

Produkt Dokumentation



PSIT4

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

18.5.2020

Version 1.0

Inhaltsverzeichnis

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

Gender-Klausel

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

Kontaktinformationen

Bei Problemen, die nicht mit der Produkt Dokumentation gelöst werden konnten, kontaktieren Sie den Support unter support@rehearsal.ch.

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 zu diesem Zweck eine Musikgruppe oder starten eine Solo-Karriere, sowohl im professionellen wie im Amateurbereich. In fast allen Fällen sind dabei aber mehrere Leute in die Entwicklung der Musikproduktion involviert.

Bei der Arbeit an verschiedenen Musikstücken ist es schwierig den Überblick zu behalten, da Verbesserungsvorschläge zu den verschiedenen Versionen nicht auffindbar oder nur schwer zuzuordnen sind.

reHEARsal ist eine Webapplikation für Musiker, welche die Verwaltung von Musikstücken im Produktionsprozess und die gemeinschaftliche, kreative Zusammenarbeit in Bezug auf Musikproduktion vereinfacht. Sie ermöglicht dem Benutzer, einen Account zu erstellen, um mit anderen Benutzern an der eigenen Musik zu arbeiten. Zum Beispiel kann pro Probe eine Session erstellt werden, in welcher die verschiedenen Tracks angezeigt, deren Versionen abgespielt, neue Tracks hinzugefügt und neue Versionen hochgeladen werden können. Die einzelnen Versionen können einfach untereinander verglichen werden.

Darüber hinaus bietet *reHEARsal* eine Kommentarfunktion, mit welcher es möglich ist, zu einzelnen Versionen Kommentare zu verfassen. Diese können sogar an spezifischen Stellen (genaue Zeit in einer Version eines Songs) abgelegt und abermals kommentiert werden. Damit der Benutzer schnell seine Arbeit findet, wird ihm eine Übersicht mit seinen Beiträgen bereitgestellt, wo er auch über neue Aufnahmen und Kommentare seiner Gruppen informiert wird. *reHEARsals* Zielkunden sind Musiker, Produzenten, sowie deren Gastmusiker.

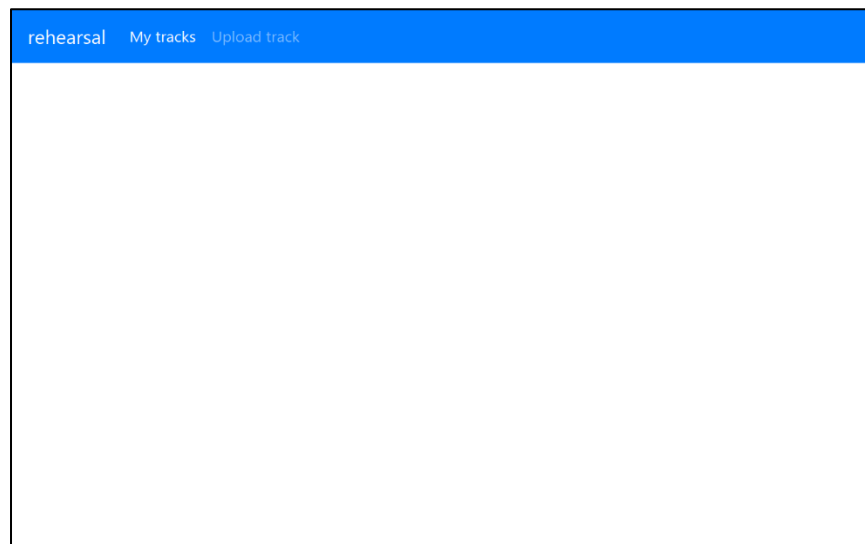


Abbildung 1: Home Page

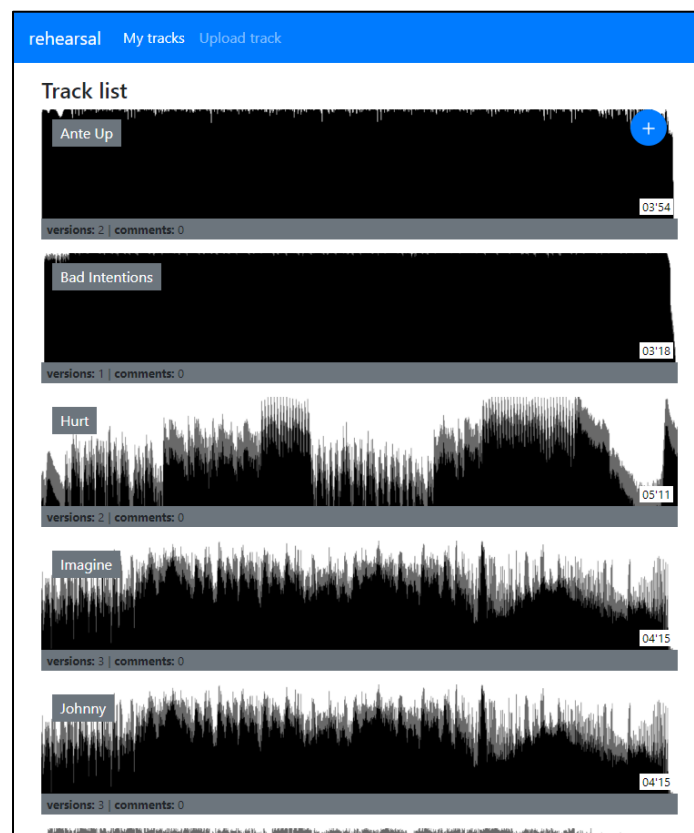


Abbildung 2: Track Page

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 wird 26. Mai 2020 abgeschlossen. Dies ergibt fünf zweiwöchige Sprints, dazu ein einwöchiger sechster Sprint.

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 in den 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.

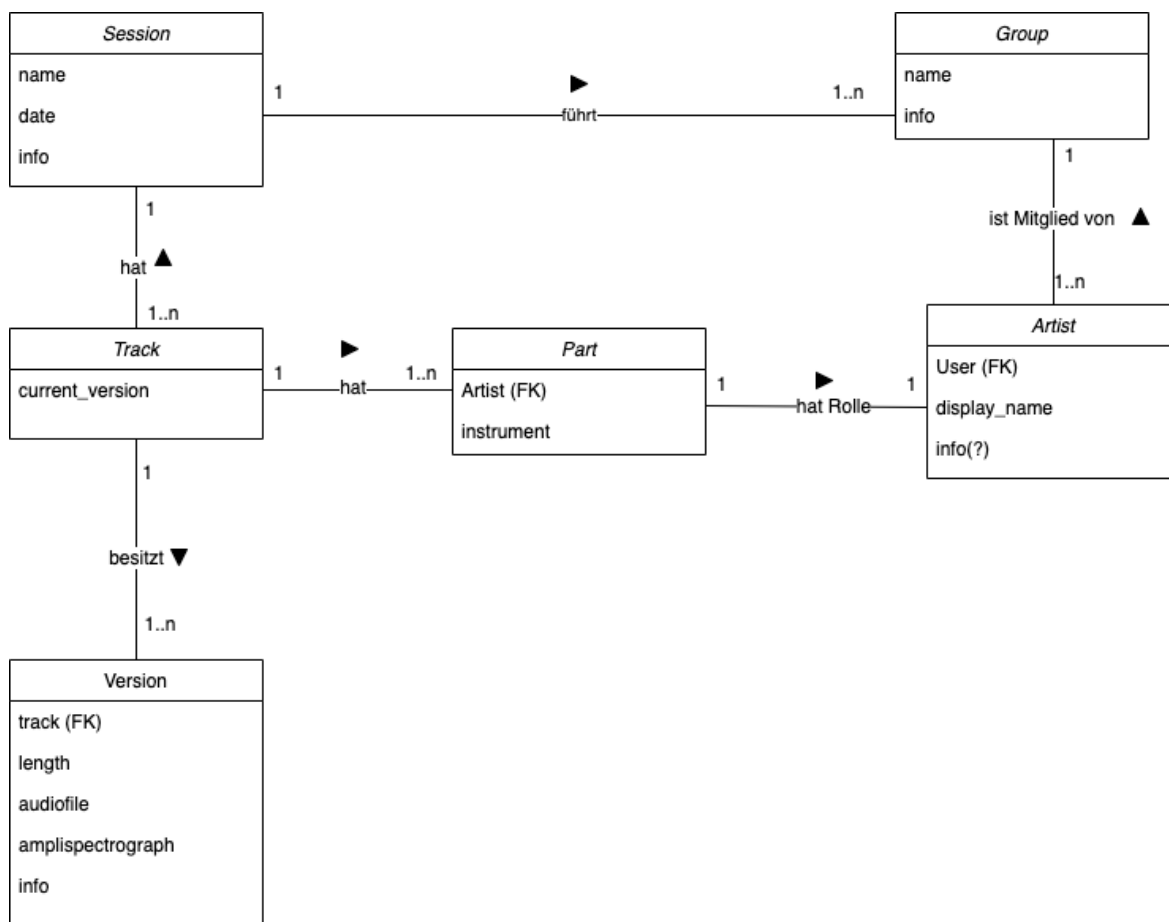


Abbildung 3: Domänenmodell

7. Externe Schnittstellen

Beim Seitenaufruf liefert die Applikation ein kompiliertes HTML-File (mit allem benötigten Frontend-Code) an den Browser. Die Frontendapplikation, welche ab da benutzerinnenseitig im Browser residiert, interagiert mit dem Browser über API-Aufrufe und erhält vom Browser Antworten im JSON-Format.

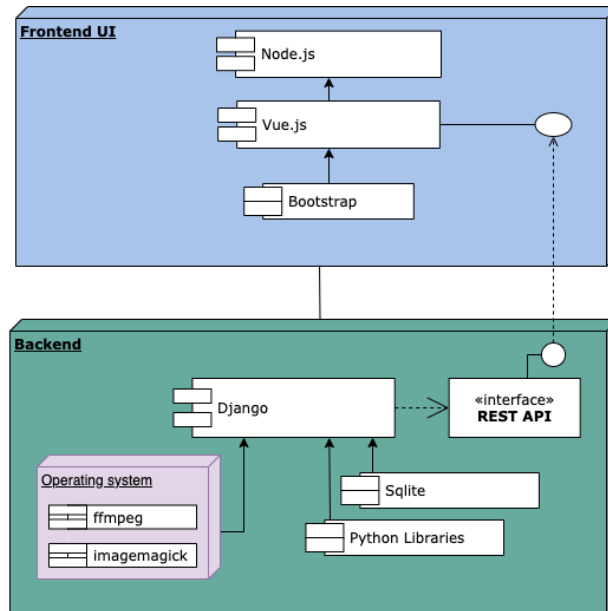


Abbildung 4: Externe Schnittstellen

8. Infrastructure Architecture

Als Infrastruktur benutzen wir den von der ZHAW zur Verfügung gestellten Server auf SWITCH, worauf wir die Applikation laufen lassen. Das Betriebssystem der Infrastruktur ist Ubuntu 18.04 LTS. Als physische Komponente benutzen wir nur unsere Computer und die Hardware, welche von unseren benutzten Tools zur Verfügung gestellt werden.

9. Installation

Um die Software zu nutzen, ist keine Installation notwendig. Auf der Domain rehearsal.ch kann man die Applikation laufen lassen. Falls man die Software lokal starten möchte, kann man auf GitHub den gewünschten Branch in seine Entwicklungsumgebung klonen, um anschliessend Frontend- und Backendserver zu starten. Hierbei kann man die Software auch im Code konfigurieren.

Wir arbeiten mit einem Master Branch, einem Development Branch und diversen Feature Branches. Das Deployment ist automatisiert und findet zu jeder vollen Stunde statt. Der Development Branch wird auf dev.rehearsal.ch verteilt und der Master Branch auf rehearsal.ch. Sobald ein Feature fertig implementiert und getestet ist, wird es auf den Development Branch gemerged. Ist der Development Branch auf einer stabilen Version, wird er in den Master gemerged.

10. Operation und Support

Die Software wird anhand von Tests auf verschiedenen Ebenen überwacht. Wir haben Unit Tests im Frontend und im Backend. Im Frontend benutzen wir End-to-End Tests mit Nightwatch.js (Selenium Webdriver) und im Backend wird Pytest verwendet um die Programmierlogik zu überprüfen. Schlägen diese Tests fehl, wird anschliessend ein Fehler geloggt. Diese Wartung findet manuell statt, sowie die anschliessende Auswertung. Bei allfälligen Störungen, wenden Sie sich an den Support von *reHEARsal* (siehe Kontaktinformation).

11. 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.

12. Glossar

Begriff	Beschreibung
Track	Ein Musikstück -- oder genauer: die Vorstellung, das Ziel eines Musikstücks.
Version	Ein Audiofile, welches versucht die Ideen eines Tracks umzusetzen.
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.
API	Eine limitierte Anzahl an Funktionen, welche eine Software nach aussen freigibt um zu kommunizieren.
Django	CMS Framework (geschrieben in Python)
JavaScript	Kurz: JS, Skriptsprache für dynamisches HTML in Webbrowsern
JSON	Ein Dateiformat um Daten zwischen Anwendungen auszutauschen.
Node.js	Plattform für JavaScript Webanwendungen
Vue.js	Frontend Framework zur Entwicklung von User Interfaces
Python	Eine Programmiersprache
REST API	Eine Definition wie zwei unabhängige Systeme miteinander kommunizieren sollen.
Bootstrap	Layout Framework
TimeComment	Eine Bemerkung, welche eine genaue Stelle einer bestimmten Version kommentiert.
VersionComment	Ein Kommentar, der auf eine bestimmte Version zutrifft.

13. Abbildungsverzeichnis

Abbildung 1: Home Page	4
Abbildung 2: Track Page	4
Abbildung 3: Domänenmodell	7
Abbildung 4: Externe Schittstellen	8