

# Diplomarbeit

Name



AUTOR: TEST1

# 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	Technische Lösungen . . . . .	8

# Kapitel 1

## Umsetzung - Irena Bala

### 1.1 Allgemeine Beschreibungen

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

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

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

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

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

### 1.1.1 Chatbot

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

#### Einführung in Chatbot

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

#### Umsetzung von Chatbot

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

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

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

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

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

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

### 1.1.2 Server

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

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

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

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

### 1.1.3 Technologien

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

Tabelle 1.1: 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 Browser und Server

### Was ist Apache HTTP Server?

Der Apache HTTP<sup>1</sup> Server ist ein weltweit verbreiteter Web Server. Dieser Server ist Open Source, das bedeutet, dass es keine Lizenz gekauft werden soll, um es zu verwenden. Es ist kompatibel auf allen kohärenteren Betriebssystemen, beispielsweise Linux, Windows, Mac OS und andere. Es bietet viele Versionen an, die zu unterschiedlichen Anwendungsgebiete passen und verbesserte Eigenschaften bereitstellen. Durch dieses Webserver können Webseiten erstellt werden. Die Erstellung der Webseiten erfolgt über serverseitige Scriptsprachen, die von dem Server selbst nicht unterstützt werden. Sie werden als Zusatzfunktionen angehängt.

---

<sup>1</sup>Hypertext Transfer Protocol

## 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>2</sup> Anforderung an den Server, auf dem die Webseite gespeichert ist.“

Der Apache HTTP Server bietet viele Funktionalitäten an, die seine Entwicklungsumgebung vergrößern. Die wichtigste davon ist die Möglichkeit der Integration eines SSL <sup>3</sup>Zertifikats. Das ermöglicht die Übertragung der Daten in einer verschlüsselten Form. Die detaillierte Funktionsweise eines SSL Zertifikats wird in den Unterkapiteln beschrieben.

## Was ist MySQL?

MySQL ist ein weitverbreitetes relationales Datenbanksystem. Die Datenbanksysteme werden allgemein zur Datenspeicherung und Datenverwaltung verwendet. Ein wichtiges Kriterium für die Datenspeicherung ist die Performanz. Diese Anforderung wird durch MySQL optimal erfüllt. Das ist auch der Grund, warum dieses Datenbanksystem so populär und bekannt ist. Die von MySQL für die Abarbeitung, Verwaltung und Systematisierung von Daten verwendete Sprache ist SQL<sup>4</sup>. MySQL ist auch eine Open Source Software, die in meisten Fällen in Verbindung mit serverseitigen Scriptsprachen wie PHP, vorkommt.

## Funktionsweise von MySQL

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

## Was ist PHP?

PHP<sup>5</sup> ist eine serverseitige Programmiersprache. Das bedeutet, dass diese Sprache, um die vom Server auszuführenden Ereignissen zu programmieren, verwendet wird. PHP ist eine sehr verbreitete Programmiersprache, die am meisten zur Erstellung und

---

<sup>2</sup>Hypertext Transfer Protocol Secure

<sup>3</sup>Secure Socket Layer

<sup>4</sup>Structured Query Language

<sup>5</sup>Parallel History Project

Programmierung von Webseiten verwendet wird. Eigentlich ist PHP sehr flexibel, denn es einen großen Schnittstellenansatz anbietet. Diese Programmiersprache kann auch im Zusammenhang mit Datenbanken genutzt werden.

### **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.

### **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.

### **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”

## Was ist Python?

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

## Merkmale von Python

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

## Was ist HTML?

HTML<sup>6</sup> ist keine Programmiersprache, die wird für die Erstellung von Inhalten bei Webseiten verwendet. Diese Inhalte können Texte, Bilder oder andere Komponente sein. HTML wird als eine Auszeichnungssprache betrachtet. Sie ist nicht nur für die Erstellung von Webseite-Inhalten zuständig, sondern auch für ihr Design. Diese Sprache liegt mithilfe von bestimmten Tags die Struktur einer Webseite fest. Im Tag werden die Inhalte gespeichert. Es gibt bestimmte Tags für verschiedene Layout-Elemente.

## 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, Zitaten) zu entkoppeln.“

---

<sup>6</sup>HyperText Markup Language

<sup>7</sup>Cascading Style Sheets



## Was ist Raspberry PI?

Der Raspberry PI ist ein Minirechner, die zur Linux Programmierung, Shell Scripting und Realisierung von technischen Projekten verwendet wird. Es braucht eine Tastatur, eine Maus, ein Netzteil, VGA<sup>8</sup> und HDMI<sup>9</sup>-VGA Konverter, damit es genutzt 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.

## 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, der der neueste und modernste Standard von SSL ist.

## SSL-Verschlüsselung

Um eine verschlüsselte Verbindung zwischen einem Browser und einem Server aufzubauen, werden SSL – Zertifikaten integriert. Mittels ein SSL Zertifikats wird die Authentizität einer Webseite überprüft. Das SSL Zertifikat wird von einer Zertifizierungsstelle, erzeugt. Diese Zertifizierungsstelle heißt 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 zwischenübertragenen Daten ein öffentlicher, -und ein privater Schlüssel. Je grösser die Länge des Schlüssels ist, desto sicherer und besser ist. Meistens werden Schlüssel mit einer Länge von 256 Bit verwendet.

## 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. Diese Funktionen sind in der eigenen Dokumentation von Telegram Bot API klar beschrieben.

## Was ist Wetter API?

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

---

<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.“

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

Tabelle 1.2: Bereiche und Technologien

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

## 1.2 Technische Lösungen

### Structured Software Design

#### Konfiguration von Raspberry PI Server

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

---

<sup>14</sup>JavaScript Object Notation

## Datenbank

- Zuerst wurde ein ERD Diagramm mit Papier gezeichnet. Das Ziel war die richtige Erstellung der benötigten Tabellen. Die Tabellen wurden mit den Spalten und ihren Datentypen erstellt. Es wurden auch die Kardinalitäten dazwischen gezeichnet. Die erstellten Tabellen waren:
  - Supplierplan
  - Stundenplan
  - Wetterdaten
  - Chatbot Users
- Basierend auf das ERD Diagramm wurden dann die Tabellen mit phpmyadmin erstellt
- Dann sind für alle Tabellen mit MySQL Workbench die entsprechenden gespeicherten Prozeduren erstellt

## 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

## Wetterdaten

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

## Anzeige

## Chatbot

### Chatbot Einrichtung:

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

## Einrichtung des Chatbots in RaspberryPI

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

## Grundlagen für die Umsetzung

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

## Funktionalitäten

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 Buttonen, die im Bot integriert werden. Falls der unregistrierte Benutzer Ja drückt, bittet er ihn die Telefonnummer einzugeben. Durch diese Telefonnummer wird dann der Benutzer in der Datenbank gespeichert. Falls nein, wird ihm nur eine Nachricht vom Chatbot zurückgeschickt.

Die Telefonnummer, die von dem unregistrierten Benutzer eingegeben wird soll nach Inhalt überprüft werden. Wenn der Benutzer z.B Text eingibt, wird das nicht genehmigt und nicht gespeichert. Falls eine Zahl ist, soll es 10 Ziffern lang sein und mit einem vorgelegten Prefix anfangen. Ansonsten wird es nicht genehmigt und der unregistrierter Benutzer kriegt wieder Möglichkeiten vom Chatbot die Nummer richtig zu schreiben.

Die Informationen über den unregistrierten Benutzer wird der Administrator bekommen. Er soll dann die Genehmigung geben, ob diese Person schon eine Interaktion mit dem Chatbot haben darf oder nicht.

### Normale Benutzer:

Die Daten über die normalen Benutzer sind in der Datenbank gespeichert. Diese Benutzer sind von dem Administrator genehmigt. Die Nachrichten, die diese normalen

Benutzer zum Chatbot schicken werden nach Inhalt überprüft. Falls sie ein Bild schicken, wird dieses Bild auf die Datenbank gespeichert und auf dem Bildschirm angezeigt. Wenn aber der Benutzer eine Nachricht schickt, die kein Bild ist, wird es keine Interaktion mit dem Bot geben, weil dem Bot interessieren die anderen Nachrichtformate nicht.

### Administrator:

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

### Umsetzung der Funktionalitäten:

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

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

Listing 1.1: Chatbot 1

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

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

Listing 1.2: Chatbot 2

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

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

Listing 1.3: Chatbot 3

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

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

Listing 1.4: Chatbot 3

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

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

Listing 1.5: Chatbot 4

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

```
def on_callback_query(msg):
    query_id, from_id, query_data=
    telepot.glance(msg, flavor='callback_query')
    bot.answerCallbackQuery(query_id, text='getIt')
    if query_data=='press1':
        bot.sendMessage(from_id, text="ok")
        query2=("Select c_id from tb_infotainment_chatbot_users
                where c_id=%s") %(int(from_id))
        count1=curs.execute(query2)
    if count1 > 0:
        bot.sendMessage(from_id, text="Please write your
        phone number below")
```

Listing 1.6: Chatbot 5

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

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

Listing 1.7: Chatbot 6

## Konkrete Beispiele mit Bildern

In diesem Unterkapitel werden Bilder dargestellt, die die konkrete Umsetzung des Chatbots zeigen.

- Dieses Bild zeigt die Möglichkeit, die der Chatbot dem unregistrierten Benutzer gibt





Abbildung 1.1: Chatbot

# Abbildungsverzeichnis

1.1	Chatbot . . . . .	15
-----	-------------------	----

# Tabellenverzeichnis

1.1	Technologien . . . . .	3
1.2	Bereiche und Technologien . . . . .	8

# Literatur