

Produkt-Vision



PSIT4

Semanur Cerkez, Vincent Schmid, Mahmoud Sino,
Patric (Ryan) Steiger, Gabriele Pace, Jan-Erik Walker,
Gabriel Wicki, Nicola Meier

9.3.2020

Version 1.0

Inhaltsverzeichnis

1. Kontext	2
2. Ziele und Hauptfunktionen	4
Hauptfunktionen	4
Minimum Viable Product	4
Track-Page.....	4
Versions-Page.....	4
Version abspielen.....	4
Visualisierung der Waveform.....	4
3. Qualitätsattribute.....	5
Performance	5
Verfügbarkeit	5
Sicherheit	5
Internationalisierung.....	5
Rechtliche Anforderungen	5
4. Bekannte Beschränkungen	5
5. Verwendete Prinzipien.....	6
6. Architektur.....	6
Domänenmodell	6
7. Entscheidungs-Logbuch.....	7
8. Glossar	7
9. Tabellenverzeichnis	7
10. Abbildungsverzeichnis.....	7

Gender-Klausel

Aus Gründen der Lesbarkeit wird die männliche Sprachform verwendet, die gleichzeitig für die weibliche Form gilt.

1. Kontext

Heutzutage ist es einfacher denn je, Musik aufzunehmen und zu produzieren. Sei dies mit Instrumenten, mit Gesang oder auch nur am Computer. Viele Personen gründen hierfür eine Musikgruppe oder starten eine Solo-Karriere. Dies sowohl im professionellen wie auch im Amateurbereich. In fast allen Fällen sind dabei aber mehrere Leute in die Entwicklung involviert. Dies erfordert eine strukturierte Herangehensweise, welche in vielen Fällen nicht vorhanden ist. Musikgruppen speichern ihre Songs auf verschiedenen Geräten und bei verschiedenen Personen ab. Oft gehen Versionen oder sogar ganze Tracks und deren Entwicklungsprozess verloren oder man findet ein gewisses Artefakt wie zum Beispiel eine Jam-Session (freie Improvisation) oder ein Feedback erst nach einer langen Suchaktion wieder.

Beim Arbeiten an den verschiedenen Tracks ist es manchmal schwierig den Überblick zu behalten, da die Verbesserungsvorschläge zu den verschiedenen Versionen nicht auffindbar oder nur schwer zuordnungsbar sind und niemand so richtig weiss, wo was ist. Leute behelfen sich oft nur mit simplen Ordnerstrukturen und Gruppen-Chats um dieses Problem zu lösen. Dies endet meist in einem Chaos. Besser wäre es, wenn man als Musikgruppe seine Aufnahmen in einer Applikation strukturiert ablegen, anhören und diskutieren kann.

ReHEARSal ist eine Webapplikation für Musiker zum Verwalten und Diskutieren ihrer Aufnahmen und spricht somit das oben dargelegte Problem an. Es ermöglicht dem Benutzer, einen Account zu erstellen, um mit anderen Benutzern an der eigenen Musik zu arbeiten. Hierfür kann man Gruppen und anschliessend einzelne Studio Sessions erstellen. Pro Gruppe können wir z.B. für jede Probe eine Session erstellen, in welcher wir die verschiedenen Tracks anzeigen und abspielen, sowie neue Tracks oder Versionen hochladen können. Die einzelnen Versionen können einfach untereinander verglichen werden.

ReHEARSal bietet eine Kommentarfunktion, mit der es möglich ist, sowohl zu den einzelnen Tracks wie auch zu einzelnen Versionen Kommentare zu verfassen. Zudem können die Kommentare auch an spezifischen Stellen (genaue Zeit in einer Version eines Tracks) abgelegt und abermals kommentiert werden. Damit der Benutzer schnell seine Arbeit findet, wird ihm eine Übersicht seiner Beiträge bereitgestellt, wo er auch über neue Aufnahmen und Kommentare seiner Gruppen informiert wird. ReHEARSALs Zielkunden sind Musiker, Produzenten, sowie deren Gastmusiker.

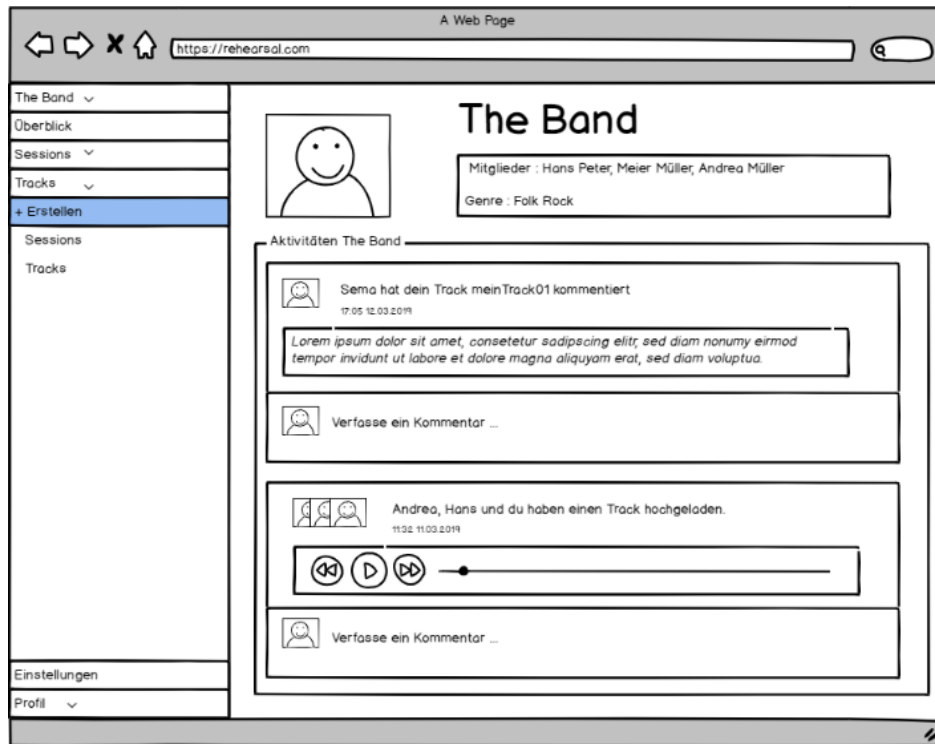


Abbildung 1: Home Page

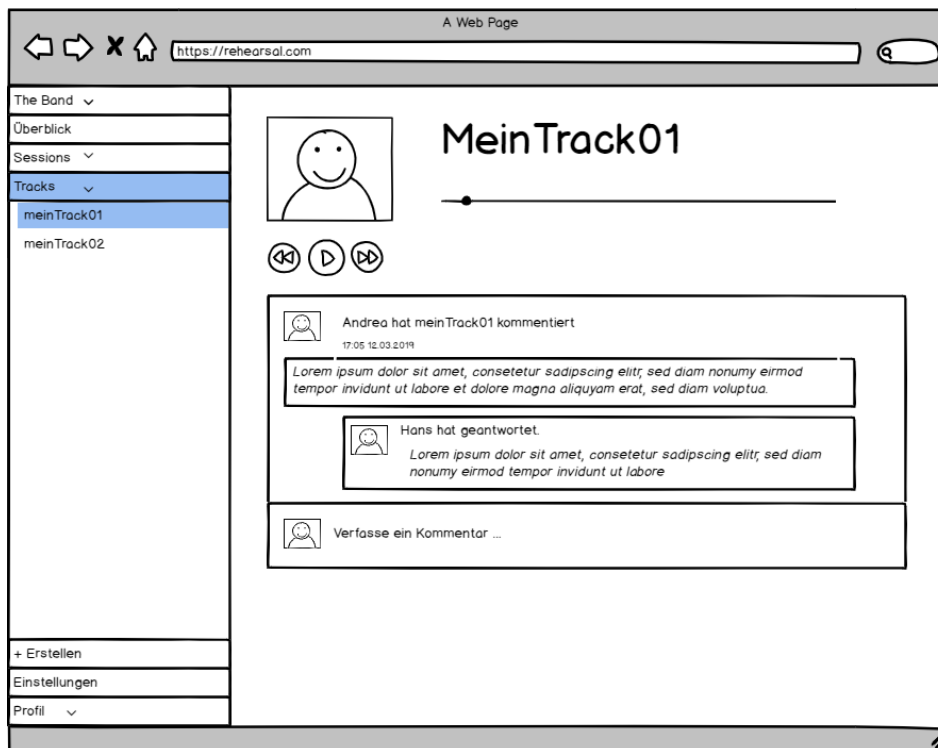


Abbildung 2: Track Page

2. Ziele und Hauptfunktionen

Das Ziel der Applikation ist die Schaffung einer Plattform, auf welcher unterschiedliche Versionen von Tracks kommentiert und abgespielt werden können.

Hauptfunktionen

- Erstellen von Sessions
- Tracks erstellen und Versionen hochladen
- Abspielen von Versionen
- Übersicht der Tracks an welchen man beteiligt ist → Dashboard
- Einordnen von Tracks in Sessions
- Verwalten von Versionen
- Version an einer bestimmten Stelle markieren und dort eine Diskussion starten

Tracks und Kommentare müssen so abgespeichert werden, dass die richtigen Benutzer Zugriff auf die richtigen Daten haben. Neue Tracks werden weiter oben angezeigt.

Minimum Viable Product

Track-Page

In der Track-Page werden die verschiedenen Versionen der Tracks angezeigt.

Über einen Button kann man neue Versionen eines Tracks hinzufügen.

Versions-Page

Über Links kann man auf die Version-Page der verschiedenen Versionen kommen. Auf der Versions-Page werden Kommentare, ein Feld für einen neuen Kommentar und ein Button zum Absenden des Kommentars angezeigt.

Version abspielen

Die hochgeladene Version eines Songs, kann auf der Versions-Page abgespielt werden.

Visualisierung der Waveform

Die Waveform einer Version, wird während dem Abspielen der Version dargestellt.

3. Qualitätsattribute

Nach ISO/IEC 25010 Softwarequalitätsmerkmale abgeleitet:

Performance

- Die Latenz sollte 200ms nicht überschreiten.
- 5 Bands à 5 Mitglieder sollten gleichzeitig Tracks abspielen können, ohne Verzögerungen zu bemerken.
- Jede Band sollte 5 GB Daten auf dem Server abspeichern können.
- Die hochgeladenen Audiodateien werden nicht komprimiert.

Verfügbarkeit

- Der Service kann zum Unterhalt einmal pro Woche für wenige Stunden vom Netz genommen werden.
- Der Services bietet eine sichere Speicherung der Daten der Nutzer.

Sicherheit

- Die Benutzer müssen sich einloggen, um Zugriff auf Ihre Daten zu erlangen.
- Passwörter werden verschlüsselt auf der Datenbank abgelegt.

Internationalisierung

Die Sprache der Applikation ist Englisch.

Rechtliche Anforderungen

Die Tracks der Nutzer werden nicht auf Copyright geprüft und können nur innerhalb der Band geteilt werden.

4. Bekannte Beschränkungen

Das Projekt muss bis Mitte Mai abgeschlossen sein. Dies ergibt fünf zweiwöchige Sprints.

Es ist bereits Quellcode vorhanden, welcher das Grundgerüst vorgibt. Der Code ist in Python verfasst und verwendet das Django Webframework.

Da die meisten Teammitglieder Kenntnisse mit Vue.js haben, ist dieses Framework das Webframework der Wahl. Das Django Backend wird mit dem Frontend über eine REST API-Kommunizieren.

Das Team besteht aus acht Mitgliedern, einige Teammitglieder besitzen bereits Python Kenntnisse andere haben grössere Frontend Erfahrung. Das Resultat ist eine Zusammensetzung welche in der Lage sein sollte das Projekt kompetent umzusetzen.

5. Verwendete Prinzipien

Meist verwendete Prinzipien werden nach GRASP abgeleitet:

- Verwendung von SOLID (Single responsibility principle, Open/Closed principle, Liskov substitutions principle, Interface segregation principle, Dependency Inversion principle).
- DRY (don't repeat yourself).
- Hohe Kohäsion und minimale Kopplung.
- Verwendung von Schichtmodellen.
- Keine Business Logik im Views.
- kein Datenbankzugriff aus Views, Verwendung von ORM.

6. Architektur

Domänenmodell

Eine Gruppe erstellt eine Session. Eine Session kann einen oder mehrere Tracks enthalten. Ein Track selbst besteht aus mindestens einer Version. Jedes Mitglied der Gruppe hat einen Part im Track.

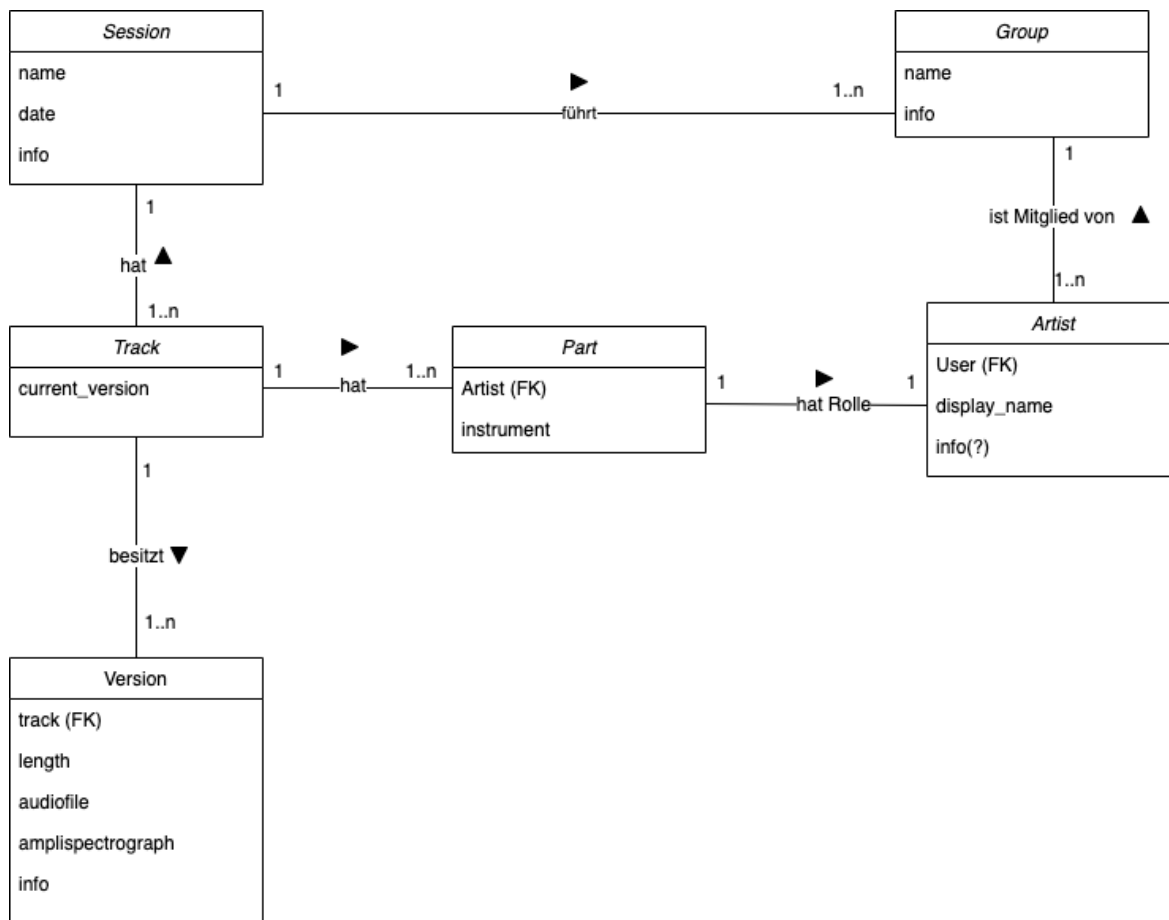


Abbildung 3: Domänenmodell

7. Entscheidungs-Logbuch

Wir haben uns für Vue.js entschieden, da im Team am meisten Kenntnisse mit diesem Framework vorhanden sind. Vue.js ist ein modernes Framework mit verständlicher Syntax.

Python empfiehlt sich für statistische Analysen von Daten und ist eine gut zugängliche Programmiersprache

Django ist ein bewährtes Webframework für Python welches einfach zu verstehen ist, das Framework lässt sich auch gut skalieren.

8. Glossar

Begriff	Beschreibung
Track	Ein Projekt an welchem die ganze Band arbeitet
Version	Der Tracks besteht aus Versionen. Die Versionen können abgespielt und kommentiert werden.
Session	Wenn die Band sich Trifft und Musikstücke aufnimmt
Part	Für jedes Instrument gibt es einen Part. Alle Parts zusammengeführt ergeben den Track.
Waveform	Die Bildliche Darstellung eines Musikstückes, in Form eines Histogramms, welche die Lautstärke pro Zeiteinheit beschreibt.

Tabelle 1: Glossar

9. Tabellenverzeichnis

Tabelle 1: Glossar	7
--------------------------	---

10. Abbildungsverzeichnis

Abbildung 1: Home Page.....	3
Abbildung 2: Track Page.....	3
Abbildung 2: Domänenmodell	6