



Projektbericht

Softwaredesign (MECH-B-3-SWD-SWD-ILV)

Bachelor - Mechatronik, Design und Innovation

3. Semester

Dozent: DR. RER. POL. JULIAN HUBER

Gruppe: BA-MECH-22-2B

Autor: Slyvka, Reimeir

1. März 2024

Inhaltsverzeichnis

1	Einleitung	1
2	Installationsanleitung	1
3	Programmfunktion u. Erweiterungen	1
3.1	UML-Diagramm	2
3.2	Recognize über Mikrofon	3
3.3	Erstellen eines Histogramms	3
3.4	Waveform des Musikstücks	3
3.5	Ausgabe eines YouTube Links	4
3.6	Meta-Daten	4
	Abbildungsverzeichnis	III
	Tabellenverzeichnis	IV
	Literaturverzeichnis	V

1 Einleitung

Für das Abschlussprojekt in Softwaredesign mussten wir eine Musikerkennungssoftware mit Web-UI entwickeln. Über das UI können neue Songs angelernt werden, welche dann, mit einem kurzen Abschnitt des Lieds, identifiziert werden können. Als Hilfestellung durften wir ein GitHub Repository namens `abracadabra[1]` verwenden.

2 Installationsanleitung

Um die von uns erstellte Software Musikerkennung 3000 zu verwenden muss folgendes durchgeführt werden:

- Das Git-Repository muss kopiert werden. Das kann über den Link <https://github.com/slyvka800/Musikerkennung-3000.git> geklont werden.
- Damit die Software funktioniert müssen die benötigten Packages installiert werden. Die verwendeten Pakete können mit dem Befehl `pip install -r requirements.txt` installiert werden. Diese Datei befindet sich in dem Git Repository.
- Mit dem Befehl `streamlit run User.Interface.py` wird das Web-UI im Browser gestartet. Diese Datei befindet sich auf folgendem Pfad in dem Repository:
`\build\lib\abracadabra\classes`

3 Programmfunktion u. Erweiterungen

Die Minimalanforderungen für die Software sind, dass man über das Web-UI Musikstücke einlernen kann und diese dann identifiziert werden können. Die hochgeladenen Songs müssen im Browser abgespielt werden können. In Abbildung 3.1 ist die Startseite des User Interface zu sehen. Über den Reiter auf der linken Seite kann man auswählen ob man ein Musikstück anlernen oder erkennen will. Wenn ein Musikstück angelernt wird, wird die Funktion, welche für das Fingerprinting zuständig ist, aufgerufen. Es muss eingegeben werden wie der Song heißt, von wem er ist und aus welchem Album. Anschließend werden die Daten in der Datenbank, welche mit `tinydb` realisiert wurde, gespeichert. Beim Erkennen eines Lieds wird die Funktion `Recognize` aufgerufen. In dieser Funktion wird wiederum über Fingerprinting verglichen ob das Musikstück in der Datenbank vorhanden ist. Wenn dies der Fall ist, wird der erkannte Song ausgegeben.

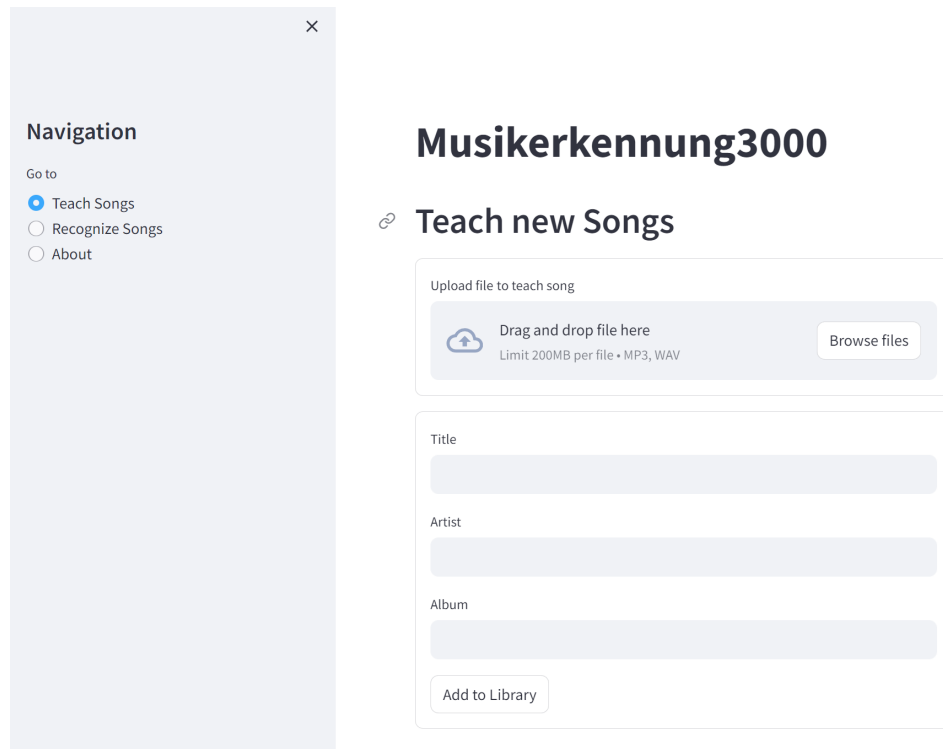


Abbildung 3.1: Startseite Musikererkennung 3000

3.1 UML-DIAGRAMM

In der Abbildung 3.2 wird die Programmarchitektur anhand eines UML-Diagramms veranschaulicht.

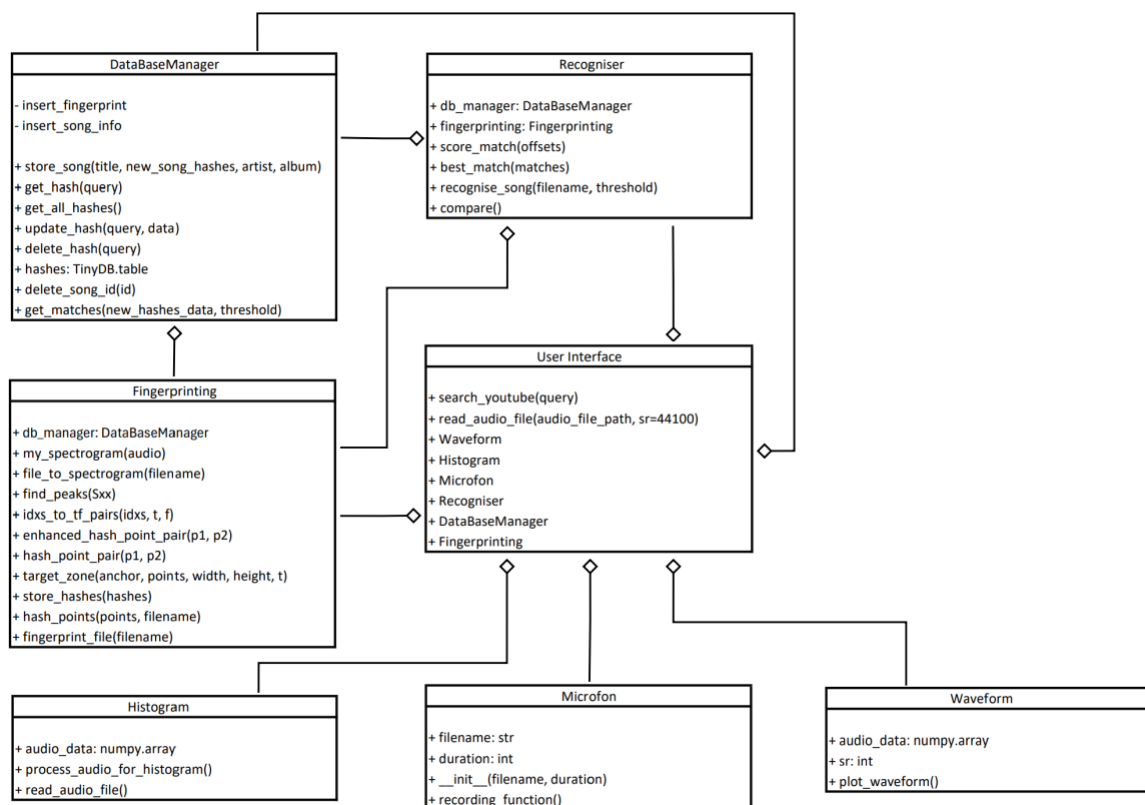


Abbildung 3.2: UML-Diagramm

3.2 RECOGNIZE ÜBER MIKROFON

Eine unserer Erweiterungen ist, dass Musikstücke auch über das Mikrofon des PCs erkannt werden können. Dies kann über den Button Start Recording durchgeführt werden.

3.3 ERSTELLEN EINES HISTOGRAMMS

In Abbildung 3.3 ist eine weitere Erweiterung zu sehen. Es wird abgebildet, welche Frequenz und welche Amplitude die angelernten Lieder haben.

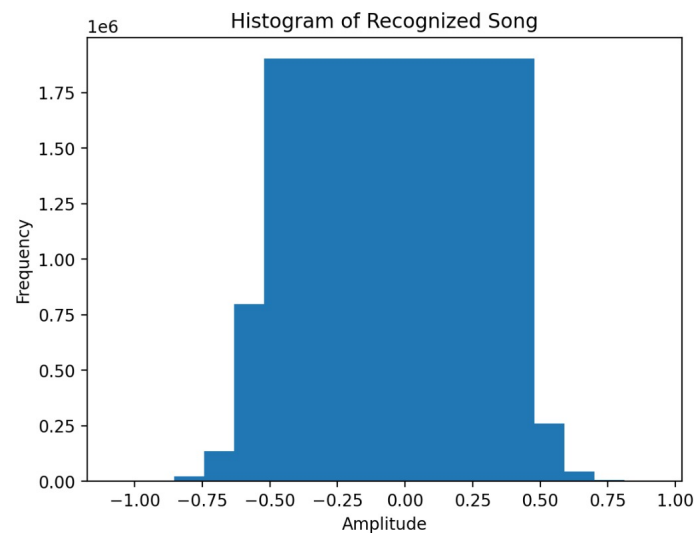


Abbildung 3.3: Histogramm

3.4 WAVEFORM DES MUSIKSTÜCKS

In der Web-UI Anwendung wird ein Plot, wie in Abbildung 3.4 zu sehen ist, des Musikstücks ausgegeben.

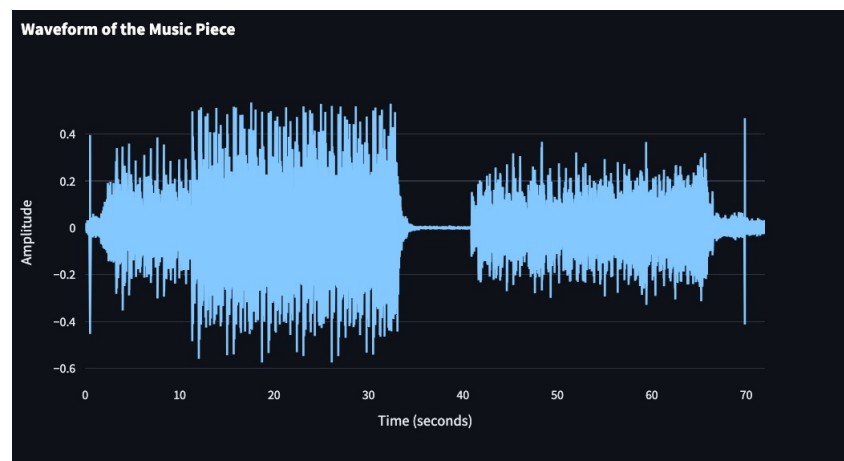


Abbildung 3.4: Waveform

3.5 AUSGABE EINES YOUTUBE LINKS

Wenn die Software ein Musikstück erkennt, wird im User Interface ein YouTube Link zum jeweiligen Song bereitgestellt.

3.6 META-DATEN

Wenn ein Song erkannt wird, wird im User Interface angezeigt von welchem Künstler das Musikstück ist, der Name des Musikstücks und aus welchem Album.

Abbildungsverzeichnis

3.1	Startseite Musikererkennung 3000	2
3.2	UML-Diagramm	2
3.3	Histogramm	3
3.4	Waveform	3

Tabellenverzeichnis

Literaturverzeichnis

- [1] Cameron MacLeod, “Abracadabra (<https://github.com/notexactlyawe/abracadabra>),” GitHub.