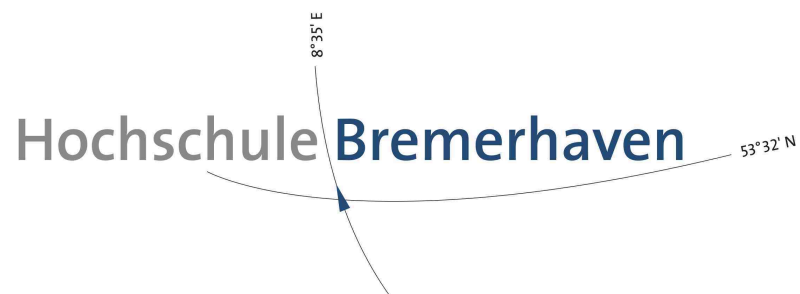


Bachelorprojekt: Pepper



Eingereicht von:

Benjamin Thomas Schwertfeger 36036

Kristian Kellermann 35751

Jacob Menge xxxxx

Kursleitung: Prof. Dr. Nadja Petram

Technik für Wirtschaftsinformatik
Hochschule Bremerhaven

Eigenständigkeitserklärung

Hiermit erklären wir, dass wir die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben.

Alle sinngemäß und wörtlich übernommenen Textstellen aus fremden Quellen wurden kenntlich gemacht.

Name: _____ Unterschrift: _____
Ort, Datum: _____

Name: _____ Unterschrift: _____
Ort, Datum: _____

Name: _____ Unterschrift: _____
Ort, Datum: _____

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Abstrakt	iv
1 Dokumentation und Struktur	1
2 Motivation und Zielsetzung	3
3 Erste Schritte: Das Grundgerüst	4
3.1 Allgemein	4
3.2 Wer ist dieser Pepper?	4
3.3 Projektmanagement	4
3.4 Persona	5
3.5 Datensammlung	6
4 Einrichtung des Laptops	8
4.1 Technische Details	8
4.2 Allgemein	8
5 Anwendungsfall - Hochschule	10
5.1 Die Vorüberlegungen	10
5.2 Hard- und Softwarespezifikationen	11
5.3 Implementierung	11
6 Anwendungsfall - XXX	12
6.1 Allgemein	12
6.2 Die Herkunft der Daten	12
6.3 Erläuterungen zum Quellcode	12
7 Das Backend	13
7.1 Allgemein	13
7.2 Implementierung	13
7.2.1 NodeJS, npm, Express	14
7.2.2 Endpunkte	14
7.3 Installation	14
8 Abschließende Worte	15

Abbildungsverzeichnis

Abstrakt

Diese Dokumentation ist im Rahmen des Bachelorprojektes von Benjamin T. Schwertfeger, Jacob Menge und Kristian Kellermann entstanden und dient neben der Dokumentation des selbst gewählten Projektes ebenfalls als Anforderung für das Bestehen des Bachelorstudiums.

Die Studenten haben sich dazu entschieden, eine Anwendung für den humanoiden Roboter Pepper, welcher sich derzeit im Besitz der Hochschule Bremerhaven befindet zu entwickeln. Hierzu werden verschiedene Anwendungsfälle implementiert, welche durch eine selbst implementierte Web-Schnittstelle in Form einer Backend Webanwendung unterstützt wird.

Ziel ist es, dem Roboter das Interagieren mit Menschen beizubringen, sodass sinnvolle Interaktionen zwischen Mensch und Maschine zustande kommen. Zudem sollen mit Hilfe der Webanwendung zusätzlich dynamisch Informationen an Pepper weitergegeben und von ihm abgerufen werden, sodass Informationen zu Interaktionen, wie Dauer, Intensität und Erfolg des Gespräches gespeichert und ausgewertet werden können.

Dieses Projekt wird von Frau Prof. Dr. Nadja Petram begleitet. Es wurden keine Rahmenbedingungen festgelegt.

Kapitel 1

Dokumentation und Struktur

Diese Dokumentation ist in XXX Kapitel gegliedert, welche sich mit den einzelnen Aspekten unseres Projektes auseinandersetzen. Die Stuktur sieht wie folgt aus:

- *Kapitel 1*: Dokumentation und Struktur
- *Kapitel 2*: Motivation und Zielsetzung
- *Kapitel 3*: Erste Schritte und Installation
- *Kapitel 4*: Anwendungsfall - Hochschule
- *Kapitel 5*: Anwendungsfall - XXX
- *Kapitel 6*: Das Backend
- *Kapitel 7*: Möglichkeiten der Erweiterung
- *Kapitel 12*: Abschließende Wort

Zu sehen ist, dass wir nach der Erläuterung unserer Struktur und des Grundes für diese Arbeit von vorn bis hinten durch unser Projekt gehen. Nachdem wir unser Grundgerüst und unsere ersten Schritte einschließlich Installation dargelegt haben, werden wir zwei Anwendungsfälle zum Einsatz von Pepper Aufzeigen, sowie die von uns Implementierten Funktionen aufzeigen.

Ebenfalls werden wir die Herangehensweise, sowie die technische Umsetzung und Hürden, mit denen wir zu kämpfen gehabt haben erläutern, um alle Aspekte der Implementierung von Software für den Roboter Pepper abzudecken.

Nachfolgend haben wir die Fragestellungen festgehalten, auf welche wir in den jeweiligen Abschnitten eingehen werden.

Kapitel	Fragestellung
1. Dokumentation und Strktur	Wie ist diese Arbeit aufgebaut?
2. Motivation und Zielsetzung	Was wollen wir hiermit erreichen?
3. Erste Schritte und Installation	Was ist das Fundament unseres Softwareprojektes?
4. Anwendungsfall - Hochschule	Wie kann Pepper der Hochschule helfen?
5. Anwendungsfall - Hochschule	Was kann Pepper als Stadtführer?
6. Das Backend	Welche Prozesse laufen im Hintergrund?
7. Möglichkeiten der Erweiterung	Wie kann es weiter gehen?
12. Abschließende Worte	Was haben wir aus diesem Projekt mitgenommen?

Im Anschluss dieser Dokumenation ist das Literaturverzeichnis mit allen Referenzen zu finden.

Kapitel 2

Motivation und Zielsetzung

Wie alle Bachelorstudierende kommt irgendwann die Zeit, in der es darum geht, ein Projekt zu finden, mit welchem man sich über einen längeren Zeitraum beschäftigt, um dies im Rahmen des Studiums schriftlich niederzulegen. Wir drei sind von der Welt der Programmierung begeistert und haben von Frau Dr. Petram das Angebot erhalten, bei ihr das Bachelorprojekt durchzuführen. Da wir im Laufe unseres Studiums viele Kurse bei ihr besucht haben, unter anderem “KI - maschinelles Lernen“ und “Big Data“ waren wir sofort begeistert, als wir gehört haben, mit dem humanoiden Roboter Pepper arbeiten zu dürfen.

Wir haben von ehemaligen Masterstudierenden eine Vorführung einer laufenden Anwendung bekommen und haben uns natürlich direkt gefragt, was wir denn tolles entwickeln können, damit nicht nur wir, sondern auch die Hochschule den meisten Mehrwert davon erhalten.

Aufgrund dieser Fragestellung haben wir uns dazu entschieden, dass wir zwei Anwendungsfälle zum Einsatz des Peppers implementieren wollen, um diese auch mit Hinblick auf den Tag der Informatik vorstellen zu können.

Neben der Entwicklung dieser Anwendungsfälle wollen ebenfalls unser Verständnis für die Thematik KI und Big Data ausbauen, indem wir ein Backend schaffen, welches mit Hilfe von Pepper Daten sammelt, um diese in einem späteren Prozess anderweitig zu verwenden.

Kapitel 3

Erste Schritte: Das Grundgerüst

3.1 Allgemein

Wir haben bisher noch kaum Berührung mit Robotern im Alltag erlebt und daher ist unsere Erfahrung im Bereich der Roboterprogrammierung sehr begrenzt. Zum Glück haben uns ehemalige Masterstudierende einen schnellen Einstieg ermöglicht, denn diese haben eine Anwendung für das Unternehmen “Erlebnis Bremerhaven” entwickelt und uns deren Quellcode zur Verfügung gestellt.

Wir haben sehr schnell gemerkt, dass uns dies zwar eine gute Hilfe für den Anfang ist, jedoch gibt es mittlerweile deutlich effizientere Methoden einzelne Komponenten miteinander zu verknüpfen.

3.2 Wer ist dieser Pepper?

3.3 Projektmanagement

Um dieses Projekt erfolgreich durchzuführen und zielorientiert zu arbeiten, treffen wir uns wöchentlich in der Hochschule und besprechen unsere neuesten Ideen und Implementierungen. Auch zwischendurch stehen wir im Kontakt, um Probleme und Schwierigkeiten schnell zu beheben.

Damit wir gemeinsam an einer Codebasis arbeiten können, haben wir zu Anfang unseres Projektes die Organisation [HBV-Pepper](#) auf [GitHub](#) angelegt. Jegliche Beteiligung, sowie verschiedene Versionen können dort eingesehen werden. Zum Ende des Projektes haben sich hier 4 Repositories von uns angesammelt, welche folgende Themen beinhalten:

- Pepper Anwendung
- Backend Serveranwendung
- Dokumentation

- Sonstiges

Das erste Repo beinhaltet unsere lauffähige Anwendung, welche die von uns implementierten Anwendungsfälle, sowie sonstige Funktionalitäten für den Roboter Pepper beinhaltet.

Der Backend Serveranwendung auf welchem wir in Kapitel `[HIER LINK SETZEN]` eingehen, ist unsere Schnittstelle, mit welcher wir Informationen, die Pepper durch seine Interaktionen mit der Umgebung sammelt, abfangen und speichern. Zudem Versorgen wir Pepper mit Informationen, Beispiel hierfür sind Informationen zum Stunden- und Mensaplan.

Das Repositorie zur Dokumentation beinhaltet alle zu diesem Dokument gehörenden Dateien. Da wir diese mit LaTeX anfertigen, lohnt es sich auch dies zu versionieren.

Für kleine Skripte, Tests und andere zu keinem festen Thema dazugehörigen Dokumente haben wir das Repositorie “Sonstiges“ angelegt.

Wir werden in diesem Dokument neben der Implementierung auch auf das Herunterladen, Installieren und Einrichten unserer Software eingehen, damit Studierende, die dieses Projekt weiterführen wollen, die Möglichkeit haben, mit einem etwas fortgeschrittenem Projekt zu starten. Darüber hinaus befindet sich in jedem Stammverzeichnis eine Readme, welche Anforderungen und erste Schritte aufzeigen, um einen schnellen Einstieg zu ermöglichen.

3.4 Persona

Damit wir Pepper mit allen möglichen Antworten auf alle möglichen Fragen vorbereiten können, müssen wir uns zuerst überlegen, welche Personengruppe mit Pepper sprechen wird. Auf dieser Basis können wir verschiedene Gesprächsthemen entwickeln, die für die Personengruppe interessant sein könnten.

Da wir Pepper als Hochschul Assistenzen einsetzen wollen, haben wir es mit folgenden Personas zu tun:

- Student
- Dozent
- Angestellter der Hochschule
- Besucher (Zukünftige Studenten, Interessenten)

Hauptsächlich soll Pepper für Studenten und insbesondere für Erstsemestler eingesetzt werden, die zum Beispiel nicht wissen, wo sich ein bestimmter Raum oder ein Gebäude befindet oder wann und wo die nächste Vorlesung stattfindet. Hierbei soll es auch die Möglichkeit geben, den kompletten Stundenplan für jeden Studiengang und jedes (zur

Zeit verfügbare) Semester anzeigen zu lassen, um auf jede Situation eine Antwort parat zu haben.

Ein weiterer Faktor ist der Altersintervall mit dem Pepper zu tun haben wird. Ältere Studenten, die sich auch in einem höheren Semester befinden, sind Themen wie Raumfindung oder Stundenplan eher uninteressant. Dafür könnten Themen wie der Mensaplan oder Informationen über die Hochschule und über Pepper selber interessant sein. Außerdem wird Pepper in der Lage sein, mit dem User ein kurzes Smalltalk Gespräch zu führen. Somit ist Pepper auch für alle Altersgruppen interessant. Diese Themen könnten für alle weitere Personas ebenfalls interessant sein. Vor allem Personen, die technikaffin sind oder das Thema Roboter einfach spannend finden, werden Pepper ebenfalls verwenden, weswegen wir für diese Personengruppe passende Gesprächsthemen und lustige Animationen vorbereiten.

Für Dozenten und Angestellte der Hochschule sind Themen wie der Stundenplan und die Raumfindung eher uninteressant und der Mensaplan wieder interessanter. Diese beiden Personas werden Pepper sehr wahrscheinlich am wenigsten verwenden.

Für Besucher sind Themen wie Informationen über Studiengänge und allgemein zur Hochschule eher interessanter als alle anderen Themen.

Um genau sagen zu können, welche Personengruppe sich mit Pepper unterhalten und über welche Themen sie sprechen, müssen wir Pepper zuerst sehr oft in der Praxis einsetzen und verschiedene Daten sammeln. Dazu werden wir im Punkt “Datensammlung” näher eingehen.

3.5 Datensammlung

Sobald alle Anwendungsfälle in Pepper integriert sind und er für den Praktischen Nutzen einsatzbereit ist, haben wir uns überlegt, verschiedene Daten zu sammeln um eine Art Feedback zu erhalten und diese graphisch darzustellen.

Folgende Daten sollen gesammelt werden:

- Distanz
- Alter
- Geschlecht
- Emotion/Stimmung
- Mimic
- Dialog Zeit
- ...

Diese Daten sollen während des Gesprächs zwischen Pepper und dem User gesammelt werden und zu unserem Node Server gesendet werden. Auf dem Node Server werden diese anschließend weiter verarbeitet und vorerst in einer MySQL Datenbank abgespeichert.

Um die Daten sammeln zu können, bietet Softbanks praktischerweise verschiedene KI orientierte Funktionen an, wie zum Beispiel die Gesichts,- oder Alterserkennung.

Diese Daten sollen dafür eingesetzt werden, um Pepper in Zukunft weiter zu optimieren und folgende Fragen zu beantworten:

- War das Gespräch erfolgreich und konnte Pepper helfen?
- Hatte der User Freude bei der Benutzung des Roboters?
- Hatte der User Schwierigkeiten, sich mit dem Pepper zu unterhalten oder verlief alles reibungslos?
- Welche Personengruppe hat mit Pepper am meisten gesprochen?
- Wie lange verläuft ein Gespräch im Durchschnitt?
- Gab es Fragen, die nicht beantwortet werden konnten?
- Welcher Anwendungsfall wurde wie oft verwendet?
- Gab es Systemfehler?
- Was ist die optimale Ansprech Distanz zwischen User und Roboter?
- Welches Semester und/oder Studiengang ist der User

Kapitel 4

Einrichtung des Laptops

4.1 Technische Details

- Intel Core i9-11900K Prozessor (bis zu 5,3 GHz), Octa-Core
- 43,9 cm (17,3") Full HD 16:9 Display (entspiegelt), Webcam
- 32 GB RAM, 1.500 GB SSD, Fingerprint
- 1,5TB SSD
- NVIDIA GeForce RTX3080 Grafik (16384 MB), HDMI, Thunderbolt 4, MiniDP
- Windows 10 Professional 64 Bit, 4,6 kg

Link: https://www.cyberport.de/?DEEP=1C10-3HP&APID=276&gclid=CjwKCAjwhaakBhBcEiwA8acsHlgHRYDqBoC2FoQAvD_BwE

4.2 Allgemein

Zu Beginn des Projekts haben wir einen XMG Ultra 17,3 Laptop erhalten, den wir beliebig verwenden durften. Wir haben uns Gedanken darüber gemacht, wie wir den Laptop am Sinnvollsten verwenden können und welches Betriebssystem wir verwenden.

Wir haben uns dazu entschieden, den Laptop als Speicherort für die Daten zu benutzen, die wir mit Pepper generieren und speichern wollen. Außerdem soll der Laptop als gemeinsames System zur Kommunikation und Datentransfer dienen. Dafür war die Überlegung, ein Nextcloud Server zu implementieren und Open SSH zu verwenden um auch von außerhalb Zugriff zu gewährleisten. Des Weiteren hatten wir auch die Idee, den Laptop als Workstation zu benutzen, um gemeinsam in Android Studio zu arbeiten oder weitere Applikationen wie Python Anaconda zu nutzen.

Die Wahl des Betriebssystems fiel uns schwer, da jeder seine eigenen Vorlieben besaß. Am Ende konnten wir uns allerdings eindeutig entscheiden. Für den Nextcloud Server, sowie das Speichern der Daten in eine Datenbank haben wir Ubuntu Server verwendet.

Hier war auch die Überlegung die Desktop version zu verwenden, allerdings bräuchten wir diesen für unser Vorhaben nicht unbedingt und zur Not hätte man sich den Desktop später immer noch einrichten können. Der Vorteil an Ubuntu Server ist, dass man von Null anfängt und keine unnötigen Applikationen bereits Installiert sind, so wie es mit der Ubuntu Desktop Version der Fall ist.

Da wir Ubuntu Server nicht als Betriebssystem für die Workstation verwenden konnten haben wir uns entschieden eine zweite Partition zu erstellen und Windows 10 einzurichten.

Einrichtung der Betriebssysteme ...

Kapitel 5

Anwendungsfall - Hochschule

5.1 Die Vorüberlegungen

Nachdem wir uns darauf geeinigt haben, mindestens einen Anwendungsfall für den Roboter Pepper zu entwickeln, und sich dieser auf den Alltag an der Hochschule Bremerhaven beziehen soll, haben wir uns damit konfrontiert gesehen, uns Gedanken darüber zu machen, wie genau eine Interaktion zwischen einem Studenten oder einer Lehrkraft und dem Roboter stattfinden kann. Aufgrund der den aktuellen Beschlüssen der Regierung ist das Leben auf dem Campus fast zum Stillstand gekommen, wodurch sich für Pepper wenige Einsatzgebiete ergeben. Wir gehen jedoch davon aus, zu einer neuen Normalität zurück zu gelangen, und werden Pepper dahingehend vorbereiten.

Folgende Fragen sind für uns zentral:

- Wie sieht der Alltag auf dem Campus aus?
- Wann soll Pepper mit wem interagieren?
- Welche Funktionalitäten wollen wir implementieren?
- Wo setzen wir welche Werkzeuge ein?
- Wo fangen wir an?

Wir sind dann ziemlich schnell darauf gekommen, dass wir eine klare Struktur benötigen, um unsere Aufgaben und Ziele klar zu definieren. Daraufhin ist folgende Skizze entstanden, welche grob die Hauptfunktionalitäten unseres Projektes festhält:

HIER WIRKLICH EINE SKIZZE ANFERTIGEN

In Abb. ist zu sehen, dass wir folgende Grundfunktionalitäten definiert haben:

- Stundenplan

- Raum- und Lageplan
- Speisekarte der Mensa
- Smalltalk

Zudem soll die Interaktion mit Pepper für alle möglich sein. Studierende und Lehrende sollen erfragen können, wo welche Vorlesung stattfindet, in der Kaffeteria oder Mensa soll sich Pepper mit Gästen unterhalten, Smalltalk führen oder Empfehlungen für bestimmte Speisen ausgeben. Darüber hinaus wollen wir Pepper das lückenlose Interagieren beibringen, damit er an besonderen Veranstaltungen, wie dem Tag der offenen Tür oder dem Tag der Informatik, Interessenten Informationen über die Hochschule, das Leben auf dem Campus, sowie allgemeine Empfehlungen für die Stadt Bremerhaven ausgeben kann.

5.2 Hard- und Softwarespezifikationen

Da Peppers Tablet auf Android basiert, liegt es auf der Hand, Android Studio als Entwicklungsumgebung zu benutzen. Der Einfachheit halber, ist nachfolgend eine Tabelle, welche die von uns für die Pepper Anwendung benutzte Software mit den entsprechenden Versionsnummern auflistet.

TABELLE 5.1: Hard- und Softwarespezifikationen

Software / Tool	Version / Spezifikation	Beschreibung
Android Studio	Arctic Fox 2020.3.1 Patch 4	IDE der IntelliJ Plattform
Gradle	7.0.3	Build Tool
Android SDK	12	Android Framework
Pepper API	v1.9 API 23	Entwicklungstools für Pepper mit Emulator, Deploy und Debug Funktion
Android API SDK	31	Framework zum Verbinden und Installieren von Software auf Geräten mit Android OS
Java SDK	1.8	Basis der Programmiersprache Java und dessen Grundfunktionalitäten

Mit abweichenden Versionen kann es zu Kompatibilitätsproblemen kommen.

Der Roboter Pepper selbst, ist XXX groß, wurde am XXX von XXX gekauft und ist seit XXX im Besitz der Hochschule und läuft derzeit mit der Version XXX.

5.3 Implementierung

Kapitel 6

Anwendungsfall - XXX

6.1 Allgemein

Wir haben im Kapitel `¡HIER KAPITEL EINFÜGEN!` die Soft- und Hardwareanforderungen definiert und werden dies für diesen Anwendungsfall nicht noch einmal durchführen, da auch dieser Use-Case in der selben Applikation, integriert ist.

6.2 Die Herkunft der Daten

6.3 Erläuterungen zum Quellcode

Kapitel 7

Das Backend

7.1 Allgemein

In Kapitel **KAPITEL VERWEIS EINFÜGEN** und **KAPITEL VERWEIS EINFÜGEN** haben wir das Thema Backend Services des öfteren erwähnt, daher werden wir uns nun diesem Bereich widmen.

Wir als Studenten haben durch das Studium, aber auch durch Nebenjobs und private Projekte Erfahrungen mit verschiedenen Programmierkonzepten sammeln können. Da wir bei der Entwicklung der Pepper Anwendung jedoch nur zwischen den Sprachen Kotlin und Java entscheiden konnten, haben wir zu Beginn unseres Projektes, das Gefühl gehabt, in unserer Entwicklung, was die Nutzung und Implementierung verschiedener Methoden angeht sehr eingeschränkt zu sein.

Genau hier war es uns von Vorteil, Kenntnisse zu NodeJS und Javascript gesammelt zu haben, denn damit ist es uns möglich gewesen, einen Express Server mit Hilfe der Laufzeitumgebung NodeJS aufzusetzen, auf welchem wir vielfältige Möglichkeiten zur Implementierung von Komplexen Vorgängen haben.

7.2 Implementierung

Damit wir eine Anwendung erstellen können, auf die wir über eine URL von außen zugreifen können, brauchen wir natürlich einen Webserver. Da wir in der Hochschule verschiedene Server benutzen dürfen, haben wir uns mit Prof. Dr. Oliver Radfelder abgesprochen, welcher uns daraufhin einen gemeinsamen Benutzeraccount mit Namen “hbv-kms“ auf dem Server Hopper angelegt hat. Für unseren gemeinsamen Account steht ein Dockercontainer zur Verfügung, auf welchem wir unsere Anwendung bereitstellen. Von außen ist unsere Anwendung dann über die URL <https://informatik.hs-bremerhaven.de/docker-hbv-kms-http> erreichbar.

7.2.1 NodeJS, npm, Express

Unsere Webanwendung basiert auf NodeJS und JavaScript. NodeJS ist eine Laufzeitumgebung, welche JavaScript außerhalb des Browsers ausführen kann und somit wichtige Prozesse auf dem Server anstatt beim Client ausführen kann. NodeJS hat einen eigenen Paketmanager, npm (Node Package Manager), mit welchem sich vielfältige Libraries und Frameworks installieren und ausführen lassen.

... wir nutzen das Modul Express, welches es uns ermöglicht, ohne großen Aufwand, eine völlig anpassbare Webanwendung zu erstellen.

7.2.2 Endpunkte

Endpunkte, auch Routes genannt, sind Schnittstellen, an denen man von außen zugreifen kann. Ein Beispiel für eine Route innerhalb der Anwendung ist: /docker-hbv-kms-http/timetable. Anzumerken ist, dass jede unserer Routes mit /docker-hbv-kms-http beginnt, da dies das der Stamm unseres Anwendungspfades innerhalb des Hochschul-webservices ist.

7.3 Installation

erst dies, dann dass und ananas ggf. muss das Ddeploy Skript angepasst werden

Kapitel 8

Abschließende Worte

Wir haben in diesem Projekt sehr viele neue Methoden und Möglichkeiten zur Entwicklung verteilter Anwendungen kennen gelernt. Der Entwurf und die Implementierung einer solch umfangreichen Webseite bzw. Webanwendung hat uns sehr gut gefallen, da wir hier im Vergleich zu vielen anderen Kursen unseres Studiums, Methoden und Werkzeuge aus der Vorlesung direkt anwenden und ausbauen konnten.

Auch die Arbeit im Team war sehr angenehm. Wir haben zum Teil täglich miteinander den Diskurs gesucht, neue Ideen ausgetauscht miteinander gearbeitet. Hierbei konnte jeder etwas von dem Anderen lernen, sodass nach einem Treffen alle etwas schlauer und mit einem guten Gefühl den Jitsi Raum verlassen konnten.

Wir nehmen aus diesem Projekt viele neue Eindrücke und vor Allem neues Wissen mit. Webentwicklung war für uns drei vor diesem Kurs, ein bisher »nur« interessantes Thema, von dem wir mal gehört haben, mit dem wir aber keine praktischen Erfahrungen teilen konnten. Die intensive Auseinandersetzung mit JavaScript und php hat bisher jedoch bei jedem von uns gefehlt, daher sind wir sehr froh auch dies endlich nachgeholt zu haben.

Literaturverzeichnis

- [1] Leonhard Thomas Schwertfeger. Bild: Mygain - trading logo, Juli 2021.
- [2] Investopia Tim Smith. Brokers. <https://www.investopedia.com/terms/b/broker.asp>, letzter Zugriff am 05. Juli 2021.