

**Philosophische** Fakultät III

Sprach- , Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Medieninformatik

Praxisseminar Software Engineering

Modul: MEI-M 26 (M.Sc.)

SS 2015

Leitung: Christian Wolff, Rafael Wimmer

Transkriptions-Werkzeug

Hoerburger-Aufnahmen

Software Requirements Specification –Anforderungsspezifikation

Version 1.0

Martin Kocur, Stefan Covaci, Murat Firat, Daniel Schmidl

1580400,1466192,1586591,1567694

Medieninformatik

1. Semester M.Sc., 2. Semester M.Sc., 1. Semester M.Sc., 1. Semester M.Sc.

Brunhuberstr. 12a

93053 Regensburg

Tel.: 0941/20900043

E-Mail: stefan.covaci@stud.uni-regensburg.de

Abgegeben am 02.06.2015

Inhalt

[0 Versionsgeschichte 5](#_Toc420337794)

[1 Einführung 6](#_Toc420337795)

[1.1 Zweck des Dokuments 6](#_Toc420337796)

[1.2 Projektumfang (Scope) 7](#_Toc420337797)

[1.3 Definitionen, Akronyme und Abkürzungen 7](#_Toc420337798)

[1.4 Referenzen und Quellenangaben 7](#_Toc420337799)

[1.5 Übersicht über das Dokument 8](#_Toc420337800)

[2 Allgemeine Beschreibung 9](#_Toc420337801)

[2.1 Produktperspektive 9](#_Toc420337802)

[2.2 Produktfunktionen 9](#_Toc420337803)

[2.3 Benutzergruppen und Charakteristika 10](#_Toc420337804)

[2.4 Zielplattform 11](#_Toc420337805)

[2.5 Einschränkungen, Annahmen und Abhängigkeiten 11](#_Toc420337806)

[2.6 Benutzerhandbuch und Dokumentation 11](#_Toc420337807)

[3 Spezifische Anforderungen 12](#_Toc420337808)

[3.1 (Externe) Schnittstellen 12](#_Toc420337809)

[3.1.1 Benutzerschnittstellen 12](#_Toc420337810)

[3.1.2 Hardware-Schnittstellen 13](#_Toc420337811)

[3.1.3 Software-Schnittstellen 13](#_Toc420337812)

[3.1.4 Kommunikationsschnittstellen 14](#_Toc420337813)

[3.2 Funktionale Anforderungen 14](#_Toc420337814)

[3.3 Anwendungsfälle (Use Cases) 16](#_Toc420337815)

[4 Nicht-funktionale Anforderungen 21](#_Toc420337816)

[4.1 Performance 21](#_Toc420337817)

[4.2 Sicherheit 21](#_Toc420337818)

[4.3 Qualität 21](#_Toc420337819)

[4.3.1 Ausfallsicherheit 22](#_Toc420337820)

[4.3.2 Erreichbarkeit 22](#_Toc420337821)

[4.3.3 Wartbarkeit 22](#_Toc420337822)

[4.3.4 Erweiterbarkeit 22](#_Toc420337823)

[4.3.5 Usability 22](#_Toc420337824)

[4.4 Datenmodell 22](#_Toc420337825)

[4.5 Sonstige Anforderungen 22](#_Toc420337826)

Abbildungen

[Abbildung 1: TOP 10 Qualitätskriterien eines SRS. 5](#_Toc420337827)

[Abbildung 2. Schnittstellen-Diagramm 9](#_Toc420337828)

[Abbildung 3. Beispielhafter Workflow im Umgang mit dem Annotations-Tool 10](#_Toc420337829)

[Abbildung 4. Google Drive Drag & Drop Funktion als Designanregung 12](#_Toc420337830)

[Abbildung 5. Reaper Audioprogramm als Designanregung für das Setzen von Markierungen 13](#_Toc420337831)

[Abbildung 6. Standard HTML Upload (<!selfHTML.wiki/>, n.d.) 15](#_Toc420337832)

Tabellen

# Versionsgeschichte

|  |  |  |
| --- | --- | --- |
| Version | Datum | Beschreibung |
| 1.0 | 24.05.2015 | Vorläufige Version |
| 1.1 | 31.05.2015 | Korrigierte Version |

# Einführung

## Zweck des Dokuments

Dieses Dokument dient als nähere Beschreibung unseres Projektes innerhalb des Praxisseminars im Sommersemester 2015 des Masterstudiengangs Medieninformatik an der Universität Regensburg. Hiermit soll die von uns hergestellte Software sowohl vorgestellt, als auch genauer erklärt werden. Dabei sollen den Lesern dieser Software Requirements Specification alle Fragen bezüglich des Produktes beantwortet. Dieses Dokument hat demzufolge den Zweck mit Hilfe der verschiedenen Anforderungsspezifikationen das Produkt sowohl dem Kunden als auch der Entwicklung verständlich zu machen. Somit sollen anhand dieses SRS die Hauptfunktionen, Aufgaben, Use Cases und Ziele des Produkts beschrieben werden. Außerdem wird in diesem Dokument aufgelistet, welche Benutzergruppen und Zielplattform diese Software anspricht. Verschiedene externe Schnittstellen, wie z.B. Benutzerschnittstellen, Hardware-/ Softwareschnittstellen, sowie Kommunikationsschnittstellen werden durch dieses Dokument vorgestellt und erklärt. Im Übrigen werden Qualitätsanforderungen, Sicherheitsanforderungen und Performanceanforderungen beschrieben, um die Software optimal benutzbar machen zu können. Dieses Dokument dient somit als Benutzerhandbuch für das im Folgenden beschriebene System.

Das Produkt, das in dieser SRS beschrieben werden soll, ist ein Transkriptionswerkzeug für die Hoerburger - Aufnahmen. Damit können digitalisierte Tonbandaufnahmen, in diesem Fall Tonbandaufnahmen von Prof. Dr. Felix Hoerburger, die er in verschiedenen Forschungsreisen festgehalten hat, bearbeitet werden. Hierbei handelt es sich um die erste Produktversion.

Die Zielgruppe dieses Dokuments sind hauptsächlich Bibliotheksmitarbeiter, wissenschaftliche Hilfskräfte und studentische Hilfskräfte der Universität Regensburg, die diese Audiodaten erfassen und mit diesem Tool arbeiten werden, um die bis jetzt umfangreichen und unstrukturierten Hoerburger- Aufnahmen bearbeiten zu können. Außerdem zählen neben den Mitarbeitern der Universität Regensburg weitere Wissenschaftler, sowie interessierte Laien, die sich mit diesem Thema befassen, zu der Zielgruppe dieses Tools und somit auch zur Zielgruppe dieses Dokuments.

Den Lesern dieses Dokuments erwarten demzufolge eine genaue Spezifikation der Anforderungen für die digitale Bearbeitung der Hoerburger- Aufnahmen aus Sicht des End-Anwenders und der Sicht des Entwicklers, indem alle Funktionalitäten, Aufgaben und Eigenschaften des Produkts beschrieben und erklärt werden.

## Projektumfang (Scope)

Wie oben schon erwähnt, hat das von uns hergestellte Tool die Aufgabe den Usern ein Transkriptionswerkzeug für Hoerburger- Aufnahmen zur Verfügung zu stellen. Die Universität Regensburg ist im Besitz von umfangreichen Tonbandaufnahmen, die von Prof. Dr. Felix Hoerburger in zahlreichen Forschungsreisen aufgezeichnet wurden. Diese sind zwar digitalisiert, jedoch sind diese ungeordnet und nicht den entsprechenden Metadaten zugeordnet. Die jeweiligen Metadaten zu diesen Tonbandaufnahmen sind größtenteils vorhanden, aber nicht komplett vollständig oder können noch mit weiteren Metadaten ergänzt werden. Ziel ist es ein Bearbeitungstool für die User zu schaffen, mit dem eine Audiodatei in die Benutzeroberfläche hochgeladen und anschließend Markierungen gesetzt werden können, um bestimmte Teile der Tonbandaufnahmen zu den jeweiligen Metadaten zuzuordnen. Das heißt, beispielsweiße sollte es den Nutzern ermöglicht werden Anfangs- und Endmarkierungen zu setzen, um damit ein bestimmtes Lied abzugrenzen und diesen Teil der Audiodatei zu den jeweiligen dazugehörigen Metadaten zuweisen zu können. Eine Erweiterung der Metainformationen ist ebenfalls möglich. Folglich hat dieses System das Ziel, die Tonbandaufnahmen von Prof. Dr. Felix Hoerburger durch Bearbeitung, bzw. durch das Setzen von Markierungen und der Zuordnung der dazugehörigen Metadaten, zu strukturieren und übersichtlicher darzustellen.

## Definitionen, Akronyme und Abkürzungen

FTP = File Transfer Protocol

HTTP = Hypertext Transfer Protocol

HTML = Hypertext Markup Language

PHP = Hypertext Preprocessor

## Referenzen und Quellenangaben

<!selfHTML.wiki/> (n.d.). *HTML/ Formulare/ Felder/ Datei-Upload* . Retrieved 20 Mai, 2015, from http://wiki.selfhtml.org/extensions/Selfhtml/example.php/Beispiel:HTML\_input-Element10.html

## Übersicht über das Dokument

In den folgenden Teilen dieser Dokumentation wird zunächst eine allgemeine Beschreibung des Produktes formuliert. Dabei werden Themen wie die Produktperspektive, die Hauptfunktionen und Ziele der Software, die verschiedenen Nutzergruppen und Charakteristika, die Zielplattform, d.h. für welche Plattform die Software geeignet ist, angesprochen. Es werden auch Einschränkungen, Annahmen und Abhängigkeiten dargestellt. Im nächsten Gliederungspunkt werden spezifische Anforderungen dargestellt. Dabei werden die externen Schnittstellen, wie Benutzerschnittstellen, Hardware- Schnittstellen, Software- Schnittstellen und Kommunikationsschnittstellen erwähnt. Anschließend werden funktionale Anforderungen aufgezählt und gezeigt welche Features und Funktionen ausgeführt werden können. Im letzten Punkt dieses Abschnittes werden sämtliche Anwendungsfälle dokumentiert. Zuletzt werden in dieser SRS die nicht- funktionalen Anforderungen, wie Performance, Sicherheit und Qualität dem Leser gezeigt. Anforderungen an das Datenmodell und sonstige Anforderungen werden ebenfalls in diesem Teil dokumentiert.

# Allgemeine Beschreibung

## Produktperspektive

Der Ursprung des Produkts liegt in den Aufnahmen von Dr. Felix Hoerburger aus den 50er und 60er Jahren, auf denen er Volksmusik aus verschiedensten Kulturen aufnahm. Diese liegen nun digitalisiert, aber größtenteils noch ungeordnet, vor. In den meisten Fällen handelt es sich dabei um große .WAV-Dateien, welche wiederum aus einzelnen Tracks bestehen. Die Aufnahmen sollen mit Hilfe unseres Tools mit zusätzlichen Informationen (Metadaten) ausgestattet und die einzelnen Tracks hervorgehoben werden können. Als Hauptkomponente dient die Amplitudenansicht ganzer Audio-Dateien.

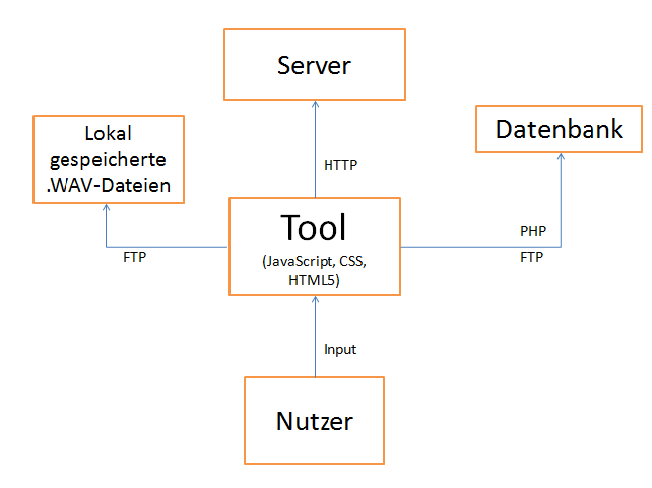


Abbildung . Schnittstellen-Diagramm

## Produktfunktionen

Eine Hauptfunktion des Tools ist die Möglichkeit, in den Audio-Dateien mit Hilfe der Amplitudenansicht Markierungen setzen zu können um einzelne Tracks zu identifizieren und voneinander abzugrenzen. Da sich zumeist mehrere Tracks in einer Datei befinden, ist es daher auch wichtig die Möglichkeit zu haben, mehrere Markierungen zu setzen. Des Weiteren müssen die markierten Tracks auch mit Metadaten ausgestattet werden können, um sie näher zu beschreiben. Die vom Nutzer gesetzten Markierungen und Metadaten zu einzelnen Tracks sollen auf einem Server gespeichert werden.

Zu den Funktionen des Tools gehört der Upload einer Audiodatei in den Browser, welche anschließend in Amplitudenanzeige visualisiert wird. Wurde eine Audiodatei bereits bearbeitet und mit Markierungen versehen, sollen diese automatisch aus der Datenbank geladen und angezeigt werden.

Des Weiteren müssen die markierten Tracks auch mit Metadaten ausgestattet werden können, um sie näher zu beschreiben. Die vom Nutzer gesetzten Markierungen und Metadaten zu einzelnen Tracks sollen auf einem Server gespeichert werden und somit jederzeit wieder abrufbar sein. Diese können anschließend eingesehen und erweitert werden.



Abbildung . Beispielhafter Workflow im Umgang mit dem Annotations-Tool

## Benutzergruppen und Charakteristika

Als Nutzergruppen für unser Annotations-Tool kommen vor allem Bibliotheksmitarbeiter der Universität, welche sich mit den Aufnahmen von Hoerburger beschäftigen, in Frage, ebenso studentische sowie wissenschaftliche Hilfskräfte. Außerdem bietet sich dieses Tool auch für Wissenschaftler an, die sich mit diesem Thema befassen. Das Ziel dieser Nutzergruppe ist die Aufbereitung der Aufnahmen durch Identifizierung der einzelnen Tracks und deren Ausstattung mit Metadaten. Des Weiteren können auch interessierte Laien von dem Werkzeug Gebrauch machen, um eventuell zusätzliches Wissen zu der Sammlung beizutragen.

## Zielplattform

Zunächst wird die Software für den Browser "Google Chrome" entwickelt, da sich hier programmiertechnisch gesehen, die wenigsten Probleme ergeben. Auch die anderen gängigen Browsern sollten das Tool öffnen können (mit der Ausnahme von IE, da dieser Browser „Web Audio API“ nicht unterstützt).

## Einschränkungen, Annahmen und Abhängigkeiten

Einschränkungen ergeben sich bei der Größe der einzulesenden Datei. Übersteigt diese einen gewissen Wert, so ist das Tool nicht mehr in der Lage die Datei in der Amplitudenansicht darzustellen. Außerdem können zunächst nur Hoerburger-Aufnahmen bearbeitet werden, da hierfür eine ID notwendig ist, welche vorerst nur diese besitzen.

Als externe Schnittstellen benötigen wir eine Datenbank zum Speichern der Dateien, sowie Zugang zu ihrer technischen Dokumentation.

## Benutzerhandbuch und Dokumentation

Bei Abschluss des Projekts stehen den Nutzern, Stakeholdern und Auftraggebern eine schriftliche Anleitung zur Nutzung des Tools vor.

# Spezifische Anforderungen

## (Externe) Schnittstellen

### Benutzerschnittstellen

*Beschreiben Sie die umzusetzenden Schnittstellen zwischen Nutzer und System: Wo ist eine Nutzerinteraktion notwendig (siehe Abschnitt Anwendungsfälle). Auf welche Prinzipien und Styleguides (z.B. Windows 8 User Experience Guidelines) kann aufgesetzt werden. Definieren Sie das Interaktionskonzept (Art der Ein- und Ausgabe). Beziehen Sie plattformspezifische Guidelines mit ein. Gerne können Sie hier auch (erste) UI-Sketches einfügen.*

1. Upload einer Audiodatei

Die erste Form der Interaktion des Nutzers besteht darin, eine beliebige Audio-Datei per HTML Standardupload oder Drag & Drop hochzuladen. Für die Drag & Drop Funktion kann man Anregungen von Google Drive oder anderen Portalen finden (siehe Abbildung 2). Auf diese Weise soll gleich eine Wellenform der hochgeladenen Audiodatei resultieren und dargestellt werden.

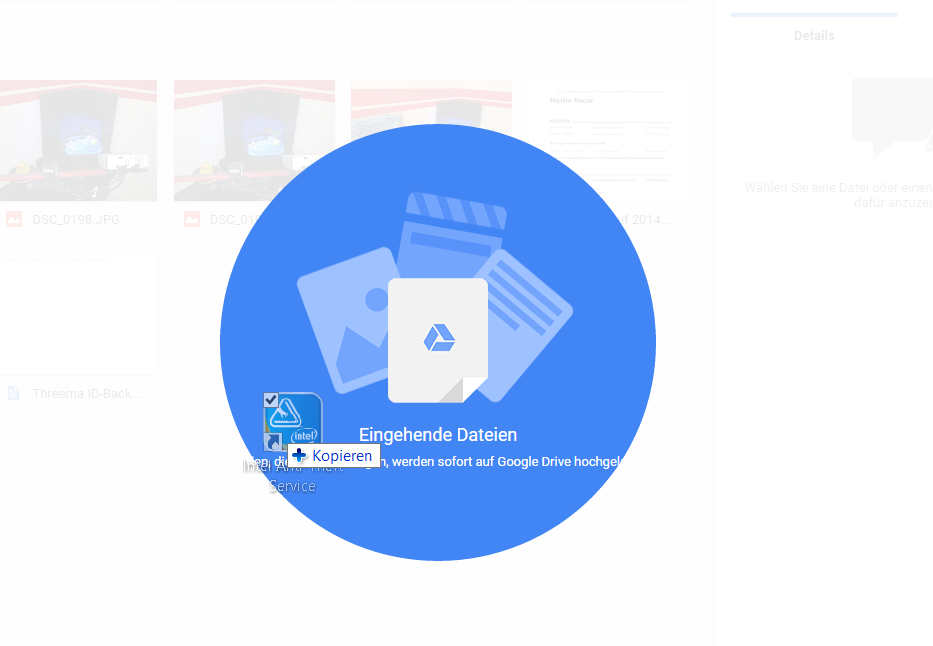


Abbildung . Google Drive Drag & Drop Funktion als Designanregung

1. Setzen von Markierungen

Außerdem können die Audiodateien mit Hilfe der Audiowellen durch Setzen von Markierungen in einzelne Tracks unterteilt werden. Dadurch können zu jedem Track Informationen in Form von Metadaten abgerufen und auch hinzugefügt werden. Bereits bestehende Tracks inklusive der Metadaten können ebenfalls ausgewählt werden. Die Markierungen werden in Form von Strichen gestaltet. Abbildung 3 zeigt die Idee des Markierens durch Klick auf die gewünschte Position auf der Audiowelle.



Abbildung . Reaper Audioprogramm als Designanregung für das Setzen von Markierungen

1. Metadaten abrufen/hinzufügen

Durch Markieren eines Abschnittes, können explizit für diesen Abschnitt Informationen hinzugefügt oder erhalten werden. Dazu klappt ein bestimmter Bereich im Fenster unter der Wellendarstellung auf, um die Metadaten des Bereichs anzuzeigen und ggf. das Hinzufügen von Informationen zu ermöglichen.

### Software-Schnittstellen

*Welche Schnittstellen zu andere Software sind nötig und wie sollen diese umgesetzt werden (Datenbanken, Bibliotheken etc.). Welche Dienste werden benötigt und wie sollen diese eingebunden werden?*

1. Datenbank für das Speichern der Metadaten

Um die Informationen zu den einzelnen Audiodateien und deren Abschnitte zu speichern, wird eine Datenbank benutzt. Diese speichert anhand einer ID die jeweilige Audiodatei, die einzelnen Tracks und die dazugehörigen Metadaten. Durch die Verknüpfung in der Datenbank können auf diese Weise Daten abgerufen und hinzugefügt/abgespeichert werden.

1. Software Bibliotheken

Des Weiteren werden auch unterschiedliche Bibliotheken in der Software verwendet, um verschiedene Darstellungen und Funktionen des Tools selber zu ermöglichen.

* Wavesurfer.js: Diese Javascript Bibliothek ermöglicht die Darstellung der Wellenform der Audiodaten
* Jquery: Diese Javascript Bibliothek ermöglicht Animationen des Tools
* BBC Bibliothek : Diese Javascript Bibliothek enthält einige Animationen und Funktionen, die sich unter anderem für das Verarbeiten von Audiodaten eignen.

Mit Hilfe dieser einzelnen Bibliotheken und darin enthalten Funktionen sollen die Audiodatenverarbeitung und Nutzerinteraktionen implementiert werden.

### Kommunikationsschnittstellen

*Beschreiben Sie alle Kommunikationsfunktionen, die für das Produkt benötigt werden (z.B. E-Mail, Server, Kommunikationsprotokolle etc.). Identifizieren Sie anzuwendende Standards (z.B. FTP, HTTP etc.). Auf welcher Basis sollen die Daten ggf. verschlüsselt werden (Standards etc.).*

1. File Transfer Protocol

Um die einzelnen Metadaten der jeweiligen Audiodatei abzurufen bzw. für den markierten Abschnitt hinzuzufügen, wird mit Hilfe des FTP mit der Datenbank kommuniziert. Diese stellte die Informationsdaten zur Verfügung.

1. Hypertext Transfer Protocol

Mit Hilfe des http wird das Audiotool im Browser geladen. Dabei wird eine Anfrage an den Host gerichtet, der gewünschten Link zurückgegeben und angezeigt.

## Funktionale Anforderungen

1. Upload einer Audiodatei

Das Tool ermöglicht eine Hoerburger Audiodatei in den Browser hochzuladen. Dies kann über Drag & Drop oder dem Standard HTML Upload geschehen. Abbildung 4 zeigt den Standard HTML Upload.



Abbildung . Standard HTML Upload (<!selfHTML.wiki/>, n.d.)

1. Visualisierung der hochgeladenen Audiodatei in Amplitudenform

Sobald eine Audiodatei im WAV-Format hochgeladen wird, wird automatisch im Browser die dazugehörige Wellenform angezeigt. Auf diese Weise sind die einzelnen Amplituden des Audiosignals ersichtlich.

1. Visualisierung der bereits markierten/vorhandenen Tracks

In der Wellendarstellung soll automatisch, durch Zugriff auf die Datenbank, die Darstellung der einzelnen Tracks ermöglicht werden. Farbig markierte Trennstriche in der Audiowelle markieren einen Abschnitt, der in der Datenbank als Track gespeichert ist. Jeweils gleichfarbige Trennstriche kennzeichnen einen Track. Die Farben wechseln für die unterschiedlichen Abschnitte.

1. Markieren von Tracks

Sobald sich der Nutzer mit der Maus über die Wellenform begibt, kann er Abschnitte, in Form der in Punkt 3 bereits erwähnten Trennstriche, kennzeichnen. Damit sich durch die Markierungen keine Tracks überschneiden bzw. ein Track mittendrin abgeschnitten wird, werden Plausibilitätsabfragen in der Programmierung verwendet. Durch diese wird der Nutzer in bestimmten Fällen keine Markierungen setzen dürfen:

* Zwischen einem bereits markiertem Track
* Bei gewissen Amplitudenausschlägen

Folglich wird mit Hilfe der Amplitude ermittelt, wann ein Track eventuell zu Ende sein könnte. Beispielsweise eignen sich Pausen in der Aufnahme (kein bis sehr geringer Amplitudenausschlag) als Indikatoren für ein Trackende bzw. Trackanfang.

Ist ein Trackanfang/ Trackende markiert, kann nun die Wellenform nach rechts/links gescrollt werden, um den nächsten Strich zu setzen. Bei nicht erlaubter Trennstrichsetzung erscheint eine Fehlermeldung und die unerlaubte Trennmarkierung wird gelöscht.

1. Visualisierung der Informationen der einzelnen Abschnitte einer kompletten Audiodatei

Links neben dem Bereich der dargestellten Audiowellen befindet sich ein Feld mit den bereits markierten Tracks. Klickt man auf einen dieser Tracks, wird der dazugehörige Audioteil abgespielt. Unter der Wellendarstellung öffnet sich ein Feld, welches bereits in der Datenbank gespeicherte Informationen zu diesem Abschnitt anzeigt.

1. Hinzufügen von Informationen

Ein Button unter dem Informationsfeld ermöglicht dem Nutzer neue Informationen zu diesem Abschnitt hinzuzufügen und durch einen weiteren Button zu bestätigen. Dadurch werden die hinzugefügten Daten in der Datenbank gespeichert und ab diesem Zeitpunkt jedermann in diesem Feld für diesen Abschnitt angezeigt.

1. Speichern / Löschen von Informationen und gesetzten Tracks

Aktuelle Änderungen können durch die "Rückgängig"- Taste im linken oberen Eck widerrufen werden. Solange der in Punkt 6 beschriebener Bestätigungsbutton nicht geklickt wird, wird nichts in der Datenbank gespeichert und ist bei Schließen des Browsers wieder gelöscht. Erst durch den Bestätigungsbutton werden Änderungen übernommen. Alle bereits übernommenem Änderungen und in der Datenbank gespeicherte Metadaten können nicht durch den Nutzer eliminiert bzw. verändert werden.

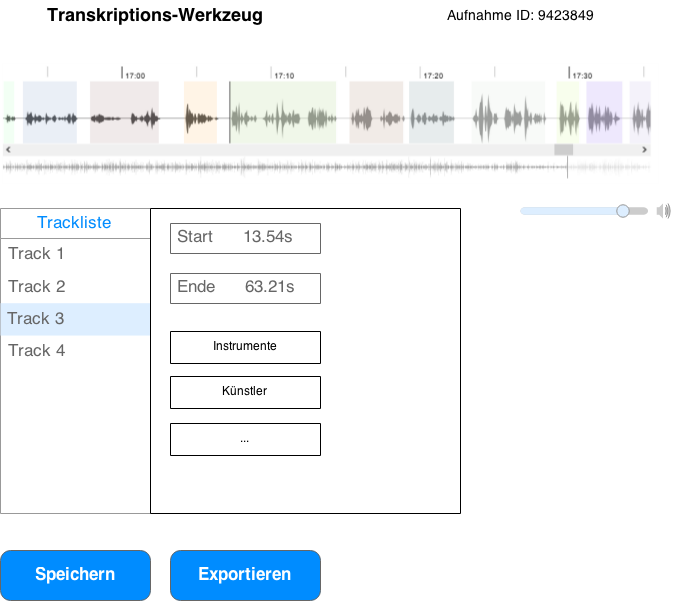


Abbildung . Erstes Mockup des Transkriptions-Tools

## Anwendungsfälle (Use Cases)

*Beschreiben Sie hier im Detail die Anwendungsfälle, die Ihr Produkt leisten soll. Welche Ziele sollen mit Ihrem System erreicht werden können (vgl. auch User Stories)? Wie sollen diese umgesetzt werden? Bitte nehmen Sie auch eine Priorisierung vor. Gerne können Sie zusätzlich die Modellierung in UML zur Veranschaulichung nutzen. Verwenden Sie bitte auf jeden Fall das Template, welches in abgebildet ist. zeigt ein Beispiel.*

1. *Nutzerinteraktion*

|  |  |
| --- | --- |
| Use Case 01 | Upload der Audiodatei |
| Priorisierung | \*\*\* |
| Beschreibung | Der Nutzer muss die Audiodatei (.WAV) in das Tool hochladen |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Drag & Drop |
| Flow of Events  (Interaktionsschritte) | Der Nutzer ruft die Seite auf. Er wird dazu aufgefordert seine Datei in einen markierten Bereich zu laden. Feedback begleitet den Uploadvorgang. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) | Standardupload („Datei hinzufügen“- Button) |
| Nicht-funktionale Anforderung | Der Uploadvorgang soll nicht zu lange dauern! (Dateigröße) |

|  |  |
| --- | --- |
| Use Case 02 | Setzen von Markierungen |
| Priorisierung | \*\*\* |
| Beschreibung | Der Nutzer kann die Audiodatei in Abschnitte unterteilen ( Mausklick für Trackanfang und weiterer Mausklick für Trackende ) |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Mausklick |
| Flow of Events  (Interaktionsschritte) | Durch einen Mausklick auf die gewünschte Amplitude kann der/ das Trackanfang/ Trackende gesetzt werden. Durch weiteren Mausklick auf einen anderen Ort in der Wellendarstellung das / der Trackende/ Trackanfang |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) | Bei fehlerhafte Tracksetzung wird der Vorgang rückgängig gemacht und neue Markierungen können hinzugefügt werden:   1. Wird ein Track zwischen einem bereits markierten Track gesetzt 2. Wird eine Markierung mitten im Track gesetzt (Amplitude schlägt aus). |
| Nicht-funktionale Anforderung |  |

|  |  |
| --- | --- |
| Use Case 03 | Anzeigen der Metainformationen |
| Priorisierung | \*\*\* |
| Beschreibung | Der Nutzer kann die Informationen zu jedem Track einsehen |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Mausklick |
| Flow of Events  (Interaktionsschritte) | Durch Mausklick auf einen angezeigten Track (Liste mit allen Tracks der Audiodatei) neben der Wellendarstellung, öffnet sich darunter ein Bereich mit den einzelnen Informationen. Der geklickte Track wird abgespielt. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) | Ein Klick auf den bereits gesetzten Trennstrich ermöglich ebenfalls die gleiche Funktion |
| Nicht-funktionale Anforderung |  |

|  |  |
| --- | --- |
| Use Case 04 | Hinzufügen von Informationen zu gesetztem Track |
| Priorisierung | \*\*\* |
| Beschreibung | Der Nutzer kann Informationen zu einer Track hinzufügen |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Button Klick („ Information hinzufügen“) |
| Flow of Events  (Interaktionsschritte) | Wird ein Track angeklickt, kann im unteren „Information“ – Bereich durch Klick auf den dazugehörigen Button weitere Information hinzugefügt werden. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) |  |
| Nicht-funktionale Anforderung | Die Berechnung der Wellenform soll zügig durchgeführt werden ( Dateiupload nimmt selber schon Zeit in Anspruch ) |

|  |  |
| --- | --- |
| Use Case 05 | Abspeichern der Daten |
| Priorisierung | \*\*\* |
| Beschreibung | Speichern der Informationen und gesetzten Tracks |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Button Klick („Speichern“) |
| Flow of Events  (Interaktionsschritte) | Durch Klick auf den Speicherbutton werden alle Veränderungen übernommen und in der Datenbank gespeichert. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) |  |
| Nicht-funktionale Anforderung | Das übermitteln der Daten soll schnell durchgeführt werden |

|  |  |
| --- | --- |
| Use Case 06 | Aktion rückgängig machen |
| Priorisierung | \*\* |
| Beschreibung | Veränderungen wieder rückgängig machen |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Button Klick („Rückgängig“) |
| Flow of Events  (Interaktionsschritte) | Durch Klick auf den Rückgängigbutton wird die letzte Veränderung gelöscht. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) | Manuelles Löschen der Veränderung |
| Nicht-funktionale Anforderung |  |

|  |  |
| --- | --- |
| Use Case 07 | Tracks löschen |
| Priorisierung | \*\*\* |
| Beschreibung | Noch nicht in der Datenbank gespeicherte Tracks löschen |
| Hauptakteur | Nutzer |
| Auslöser (Trigger) | Mausklick (rechts) + Delete |
| Flow of Events  (Interaktionsschritte) | Durch Klick der rechten Maustaste auf den gewünschten Trennstrich und dann „Delete“ wird der Track gelöscht. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) | Durch Button „Tracks löschen“ gewünschten Markierungsstrich durch Klick der linken Maustaste entfernen. |
| Nicht-funktionale Anforderung |  |

1. *Tool / Datenbank*

|  |  |
| --- | --- |
| Use Case 08 | Berechnung der Wellenform |
| Priorisierung | \*\*\* |
| Beschreibung | Das Tool berechnet mit Hilfe von Bibliotheken (Wavesurfer.js, BBC ) die Amplitudenform der Audiodatei. |
| Hauptakteur | Tool |
| Auslöser (Trigger) | Dateiupload |
| Flow of Events  (Interaktionsschritte) | Nach dem Dateiupload berechnet das Tool automatisch die Wellenform |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) |  |
| Nicht-funktionale Anforderung |  |

|  |  |
| --- | --- |
| Use Case 09 | Abruf der Daten aus der Datenbank |
| Priorisierung | \*\*\* |
| Beschreibung | Das Tool greift auf die in der Datenbank gespeicherten Daten zu. |
| Hauptakteur | Tool, Datenbank |
| Auslöser (Trigger) | Wellenform der Audiodatei ist berechnet. |
| Flow of Events  (Interaktionsschritte) | Nach der Berechnung der Wellenform, greift das Tool auf die Datenbank zu und holt sich die zur Audiodatei passenden Metadaten (gesetzte Tracks, Informationen). Feedback begleitet weiterhin diesen Prozess. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) | Direkt nach dem Dateiupload werden die Daten aus der Datenbank angefordert |
| Nicht-funktionale Anforderung |  |

|  |  |
| --- | --- |
| Use Case 10 | Anzeigen der Wellendarstellung |
| Priorisierung | \*\*\* |
| Beschreibung | Das Tool visualisiert die Amplitudenform der hochgeladenen Audiodatei |
| Hauptakteur | Tool |
| Auslöser (Trigger) | Dateiupload |
| Flow of Events  (Interaktionsschritte) | Nach dem Dateiupload und der Datenbeschaffung aus der Datenbank, visualisiert das Tool die Wellendarstellung mit den dazugehörigen Tracks und Metainformationen. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) |  |
| Nicht-funktionale Anforderung | Die Berechnung der Wellenform soll zügig durchgeführt werden ( Dateiupload nimmt selber schon Zeit in Anspruch ) |

|  |  |
| --- | --- |
| Use Case 11 | Speichermeldung |
| Priorisierung | \*\*\* |
| Beschreibung | Beim Verlassen des Tools wird ein Warnhinweis angezeigt. |
| Hauptakteur | Tool |
| Auslöser (Trigger) | Tool verlassen |
| Flow of Events  (Interaktionsschritte) | Wird das Tool durch eine neue URL oder den Schließen Button des Browsers verlassen, wird davor eine Warnung angezeigt, die dem Benutzer an das Speichern der Daten erinnern soll. |
| Alternative Flow/ Erweiterungen  (Variierende Interaktionsschritte) |  |
| Nicht-funktionale Anforderung |  |

# Nicht-funktionale Anforderungen

## Performance

*Welche Voraussetzung hinsichtlich der Performanz müssen gewährleistet werden, damit das Produkt reibungslos funktioniert? Gibt es Faktoren, die dies beeinflussen?*

Dafür dass das System als Webanwendung verwendet werden soll, ist von einer Echtzeit-Bearbeitung auszugehen. Man geht aber trotzdem von einer bestimmten Latenz aus, wegen Abhängigkeiten vom Dritte (Datenbank, Server etc.). Das Tool soll mit große Audio-Dateien arbeiten (>600 MB WAV Dateien). Die Performanz des Tools soll mit Hilfe der Anwendung neuester Web-Technologien (Web Audio usw.) gewährleistet werden.

## Sicherheit

*Welche Anforderungen an die Sicherheit müssen gestellt werden, um z.B. Datensicherheit oder -Verlust zu verhindern. Bedarf es einer Datenverschlüsselung? Welche Maßnahmen müssen für die Gewährleistung der Sicherheit eingeplant werden? Welche Auswirkungen auf Implementierung oder Design sind absehbar? Gibt es bestimmte Zertifikate, die hier Verwendung finden?*

Die Metadaten, welche der Benutzer in der Datenbank über das Tool einfügen soll, sollten die üblichen Datenschutz-Richtlinien respektieren. Diese werden über die Anmeldepflicht mit Kennwort des Nutzers ermöglicht.

## Qualität

### Ausfallsicherheit

Das System soll versuchen den Ausfall zu verhindern. Wenn das nicht möglich ist, soll der Verlust der zum Zeitpunkt des Ausfalls eingegeben Daten verhindert werden. Dies wird erreicht, indem die Daten schon bei der Eingabe zwischengespeichert werden.

### Erreichbarkeit

Die Systemerreichbarkeit sollte auch gewährleistet werden. Dies ist aber vom Server abhängig, deswegen kann hier die Sicherheit nicht gegeben werden. Falls nicht möglich, soll der Benutzer davon benachrichtigt werden.

### Wartbarkeit

Das System soll eine leichte Wartung dessen ermöglichen, um Fehler beheben zu können. Dies wird gewährleistet, indem das Programmiercode frei zugänglich über die die GitHub Plattform gemacht wird.

### Erweiterbarkeit

Das System soll Platz auch für Erweiterungen lassen, falls andere Entwickler diesen erweitern möchten. Dafür das die Software quelloffen ist, sollte diese Qualität gewährleistet sein.

### Usability

Die Software Ergonomie des Systems ist für den Benutzer entscheidend. Dieses Merkmal wird durch umfangreiche Usability Tests erreicht.

## Datenmodell

*Welche Anforderungen sind an das Datenmodell zu stellen? Welche Art von Modell soll eingesetzt werden?*

Da man mit Datenbank-Systeme zu tun haben werden, sollten bestimmte Relationen oder Abhängigkeiten zwischen die Datenelemente geklärt werden.

Als Datenmodelle werden sowohl konzeptuelle Datenbankschemas (z.B. ER-Diagramme, UML-Diagramme), logische als auch physische Datenbankschemas eingesetzt. Ob hierarchische-, Netzwerk-, relationale- oder objekt-relationale Modelle verwendet werden, wird im Laufe des Projekts geklärt werden.

## Sonstige Anforderungen

*Welche weiteren Anforderungen (egal welcher Art) gibt es an das zu entwickelnde Produkt?*