.13	Vollständige Erstellung der Tabellen im Client	19
1.14	Hinzufügen von Datensätzen zu den Tabellen	19
1.15	Wetterdaten	20
1.16	Chatbot Multilanguage	27
1.17	Hauptmenü für den Administrator	28
1.18	Auflistung der Chatbot-Benutzer	28
1.19	Auflistung der unregistrierten Benutzer	29
1.20	Keine Unregistrierte Benutzer	29
1.21	Genehmigung des Administrators	29
1.22	Ablehnung des Administrators	29
1.23	Notfallwarnmeldung	30
1.24	Benutzer Blockierung	30
1.25	Spracheinstellungen	30
1.26	Hauptmenü für den unregistrierten Benutzer	31
1.27	Registrierung beim Chatbot	31
1.28	Falsche Eingabe des Telefonnummers	32
1.29	Telefonnummer zu kurz	32
1.30	Korrekte Eingabe des Telefonnummers	32
1.31	Das Warten der Benutzer auf die Bestätigung des Administrators	33
1.32	Keine Registrierung	33
1.33	Spracheinstellungen für unregistrierte Benutzer	33
1.34	Hauptmenü für einen normalen Benutzer	34
1.35	Bilderfunktionalität	34
1.36	Spracheinstellungen für einen normalen Benutzer	34

1.37	Notfallwarmeldung vom Administrator	35
1.38	Nachricht von einem blockierten Benutzer	35
1.39	Nicht akzeptierter Benutzer	36
1.40	Akzeptierter Benutzer	36

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).