Höhere technische Schule fur Informationstechnologie
Shkolla e mesme profesionale private për teknologji informacioni
Österreichiche 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, 13. Dezember 2019

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 minikompjutera 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 mënyre apo në një tjetër thyejnë monotoninë gjatë rutinës ditore shkollore.

Danksagung

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 Wasserfall 9

3 Dokumentation des Projektverlaufs **12**

3.1 Beschreibungen des Arbeitsverlaufs 12

3.1.1 Irena Bala 12

3.1.2 Aldo Sheldija 17

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 Technische Lösungen 28

4.2.1 Structured Software Design 28

4.2.2 Konfiguration von Raspberry PI Server 28

4.2.3 Datenbank . . . 29
4.2.4 SSL Verschlüsselung . . . 30
4.2.5 Wetterdaten . . . 31
4.2.6 Anzeige . . . 31
4.2.7 Chatbot . . . 31
4.2.8 Chatbot Einrichtung: . . . 32
4.2.9 Einrichtung des Chatbots in RaspberryPI . . . 32
4.2.10 Grundlagen für die Umsetzung von Chatbot . . . 32
4.2.11 Funktionalitäten von Chatbot . . . 33
4.2.12 Unregistrierte Benutzer: . . . 33
4.2.13 Konkrete Beispiele mit Bildern . . . 36
4.3 Probleme, Herausforderungen und deren Lösung . . . 46
4.4 Qualitätssicherung, Controlling . . . 46
4.5 Ergebnisse - Irena Bala . . . 46
4.5.1 Implementierung . . . 47
4.6 Handbuch für die Bedienung . . . 47
4.6.1 Beschreibung der Bedienung als User . . . 47
4.6.2 Beschreibung der Bedienung als Administrator . . . 48
4.7 Evaluierung und Resümee . . . 48
4.7.1 Planung vs Realisierung . . . 48
4.7.2 Wertschöpfung und Lessons Learned . . . 48

5 Umsetzung - Aldo Sheldija 49
5.1 Allgemeine Beschreibung . . . 49
5.2 Technologien . . . 49
5.2.1 Bootstrap . . . 49
5.2.2 PHP . . . 50
5.2.3 JavaScript . . . 50
5.2.4 DataTables . . . 51
5.2.5 Font Awesome . . . 51
5.2.6 MySQL . . . 51
5.2.7 Apache . . . 52
5.3 Structed Software Design für den Client . . . 52
5.3.1 Start . . . 52
5.3.2 Zeitsteuerung . . . 53
5.3.3 Daten holen . . . 53
5.3.4 Buffer . . . 53
5.3.5 Anzeige Prepare . . . 54
5.3.6 Log . . . 54

5.4 Datenbank 54
5.4.1 ER Diagramm 54
5.5 Admin Webseite 55
5.5.1 Login 56
5.5.2 Passwort Zurücksetzen 56
5.5.3 Menü 57
5.5.4 Rechte in die Webseite 57
5.5.5 Webseite auf mehrere Sprachen 58
5.6 Client 58
5.6.1 Layout 1 58
5.7 Wordpress Webseite API 59
5.8 Herausforderungen und deren Lösung 59

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 selbsterstellen 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 Anzeigezeiten 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.

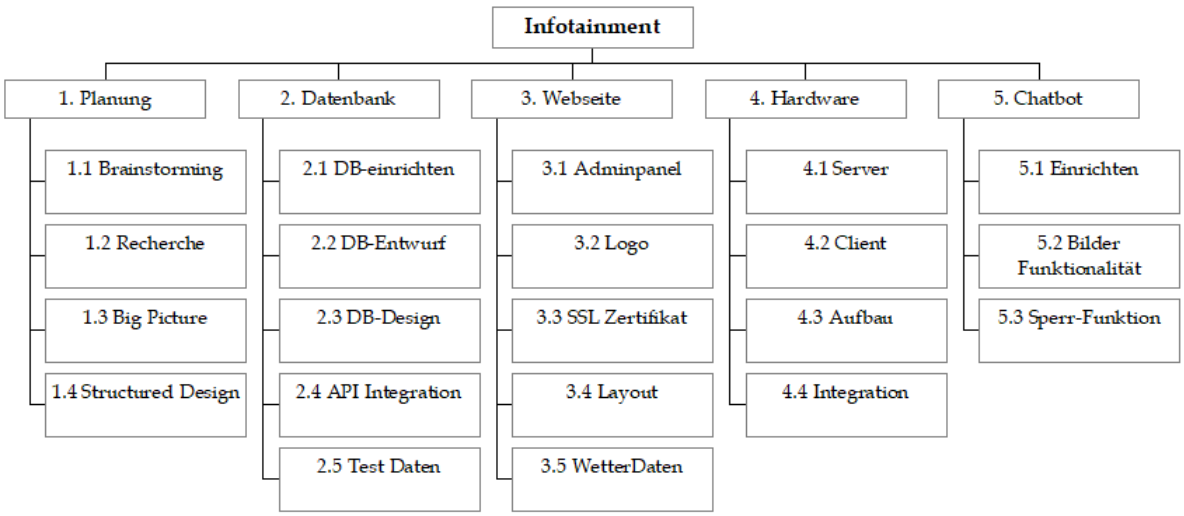

Abbildung 2.1: Projektstrukturplan

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 Zertifikate Keys anlegen - SSL Zertifikate 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

2.3 Projektmanagementmethode

2.3.1 Structed Design

Structed Design ist eine systematische Methode, die verwendet wird, um eine Software so gut wie möglich zu beschreiben. Mit dieser Methode wird das Architekturdesign, die fließenden Daten und Signale, sowie alle Schnittstellen sehr leicht und deutlich beschrieben. Ein weiterer Vorteil dieser Methode ist, dass es sehr leicht zu verstehen ist. In diesem Projekt wurde Structed Design verwendet, aufgrund der guten Möglichkeit, dass sie bietet, Iterationen zu verwenden und tiefer in die wichtigsten Ebenen der Arbeit zu gehen.

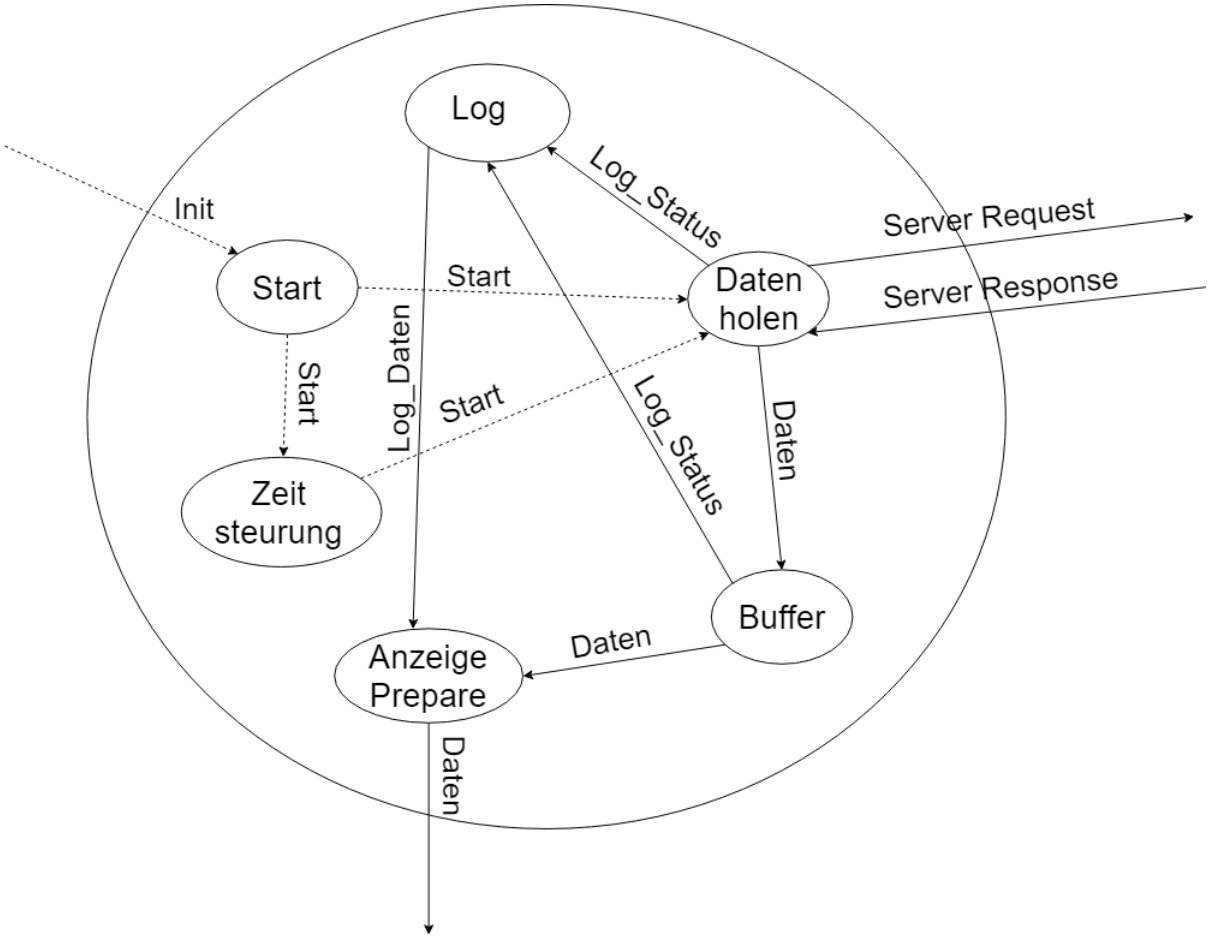

Abbildung 2.2: Structed Design - Client

.

2.3.2 Wasserfall

Das Wasserfall Modell(Abbildung 2.3.2) ist die zweite Projektmanagement Methode, die von uns gewählt wurde. Diese Methode teilt die wichtigsten Prozesse und Phasen

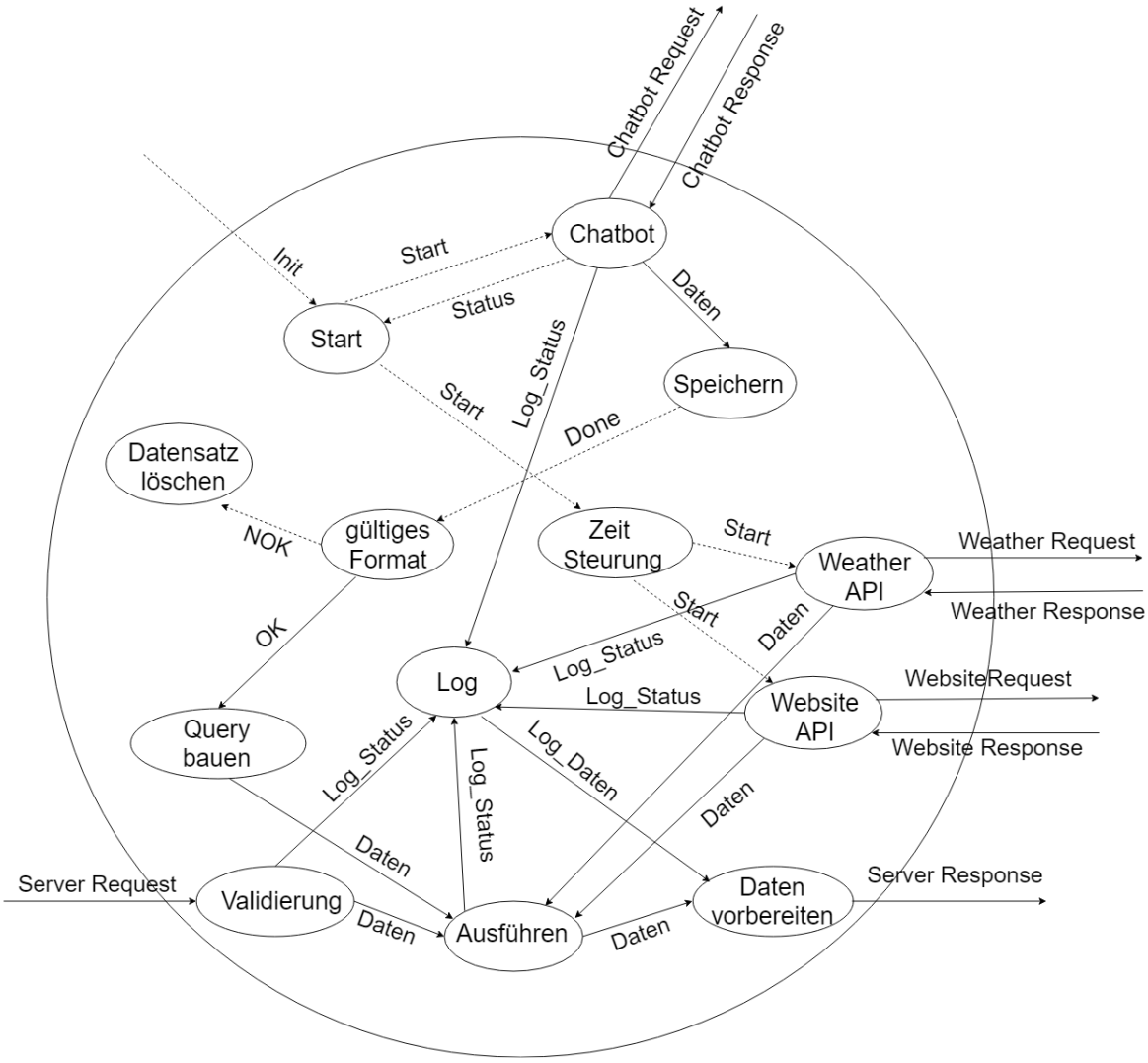

Abbildung 2.3: Structed Design - Server

des Projekts so, dass sie sequenziell nacheinander bearbeitet werden und dass jede Phase von der vorherigen Phase abhängig ist. Die wichtigsten Schritte eines Wasserfallmodells sind: Anforderung, Analyse, Entwurf, Realisierung, Test, Systemintegration und Systemabnahme. Das Projekt geht durch diese sieben Schritte bzw. Phasen sequentiell durch, bis die Arbeit abgeschlossen ist. Der Name dieses Modells leitet sich aufgrund des Fließens des Projektes durch alle Phasen ab. Das Modell wird sehr gerne bei der Erstellung von Webseiten oder bei Entwurf von Datenbanken angewendet. Obwohl diese Methode sehr langwierig sein kann und nicht sehr flexibel ist (Fehlerkorrektur), ermöglicht es eine sehr klare und leichte Planung und Organisation der Arbeit, sowie eine gute Zielabgrenzung und Aufwandsabschätzung.Die oben genannten Merkmale sind der Grund, warum diese Methode für dieses Projekt ausgewählt wurde.

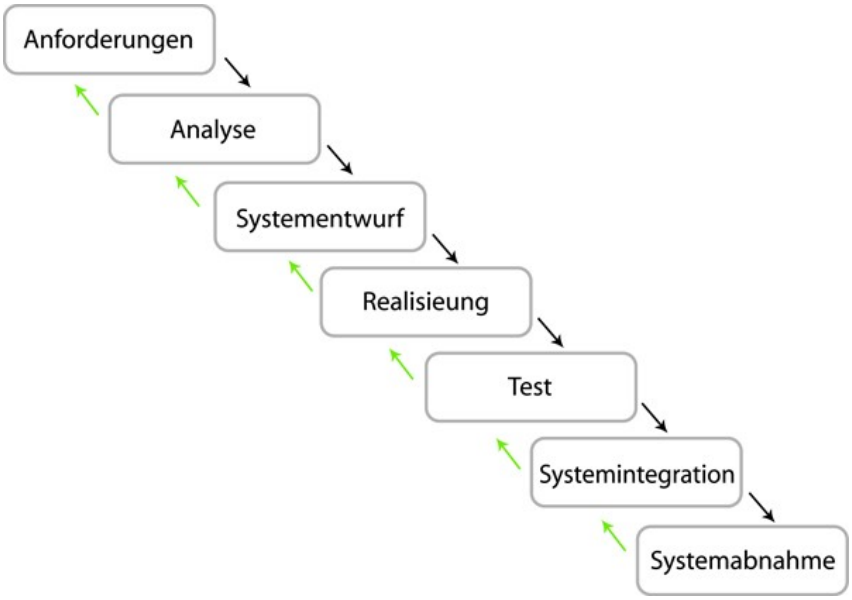

Abbildung 2.4: Wasserfallmodell

Kapitel 3

Dokumentation des Projektverlaufs

3.1 Beschreibungen des Arbeitsverlaufs

3.1.1 Irena Bala

Detailed report

toggl

2019-09-01 - 2019-12-28
Total 109 h 15 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.** Infotainment	**5:00:00** 13:00-18:00	Irebal14
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.** Infotainment	**2:00:00** 14:00-16:00	Irebal14
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.** Infotainment	**4:30:00** 16:30-21:00	Irebal14
09-11	**Die richtigen Materialien für die Diplomarbeit wurden gewählt. Dann wurde eine Hardwareliste geschrieben, damit die benötigten Materialien bestellt werden könnten.** Infotainment	**1:30:00** 14:00-15:30	Irebal14
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.** Infotainment	**1:15:00** 16:00-17:15	Irebal14
09-12	**Der Diplomarbeitsantrag wurde vorbereitet. Dazu gehört die Formulierung der Ziele, Aufgabestellung in drei verschiedenen Sprachen, die Aufgabenteilung, die Erstellung der Arbeitspaketen usw.** Infotainment	**5:00:00** 15:00-20:00	Irebal14
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.** Infotainment	**4:00:00** 12:00-16:00	Irebal14
09-13	**Einige Punkte im Antrag wurden gemäß den Anweisungen des Betreuers korrigiert.** Infotainment	**1:30:00** 17:00-18:30	Irebal14
09-13	**Es wurden alle benötigten Informationen für die Einreichung unserer Diplomarbeit bei dem Plattform Diplomarbeits-DB ergänzt.** Infotainment	**1:00:00** 19:00-20:00	Irebal14
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.** Infotainment	**1:00:00** 14:00-15:00	Irebal14
09-22	**Es wurde ein ER Diagramm im 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.** Infotainment	**3:00:00** 13:30-16:30	Irebal14
09-25	**Das ER-Diagramm wurde in der Gruppe besprochen. Es gab ein paar Änderungen, die dann verbessert worden sind.** Infotainment	**1:00:00** 15:00-16:00	Irebal14

09-30	**Die Tabellen Supplierplan, Layout, LayoutSections, Weather, Unterricht wurden in MySqlWorkbench mit den entsprechenden Spalten, Datentypen, Primär und Fremdschlüssel durch Skripts erstellt.** Infotainment	**1:30:00** 15:00-16:30	Irebal14
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 Diplomarbeitsmitglied gegeben.** Infotainment	**1:15:00** 20:00-21:15	Irebal14
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 Diplomarbeitsmitglied gegeben.** Infotainment	**1:15:00** 20:00-21:15	Irebal14
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 verschiedener Anzahl von Teilbereichen, wo die Informationen dargestellt werden könnten.** Infotainment	**2:30:00** 10:00-12:30	Irebal14
10-12	**An diesem Tag wurden nur die folgenden Kapiteln geschrieben: Aufgabenstellung, Idee, Thema, Allgemeines und die Planung.** Infotainment	**4:30:00** 14:00-18:30	Irebal14
10-13	**An diesem Tag wurden die restlichen Kapiteln geschrieben: Projektplanung (Arbeitspaketen), Team, Projektmanagementmethode und die Projektziele.** Infotainment	**2:30:00** 14:00-16:30	Irebal14
10-13	**Einige Punkte wurden nach den Anweisungen des Betreuers korrigiert.** Infotainment	**1:30:00** 17:00-18:30	Irebal14
10-18	**Zuerst wurden generelle Informationen über APIs gesammelt. Es wurde nach 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.** Infotainment	**4:30:00** 13:00-17:30	Irebal14
10-20	**Es wurde eine Stored Procedure erstellt, die als Supplierer, die jenigen Personen darstellt, die nicht fehlen, die kein Unterricht haben und eine Supplierstunde haben. Diese Lehrer wurden in einer bestimmten Reihenfolge geordnet.** Infotainment	**5:00:00** 15:00-20:00	Irebal14
10-26	**Es wurden Prozeduren für die restlichen Tabellen erstellt, falls sie für die Zukunft notwendig sind.** Infotainment	**4:00:00** 11:00-15:00	Irebal14
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.** Infotainment	**2:00:00** 16:00-18:00	Irebal14
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 Pakette für MySql), python, telepot (die Pakete für Chatbot). Es wurde die IP-Adresse des Servers angelegt. Danach wurde 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.** Infotainment	**2:00:00** 12:00-14:00	Irebal14

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 letztens wurde SSL Virtual Host aktiviert und apache neugestartet.** Infotainment	**2:15:00** 17:00-19:15	Irebal14
11-23	**Es wurden eine intensive Recherche über Chatbot gemacht. Es wurde nach einer guten API gesucht. Danach wurden Beispiele geschaut und analysiert.** Infotainment	**2:00:00** 13:00-15:00	Irebal14
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 Paketen 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 zurückkriegen.** Infotainment	**1:30:00** 16:00-17:30	Irebal14
11-30	**Es wurden die verschiedenen Funktionalitäten programmiert. Zuerst wurde übeprüft ob der jenige, der zum Chatbot schreibt, der Administrator ist oder nicht. Falls ja, werden ihn 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 ihn eine Message zurückgeschickt, dass er kein Administrator ist.** Infotainment	**2:30:00** 18:00-20:30	Irebal14
12-01	**Es wurde eine neue Tabelle in der Datenbank für die Benutzer von Chatbot erstellt. Zusätzlich wurde Token auch in der Datenbank gespeichert.** Infotainment	**0:45:00** 11:30-12:15	Irebal14
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.** Infotainment	**2:00:00** 16:00-18:00	Irebal14
12-03	**Den Benutzern, die für 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 druckt, wird Chatbot ihn sagen, die Telefonnummer zu schreiben. Die Telefonnummer wird dann in der Datenbank gespeichert zusammen mit der Chat-Id, die eindeutig ist und mit den anderen Datensätzen. Wenn die Telefonnummer geschickt wird, wird dieser Benutzer warten bis der Administrator seine Registrierung genehmigt hat.** Infotainment	**7:00:00** 15:00-22:00	Irebal14
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.** Infotainment	**4:30:00** 16:00-20:30	Irebal14
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.** Infotainment	**5:00:00** 16:00-21:00	Irebal14
12-06	**Es wurde versucht, die Kalenderinformationen so zu adaptieren, dass sie in allen Teilbereichen des Anzeigelayouts hineinpassen.** Infotainment	**4:30:00** 17:00-21:30	Irebal14

12-07	**Es wurde versucht, die Wetterdaten so zu adaptieren, dass sie in allen Teilbereichen des Anzeigelayouts hineinpassen.** Infotainment	**4:30:00** 16:00-20:30	Irebal14
12-08	**Es wurde versucht, den Stundeplan 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.** Infotainment	**6:00:00** 13:00-19:00	Irebal14
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.** Infotainment	**3:30:00** 14:00-17:30	Irebal14

Created with toggl.com

3.1.2 Aldo Sheldija

Detailed report

toggl

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

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

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

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 also darin, das System zu vervollständigen und sein Entwicklungsgemeinde zu vergrößern.
Diese Komponente wurden in unterschiedlichen Bereichen unterteilt. Der erste Bereich ist die Datenbank. Die Datenbank ist der wesentliche Bestandteil des Systems, weil die Basis für die Speicherung der benötigten Daten bildet. In die Datenbank wurden alle von den verwendeten APIs gekommene Daten gespeichert. Außerdem wurden dort auch die Schuldaten gelegen. Diese Daten sind ein separater Teil des Projekts, die in den folgenden Kapiteln genauer erklärt werden.
Die Darstellung an Bildschirm von den gespeicherten Daten wurde durch eine Admin-Webseite realisiert. Es wurden viele Layouts für die Anzeige (Bildschirm) entworfen, damit die Informationen auf unterschiedlichen Weisen dargestellt werden.
Das Infotainment System funktioniert wie die meisten Systeme nach dem Server- Client Prinzip. Sowohl der Server, als auch der Client werden näher betrachtet, denn sie die Hauptbereiche des ganzen Systems sind. Ein weiterer interessanter Bereich ist der Chatbot. Chatbot wurde deswegen implementiert, weil es die Interaktion des Menschen mit dem System ermöglicht und dadurch wurde angenommen, dass die Umsetzung dieses Komponentes das Interesse des Menschen an dem System erhöhen wird.
Dies war eine allgemeine Beschreibung von den bis jetzige erreichte Ergebnisse. Entsprechend der jeweiligen Arbeitsaufteilung werden in den folgenden Kapiteln einige von den oben genannten Aspekten näher erläutert.

4.1.1 Chatbot

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

Einführung in Chatbot

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

Umsetzung von Chatbot

Chatbot wurde bei dieser Diplomarbeit so implementiert, dass es den Schülern die Möglichkeit gibt, aufgenommene Bilder zum Chatbot zu schicken und dann diese Bilder werden automatisch auf dem Bildschirm angezeigt. Das ist auch die wesentliche Funktionalität von Chatbot bei diesem Projekt.
Um den Chatbot zu implementieren, haben viele kleine Prozesse stattgefunden, die als weitere oder zusätzliche Funktionen angesehen werden können.
Zuerst wurde Telegram Bot API als eine Schnittstelle für die Chatbot-Implementierung ausgewählt. Durch diese API können neue Bots erstellt, bearbeitet und verändert werden. Telegram Bot API funktioniert gleich wie die anderen Kommunikationsapplikation z.B Whatsapp. Der wesentliche Unterschied ist, dass bei dieser Applikation wird nicht nur die Möglichkeit mit anderen Menschen zu chatten angeboten, sondern auch mit Chatbots. Jedes Bot, das erstellt wird, bekommt ein Token, dass eindeutig für das Bot ist, wie die Telefonnummer für uns eindeutig ist.
Die komplette Funktionalität des Bots wurde in RaspberryPI Server, mithilfe der Python Programmiersprache programmiert.
Je nachdem ob die Person, die mit dem Bot chatten will, als ein normaler Benutzer oder ein Administrator in der Datenbank definiert ist, werden ihm verschiedene Funktionen im Zusammenhang mit dem Bot zur Verfügung gestellt. Die Nachrichten, die zu dem Bot geschickt werden, werden nach Inhalt überprüft. Basierend auf Inhalt der Nachrichten, wird der Bot auf verschiedene Weisen reagieren.
Die Art der Umsetzung bzw. Realisierung aller diesen Funktionen kann im Unterkapitel 2.1 gelesen werden.

4.1.2 Server

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

Im Server liegen aber alle benötigten Informationen für die Darstellung. Diese Informationen sind auf der Datenbank gespeichert, die auf dem Server stattfindet.
Der Server beinhaltet auch die grundlegenden Skripts für die Chatbot-Implementierung und für die Programmierung der Admin-Webseite. Im Server wurde auch das SSL Zertifikat für eine sichere Datenübertragung erstellt.
Die Kommunikation zwischen dem Client und dem Server wird dann aufgebaut, wenn Daten von dem Server ausgewählt und zum Client geschickt werden. Der Server ist der grundlegende Teil des Projekts. Es enthält alle benötigten Ressourcen für die vollständige Umsetzung des Systems.

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

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

Tabelle 4.1: Technologien

Was ist Apache HTTP Server?

Der Apache HTTP[1] Server ist ein weltweit verbreitender Web Server. Dieser Server ist Open Source, das bedeutet, dass es keine Lizenz gekauft werden soll, um es zu verwenden. Es ist kompatibel auf allen kohärenteren Betriebssystemen, beispielsweise Linux, Windows, Mac OS und andere. Es bietet viele Versionen an, die zu unterschiedlichen Anwendungsgebiete passen und verbesserte Eigenschaften bereitstellen. Durch dieses Webservers 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

[1] Hypertext Transfer Protocol

SSL [2]Zertifikats. Das ermöglicht die Übertragung der Daten in einer verschlüsselten Form. Die detaillierte Funktionsweise eines SSL Zertifikats wird in den Unterkapiteln beschrieben. [**50˙apache**]

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.“ [**50˙apache**]

Was ist MySQL?

MySQL ist ein weitverbreitetes relationales Datenbanksystem. Die Datenbanksysteme werden allgemein zur Datenspeicherung und Datenverwaltung verwendet. Ein wichtiges Kriterium für die Datenspeicherung ist die Performanz. Diese Anforderung wird durch MySQL optimal erfüllt. Das ist auch der Grund, warum dieses Datenbanksystem so populär und bekannt ist. Die von MySQL für die Abarbeitung, Verwaltung und Systematisierung von Daten verwendete Sprache ist SQL[4]. MySQL ist auch eine Open Source Software, die in meisten Fällen in Verbindung mit serverseitigen Scriptsprachen wie PHP, vorkommt. [**50˙mysql**]

Funktionsweise von MySQL

Das MySQL Datenbanksystem wird sehr häufig implementiert. Es gibt viele Unternehmen und Institutionen, die ihre Daten über eine gewisse Zeit speichern wollen. Das MySQL Datenbanksystem, das die Daten beinhaltet, wird als ein Server vorgesehen. Jeder, der versucht, Zugriff auf diese Daten zu haben, wird als ein Client vorgesehen. Der Server kann die erforderliche Zugänglichkeit erlauben oder nicht. Das hängt von den Clientrechten ab. Die Daten können von den Clients selektiert, bearbeitet oder gelöscht werden. Diese Ereignisse erfolgen durch SQL-Abfragen. Die SQL-Abfragen werden mithilfe der SQL Datenbanksprache erstellt. [**50˙mysql**]

Was ist PHP?

PHP[5] ist eine serverseitige Programmiersprache. Das bedeutet, dass diese Sprache, um die vom Server auszuführenden Ereignissen zu programmieren, verwendet wird.

[2]Secure Socket Layer
[3]Hypertext Transfer Protocol Secure
[4]Structured Query Languague
[5]Parallel History Project

PHP ist eine sehr verbreitete Programmiersprache, die am meisten zur Erstellung und Programmierung von Webseiten verwendet wird. Eigentlich ist PHP sehr flexibel, denn es einen großen Schnittstellenansatz anbietet. Diese Programmiersprache kann auch im Zusammenhang mit Datenbanken genutzt werden. [**50˙php**]

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. [**50˙php**]

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. [**50˙javascript**]

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" [**50˙javascript**]

Was ist Python?

Python ist eine objektorientierte Programmiersprache, aber kann auch in prozedurale Programmierung verwendet werden. Sie wurde ausschließlich zum Zweck der einfach einprägsamen Syntax entwickelt. Andererseits haben die Entwickler der Systematisierung des Codes große Bedeutung beigemessen. Wegen dieser angewandten Eigenschaften kann Python in die Gruppe der leichten Programmiersprachen aufgenommen werden. Diese Programmiersprache wird viel verwendet, aber was die Anwendungsumgebung besonders erhöht, ist die Möglichkeit andere Module anzuhängen. Es ist auch eine Open Source Software, der von den Programmierern verwendet, verändert, angepasst bzw. bearbeitet werden kann. Es wird meistens für komplexe Aufgaben verwendet werden, deswegen wird es als eine Hochsprache betrachtet. [**50˙python**]

Merkmale von Python

- Einfach einprägsame Syntax
- Objektorientierte und prozedurale Programmiersprache
- Open Source
- Hoches Niveau Programmiersprache
- Leicht veränderbare Programmiersprache[**50˙python**]

Was ist HTML?

HTML[6] ist keine Programmiersprache, die wird für die Erstellung von Inhalten bei Webseiten verwendet. Diese Inhalte können Texte, Bilder oder andere Komponente sein. HTML wird als eine Auszeichnungssprache betrachtet. Sie ist nicht nur für die Erstellung von Webseite-Inhalten zuständig, sondern auch für ihr Design. Diese Sprache liegt mithilfe von bestimmten Tags die Struktur einer Webseite fest. Im Tag werden die Inhalte gespeichert. Es gibt bestimmte Tags für verschiedene Layout-Elemente.[**50˙html**]

Was ist CSS?

CSS[7] wird im Zusammenhang mit HTML verwendet. Diese Methode wird unten genauer betrachtet.
„CSS steht für Cascading-Style-Sheets und ist eine Möglichkeit für HTML-Dokumente, den Inhalt einer Seite von den Designanweisungen der einzelnen Elemente, wie zum Beispiel Überschriften, Zitaten) zu entkoppeln.“ [**50˙css**]

[6] HyperText Markup Language
[7] Cascading Style Sheets

Was ist Raspberry PI?

Der Raspberry PI ist ein Minirechner, die zur Linux Programmierung, Shell Scripting und Realisierung von technischen Projekten verwendet wird. Es braucht eine Tastatur, eine Maus, ein Netzteil, VGA[8] und HDMI[9]-VGA Konverter, damit es genutzt werden kann. Die Konfiguration von einem Raspberry PI erfolgt durch eine SD-Karte. Diese SD[10]-Karte beinhaltet das Image, wo das Betriebssystem liegt. Ein Raspberry PI kann in Zusammenhang mit vielen anderen Komponenten verwendet werden.[**50˙raspi**]

Was ist SSL?

SSL steht für Secure Socket Layer und ist für die verschlüsselte Übertragung der Daten vom Browser zum Server verantwortlich. Die Verbindung zwischen dem Server und dem Browser erfolgt durch das HTTPS Protokoll. Das ist ein Kommunikationsprotokoll, das eine verschlüsselte Datenübertragung ermöglicht. Heutzutage wird TLS[11] am meisten verwendet, der der neueste und modernste Standard von SSL ist.[**50˙ssl**]

SSL-Verschlüsselung

Um eine verschlüsselte Verbindung zwischen einem Browser und einem Server aufzubauen, werden SSL – Zertifikaten integriert. Mittels ein SSL Zertifikats wird die Authentizität einer Webseite überprüft. Das SSL Zertifikat wird von einer Zertifizierungsstelle, erzeugt. Diese Zertifizierungsstelle heißt CA[12] und erfordert einige Daten von dem Antragsteller, die für die Erstellung des Zertifikats notwendig sind. Als nächstes, erzeugt der Antragsteller für die Entschlüsselung und Verschlüsselung der zwischenübertragenen Daten ein öffentlicher, -und ein privater Schlüssel. Je grösser die Lange des Schlüssels ist, desto sicherer und besser ist. Meistens werden Schlüssel mit einer Länge von 256 Bit verwendet.[**50˙ssl**]

Was ist Telegram Bot API?

Telegram Bot API[13] ist eine Schnittstelle, die die Implementierung von Chatbot ermöglicht. Es bietet verschiedene Funktionen an, nämlich die Einrichtung, Erstellung und die Verarbeitung von Bots. Diese Funktionen sind in der eigenen Dokumentation von Telegram Bot API klar beschrieben. [**50˙telegram**]

Was ist Wetter API?

„Wetter APIs sind Schnittstellen, die die Verbindung zu einer großen Wettervorhersage-Datenbank und die Aufnahme benötigter Daten ermöglichen.“ [**50˙Wetter**]

[8] Video Graphics Array
[9] High-Definition Multimedia Interface
[10] Secure Digital Memory Card
[11] Transport Layer Security
[12] Certificate Authority
[13] Application Programming Interface

Was ist JSON?

„JSON[14] 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.“
[**50˙json**]
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 Technische Lösungen

4.2.1 Structured Software Design

4.2.2 Konfiguration von Raspberry PI Server

- Zuerst wurde ein Image daraufgespielt, die das Betriebssystem von Raspberry PI beinhaltet
- Danach wurden die folgenden gebrauchten Paketen installiert: git, vim, apache2, python-pip, telepot, php php-mbstring, mariadb-server php-mysql, phpmyadmin
- In der Konfigurationsdatei wurde die IP-Adresse von dem Raspberry PI angelegt
- Der nächste Schritt war die Erstellung der Datenbank und der dazugehörigen Tabellen
- Dann wurden die Benutzer angelegt und die Rechte vergeben
- Als letztens wurde SSH[15] aktiviert, damit eine sichere Verbindung zu diesem Server von einem externen Gerät ermöglicht werden kann

[14] JavaScript Object Notation
[15] Secure Shell

./figures/server.png

Abbildung 4.1: Server

4.2.3 Datenbank

- Zuerst wurde ein ER[16] Diagramm mit Papier gezeichnet. Das Ziel war die richtige Erstellung der benötigten Tabellen. Die Tabellen wurden mit den Spalten und ihren Datentypen erstellt. Es wurden auch die Kardinalitäten dazwischen gezeichnet. Die erstellten Tabellen waren:
  - Supplierplan
  - Stundenplan
  - Wetterdaten

[16]Entity Relationship

  - Chatbot Users

- Basierend auf das ER Diagramm wurden dann die Tabellen mit phpmyadmin erstellt

- Dann sind für alle Tabellen mit MySQL Workbench die entsprechenden gespeicherten Prozeduren erstellt

./figures/erd_irena.png

Abbildung 4.2: ERD

4.2.4 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 Virtuellhost mit SSL aktiviert.
- Es wurde ein System reboot gemacht und der Apache Server noch einmal gestartet [**50˙SSS**]

4.2.5 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

4.2.6 Anzeige

4.2.7 Chatbot

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

4.2.8 Chatbot Einrichtung:

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

4.2.9 Einrichtung des Chatbots in RaspberryPI

Die Entwicklung und die Programmierung von Chatbot wurde im RaspberryPI Server gemacht. Damit der RaspberryPI mit dem Telegram Bot API verbunden werden kann, sollten im Raspberry PI zwei Paketen installiert werden. Diese Paketen sind telepot und python-pip. Telepot ist die Pakete, die die Verbindung zum Telegram Bot API erstellt und mit einem Python-Framework funktioniert.[**50˙telegram**]

4.2.10 Grundlagen für die Umsetzung von Chatbot

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

4.2.11 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 ihn verschiedene Funktionalitäten angeboten.

4.2.12 Unregistrierte Benutzer:

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

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

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

Normale Benutzer:

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

Administrator:

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

Umsetzung der Funktionalitäten:

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

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

Listing 4.1: Chatbot 1

- Falls diese Person registriert ist, wird überprüft ob er ein Administrator oder ein normaler Benutzer ist. Falls er ein Administrator ist, werden die Nachrichten, die er zum Chatbot sendet analysiert und die entsprechenden Ergebnisse zurückgeschickt.

- Wenn der Administrator /users zum Bot schreibt, bedeutet dass er die normalen Benutzer sehen will. Es wird ein Query geschrieben, die diese Benutzer aus der Datenbank selektiert. An dem Administrator wird nur die ID des Benutzers geschickt.

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

Listing 4.2: Chatbot 2

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

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

Listing 4.3: Chatbot 3

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

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

Listing 4.4: Chatbot 3

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

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

Listing 4.5: Chatbot 4

- Falls ein normaler Benutzer die oberen Befehle probiert, wird der Chatbot ihn eine Nachricht zurückschicken, dass er nicht der Administrator ist.

- Durch diese Funktion wird überprüft, ob die Person, die den Button Ja gedruckt hat, registriert ist oder nicht. Falls nein bittet Chatbot ihn, die Telefonnummer einzuschreiben.

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

Listing 4.6: Chatbot 5

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

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

Listing 4.7: Chatbot 6

4.2.13 Konkrete Beispiele mit Bildern

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

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

./figures/chatbotbild88.jpg

Abbildung 4.3: Auflistung der Chatbot-Benutzer

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

./figures/chatbotbild9.jpeg

Abbildung 4.4: Auflistung der unregistrierten Benutzer

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

./figures/chatbotbild11.jpeg

Abbildung 4.5: Benutzer Blockierung

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

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

./figures/chatbotbild.jpeg

Abbildung 4.6: Registrierung beim Chatbot

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

./figures/chatbotbild2.jpeg

Abbildung 4.7: Falsche Eingabe des Telefonnummers beim Chatbot

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

./figures/chatbotbild5.jpeg

Abbildung 4.8: Falsches Format des Telefonnummers beim Chatbot

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

./figures/chatbotbild6.jpeg

Abbildung 4.9: Eingabe einer falschen Telefonnummer beim Chatbot

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

./figures/chatbotbild4.jpeg

Abbildung 4.10: Korrekte Eingabe der Telefonnummer beim Chatbot

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

./figures/chatbotbild7.jpeg

Abbildung 4.11: Das Warten der Benutzer auf die Administrator Bestätigung

In dem folgenden Bild wird die Interaktion eines blockierten Benutzer mit dem Chatbot dargestellt.

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

./figures/catboti.jpg

Abbildung 4.12: Nachricht von einem blockierten Benutzer

4.3 Probleme, Herausforderungen und deren Lösung

4.4 Qualitätssicherung, Controlling

4.5 Ergebnisse - Irena Bala

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst.

4.5.1 Implementierung

Die wichtigsten Ergebnisse der Arbeit sind:

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

4.6 Handbuch für die Bedienung

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

4.6.1 Beschreibung der Bedienung als User

Die Bedienung von Chatbot ist eigentlich sehr leicht. Ein normaler Benutzer ist in der Lage, Bilder zum Chatbot zu schicken und die danach in die Datenbank gespeichert und automatisch auf dem Bildschirm angezeigt.
Zuerst soll im Handy Telegram Applikation heruntergeladen werden. Danach soll nach dem Benutzername Infotainment gesucht werden. Der Chat mit dem Infotainment Bot wird geöffnet. Jeder Benutzer, die eine Interaktion mit Chatbot haben will, soll zuerst registrieren.
Sobald der Chatbot eine Nachricht von jemand, der für das erste Mal an ihm etwas schickt, wird der Chatbot ihm fragen ob er registrieren will oder nicht. Die Wahl kommt in Form von 2 Buttons, die die Antworten Ja und Nein beinhalten.
Für die Registrierung soll Ja geklickt werden und dann die Telefonnummer für die Registrierung soll auch eingegeben werden.
Danach soll gewartet, 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.

4.6.2 Beschreibung der Bedienung als Administrator

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

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

4.7 Evaluierung und Resümee

4.7.1 Planung vs Realisierung

4.7.2 Wertschöpfung und Lessons Learned

Kapitel 5

Umsetzung - Aldo Sheldija

In diesem Kapitel wird die Vorgehensweise der Technische Lösung bei der Admin Webseite und Client erklärt. Weiter werden auch die verwendeten Technologien aufgelistet.

5.1 Allgemeine Beschreibung

Dieser Diplomarbeit besteht aus mehreren unterschiedlichen Teilen, die am Ende zusammengeführt werden um das ganze Infotainmentsystem zu geben. Admin Webseite oder als Admin Panel genannt ist eine von diesen Komponenten. Bei dieser Webseite wird die ganze Verwaltung des Systems gemacht. Es werden alle Informationen geschrieben und gewählt, die bei verschiedenen Anzeigen dargestellt werden sollen. Anderseits ist die Anzeige oder als Client bezeichnet, indem alle diese Informationen angezeigt werden.

5.2 Technologien

Dieses Teil setzt sich mit der Beschreibung der Technologien, die für die Erstellung des Admin Webseite und der Client verwenden wurden auseinander. Es werden die Technologien erklärt und wozu die in dieses Projekt verwendet wurden.

5.2.1 Bootstrap

Bootstrap (Abb. 5.1) ist eine Open-Source und Kostenloses CSS[17] Framework, die für die Herstellung von unterschiedlichen responsiven Webseiten hilft. Dieses Framework ist nicht nur aus CSS basiert, sondern die enthält auch JavaScript, die die Interaktion von dynamischer Webseite erhöht. Bootstrap stellt fertige CSS Klassen zur Verfügung, die für unterschiedliche Bildschirmgrößen automatisch angepasst werden. Das wird durch ein Grid System ermöglicht. Bootstrap wurde verwendet um das Layout von der Admin Webseite zu machen. Da unsere Webseite eine Responsives Layout haben soll, ist Bootstrap verwendet. Mit Bootstrap wird das gesamte Layout dieser Webseite gemacht.

[17] Cascading Style Sheet

Abbildung 5.1: Bootstrap Logo[**40˙bootstrap˙img**]

5.2.2 PHP

PHP (Abb. 5.2) stand für "Personal Home Page Tools" und jetzt heißt PHP Hypertext Preprocessor. Die aktuellste Version von PHP ist 7.4.0, die im Dezember 2019 veröffentlicht. PHP ist eine Serverseitige Programmiersprache, die für die Erstellung von dynamischen Webseiten dient. Mit Hilfe von PDO[18] wird die Verbindung mit der Datenbank gemacht. PHP ist fast von alle Webhosting Services verfügbar und unterstützt. Durch PHP wird das ganze backend der Webseite gemacht. Mit PDO wird die Verbindung mit der Datenbank ermöglicht. PDO ist sicher von SQL-Injektionen und schneller als MySQLi.

Abbildung 5.2: PHP Logo[**40˙php˙img**]

5.2.3 JavaScript

JavaScript JavaScript oder auch kurz JS genannt, ist ein objektorientierte Programmiersprache. Diese Programmiersprache wird gerne bei der Erstellung von Webseiten verwendet, weil die Webbrowser eine JavaScript Engine implementiert haben.[**40˙js**]
In dieser Diplomarbeit wird die JavaScript gebraucht um die automatische Darstellung von unterschiedlichen Inhalten in der Anzeige darzustellen, ohne die Webseite neu zu laden.

Abbildung 5.3: JavaScript Logo[**40˙js˙img**]

[18]PHP Data Objects

5.2.4 DataTables

DataTables ist ein Plug-In für die jQuery-Javascript-Bibliothek. Es ist ein hochflexibles Tool, das auf den Grundlagen der fortschreitenden Verbesserung aufbaut und alle diese erweiterten Funktionen zu jeder HTML-Tabelle hinzufügt. DataTables bietet sehr viele Vorteile, wenn es zu den Tabellen geht wie: schnelleres und sofortiges Suchen, Seitennummerierung, Fast jede Datenquelle benutzen, Leichte Integration mit Bootstrap, freie Open Source Software, Mobilfreundlich und Export in den verschiedenen Formaten. Alle Tabellen werden mit Hilfe von diesem Plug-In erstellt.[**40˙datatables**]

Abbildung 5.4: DataTables Logo[**40˙datatables˙img**]

5.2.5 Font Awesome

Font Awesome(Abb. 5.5) ist ein Toolkit für Schriftarten und Symbole, das auf CSS und LESS basiert. Es wurde von Dave Gandy für die Verwendung mit Twitter Bootstrap erstellt und später in die Bootstrap-CDN aufgenommen. Mit Font Awesome erhaltet man skalierbare Vektor-Icons, die sofort angepasst werden können. Die Änderungen, die gemacht werden können sind in Bezug auf Größe, Farbe, Schlagschatten und alles, was mit CSS möglich ist.

Abbildung 5.5: Font Awesome Logo[**40˙font˙awesome˙img**]

5.2.6 MySQL

MySQL[19]Für die Relationale Datenbank Management System wurde MySQL (Abb. 5.6) gewählt. Dieses Datenbankverwaltungssystem wird gerne für kleine Projekte verwendet, weil es Open-Source und kostenlos ist. MySQL ist eine Software um Datenbank zu erstellen und zu verwalten. MySQL ermöglicht kontrollierten Zugriff auf die Datenbank. Die Daten werden auf Tabellen Struktur gespeichert, die in Dateien auf der Festplatte gespeichert sind. Diese Software ist kompatibel auf mehrere unterschiedlichen Plattformen wie z.B. Windows, Linux, MacOS usw.

[19]My Structured Query Language

Abbildung 5.6: MySQL Logo[**40˙mysql˙img**]

5.2.7 Apache

Das Apache HTTP Server-Projekt ist ein Versuch, einen Open-Source-HTTP-Server für moderne Betriebssysteme wie UNIX und Windows zu entwickeln und zu warten. Ziel dieses Projekts ist es, einen sicheren, effizienten und erweiterbaren Server bereitzustellen, der HTTP-Dienste synchron mit den aktuellen HTTP-Standards bereitstellt. Dieser Server verwendet die Port 80 für HTTP und Port 443 für HTTPS. Die erste Version wurde im Jahr 1995 und der letzte stabile Version ist 2.4.41, der im August 2019 veröffentlicht wurde.[**40˙apache**]

Abbildung 5.7: Apache HTTP Server Logo[**40˙apache˙img**]

5.3 Structed Software Design für den Client

Mithilfe von Structed Design wurde der Planung der Client gemacht. Diese Structed Design ist auf der Abbildung 5.8 dargestellt. Dieser Teil ist verantwortlich für die automatische Darstellung der Informationen in der Anzeige. Der Client kommuniziert mit der Datenbank, die im Server liegt und von dem die Daten geholt werden. Die wichtigste Funktionsblöcke dieser Structed Design sind Zeitsteuerung, Anzeige Prepare und das Log, die in den kommenden Teilen weiter erklärt werden.

5.3.1 Start

Beim Hochfahren des Systems bei dem Client wird ein Signal geschickt um das Programm zu starten. Dieses Signal ist mit Init bezeichnet. Dieser Funktionsblock wird dann die Grund Initialisierung machen. Es wird versucht eine Verbindung mit der Datenbank aufzubauen. In der Datenbank wird geschaut ob diese Anzeige schon registriert ist. Falls das nicht der Fall ist wird diese Anzeige registriert und die Anzeige kann die Daten bekommen. Start schickt auch ein Signal an der Zeitsteuerung, damit er beginnen zu arbeiten kann. Sobald diese Schritte fertig sind wird der Startblock fertig sein und das Programm läuft weiter.

5.3.2 Zeitsteuerung

Zeit Steuerung wird gestartet, wenn das Signal von Start gekommen ist. Nachdem Signal wird dieser Block gestartet. Dieser Block schaut wann soll er die neuen Informationen von der Datenbank bekommen und mach eine Anfrage. Dieser Teil wird immer im laufen sein, damit der Anzeige die Aktuellste Informationen beinhaltet.

5.3.3 Daten holen

Daten holen, wie der Name schon sagt beschäftigt sich mit dem Erhalt der Datensätze aus der Datenbank. Es werden die Queries aufgebaut um die gewünschten Daten zu bekommen. Diese Query werden weiter zu dem Daten, die auf dem Server liegen, weitergeleiten. Die Response wird auch von diesem Teil gehandelt. Die Daten werden in dem richtigen Format umgewandelt und weitergeleitet.

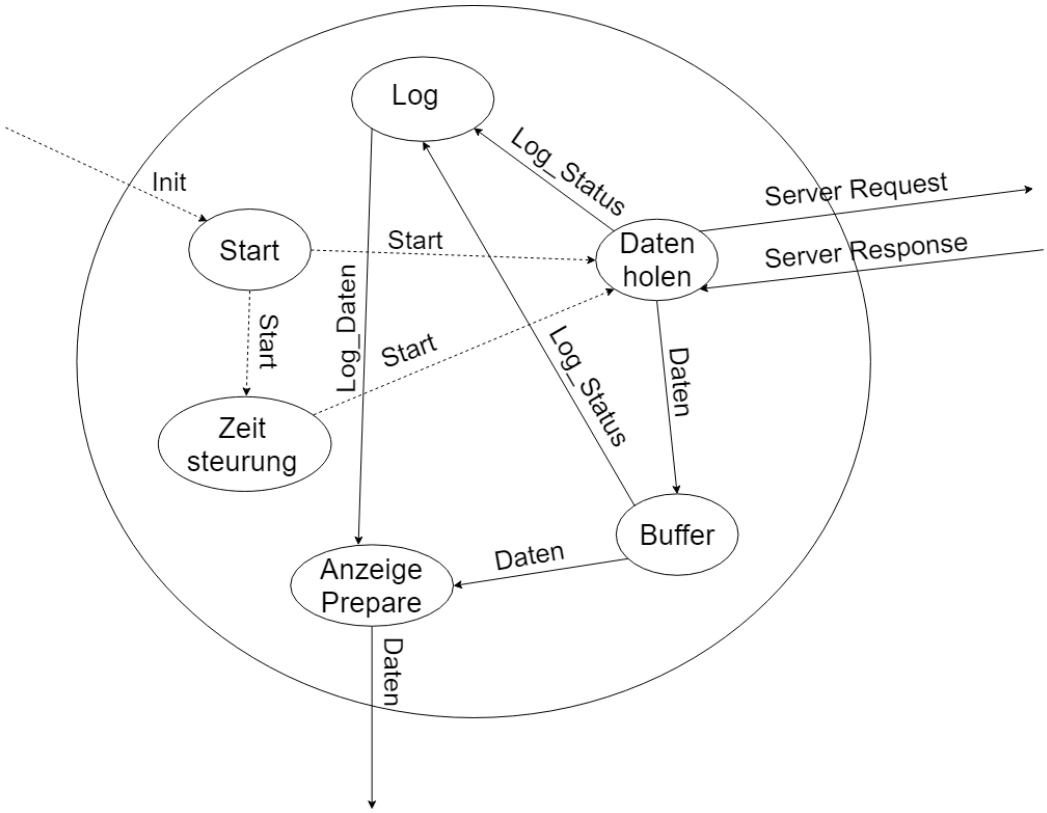

Abbildung 5.8: Structed Software Design für den Client

5.3.4 Buffer

Die Aufgabe dieser Teil, ist es die Daten, die er von dem vorherigen Block bekommen hat, zu schauen ob sie in die Lokale Datenbank gespeichert sind. Wenn diese Daten neu und nicht drinnen sind, werden die gespeichert. Ansonsten werden sie zu die Anzeige Prepare geschickt.

5.3.5 Anzeige Prepare

In dieser Funktionsblock findet die Umwandlung der Daten für die Richtige Darstellung statt. Die Daten werden gezielt für diese Anzeige umgewandelt und angepasst. Die Informationen sind in den passenden GUI[20] Bereich hinzugefügt. Diese Daten werden dann im Bildschirm angezeigt.

5.3.6 Log

In Log werden alle Ereignisse, die in alle Blöcke, die vorher erwähnt wurden, auftreten mitprotokolliert und gespeichert. Egal ob die Nachrichten erfolgreich oder nicht gekommen sind, werden alle diese in eine einzelne Protokoll Datei geschrieben. So ist es möglich jederzeit die Fehler, die in das System aufgetreten sind, anzuschauen und mit diesem Fehler umgehen.

5.4 Datenbank

Wie jede Website benötigt auch das InfotainmentSystem eine Datenbank, in der die Daten gespeichert und gesichert werden können. Die Datenbank besteht aus Entitäten, Beziehungen und Attributen. Eine Datenbank ist ein elektronisches Verwaltungssystem, das besonders bei großen Datenmengen effizient, widerspruchsfrei und dauerhaft umgehen und logische Zusammenhänge digital abbilden können muss. Diese Informationen sind in Zeilen, Spalten und Tabellen organisiert und werden indiziert, um das finden relevanter Informationen zu erleichtern.

5.4.1 ER Diagramm

Zuerst wurde ein ER-Diagramm[21] (Abbildung 5.9) für die Datenbank erstellt. Das ER-Modell bedeutet Entity-Relationship-Modell. Bevor die Datenbank erstellt wird, muss sie zuerst konzipiert werden. Zu diesem Punkt gehört die Erstellung eines ER-Diagrammes. Um ein ER-Diagramm zu erstellen, braucht man die Hilfe von MYSQL-Workbench. Dazu werden im Detail alle Entitäten mit ihren dazugehörigen Attributen und Relationen zwischen einander geschrieben. Es besteht aus einer Grafik und einer Beschreibung der darin verwendeten Elemente, wobei deren Bedeutung und ihre Struktur dargestellt wird. Grundlage der Entity-Relationship Modelle ist die Typisierung von Objekten, ihrer Beziehungen untereinander und der über sie zuführenden Informationen beziehungsweise Attribute. Hier in dieser Diplomarbeit werden alle Tabellen, die wir für die Datenbank benötigen, gezeigt. Von diesem ER-Diagramm wurde ein SQL-Skript erstellt und im Raspberry Pi ausgeführt.

[20] Graphical User Interface
[21] Entity Relationship Diagram

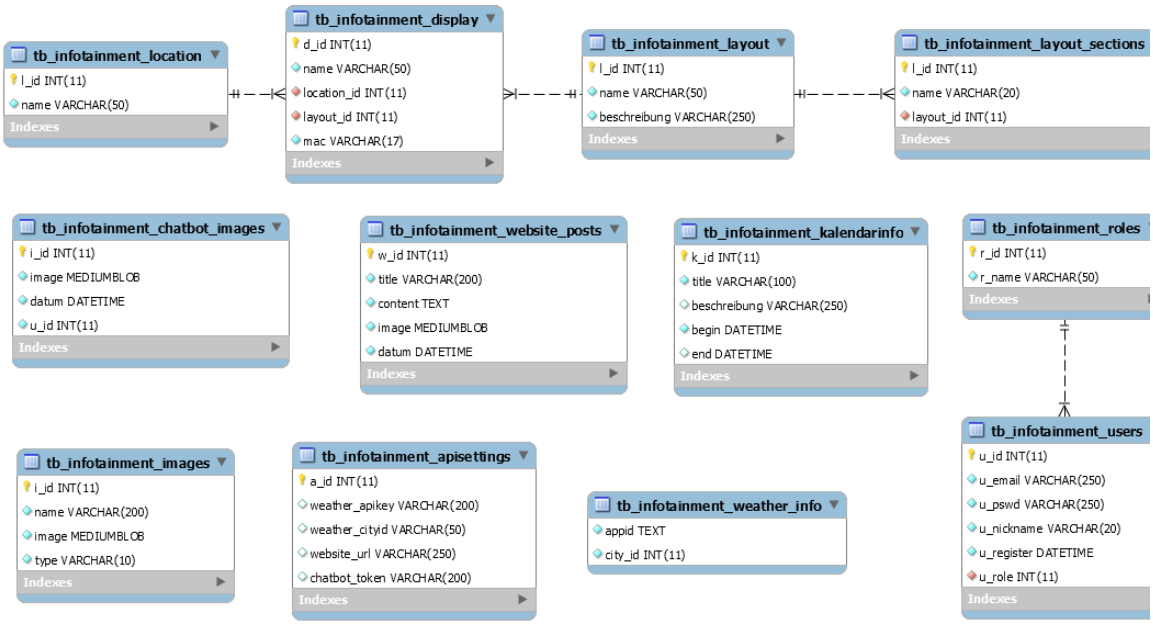

Abbildung 5.9: ERD Diagramm

5.5 Admin Webseite

Ein sehr wichtiger Teil dieser Diplomarbeit ist auch die Admin Webseite. In dieser Webseite hat der Admin die Möglichkeit, alle Anzeigen zu verwalten und denen mit unterschiedlichen Inhalten zu befüllen. Durch diese Webseite wird die Verwaltung einfacher und übersichtlicher gemacht. Dort wird die Gelegenheit zu dem Admin gegeben, Anzeige hinzufügen, ändern und löschen, verschiedenen Layouts für unterschiedliche Anzeigen anzuordnen, Bilder zu verwalten und Benutzer zu administrieren. Ein anderes Feature zu dieser Webseite ist auch der Supplierplan. Hier kann der Stundeplan für bestimmte Tagen gemacht werden, damit er in die Anzeige dargestellte werden kann. Es wurde so gemacht, dass der Admin nur in ein paar Klicks das erledigen kann. Da unser Diploma nicht nur aus dieser Webseite besteht, wird unsere Webseite eine Schnittstelle sein, um die API-Key von Wetterbericht, URL der Webseite, von dem die letzten Posts gespeichert werden, und Chatbot Token zu verwalten.
Diese Webseite wurde so konzipiert, dass der Benutzer, der das verwenden wird, die Informationen und die Bedienung dieser Webseite sehr leicht und ohne Probleme folgt. Die Website ist auch in mehrere Sprachen angebetet, so dass noch leichter wird.
Wie gesehen, der Inhalt in dieser Webseite ändert ständig und deshalb ist unsere Webseite, eine dynamische Website, die mit PHP und MySQL gemacht wird. Für die Darstellung und das Design wurde Bootstrap verwendet. Um die Tabellen am besten darzustellen wurde auch Datables, ein JavaScript Plugin, verwendet. Regex wurde verwendet um die Validierung von unterschiedlichen User Input zu überprüfen und sicherstellen, dass der Input richtig ist.
Damit nicht jeder diesen Inhalten ändern und löschen kann, ist diese Webseite auch mit einen Login Bereich geschützt. Nur bestimmte Personen, abhängig von welchem Rechten sie haben können dann die zugehörigen Änderungen machen. Falls ein Be-

nutzer das Passwort vergisst, hat er die Möglichkeit das Passwort zurückzusetzen und wartet bis der Admin das bestätigt hat.
Der Zugriff auf dieser Webseite ist auf jedes Gerät und von jedem Ort aus möglich. Um diese Webseite anzuschauen ist nur ein Internet Verbindung notwendig.

5.5.1 Login

Damit nicht alle unsere Webseite verwenden können, müssen sie die Benutzer sich auf der Webseite einloggen (Abbildung 5.10). Die Admin Webseite ist durch eine Login Seite geschützt, die dem Benutzer folgende Informationen anfordert: die E-Mail-Adresse und das Passwort. Dieses Formular heißt Login.php. Wenn der Benutzer auf der Sign-in Button klickt werden seine Anmeldedaten überprüft ob sie korrekt sind. Ist das der Fall wird der Benutzer eingeloggt. Wenn der User in der Datenbank nicht existiert, wird ihm ein Error Meldung gezeigt damit er weißt, dass er zuerst registriert sein muss um in die Webseite einloggen zu können. Wenn der Benutzer bei der Erfüllung des Formulars das Passwort falsch eingibt, aber die E-Mail-Adresse ist richtig und der Benutzer existiert, wird ihm die Error Meldung für die Falsche Passwort angezeigt. So ist dieser Login sehr einfach und Benutzerfreundlich. Bei dieser Seite findet man auch einen Link um das Passwort zurückzusetzen.

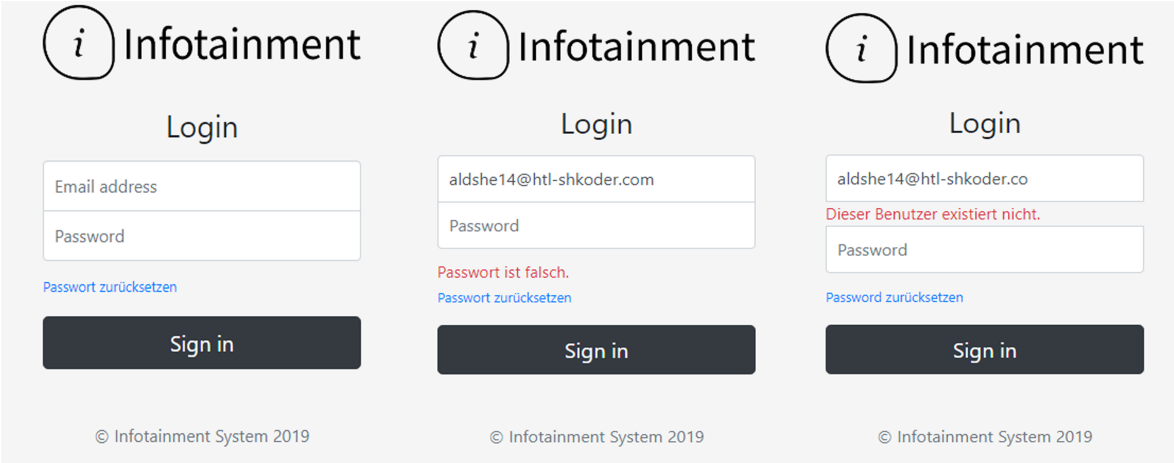

Abbildung 5.10: Anmelden Seite

5.5.2 Passwort Zurücksetzen

Ab und zu passiert, dass die Benutzer das Passwort vergessen, egal auf welche Gründe. So muss diese Webseite die Möglichkeit anbieten das Passwort zurückzusetzen .Das ist auf die Abbildung 5.11 dargestellt Der Benutzer muss das E-Mail zwei Mal schreiben und auf Zurücksetzen draufklicken. Wenn die E-Mail-Adressen nicht überein stimmen wird ein Error Meldung angezeigt. Stimmen die E-Mail-Adresse überein, wird eine Eintrag auf der Datenbank gespeichert damit der Admin das zurücksetzen kann.

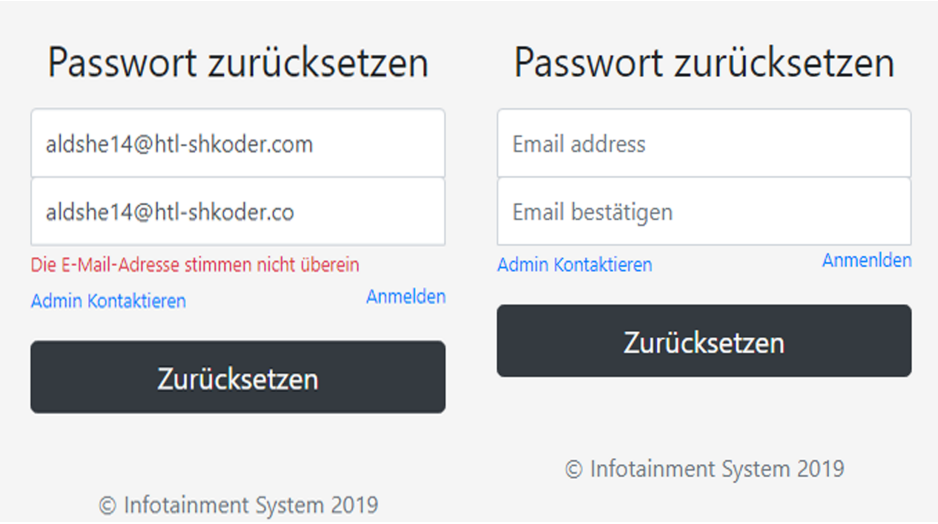

Abbildung 5.11: Passwort zurücksetzen

5.5.3 Menü

Das Menü wird ganz oben als Header sein. Für das Das Menü ist unterschiedlich für den Admin und für normale Users. Der Admin hat mehrere Möglichkeiten zu wählen und das ist auf der Abbildung 5.12 ersichtlich. Normaler Users werden weniger Wahl Möglichkeiten haben und die haben auch beschränke Zugriff (Abb. 5.13). Die Seiten Settings und Users werden nur für Benutzer mit Adminrechte erreichbar.

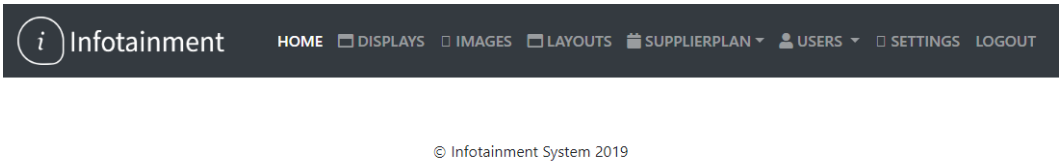

Abbildung 5.12: Menu der Admin Panel

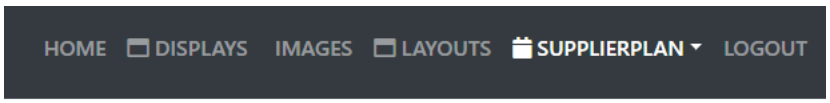

Abbildung 5.13: Menu für normaler User

5.5.4 Rechte in die Webseite

Wie es vorher erwähnt wurde, wird die Webseite unterschiedliche Benutzer Rechte haben. Die zwei Hauptrollen sind Admin und normaler User. Die unterschied zwischen den beiden Benutzern wird von die Rolle Attribut in der Tabelle Rolle gemacht (Siehe Abb. 5.9). Diese Attribute werden in den Sessionen gespeichert. Es wurden Sessions gewählt, weil sie sicherer als Cookies sind.[**40˙sessions˙cookies**]

```
$_SESSION['user_id'] = $user['u_id'];
$_SESSION['loggedin'] = true;
$_SESSION['email'] = $user['u_email'];
$_SESSION['role'] = $user['u_role'];
```

Listing 5.1: Speicherung der Benutzer Informationen in Sessions

5.5.5 Webseite auf mehrere Sprachen

5.6 Client

Zu diese Diplomarbeit gehört auch der Client dazu. Als Client wird bezeichnet die Anzeige, in dem die Informationen dargestellt werden. Der Client besteht aus einem RaspberryPi, der mit Internet verbindet wird (WLAN[22] oder LAN[23] Verbindung), und mit dem Server kommunizieren wird. Der RaspberryPi hat einen HDMI[24] Anschluss, der mit einem Bildschirm verbindet wird. Um das System mehr kompatibel und umfassend, werden unterschiedliche Bildschirmgrößen unterstützt. Der Anzeige wird automatisch registriert in der Datenbank mithilfe von einem Skript, der die Informationen beim Hochfahren des Geräts zu Datenbank schickt. Diese Anzeige wird beim Hochfahren des Betriebssystems die Webseite von der Anzeige öffnen und das Anzeigen lassen. Diese Webseite wird im Vollbild geöffnet. Zusätzlich werden die Bildschirmaustastung, Bildschirmschoner und Energieverwaltungssystem ausgeschaltet, damit die Anzeige immer eingeschaltet bleibt. Damit das System nicht abhängig von dem Internet ist, werden die Informationen für diese Anzeige von der Datenbank, die im Server liegt, kopiert und auf der lokalen Datenbank gespeichert.
In dem Client laufen eine Apache Server, MySQL Datenbank und PHP, die für die Anzeige notwendig sind um die Informationen anzuzeigen. Falls der Anzeige noch nicht von der Admin freigegeben wurde, wird die Anzeige zu eine Welcome Seite umgeleitet, in dem nur die MAC-Adresse der Anzeige dargestellt wird.

5.6.1 Layout 1

Die Anzeige wird unterschiedliche Layouts beinhalten. Eine von denen ist der Erste Layout. Dieses Layout wird in 5 Bereiche unterteilt. Header – Hier wird die Logo platziert zusammen mit der aktuellen Uhrzeit und Datum.

- *Body* – Es können unterschiedliche Komponnenten dargesttelt werden wie: Supplierplan, Stundenplan usw.
- *Widget 1* – Hauptsächlich für das Wetter und Post der Webseite geeignet. Hier können auch die Kalenderinformationen angezeigt werden.
- *Widget 2* – Es können dieselben Komponenten wie bei der 1. Widget angezeigt.

[22] Wireless LAN
[23] Local Area Network
[24] High Definition Multimedia Interface

- *Footer* – Unterschiedliche Texte in Form einer beweglichen Text.

Auf der Abbildung 5.14 wird ein mögliches Beispiel dargestellt.

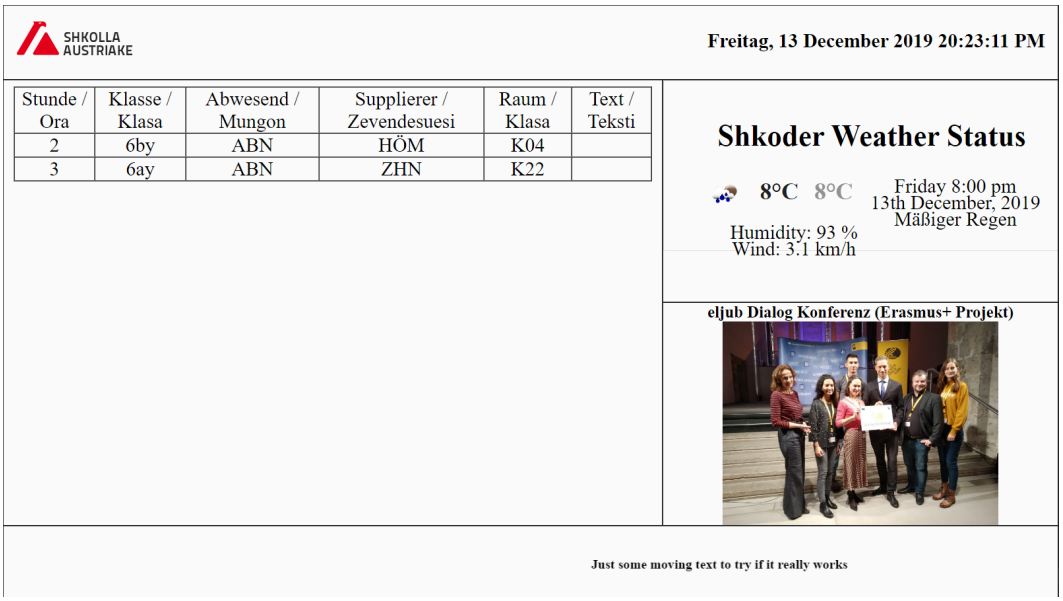

Abbildung 5.14: Layout 1 der Anzeige

5.7 Wordpress Webseite API

WordPress wird heutzutage sehr viel verwendet um Webseite zu bauen. Die Erleichterungen der WordPress bieten, ermöglicht auch die Benutzer, die keine Programmierkentnisse haben, Webseite zu erstellten und zu verwaltet Durch Plug-Ins kann der WordPress erweitert und unterschiedliche Funktionalitäten eingefügt werden. Jetpack ist ein von denen, der zur Sicherheit, Leistung und Site-Management hilf. Dieses Plug-In bietet auch eine API[25] zum WordPress. Dieser API wurde verwendet um die Post einer bestimmte seite zu bekommen. Es muss eine nachfrage and der unteren URL geschickt.

http://<domainname>/wp-json/wp/v2/posts

Die Response ist in JSON Format und die wird dann mit Hilfe von PHP geparst und in der Datenbank gespeichert.

5.8 Herausforderungen und deren Lösung

Eine der größten Herausforderungen war beim Supplierplan getroffen. Als vorher schon erwähnt wurde, das Supplierplan dient darum, die Lehrer die fehlen mit anderen Lehrern substituieren bzw ersetzen. In diesem Bezug kann es sein, dass ein Lehrer bzw eine Lehrerin nicht während den ganzen Tag fehlen wird. Es gibt Situationen und Tagen,

[25] Application Programming Interface

in dem ein/e Lehrer/Lehrerin nur für eine bestimmte Zeit fehlt. Beispielweise haben sie etwas Wichtiges zu tun, dass aber nur 2 Unterrichtsstunde dauert. In diesem Fall muss es gesichert werden, dass dieser Lehrer von einem Kollegen während dieser Zeit substituiert wird.
Um dieses Problem zu lösen und um den Admin die Möglichkeit zu geben, ein Lehrer nur für eine bestimmte Stundenanzahl zu ersetzen, wurde ein kleine Checkbox erstellt. Es geht um eine Dropdown-Liste, bei der alle Lehrern die bereit zum supplieren sind stehen. Um es je einfacher wie möglicher zu machen, werden diese Lehrern sortiert nach der besten Möglichkeit und Günstigkeit für sie. Zum Beispiel wenn ein Lehrer in der 3.Stunde fehlt, dann steht ein supplierbereiter Lehrer, dass erst um 4.Stunde in der Schule kommen musst, höher bzw. vorher in der Dropdown-Liste als ein Lehrer, der wegen seines normalen Stundenplans in der 8. Stunde Unterricht hat. Je näherer die normale Unterrichtsstunde eines Lehrers mit der zu supplierten Stunden ist, desto höher steht er in der Liste.

Abbildungsverzeichnis

2.1 Projektstrukturplan 5
2.2 Structed Design - Client 9
2.3 Structed Design - Server 10
2.4 Wasserfallmodell 11

4.1 Server 29
4.2 ERD 30
4.3 Auflistung der Chatbot-Benutzer 37
4.4 Auflistung der unregistrierten Benutzer 38
4.5 Benutzer Blockierung 39
4.6 Registrierung beim Chatbot 40
4.7 Falsche Eingabe des Telefonnummers beim Chatbot 41
4.8 Falsches Format des Telefonnummers beim Chatbot 42
4.9 Eingabe einer falschen Telefonnummer beim Chatbot 43
4.10 Korrekte Eingabe der Telefonnummer beim Chatbot 44
4.11 Das Warten der Benutzer auf die Administrator Bestätigung 45
4.12 Nachricht von einem blockierten Benutzer 46

5.1 Bootstrap Logo[**40˙bootstrap˙img**] 50
5.2 PHP Logo[**40˙php˙img**] 50
5.3 JavaScript Logo[**40˙js˙img**] 50
5.4 DataTables Logo[**40˙datatables˙img**] 51
5.5 Font Awesome Logo[**40˙font˙awesome˙img**] 51
5.6 MySQL Logo[**40˙mysql˙img**] 52
5.7 Apache HTTP Server Logo[**40˙apache˙img**] 52
5.8 Structed Software Design für den Client 53
5.9 ERD Diagramm 55
5.10 Anmelden Seite 56
5.11 Passwort zurücksetzen 57
5.12 Menu der Admin Panel 57
5.13 Menu für normaler User 57
5.14 Layout 1 der Anzeige 59

Tabellenverzeichnis

2.1 Meilensteine 4
2.2 Arbeitspakete 8

4.1 Technologien 23
4.2 Bereiche und Technologien 28

Quellcodeverzeichnis

4.1 Chatbot 1 34
4.2 Chatbot 2 34
4.3 Chatbot 3 35
4.4 Chatbot 3 35
4.5 Chatbot 4 35
4.6 Chatbot 5 35
4.7 Chatbot 6 36
5.1 Speicherung der Benutzer Informationen in Sessions 58

Literatur