



Abschlussprüfung Sommer 2018 / 2019

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Vereinheitlichung und Digitalisierung der Materialeingangsprüfungsberichte mithilfe einer zentralen Datenbankanbindung im Synchronbereich

Auszubildender:

Rico Krüger

Beermannstr. 18

12435 Berlin

BERLINER SYNCHRON
WENZEL LÜDECKE 
• EIN UNTERNEHMEN DER S&L MEDIEN GRUPPE

Ausbildungsstätte

Berliner Synchron GmbH

EUREF-Campus 10-11

10829 Berlin

Abgabetermin:

Berlin, den 15.06.2018

Dieses Werk, einschließlich seiner Teile, ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.



Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Listings	V
Abkürzungsverzeichnis	VI
1 Einleitung	1
1.1 Projektumfeld	1
1.2 Projektziel	1
1.3 Projektbegründung	2
1.4 Projektschnittstellen	2
1.5 Projektabgrenzung	3
2 Projektplanung	3
2.1 Projektphasen	3
2.2 Zeitplanung	3
2.3 Abweichungen vom Projektantrag	4
2.4 Ressourcenplanung	4
2.5 Entwicklungsprozess	4
3 Analysephase	4
3.1 Ist-Analyse	4
3.2 Wirtschaftlichkeitsanalyse	5
3.2.1 „Make or Buy“-Entscheidung	5
3.2.2 Projektkosten	5
3.2.3 Amortisationsdauer	6
3.3 Nutzwertanalyse	6
3.4 Qualitätsanforderungen	6
3.5 Lastenheft	7
3.6 Zwischenstand	7
4 Entwurfsphase	8
4.1 Zielplattform	8
4.2 Architekturdesign	8
4.3 Entwurf der Benutzeroberfläche	8
4.4 Datenmodell	9
4.5 Fachkonzept	9
4.6 Maßnahmen zur Qualitätssicherung	9
4.7 Pflichtenheft	9



Inhaltsverzeichnis

4.8	Zwischenstand	10
5	Implementierungsphase	10
5.1	Implementierung der Projektumgebung	11
5.2	Implementierung des Webservers und des Routings	11
5.3	Implementierung der Datenbank	11
5.4	Implementierung der Model-Klassen	11
5.5	Implementierung des Fachkonzepts	12
5.6	Implementierung der Benutzeroberfläche	12
5.7	Zwischenstand	12
6	Qualitätssicherung und Abnahme	12
6.1	Zwischenstand	13
7	Einführungsphase	13
7.1	Geplante Einführung	13
7.2	Benutzerschulung	13
7.3	Zwischenstand	13
8	Dokumentation	13
8.1	Benutzerdokumentation	14
8.2	Zwischenstand	14
9	Fazit	14
9.1	Soll-/Ist-Vergleich	14
9.2	Gewonnene Erkenntnisse	15
9.3	Ausblick	15
	Literaturverzeichnis	16
	Eidesstattliche Erklärung	17
A	Anhang	i
A.1	Zeitplanung	i
A.2	Projekt-Ressourcen	ii
A.3	Mockups	iii
A.4	Datenbankentwurf	v
A.5	Testprotokoll	vi
A.5.1	Routes.rb	vii
A.5.2	Migration.rb	vii
A.5.3	Model.rb	viii
A.5.4	Controller.rb	ix
A.5.5	View.html	xi
A.6	Auszug Benutzerdokumentation	xiv

Abbildungsverzeichnis

1	Mockup - Übersicht der Materialeingangsberichte	iii
2	Mockup - Eingabeformular eines Materialeingangsberichts	iv
3	Entwurf der MySQL - Datenbank	v
4	Auszug aus dem Testprotokoll	vi

Tabellenverzeichnis

1	Zeitplanung	3
2	Projektkosten	6
3	Zwischenstand nach der Analysephase	7
4	Zwischenstand nach der Entwurfsphase	10
5	Zwischenstand nach der Implementierungsphase	12
6	Zwischenstand nach der Abnahmephase	13
7	Zwischenstand nach der Einführungsphase	13
8	Zwischenstand nach der Dokumentation	14
9	Soll-/Ist-Vergleich	14

Listings

Listings/routes.rb	vii
Listings/migration.rb	vii
Listings/model.rb	viii
Listings/controller.rb	ix
Listings/view.html.haml	xi

Abkürzungsverzeichnis

IHK	Industrie- und Handelskammer
BSG	Berliner Synchron GmbH
MEP	Materialeingangsprüfungsbericht
QC	Quality Check
Ruby	Objektorientierte Programmiersprache
Framework	Programmiergerüst
Rails	Framework für Webanwendungen
RoR	Ruby on Rails
SQL	Structured Query Language: Programmiersprache für Datenbankabfrage
MySQL	My Structured Query Language: Relationale Datenbank
HTTP	Hypertext Transfer Protocol: Protokoll zur Datenübertragung
mysql2	MySQL -Datenbank-Connector
gem	Bezeichnung für Bibliotheken und Frameworks in Ruby
HTML	Hypertext Markup Language: Programmiersprache zur Strukturierung einer Webseite
HAML	HTML abstraction markup language: Programmiersprache zur Generierung von HTML-Code; Erlaubt u.a. die Ausführung von Ruby-Code
CSS	Cascading Style Sheets: Programmiersprache zum Designen von Webseiten
Javascript	Scriptsprache zum Erstellen dynamischer Webseiten
Bootstrap	HTML -, CSS -, Javascript-Framework
Model	Datenmodell
View	Darstellung
Controller	Programmsteuerung
MVC	Model-View-Controller : Architekturdesign zur Trennung der Software in drei Komponenten: Model , View und Controller
CRUD	Create Read Update Delete
GUI	Graphical User Interface

Abkürzungsverzeichnis

rvm	Ruby Version Manager
IDE	Integrated Development Environment

1 Einleitung

1.1 Projektumfeld

Ausbildungsbetrieb: Die Berliner Synchron GmbH (**BSG**) ist ein mittelständisches, international agierendes Unternehmen, dessen Kerngebiet die deutsche Synchronisation englischsprachiger Filme und Serien ist. 1949 als erstes Synchronunternehmen Deutschlands gegründet, hält die **BSG** nach wie vor eine Spitzenstellung innerhalb der Synchronbranche. Seit 2016 ist das Unternehmen Teil der S&L Medien Gruppe GmbH, die es sich zum Ziel gesetzt hat, ein allumfassendes Unterhaltungsmarketing zu bieten. Neben den rund 60 festen Mitarbeitern, greift die **BSG** auf einen Pool aus über 3 000 Synchronschauspielern, Autoren, Regisseuren, Cuttern und Takern zurück. Die interne IT-Abteilung des Unternehmens beschäftigt derzeit eine Vollzeitkraft und einen Auszubildenden. Unterstützt wird diese durch die IT-Abteilung der S&L Medien Gruppe sowie durch einen technischen Dienstleister vor Ort. Das Aufgabenfeld der IT-Abteilung umfasst den Aufbau und die Wartung des firmeninternen Netzwerks, die Administration der eigenen Server, die Entwicklung und Pflege der in der Firma eingesetzten Systeme sowie die alltägliche Benutzerunterstützung.

Auftraggeber: Der Auftraggeber ist in diesem Projekt der Ausbildungsbetrieb – die **BSG**.

Projektbeschreibung: Derzeit befindet sich eine neue Software – Copper in der Testphase. Diese wird von einem externen Dienstleister in Zusammenarbeit mit der **BSG** programmiert. Mit Copper sollen alle Tätigkeiten von der Projekterstellung über die Projektumsetzung bis hin zur Projektauswertung gesteuert und alle betrieblichen Prozesse über ein Webinterface zentralisiert dargestellt werden. Eine zusätzliche Webanwendung soll für die neue Software entwickelt werden. Diese Webanwendung soll ein Eingabeformular für einen Materialeingangsprüfungsbericht (**MEP**) bereitstellen und die vom Benutzer eingegebenen Daten in eine zentrale Datenbank speichern.

1.2 Projektziel

Projekthintergrund: Die Synchronisation von Filmen und Serien wird jeweils als ein Projekt bezeichnet. Die Akte eines Kinofilms bzw. die Episoden einer Serie bilden dabei einzelne Teilprojekte. Um diese Teilprojekte in deutscher Sprache aufzunehmen, erhält die Berliner Synchron GmbH von seinen Kunden, meist ausländische Produktionsstudios, sowohl das zu synchronisierende Video- als auch diverse Audiomaterialien. Dabei handelt es sich um unterschiedliche Audiotypen wie Originalton, Musik, Effekte und andere, von denen es wiederum unterschiedliche Versionen gibt. Bevor das gelieferte Material in den Produktionsprozess integriert werden kann, muss dieses analysiert werden. Dabei wird es auf Fehler überprüft und die Qualität des angelieferten Materials wird bewertet. Diese Bewertung kann dazu führen, dass das Audiomaterial überarbeitet oder sogar erneut vom Kunden zugesandt werden muss. Bisher wird das Ergebnis dieser Prüfung formfrei, meist in Word-Dateien,

1 Einleitung

verfasst. Danach werden diese in den entsprechenden Projektordnern gespeichert und die betreffenden Projektbeteiligten per E-Mail informiert.

Ziel des Projekts: Ziel dieses Projektes ist die Entwicklung einer einfachen und leicht bedienbaren Webanwendung zur Erfassung, Darstellung und zentralen Speicherung der Daten der Materialeingangsprüfung und deren direkter Zuordnung zu einem Teilprojekt. Alle Mitarbeiter sollen nach einer kurzen Einführung in der Lage sein, ein solches Formular problemlos ausfüllen zu können.

1.3 Projektbegründung

Nutzen des Projekts: Durch ein standardisiertes Formular mit zum Großteil vorgegebenen Werten über Auswahlfeldern soll die Erstellung eines solchen Meldeberichtes erleichtert und damit beschleunigt werden. Das Projekt setzt auf einen weiteren wichtigen Schwerpunkt: dem Controlling. Neben der Kostenerfassung für den Materialeingangsbericht durch Copper, lässt die zentralisierte Erfassung der Daten und die direkte Zuordnung zu den Projekten und Kunden auch eine übersichtliche und gefilterte Darstellung aller **MEPs** zu. Damit lassen sich Rückschlüsse auf anfallende Kosten für zukünftige Projekte ziehen. Schickt ein Kunde regelmäßig fehlerhaftes Material, verzögert sich die Produktion und die Kosten steigen. Das Erstellen eines solchen Berichts ist auf Grund seines Umfangs nicht Teil dieses Projekts und wird erst in einem Nachfolgeprojekt entwickelt. Die Erstellung und Speicherung der Daten eines Materialeingangs stellt lediglich die Grundlage dar.

Motivation: Der Auftraggeber ist daran interessiert, ein allumfassendes System für die Projektabwicklung zu haben, welches alle Projektschritte dokumentiert und darstellt. Zu diesem Zweck wurde die Software Copper in Auftrag gegeben. Sie wird von einem externen Dienstleister entwickelt und nach erfolgreicher Einführung von der eigenen IT-Abteilung erweitert werden.

1.4 Projektschnittstellen

Copper ist eine Webapplikation, die mit **Ruby on Rails (RoR)** erstellt wurde. Da diese Anwendung später auf einem firmeneigenen Server ausgeführt und hauptsächlich im lokalen Netzwerk eingesetzt wird, soll diese vor allem über den in der **BSG** eingesetzten Webbrowser – Google Chrome¹ ausführbar sein. Zur Erstellung der Webseite wird **Bootstrap** verwendet. Die browserbasierten Nutzereingaben werden dann mittels **HTTP** von einem Apache-Webserver entgegen genommen und an die **Rails**-Applikation übermittelt. Dort werden die Anfragen bearbeitet und beantwortet, sowie ggf. in eine lokale **MySQL**-Datenbank geschrieben oder ausgelesen. Die Anwendung ist vor allem für die Quality Check (**QC**)-Abteilung im Unternehmen gedacht. Die Verantwortliche der **QC**-Abteilung sowie der Ausbilder sind für die Projektabnahme verantwortlich.

¹<https://www.google.de/chrome/>

1.5 Projektabgrenzung

Was dieses Projekt nicht bietet: Eine automatisierte Auswertung sowie sonstige Zusammenfassungen außerhalb eines Teilprojektes werden nicht vorgenommen. Dieses Projekt stellt lediglich die Grundlage dar. Zudem wird die Software während des Projektes noch nicht in Copper integriert. (Siehe [Siehe Abweichung vom Projektantrag](#)).

2 Projektplanung

Der Projektplanung ging ein Meeting zwischen der Geschäftsführung, einer Mitarbeiterin, welche zum Großteil Materialeingangsberichte erfasst und der IT-Abteilung voraus. Nachdem die Geschäftsführung sich nach einer Zusammenfassung der Materialeingangsprüfungsberichte für einen bestimmten Kunden erkundigte bzw. nach der Möglichkeit einer solchen, wurde schnell klar, dass dies auf Basis der aktuellen Arbeitsweise nicht gewährleistet werden kann. Für eine übersichtliche, gefilterte Darstellung solcher Berichte würde eine gewisse Standardisierung und zentrale Speicherung der Daten benötigt. Auf dieser Grundlage entstand die Idee eine Erweiterung für Copper zu entwickeln.

2.1 Projektphasen

Aufgrund des von der Industrie- und Handelskammer ([IHK](#)) vorgegebenen begrenzten Zeitrahmens, wurde das Projekt in ein Teilprojekt gewandelt. Nach der positiven Rückmeldung auf meinen Projektantrag wurde das Projekt in den Tagesablauf meiner betrieblichen Arbeit integriert und mit ca. 30 Wochenstunden bearbeitet.

2.2 Zeitplanung

Für die Umsetzung des Projektes stehen seitens der Anforderungen der [IHK](#) 70 Stunden zur Verfügung. Diese wurden zur Antragstellung auf die einzelnen Phasen verteilt. Die grobe Zeitplanung der Hauptphasen kann nachfolgender Tabelle 1 Zeitplanung entnommen werden. Eine ausführlichere Zeitplanung findet sich im Anhang [A.1: Zeitplanung](#) auf Seite [i](#).

Projektphase	Geplante Zeit
Analyse	6 h
Entwurf	14 h
Implementierung	37 h
Abnahme und Einführung	2 h
Dokumentation	11 h
Gesamt	70 h

Tabelle 1: Zeitplanung

3 Analysephase

2.3 Abweichungen vom Projektantrag

Bei der Erstellung des Projektantrags wurde geplant, das Projekt auf einer bereits vorhandenen Testumgebung aufzusetzen und in Absprache mit dem Entwickler von Copper anzufertigen. Da der Entwickler von Copper während der Projektarbeit weder erreichbar, noch ein Zugriff auf die Testumgebung vorhanden war, musste das Projekt mit den zur Verfügung stehenden Daten zur Zeit der Erstellung des Projektantrags nachgebaut werden. Somit ergaben sich erhebliche Änderungen im Vergleich zum Projektantrag. Ferner konnte aufgrund der am Ende fehlenden Zeit durch die vorhergehenden, nicht eingeplanten Entwicklungsarbeiten keine automatisierten Tests des entwickelnden Materialeingangsberichts mehr durchgeführt werden. Das hätte das zusätzliche Aufsetzen einer Testumgebung bedurft, welche am Ende nicht einmal den tatsächlichen Produktionseinsatz der Anwendung sichergestellt hätte. Daher wurde in Absprache mit dem Ausbilder beschlossen, lediglich ein Testprotokoll zu erstellen. Weiterhin wurde auf die Erstellung einer Entwicklerdokumentation verzichtet.

2.4 Ressourcenplanung

Alle benötigten Ressourcen zur Durchführung des Projekts sind bereits im Unternehmen vorhanden bzw. können online heruntergeladen werden. Eine Auflistung befindet sich im Anhang **A.2: Projekt-Ressourcen** auf Seite **ii**.

2.5 Entwicklungsprozess

Die Entwicklung des Projekts erfolgt nach Rücksprache mit der verantwortlichen Mitarbeiterin für Qualitätsprüfung im Unternehmen. Ansonsten wird das Projekt weitestgehend nach dem Wasserfallmodell in Eigenregie erstellt.

3 Analysephase

Die erste Analysephase wurde bereits zum Zeitpunkt der Antragstellung des **IHK**-Projektes durchgeführt. Dort wurde bereits mit dem externen Entwickler gesprochen und auf der Grundlage eines Skripts, geschrieben in der Datenbanksprache **SQL**, ein vorläufiges Datenbankmodell für das Projekt erstellt und eingegrenzt.

3.1 Ist-Analyse

In der **BSG** wird sowohl das eingehende als auch das ausgehende Audio- und Videomaterial analysiert und auf Fehler überprüft. Bei der Analyse werden neben möglichen Fehlern auch Meta-Daten wie das Materialeingangsdatum, Audioformat oder die Version gesammelt. Zudem wird entschieden, ob das Material nochmal überarbeitet oder gar vom Kunden neu übermittelt werden muss, bevor es in die

3 Analysephase

Produktion geht. Eine solche Prüfung wird bisher formlos, meist in Word-Dateien geschrieben. Danach werden die für das Projekt verantwortlichen Mitarbeiter per E-Mail über die Fertigstellung und auf mögliche Mängel hingewiesen. Bei einem solchen Bericht gibt es viele Informationen, welche nur einen begrenzt möglichen Datensatz aufweisen. Weiterhin werden dann die Berichte in den entsprechenden Netzwerkordnern der einzelnen Teilprojekte gespeichert. Das Erhalten eines Überblicks über die Materialeingänge und deren Fehler für ein Teilprojekt ist somit äußerst schwierig. Will man einen solchen Überblick über ein komplettes Projekt oder gar für einen Kunden erhalten, ist dies nahezu unmöglich. Dies erschwert die Kostenberechnung und das Ziehen von Rückschlüssen für künftige Projekte.

3.2 Wirtschaftlichkeitsanalyse

Die Wirtschaftlichkeit des neuen standardisierten Berichts zur Qualitätssicherung steht außer Frage. Die derzeitige Darstellung und Bearbeitung ist unübersichtlich und ineffizient. Die Vielzahl der Erstellung von Berichten würde sich in jedem Fall mittelfristig gesehen, refinanzieren. Mit der Inbetriebnahme der neuen Software sollen zudem alle übrigen Projektprozesse dargestellt und deren Kosten ermittelt werden. Warum also die Erstellung eines solchen Berichts ausschließen? Das würde wiederum das Controlling erschweren und auch für diese Kostenstelle einen zusätzlichen Zeit- und damit Kostenaufwand bedeuten.

3.2.1 „Make or Buy“-Entscheidung

Da die Synchronbranche sehr speziell ist und es keine standardisierte Arbeitsweise oder Projektsoftware für diesen Bereich gibt, entschied sich die **BSG** einen externen Dienstleister mit der Entwicklung einer Software, zugeschnitten auf die Bedürfnisse des Unternehmens, zu engagieren. Von Anfang an war klar, dass dieses Projekt, selbst nach Einführung, von der unternehmenseigenen IT-Abteilung stetig weiterentwickelt werden soll und muss. Da die neue Software noch in den Händen des Entwicklers ist, stellt sich nur die Frage, ob dieser den neuen Bericht erstellt oder der firmeneigene Mitarbeiter / Auszubildende. Aus Zeit- und Kostengesichtspunkten ist der Auszubildende für das Unternehmen am geeignetsten.

3.2.2 Projektkosten

Die Kosten zur Durchführung des Projekts ergeben sich aus den Gehältern der beteiligten Mitarbeiter, die für die Mitarbeiter aufzuwendenden Sozialabgaben, sowie die Gemeinkosten sonstiger Ressourcen. Bei 20 Arbeitstagen monatlich erhält der Auszubildende 884 € brutto. Der Arbeitgeberanteil zur Sozialversicherung beträgt dabei 229,53 €. Für den Personalaufwand der übrigen Mitarbeiter wird mit einer Pauschale von 30 € / Stunde gerechnet.

$$8 \text{ h/Tag} \cdot 20 \text{ Tage/Monat} = 160 \text{ h/Monat} \quad (1)$$

3 Analysephase

$$\frac{884,00 \text{ €} + 229,53 \text{ €/Monat}}{160 \text{ h/Monat}} = \frac{1113,53 \text{ €/Monat}}{160 \text{ h/Monat}} = 6,96 \text{ €/h} \quad (2)$$

Es ergibt sich ein Stundenlohn von 6,96 €. Die Durchführungszeit des Projekts beträgt 70 Stunden. Die Nutzung von Ressourcen wird hier mit einem pauschalen Stundensatz von 10 € berücksichtigt. Eine Aufstellung der Kosten befindet sich in Tabelle 2. Die Kosten belaufen sich auf insgesamt 1337,20 €.

Vorgang	Zeit	Kosten pro Stunde	Kosten
Entwicklungskosten	70 h	6,96 €	487,20 €
Fachgespräche	1 h	30 €	30 €
Meeting QC - Mitarbeiterin	2 h	30 €	60 €
Benutzerschulung	0,5 h	30 € · 4	60 €
Ressourcen	70 h	10 €	700 €
Kosten			1337,20 €

Tabelle 2: Projektkosten

3.2.3 Amortisationsdauer

Geht man von einer Zeitersparnis bei der Erstellung von 10 Minuten pro Bericht aus, bei zwei Berichten pro Tag, ergibt sich eine durchschnittliche monatliche Zeitersparnis von 400 Minuten über 6,6 Stunden. Bei der veranschlagten Mitarbeiterpauschale ergibt sich eine monatliche Kostenersparnis von 200 €. Damit wäre dieses Projekt nach ca. 7 Monaten amortisiert. Die tatsächliche Amortisationsdauer des Projektes ist zum jetzigen Zeitpunkt noch nicht absehbar, da es nur den ersten Teil des Projekts zur Qualitätssicherung darstellt. Dies stellt wiederum auch nur einen Teil des viel größeren Projektes – Copper dar.

3.3 Nutzwertanalyse

Neben der Zeitersparnis und Minimierung der Fehleranfälligkeit liegt der größte Nutzen des Projekts in der Standardisierung der **MEPs**, sowie der späteren Integration in Copper. Auf dieser Grundlage lassen sich die tatsächlich aufgetreten Projektkosten noch genauer kalkulieren und es ist möglich Rückschlüsse auf Kosten zukünftiger Projekte zu ziehen.

3.4 Qualitätsanforderungen

Die Webanwendung soll intuitiv und leicht bedienbar für die im Tonbereich qualifizierten Mitarbeiter sein. Schlüsseldatensätze für spätere Auswertungen sollen auswählbar sein. Zudem muss es für etwaige Fehler Kommentarfelder für genauere Erläuterungen geben. Die Berichte müssen sich eindeutig unterscheiden und einem Teilprojekt zugeordnet sein. Es muss möglich sein, sich die erstellten Berichte

3 Analysephase

anzusehen und diese gegebenenfalls zu editieren. Zudem muss es eine Übersicht für die erstellten **MEPs** eines Teilprojekts geben. Die komplette Ansicht soll über Google-Chrome skalierbar sein.

3.5 Lastenheft

Nachfolgend werden die wichtigsten Anforderungen für das Projekt dargestellt.

Anforderungen an die Benutzungsoberfläche:

- LB10:** Die Berichte müssen online über den Chrome-Webbrowser erreichbar sein.
- LB20:** Die Ansicht muss skalierbar sein.
- LB30:** Die Bedienung und Formularfelder müssen intuitiv und selbsterklärend sein.
- LB40:** Sich wiederholende standardisierte Datensätze müssen auswählbar sein.
- LB50:** Es muss möglich sein zu diversen Datensätze, Kommentare zu schreiben.
- LB60:** Erstellte Berichte müssen editierbar sein.
- LB70:** Für die erstellten Berichte muss es eine Anzeige geben.
- LB80:** Es muss eine Übersicht aller Berichte eines Teilprojekts geben.

Funktionelle Anforderungen:

- LF10:** Die Berichte müssen eindeutig voneinander unterschieden werden können.
- LF20:** Die Navigation zu den Berichten muss vorhanden sein.

Sonstige Anforderungen:

- LS10:** Die Anwendung muss anhand der vorhandenen Information in Copper integrierbar sein.
- LS20:** Das Projekt muss für das Anschlussprojekt erweiterbar sein.

3.6 Zwischenstand

Tabelle 3 zeigt den Zwischenstand nach der Analysephase.

Vorgang	Geplant	Tatsächlich	Differenz
Durchführung einer Ist-Analyse	2 h	3 h	+1 h
Wirtschaftlichkeitsanalyse und Amotisationsrechnung	1 h	1 h	
Erstellung eines Lastenhefts	3 h	3 h	
Analysephase	6 h	7 h	1 h

Tabelle 3: Zwischenstand nach der Analysephase

4 Entwurfsphase

Die meisten, jedoch nicht alle, Entscheidungen über verwendete Technologien stehen bereits mit Projektbeginn fest. Denn sie sind von der sich in der Entwicklung befindenden Software – Copper abhängig. In der Entwurfsphase ist es das Ziel, anhand der zur Verfügung stehenden Informationen einen Designplan für die Anwendung zur späteren Integration in Copper zu entwickeln.

4.1 Zielplattform

Die **RoR** Webanwendung soll unter dem Apple-Betriebssystem macOS laufen. Die Daten werden in einer lokal installierten **MySQL**-Datenbank gespeichert. Die Anbindung zur Datenbank erfolgt über den **mysql2 gem**. Zum lokalen Aufrufen und Testen der Applikation über den Chrome-Browser wird der Thin-Webserver verwendet. Die Realisierung der skalierbaren Webanwendung wird über die Auszeichnungssprachen **HAML**, **CSS** und **Bootstrap** sichergestellt.

4.2 Architekturdesign

Als Architekturdesign wird das in **Rails** üblicherweise eingesetzte Model-View-Controller(**MVC**)-Muster genutzt. Dabei repräsentieren die **Model**-Klassen die Tabellen der Datenbank. Für die Kommunikation zwischen den **Models**, der Datenhaltungsschicht, und der Datenbank sorgt der **mysql2 gem**. Die browserbasierten Darstellungen und Formulare, die **GUI**-Schicht, werden in der **View** repräsentiert. Die Kommunikation und Logik zwischen **View** und **Model** übernimmt der **Controller**. Die ankommenden **HTTP**-Anfragen und Parameter werden über entsprechend definierte Routen an den **Controller** weitergeleitet, welcher dann anhand dieser die sogenannten Create Read Update Delete (**CRUD**)-Operationen durchführt.

4.3 Entwurf der Benutzeroberfläche

Da die Anwendung primär von ausgebildetem Fachpersonal in der Tonbranche verwendet wird, können auch Fachbegriffe oder Abkürzungen verwendet werden. Die Bewertung des Materials sollte jedoch für jedermann verständlich sein. Die Skalierbarkeit der Anwendung wird durch **Bootstrap** realisiert. Da die Anwendung von internem Personal genutzt und in ein anderes Projekt eingepflegt werden soll, wird auf ein ausgeklügeltes, repräsentatives Design verzichtet. Für Eingabefelder mit begrenzten Werten werden Dropdown-Menüs erzeugt. Zur eindeutigen Kennzeichnung jeder Eingabe werden Label verwendet. Die Datumseingabe erfolgt mithilfe eines Datumswählers. Screenshots der Mockups finden sich im Anhang **A.3: Mockups** auf Seite **iii**.

4.4 Datenmodell

Während der Vorbereitung des Projektantrags ließ mir der externe Entwickler einen kleinen Ausschnitt der Datenbank in Form eines **SQL**-Skripts zukommen. Auf dieser Grundlage muss der bestehende Ausschnitt der Datenbank mit den entsprechenden Namenskonventionen erweitert werden. Dabei werden die benötigten Tabellen über Rails migriert. Es werden Tabellen für alle auszufüllenden Felder mit begrenzten Datensätzen erstellt. Eine Tabelle für den Materialeingangsbericht sammelt die eingegebenen Daten und ordnet diese der Tabelle des Teilprojekts zu, indem sie die entsprechenden Datensätze der anderen Tabellen referenziert. Für jede Tabelle muss in Rails eine Klasse angelegt werden. Dort müssen die Eingaben dann validiert werden, bevor sie tatsächlich in die Datenbank geschrieben werden. Ein Entwurfsdiagramm der Datenbank findet sich im Anhang **A.4: Datenbankentwurf** auf Seite **v**.

4.5 Fachkonzept

Für die Umsetzung der Logik muss jeweils ein Controller für ein Projekt und ein Teilprojekt angelegt werden, um die grundlegende Struktur von Copper zu simulieren. Dabei beinhalten diese nur die grundlegendsten Methoden, welche es mir gestatten sollen, mich über den Browser zu meinen **Views** zu navigieren und den zu erstellenden Controller für die Berichte anzusprechen. Auf diesen werden dann die **CRUD**-Operationen definiert.

4.6 Maßnahmen zur Qualitätssicherung

Während der Bearbeitung des Projekts wurde die Funktionalität der Anwendung kontinuierlich getestet. Dabei wurden die Links ausgeführt, die Routen getestet und nach Eingaben die Datenbank auf die Richtigkeit der Werte geprüft. In Zusammenarbeit mit der **QC**-Mitarbeiterin wird ein für die Abnahme erforderliches Testprotokoll erstellt. Ein Auszug des Protokolls befindet sich im Anhang **A.5: Testprotokoll** auf Seite **vi**.

4.7 Pflichtenheft

Nachfolgend werden die wichtigsten vereinbarten Pflichten des Projekts aufgelistet, um eine Standardisierung der Eingangsberichte, die Integrierbarkeit in Copper sowie die Erweiterbarkeit für die Erstellung späterer Auswertungen zu gewährleisten.

Pflichten der Benutzungsoberfläche:

PB10: Es muss eine Webanwendung programmiert und ständig mit Chrome getestet werden.

PB20: Es muss Bootstrap verwendet und die Skalierbarkeit ständig über den Webbrowser getestet werden.

PB30: Jede Eingabe muss ein Label erhalten. Die Namensbezeichnung erfolgt in Absprache mit der zuständigen Mitarbeiterin für Qualitätssicherung.

5 Implementierungsphase

PB40: Es muss Dropdown-Menüs und Checkboxes geben.

PB50: Es muss Textfelder und Textareas geben. Diese müssen optisch zu den Dropdown-Menüs und Checkboxes zugeordnet werden.

PB60: Es muss eine initiale tabellarische Anzeige der Berichte nach Auswahl des Teilprojekts geben. Dort müssen der Audiotyp, die Version, die Bewertung, sowie Links zum Anzeigen, Bearbeiten und Löschen erscheinen.

PB70: Es muss ein Webformular mit den o.g. Konventionen geben.

PB80: Es muss einen Link zum Bearbeiten geben, welcher das unter **PB70** definierte Formular mit den entsprechenden Datensätzen lädt.

PB90: Es gibt einen Link zur Anzeige, welcher das unter **PB70** definierte Formular mit deaktivierten Eingabefelder lädt.

Funktionelle Pflichten:

PF10: Berichte mit gleichem Teilprojekt, Audiotyp und Audioformat müssen eine unterschiedliche Versionsnummer haben.

PF20: Über die Auswahl der Teilprojekte muss die Auswahl eines jeden zugehörigen Berichts erfolgen.

Sonstige Pflichten:

PS10: Alle bekannten Informationen über Copper vom Betriebssystem bis zu den Versionen der Frameworks und Bibliotheken werden verwendet.

PS20: Die Daten werden in eine relationale MySQL-Datenbank gespeichert.

4.8 Zwischenstand

Tabelle 4 zeigt den Zwischenstand nach der Entwurfsphase.

Vorgang	Geplant	Tatsächlich	Differenz
Entwurf der Projektumgebung	1 h	1 h	
Entwurf der Benutzeroberflächen	3 h	4 h	+1 h
Entwurf der erweiterten Datenbank	4 h	4 h	
Erstellung eines Testprotokolls	2 h	2 h	
Erstellung des Pflichtenhefts	4 h	4 h	
Entwurfsphase	14 h	15 h	+1 h

Tabelle 4: Zwischenstand nach der Entwurfsphase

5 Implementierungsphase

Bevor mit der eigentlichen Programmierung und Implementierung begonnen werden kann, muss direkt am Anfang eine dazugehörige Entwicklungsumgebung aufgesetzt werden. Dies war zum Zeitpunkt des

5 Implementierungsphase

Projektantrags nicht bekannt. Während der gesamten Dauer des Projekts wurde regelmäßig über die Versionsverwaltungssoftware Github gesichert.

5.1 Implementierung der Projektumgebung

Nach einem Backup des bisherigen Systems auf einem MacBook Pro, werden **Ruby**, **Rails** und alle weiteren benötigten **Frameworks** und Bibliotheken mit den bekannten Informationen eingebunden und deren Funktionstüchtigkeit getestet. Der **MySQL**-Server, Ruby Version Manager (**rvm**), über den **Ruby** installiert wird, sowie das Datenbank-Management-System Sequel Pro werden mit Hilfe des OS X Paket Managers Homebrew installiert. Über den Befehl **rails new app** wird ein neues Projekt erstellt. Als Entwicklungsumgebung (**IDE**) wird Rubymine verwendet.

5.2 Implementierung des Webservers und des Routings

Als Web Server wird Thin verwendet, welcher leicht als **gem** über **Ruby** eingebunden werden kann und keiner weiteren Konfiguration bedarf. In der Projektdatei **config/routes.rb** werden die Routen zum Controller mit den entsprechenden Operationen definiert. Siehe Anhang **A.5.1: Routes.rb** auf Seite **vii**.

5.3 Implementierung der Datenbank

Nach der Erstellung eines Projekts wird über den Befehl **rake db:create** die Datenbank erstellt und danach das vom externen Entwickler erhaltene **SQL**-Skript importiert. Migrationen für die neuen Tabellen werden in **Rails** geschrieben und über **rake db:migrate** der Datenbank hinzugefügt. Das hat den Vorteil, dass man jederzeit seine selbst erstellten Migrationen zurücksetzen, diese editieren und der Datenbank wieder hinzufügen kann. Über ein selbst erstelltes **SQL**-Skript werden dann die Daten für die Tabellen der Dropdown-Menüs importiert. Ein Auszug der Migration für die **MEP**-Tabelle findet sich im Anhang **A.5.2: Migration.rb** auf Seite **vii**.

5.4 Implementierung der Model-Klassen

Für jede Tabelle der Datenbank wird eine Klasse erstellt. Dies geschieht über den Befehl **rails generate model [Tabellenname]**. Dabei ist darauf zu achten, dass man sich an den Namenskonventionen von **Rails** orientiert. **Rails** kann so die entsprechenden **Models** der jeweiligen Tabelle automatisch zuordnen und die Datenbankabfragen generieren. Danach werden in den **Model**-Klassen die Relationen zueinander nach dem Vorbild der Datenbank definiert und Operationen zur Gültigkeitsprüfung der an die Tabelle zu übergebenden Daten geschrieben. Ein Auszug des **MEP-Models** findet sich im Anhang **A.5.3: Model.rb** auf Seite **viii**.

5.5 Implementierung des Fachkonzepts

Über den Befehl `rails generate controller [Controllernamen]` wird eine neue Klasse für die entsprechenden Controller erstellt. Entsprechend der Namenskonventionen werden durch **Rails** zusätzliche Dateien wie die **Views**, verschiedene Hilfsdateien, sowie Pfade zu den **Views** der Methoden des **Controllers** angelegt. Anhand der Methodennamen erfolgt dann das Routing zu dessen Operationen. Die Methoden im **Controller** entsprechen den üblichen **CRUD**-Operationen. Ein Auszug des erstellten **Controllers** findet sich im Anhang **A.5.4: Controller.rb** auf Seite **ix**.

5.6 Implementierung der Benutzeroberfläche

Für die Ansicht der Projekte und Teilprojekte werden nur die nötigsten Dateien und Links für die Weiterleitung zum Materialeingangsbericht implementiert. Die Ansichten für den Bericht werden mit der HTML – Abstraktionssprache (**HAML**) geschrieben. Über die Index-Datei des Berichts werden alle zu dem Teilprojekt gespeicherten Berichte aufgelistet und zu den entsprechenden Ansichten verlinkt. Zudem wird ein Link zur Erstellung eines neuen Berichts generiert. Für das Anzeigen, Bearbeiten und Erstellen wird jeweils das gleiche „Partial“ – Formular mit aktivierten oder deaktivieren Eingabefeldern geladen. Die Datenauswahl in den Dropdown-Menüs wird aus den entsprechenden Tabellen der Datenbank befüllt. Ein Auszug des HAML - Formulars findet sich im Anhang **A.5.5: View.html** auf Seite **xi**. Ein Screenshot der daraus generierten HTML-Ansicht findet sich im Anhang **A.6: Auszug Benutzerdokumentation** auf Seite **xiv**.

5.7 Zwischenstand

Tabelle **5** zeigt den Zwischenstand nach der Implementierungsphase.

Vorgang	Geplant	Tatsächlich	Differenz
Implementierung der Projektumgebung	6 h	9 h	+3 h
Implementierung der Datenbank und Migrationen	8 h	6 h	-2 h
Implementierung der Models	6 h	6 h	
Implementierung der Logik für Darstellung und Datenhaltung	8 h	7 h	-1 h
Implementierung der Views	9 h	10 h	+1 h
Implementierungsphase	37 h	38 h	+1 h

Tabelle 5: Zwischenstand nach der Implementierungsphase

6 Qualitätssicherung und Abnahme

Die Abnahme erfolgt durch die Verantwortliche der **QC**-Abteilung und dem Ausbilder. Dabei werden erneut die Daten aus dem Testprotokoll eingegeben. Die Aus- und Eingabe sowie das äußere Design

7 Einführungsphase

wird durch die QC-Leiterin geprüft, wohingegen die technischen Spezifikationen für Copper durch den Ausbilder kontrolliert werden.

6.1 Zwischenstand

Tabelle 6 zeigt den Zwischenstand nach der Abnahmephase.

Vorgang	Geplant	Tatsächlich	Differenz
Qualitätssicherung und Abnahme	1 h	1 h	
Abnahme	1 h	1 h	

Tabelle 6: Zwischenstand nach der Abnahmephase

7 Einführungsphase

7.1 Geplante Einführung

Die tatsächliche Inbetriebnahme der Anwendung kann derzeit noch nicht erfolgen, da dieses Projekt abhängig von der Integration in Copper ist.

7.2 Benutzerschulung

Den Sounddesignern wird das Projekt demonstriert und die Bedienung erklärt.

7.3 Zwischenstand

Tabelle 7 zeigt den Zwischenstand nach der Einführungsphase.

Vorgang	Geplant	Tatsächlich	Differenz
Benutzerschulung	1 h	1 h	
Einführung	1 h	1 h	

Tabelle 7: Zwischenstand nach der Einführungsphase

8 Dokumentation

Diese Projektdokumentation wurde im Rahmen der IHK - Abschlussprüfung zum IT-Fachinformatiker für Anwendungsentwicklung mit \LaTeX erstellt.

8.1 Benutzerdokumentation

Eine vorläufige Benutzerdokumentation mit entsprechenden Hinweisen für die Bedienung wurde mit Microsoft-Word geschrieben und in einen für Handbücher vorgesehenen Netzwerkordner im PDF-Format gespeichert. Ein Auszug befindet sich im Anhang [A.6: Auszug Benutzerdokumentation](#) auf Seite [xiv](#).

8.2 Zwischenstand

Tabelle 8 zeigt den Zwischenstand nach der Dokumentation.

Vorgang	Geplant	Tatsächlich	Differenz
Erstellung der Projektdokumentation	10 h	12 h	+2 h
Erstellen der Benutzerdokumentation	1 h	1 h	
Dokumentation	11 h	13 h	

Tabelle 8: Zwischenstand nach der Dokumentation

9 Fazit

9.1 Soll-/Ist-Vergleich

Auch wenn ich mir den Projektablauf und das Ergebnis zum Zeitpunkt der Antragsstellung gänzlich anders vorgestellt habe, sind die Kollegen mit dem neuen Formular sehr zufrieden und würden es gerne schon in Betrieb nehmen.

Insgesamt weicht die geplante Zeit von der tatsächlich benötigten Zeit nur geringfügig ab. Betrachtet man jedoch die einzelnen Teile, fällt auf, dass es nicht in Gänze einzuschätzen war, wie groß der tatsächliche Aufwand dieser jeweils werden würde. Tabelle 9 zeigt einen Vergleich der veranschlagten sowie tatsächlich aufgewendeten Zeit.

Phase	Geplant	Tatsächlich	Differenz
<i>Analysephase</i>	6 h	7 h	+1 h
<i>Entwurfsphase</i>	14 h	15 h	+1 h
<i>Implementierungsphase</i>	37 h	38 h	+1 h
<i>Abnahme</i>	1 h	1 h	
<i>Einführung</i>	1 h	1 h	
Dokumentation	11 h	13 h	
Gesamt	70 h	75 h	+5 h

Tabelle 9: Soll-/Ist-Vergleich

9.2 Gewonnene Erkenntnisse

Das Projekt in kompletter Eigenregie zu erstellen, kostete zwar einerseits viel Kraft, aber hat mich persönlich weitergebracht. Jeder Schritt in der Erstellung des Projekts machte aus mir einen besseren Entwickler, auch wenn es mir vor Augen führte wieviel Sachen es noch über **Ruby**, **Rails** und die komplette Webprogrammierung es noch zu lernen und zu verinnerlichen gilt, da dies wirklich, ausgenommen der Projektdokumentation, eine Sache ist, die mir Freude bereitet und für die ich gerne morgens zur Arbeit gehe.

9.3 Ausblick

Wenn man für die Anwendung noch ein paar Sicherheitsrichtlinien hinzufügt, die Erstellung von Projekten und Teilprojekten auf ähnlicher Weise wie den Audiobericht generiert, noch ein wenig mehr Zeit dem Testen widmet, könnte diese Anwendung tatsächlich in den Betrieb gehen. Da dies jedoch einen zusätzlichen Mehraufwand für die Firma bedeutet, muss dies noch von der Geschäftsführung entschieden werden. Dabei spielt vor allem der tatsächliche Zeitpunkt der Implementierung von Copper eine Rolle.

Literaturverzeichnis

<http://getbootstrap.com/docs/3.3/css/> 2011

[HTTP://GETBOOTSTRAP.COM/DOCS/3.3/CSS/](http://getbootstrap.com/docs/3.3/css/): *CSS, Global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system*. 2011. – <http://getbootstrap.com/docs/3.3/css/>, Aufgerufen 2018-06-14

<http://guides.rubyonrails.org/v4.2/>

[HTTP://GUIDES.RUBYONRAILS.ORG/V4.2/](http://guides.rubyonrails.org/v4.2/): *Ruby on Rails Guides (v4.2.10)*

<http://haml.info/tutorial.html> 2016

[HTTP://HAML.INFO/TUTORIAL.HTML](http://haml.info/tutorial.html): *Haml Tutorial*. 2016. – <http://haml.info/tutorial.html>, Aufgerufen 2018-06-14

texstudio.sourceforge.net 2017

[TEXSTUDIO.SOURCEFORGE.NET](http://texstudio.sourceforge.net), User M. Manual: *TexStudio Manual, User Manual*. 2017. – http://texstudio.sourceforge.net/manual/current/usermanual_en.html, Aufgerufen 2018-06-14

www.latex-tutorial.com 2015

[TUTORIAL.COM](http://www.latex-tutorial.com), Bibtex [www.latex](http://www.latex-tutorial.com) Tutorial: *Bibliography in LaTeX with Bibtex/Biblatex*. 2015. – <https://www.latex-tutorial.com/tutorials/beginners/latex-bibtex/>, Aufgerufen 2018-06-14

Eidesstattliche Erklärung

Eidesstattliche Erklärung

Ich, Rico Krüger, versichere hiermit, dass ich meine **Dokumentation zur betrieblichen Projektarbeit** mit dem Thema

Vereinheitlichung und Digitalisierung der Materialeingangsprüfungsberichte mithilfe einer zentralen Datenbankanbindung im Synchronbereich

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Berlin, den 15.06.2018

RICO KRÜGER

A Anhang

A.1 Zeitplanung

Vorgang	Geplant	Tatsächlich	Differenz
Durchführung einer Ist-Analyse	2 h	3 h	+1 h
Wirtschaftlichkeitsanalyse und Amotisationsrechnung	1 h	1 h	
Erstellung eines Lastenhefts	3 h	3 h	
Analysephase	6 h	7 h	+1 h
Entwurf der Projektumgebung	1 h	1 h	
Entwurf der Benutzeroberflächen	3 h	4 h	+1 h
Entwurf der erweiterten Datenbank	4 h	4 h	
Erstellung eines Testprotokolls	2 h	2 h	
Erstellung des Pflichtenhefts	4 h	4 h	
Entwurfsphase	14 h	15 h	+1 h
Implementierung der Projektumgebung	6 h	9 h	+3 h
Implementierung der Datenbank und Migrationen	8 h	6 h	-2 h
Implementierung der Models	6 h	6 h	
Implementierung der Logik für Darstellung und Datenhaltung	8 h	7 h	-1 h
Implementierung der Views	9 h	10 h	+1 h
Implementierungsphase	37 h	38 h	+1 h
Qualitätssicherung und Abnahme	1 h	1 h	
Qualitätssicherung und Abnahme	1 h	1 h	
Einführungsphase	1 h	1 h	
Einführungsphase	1 h	1 h	
Erstellen der Projektdokumentation	10 h	12 h	+2 h
Erstellung der Benutzerdokumentation	1 h	1 h	
Dokumentation	11 h	13 h	+2 h
Gesamt	70 h	75 h	+5 h

A.2 Projekt-Ressourcen

Projekt-Ressourcen

Beteiligte Personen

- Rico Krüger, Auszubildender, IT-Fachinformatiker – Anwendungsentwicklung, Berliner Synchron GmbH
- Marko Baroke, Ausbilder, Senior IT-Administrator, Berliner Synchron GmbH
- Carina Panicke, Tonmeisterin, Leiterin Quality Check, Berliner Synchron GmbH

Hardware

- MacBook Pro 2017 (3,5 GHz Intel Core i7, 16 GB DDR3 RAM, 500 GB SSD)
- 2 Monitore (BenQ, 24 Zoll)
- Maus, Tastatur

Software, Frameworks, Services

- **Betriebssystem:** macOS Serra (v 10.12.6)
- **Programmiersprache:** Ruby (v. 2.3.1), HAML, SQL
- **Frameworks und Bibliotheken:** Rails (v. 4.2.7.1), Bootstrap (v. 3.3), mysql2 (v. 0.3.17), sass-rails (v. 5.0), uglifier (v. 1.3.0), jquery-rails, turbolinks, spring, haml-rails, thin, bootstrap-datepicker-rails, minitest
- **Sonstige Software und Services:** Homebrew, Ruby Version Manager, Rubymine, Github, MySQL (v. 5.6), Sequel Pro, Draw.io, Balsamiq (v. 3.5), LATEX, Texstudio, Microsoft Office

Sonstige Ressourcen

- Büroraum, Strom, Internet

A.3 Mockups

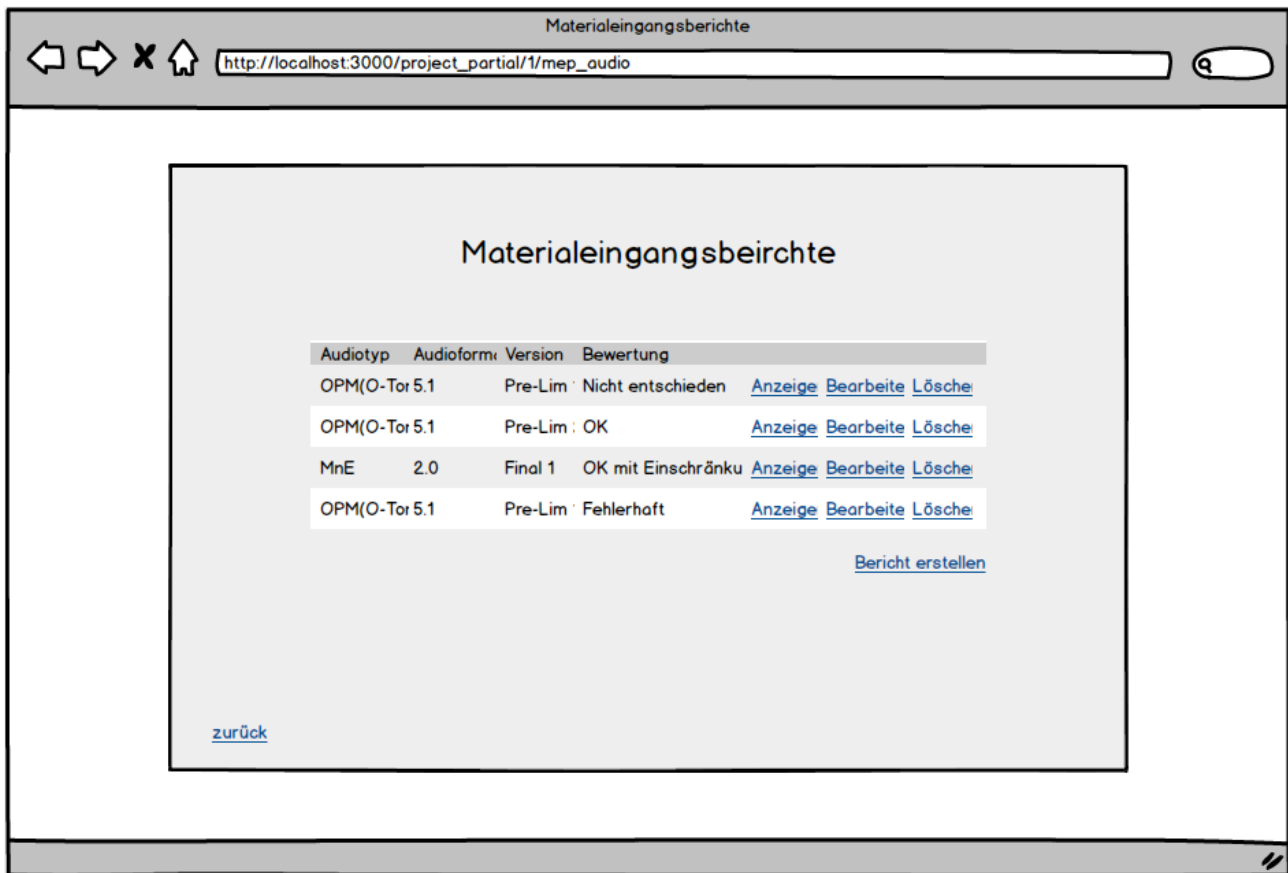


Abbildung 1: Mockup - Übersicht der Materialeingangsberichte

The mockup shows a web browser window with the title 'Materialeingangsberichte'. The address bar contains the URL 'http://localhost:3000/project_partials/1/new'. The main content area displays a form titled 'Materialeingangsbeirchte' (note the typo in the mockup). The form is organized into several sections:

- Audio Data:** Includes dropdowns for 'Audiotyp' (set to 'O-Ton (OPM)'), 'Audioformat' (set to 'Mono'), and 'Geschwindigkeit' (set to '23.976').
- Versioning:** Includes dropdowns for 'Version' (set to 'Pre-Lim') and 'Versionsnummer' (set to '0').
- Date and Calendar:** A 'Datum' field shows '02.06.2018'. A calendar widget for June 2018 is open, showing the date '02' selected.
- External Name:** A text input field for 'Externer Name'.
- Technical Data:** Includes input fields for 'dBTP max', 'LRA', and 'LUFS'.
- Checkboxes:** Two checkboxes labeled 'Synchorn' and 'Vollständig' are present, both currently unchecked.
- Comments:** Four text input fields labeled 'Kommentar' are provided for additional information.
- Channels and Status:** Includes dropdowns for 'Kanäle' (set to 'Nicht geprüft') and 'Störung' (set to 'Nicht geprüft').
- Rating:** A dropdown for 'Bewertung' is set to 'Nicht entschieden'.
- Actions:** A blue button labeled 'Speichern' (Save) is located at the bottom right of the form.

Abbildung 2: Mockup - Eingabeformular eines Materialeingangsberichts

A Anhang

A.4 Datenbankentwurf

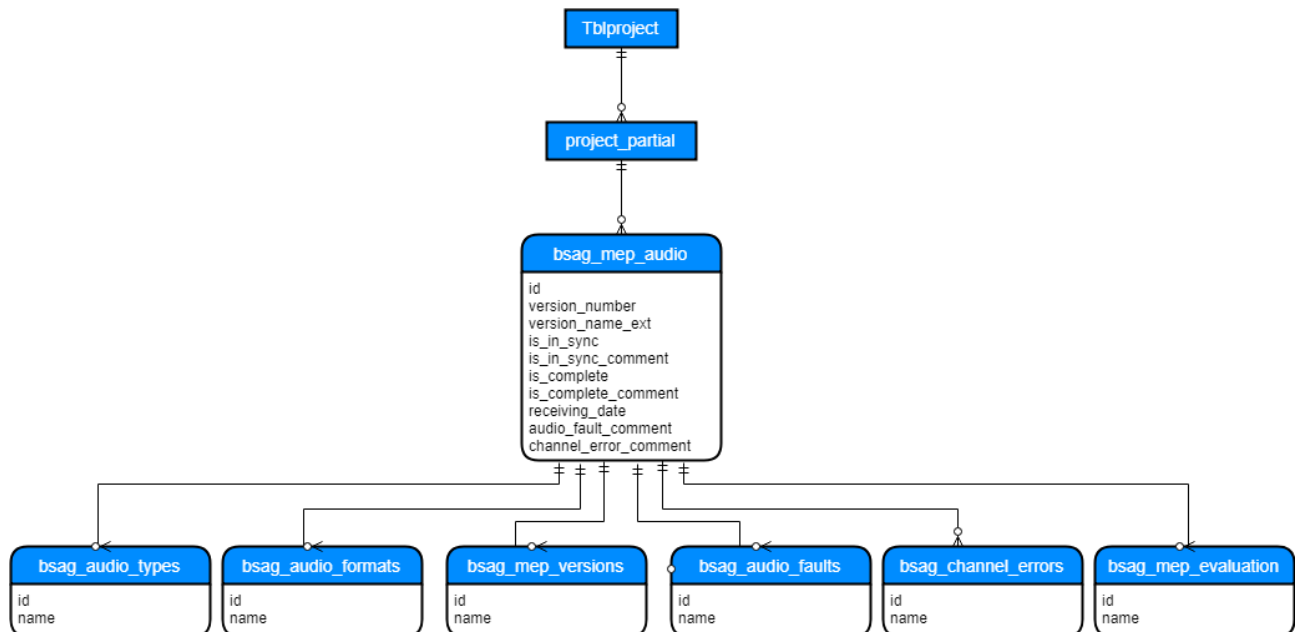


Abbildung 3: Entwurf der MySQL - Datenbank

A Anhang

A.5 Testprotokoll

Teil- projekt	Test- datesatz	Audio- typ	Audio- format	Audio- version	Versions- nummer	Eingangs- datum	Neu / Bearbeiten / Speichern / Löschen	Erwartungs- horizont	Ergebnis	Auswertung
1	1	OPM	5.1	Final	1	08.06.2018	N / S	WAHR	WAHR	WAHR
1	1	MnE	2.0	Pre-Lim	2	08.06.2018	B / S	WAHR	WAHR	WAHR
1	1	MnE	2.0	Pre-Lim	2		B / S	FALSCH	FALSCH	WAHR
1	1	MnE	2.0	Pre-Lim	2	08.06.2019	B / S	FALSCH	FALSCH	WAHR
1	1	MnE	2.0	Pre-Lim	2	08.06.2018	L	WAHR	WAHR	WAHR
1	2	MnE	2.0	Final	2	08.06.2018	N / S	WAHR	WAHR	WAHR
1	3	MnE	2.0	Final	2	08.06.2018	N / S	FALSCH	FALSCH	WAHR
1	3	MnE	2.0	Final	1	08.06.2018	N / S	WAHR	WAHR	WAHR
1	4	MnE	2.0	Pre-Lim	1	08.06.2018	N / S	WAHR	WAHR	WAHR
1	5	MnE	2.0	Pre-Lim	1	08.06.2018	N / S	FALSCH	FALSCH	WAHR
1	5	MnE	2.0	Pre-Lim	2	08.06.2018	N / S	WAHR	WAHR	WAHR
1	6	MnE	5.1	Pre-Lim	2	08.06.2018	N / S	WAHR	WAHR	WAHR
1	7	MnE	atmos	Pre-Lim	2	08.06.2018	N / S	WAHR	WAHR	WAHR
1	7	MnE	5.1	Pre-Lim	2	08.06.2018	B / S	FALSCH	FALSCH	WAHR
1	7	OPM	5.1	Pre-Lim	2	08.06.2018	B / S	WAHR	WAHR	WAHR
1	7	MnE	2.0	Pre-Lim	2	08.06.2018	B / S	WAHR	WAHR	WAHR
1	7	MnE	2.0	Pre-Lim	2	08.06.2018	L	WAHR	WAHR	WAHR
1	6	MnE	5.1	Pre-Lim	2	08.06.2018	L	WAHR	WAHR	WAHR
1	5	MnE	2.0	Pre-Lim	2	08.06.2018	L	WAHR	WAHR	WAHR
1	4	MnE	2.0	Pre-Lim	1	08.06.2018	L	WAHR	WAHR	WAHR
1	3	MnE	2.0	Final	1	08.06.2018	L	WAHR	WAHR	WAHR
1	2	MnE	2.0	Final	2	08.06.2018	L	WAHR	WAHR	WAHR
1	8	FX	2.1	Locked	5	08.06.2018	N / S	WAHR	WAHR	WAHR
2	9	FX	2.1	Locked	5	08.06.2018	N / S	WAHR	WAHR	WAHR
3	10	FX	2.1	Locked	5	08.06.2018	N / S	WAHR	WAHR	WAHR

Abbildung 4: Auszug aus dem Testprotokoll

A Anhang

A.5.1 Routes.rb

```
1 Rails.application.routes.draw do
2   resources :projects do
3     resources :project_partials, only: [:index, :new, :create]
4   end
5   resources :project_partials, only: [:show, :edit, :update, :destroy] do
6     resources :mep_audios, only: [:index, :new, :create]
7   end
8   resources :mep_audios, only: [:show, :edit, :update, :destroy]
9   root 'projects#index'
10 end
```

A.5.2 Migration.rb

```
1 class CreateBsgMepAudios < ActiveRecord::Migration
2   def change
3     create_table :bsag_mep_audios do |t|
4       t.references :bsag_project_partial, null: false, index: true, foreign_key: true
5       t.references :bsag_audio_type, null: false, index: true, foreign_key: true
6       t.date :receiving_date, null: false
7       t.references :bsag_mep_version, null: false, index: true, foreign_key: true
8       t.integer :version_number, limit: 2, null: false
9       t.string :version_name_ext, limit: 30
10      t.references :bsag_audio_format, null: false, index: true, foreign_key: true
11      t.references :bsag_audio_speed, null: false, index: true, foreign_key: true
12      t.boolean :is_complete
13      t.string :is_complete_comment, null: false
14      t.boolean :is_in_sync
15      t.string :is_in_sync_comment, null: false
16      t.decimal :dbtp_max, precision: 3, scale: 1
17      t.decimal :lra, precision: 3, scale: 1
18      t.decimal :lufs, precision: 3, scale: 1
19      t.references :bsag_channel_error, null: false, index: true, foreign_key: true
20      t.string :channel_error_comment
21      t.references :bsag_audio_fault, null: false, index: true, foreign_key: true
22      t.string :audio_fault_comment
23      t.references :bsag_mep_evaluation, null: false, index: true, foreign_key: true
24      t.timestamps null: false
25    end
26  end
27 end
```


A.5.3 Model.rb

```
1 class BsapMepAudio < ActiveRecord::Base
2   belongs_to :project_partial, class_name: 'BsapProjectPartial', foreign_key: 'bsag_project_partial_id'
3   belongs_to :audio_type, class_name: 'BsapAudioType', foreign_key: 'bsag_audio_type_id'
4   belongs_to :mep_version, class_name: 'BsapMepVersion', foreign_key: 'bsag_mep_version_id'
5   belongs_to :audio_format, class_name: 'BsapAudioFormat', foreign_key: 'bsag_audio_format_id'
6   belongs_to :audio_speed, class_name: 'BsapAudioSpeed', foreign_key: 'bsag_audio_speed_id'
7   belongs_to :channel_error, class_name: 'BsapChannelError', foreign_key: 'bsag_channel_error_id'
8   belongs_to :audio_faults, class_name: 'BsapAudioFault', foreign_key: 'bsag_audio_fault_id'
9   belongs_to :evaluation, class_name: 'BsapMepEvaluation', foreign_key: 'bsag_mep_evaluation_id'
10
11   validates :version_number, uniqueness: { scope: [:project_partial, :audio_type, :audio_format, :mep_version] }
12   validates :bsag_audio_fault_id, presence: true
13   validates_length_of :version_name_ext, maximum: 30
14   validates_length_of :is_complete_comment,
15                     :is_in_sync_comment,
16                     :channel_error_comment,
17                     :audio_fault_comment,
18                     maximum: 255
19   # validates decimal format
20   validates :dbtp_max_before_type_cast, :lra_before_type_cast, :lufs_before_type_cast,
21           numericality: { only_decimal: true },
22           allow_blank: true,
23           format: { with: /\A[+-]?\d{0,2}(?!\\d)(\\.\\d{0,1})?\\z/ }
24   validate :receiving_date_cannot_be_in_the_future if validates_presence_of :receiving_date
25   def receiving_date_cannot_be_in_the_future
26     errors.add(:receiving_date, "can't be in the future") if receiving_date.presence && receiving_date > Date.
27       today
28   end
29 end
```

A.5.4 Controller.rb

```
1 class MepAudiosController < ApplicationController
2   def index
3     @project_partial = BsagProjectPartial.find(params[:project_partial_id])
4     @mep_audios = @project_partial.mep_audios.all
5   end
6
7   def show
8     @mep_audio = BsagMepAudio.find(params[:id])
9   end
10
11  def new
12    @mep_audio = BsagMepAudio.new
13  end
14
15  def edit
16    @mep_audio = BsagMepAudio.find(params[:id])
17  end
18
19  def create
20    @project_partial = BsagProjectPartial.find(params[:project_partial_id])
21    @mep_audio = @project_partial.mep_audios.create(mep_audio_params)
22    if @mep_audio.save
23      redirect_to project_partial_mep_audios_path
24    else
25      render 'new'
26    end
27  end
28
29  def update
30    @mep_audio = BsagMepAudio.find(params[:id])
31    if @mep_audio.update(mep_audio_params)
32      redirect_to project_partial_mep_audios_path(@mep_audio.bsag_project_partial_id)
33    else
34      render 'edit'
35    end
36  end
37
38  def destroy
39    @mep_audio = BsagMepAudio.find(params[:id])
40    @mep_audio.destroy
41    redirect_to project_partial_mep_audios_path(@mep_audio.bsag_project_partial_id)
42  end
43
44  private
45
46  def mep_audio_params
47    params.require(:bsag_mep_audio).permit(:bsag_project_partial_id,
48                                             :bsag_audio_type_id,
49                                             :receiving_date,
```

A Anhang

```
50      :bsag_mep_version_id,  
51      :version_number,  
52      :version_name_ext,  
53      :bsag_audio_format_id,  
54      :bsag_audio_speed_id,  
55      :is_complete,  
56      :is_complete_comment,  
57      :is_in_sync,  
58      :is_in_sync_comment,  
59      :dbtp_max, :lra, :lufs ,  
60      :bsag_channel_error_id,  
61      :channel_error_comment,  
62      :bsag_audio_fault_id,  
63      :audio_fault_comment,  
64      :bsag_mep_evaluation_id)  
65  end  
66  end
```

A.5.5 View.html

```

1  – disabled ||= false
2
3  = form_for @mep_audio, url: { action: action_name }, html: { class: "form-group" } do |form|
4    – if @mep_audio.errors.any?
5      .group
6        .alert .alert –danger
7          %h2
8            = pluralize(@mep_audio.errors.count, "errors")
9            prohibited this mep from being saved:
10           %ul
11             – @mep_audio.errors.full_messages.each do |msg|
12               %li= msg
13
14     .row
15       .group
16         .col-xs-12
17         .form-group-row
18         .col-xs-6.col-md-3
19           = form.label :bsag_mep_version_id, "Audiotyp"
20           = form.collection_select :bsag_audio_type_id, BsagAudioType.order(:id), :id, :name,
21             {}, { class: "form-control", disabled: disabled }
22
23         .form-group-row
24         .col-xs-6.col-md-3
25           = form.label :bsag_audio_format_id, "Audioformat"
26           = form.collection_select :bsag_audio_format_id, BsagAudioFormat.order(:id), :id, :name,
27             {}, { class: "form-control", disabled: disabled }
28
29         .form-group-row
30         .col-xs-12.col-md-6
31           = form.label :receiving_date
32           = form.text_field :receiving_date, { class: "form-control", "data-provide" => "datepicker", "data-date
33             –format" => "dd.mm.yyyy", disabled: disabled,
34             value: (@mep_audio.receiving_date ? @mep_audio.receiving_date.strftime("%d.%m.%
35             Y") : Date.today.strftime("%d.%m.%Y")) }
36
37     .row
38       .group
39       .col-xs-12
40       .form-group-row
41       .col-xs-6.col-md-3
42         = form.label :bsag_mep_version_id, "Audioversion"
43         = form.collection_select (:bsag_mep_version_id, BsagMepVersion.order(:id), :id, :name,
44           {}, { class: "form-control", disabled: disabled })
45
46       .form-group-row
47       .col-xs-6.col-md-3
48         = form.label :version_number, "Versionsnummer"
49         = form.select :version_number, (0..9), {}, { class: "form-control", disabled: disabled }
50
51       .form-group-row
52       .col-xs-12.col-md-6
53         = form.label :version_name_ext, "Externer Name"
54         = form.text_field :version_name_ext, { class: "form-control text_area", disabled: disabled }
55
56     .row

```

A Anhang

```

48     .group
49     .col-xs-12
50     .form-group-row
51     .col-xs-6.col-md-3
52     = form.label :bsag_audio_speed_id, "Geschwindigkeit"
53     = form.collection_select :bsag_audio_speed_id, BsagAudioSpeed.order(:id), :id, :speed,
54     {}, { class: "form-control", disabled: disabled }
55     .form-group-row
56     .col-xs-6.col-md-3
57     = form.label :dbtp_max, "dBTP max"
58     = form.text_field :dbtp_max, { class: "form-control text_area", disabled: disabled }
59     .form-group-row
60     .col-xs-6.col-md-3
61     = form.label :lra, "LRA"
62     = form.text_field :lra, { class: "form-control text_area", disabled: disabled }
63     .form-group-row
64     .col-xs-6.col-md-3
65     = form.label :lufs, "LUFS"
66     = form.text_field :lufs, { class: "form-control text_area", disabled: disabled }
67 .row
68     .group
69     .col-xs-12
70     .form-group-row
71     .col-xs-12.col-md-3
72     = form.label :is_complete, "Vollstaendig"
73     = form.check_box :is_complete, { class: "form-control no-border", disabled: disabled }
74     .form-group-row
75     .col-xs-12.col-md-9
76     = form.label :is_complete_comment, "Kommentar"
77     = form.text_area :is_complete_comment, { class: "form-control text_area", disabled: disabled }
78 .row
79     .group
80     .col-xs-12
81     .form-group-row
82     .col-xs-12.col-md-3
83     = form.label :is_in_sync, "Synchron"
84     = form.check_box :is_in_sync, { class: "form-control no-border", disabled: disabled }
85     .form-group-row
86     .col-xs-12.col-md-9
87     = form.label :is_in_sync_comment, "Kommentar"
88     = form.text_area :is_in_sync_comment, { class: "form-control text_area", disabled: disabled }
89 .row
90     .group
91     .col-xs-12
92     .form-group-row
93     .col-xs-12.col-md-3
94     = form.label :bsag_channel_error_id, "Kanaele"
95     = form.collection_select :bsag_channel_error_id, BsagChannelError.order(:id), :id, :name,
96     {}, { class: "form-control", disabled: disabled }
97     .form-group-row
98     .col-xs-12.col-md-9
    
```



A Anhang

```
99         = form.label :channel_error_comment, "Kommentar"
100         = form.text_area :channel_error_comment, { class: "form-control text_area", disabled: disabled }
101     .row
102     .group
103     .col-xs-12
104     .form-group-row
105     .col-xs-12.col-md-3
106         = form.label :bsag_audio_fault_id, "Stoerung"
107         = form.collection_select :bsag_audio_fault_id, BsagAudioFault.order(:id), :id, :name,
108             {}, { class: "form-control", disabled: disabled }
109     .form-group-row
110     .col-xs-12.col-md-9
111         = form.label :audio_fault_comment, "Kommentar"
112         = form.text_area :audio_fault_comment, { class: "form-control text_area", disabled: disabled }
113 .row
114 .group
115 .col-xs-12
116 .form-group-row
117 .col-xs-12.col-md-3
118     = form.label :bsag_mep_evaluation_id, "Bewertung"
119     = form.collection_select :bsag_mep_evaluation_id, BsagMepEvaluation.order(:id), :id, :name,
120         {}, { class: "form-control", disabled: disabled }
121 .form-group-row
122 .col-xs-12.col-md-9
123     %label Aenderungen speichern
124     = form.button :Speichern, name: :create, class: "btn btn-primary btn-block", disabled: disabled
```

A.6 Auszug Benutzerdokumentation

Erstellung eines Materialeingangsprüfungsberichts für Audiodateien - Benutzerdokumentation

The screenshot shows a web browser window with the URL `localhost:3000/project_partials/1/mep_audios/new`. The page title is "Bericht erstellen". The form contains the following fields and controls:

- Audiotyp:** Dropdown menu with "O-Ton (OPM)" selected.
- Audioformat:** Dropdown menu with "Mono" selected.
- Eingangsdatum:** Text input field with "10.06.2018" entered.
- Audioversion:** Dropdown menu with "Pre-Lim" selected.
- Versionsnummer:** Text input field with "0" entered.
- Externer Name:** Text input field (empty).
- Geschwindigkeit:** Text input field with "23.976" entered.
- dBTP max:** Text input field (empty).
- LRA:** Text input field (empty).
- LUFS:** Text input field (empty).
- Vollständig:** Checkbox (unchecked).
- Kommentar:** Text area (empty).
- Synchron:** Checkbox (unchecked).
- Kommentar:** Text area (empty).
- Kanäle:** Dropdown menu with "nicht geprüft" selected.
- Kommentar:** Text area (empty).
- Störung:** Dropdown menu with "nicht geprüft" selected.
- Kommentar:** Text area (empty).
- Bewertung:** Dropdown menu with "Nicht entschieden" selected.
- Änderungen speichern:** Button with "Speichern" label.
- zurück:** Link at the bottom left.

Erstellung:

- Durch einfaches Anklicken der Formularfelder wird der Audiotyp, die Audioversion usw. ausgewählt.
- Ein Bericht kann für eine Serie / einen Akt nicht doppelt angelegt werden. D.h. ein Bericht eines Teilprojektes kann nicht den gleichen Audiotyp, Audioformat, Audioversion und Versionsnummer besitzen. Sollten die ersten drei genannten Dinge übereinstimmen, wählt eine höhere Version. Das Eingabefeld für „Externer Name“ dient zum Eintragen der Namensbezeichnung des Kunden, kann aber auch für sonstige Versionshinweise genutzt werden.
- Beachtet, dass zwingend das Eingangsdatum eingetragen sein muss. Ein zukünftiges Datum ist nicht gestattet.
- In den Feldern dBTP max, LRA, LUFS werden die entsprechenden Werte eingetragen. (Es kann maximal eine Nachkommastelle eingetragen werden).
- In den Checkboxes wird angegeben, ob das Material vollständig bzw. synchron ist. Achtet darauf, dass diese standardmäßig deaktiviert sind, was zur Folge hat, dass der Bericht als unvollständig bzw. asynchron gewertet wird. Diese sollten dringend noch einmal überprüft werden, bevor man sich für eine Bewertung entscheidet.
- In den Kommentarfelder, welche sich neben den Auswahlfelder befinden, können Bemerkungen, wie z.B. der Timecode, notiert werden an der eine Störung aufgetreten ist.
- Über Speichern werden die Daten gesichert und können später erneut abgerufen und bearbeitet werden.
- Bei Falscheingaben erhaltet ihr eine entsprechende Fehlermeldung.

Hinweis:

- Sollte eine möglich Auswahl für ein Dropdown-Menü fehlen, kann die IT-Abteilung kontaktiert und gebeten werden die entsprechende Auswahl in die Datenbank einzupflegen.