

Fachartikel



reHEARsal

PSIT4

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

Inhaltsverzeichnis

1	Abstract	2
2	Einleitung	3
2.1	Problemstellung	3
2.2	Problemanalyse	3
2.2.1	Referenz auf eine Stelle durch eine Zeitangabe	3
2.2.2	Referenz auf eine Stelle durch musikalische Begriffe	3
2.2.3	Referenz durch absolute Taktangabe	4
2.3	Use-Case-Analyse	4
2.3.1	Workflow	4
2.4	Lösungsansatz	6
2.4.1	Technologie	6
3	Resultate	7
3.1	Die Applikation aus Nutzerinnensicht	7
3.2	Systemarchitektur	8
3.2.1	Frontend	8
3.2.2	Backend	9
3.2.3	Interaktion	9
4	Diskussionen und Ausblick	10
4.1	Visuelle Orientierung durch Wellenform	10
4.2	Kommentare als Threads	10
4.3	Authentisierung oder anonyme Gruppen?	11
5	Literaturverzeichnis	12
6	Abbildungsverzeichnis	12
7	Versionierung	12
8	Glossar	13

Gender-Klausel

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

Kontaktinformationen

Gabriel Wicki	Product Owner
Mahmoud Sino	Scrum Master
Semanur Cerkez	Mitarbeiterin
Nicola Meier	Mitarbeiter
Gabriele Pace	Mitarbeiter
Vincent Schmid	Mitarbeiter
Patric (Ryan) Steiger	Mitarbeiter
Jan-Erik Walker	Mitarbeiter

wickigab@students.zhaw.ch
sinomah1@students.zhaw.ch
cerkesem@students.zhaw.ch
meierni9@students.zhaw.ch
pacegab1@students.zhaw.ch
schmivin@students.zhaw.ch
steigpa1@students.zhaw.ch
walkejan@students.zhaw.ch

1 Abstract

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 Musikerinnen, welche die Verwaltung von Musikstücken im Produktionsprozess und die gemeinschaftliche, kreative Zusammenarbeit im Bezug auf Musikproduktion vereinfacht. Sie ermöglicht der Benutzerin, einen Account zu erstellen, um mit anderen Benutzerinnen 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 die Benutzerin schnell ihre Arbeit findet, wird ihr eine Übersicht eigener Beiträge bereitgestellt, wo sie auch über neue Aufnahmen und Kommentare ihrer Gruppen informiert wird. *reHEARsal's* Zielkundinnen sind Musikerinnen, Produzentinnen, sowie deren Gastmusikerinnen.

2 Einleitung

2.1 Problemstellung

Die aktuelle COVID-19 Pandemie mit dem globalen Lockdown zeigt eindeutig, wie viele Bereiche des Lebens nach wie vor von physischer Präsenz abhängig sind. Nicht nur Freizeit und soziale Interaktion, sondern auch alltägliche Arbeitsabläufe, wie z.B. die gemeinschaftliche Arbeit an einem Dokument oder die Koordination engmaschiger Abläufe im Beruf zeigen auf, was in der Kreativindustrie Standard ist: ohne physische Präsenz und ohne passende Werkzeuge ist die Zusammenarbeit aus der Ferne kaum möglich. Wo uns Programmiererinnen und IT-Angestellte weit voraus sind, gilt es sowohl für das Büro, das Sozialleben oder eben auch die Kreativindustrie Werkzeuge zu schaffen.

Bands proben regelmässig zusammen, er- und bearbeiten meist in kollektiven Strukturen ihr Repertoire, arbeiten inkrementell und iterativ, und nehmen jeweils die neuen Proben wieder auf. Selbständige Arbeiten an den Stücken (Üben, Texte schreiben, Melodien und Riffs ausdenken, etc.) sind möglich, sollten jedoch Änderungen mitgeteilt, etwas rückgefragt, etwas Neues mit einer Bandkollegin abgemacht werden, scheitert dies meist an der Kommunikation. Zwar ist es sprachlich möglich, korrekt und genau zu formulieren jedoch zeigen folgende Beispiele, warum das nicht nur unpraktisch, sondern oft ein Dealbreaker ist.

2.2 Problemanalyse

2.2.1 Referenz auf eine Stelle durch eine Zeitangabe

- «Hallo X! Im File 'Bandprobe vom 23. Mai 2020 Track 01.mp3' bei 3 Minuten 12 Sekunden spielst du einen falschen Akkord»

Vorteil dieser Formulierung: Wenn die Datei vorhanden ist, kann der Fehler schnell entdeckt und die Zuweisung dessen rasch bestätigt werden.

Nachteil: Musikerinnen orientieren sich meist in sich wiederholenden Formen. Hier müsste die Musikerin X die Datei erst finden, dann öffnen, dann anerkennen, dass sie da falsch spielt, dann herausfinden zu welcher Form und zu welchem Teil das gehört - was bedeutet, dass das ganze Stück von vorne angehört werden muss.

2.2.2 Referenz auf eine Stelle durch musikalische Begriffe

- «Im Song 'Don't Eat The Yellow Snow' holpert der Wechsel vom Refrain in die dritte Strophe»

Musikerinnen wissen sofort welche Stelle gemeint ist. Tontechnikerinnen analysieren die Stücke eher Technisch (und beschreiben Stellen als Zeitangaben) und müssen - um die Stelle zu finden – erst den ganzen Song durchhören - und wissen, welcher Teil als *Intro*, *Refrain* oder *Strophe* bezeichnet wird.

2.2.3 Referenz durch absolute Taktangabe

- «Beim Mixdown der Ballade 'Solitude' habe ich 22. Takt auf den vierten Schlag ein Störgeräusch wahrgenommen»

In Musikproduktionen, welche auf Partituren und zentralisierter Entscheidungshierarchie aufbauen (vor allem, aber nicht nur Klassische Musikproduktionen mit Dirigentinnen), funktioniert dies einwandfrei. Alle betreffenden Beteiligten verfügen über die notwendigen Notenblätter und selbst die Technikerinnen orientieren sich daran. Absolute Taktangaben sind aber sowohl für instrumentelle Rock-, Pop- oder Jazzproduktionen, als auch für elektronische Produktionen aller Art bestenfalls befremdend.

2.3 Use-Case-Analyse

Wir werden die Lösung der gerade dargelegten Problemstellung anhand zweier Use-Case-Analysen erläutern: Auf der einen Seite gibt es die klassische Rock-, Pop- oder Jazzband, welche an Bandproben Aufnahmen erstellt und jeweils zwischen den Proben die Aufnahmen anhört (z.B. zum Üben). Die Kommunikation soll möglichst hierarchiefrei, die Files übersichtlich angeordnet sein.

Auf der anderen Seite haben wir einen Use-Case für elektronische Musikproduzentinnen und Tontechnikerinnen analysiert. Sobald eine Band im Studio war und die Aufnahmen eingespielt sind, geht es um den Mixdown [1] der Musik. Beim Mixdown mischt eine Tontechnikerin (die sogenannte Mischerin) die einzelnen Spuren zu einem (Stereo) Track zusammen. Der Mixdown ist eine anspruchsvolle Aufgabe – darin wird festgelegt, was wann wie laut schallen, welche Stimmen oder Instrumente in welchen Klangfarben erklingen und wie die räumliche Aufteilung der Gruppe wahrgenommen wird. Da diese ästhetischen Merkmale minutiös und bis ins letzte Detail ausgearbeitet werden müssen, um den hohen Ansprüche der Musikerinnen zu genügen, braucht es viel Geduld.

2.3.1 Workflow

2.3.1.1 Session anlegen und Kollaborateurinnen einladen

2.3.1.1.1 Use-Case A (Band)

Eine Bandleaderin (Inhaberin eines Accounts auf rehearsal.ch¹) erstellt eine neue Session. Sie gibt einen Namen, Datum und Zeit und beschreibt die Session mit ihren Worten. Sie fügt Mitmusikerinnen hinzu, welche – falls sie noch keinen Account im System haben – per E-Mail eingeladen werden.

¹ reHEARsal Webseite.

Dann erstellt die Bandleaderin einen oder mehrere Tracks in der Session. Zu jedem Track wird mindestens eine Version hinzugefügt.

2.3.1.1.2 Use-Case B (Studio)

Die Mischerin erstellt für jeden Aufnahmetag eine Session und füllt sie mit den an diesem Tag eingespielten Stücken (Tracks und Versionen) - sogenannte *rough-mixes*. Sobald die Aufnahmen abgeschlossen sind, wird eine neue Session erstellt, die «Mix Session». Darin wird für jedes aufgenommene Musikstück ein Track angelegt mit dem jeweiligen rough-mix als erster Version.

2.3.1.2 Bearbeitung

2.3.1.2.1 Use-Case A (Band)

Sobald die Versionen angelegt und die Musikerinnen eingeladen sind, können Kommentare angelegt werden, um Unklarheiten zu beseitigen. Zum Beispiel möchte Anna wissen, ob die neueste Version von «Wind Up Workin In A Gas Station» nicht zu langsam eingespielt wurde. Und Erika wundert sich, ob sich Anna beim Lied «Panzerballett zum Wiener Walzer» im dritten Takt des Intros verspielt hat oder ob in ihren eigenen Notizen ein falscher Akkord steht.

Also legt Anna einen *VersionComment* bei der neuesten Version vom Track «Wind Up Workin In A Gas Station» an. Und Erika erstellt einen *TimeComment* in der richtigen Version an genau der Stelle, wo sie meint, Anna hätte sich verspielt.

So können sich die Musikerinnen von zuhause aus über die Richtigkeit ihrer kreativen Vorhaben vergewissern und sind bei der kommenden Probe noch produktiver, denn alle Verständnisfragen und zu klärenden Angelegenheiten, konnten schon im Voraus aus dem Weg geräumt werden.

2.3.1.2.2 Use-Case B (Studio)

Sobald die rough-mixes angelegt und die Musikerinnen eingeladen sind, können Fragen ausgetauscht werden. Zum Beispiel braucht die Mischerin zu jedem Track ein oder mehrere Vorbilder (z.B. ein fertiges Stück einer anderen Musikgruppe, welchem der abzumischende Song klangästhetisch entsprechen soll), oder es wurden beim Track «What's The Ugliest Part Of Your Body» unterschiedliche Rohfassungen eingespielt und sie muss wissen, mit welcher sie nun arbeiten soll.

Sobald genügend Informationen vorhanden sind, macht sie sich an die Arbeit, um dann wiederum die bearbeiteten Versionen wieder hochzuladen. Dann möchte die Bandleaderin wissen, warum ihre Gitarre so quietscht an der einen Stelle, die Bassistin fragt, ob ihr Bass absichtlich so leise ist, etc.

2.4 Lösungsansatz

Das Problem konzentriert sich offensichtlich um die Frage, wie Musikerinnen ihre Mitmusikerinnen auf ganz genau spezifizierbare Details hinweisen können. Damit sind sowohl *Tracks* (also Song-Ideen, welche z.B. aus Text und Begleitakkorden bestehen), *Versionen* (Aufnahmen, welche einzeln oder zusammen gefertigt wurden) als auch Stellen in Versionen gemeint.

2.4.1 Technologie

Technisch gesehen handelt es sich bei den *Versionen* um Audio-Dateien. Versions-Stellen können als relative Angaben, Kommentare als Ketten von Computer-kodierten Zeichen verstanden werden. Zur Administration der Daten greifen wir auf Datenbanktechnologie zurück.

Da keine Vorgaben die Mensch-Computer-Interaktion betreffend vorhanden waren, wurde beschlossen, die Applikation als Webseite umzusetzen. So kann die Applikation möglichst vielen Nutzerinnen zur Verfügung gestellt werden.

3 Resultate

reHEARsal ist ein Werkzeugkasten, welcher sowohl zur Archivierung, für das referenzielle Kommentarsystem, die moderner Datenaufbereitung und ihre ansprechender Darstellung benutzt wird.

3.1 Die Applikation aus Nutzerinnensicht

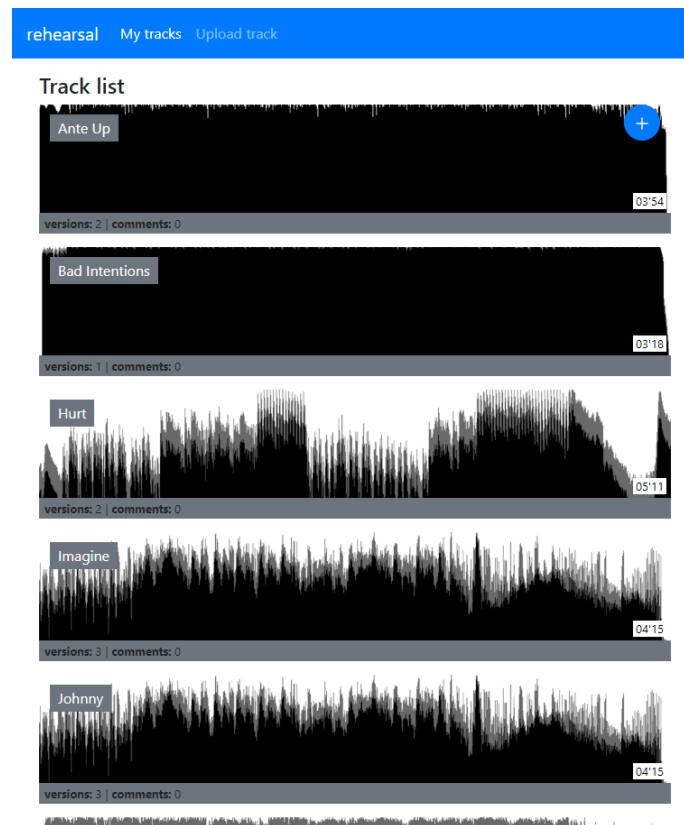


Abbildung 1: Trackliste

Wird eine *Session* ausgewählt, erhält die Nutzerin umgehend eine Übersicht über die darin enthaltenen *Tracks*.

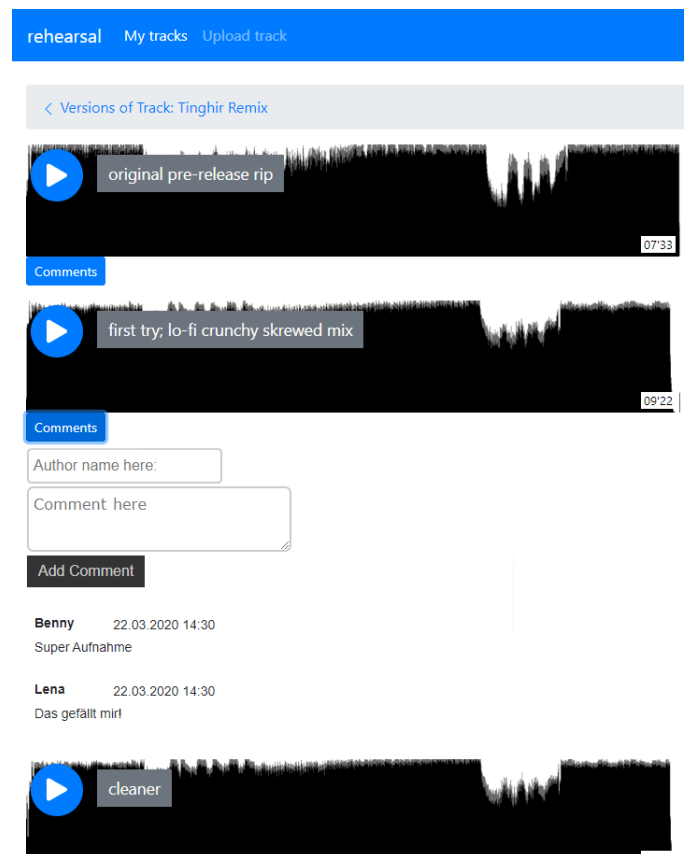


Abbildung 2: Versionsliste

Wird ein Track aus obiger Listenansicht ausgewählt, wird eine Übersicht zu den vorhandenen Versionen angezeigt.

3.2 Systemarchitektur

Moderne Webapplikationen werden in den allermeisten Fällen in einer sogenannten Two-Tier Architecture umgesetzt. Dabei wird zwischen Frontend (vordergründige Ansichtsaufbereitung für die Endbenutzerinnen) und Backend (hintergründige Informationsverarbeitung) unterschieden.

3.2.1 Frontend

Mit *Frontend* wird derjenige Teil einer Applikation bezeichnet, mit welchem eine Endbenutzerin direkt interagiert. Dies wurde in den ersten Jahren des WWW mittels Hypertext Markup Language (HTML) umgesetzt. Obwohl HTML nach wie vor der Standard für die Interaktion zwischen Endbenutzerin und Webapplikation bleibt, wird für moderne Webapplikationen kaum mehr HTML-Code von Hand geschrieben. Es existiert eine Vielzahl sogenannter Frontend-Frameworks, welche die programmatische Generierung von HTML durch die Bereitstellung layoutfreundlicherer Werkzeuge vereinfachen. Von diesen haben wir uns für *Vue.js* entschieden, da ein Teil des Teams mit dieser Technologie bereits Erfahrung hatte.

3.2.2 Backend

Mit *Backend* wird derjenige Teil einer Applikation bezeichnet, welcher nicht für Endnutzerinnen sichtbar ist. Das *Backend* ist vor allem für die Datenverarbeitung, die zentrale Speicherung und die Datenaufbereitung zuständig. Das Backend kann aber auch Administrationsfunktionalitäten erfüllen und für Fremdsysteme eine sogenannte API (Application Programming Interface) zur Verfügung stellen.

Mit dem Aufkommen des sogenannten Web 2.0 [2] wurde die User-Interaction für die Inhalte des World Wide Web ins Zentrum gerückt. Deshalb entstanden ab dem Millennium nicht nur etliche unterschiedliche Content Management Systeme (CMS, wie z.B. Wordpress [3], DokuWiki [4], u.v.a.m.), sondern auch Frameworks, um dem eigenen Nutzen entsprechende CMS herzustellen (z.B. Ruby on Rails [5] oder Django [6]). Da Kenntnisse der Programmiersprache *Python* (in welcher *Django* geschrieben ist) im Team vorhanden waren und Python dafür bekannt ist, auch komplexere Signal- und Datenverarbeitung einfach und rasch zu ermöglichen, hat sich das Team für die Implementation des Backends für Python/Django entschieden.

3.2.3 Interaktion

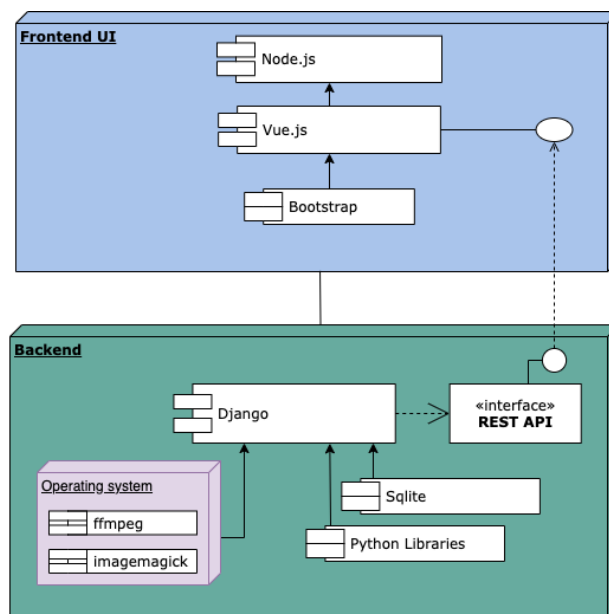


Abbildung 3: Systemarchitektur

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.

4 Diskussionen und Ausblick

4.1 Visuelle Orientierung durch Wellenform

Das Team wurde sich schnell einig, dass zur Orientierung innerhalb einer Audio-Datei, visuelle Hilfen [7] von Vorteil sind. Aus der Mathematik sind uns statistische Mittel bekannt, um grosse Zahlenreihen (was Audio-Dateien letztlich sind) graphisch darzustellen. Im Zuge des Projektes wurde deshalb die Dienstklasse AmpliSpectrogram geschaffen, welche Audiodaten einliest und mittels einem RMS-Algorithmus eine Grafik erzeugt [8]. Die schwarzen Stellen sind dabei auf den Mitten-Anteil einer Audiodatei zurückzuführen, die grauen Stellen auf den Seiten-Anteil.



Abbildung 4: AmpliSpectrogramm des Lieds Never Gonna Give You Up

Kommentare können nun an den referenzierten Stellen visuell dargestellt werden. Musikerinnen und Technikerinnen orientieren sich anhand der Wellenform.

In einer ersten Planungsphase wurde angedacht, über die Farbe die Intensitätsverteilung im Hörbereich darzustellen (analog zum Farbspektrum: basslastige Stellen werden rot oder rötlich dargestellt, mittenlastige Stellen grün, höhenlastige Stellen blau), woher auch der Name AmpliSpectrogram stammt, da sowohl Amplitude, als auch Frequenzspektrum grafisch dargestellt werden. Nach Abwägen von Aufwand und Nutzen wurde die Umsetzung aber tief priorisiert und ist in der aktuellen Version von reHEARsal nicht vorhanden.

4.2 Kommentare als Threads

Wahrscheinlich ist keine Errungenschaft des World-Wide-Web so oft implementiert worden, wie die Kommentarfunktion. Deshalb gibt es unterschiedliche Philosophien und Umsetzungen zu dieser so trivial wirkenden Aufgabe. Die dabei aufgekommenen Fragen lauten:

- Soll auf spezifische Kommentare geantwortet werden können oder werden Kommentare stets zeitlich linear dargestellt?

Wenn jeder Kommentar zu einem Referenzpunkt eines jeden (nachfolgenden) Kommentars werden kann, folgt daraus ein baumartige Struktur. Diese könnte zwar Diskussionen erleichtern, fordert aber sowohl im Frontend (graphische Darstellung der Applikation), als auch im Backend (hintergründige Datenverarbeitung und –aufbereitung) einen unproportionalen Mehraufwand. Es wurde sich geeinigt, das Tool schlicht und einfach zu halten, für

Diskussionen werden die Nutzerinnen angehalten bereits bestehende Möglichkeiten auszuschöpfen (z.B. an der Bandprobe, per E-Mail, Chat oder Videochat).

- Sollen an der selben Stelle in einer Version mehrere Kommentar-Threads möglich sein?

Der Einfachheit halber wurde entschieden, dass dies nicht möglich sein soll.

- Sollen auch Kommentare zu *Tracks* oder *Sessions* möglich sein?

Tracks beschreiben in reHEARsal im Gegensatz zu Versionen eher ideelle (zu verstehen als platonische Idealvorstellungen) als materielle (konkrete, jemals existierende) Konstrukte, was das Team vor diese Frage stellte. Soll eine Idee kommentiert werden können?

Da die konkrete Nutzung eines Produkts nie von den Produzentinnen vorausgesehen werden kann, sich das Team keinen dadurch entstehenden Schaden vorstellen kann und die Umsetzung verhältnismässig trivial ist, wird diese Funktion in die Planung aufgenommen (jedoch sehr tief priorisiert).

4.3 Authentisierung oder anonyme Gruppen?

Im Verlauf der Entwicklung der Applikation, kam die Frage auf, ob eine ordentliche Benutzerinnen-Authentisierung nicht der Einfachheit der Applikation im Wege stünde. Herkömmliche Authentisierung auf dem World-Wide-Web hat die Aufgabe, boshafte Nutzerinnen davon abzuhalten Schaden anzurichten. Da unsere Applikation aber nicht eine Schnittstelle zwischen Kreierenden und Konsumentinnen darstellen soll, trifft diese Aufgabe nicht auf unsere Anwendung zu. Das Team entschied sich also dazu, für erstellte Sessions eindeutige Strings zu generieren (ein englisches Adjektiv und ein englisches Nomen, verbunden mit einem «-»), über welche die Zusammenarbeit ermöglicht wird – das heisst: Jede Person mit dem Link kann die Daten bearbeiten.

5 Literaturverzeichnis

- [1] Wikipedia, «Audio mixing (recorded music),» [Online]. Available: [https://en.wikipedia.org/wiki/Audio_mixing_\(recorded_music\)](https://en.wikipedia.org/wiki/Audio_mixing_(recorded_music)). [Zugriff am 11. Mai 2020].
- [2] Wikipedia, «Web 2.0,» [Online]. Available: https://en.wikipedia.org/wiki/Web_2.0. [Zugriff am 11. Mai 2020].
- [3] Wordpress, «Wordpress,» [Online]. Available: <https://wordpress.org>. [Zugriff am 11. Mai 2020].
- [4] DokuWiki, «DokuWiki,» [Online]. Available: <https://dokuwiki.org>. [Zugriff am 11. Mai 2020].
- [5] Wikipedia, «Ruby on Rails,» [Online]. Available: https://en.wikipedia.org/wiki/Ruby_on_Rails. [Zugriff am 11. Mai 2020].
- [6] Wikipedia, «Django (web framework),» [Online]. Available: [https://en.wikipedia.org/wiki/Django_\(web_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)). [Zugriff am 11. Mai 2020].
- [7] Soundcloud, «Soundcloud,» [Online]. Available: <https://soundcloud.com>. [Zugriff am 11. Mai 2020].
- [8] Wikipedia, «Root mean square,» [Online]. Available: https://en.wikipedia.org/wiki/Root_mean_square. [Zugriff am 11. Mai 2020].

6 Abbildungsverzeichnis

Abbildung 1: Trackliste	7
Abbildung 2: Versionsliste.....	8
Abbildung 3: Systemarchitektur	9
Abbildung 4: AmpliSpectrogram des Lieds Never Gonna Give You Up.....	10

7 Versionierung

Version	Beschreibung
1.0	Fachartikel Entwurf.
2.0	Fachartikel (endgültig).

8 Glossar

Begriff	Bedeutung
AmpliSpectrogram	Eine Grafik, welche auf der X-Achse die Zeit, auf der Y-Achse die Lautstärke und mit Farbe über die X-Achse die Intensitäten im Hörbereich darstellt.
API	Eine limitierte Anzahl an Funktionen, welche eine Software nach aussen freigibt um zu kommunizieren.
Backend	Siehe Abschnitt 3.2.2
Bootstrap	Layout Framework
CMS	Content-Management-System; eine Applikation zur Verwaltung unterschiedlicher Inhalte
Django	CMS Framework (geschrieben in Python)
DokuWiki	CMS für zusammenhängendes Wissen (wikipedia)
Framework	Entwicklungsrahmen, mit welchem einfacher Software für bestimmte Zwecke oder in bestimmten Umgebungen realisiert werden können
Frontend	Nutzerinnenseitige Schnittstelle zur Anwendung (siehe Abschnitt 3.2.1)
HTML	«Hypertext Markup Language», der Standard um Dokumente zu erstellen, welche im Internet dargestellt werden sollten.
Interface	Schnittstelle
JavaScript	Kurz: JS, Skriptsprache für dynamisches HTML in Webbrowsern
JSON	Ein Dateiformat um Daten zwischen Anwendungen auszutauschen.
Library	Eine Ansammlung von Scripts zu einem Thema, welche geteilt werden kann.
Mixdown	Prozess, in welchem aus vielen Aufnahmespuren eine oder mehrere – für ein Publikum fabrizierte – Spuren entstehen.
Node.js	Plattform für JavaScript Webanwendungen
Part	Für jedes Instrument gibt es einen Part. Alle Parts zusammengeführt ergeben den Track.
Python	Eine Programmiersprache
REST API	Eine Definition wie zwei unabhängige Systeme miteinander kommunizieren sollen.
RMS-Algorithmus	Statistisches Mittel, mit welchem Lautstärke gemessen werden kann
Ruby on Rails	CMS Framework
Session	Eine Sammlung von Tracks (z.B. eine spezifische Recording-Session, oder die Tracks, die auf das nächste Album sollen).
SQLite	Ein Datenbanksystem
Thread	Ein linearer Zusammenhang
TimeComment	Eine Bemerkung, welche eine genaue Stelle einer bestimmten Version kommentiert.
Track	Ein Musikstück -- oder genauer: die Vorstellung, das Ziel eines Musikstücks.
UI	«User Interface», also Die grafische Benutzeroberfläche einer Software.
Use-Case	Anwendungsfall
Version	Ein Audiofile, welches versucht die Ideen eines Tracks umzusetzen.
VersionComment	Ein Kommentar, der auf eine bestimmte Version zutrifft.

Vue.js	Frontend Framework zur Entwicklung von User Interfaces
Webbrowser	Programm, mit welchem Internetseiten im WWW aufgerufen werden
Wordpress	CMS für Blogsysteme
WWW	«World Wide Web»
imagemagick	Softwarepaket zur Erstellung und Bearbeitung von Raster- und Vektorgrafiken
ffmpeg	Programmbibliotheken, die digitales Video- und Audiomaterial aufnehmen, konvertieren kann