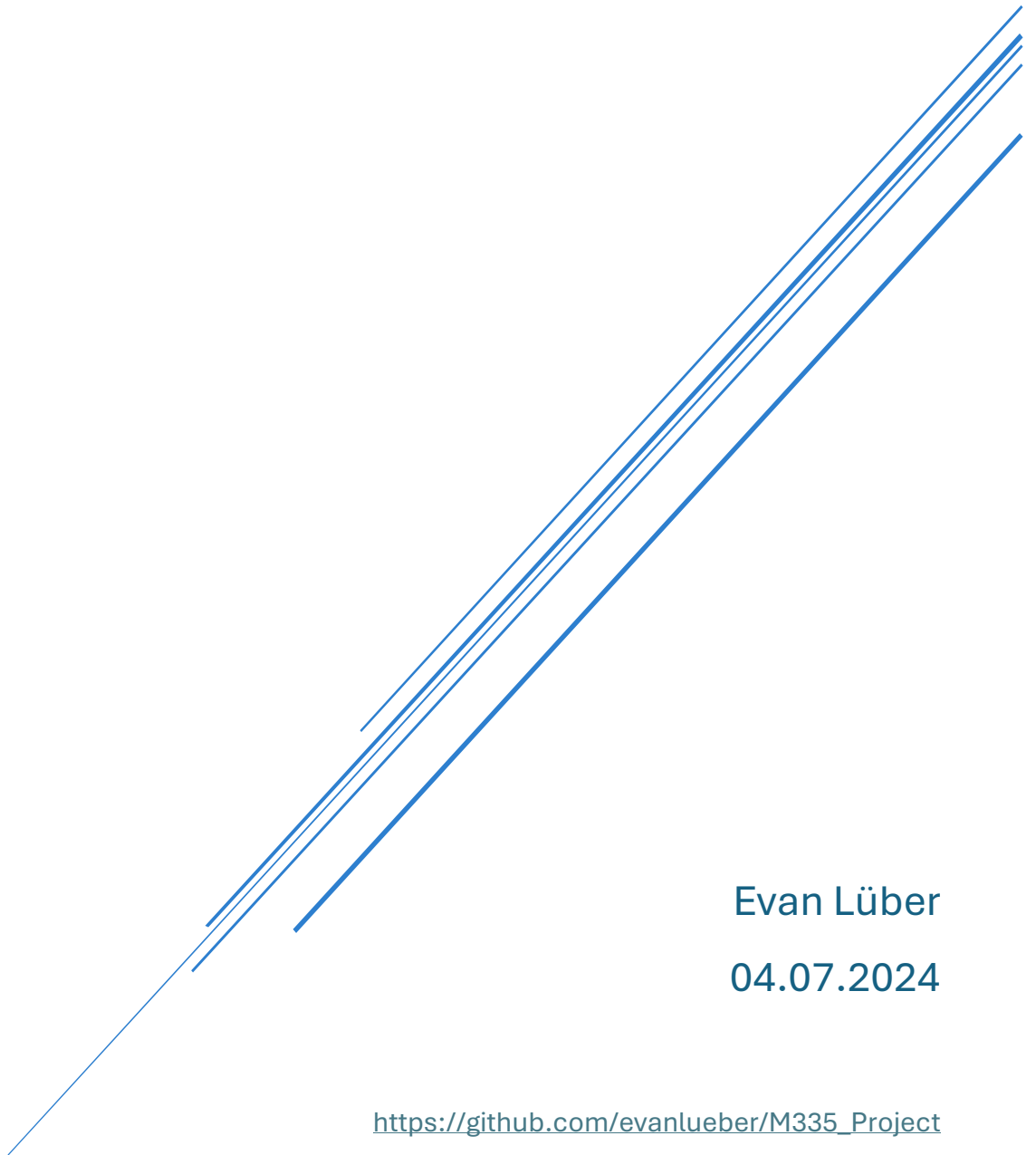


# DOKUMENTATION

M335 LB-B



Evan Lüber

04.07.2024

[https://github.com/evanlueber/M335\\_Project](https://github.com/evanlueber/M335_Project)

## Inhalt

1	Kurzbeschreibung.....	2
2	Funktionale Anforderungen .....	2
3	Nicht Funktionale Anforderungen .....	2
4	Anwendungsfalldiagramme.....	3
4.1	Audio aufnehmen .....	3
4.2	Audio löschen.....	3
4.3	Audio anhören .....	4
4.4	Text schreiben und abspielen .....	4
5	Testkonzept .....	5
5.1	Testumgebung .....	5
5.2	Testmethode.....	5
5.3	Testfälle .....	5
6	Layer-Diagramm .....	6
7	Komponenten-Diagramm.....	7
8	Quellenverzeichnis .....	8
8.1	Hilfsquellen Code .....	8
8.2	Abbildungen .....	8

# 1 Kurzbeschreibung

yApp ist eine App, die den Zweck der Audioaufnahme und der Textvorlesung erfüllt. Mit yAPP es ist möglich seine Gedanken per Tonaufnahme festzuhalten. Diese Audios werden in der App gespeichert und können angehört werden. Wenn die Aufnahme nicht mehr benötigt wird, kann sie ganz einfach gelöscht werden. Ausserdem enthält yAPP eine Funktion, mit der man sich einen Text vorlesen lassen kann, denn man zuvor in ein Textfeld eingegeben hat.

## 2 Funktionale Anforderungen

- Als Sensor wird der Beschleunigungssensor verwendet. Er soll erkennen, wann das Handy horizontal oder vertikal ist, um die App auf dem Handy mitzudrehen.
- Als Aktuator wird der Lautsprecher eingesetzt. Er ist dazu da, die Aufnahme oder einen geschriebenen Text in Sprache wiederzugeben.
- Als externe Schnittstelle wird eine Text-To-Speech Package verwendet, um den geschriebenen Text des Benutzers in Sprache umzuwandeln.
- Damit die Aufnahmen nach dem Schliessen der App nicht verschwinden, sondern beim wieder Öffnen der App noch vorhanden sind, wird ein lokaler Speicher verwendet.
- Die Aufnahme, die Liste der Aufnahmen und die Text-To-Speech Seite wird je in einem eigenen Tab zu finden sein.
- Es besteht eine Interaktion mit der Systemapplikation, beim Verwenden des Mikrofons für das Ausnehmen eines Audios.
- Für das App-Icon wird ein Bild verwendet, welches von GPT-4 generiert wurde.

## 3 Nicht Funktionale Anforderungen

- App soll keine Bugs haben
- App soll auf IOS und Android funktionieren
- App soll einfach zu bedienen sein

## 4 Anwendungsfalldiagramme

### 4.1 Audio aufnehmen

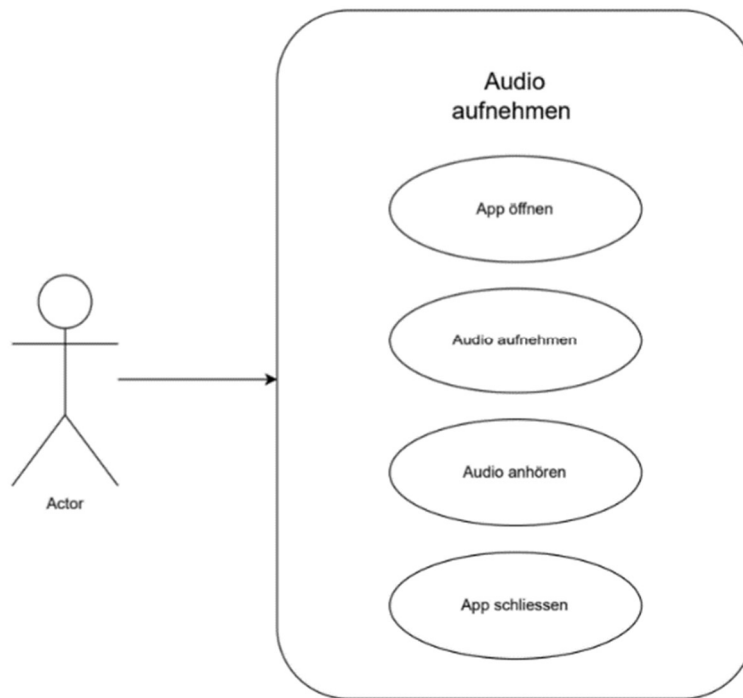


Abbildung 1: Audio aufnehmen

### 4.2 Audio löschen

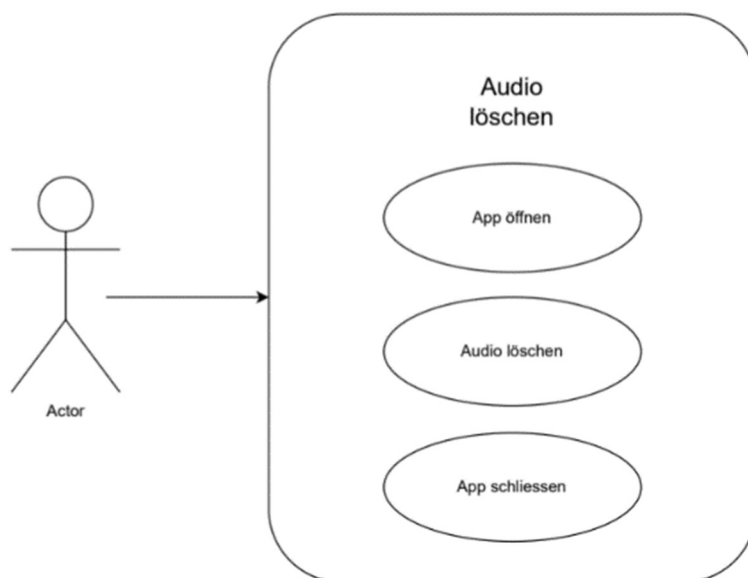


Abbildung 2: Audio löschen

### 4.3 Audio anhören

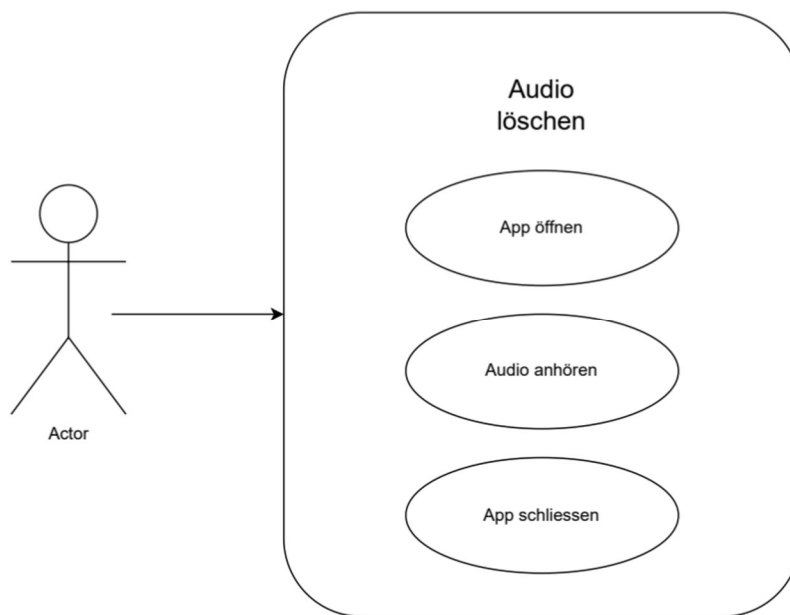


Abbildung 3: Audio anhören

### 4.4 Text schreiben und abspielen

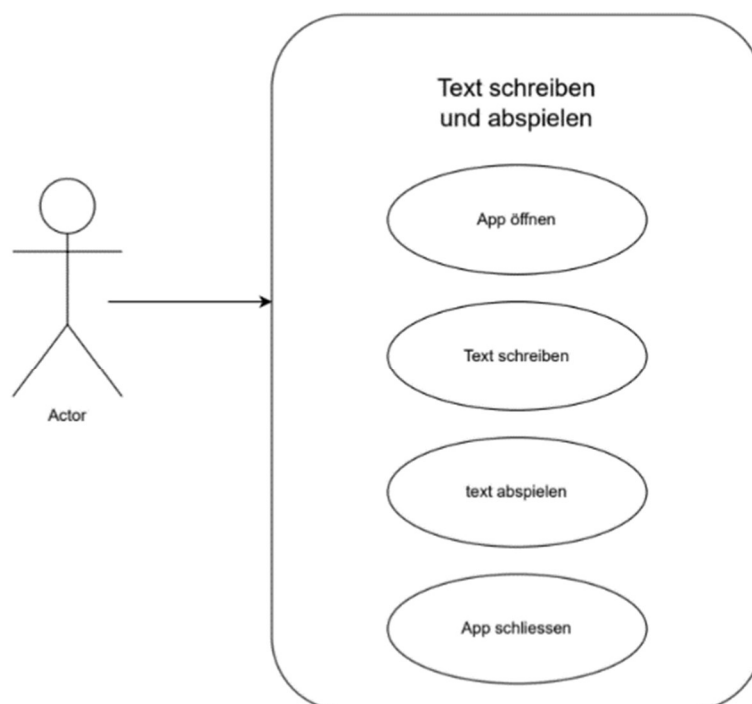


Abbildung 4: Text schreiben und abspielen

## 5 Testkonzept

### 5.1 Testumgebung

- iPhone 11, IOS 16.6.1
- iPhone 15, IOS 18.0 BETA
- Samsung A53,
  - One UI version 6.1,
  - Android version 14

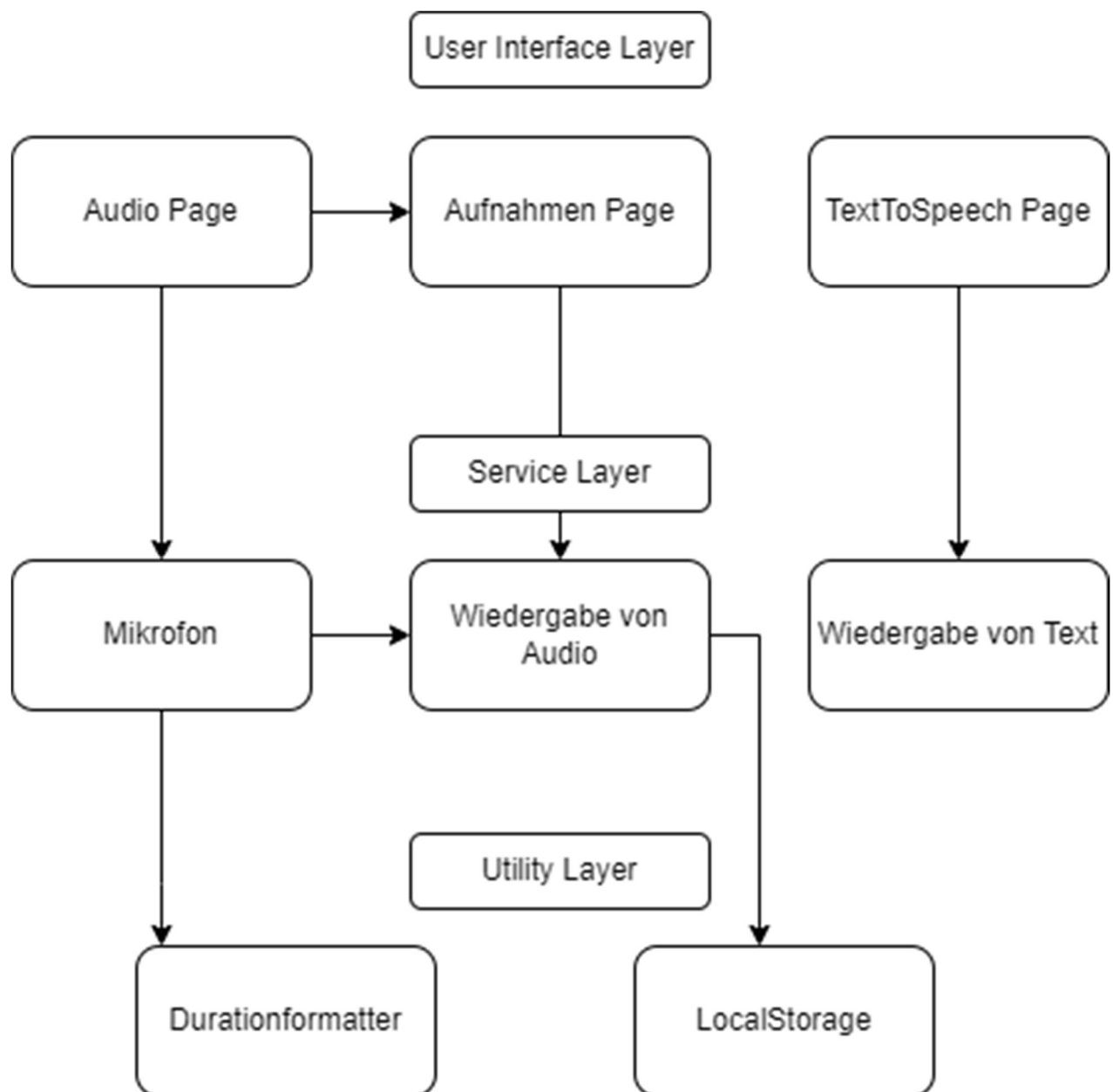
### 5.2 Testmethode

- Manuelle Tests: Testen durch Benutzer, um Benutzerfreundlichkeit zu gewährleisten.
- White-Box: Testen der Logik und Funktion der App
- Black-Box: Funktionstest der Benutzeroberfläche

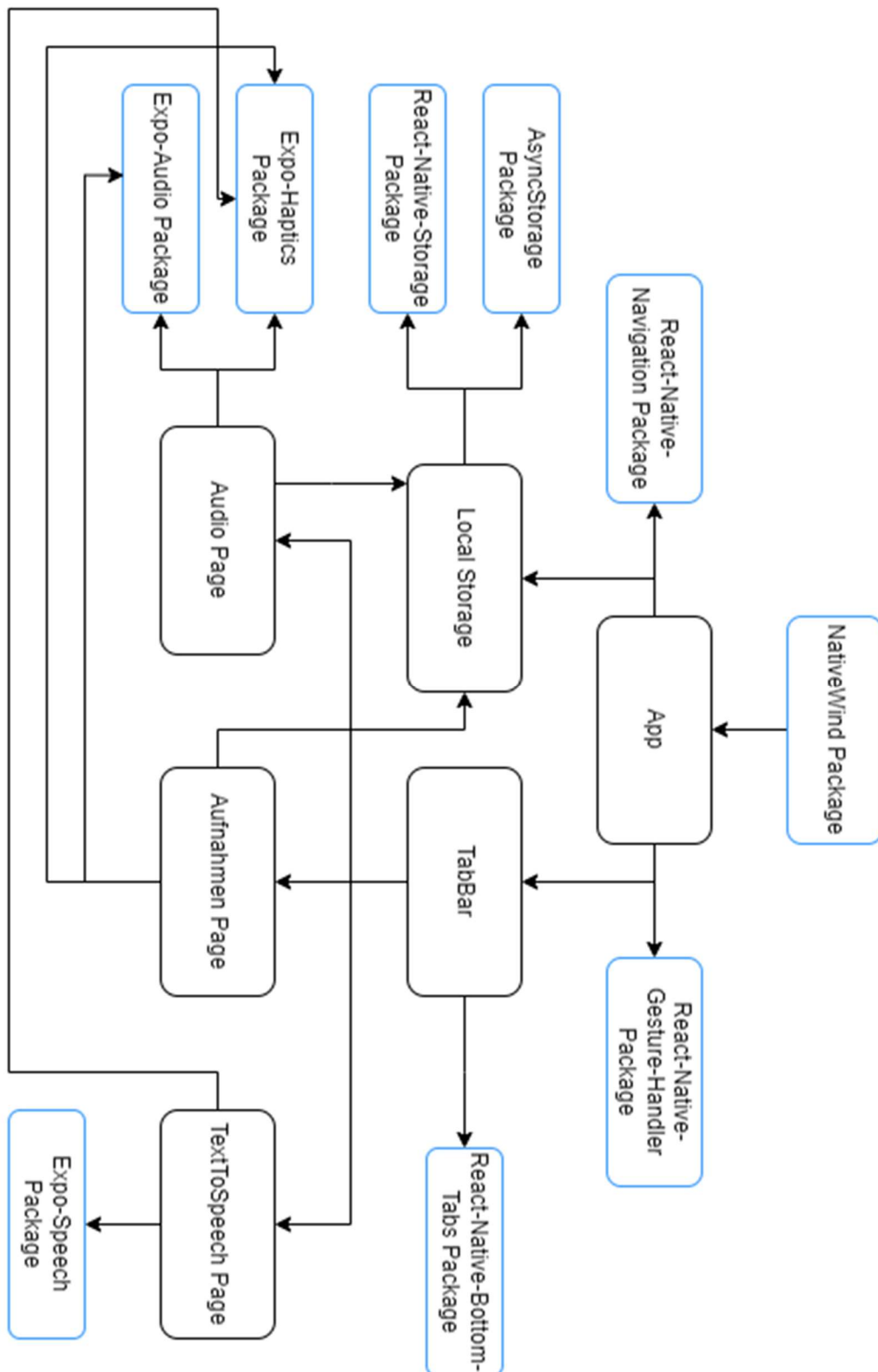
### 5.3 Testfälle

- Checken von Sound beim Starten der Aufnahme
- Testen der Wiedergabe-Funktion
- Testen der Lösch-Funktion
- Testen der Text-To-Speech Funktion
- Abrufen des lokalen Speichers
- Benutzeroberfläche auf unterschiedlichen Betriebssystemen

## 6 Layer-Diagramm



## 7 Komponenten-Diagramm





## 8 Quellenverzeichnis

### 8.1 Hilfsquellen Code

React Native Dokumentation, <https://reactnative.dev/docs/getting-started>

Expo Dokumentation, <https://docs.expo.dev/>

freeDomCamp, <https://www.freecodecamp.org/news/tailwindcss-in-react-native-expo/>

Github, Benutzer: sunnylqm, <https://github.com/sunnylqm/react-native-storage>

Github Copilot, <https://github.com/features/copilot>

### 8.2 Abbildungen

Abbildung 1: Audio aufnehmen .....	3
Abbildung 2: Audio löschen .....	3
Abbildung 3: Audio anhören .....	4
Abbildung 4: Text schreiben und abspielen .....	4