

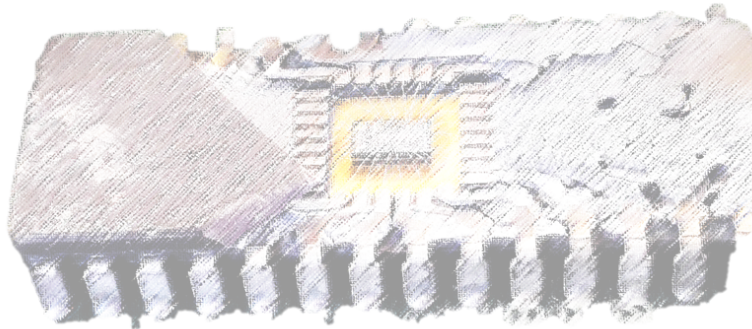
Höhere technische Schule für Informationstechnologie
Shkolla e mesme profesionale private për teknologji informacioni

Österreichische Schule Peter Mahringer
Shkolla Austriake Shkodër

Infotainment System

Diplomarbeit Nr. 20.08

Klasse 5ay, Schuljahr 2019/20



Ausgeführt von: Irena Bala
Aldo Sheldija

Projektbetreuer 1: Ing. Welat Abdall, MSc
Projektbetreuer 2: Dipl. Ing (FH) Dominik Stocklasser MSc
Projektbetreuer 3: Frenk Kasmi, MSc

Shkoder, 3. März 2020

Eidesstattliche Erklärung

Wir versichern, dass wir die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt haben. Wir haben uns keiner anderen als der im beigefügten Quellenverzeichnis angegebenen Hilfsmittel bedient. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Irena Bala

Ort, Datum

Unterschrift

Aldo Sheldija

Ort, Datum

Unterschrift

Approbation Datum u. Unterschrift	PrüferIn	IT-Koordinator/Direktion
--------------------------------------	----------	--------------------------

Sämtliche in dieser Diplomarbeit verwendeten personenbezogenen Bezeichnungen sind geschlechtsneutral zu verstehen.

Kurzfassung

In der vorliegenden Diplomarbeit geht es um die Entwicklung eines Systems, wo die wichtigsten Ankündigungen des Tages auf einem Bildschirm dargestellt werden. Auf diese Weise wird eine Innovation im Erhalt der Informationen für die Schule erreicht.

Deswegen war es notwendig, ein System zu erstellen, wo alle diese Informationen in einer digitalisierten Form angezeigt werden. Dieses System heißt Infotainment-System. Der erstellte Prototyp besteht aus einem Raspberry PI Client und einem Raspberry PI Server. Im Raspberry PI Client befindet sich eine Website, die mit dem Server kommuniziert. Der Administrator kann sich bei der Webseite einloggen und um dort die verschiedenen Funktionalitäten zu verwalten. Er erhält die Möglichkeit, die Informationen auszuwählen, die er auf dem Bildschirm anzeigen lassen will, sowie auch das entsprechende Layout.

Die Informationen werden direkt aus einer von uns erstellten Datenbank ausgewählt, die schulrelevante Informationen wie Unterrichtsplan, Lehrerplan usw. enthält. Am Bildschirm können auch noch Kalenderinformationen (Termine, Olympiade), Wetterdaten und letzter Post von der Webseite der Schule dargestellt werden. Die Wetterdaten und den letzten Post bekommen wir mithilfe von APIs. Unser System kann mehrere Bildschirme mit unterschiedlichen Inhalten haben. Alle Bildschirme können auf relativ einfache und effiziente Weise von der Webseite aus verwaltet werden.

Das Infotainment-System ist auch zu Unterhaltungszwecken zu verwenden. Über Chatbot haben die Schüler die Möglichkeit, verschiedene Fotos zu posten, die auf dem Bildschirm angezeigt werden. Dies können Bilder des Alltags sein, die den Schultag auf die eine oder andere Weise interessanter machen.

Abstract

This diploma thesis deals with the development of a system where the most important announcements of the day are displayed on a screen. In this way, an innovation in obtaining information for the school is achieved. The previous method of writing information on paper was very tiresome and time consuming. Every day, new papers were printed, which were then hung on the information board.

That was not an appealing task to the students. Therefore, it was necessary to create a system where all this information is displayed in a digitized form. This system is called infotainment system and consists of a Raspberry PI Client and a Raspberry PI Server. The Raspberry PI Client has a Website that communicates with the server. The administrator can log in to the website where they have the opportunity to select the information they want to display on the screen, as well as the corresponding layout. The information is selected directly from a database created by our team that contains school-related information such as lesson plan, teacher plan, and so on.

Calendar information (dates, Olympics), weather data and the latest posts from the school's website can also be displayed on the screen. We weather data and the latest posts can be received using APIs.

Our system can have multiple screens with different content. All screens can be managed in a relatively simple and efficient way from the website. The infotainment system can also be used for entertainment purposes. Chatbot allows students to post various photos which are going to be displayed on screen. The photographs will depict the daily school life of a Peter Mahringer student.

Përmbledhje

Ideja e këtij projekti është realizimi i një sistemi, që ka si funksion paraqitjen e njoftimeve më të rëndësishme të ditës në një ekran. Në këtë mënyrë, do të arrihet një risi në mënyrën e transmetimit së informacioneve përkatëse për shkollën.

Metoda e mëparshme e përcjelljes së informacionit ka qenë tepër e lodhshme dhe kërkonte një kohë të konsiderueshme. Prandaj, ka qenë i nevojshëm krijimi i një sistemi ku të gjitha informacionet të paraqiten në një formë të dixhitalizuar. Prototipi që ne kemi krijuar përbëhet nga dy minikompjuterat Raspberry PI, ku njëri kryen funksionin e klientit dhe tjetri atë të serverit. Tek klienti ndodhet një uebit që komunikon në mënyrë direkte me serverin. Administratori mund të logohet aty dhe të gjejë funksionalitete të ndryshme që i disponohen. Ai ka mundësinë që të zgjedhë informacionet që dëshiron të paraqesë në ekran, duke i selektuar ato direkt nga një bazë të dhënash. Kjo bazë e dhënash është krijuar nga ne dhe mbart informacionet relevante për shkollën. Administratori mund të zgjedhë edhe një strukturë përkatëse për mënyrën e paraqitjes së këtyre të dhënave.

Në ekran do të tregohen edhe njoftime të rëndësishme si për shembull datat e olimpiadave apo pushimeve. Një funksionalitet i mëtejshëm konsiston në paraqitjen e të dhënave për motin apo edhe posti i fundit i uebit të shkollës. Kjo arrihet me ndihmë të API-ve. Një tjetër tipar inovativ është fakti që sistemi ynë mund të përdoret edhe për qëllime argëtuese. Nëpërmjet Chatbotit, që është një komponent i inteligjencës artificiale, studentët mund të postojnë vetë foto të ndryshme në ekran. Këto mund të jenë fotografi të jetës së përditshme, që në një mënyrë apo në një tjetër thyejnë monotoninë gjatë rutinës ditore shkollore.

Danksagung

Wir möchten unserem Betreuer, Herrn Welat Abdall, für die Räte, die er uns gegeben hat, und für die Bereitschaft, die er uns gezeigt hat, danken. Wir möchten uns auch bei Professor Dominik Stocklasser bedanken, der uns auch bei unserer Arbeit unterstützt hat und immer bereit war, uns zu helfen. Ein großes Dankeschön geht auch an die Professoren Florian Offner und Christoph Schratt, die uns Anweisungen gegeben haben und uns bei den verschiedenen Problemen, die wir gehabt haben, geholfen haben.

Inhaltsverzeichnis

1	Allgemeines	1
1.1	Idee, Thema, Aufgabenstellung	1
1.2	Team	1
1.3	Allgemeines	2
2	Planung	3
2.1	Projektziele	3
2.1.1	Muss-Ziele	3
2.1.2	Optionale-Ziele	4
2.1.3	Nicht-Ziele	4
2.2	Projektplanung	4
2.2.1	Meilensteine	4
2.2.2	Projektstrukturplan	4
2.2.3	Arbeitspakete	5
2.3	Projektmanagementmethode	9
2.3.1	Structed Design	9
2.3.2	Hybrides Modell	9
3	Dokumentation des Projektverlaufs	12
3.1	Beschreibungen des Arbeitsverlaufs	12
3.1.1	Irena Bala	13
3.1.2	Aldo Sheldija	18
4	Umsetzung - Irena Bala	21
4.1	Allgemeine Beschreibungen	21
4.1.1	Chatbot	22
4.1.2	Server	22
4.1.3	Technologien	23
4.2	Structured Software Design	28
4.3	Konfiguration von Raspberry PI Server	29
4.4	Datenbank	30

4.5	SSL Verschlüsselung	31
4.6	Anzeige	32
4.7	Offline Betrieb	32
4.8	Wetterdaten	33
4.9	Chatbot	34
4.9.1	Chatbot Einrichtung	34
4.9.2	Einrichtung des Chatbots in RaspberryPI	35
4.9.3	Grundlagen für die Umsetzung von Chatbot	35
4.9.4	Funktionalitäten von Chatbot	35
4.9.5	Umsetzung der Funktionalitäten	37
4.9.6	Konkrete Beispiele mit Bildern	42
4.10	Probleme, Herausforderungen und deren Lösung	57
4.11	Qualitätssicherung, Controlling	58
4.12	Ergebnisse - Irena Bala	58
4.12.1	Implementierung	59
4.13	Handbuch für die Bedienung	59
4.13.1	Beschreibung der Bedienung als User	59
4.13.2	Beschreibung der Bedienung als Administrator	60
4.14	Evaluierung und Resümee	60
4.14.1	Planung vs Realisierung	60
4.14.2	Wertschöpfung und Lessons Learned	61

Kapitel 1

Allgemeines

1.1 Idee, Thema, Aufgabenstellung

Diese Diplomarbeit wird von zwei Schülerinnen der Österreichischen Schule “Peter Mahringer” in Shkodra geschrieben. Die Idee des Projekts ist, ein System zu entwickeln, wo die wichtigsten Benachrichtigungen des Tages für unsere Schule auf einem Bildschirm dargestellt werden. Das war notwendig, weil die frühere Arbeit sehr aufwendig und ungünstig war.

Dieses Thema ist sehr wichtig, weil in der heutigen Zeit es eine weitverbreitete Umgebung für die Anwendungen dieses Systems gibt. Bei vielen Unternehmen ist es erforderlich, die Informationen so schnell wie möglich darzustellen, damit die Kunden immer auf dem Laufenden sind. Momentan wird dieses System für die Schule angepasst.

Es wird eine Webseite mit Login programmiert, wobei der Administrator die Möglichkeit hat, verschiedene Informationen mit dem passenden Layout auf dem Bildschirm anzeigen zu lassen. Diese Informationen werden direkt aus einer selbsterstellten Datenbank selektiert. Zusätzlich, werden auch Kalenderinformationen wie z.B Olympiaden, wichtige Termine aber auch Wetterdaten oder der letzte Post von der Webseite dargestellt. Die Wetterdaten und der letzte Post von der Homepage der Schule werden mithilfe von APIs aus dem Internet geholt. Dieses System bietet viele Bildschirme mit unterschiedlichen Inhalten an. Der Administrator kann die Inhalte von fern für jeden Bildschirm ändern.

Eine weitere Funktionalität des Systems wird Chatbot sein. Chatbot ist ein sehr wichtiger Teil von künstlicher Intelligenz, deswegen war es wichtig, diesen Komponent zu involvieren.

1.2 Team

Das Projektteam besteht aus zwei Personen: Irena Bala und Aldo Sheldija. Irena Bala ist die Projektleiterin und Aldo Sheldija ist stellvertretender Projektleiter. Seit zwei Jahren sind sie in einer Klasse zusammen. Sie haben aber auch früher zusammengearbeitet, deswegen kennen sie sich gut. Die Mitglieder dieses Teams haben sich in der vierten Klasse für denselben Schwerpunkt entschlossen; nämlich für den Schwerpunkt

Systemtechnik. Sie haben gemeinsame Interesse an Softwareprogrammierung und an eingebettete Systeme. Allgemein haben sie auch andere Fähigkeiten. Aldo Sheldija hat in der Vergangenheit viele Websites erstellt, während Irena Bala viel Erfahrung mit Datenbanken hat. Auf diese Weise ergänzen sie ihre Kompetenzen gegenseitig, um das Projekt erfolgreich abzuschließen. Sie verfolgen den gleichen Zweck, um dieses Projekt optimal durchzuführen.

Die Aufgabenteilung in dieser Diplomarbeit ist folgende:

Irena Bala führt unter Zuhilfenahme eines Structed Design die Planung des Systems durch. Sie wird für die Konfiguration vom Raspberry PI Server, sowie für die Erstellung der SSL-Zertifikate und für die Einrichtung von der Datenbank verantwortlich sein. Das Design von der Datenbank, die Einrichtung des Systems und die API-Integration für Wetterdaten werden auch von ihr durchgeführt. Zusätzlich wird sie Chatbot einrichten und die Sperr-Funktion von unpassenden Bildern programmieren.

Aldo Sheldija ist zuständig für die Planung des Systems mithilfe von einem Big Picture und für die Entwicklung von dem Logo. Er wird für die Konfiguration vom Raspberry PI Client, sowie für die Erstellung des Admin-Panels und für die Erstellung von Bilder-Funktionalität beim Chatbot verantwortlich sein. Der Entwurf der Datenbank und die API-Integration für den letzten Post von der Webseite der Schule wird von ihm durchgeführt. Zusätzlich wird er das System aufbauen und testen.

1.3 Allgemeines

Damit das Projekt optimal umgesetzt werden kann, werden einige zusätzliche Funktionen berücksichtigt. Beispielsweise wird die Website mit einem SSL-Zertifikat verschlüsselt. Bei einer Unterbrechung der Netzwerkverbindung werden anstelle eines schwarzen Bildes, die zuletzt dargestellten Informationen für die Dauer dieser Unterbrechung am Bildschirm dargestellt. Die Lösungen dafür werden in den folgenden Kapiteln beschrieben.

Kapitel 2

Planung

2.1 Projektziele

2.1.1 Muss-Ziele

- Es werden Bilder auf der Webseite hochgeladen und automatisch angezeigt.
- Es werden Kalenderinformationen wie Feiertage, wichtige Termine (Olympiaden usw.) auf dem Bildschirm dargestellt.
- Der Stundenplan der Klassen wird angezeigt.
- Der aktuelle Supplierplan wird angezeigt.
- Das System soll ein flexibles Layout haben. Ein Vollbild, zwei oder mehr Panels, Größen und Platzierungen wie Informationen auf Bildschirmen werden angezeigt.
- Der tägliche Wetterbericht wird angezeigt.
- Die aktuelle Uhrzeit und das Datum werden auf dem Bildschirm dargestellt.
- Die leeren Klassen werden angezeigt.
- Verschiedene Bildschirmgrößen werden unterstützt.
- Ein SSL Zertifikat für die Client-Serververbindung wird eingerichtet.
- Es werden mehrere Anzeigen mit verschiedenen Inhalten angeboten.
- Chatbot wird auch implementiert, um die Bilder hochzuladen und sperren.
- Es werden unterschiedliche Anzeigzeiten für die unterschiedlichen GUI Bereiche unterstützt.
- Es wird ein funktionierendes System auch im offline Betrieb erreicht.

2.1.2 Optionale-Ziele

- Videos können hochgeladen und freigegeben werden. Die gängigsten Videoformate werden unterstützt.
- Das System kann auf mehrere Sprachen angeboten werden.
- Die Notfallwarnungen können dargestellt werden.

2.1.3 Nicht-Ziele

- Es wird Interaktion des Benutzers mit dem Bildschirm geben.
- Der Login auf die Webseite mit Gesichtserkennung und Office365 wird nicht möglich sein.
- Die Dateien können nicht von anderen Plattformen geholt werden.
- Es wird Audiounterstützung geben.

2.2 Projektplanung

2.2.1 Meilensteine

Die Meilensteine sind Orientierungspunkte, die am Ende der Projektplanung definiert werden. Durch diese Punkte wird der Weg vom Beginn bis zum Ende des Projekts in strukturierter Form beschrieben. Die Meilensteine sind sehr wichtig, weil sie die Weiterführung des Projekts bestimmen.

Datum	Meilenstein
04.10.2019	Implementierung der Anzeige-Struktur
18.10.2019	Initialisierung des Systems (Datenbank)
31.10.2019	Raspberry PI Konfiguration (Server, Client)
29.11.2019	Erstellung der Admin-Webseite
29.11.2019	Bilder-Verwaltung
05.12.2019	Chatbot Implementierung
20.12.2019	Verschlüsselung der Webseite durch SSL-Zertifikate

Tabelle 2.1: Meilensteine

2.2.2 Projektstrukturplan

Bei einem Projektstrukturplan ist das Projekt in Teilaufgaben und Arbeitspaketen unterteilt. Jedes Arbeitspaket gibt eine Aufgabe an, für die einer von der Gruppe verantwortlich ist. Der Projektstrukturplan ist ein sehr wichtiger Teil der Projektplanung, weil es bei der Definition der Ziele und bei der Darstellung der Arbeitspakete in strukturierter Form hilft.

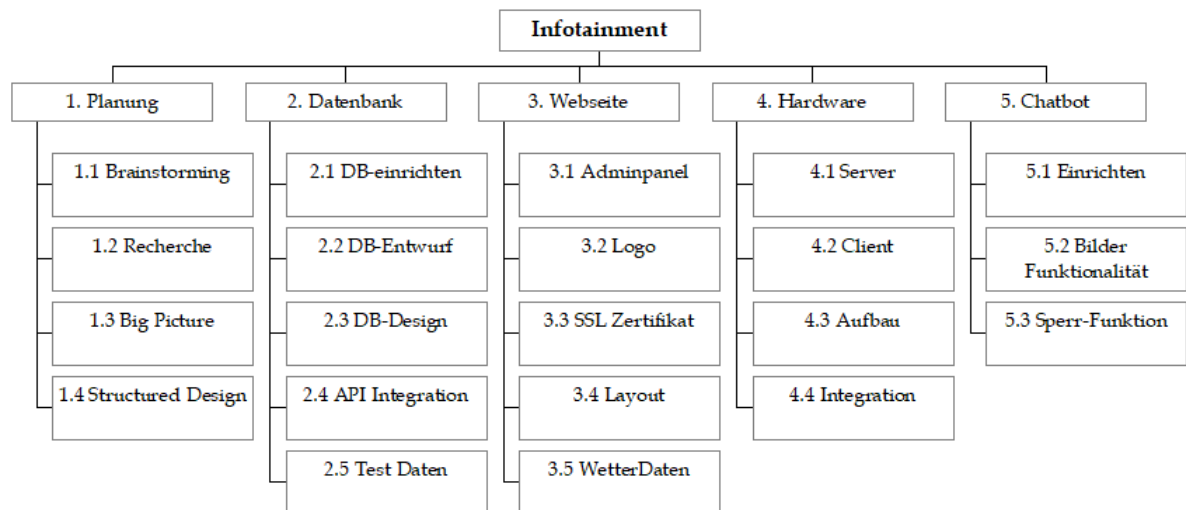


Abbildung 2.1: Projektstrukturplan

2.2.3 Arbeitspakete

Die Arbeitspakete sind die Hauptelemente eines Projekts, die ein definiertes Ergebnis, Start-und Endzeitpunkt haben. Sie werden nicht weiter unterteilt. Jedes Mitglied des Projektteams ist für einige Arbeitspakete zuständig.

Arbeitspaket AP 1.1: Brainstorming	Verantwortung: Aldo
Beginn: 16/10/2019 Ende: 22/10/2019 Mitarbeit: Aldo, Irena	Beschreibung: - Idee suchen - Idee Generierung - Idee sammeln - Idee bearbeiten
Arbeitspaket AP 1.2: Recherche	Verantwortung: Irena
Beginn: 24/10/2019 Ende: 29/10/2019 Mitarbeit: Aldo, Irena	Beschreibung: - Informationen für ähnliche Systeme suchen - Informationen finden - Informationen bearbeiten
Arbeitspaket AP 1.3: Big Picture	Verantwortung: Aldo
Beginn: 30/09/2019 Ende: 03/10/2019 Mitarbeit: Aldo, Irena	Beschreibung: - Umwelt und Rahmenbedingungen definieren - Daten bzw. Steuerungsanweisungsbeschreibungen - Iterationen - Digitalisierung
Arbeitspaket AP 1.4: Structured Design	Verantwortung: Irena

Beginn: 03/10/2019 Ende: 06/10/2019 Mitarbeit: Aldo, Irena	Beschreibung: - Festlegung der Schnittstellen und Grenzen jedes Modules - Schnittstellenbeschreibung - Funktionsbeschreibung - Modes - Iterationen - Digitalisierung
Arbeitspaket AP 2.1: DB einrichten	Verantwortung: Irena
Beginn: 07/10/2019 Ende: 13/10/2019 Mitarbeit: Irena	Beschreibung: - DB erstellen - User anlegen - Rechte vergeben
Arbeitspaket AP 2.2: DB Entwurf	Verantwortung: Aldo
Beginn: 14/10/2019 Ende: 18/10/2019 Mitarbeit: Aldo	Beschreibung: - Konzeptueller Entwurf - Implementationsentwurf - Physischer Entwurf
Arbeitspaket AP 2.3: DB-Design	Verantwortung: Irena
Beginn: 22/10/2019 Ende: 26/10/2019 Mitarbeit: Aldo, Irena	Beschreibung: Entwurf der Tabellen für: - Administrator - Bilder - Klassenplan - Lehrerplan - Supplierplan - Stundenplan - Lehrer - Wetter
Arbeitspaket AP 2.4: API Integration	Verantwortung: Aldo
Beginn: 10/12/2019 Ende: 13/12/2019 Mitarbeit: Aldo	Beschreibung: - API Dokumentation lesen - API Funktionalität anschauen - API als Schnittstelle verwenden - Daten aus vorhandenen und älteren Datenquellen durch API bekommen
Arbeitspaket AP 2.5: Test Daten	Verantwortung: Aldo
Beginn: 02/12/2019 Ende: 22/12/2019 Mitarbeit: Aldo	Beschreibung: - Dummy Daten erzeugen - Tabelle füllen - Proben durchführen
Arbeitspaket AP 3.1: Admin Panel	Verantwortung: Aldo

Beginn: 01/11/2019 Ende: 28/11/2019 Mitarbeit: Aldo, Irena	Beschreibung: - Login - Einstellungen - Gewünschte - Informationen selektieren - Informationen anzeigen lassen
Arbeitspaket AP 3.2: Logo	Verantwortung: Aldo
Beginn: 09/10/2019 Ende: 13/10/2019 Mitarbeit: Aldo	Beschreibung: - Wortmarke finden - Symbol Marke finden - Kombination (Wort und Symbol) auswählen - Entwicklung - Entwurf - Digitalisierung
Arbeitspaket AP 3.3: SSL Zertifikate	Verantwortung: Irena
Beginn: 11/12/2019 Ende: 13/12/2019 Mitarbeit: Irena	Beschreibung: - SSL Zertifikat Keys anlegen - SSL Zertifikat einrichten - HTTPS verwenden
Arbeitspaket AP 3.4: Layout	Verantwortung: Irena
Beginn: 16/12/2019 Ende: 22/12/2019 Mitarbeit: Aldo, Irena	Beschreibung: - Bootstrap - Responsives - Webdesign - Navigationsleiste - Bilder - Typographie
Arbeitspaket AP 3.5: Wetterdaten	Verantwortung: Irena
Beginn: 25/11/2019 Ende: 30/11/2019 Mitarbeit: Irena	Beschreibung: - API Integration - Format wählen (XML, JSON) - Datenbank erstellen - Daten auf die Webseite anzeigen lassen
Arbeitspaket AP 4.1: Server	Verantwortung: Irena
Beginn: 30/10/2019 Ende: 3/11/2019 Mitarbeit: Irena	Beschreibung: - Betriebssystem installieren - Programme herunterladen und installieren - Apache Server einrichten - MySQL Datenbank anlegen
Arbeitspaket AP 4.2: Client	Verantwortung: Aldo

Beginn: 21/10/2019 Ende: 2/11/2019 Mitarbeit: Aldo	Beschreibung: - Betriebssystem installieren - Programme herunterladen und installieren - Apache Server einrichten - MySQL Datenbank anlegen
Arbeitspaket AP 4.3: Aufbau	Verantwortung: Aldo
Beginn: 04/11/2019 Ende: 08/11/2019 Mitarbeit: Aldo, Irena	Beschreibung: - RaspberryPi Gehäuse - RaspberryPi am Bildschirm anbringen - Bildschirm an die Wand hängen
Arbeitspaket AP 4.4: Einrichtung des Systems	Verantwortung: Irena
Beginn: 11/11/2019 Ende: 30/11/2019 Mitarbeit: Irena	Beschreibung: - WLAN Verbindung - Verbindung mit der Datenbank
Arbeitspaket AP 5.1: Chatbot Einrichten	Verantwortung: Irena
Beginn: 18/11/2019 Ende: 24/11/2019 Mitarbeit: Irena	Beschreibung: - API Dokumentation anschauen - Account anlegen - Registrieren - Bilder, Videos posten
Arbeitspaket AP 5.2: Bilder Funktionalität	Verantwortung: Aldo
Beginn: 25/11/2019 Ende: 01/12/2019 Mitarbeit: Aldo	Beschreibung: - Bilder hochladen - Bilder in der Datenbank speichern - Bilder von der Datenbank selektieren und auf der Website anzeigen lassen
Arbeitspaket AP 5.3: Sperr-Funktion	Verantwortung: Irena
Beginn: 03/12/2019 Ende: 07/12/2019 Mitarbeit: Irena	Beschreibung: - ID des Bildes schicken - Das Telefonnummer sperren, damit er keine Bilder mehr hochladen kann - Das Bild am Bildschirm löschen - Das Bild aus der Datenbank entfernen

Tabelle 2.2: Arbeitspakete

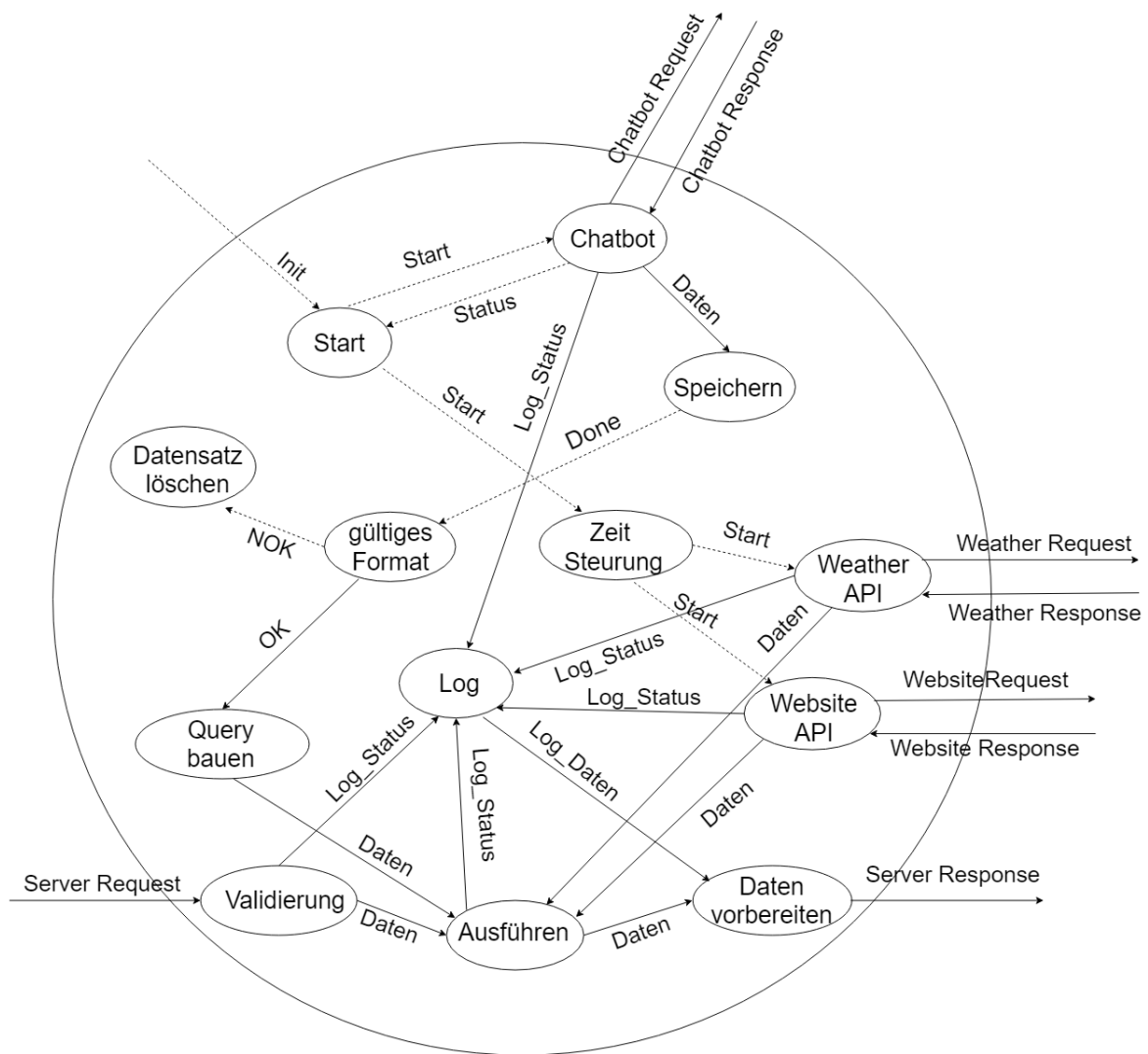


Abbildung 2.3: Structed Design - Server

dass wir eine Mischung von den traditionellen aber auch agilen Methoden brauchen. Wir müssten sehr viele Sprints machen, sehr viel testen und auch die Schritte sequenziell durchführen. Es wurde dieses Modell ausgewählt, weil für bestimmte Anforderungen wurden traditionellen Ansätze gebraucht, wobei für die anderen die agilen Ansätze gebraucht wurden. Das System ist vielfältig und komplex, deswegen war es notwendig, die Methoden für das Projektmanagement miteinander zu vermischen. Hybrid bedeutet bunt, vielfältig, komplex, gemischt. Das hybrid Modell ist modern, vielfältig und entspricht genau der Komplexität des Projekts.

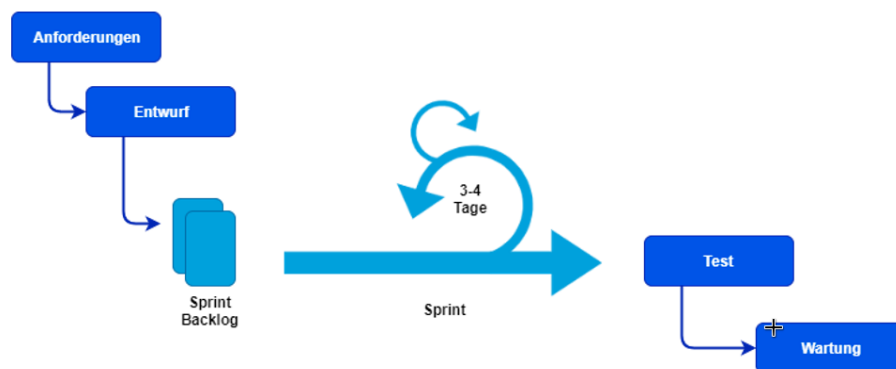


Abbildung 2.4: Hybrides Modell

Kapitel 3

Dokumentation des Projektverlaufs

3.1 Beschreibungen des Arbeitsverlaufs

3.1.1 Irena Bala

Detailed report



2019-09-01 - 2020-03-05

Total 187 h 00 min

Date	Description	Duration	User
09-09	Es wurden Recherche im Internet gemacht, um generelle Informationen über digital signatures zu sammeln. Es wurde nach Ideen über die Art der Implementierung und über die wesentlichen Funktionalitäten von solchen Systemen gesucht.	5:00:00	Irebal14
	Infotainment	13:00-18:00	
09-10	Durchführung einer Besprechung in der Gruppe über die Art der Realisierung dieser Diplomarbeit. Es wurden die Funktionalitäten dieses Systems diskutiert. Es wurden auch die zusätzlichen Eigenschaften, die dieses System anbieten sollte, überlegt.	2:00:00	Irebal14
	Infotainment	14:00-16:00	
09-10	Fertigstellung eines Structed Design, die eine sehr wichtige Methode bei der Planung eines Projekts ist. Das wurde händisch gezeichnet und nach vielen Versuchen gab es ein Endergebnis. Das Endergebnis stellte die wesentlichen Komponenten des Systems dar und die Schnittstellen dazwischen. Diese Methode gab eine übersichtlichere Darstellung des Grundkonzepts.	4:30:00	Irebal14
	Infotainment	16:30-21:00	
09-11	Die richtigen Materialien für die Diplomarbeit wurden gewählt. Dann wurde eine Hardwareliste geschrieben, damit die benötigten Materialien bestellt werden könnten.	1:30:00	Irebal14
	Infotainment	14:00-15:30	
09-11	Nachdem das Structured Design im Papier gezeichnet wurde, war es notwendig die gezeichnete Methode auch digital darzustellen. Die Digitalisierung wurde mithilfe von dem Programm Gravit Designer gemacht.	1:15:00	Irebal14
	Infotainment	16:00-17:15	
09-12	Der Diplomarbeitsantrag wurde vorbereitet. Dazu gehört die Formulierung der Ziele, Aufgabenteilung in drei verschiedenen Sprachen, die Aufgabenteilung, die Erstellung der Arbeitspaketen usw.	5:00:00	Irebal14
	Infotainment	15:00-20:00	
09-13	Es wurden die anderen Punkte der Diplomarbeit geschrieben, wie z.B die geplanten Ergebnisse der Arbeit, FMEA - Analyse, Ressourcen und Kostenplan, Balkenplan usw.	4:00:00	Irebal14
	Infotainment	12:00-16:00	
09-13	Einige Punkte im Antrag wurden gemäß den Anweisungen des Betreuers korrigiert.	1:30:00	Irebal14
	Infotainment	17:00-18:30	
09-13	Es wurden alle benötigten Informationen für die Einreichung unserer Diplomarbeit bei dem Plattform Diplomarbeits-DB ergänzt.	1:00:00	Irebal14
	Infotainment	19:00-20:00	
09-20	Zuerst wurden der XAMPP und Apache Server gestartet. Danach wurde die Startseite von XAMPP geöffnet. Dort sollte nach dem phpMyAdmin gesucht werden. Im phpMyAdmin wurde dann eine Datenbank für die Diplomarbeit erstellt.	1:00:00	Irebal14
	Infotainment	14:00-15:00	
09-22	Es wurde ein ER Diagramm am 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 zwischen den Tabellen erstellt.	3:00:00	Irebal14
	Infotainment	13:30-16:30	
09-25	Das ER-Diagramm wurde in der Gruppe besprochen. Es gab ein paar Änderungen, die dann verbessert wurden.	1:00:00	Irebal14
	Infotainment	15:00-16:00	

09-30	Die Tabellen Supplierplan, Layout, LayoutSections, Weather, Unterricht wurden in MySQLWorkbench mit den entsprechenden Spalten, Datentypen, Primär und Fremdschlüssel durch Skripts erstellt.	1:30:00	Irebal14
	Infotainment	15:00-16:30	
10-06	Zuerst wurden die Kenntnisse über GIT - Versionskontrolle wiederholt. Danach wurde ein git repository eingerichtet. Danach erfolgte die Erstellung einer Kopie des Remote-Repositorys. Im GitHub wurde ein Account erstellt. Dort wurden die Rechte des Projekts zu dem anderen Diplomarbeitmitglied gegeben.	1:15:00	Irebal14
	Infotainment	20:00-21:15	
10-06	Zuerst wurden die Kenntnisse über GIT - Versionskontrolle wiederholt. Danach wurde ein git repository eingerichtet. Danach erfolgte die Erstellung einer Kopie des Remote-Repositorys. Im GitHub wurde ein Account erstellt. Dort wurden die Rechte des Projekts zu dem anderen Diplomarbeitmitglied gegeben.	1:15:00	Irebal14
	Infotainment	20:00-21:15	
10-12	Für die Layouts wurden PHP-Skripten in Visual Studio Code erstellt. Durch CSS und Bootstrap wurden die möglichen Layouts für die Webseite vorbereitet. Die Layouts bestanden aus verschiedenen Teilbereichen, wo die Informationen dargestellt werden könnten.	2:30:00	Irebal14
	Infotainment	10:00-12:30	
10-12	An diesem Tag wurden nur die folgenden Kapiteln geschrieben: Aufgabenstellung, Idee, Thema, Allgemeines und die Planung.	4:30:00	Irebal14
	Infotainment	14:00-18:30	
10-13	An diesem Tag wurden die restlichen Kapiteln geschrieben: Projektplanung (Arbeitspakete), Team, Projektmanagementmethode und die Projektziele.	2:30:00	Irebal14
	Infotainment	14:00-16:30	
10-13	Einige Punkte wurden nach den Anweisungen des Betreuers korrigiert.	1:30:00	Irebal14
	Infotainment	17:00-18:30	
10-18	Zuerst wurden generelle Informationen über APIs gesammelt. Es wurde nach einer API gesucht, die die richtige Wetterdaten zurückliefert. Es wurde die openweathermap.org API verwendet um die Wetterdaten zu holen und auf dem Bildschirm anzeigen zu lassen. Zuerst wurde ein Account beim home.openweathermap.org erstellt. Dann wurde ein API - Schlüssel generiert und damit wird eine API Call gemacht. Die API Response wird durch JSON im passenden Format umgewandelt. Danach erfolgt die Anzeige der Daten auf die Webseite.	4:30:00	Irebal14
	Infotainment	13:00-17:30	
10-20	Es wurde eine Stored Procedure erstellt, die als Supplierer, diejenigen Personen darstellt, die nicht fehlen, die kein Unterricht haben und eine Suppliertunde haben. Diese Lehrer wurden in einer bestimmten Reihenfolge geordnet.	5:00:00	Irebal14
	Infotainment	15:00-20:00	
10-26	Es wurden Prozeduren für die restlichen Tabellen erstellt, falls sie für die Zukunft notwendig sind.	4:00:00	Irebal14
	Infotainment	11:00-15:00	
11-03	Es wurden Views für die Tabellen Stundenplan, Kalenderinformationen und Supplierplan erstellt. Die Erstellung der Views war notwendig für die Darstellung dieser Daten am Bildschirm.	2:00:00	Irebal14
	Infotainment	16:00-18:00	
11-09	Zuerst wurde das Image, welches das Betriebssystem beinhaltet, am RaspberryPI installiert. Nach dieser Grundkonfiguration wurden dann die benötigten Paketen im Server installiert. Diese Paketen sind: git, vim (Texteditor), apache2, php, mariadb-server (die Pakete für MySQL), python, telepot (die Pakete für Chatbot). Es wurde die IP-Adresse des Servers angelegt. Danach wurden die Datenbank und die Benutzer erstellt. Zu den Benutzern wurden die Rechte vergeben. Danach wurde die Konfigurationsdatei der Datenbank exportiert, und zum Server geschickt. Diese Konfigurationsdatei beinhaltete alle Datensätze, Tabellen usw. und sie wurde exportiert, damit diese Datensätze im Server gespeichert werden könnten.	2:00:00	Irebal14
	Infotainment	12:00-14:00	

11-17	Zuerst wurde ein private key und ein Zertifikat generiert, danach wurde Apache so konfiguriert, dass das SSL-Zertifikat davon unterstützt werden kann. Die SSL-Dateien wurden in Virtual Host Datei angelegt. Als letztes wurde SSL Virtual Host aktiviert und Apache neugestartet.	2:15:00	Irebal14
	Infotainment	17:00-19:15	
11-23	Es wurde eine intensive Recherche über Chatbot gemacht. Es wurden nach einer guten API gesucht. Danach wurden an Beispiele geschaut und analysiert.	2:00:00	Irebal14
	Infotainment	13:00-15:00	
11-28	Es wurde die Telegram-Bot Appi für die Chatbot Einrichtung verwendet. Zuerst wurde Telegram Applikation im Handy installiert. Danach wurde ein neues Bot erstellt und zusätzlich erhielt ich auch ein eindeutiges Token dazu. Dann wurde ein Python-Script erstellt, die für die Konfiguration vom Chatbot relevant ist. In diesem Script wurde der Import der benötigten Pakete und die Speicherung vom Token gemacht. Durch dieses Token könnte der Zugriff zu unserem Bot gemacht werden. Dieses Script sollte zuerst nur eine Message zum Telegram schicken und eine Antwort bekommen.	1:30:00	Irebal14
	Infotainment	16:00-17:30	
11-30	Es wurden die verschiedenen Funktionalitäten programmiert. Zuerst wurde überprüft ob derjenige, der zum Chatbot schreibt, der Administrator ist oder nicht. Falls ja, werden ihm verschiedene Möglichkeiten angeboten, wie z.B alle Benutzer anschauen, bestimmte Benutzer blockieren, damit sie nicht mehr Bilder zum Chatbot schicken können. Falls eine Person kein Administrator ist und diese Commands probiert, wird ihm eine Message zurückgeschickt, dass er kein Administrator ist.	2:30:00	Irebal14
	Infotainment	18:00-20:30	
12-01	Es wurde eine neue Tabelle in der Datenbank für die Benutzer von Chatbot erstellt. Zusätzlich wurde Token auch in der Datenbank gespeichert.	0:45:00	Irebal14
	Infotainment	11:30-12:15	
12-02	Es wurde überprüft, ob die Person, die zum Chatbot schreibt vom Administrator blockiert ist oder nicht. Falls ja, wird ihm eine Message zurückgeschickt, dass er geblockt worden ist. Falls nein, wird er die Möglichkeit haben, Bilder zu schicken.	2:00:00	Irebal14
	Infotainment	16:00-18:00	
12-03	Den Benutzern, die das erste Mal zum Chatbot etwas schicken, wird die Möglichkeit gegeben, sich im Chatbot zu registrieren. Es wurden zwei Buttons integriert, wo man Ja oder Nein wählen kann. Das heißt, wenn ja gedrückt wird, wird der Benutzer registriert, wenn nein nicht. Wenn man Ja drückt, wird Chatbot ihm auffordern, die Telefonnummer zu schreiben. Wenn die Telefonnummer geschickt wird, wird dieser Benutzer warten bis der Administrator seine Registrierung genehmigt hat.	7:00:00	Irebal14
	Infotainment	15:00-22:00	
12-04	Es wurden die weiteren Funktionalitäten des Administrators programmiert. Er kann die unregistrierten Benutzer sehen, diese akzeptieren oder nicht. Wenn der Administrator diese Benutzer nicht akzeptiert, wird denen eine Nachricht zurückgeschrieben, wo drinnen steht, dass sie vom Administrator nicht akzeptiert worden sind, deswegen können sie keine Bilder oder sonst etwas zum Chatbot schicken.	4:30:00	Irebal14
	Infotainment	16:00-20:30	
12-05	Chatbot wurde fertig programmiert. Es wird die Überprüfung gemacht, ob der Benutzer seine Telefonnummer richtig geschrieben hat oder nicht, und es wird überprüft ob er Text oder Zahlen geschickt hat. Das ist wichtig, weil in der Datenbank sollte ein korrektes Format gespeichert werden. Falls er die Telefonnummer richtig geschrieben hat, wird es in die Datenbank gespeichert. Ansonsten wird er die Möglichkeit haben, es noch einmal zu schreiben. Zu den Administratoren wird eine Nachricht geschickt, wo er sieht was für Möglichkeiten er hat, dasselbe wird auch zu den normalen Benutzern geschickt.	5:00:00	Irebal14
	Infotainment	16:00-21:00	
12-06	Es wurde versucht, die Kalenderinformationen so zu adaptieren, dass sie in allen Teilbereichen des Anzeigelayouts hineinpassen.	4:30:00	Irebal14
	Infotainment	17:00-21:30	
12-07	Es wurde versucht, die Wetterdaten so zu adaptieren, dass sie in allen Teilbereichen des Anzeigelayouts hineinpassen.	4:30:00	Irebal14
	Infotainment	16:00-20:30	

12-08	Es wurde versucht, den Stundepplan in bestimmten Teilbereichen des Anzeigelayouts zu adaptieren. Wenn es viele Datensätze gibt, wurde diese Tabellen in zwei Tabellen geteilt und der Administrator wählt selbst, welche Tabelle er anzeigen will.	6:00:00	Irebal14
	Infotainment	13:00-19:00	
12-09	Es wurde versucht, den Supplierplan in bestimmten Teilbereichen des Anzeigelayouts zu adaptieren. Wenn es viele Datensätze gibt, wurde diese Tabellen in zwei Tabellen geteilt und der Administrator wählt selbst, welche Tabelle er anzeigen will.	3:30:00	Irebal14
	Infotainment	14:00-17:30	
01-09	Es wurde die Anzeige der Kalenderinformationen vorbereitet. Diese Informationen werden in Tabellen angezeigt und der Administrator kann selbst entscheiden, wie lange diese Informationen angezeigt werden sollen.	2:30:00	Irebal14
	Infotainment	16:00-18:30	
01-22	Es wurden eine View aus der Tabelle Unterricht und danach auch eine gespeicherte Prozedur erstellt, damit nur die leeren Klassen angezeigt wurden.	2:00:00	Irebal14
	Infotainment	18:30-20:30	
01-23	Es wurde die Anzeige der leeren Klassen vorbereitet. Die leere Klassen werden nicht in einer Tabelle angezeigt, weil es kann sein, dass die Datenmenge gross ist. Sie werden aber zeilenweise dargestellt.	2:30:00	Irebal14
	Infotainment	15:30-18:00	
01-31	Korrektur der Diplomarbeitsdokumentation	2:30:00	Irebal14
	Infotainment	16:00-18:30	
01-31	Es wurde die Anzeige von dem Stundenplan vorbereitet. Die Daten sollten in einer Tabelle dargestellt, wobei jeder Klasse in 2 Gruppen geteilt werden sollte. Es wurden die Lehrer, Fach und Raum fuer jede Klasse und Gruppe in dieser Tabelle angezeigt.	3:30:00	Irebal14
	Infotainment	19:00-22:30	
02-05	Weiter mit der Vorbereitung der Anzeige für den Stundenplan.	3:00:00	Irebal14
	Infotainment	19:00-22:00	
02-08	Es wurde darüber nachgedacht, wie der Chatbot so implementiert wird, dass er in verschiedenen Sprachen angeboten wird	2:15:00	Irebal14
	Infotainment	08:00-10:15	
02-09	Es wurde eine Tabelle in der Datenbank erstellt, wo alle Nachrichten, die vom Chatbot gesendet werden, dort in zwei verschiedenen Sprachen gespeichert wurden.	2:00:00	Irebal14
	Infotainment	09:15-11:15	
02-15	Implementierung des Chatbots in zwei Sprachen(Englisch, Deutsch). Es wurde in Python programmiert.	4:15:00	Irebal14
	Infotainment	09:30-13:45	
02-16	Weitere Implementierung des Chatbots in zwei Sprachen.	2:45:00	Irebal14
	Infotainment	20:45-23:30	
02-18	Chatbot testen	2:00:00	Irebal14
	Infotainment	16:00-18:00	
02-22	Implementierung der Notfallwarnungen beim Chatbot. Chatbot testen.	2:30:00	Irebal14
	Infotainment	20:00-22:30	
02-23	Optimierung der Funktionalitäten beim Chatbot. Korrigieren von Fällen, in denen der Chatbot dem Benutzer nicht die richtigen Nachrichten sendete hat. Erstellung einer Menü mit Hauptfunktionen für den Administrator, normale Benutzer und unregistrierte Benutzer.	3:30:00	Irebal14
	Infotainment	08:00-11:30	

02-23	Recherche über Datensynchronisation zwischen Client und Server für die Implementierung von Offline Betrieb.	2:30:00	Irebal14
	Infotainment	16:30-19:00	
02-24	Diskussion mit Lehrern darüber, wie eine Datensynchronisation zwischen einem Server und einem Client manuell implementiert werden kann.	1:30:00	Irebal14
	Infotainment	15:00-16:30	
02-25	Es wurden auf Papier Beispiele gezeichnet, wie dann die manuelle Datensynchronisation programmiert werden konnte.	2:00:00	Irebal14
	Infotainment	17:30-19:30	
02-25	Offline Betrieb Implementierung. Zuerst wurde eine Grundstruktur programmiert, es wurde eine Verbindung mit dem Server und mit den Clients aufgebaut. Es wurden danach alle Daten davon selektiert und miteinander verglichen.	4:00:00	Irebal14
	Infotainment	20:30-00:30	
02-26	Offline Betrieb Implementierung. Es wurden die Unterschiede zwischen Client und Server mithilfe von PHP Funktionen genommen. Was im Server war und in Client noch nicht, wurde auch im Client hinzugefügt, damit sie den gleichen Zustand bekommen.	8:00:00	Irebal14
	Infotainment	16:00-00:00	
02-27	Offline Betrieb Implementierung. Korrektur der Fehler. Es wurde viel getestet.	8:30:00	Irebal14
	Infotainment	15:00-23:30	
02-28	Besprechung in der Gruppe über die bisherigen erreichten Ergebnisse.	1:30:00	Irebal14
	Infotainment	13:30-15:00	
02-29	Es wurde versucht, den Fehler bei der Synchronisation von den Bilder von Server zum Client zu korrigieren. Die Bilder wurden in blob Datentyp gespeichert, daher gab es Probleme bei der Datenuebertragung.	4:00:00	Irebal14
	Infotainment	09:30-13:30	
02-29	Diplomarbeitsdokumentation. Es wurden die fehlenden Kapiteln geschrieben.	6:00:00	Irebal14
	Infotainment	14:30-20:30	
03-01	Weiter mit der Diplomarbeitsdokumentation	4:30:00	Irebal14
	Infotainment	16:30-21:00	

Created with toggl.com

3.1.2 Aldo Sheldija

Detailed report



2019-09-10 - 2019-12-08

Total 111 h 20 min

Date	Description	Duration	User
09-10	Recherche für andere ähnliche Projekte, um die Zeile unseres Projekts zu definieren	1:35:00	Aldshe14
	Infotainment - [Dokumentation]	19:25-21:00	
09-11	#Dokumentation - DA-Antrag ausfüllen	1:00:00	Aldshe14
	Infotainment - [Dokumentation]	17:15-18:15	
09-12	#Dokumentation - DA-Antrag verbessern	1:30:00	Aldshe14
	Infotainment - [Dokumentation]	18:00-19:30	
09-16	#AdminPanel - Recherche für die Features der AdminPanel	1:00:00	Aldshe14
	Infotainment - [AdminPanel]	17:30-18:30	
09-19	#AdminPanel - Recherche für die Darstellung von Informationen in verschiedene GUI Bereiche	1:40:00	Aldshe14
	Infotainment - [AdminPanel]	18:00-19:40	
09-23	#Logo - Erste Konzept	3:00:00	Aldshe14
	Infotainment - [Logo]	18:30-21:30	
09-25	#Logo - Verbesserung der Logo	1:30:00	Aldshe14
	Infotainment - [Logo]	17:30-19:00	
09-26	#Logo - Digitalisierung des Logo	2:30:00	Aldshe14
	Infotainment - [Logo]	20:30-23:00	
09-27	#AdminPanel - Grunstruktur der PHP Dateien	4:30:00	Aldshe14
	Infotainment - [AdminPanel]	18:00-22:30	
09-28	#Display - Layout 1 / 5 Sections Layout mit Kopf und Fußzeile	1:30:00	Aldshe14
	Infotainment - [Display]	10:20-11:50	
09-28	#Display - Layout 2 / 3 Sections Layout mit Kopf und Fußzeile	1:15:00	Aldshe14
	Infotainment - [Display]	11:20-12:35	
09-29	#Client - Raspbian Image auf SD-Karte spielen und das System konfigurieren	5:25:00	Aldshe14
	Infotainment - [Client]	15:00-20:25	
10-02	#Datenbank - ERD Diagramm	3:00:00	Aldshe14
	Infotainment - [Datenbank]	17:30-20:30	
10-03	#Datenbank - Tabellen erstellen	2:45:00	Aldshe14
	Infotainment - [Datenbank]	18:15-21:00	
10-05	#AdminPanel - Datatables Plug-in für jQuery konfiguriert	6:00:00	Aldshe14
	Infotainment - [AdminPanel]	13:00-19:00	
10-08	#Client - Apache, MariaDB Server, PHP, Git, VIM Installiert am System	2:00:00	Aldshe14
	Infotainment - [Client]	16:00-18:00	

10-10	#AdminPanel - Fontawesome Icons integriert im Offline Betrieb	1:00:00	Aldshe14
	Infotainment - [AdminPanel]	18:00-19:00	
10-11	#AdminPanel - Login Bereich	6:30:00	Aldshe14
	Infotainment - [AdminPanel]	14:30-21:00	
10-12	#AdminPanel - Die Verbindung wurde mit der neuen Datenbank aktualisiert, die auf dem Server in der Schule gehostet wird	0:30:00	Aldshe14
	Infotainment - [AdminPanel]	16:30-17:00	
10-12	#Datenbank - Export gemacht und in Git Repository gespeichert, um diese Datenbank in der neuen Server zu importieren	1:00:00	Aldshe14
	Infotainment - [AdminPanel]	17:00-18:00	
10-16	#Display - Datum und Uhrzeit mit jQuery darstellen auf Deutsch	1:00:00	Aldshe14
	Infotainment - [Display]	19:00-20:00	
10-18	#AdminPanel - Admin und Einfacher Benutzer unterschieden	2:00:00	Aldshe14
	Infotainment - [AdminPanel]	11:00-13:00	
10-19	#AdminPanel - Passwort zurücksetzen	4:45:00	Aldshe14
	Infotainment - [AdminPanel]	17:45-22:30	
10-22	#AdminPanel - Stundplan für eine bestimmte Tag anzeigen mit Hilfe von einer Dropdown	3:00:00	Aldshe14
	Infotainment - [AdminPanel]	19:00-22:00	
10-24	#AdminPanel - Supplierlehrer bei der Supplierplan zeigen, die die Kriterien erfüllen	1:00:00	Aldshe14
	Infotainment - [AdminPanel]	18:30-19:30	
10-25	#AdminPanel - Supplierplan für ein Lehrer machen und bestimmte Stunden	4:30:00	Aldshe14
	Infotainment - [AdminPanel]	18:00-22:30	
10-27	#AdminPanel - Neue User anlegen mit unterschiedliche Rechte und löschen	6:00:00	Aldshe14
	Infotainment - [AdminPanel]	16:00-22:00	
10-30	#AdminPanel - Fehlendelehrer selektieren aus der Stundenplantabelle (Editieren und Löschen)	2:20:00	Aldshe14
	Infotainment - [AdminPanel]	17:40-20:00	
11-02	#Display - Welcome Page hinzugefügt, wenn der Player noch nicht aktiviert wurde	5:30:00	Aldshe14
	Infotainment - [Display]	14:30-20:00	
11-03	#AdminPanel - Supplierplan ändern und löschen	3:30:00	Aldshe14
	Infotainment - [AdminPanel]	19:00-22:30	
11-06	#AdminPanel - Logout Funktionalität	2:00:00	Aldshe14
	Infotainment - [AdminPanel]	18:40-20:40	
11-07	#AdminPanel - Passwort zurücksetzen beim Login Updaten	1:30:00	Aldshe14
	Infotainment - [AdminPanel]	13:30-15:00	
11-09	#Display - Letzte Post der Webseite der Schule anzeigen	4:45:00	Aldshe14
	Infotainment - [Display]	14:30-19:15	
11-11	#Client - Chromium Installieren für die Anzeige von Informationen	1:00:00	Aldshe14
	Infotainment - [Client]	16:45-17:45	

11-13	#AdminPanel - Kopfzeile der Tabelle Fett darstellen Updated	1:00:00	Aldshe14
	Infotainment - [AdminPanel]	15:00-16:00	
11-15	#Client - Webseite beim Hochfahren öffnen	2:00:00	Aldshe14
	Infotainment - [Client]	15:47-17:47	
11-16	#AdminPanel - Stundenplan importieren und alte Daten löschen	2:00:00	Aldshe14
	Infotainment - [AdminPanel]	18:10-20:10	
11-21	#Datenbank - Automatische ausführung von PHP Scripts mit hilfe von Cronjobs um die neue Posts von der Webseite zu bekommen	1:30:00	Aldshe14
	Infotainment - [Datenbank]	17:50-19:20	
11-26	#Display - Logo im Header platzieren und darstellen	1:00:00	Aldshe14
	Infotainment - [Display]	20:00-21:00	
11-29	#AdminPanel - Bilder Hochladen	2:00:00	Aldshe14
	Infotainment - [AdminPanel]	18:45-20:45	
11-30	#Display - Anzeige von Bildern	2:00:00	Aldshe14
	Infotainment - [Display]	16:12-18:12	
12-01	#AdminPanel - Settings Seite erstellt um die API-Key, Telegram token und URL der Webseite zu editieren	3:00:00	Aldshe14
	Infotainment - [AdminPanel]	17:05-20:05	
12-04	#AdminPanel - Anzeige hinzufügen, editieren und löschen	2:00:00	Aldshe14
	Infotainment - [AdminPanel]	18:00-20:00	
12-07	#AdminPanel - Kalendarinformationen hinzufügen	2:50:00	Aldshe14
	Infotainment - [AdminPanel]	09:52-12:42	

Created with toggl.com

Kapitel 4

Umsetzung - Irena Bala

4.1 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.

4.1.1 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, dass 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 1.9.5 gelesen werden.

4.1.2 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.

4.1.3 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 4.1: Technologien

Was ist Apache HTTP Server?

Der Apache HTTP¹ 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

¹Hypertext Transfer Protocol

SSL ²Zertifikats. 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 ³ 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 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 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?

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

Was ist SSL?

SSL steht für Secure Socket Layer und ist für die verschlüsselte Übertragung der Daten vom Browser zum Server verantwortlich. Die Verbindung zwischen dem Server und dem Browser erfolgt durch das HTTPS-Protokoll. Das ist ein Kommunikationsprotokoll, das eine verschlüsselte Datenübertragung ermöglicht. Heutzutage wird TLS¹¹ am meisten verwendet, 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 CA¹² und erfordert einige Daten von dem Antragsteller, die für die Erstellung des Zertifikats notwendig sind. Als nächstes, erzeugt der Antragsteller für die Entschlüsselung und Verschlüsselung der 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 API¹³ 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]

⁸Video Graphics Array

⁹High-Definition Multimedia Interface

¹⁰Secure Digital Memory Card

¹¹Transport Layer Security

¹²Certificate Authority

¹³Application Programming Interface

Was ist JSON?

„JSON¹⁴ bietet einen einfachen Standard für die strukturierte Kodierung von Daten in Form von menschenlesbarem Text. Dies bietet Vorteile bei einer automatisierten Weiterverarbeitung, macht sie aber auch einer manuellen Inspektion und Überarbeitung besser zugänglich.“ [8]

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

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

Tabelle 4.2: Bereiche und Technologien

4.2 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.

¹⁴JavaScript Object Notation

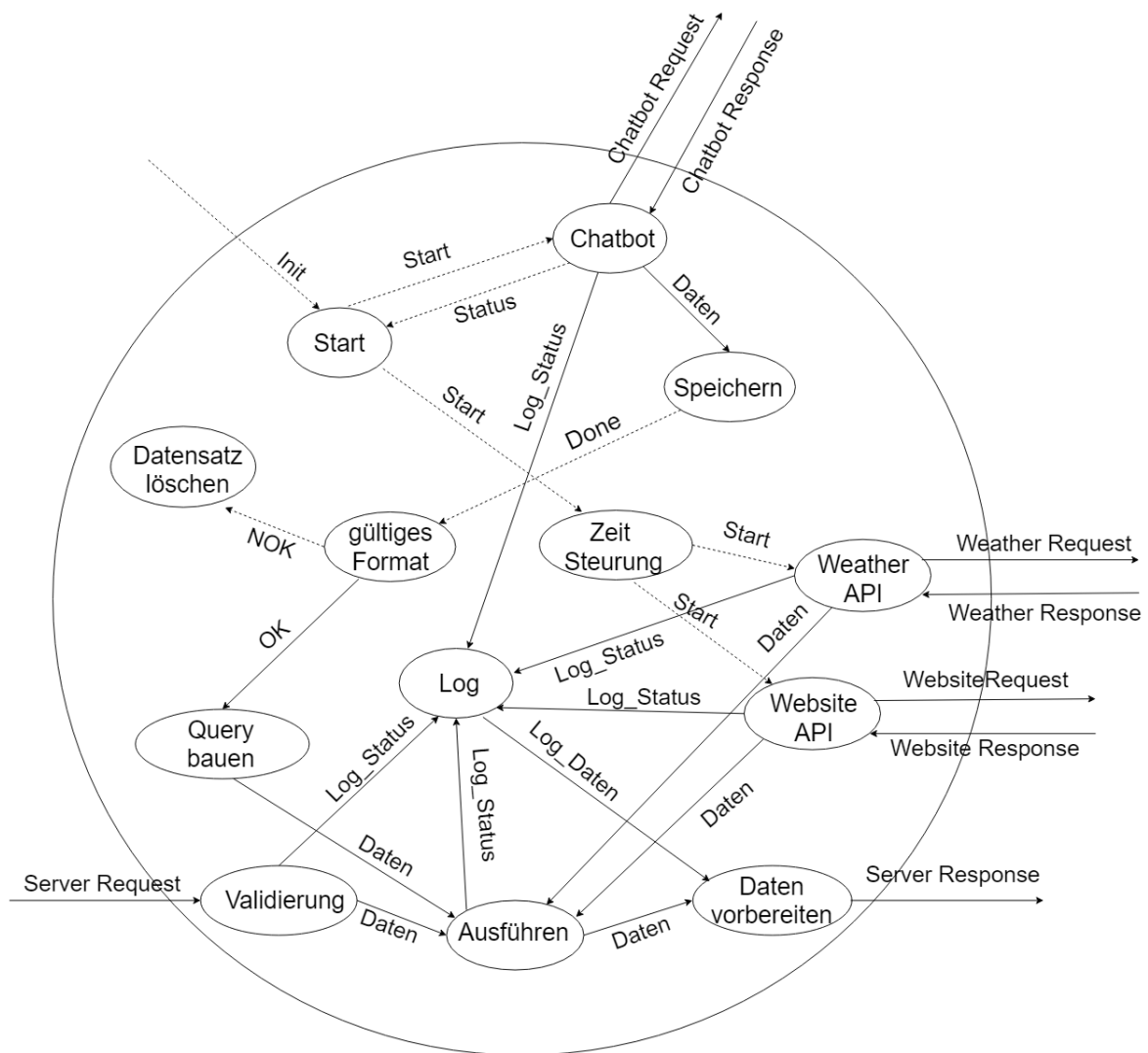


Abbildung 4.1: Structured Software Design vom Server

4.3 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 SSH¹⁵ aktiviert, damit eine sichere Verbindung zu diesem Server von einem externen Gerät ermöglicht werden kann.

4.4 Datenbank

- Zuerst wurde ein ER¹⁶ 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.

¹⁵Secure Shell

¹⁶Entity Relationship

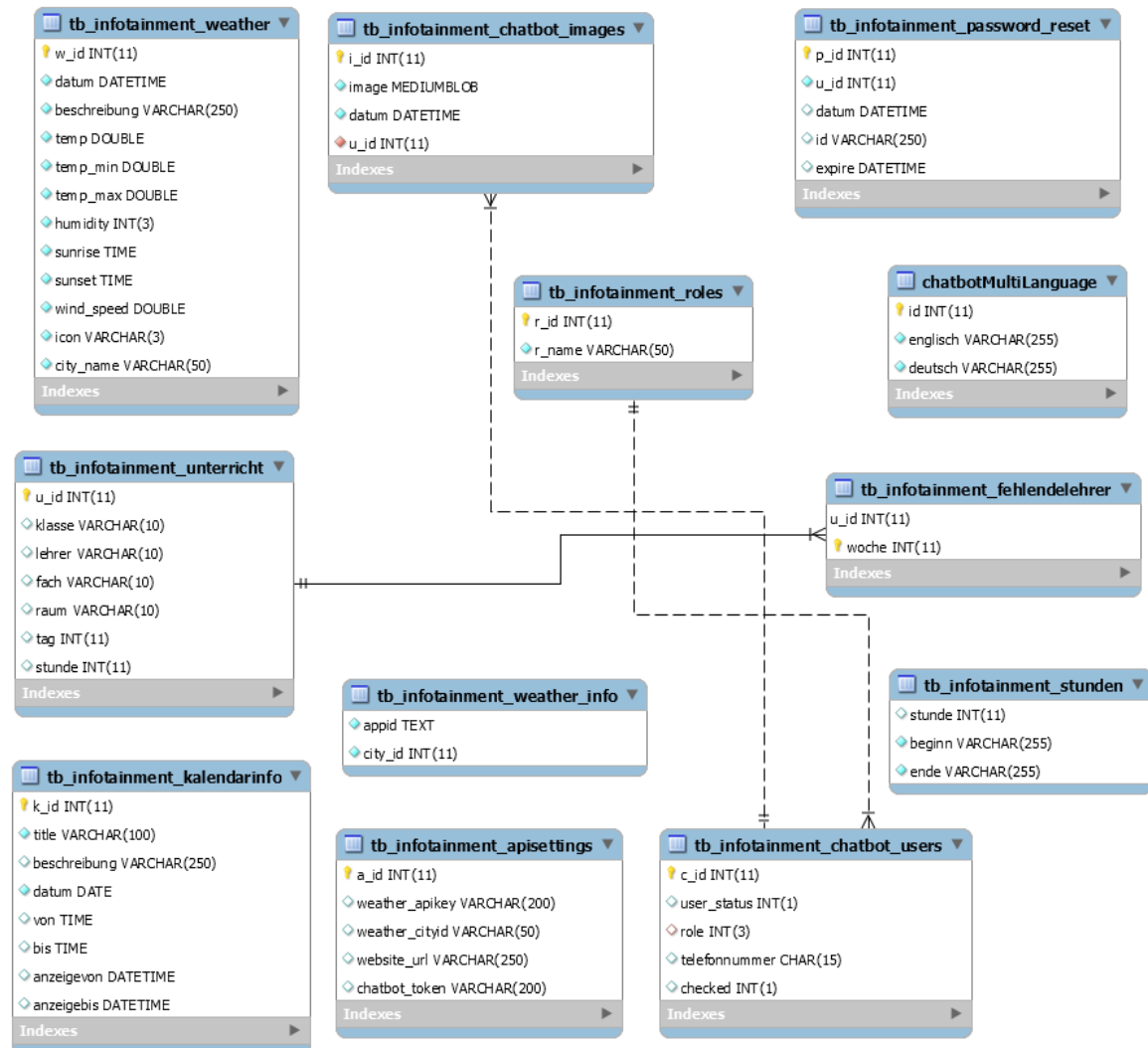


Abbildung 4.2: ERD

4.5 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]

4.6 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.

Die Tabellen Kalenderinformationen und Stundenplan wurden aus der Datenbank selektiert und dann mit PHP dargestellt. Die Kalenderinformationen sind auf die Abbildung 1.3 ersichtlich. Die leere Klassen sind durch die Erstellung von einer View für die Tabelle Unterricht dargestellt. Die leere Klassen sind auf die Abbildung 1.4 ersichtlich. Für den Supplierplan wurde eine gespeicherte Prozedur erstellt. Auf die Abbildung 1.4 ist die Darstellung der Wetterdaten auf dem Bildschirm 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 4.3: Kalenderinformationen

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

Abbildung 4.4: Leere Klassen für Montag, die zweite Stunde

4.7 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 Stromausfall 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 Ereigniss 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 4.1: 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 4.2: 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.

4.8 Wetterdaten

- Als erster Schritt erfolgte die Registrierung bei openweathermap.org.

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

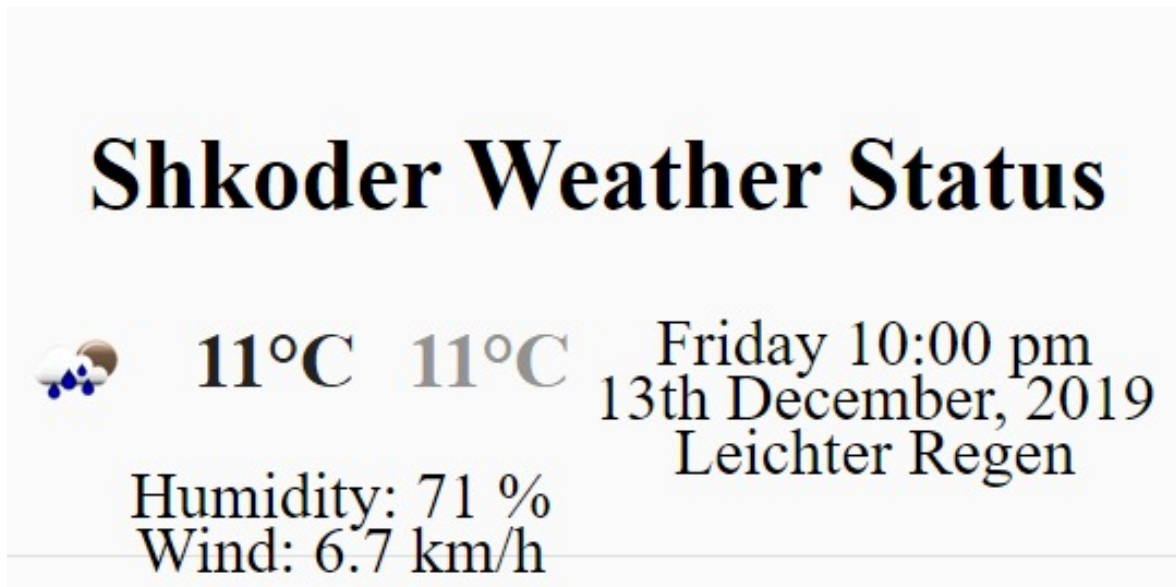


Abbildung 4.5: Wetterdaten

4.9 Chatbot

In diesem Unterkapitel wird der Chatbot und dessen Umsetzung erklärt.

4.9.1 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.

4.9.2 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]

4.9.3 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.

4.9.4 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.

4.9.5 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 4.3: 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 4.4: 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 4.5: 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 4.6: 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 4.7: 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 4.8: 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 4.9: 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 4.10: 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 4.11: 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 4.12: 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 4.13: 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 4.6: Chatbot Multilanguage

4.9.6 Konkrete Beispiele mit Bildern

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

- Hauptmenü für den Administrator

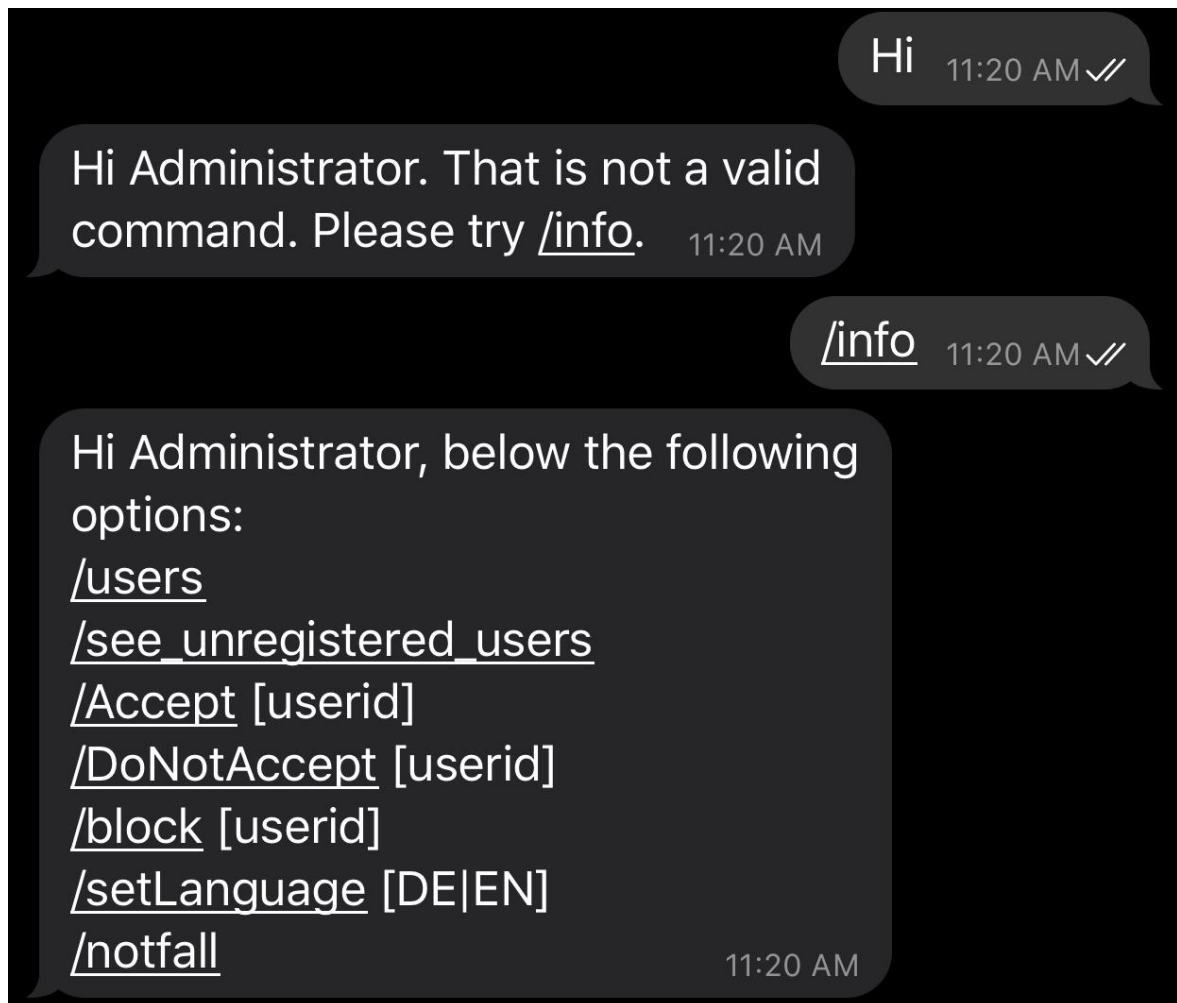


Abbildung 4.7: Hauptmenü für den Administrator

- 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.

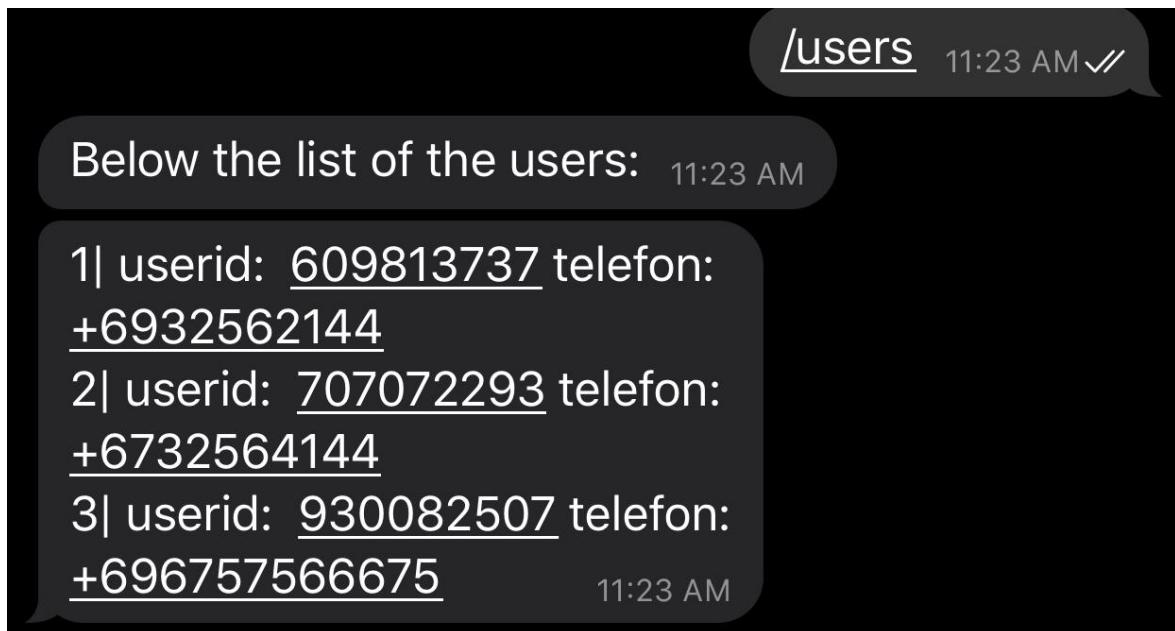


Abbildung 4.8: Auflistung der Chatbot-Benutzer

- 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.

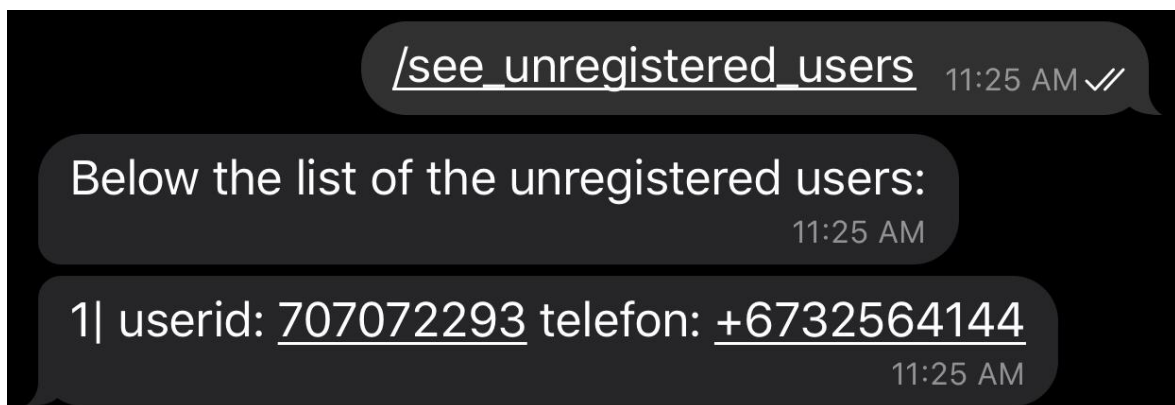


Abbildung 4.9: Auflistung der unregistrierten Benutzer

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

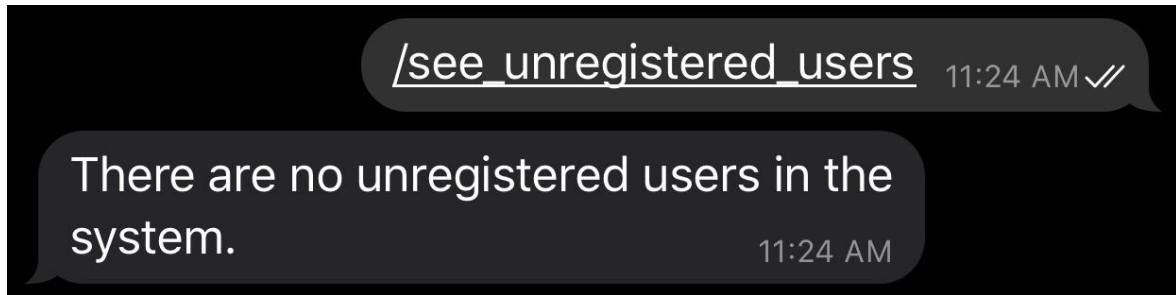


Abbildung 4.10: Keine Unregistrierte Benutzer

- 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.

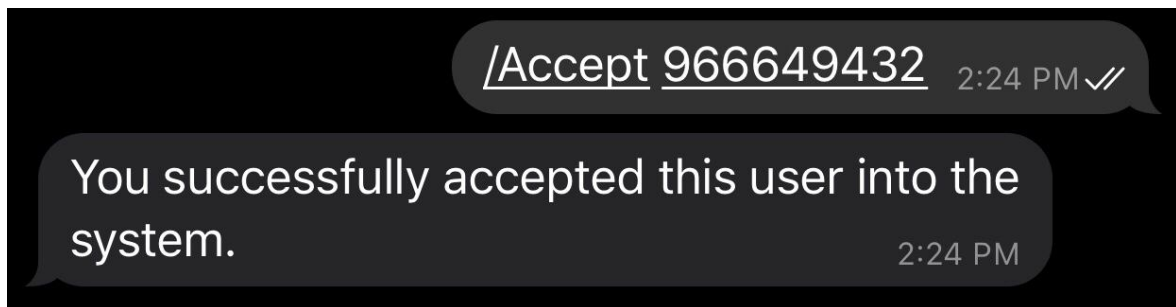


Abbildung 4.11: Genehmigung des Administrators

- 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.

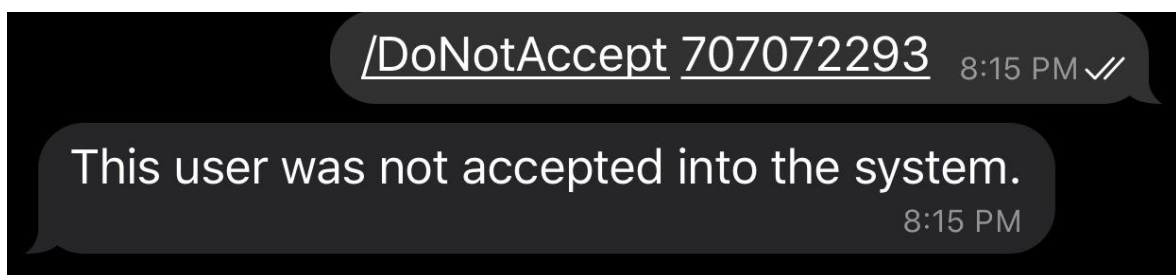


Abbildung 4.12: Ablehnung des Administrators

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

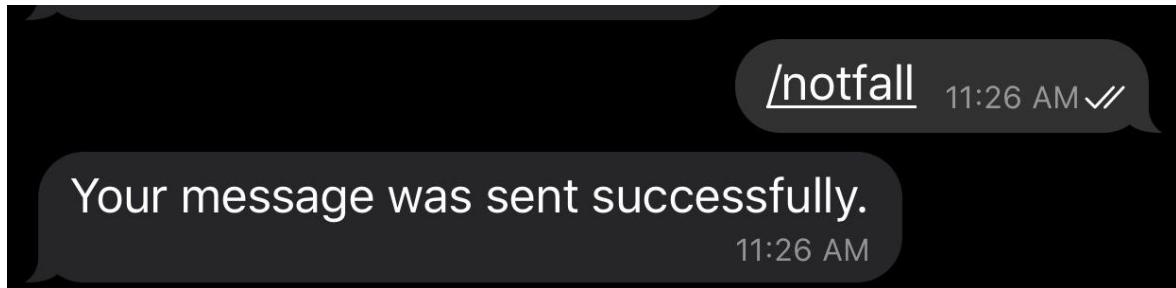


Abbildung 4.13: Notfallwarnmeldung

- 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.

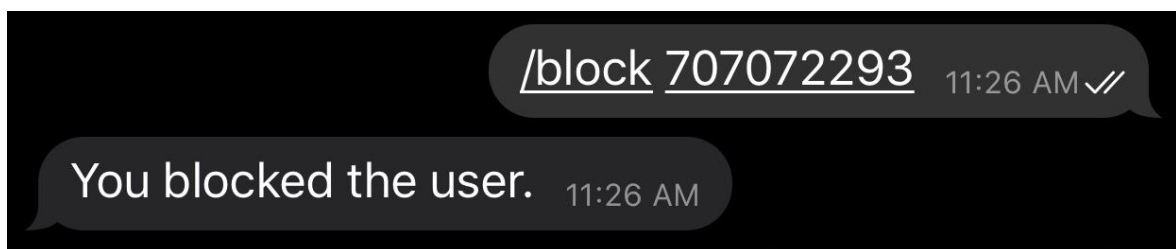


Abbildung 4.14: Benutzer Blockierung

- 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.

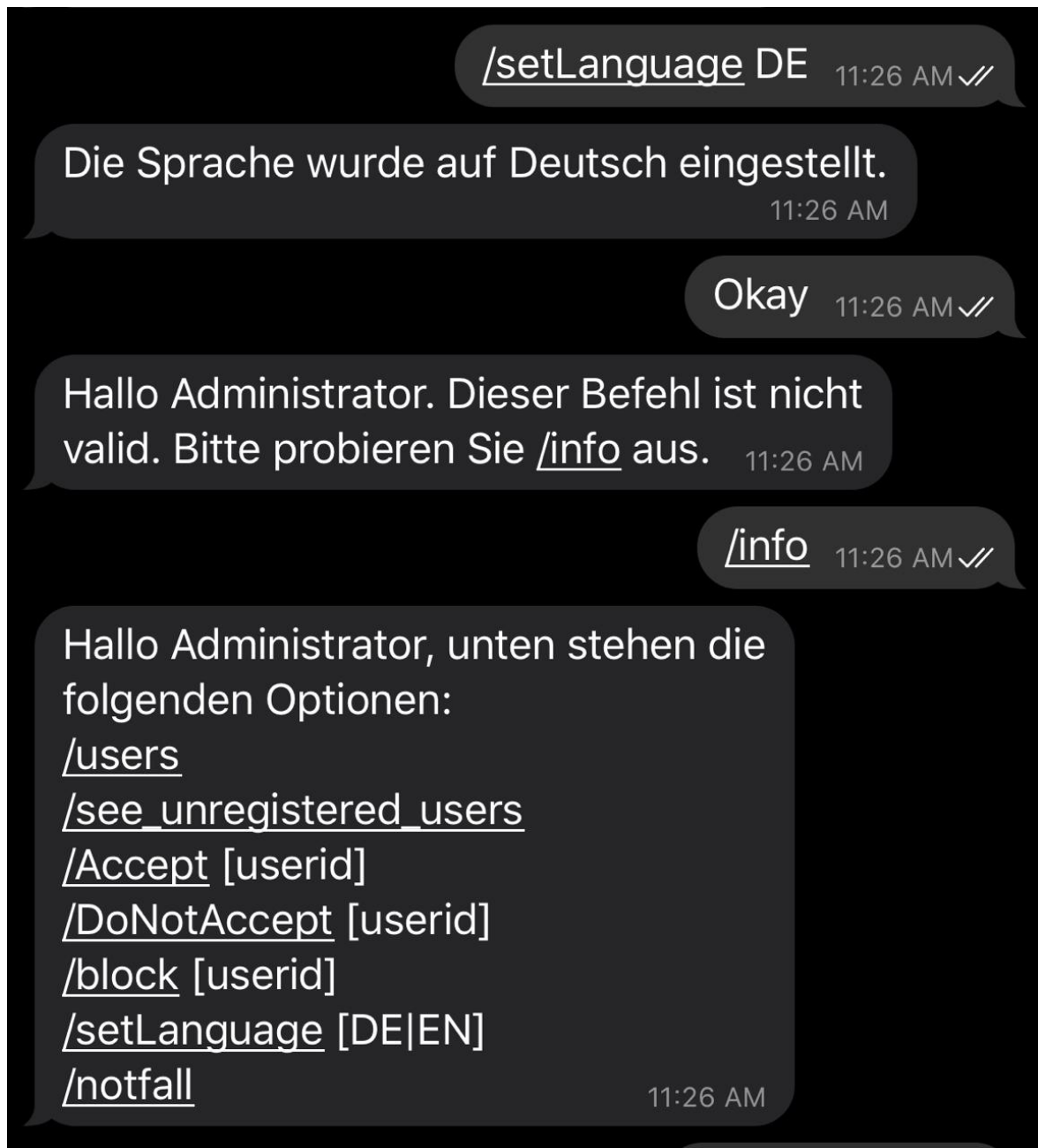


Abbildung 4.15: Spracheinstellungen

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

- Hauptmenü für den unregistrierten Benutzer

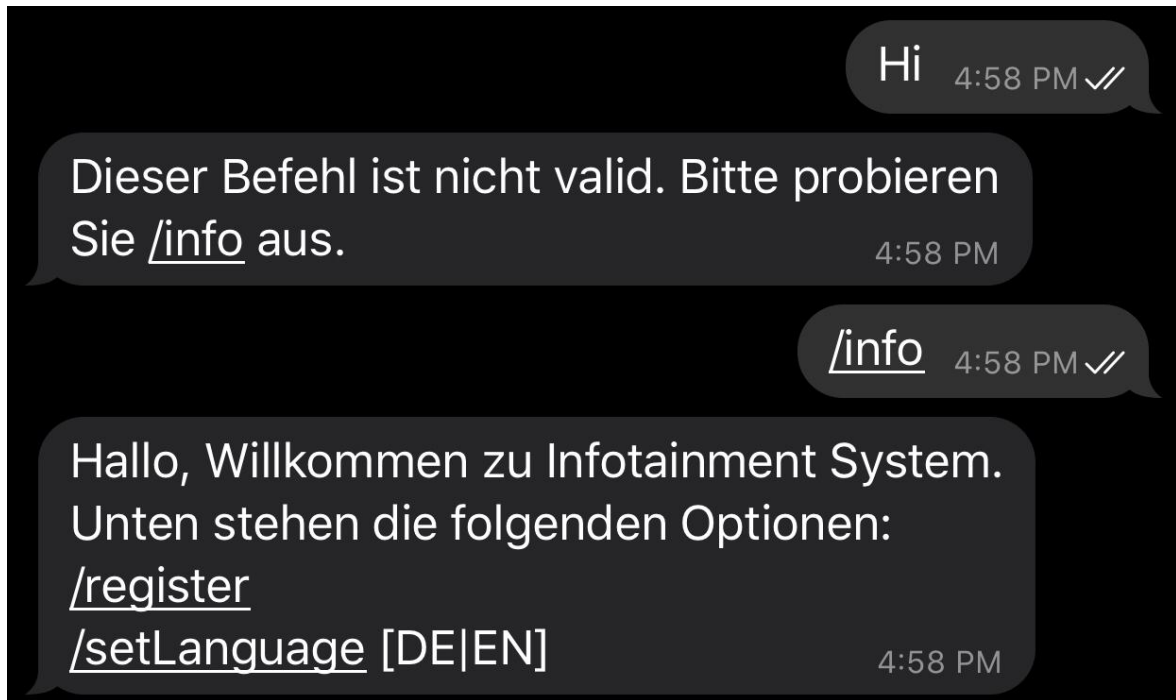


Abbildung 4.16: Hauptmenü für den unregistrierten Benutzer

- 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.

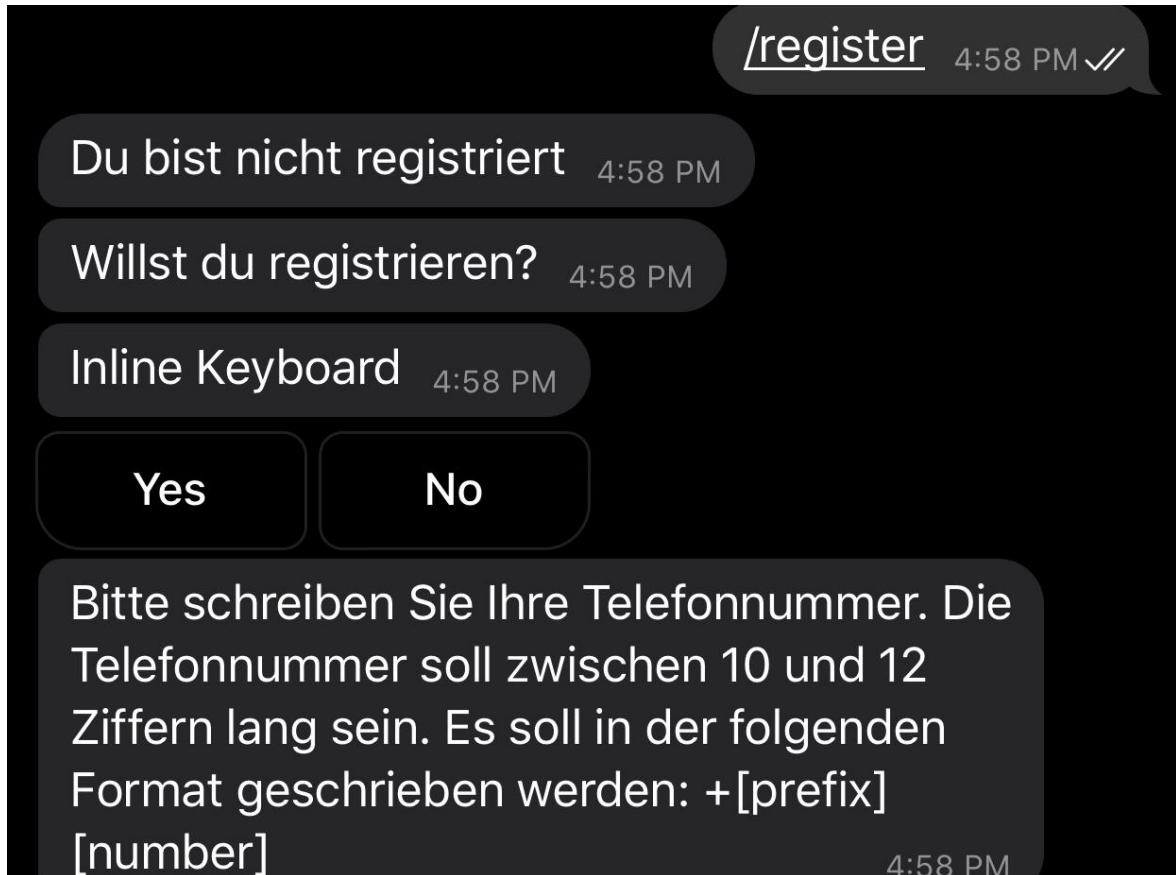


Abbildung 4.17: Registrierung beim Chatbot

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

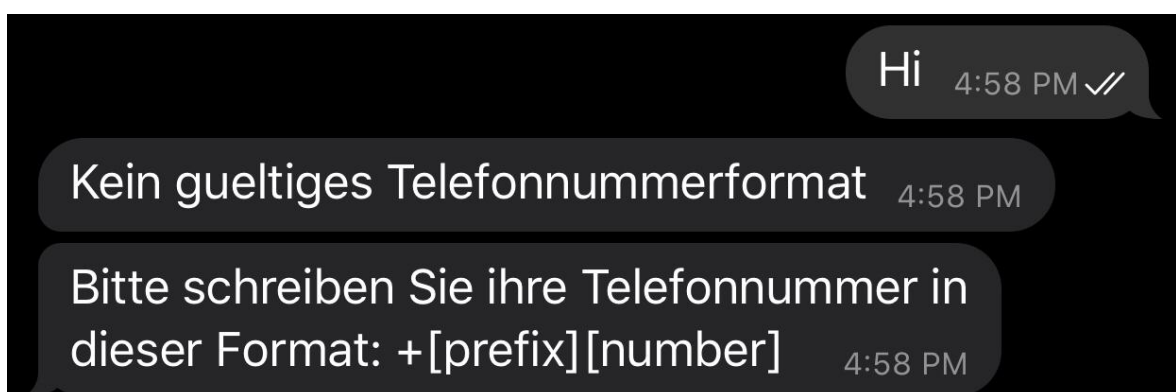


Abbildung 4.18: Falsche Eingabe des Telefonnummers

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

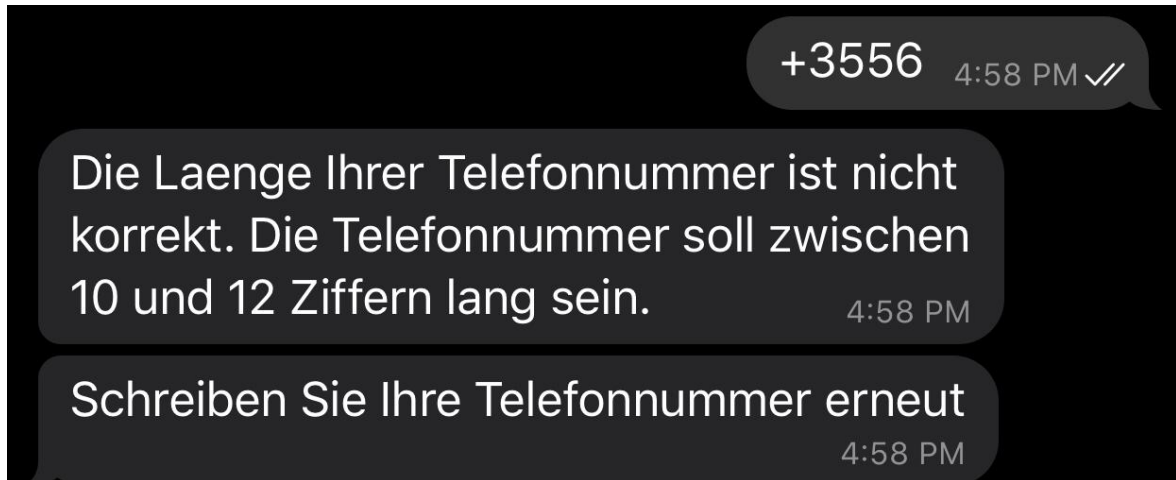


Abbildung 4.19: Telefonnummer zu kurz

- 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.

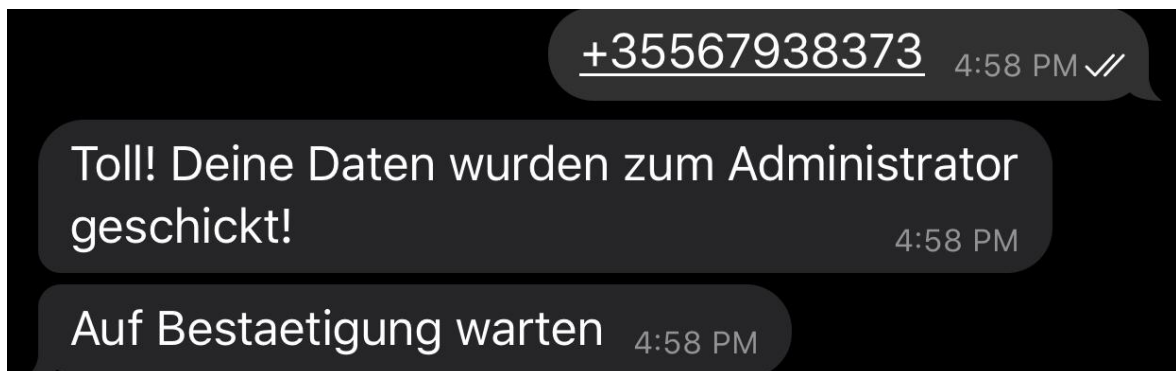


Abbildung 4.20: Korrekte Eingabe des Telefonnummers

- 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.

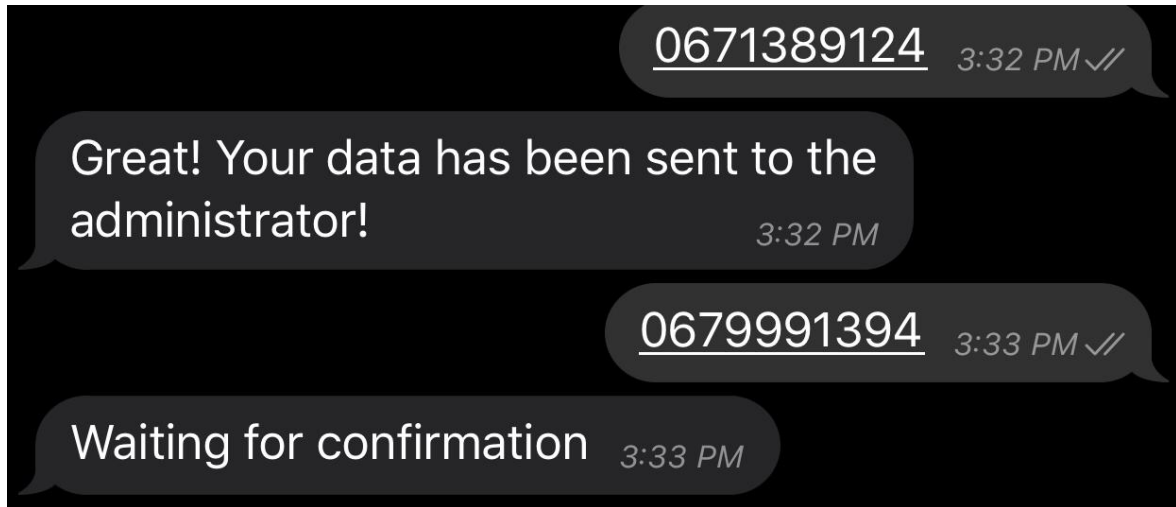


Abbildung 4.21: Das Warten der Benutzer auf die Administrator Bestätigung

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

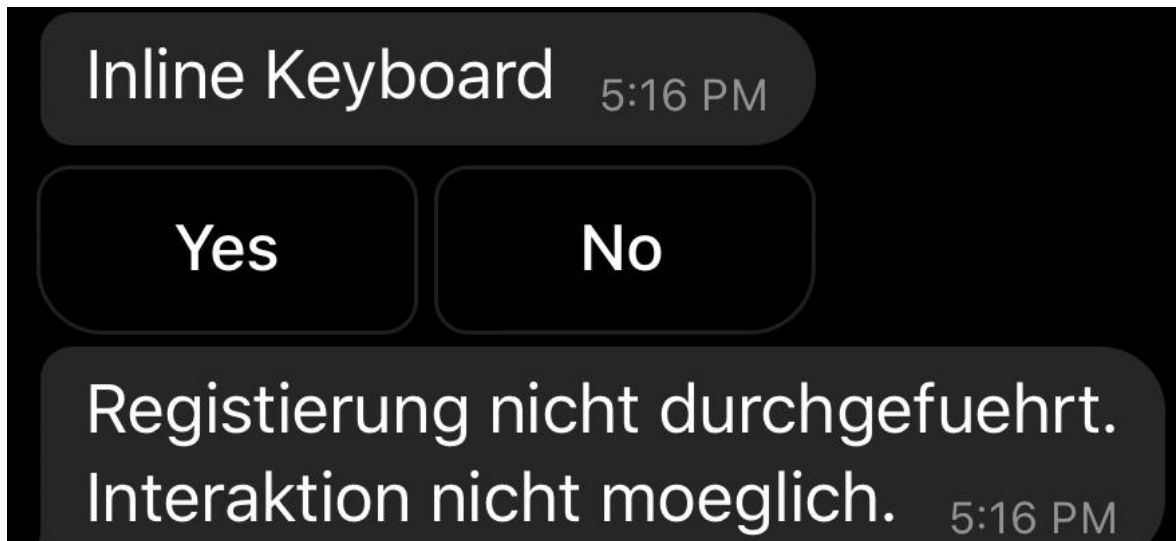


Abbildung 4.22: Keine Registrierung

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

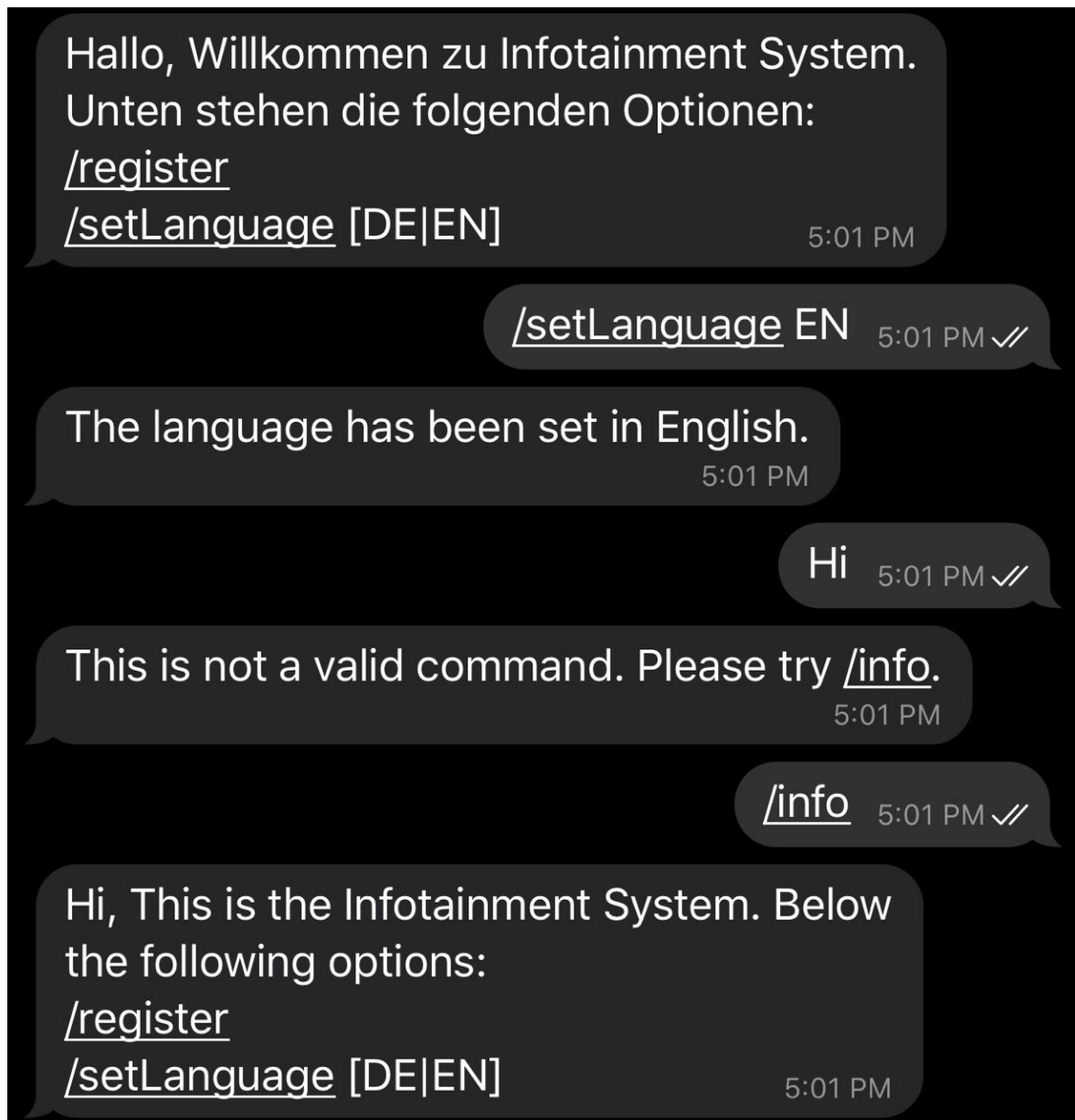


Abbildung 4.23: Spracheinstellungen für unregistrierte Benutzer

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

- Hauptmenür für einen normalen Benutzer

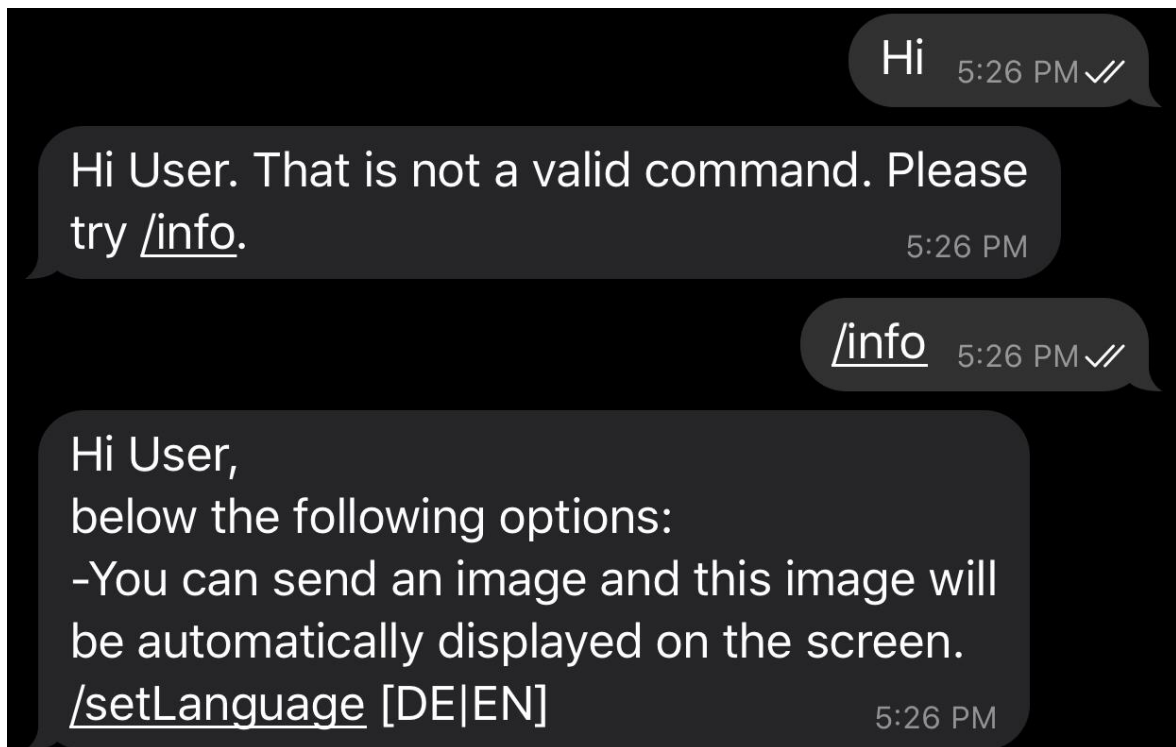


Abbildung 4.24: Hauptmenür für einen normalen Benutzer

- Bilderfunktionalität

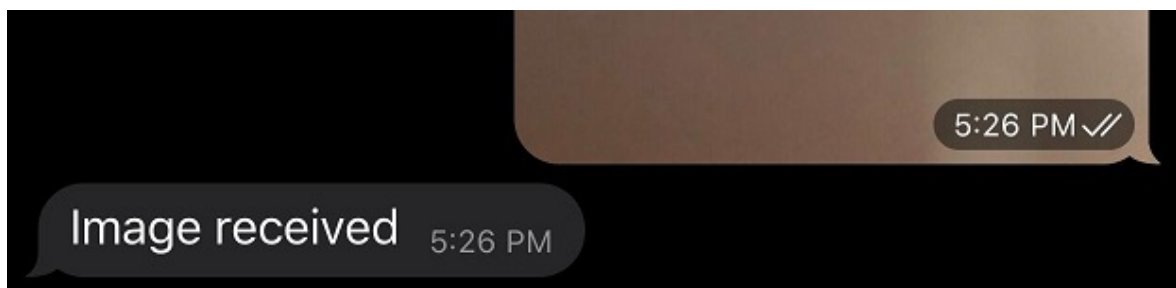


Abbildung 4.25: Bilderfunktionalität

- Spracheinstellungen für einen normalen Benutzer

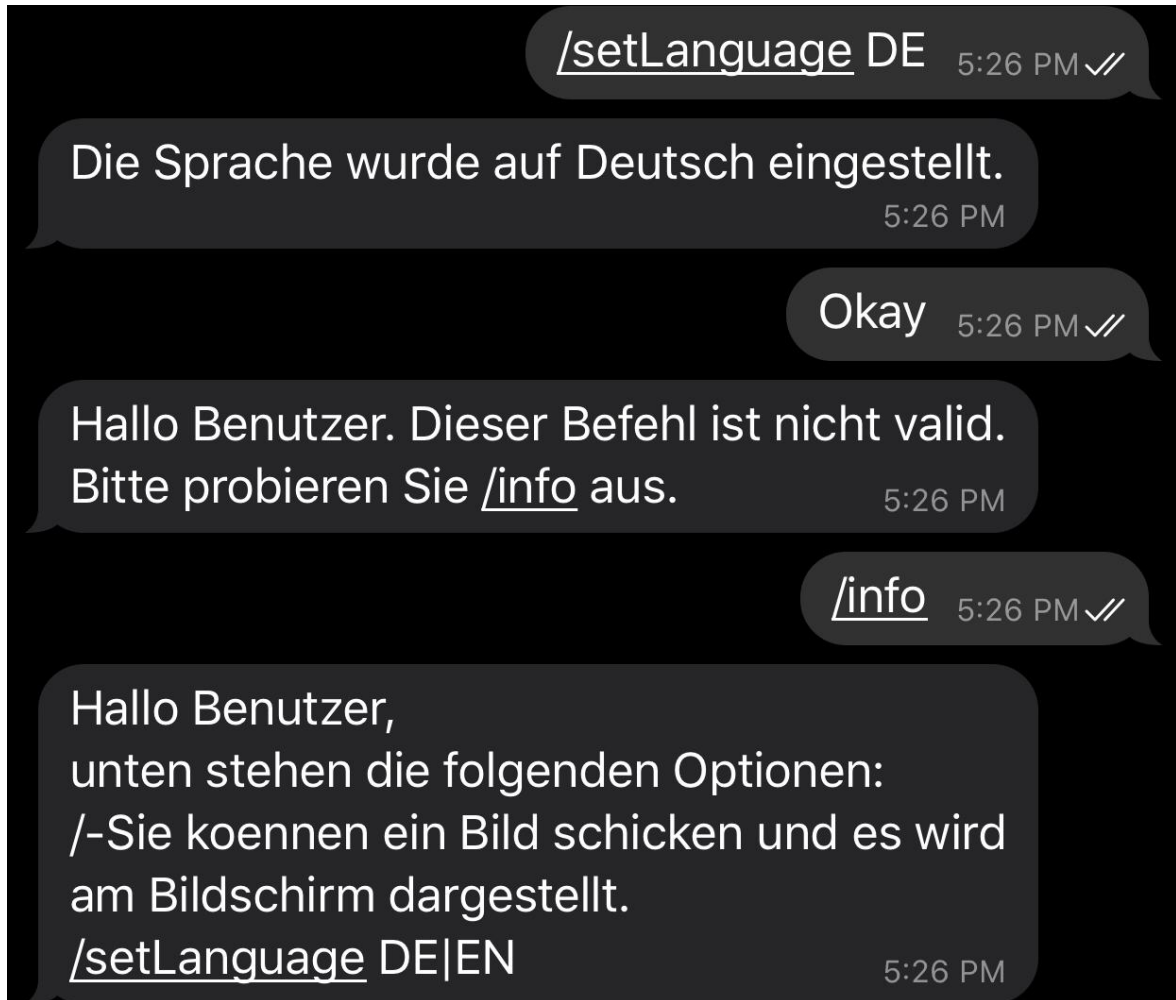


Abbildung 4.26: Spracheinstellungen für einen normalen Benutzer

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

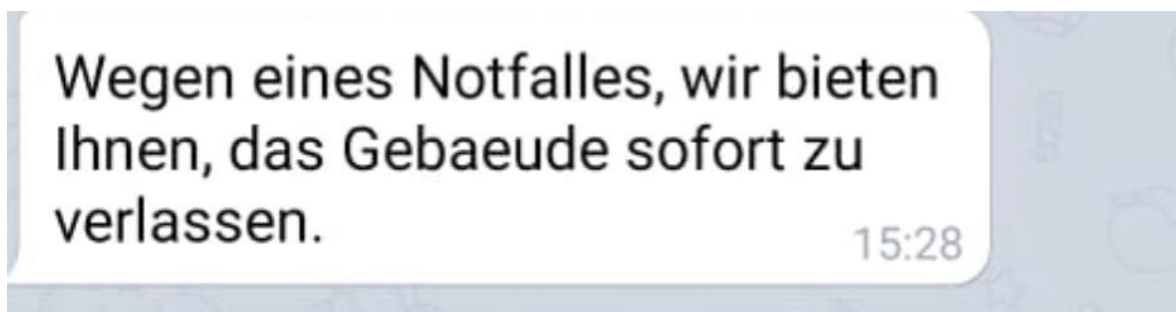


Abbildung 4.27: Notfallwarnmeldung vom Administrator

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.

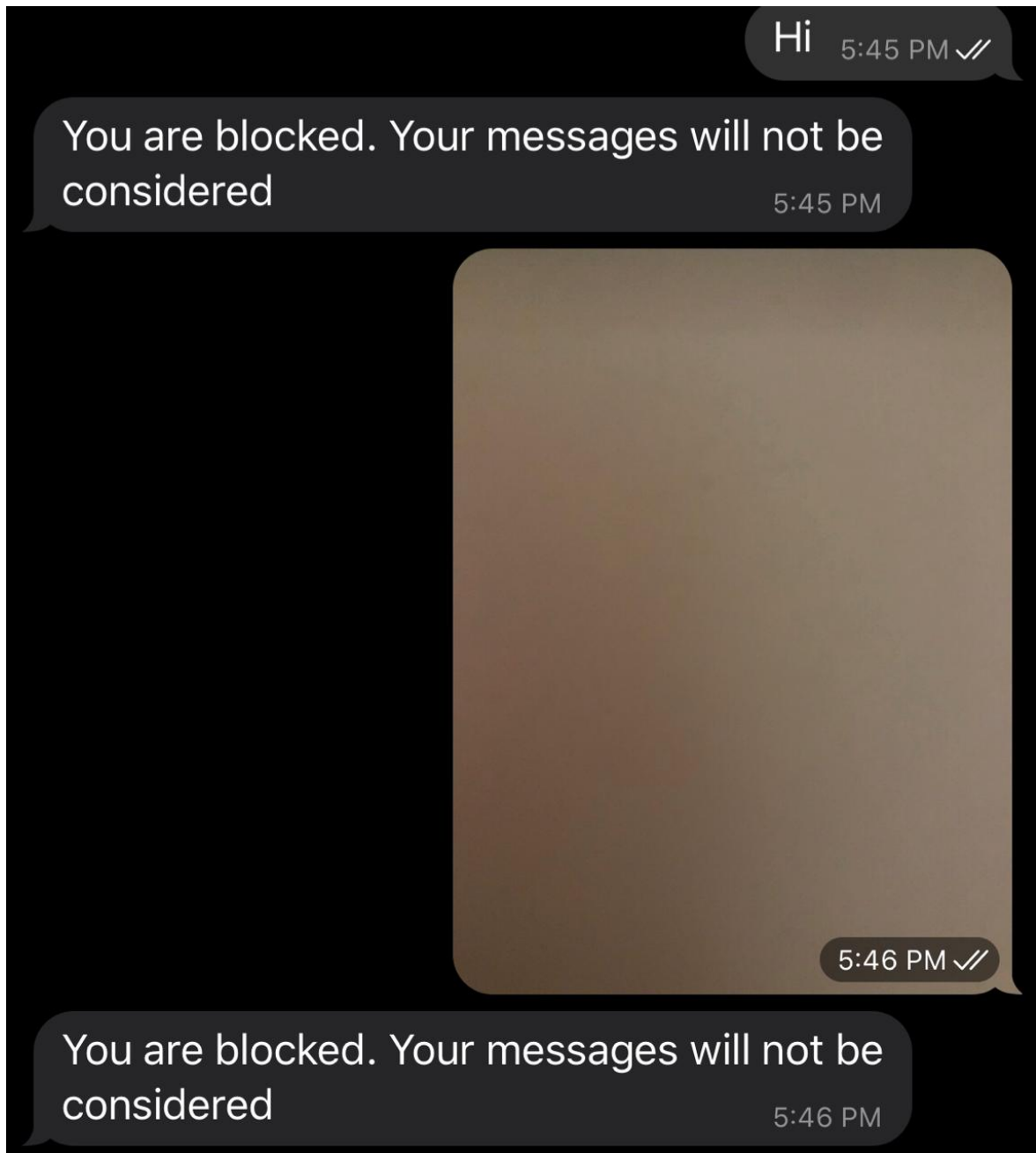


Abbildung 4.28: Nachricht von einem blockierten Benutzer

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.

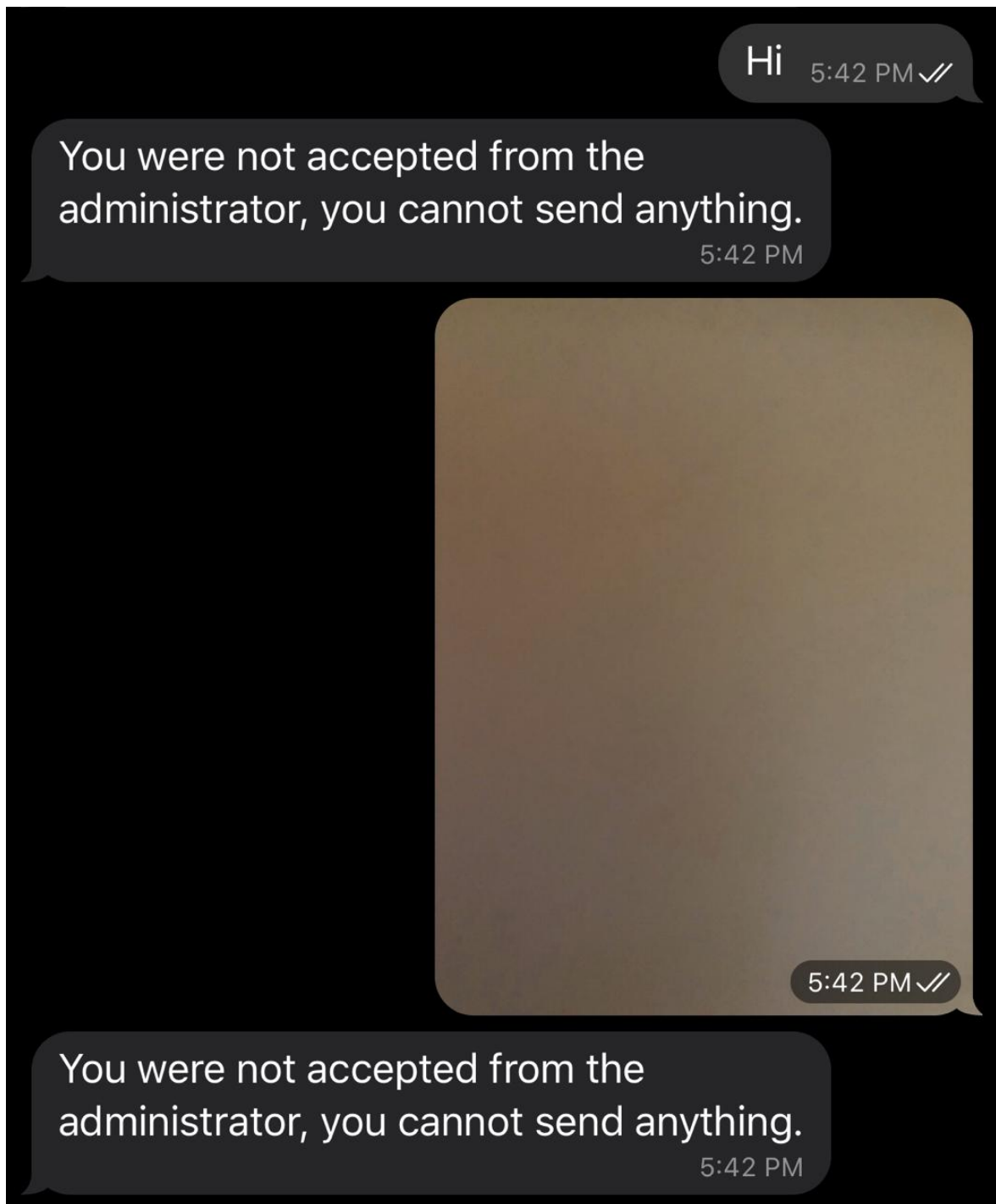


Abbildung 4.29: Nicht akzeptierter Benutzer

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.

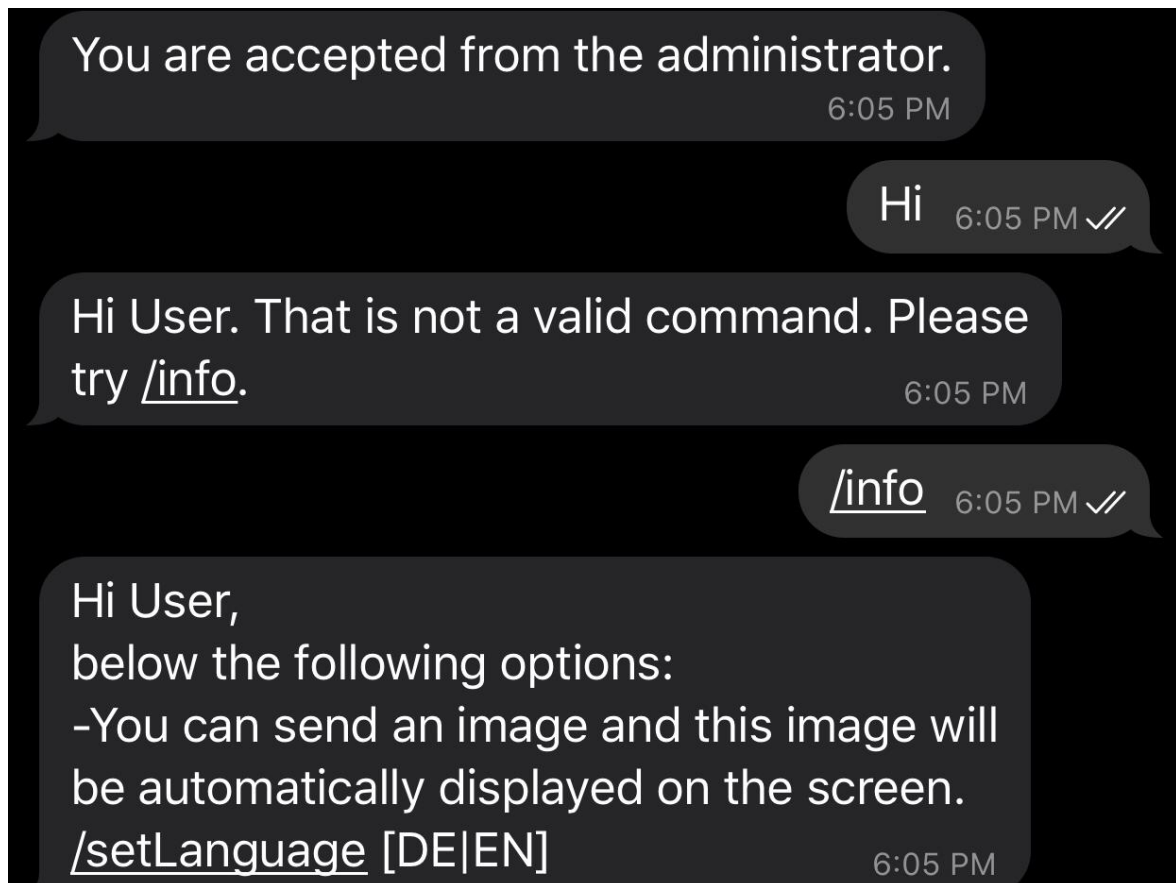


Abbildung 4.30: Akzeptierter Benutzer

4.10 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

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.

Offline Betrieb

Die Implementierung von Offline Betrieb war eine eigene Herausforderung. Zuerst sollte das Prinzip und der Weg der Programmierung genau überlegen 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. 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 konnte sein, dass der Chatbot die falschen Nachrichten zu den Benutzern schickte und die Funktionalitäten stimmten dann nicht ganz.

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.

4.11 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.

4.12 Ergebnisse - Irena Bala

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst.

4.12.1 Implementierung

Die wichtigsten Ergebnisse der Arbeit sind:

- Die Implementierung von einer Hauptkomponente der künstlichen Intelligenz wie Chatbot, die das entwickelte System interessanter macht. Diese Komponente wird multisprachig angeboten.
- Designvorbereitung für die Anzeige
- 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
- Implementierung von Offline Betrieb
- SSL-Verschlüsselung für eine sichere Datenübertragung
- Einrichtung des Systems

4.13 Handbuch für die Bedienung

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

4.13.1 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 Benutzername Infotainment gesucht werden. Der Chat mit dem Infotainment Bot wird geöffnet. Jeder Benutzer, die 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 ihm 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. Die Benutzer haben die Möglichkeit, selbst die Sprache auszuwählen, in der die Interaktion mit dem Chatbot erfolgen soll.

4.13.2 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
- 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 schauen will, soll er `/info` eingeben.

4.14 Evaluierung und Resümee

4.14.1 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)
- 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, 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.

4.14.2 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

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, deswegen war diese Programmiersprache sehr hilfreich und ist sehr oft vorgekommen. 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.

Abbildungsverzeichnis

2.1	Projektstrukturplan	5
2.2	Structed Design - Client	9
2.3	Structed Design - Server	10
2.4	Hybrides Modell	11
4.1	Structured Software Design vom Server	29
4.2	ERD	31
4.3	Kalenderinformationen	32
4.4	Leere Klassen für Montag, die zweite Stunde	32
4.5	Wetterdaten	34
4.6	Chatbot Multilanguage	42
4.7	Hauptmenü für den Administrator	43
4.8	Auflistung der Chatbot-Benutzer	44
4.9	Auflistung der unregistrierten Benutzer	44
4.10	Keine Unregistrierte Benutzer	45
4.11	Genehmigung des Administrators	45
4.12	Ablehnung des Administrators	45
4.13	Notfallwarnmeldung	46
4.14	Benutzer Blockierung	46
4.15	Spracheinstellungen	47
4.16	Hauptmenü für den unregistrierten Benutzer	48
4.17	Registrierung beim Chatbot	49
4.18	Falsche Eingabe des Telefonnummers	49
4.19	Telefonnummer zu kurz	50
4.20	Korrekte Eingabe des Telefonnummers	50
4.21	Das Warten der Benutzer auf die Administrator Bestätigung	51
4.22	Keine Registrierung	51
4.23	Spracheinstellungen für unregistrierte Benutzer	52
4.24	Hauptmenür für einen normalen Benutzer	53
4.25	Bilderfunktionalität	53
4.26	Spracheinstellungen für einen normalen Benutzer	54
4.27	Notfallwarnmeldung vom Administrator	54
4.28	Nachricht von einem blockierten Benutzer	55
4.29	Nicht akzeptierter Benutzer	56
4.30	Akzeptierter Benutzer	57

Tabellenverzeichnis

2.1	Meilensteine	4
2.2	Arbeitspakete	8
4.1	Technologien	23
4.2	Bereiche und Technologien	28

Quellcodeverzeichnis

4.1	Implementierung von Offline Betrieb	33
4.2	Offline Betrieb-Erstellung der fehlenden Tabellen im Client	33
4.3	Select Query für die registrierten Benutzer in die Datenbank	38
4.4	Darstellung aller Benutzer	38
4.5	Darstellung der unregistrierten Benutzer	39
4.6	Genehmigungen für die neue Registrierungen	39
4.7	Genehmigungen für die neue Registrierungen	39
4.8	Blockierung der Benutzer	40
4.9	Notfall Nachricht	40
4.10	Spracheinstellungen	40
4.11	Hauptmenü für den Administrator	41
4.12	Überprüfung der Eingabe der Benutzer	41
4.13	Hinzufügen eines neuen Benutzers zur Datenbank	41

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).