



---

# **BACHELORARBEIT**

---

Herr  
**Leon Erdhütter**

**Plattformübergreifende Daten-  
akquise und -visualisierung  
im Projektmanagement durch  
Chatbots.**

Mittweida, 2022

Fakultät: Angewandte Computer- und Biowissenschaften

---

## **BACHELORARBEIT**

---

# **Plattformübergreifende Datenakquise und -visualisierung im Projektmanagement durch Chatbots.**

Autor:  
**Herr Leon Erdhütter**

Studiengang:  
**Medieninformatik und interaktives Entertainment**

Seminargruppe:  
**MI18w2-B**

Erstprüfer:  
**Prof. Alexander Marbach**

Zweitprüfer:  
**Dipl.-Designer Norbert Rasch**

Einreichung:  
**Mittweida, 08.08.2022**

Verteidigung/Bewertung:  
**Mittweida, 2022**

# **BACHELOR THESIS**

---

## **Cross-platform data acquisition and visualization in project management through chatbots.**

author:

**Mr. Leon Erdhütter**

course of studies:

**Media Informatics and interactive entertainment**

seminar group:

**MI18w2-B**

first examiner:

**Prof. Alexander Marbach**

second examiner:

**Dipl.-Designer Norbert Rasch**

submission:

**Mittweida, 08.08.2022**

defence/evaluation:

**Mittweida, 2022**

## **Bibliografische Angaben:**

Erdhütter, Leon:

Plattformübergreifende Datenakquise und -visualisierung im Projektmanagement durch Chatbots.

Cross-platform data acquisition and visualization in project management through chatbots.

59 Seiten, Hochschule Mittweida, University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2022

## **Abstract**

Das Ziel der vorliegenden Bachelorarbeit ist es zu beantworten, inwieweit Chatbots die Aufgaben von Projektmitgliedern vor allem im Bereich Projektmanagement optimieren können. Dabei wird auf theoretische Grundlagen aufbauend, eine Analyse zur Umsetzung eines solchen Systems durchgeführt, die vor allem auf die Anforderungen innerhalb eines Projektmanagements Bezug nimmt. Im Ergebnis der Ausarbeitung soll mithilfe eines entwickelten Prototyps belegt werden, ob die genannten künstlich intelligenten Systeme es möglich machen können, gewisse Arbeitsabläufe zu erleichtern beziehungsweise diese den Projektmanagern komplett abzunehmen. Innerhalb eines abschließenden Probandentests soll demonstriert werden, dass der Einsatz des Prototyps in einem größeren Projekt sinnvoll sein kann. Weiterhin soll ebenfalls gezeigt werden, dass eine Moral- sowie Arbeitssteigerung aller Projektmitglieder durch den Prototyp möglich ist.

# Danksagung

Vorab möchte ich mich an dieser Stelle bei all denjenigen bedanken, welche mich bei der Erstellung und Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Zuerst geht dabei ein besonderer Dank an Herrn Prof. Alexander Marbach, welcher meine Bachelorarbeit als Erstprüfer betreute und diese auch am Ende begutachtet hat. Für die schnelle Unterstützung bei egal welchem Anliegen, hilfreiche Anregungen bei verschiedenen Themen und die konstruktive Kritik während der Erstellung der Arbeit, möchte ich an dieser Stelle noch einmal Danke sagen.

Weiterhin danke ich Herrn Dipl.-Designer Norbert Rasch für die Unterstützung als Zweitprüfer. Vielen Dank für schnelle Beihilfe und für das Lesen und Bewerten dieser Bachelorarbeit.

Einen weiteren Dank möchte ich auch an Laurenz Bischoff richten, welcher mir in einem Interview aussagekräftige Fakten im Bereich Projektmanagement zur beta Mittweida geliefert hat.

Ebenfalls danke ich allen Teilnehmenden des Probandentests für ihre Zeit sowie für das hilfreiche Feedback, welches ich bekommen habe.

Abschließend möchte ich mich bei meiner Familie und meinen Freunden bedanken, die immer ein offenes Ohr für mich hatten sowie mich durch Korrekturlesen und durch wertvolle Anregungen beim Schreiben der Bachelorarbeit unterstützt haben.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>I</b>
<b>Abbildungsverzeichnis .....</b>	<b>IV</b>
<b>Tabellenverzeichnis .....</b>	<b>VI</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VII</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Zielsetzung und Hintergrund .....	1
1.3 Aufbau der Arbeit .....	2
<b>2 Theoretische Grundlagen .....</b>	<b>3</b>
2.1 Überblick und Definitionen .....	3
2.1.1 Begriffsbestimmungen.....	3
2.1.2 Mensch-Computer-Interaktion .....	4
2.1.3 Interaction Design .....	4
2.1.4 User Centered Design.....	4
2.1.5 User Interface.....	5
2.1.6 User Experience.....	5
2.1.7 Application Programming Interface.....	6
2.1.8 Künstliche Intelligenz.....	7
2.2 Projektmanagement .....	8
2.2.1 Was ist Projektmanagement.....	8
2.2.2 Historische Entwicklung .....	9
2.2.3 Projektarten.....	9
2.2.4 Projektmanagement Tools.....	10
2.2.5 Charakteristik im Projektmanagement.....	11
2.2.6 Richtlinien für Effektives Projektmanagement.....	11
2.2.7 Projektstrategie im Game Developement .....	13
2.3 Chatbots .....	14
2.3.1 Einführung.....	14
2.3.2 Geschichte der Chatbots.....	14
2.3.3 Nutzung von Chatbots.....	15
2.3.4 Arten von Chatbots .....	15

2.3.5	Design und Entwicklung von Chatbots .....	16
2.3.6	Programmierung von Chatbots.....	18
2.4	Pipedream.....	18
2.4.1	Was ist Pipedream .....	18
2.4.2	Entwicklung mit Pipedream .....	18
2.5	Kommunikationsplattform Discord.....	19
2.5.1	Allgemeines .....	20
2.5.2	Wer sind die Konsumenten .....	20
2.5.3	Discord Chatbot .....	22
2.5.4	Discord Chatbot Funktionen.....	22
2.6	beta Mittweida.....	23
2.6.1	Was ist die beta.....	23
2.6.2	Verwendete Projektmanagement Tools.....	23
2.6.3	Probleme der beta.....	24
<b>3</b>	<b>Analyse und Konzeption.....</b>	<b>25</b>
3.1	Anforderungsanalyse .....	25
3.1.1	Anwendungsszenario .....	25
3.1.2	Anforderungen an den Prototyp .....	27
3.1.3	Funktionale Anforderungen .....	28
3.1.4	Nicht-funktionale Anforderungen .....	29
3.2	Entwicklung eines Konzeptes.....	30
3.2.1	Grafische Benutzeroberfläche .....	30
3.2.2	Funktionen und Kommandos.....	31
<b>4</b>	<b>Implementierung .....</b>	<b>33</b>
4.1	Prototyp als Discord Bot.....	33
4.1.1	Programmierung mit Python und MySQL .....	33
4.1.2	Datenbankerstellung .....	34
4.1.3	Verwirklichung des Chatbots .....	35
4.1.4	Funktionen und Abläufe .....	36
4.2	Benutzereingaben .....	41
4.2.1	Übermittlung von Kommandos .....	41
4.2.2	Senden und Empfangen von Daten.....	44
4.3	Test des Prototyps .....	45
4.3.1	Testkonzeption und Ablauf.....	46
4.3.2	Durchführung der Testphase.....	47
4.3.3	Auswertung der Testphase.....	49

4.3.4	Fazit .....	57
<b>5</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>58</b>
	<b>Literatur .....</b>	<b>60</b>
	<b>Anlagen, Teil 1 .....</b>	<b>63</b>
	<b>Anlagen, Teil 2.....</b>	<b>73</b>
	<b>Selbstständigkeitserklärung</b>	



# Abbildungsverzeichnis

Abbildung 1: UCD-Prozess.....	5
Abbildung 2: User Experience Ablauf .....	6
Abbildung 3: Anteil der Unternehmen mit KI-Einsatz in Deutschland .....	7
Abbildung 4: Bedingungen für ein ideales Projektmanagement .....	12
Abbildung 5: Übersicht eines dreistufigen iterativen Entwicklungsprozess .....	13
Abbildung 6: Allgemeine Chatbot-Architektur .....	17
Abbildung 7: Beispiel eines Pipedream Workflows .....	19
Abbildung 8: Discord Nutzerzahlen der Jahre 2017-2021 .....	20
Abbildung 9: Discord Servertypen mit den meisten Nutzern .....	21
Abbildung 10: Anwendungsdiagramm .....	26
Abbildung 11: Einteilung der Anforderungen an ein Software-System .....	28
Abbildung 12: Konzept der Benutzeroberfläche.....	30
Abbildung 13: Datenbanktabelle mit Eintrag.....	35
Abbildung 14: Python und Slash Command Attribute .....	36
Abbildung 15: Slash Command für die Registrierung .....	38
Abbildung 16: Slash Command für die gesamt Bestenliste.....	39
Abbildung 17: Ausgabe der gesamt Bestenliste in Discord.....	40
Abbildung 18: Bildbearbeitung im Drop Slash Command .....	40
Abbildung 19: Karten Ausgabe .....	41
Abbildung 20: Discord Slash Commands .....	42
Abbildung 21: Herstellen einer Datenbankverbindung in Python .....	44
Abbildung 22: MySQL Insert Statement in Python.....	44
Abbildung 23: MySQL Update Statement in Python .....	45
Abbildung 24: MySQL Select Statement in Python.....	45
Abbildung 25: Fragebogen Auswertung – Persönliche Informationen.....	49

Abbildung 26: Fragebogen Auswertung – Projektmanagement .....	50
Abbildung 27: Fragebogen Auswertung – Chatbots.....	51
Abbildung 28: Fragebogen Auswertung – Kommunikationsplattform Discord.....	52
Abbildung 29: Fragebogen Auswertung – Chatbot Präferenzen .....	53
Abbildung 30: Fragebogen Auswertung – Bewertung des Prototyps .....	54
Abbildung 31: Fragebogen Auswertung – Bewertung des beta Card Game .....	55
Abbildung 32: Fragebogen Auswertung – Ausblick des Prototypen.....	56

# Tabellenverzeichnis

Tabelle 1: Übersicht der Rahmenbedingungen in verschiedenen Projektarten .....	9
Tabelle 2: Übersicht verschiedener Projektmanagementtools und Whiteboards.....	10
Tabelle 3: Beschreibung der Python und Slash Command Attribute.....	37
Tabelle 4: Testkonzept des Prototypen .....	47

# Abkürzungsverzeichnis

<b>AIML</b>	Artificial Intelligence Markup Language
<b>API</b>	Application Programming Interface
<b>CLI</b>	Command Line Interface
<b>GUI</b>	Graphical User Interface
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IKT</b>	Informations- und Kommunikationstechnik
<b>IPMA</b>	International Project Management Association
<b>KI</b>	Künstliche Intelligenz
<b>MCI</b>	Mensch-Computer-Interaktion
<b>MMI</b>	Mensch-Maschine-Interaktion
<b>NLP</b>	Natural Language Processing
<b>NLU</b>	Natural Language Understanding
<b>TP</b>	Taskpoints
<b>TUI</b>	Text User Interface
<b>UCD</b>	User Centered Design
<b>UI</b>	User Interface
<b>URL</b>	Uniform Resource Locator
<b>UX</b>	User Experience
<b>VoIP</b>	Voice over IP
<b>VUI</b>	Voice User Interface



# 1 Einleitung

Im einführenden Kapitel werden Motivation, Problemstellung sowie das Ziel dieser Bachelorarbeit besprochen. Weiterhin erfolgt ein kurzer Einblick zu den einzelnen Kapiteln und damit auch, was von dieser Arbeit erwartet werden kann.

## 1.1 Motivation

Im heutigen Zeitalter erleben wir eine immer weiter voranschreitende Digitalisierung in den verschiedensten Bereichen. Viele Unternehmen sowie andere Einrichtungen planen auf dieser Basis immer mehr erfolgsorientierte Projekte. Ein wichtiger und entscheidender Punkt ist dabei die Planung des richtigen Projektmanagements. Gerade dieser Bereich kann über einen möglichen Erfolg oder aber auch Misserfolg des gesamten Projektes entscheiden. Da diese Planung eine sehr große Herausforderung sein kann, stellt sich die Frage, inwieweit sich diese mithilfe künstlicher Intelligenz (KI, vgl. Abschnitt 2.1.8) optimieren lässt.

Ein gutes Projektmanagement kann eine sehr anstrengende und auch zeitaufwendige Aufgabe sein und dabei wurde auch der Bedarf festgestellt, dass sich daran etwas ändern muss. Es gibt teilweise einfach zu viele Aufgaben, um welche sich ein Projektmanager gleichzeitig kümmern muss, wie zum Beispiel das Koordinieren mit verschiedenen Projektmanagement-Hilfsmitteln, das Bewerten der Arbeitsweise von Projektmitarbeitern, die Verwaltung des zeitlichen Rahmens und noch vieles mehr. Bei dieser hohen Anzahl an Tätigkeiten ist es auch nicht ausgeschlossen, dass Fehler gemacht werden, da am Ende der Überblick verloren geht.

Aufgrund des heutigen Entwicklungsstandes ist es daher verständlich, dass eine mögliche Lösung für dieses Problem künstlich intelligente Systeme in Form von Chatbots sind. Diese sind teilweise so weit entwickelt, dass sie Menschen in vielen anderen Bereichen das Leben erleichtern. Deshalb ist es naheliegend, dass ein solcher Chatbot auch einen positiven Einfluss im Bereich Projektmanagement haben kann.

Schon seit langem ist künstliche Intelligenz für mich ein sehr spannendes sowie wichtiges Thema und auch gar nicht mehr aus meinem sowie unser aller Leben wegzudenken. Selbst habe ich mich in diesem Bereich schon viel beschäftigt, viele Nachforschungen angestellt und auch schon eigene Chatbots konzeptioniert und programmiert. Dabei konnte ich mir schon ein solides Grundwissen zu diesem Thema aufbauen und denke, dass dies eine gute Ausgangslage für meine Untersuchungen sowie meine wissenschaftliche Arbeit ist.

## 1.2 Zielsetzung und Hintergrund

In Kooperation mit der beta 2023 Mittweida (vgl. Abschnitt 2.6) wird vor allem im Projektmanagement erörtert, was die wichtigsten und schwierigsten Aufgaben im Projekt sind. Ziel dabei ist herauszufinden, wo die Projektmanager dringend Hilfe benötigen und wo diese die größten Hürden sehen.

Dafür soll ein geeigneter Prototyp entwickelt werden, der es schaffen soll, diese Probleme zu mindern beziehungsweise vollständig zu beheben. Dieser soll in Form eines Discord Chatbots entstehen, da diese zum einen sehr großen Funktionsumfang bieten und viele Projekte, wie zum Beispiel die beta Mittweida schon seit Jahren mit der Kommunikationsplattform Discord (vgl. Abschnitt 2.5) arbeiten. Weiterhin soll dieser Prototyp mit der Internetplattform Pipedream (vgl. Abschnitt 2.4) verbunden werden, umso auf die Application Programming Interface (API, vgl. Abschnitt 2.1.7) von beispielsweise Trello, Git und anderen verwendeten Tools (vgl. Abschnitt 2.1.1) zugreifen zu können.

Der Chatbot soll außerdem anhand eines Probandentests analysiert werden, um dessen Gebrauchstauglichkeit innerhalb eines Projektes einschätzen zu können. Basierend darauf sollen durch das Ergebnis mögliche Probleme gefunden werden, welche dann später durch Verbesserungen behoben werden. Entwickelt werden soll der Prototyp neben Pipedream, mittels der Programmiersprache Python (vgl. Abschnitt 4.1.1) und dem Datenbankverwaltungssystem MySQL (vgl. Abschnitt 4.1.1).

Grundsätzlich gilt also, den Arbeitsaufwand von Projektmanagern zu senken und für eine geringere Fehlerquote sowie eine bessere Überschaubarkeit im Projektmanagement zu sorgen. Auch bei den restlichen Projektmitgliedern soll sich eine Moral- sowie Arbeitssteigerung erkennbar machen.

### **1.3 Aufbau der Arbeit**

Die vorliegende Arbeit setzt sich aus mehreren Teilen zusammen. Beginnend werden im Theorieteil die nötigen Grundlagen, wie Definitionen und allgemeine Themen zur Benutzerfreundlichkeit erläutert. Es folgt ein theoretischer Teil zum Thema Projektmanagement, was dieses ausmacht und welche Tools es dafür gibt. Weiterhin wird erklärt, was Chatbots sind, wo sie zum Einsatz kommen, welche Arten es gibt und wie man diese entwickelt. Nach einer anschließenden Einführung in die Kommunikationsplattform Discord, mit den Schwerpunkten wie Discord funktioniert, wer die Konsumenten sind und wie weit die Discord Bots bisher entwickelt sind, folgt abschließend noch eine Erläuterung der beta Mittweida.

Der praktische Hauptteil der Arbeit beschäftigt sich mit der Erarbeitung und Entwicklung eines ersten Konzeptes. Daraus soll nach umfassender Prüfung und Evaluation ein funktionstüchtiger Prototyp entstehen. Für diesen Prototyp wird außerdem ein mögliches Anwendungsszenario beschrieben, welches die Grundlage der Anforderungsanalyse bezüglich Funktionalität, technischen Umsetzung und Qualität bildet.

Hierauf aufbauend ergibt sich eine detaillierte Beschreibung der umgesetzten Funktionen des Prototyps. Weiterhin soll darauf eingegangen werden, wie die Programmierung abgelaufen ist und womit der Chatbot plattformübergreifend verknüpft wurde. Außerdem wird erklärt, wie die Eingaben vom Benutzer auszusehen haben und wie der Prototyp mit diesen umgeht. Darüber hinaus soll die Durchführung eines Probandentests zur Bewertung und Einschätzung des Chatbots anhand eines entsprechenden Fragebogens ausgewertet werden. Im Test gefundene Probleme und Verbesserungsvorschläge sollen direkt im Anschluss implementiert werden. Allerdings werden zu aufwendige Verbesserungen ausschließlich im Ausblick noch einmal aufgegriffen. Den Abschluss der Arbeit bildet eine Zusammenfassung mit Ausblick, in dem die wichtigsten Erkenntnisse der Arbeit noch einmal aufgegriffen werden.

## 2 Theoretische Grundlagen

Um diese Arbeit vollständig verstehen zu können, ist es notwendig, die nötigen theoretischen Grundlagen zu betrachten. Dadurch wird ein Fundament geschaffen, welches von großer Bedeutung für das weitere Vorgehen ist. Da sich diese Arbeit hauptsächlich den Gebieten Projektmanagement und Chatbots widmet, ist es wichtig, auf diese und zusammenhängende Bereiche genau einzugehen, um damit ein grundlegendes Verständnis der Thematik zu schaffen. Auch der Einblick in bedeutende Begriffe und Definitionen soll in diesem Kapitel sicher gestellt werden. Im weiteren Verlauf wird die Bachelorarbeit auf dieser Grundlage aufgebaut.

### 2.1 Überblick und Definitionen

In der heutigen Zeit spielt die Entwicklung von innovativen Interaktionstechniken sowie Benutzungsschnittstellen eine große Rolle für die Verwirklichung erfolgsorientierter und vor allem ubiquitärer Anwendungen. Es gibt mittlerweile zahlreiche interaktive Systeme in unserem Leben, welche aber teilweise enorme Unterschiede in der Benutzungsqualität aufweisen. Aus diesem Grund ist es unumgänglich, auf Faktoren zu schauen, welche die Zielführung eines Produktes bestimmen sowie auch die Möglichkeiten zu betrachten, diese Faktoren bei der Entwicklung und Gestaltung umzusetzen. Über die letzten Jahrzehnte haben sich verschiedene Fachgebiete entwickelt, die hinsichtlich der Entwicklung von Anwendungen einen nutzerorientierten Ansatz beinhalten und im folgenden genauer betrachtet werden sollen.

#### 2.1.1 Begriffsbestimmungen

Für ein gutes Verständnis dieser Bachelorarbeit ist es wichtig, verschiedene Begriffe genau zu definieren, um so Verwechslungen sowie Konfusion vorzubeugen.

- **Tools:** Unter dem Begriff Tools versteht man im jeweiligen Kontext eine Software, die dem Bearbeiter einem Werkzeug ähnlich hilft, Problemstellungen zu lösen.
- **Repository:** Ein Repository ist eine spezielle Datenbank, die vom System zum Speichern von Modellen und deren Komponenten verwendet wird. Die grundlegende Funktion eines Repositories besteht darin, gespeicherte Modelle und ihre Komponenten zu versionieren und deren Abruf zu unterstützen (vgl. Leymann o. D.).
- **Commit:** Ein Commit ist ein Begriff aus der Softwareentwicklung, der die bestätigte Freigabe einer oder mehrerer Änderungen beschreibt. Es dient sowohl zur Persistenz von Daten in Datenbanken als auch zur Eingabe von Programmquellcode in Versionskontrollsystemen (vgl. Commit: Wikipedia 2022).
- **Slash Command:** Ein Slash Command (dt. Slash Kommando) ist ein in der Kommunikationsplattform Discord (vgl. Abschnitt 2.5) integriertes Kommunikationssystem für Chatbots. Die Besonderheit daran ist, dass jedes Kommando mit einem Slash beginnt.



- **Taskpoints:** Die Taskpoints (TP) sind ein in der beta Mittweida genutzter Bewertungsmaßstab für die einzelnen Projektmitglieder. Der zu erstellende Prototyp wird auf diese zurückgreifen, kann jedoch in einem anderen Projekt auf jedes beliebige andere Bewertungssystem umbenannt werden.
- **beta Card Game:** Das beta Card Game ist ein Eigenname für das entwickelte Kartenspiel des Prototyps. Um Konfusion vorzubeugen, wird hier verdeutlicht, dass dieses nicht im Zusammenhang mit der beta Mittweida steht und von jedem beliebigen Projekt genutzt werden kann.

### 2.1.2 Mensch-Computer-Interaktion

Die Mensch-Computer-Interaktion (engl. Human-Computer-Interaction, auch als MCI abgekürzt) untersucht, wie Benutzer mit softwarebasierten Anwendungen umgehen. Ein übergreifender Bereich ist die Mensch-Maschine-Interaktion (MMI), die ähnlichen Fragestellungen rund um die nutzer- und kontextbezogene Gestaltung von Informationssystemen adressiert. Allerdings verallgemeinert MMI den Interaktionspartner des Menschen zur Maschine (vgl. Koch et al. 2020, S. 381).

Oftmals stehen neuartige Interaktionsformen und Technologien, sowie Ein- und Ausgabemedien, im Fokus. Die psychologischen Rahmenbedingungen, in denen sich die Benutzer befinden, während sie diese Technologien nutzen, spielen eine bedeutende Rolle. Ein zentraler Faktor ist die Effizienz, womit die Technologien eingesetzt werden können, zum Beispiel bei der Industrie, wo ganze Generationen von Anlagen mit Touchscreens ausgestattet werden oder bei der Erstellung von hochleistungsfähigen Anwendungen im Finanzbereich. Mittlerweile spielen auch Mobile Technologien eine ebenso große Rolle wie Produkte sowie Anwendungen im Bereich Unterhaltung oder Kommunikation (vgl. Richter/Flückiger 2016, S. 7).

### 2.1.3 Interaction Design

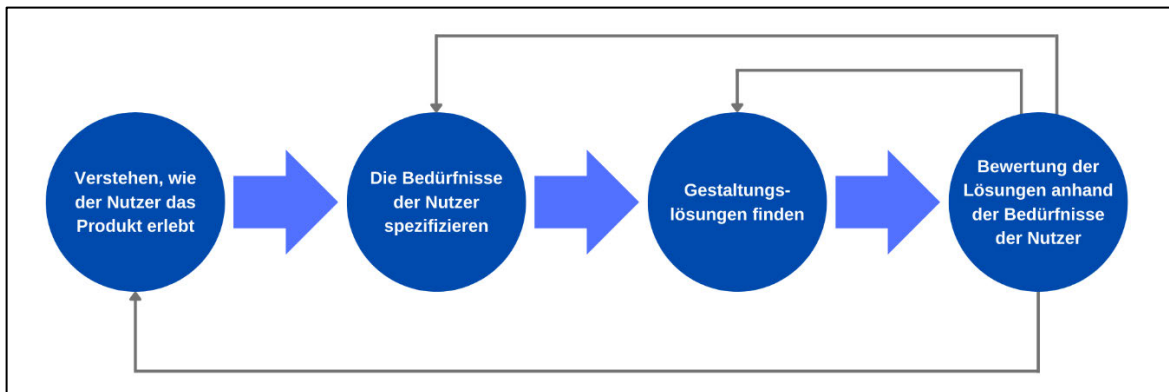
Die Erstellung und Verwirklichung von interaktiven Systemen sind darauf ausgerichtet, die Aufgaben der Benutzer bestmöglich zu unterstützen. Dialoge werden auf eine Weise entwickelt, dass sie leicht zu verstehen sind und effizient zum gewünschten Ergebnis führen. Dabei existieren für die unterschiedlichen Plattformen und Technologien verschiedene Interaktionsprinzipien sowie Richtlinien. Es ist trotzdem unumgänglich, Prototypen zu entwerfen und eine Bewertung durch die Benutzer für eine optimale Gestaltung der Benutzerschnittstelle vorzunehmen (vgl. Richter/Flückiger 2016, S. 8).

Interaction Design hat es sich als Fachgebiet zur Aufgabe gemacht, die Herleitung und den Entwurf geeigneter Interaktionskonzepte für digitale Produkte und Anwendungen zu optimieren. Darüber hinaus steht eine gute User Experience (UX, vgl. Abschnitt 2.1.6) im weiteren Sinne im Vordergrund. Mittels nutzerorientierter Methoden werden dabei die notwendigen Funktionen, Informationen und Abläufe vereinbart und mit Benutzern getestet. Interaction Design enthält dabei drei grundlegende Aspekte. Als erstes ein Verständnis für gute Gestaltung und Designprozesse mit Fokus auf interaktive Produkte, zum Beispiel aus den Disziplinen der grafischen Gestaltung oder dem Industriedesign. Als weiteren Punkt die kreativen, gestalterischen Techniken unter Beachtung der zu vermittelnden Gefühlswelten. Zuletzt das ausführliche Wissen der eingesetzten Technologie der Benutzerschnittstelle (vgl. Richter/Flückiger 2016, S. 192).

### 2.1.4 User Centered Design

Der Begriff User Centered Design (UCD) bezieht sich auf verschiedene Gestaltungsprozesse, die den zukünftigen Benutzer in den Mittelpunkt der Entwicklung stellen. Mit der

Betonung auf Design wird damit ausgedrückt, dass sowohl Interaktions- als auch Gestaltungsaspekte, also zum Beispiel die Gestaltung von Dialogabläufen, die Gestaltung der Form von physischen Produkten sowie Bedienelementen, aber auch das grafische Design wichtige Faktoren für eine optimale Benutzung sind und von Beginn an beachtet werden müssen (vgl. Richter/Flückiger 2016, S. 9).



**Abbildung 1: UCD-Prozess, Quelle: Eigene Darstellung nach Frigola (2021)**

Wie in Abbildung 1 zu sehen, umfasst der UCD-Prozess insgesamt vier verschiedene Phasen. Zuerst muss verstanden werden, wie der Benutzer das Produkt oder Ähnliches erlebt. Im nächsten Schritt werden die Bedürfnisse des Endbenutzers angegeben. Dabei wird basierend auf der durchgeführten Recherche eingegrenzt, welches Endbenutzerproblem am wichtigsten zu lösen ist. Als Nächstes werden Lösungen entworfen, die bei dem Problem des Endbenutzers helfen sollen. An diesem Punkt wird außerdem mit der Entwicklung des Produkts begonnen. Im letzten Schritt wird das Design anhand der Anforderungen des Endbenutzers bewertet und überprüft, ob er dessen Probleme löst (vgl. Frigola 2021).

### 2.1.5 User Interface

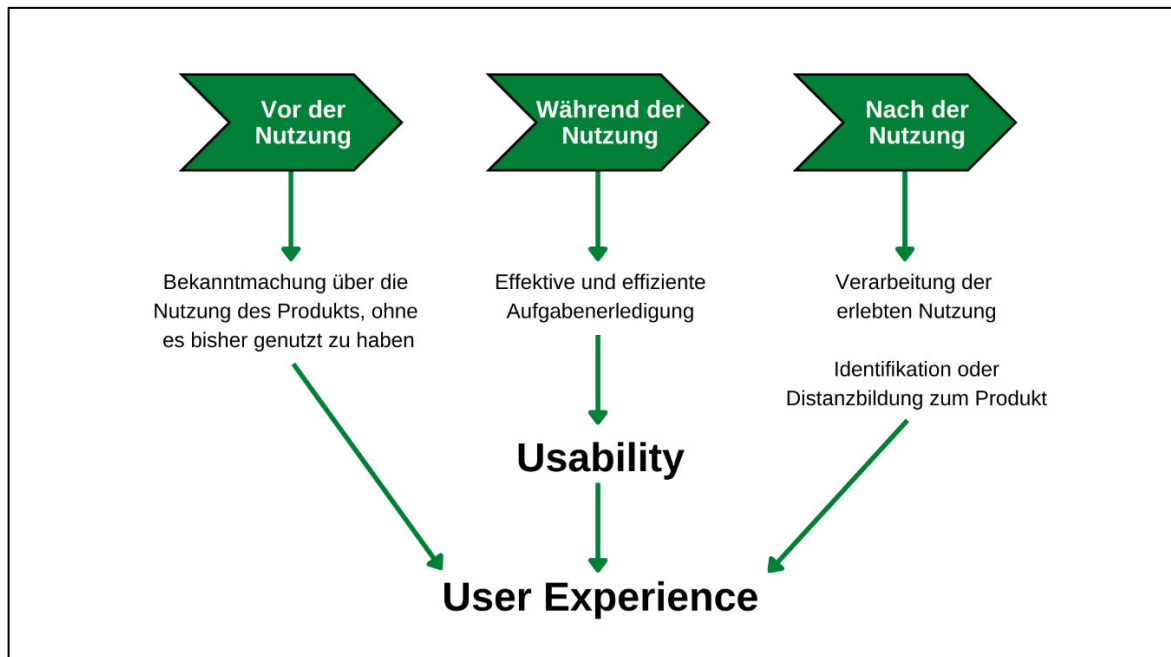
Das User Interface (UI) ist die Benutzeroberfläche bzw. die Benutzerschnittstelle, welche definiert, was ein Benutzer sieht und womit er interagiert. Da es oft unmöglich ist, direkt mit einem Programm oder einer Maschine zu kommunizieren, fungiert das UI als menschlicher Kommunikator (vgl. o. V. 2021, SEO-Küche).

Prinzipiell ist das UI in verschiedene Interaktionsebenen unterteilt, die wiederum unterschiedliche Eingabemöglichkeiten beinhalten. Bei dem Text User Interface (TUI) erfolgt die Eingabe durch Text, während bei dem Voice User Interface (VUI) die Eingabe durch die Stimme erfolgt. Beim Command Line Interface (CLI) werden Befehle direkt über eine Kommandozeile eingegeben. Die heute am weitesten verbreitete Interaktionsebene ist das Graphical User Interface (GUI). Dieses umfasst eine grafische Benutzerschnittstelle, die es dem Benutzer ermöglicht, mit einem elektronischen Gerät über grafische Symbole und Audioindikatoren, anstatt über eine textbasierte Benutzerschnittstelle zu interagieren (vgl. o. V. 2021, SEO-Küche).

### 2.1.6 User Experience

Der Begriff User Experience (UX) bezieht sich auf die Gesamterfahrung eines Benutzers in Bezug auf die wahrgenommene Qualität von Interaktionen bei der Verwendung eines Programms, Produkts oder einer Dienstleistung. Sie ist ein wesentlicher Erfolgsfaktor digitaler Konzepte. Neben der Funktion wurde auch den Begriffen Ästhetik und Emotion eine besondere Bedeutung beigemessen. Im Fokus stehen neben kommerziellen Anwendungen auch

Entertainment-Bereiche wie zum Beispiel Videospiele. Aufgrund der umfassenden und vielfältigen Sichtweise hat sich der Begriff UX auf unterschiedliche Weise durchgesetzt. Oft werden die Bezeichnungen User Experience und Usability (dt. Gebrauchstauglichkeit) synonym verwendet, dies ist aber nur teilweise korrekt (vgl. Richter/Flückiger 2016, S. 9).



**Abbildung 2: User Experience Ablauf, Quelle: Eigene Darstellung nach Becker (2015)**

Wie in Abbildung 2 zu sehen ist, umfasst der komplette User Experience Ablauf insgesamt drei Phasen. In der ersten Phase soll sich der Benutzer mit der Nutzung des Produktes bekanntmachen. Als Nächstes folgt die Phase während der Nutzung, in welcher eine effektive und effiziente Aufgabenerledigung erfolgt. Zuletzt ergibt sich noch die Phase nach der Nutzung. Hier wird das Erlebte verarbeitet und es folgt eine Identifizierung mit dem Produkt oder eine Distanzbildung zum Produkt. Alle genannten Phasen bilden zusammen die User Experience. Als Usability lediglich nur die zweite Phase bezeichnet (vgl. Becker 2015).

### 2.1.7 Application Programming Interface

Die Application Programming Interface (API) bezeichnet eine Programmierschnittstelle, dessen Anbindung auf Quelltext-Ebene erfolgt. Programmierschnittstellen dienen dem standardisierten Austausch von Informationen zwischen Anwendungen und einzelnen Programmteilen. Die Übertragung von Daten und Befehlen ist nach der zuvor definierten Syntax strukturiert. APIs ermöglichen es, die Programmierung zu modularisieren und damit zu vereinfachen. Die einzelnen über die API verbundenen Programmteile implementieren spezifische Funktionalitäten und sind klar vom Rest der Anwendung getrennt. Die Kommunikation einzelner Programmmodule kann nur über genau spezifizierte Schnittstellen erfolgen. Die API definiert die Form, in der Informationen und Daten von Anwendungsmodulen empfangen und zurückgesendet werden. Wird eine Programmierschnittstelle bereitgestellt, geschieht dies in der Regel zusammen mit einer ausführlichen Dokumentation der einzelnen Funktionen, der genauen Syntax und möglicher Parameter. Über APIs kann auf Hardwarekomponenten, Datenbanken, einzelne Programmfunktionen oder Schnittstellen und andere Elemente zugegriffen werden (vgl. Luber 2017).

## 2.1.8 Künstliche Intelligenz

Der Bereich der künstlichen Intelligenz (KI) hat über die letzten Jahre den Erfolg verzeichnet, die Effizienz intelligenter Techniken dank neuer technologischer und methodischer Fortschritte beachtlich zu steigern. Neben der Verfügbarkeit leistungsfähigerer Hardware sind neue Machine-Learning-Verfahren, insbesondere Deep-Learning-Verfahren auf Basis neuronaler Techniken und die wachsende Verfügbarkeit sehr großer Datenmengen (Big Data) Schlüsselfaktoren, die in der Breite zu erstaunlichen Durchbrüchen geführt haben. Darunter zählen zum Beispiel Anwendungsbereiche wie Bild- und Sprachverständnis, Datenanalyse, Empfehlungssysteme und die automatische Generierung künstlerischer Artefakte (vgl. Koch et al. 2020, S. 381).

Künstliche Intelligenz bezieht sich also auf die Fähigkeit einer Maschine, die kognitiven Aufgaben zu erfüllen, die wir mit dem menschlichen Verstand assoziieren. Dazu gehören Wahrnehmungsfähigkeiten ebenso wie die Fähigkeit zu argumentieren, selbstständig zu lernen und somit eigenständig Problemlösungen zu finden (vgl. Kreutzer/ Sirrenberg 2019, S. 3).

In der heutigen Zeit erstreckt sich der Gebrauch von KI über immer mehr Bereiche. Beispiele dafür sind persönliche Assistenten wie Alexa, Siri oder Google Home, welche durch Spracheingaben Musik abspielen, Einkaufslisten erstellen oder Termine vereinbaren können. Auch Übersetzungshilfen im Internet wie Google Translate oder DeepL bedienen sich ebenfalls an KI-Anwendungen. Ein weiteres Einsatzfeld von künstlicher Intelligenz stellen Roboter dar. Diese kommen zum Beispiel im Bereich autonomes Fahren vor, wobei sie sogar die Rolle als Fahrer übernehmen (vgl. Kreutzer/Sirrenberg 2019, S. 1).

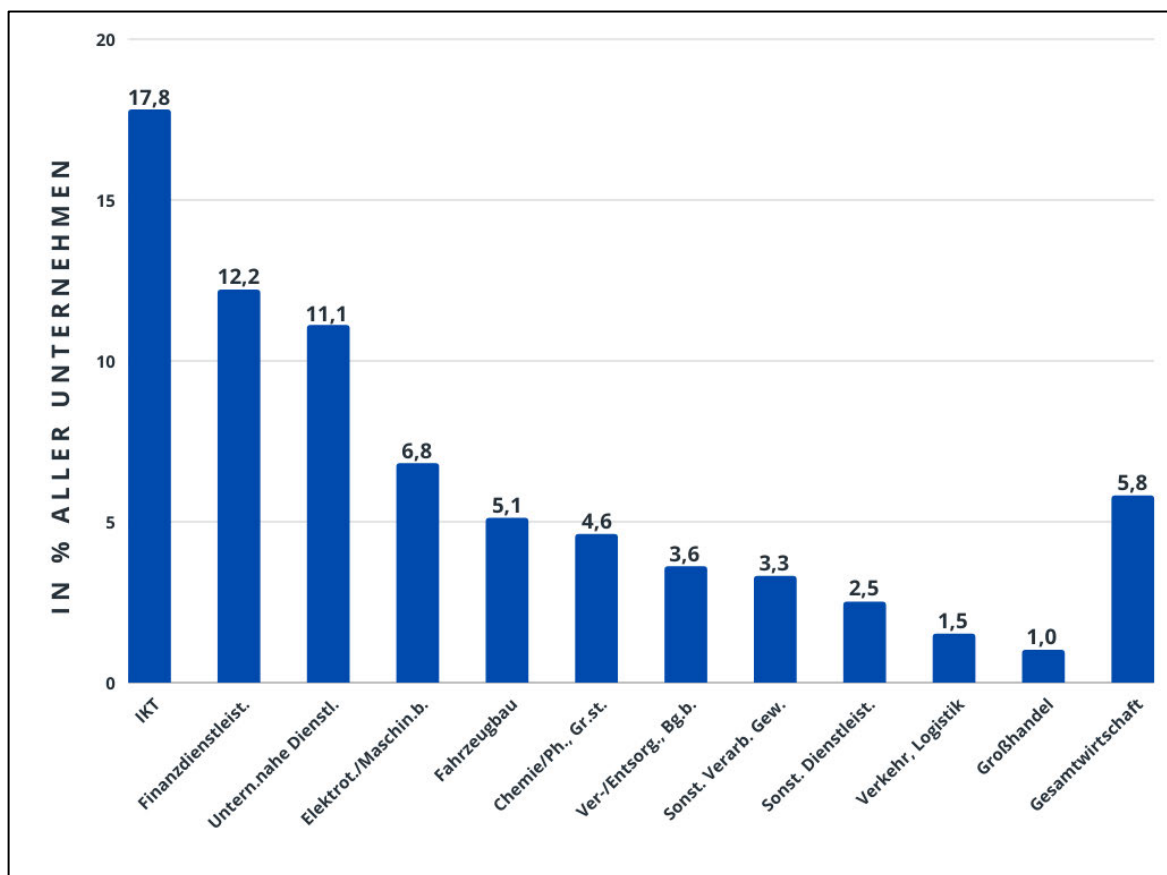


Abbildung 3: Anteil der Unternehmen mit KI-Einsatz in Deutschland, Quelle: Eigene Darstellung nach Rammer et al. (2020), S.7

Die Verbreitung von künstlicher Intelligenz erstreckt sich heutzutage auf unterschiedliche Funktionsbereiche verschiedener Unternehmen. Im Jahr 2019 setzten rund 17.500 Betriebe in Deutschland Verfahren der künstlichen Intelligenz ein. Wie man Abbildung 3 entnehmen kann, ergibt dies in der Gesamtwirtschaft einen Anteil von 5,8 %. In der Informations- und Kommunikationstechnik (IKT) ist dieser Anteil mit 17,8 % deutlich höher. Dies kommt zustande, da die IKT-Branche nicht nur Nutzer von KI für eigene Produkte und Prozesse ist, sondern diese auch für andere Branchen entwickelt. Eine weitere hohe Verbreitung ergibt sich im Bereich Finanzdienstleistungen (u. a. Anlageberatung und -vermittlung, Eigenhandel) mit 12,2 % und in den unternehmensnahen Dienstleistungen (Ingenieurbüros, Werbung) mit 11 %. In den weiteren dargestellten Bereichen lässt sich allerdings nur eine deutlich geringere Verbreitung aufweisen (vgl. Rammer et al. 2020, S. 7).

## 2.2 Projektmanagement

Die steigende Dynamik in Wirtschaft und Wissenschaft, sich verkleinernde Produktlebenszyklen, intensiver Wettbewerb und sich ändernde Umweltbedingungen stellen Unternehmen immer wieder vor neue, komplexe und oft unbekannte Herausforderungen. In der Vergangenheit hat man diese einmaligen Aufgaben unter Beachtung der Leistungs-, Kosten- und Terminziele meistens in einer hierarchisch geprägten Organisationsform durchgeführt. Die Bewältigung dieser Aufgaben erfordert große Flexibilität, Kreativität und innovatives Denken sowie die Zusammenarbeit von Mitarbeitern unterschiedlicher Qualifikationen und Niveaus. Um den Anforderungen dieser Aufgaben gerecht zu werden, ist das Projektmanagement eine bereichsübergreifende ressourcenbündelnde Organisationsform (vgl. Witt 2000, S. 11).

Für den weiteren Verlauf dieser Arbeit ist es bedeutsam, innerhalb dieses Themengebiets ein grundlegendes Verständnis aufzubauen. Dabei soll geklärt werden, was man sich unter Projektmanagement vorstellen kann, wie es in der heutigen Zeit funktioniert, welche Anwendungen es bereits dafür gibt, wie ein gut funktionierendes Projektmanagement aussieht und wie dieses im Game Development Bereich zum Einsatz kommt. Daraus lässt sich ableiten, was für einen Chatbot im Projektmanagement wichtig ist, worauf bei der Implementierung geachtet werden muss und welchen Funktionsumfang er bieten muss.

### 2.2.1 Was ist Projektmanagement

Projektmanagement dient als Oberbegriff für alle Planungs-, Überwachungs-, Koordinations- und Steuerungsmaßnahmen, die zur Transformation beziehungsweise Neugestaltung eines Systems sowie Prozesses oder zur Problemlösung erforderlich sind. Der Prozess der Lösungsfindung, die dafür notwendigen Mittel, ihr Einsatz sowie ihre Abstimmung sind dabei wichtiger als die Lösung selbst. Oft wird Projektmanagement als das Gleiche angesehen wie Linienmanagement. Dies ist nicht richtig, denn Linienmanagement befasst sich eher mit der Führung des sogenannten laufenden Geschäfts und der beteiligten Organisationen (vgl. Kuster et al. 2019, S. 13).

Egal ob es um die Entwicklung neuer Produkte oder Dienstleistungen, den Bau neuer Produktionsstätten, die Einführung von Qualitätsmanagementsystemen für Pflegeheime, den Zusammenschluss von Unternehmen mit Wettbewerbern oder den Bau großer Flughäfen geht, fast alle Unternehmensformen sind ebenso wie die öffentliche Verwaltung vielen Anforderungen ausgesetzt, die nicht zum täglichen Geschäft oder zum Schwerpunkt ihrer Organisation gehören. All dies Genannte wird in Form von Projekten organisiert. Diese Projekte sind allerdings neu für die jeweiligen Unternehmen und werden daher auch als innovativ bezeichnet. Zur möglichst optimalen Durchführung der genannten Projekte hat sich die Organisationsform Projektmanagement entwickelt und bietet ein Gesamtkonzept vieler Methoden und Werkzeuge für eine möglichst effiziente und erfolgsorientierte Abwicklung (vgl. Bergmann / Garrecht 2021, S. 309 ff.).

## 2.2.2 Historische Entwicklung

Seit der Umsetzung von Großprojekten in der Menschheitsgeschichte konnte man von implizitem Projektmanagement sprechen, darunter zählen zum Beispiel der Bau der Pyramiden, der großen Mauer oder die Expeditionen der großen modernen Entdecker. Der Bau des Hoover-Staudamms, das Manhattan-Projekt zur Entwicklung der Atombombe und das Apollo-Programm der NASA wurden als erste Großprojekte zur Förderung eines systematischen Projektmanagements und der Entwicklung und Nutzung seiner Instrumente aufgeführt. Die ersten Projektmanagementorganisationen und Verbände wurden Mitte der 1960er-Jahre gegründet, wie zum Beispiel die International Project Management Association (IPMA) 1965 in Wien. Im Zuge der Professionalisierung des Projektmanagements haben sich verschiedene Methodiken und Regelwerke zu einem Standard in dessen Bereich entwickelt. Dabei handelt es sich um Best-Practice-Verfahren, welche ursprünglich für IT-Projekte vorgesehen waren, aber mittlerweile zum Managen aller Projekte da sind. Allerdings gibt es in vielen Unternehmen andere interne Standards sowie Richtlinien zum Projektmanagement (vgl. Bergmann / Garrecht 2021, S. 316 ff.).

## 2.2.3 Projektarten

Grundsätzlich lassen sich Projekte in vier verschiedene Kategorien unterteilen. Darunter zählen Investitionsprojekte, Organisationsprojekte, F & E-Projekte sowie komplexe Dienstleistungsprojekte.

Projektart	Parameter			
	Ziele / Ergebnisse	Termine	Ressourcen	Kosten
Investitionsprojekte	fest	normalerweise fest	variabel	normalerweise fest
Organisationsprojekte	fest	fest	variabel	fest
F & E-Projekte	variabel	variabel	fest	variabel
Komplexe Dienstleistungsprojekte	fest	fest	fest	fest

**Tabelle 1: Übersicht der Rahmenbedingungen in verschiedenen Projektarten, Quelle: Eigene Darstellung nach Bergmann / Garrecht (2021), S. 315**

Wie in Tabelle 1 zu sehen ist, können die Projektarten in unterschiedliche Parameter eingeteilt werden. Diese Parameter sind voneinander abhängig. Bei den Investitionsprojekten sind die Ziele und Ergebnisse fest, die Termine normalerweise fest, die Ressourcen Variable und die Kosten normalerweise fest. Beispiele für solche Projekte sind die Einrichtung einer neuen Produktionshalle oder die Installation eines IT-Netzwerkes. Organisationsprojekte haben feste Ziele und Ergebnisse, feste Termine, feste Kosten und variable Ressourcen. Die Entwicklung eines neuen Marketingkonzeptes sowie die Organisation einer Messerveranstaltung oder anderen Events sind dabei Beispiele dieser Projektart. F & E-Projekte haben variable Ziele und Ergebnisse, variable Termine, variable Kosten und feste

Ressourcen. Hierzu zählen zum Beispiel die Neuentwicklung einer Maschine, eines Medikaments oder einer Software. Die komplexen Dienstleistungsprojekte haben feste Ziele und Ergebnisse, feste Termine, feste Ressource und feste Kosten. Beispiele dafür sind die Verlegung eines Braunkohlebagger und groß angelegte Instandhaltungsmaßnahmen (vgl. Bergmann / Garrecht 2021, S. 315).

## 2.2.4 Projektmanagement Tools

Heutzutage gibt es eine große Anzahl von verschiedenen Projektmanagement-Tools, die sehr nützlich sein können. Der Einsatz eines solchen Tools ist jedoch nur dann sinnvoll, wenn alle Mitglieder des Projektteams damit arbeiten. Dies ist nicht der Fall, wenn viele Projektablaufe nicht nur über das ausgewählte Tool, sondern auch per Mail und anderen Kanälen kommuniziert werden. Es ist also sehr wichtig, sich für ein Tool zu entscheiden und dieses konsequent in der Projektarbeit einzusetzen. Je früher diese Entscheidung für den Einsatz eines Tools getroffen wird, desto sicherer ist es, diese Arbeitsgrundlage beizubehalten (vgl. Seyhan 2021, S. 54).

Mittlerweile gibt es einige Projektmanagement-Tools wie zum Beispiel Trello, Asana, Bitrix24, Monday oder Zoho. Allerdings sind nicht alle gleichermaßen für die Anforderungen und den Gebrauch verschiedener Projektmitarbeiter geeignet. Je nach Wunsch und Bedarf der unterschiedlichen Teams haben alle Tools ganz unterschiedliche Stärken sowie Schwächen und bringen auch unterschiedliche Funktionen mit sich, welche sich mal mehr und mal weniger eignen. Neben Projektmanagement-Tools gibt es außerdem viele verschiedene Kommunikationstools, die ergänzend oder als Ersatz benutzt werden können. Bekannte Vertreter dabei sind Slack, Rooftop sowie Discord. Weiterhin gibt es auch Online-Whiteboards wie zum Beispiel Miro, Collaboard, Microsoft Whiteboard oder Limnu. Es lässt sich schlussfolgern, dass die Möglichkeiten nahezu grenzenlos sind und jede mögliche Art der Zusammenarbeit realisiert wird (vgl. Seyhan 2021, S. 55).

	Vorteile	Nachteile	Preis pro Monat
<b>Trello</b>	<ul style="list-style-type: none"> <li>Einfach und übersichtliche Bedienung</li> <li>Listen und Boards (Karten) mit vielen zusätzlichen Features</li> </ul>	<ul style="list-style-type: none"> <li>Kommt schnell an seine Grenzen bei komplexen Projekten</li> <li>Mangelnde Gantt Diagramm und Ressourcen-Management</li> </ul>	<ul style="list-style-type: none"> <li>Free: 0 EUR</li> <li>Standard: 4,75 EUR</li> <li>Premium: 9,50 EUR</li> <li>Enterprise: 16,60 EUR</li> </ul>
<b>Asana</b>	<ul style="list-style-type: none"> <li>Populär und einfache Bedienung</li> <li>Umfangreiches PM-Tool</li> <li>Echtzeitkommunikation und Projektportfolio</li> </ul>	<ul style="list-style-type: none"> <li>Wirkt auf den ersten Blick überladen</li> <li>Verlangt viel Zeit für eine gute Einarbeitung</li> </ul>	<ul style="list-style-type: none"> <li>Basic: 0 EUR</li> <li>Premium: 10,99 EUR</li> <li>Business: 24,99 EUR</li> </ul>
<b>Bitrix24</b>	<ul style="list-style-type: none"> <li>Professionelles Tool, dass über Projektmanagement hinaus geht</li> <li>Umfangreich, Echtzeitkommunikation, Cloud Basiert</li> </ul>	<ul style="list-style-type: none"> <li>Für Unternehmen mit komplexen Projekten</li> <li>Viele Funktionen in der kostenlosen Version nicht enthalten</li> </ul>	<ul style="list-style-type: none"> <li>Free: 0 EUR</li> <li>Basic: 49 EUR</li> <li>Standard: 99 EUR</li> <li>Professional: 199 EUR</li> </ul>
<b>Slack</b>	<ul style="list-style-type: none"> <li>Schnelle Kommunikation</li> <li>Öffentliche und Private Kanäle</li> <li>Integration mit anderen Tools und Diensten</li> </ul>	<ul style="list-style-type: none"> <li>Nachrichten können unorganisiert und chaotisch werden</li> <li>Kann oft sehr oberflächlich sein</li> </ul>	<ul style="list-style-type: none"> <li>Free: 0 EUR</li> <li>Pro: 6,30 EUR</li> <li>Business+: 11,90 EUR</li> <li>Enterprise Grid: unbekannt</li> </ul>
<b>Limnu</b>	<ul style="list-style-type: none"> <li>Sehr gute Funktionalität</li> <li>Benutzerfreundlich und intuitives Design</li> </ul>	<ul style="list-style-type: none"> <li>Verwendung eines separaten Programms neben Limnu erforderlich</li> </ul>	<ul style="list-style-type: none"> <li>Free: 0 EUR</li> <li>Standard: 4,75 EUR</li> <li>Team: 7,60 EUR</li> <li>Enterprise: unbekannt</li> </ul>

Weitere Tools: monday, Celoxis, ClickUp, Smartsheet, Forecast.app, Kintone, Easy Projects, Ravetree, OpenProject, Wrike, MeisterTask, GanttProject, Zoho Projects, Libreplan, 2-Plan, Taiga.io, Nifty, Factro, Teamwork, Excel, Discord, Freedcamp etc.

**Tabelle 2: Übersicht verschiedener Projektmanagementtools und Whiteboards, Quelle: Eigene Darstellung nach Seyhan (2021), S. 56**

Tabelle 2 zeigt eine Übersicht von drei verschiedenen Projektmanagementtools, ein Kommunikationstool sowie ein Whiteboard. Dabei wird eingegangen auf die Vorteile, Nachteile und den Preis der einzelnen Tools. Außerdem werden weitere wichtige und bekannte Tools namentlich erwähnt.

## 2.2.5 Charakteristik im Projektmanagement

Projekte sollen Probleme im Unternehmen lösen, Verbesserungen und Innovationen hervorbringen und bei der Erstellung komplexer Anlagen helfen. Die sehr hohen Erwartungen prägen die Zielsetzung des Projektmanagements, da sie eine effektive Arbeitsstruktur erfordern. Charakteristisch für die Vorgehensweise sind dabei die Ziel- und Ergebnisorientierung, das Auftragsdenken, die Ganzheitlichkeit sowie flache Hierarchien (vgl. Kraus / Westermann 2019, S. 7).

Ziel- und Ergebnisorientierung bedeutet, die Orientierungs- und Motivationsfunktion der Ziele zu nutzen und gleichzeitig die Möglichkeit für alternative Lösungswege für Innovationen zu geben. Das setzt Selbstinitiative und Selbstständigkeit auf der Seite des Mitarbeiters voraus. Den Teams wird die freie Wahl gelassen, welchen Weg sie einschlagen, nur das Ziel ist vorherbestimmt (vgl. Kraus / Westermann 2019, S. 7).

Hinter dem Gedanken, einen Auftrag zu vergeben, steckt die Führungsphilosophie des Managements nach Zielvereinbarungen. Einem Projektmitarbeiter kommt dabei eine andere Funktion zu als dem Linienmitarbeiter. Er ist dabei Auftragnehmer und wird nicht als Arbeitnehmer bezeichnet. Er ist verantwortlich für die Erfüllung seiner Aufgaben gemäß den Projektzielen und hilft bei der Definition seiner Verantwortlichkeiten. Durch die Mitwirkung an der Auftragsdefinition kann eine stärkere Identifikation mit dem Projekt erreicht werden (vgl. Kraus / Westermann 2019, S. 7).

Ganzheitlichkeit bedeutet die umfassende Betrachtung und Behandlung einer Aufgabe. Es geht nicht darum, dass jeder seinen Job erledigt, sondern wie das vorgegebene Ziel erreicht wird (vgl. Kraus / Westermann 2019, S. 7).

Flache Hierarchien sorgen für eine bessere Ausrichtung und Motivation. In der Projektarbeit sind kurze Informations- und Entscheidungswege wichtig, denn ein wesentlicher Erfolgsfaktor im Projektmanagement ist die „Personalisierung“ der Arbeit (vgl. Kraus / Westermann 2019, S. 7).

## 2.2.6 Richtlinien für Effektives Projektmanagement

Effektives Projektmanagement hängt von zwei wesentlichen Faktoren ab. Dazu gehören der Projektmanager sowie der Ablauf des Projektmanagements allgemein. Im nachfolgenden sollen nun ein optimaler Projektmanager sowie ein ideales Projektmanagement beschrieben werden.

Ein effektives Projektmanagement umfasst für den Projektmanager eine Vielzahl von Richtlinien, an welche er sich halten sollte. Zuerst gehört dazu, dass bitten der Teammitglieder um Beiträge anstatt ihnen Befehle zu erteilen. Weiterhin sollte das komplette Team mit in die Projektplanung einbezogen werden, wobei man mit dem Geplanten so flexibel wie möglich umgehen sollte. Mitarbeiter sollen ermutigt werden, ihre Probleme mit dem Projektmanager zu besprechen, um später unüberschaubare Komplikationen zu vermeiden. Außerdem ist es wichtig niemals in Panik zu geraten, wenn Probleme auftauchen und diese auch nicht persönlich zu nehmen. Vor dem Treffen wichtiger Entscheidungen sollte man sich über die Fakten informieren und prüfen, ob das Team zu einem Konsens finden kann. Des Weiteren sollte dafür gesorgt werden, dass Mitarbeiter jeden Tag mehrere Stunden



ohne Unterbrechungen arbeiten können. Dem Team gegenüber sollte man offen sein und keine Informationen verheimlichen. Weiterhin sollte man als Projektmanager nicht nur zu normalen Zeiten arbeiten, wenn man sein Team darum gebeten hat, Überstunden zu machen. Zuletzt ist es auch noch sehr wichtig, den Mitarbeitern nach Abschluss des Projektes eine angemessene Belohnung zu überreichen (vgl. Novak 2012, S. 372).

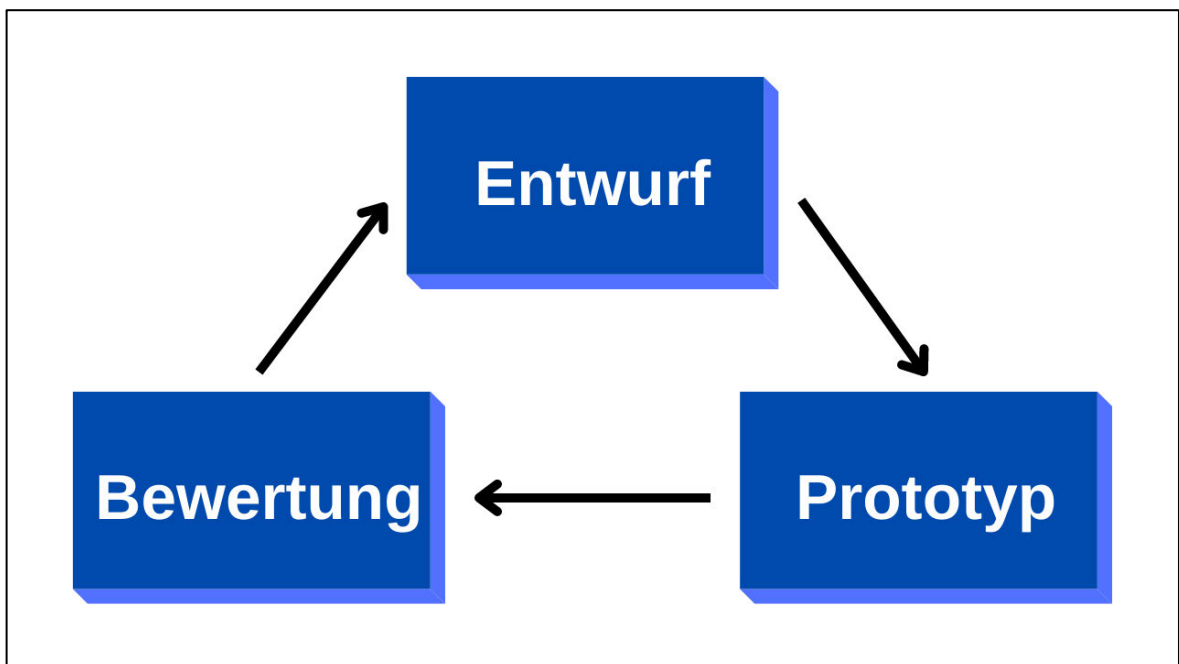
<b>Bedingungen für ein ideales Projektmanagement</b>		
<p><b>Ziel</b></p> <p>Ziele geben Aufschluss darüber, wie ein Plan zur Erreichung des Ziels aussehen soll. Sie sollten spezifisch, messbar, erreichbar, relevant und zeitgebunden sein.</p>	<p><b>Fähigkeiten</b></p> <p>Die wichtigste Ressource sind die Menschen. Gehen Sie zu denjenigen, die die Fähigkeiten, die Erfahrung und den Willen haben, auf höchstem Niveau zu arbeiten.</p>	<p><b>Ressourcen</b></p> <p>Ermitteln Sie erforderliche Ressourcen wie Kapital, Personal, Ausrüstung, Raum, Zeit sowie alles, was für die Ausführung der Aufgabe erforderlich ist. Ohne die richtigen Ressourcen wird ein Projekt nicht erfolgreich sein.</p>
<p><b>Entscheidungen</b></p> <p>Legen Sie fest, wer welche Entscheidungen treffen darf, was von einem Ausschuss geprüft werden muss und wie schnell Entscheidungen getroffen werden. Wenn zu viele Personen beteiligt sind, steigt das Risiko, in eine Analyse-Lähmung zu verfallen.</p>	<p><b>Debatten</b></p> <p>Versuchen Sie, eine Situation aus mehreren Blickwinkeln zu analysieren. Ermutigen Sie Ihr Team, alle Möglichkeiten auszuloten. Stoßen Sie gesunde Debatten an, um die gegensätzlichen Seiten zu erforschen.</p>	<p><b>Zieltag</b></p> <p>Zeit ist ein sehr wichtiger Faktor, und Ihr Team muss die Aufgaben rechtzeitig erledigen, um das Projekt bis zum Zieltermin abzuschließen. Wenn dies nicht der Fall ist, werden mehr Ressourcen benötigt, um das gleiche Ziel zu erreichen.</p>
<p><b>Untersuchung</b></p> <p>Gehen Sie alle möglichen Optionen durch. Wägen Sie die Vor- und Nachteile sorgfältig ab und versuchen Sie, die Option zu finden, die den größten Nutzen bringt.</p>	<p><b>Prioritäten</b></p> <p>Überprüfen Sie ständig die Prioritäten, denn was heute wichtig ist, ist morgen vielleicht nicht mehr so wichtig. Ihr Team muss wissen, wie sich der Kontext ändert, damit Sie den Plan entsprechend anpassen können.</p>	<p><b>Transparenz</b></p> <p>Sie müssen wissen, wer woran arbeitet und wer seine Aufgaben fristgerecht erledigt bzw. nicht erledigt. Es ist viel besser, rote Flaggen zu sehen und gemeinsam daran zu arbeiten, Probleme frühzeitig zu lösen.</p>
<p><b>Rechenschaftspflicht</b></p> <p>Dies ist nicht zu verwechseln mit Verantwortung. Teams können sich die Verantwortung teilen, aber jeder Einzelne innerhalb eines Teams ist für die Erledigung der Aufgaben selbst verantwortlich.</p>	<p><b>Rote Flaggen</b></p> <p>Schaffen Sie ein Umfeld, in dem die Mitarbeiter bereit sind, Probleme anzusprechen. Teams müssen zusammenarbeiten, um Probleme zu lösen, und einander vertrauen, um ihre wirkliche Meinung zu äußern.</p>	<p><b>Kritischer Pfad</b></p> <p>Dieser Begriff bezieht sich auf den längsten Weg zum Erreichen eines Ergebnisses, das von der Ausführung zahlreicher Prozesse abhängt. Stellen Sie sicher, dass der kritische Pfad während der gesamten Projektlaufzeit ermittelt wird.</p>

Abbildung 4: Bedingungen für ein ideales Projektmanagement, Quelle: Eigene Darstellung nach Mouawad (2014)

Ein ideales Projektmanagement umfasst eine Vielzahl von Bedingungen, an welche man sich halten sollte. In Abbildung 4 werden dazu die wichtigsten Bedingungen genannt sowie kurz beschrieben.

### 2.2.7 Projektstrategie im Game Development

Die Entwicklung von Spielen ähnelt mehr der Entwicklung von Software als andere Formen der Unterhaltung. Genauso wie Spiele keine Filme sind, ist auch der Prozess, in dem sie entwickelt werden, auch nicht mit der Filmproduktion vergleichbar. Der iterative Entwicklungsprozess, der in der Software und Web-Entwicklung verwendet wird, funktioniert auch bei der Entwicklung von Spielen sehr gut. Dieses Modell umfasst einen zirkulären dreistufigen Prozess (vgl. Novak 2012, S. 366).



**Abbildung 5: Übersicht eines dreistufigen iterativen Entwicklungsprozess, Quelle: Eigene Darstellung nach Novack (2012), S. 367**

Wie in Abbildung 5 zu sehen, gehören dazu Entwurf, Prototyp und Bewertung. Nachdem die Entwurfsphase abgeschlossen ist, wird ein Prototyp entwickelt. Sobald der Prototyp in der Evaluierungsphase ausprobiert sowie getestet wurde, entscheidet das Team, was gut funktioniert, was nicht, und kehrt dann in die Entwurfsphase zurück, ändert den Prototyp, um ihn erneut für Tests zur Verfügung zu stellen. Dieser Prozess wird je nach Bedarf wiederholt, bis das Spiel vollständig fertiggestellt ist. Der Schlüssel zu diesem Modell ist die kontinuierliche Verfeinerung des Designs von dem Spiel auf der Grundlage dessen, was bisher erstellt wurde. Jeder Prototyp sollte seinen eigenen vollständigen Entwicklungszyklus haben unter Berücksichtigung der Anforderungen sowie Ergebnisse und dem Zeitplan. Es sollen potenzielle Akteure in die Evaluierungsphase mit einbezogen werden, damit Sie die Perspektive eines Spielers in Bezug auf Benutzerfreundlichkeit, Gameplay und Spaßfaktor einnehmen können (vgl. Novak 2012, S. 367).

## 2.3 Chatbots

Chatbots erlangen im heutigen Zeitalter an immer mehr Einfluss. Sie ermöglichen uns Menschen, mit Computersystemen in natürlicher Sprache zu kommunizieren. Im einfachsten Fall extrahiert der Chatbot seine Absicht aus dem, was der Benutzer gesagt hat, fragt die Wissensdatenbank nach fehlenden Informationen ab und bereitet die Antwort für den Benutzer auf. Somit stellen Chatbots eine Schnittstelle zwischen Informationen und Nutzern dar, was in verschiedenen Unternehmen mehrere Vorteile mit sich bringen kann (vgl. Stucki et al. 2018, S. 725).

Um einen Prototyp in Form eines Chatbots zu entwickeln ist es essenziell, dieses Themengebiet vorher genauestens zu untersuchen. Dabei soll darauf eingegangen werden, was Chatbots sind, wie die Geschichte der Chatbots aussah und sie heute so beliebt geworden sind, welchen Nutzen sie für uns Menschen haben, in welche Arten sie unterteilt werden, wie sie entworfen und entwickelt werden und zuletzt noch, welche Programmierungsmöglichkeiten es für sie gibt. Dadurch soll vor allem das Potenzial und die verschiedenen Entwicklungsmöglichkeiten von Chatbots besser verstanden werden und außerdem soll es eine Grundlage für die Erschaffung des Prototyps bieten.

### 2.3.1 Einführung

Wenn man von Chatbots spricht, handelt es sich dabei um ein Computerprogramm, das wie eine intelligente Entität reagiert, wenn man sich mit diesem über Text oder Sprache unterhält. Weiterhin versteht er eine oder mehrere menschliche Sprachen, was gleichzeitig ein Teilbereich der künstlichen Intelligenz ist und Natural Language Processing (NLP) genannt wird. Innerhalb eines englischen Lexikons wird ein Chatbot definiert als „A computer program designed to simulate conversation with human users, especially over the internet“ (vgl. o. V. Definition of chatbot in English by Lexico Dictionaries). Die Bezeichnungen für Chatbots gehen weit auseinander und daher werden diese auch als Smart Bots, interaktive Agenten, digitale Assistenten oder künstliche Konversationseinheiten bezeichnet. Weiterhin können Chatbots menschliche Konversationen nachahmen, Benutzer unterhalten und nebenbei auch nützlich für Bereiche wie Bildung, Informationsbeschaffung, Wirtschaft sowie E-Commerce sein. Eine große Popularität erlangen sie durch die vielen Vorteile, welche sie für Benutzer und Entwickler bieten. Ein Großteil der Implementierungen sind plattformunabhängig und stehen den Nutzern sofort zur Verfügung, ohne dass irgendwelche Installationen vorgenommen werden müssen. Außerdem sind die Zuverlässigkeit der Kommunikation, schnelle und unkomplizierte Entwicklungsiterationen, fehlende Versionsfragmentierung und ein meist geringer Designaufwand für die Schnittstelle von großem Vorteil für Entwickler (vgl. Maglogiannis et al. 2020, S. 373 ff.).

### 2.3.2 Geschichte der Chatbots

Der Startschuss der Chatbots erfolgte 1950, als Alan Turing den Turing-Test "Können Maschinen denken?" Vorschlag, aus welchem die Idee eines Chatbots populär wurde. Der erste bekannte Chatbot war Eliza, welcher 1966 entwickelt wurde und als Psychotherapeut fungieren sollte, indem er dem Benutzer Äußerungen in Form von Fragen zurückgab. Er verwendete einen einfachen Musterabgleich und einen auf Vorlagen basierenden Antwortmechanismus. Seine Konversationsfähigkeit war nicht weit entwickelt, aber sie reichte aus, um die Menschen in einer Zeit, in der sie nicht an die Interaktion mit Computern gewöhnt waren, zu verwirren und ihnen die Motivation zu geben, andere Chatbots zu entwickeln. Eine Verbesserung brachte später ein Chatbot mit Persönlichkeit namens PARRY, welcher 1972 entwickelt wurde. Einige Zeit später, im Jahr 1995, wurde der Chatbot ALICE entwickelt, der in den Jahren 2000, 2001 und 2004 den Loebner-Preis gewann. Er war der erste Computer, der den Rang des "menschlichsten Computers" erhielt. ALICE funktionierte über einen einfachen Pattern-Matching-Algorithmus, wobei die zugrunde liegende Intelligenz auf

der Artificial Intelligence Markup Language (AIML) basiert, die es den Entwicklern ermöglicht, die Bausteine des Chatbot-Wissens zu definieren. Chatbots wie SmarterChild aus dem Jahr 2001 werden nach der Entwicklung über Messenger-Anwendungen verfügbar gemacht. Daraus folgte die Entwicklung von virtuellen persönlichen Assistenten wie Apple Siri, Microsoft Cortana, Amazon Alexa, Google Assistant und IBM Watson. Ab diesem Moment stieg das Interesse der Menschen an Chatbots rapide an (vgl. Maglogiannis et al. 2020, S. 374).

### 2.3.3 Nutzung von Chatbots

Chatbots scheinen sehr vielversprechend zu sein, wenn es darum geht, Nutzern schnelle und bequeme Unterstützung zu bieten, welche speziell auf ihre Fragen Bezugnahmen. Die häufigste Motivation für Chatbot-Benutzer ist die Produktivität, während andere Motive, Unterhaltung, soziale Faktoren und die Begegnung mit neuen Dingen sind. Um die oben genannten Interessen auszugleichen, sollte jedoch ein Chatbot gebaut werden, der gleichzeitig als Werkzeug, Spielzeug und Freund fungiert. Die Reduzierung der Kundendienstkosten und die Möglichkeit, viele Benutzer gleichzeitig bedienen zu können, ist einer der Gründe, warum Chatbots in Unternehmen so beliebt sind. Sie werden nicht mehr nur als Assistenten gesehen, sondern rücken durch ihre Interaktion dem Nutzer als ein freundlicher Begleiter immer näher. Durch maschinelles Lernen können beispielsweise Kundendienst-Chatbots Gefühle erkennen und wie ein menschlicher Mitarbeiter eine emotionale Verbindung zu Kunden aufbauen. Das Vertrauen eines Nutzers in einen solchen Bot hängt von Faktoren ab, die mit dem Chatbot selbst zusammenhängen, zum Beispiel wie menschenähnlich er reagiert, wie er sich präsentiert und wie professionell er sich verhält. Es hängt aber auch von Faktoren ab, die mit dem Kontext des Dienstes zusammenhängen, wie z. B. der Marke des Chatbot-Hosts, der Privatsphäre und Sicherheit sowie anderen Risikofaktoren im Zusammenhang mit dem Thema der Anfrage. Es bleibt also zu vermuten, dass wir bald in einer Welt leben, in der Gesprächspartner Menschen oder Chatbots sein können und in vielen Fällen es nicht auszuschließen ist, dass keiner unterscheiden kann mit was für einem Gesprächspartner er gerade kommuniziert (vgl. Maglogiannis et al. 2020, S. 375 ff.).

### 2.3.4 Arten von Chatbots

Chatbots können anhand verschiedener Parameter klassifiziert werden, darunter zählen Wissensbereich, angebotener Dienst, Ziele, Methode zur Verarbeitung von Eingaben und zur Generierung von Antworten, menschlicher Beistand sowie die Erstellungsmethode. (vgl. Maglogiannis et al. 2020, S. 377).

Die Klassifizierung auf Grundlage des Wissensbereichs berücksichtigt das Wissen, auf das ein Chatbot Zugriff hat, oder die Menge an Daten, auf die er trainiert wurde. Chatbots mit offenem Wissensgebiet können über allgemeine Themen sprechen sowie entsprechende Antworten geben, während Chatbots mit geschlossenem Wissensgebiet auf ein bestimmten Bereich fokussiert sind und möglicherweise nicht auf Fragen anderer Gebiete antworten können (vgl. Maglogiannis et al. 2020, S. 378).

Die Einordnung auf Grundlage des angebotenen Dienstes berücksichtigt die gefühlsmäßige Nähe des Chatbots zum Nutzer, das Ausmaß der Interaktion, die stattfindet und hängt auch von der Aufgabe ab, die der Chatbot ausführt. Interpersonelle Chatbots sind im Bereich der Kommunikation angesiedelt und bieten Dienste wie Restaurantbuchungen oder Flugbuchungen an. Sie sind keine Begleiter des Nutzers, sondern nehmen Informationen auf und geben sie an den Benutzer weiter. Sie können eine Persönlichkeit haben, können freundlich sein und werden sich wahrscheinlich an Informationen über den Nutzer erinnern, aber sie sind nicht dazu verpflichtet, dass sie dies tun. Intrapersonale Chatbots existieren im persönlichen Bereich des Nutzers, zum Beispiel in Chat-Apps wie Messenger, Slack oder

WhatsApp. Sie sind Begleiter des Nutzers und verstehen ihn wie ein Mensch (vgl. Maglogiannis et al. 2020, S. 378).

Die Einteilung nach den Zielen berücksichtigt das primäre Bestreben, das Chatbots erreichen wollen. Informative Chatbots sollen den Benutzer mit Informationen versorgen, die zuvor gespeichert werden oder aus einer festen Quelle verfügbar sind, wie zum Beispiel FAQ-Chatbots. Konversationelle Chatbots stellen sich dem Nutzer wie ein anderes menschliches Wesen vor, und ihr Ziel ist es, korrekt auf die ihnen gegebene Antwort zu reagieren. Aufgabenbasierte Chatbots führen spezifische Aufgaben aus, beispielsweise einen Flug zu buchen oder jemandem zu helfen. Diese Chatbots sind intelligent, wenn es darum geht, nach Informationen zu fragen und die Eingaben des Nutzers zu verstehen (vgl. Maglogiannis et al. 2020, S. 378).

Die Klassifizierung nach der Methode der Eingabeverarbeitung und Antwortgenerierung berücksichtigt die Arbeitsweise der Verarbeitung von Eingaben und der Generierung von Antworten. Es gibt drei Modelle, die verwendet werden, um die entsprechenden Antworten zu erzeugen. Dazu zählen das regelbasierte Modell, das abrufbasierte Modell und das generative Modell. Chatbots mit regelbasiertem Modell sind die Art von Architektur, mit der die meisten konstruiert werden. Sie wählen die Antwort des Systems auf der Grundlage eines fest definierten Satzes von Regeln aus, die auf der Erkennung der komplexen Form des eingegebenen Textes basieren, ohne neue Textantworten zu erstellen. Ein wenig anders ist dagegen das abrufbasierte Modell, das mehr Flexibilität bietet, da es verfügbare Ressourcen mithilfe von APIs abfragt und analysiert. Ein abrufbasierter Chatbot ruft einige Antwortkandidaten aus einem Index ab, bevor er den passenden Ansatz für die Antwortauswahl anwendet. Das generative Modell generiert Antworten auf eine andere Art und Weise als die anderen drei Modelle, und zwar basierend auf aktuellen und früheren Nutzernachrichten. Diese Chatbots kommen den Menschen am ähnlichsten und nutzen Algorithmen des maschinellen Lernens und Techniken des Deep Learning (vgl. Maglogiannis et al. 2020, S. 378).

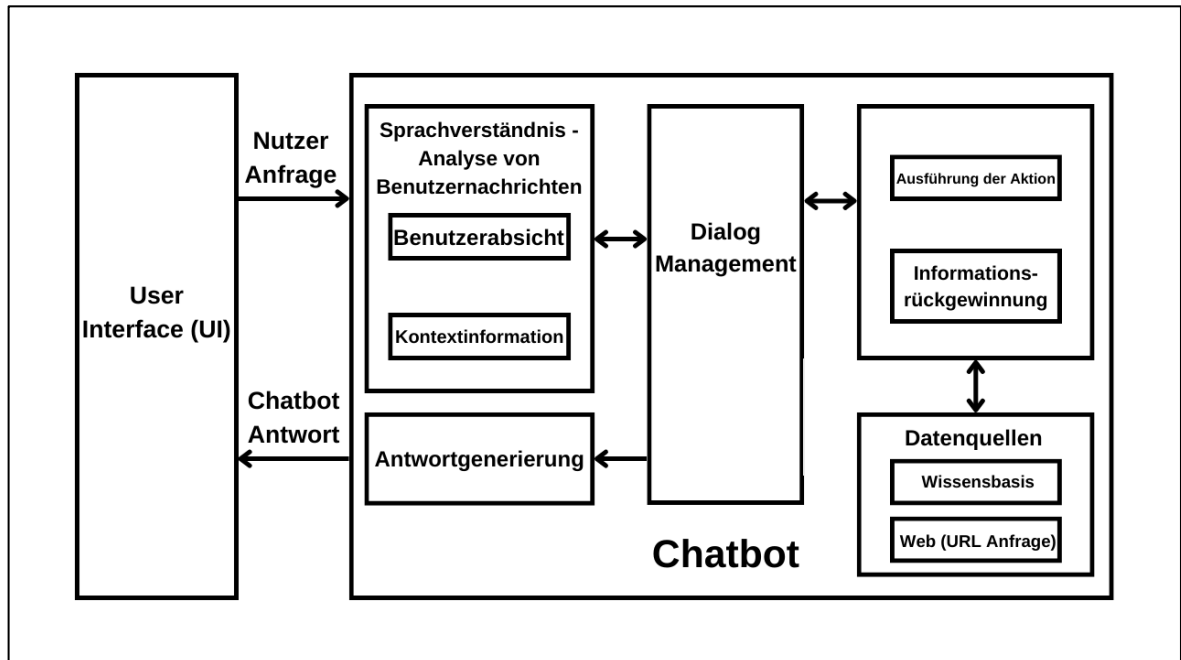
Eine weitere Unterteilung für Chatbots berücksichtigt den Anteil menschlicher Hilfe an den Komponenten. Menschlich unterstützte Chatbots nutzen die menschliche Computertechnik für ein einziges Element. Freiberufler oder Vollzeitangestellte können ihre Intelligenz in die Chatbot-Logik einbringen, um die Lücken zu füllen, die durch die Einschränkungen vollautomatischer Chatbots entstehen. Zwar bietet die menschliche Berechnung im Vergleich zu regelbasierten Algorithmen und maschinellem Lernen mehr Flexibilität und Robustheit, doch kann sie eine bestimmte Information nicht so schnell verarbeiten wie eine Maschine, was die Bearbeitung weiterer Nutzeranfragen erschwert (vgl. Maglogiannis et al. 2020, S. 379).

Chatbots können auch nach den von ihrer Entwicklungsplattform bereitgestellten Berechtigungen eingeteilt werden. Open-Source-Plattformen bieten beispielsweise dem Chatbot-Designer die Möglichkeit, in die meisten Aspekte der Implementierung einzugreifen. Geschlossene Plattformen fungieren in der Regel als Black Box, was je nach Projektanforderungen ein erheblicher Nachteil sein kann. Außerdem kann man davon ausgehen, dass Chatbots, die auf der Grundlage von Plattformen großer Unternehmen entwickelt werden, von der großen Menge an Daten, die diese Unternehmen sammeln, profitieren können (vgl. Maglogiannis et al. 2020, S. 379).

### **2.3.5 Design und Entwicklung von Chatbots**

Der Entwurf und die Entwicklung eines Chatbots umfassen eine Vielzahl von verschiedenen Techniken. Zu verstehen, was der Chatbot bieten soll und in welche Kategorie er fällt, hilft den Entwicklern bei der Auswahl der Algorithmen bzw. der Plattformen und Tools für die Erschaffung. Gleichzeitig hilft es auch den Endnutzern zu verstehen, was sie erwarten kön-

nen. Die Anforderungen für die Entwicklung eines Chatbots umfassen eine genaue Wissensrepräsentation, eine Strategie zur Generierung von Antworten und eine Reihe von vordefinierten neutralen Antworten, die gegeben werden können. Der erste Schritt bei der Entwicklung eines Systems besteht darin, es nach einem Standard in seine Bestandteile zu zerlegen, sodass ein modularer Entwicklungsansatz verfolgt werden kann (vgl. Maglogiannis et al. 2020, S. 379).



**Abbildung 6: Allgemeine Chatbot-Architektur, Quelle: Eigene Darstellung nach Maglogiannis et al. (2020), S. 380**

In Abb. 6 wird eine allgemeine Chatbot-Architektur vorgestellt. Der Prozess beginnt mit der Anfrage eines Benutzers an den Chatbot zum Beispiel über eine Messenger-App wie Facebook, Slack, WhatsApp, Skype oder einer anderen App mit Text- oder Spracheingabe wie Amazon Echo. Nachdem der Chatbot die Benutzeranfrage erhalten hat, analysiert die Sprachverständniskomponente diese, um die Absicht des Benutzers und die damit verbundenen Informationen abzuleiten. Sobald ein Chatbot die bestmögliche Interpretation erreicht hat, muss er entscheiden, was sein weiteres Vorgehen ist. Er kann direkt auf die neuen Informationen reagieren, alles, was er verstanden merken und abwarten, was als Nächstes passiert, weitere Kontextinformationen anfordern oder um eine Erklärung bitten. Wenn die Anfrage verstanden wurde, werden Aktionen ausgeführt und Informationen abgerufen. Der Chatbot führt die angeforderten Aktionen aus oder ruft die gewünschten Daten aus seinen Datenquellen ab, bei denen es sich um eine Datenbank, die sogenannte Wissensbasis des Chatbots, oder um externe Ressourcen handeln kann, auf die über einen API-Aufruf zugegriffen wird. Nach dem Abruf verwendet die Antwortgenerierungskomponente die natürliche Sprachgenerierung, um eine menschenähnliche Antwort an den Benutzer zu erstellen, die auf der Absicht und den Kontextinformationen basiert, die von der Komponente zur Analyse der Benutzernachricht zurückgegeben wurde. Die Dialog Management Komponente hält und aktualisiert den Kontext der Konversation, das heißt die aktuelle Absicht identifizierte Entitäten oder fehlende Entitäten, die zur Erfüllung der Benutzeranfragen erforderlich sind. Darüber hinaus fordert sie fehlende Informationen an, verarbeitet Klarstellungen der Benutzer und stellt Folgefragen (vgl. Maglogiannis et al. 2020, S. 379 ff.).

### 2.3.6 Programmierung von Chatbots

Für die Entwicklung eines Chatbots stehen viele kommerzielle und Open-Source-Optionen zur Verfügung. Die Zahl der Chatbot-Technologien ist bereits überwältigend und wächst täglich. Chatbots werden bisher auf zwei Arten entwickelt: mit einer Programmiersprache wie Java, Clojure, Python, C++, PHP, Ruby und Lisp oder mit modernen Plattformen. Derzeit gibt es sechs führende Natural Language Understanding (NLU, dt. Das Verstehen natürlicher Sprache) Cloud-Plattformen, die Entwickler nutzen können, um Anwendungen zu erstellen, dazu gehören DialogFlow von Google, wit.ai von Facebook, Microsoft LUIS, IBM Watson Conversation, Amazon Lex und SAP Conversation AI. Alle diese Plattformen werden durch maschinelles Lernen unterstützt (vgl. Maglogiannis et al. 2020, S. 380).

## 2.4 Pipedream

Ein gutes Projektmanagement erfordert meistens viele verschiedene Programme, Webseiten sowie andere Tools, um zu funktionieren. Darunter zählen zum Beispiel Trello, Git, oder auch Asana. Durch diese Vielzahl an Anwendungen ist es manchmal schwer, den Überblick zu behalten. Dementsprechend ist das Ziel, die gesamte Projektmanagement abhängige Software an einem zentralen Ort zu vereinen. Problem an der ganzen Sache sind die oft schwierigen Zugriffsmöglichkeiten auf die API dieser einzelnen Anwendungen. Diese sind oftmals sehr begrenzt, sodass es so gut wie unmöglich ist, auf diese zuzugreifen. Dies sorgt für Aufsehen und das Verärgern vieler verschiedener Entwickler. Aus diesen Problemen heraus haben sich verschiedene Webseiten und Anwendungen gebildet, welche als Ziel haben, Entwicklern so angenehm und einfach wie möglich diese Schnittstellen zur Verfügung zu stellen. Eine sehr bekannte und auch viel genutzte Webseite ist dabei Pipedream.

### 2.4.1 Was ist Pipedream

Pipedream ist eine Integrationsplattform für Entwickler und eines der schnellsten Wege, jeden Prozess zu automatisieren, der APIs miteinander verbindet. Dabei bietet es eine kostenlose, gehostete Plattform für die Verbindung von Softwaresystemen und die Entwicklung ereignisgesteuerter Automatisierungen. Pipedream verfügt über mehr als 700 vollständig integrierte Anwendungen, sodass es möglich ist, mithilfe vorgefertigter Komponenten schnell eine Verbindung herstellen zu können, um beispielsweise Events aus Git abzurufen, eine neue Zeile zu Google Sheets hinzuzufügen und noch vieles mehr. Ebenfalls ist es möglich, benutzerdefinierten Node.js-, Python-, Golang- oder Bash-Code mit einzubinden (vgl. o. V. Introduction to Pipedream).

### 2.4.2 Entwicklung mit Pipedream

Um Pipedream mit all seinen Möglichkeiten verwenden zu können, ist es zunächst notwendig einen kostenlosen Account zu erstellen. Ist das erledigt und die Anmeldung abgeschlossen, landet man auch schon auf der Startseite. Hier wird nun ein neuer Workflow (dt. Arbeitsablauf) erstellt. Jeder Workflow beginnt mit einem Trigger (dt. Auslöser), welcher der Nutzer aus einer Vielzahl von verschiedenen Vorlagen auswählen kann. Dieser Trigger initiiert die Auslösung eines Workflows, was bedeutet so oft wie der Trigger ausgelöst wird, wird auch der Workflow gestartet. Ein Beispiel hierfür wäre ein Hypertext Transfer Protocol (HTTP) Trigger, welcher HTTP-Anfragen akzeptiert. Jedes Mal, wenn ein Nutzer an diese eindeutige Uniform Resource Locator (URL) eine HTTP-Anfrage sendet, wird auch der Workflow ausgeführt. Um ein gewünschtes Ergebnis begutachten zu können, müssen noch Aktionen sowie Codeschritte betrachtet werden, welche die Logik des Workflows steuern. Der Begriff Aktionen beschreibt vorgefertigte Codeschritte, mit denen eine Verbindung zu Hunderten von APIs hergestellt werden können, ohne dabei eigenen Code schreiben zu

müssen. Sollte doch mehr Kontrolle nötig sein, als die Standardaktionen bieten, ist es möglich, mit den Codeschritten beliebigen eigenen Node.js-Code zu schreiben und auszuführen (vgl. o. V. Pipedream Workflows and Steps).

The image shows a Pipedream workflow configuration with two main sections: a Trigger and an Action.

**Trigger Section:** Titled "Trigger" in red. It shows a "New Commit (Instant)" trigger. A red box highlights the configuration details: Component: New Commit (Instant) view on Github, Apps: Gitlab, GitLab Account: thorschnor7@gmail.com, Project ID: thorschnor7/beta-bot-project (35751477), and Branch Name: master. Red arrows point to these details with the text "Ausführen wenn ein neuer Commit gemacht wurde" and "Login Daten für GitLab Repository".

**Action Section:** Titled "Aktion" in blue. It shows a "send\_message\_advanced" action. A blue box highlights the configuration details: Discord Webhook Account: thorschnor7@gmail.com, Message: Ein neuer Commit wurde gemacht von: {{steps.trigger.event.authorName}} \n Uhrzeit: {{steps.trigger.event.authoredDate}} \n --- Nachricht --- {{steps.trigger.event.message}}, Thread ID: -, Username: beta Bot, Avatar URL: https://pbs.twimg.com/profile\_images/1388950791369732097/JYm4C9qe\_400x400.jpg, Include Link To Workflow: false, and Embeds: {{ [{"title": "Ein neuer Commit wurde gemacht!", "description": "Ersteller: \*\*steps.trigger.event.committerName\*\* \n Uhrzeit: \*\*steps.trigger.event.authoredDate\*\* \n Nachricht: \*\*steps.trigger.event.message\*\*"} ] }}. A blue arrow points to the message field with the text "Verknüpfter Discord Account" and "Nachricht, welche in Discord gesendet wird".

Abbildung 7: Beispiel eines Pipedream Workflows, Quelle: Eigene Darstellung unter Zuhilfenahme von Pipedream

In Abbildung 7 wird ein typischer Aufbau eines Pipedream Workflows gezeigt. In diesem Beispiel wird eine Verbindung von dem Versionsverwaltungstool GitLab über Pipedream bis hin zur Kommunikationsplattform Discord realisiert. Dabei wird innerhalb des Trigger Bereichs vereinbart, dass der Workflow gestartet werden soll, sobald unter dem angegebenen GitLab Repository ein Commit stattfindet. Ist das erfolgt, wird im Bereich Aktion eine Nachricht über den verknüpften Chatbot Account auf Discord gesendet. Dabei lässt sich auch auf verschiedene Variablen des Triggers zugreifen wie zum Beispiel auf die Information von wem der Commit stammt und wann er stattgefunden hat.

## 2.5 Kommunikationsplattform Discord

„Stell dir einen Ort vor, an dem du Teil eines Schulkclubs, einer Gaming-Gruppe oder einer weltweiten Kunst-Community sein kannst. Ein Ort, an dem du einfach Zeit mit Freunden verbringen kannst. Ein Ort, an dem es leicht ist, sich jederzeit zu treffen und zu unterhalten“ (vgl. o. V. Discord Hauptseite).

Die Kommunikationsplattform Discord hat sich über die letzten Jahre zu einer der größten Anwendungen in seiner Branche entwickelt. Dabei liegt es nahe, dass auch der Funktionsumfang, welche die Plattform bietet, ständig weiter fortschreitet. Eine besonders herausstechende und sehr weit entwickelte Funktion sind dabei die Discord Chatbots. Aus diesem Grund wurde Discord auch als Anwendung für den Prototypen ausgewählt und muss im



nachfolgenden in verschiedenen Bereichen genau analysiert werden. Dabei wird eingegangen auf allgemeine Fakten, wer die Konsumenten sind, was Discord Chatbots sind und welchen Funktionsumfang diese mitbringen.

### 2.5.1 Allgemeines

Veröffentlicht wurde die Kommunikationsplattform Discord im Mai 2015 unter dem Domain-Namen „discordapp.com“. Discord ist eine proprietäre Voice over IP (VoIP) Anwendung sowie digitale Distributionsplattform, welche speziell für Videospiele-Communities entwickelt wurde. Sie erlaubt Text-, Bild-, Video- und Audiokommunikation zwischen Benutzern, welche sich in einem Chat Kanal befinden. Die Anwendung steht für verschiedene Systeme zur Verfügung, darunter befindet sich Windows, macOS, Android, iOS, Linux und sogar diverse Webbrowser. Alle wichtigen Grundfunktionen von Discord sind komplett kostenlos und für jeden frei zugänglich. Erweiterte aber für den normalen Benutzer irrelevante Funktionen erfordern jedoch ein kostenpflichtiges Discord Nitro Abonnement. Dieses kann entweder für 9,99\$ (umgerechnet ca. 9,31€) im Monats- oder für 99,99\$ (umgerechnet ca. 93,16€) im Jahresabonnement erworben werden. Die Erstellung und Wartung eines Discord Servers ist ebenfalls mit keinen Kosten verbunden und auch für technisch unerfahrene Menschen aufgrund von vielen Onlinehilfen einfach zu bewerkstelligen. Als Server werden dabei eine Sammlung von dauerhaften privaten oder Gemeinschaftlichen, Chaträumen und Voice-Chat-Kanälen bezeichnet (vgl. Cacho 2020, S. 2).

### 2.5.2 Wer sind die Konsumenten

Um zu verstehen, wer die Konsumenten sind, ist es naheliegend, sich anzuschauen, welche Interessengebiete die einzelnen Discord Nutzer verfolgen. Doch bevor man sich diese anschauen kann, muss zunächst geklärt werden wie viele registrierte sowie aktive Nutzer Discord über die einzelnen Jahre aufgebaut hat.

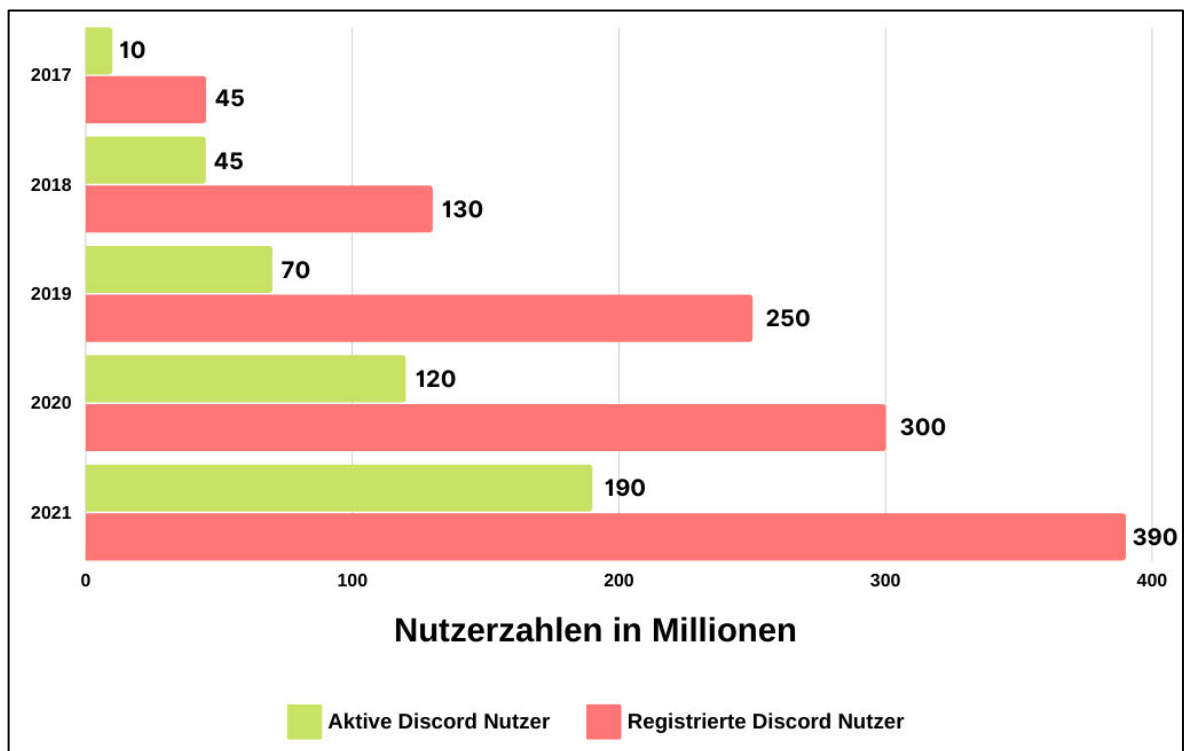
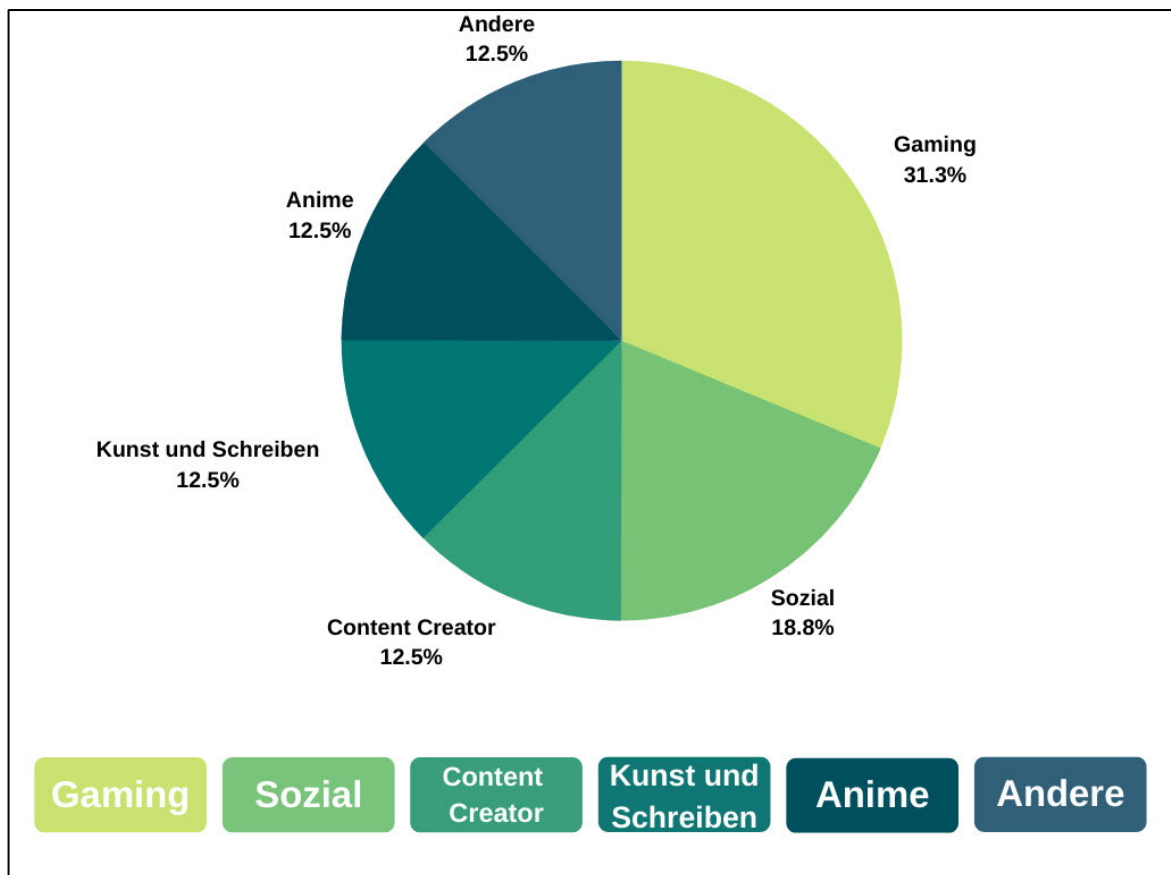


Abbildung 8: Discord Nutzerzahlen der Jahre 2017-2021, Quelle: Eigene Darstellung nach Campbell (2022)

Wie in Abbildung 8 zu sehen, konnte sich Discord bereits im Jahr 2017 eine beachtliche Summe von 45 Millionen registrierten Benutzern sowie 10 Millionen aktiven Benutzern aufbauen. Dies stieg über die Jahre rasant an und so konnte man 2019 stolze 250 Millionen registrierte Nutzer sowie 70 Millionen aktive Benutzer erfassen. Im letzten Jahr schlug man aber noch einmal alle bisher aufgezeichneten Benutzerdaten und so konnte man insgesamt 390 Millionen registrierte- und 190 Millionen aktive Benutzer vorweisen (vgl. Campbell 2022).

Da es keine offiziellen Statistiken darüber gibt, welche Interessen die einzelnen Nutzer nachverfolgen, können wir dies nur anhand der Typen von bisher erstellten Discord Servern abschätzen.



**Abbildung 9: Discord Servertypen mit den meisten Nutzern, Quelle: Eigene Darstellung nach Benet (2021)**

Abbildung 9 kann entnommen werden, dass der Gamingbereich, welcher gleichzeitig der Hauptfokus von Discord ist, mit 31,3% den höchsten Anteil aller Server beansprucht. Mit 18,8% kommt dahinter der soziale Bereich. Die Bereiche Anime, Kunst und Schreiben sowie Content Creator (Ersteller von Inhalten) liegen mit 12,5% gleich auf. Die restlichen 12,5% aller in Discord erstellten Server zählen zu der Kategorie andere (vgl. Benet 2021).

Es lässt sich also schlussfolgern, dass fast ein Drittel aller Discord Nutzer Gamer sind. Allerdings wird die Plattform von immer mehr Menschen anderer Interessen genutzt, wie zum Beispiel im sozialen Bereich. Daraus folgt, dass sich Discord schon längst nicht mehr nur an den Gamingbereich richtet. Heutzutage ist es für viele Menschen schon Alltag, Discord auch auf ganz andere Weise zu benutzen. Zum Beispiel werden während der Covid-19-Pandemie ganze Unterrichtseinheiten von Schulen in Discord übertragen, welche am Ende

auch sehr positiv angekommen sind. Discord ist mittlerweile mehr als nur eine Kommunikationsplattform für Gamer, wie es ursprünglich gedacht war und es bleibt gespannt zu beobachten, wie sie sich in den nächsten Jahren entwickeln wird.

### **2.5.3 Discord Chatbot**

Durch Discords stetiges Wachstum und der Tatsache einer steigenden weltweiten Beliebtheit, sind automatisierte Prozesse wie das Verbannen unangemessener Nutzer und die Reaktion auf Nutzeranfragen für das Gedeihen und Wachsen einer Community unerlässlich. Automatisierte Programme, die wie Benutzer aussehen und sich wie diese verhalten und automatisch auf Ereignisse und Befehle in Discord reagieren, werden Bot User (dt. Bot Benutzer oder Bots) genannt. Discord Bots haben im heutigen Entwicklungsstand nahezu unbegrenzte Einsatzmöglichkeiten. Es gibt sie fast überall auf Discord, wobei jeder Chatbot für einen ganz bestimmten Zweck gemacht wurde. Darunter zählt zum Beispiel das Abspielen von Musik anderer Plattformen wie YouTube oder Spotify sowie das Generieren von Tokens oder aber auch ein RPG-Text-Spiel. Es ist auch möglich, dass der Chatbot automatisch auf den Beitritt eines neuen Mitglieds auf einem bestimmten Server reagiert. Dabei kann man sein Verhalten sogar kontextabhängig anpassen und steuern, wie er mit jedem neuen Benutzer interagieren soll. In einer Welt, in der Kommunikation für viele Menschen bedeutend ist, ist es wichtig, dass Plattformen, die soziale und kommunikative Funktionen anbieten, interessant und interaktiv sind. Discord Bots haben im Vergleich zu normalen Benutzern kein Limit, auf wie vielen Servern sie gleichzeitig sein können (vgl. Anirudh et al. 2021, S. 532 ff.).

### **2.5.4 Discord Chatbot Funktionen**

Wie bereits erwähnt, ist der Funktionsumfang eines Discord Bots sehr umfassend. Einige wichtige Funktionen sind zum Beispiel das Posten sowie Abrufen von Texten, Bildern und Videos aus verschiedenen Plattformen wie Reddit oder Twitter. Auch das aktive Lesen von Nachrichten und Chatverläufen eines Servers sowie dass aus diesen Texten erkennen von Befehlen mit anschließender Reaktion darauf ist mit einem Discord Bot möglich. Außerdem können Mitgliedern Namen und verschiedene vom Bot selbst erstellte Rollen zugewiesen werden und auch Erinnerungen können für verschiedene Anlässe erstellt werden. Weiterhin können Bots benutzerspezifische Umfragen erstellen und diese auch auswerten. Im Grunde genommen können sie die komplette Benutzerverwaltung eines Servers übernehmen, dazu zählt das Verbannen unangebrachter Nutzer, die Moderation von Textchats und das direkte Anschreiben von bestimmten Benutzern, wenn diese etwas auf dem Server erreicht haben. Sie können auf Datenbanken zugreifen und wichtige Daten darin speichern beziehungsweise abrufen. Anders als normale Benutzer können Discord Bots auch Nachrichten in Form eines Discord „Embeds“ (dt. Einbettung) senden, welche im Vergleich zu normalen Nachrichten einen viel höheren Formatierungs- sowie Gestaltungsaspekt bieten und so übersichtlicher und anschaulicher den Benutzern zur Verfügung gestellt werden können. Auch auf interaktive Elemente wie Buttons, Drop Down Menüs und Texteingabefelder haben Discord Bots Zugriff und können diese auch nach Belieben gestalten (vgl. Anirudh et al. 2021, S. 534).

Dies ist längst nicht alles, wozu ein Discord Bot benutzt werden kann, aber zumindest sind die wichtigsten und für diese Bachelorarbeit relevantesten Funktionen genannt.

## 2.6 beta Mittweida

Da der im Rahmen dieser Arbeit entwickelte Prototyp in Zukunft seinen Einsatz in einem größeren Projekt finden soll und die beta Mittweida sehr nah an diese Vorstellung herankommt, ist es sehr interessant sowie informativ, diese für den weiteren Verlauf genauer zu betrachten. Dabei ist es bedeutend herauszufinden, was die beta eigentlich ist und wie dessen bisheriger Projektmanagement Bereich aussieht. Außerdem soll basierend auf diesem Gebiet, auf die verwendete Software eingegangen sowie aus verschiedenen Gesprächen mit Projektmanagern, deren Probleme abgeleitet werden.

### 2.6.1 Was ist die beta

Die „Base of Entertainment, Technology and Arts“ (beta) ist eine Veranstaltung, welche einmal im Jahr in Mittweida stattfindet. Dabei bekommen alle Studierende des Abschlussjahrgangs aus der Studienrichtung Medieninformatik und Interaktives Entertainment die Aufgabe gestellt, ein Videospiel zu entwickeln und es auf der eigenen Abschlussveranstaltung zu präsentieren. Außerdem werden dazu namhafte Mitglieder der Games-Branche eingeladen, welche über Erfahrungen aus der Praxis berichten und so einen kleinen Einblick in die Spielindustrie bieten (vgl. o. V. Das Event für interaktive Unterhaltung).

Alle Entwickler werden dabei in unterschiedliche Teams und damit auch Aufgabenbereiche eingeteilt. Den Anfang macht das Team „Gamedesign“, welches die geistigen Schöpfer der Spielidee sind und außerdem das Spielkonzept entwickeln. Team „Art“ kümmert sich um den optischen Teil des Spiels und sorgt für eine ansprechende Darstellung der Spielidee. Im „Tech“ Team wird mithilfe der Ideen und den grafischen Elementen dafür gesorgt, dass alles in der Spieleentwicklung Software zum Leben erwacht und funktioniert. Für die Erstellung von Arbeitsplänen, verteilen der Stunden und Termine sowie der Vermeidung von Chaos setzt sich das Projektmanagement Team ein. Als letztes gibt es noch das Event und Marketingteam, dessen Aufgabe daraus besteht, sich um Social Media, Sponsoring, Finanzplanung und der Organisation des gesamten Events zu kümmern (vgl. o. V. Das Team der beta 2022).

Unter Einbeziehung von Abschnitt 2.2.3 lässt sich schlussfolgern, dass es sich bei der beta um ein Organisationsprojekt handelt. Das liegt daran, dass es während des Projektes feste Ziele und Ergebnisse, Termine sowie Kosten gibt. Ein weiterer Hinweis sind außerdem die variablen Ressourcen. Es lässt sich weiterhin erkennen, dass der zu entwickelnde Prototyp innerhalb dieser Projektart für einen optimalen Gebrauch einzuordnen ist.

### 2.6.2 Verwendete Projektmanagement Tools

Die beta Mittweida verwendet hauptsächlich den Aufgaben-Verwaltungs-Onlinedienst Trello für alles, was mit den verschiedenen Aufgaben zu tun hat. Darunter zählt zum Beispiel Aufgabenverwaltung sowie Aufgabenverteilung an die Entwickler. Diese Aufgaben können dann aus den zuvor vom Projektmanagement erstellten Karten entnommen, bearbeiten und als erledigt gemeldet werden. Weiterhin gibt es neben dem Trelloboard für die Entwickler auch noch ein eigenes für das Projektmanagement, wo alle Aufgaben, die aktiv in der jeweiligen Woche existieren, gesammelt und noch einmal genau überprüft werden. Es folgt eine Begutachtung, welche Anforderungen abgearbeitet sind und ob irgendwelche Fehler existieren, wie zum Beispiel das unabsichtliche Abhaken einer Aufgabe, welche noch nicht erledigt ist. Außerdem arbeitet die beta mit einigen Excel-Tabellen, welche in verschiedenen Bereichen zum Einsatz kommen. Bei einer dieser Tabellen handelt es sich um einen Wochenplan, in welchem alle anstehenden Meetings mit Datum und Uhrzeit enthalten sind. Zu diesen Meetings können die Entwickler auch noch mal ihre Wünsche und Probleme ansprechen, welche dann vom Projektmanagement Team bearbeitet werden. Eine weitere Tabelle ist die TP Liste, welche alle Informationen über die erreichten TP jedes

einzelnen Entwicklers enthält und jede Woche vom Projektmanagement Team aktualisiert wird. Weiterhin gibt es einen Zeitplan für das ganze Semester, welches in drei Bereiche unterteilt ist. Diese Bereiche sind Meilensteine, Sprints und Wochenplanung. Darüber hinaus gibt es auch Entwicklerberichte, welche in Google Forms erstellt werden. Darin sind die Aufgaben pro Woche der einzelnen Entwickler enthalten sowie eine Rückmeldung, ob alle TP versendet sind oder ob es irgendwo Abzug gab. Weiterhin gibt es jede Woche einen Wochenrückblick, in dem jeder Entwickler erwähnt, wie es ihm geht, was für einen Fortschritt er im Team gemacht hat und ob er Probleme mit irgendwas hat. Neben all diesen Tools verwendet die beta außerdem noch Git für die Versionskontrolle des Projektes und zur Übersicht der programmiertechnischen Umsetzung (vgl. Bischoff 2022).

Im Hinblick auf Abschnitt 2.2.4 lässt sich feststellen, dass die beta auf gängige Projektmanagement- sowie Kommunikationstools wie zum Beispiel Trello, Excel und Discord zurückgreift. Mit Blick auf die enorme Anzahl an verschiedenen Tools nutzt die beta zwar nur wenige, aber für das Projektmanagement ist dies vollkommen ausreichend. Mithilfe von Whiteboards arbeitet die beta allerdings nicht.

### **2.6.3 Probleme der beta**

Eines der größten Probleme der beta ist, dass an vielen Stellen mangelnde Kommunikation herrscht. Es fehlen einfach viele Informationen, welche an manchen Punkten für Probleme im Projekt sorgen. Außerdem sind viele Projektmanager nicht streng genug mit den Entwicklern. Dadurch entstehen in einigen Bereichen Zeitprobleme und Rückschläge im Projekt. Die Projektmanager müssen aktiv darauf hingewiesen werden, dass sie mehr Druck bei den Entwicklern ausüben und nicht immer alles einfach so hinnehmen. Ein weiteres Problem ist, dass sich einige der Projektmanager überarbeitet haben und nebenbei auch noch die Aufgaben des Heads mit übernommen haben, was nicht ihre Aufgabe ist. Das lag auch daran, dass viele Aufgaben nicht vollständig und ordentlich geklärt waren zwischen Entwickler und Head. Ein weiteres Problem ist, dass viele Entwickler nicht richtig mit Trello und Git arbeiten. Diese schauen oft nicht in dieses Tool hinein und wissen dann oft nicht, was eigentlich Sache ist. Weiterhin ist es problematisch, dass man nicht wirklich weiß, wo die einzelnen Entwickler im Projekt stehen, dies könnte zum Beispiel durch Erstellen von TP Diagrammen sehr gut dargestellt werden und sehr weiterhelfen (vgl. Bischoff 2022).

Vergleicht man die beta mit den in Abschnitt 2.2.6 aufgeführten Richtlinien für ein optimales Projektmanagement, fällt auf, dass es dabei an einigen Stellen zu großen Komplikationen kommt. Nachfolgend sollen einige Beispiele genannt werden. Das erste Problem bezieht sich dabei auf den Punkt Fähigkeiten. Ein optimales Projektmanagement setzt voraus, dass Menschen im Projekt mitarbeiten, welche sehr gute Fähigkeiten, viel Erfahrung und Arbeit auf dem höchsten Niveau in ihrem Bereich mitbringen. Bei der beta handelt es sich jedoch um noch lernende sowie meist unerfahrene Menschen in ihrem Fachgebiet. Ein weiterer wichtiger Ansatz ist der Zieltag. Dabei wird vorausgesetzt, dass alle Aufgaben bis zum Zieltermin abgeschlossen werden müssen. Dies ist bei der beta jedoch oft schwierig, da man sich oft bei der Einhaltung der Termine verschätzt. Außerdem ist Transparenz auch eine wichtige Richtung, bei der es darum geht, immer vollen Überblick darüber zu haben, wer woran gerade arbeitet und wer seine Aufgaben fristgerecht erledigt. Auch das ist bei der beta ein großes Problem, da es an vielen Stellen zu schlechter bis keiner Kommunikation kommt und oft nicht streng genug mit der Abnahme der Aufgaben umgegangen wird. Natürlich können noch weitere Gebiete verglichen werden, in welchen große Unterschiede bemerkbar sind, jedoch soll dies für einen Einblick ausreichen.

## 3 Analyse und Konzeption

In Verbindung mit der beta Mittweida soll ein Prototyp entwickelt werden, welcher in der Lage sein soll, dem Projektmanagement Team verschiedene Aufgaben und Arbeitsprozesse zu erleichtern beziehungsweise diese vollständig zu übernehmen. Weiterhin soll dieser auch zur Motivations- sowie Leistungssteigerung der gesamten Projektteilnehmer beitragen. In diesem Kapitel werden Anwendungsszenario und Anforderungen an den zu realisierenden Prototyp im Rahmen der Anforderungsanalyse beschrieben. Anschließend wird dazu ein Konzept für den Prototyp angefertigt.

### 3.1 Anforderungsanalyse

Im Nachfolgenden soll nun eine Anforderungsanalyse erstellt werden, um die Ansprüche an das zu entwickelnde System festzustellen, einzuteilen und zu prüfen. Dabei wird zuerst ein mögliches Anwendungsszenario mit Bezug zur beta erstellt. Weiterhin sollen die Anforderungen an den Prototyp geklärt und mithilfe von funktionalen und nicht funktionalen Anforderungen beschrieben werden.

#### 3.1.1 Anwendungsszenario

Das Hauptmotiv dieser Arbeit besteht darin, den Benutzenden einen Assistenten in Form eines Chatbots zur Verfügung zu stellen. Dieser unterteilt sich dabei in zwei wichtige Aufgabenbereiche. Zum einen soll er Projektmanagern bei ihren Aufgaben unterstützen, diese ihnen abnehmen oder zumindest stark vereinfachen. Zum anderen soll er bei allen Nutzern durch spielerische Funktionen zu einer Leistungs- und Motivationssteigerung beitragen. Im nachfolgenden Abschnitt wird ein Anwendungsszenario mit besonderem Wert auf den Hauptaspekt Assistenz beschrieben. Im folgenden Szenario wird die Interaktion zwischen Benutzer und Prototyp dargestellt.

Der Benutzer „Anton“ befindet sich aktuell im vierten Semester seines Studiengangs Medieninformatik und Interaktives Entertainment an der Hochschule Mittweida. Das Besondere an diesem Semester ist der Start des beta Projekts. Anton hat sich für das Projektmanagement Team entschieden und merkt schnell, dass seine Aufgaben sehr umfangreich sowie auch zeitaufwendig sind. Viele dieser Aufgaben, denkt sich Anton, müssen doch im heutigen Zeitalter einfacher mit weniger bis gar keinen Aufwand sowie automatisch zu erledigen sein. Dazu kommt noch das Antons Motivation und Leistungsbereitschaft für das Projekt etwas am Schleifen sind, weil ihn diese ganzen Herausforderungen über einen längeren Zeitraum stark ausgelastet haben. Glücklicherweise benutzt die beta Mittweida die Kommunikationsplattform Discord und Anton entscheidet sich dafür, den beta Bot auf den Discord Server einzuladen.

Nachdem der Chatbot auf dem beta Discord Server angekommen ist, beginnt Anton auch schon mit der Einrichtung. Dabei besucht er mithilfe seines Webbrowsers die Plattform Pipedream. Hier verbindet er den von der beta genutzten Trello und Git Account über die vorgefertigten Workflows mit dem Chatbot. Weiterhin möchte Anton wissen, an welchem Tag eigentlich die nächste beta stattfindet und welche Person aktuell die meisten TP im Verlauf des Projektes gesammelt hat.

Die Interaktion zwischen dem Chatbot (C) und Anton (A) kann für dieses Beispiel anhand eines kleinen Ausschnitts folgendermaßen aussehen:

**A:** /betadate

**C:** The next beta will take place on Sunday the 14.4.2024.

**A:** /leaderboard total

**C:** „Generierung und Ausgabe der Bestenliste.“ (Siehe Abbildung 17)

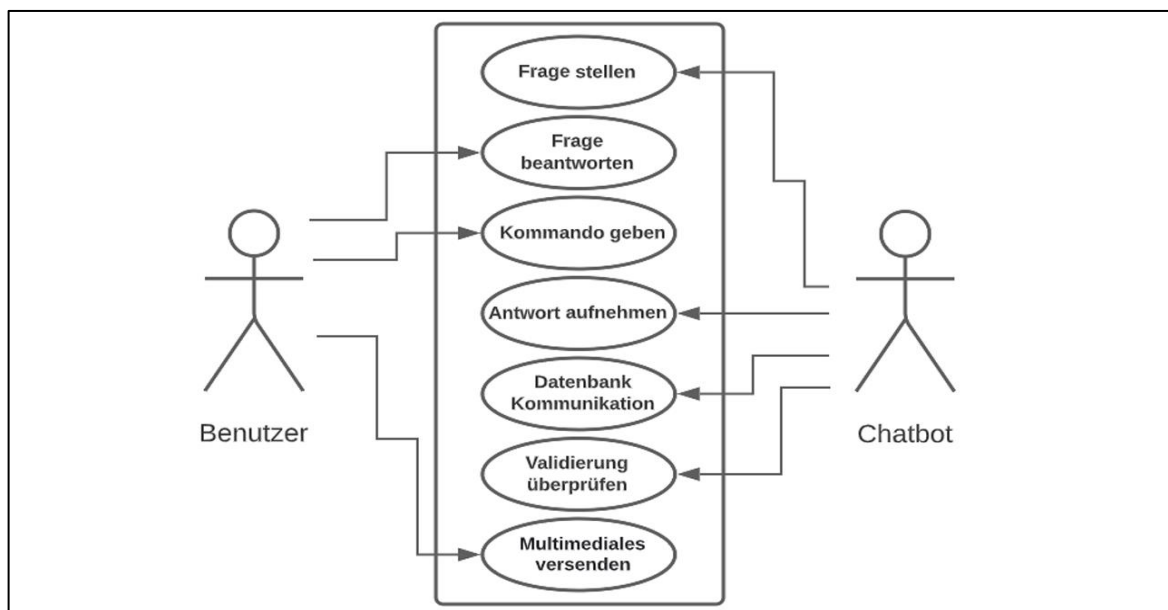
Weiterhin hat Anton das Kartenspiel des beta Bot entdeckt und möchte sich selbst als Karte dort einfügen, um am Spiel teilnehmen zu können. Dafür muss er dem Chatbot ein paar Parameter übergeben.

Die Interaktion zwischen dem Chatbot (C) und Anton (A) kann für dieses Beispiel anhand eines kleinen Ausschnitts folgendermaßen aussehen:

**A:** /user registration name: Bauer forename: Anton team: GD

**C:** You have successfully registered.

Anton probiert im Folgenden noch einige Zeit die Funktionen des beta Bot aus. Er macht unter anderem TP Eintragungen, damit sich alle Entwickler wichtige Informationen zu sich selbst holen können, der Chatbot eine neue wöchentliche Bestenliste berechnen kann und im Kartenspiel die Werte der einzelnen Entwickler aktualisiert werden und so weiter.



**Abbildung 10: Anwendungsdigramm, Quelle: Eigene Darstellung**

Abbildung 10 enthält eine visuelle Darstellung der grundlegenden Funktionalitäten mit dem Endnutzer als Akteur innerhalb eines Anwendungsdigramms.

### 3.1.2 Anforderungen an den Prototyp

Da nun einige Funktionalitäten anhand eines Anwendungsszenarios veranschaulicht sind, werden innerhalb dieses Abschnitts die Anforderungen an den zu entwickelnden Prototyp ausführlich beschrieben und festgelegt.

Die Formulierung der Anforderungen orientiert sich hierbei an der „Schablone für das Lastenheft, Pflichtenheft und Glossar“ nach Balzert (vgl. Balzert et al. 2009: 492 ff). Grund dafür ist, dass diese Anforderungsschablone für kleine Softwareprojekte und Prototypen empfohlen wird. Jede Anforderung beginnt mit einem Buchstaben, gefolgt von einer Zahl, die in Schrägstriche eingeschlossen ist. Im Folgenden werden die in dieser Arbeit verwendeten Abkürzungen der Anforderungstypen nach Balzert erklärt (vgl. Balzert et al. 2009: 457 ff):

**/F10/** Funktionale Anforderungen: Welche Fähigkeiten erwartet der Benutzende von der Anwendung bezüglich der Zielrichtung?

**/NF10/** Nicht-funktionale Anforderungen: Welche Qualitätsanforderungen gibt es an die Anwendung?

Dies ist laut Balzert jedoch nicht alles, was in eine umfangreiche Analyse hineingehört. Da dies aber den Rahmen dieser Bachelorarbeit sprengen würde, werden nur die funktionalen und nicht-funktionalen Anforderungen behandelt. Es folgt ein kleiner Überblick über die weiteren Anforderungstypen nach Balzert:

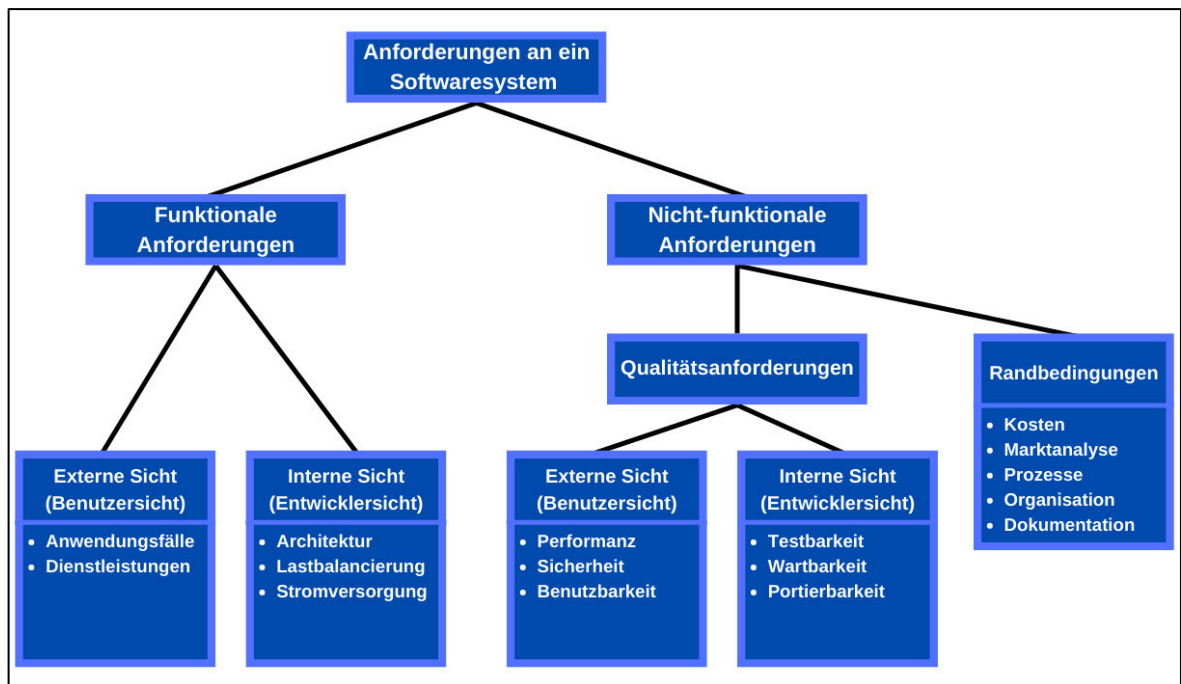
**/V10/** Vision des Projekts: Was soll am Ende erreicht werden?

**/Z10/** Ziel des Projekts: Auf welche Weise kann man die Visionen verfeinern sowie erleichtern?

**/R10/** Rahmenbedingungen: Was gibt es für organisatorische und technische Einschränkungen für die Anwendung?

Nachfolgend werden die genannten Anforderungen in Bezug auf die Bedürfnisse des Prototyps spezifiziert.





**Abbildung 11: Einteilung der Anforderungen an ein Software-System, Quelle: Eigene Darstellung nach Peterjohann (2021)**

In Abbildung 11 ist die inhaltliche Unterteilung von funktionalen und nicht-funktionalen Anforderungen noch einmal dargestellt und mit Beispielen versehen.

### 3.1.3 Funktionale Anforderungen

Durch das Beschreiben der funktionalen Anforderungen werden die angestrebten Funktionalitäten sowie das Verhalten des Systems definiert. Im Kontext des Prototyps lauten sie wie folgt:

**/F10/** Der Chatbot soll anhand von Discord Slash Commands angesprochen und gesteuert werden.

**/F11/** Sind notwendige Felder der Slash Commands nicht ausgefüllt, muss der Chatbot die Benutzenden darüber informieren.

**/F12/** Der Chatbot muss die Kommunikation mit Benutzenden in natürlicher Sprache führen.

**/F13/** Der Chatbot soll auf Discord interne Interaktionsmöglichkeiten wie zum Beispiel Buttons, Modals oder Dropdowns zugreifen können und so den benutzenden interaktive Möglichkeiten zur Datenübermittlung bieten.

**/F14/** Der Chatbot muss mit einer Datenbank kommunizieren, um so Daten abzuspeichern sowie abzurufen.

**/F15/** Der Chatbot muss anhand von gegebenen Daten Berechnungen anstellen und diese ausgeben können.

**/F16/** Der Chatbot muss in der Lage sein, auf die API unterschiedlicher Plattformen wie zum Beispiel Trello oder Git zuzugreifen.

**/F17/** Der Chatbot muss den Benutzenden nach dem Versenden von Daten darüber informieren, ob es erfolgreich war.

**/F18/** Der Chatbot soll Bilder, Emotes sowie Text versenden und diese auch in Form von Discord „Embeds“ darstellen können.

**/F19/** Der Chatbot soll in der Lage sein, bereits gesendete Nachrichten zu einem späteren Zeitpunkt zu bearbeiten.

**/F20/** Mindestens zwischen zwei Benutzenden soll es möglich sein, über den Chatbot miteinander zu interagieren.

**/F21/** Der Chatbot muss die Möglichkeit bieten, bereits vom Benutzenden gesendete persönliche Informationen im Nachhinein noch einmal zu bearbeiten.

### **3.1.4 Nicht-funktionale Anforderungen**

Die nicht-funktionalen Anforderungen definieren, in welcher Qualität die beschriebenen Funktionen umgesetzt werden sollen. Damit besitzen sie unter anderem eine große Auswirkung auf den Verbrauch von Ressourcen. Im Kontext des Prototyps lauten sie wie folgt:

**/NF10/** Der Chatbot soll innerhalb von maximal drei Sekunden auf die Eingaben der Benutzenden reagieren.

**/NF11/** Der Chatbot soll von der aktuellen Version der Anwendung Discord unterstützt werden.

**/NF12/** Die Verarbeitung von sensiblen Daten, zum Beispiel Textnachrichten zwischen Benutzenden und Chatbot, werden nicht an dritte übergeben und Außenstehende erhalten außerdem keinen Zugriff darauf.

**/NF13/** Der Chatbot soll jederzeit online und damit auch für jeden auf dem Discord Server verfügbar sein.

**/NF14/** Die Benutzenden sollen nach dem Interagieren oder Schreiben mit dem Chatbot erfahren, ob alles korrekt abgesendet wurde oder ein Problem aufgetreten ist.

**/NF15/** Die Benutzenden sollen im Falle auftretender Fehlerquellen egal welchen Bereichs informiert werden.

**/NF16/** Der Chatbot soll durch einen Hyperlink auf jeden beliebigen Discord Server eingeladen werden können.

## 3.2 Entwicklung eines Konzeptes

Nachdem die Anforderungen an den Prototyp definiert sind, soll im Folgenden dessen Konzept erstellt werden. Besonders Wert wird dabei auf den Aufbau des Graphical User Interface (GUI) und dessen dazugehörige Funktionalitäten gelegt, um dieses letztendlich so gut wie möglich in der Implementierung zu realisieren.

### 3.2.1 Grafische Benutzeroberfläche

Das Ziel des GUI-Entwurfs ist es, die erarbeiteten Anforderungen in eine bildlich sichtbare Benutzeroberfläche umzusetzen. Dadurch, dass der Chatbot auf der Kommunikationsplattform Discord funktionieren soll, ist der Spielraum für die Benutzeroberfläche sehr begrenzt. Das bedeutet, dass es nur teilweise möglich ist, bei der Bestimmung des GUI mitzuwirken. Es soll dennoch eine aufgabenangemessene Benutzungsschnittstelle entworfen werden, welche die ausgearbeiteten Anforderungen am besten grafisch unterstützt.

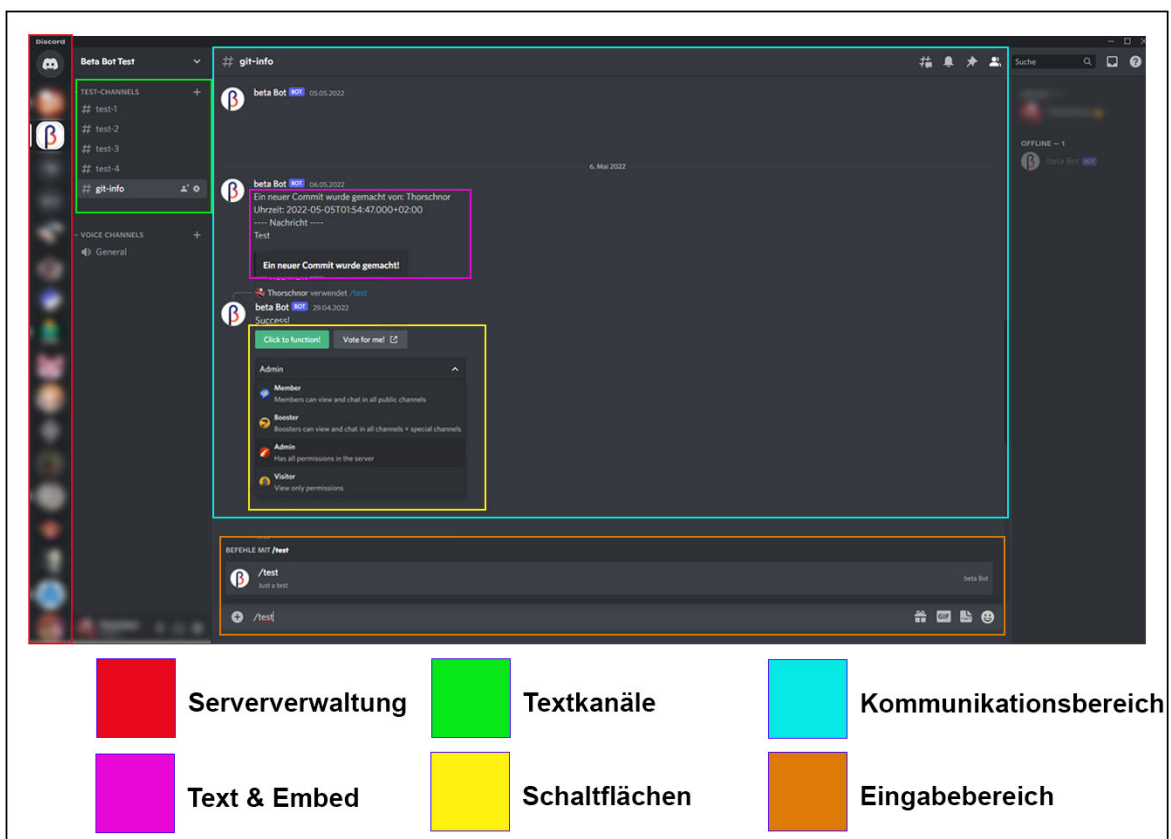


Abbildung 12: Konzept der Benutzeroberfläche, Quelle: Eigene Darstellung unter Zuhilfenahme der Kommunikationsplattform Discord

Wie die Benutzeroberfläche aussehen soll, ist in Abbildung 12 dargestellt. Die Serververwaltung enthält alle Discord Server, welchem der Benutzer beigetreten ist. Die Textkanäle sind meistens nach verschiedenen Kategorien benannt, sodass zum Beispiel eine neue Git Commit Benachrichtigung vom Chatbot auch nur in den dafür vorgesehenen „git-info“ Kanal gesendet wird. Der Kommunikationsbereich enthält alle vom Benutzer oder Chatbot gesendeten Nachrichten in zeitlich richtiger Reihenfolge. Weiterhin sendet der Chatbot Texte sowie Discord „Embeds“ in den Kommunikationsbereich. Für eine interaktive Benutzung sorgen Schaltflächen in Form von Buttons, Dropdowns sowie Modals. Diese sind auch im Kommunikationsbereich enthalten. Zuletzt gibt es noch den Eingabebereich, in der der Benutzer die Befehle und Nachrichten an den Chatbot sendet. Dabei werden, sobald der Benutzer

einen Slash eingibt, verschiedene Vorschläge von Befehlen mit entsprechender Beschreibung angezeigt.

### 3.2.2 Funktionen und Kommandos

In diesem Abschnitt werden alle Funktionen, welche der Chatbot beinhalten soll, sowie Kommandos, auf welche er reagieren soll, aufgeführt. Im Bereich Funktionen soll der Chatbot automatisch eine Nachricht in Discord ausgeben, wenn auf GitLab ein Commit erfolgt ist, ein neuer Branch erstellt wurde und ein Merge Request stattgefunden hat. Beim Aufgabenverwaltungsdienst Trello soll eine Nachricht ausgegeben werden, wenn eine neue Karte erstellt wurde, eine Karte verschoben wurde, eine Karte aktualisiert wurde und ein neuer Kommentar zu einer Karte hinzugefügt wurde. Im nachfolgenden soll nun eine Liste mit allen vorhandenen Kommandos dargestellt werden. Dabei wird zuerst der Discord Slash Command genannt, wie ihn der Benutzer eingeben soll sowie eine kurze Beschreibung über dessen Funktion:

**/album add:** Fügt eine Karte zu einem Album hinzu.

**/album background:** Lässt den Hintergrund eines Albums ändern.

**/album collection:** Gibt alle im Besitz des Benutzers befindliche Alben untereinander aufgelistet aus.

**/album create:** Erstellt ein neues Album.

**/album delete:** Löscht ein gesamtes Album.

**/album remove:** Löscht eine Karte aus einem Album heraus.

**/album view:** Generiert und gibt ein Album als Bilddatei aus.

**/eventdate:** Gibt das Datum aus, an welchem das nächste Event stattfindet.

**/burn:** Zerstört und löscht eine Karte.

**/card collection:** Gibt alle im Besitz des Benutzers befindliche Karten untereinander aufgelistet aus.

**/coin amount:** Zeigt die aktuell im Besitz befindliche Goldanzahl an.

**/developer information:** Gibt detaillierte Informationen zu einem Projektmitglied aus.

**/developer update:** Aktualisiert die Aufgaben sowie Kommentare zu Projektmitgliedern.

**/drop:** Es werden drei zufällige Karten ausgegeben, welche von Benutzenden genommen werden können.

**/give:** Übergibt einer Karte oder Gold an einen ausgewählten Benutzer.

**/info:** Gibt detaillierte Informationen zum beta Card Game aus.

**/leaderboard total:** Gibt die aktuelle Gesamt TP Bestenliste der Projektmitglieder aus.

**/leaderboard weekly:** Gibt die wöchentliche TP Bestenliste der Projektmitglieder aus.

**/shop:** Generiert einen Shop, in dem Gold gegen bestimmte Karten getauscht werden kann.

**/update date:** Lässt das Datum für das nächste Event aktualisieren.

**/upload tp:** Lädt eine CSV Datei in die Datenbank, um die TP der einzelnen Projektmitglieder zu aktualisieren.

**/user edit:** Lässt den Benutzer seine Registrierungsinformationen editieren und damit gewünschte Daten ändern.

**/user registration:** Registriert einen Benutzer für das Kartenspiel.

## 4 Implementierung

Dieser Abschnitt beschäftigt sich mit der Umsetzung der in Kapitel 3 spezifizierten funktionalen und nicht-funktionalen Anforderungen an den Prototypen. Begonnen wird dabei mit der endgültigen Entwicklungsumgebung, gefolgt von der Vorstellung und näheren Beschreibung der genutzten Programmiersprachen und Datenbankverwaltungssystemen. Anschließend wird die Implementierung von Schlüsselfunktionen im Detail beschrieben. Weiterhin sollen die Benutzereingaben genau definiert sowie auf die Benutzerfreundlichkeit eingegangen werden. Es folgt außerdem eine Testphase mit Konzepterstellung, Durchführung und Auswertung. Die in Abschnitt 2.1 untersuchten Begrifflichkeiten Mensch-Computer-Interaktion, Interaction Design, User Centred Design, User Interface und die User Experience sollen ebenfalls anhand des Prototyps innerhalb der Testphase beachtet und dessen Erkenntnisse abschließend anhand von einem Fazit des gesamten Kapitels nachgewiesen werden.

### 4.1 Prototyp als Discord Bot

Der entwickelte Prototyp soll nach erfolgreichem Abschluss der Entwicklungsphase als Discord Bot verfügbar sein. Das bedeutet, es wird eine KI erschaffen, die eine Reihe von nützlichen automatisierten Aufgaben und Bot-Befehlen auf verschiedenen Discord Servern ausführen kann. Der Chatbot besteht hauptsächlich aus Python Code und arbeitet mit einer Datenbank im Hintergrund. Die Datenbank dient zum Speichern und Abrufen von verschiedenen Datensätzen. Im Hinblick auf Abschnitt 2.3.4 lässt sich die Art des Prototypen folgendermaßen bestimmen. Es handelt sich dabei um einen Chatbot mit geschlossenem Wissensgebiet, da dieser auf den Bereich Projektmanagement spezialisiert ist. Weiterhin operiert er interpersonell, da er kein Begleiter des Benutzers ist, sondern Informationen aufnimmt und diese weiterleitet. Außerdem ist der Prototyp aufgabenbasiert, da er spezifische Aufgaben ausführt. Im Bereich der Methode zur Eingabeverarbeitung und Antwortgenerierung gehört er zum regelbasierten Modell. Das liegt daran, dass der Chatbot seine Antworten auf Grundlage fest definierter Regeln auswählt. Des Weiteren soll der Prototyp in Anlehnung an die in Abschnitt 2.3.5 vorgestellte allgemeine Chatbot-Architektur realisiert werden.

Für den Versuch einer Moral- und Arbeitssteigerung wurde das beta Card Game mithilfe des Prototyps umgesetzt. Da es sich dabei um ein Spiel handelt, wurde für die Erschaffung, auf den dreistufigen iterativen Entwicklungsprozess aus Abschnitt 2.2.7 zurückgegriffen, da dieser ein optimales Konzept für die Erstellung bietet.

#### 4.1.1 Programmierung mit Python und MySQL

Der Prototyp wurde innerhalb der Entwicklungsumgebung Pycharm erstellt und basiert auf der universellen, interpretierten und höheren Programmiersprache Python sowie dem weltweit verbreiteten relationalen Datenbankverwaltungssystem MySQL. Demnach hat man sich im Bezug auf Abschnitt 2.3.6 für eine Entwicklung mit einer Programmiersprache und gegen die Verwendung einer modernen Plattform entschieden. Dabei wurde sich aus mehreren Gründen für Python entschieden. Python ist eine Moderne und auch stets weiterentwickelte Programmiersprache, sie bietet für sehr viele Anwendungsprobleme eine Lösung, sie ist leichter zu erlernen als andere Sprachen, was auch an der gut lesbaren Struktur des Codes liegt und wodurch außerdem schnell Bugs gefunden sowie behoben werden können.

Außerdem ist Python dynamisch typisiert, was den Typ von Variablen oder Funktionsargumenten zu definieren unnötig macht (vgl. Matrone 2020). Neben der sehr umfassenden Standardbibliothek, welche Python mitbringt, werden weitere für den Prototypen unerlässliche Bibliotheken hinzugefügt. Darunter zählen als wichtigste Bibliotheken Pycord, was als Schnittstelle und Zugriffspunkt auf die API von Discord verwendet wurde, MySQL Connector Python, welches benutzt wurde, um eine Verbindung zur Datenbank herzustellen und daraus Daten abzurufen beziehungsweise hochzuladen, Pillow, welches zur Bearbeitung und Erstellung von Bildern benutzt wurde, und NumPy, welches die Funktionalitäten von Arrays erweitert. Die Programmierung erfolgte unter Verwendung der Version Python 3.10. Weiterhin wurde mithilfe von MySQL Statements Daten in die Datenbank eingefügt, abgerufen, aktualisiert und gelöscht. Es wurde sich dabei aufgrund der vielen Vorteile für MySQL entschieden. Die Wichtigsten sind, dass es als sehr sicher und stabil gilt, dass es sehr flexibel ist, Schnittstellen zu vielen Skriptsprachen bietet, es einfach zu bedienen ist und auf vielen verschiedenen Plattformen verwendet werden kann (vgl. o. V. RYTE Wiki 2021).

## 4.1.2 Datenbankerstellung

Wie bereits in Abschnitt 4.1.1 erwähnt, wurde für die Erstellung der Datenbank das Datenbankverwaltungssystem MySQL verwendet. Der genutzte Datenbank-Server-Typ ist MariaDB mit der Server-Version 10.4.24-MariaDB. Zur Administration der Datenbank wurde die freie Webanwendung phpMyAdmin genutzt in der Version 5.2.0. Für die Realisierung der Datenbank wurde das Programmpaket XAMPP genutzt. Dieses ermöglichte das Installieren sowie Konfigurieren des Webservers Apache unter Zuhilfenahme der Datenbank MariaDB (vgl. XAMPP: Wikipedia 2022). Mithilfe der grafischen Benutzeroberfläche von „phpMyAdmin“ war es möglich, die Datenbank aufzubauen. Dafür wird zunächst die Datenbank beta bot erstellt. In dieser sind die folgenden Tabellen für den Prototyp konfiguriert:

- **album:** Diese Tabelle beinhaltet alle Informationen über erstellte Alben innerhalb des beta Card Game. Darunter zählt die Discord ID des Besitzers von dem Album, der Name des Albums, die IDs der in dem Album befindlichen Karten, der freie Platz im Album sowie der Name des ausgewählten Hintergrundes.
- **beta\_date:** Diese Tabelle beinhaltet Informationen zum Datum des nächsten Events sowie in welcher Woche sich das Projekt aktuell befindet.
- **beta\_member:** In dieser Tabelle sind Informationen für alle registrierten Benutzer hinterlegt. Dazu zählen persönliche Informationen wie Name, Discord ID aber auch die Anzahl an Gold, verfügbare Alben und eine Information, ob ein Bild hinterlegt wurde oder nicht.
- **generated\_user\_cards:** Diese Tabelle enthält alle im Verlaufe des beta Card Game generierten Karten. Dazu gehört Informationen zu der Karten ID, Name, Seltenheit und Datum der Ziehung.
- **member\_comments:** In dieser Tabelle sind Aufgaben und Kommentare zu einzelnen Entwicklern der beta in Textform hinterlegt.
- **tp\_list:** Diese Tabelle enthält Informationen zu den erreichten TP pro Woche jedes einzelnen Projektmitglieds. Außerdem sind der TP durchschnitt sowie die Gesamtanzahl an TP enthalten.

Nachfolgend soll nun ein kurzes Beispiel zum Aufbau einer solchen Tabelle gezeigt werden.

ID	Name	Vorname	Matriculation	MtkNr	Discord_ID	Team	Betacoins	Albums_Left	Custom_Image
1	Erdhütter	Leon	MI18w2-B	49860	262699738405535745	GD	34143	9	0

**Abbildung 13: Datenbanktabelle mit Eintrag, Quelle: Eigene Darstellung unter Zuhilfenahme von phpMyAdmin**

Abbildung 13 zeigt die Datenbanktabelle beta\_member mit einem darin enthaltenen Eintrag innerhalb der grafischen Benutzeroberfläche von „phpMyAdmin“. Es ist zu erkennen, dass jeder Wert einer Reihe einer Tabellenspalte zugeordnet ist. Somit ist es möglich, gezielt den Eintrag abzurufen, welcher gerade vom Chatbot benötigt wird. Außerdem wurde eine ID als Primärschlüssel erstellt, um sicherzugehen, dass jeder Eintrag einzigartig ist und auch gelöscht werden kann.

### 4.1.3 Verwirklichung des Chatbots

Der erste Schritt, um den Discord Bot zu verwirklichen, ist, dass er zunächst im Discord Developer Portal als solcher neu erstellt werden muss. Dazu loggt man sich auf der Developer Webseite mit seinem bestehendem Discord Account ein und kann dann über den Reiter Applikationen eine neue Applikation erstellen. Sobald dies getan ist, kann man den Discord Bot auch schon konfigurieren. Dabei wird ein beliebiges Logo als Profilbild eingesetzt, der Name in die gewünschte Bezeichnung geändert sowie eine kurze Beschreibung des Chatbots erstellt. Weiterhin werden die benötigten Rechte des Bots für die Discord Server definiert und anschließend in einen Einladungslink konvertiert. Wird der Bot zu einem Beitritt eingeladen, werden ihm direkt alle benötigten Privilegien gewährt. Abschließend werden eine Applikation ID, ein öffentlicher Schlüssel sowie ein Token zur Authentifizierung des Discord Bots innerhalb der Programmierschnittstelle generiert.

Im nächsten Schritt wird die Entwicklungsumgebung PyCharm installiert und eingerichtet. Außerdem werden noch benötigte Bibliotheken eingefügt. Parallel dazu wird die in Abschnitt 4.1.2 erstellte Datenbank aufgebaut und konfiguriert. Eine „main“ Datei konnte in PyCharm erstellt werden und in ihr erfolgten erforderliche Discord Rechte vergabe, Fehler Events sowie Token Verarbeitungen. Dazu kommt die Verwendung von Discord „cogs“, welche es ermöglichen, den gesamten Python Code über mehrere Dateien zu verteilen, um so eine bessere Übersicht zu gewähren. Im nächsten Schritt werden die Slash Commands über mehrere „cogs“ erschaffen und für diese die nötige Logik sowie eine Datenbank Kommunikation erstellt. Ausführen kann man den Discord Bot über PyCharm und die lokale Datenbank über das XAMPP Control Panel.



#### 4.1.4 Funktionen und Abläufe

Nachfolgend sollen einige der wichtigsten Schlüsselfunktionen beschrieben werden, auf denen der Prototyp aufgebaut ist.

```
class CardCommands(commands.Cog):
    def __init__(self, bot):
        self.bot = bot

    card = SlashCommandGroup("card", "Card Group")

    @card.command(guild_ids=[969395923929137162], name="collection", description="View your card collection or "
                                                                    "that of another player in a sort "
                                                                    "order of your choice.")

    async def cardinv(self, ctx,
                     sorting: Option[str], "Your desired sorting.", choices=["Alphabet", "Date", "Rarity"],
                     required=True,
                     discorduser: Option(discord.User, "The player whose cards collection you want to see.",
                                         required=False)
    ):

```

Abbildung 14: Python und Slash Command Attribute, Quelle: Quellcode

class	<ul style="list-style-type: none"> <li>eine Klasse in Python ist wie ein Objekt Konstruktor oder eine "Blaupause" für die Erstellung von Objekten</li> </ul>
CardCommands(commands.Cog)	<ul style="list-style-type: none"> <li>der Name der Klasse mit Registrierung dieser als Discord cog</li> </ul>
def __init__(self, bot)	<ul style="list-style-type: none"> <li>mit dem Schlüsselwort def wird in Python der Beginn einer Funktion definiert</li> <li>eine Funktion ist eine Gruppe von zusammenhängenden Anweisungen, die eine bestimmte Aufgabe erfüllen</li> <li>alle Klassen haben eine Funktion namens __init__(), welche immer ausgeführt wird, wenn die Klasse initiiert wird</li> <li>der self-Parameter ist ein Verweis auf die aktuelle Instanz der Klasse und wird für den Zugriff auf Variablen verwendet, die zur Klasse gehören</li> </ul>
self.bot = bot	<ul style="list-style-type: none"> <li>unter Verwendung der __init__() Funktion können Objekteigenschaften oder anderen Operationen, die beim Erstellen des Objekts ausgeführt werden, Werte zugewiesen bekommen</li> </ul>
card = SlashCommandGroup("card", "Card Group")	<ul style="list-style-type: none"> <li>Slash Command Groups müssen immer in einem Attribut gespeichert werden</li> <li>eine Slash Command Group verknüpft verschiedene Slash Commands miteinander, was zur Übersichtlichkeit im Discord Texteingabefeld beiträgt</li> </ul>
@card.command	<ul style="list-style-type: none"> <li>Kennzeichnung der nachfolgenden Funktion als Slash Command</li> </ul>

	<ul style="list-style-type: none"> <li>wird das Attribut der Slash Command Group verwendet, wird das Nachfolgende Kommando als dieses registriert</li> </ul>
guild_ids	<ul style="list-style-type: none"> <li>Eintragung der ID des Discord Servers, auf dem die Kommandos verfügbar sein sollen</li> </ul>
name	<ul style="list-style-type: none"> <li>der Name des Kommandos, wie ihn der Benutzer später in Discord vorfindet</li> </ul>
description	<ul style="list-style-type: none"> <li>eine Beschreibung des Kommandos, wie ihn der Benutzer später in Discord vorfindet</li> </ul>
async def cardinv(self, ctx)	<ul style="list-style-type: none"> <li>die Syntax <code>async def</code> leitet entweder eine native Coroutine oder einen asynchronen Generator ein</li> <li>nach dieser Syntax folgt der Name, welcher einmalig sein muss</li> <li>ctx ist der Kontext und enthält Informationen über den Benutzer, welcher das Kommando ausführt</li> </ul>
sorting: Option	<ul style="list-style-type: none"> <li>Slash Commands lassen sich verschiedene Eingabeoptionen anfügen</li> <li>speichert man jede vom Benutzer eingegebenen Daten in Attribute und markiert diese mit dem Schlüsselwort „Option“</li> </ul>
str	<ul style="list-style-type: none"> <li>an dieser Stelle wird der Datentyp für die jeweilige Option vereinbart, welcher vom Benutzer erwartet wird</li> <li>dieser kann neben str für Zeichen auch int für Zahlen oder sogar ein Discord eigener Datentyp sein wie zum Beispiel <code>discord.user</code></li> </ul>
choices	<ul style="list-style-type: none"> <li>über das Schlüsselwort „choices“ lassen sich vorgefertigte Auswahlmöglichkeiten erstellen, aus welchem der Benutzer bei der Eingabe wählen kann</li> </ul>
required	<ul style="list-style-type: none"> <li>das Schlüsselwort „required“ gibt Auskunft darüber, ob die Tätigkeit dieser Eingabeoption zwingend erforderlich ist oder nicht</li> </ul>
):	<ul style="list-style-type: none"> <li>mit dem Schließen der Klammer und einem Doppelpunkt wird das Ende der Slash Command Konfiguration definiert</li> <li>alles was danach folgt, ist der eigentliche Code, welcher dem Chatbot sagt, was er tun soll</li> </ul>

**Tabelle 3: Beschreibung der Python und Slash Command Attribute, Quelle: Eigene Darstellung**

Da die Discord Slash Commands sehr oft einem gleichen Aufbau folgen, verschaffen Abbildung 14 und Tabelle 3 zunächst einen Überblick über deren Aufbau und Funktion. Die in Abbildung 14 hervorgehobenen Attribute werden in der nachfolgenden Tabelle genau erklärt und sind wichtig, um die weiteren Funktionen zu verstehen. Nachdem die verschiedenen Attribute erläutert sind, erfolgt nun ein Einblick in einige der Funktionalitäten des Prototyps.

```

databaseConnect = mysql.connector.connect(
    host=hostname,
    user=username,
    password=passwordname,
    database=databasename
)
cursor = databaseConnect.cursor(buffered=True)

cursor.execute("SELECT Discord_ID from beta_member WHERE Discord_ID = %s", (str(ctx.author.id),))
registeredusers = cursor.rowcount

if registeredusers != 0:
    await ctx.respond("You are already registered. To make changes to it please use /useredit.")
    databaseConnect.close()
    return

customimage = 0
if imagefile is not None:
    if imagefile.url.endswith('.jpg'):
        await imagefile.save("images/developer_images/" + str(ctx.author.id) + ".jpg")
        customimage = 1
    else:
        await ctx.respond("You have not uploaded a .jpg file.")
        databaseConnect.close()
        return

cursor.execute("INSERT INTO beta_member(Name, Vorname, Matriculation, MtkNr, Discord_ID, Team, Custom_Image)
    VALUES(%s, %s, %s, %s, %s, %s, %s)", (name, vorname, matriculation, matriculationnumber,
        str(ctx.author.id), team, customimage), )

databaseConnect.commit()
databaseConnect.close()

await ctx.respond("You have successfully registered.")

```

**Abbildung 15: Slash Command für die Registrierung, Quelle: Quellcode**

Die erste Funktion kann Abbildung 15 entnommen werden und zeigt die Umsetzung des Registrierungskommandos mit Python Code innerhalb der Entwicklungsumgebung Py-Charm. Zuerst wird für dieses Kommando eine Verbindung zur Datenbank erstellt und ein Cursor erzeugt. Mithilfe dieses Cursors wird abgefragt, wie oft die Discord ID des Benutzers bereits in der Datenbank enthalten ist und der Wert in einer Variablen gespeichert. Hat die Variable einen Wert größer 0, benachrichtigt der Chatbot den Benutzer, dass er bereits registriert ist und das Kommando endet. Ist der Wert 0, wird im nächsten Schritt überprüft, ob der Benutzer ein optionales Profilbild angehängt hat. Ist dies der Fall, überprüft der Chatbot, ob es sich dabei um eine JPG Datei handelt. Ist dies korrekt, speichert er das Bild lokal ab und vergibt ihm als Name die Discord ID des Benutzers. Sollte es sich um einen anderen Dateityp handeln, wird der Benutzer darauf hingewiesen und das Kommando endet. Im nächsten Schritt erstellt der Chatbot eine neue Reihe in der Datenbank und lädt alle zuvor vom Benutzer getätigten Eingaben hoch. Abschließend werden diese Eingaben bestätigt und die Datenbankverbindung geschlossen. Außerdem gibt der Chatbot noch eine Nachricht aus, dass alles funktioniert hat. Am Beispiel von Abbildung 13 wird ein Datenbankeintrag dargestellt.

```

cursor.execute("SELECT Name, Vorname, Total FROM tp_list WHERE Total > 0 AND Name != %s ORDER BY Total DESC",
              ("Average", ))
foundusers = cursor.rowcount
allentries = cursor.fetchall()

if foundusers > 10:
    foundusers = 10

x = 0
resultstring = ""
resultstring2 = ""
while x < foundusers:
    if x == 0:
        resultstring = resultstring + ":first_place: "
    elif x == 1:
        resultstring = resultstring + ":second_place: "
    elif x == 2:
        resultstring = resultstring + ":third_place: "
    else:
        resultstring = resultstring + str(x + 1) + ". "

    resultstring = resultstring + allentries[x][1] + " " + allentries[x][0]

    resultstring2 = resultstring2 + "*" + str(allentries[x][2]) + " TP" + "*"

    resultstring = resultstring + "\n"
    resultstring2 = resultstring2 + "\n"
    x += 1

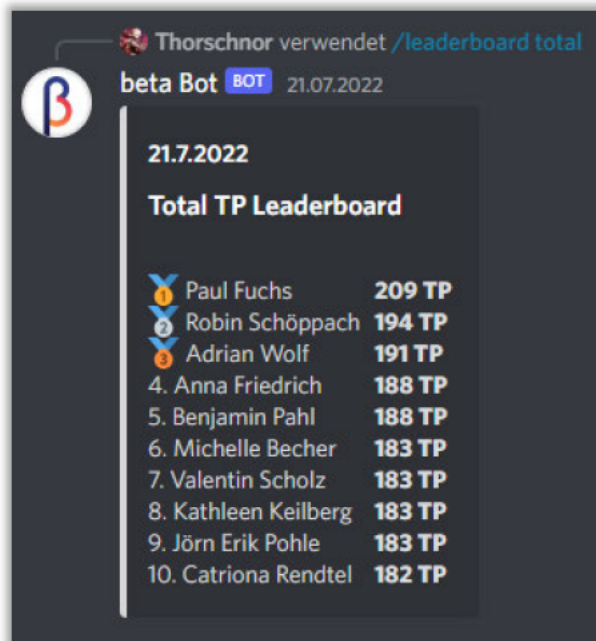
today = datetime.today().date()
date_format = "%Y-%m-%d"
dayformatted = datetime.strptime(str(today), date_format)

embedw1 = discord.Embed(title="Total TP Leaderboard", color=0x060606)
embedw1.set_author(name=str(dayformatted.day) + "." + str(dayformatted.month) + "." + str(dayformatted.year))
embedw1.add_field(name="\u200b", value=resultstring, inline=True)
embedw1.add_field(name="\u200b", value=resultstring2, inline=True)
await ctx.respond(embed=embedw1)
databaseConnect.close()

```

**Abbildung 16: Slash Command für die Gesamtbestenliste, Quelle: Quellcode**

Abbildung 16 zeigt den Codeabschnitt für das Ausgeben der Gesamtbestenliste. Dazu ist es zunächst erforderlich, die benötigten Werte aus der Datenbank abzurufen. Diese werden auch direkt absteigend geordnet in ein Array gespeichert. Werden mehr als zehn Benutzer gefunden, wird die Anzahl der gefundenen auf zehn gesetzt, um so immer nur die Top 10 von allen Entwicklern zu bekommen. Als Nächstes werden die Strings generiert. Dabei sind dem ersten, zweiten und dritten Platz jeweils ein Emote vorangestellt, gefolgt von der Zahl des Schleifendurchlaufes und dem gespeicherten Array Wert des Namens des Entwicklers, welcher zuvor aus der Datenbank abgerufen wird. Weiterhin wird das heutige Datum ermittelt und formatiert. Abschließend wird ein Discord „Embed“ erstellt und das Datum sowie die generierten Strings eingefügt.



**Abbildung 17: Ausgabe der Gesamtbestenliste in Discord, Quelle: Eigene Darstellung**

Abbildung 17 zeigt, wie das Endergebnis fertig generiert in Discord aussieht. Dabei wird oben das Datum genannt sowie eine kurze Überschrift mit einer Beschreibung, was dargestellt ist. Weiterhin wird immer zuerst die Platzierung genannt, gefolgt von dem Namen des Entwicklers und dessen gesamt TP.

```

beta_background.paste(cardimage[0], (45, 102), cardimage[0].convert("RGBA"))
beta_background.paste(secretimage, (325, 305), secretimage.convert("RGBA"))
beta_background.paste(secretimage, (604, 101), secretimage.convert("RGBA"))

beta_background.paste(beta_background2, (0, 0), beta_background2.convert("RGBA"))

frames[0].paste(textimage[0], (102, 79 - fontoffsetup[0]), textimage[0].convert("RGBA"))
frames[0].paste(textimage[1], (102, 252 - fontoffsetdown[0]), textimage[1].convert("RGBA"))

beta_background.paste(frames[0], (3, -14), frames[0].convert("RGBA"))
beta_background.paste(secretframe, (284, 190), secretframe.convert("RGBA"))
beta_background.paste(secretframe, (564, -14), secretframe.convert("RGBA"))

beta_background = beta_background.convert("RGB")

tempimage = BytesIO()
beta_background.save(tempimage, format="JPEG", optimize=True)
tempimage.seek(0)
file = discord.File(tempimage, "drop.jpg")

embeddrop = discord.Embed(title="Card Drop")
embeddrop.set_image(url="attachment://drop.jpg")
drop = await ctx.send(file=file, embed=embeddrop)

```

**Abbildung 18: Bildbearbeitung im Kartenausgabe Slash Command, Quelle: Quellcode**

Ein abschließendes Beispiel soll Abbildung 18 zeigen. Dabei sieht man einen Ausschnitt aus dem Kartenausgabe Slash Command. Hier soll noch einmal verdeutlicht werden, wie der Code bei einem Bildbearbeitungsvorgang aussieht. Zuerst wird ein Bild, welches lokal gespeichert ist, als Hintergrund in einer Variablen geöffnet. Weiterhin werden Schriften für die Bilder generiert und als eigenes Bild abgespeichert. In Abbildung 18 werden diese einzelnen Elemente aus Benutzerbild, Rahmen und Schrift übereinandergelegt und als „RGBA“ konvertiert, um Transparenz darstellen zu können. Später wird das zusammengestellte Bild wieder in „RGB“ konvertiert, um am Ende ein JPG Bild ausgeben zu können, welches platzsparend ist. Dieses Bild wird danach temporär in einer Variablen gespeichert, sodass es nicht nötig ist, das Bild lokal abzuspeichern.



**Abbildung 19: Karten Ausgabe, Quelle: Eigene Darstellung**

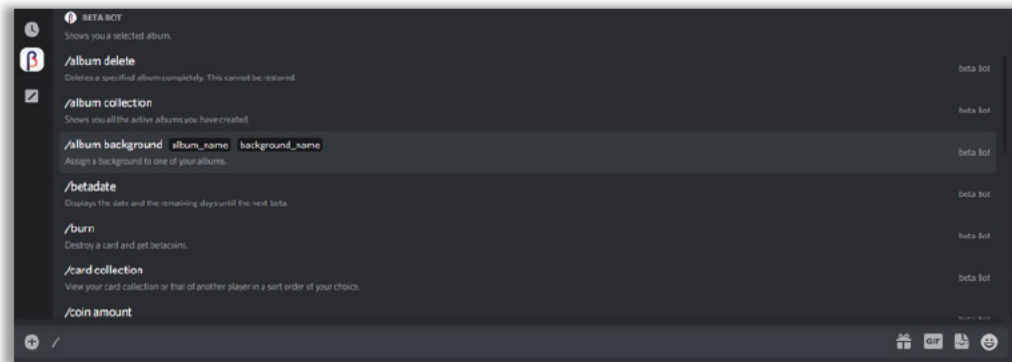
Das fertig generierte Bild ist in Abbildung 19 zu sehen. Dieses wurde nun in ein „Embed“ als Anhang miteingefügt und im Discord gesendet. Durch drei verschiedene Emotes der Zahlen 1 - 3 können die Benutzer sich für eine der Karten entscheiden.

## 4.2 Benutzereingaben

In diesem Abschnitt soll geklärt werden, wie der Benutzer die Kommandos an den Chatbot übermittelt. Dabei soll genau darauf eingegangen werden, was dazu erforderlich ist bzw. welche Angaben der Benutzer bei jedem Kommando tätigen muss. Weiterhin erfolgt eine Erklärung, wie diese Daten, welche der Benutzer dem Chatbot gesendet hat, gespeichert und abgerufen werden.

### 4.2.1 Übermittlung von Kommandos

In diesem Abschnitt soll zum einen geklärt werden, wie der Benutzer die einzelnen Kommandos an den Chatbot übermittelt sowie welche Parameter er dafür benutzen muss. Die Übermittlung von Kommandos erfolgt über Discord Slash Commands.



**Abbildung 20: Discord Slash Commands, Quelle: Eigene Darstellung unter Zuhilfenahme der Kommunikationsplattform Discord**

Abbildung 20 zeigt eine Discord Texteingabezeile. Wird in diese ein Slash eingefügt, bekommt man zum einen Vorschläge für die in Discord integrierten Slash Commands sowie auch die eigenen von Chatbots, welche sich auf dem Server befinden und diese unterstützen. Bewegt man die Maus über einen solchen Slash Command, wird einem auch schon angezeigt, welche Eingaben (Parameter) der Benutzer tätigen muss, um diesen auszuführen. Mit einem Mausklick wird der ausgewählte Slash Command in die Texteingabezeile übernommen und man kann die einzelnen Parameter nach Belieben eintragen. Hat man zum Beispiel Text in ein Feld eingetragen, was eigentlich für eine Zahl vorbestimmt ist, wird der Benutzer vom Chatbot darauf hingewiesen und fordert ihn auf, dies zu ändern. Abschließend muss der Nutzer nur noch das Kommando mit der Enter Taste absenden und der Chatbot beginnt mit der Verarbeitung. Im Folgenden werden alle vorhandenen Slash Commands aufgeführt und mit den erforderlichen Parametern beschrieben.

**/album add:** Für dieses Kommando muss der Benutzer den Namen des Albums angeben sowie die ID der Karte, welche in das Album eingefügt werden soll.

**/album background:** Für dieses Kommando muss der Benutzer den Namen des Albums angeben sowie aus einer Auswahl von vier Optionen den gewünschten Hintergrund für das Album wählen.

**/album collection:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/album create:** Für dieses Kommando muss der Benutzer den Namen des Albums angeben, also wie es benannt werden soll.

**/album delete:** Für dieses Kommando muss der Benutzer den Namen des zu löschenden Albums angeben.

**/album remove:** Für dieses Kommando muss der Benutzer den Namen des Albums angeben sowie die ID der Karte, welche aus dem Album entfernt werden soll.

**/album view:** Für dieses Kommando muss der Benutzer den Namen des Albums angeben, welches er anschauen möchte.

**/eventdate:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/burn:** Für dieses Kommando muss der Benutzer die ID von der Karte angeben, welche er zerstören möchte.

**/card collection:** Für dieses Kommando muss der Benutzer auswählen, ob er die Karten nach Alphabet, Datum oder Seltenheit sortiert ausgeben möchte und optional kann er einen Discord Benutzer angeben, von welchem er die gesammelten Karten anschauen kann.

**/coin amount:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/developer information:** Für dieses Kommando muss der Benutzer den Vornamen und Namen des beta Entwicklers angeben, von welchem er die Informationen haben möchte.

**/developer update:** Für dieses Kommando muss der Benutzer den Vornamen und Namen des beta Entwicklers angeben, bei dem er die Aufgaben und Kommentare aktualisieren möchte.

**/drop:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/give:** Für dieses Kommando muss der Benutzer einen anderen Discord Benutzer als Handelspartner angeben sowie ob er Gold oder eine Karte an diesen geben möchte. Basierend auf dieser Auswahl muss er entweder angeben, wie viel Gold er abgeben möchte oder die Karten ID eingeben von der Karte, welcher er versenden möchte.

**/info:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/leaderboard total:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/leaderboard weekly:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden. Optional kann der Benutzer jedoch eine Woche angeben, von der er die Bestenliste sehen möchte.

**/shop:** Für dieses Kommando sind keine Parameter erforderlich und es kann direkt ausgeführt werden.

**/update date:** Für dieses Kommando muss der Benutzer ein Tag, Monat sowie ein Jahr eingeben, wann das nächste Event stattfindet.

**/upload tp:** Für dieses Kommando muss der Benutzer eine CSV Datei anhängen, die die TP Informationen an die Datenbank sendet.

**/user edit:** Für dieses Kommando kann der Benutzer Vorname, Name, Matrikel, Matrikelnummer und ein Profilbild angeben, welches in der Datenbank aktualisiert wird.

**/user registration:** Für dieses Kommando muss der Benutzer Vorname, Name, Matrikel, Matrikelnummer angeben sowie aus einer Auswahl das zugehörige Team wählen. Optional kann noch ein Profilbild angehängt werden.



## 4.2.2 Senden und Empfangen von Daten

Da das Senden und Empfangen von Daten ausschließlich über die Datenbank erfolgt, muss zunächst geklärt werden, wie die Verbindung zu dieser überhaupt zustande kommt.

```
hostname = "localhost"
username = "root"
passwordname = ""
databasename = "beta_bot"

databaseConnect = mysql.connector.connect(
    host=hostname,
    user=username,
    password=passwordname,
    database=databasename
)
cursor = databaseConnect.cursor(buffered=True)
```

Abbildung 21: Herstellen einer Datenbankverbindung in Python, Quelle: Quellcode

In Abbildung 21 sieht man den Aufbau einer Verbindungsherstellung zu einer MySQL Datenbank. Dafür werden zunächst vier String Variablen vereinbart, welche die nötigen Verbindungsinformationen enthalten. Wenn sich diese Daten ändern, muss dies nur im jeweiligen String einmal ersetzt und nicht für jeden Verbindungsversuch umgeschrieben werden. Diese String Variablen setzt man dann in das dafür vorgesehene Feld der „mysql.connector.connect“ Methode ein. Abschließend muss noch ein Cursor erstellt werden, welches durch die erstellte „databaseConnect“ Variable bewerkstelligt werden kann. Ein Cursor erlaubt eine zeilenweise Verarbeitung der Ergebnismengen. Sobald man die Datenbank nicht weiter benötigt, ist es wichtig, diese mit dem Schlüsselwort „close“ zu schließen.

Wie bereits in Abschnitt 4.1.1 erwähnt, erfolgt das Senden und Empfangen von Daten über sogenannte MySQL Statements. Während der Programmierung des Prototyps werden drei verschiedene Arten dieser Statements genutzt. Diese sollen im Folgenden genauer beschrieben werden.

```
cursor.execute("INSERT INTO generated_user_cards(Card_ID, Discord_ID, Name, Vorname, Team, "
              "Rarity_Points, Rarity, Obtained) VALUES(%s, %s, %s, %s, %s, %s, %s, %s)",
              (completecode, str(winnerid), cardStats[0], cardStats[1], cardStats[2],
              raritypoints[0], totalrarity[0], datetime.now(), ))
databaseConnect.commit()
```

Abbildung 22: MySQL Insert Statement in Python, Quelle: Quellcode

In Abbildung 22 kann man den Aufbau eines Insert Statements sehen. Eine Gemeinsamkeit von allen Statements ist, das sie alle über den Cursor ausgeführt werden, dies geschieht über „cursor.execute“. Ein Insert Statement beginnt immer mit „INSERT INTO“ gefolgt von dem Tabellennamen, in der eine Neue Zeile eingefügt werden soll. In Klammern werden danach alle Tabellen Spalten genannt, in welche ein Wert später hinzugefügt wird. Es folgt der Begriff „VALUES“ und in Klammern dahinter die einzelnen Werte, welche eingetragen

werden sollen. Möchte man eine in Python erstellte Variable einfügen, gibt man zunächst ein „%s“ als Platzhalter an. Für jeden Platzhalter muss am Ende eine Variablen eingefügt werden. Abschließend muss noch ein „Datenbank Commit“ erfolgen, um die Übertragung der Daten zu bestätigen sowie freizuschalten.

```
cursor.execute("UPDATE album SET Background = %s WHERE Discord_ID = %s AND Album_Name = %s",
              (background_name, str(ctx.author.id), album_name, ))
databaseConnect.commit()
```

**Abbildung 23: MySQL Update Statement in Python, Quelle: Quellcode**

In Abbildung 23 wird der Aufbau eines Update Statements dargestellt. Neben dem bereits Genannten, ist an dieser Stelle neu, dass das Statement mit „UPDATE“ beginnt, gefolgt vom Namen der Tabelle, welche aktualisiert werden soll. Es folgt das Wort „SET“ zusammen mit dem Namen der zu aktualisierenden Spalte sowie dessen neuer Wert. Nach dem Wort „WHERE“ wird eingegrenzt, welche Zeilen aktualisiert werden sollen. Das Wort „AND“ sorgt für mehr Bedingungen, um so noch genauer die Zeilen aktualisieren zu können.

```
cursor.execute("SELECT Name, Vorname, Team FROM beta_member WHERE Discord_ID = %s",
              (droppedcards[2],))
countrows = cursor.rowcount
cardStats = cursor.fetchone()
```

**Abbildung 24: MySQL Select Statement in Python, Quelle: Quellcode**

In Abbildung 24 wird der Aufbau eines Select Statements veranschaulicht. Die bisher gezeigten Statements haben alle die Funktion, Daten an die Datenbank zu versenden. Mit dem Select Statement ruft man diese Daten aus der Datenbank ab. Begonnen wird dabei mit dem Wort „SELECT“ gefolgt von den Tabellenspalten, welche man abrufen möchte. Es folgt die Bezeichnung „FROM“ und der Name der Tabelle, aus der die Daten abgerufen werden sollen. Abschließend folgt mit „WHERE“ die Bedingung zum Abrufen der Daten. Durch das Schlüsselwort „rowcount“ lässt sich in eine Variable die Anzahl an gefundenen Zeilen speichern. Durch die Schlüsselwörter „fetchone“ / „fetchall“ kann eine/mehrere Zeileneinträge in ein Array gespeichert werden.

### 4.3 Test des Prototyps

Während dieser Bachelorarbeit wurde zu Test- und Evaluierungsabsichten des Prototyps ein Probandentest durchgeführt und anschließend mithilfe einer digitalen Umfrage beurteilt (siehe Anhang 1). Im Folgenden soll auf die Erstellung und den Ablauf des Tests genauer eingegangen werden. Weiterhin soll geklärt werden, wie die Organisation sowie Durchführung der Testphase verlaufen ist sowie was am Ende bei der Auswertung herausgekommen ist. Zuletzt folgt zu dem gesamten Probandentest ein Fazit.

### 4.3.1 Testkonzeption und Ablauf

Der entstandene Prototyp hat mit der Arbeitserleichterung sowie Moral- und Leistungssteigerung der Mitglieder eines Projektes bereits eine vorgegebene Zielsetzung. Er soll vor allem Projektmanager bei ihrer Arbeit unterstützen und gleichzeitig allen Projektteilnehmer einen Motivationsschub in ihrem Projekt geben. Bei der Wahl der Probanden liegt es nahe, den entwickelten Prototyp beim diesjährigen beta Projekt einzusetzen und von allen Entwicklern, Feedback dafür abzuholen. Leider befindet sich die beta aktuell in den Semesterferien, was ein Testen des Prototyps unmöglich macht. Allerdings spricht der Chatbot nicht nur speziell das beta Projekt an, sondern auch jedes andere Projekt, welches mit Discord arbeitet und ein Projektmanagement Team besitzt. Es wurden neun Personen eingeladen, den Prototypen mit allen seinen Funktionen und Slash Commands zu testen sowie anschließend ihre Eindrücke innerhalb einer Umfrage festzuhalten. Die Personen wurden dafür in drei verschiedene Gruppen eingeteilt. Dabei wurden diese nach ehemaligem beta Teilnehmer sowie Alter unterteilt. Daraus lässt sich schlussfolgern, dass für den Test auch ältere Personen mit wenig Internet- und Softwarebezug eingeladen wurden, um auch vor allem die Benutzerfreundlichkeit und Bedienbarkeit des Prototyps zu testen.

Die drei Testgruppen können folgendermaßen dargestellt werden:

- **Testgruppe 1 – Probanden {1}, {2}:**  
Ehemalige beta Teilnehmer im Alter von 22 bis 24 Jahre, verstärkter Internet- und Softwarebezug
- **Testgruppe 2 – Probanden {3}, {4}, {5}, {6}, {7}:**  
Personen, welche an eigenen Projekten Arbeiten im Alter von 20 bis 29 Jahre, verstärkter Internet- und Softwarebezug
- **Testgruppe 3 – Probanden {8}, {9}:**  
Personen im Alter von 55 bis 61 Jahre, wenig Internet- und Softwarebezug

Während des gesamten Tests wurden alle Probanden, welche zu dem Zeitpunkt nicht physisch anwesend waren, dazu aufgefordert, ihren Bildschirm als Livestream zu übertragen. Dadurch war es möglich, bei allen Probanden ihr Vorgehen bei der Interaktion mit dem Chatbot sowie dem Ausfüllen der Umfrage genau zu beobachten. Bei der Umfrage wurde sich für offene und geschlossene Fragen entschieden, um den Personen so die Freiheit zu geben, ihre Wünsche und Gedanken als freies Feedback äußern zu können. Die nachfolgende Tabelle soll wichtige Informationen zum Konzept sowie Ablauf des Probandentests noch einmal zusammenfassen.

Ziel des Prototyps	<ul style="list-style-type: none"> <li>• herauszufinden, ob er den Benutzern ihre Arbeit erleichtert und ob er zu einer Moral- sowie Arbeitssteigerung beitragen kann</li> </ul>
Erforderliche Informationen für die Probanden	<ul style="list-style-type: none"> <li>• Verwendungszweck des Prototyps</li> <li>• Einladungslink zum Discord Server</li> <li>• Erteilte Berechtigungen auf dem Discord Server</li> <li>• Nutzung von Slash Commands</li> <li>• Link zum Download einer CSV Datei</li> </ul>
Erwarteter Ablauf während des Tests	<ul style="list-style-type: none"> <li>• dem Discord Server über bereitgestellten Link beitreten</li> <li>• benötigten Rechte auf dem Discord Server erhalten</li> </ul>

	<ul style="list-style-type: none"> <li>• die Projektmanagement orientierten Slash Commands ausführen</li> <li>• die beta Card Game Slash Commands ausführen</li> </ul>
Erfolgskriterien	<ul style="list-style-type: none"> <li>• der Prototyp hat alle Anfragen der Probanden erfolgreich bewältigt</li> <li>• der Prototyp weist keine störenden Fehler auf</li> <li>• die Probanden wissen den Prototyp gezielt für ihre Probleme einzusetzen</li> <li>• die Probanden haben Spaß am beta Card Game</li> </ul>
Überwachung	<ul style="list-style-type: none"> <li>• Beobachtungen zum Vorgehen der Probanden, während sie mit dem Prototyp interagieren</li> </ul>
Umfrage	<ul style="list-style-type: none"> <li>• erfolgt im Anschluss und ist von jedem Probanden auszufüllen</li> </ul>

**Tabelle 4: Testkonzept des Prototypen, Quelle: Eigene Darstellung**

### 4.3.2 Durchführung der Testphase

Vor Beginn jeder einzelnen Testphase wurde den Probanden Hintergrund und Zweck des Prototyps erklärt, um einen reibungslosen Ablauf ohne Unterbrechungen zu gewährleisten. Auch eine Aufgabenstellung wurde im Vorfeld skizziert, um einen klaren Ablauf für den Test zu etablieren. Allen Probanden wurde die gleiche Aufgabe gestellt, die wie folgt lautet:

1. Klicken Sie auf den bereitgestellten Link, um auf den „Bot Test“ Discord Server zu gelangen.
2. Schauen Sie sich die Rechte an, welche Sie auf dem Server erhalten haben, damit Sie wissen, ob Sie Admin Kommandos ausführen können oder nicht. Sollten Sie keine Rechte besitzen, weisen Sie den Veranstalter darauf hin.
3. Klicken Sie in der Discord Kanal Hierarchie auf den Text-Kanal mit ihrem Namen.
4. Testen Sie bitte die folgenden Kommandos:
  - /betadate
  - /update betadate
  - /developer information, unter Eingabe des Namens: Leon Erdhütter
  - /developer update, unter Eingabe des Namens: Leon Erdhütter
  - /leaderboard total
  - /leaderboard weekly, auch unter Eingabe einer Zahl für die Woche
  - /upload tp, verwenden Sie die dafür bereit gestellte CSV Datei, in welcher Sie gerne Änderungen an den Zahlen vornehmen dürfen
5. Benutzen Sie bitte das Kommando /user registration um sich für das beta Card Game zu registrieren.
6. Versuchen Sie nun alle bisher noch nicht ausgeführten Kommandos zu verwenden, wie das Ausgeben einer Karte, das Erstellen eines Albums und so weiter.

7. Um 14 Uhr erfolgt ein Commit auf dem GitLab Repository des Chatbots sowie eine neue Kartenerstellung auf dem Chatbot Trelloboard. Bitte schauen Sie sich die Ausgaben des Prototypen im Textkanal „git-info“ und „trello-info“ an.
8. Sollten Sie fertig mit dem Testen sein, befindet sich im Textkanal „survey“ ein Link zu einer Umfrage, welche Sie bitte in Ruhe bearbeiten sollen (Ab diesem Punkt wird der Bildschirm der Probanden nicht länger übertragen und beobachtet).

Während der Prototyp getestet wurde, wurden wie bereits angedeutet, Beobachtungen über das Vorgehen der Probanden {P} sowie deren Denkweise durch akustische Äußerungen mitverfolgt. Im Nachfolgenden sollen diese zusammengefasst beschrieben werden:

- P {1}, {2}, {3}, {4}, {5}, {6} und {7} schaffen es ohne Probleme auf den Discord Server und beginnen direkt mit dem Prototyp zu interagieren.
- P {8} und {9} hatten Probleme mit dem Umgang von Discord und benötigten etwas Hilfe, um auf den Server sowie den entsprechenden Textkanal zu kommen.
- P {1} und {4} versuchten zuerst über den Direktnachrichtenverlauf mit dem Chatbot zu kommunizieren anstatt über den dafür vorgesehenen Textkanal.
- P {6} und {7} schauten sich zunächst das Discord Profil des Chatbots an.
- P {2}, {3}, {5}, {8} und {9} starteten direkt mit der Eingabe der in der Aufgabenstellung definierten Slash Commands.
- P {1}, {4} und {5} versuchen durch gezielte Falscheingaben von Variablen sowie durch Hochladen von falschen Dateitypen den Prototyp auf Fehler zu testen.
- P {1} äußerte sich positiv über das Aussehen und der Anordnung der Bestenlisten.
- P {5} war positiv überrascht, dass der Prototyp für jeden Fehlertest eine passende Antwort parat hat, worin er erklärt, was an der Eingabe falsch gemacht wurde.
- P {1}, {2}, {4}, {5}, {6} und {7} haben sichtlich viel Spaß am beta Card Game.
- P {1} und {2} sind fasziniert von der gegenseitigen Interaktion über den Chatbot beim Tauschen einer Karte.
- P {3}, {8} und {9} haben zunächst Probleme beim beta Card Game und wissen nicht, was sie machen sollen.
- P {1}, {2}, {4} und {6} sind von der plattformübergreifenden Kommunikation mit Trello und Git begeistert und wollen diese Methode für ihre Projekte auch nutzen.
- P {1}, {3} und {4} benötigten für den Test die wenigste Zeit (15 Minuten).
- P {8} und {9} benötigten für den Test die meiste Zeit (25-30 Minuten).
- Alle können das Testen erfolgreich abschließen und füllen im letzten Schritt die für den Test erstellte Umfrage aus.

### 4.3.3 Auswertung der Testphase

Wie bereits erwähnt, wurden alle Probanden nach der Verwendung des Prototyps dazu angehalten, ihre gesammelten Eindrücke und Erkenntnisse in einer Umfrage festzuhalten und diesen somit zu bewerten. Das Ausfüllen des Fragebogens erfolgte anonym und alle neun Probanden nahmen daran teil. Nachfolgend sollen nun die verschiedenen Fragen mit ihren entsprechenden Ergebnissen ausgewertet werden. Da für die Umfrage die kostenlose Umfrageverwaltungssoftware „Google Formulare“ verwendet wurde, ist eine Auswertung in Form von Diagrammen bereits automatisch darüber erfolgt. Diese generierten Diagramme werden im Folgenden zur Veranschaulichung verwendet. Diagramme mit ähnlichem Kontext zu bereits ausgewerteten sowie weniger relevante Fragen werden in Anhang 2 angefügt.

#### Befragung zur Person

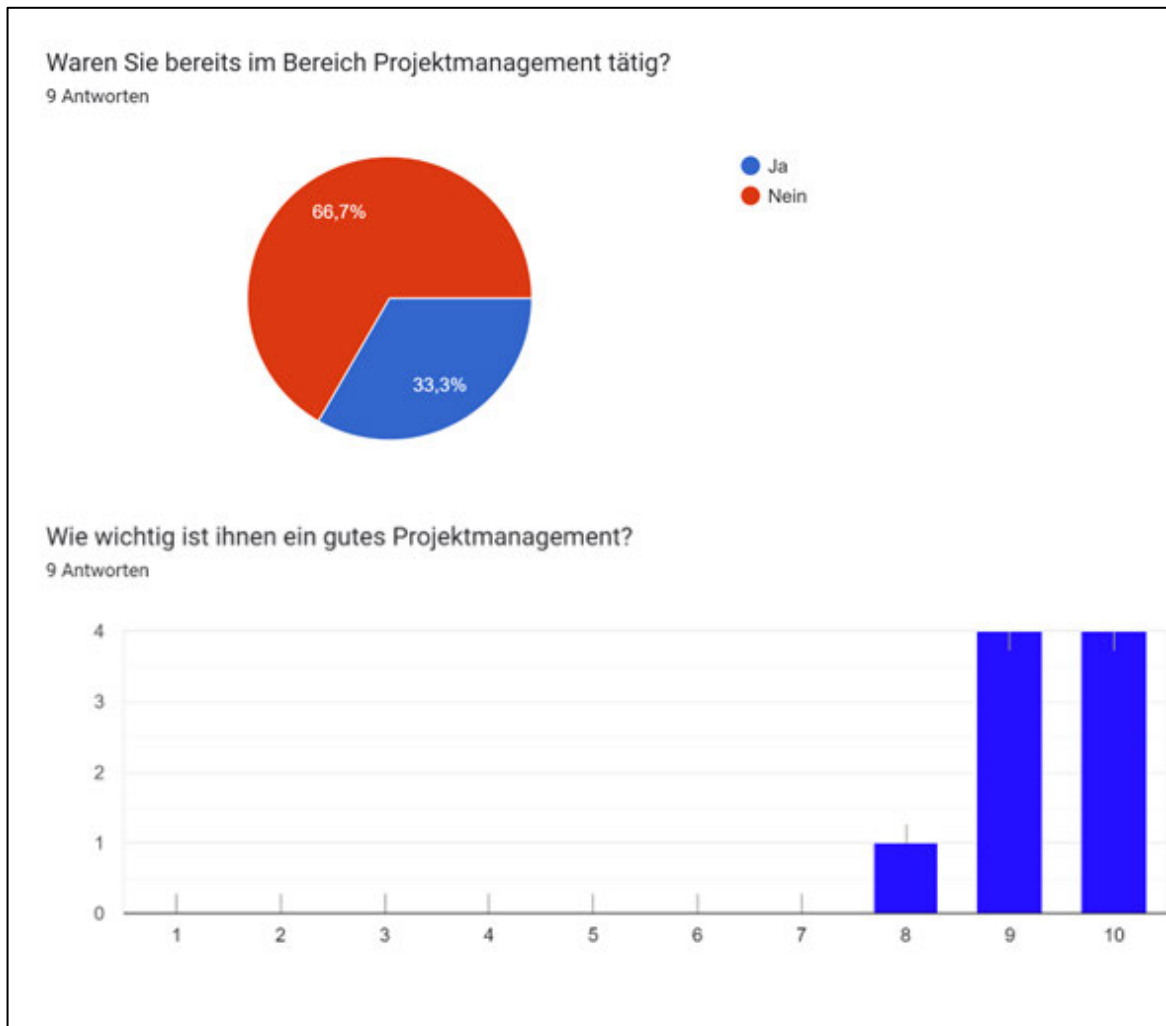


Abbildung 25: Fragebogen Auswertung – Persönliche Informationen, Quelle: Umfrage über Google Formulare

Zu Beginn der Umfrage wurden zunächst einige Fragen zur Person der Probanden gestellt. Dabei wurde zunächst das Alter abgefragt, was ergab, dass der Großteil der Tester im Altersbereich zwischen 20 und 30 Jahren lag. Lediglich zwei Personen waren mit 55 sowie

61 Jahren deutlich älter. Durch diese Altersunterschiede sowie auch der Internet- und Softwareaffinität der Probanden war es möglich, den Prototyp sehr gut auf Benutzerfreundlichkeit und Verständlichkeit zu prüfen. Herausgekommen ist dabei, dass die älteren Personen zwar deutlich mehr Probleme mit dem Umgang des Prototyps hatten, aber mit etwas Hilfe den gesamten Test sehr gut bewerkstelligen konnten. Abbildung 25 enthält eine Statistische Auswertung der weiteren Fragen zur Person. Es fiel auf, dass knapp über die Hälfte der Probanden einen Bachelor oder Masterabschluss anstreben und die verbleibenden vier die Antwortmöglichkeit andere gewählt haben. Bei der Frage nach der beta Angehörigkeit, kam heraus, dass zwei der neun Probanden bereits an einer beta teilnehmen oder teilgenommen haben. Diese gehörten dabei dem Team Eventmanagement sowie Gamedesign an.

### Befragung zum Thema Projektmanagement

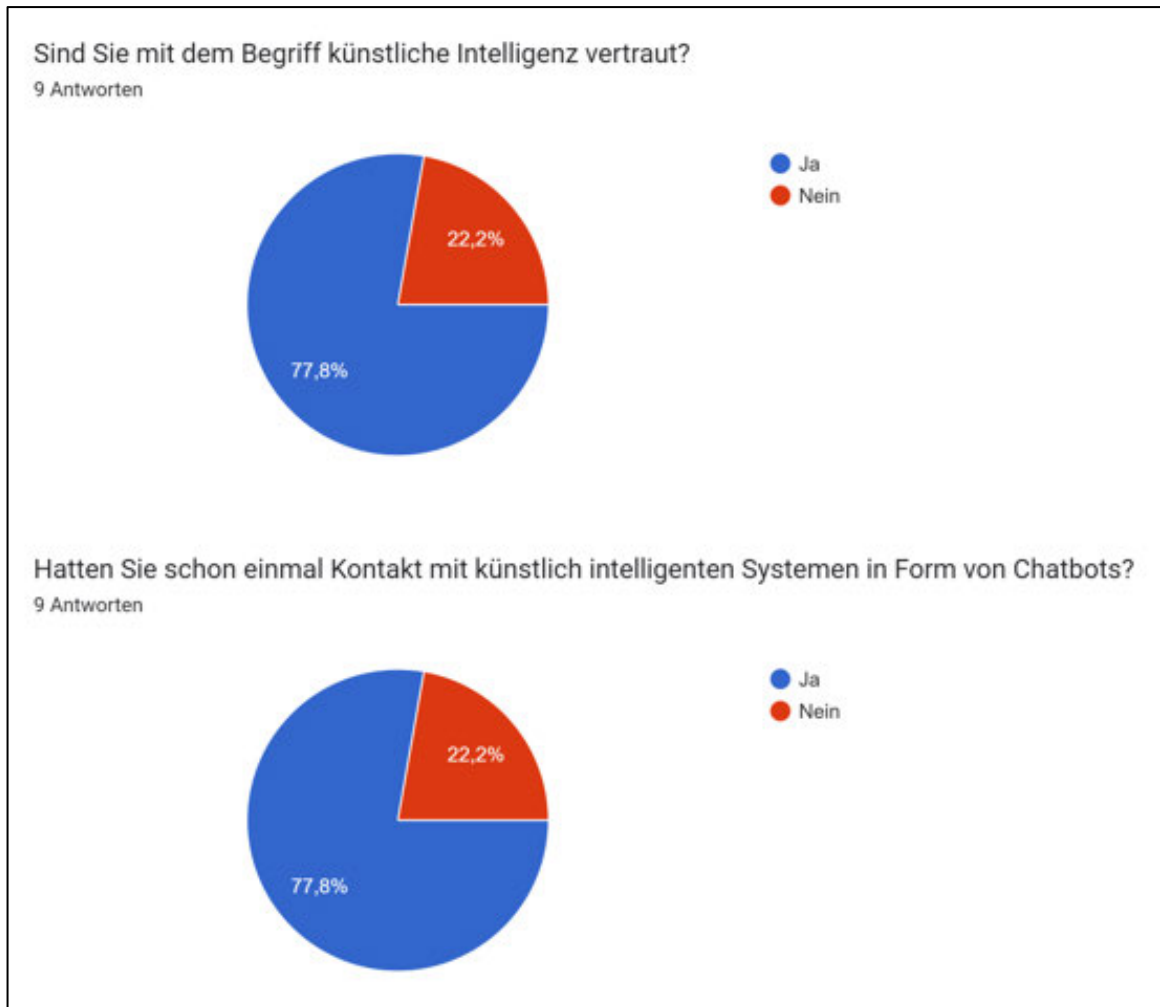


**Abbildung 26: Fragebogen Auswertung – Projektmanagement, Quelle: Umfrage über Google Formulare**

Im weiteren Verlauf der Umfrage wurden die Teilnehmer zum Thema Projektmanagement befragt. Dabei war zunächst interessant zu wissen, ob die Tester bereits selbst im Projektmanagement tätig waren. Abbildung 26 kann entnommen werden, dass tatsächlich ein Drittel der Befragten als Projektmanager beschäftigt waren oder es immer noch sind. Diese Personen haben bereits Erfahrungen im Bereich Projektmanagement gesammelt und wissen, worauf zu achten ist. Dies macht sie besonders wertvoll, um festzustellen, ob der Prototyp im Projektmanagement hilfreich sein kann. Weiterhin wurde auch nach der Bedeutsamkeit eines guten Projektmanagements gefragt. In Abbildung 26 ist zu erkennen, dass alle Probanden bei dieser Frage mit acht oder mehr Punkten abgestimmt haben und somit

ein gutes Projektmanagement für sie sehr wichtig in einem Projekt ist. Außerdem wurden die Probanden dazu Angehalten, ihre Antwort zu begründen. Dabei kam heraus, dass viele der Meinung sind, je besser das Projektmanagement ist, desto erfolgreicher verläuft am Ende das Projekt. Weiterhin hält es alle anderen Teams zusammen und sorgt für einen reibungslosen Projektverlauf.

### Befragung zum Thema Chatbots

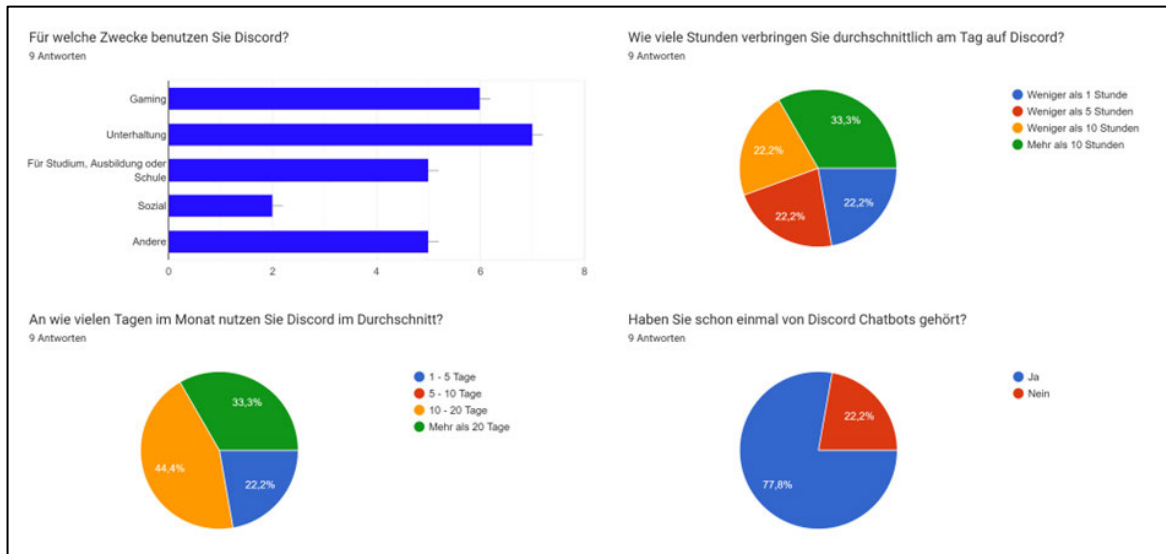


**Abbildung 27: Fragebogen Auswertung – Chatbots, Quelle: Umfrage über Google Formulare**

Ein weiterer wichtiger Punkt ist es, die Probanden zum Thema Chatbots zu befragen. Dabei wurde zunächst gefragt, ob diese überhaupt mit dem Begriff künstliche Intelligenz vertraut sind beziehungsweise sich darunter etwas vorstellen können. Mithilfe von Abbildung 27 ist zu sehen, dass sieben von neun befragten bereits mit künstlicher Intelligenz zu tun haben und wissen, was es bedeutet. Nur zwei Personen wussten mit dem Begriff nichts beziehungsweise wenig anzufangen. Von den eben genannten sieben Personen sind alle mindestens einmal mit künstlich intelligenten Systemen in Kontakt getreten. Auf die Frage, wo dieser Kontakt auftrat, haben alle direkt auf Online-Webseiten verwiesen, wo solche Chatbots den Benutzern in verschiedenen Bereichen helfen. Darunter zählt zum Beispiel das Ausfüllen eines Formulars oder das Erklären von einzelnen Funktionen der Webseite. Ein Proband gab an, dass er in einem KI-Labor bereits mit einem Roboter kommuniziert hat.



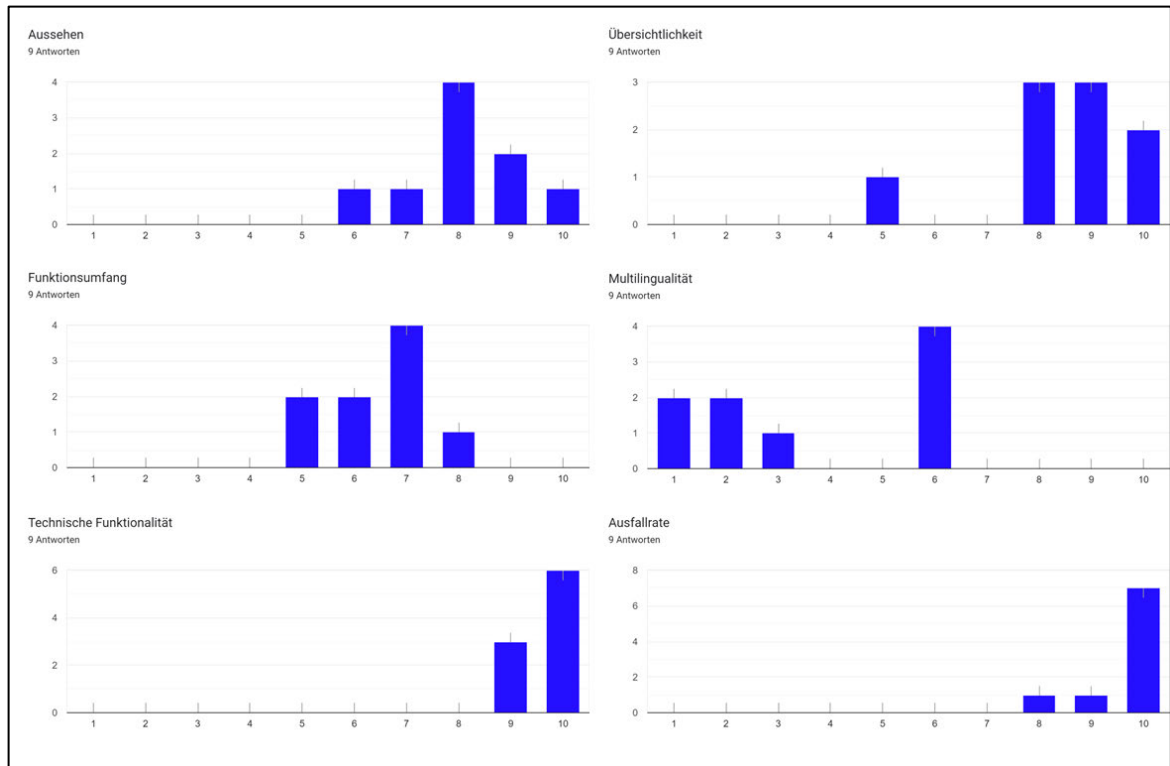
## Befragung zum Thema Kommunikationsplattform Discord



**Abbildung 28: Fragebogen Auswertung – Kommunikationsplattform Discord, Quelle: Umfrage über Google Formulare**

Da es sich bei dem Prototyp um einen Discord Chatbot handelt, ist es wichtig, auch diesen Punkt bei den Probanden abzufragen. Dabei soll auch herausgefunden werden, ob Discord als Anwendung für den Prototyp eine gute Wahl war oder es bessere Alternativen gibt. Zunächst wurde nachgefragt, wann die Probanden begonnen haben, Discord zu nutzen beziehungsweise ob sie überhaupt schon mal mit Discord zu tun hatten. Dabei gehen die Jahre der ersten Nutzung aller Tester weit auseinander und von 2015 bis 2022 ist fast alles enthalten. Lediglich zwei Probanden hatten vorher noch nie Discord benutzt und für diese war es das erste Mal im Umgang mit der Kommunikationsplattform. Auf die Frage, für welche Zwecke die Teilnehmer Discord nutzen, kann man Abbildung 28 entnehmen, dass die meisten es für Unterhaltung und Gaming benutzen. Aber auch für schulische sowie andere Zwecke wird Discord von den Probanden viel verwendet. Discord im sozialen Bereich wurde nur von zwei der neun Tester genutzt. Abbildung 28 zeigt weiterhin, dass der größte Teil der Probanden Discord zwischen zehn und zwanzig Tagen im Monat verwendet. Weitere drei Tester gaben an, Discord sogar an mehr als zwanzig Tagen im Monat zu verwenden, was ein hohes Konsumverhalten der Kommunikationsplattform vermuten lässt. Dies bestätigt auch noch mal die Nutzungsdauer, bei der die meisten Probanden Angaben Discord mehr als zehn Stunden täglich zu nutzen. Es lässt sich hierzu schlussfolgern, dass Discord eine wahrscheinlich sehr gute Wahl für den Prototypen ist, da das Konsumverhalten der Nutzer sehr hoch ist. Abschließend wurde auch nach dem speziellen Fall Discord Chatbots gefragt und Abbildung 28 lässt sich entnehmen, dass sieben von neun Testern auch damit bereits vertraut sind.

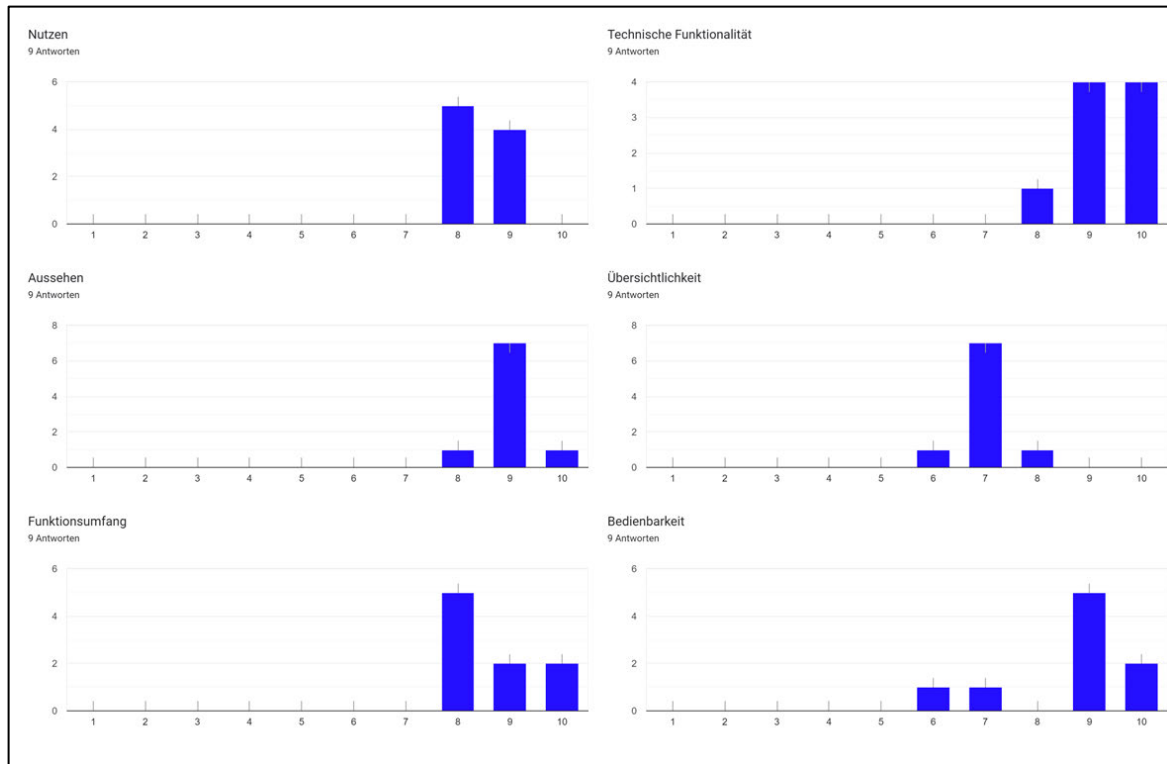
## Befragung zum Thema Chatbot Präferenzen



**Abbildung 29: Fragebogen Auswertung – Chatbot Präferenzen, Quelle: Umfrage über Google Formulare**

Während der Umfrage sollten die Probanden angeben, was für sie bei der Erstellung eines Chatbots wichtig ist. Dafür wurden sechs wichtige Präferenzkriterien ausgewählt, welche anschließend auf einer Skala von 1 (unwichtig) bis 10 (sehr wichtig) bewertet wurden. Daraus soll sich zeigen, welche Punkte bei einer anschließenden Überarbeitung des Prototyps am wichtigsten sind. In Abbildung 29 kann man die von den Probanden präferierten Ergebnisse ausgewertet betrachten. Dabei lässt sich erkennen, dass den Probanden eine technische Funktionalität sowie eine geringe Ausfallrate des Chatbots am wichtigsten ist. Es wird also viel Wert daraufgelegt, dass der Prototyp einwandfrei ohne Fehler oder Bugs funktioniert und dass er am besten immer online und erreichbar ist. Auch auf das Aussehen der ausgegebenen Nachrichten und die Übersichtlichkeit während der Interaktion mit dem Chatbot wird von den Probanden als wichtig empfunden. Als weniger wichtig als die anderen Punkte wurde der Funktionsumfang eingestuft. Daraus lässt sich erkennen, dass die Probanden nicht sehr viel Wert auf viele Nebenfunktionen legen und mit grundlegenden sowie funktionierenden Funktionen zufrieden sind. Als eher unwichtig wurde die Multilingualität eingestuft. Dies bedeutet, dass es zwar schön wäre, wenn der Chatbot mehrere Sprachen beherrscht, es aber nicht weiter schlimm ist, wenn er es nicht tut.

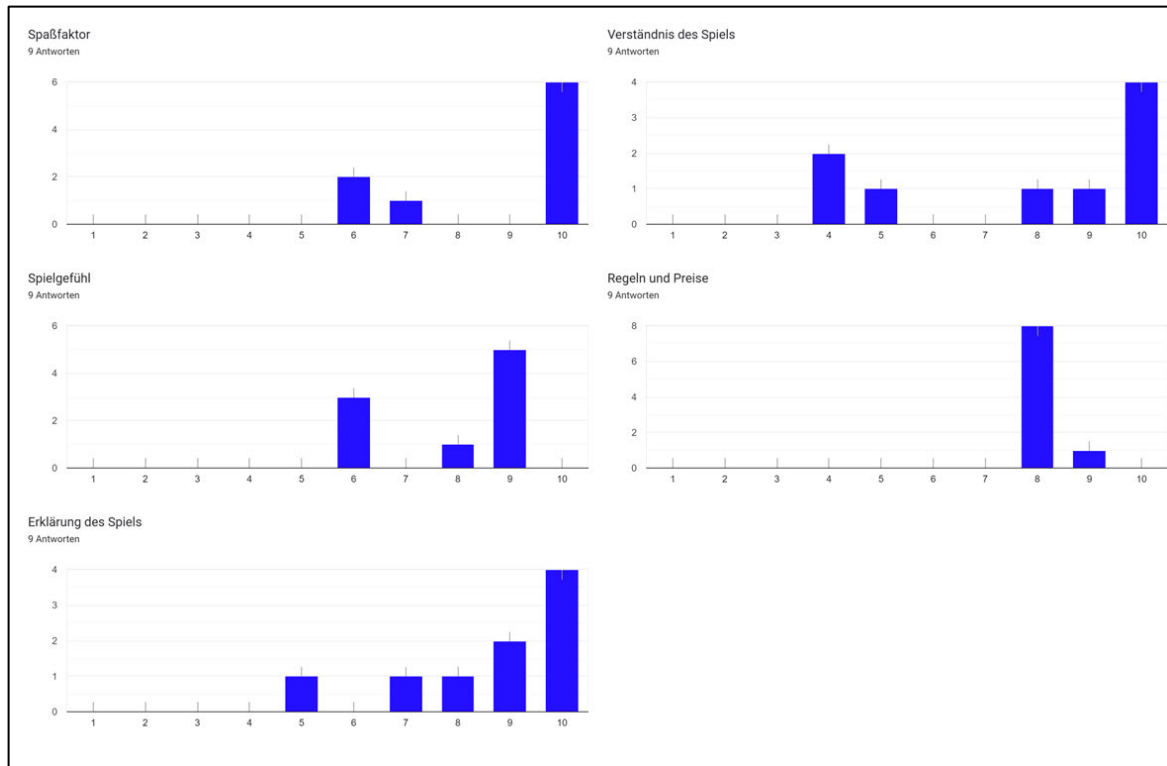
## Bewertung des Prototyps



**Abbildung 30: Fragebogen Auswertung – Bewertung des Prototyps, Quelle: Umfrage über Google Formulare**

Ähnlich wie bei den Chatbot Präferenzen wurde für die Bewertung des Prototyps sechs Kriterien ausgewählt, bei denen die Probanden auf einer Skala von 1 (schlecht) und 10 (sehr gut) bewertet haben. Dabei stach heraus, dass die technische Funktionalität, die beste Bewertung bekommen hat, wie in Abbildung 30 zu sehen ist. Demnach kann gesagt werden, dass der Prototyp während des Tests kaum Fehler aufgewiesen hat. Auch als sehr positiv wurde der Nutzen, das Aussehen der Chatnachrichten sowie der Funktionsumfang empfunden. Dazu gab es auch direktes Feedback von den Probanden, in dem sie den Aufbau und die interaktiven Möglichkeiten der Chatnachrichten lobten. Weiterhin hat es den Probanden gefallen, dass es nicht zu viele beziehungsweise zu wenige Kommandos waren und der Prototyp damit nicht überladen beziehungsweise nutzlos wirkte. Auch sehr gut bewertet, bis auf zwei stimmen, wurde die Bedienbarkeit. Das liegt wahrscheinlich an der eher ungewöhnlichen Bedienung des Prototyps mit den Slash Commands. Dies wird demnach von jeder Person anders angenommen. Eine solide Bewertung bekam die Übersichtlichkeit, dies liegt vor allem daran, dass die Slash Commands alle untereinander ohne Gruppierung und Sortierung gereiht sind. Leider unterstützt Discord zum Zeitpunkt der Erstellung des Prototyps keine brauchbare Alternative.

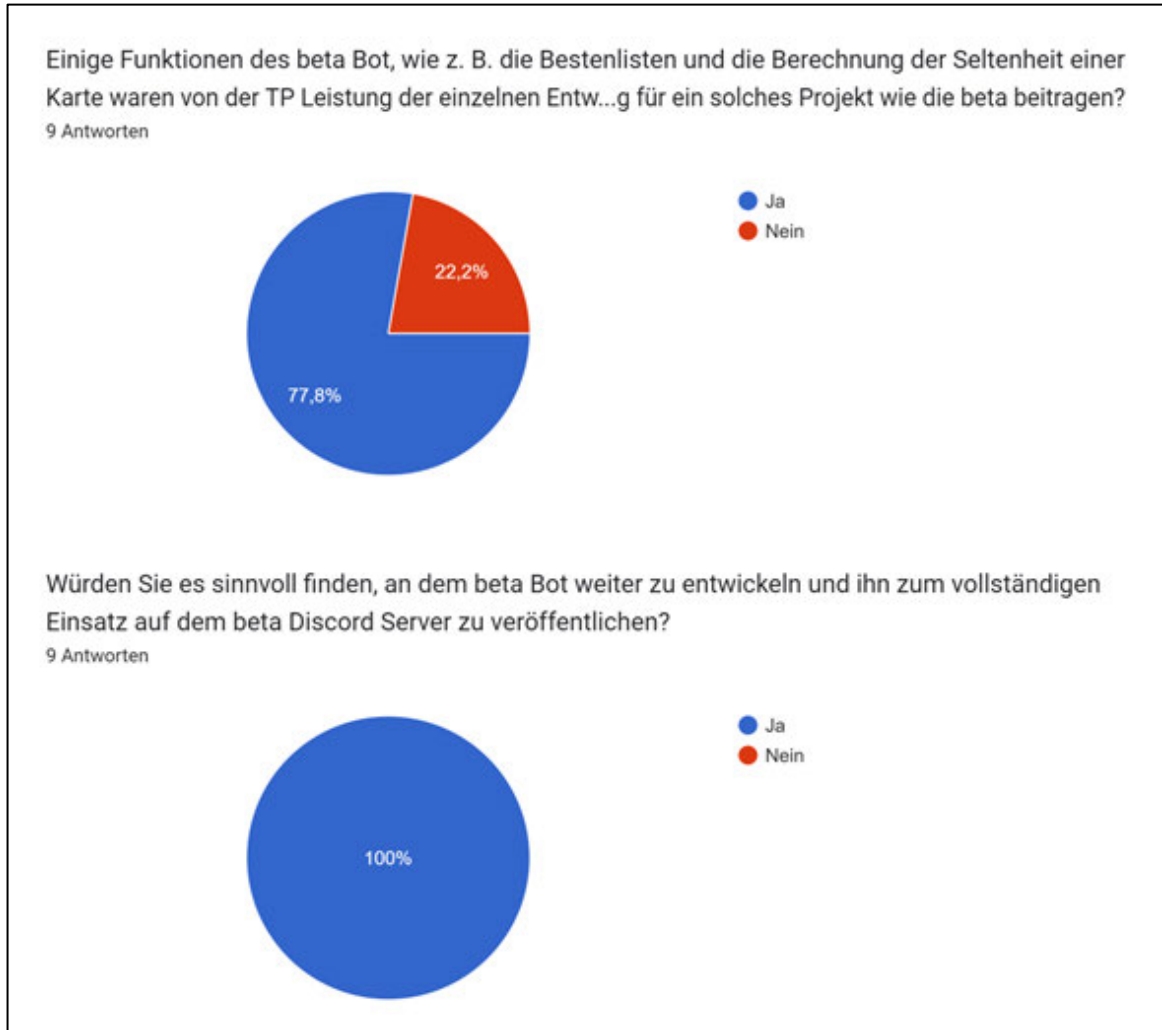
## Bewertung des beta Card Game



**Abbildung 31: Fragebogen Auswertung – Bewertung des beta Card Game, Quelle: Umfrage über Google Formulare**

Auch bei diesem Punkt wurde wieder eine Bewertung anhand von insgesamt fünf Kriterien von den Probanden gefordert. Auch hier ging die Skala von 1 (schlecht) bis 10 (sehr gut). Direkt in Abbildung 31 zu erkennen ist, dass die Meinungen der Tester weiter auseinandergehen als bei den vorherigen Bewertungen. Im Bereich Spaßfaktor gaben zwei Drittel der befragten an, dass sie sehr viel Freude mit dem Kartenspiel hatten und ein Drittel, dass sie zwar Spaß hatten, aber nicht so viel wie die anderen. Dasselbe gilt auch für das Spielgefühl, was bei der Mehrheit als sehr gut bewertet wurde, aber dennoch Schwächen aufweist. Die Erklärung des Spiels wurde auch als sehr gut empfunden, auch wenn das wahrscheinlich daran lag, dass der Verfasser in vielen Punkten ausgeholfen hat und diese nötige Erklärung nicht vom Prototyp selbst kam. Demnach kam auch der Wunsch nach einem Informationskommando zum Kartenspiel auf. Auch beim Verständnis des Spiels gab es große Unterschiede. Während zwei Drittel das Spiel sehr schnell verstanden haben, gab es bei einem Drittel doch einige Probleme, was sich in Abbildung 31 widerspiegelt. Bei den Regeln und Preisen waren sich fast alle einig und bewerteten dieses als gut. Auch hier gab es Anregungen und Wünsche von zu überkauften Preisen bis hin zu Änderungen an den Zieh-Wahrscheinlichkeiten der Karten.

## Fragen zum Ausblick des Prototyps



**Abbildung 32: Fragebogen Auswertung – Ausblick des Prototyps, Quelle: Umfrage über Google Formulare**

Zum Abschluss der Umfrage wurden noch einmal zwei für diese Arbeit sehr wichtigen Fragen gestellt. Die erste Frage ist dabei, ob die Probanden glauben, dass die Nutzung von Bestenlisten oder die Berechnung der Seltenheit einer Karte basierend auf den Einzelleistungen der Entwickler zu einer Moral- beziehungsweise Arbeitssteigerung in einem Projekt beitragen kann. In Abbildung 32 kann man sehen, dass sieben der neun Tester an ein solches System glauben und äußerten außerdem, dass sie sich selbst noch mehr in einem solchen Projekt steigern würden, um zum einen in der Bestenliste verewigt zu sein sowie eine äußerst seltene Karte ziehen zu können. Abschließend wurden die Probanden danach gefragt, ob sie es sinnvoll finden würden, den Prototyp weiterzuentwickeln und für viele Projekte zugänglich zu machen. Wie Abbildung 32 zu entnehmen ist, sind sich alle Tester einig und denken, dass es sich auf jeden Fall lohnen würde, den Prototypen weiterzuentwickeln.

#### 4.3.4 Fazit

Letzten Endes lässt sich zur Bewertung des gesamten Probandentests sagen, dass der Prototyp insgesamt sehr positiv von allen Teilnehmern aufgenommen wurde. Es gab vermehrt sehr gutes Feedback, welche vor allem das Aussehen der Chatnachrichten, die Falscheingaben Bearbeitung sowie das beta Card Game beinhalteten. Weiterhin waren das Arbeiten und Interagieren mit dem Prototyp für die meisten Teilnehmer gut verständlich und wurde sogar als intuitiv bezeichnet. Auch das Arbeiten mit den Discord Slash Commands war für niemanden eine Hürde und wurde als sehr gutes Mittel zur Kommunikation mit dem Prototyp aufgenommen.

Auffälligkeiten während des Tests waren Unsicherheiten bei der Kommunikation vor allem älterer Teilnehmer, da der Prototyp nur in Englisch kommuniziert und nicht in Deutsch. Weiterhin wurde angemerkt, dass die Übersichtlichkeit an manchen Stellen und vor allem der Kommandos in Discord nicht optimal dargestellt werden. Außerdem war das beta Card Game nicht für alle Teilnehmer schlüssig und sorgte an einige Stellen für Verwirrung. Zuletzt war ein weiteres Problem, dass der Prototyp Lokal über den Computer des Verfassers lief und somit zum einen nur eine schlechte, aber für den Test ausreichende Internetanbindung besaß und zum anderen somit außerhalb des Tests nicht online sein konnte.

Basierend darauf wurden verschiedene Wünsche von den Probanden geäußert. Darunter zählen Multilingualität, sodass der Prototyp neben Englisch auch auf Deutsch bedient werden kann sowie eine bessere Übersicht der Slash Commands in der Textzeile. Leider ist dies aber von Discord Seite aus beschränkt und lässt dafür wenig Spielraum übrig. Weiterhin wird sich ein Kommando gewünscht, welches eine komplette Erklärung zum beta Card Game mit all seinen Funktionen beinhaltet. Zuletzt wurde noch der Wunsch nach einer geringen Ausfallrate geäußert, sodass der Prototyp jederzeit online zu erreichen ist. Dies ist leider mit Kosten verbunden, da so gut wie alle Hostinganbieter im Discord Chatbot Bereich nur im monatlichen Abo erhältlich sind.

Auch die wichtigen Begrifflichkeiten, welche in Abschnitt 2.1 untersucht wurden, haben innerhalb der Testphase Beachtung gefunden. So wurde beispielsweise anhand der Mensch-Computer-Interaktion beachtet, wie die Probanden mit dem Prototyp umgegangen sind. Weiterhin wurde innerhalb des Interaction Design versucht, die Aufgaben der Tester mit dem Chatbot bestmöglich zu unterstützen. Auch im Bereich User Centred Design wurde sich an den in Abschnitt 2.1.4 definierten UCD-Prozess gehalten und der Prototyp anhand der vier Phasen auf die Bedürfnisse der Benutzer abgestimmt. Das User Interface wurde als eine angenehme Benutzeroberfläche mit guten Interaktionsmöglichkeiten von den Probanden bewertet. Auch die Gesamterfahrung im Bezug auf die User Experience wurde sehr positiv von den Testern beurteilt.

Bei der weiteren Bewertung des Prototyps fiel auf, dass die von den Probanden als wichtig Präferierten Kriterien gut bis ausgezeichnet erfüllt wurden. Weiterhin übertrafen die ausgezeichnete Bewertung und das angenehme Feedback der Mehrheit der Tester die Erwartungen des Verfassers. In Bezug auf den abschließenden Teil der Umfrage lässt das Bewertungssystem innerhalb des Prototyps zur Moral- und Arbeit Steigerung der Projektmitglieder vermuten, dass es tatsächlich funktionieren kann. Weiterhin waren alle Probanden der Meinung, dass es sich lohnt, den Prototypen weiterzuentwickeln, was die Annahme erzeugt, dass er seinen Nutzen sehr gut erfüllt.

Besonders auffällige oder schwerwiegende Fehler beziehungsweise Probleme sind während des gesamten Tests nicht aufgetreten. Die gesammelten Verbesserungsvorschläge der Probanden wurden zum Teil nach der Testphase direkt implementiert, sodass der zum Teil genutzte dreistufige iterative Entwicklungsprozess beibehalten wurde. Aufwendige Verbesserungen, welche im Rahmen dieser Arbeit vor allem aus Zeitgründen nicht mehr geleistet werden können, werden im Ausblick noch einmal aufgeführt.

## 5 Zusammenfassung und Ausblick

Das Ziel der vorliegenden Bachelorarbeit war es, herauszufinden, ob künstlich intelligente Systeme in Form von Chatbots dazu beitragen können, Projekte vor allem im Bereich Projektmanagement zu unterstützen. Aber auch außerhalb des Projektmanagements galt es zu untersuchen, ob verschiedene spielerische Funktionen zu einer Moral- und Arbeitssteigerung bei verschiedenen Projektmitgliedern beitragen kann. Mithilfe eines entwickelten Prototypen sollte die Erfüllung dieses Ziels angestrebt werden. Dafür wurden verschiedene Personen zu einem Probandentest eingeladen, welche den Prototypen ausgiebig testen und anschließend bewerten sollten. Nach der Finalisierung des Prototyps kann dieser vollständig in ein größeres Projekt wie zum Beispiel der beta Mittweida integriert werden.

Zu Beginn wurde eine ausführliche Recherche zum Thema der Bachelorarbeit betrieben, welche sich in mehrere Phasen unterteilt. Dabei wurden zuerst wichtige und für den Kontext der Arbeit bedeutsame Begrifflichkeiten erklärt sowie tiefgreifende Einblicke in die Thematik Projektmanagement sowie Chatbots geboten. Weiterhin wurden für den späteren Prototypen weitere unabdingliche Themen untersucht, wozu Pipedream, die Kommunikationsplattform Discord sowie die beta Mittweida gehören. Die gewonnenen Erkenntnisse konnten im Verlauf der Arbeit sehr gut unterstützen und haben für eine solide Grundlage gesorgt, auf welcher alles Weitere optimal aufbauen konnte.

Um sich gut auf die Implementierungsphase vorzubereiten, wurden zunächst die Anforderungen an den Prototyp mit dem Definieren eines Anwendungsszenarios in der Konzeptionsphase eingeleitet. Außerdem wurden die einzelnen funktionalen sowie nicht-funktionalen Anforderungen beschrieben. Weiterhin wurde eine grafische Benutzeroberfläche konzipiert, anhand der verfügbaren Mitteln, welche Discord zu bieten hatte. Abschließend wurde eine Übersicht erstellt, mit allen Kommandos, welcher der Prototyp beinhalten sollte. Für einen Discord Chatbot als Prototyp wurde sich entschieden, weil diese sehr viel zu bieten haben, Discord eine große Anzahl an Nutzern hat und viele Projekte, einschließlich der beta die Kommunikationsplattform intensiv nutzen. Eine der größten Hürden war es jedoch, so viel wie möglich aus dem Chatbot herauszuholen, da viele Funktionen an der ein oder anderen Stelle doch etwas beschränkt waren.

Die Implementierungsphase verlief ohne größere Probleme und schnell konnte die Entwicklungsumgebung sowie die Datenbank aufgesetzt und miteinander verknüpft werden. Die Programmiersprache Python war dafür eine sehr gute Wahl, da sich der Prototyp damit perfekt umsetzen ließ und auch die erweiterten Bibliotheken dafür gesorgt haben, dass viele Funktionen makellos funktionieren. Auch die zuvor definierten Kommandos wurden alle umgesetzt und auf Fehler getestet. Weiterhin wurden die Benutzereingaben klar definiert und am Ende ein Probandentest mit neun Teilnehmern durchgeführt.

Die Auswertung der Testphase hat vor allem die nachfolgenden Aspekte zum Vorschein gebracht. Der Prototyp konnte in der Gesamtbewertung die Probanden überzeugen. Vor allem der Nutzen, das Aussehen der generierten Chatnachrichten, der Funktionsumfang und die technische Funktionalität stachen besonders positiv hervor. Zwar auch sehr positiv aber mit ein paar Abzügen wurde die Bedienbarkeit ausgewertet. Dies lässt sich damit erklären, dass das Slash Command System nicht von jedem gut aufgenommen wurde und es an vielen Stellen deutliche Probleme beim Verständnis gab. Auch solide, aber im Vergleich zu den anderen Kriterien am schlechtesten wurde die Übersichtlichkeit bewertet. Das liegt vor allem an der unübersichtlichen Präsentation der Kommandos über die Textzeile innerhalb von Discord.

Innerhalb der Auswertung der Testphase wurden von den Probanden einige Anmerkungen und Wünsche geäußert, welche im Folgenden noch einmal aufgegriffen werden sollen. Da wäre zuerst der Wunsch nach Multilingualität des Chatbots. Im Moment kommuniziert dieser nur in englischer Sprache. Es wurde sich gezielt dafür entschieden, da Englisch zum einen die Weltsprache ist und zum anderen es immer sein kann, das in einem Projekt, zum Beispiel auch bei der beta, eine Person mitarbeitet, die nicht deutsch spricht, welche dann gar nicht mit dem Chatbot interagieren könnte. Zum Zeitpunkt der Erstellung des Prototyps war es von Discord Seite aus nur machbar, mehrere Sprachen zu unterstützen, wenn man mehrere Chatbots programmiert. Da dies aber keine wirklich gute Lösung ist, wurde davon abgesehen und dieser Lösungsweg nicht weiter verfolgt. Zum jetzigen Stand hat Discord einige Updates bekommen und tatsächlich auch eine Unterstützung von Mehrsprachlichkeit realisiert. Diese befindet sich aber in einer frühen Testphase und es ist nicht klar, dass alles problemlos funktioniert. Sollte der Prototyp weiter entwickelt werden, ist dies ein Ansatz, welcher auf jeden Fall verfolgt werden sollte. Doch gerade aus zeitlichen Gründen sowie dem Aufwand, alle Chatnachrichten noch einmal zu übersetzen, konnte dies im Rahmen der Arbeit nicht geschafft werden. Ein weiterer Wunsch war es, die Übersichtlichkeit vor allem der Slash Commands zu verbessern und in einzelne Gruppen einzuteilen. Leider ist auch dies nicht mehr möglich, innerhalb dieser Arbeit durchzuführen, da es seitens Discord noch keine Unterstützung dafür gibt. Auch hier bleibt zu hoffen, dass die Slash Commands irgendwann noch einmal überarbeitet werden und diese Funktion implementieren. Sollte dies der Fall sein, ist es auch problemlos möglich, den Prototypen auf dieses System umzustellen.

Andere Verbesserungsvorschläge und Wünsche, welche vom zu erbringenden Aufwand geringgehalten werden konnten, wurde direkt im Anschluss an die Testphase umgesetzt. Dazu zählen, dass Einfügen eines Informationskommandos für das beta Card Game. Dieses gibt nun detaillierte Informationen zu allen möglichen beta Card Game Slash Commands aus und erklärt außerdem die Wahrscheinlichkeiten und Regeln jedes einzelnen Kommandos. Weiterhin wurden viele Anpassungen an den Preisen sowie Wahrscheinlichkeiten vorgenommen, um das beta Card Game fairer zu gestalten. Einige kleine aufgetretene Bugs während der Testphase wurden ebenfalls bearbeitet, um somit eine fehlerfreie Arbeit des Prototyps zu gewährleisten.

Auch der Versuch, ein System zu entwickeln, welches zur Moral- und Arbeitssteigerung aller Projektmitglieder beiträgt, konnte seine ersten Erfolge verbuchen. So waren die meisten Tester überzeugt, dass ein solches System bei ihnen so etwas auslösen könnte, um am Ende in der Bestenliste ganz oben zu sein oder eine sehr seltene Karte von sich selbst für das beta Card Game zu generieren. Ein weiterer Ansatz und um dieses System noch weiter zu stärken, ist es, ein Kampfsystem zwischen den Karten zu implementieren. Dadurch bekommen seltene Karten noch mehr Wert, was wiederum die Projektmitglieder dazu bringen könnte, noch mehr Leistung in das Projekt zu stecken. Dies ist aber auch nur ein weiterer Ausblick für den Prototypen und letzten Endes lässt sich sagen, dass noch sehr viel aus diesem herausgeholt werden kann.

Doch die abschließende Frage, ob sich der Prototyp in der Praxis wirklich für ein Projekt eignet und ob er dort in erster Linie die Projektmanager sowie auch die restlichen Projektteilnehmer unterstützen kann, bleibt offen. Innerhalb des Probandentests kam heraus, dass alle Teilnehmer der Meinung sind, dass das Potenzial des Chatbots auf jeden Fall da ist und dieser sehr viel Unterstützung leisten kann. Dadurch lässt sich auch ein Erfolg verbuchen im Hinblick auf das Ziel der Bachelorarbeit. Demnach ist von einer Bestätigung auf die Frage, ob Chatbots in der Lage sind, Aufgaben von Projektmitgliedern zu optimieren auszugehen. Weiterhin lässt sich ebenfalls vermuten, dass das System zur Moral- und Arbeitssteigerung verschiedener Projektmitglieder aufgrund bereits genannter Aspekte funktionieren kann. Jedoch für ein endgültiges Resultat sowie einen vollständigen Nachweis muss der Prototyp zunächst über einen längeren Zeitraum in einem größeren Projekt bei häufiger Nutzung zum Einsatz kommen.



# Literatur

- Anirudh, Verma / Tyagi Shashikant / Mathur Gauri (2021): A Comprehensive Review on Bot Discord Bot, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, [online] doi: 10.32628/CSEIT2172100
- Balzert, Helmut / Heide Balzert / Rainer Koschke / Uwe Lämmel / Peter Liggesmeyer / Jochen Quante (2009): Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3. Auflage, Spektrum Akademischer Verlag
- Becker, Dr. Nils (2015): User Experience ungleich Usability, Johner Institut, [online] <https://www.johner-institut.de/blog/iec-62366-usability/user-experience/>
- Benet, Lela (2021): Reasons Why People Use Discord, Medium, [online] <https://medium.com/cbblog/reasons-why-people-use-discord-e199591e0a29>
- Bergmann, Rainer / Martin Garrecht (2021): Organisation und Projektmanagement, 3. Auflage, BA Kompakt, [online] doi: 10.1007/978-3-662-63754-8
- Bischoff, Laurenz (2022): Projektmanagement Head der beta 2023, Interview
- Cacho, Jorge Fonseca (2020): Using Discord to Improve Student Communication, Engagement, and Performance, UNLV Best Teaching Practices Expo. 95. [https://digitalscholarship.unlv.edu/btp\\_expo/95](https://digitalscholarship.unlv.edu/btp_expo/95)
- Campbell, Stefan (2022): Discord Statistics 2022: How Many People Use Discord? The Small Business Blog, [online] <https://thesmallbusinessblog.net/discord-statistics/>
- Chatbot | Definition of chatbot in English by Lexico Dictionaries, [online] <https://www.lexico.com/en/definition/chatbot>
- Commit (2022): Wikipedia, [online] <https://de.wikipedia.org/wiki/Commit#:~:text=Commit%20ist%20ein%20Ausdruck%20aus,Programm%2DQuelltext%20in%20Versionsverwaltungssystemen%20verwendet>
- Das Event für Interaktive Unterhaltung, Games Studieren, [online] <https://games-studieren.hs-mittweida.de/beta/>
- Das Team der beta 2022, Games Studieren, [online] <https://games-studieren.hs-mittweida.de/beta/team/>
- Discord (2022): [online] [www.discord.com](http://www.discord.com)
- Frigola, Genís (2021): User-centred design & other frameworks, Medium, [online] <https://medium.com/@genis.hti/user-centred-design-other-frameworks-e89c492a9f6c>
- Introduction to Pipedream, Pipedream, [online] <https://pipedream.com/docs/>
- Koch, Michael / Jürgen Ziegler / Christian Reuter / Thomas Schlegel / Michael Prilla (2020): Mensch-Computer-Interaktion als zentrales Gebiet der Informatik – Bestandsaufnahme, Trends und Herausforderungen, in Informatik Spektrum, Bd. 43, Nr. 6, S. 381-387, [online] doi: 10.1007/s00287-020-01299-8

- Kuster, Jürg / Christian Bachmann / Eugen Huber / Mike Hubmann / Robert Lippmann / Emil Schneider / Patrick Schneider / Urs Witschi / Roger Wüst (2019): Handbuch Projektmanagement, 4. Auflage, Springer Gabler, [online] doi: 10.1007/978-3-662-57878-0
- Kraus, Georg / Reinhold Westermann (2019): Projektmanagement mit System, 6. Auflage, Springer Gabler, [online] doi: 10.1007/978-3-658-24521-4\_1
- Kreutzer, Ralf T. / Marie Sirrenberg (2019): Künstliche Intelligenz verstehen, Springer Gabler, [online] doi: 10.1007/978-3-658-25561-9
- Laymann, Prof. Dr. Frank (o. D.): Repository, Springer Gabler, [online] <https://wirtschaftslexikon.gabler.de/definition/repository-52691>
- Luber, Dipl.-Ing. (FH) Stefan (2017): Was ist eine API? Dev Insider, [online] <https://www.dev-insider.de/was-ist-eine-api-a-583923/>
- Maglogiannis, I. / et al. (2020): An Overview of Chatbot Technology, Springer Nature Switzerland, [online] doi: 10.1007/978-3-030-49186-4\_31
- Matrone, Glorinda (2000): Teamentwicklung im Projektmanagement, Springer Fachmedien Wiesbaden GmbH, [online] doi: 10.1007/978-3-322-99302-1
- Mouawad, Fred (2014): 13 Project Management Terms You Should Know, Taskworld.com
- Novak, Jeannie (2012): Game Development Essentials, 3. Auflage, Delmar Cengage Learning [online]
- Peterjohann, Horst (2021): Die Klassifikation von Anforderungen, Peterjohann Consulting, [online] <https://www.peterjohann-consulting.de/klassifikation-von-anforderungen/>
- Rammer, Christian / Irene Bertschek / Bettina Schuck / Vera Demary / Henry Goecke (2020): Einsatz von Künstlicher Intelligenz in der Deutschen Wirtschaft: Stand der KI-Nutzung im Jahr 2019, ZEW-Gutachten und Forschungsberichte, Bundesministerium für Wirtschaft und Energie, Berlin
- Richter, Michael / Markus D. Flückiger (2016): Usability und UX kompakt, 4. Auflage, Springer Vieweg, [online] doi: 10.1007/978-3-662-49828-6
- Ryte Wiki (2021): MySQL, [online] <https://de.ryte.com/wiki/MySQL#:~:text=Das%20Datenbanksystem%20MySQL%20bietet%20Anwendern,als%20besonders%20ein fach%20zu%20bedienen.>
- Seyhan, Levend (2021): Projektmanagement im Ehrenamt, 2. Auflage, essentials, [online] doi: 10.1007/978-3-658-35036-9
- Stucki, Toni / Sara D'Onofrio / Edy Portmann (2018): Chatbot– Der digitale Helfer im Unternehmen: Praxisbeispiele der Schweizerischen Post, CrossMark, [online] doi: 10.1365/s40702-018-0424-8
- Witt, Matthias M. (2020): Vorteile von Python als Programmiersprache, Datamics, [online] <https://blog.datamics.com/vorteile-von-python-als-programmiersprache-29cf1b0453b>

Workflows and Steps, Pipedream, [online] <https://pipedream.com/docs/workflows/steps/#types-of-steps>

XAMPP (2022): Wikipedia, [online] <https://de.wikipedia.org/wiki/XAMPP>

# Anlagen, Teil 1

Die hier angefügte Umfrage dient als Ausgangspunkt für die Auswertung des Probandentest aus Kapitel 4.3.

The image shows a Google Form titled "beta Bot Umfrage" from Hochschule Mittweida. The header features the university's logo and name. The main text expresses gratitude and explains the purpose of the survey. Below the text, there is a field for the user's email address, which is currently empty. A progress bar indicates the user is on page 1 of 12. A "Weiter" button is visible, along with a link to "Alle Eingaben löschen". At the bottom, there is a warning about passwords and a "Google Formulare" logo.

**HOCHSCHULE  
MITTWEIDA**  
University of Applied Sciences

## beta Bot Umfrage

Vielen Dank, dass Sie sich Zeit nehmen, um diese Umfrage auszufüllen und mich damit bei meiner Bachelorarbeit unterstützen. Ihr Feedback ist sehr wichtig für den weiteren Verlauf der Arbeit und ich bitte daher darum, alle Fragen vollständig sowie ehrlich zu beantworten. Bitte nehmen Sie nur an der Umfrage teil, wenn sie bereits den beta Bot Prototypen ausreichend getestet haben. Die durchschnittliche Dauer dieser Umfrage liegt bei 8 Minuten.

leon.erdhuetter@gmail.com wird nicht geteilt [Konto wechseln](#)

[Weiter](#) 0% Seite 1 von 12 [Alle Eingaben löschen](#)

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 33: Fragebogen - Einstieg, Quelle: Umfrage über Google Formulare**

**Angaben zur Person**

Wie alt sind Sie? \*

Meine Antwort \_\_\_\_\_

Welchen Abschluss streben sie aktuell an? \*

Bachelor

Master

Andere

Gehören Sie zum derzeitigen beta Team oder haben in der Vergangenheit an einer beta teilgenommen? \*

Ja

Nein

Zurück Weiter

Seite 2 von 12

Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 34: Fragebogen – Angaben zur Person, Quelle: Umfrage über Google Formulare**

beta Mittweida

Welchem Team gehörten Sie während ihrer beta Zeit an? \*

GD

Art

Tech

PM

EVM

Andere

Zurück Weiter

Seite 3 von 12

Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 35: Fragebogen – Fragen zur beta Mittweida, Quelle: Umfrage über Google Formulare**

**Fragen zum Thema Projektmanagement**

Waren Sie bereits im Bereich Projektmanagement tätig? \*

Ja

Nein

---

Wie wichtig ist Ihnen ein gutes Projektmanagement? \*

1 2 3 4 5 6 7 8 9 10

Egal           Wichtig

---

Begründen Sie Ihre Antwort

Meine Antwort

---

Zurück Weiter

Seite 4 von 12

Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 36: Fragebogen – Fragen zum Thema Projektmanagement, Quelle: Umfrage über Google Formulare**

**Fragen zum Thema Chatbots**

Sind Sie mit dem Begriff künstliche Intelligenz vertraut? \*

Ja

Nein

---

Hatten Sie schon einmal Kontakt mit künstlich intelligenten Systemen in Form von Chatbots? \*

Ja

Nein

Zurück Weiter

Seite 5 von 12 Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 37: Fragebogen – Fragen zum Thema Chatbots, Quelle: Umfrage über Google Formulare**

**Fragen zum Thema Chatbots**

Sie haben angegeben, dass Sie schon einmal mit Chatbots in Berührung gekommen sind. Wo trat dieser Kontakt auf?

Meine Antwort

---

Zurück Weiter

Seite 6 von 12 Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 38: Fragebogen – Weitere Fragen zum Thema Chatbots, Quelle: Umfrage über Google Formulare**



**Fragen zur Kommunikationsplattform Discord**

In welchem Jahr haben Sie angefangen, Discord zu benutzen? \*

Meine Antwort \_\_\_\_\_

Für welche Zwecke benutzen Sie Discord? \*

Gaming

Unterhaltung

Für Studium, Ausbildung oder Schule

Sozial

Andere

An wie vielen Tagen im Monat nutzen Sie Discord im Durchschnitt? \*

1 - 5 Tage

5 - 10 Tage

10 - 20 Tage

Mehr als 20 Tage

Wie viele Stunden verbringen Sie durchschnittlich am Tag auf Discord? \*

Weniger als 1 Stunde

Weniger als 5 Stunden

Weniger als 10 Stunden

Mehr als 10 Stunden

Haben Sie schon einmal von Discord Chatbots gehört? \*

Ja

Nein

Zurück Weiter

Seite 7 von 12

Alle Eingaben löschen

**Abbildung 39: Fragebogen – Fragen zur Kommunikationsplattform Discord, Quelle: Umfrage über Google Formulare**

**Chatbot Präferenzen**

Wir nehmen an, sie würden einen Discord Chatbot entwickeln, wie wichtig sind Ihnen dabei folgende Kriterien bei der Umsetzung?

**Aussehen \***

1   2   3   4   5   6   7   8   9   10

Egal           Wichtig

**Funktionsumfang \***

1   2   3   4   5   6   7   8   9   10

Egal           Wichtig

**Technische Funktionalität \***

1   2   3   4   5   6   7   8   9   10

Egal           Wichtig

**Übersichtlichkeit \***

1   2   3   4   5   6   7   8   9   10

Egal           Wichtig

**Multilingualität \***

Ist es wichtig, dass der Chatbot mehr als nur eine Sprache zur Kommunikation beherrscht?

1   2   3   4   5   6   7   8   9   10

Egal           Wichtig

**Ausfallrate \***

Ist es wichtig, dass der Chatbot 24/7 Online und somit zu erreichen ist?

1   2   3   4   5   6   7   8   9   10

Egal           Wichtig

Hier können Sie weitere Ihrer Meinung nach wichtige Punkte anfügen.

Meine Antwort

Zurück
WeiterSeite 8 von 12
Alle Eingaben löschen

**Abbildung 40: Fragebogen – Chatbot Präferenzen, Quelle: Umfrage über Google Formulare**

**Prototyp Bewertung**

Bitte bewerten Sie nun den Prototypen innerhalb der nachfolgenden Punkte. Bitte beziehen Sie ihre Bewertung dabei nicht nur auf das beta Card Game.

**Nutzen \***

1 2 3 4 5 6 7 8 9 10

Schlecht            Sehr gut

**Aussehen \***

Finden Sie die Gestaltung der ausgegebenen Nachrichten ansprechend?

1 2 3 4 5 6 7 8 9 10

Schlecht            Sehr gut

**Funktionsumfang \***

1 2 3 4 5 6 7 8 9 10

Schlecht            Sehr gut

**Technische Funktionalität \***

Funktioniert das Interagieren mit dem Chatbot bugfrei?

1 2 3 4 5 6 7 8 9 10

Schlecht            Sehr gut

**Übersichtlichkeit \***

1 2 3 4 5 6 7 8 9 10

Schlecht            Sehr gut

**Bedienbarkeit \***

Verfügt der Chatbot über eine angenehme Bedienung oder hatten Sie Probleme damit?

1 2 3 4 5 6 7 8 9 10

Schlecht            Sehr gut

Hier können Sie gerne weiteres Feedback zum Prototypen dalassen.

Meine Antwort

Zurück
Weiter

Seite 9 von 12

Alle Eingaben  
löschen

**Abbildung 41: Fragebogen – Bewertung des Prototyps, Quelle: Umfrage über Google Formulare**

**Beta card Game Bewertung**

Bitte bewerten Sie nun das beta Card Game innerhalb der nachfolgenden Punkte.

**Spaßfaktor \***

1 2 3 4 5 6 7 8 9 10

Niedrig           Hoch

**Spielgefühl \***

Wie hat sich das Spielen angefühlt?

1 2 3 4 5 6 7 8 9 10

Schlecht           Sehr gut

**Erklärung des Spiels \***

Wurden Ihnen das Spiel sowie die Regeln klar geschildert?

1 2 3 4 5 6 7 8 9 10

Schlecht           Sehr gut

**Verständnis des Spiels \***

Haben Sie schnell gelernt, wie das Spiel funktioniert?

1 2 3 4 5 6 7 8 9 10

Schlecht           Sehr gut

**Regeln und Preise \***

Waren Sie mit dem Regelwerk sowie der betacoin Preise, z. B. beim Kauf eines Albums zufrieden?

1 2 3 4 5 6 7 8 9 10

Schlecht           Sehr gut

Hier können Sie gerne weiteres Feedback zum beta Card Game dalassen sowie Erweiterungswünsche äußern.

Meine Antwort

Zurück
Weiter

Seite 10 von 12

Alle Eingaben löschen

**Abbildung 42: Fragebogen – Bewertung des beta Card Game, Quelle: Umfrage über Google Formulare**

**Ausblick**

Einige Funktionen des beta Bot, wie z. B. die Bestenlisten und die Berechnung der <sup>\*</sup> Seltenheit einer Karte waren von der TP Leistung der einzelnen Entwickler abhängig. Könnte ein solches System bei Ihnen zu einer Motivations- bzw. Leistungssteigerung für ein solches Projekt wie die beta beitragen?

Ja

Nein

Würden Sie es sinnvoll finden, an dem beta Bot weiter zu entwickeln und ihn zum <sup>\*</sup> vollständigen Einsatz auf dem beta Discord Server zu veröffentlichen?

Ja

Nein

Begründen Sie ihre Entscheidung.

Meine Antwort

Zurück Weiter

Seite 11 von 12

Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

**Abbildung 43: Fragebogen – Ausblick, Quelle: Umfrage über Google Formulare**

**Vielen Dank für die Teilnahme**

Wir sind nun am Ende dieser Umfrage angekommen.

Bitte betätigen Sie abschließend den Button zum Absenden der Daten.

Zurück **Senden**

Seite 12 von 12

Alle Eingaben löschen

Geben Sie niemals Passwörter über Google Formulare weiter.

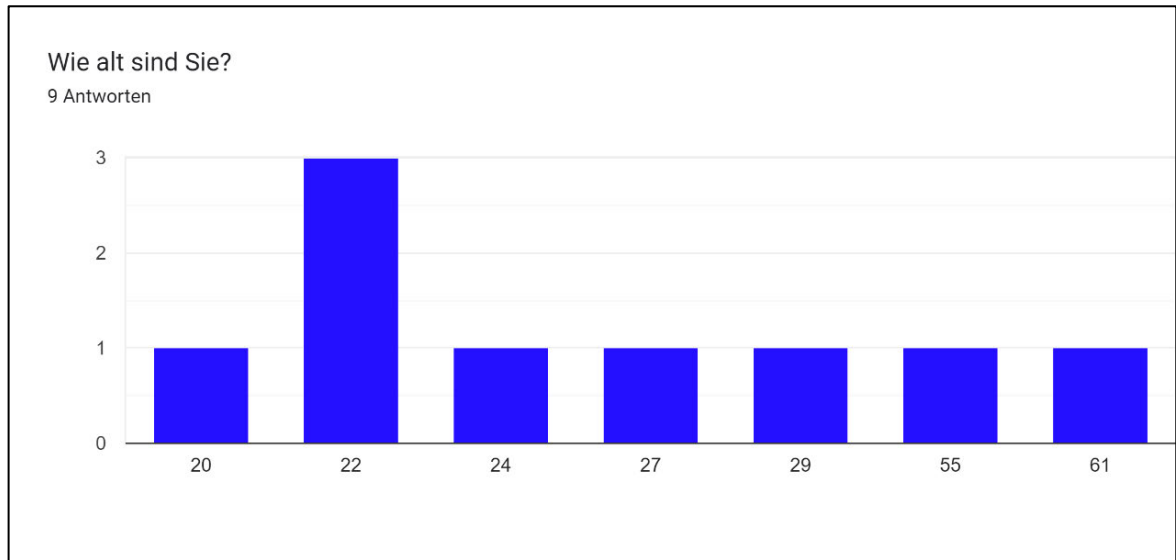
Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

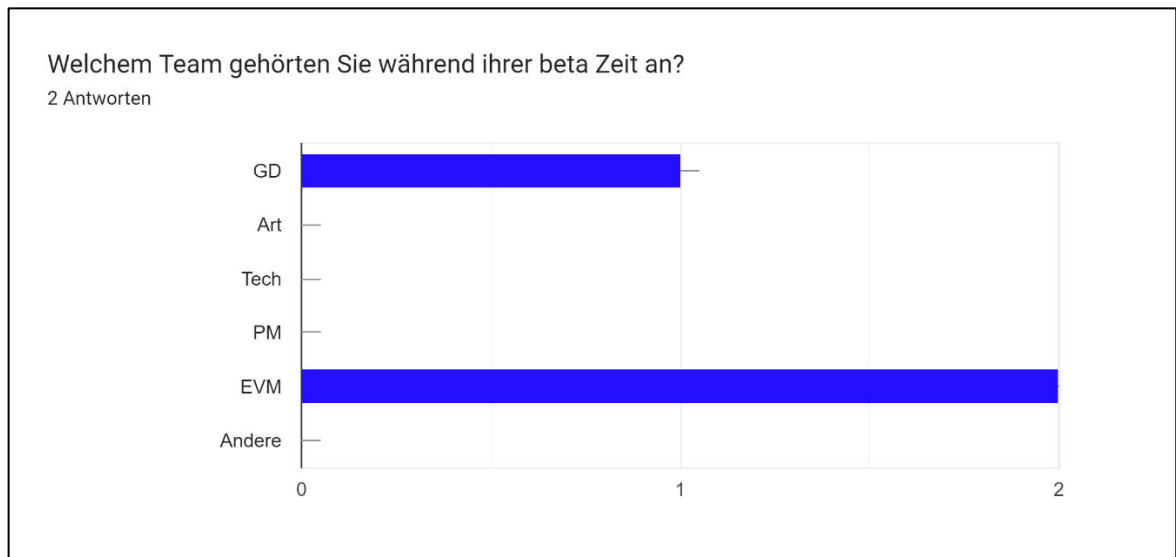
**Abbildung 44: Fragebogen – Abschließende Worte, Quelle: Umfrage über Google Formulare**

## Anlagen, Teil 2

Im Folgenden sind weitere Diagramme zu Kapitel 4.3.3 dargestellt, welche ähnliche oder weniger relevante Ergebnisse im Vergleich zu anderen Fragen aufweisen.



**Abbildung 45: Fragebogen Auswertung – Alter der Probanden, Quelle: Umfrage über Google Formulare**



**Abbildung 46: Fragebogen Auswertung – Teamangehörigkeit während der beta, Quelle: Umfrage über Google Formulare**



**Abbildung 47: Fragebogen Auswertung – Erster Kontakt mit Discord, Quelle: Umfrage über Google Formulare**

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 08. August 2022



Leon Erdhütter