



# **CHATCHARM**

**Burgherr, Studer, Matarese, Kistler**

## Inhaltsverzeichnis

Informieren.....	3
Planen.....	3
Entscheiden .....	4
Realisieren .....	4
Kontrollieren.....	5
Auswerten .....	8

## Informieren

Wir möchten eine mobile Chat-App erstellen, bei der man mit einer Künstlichen Intelligenz kommunizieren kann. Die Künstliche Intelligenz soll dabei von uns selbst programmiert und trainiert werden. Die Chat-App soll dabei so aufgebaut sein, dass jede Person allein einen Chat mit der AI hat. Der Verlauf des Chats soll auf dem Handy gespeichert werden, damit der Nutzer nahtlos in den Chat einsteigen kann, wenn er die App startet.

Wir werden mit React-Native arbeiten, da wir im Modul 335 damit arbeiten. Als Programmiersprachen werden wir daher HTML, CSS und JavaScript verwenden.

Um das Programm während der Entwicklung am Handy testen zu können, verwenden wir Expo. Dies ist eine App, um mittels eines QR-Codes die zu testende Software auf dem Handy temporär zu öffnen.

## Anforderungen

Nr.	Anforderung	Muss/Kann	Funkt/Rand/Qua
1	Der User kann Nachrichten durch ein Textfeld schreiben	Muss	<b>Funktional</b>
2	Die KI antwortet dem User zurück	Muss	<b>Funktional</b>
3	Die Nachrichten werden über den lokalen Speicher gespeichert	Kann	<b>Funktional</b>
4	Nachrichten können vom User gelöscht werden	Kann	<b>Funktional</b>
5	Die Applikation läuft auf dem Handy	Muss	<b>Rand</b>
6	Das React-Native Framework wurde verwendet	Muss	<b>Rand</b>
7	Man kann sich registrieren	Kann	<b>Funktional</b>
8	Man kann sich einloggen	Kann	<b>Funktional</b>
9	Der User kann ein Profilbild über die API festlegen	Kann	Funktional
10	Die App gibt einem Fehlerausgaben und Erfolgsmeldungen	Kann	Funktional

## Planen

Nr.	Arbeitspaket	Dauer	Verantwortlich	Deadline
0	Expo wurde bei allen installiert und funktioniert	2x45	Alle	<b>14.11.23</b>
1	Textfeld Komponente erstellen	2x45	Olivier Kistler	<b>21.11.23</b>
2	Nachrichten Komponente erstellen	3x45	Olivier Kistler	<b>21.11.23</b>
3	KI an die Applikation anbinden	8x45	Randon Studer	<b>28.11.23</b>
4	KI gibt dem User eine Antwort zurück	2x45	Randon Studer	<b>28.11.23</b>
5	Chat Komponente erstellen	2x45	Olivier Kistler	<b>28.11.23</b>
6	Chatnachrichten werden auf den lokalen Speicher gespeichert	3x45	Olivier Kistler	<b>05.12.23</b>

7	Die Nachrichten können gelöscht werden	1x45	Olivier Kistler	05.12.23
8	Chat wird auch aus dem Speicher gelöscht	1x45	Olivier Kistler	05.12.23
9	CSS erstellen	8x45	Olivier Kistler/Samuel Matarese	12.12.23
10	Button Komponente erstellen	3x45	Olivier Kistler	21.11.23
11	Dropdown für Settings	2x45	Olivier Kistler	05.12.23
12	Der User kann sich ausloggen	2x45	Olivier Kistler	05.12.23
13	Die App ist responsiv (Desktop, Handy)	5x45	Olivier Kistler	12.12.23
14	Fehlermeldungen und Erfolgsmeldungen werden angezeigt	2x45	Samuel Matarese	05.12.23
15	Profilbild kann zum User hinzugefügt werden	2x45	Samuel Matarese	5.12.23
16	Profilbild wird in der API gespeichert	2x45	Samuel Matarese	28.11.23
17	Profilbild wird bei jeder Nachricht angezeigt	2x45	Olivier Kistler	5.12.23
18	API für Login mit JWT	8x45	Samuel Matarese	5.12.23
19	Testfälle schreiben	3x45	Samuel Matarese	12.12.23
20	Kontrollieren	2x45	Samuel Matarese	12.12.23
21	Auswerten (Portfolio schreiben)	3x45	Alle	19.12.23
22	Projektdokumentation	5x45	Samuel Matarese	19.12.23
23	KI trainieren		Randon Studer	19.12.23

## Entscheiden

Wir haben uns dazu entschieden, als KI eine selbst erstellte KI von Randon zu wählen, die mit Python geschrieben wurde.

## Realisieren

Unser Projekt befindet sich in einem GitHub Repository:

<https://github.com/Don-Borito/chat-app>

Wenn sie unsere App testen möchten, bedenken sie das der erste API Call ein wenig länger dauern könnte, da die API ihre Zeit braucht um aufzustarten.

# Kontrollieren

## Testfälle

Testnr.	Anforderungsnr.	Vorbereitung	Eingabe	Ausgabe
1.1	1	Applikation wurde gestartet	1.Klick auf das Inputfeld 2.Eingabe von „Hallo“ 3.Auf „send“ klicken	4.„Hallo“ wird mit grünem Hintergrund angezeigt.
1.2	1	Applikation wurde gestartet	1.Klick auf das Inputfeld 2.Klick auf „send“	3.Keine Ausgabe
2.3	2	Applikation wurde gestartet	1.Klick auf das Inputfeld 2.Eingabe von „Hallo“ 3.Auf „send“ klicken	4.„Hallo“ wird mit grünem Hintergrund angezeigt. 5.Antwort von KI wird angezeigt
3.4	3	Eine Nachricht wurde vom User erstellt.	1.App neu laden	2.Nachricht ist immer noch oben zu sehen
4.5	4	2 Nachrichten wurden vom User erstellt	1.Klick auf Zahnradsymbol 2.Klick auf „Delete Chat“	3.Nachrichten sind nicht mehr zu sehen
4.6	4	2 Nachrichten wurden vom User erstellt	1.Klick auf Zahnradsymbol 2.Klick auf „Delete Chat“ 4.App Reload	3.Nachrichten sind nicht mehr zu sehen 5.Nachrichten sind nicht mehr zu sehen
5.7	5	Visual Studio Code offen haben mit Terminal	1. «npx expo start» ins Terminal eingeben 3.Code mit Handy scannen	2.QR-Code wird angezeigt 4.App wird angezeigt
7.8	7+10	App wurde gestartet	1.Klick auf Zahnrad 2.Klick auf «Register» 4.<Test > in Name Inputfeld eingeben 5.<test > in Passwortfeld eingeben	3. «Register» Modal wird angezeigt 6. «Registration succeeded» wird als Erfolgsausgabe abgegeben
7.9	7+10	App wurde gestartet	1.Klick auf Zahnrad 2.Klick auf «Register» 4.Klick auf «Register»	3. «Register» Modal wird angezeigt 5. <Fehlerausgabe> wird angezeigt
8.10	8+10	Registrierung in 7.8 wurde gemacht	1.Klick auf Zahnrad 2.Klick auf «Register» 4. «already have an account?» anklicken 6.<Test > in Name Inputfeld eingeben 7.<test > in Passwortfeld eingeben 8.«Login» anklicken	3. «Register» Modal wird angezeigt 5. «Login» Modal wird angezeigt 9. «Login succeeded» wird als Erfolgsausgabe abgegeben

8.11	8+10	App wurde gestartet	1.Klick auf Zahnrad 2.Klick auf «Register» 4. «already have an account?» anklicken 6.«Login» anklicken	3. «Register» Modal wird angezeigt 5. «Login» Modal wird angezeigt 7.<Fehlermeldung> wird angezeigt
8.12	8+10	App wurde gestartet	1.Klick auf Zahnrad 2.Klick auf «Register» 4. «already have an account?» anklicken 6.<falsche Daten > in Name Inputfeld eingeben 7.<falsche Daten > in Passwortfeld eingeben	3. «Register» Modal wird angezeigt 5. «Login» Modal wird angezeigt 8.<Fehlerausgabe> wird angezeigt
9.13	9	Der User ist angemeldet (Test 8.10)	1.Klick auf Zahnrad 2.Klick auf «Select profile Image» 4.Klick auf «Open Gallery» 6.<beliebiges Foto> wählen 8. «Select this Image» klicken	3. «Upload a Profile Image» Modal wird angezeigt 5. <Fotos auf dem Iphone> werden angezeigt 7.Bild wird im Modal angezeigt 9. <Erfolgsausgabe> wird angezeigt
9.14	9	Der User hat ein Bild erfolgreich hochgeladen (Test 9.13) und die App neu geladen.	1.Klick auf Inputfeld 2. «Hallo» eingeben 3. «send» klicken	4. «Hallo» und festgelegtes Profilbild wird angezeigt

## Testprotokoll

Testnr	Status	Tester	Datum
1.1	OK	Samuel Matarese	12.12.2023
1.2	OK	Samuel Matarese	12.12.2023
2.3	NOK	Samuel Matarese	19.12.2023
3.4	OK	Samuel Matarese	12.12.2023
4.5	OK	Samuel Matarese	12.12.2023
4.6	OK	Samuel Matarese	12.12.2023
5.7	OK	Samuel Matarese	12.12.2023
7.8	OK	Samuel Matarese	12.12.2023
7.9	OK	Samuel Matarese	12.12.2023
8.10	OK	Samuel Matarese	12.12.2023
8.11	OK	Samuel Matarese	12.12.2023
8.12	OK	Samuel Matarese	12.12.2023
9.13	OK	Samuel Matarese	12.12.2023
9.14	OK	Samuel Matarese	12.12.2023

## Testprotokoll

Alle Testfälle ausser Testfall 2.3 haben funktioniert. Die KI konnte nicht an das Frontend angebunden werden, da es Schwierigkeiten mit dem Image gab. Das bedeutet, dass unsere App so noch nicht ausgeliefert werden kann.

## Auswerten

Die Auswertung befindet sich individuell auf den verschiedenen Portfolios