

# Diplomarbeit

Name



AUTOR: IRENA BALA

# Inhaltsverzeichnis

<b>1</b>	<b>Umsetzung - Irena Bala</b>	<b>1</b>
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	Offline Betrieb . . . . .	11
1.8	Wetterdaten . . . . .	13
1.9	Chatbot . . . . .	13
1.9.1	Chatbot Einrichtung: . . . . .	13
1.9.2	Einrichtung des Chatbots in RaspberryPI . . . . .	14
1.9.3	Grundlagen für die Umsetzung von Chatbot . . . . .	14
1.9.4	Funktionalitäten von Chatbot . . . . .	14
1.9.5	Umsetzung der Funktionalitäten . . . . .	16
1.9.6	Konkrete Beispiele mit Bildern . . . . .	21
1.10	Probleme, Herausforderungen und deren Lösung . . . . .	36
1.11	Qualitätssicherung, Controlling . . . . .	37
1.12	Ergebnisse - Irena Bala . . . . .	37
1.12.1	Implementierung . . . . .	37
1.13	Handbuch für die Bedienung . . . . .	38
1.13.1	Beschreibung der Bedienung als User . . . . .	38
1.13.2	Beschreibung der Bedienung als Administrator . . . . .	38
1.14	Evaluierung und Resümee . . . . .	39
1.14.1	Planung vs Realisierung . . . . .	39
1.14.2	Wertschöpfung und Lessons Learned . . . . .	39

# 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 dieses Komponentes das Interesse des Menschen an dem System erhöhen wird.

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

#### Einführung in Chatbot

Chatbot ist eine sehr wichtige Komponente 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. Jedes Bot, das erstellt wird, bekommt ein Token, das eindeutig für das 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 2.1 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. das Bildschirm, in dem die von dem Server erhaltene Daten darstellt.

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 HTTP<sup>1</sup> Server ist ein weltweit verbreiteter Webserver. Dieser Server ist Open Source, das bedeutet, dass dafür keine Lizenz gekauft werden muss, 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 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

---

<sup>1</sup>Hypertext Transfer Protocol

SSL <sup>2</sup>Zertifikats. Das ermöglicht die Übertragung der Daten in einer verschlüsselten 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 <sup>3</sup> 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 auf einer bestimmten Format. 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 SQL<sup>4</sup>. 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?

PHP<sup>5</sup> ist eine serverseitige Programmiersprache. Das bedeutet, dass diese Sprache zur Programmierung von der vom Server auszuführenden Ereignissen verwendet wird. PHP

---

<sup>2</sup>Secure Socket Layer

<sup>3</sup>Hypertext Transfer Protocol Secure

<sup>4</sup>Structured Query Language

<sup>5</sup>Parallel 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 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, 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?

HTML<sup>6</sup> 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?

CSS<sup>7</sup> 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]

---

<sup>6</sup>HyperText Markup Language

<sup>7</sup>Cascading Style Sheets



## Was ist Raspberry PI?

Der Raspberry PI ist ein Minirechner, der zur Linux Programmierung, Shell Scripting und Realisierung von technischen Projekten verwendet wird. Es braucht eine Tastatur, eine Maus, einen Netzteil, VGA<sup>8</sup> und HDMI<sup>9</sup>-VGA Konverter, damit es benutzt werden kann. Die Konfiguration von einem Raspberry PI erfolgt durch eine SD-Karte. Diese SD<sup>10</sup>-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 TLS<sup>11</sup> 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 CA<sup>12</sup> 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 API<sup>13</sup> ist eine Schnittstelle, die die Implementierung von Chatbot ermöglicht. Es bietet verschiedene Funktionen an, nämlich die Einrichtung, Erstellung und die Verarbeitung von Bots. Die genannten Funktionen sind in der Dokumentation vom 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]

---

<sup>8</sup>Video Graphics Array

<sup>9</sup>High-Definition Multimedia Interface

<sup>10</sup>Secure Digital Memory Card

<sup>11</sup>Transport Layer Security

<sup>12</sup>Certificate Authority

<sup>13</sup>Application Programming Interface

## Was ist JSON?

„JSON<sup>14</sup> 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 4.1 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 Komponente des Systems, also Teil des Servers. Der Pfeil bei der unteren Abbildung, 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 eingeschaltet sind, schicken sie eine Statusmeldung zum Server, wie in der Abbildung 1.1 ersichtlich ist.

Nachdem der Chatbot eingeschaltet 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 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.

<sup>14</sup>JavaScript Object Notation

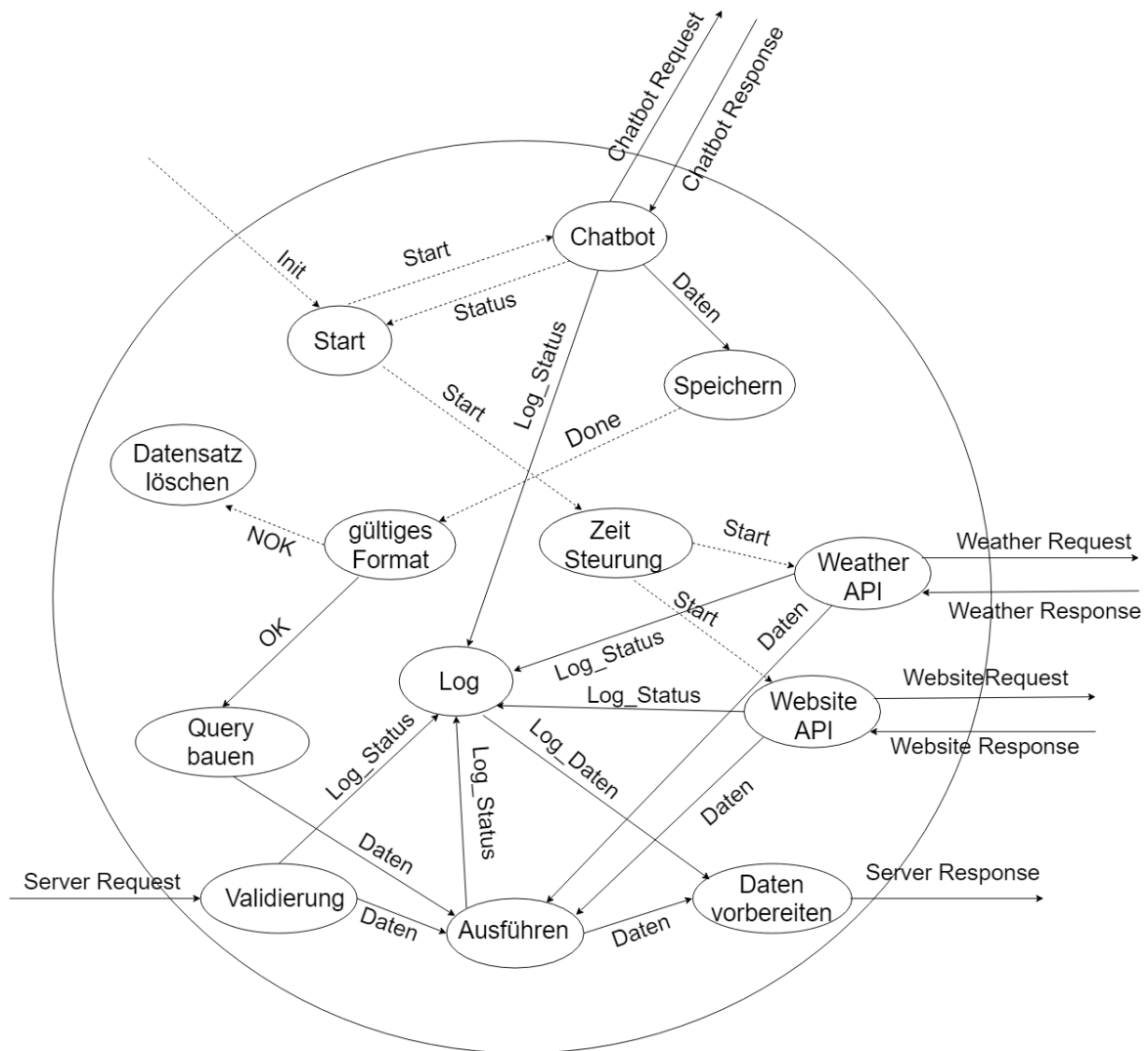


Abbildung 1.1: Structured Software Design vom Server

## 1.3 Konfiguration von Raspberry PI Server

Folgende Schritte sind für die Konfiguration vom RaspberryPI notwendig:

- Zuerst wurde ein Image 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 SSH<sup>15</sup> aktiviert, damit eine sichere Verbindung zu diesem Server von einem externen Gerät ermöglicht werden kann.

## 1.4 Datenbank

- Zuerst wurde ein ER<sup>16</sup> 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
- 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

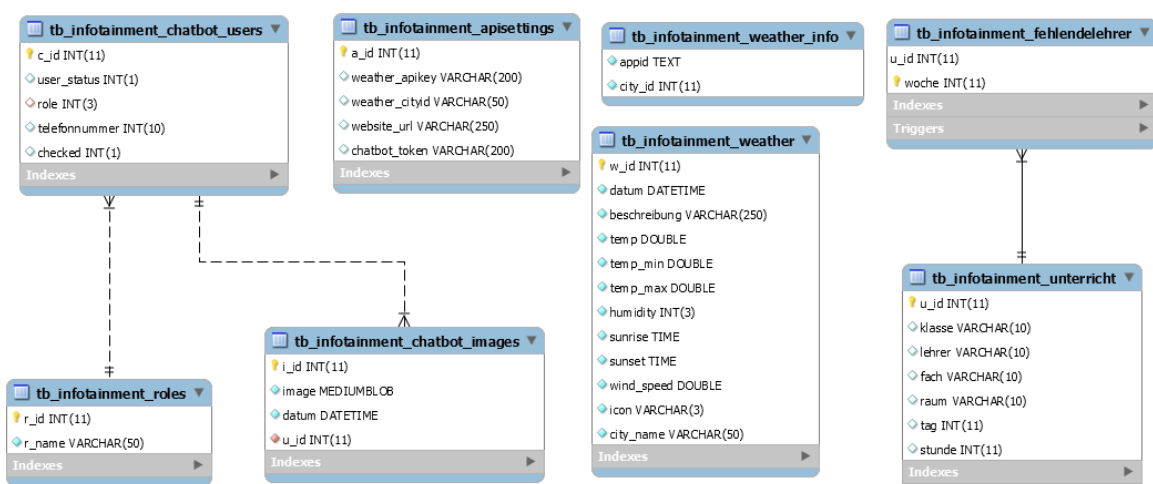


Abbildung 1.2: ERD

<sup>15</sup>Secure Shell

<sup>16</sup>Entity 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, Supplierplan und die leere Klassen 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 selbst vom Administrator geschriebene Informationen. Diese Informationen sollten so angepasst werden, dass sie in unterschiedlichen Teilbereichen des Layouts hinzugefügt werden können und auch von den anderen lesbar sind.

Die Tabellen Supplierplan und Stundenplan wurden aus der Datenbank selektiert und dann mit PHP dargestellt. Die Formatierung wurde ebenfalls mit JavaScript gemacht. Bei Tabellen mit vielen Datensätzen wurde die Tabelle in zwei Seiten aufgeteilt, damit es richtig im Layout der Anzeige angepasst werden kann. Auf die Abbildung 4.3 ist die Darstellung der Wetterdaten auf dem Bildschirm ersichtlich.

## 1.7 Offline Betrieb

Der Offline-Betrieb ist aus mehreren Gründen für das Infotainment-System sehr wichtig. Durch den Offline-Betrieb können die Displays, die Clients sind, bei Stromausfall nicht vollständig ausgeschaltet werden, 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 allen Tabellennamen 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 SShow Columns aufgelistet.

```
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.

```
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 von dem 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 Wetterdaten

- Als erster Schritt erfolgte die Registrierung bei [openweathermap.org](https://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.

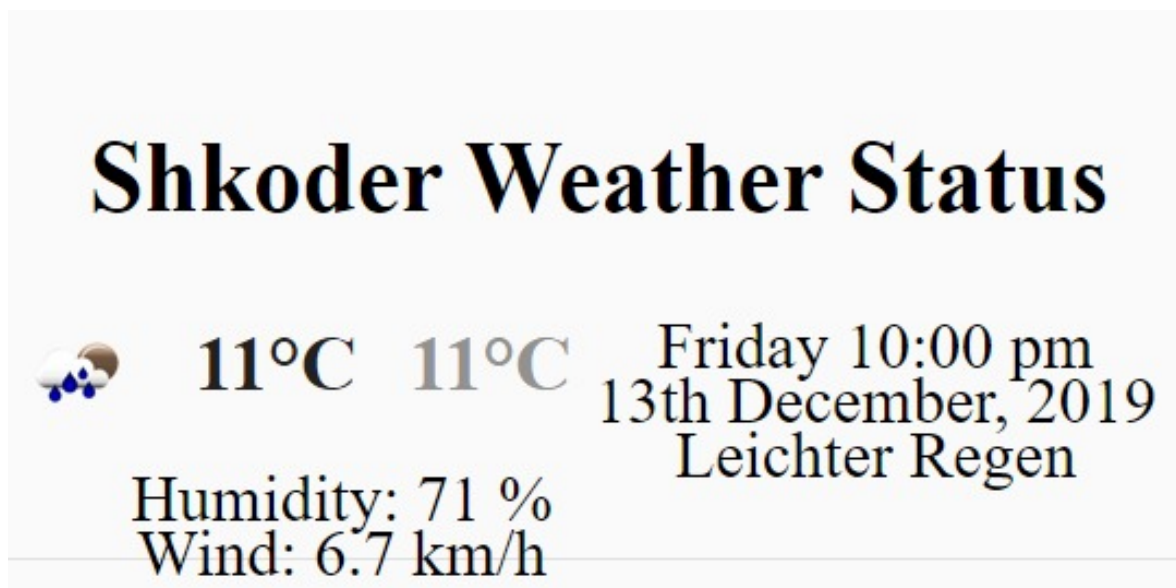


Abbildung 1.3: Wetterdaten

## 1.9 Chatbot

In diesem Unterkapitel wird der Chatbot und dessen Umsetzung erklärt.

### 1.9.1 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 nach, 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 von diesen Schritten, ist der Bot fertig erstellt geworden. Es könnte dann mit dem angegebenen Benutzernamen durchsucht werden und der entsprechende Chat damit geöffnet werden.

### 1.9.2 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]

### 1.9.3 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.

### 1.9.4 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, dass 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.

### **1.9.5 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.

```
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 was für 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 diese 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 momentan im System keine Benutzer sind.

```
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 sind, wird dem Administrator eine Nachricht geschickt: "Es gibt keine unregistrierte Benutzer".

```
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 will. 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 einfach nicht genehmigt wurde.

```
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.

```
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 denen eine Nachricht geschickt, dass sie blockiert sind und die Interaktion nicht möglich ist.

```
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.

```
1 if command=='/notfall':
2     querynotfall=("select c_id, user_status from tb_infotainment_chatbot_
3     users where role!='777'")
4     curs.execute(querynotfall)
5     variablenotfall=curs.fetchall()
6     for not in variablenotfall:
7         idd=not[0]
8         bot.sendMessage(str(idd),txt[26][language])
9     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.

```
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.

```
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 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         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.

```
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 in der Abbildung 1.4 ersichtlich. Es wurde in dem Python-Skript, wo Chatbot programmiert wurde, eine Variable erstellt, die mit dem Wert 1 initialisiert ist. Solange diese Variable das Wert 1 hat, werden zu den Benutzern nur Nachrichten auf Englisch geschickt, weil das Wert 1, bedeutet, dass es 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 Telefonnummernformat
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 1.4: Chatbot Multilanguage

### 1.9.6 Konkrete Beispiele mit Bildern

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

- Hauptmenü für den Administrator

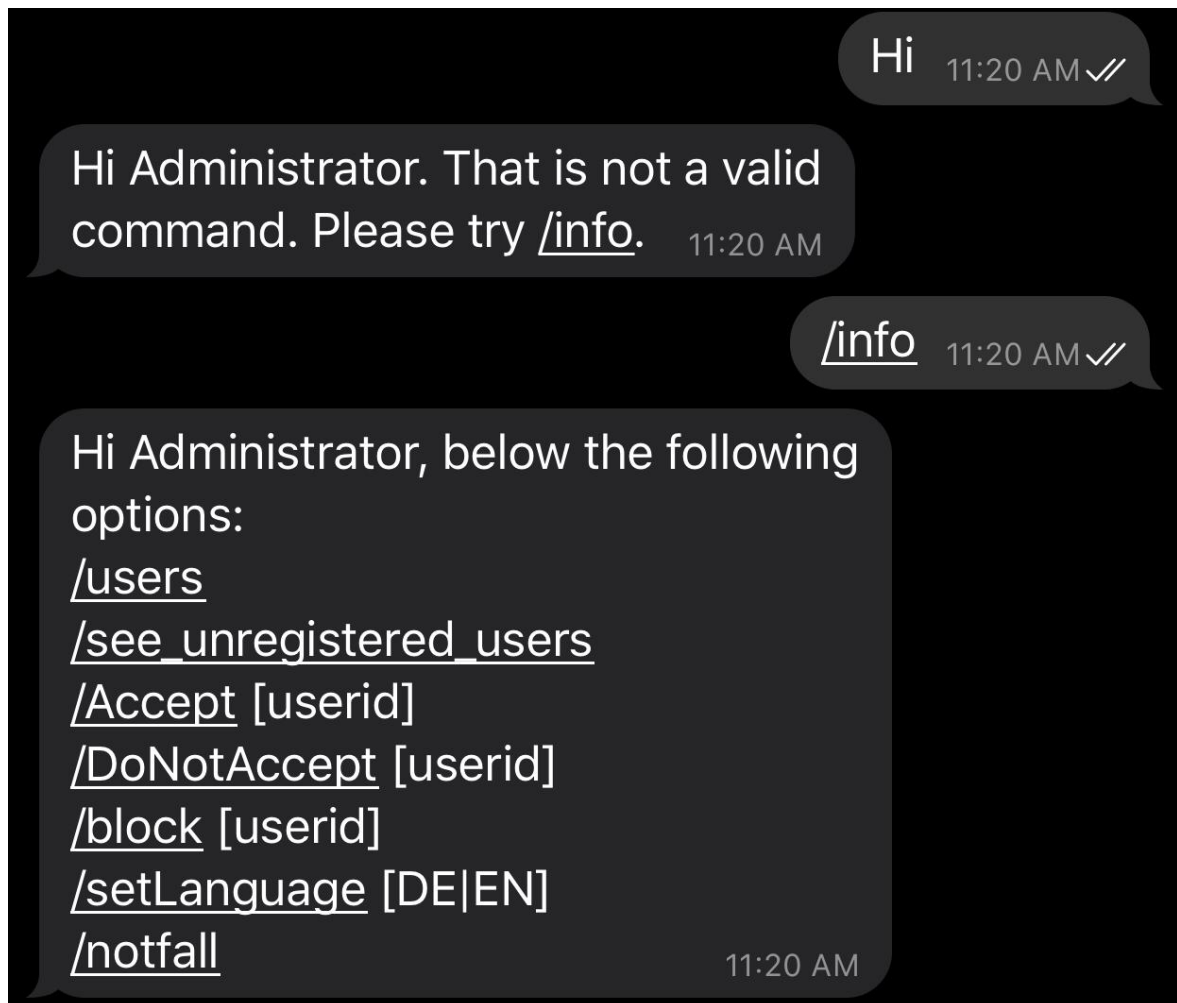


Abbildung 1.5: Hauptmenü für den Administrator

- 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.



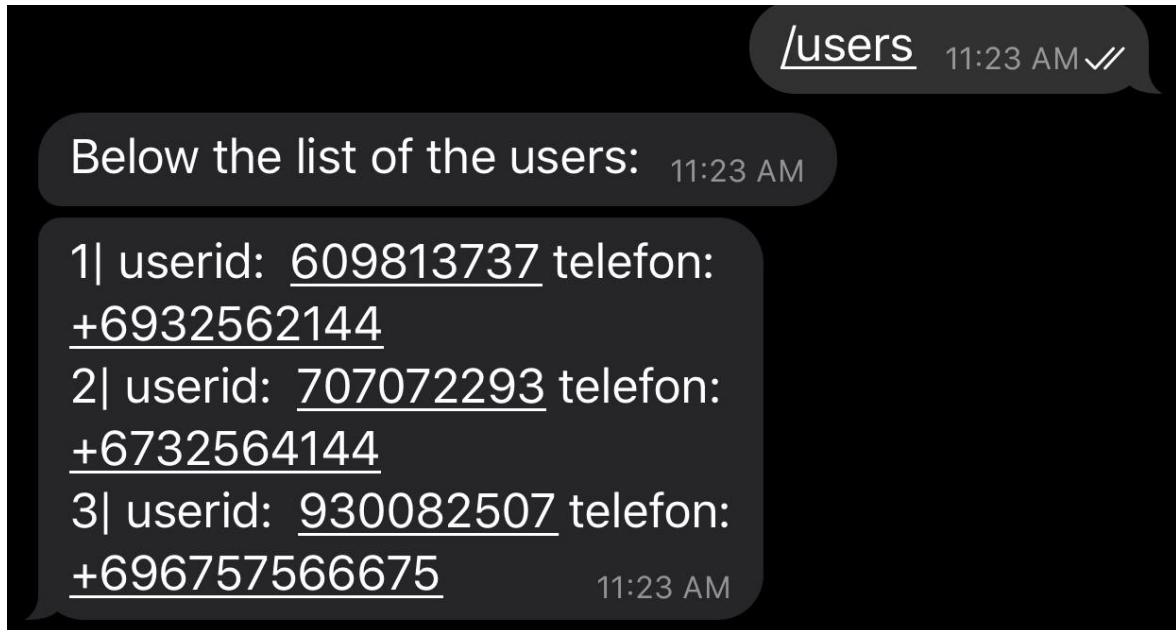


Abbildung 1.6: Auflistung der Chatbot-Benutzer

- 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.

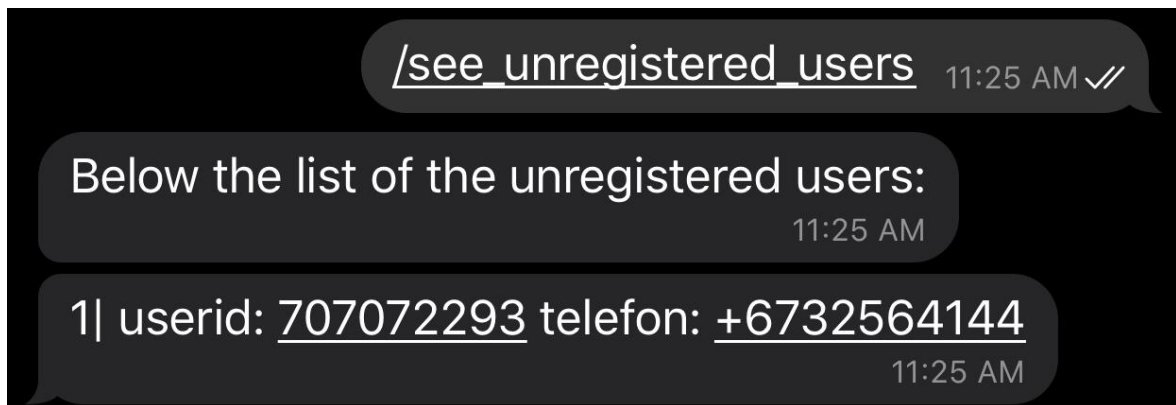


Abbildung 1.7: Auflistung der unregistrierten Benutzer

- Falls es keine unregistrierte Benutzer gibt, schickt der Infotainment Bot eine Nachricht zum Administrator, dass es bereits keine unregistrierte Benutzer im System sind.

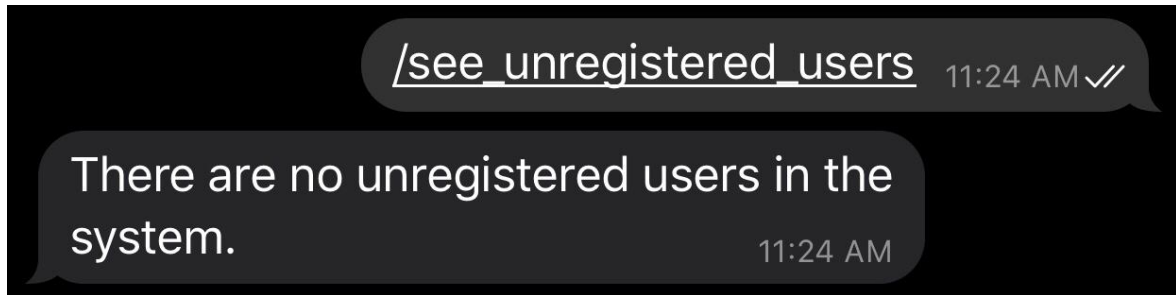


Abbildung 1.8: Keine Unregistrierte Benutzer

- 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.

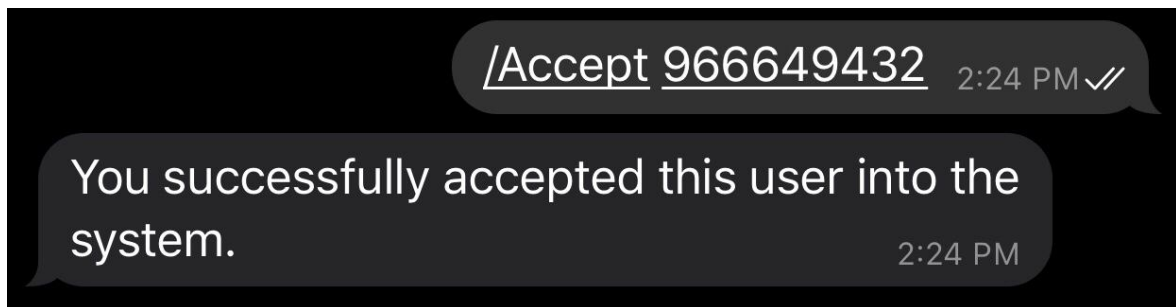


Abbildung 1.9: Genehmigung des Administrators

- 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.

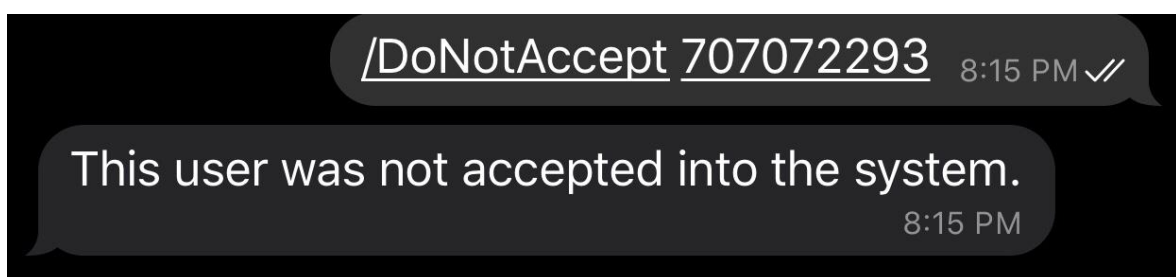


Abbildung 1.10: Ablehnung des Administrators

- Funktion des Administrators, eine Notfallwarnmeldung an alle Benutzern zu schicken. Der Administrator erhält dann eine Bestätigungsnachricht vom Chatbot.

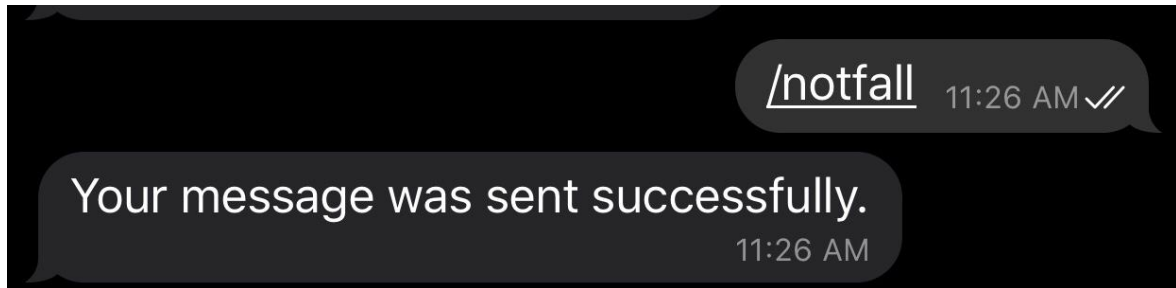


Abbildung 1.11: Notfallwarnmeldung

- 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.

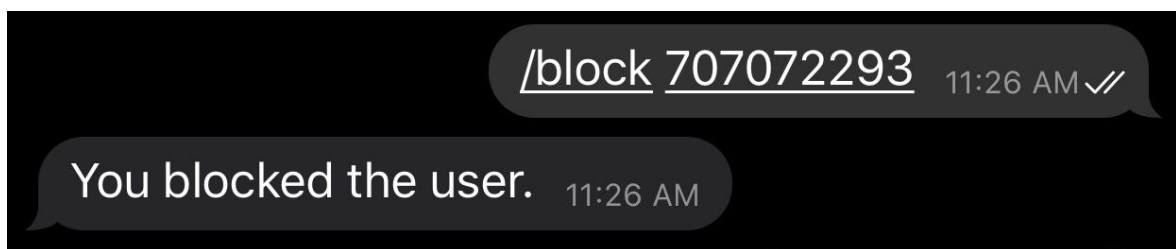


Abbildung 1.12: Benutzer Blockierung

- Funktion des Administrators, die Sprache zu ändern. In der folgenden Abbildung wurde die Sprache von dem Administrator auf deutsch geändert. Er kriegt eine Bestätigungsnachricht vom Chatbot, dass die Sprache auf deutsch eingestellt wurde. Das bedeutet, dass die Interaktion wird in der ausgewählten Sprache folgen und der Administrator wird alle Meldungen nur in dieser Sprache vom Chatbot bekommen.

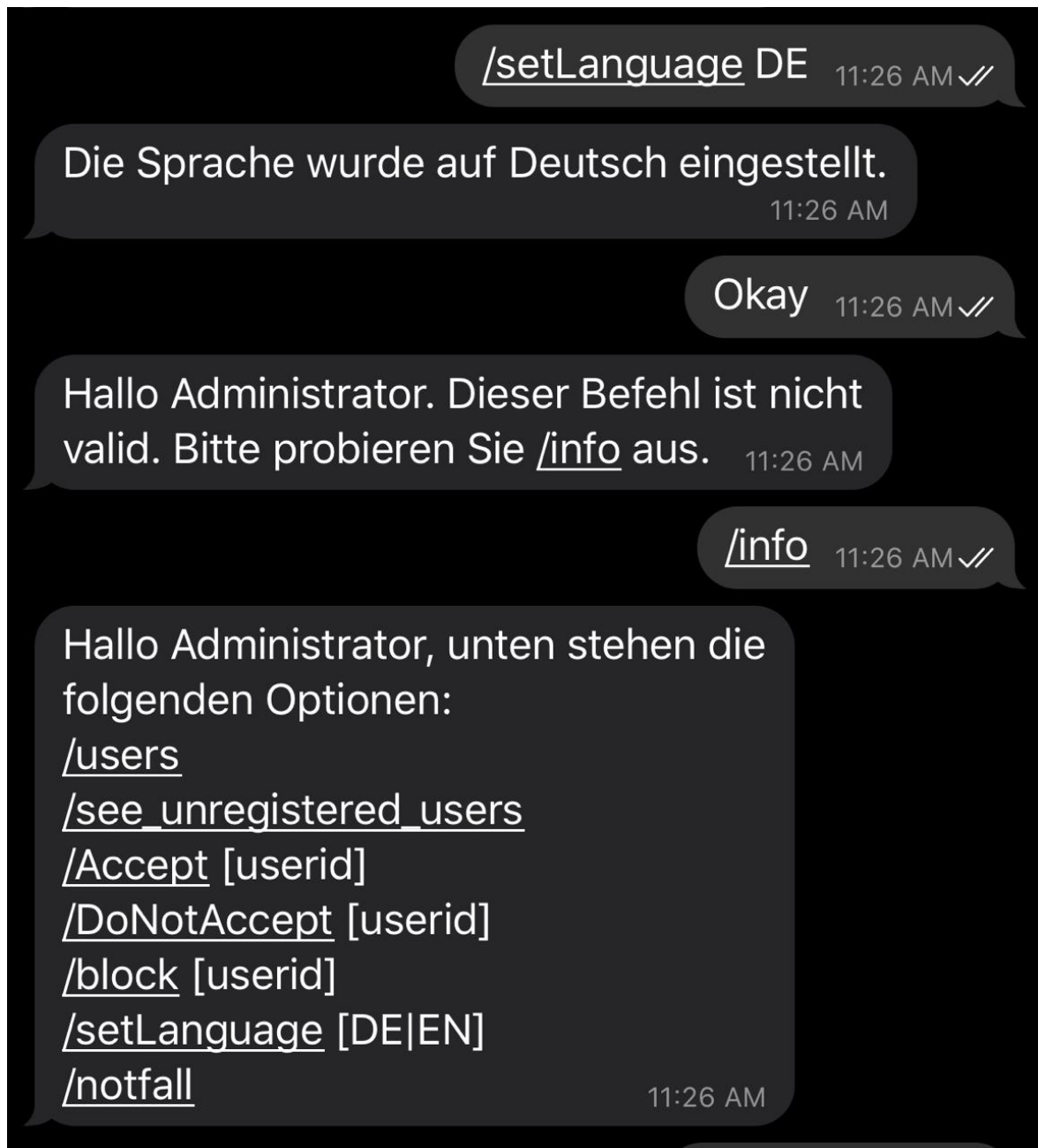


Abbildung 1.13: Spracheinstellungen

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

- Hauptmenü für den unregistrierten Benutzer

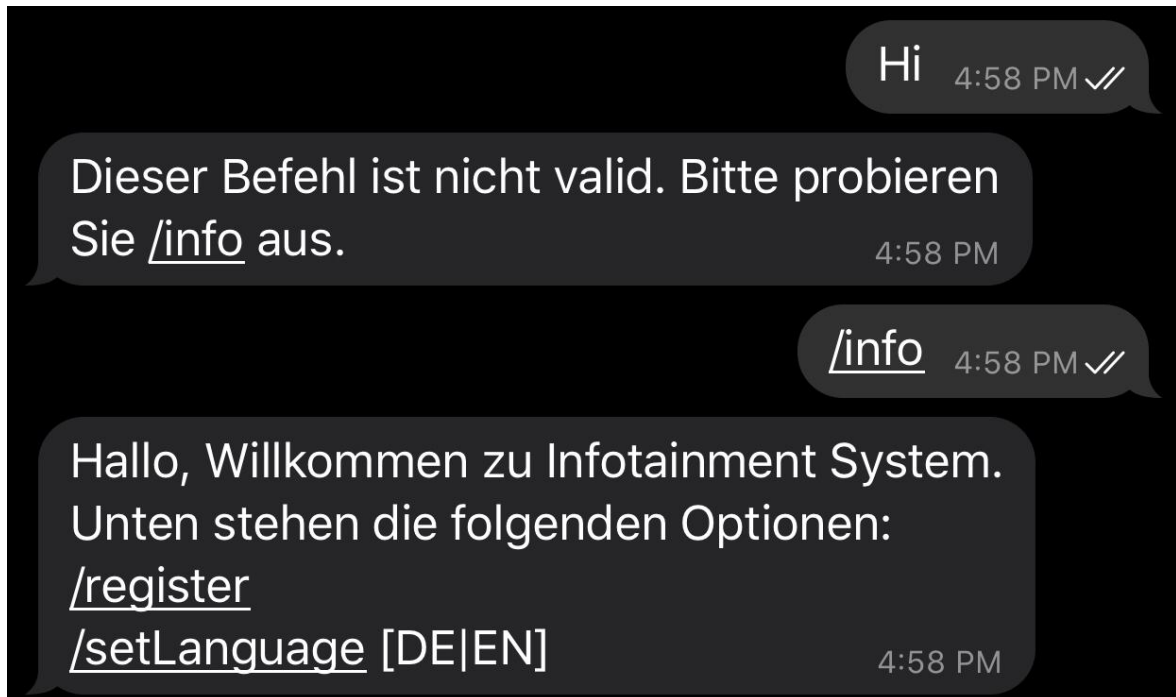


Abbildung 1.14: Hauptmenü für den unregistrierten Benutzer

- 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 ist, in welcher Format die Telefonnummer geschrieben werden soll.

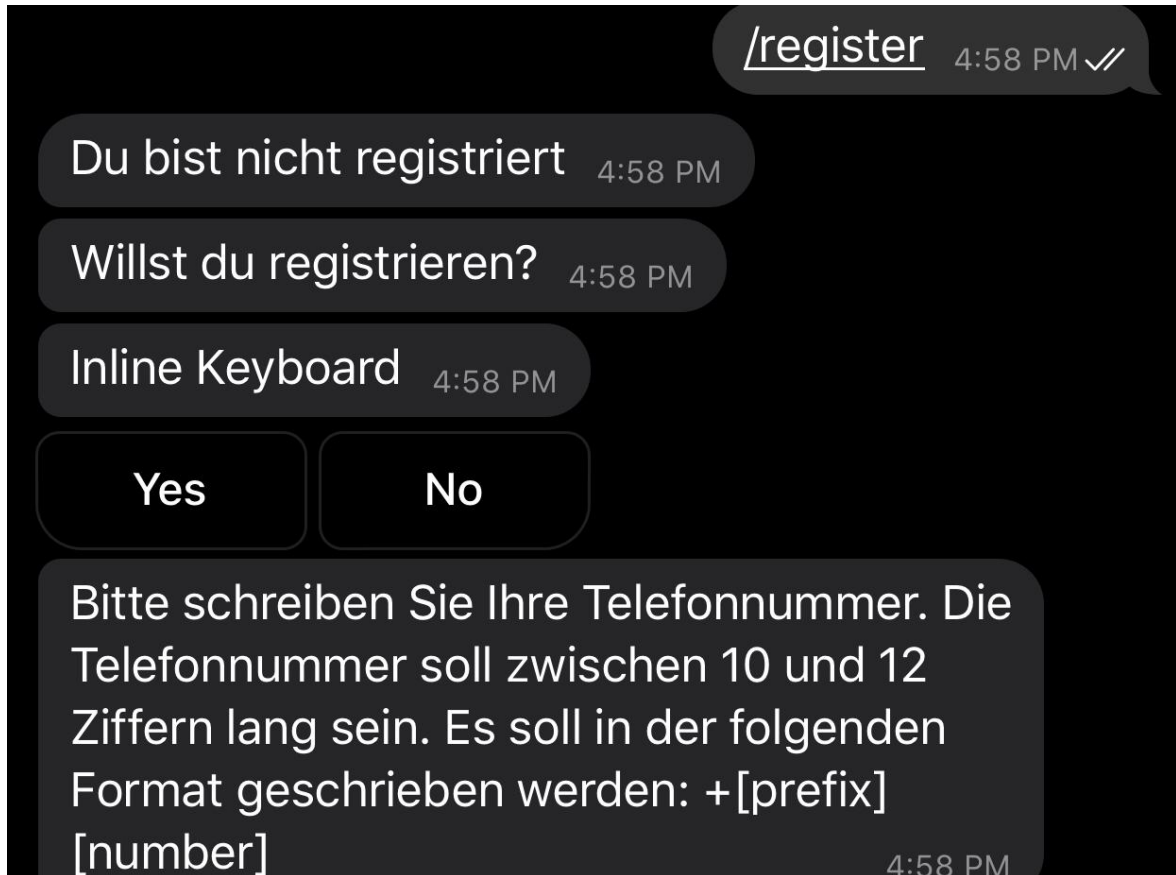


Abbildung 1.15: Registrierung beim Chatbot

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

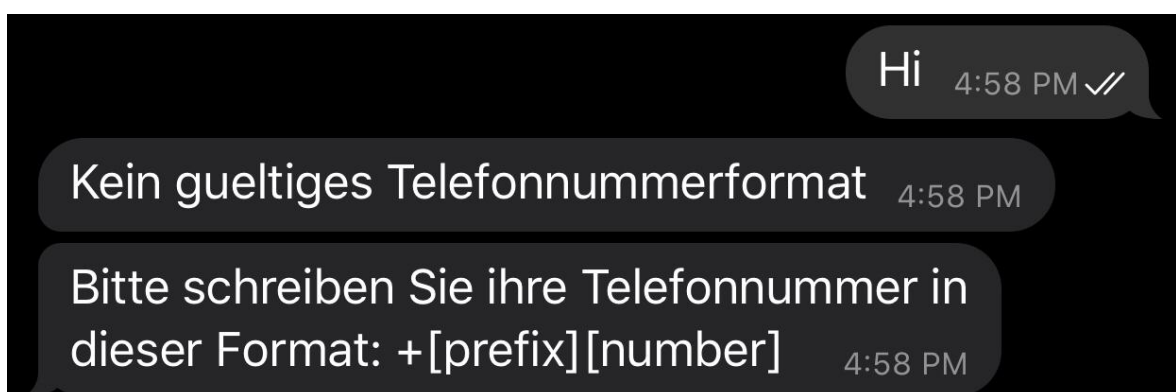


Abbildung 1.16: Falsche Eingabe des Telefonnummers

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

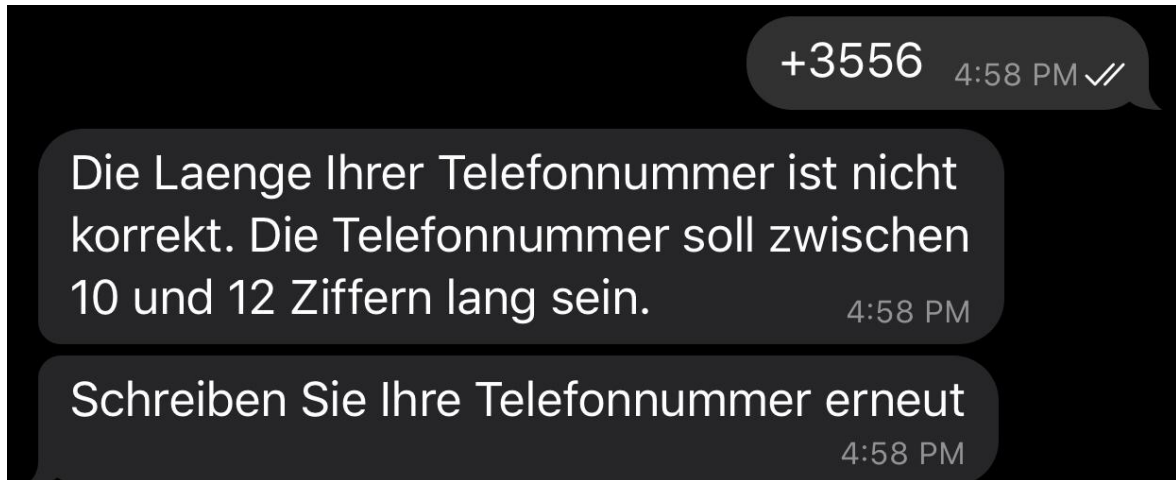


Abbildung 1.17: Telefonnummer zu kurz

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

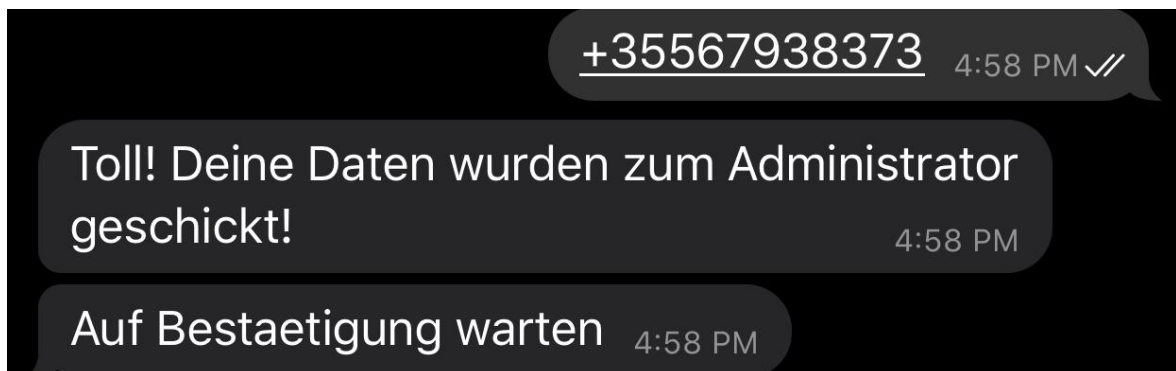


Abbildung 1.18: Korrekte Eingabe des Telefonnummers

- 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 andere Nachricht, die später kommen werden nicht berücksichtigt bis der Administrator die Genehmigung gegeben hat.

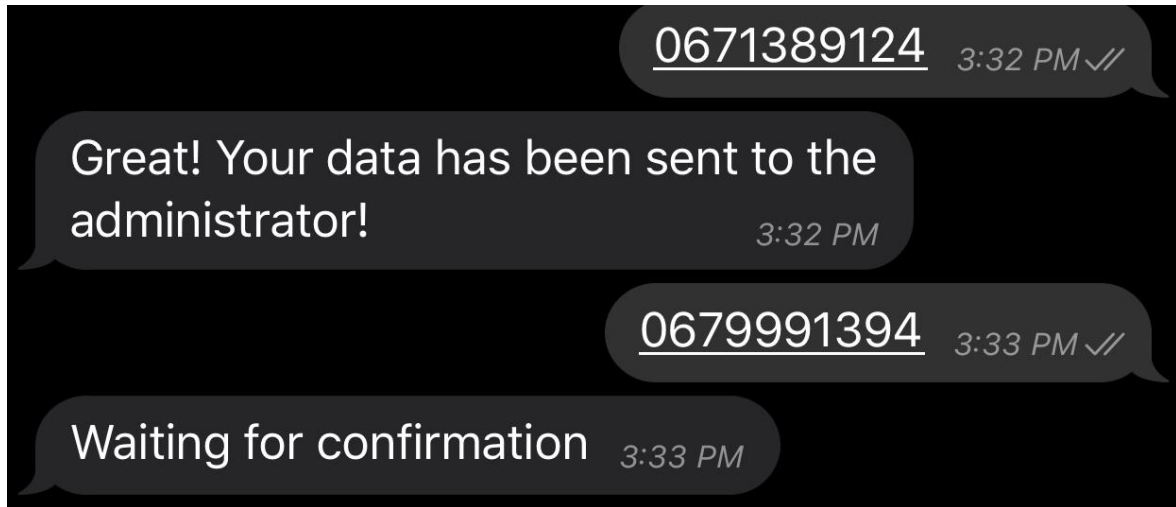


Abbildung 1.19: Das Warten der Benutzer auf die Administrator Bestätigung

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

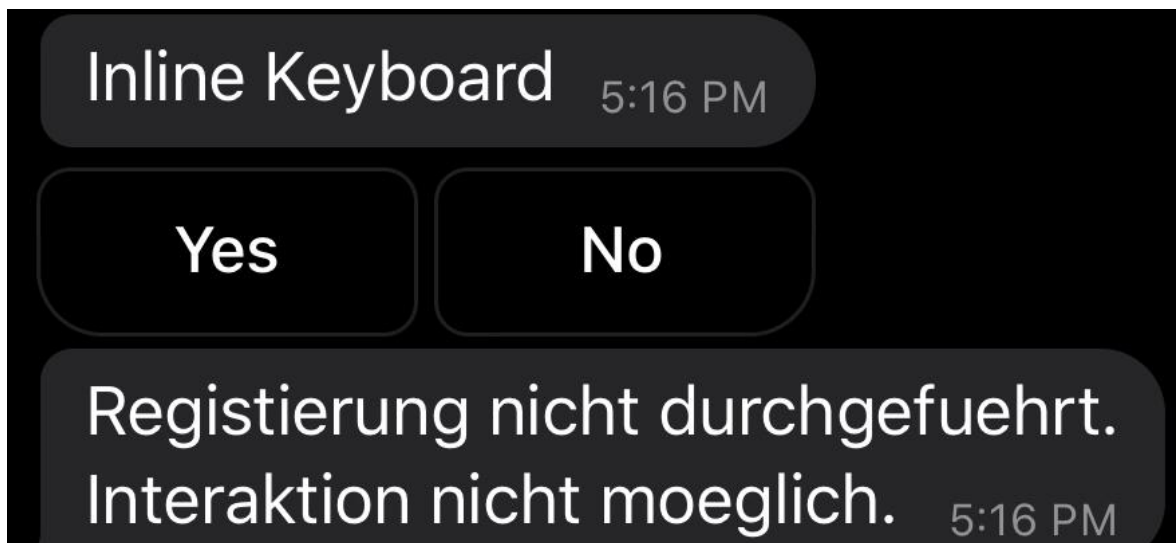


Abbildung 1.20: Keine Registrierung

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



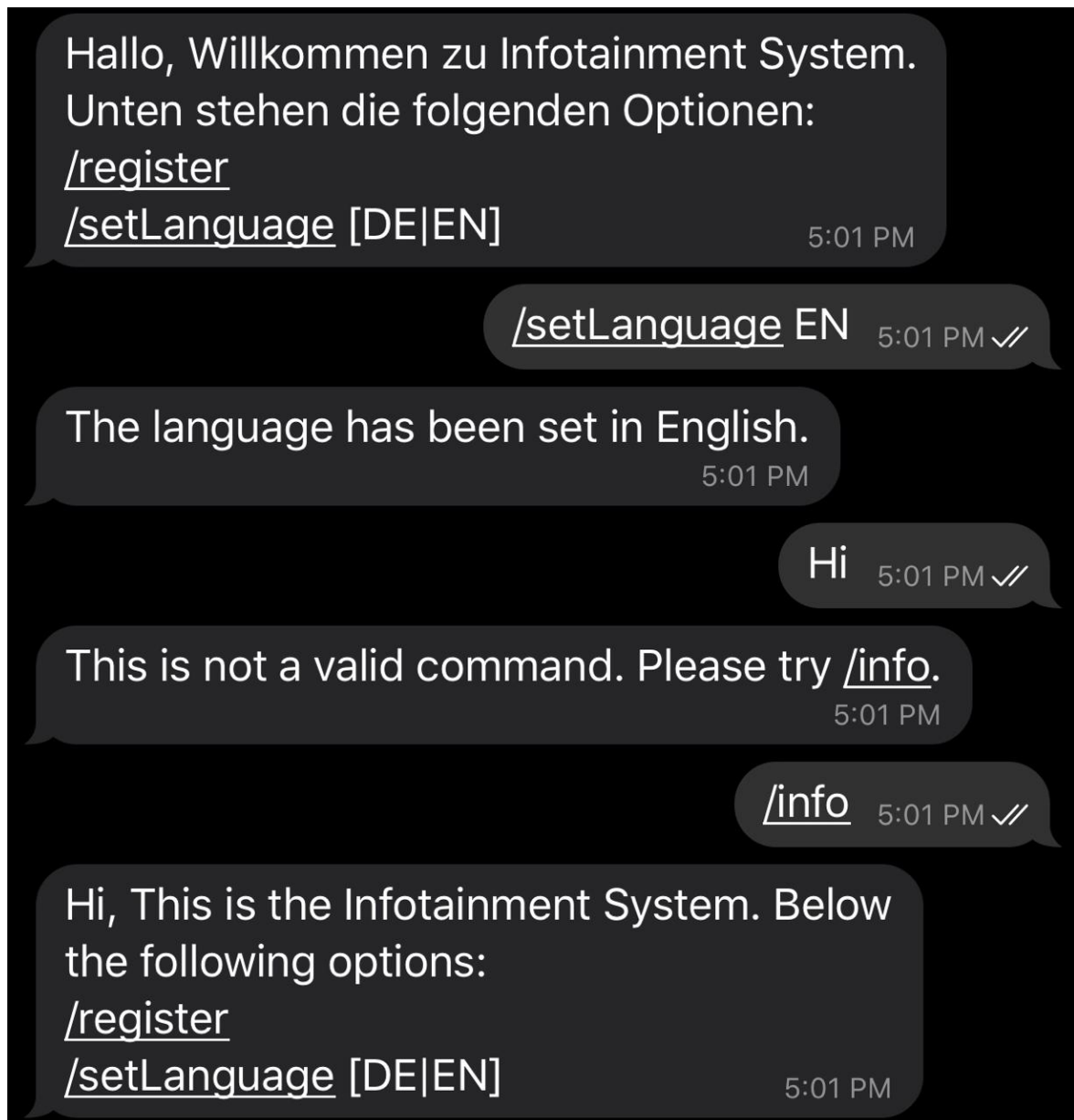


Abbildung 1.21: Spracheinstellungen für unregistrierte Benutzer

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

- Hauptmenür für einen normalen Benutzer

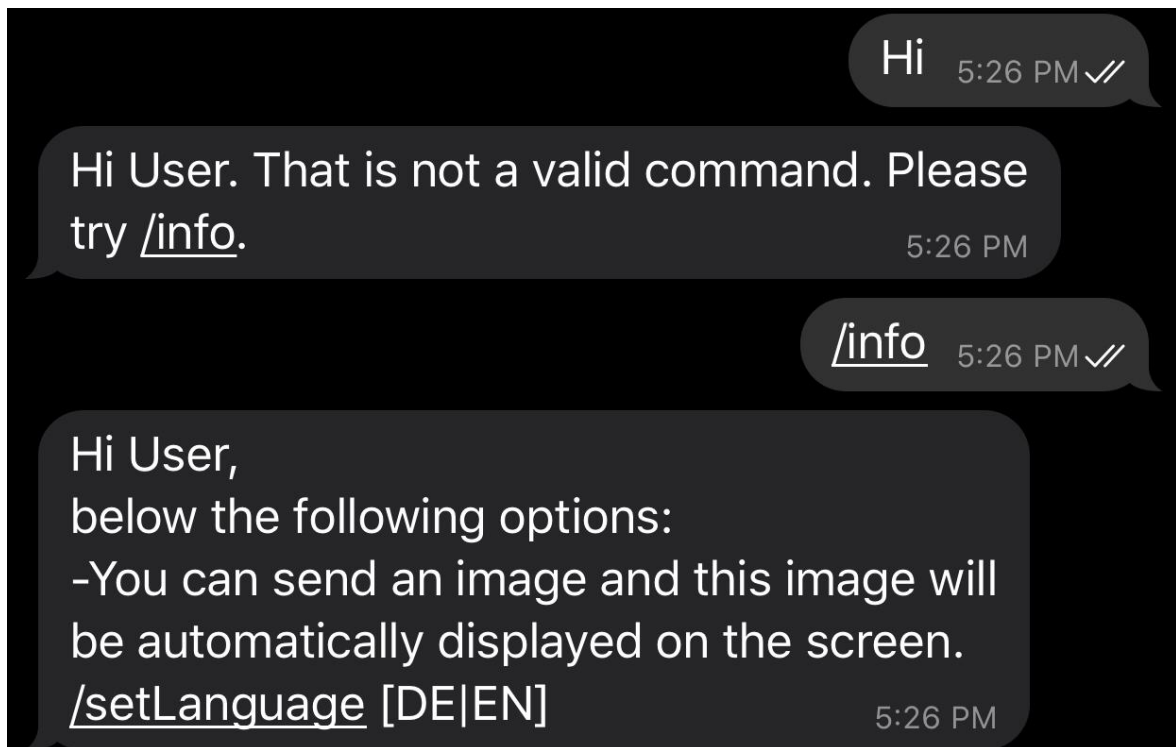


Abbildung 1.22: Hauptmenür für einen normalen Benutzer

- Bilderfunktionalität

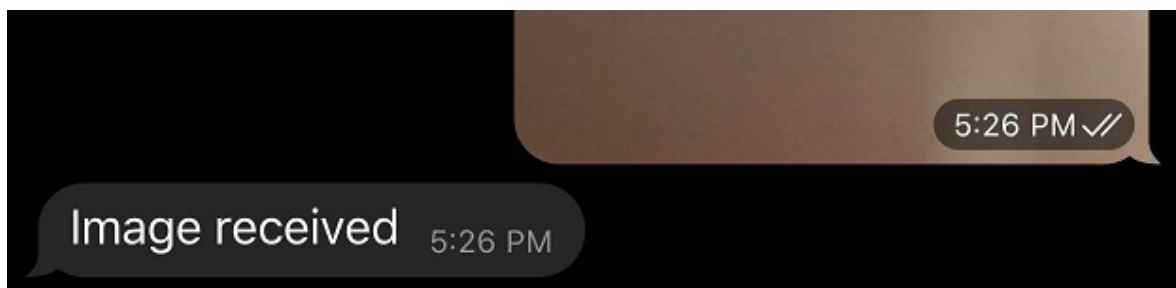


Abbildung 1.23: Bilderfunktionalität

- Spracheinstellungen für einen normalen Benutzer

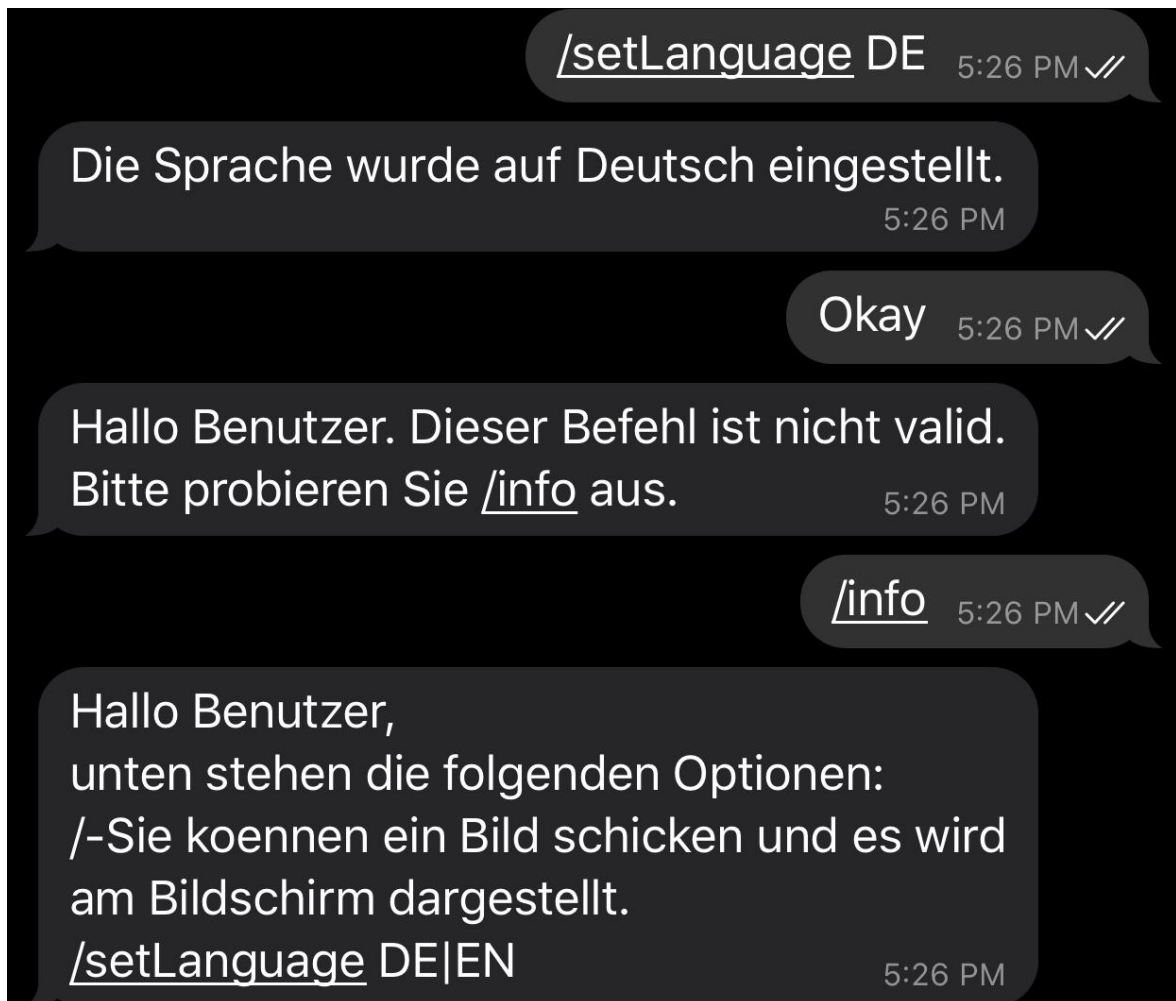


Abbildung 1.24: Spracheinstellungen für einen normalen Benutzer

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

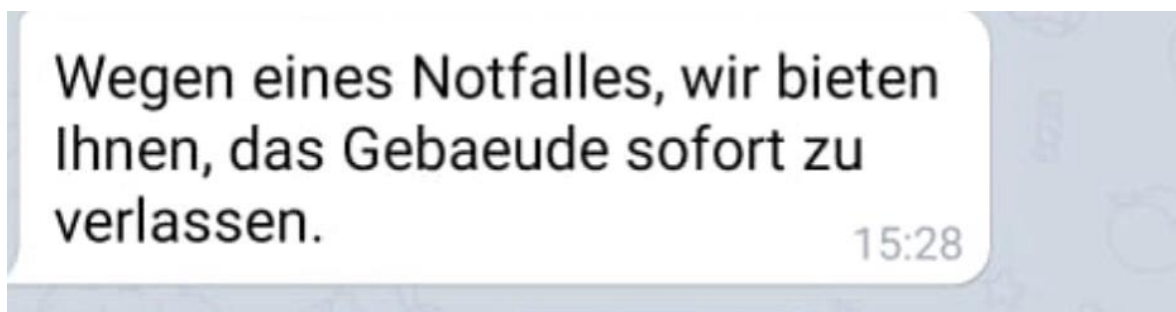


Abbildung 1.25: Notfallwarnmeldung vom Administrator

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.

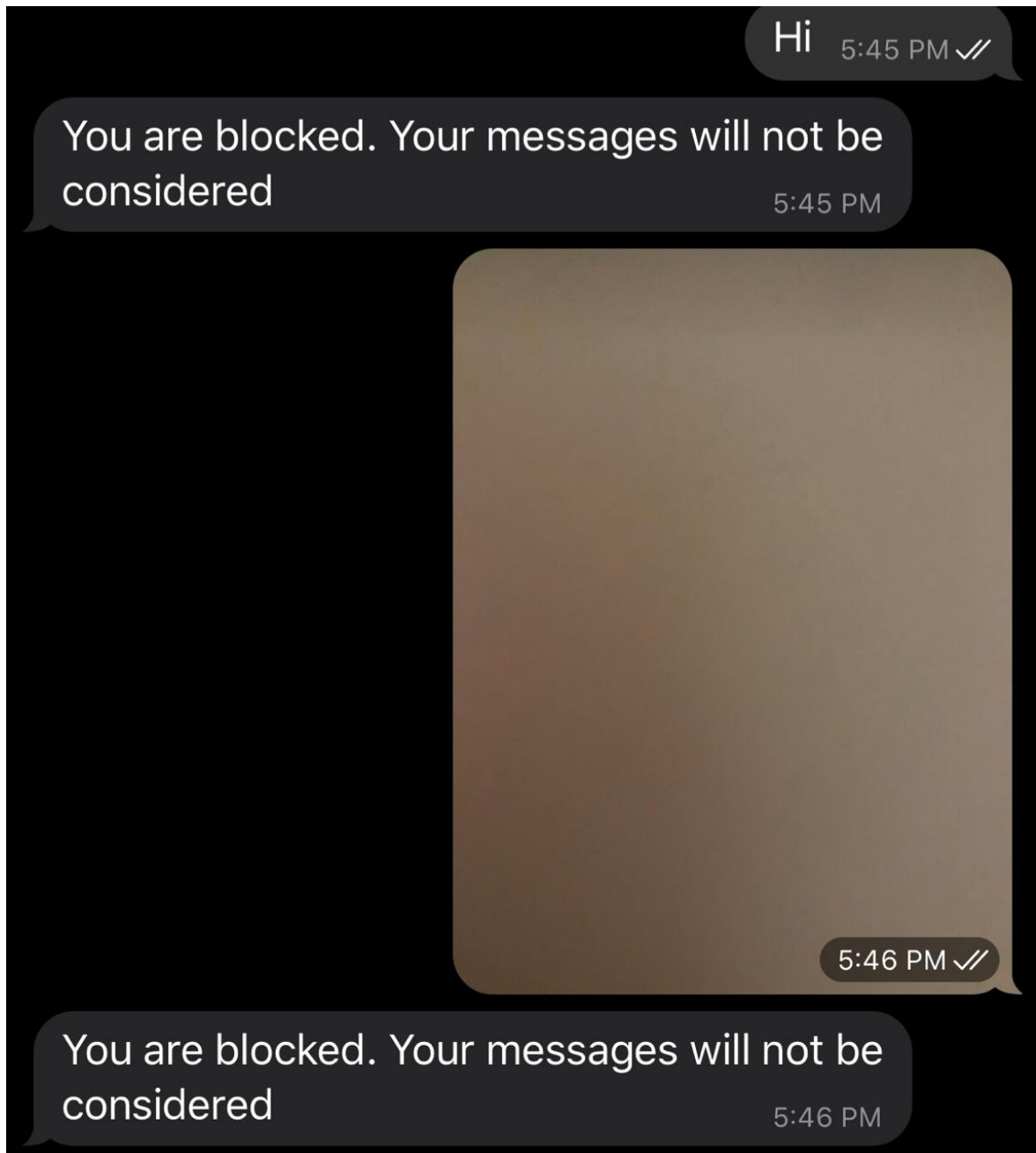


Abbildung 1.26: Nachricht von einem blockierten Benutzer

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 blockiert ist.

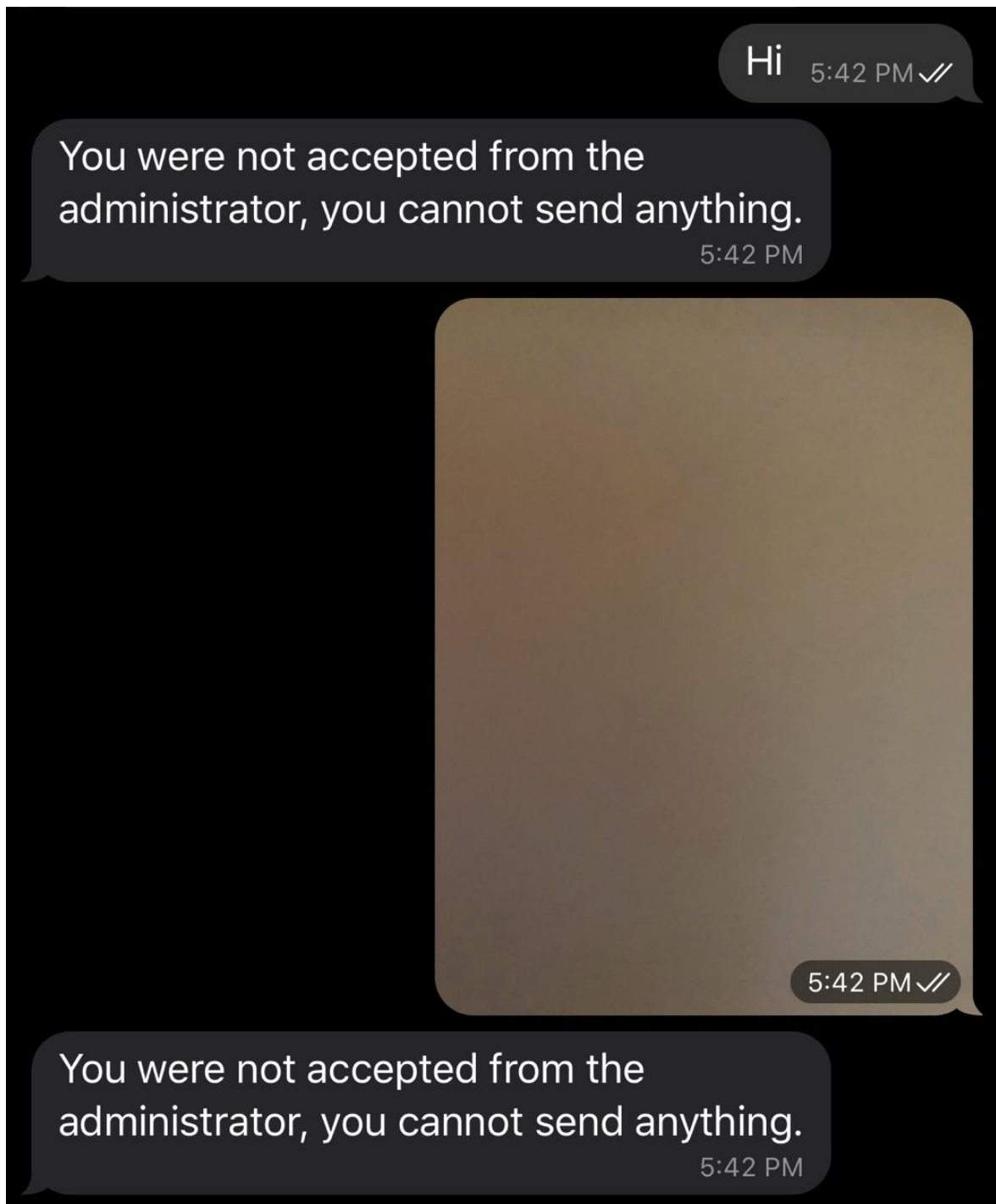


Abbildung 1.27: Nicht akzeptierter Benutzer

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

- Wenn ein Benutzer von dem Administrator akzeptiert wird, wird der Infotainment Bot ihm eine Nachricht schicken, dass er gerade genehmigt wurde und dass die Interaktion mit dem Chatbot möglich ist. Dieser Benutzer kriegt den Status einer normalen Benutzer.

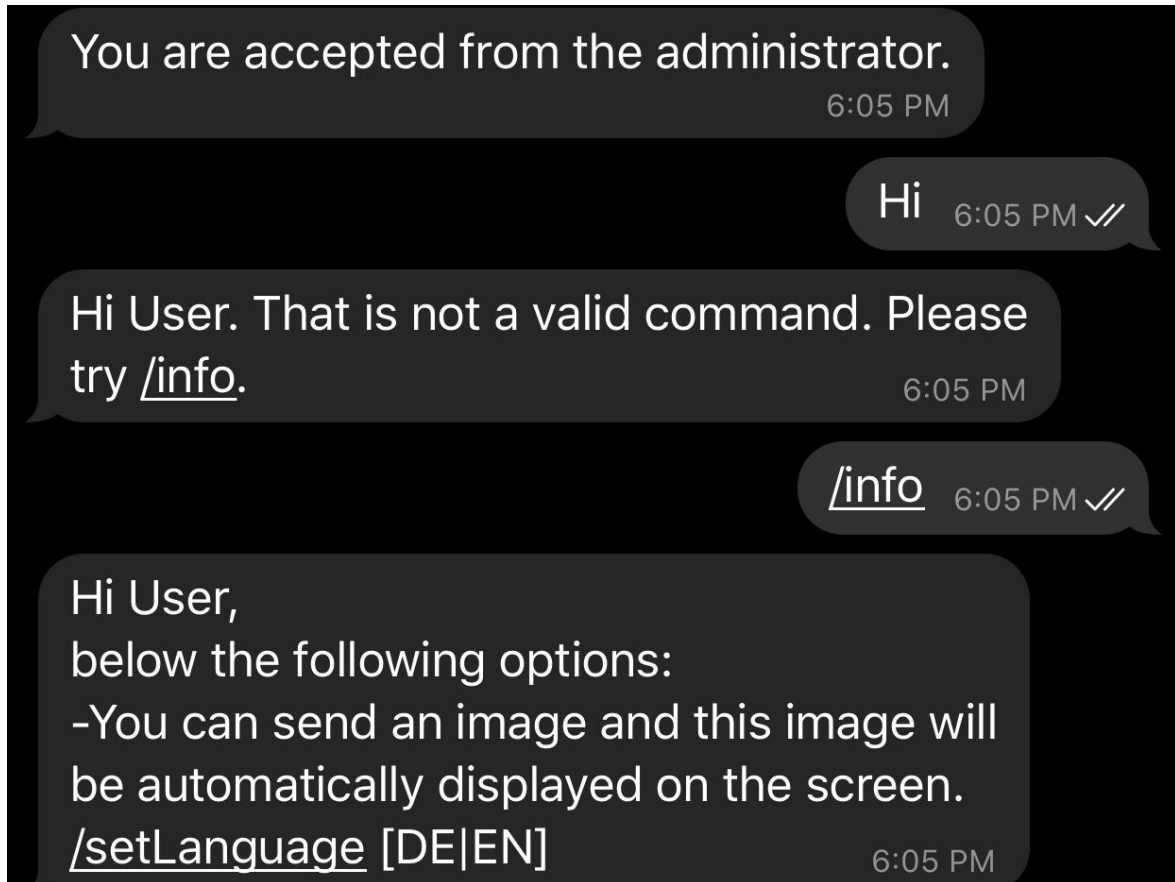


Abbildung 1.28: Akzeptierter Benutzer

## 1.10 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 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 Zeil gebraucht. Das sollte nicht für die Wetterdaten, sondern auch für die Kalenderinformationen gemacht werden.

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

### **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.

### **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 anzeigte, 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 sehr komplex und hat viel Zeit benötigt, bis eine Lösung gefunden wurde.

## **1.11 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 oder die Darstellung von Wetterdaten auf dem Bildschirm. Chatbot ist eine wichtige Komponente für die Qualitätsicherung, weil es viele Funktionen anbietet.

## **1.12 Ergebnisse - Irena Bala**

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst.

### **1.12.1 Implementierung**

Die wichtigsten Ergebnisse der Arbeit sind:



- Die Implementierung von einer Hauptkomponente der künstlichen Intelligenz wie Chatbot, die das entwickelte System interessanter macht.
- Designvorbereitung für die Anzeige, damit dort die Informationen auf unterschiedliche Weise dargestellt werden können
- 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 die Basis für die Zusammensetzung aller Komponenten anbietet
- Integration des Systems, damit alle Komponenten miteinander verbunden werden können

## 1.13 Handbuch für die Bedienung

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

### 1.13.1 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.

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, der für das erste Mal an ihm etwas schickt, wird der Chatbot ihn fragen ob er sich 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 muss die Telefonnummer eingegeben werden.

Danach soll gewartet, 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.

### 1.13.2 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, 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

## 1.14 Evaluierung und Resümee

### 1.14.1 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)

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, gebaut. Während der Arbeit wurden viele Adaptierungen gemacht. Es sind mehrere Tabellen als geplant erstellt. Eine Tabelle davon ist die Tabelle für die Chatbot Benutzer. Diese 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 miteinander in Zusammenhang sind und wenn eine davon geändert wurde, hatte diese Konsequenzen auf andere Komponenten.

### 1.14.2 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

# Abbildungsverzeichnis

1.1	Structured Software Design vom Server . . . . .	9
1.2	ERD . . . . .	10
1.3	Wetterdaten . . . . .	13
1.4	Chatbot Multilanguage . . . . .	21
1.5	Hauptmenü für den Administrator . . . . .	22
1.6	Auflistung der Chatbot-Benutzer . . . . .	23
1.7	Auflistung der unregistrierten Benutzer . . . . .	23
1.8	Keine Unregistrierte Benutzer . . . . .	24
1.9	Genehmigung des Administrators . . . . .	24
1.10	Ablehnung des Administrators . . . . .	24
1.11	Notfallwarnmeldung . . . . .	25
1.12	Benutzer Blockierung . . . . .	25
1.13	Spracheinstellungen . . . . .	26
1.14	Hauptmenü für den unregistrierten Benutzer . . . . .	27
1.15	Registrierung beim Chatbot . . . . .	28
1.16	Falsche Eingabe des Telefonnummers . . . . .	28
1.17	Telefonnummer zu kurz . . . . .	29
1.18	Korrekte Eingabe des Telefonnummers . . . . .	29
1.19	Das Warten der Benutzer auf die Administrator Bestätigung . . . . .	30
1.20	Keine Registrierung . . . . .	30
1.21	Spracheinstellungen für unregistrierte Benutzer . . . . .	31
1.22	Hauptmenür für einen normalen Benutzer . . . . .	32
1.23	Bilderfunktionalität . . . . .	32
1.24	Spracheinstellungen für einen normalen Benutzer . . . . .	33
1.25	Notfallwarnmeldung vom Administrator . . . . .	33
1.26	Nachricht von einem blockierten Benutzer . . . . .	34
1.27	Nicht akzeptierter Benutzer . . . . .	35
1.28	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](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).