

# Repräsentation von Kurseinheiten der FernUniversität als Hyperaudio-Dokumente in Moodle: Design und Implementierung

Bachelorarbeit

eingereicht von  
**Michael Lämmermann**  
(Matrikelnummer 9611711)

angefertigt am  
Lehrgebiet Kooperative Systeme  
Fakultät Mathematik und Informatik  
FernUniversität in Hagen

Betreuer  
Dr. Niels Seidel

September 2018



**Michael Lämmermann**

**Repräsentation von Kurseinheiten der FernUniversität als Hyperaudio-Dokumente  
in Moodle: Design und Implementierung**



## **Zusammenfassung**

Aktuell werden Lerninhalte an der FernUniversität in Hagen hauptsächlich in Form von textlastigen Kurseinheiten vermittelt. Diese schränken die Studierenden, bezogen auf die Tätigkeiten, welche beim Lernen durchgeführt werden können, ein. Aus diesem Grund lässt sich das Lernen aktuell schwer in Alltagstätigkeiten integrieren. Ziel dieser Arbeit soll es sein, eine auditive Lernumgebung zu schaffen, welche alle Inhalte der bestehende Kurseinheiten vermitteln kann. Die auditive Lernumgebung soll es den Studierenden ermöglichen, zukünftig mehr Zeit zum Lernen wahrzunehmen. Um dieses Ziel zu erreichen, wird ein Moodle-Plugin entwickelt, welches eine alternative Repräsentation der Kurseinheiten bietet und zudem Kommunikationsmöglichkeiten für Studierende und Lehrende schafft.

## **Summary**

Currently, learning content is primarily provided in the form of text-heavy course units at the FernUniversität in Hagen. Those restrict the students in terms of activities which can be executed during studying. For that reason learning is currently difficult to integrate in everyday activities. The goal of this work is to create an auditory learning environment imparting all the content of the existing course units. The auditory learning environment should enable students to use more time for studying. In order to reach this goal, a Moodle plugin is developed, offering an alternative representation for course units while also creating the means for communication between students and teachers.

Michael Lämmermann. *Repräsentation von Kurseinheiten der FernUniversität als Hyperaudio-Dokumente in Moodle: Design und Implementierung*. Bachelorarbeit. Fakultät Mathematik und Informatik, FernUniversität in Hagen, 2018.

Diese Publikation ist unter *Creative Commons – Namensnennung 3.0 Deutschland* lizenziert und darf als Ganzes oder ausschnittweise vervielfältigt, verbreitet und öffentlich zugänglich gemacht werden, sofern dies im Text nicht anders vermerkt ist.



Autor: Michael Lämmermann  
Gestaltung und Satz: Michael Lämmermann/  $\text{\LaTeX}$   
Datum: 9. September 2018

# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>9</b>
1.1 Motivation	9
1.2 Problemstellung	9
1.2.1 Ist-Situation	9
1.2.2 Probleme	10
1.3 Zielsetzung und Forschungsfragen	12
<b>2 Grundlagen</b>	<b>13</b>
2.1 Moodle	13
2.2 Kooperation im Lernumfeld	14
2.3 Hypermedia	15
2.3.1 Grundbegriffe	16
2.3.2 Lernen mit Hypermedia	16
2.3.3 Hyperaudio-Repräsentation von Kurseinheiten	17
2.4 Zusammenfassung	18
<b>3 Analyse</b>	<b>19</b>
3.1 Zielgruppe	19
3.1.1 Personas	19
3.1.2 User Stories	20
3.2 Anforderungsdefinition	23
3.2.1 Anforderungen der Administratoren	23
3.2.2 Anforderungen der Nutzenden	23
3.3 Möglichkeiten der Moodle-Plugin-Entwicklung	24
3.4 Aktueller Stand der Technik	26
3.4.1 Etablierte Audio- und Video-Plattformen	26
3.4.2 Technologien für den Einsatz in Moodle	28
3.5 Zusammenfassung	31
<b>4 Konzept</b>	<b>32</b>
4.1 Zusammenhänge der Komponenten der Hyperaudio-Anwendung	32
4.1.1 Komponenten	32
4.1.2 Zusammenhänge	32
4.2 Definition des Schnittstellenformats für Hyperaudio-Dokumente	33
4.3 Datenbankentwurf	34
4.4 Gestaltung der Benutzeroberfläche	36
4.4.1 Zusatzinhalte	36
4.4.2 Mediensteuerung inklusive Visualisierungen der Annotationen	37
4.4.3 Galerie	39
4.4.4 Kommentare und Notizen	39
4.4.5 Zusammenführen der Elemente	42
4.4.6 Mobile Version	42

## Inhaltsverzeichnis

4.5 Zusammenfassung . . . . .	45
<b>5 Implementierung . . . . .</b>	<b>46</b>
5.1 Struktur des Hyperaudio-Plugins . . . . .	46
5.2 Iterative Entwicklung . . . . .	48
5.2.1 Speichern und Abspielen einer Audio-Datei . . . . .	49
5.2.2 Speichern und Anzeige von Zusatzinhalten . . . . .	51
5.2.3 Einbindung der Konfigurationsdatei . . . . .	54
5.2.4 Speichern und Anzeige von Kommentaren . . . . .	55
5.2.5 Antworten auf Kommentare . . . . .	58
5.2.6 Notizen . . . . .	59
5.2.7 Audio Cues . . . . .	60
5.2.8 Galerie der Zusatzinhalte . . . . .	61
5.2.9 Zeitabhängige Visualisierung der Kommentare und Notizen . . . . .	62
5.2.10 Lesezeichen . . . . .	64
5.2.11 Suche, Filter und Sortierung bei Kommentaren . . . . .	65
5.2.12 Unterstützung mobiler Endgeräte . . . . .	66
5.3 Zusammenfassung . . . . .	71
<b>6 Evaluation . . . . .</b>	<b>72</b>
6.1 Validierung der User Stories und Anforderungen . . . . .	72
6.2 Evaluation des Hyperaudio-Plugins . . . . .	79
6.2.1 Rolle der Nutzenden . . . . .	79
6.2.2 Rolle der Lehrenden . . . . .	79
6.3 Verbesserungsvorschläge . . . . .	80
6.4 Zusammenfassung . . . . .	81
<b>7 Fazit . . . . .</b>	<b>82</b>
7.1 Zusammenfassung . . . . .	82
7.2 Ausblick . . . . .	83
<b>A Auswertungen . . . . .</b>	<b>85</b>
A.1 Textanteil der Kurseinheiten . . . . .	85
A.2 Einsatz von Moodle . . . . .	88
<b>Literaturverzeichnis . . . . .</b>	<b>91</b>
<b>Abbildungsverzeichnis . . . . .</b>	<b>95</b>
<b>Tabellenverzeichnis . . . . .</b>	<b>97</b>
<b>Auflistungsverzeichnis . . . . .</b>	<b>99</b>
<b>Abkürzungsverzeichnis . . . . .</b>	<b>101</b>

# 1 Einführung

Diese Arbeit beschäftigt sich mit dem Design und der Implementierung eines Plugins für die Lernplattform Moodle, welches es ermöglichen soll, den Studierenden die Lerninhalte mittels Hyperaudio-Dokumenten bereitzustellen. Ziel dieses Plugins ist die Erweiterung der Lernmöglichkeiten an der FernUniversität in Hagen, um die Studierenden beim Erreichen ihrer Lernziele besser zu unterstützen.

## 1.1 Motivation

Die Motivation zur Behandlung dieses Themas besteht darin, dass 80% der Studierenden gleichfalls neben dem Studium arbeiten (FernUniversität in Hagen, 2018c). Unter diesen Umständen beschäftigen sich viele Studierende erst kurz vor der Prüfung - dafür aber entsprechend intensiv - mit den Lerninhalten. An der Fakultät Mathematik/Informatik und auch an der Fakultät für Wirtschaftswissenschaften bestehen diese Lerninhalte zu einem guten Teil aus textlastigen Kurseinheiten, die Abbildungen und Formeln enthalten.

Diese Aussage kann anhand der Pflichtmodule des Bachelorstudiengangs Wirtschaftsinformatik bestätigt werden. Die Pflichtmodule an der Fakultät Mathematik/Informatik weisen einen Textanteil von 61,08% auf. An der Fakultät für Wirtschaftswissenschaften liegt der Anteil in den Pflichtmodulen nochmals höher bei 64,24%<sup>1</sup>.

Eher selten werden auch Videos angeboten, in welchen bestimmte Lerninhalte aus den Kurseinheiten nochmals rekapituliert werden. Hier sind als Beispiel die Videos von Univ.-Prof. Dr. Ulrike Baumöl zum Kurs „Informationsmanagement“ zu nennen.

Die Idee besteht nun darin, den Lernenden erstens eine alternative Repräsentation (Modalität) der Lerninhalte anzubieten und ihnen zweitens das Lernen während ungenutzter Alltagssituationen zu ermöglichen (z.B. lange Autofahrten, Pendeln in Bus und Bahn, beim Joggen, etc.). Auf diese Weise könnten die Lernenden die Inhalte häufiger rezipieren und einüben. Ergänzt um gute E-Assessments (Selbsttests) hätten sie in Summe eine Chance, sich frühzeitig und kontinuierlich auf die Prüfung vorzubereiten und vielleicht bessere Lernerfolge zu erzielen.

## 1.2 Problemstellung

Diese Arbeit wird sich in diesem Zusammenhang vor allem mit dem Problem des hohen Textanteils vieler Kurse und dem damit verbundenen Lernverhalten beschäftigen. Doch zunächst soll die Ist-Situation für die Studierenden an der FernUniversität in Hagen beschrieben werden.

### 1.2.1 Ist-Situation

Jeder Studierende hat Zugriff auf den *Virtuellen Studienplatz*, häufig auch *Virtuelle Universität* (VU) oder *Lernraum Virtuelle Universität* (LVU) genannt. Hierbei handelt es sich um ein eigenentwickeltes Webportal der FernUniversität in Hagen. Der *Virtuelle Studienplatz* stellt unter anderem das

---

<sup>1</sup>Genauere Informationen zu dieser Auswertung können im Anhang A.1 nachgeschlagen werden.

## 1 Einführung

Kurs- und Studiumsportal der FernUniversität dar. Hierüber können die Studierenden Kurse belegen, die Rückmeldung für das nächste Semester vornehmen oder ihre persönlichen Daten einsehen und bearbeiten. Auch eine Übersicht über das Veranstaltungsangebot wird dem Studierenden geboten. Zusätzlich bietet der *Virtuelle Studienplatz* eine Übersicht über alle belegten Kurse des Studierenden (FernUniversität in Hagen, 2018d). Mittels dieser Übersicht kann der Studierende direkt auf das jeweilige persönliche Kursportal seiner belegten Kurse gelangen. Neben allgemeinen Informationen zu dem Kurs bietet das Kursportal unter anderem Zugriff auf die Online-Studieninhalte (z.B. Kurseinheiten<sup>2</sup>, Einsendeaufgaben, Musterlösungen) und Verweise zu anderen Diensten (Moodle, Online-Übungssystem, Kommunikationsangebote Adobe Connect Videokonferenzen), die in diesem Kurs zur Verfügung stehen (FernUniversität in Hagen, 2014).

Neben der Möglichkeit, die Kurseinheiten als PDF (Portable Document Format) über den *Virtuellen Studienplatz* herunterzuladen, werden diese im Regelfall für die belegten Kurse automatisch in gedruckter Form an die Studierenden versendet. Die Kurseinheiten dienen als zentrales Lernmaterial für die Studierenden.

Jeder Kurs hat die Möglichkeit, zusätzliches Lernmaterial über Moodle, die zentrale Lernplattform der FernUniversität in Hagen, zur Verfügung zu stellen. Moodle bietet den Kursen „als sogenanntes Learningmanagementsystem (LMS) vielfältige Möglichkeiten zur Gestaltung der mediengestützten Lehre an“ (FernUniversität in Hagen, 2018b). Besonders hervorzuheben ist hierbei der mögliche Einsatz von Lehrvideos, Foren und Tests. Die Plattform Moodle und ihre Erweiterungsmöglichkeiten werden in Abschnitt 2.1 vorgestellt. Im Bezug auf die Pflichtmodule des Bachelorstudiengangs Wirtschaftsinformatik an der FernUniversität in Hagen wird Moodle in 21 von 30 Kursen beziehungsweise 10 von 15 Modulen eingesetzt (vgl. Anhang A.2). Auffällig ist dabei, dass während Moodle an der Fakultät für Wirtschaftswissenschaft für jeden Kurs angeboten wird, an der Fakultät für Mathematik und Informatik kein einziger Kurs der Pflichtmodule des Bachelorstudiengangs Wirtschaftsinformatik auf Moodle zurückgreift.

Durch den Einsatz von Adobe Connect besteht an der FernUniversität in Hagen auch die Möglichkeit sogenannter *Virtual Classrooms*. Dabei handelt es sich um eine Video- und Tonübertragung mit Textchat und Freigabemöglichkeiten für Präsentationen und Bildschirmhalte. „Ein *Virtual Classroom* [Hervorhebung v. Verf.] eignet sich insbesondere für Veranstaltungen, in denen die synchrone Kommunikation ein hohes Gewicht erhält: Seminare, Tutorien, Sprechstunden, Arbeitsgruppen u.Ä.“ (FernUniversität in Hagen, 2018a).

Darüber hinaus werden den Studierenden mit den *Diskussionsforen* (Newsportal) und dem *Conference Center* als Chat-System zwei weitere Systeme zur Kommunikation geboten (FernUniversität in Hagen, 2018a).

### 1.2.2 Probleme

Die im vorangegangen Abschnitt beschriebenen Lernangebote weisen jedoch Defizite im Bezug auf die Vermittlung der Lerninhalte auf. Der *Virtuelle Studienplatz* dient aktuell ausschließlich als Portal, um die Studierenden zu den von ihnen benötigen Informationen zu leiten und unterstützt somit nur indirekt die Vermittlung von Lerninhalten.

Die Kurseinheiten als zentrales Lernmaterial bestehen, wie anhand der Zahlen aus Abschnitt 1.1 erkenntlich, zum Großteil aus Text. Dies hat zur Folge, dass sich die Studierenden während der Auseinandersetzung mit den Lerninhalten nicht mit anderen Dingen beschäftigen können, welche die Aufmerksamkeit ihrer Augen und Hände benötigen. Zusätzlich besteht oft das Problem, dass Abbildungen, Formeln und Tabellen, auf die im Text verwiesen wird, nicht direkt auf der Seite ersichtlich sind, auf der diese im Text erwähnt werden. Hierdurch ist oftmals Blättern bzw. Scrollen nötig, je nachdem ob die Kurseinheit in Papierform oder digital bearbeitet wird. Dies erschwert

---

<sup>2</sup>Kurseinheiten werden an der FernUniversität in Hagen auch als Studienbriefe bezeichnet.

zusätzlich das Verinnerlichen des in der Kurseinheit zu vermittelnden Inhalts (Mayer, 2009). Im Vergleich zum Frontalunterricht bringt die Vermittlung der Lerninhalte in Form von Kurseinheiten den Nachteil mit sich, dass bei Verständnisproblemen keine direkten Fragen gestellt werden können.

Dieses Problem tritt bei Lehrvideos ebenso auf. Hier ist im besten Fall ein asynchroner Austausch mittels einer Kommentarfunktion möglich. Ähnlich wie Kurseinheiten verlangen auch Lehrvideos die durchgehende visuelle Aufmerksamkeit des Studierenden. Nur durch ununterbrochenes Betrachten eines Videos kann ein Studierender sicherstellen, dass er alle dargestellten Inhalte wahrnimmt.

Im Gegensatz zu Lehrvideos, sind die Foren und der Chat ohnehin nur als zusätzliche Kommunikationswege für die Studierenden implementiert. Es besteht das grundsätzliche Problem, dass diese Funktionalitäten nicht direkt an die Lerninhalte gekoppelt sind und deswegen separat aufgerufen werden müssen, falls beim Lernen Fragen auftreten sollten.

Die in Moodle verfügbaren Tests dienen hingegen in ihrer Form nur zur reinen Selbstkontrolle. Dadurch, dass diese Tests nicht direkt während der Erarbeitung der Lerninhalte durch Kurseinheit oder Lehrvideos erfolgen kann, wird nicht unmittelbar überprüft, ob die Inhalte korrekt verstanden wurden.

*Virtual Classrooms* zählen wiederum zu den Lerninhalt vermittelnden Angeboten. Bedingt durch die Tatsache, dass der Unterricht in *Virtual Classrooms* in Echtzeit abgehalten wird, entsteht der Nachteil, dass der Studierende nur zu einem festgelegten Zeitpunkt die Lehrveranstaltung wahrnehmen kann. Außerdem verlangen *Virtual Classrooms*, genau wie Lehrvideos, die ständige Aufmerksamkeit des Studierenden. Im Gegensatz zu Lehrvideos schaffen *Virtual Classrooms* jedoch die Möglichkeit der synchronen Kommunikation.

Zusammenfassend handelt es sich bei den Kurseinheiten, Lehrvideos und *Virtual Classrooms* um die Lerninhalt vermittelnden Angebote. Kurseinheiten und Lehrvideos stellen hierbei asynchrone Lehrmethoden dar, während es sich bei den *Virtual Classrooms* um eine synchrone Lehrmethode handelt. Die asynchronen Angebote bringen im Gegensatz zum synchronen Angebot den Vorteil mit sich, dass diese zu jeder beliebigen Zeit wahrgenommen werden können. Dennoch haben diese drei Lehrangebote gemeinsam, dass die Studierenden ihnen zumindest die volle visuelle Aufmerksamkeit beim Lernen schenken müssen.

Somit ist mit dem aktuellen Lehrangebot beispielsweise auch kein Lernen während der sportlichen Betätigung möglich. Dabei führt leichte körperliche Betätigung während des Lernens nach einer Studie von Schmidt-Kassow et al. (2013) sogar zu einem besseren Lernergebnis. Stattdessen ist der Studierende weiterhin daran gebunden im Sitzen oder vor dem Bildschirm zu lernen. Indes haben mehrere Studien gezeigt, dass langes Sitzen negative Auswirkungen auf die gesundheitliche Verfassung mit sich bringt (Tremblay et al., 2011).

Aufgrund der Tatsache, dass 80% der Studierenden an der FernUniversität in Hagen neben dem Studium ebenfalls einer Arbeit nachgehen (FernUniversität in Hagen, 2018c), muss auch die dem Studierenden zur Verfügung stehende Zeit berücksichtigt werden. Zur bezahlten Arbeit kommt immer auch unbezahlte Arbeit hinzu. Diese betrug in den Jahren 2012/2013 im Durchschnitt ca. 24,5 Stunden in der Woche für Personen ab 18 Jahren. Als unbezahlte Arbeit gelten Haushaltstätigkeiten, wie Kochen, Putzen, Gartenpflege und Einkaufen, aber auch ehrenamtliche Tätigkeiten sowie Wegzeiten (Statistisches Bundesamt, 2015). Diese unbezahlte Arbeit kann aktuell zum Großteil nicht zum Lernen verwendet werden, da durch die heutigen Lernangebote stets die volle Aufmerksamkeit des Studierenden erforderlich ist. Beispielsweise ist es unmöglich, während des Fensterputzes eine Kurseinheit in ihrer aktuellen Repräsentation zu lesen, da die Putztätigkeit bereits die volle visuelle Aufmerksamkeit sowie den Einsatz der Hände erfordert. Diese Zeit könnte jedoch zum Lernen genutzt werden, sofern die Lerninhalte in einer anderen Form präsentiert würden, die sich auf andere Sinne beschränkt. Da der Hörsinn in vielen Alltagstätigkeiten, wie dem Putzen oder Einkaufen, nicht vorrangig benötigt wird, bietet sich dazu eine Repräsentation in auditiver Form an.

### 1.3 Zielsetzung und Forschungsfragen

Hauptziel des Vorhabens ist die Gestaltung einer auditiven Repräsentation der Lerninhalte und der Entwicklung eines Plugins innerhalb von Moodle zur Wiedergabe dieser alternativen Repräsentation. Die Wiedergabe soll in die Moodle-Plattform integriert werden, da diese mit ihrem Plugin-System und dem hohen Verbreitungsgrad an der FernUniversität in Hagen gute Voraussetzungen für die Bereitstellung neuer Lehrmethoden bietet.

Im Laufe dieser Arbeit soll die Frage beantwortet werden, wie eine Hyperaudio-Lernumgebung in Moodle zu gestalten ist. Dabei ergeben sich folgende untergeordnete Forschungsfragen:

- Wie kann den Studierenden mithilfe einer Hyperaudio-Lernumgebung ermöglicht werden mehr Zeit zum Lernen nutzen zu können?
- Wie lassen sich auditive Inhalte verständlich gestalten?
- Wie lassen sich Inhalte in der Hyperaudio-Lernumgebung darstellen, die nicht in auditiver Form abgebildet werden können?
- Wie können nicht-auditive Inhalte mit den auditiven Inhalten verknüpft werden?
- Wie lassen sich Interaktionen, die eine textuelle Darstellung der Lerninhalte bietet, in der Hyperaudio-Lernumgebung umsetzen?
- Wie kann der Austausch zwischen Studierenden und Lehrenden umgesetzt werden?

Ziel dieser Arbeit soll es also sein, den Studierenden eine auditive Repräsentation der Lerninhalte anzubieten, welche es ihnen ermöglichen soll, mehr Zeit zum Lernen nutzen zu können und dabei die Effizienz des Lernens zu erhöhen.

Es muss eine Lernumgebung gestaltet werden, welche es ermöglicht, die hauptsächlich auditiven Lerninhalte auf verständliche Art und Weise bereitzustellen. Daneben müssen auch Lerninhalte, welche nicht in auditiver Form abgebildet werden können, berücksichtigt werden. Akustische Signale können unterstützend eingesetzt werden, um Studierende auf nicht-auditive Zusatzinhalte aufmerksam zu machen, die ihre tiefergehende Aufmerksamkeit erfordern. Typische Nutzerinteraktionen mit textuellen Lernmedien, wie beispielsweise Lesezeichen und Notizen oder das Durchsuchen des Inhaltsverzeichnisses, sollen hierbei erhalten bleiben.

Zusätzlich zur Bereitstellung der Lerninhalte in alternativer Form soll die Hyperaudio-Lernumgebung auch den Kommunikationsaustausch zwischen Lehrenden und Studierenden ermöglichen und fördern. Hiermit soll dem allgemeinen Problem des mangelnden sozialen Kontakts beim Fernstudium begegnet werden (Kerres und Jechle, 2002). Um den Studierenden beim Lernen möglichst große Flexibilität zu bieten, ist eine mobile Verfügbarkeit der neugestalteten Lerninhalte erstrebenswert.

## 2 Grundlagen

Als Fundament für die weitere Arbeit sollen zunächst einige Grundlagen thematisiert werden. Um das Vorhaben und das Vorgehen dieser Arbeit besser nachvollziehen zu können, wird zunächst das Modell der *Tetrade der Medieneffekte* vorgestellt. Hierbei handelt es sich um eine Idee von Marshall McLuhan, welcher sich mit den Effekten beschäftigt, die ein Medium mit sich bringt. Er hat festgestellt, dass im Wesentlichen vier Effekte von Interesse sind, welche er mit den folgenden Fragen bestimmen will (McLuhan, 1977):

1. What does the medium enhance?
2. What does the medium make obsolete?
3. What does the medium retrieve that had been obsolesced earlier?
4. What does the medium reverse or flip into when pushed to extremes?

So stellt McLuhan (1977) fest, dass das Radio eine unmittelbare und auditive Art der Kommunikation beförderte. Im gleichen Moment wurde dadurch die Bedeutsamkeit von Printmedien geschwächt. Hierdurch hat die vorangegangene auditive Kommunikation, welche durch die Einführung von Printmedien obsolet wurde, wieder an Bedeutung gewonnen. Wenn man nun das Medium an sein Limit bringt, dann befördert dies die Entwicklung hin zum Fernsehen (McLuhan, 1977).

Analog hierzu soll das Ziel der Einführung des Hyperaudio-Plugins sein, Medien wie Podcasts und Hörbücher zu erweitern, um Kurseinheiten und Präsenzveranstaltungen überflüssig zu machen. Hierdurch würde das Bildungsradio wieder reaktiviert. Wenn Hyperaudio an seine Grenzen gebracht wird, würde dies zu Video beziehungsweise Hypervideo führen. Mit diesen Gedanken im Hinterkopf werden nun die Grundlagen für diese Arbeit vorgestellt. Die betrachteten Themen beschäftigen sich ebenfalls mit Medien und deren Effekten.

### 2.1 Moodle

Bei Moodle (Modular Object-Oriented Dynamic Learning Environment) handelt es sich um ein frei verfügbares Open-Source-Learningmanagementsystem (GNU Public License), mit dem Internetbasierte Kurse entwickelt und durchgeführt werden können (Moodle, 2015c). Ziel der Lernplattform ist es, den Lehrenden, Administratoren und Lernenden ein robustes, sicheres und integriertes System zu liefern, mit dessen Hilfe sie eine personalisierte Lernumgebung gestalten können (Moodle, 2018a). Unter dieser Zielsetzung ist Moodle als Lernplattform weltweit verbreitet und hat aktuell<sup>3</sup> 101.447 registrierte Seiten in 232 Ländern mit insgesamt mehr als 130 Millionen Benutzern (Moodle, 2018f).

Zugriff auf Moodle erhält der Nutzer über die Startseite, welche auf die eigenen Bedürfnisse angepasst werden kann. Auch kann Moodle so konfiguriert werden, dass die Startseite erst nach Anmeldung an der Login-Seite erfolgen kann. Die Grundstruktur von Moodle ist, wie in Abbildung 2.1 zu sehen, anhand von Kursbereichen und Kursen organisiert. Kurse werden wiederum als Seiten repräsentiert, auf welchen die Lehrenden Arbeitsmaterialien und Aktivitäten für die Studierenden bereitstellen können. Kurse werden üblicherweise in einzelne Kursabschnitte unterteilt, in welchen die Arbeitsmaterialien und Aktivitäten eingebunden werden. Kursseiten können durch Blöcke noch um weitere zusätzliche Informationen angereichert werden.

---

<sup>3</sup>Stand: 28.07.2018

## 2 Grundlagen

Ein Beispiel für eine Kursseite ist in Abbildung 2.2 anhand des Kurses „Einführung in Mensch-Computer-Interaktion“ zu sehen. Auf der linken Seite sind innerhalb der Navigation die einzelnen Kursabschnitte, hier als Kurseinheiten beziehungsweise Einführung benannt, sichtbar. Unterhalb der Navigation sowie auf der rechten Seite sind die verschiedenen Blöcke (z.B. „Aktivitäten“, „Suche in Foren“ oder „Neue Ankündigungen“) innerhalb des Kurses angeordnet. Im mittleren Bereich befinden sich untereinander die Beschreibungen der einzelnen Kursabschnitte mit Links zu den verwendeten Arbeitsmaterialien und Aktivitäten.

Die vorab beschriebenen Kurse werden wiederum innerhalb von Kursbereichen organisiert. Es ist auch ein mehrstufiges Kursbereichssystem umsetzbar (Moodle, 2015a).

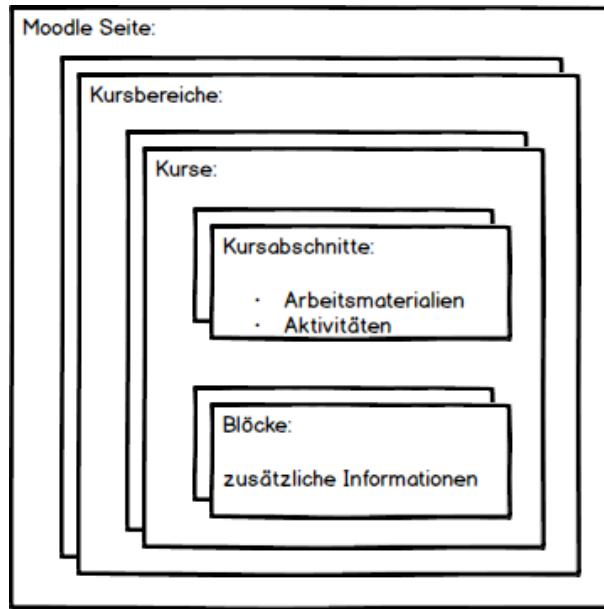


Abbildung 2.1: Schematischer Aufbau einer Moodle Seite

Technisch baut Moodle auf einem Aufbau aus Webserver und Datenbank unter Verwendung von PHP (PHP: Hypertext Preprocessor) auf. Um die zum Ziel gesetzte Personalisierbarkeit zu erreichen, setzt Moodle unter anderem auf ein Plugin-System. Plugins werden in über 50 verschiedene Plugin-Typen kategorisiert, wobei jeder dieser Typen dazu dient, einen speziellen Bereich von Moodle zu erweitern beziehungsweise anzupassen (Moodle, 2017b).

## 2.2 Kooperation im Lernumfeld

„Lernen ist in vieler Hinsicht ein sozialer Prozess, der kulturelle Einflüsse einschließt sowie soziale Aktivitäten und gemeinsames Problemlösen umfasst“ (Reinmann-Rothmeier und Mandl, 1995). Weitgefasst versteht man unter kooperativem Lernen eine Situation, in der zwei oder mehrere Personen zusammen lernen oder versuchen, zusammen zu lernen (Dillenbourg, 1999).

Während in dieser Arbeit, wie im deutschsprachigen Raum üblich, keine Unterscheidung zwischen dem *kooperativen* und dem *kollaborativen* Lernen vorgenommen wird, werden die Begriffe außerhalb des deutschsprachigen Raums häufig differenziert betrachtet (Reinmann-Rothmeier und Mandl, 2002). Eine differenzierte Betrachtung der beiden Begriffe liefert Dillenbourg et al. (1995). Demnach handelt es sich um *Kooperation*, wenn eine Problemlösung durch die Arbeitsteilung unter den Mitgliedern einer Gruppe erreicht wird, wobei jedes Mitglied für einen Teil der Problemlösung verantwortlich ist. Bei *Kollaboration* wird die Problemlösung hingegen durch gemeinsames Engagement der Gruppenmitglieder bei koordiniertem Vorgehen zur Problemlösung erreicht. Als Beispiel kann eine Gruppenarbeit mit der Aufgabe, mehrere Textabschnitte zusammenzufassen, betrachtet werden.

The screenshot shows the Moodle course page for "WS16/17 - Kurs 01697 Einführung in Mensch-Computer-Interaktion". The left sidebar contains a navigation menu with sections like "Navigation", "Startseite", "Meine Lernumgebungen", "Kurse", "Personen", and "Aktivitäten". The main content area displays course information, including the title, description, and various assignment and communication modules.

Abbildung 2.2: Kursseite des Kurses „Einführung in Mensch-Computer-Interaktion“  
(FernUniversität in Hagen, 2016)

Bei einem kooperativen Vorgehen wäre zunächst jeweils ein Gruppenmitglied für die Zusammenfassung genau eines Textabschnittes verantwortlich. In einem zweiten Schritt würden dann die einzelnen Teilergebnisse zu einem Ergebnis zusammengefasst. Bei einem kollaborativen Vorgehen wiederum würden alle Gruppenmitglieder die einzelnen Textabschnitte gemeinsam zusammenfassen.

Dem kooperativen Lernen werden positive Effekte zugesprochen, unter anderem bezüglich der Lernmotivation (Dillenbourg, 1999; Reinmann-Rothmeier und Mandl, 1995). So wird beispielsweise beim Lernen durch Lehren, einer speziellen Form des kooperativen Lernens, die Motivation der Beteiligten erhöht. Gleichzeitig werden sowohl beim Lehrenden als auch beim Lernenden effektive Lernprozesse in Gang gesetzt (Reinmann-Rothmeier und Mandl, 1995).

Reinmann-Rothmeier und Mandl (2002) beschreiben auch, welche Probleme bei netzbasierten Umgebungen entstehen können. So können neben technischen Problemen auch Widerstände seitens der Nutzer vorliegen. Oftmals ist es schwer, die Gruppenmitglieder dazu zu bringen, aktiv an den netzbasierten Szenarien teilzunehmen. Auch besteht das Problem, dass in netzbasierten Umgebungen weniger individuelles als kollektives Wissen geteilt wird als in Face-to-Face-Situationen. Aufgrund der fehlenden sozialen Hinweise reize und der Anonymität kann „es leicht zu unkontrollierter Kommunikation und heftigen Gefühlsausbrüchen, dem sog. Flaming, kommen“ (Reinmann-Rothmeier und Mandl, 2002).

## 2.3 Hypermedia

Bevor mit der genauen Konzeption und Implementierung des Hyperaudio-Plugins begonnen werden kann, werden *Hypermedia* im Allgemeinen betrachtet. Dabei soll zunächst eine Begriffsklärung

## 2 Grundlagen

durchgeführt werden. Aus diversen Erfahrungen mit *Hypermedia* können Rückschlüsse für das zu entwickelnde Moodle-Plugin und die zugrundeliegende Interpretation von *Hyperaudio* gezogen werden.

### 2.3.1 Grundbegriffe

Zu Beginn wird zunächst der Begriff *Multimedia* betrachtet. Unter *Multimedia* wird die Bereitstellung von Informationen mittels verschiedener Formate, beispielsweise Text, Audio, Bilder oder Video, bezeichnet (Mayer, 2009; Moos und Marroquin, 2010).

Der Begriff *Hypermedia* wurde das erste Mal von Ted Nelson 1965 verwendet (Nelson, 1965). In seinem Paper beschreibt er detailliert, was er sich unter einem *Hypertext* vorstellt. Hierunter versteht er ein Dokument bestehend aus geschriebenen oder bildhaften Inhalten, welche in solch einer komplexen Art und Weise miteinander verbunden sind, dass sie nicht mehr auf Papier dargestellt werden können. Es kann Zusammenfassungen, Karten über die Inhalte und deren Zusammenhänge, Annotationen, Ergänzungen oder Anmerkungen von Wissenschaftlern, die das Dokument begutachtet haben, enthalten. Nelson beschreibt das Kriterium für den Präfix *hyper* damit, dass diese Objekte nicht durch eine Konvertierung in ein einfaches lineares Medium, wie beispielsweise eine Zeichenkette umgewandelt werden können. Der wesentliche Punkt ist also, dass es sich beim Arbeiten mit *Hypermedia* um ein nicht-lineares Vorgehen handelt.

Genauer betrachtet stellt das, was Nelson (1965) sich als *Hypertext* vorgestellt hatte, nach Nielsen (2013) bereits eine Form von *Hypermedia* dar. Auch wenn die beiden Begriffe *Hypertext* und *Hypermedia* synonym verwendet werden können, werden bei strikter Betrachtung bei *Hypertext* ausschließlich Texte miteinander verbunden, während bei *Hypermedia* auch andere Medien eingebunden werden können. Gemeinsam haben beide Arten jedoch, dass der Betrachter keinen linearen Weg vorgegeben hat, sondern von einem Knoten zum anderen springen kann und sich somit seinen Weg selbst aussucht. Demnach stellt *Hypermedia* eine nicht-lineare Variante von *Multimedia* dar.

Nach dieser Logik handelt es sich bei *Hyperaudio* in seiner klassischen Form eigentlich um reine Audiosequenzen, die miteinander verknüpft sind, wobei der Zuhörer selbst entschieden kann, in welcher Reihenfolge er diese abspielt (Zumbach und Kroeber, 2006).

### 2.3.2 Lernen mit Hypermedia

Wissenschaftler beschäftigen sich schon seit vielen Jahren damit, festzustellen, welche Effekte der Einsatz von *Multimedia*, *Hypertext* und *Hypermedia* auf den Lernerfolg von Lernenden hat.

Mayer (2009) konnte im Laufe seiner Untersuchungen positive Effekte beim Einsatz von *Multimedia* nachweisen. So schnitten Studenten bei einem Transfertest besser ab, wenn sie mit Text und Bildern lernten, als wenn sie ausschließlich mit Text lernten. Mayer (2009) knüpft dieses Ergebnis aber an folgende Prinzipien, welche beim Einsatz von *Multimedia* befolgt werden müssen:

- Kohärenz: irrelevante Wörter, Töne und Bilder sollten vermieden werden
- Signalisieren: essentielle Inhalte sollten hervorgehoben sein
- Redundanz: Text sollte ausschließlich auditiv statt auditiv und visuell in Multimediapräsentationen wiedergegeben werden
- Räumliche Nähe: zusammengehörige Texte und Bilder sollten nah beieinander statt weit entfernt auf der Seite beziehungsweise dem Bildschirm dargestellt werden
- Zeitliche Nähe: zusammengehörige Texte und Bilder sollten gleichzeitig statt nacheinander auf der Seite beziehungsweise dem Bildschirm dargestellt werden
- Segmentierung: temporeiche, komplexe Multimedia–Lektionen sollten in benutzerfreundlichen Portionen statt im Ganzen präsentiert werden

- Vorbereitung: der Lernende sollte die Begrifflichkeiten und Merkmale der Lektion kennen
- Modalität: Text sollte auditiv statt visuell wiedergegeben werden
- Personalisierung: Texte sollten in dialogorientierterem Stil statt in formellem Stil wiedergegeben werden
- Stimme: Text sollte von einer freundlichen menschlichen Stimme statt einer Computerstimme wiedergegeben werden

Im Gegenzug verweisen Moos und Marroquin (2010) auf Arbeiten, nach denen eine Herausforderung bei *Multimedia* und somit auch bei *Hypermedia* darin besteht, dass die kognitive Aufnahmekapazität der Studierenden überschritten werden kann, wenn Informationen sowohl aus einem Text als auch aus einem Diagramm entnommen werden sollen (Mayer und Moreno; van Merriënboer und Ayres, nach Moos und Marroquin (2010)). Dies beruht auf der Annahme der Cognitive Load Theory, welche dem Arbeitsgedächtnis nur eine begrenzte Kapazität zuspricht (Sweller; van Merriënboer und Sweller, nach Moos und Marroquin (2010)). Diesem Problem ist laut Mayer (2009) mit den oben genannten Prinzipien zu begegnen, da diese darauf abzielen, die begrenzte Kapazität besser zu nutzen.

Ebenso wie *Multimedia* bietet *Hypertext* Vorteile. Diese bestehen beispielsweise darin, dass der Studierende den Lernweg bestimmen kann, der am besten auf seine Bedürfnisse angepasst ist. Auf der anderen Seite ist hierzu aber eine ausreichende Vorkenntnis im entsprechenden Lernbereich notwendig, um die Entscheidung treffen zu können, wie dieser Weg aussehen soll. Des Weiteren wirkt sich auch ein fehlendes Interesse des Studierenden negativ auf die Effektivität der *Hypertext*-Lernumgebung aus (Lawless und Kulikowich, nach Moos und Marroquin (2010)).

Es ist nun also nicht verwunderlich, dass *Hypermedia* als Verschmelzung von *Multimedia* und *Hypertext* ebenfalls einige Herausforderungen mit sich bringt (Moos und Marroquin, 2010). Scott und Schwartz (nach Moos und Marroquin (2010)) fordern für das Lernen mit *Hypermedia* eine Balance zwischen effektiver Navigation und Inhaltsverständnis. Dies soll durch Prozesse zur Überwachung des eigenen Lernfortschritts erreicht werden, wohingegen Untersuchungen ergeben haben, dass viele Studierende Schwierigkeiten dabei haben, diese Prozesse korrekt anzuwenden (Moos und Marroquin, 2010).

### 2.3.3 Hyperaudio-Repräsentation von Kurseinheiten

*Hyperaudio* als Sonderform von *Hypermedia* bringt spezielle Probleme mit sich. So weisen Donker und Blenn (2007) auf die Herausforderungen bei der Gestaltung von *Hyperaudio*-Anwendungen hin, welche unter anderem darin bestehen, dem Hörer die Hyperlinks innerhalb einer *Hyperaudio*-Anwendung sinnvoll darzustellen. Bei ihrer Untersuchung kamen Donker und Blenn (2007) zu dem Ergebnis, „dass sowohl eine Verdeutlichung von Links durch das Voranstellen eines Tons als auch die Variante durch das Ändern des Alters des Sprechers zu guten Ergebnissen hinsichtlich des Erkennens des Links führen.“ Die genannten vorangestellten Töne werden auch als *Audio Cues* bezeichnet.

Bei einer auditiven Repräsentation von Kurseinheiten stellen sich einige Herausforderungen, denen ebenfalls mit den Erkenntnissen von Donker und Blenn (2007) begegnet werden kann:

- Wie können spezielle Textpassagen (Auflistungen, Beispiele, Aufgaben, Definitionen, Algorithmen etc.) nachvollziehbar wiedergegeben werden?
- Wie können Zitate kenntlich gemacht werden?
- Wie soll mit Ausdrücken in Klammern umgegangen werden?
- Wie soll der Hörer auf Fußnoten hingewiesen werden?
- Wie sollen Marginalien bei der Vertonung berücksichtigt werden?
- Wie kann der Hörer über Querverweise informiert werden?

## 2 Grundlagen

In Anlehnung an Donker und Blenn (2007) könnte mit dem Einsatz verschiedener Audio Cues und Sprecher gearbeitet werden. Auch wäre ein Einsatz von *Resonance Audio*<sup>4</sup> von Google denkbar. Damit wäre es möglich, durch unterschiedliche Positionierung im Raum oder durch Verwendung verschiedener Audio-Kanäle den gerade gesprochenen Inhalt zu kategorisieren. Zusätzlich sollten auch bei der Vertonung von Kurseinheiten die Prinzipien von Mayer (2009) (vgl. Abschnitt 2.3.2) beachtet werden.

Unter Berücksichtigung der soeben genannten Herausforderungen und Lösungsideen müssen für eine auditive Repräsentation der Kurseinheiten deren Inhalte entsprechend aufbereitet werden. Inhalte, die nicht auditiv umgesetzt werden können, können in visueller Darstellung zum passenden Zeitpunkt an die auditiven Inhalte annotiert werden.

In Ergänzung zu Zumbach und Kroeber (2006) wird in dieser Arbeit unter *Hyperaudio* ein Audioinhalt verstanden, der mittels der Erweiterung durch Kommunikations- und Interaktionsmöglichkeiten (vgl. Abschnitt 1.3) sowie visuelle Inhalte um Multimedia- und Hypermedia-Elemente ergänzt wird. Die Kommunikationsmöglichkeiten sollen die Lehrenden und Studierenden in die Lage versetzen, effektiv miteinander zu kommunizieren, um die Vorzüge des kooperativen Lernens (vgl. Abschnitt 2.2) genießen zu können. Die Interaktionsmöglichkeiten sollen wiederum dafür sorgen, dass die Möglichkeiten, die eine textuelle Darstellung der Lerninhalte bietet, wie beispielsweise das Markieren von Inhalten mit Pagemarkern, erhalten bleiben.

Ein Hyperaudio-Dokument im Sinne dieser Arbeit besteht aus auditiven Inhalten und annotierten visuellen Inhalten. Bei den Hyperaudio-Dokumenten handelt es sich somit eigentlich um Multimedia-Dokumente. Erst die Kommunikations- und Interaktionsmöglichkeiten führen dazu, dass der *Hypermedia*-Aspekt erfüllt wird.

Der Vorteil dieser Betrachtungsweise von *Hyperaudio* liegt darin, dass die Herausforderungen der Navigation durch den Nutzer, die in Verbindung mit *Hypermedia* bzw. *Hypertext* normalerweise auftreten, nicht besonders prägnant sind, da es sich vorrangig um ein lineares Audio-Dokument handelt. Im zu entwickelnden Moodle-Plugin wird dem Studierenden in erster Linie ein linearer Audioinhalt vorgespielt, der um Multimedia-Elemente ergänzt wird. Die herausfordernden Elemente von *Hypermedia* kommen jedoch erst durch die Einbindung der Kommunikations- und Interaktionsmöglichkeiten ins Spiel. Dementsprechend stellt das Hyperaudio-Plugin einen guten Kompromiss zwischen den verschiedenen Lehrmethoden dar.

## 2.4 Zusammenfassung

Zu Beginn dieses Kapitels wurde zunächst auf Medien und deren Effekte eingegangen. Darauf aufbauend wurden die gewünschten Effekte, welche durch die Einführung der Hyperaudio-Lernumgebung erzielt werden sollen, festgelegt. Anschließend wurde ein Überblick über die Lernplattform Moodle gegeben, wobei dessen technische und strukturelle Gegebenheiten beleuchtet wurden. Im nächsten Schritt wurde der Begriff der Kooperation im Lernumfeld, dessen generelle Vorteile sowie dessen Probleme in netzbasierten Umgebungen beleuchtet. Dies soll dazu dienen die Bedeutung der Kommunikation und der damit verbunden Probleme für das Hyperaudio-Plugin aufzuzeigen. Um ein grundlegendes Verständnis für die Begrifflichkeit *Hyperaudio* zu schaffen, wurden die Begriffe *Multimedia*, *Hypermedia*, *Hypertext* sowie *Hyperaudio* erläutert und deren Vor- und Nachteile anhand vorhandener Studien dargelegt. Als Ergebnis der zuvor erarbeiteten Grundlagen wurden die Grundgedanken für die Repräsentation von Kurseinheiten im Hyperaudio-Format festgehalten, wobei auch auf die Probleme bei der Vertonung der Kurseinheiten und deren möglichen Lösungen eingegangen wurde.

---

<sup>4</sup><https://developers.google.com/resonance-audio/>

# 3 Analyse

Für die Konzeption und Implementierung des Hyperaudio-Plugins ist zunächst eine Analyse notwendig. Basierend auf einer Zielgruppenanalyse sollen Anforderungen an das Plugin bestimmt und priorisiert werden. Darauf aufbauend werden Möglichkeiten und Technologien zur Umsetzung des Hyperaudio-Plugins begutachtet.

## 3.1 Zielgruppe

Im ersten Schritt soll die Zielgruppe anhand von *Personas* und deren *User Stories* festgelegt werden. Diese sollen dann die Grundlage für die Definition der Anforderungen im nächsten Abschnitt darstellen.

### 3.1.1 Personas

Unter *Personas* werden fiktive Benutzer verstanden, für welche das Programm, in unserem Fall das Moodle-Plugin, designt wird (Cooper et al., 2004). Jeder Persona wird eine Rolle im Zusammenhang mit der Anwendung zugewiesen. Darüber hinaus wird die Persona ausreichend beschrieben, damit sich leicht in die Person hineinversetzt werden kann (Cohn, 2004). Dieses Vorgehen hilft dabei, möglichst authentische *User Stories* zu generieren, ohne auf echte Benutzer zurückgreifen zu müssen. Darüber hinaus ist festzustellen, dass die realen Anwender zwar Problemstellungen identifizieren können, aber nicht unbedingt in der Lage sind, Anforderungen zur Lösung dieser Probleme zu formulieren (Cooper et al., 2004). Für die Analyse und Konzeption des Hyperaudio-Plugins wird mit folgenden Personas gearbeitet:

*Prof. Dr. Karolin Schröder* ist verantwortlich für den Kurs „Einführung in die Wirtschaftsinformatik“. In diesem Kurs werden bereits erfolgreich Hyperaudio-Dokumente eingesetzt. Nachdem Prof. Dr. Schröder mit dem Start des nächsten Semesters überarbeitete Kurseinheiten anbietet, müssen nun auch die zugehörigen Hyperaudio-Dokumente auf die Notwendigkeit einer Überarbeitung hin überprüft werden. Die veralteten Hyperaudio-Dokumente müssen dann durch neuere Versionen ersetzt werden.

*Dr. Julian Schmidt* ist wissenschaftlicher Mitarbeiter und Betreuer für den Kurs „Marketing“, der ebenfalls das Lernen mittels Hyperaudio-Dokument anbietet. Dr. Julian Schmidt ist unter anderem für die Betreuung dieser verantwortlich und ist derjenige, der hier Rede und Antwort steht.

*Laura Ebert* ist 35 Jahre alt, verheiratet und hat drei Kinder. Sie geht halbtags ihrem Beruf als Anwendungsentwicklerin nach, zu dem sie 30 Minuten mit dem Bus pendelt. Neben der Arbeit kümmert sie sich zusammen mit ihrem Mann um den Haushalt und die Kinder. Laura studiert in Teilzeit den Bachelorstudiengang Informatik im ersten Semester. Das Semester hat erst vor einigen Wochen begonnen und sie entdeckt gerade Moodle für sich. Hierbei ist sie auf die Hyperaudio-Dokumente

### 3 Analyse

gestoßen und hat sich fest vorgenommen, sich im Laufe des Semesters mit deren Hilfe mit den Lerninhalten auseinanderzusetzen.

*Max Lustig* ist 24 Jahre alt, ledig und hat sich nach einer abgeschlossenen Ausbildung zum Informatikkaufmann dazu entschlossen, neben dem Beruf zu studieren. Zur Arbeit kommt er in wenigen Minuten zu Fuß. Viel Zeit verbringt er jedoch im Fitnessstudio mit Kraft- und Ausdauertraining. Er absolviert ein Vollzeitbachelorstudium der Wirtschaftsinformatik und befindet sich kurz vor der Prüfungsphase zum Ende des dritten Semesters. Max möchte sich nun auf die Klausur des Moduls „Investition und Finanzierung (BWL II)“, welche in zwei Wochen stattfindet, intensiv vorbereiten. Im Laufe des Semesters hat er bereits ausgiebig die neuen Hyperaudio-Dokumente zum Erreichen des Lernziels genutzt.

#### 3.1.2 User Stories

Unter Zuhilfenahme der entwickelten Personas werden im nächsten Schritt *User Stories* formuliert. *User Stories* beschreiben, wie die klassischen *Use Cases*, Anforderungen an ein Softwaresystem. Sie sind dabei im Vergleich wesentlich oberflächlicher und ungenauer formuliert als *Use Cases* (Werde mann, 2017). Erst im Laufe der Entwicklung werden *User Stories* konkreter und dienen am Ende dazu, das Entwicklungsergebnis zu validieren. Beim Erstellen von *User Stories* ist zu beachten, dass sogenannte *Epics*, das sind *User Stories* mit sehr großem Umfang, wenn möglich in kleinere *User Stories* aufgesplittet werden. Unter anderem ist zu beachten, dass die *User Stories* keine Abhängigkeiten untereinander aufweisen und dass deren Erfüllung überprüfbar ist. *User Stories* können im *Connextra Format* festgehalten werden, welches wie folgt aufgebaut ist (Cohn, 2004):

*Ich als (Rolle) möchte (Funktion), um (Nutzen).*

Mit den Unterschieden und Einsatzzwecken von Personas und Rollen beschäftigt sich Constantine (2006). Grundsätzlich ist demnach festzustellen, dass Personas eher aus dem *User-centered design* (Norman und Draper, 1986), Benutzerrollen dagegen aus dem *Usage-centered design* (Constantine, 1996) motiviert sind. Während sowohl Personas als auch Benutzerrollen durchaus nützlich sind, um ein Verständnis von den Nutzern eines Systems zu erhalten, unterscheiden sie sich nach Constantine (2006) in ihrer Philosophie und Relevanz für das Interaktionsdesign. Im Gegensatz zu Personas, die wie bereits im vorangegangenen Abschnitt 3.1.1 beschrieben dazu dienen, die Nutzersicht anhand möglichst realer Personen darzustellen, sollen Benutzerrollen in einer wesentlich technischeren Sicht ein abstrahiertes Modell für die Art und Weise, in der Nutzer mit dem System interagieren, bilden (Constantine, 2006).

Für die weitere Analyse soll das Beste beider Welten vereinbart werden. Um die User Stories möglichst anschaulich zu halten, werden sie anhand der vorgestellten Personas definiert. Gleichzeitig wird eine Aufteilung in Benutzerrollen vorgenommen, welche die Grundlage für die Ableitung von Anforderungen aus den User Stories bilden soll.

Für die Konzeption des Hyperaudio-Plugins ergeben sich im Wesentlichen zwei Benutzerrollen wie in Abbildung 3.1 dargestellt. *Administrierende* sind diejenigen Anwender, die Lerninhalte in Form von Hyperaudio-Dokumenten bereitstellen und verwalten. Diese Rolle kann nur von Lehrenden eingenommen werden. Die Gruppe der *Nutzenden* kann derweil aus Lehrenden sowie Studierenden bestehen, die mit Hyperaudio-Dokumenten interagieren möchten.



Abbildung 3.1: Benutzerrollen

Für das Angebot einer Hyperaudio-Lernumgebung lassen sich folgende User Stories festhalten:

- US-1: Als Administrierende möchte Prof. Dr. Karolin Schröder ein neues Hyperaudio-Dokument in ihrem Kurs „Einführung in die Wirtschaftsinformatik“ zur Verfügung stellen, um den Studierenden neue Lerninhalte bereitzustellen.
- US-2: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente aus ihrem Kurs „Einführung in die Wirtschaftsinformatik“ löschen können, um veraltete Informationen zu entfernen.
- US-3: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente aus ihrem Kurs „Einführung in die Wirtschaftsinformatik“ im Sommersemester in den darauffolgenden Kurs im Wintersemester übernehmen, um diese nicht erneut erstellen zu müssen.
- US-4: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente anderer Kurse in ihren Kurs „Einführung in die Wirtschaftsinformatik“ übernehmen, um auf die hervorragende Arbeit anderer Lehrender zurückgreifen zu können, da sich die Themen mit ihrem Kurs überschneiden.
- US-5: Als Administrierende möchte Prof. Dr. Karolin Schröder Erkenntnisse daraus gewinnen, wie die Hyperaudio-Dokumente des Kurses „Einführung in die Wirtschaftsinformatik“ von Studierenden genutzt werden, um Verbesserungspotenzial auszumachen.
- US-6: Als Administrierender möchte Dr. Julian Schmidt ein vorhandenes Hyperaudio-Dokument in dem von ihm betreuten Kurs „Marketing“ überarbeiten, um einen Fehler zu beseitigen.
- US-7: Als Nutzende möchte Prof. Dr. Karolin Schröder die bereits vorhandenen Hyperaudio-Dokumente aus ihrem Kurs „Einführung in die Wirtschaftsinformatik“ wiedergeben, um diese auf ihre Richtigkeit zu überprüfen.
- US-8: Als Nutzende möchte Prof. Dr. Karolin Schröder die Kommentare zu einem Hyperaudio-Dokument lesen und beantworten können, um auf Fragen von Studierenden einzugehen.
- US-9: Als Nutzende möchte Prof. Dr. Karolin Schröder eine Notiz zu einem Hyperaudio-Dokument machen, um ihren Gedanken festzuhalten und später darauf zurückgreifen zu können.
- US-10: Als Nutzender möchte Dr. Julian Schmidt eine gefundene Erklärungslücke in einem Hyperaudio-Dokument durch einen Kommentar zum entsprechenden Zeitpunkt schließen, um eventuellen Fragen der Studierenden vorzukommen.
- US-11: Als Nutzende möchte Laura Ebert mittels Hyperaudio-Dokument lernen, um die Zeit während Haushaltarbeiten, wie dem Bügeln, Kochen oder Putzen, und dem Pendeln sinnvoller zu nutzen.
- US-12: Als Nutzende möchte Laura Ebert erfahren, welche Hyperaudio-Dokumente in den von ihr belegten Kursen angeboten werden, um herauszufinden, mit welchen Mitteln sie sich auf die anstehenden Prüfungen vorbereiten kann.

### 3 Analyse

- US-13: Als Nutzende möchte Laura Ebert einen Kommentar verfassen, um dem Kursbetreuer und den anderen Studierenden eine Frage zu stellen.
- US-14: Als Nutzende möchte Laura Ebert ein Lesezeichen setzen, wenn eine klausurrelevante Thematik erklärt wird. Bei der Prüfungsvorbereitung möchte sie anhand dieser Lesezeichen diejenigen Themen erkennen, mit welchen sie sich besonders intensiv beschäftigen möchte.
- US-15: Als Nutzende möchte Laura Ebert ein Lesezeichen löschen, da sie den markierten Lerninhalt inzwischen beherrscht. Anhand der übrigen Lesezeichen möchte sie schnell erkennen, wo für sie noch Lernbedarf besteht.
- US-16: Als Nutzende möchte Laura Ebert eine Notiz erstellen, um ein Beispiel zu dem genannten Sachverhalt festzuhalten, sodass sie die Thematik beim nächsten Mal einfacher nachvollziehen kann.
- US-17: Als Nutzende möchte Laura Ebert die Wiedergabe eines Hyperaudio-Dokuments beenden und am nächsten Tag automatisch an derselben Stelle fortsetzen können, um das Lernen schnell wiederaufnehmen zu können.
- US-18: Als Nutzende möchte Laura Ebert die Hyperaudio-Angebote mit ihrem Smartphone in Anspruch nehmen, um auch die Zeit während des Pendelns zum Lernen nutzen zu können.
- US-19: Als Nutzender möchte Max Lustig eine alte Notiz bearbeiten, um einen Schreibfehler zu korrigieren.
- US-20: Als Nutzender möchte Max Lustig eine alte Notiz löschen, da er inzwischen Lernfortschritte gemacht hat und auf diese Notiz verzichten kann.
- US-21: Als Nutzender möchte Max Lustig schnell erkennen welche Inhalte im Hyperaudio-Dokument behandelt werden, um eine Erklärung eines bestimmten Themas zu finden.
- US-22: Als Nutzender möchte Max Lustig nach Textinhalten in Kommentaren suchen können, um schnell Erklärungen zu finden.
- US-23: Als Nutzender möchte Max Lustig die Kommentare nach Erstellungsdatum sortieren können, um sich einen Überblick über die neuesten Aktionen zu verschaffen.
- US-24: Als Nutzender möchte Max Lustig die Kommentare und persönlichen Notizen zu den Annotationszeitpunkten zuordnen können, um diese bei der Wiedergabe verfolgen zu können.
- US-25: Als Nutzender möchte Max Lustig öffentliche Kommentare und persönliche Notizen getrennt betrachten können, um die öffentliche Diskussion verfolgen beziehungsweise die eigenen Anmerkungen isoliert betrachten zu können.
- US-26: Als Nutzender möchte Max Lustig auf Kommentare antworten können, um sich mit den Studierenden und Lehrenden auszutauschen.
- US-27: Als Nutzender möchte Max Lustig erkennen, welche Hyperaudio-Dokumente er zuletzt abgespielt hat, um seinen Lernfortschritt im Auge zu behalten.
- US-28: Als Nutzender möchte Max Lustig besonders hilfreiche Hyperaudio-Dokumente als Favoriten speichern, um diese schnell als solche identifizieren zu können.
- US-29: Als Nutzender möchte Max Lustig die Markierung als Favorit entfernen können, wenn der Inhalt für ihn nicht mehr von Interesse ist.
- US-30: Als Nutzender möchte Max Lustig auf seinem Tablet Zugang zu Hyperaudio-Dokumenten haben, um die Zeit auf dem Laufband gleichzeitig zum Lernen nutzen zu können.

Natürlich könnten noch mehr User Stories formuliert werden. Die 30 genannten User Stories sollen jedoch für diese Arbeit die Basis der Anforderungsdefinition bilden. Auch die Evaluation des Entwicklungsergebnisses soll anhand der User Stories durchgeführt werden.

## 3.2 Anforderungsdefinition

Basierend auf den Personas, Rollen und User Stories können nun die Anforderungen für das Hyperaudio-Plugin definiert werden. Hierbei werden aus einer oder mehreren User Stories jeweils eine oder mehrere Anforderungen abgeleitet und zugleich mit einer Priorität versehen. Für die Priorisierung stehen die drei Prioritätsstufen *niedrig*, *mittel* und *hoch* zur Verfügung. Während im Rahmen eines partizipativen Entwicklungsansatzes eine Bewertung der User Stories oder Anforderungen durch zukünftige Nutzer in Betracht gezogen werden kann, soll in dieser Arbeit die Zielsetzung aus Kapitel 1 als Orientierung für die Priorisierung der Anforderungen dienen.

### 3.2.1 Anforderungen der Administrierenden

Es wird mit der Definition der Anforderungen der Administrierenden begonnen, welche in Tabelle 3.1 festgehalten werden.

Aus US-1 kann die Anforderung des Erstellens von Hyperaudio-Dokumenten abgeleitet werden. Dass bestehende Hyperaudio-Dokumente auch bearbeitet und gelöscht werden können sollen, ergibt sich aus US-6 und US-2. Zusammen stellen diese Anforderungen die Grundfunktionalitäten für die alternative Repräsentation der Lerninhalte dar und werden dementsprechend mit der Prioritätsstufe *hoch* versehen.

Aus US-3 und US-4 ergibt sich die Anforderung, dass Hyperaudio-Dokumente in einen anderen Kurs übernommen werden können sollen, sei es der gleiche Kurs im nächsten Semester oder ein anderer Kurs. Diese Anforderung zählt nicht zu den Grundfunktionalitäten, kann das Verwalten von Hyperaudio-Dokumenten jedoch vereinfachen und wird daher mit der Prioritätsstufe *mittel* bewertet.

Der Wunsch, Erkenntnisse aus der Nutzung von Hyperaudio-Dokumenten durch die Studierenden zu erhalten (US-5), schlägt sich in der Anforderung nach statistischen Auswertungsmöglichkeiten nieder. Für diese Anforderung wird die Priorität *niedrig* vergeben, da es sich um ergänzende Metainformationen handelt.

Tabelle 3.1: Anforderungen von Administrierenden

Nr.	Anforderung	Priorität
1	Erstellen eines Hyperaudio-Dokuments	hoch
2	Bearbeiten eines Hyperaudio-Dokuments	hoch
3	Löschen eines Hyperaudio-Dokuments	hoch
4	Übernahme eines Hyperaudio-Dokuments in einen anderen Kurs	mittel
5	Statistische Auswertungen über die Nutzung der Hyperaudio-Dokumente	niedrig

### 3.2.2 Anforderungen der Nutzenden

Die aus den User Stories der Nutzenden abgeleiteten Anforderungen sind in Tabelle 3.2 festgehalten. Die Priorisierung wird dabei nach folgenden Kriterien vorgenommen:

- *hoch*: Basisfunktionalität zum Erreichen der Zielsetzung der Arbeit
- *mittel*: Verbesserung der Interaktion mit Hyperaudio-Dokumenten und Annotationen
- *niedrig*: Vereinfachter Zugriff auf Hyperaudio-Dokumente

### 3 Analyse

Aus US-7, US-11, US-18 und US-30 ergibt sich die grundlegende Anforderung, Hyperaudio-Dokumente abspielen zu können. US-11, US-18 und US-30 führen zudem zur Anforderung der Audio Cues, mithilfe derer auf annotierte Zusatzinhalte<sup>5</sup> hingewiesen wird. Diese Audio Cues wären zwar auch für Kommentare, Notizen oder Lesezeichen denkbar, doch würde dies selbst bei Verwendung verschiedener Audio Cues dazu führen, dass der Studierende mit unnötigen Informationen (im Sinne des zu vermittelnden Lerninhaltes) überhäuft würde. Dies widerspricht jedoch Mayers Prinzipien für Multimedia (Mayer, 2009). Die Audio Cues für Zusatzinhalte können zu den Basisfunktionalitäten des Hyperaudio-Plugins gezählt werden, da erst dadurch das Ziel der größeren zeitlichen Flexibilität beim Lernen erreichbar wird (vgl. Abschnitte 1.3 und 2.3.3). Um Inhalte schnell auffinden zu können, wie in US-21 gefordert, ist zudem eine Übersicht über annotierte Zusatzinhalte nützlich.

Basierend auf US-8, US-10, US-13, US-24 und US-26 lässt sich die Anforderung an eine Kommentarfunktion ableiten, die über Möglichkeiten zum Erstellen, Anzeigen und Beantworten von Kommentaren verfügen muss. An dieser Stelle kann die Interaktion durch eine Suchfunktion innerhalb der Kommentare verbessert werden (vgl. US-22). In den *User Stories* US-9, US-16, US-19, US-20 und US-24 werden Wünsche bezüglich einer Notizfunktion formuliert. Diese lässt sich aufschlüsseln in die Anforderungen zum Erstellen, Anzeigen, Bearbeiten und Löschen von Notizen. Die Notizfunktion spiegelt das Ziel der Erhaltung typischer Nutzerinteraktionen mit textuellen Lernmedien wieder und kann das Lernen für die Studierenden erleichtern (Scutter et al., 2010). Ähnlich sind die Anforderung zum Erstellen, Anzeigen und Löschen von Lesezeichen aus US-14 und US-15 zu bewerten. Obwohl Lesezeichen, ebenso wie die Notizfunktion, zu den Basisfunktionalitäten des Plugins gezählt werden können, wird die Lesezeichenfunktion mit der Priorität *mittel* versehen. Das Herabsetzen der Priorität wird dadurch begründet, dass ein Lesezeichen im Wesentlichen einer Notiz ohne textuellen Inhalt entspricht und durch die höher priorisierte Notizfunktion abgebildet werden kann. Aus US-23, US-24 und US-25 ergibt sich zudem der Wunsch nach Filter- und Sortierungsmöglichkeiten.

Der Wunsch nach einer Favoritenfunktion für Hyperaudio-Dokumente ergibt sich aus US-28 und US-29. Daraus resultieren die Anforderungen, Favoriten zu setzen, anzuzeigen und zu löschen. Übersichten über Hyperaudio-Dokumente werden in US-12 und US-27 gefordert: im ersten Fall eine Übersicht über alle Hyperaudio-Dokumente der belegten Kurse und im zweiten Fall eine Übersicht über die zuletzt abgespielten Hyperaudio-Dokumente. Sowohl die Favoritenfunktion als auch die Übersichten stellen eine reine Optimierung der Navigation dar und verhelfen somit zu einem vereinfachten Zugriff auf Hyperaudio-Dokumente. US-17 bringt die Anforderung für eine Funktion zum Fortsetzen unterbrochener Wiedergaben bei folgenden Aufrufen in Moodle hervor. Auch diese Funktion vereinfacht die Navigation zum gewünschten Hyperaudio-Dokument beziehungsweise dessen Inhalt.

Das Verlangen, Hyperaudio-Dokumente auch auf einem Smartphone oder Tablet nutzen zu können, wie in US-18 und US-30 beschrieben, resultiert in der Anforderung zur Unterstützung mobiler Endgeräte. Da es sich um eine Verbesserung der Interaktionsmöglichkeiten handelt, wird der Anforderung die Priorität *mittel* zugewiesen.

## 3.3 Möglichkeiten der Moodle-Plugin-Entwicklung

Nachdem die Anforderungen an das Hyperaudio-Plugin definiert wurden, wird sich nun den Möglichkeiten bei der Entwicklung von Moodle-Plugins zugewendet. Für den Betrieb von Moodle werden ein Webserver, eine MySQL- oder PostgreSQL-Datenbank und PHP vorausgesetzt (Moodle, 2018e). Moodle unterstützt die Verwendung von JavaScript und die Einbindung von Thirdparty-Frameworks (Wild, 2017). Dies gepaart mit den Möglichkeiten durch den Einsatz von PHP bietet bei der Entwicklung ausreichend Möglichkeiten, um das gewünschte Plugin umzusetzen.

Des Weiteren wird die Plugin-Entwicklung von Moodle mithilfe der sogenannten *Core APIs* (Application Programming Interfaces) unterstützt. Moodle bietet für fast jeden Anwendungszweck eine

<sup>5</sup>Zusatzinhalte sind die nicht in auditiver Form repräsentierbaren Inhalte von Kurseinheiten.

Tabelle 3.2: Anforderungen der Nutzenden

Nr.	Anforderung	Priorität
1	Wiedergabe von Hyperaudio-Dokumenten	hoch
2	Hinweise auf die Darstellung von annotierten Zusatzinhalten	hoch
3	Übersicht über annotierte Zusatzinhalte	mittel
4	Kommentarfunktion bei Hyperaudio-Dokumenten	mittel
4.1	Erstellen von Kommentaren	hoch
4.2	Anzeigen von Kommentaren	hoch
4.3	Antworten auf Kommentare	hoch
4.4	Suchfunktion innerhalb der Kommentare	mittel
5	Notizfunktion bei Hyperaudio-Dokumenten	mittel
5.1	Erstellen von Notizen	hoch
5.2	Anzeigen von Notizen	hoch
5.3	Bearbeiten von Notizen	hoch
5.4	Löschen von Notizen	hoch
6	Lesezeichenfunktion bei Hyperaudio-Dokumenten	mittel
6.1	Erstellen von Lesezeichen	mittel
6.2	Anzeigen von Lesezeichen	mittel
6.3	Löschen von Lesezeichen	mittel
7	Filter- und Sortermöglichkeiten	mittel
8	Favoritenfunktion für Hyperaudio-Dokumente	niedrig
8.1	Erstellen von Favoriten	niedrig
8.2	Anzeigen von Favoriten	niedrig
8.3	Löschen von Favoriten	niedrig
9	Übersicht über alle Hyperaudio-Dokumente der belegten Kurse	niedrig
10	Übersicht über die zuletzt abgespielten Hyperaudio-Dokumente	niedrig
11	Funktion zum Fortsetzen unterbrochener Wiedergaben bei folgenden Aufrufen in Moodle	niedrig
12	Unterstützung von mobilen Endgeräten	mittel

passende API, welche über ein Objekt angesprochen werden kann (Wild, 2017). So kann beispielsweise unter Verwendung der DML-API (Data Manipulation Language) und dem dazugehörigen Objekt \$DB Zugriffe auf die Datenbank vorgenommen werden.

Zu Beginn der Entwicklung eines Moodle-Plugins steht jedoch die Frage, um welche Art von Plugin es sich handelt. Es werden nun diejenigen Plugin-Typen beschrieben, welche für das Hyperaudio-Plugin infrage kommen (Moodle, 2017b).

### 3 Analyse

#### Media Player

Mit diesem Plugin-Typ kann Moodle um alternative Player für Audio- und Videoformate, aber auch für andere Medien (z.B. Diagramme, Formeln, etc.) ergänzt werden (Moodle, 2017a). Player beziehen sich aber stets auf das reine Abspielen von Dateien oder Links zu externen Medieninhalten, wie zum Beispiel ein Link zu einem *Youtube*-Video.

#### Blöcke

Blöcke dienen dazu, Kursseiten um zusätzliche Informationen anzureichern, welche dann in der rechten oder linken Spalte angeheftet werden können (vgl. Abbildung 2.2). Blöcke können auch „angeheftet“ werden (Moodle, 2018b). Diese Anheftung kann auf verschiedene Bereiche erfolgen, beispielsweise in der gesamten Moodle-Umgebung, auf der Seite des Benutzerprofils oder der Startseite (Moodle, 2015b). Blöcke sind also nicht dazu geeignet, größere Inhalte darzustellen. Dementsprechend kommt dieser Plugin-Typ nicht für das Hyperaudio-Plugin in Frage. Es wäre aber durchaus denkbar, dass mithilfe von Blöcken die Anforderungen 8.2, 9 und 10 umgesetzt werden könnten.

#### Ressourcen

Mittels eines Plugins dieses Typs ist es möglich, dem Studierenden Inhalte zu präsentieren. Dieses Plugin erwartete jedoch keinerlei Eingabe oder Interaktion von Seiten des Studierenden und dient ausschließlich der Darstellung von Informationen (Wild, 2017).

#### Aktivitäten

Neben der Darstellung von Inhalten erwarten Aktivitäten im Gegensatz zu Ressourcen eine Art von Interaktion durch den Studierenden. Dies kann beispielsweise in Form eines Quiz, bei dem der Nutzer Antworten anhaken muss, oder in Form eines Forums, in dem der Studierende Beiträge schreibt, geschehen (Wild, 2017). Aufgrund des interaktiven Charakters stellen Aktivitäten den richtigen Plugin-Typ für das Hyperaudio-Plugin dar.

### 3.4 Aktueller Stand der Technik

Bevor mit der Konzeption und Implementierung des Hyperaudio-Plugins begonnen werden kann, ist der aktuelle Stand der Technik bezüglich der Zielsetzung dieser Arbeit zu betrachten. Hierbei werden im ersten Schritt bereits etablierte Plattformen für die Bereitstellung von Audio- und Videoinhalten begutachtet. Im zweiten Schritt werden dann vorhandene Technologien für die Umsetzung innerhalb von Moodle untersucht.

#### 3.4.1 Etablierte Audio- und Video-Plattformen

Mit dem Hintergrund, eine nach DIN EN ISO 9241 erwartungskonforme Software gestalten zu wollen, erfolgt nun zunächst eine Analyse von etablierten Systemen zur Wiedergabe von Audio- und Videoinhalten mit integrierten Kommunikationsmöglichkeiten. Da menschliches Handeln stark durch erlernte Verhaltensmuster geprägt ist, empfiehlt es sich, bei der Gestaltung von Interaktionen auf bekannte Verfahren zurückzugreifen, um den kognitiven Aufwand zum Erlernen der Bedienmöglichkeiten gering zu halten und somit eine Konzentration auf die wesentlichen Inhalte zu ermöglichen (Rampl, Hansjörg, 2007).

## SoundCloud

„Als weltweit größte Musik- und Audio-Plattform“ (SoundCloud, o.D.) bietet *SoundCloud* Künstlern eine Plattform, um ihre Musik einem breiten Publikum anzubieten. Charakteristisch für *SoundCloud* ist das Design des Players (siehe Abbildung 3.2). Zum einen wird hier die Waveform des Musikstückes angezeigt und zum anderen werden gleichzeitig mittels Thumbnails Kommentare an ebenjener Stelle des Stücks visualisiert, zu der kommentiert wurde. Beim Abspielen des Musikstücks werden die annotierten Kommentare zum jeweiligen Zeitpunkt eingeblendet. Zusätzlich bietet der Player auch durch Mouseover-Effekte auf den Thumbnails die Möglichkeit, die annotierten Kommentare zu lesen. Durch einen Klick auf das entsprechende Thumbnail kann direkt auf den Kommentar geantwortet werden. Unterhalb des Players befindet sich der Eingabebereich, um eigene Kommentare zu verfassen. Diese werden zu dem Zeitpunkt gespeichert, zu dem der Kommentar begonnen wurde. Wiederum unterhalb des Eingabebereichs befindet sich ein Bereich für die Anzeige der Kommentare. Sie werden chronologisch nach Erstellungsdatum, mit dem neusten Kommentar an oberster Stelle, dargestellt. Antworten auf Kommentare werden durch eine leichte Einrückung gekennzeichnet.



Abbildung 3.2: Player der Musik- und Audio-Plattform *SoundCloud* (SoundCloud, 2015)

## Youtube

Im Bereich der Videoplattformen gilt *Youtube* als die mit Abstand am weitesten verbreitete Videoplattform in Deutschland (Statista, 2016). Zum Abspielen der von den Nutzern hochgeladenen Videos setzt *Youtube* auf den HTML5-Player<sup>6</sup> (siehe Abbildung 3.3).



Abbildung 3.3: Player der Video-Plattform *Youtube* (Youtube, 2015)

In (Google, 2018a,b) werden Möglichkeiten aufgezeigt, Informationen an ein Video zu annotieren.

<sup>6</sup>HTML = HyperText Markup Language

### 3 Analyse

Die Informationen können Verweise auf andere Videos, Playlists und Kanäle, eine Abstimmung oder einen Link zu einer Webseite beinhalten. Einem Video können insgesamt maximal ein Abspann und bis zu fünf Infokarten mittels des integrierten Webeditors angeheftet werden. Abbildung 3.4 zeigt, wie solche Infokarten im Player dargestellt werden. Auf das Vorhandensein von Infokarten wird durch ein Infosymbol in der rechten oberen Ecke des Videos hingewiesen. Bei einem Klick auf das Symbol werden die Infokarten angezeigt. Darüber hinaus kann pro Infokarte ein beliebiger Zeitpunkt im Video festgelegt werden, zu dem ein zusätzlicher Hinweis zur Infokarte eingeblendet wird. Ein Abspann kann hingegen nur während der letzten fünf bis 20 Sekunden eines Videos angezeigt werden.



Abbildung 3.4: Anzeige der Infokarten (Youtube, 2015)

#### 3.4.2 Technologien für den Einsatz in Moodle

Im zweiten Schritt wird sich nun der Analyse bestehender Komponenten zugewendet, die als Basis für das Moodle-Plugin dienen könnten. Ziel ist es, festzustellen, ob bereits Technologien existieren, mit deren Hilfe die Idee des Hyperaudio-Plugins umgesetzt werden kann oder ob zumindest Teile davon - unter entsprechender Beachtung der Lizenzierung - sinnvoll wiederverwendet werden können. Im Zuge dessen wird so vorgegangen, dass die einzelnen vorhanden Technologien auf diesem Gebiet mit ihren Funktionen vorgestellt werden. Dabei wird deren Relevanz für die Umsetzung des Plugins begutachtet.

#### VideoJS Player

Bei dem *VideoJS Player*<sup>7</sup> handelt es sich um eine Open-Source-Bibliothek zum Abspielen von Videos und stellt damit einen HTML5-Video-Player zur Verfügung. Der *VideoJS Player* ist bereits als Standard-Plugin für die Wiedergabe von Audio- und Video-Dateien in Moodle integriert. Wie der Name schon erkennen lässt, handelt es sich hierbei um eine JavaScript-Bibliothek. Der *VideoJS Player* beschränkt sich in seiner Ausgangsversion ausschließlich auf das Abspielen von Audio- und Video-Dateien und bietet einen optionalen Fallback auf den Adobe FlashPlayer. Die Funktionalität des *VideoJS Player* kann aber über Plugins erweitert werden. Es existieren bereits zahlreiche solcher Plugins. Hier sei vor allem das Plugin *videojs-wavesurfer*<sup>8</sup> genannt, welches das *wavesurfer.js*-Framework in den *VideoJS Player* integriert. Dank der Unterstützung von Plugins ist es durchaus denkbar, den Player mittels Plugin beispielsweise um Buttons zum Erstellen von Kommentaren oder persönlichen Notizen zu erweitern. Auch wäre es denkbar, mittels eines Plugins die annotierten Kom-

<sup>7</sup>GitHub-Projekt, Apache-Lizenz 2.0: <http://videojs.com/>; <https://github.com/videojs>

<sup>8</sup>GitHub-Projekt, MIT Lizenz: <https://github.com/collab-project/videojs-wavesurfer>

mentare zu visualisieren. Grundsätzlich stellt der *VideoJS Player* somit eine gute Ausgangslage für einen Hyperaudio-Player dar.

## H5P

Mit *H5P* und dem bereits vorhanden Plugin für Moodle<sup>9</sup> ist es möglich, verschiedene Arten von interaktiven Lerninhalten zu gestalten. Dabei handelt es sich um eine Sammlung von interaktiven Komponenten, darunter *Course Presentation*, *Timeline* und *Interactive Video*. *Course Presentation* bietet die Möglichkeit, interaktive Präsentationen zu gestalten. *Timeline* kann genutzt werden um Inhalte anhand eines Zeitstrahls darzustellen. *Interactive Video* ermöglicht, ähnlich wie *Course Presentation*, die Interaktion während des Abspielens eines Videos (siehe Abbildung 3.5). Besonders erwähnenswert ist, dass sich die interaktiven Inhalte bei *H5P* innerhalb der Weboberfläche erstellen lassen. Es wäre also denkbar, eine eigene interaktive Komponente zu entwickeln, welche es ermöglicht, Hyperaudio-Dokumente als interaktiven Lerninhalt zu erstellen und abzuspielen sowie eine Kommentarfunktion zu integrieren.

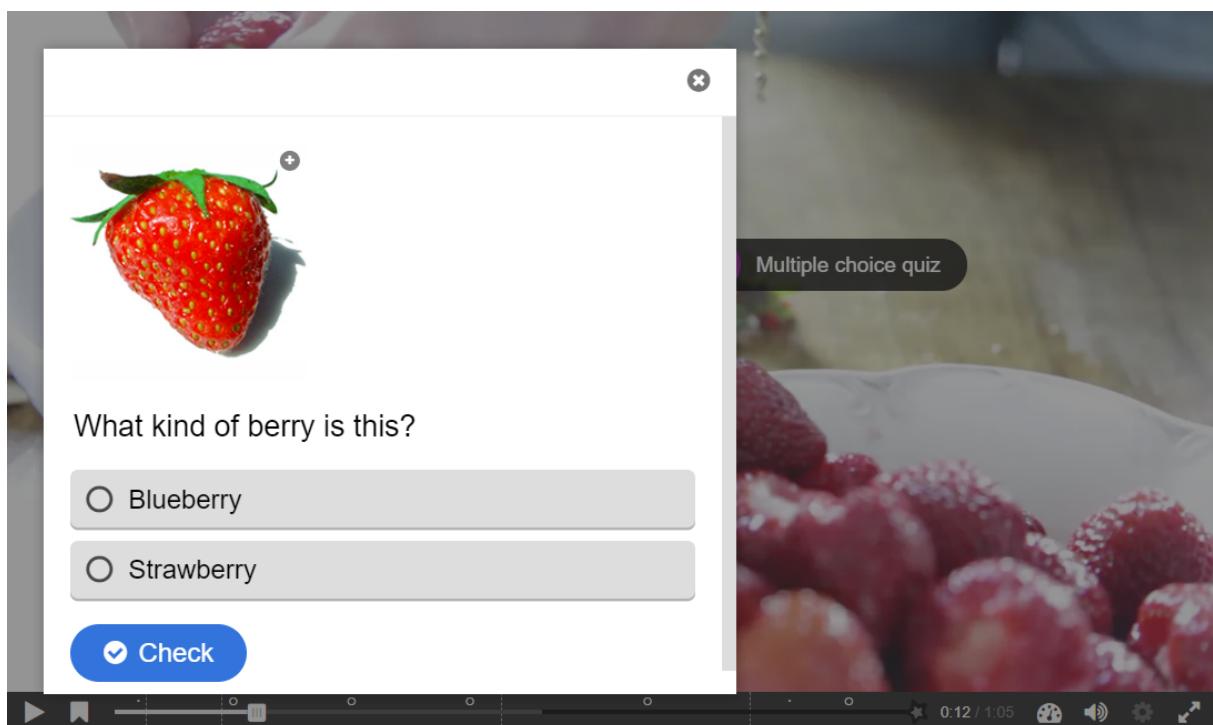


Abbildung 3.5: Auszug aus einem *Interactive Video* (H5P, 2013)

## Popcorn.js

Die Mozilla Corporation bietet mit *Popcorn.js*<sup>10</sup> eine Bibliothek an, welche neben einer standardisierten Steuerung von Medieninhalten aus verschiedenen Quellen auch die zeitabhängige Annotation von Inhalten mittels Plugins ermöglicht. Hier wäre also auch eine Entwicklung eines Plugins denkbar, mit welchem Hyperaudio-Dokumente wie gewünscht wiedergegeben werden können. Die Wartung für die Bibliothek wurde seitens Mozilla zwar eingestellt, das Projekt steht aber weiterhin auf GitHub zur Verfügung. Obwohl das Projekt nicht mehr weiterentwickelt wird, kann es durch die vorhandenen Steuerungsmöglichkeiten und das Plugin-System ein geeignetes Grundgerüst für die Entwicklung des Moodle-Plugins darstellen.

<sup>9</sup>GitHub-Projekt, GNU General Public License v2.0: <https://github.com/h5p/h5p-moodle-plugin>

<sup>10</sup>GitHub-Projekt, MIT Lizenz: <https://github.com.mozilla/popcorn-js>

### 3 Analyse

#### wavesurfer.js

Bei *wavesurfer.js*<sup>11</sup> handelt es sich um ein JavaScript-Framework, welches es ermöglicht, die Wellenform zu der abgespielten Audio-Datei visualisieren zu lassen (siehe Abbildung 3.6). Diese Basisfunktionalität wurde durch Weiterentwicklungen um nützliche Funktionen erweitert. Auf zwei dieser Weiterentwicklungen wird im Folgenden eingegangen.

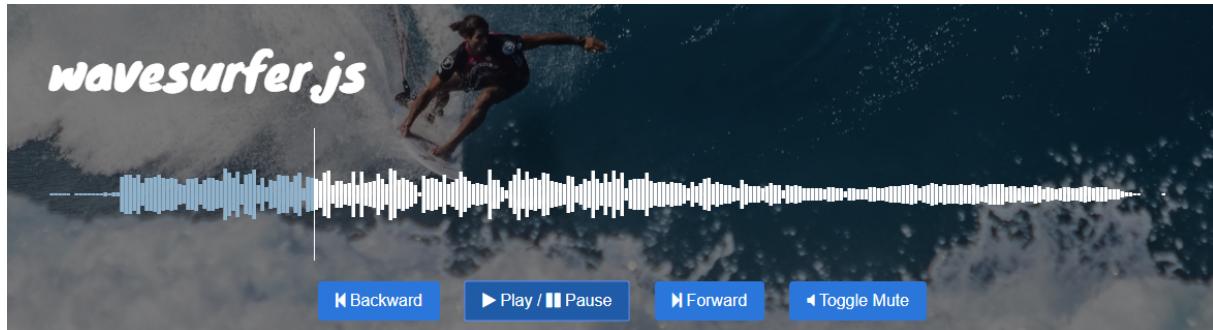


Abbildung 3.6: Der Audio-Player auf der *wavesurfer.js*-Webseite(H5P, o.D.)

Der *audio-annotator*<sup>12</sup> stellt eine auf dem *wavesurfer.js*-Framework basierende Weiterentwicklung dar, welche es mittels Weboberfläche ermöglicht, Annotationen in Form von Text an eine Audio-Datei anzuheften. Es erweitert *wavesurfer.js* also um die Möglichkeit, Annotationen an eine Datei anzuheften und bietet gleichzeitig noch eine Oberfläche, um ebendiese Annotationen vorzunehmen. Beim *BAT - BMAT Annotation Tool*<sup>13</sup> handelt es sich um eine Entwicklung basierend auf dem Framework *wavesurfer.js*. Es ermöglicht, ebenso wie *audio-annotator*, dem Benutzer mittels Weboberfläche Annotationen an einer Audio-Datei vorzunehmen. Somit bietet *BAT* logischerweise dieselben Vorzüge wie bereits der *audio-annotator*. Im Vergleich zum *audio-annotator* stellt *BAT* jedoch ein weiterentwickelteres Framework dar.

Das *wavesurfer.js* Framework - speziell mit seinen Weiterentwicklungen - bietet einige Funktionen, die für das Abspielen von Hyperaudio-Dokumenten nützlich sein könnten. Zusätzlich bietet es auch die Funktion, die entsprechenden Annotationen in einer Weboberfläche an die Audio-Dateien anzuheften. Grundsätzlich lässt sich feststellen, dass *wavesurfer.js* und seine Ableger im Vergleich zu den zuvor betrachteten Entwicklungen einen wesentlich unausgereifteren Eindruck hinterlassen.

#### timesheets.js

*timesheets.js*<sup>14</sup> ist ebenfalls ein JavaScript-Framework, welches analog zu *audio-annotator* und *BAT* die Annotation zusätzlicher Inhalte ermöglicht. Leider befindet sich das Framework aktuell nicht mehr in der Entwicklung. Aufgrund der Ähnlichkeit zu den *wavesurfer.js*-Ablegern und der eingestellten Entwicklung können hier zwar Ideen übernommen werden, als Basis für das zu entwickelnde Moodle-Plugin ist dieses Framework jedoch nicht geeignet.

Zusammenfassend ist festzustellen, dass für die Entwicklung des Plugins für Hyperaudio-Dokumente vor allem *VideoJS Player*, *H5P* und *Popcorn.js* die vielversprechendsten bestehenden Entwicklungen darstellen, da diese bereits einen hohen Entwicklungsstand haben. Unter Anbetracht der benötigten Funktionen stellen aber speziell der *VideoJS Player* und *Popcorn.js* eine gute Basis

<sup>11</sup>GitHub-Projekt, BSD-3-Clause: <https://wavesurfer-js.org>; <https://github.com/katspaugh/wavesurfer.js>

<sup>12</sup>GitHub-Projekt, BSD-2-Clause: <https://github.com/CrowdCurio/audio-annotator>

<sup>13</sup>GitHub-Projekt, GNU General Public License 3: <https://wavesurfer-js.org>; <https://github.com/BlaiMelendezCatalan/BAT>

<sup>14</sup>ehemaliges GitHub-Projekt, MIT Lizenz: <http://wam.inrialpes.fr/timesheets>

dar, da diese mit ihrem Kernelement als Player sowie durch die integrierten Plugin-Systeme für die Entwicklung von Multimedia-Elementen ausgelegt sind. Bei *H5P* müsste die Playerfunktion mit der dazugehörigen Erweiterung für Hyperaudio-Dokumente von Grund auf entwickelt werden, um eine entsprechende interaktive Komponente für Hyperaudio-Dokumente bereitzustellen zu können. Letztlich scheint *Popcorn.js* die beste Grundlage für die Entwicklung des Plugins darzustellen, da hier auch die Steuerung der Medieninhalte bereits von Grund auf ausgeprägt implementiert sind, woraus bei der Umsetzung einiger Funktionen großer Nutzen gezogen werden kann. Des Weiteren ist *Popcorn.js* als JavaScript-Bibliothek mühelos als Thirdparty-Framework in Moodle integrierbar.

## 3.5 Zusammenfassung

Im ersten Schritt der Analyse wurden Personas als mögliche Nutzer der Anwendung entwickelt. Anhand der Personas konnte die Aufteilung der Zielgruppe in Nutzende und Administrierende abgeleitet werden. Personas und Rollen wurden daraufhin zurate gezogen, um User Stories zu formulieren. Die daraus resultierenden rollenspezifischen Anforderungen wurden definiert und im Hinblick auf die Zielsetzung der Arbeit in drei Prioritätsstufen kategorisiert, welche die Basis für das Vorgehen im Implementierungsprozess bilden. Als Vorbereitung für die Implementierung dient ebenfalls die genauere Betrachtung der Möglichkeiten zur Plugin-Entwicklung in Moodle. Als Rahmenbedingung kann festgehalten werden, dass das Hyperaudio-Plugin primär als Aktivitäten-Plugin umzusetzen ist und dass Übersichten und Favoritenfunktion innerhalb von Blöcken realisiert werden können. Bei der Betrachtung des aktuellen Stands der Technik im Allgemeinen sowie im Bezug auf die Entwicklung des Plugins wurde entschieden, dass *Popcorn.js* als Grundlage für das Hyperaudio-Plugin dienen soll, da dieses mit seinen vorhanden Steuerungs- und Annotationsmöglichkeiten ein gutes Grundgerüst darstellt und problemlos in Moodle integriert werden kann.

# 4 Konzept

Mit den Erkenntnissen des vorherigen Kapitels kann sich nun der Konzeption des Hyperaudio-Plugins zugewendet werden. Dabei werden zu Beginn die Komponenten von Hyperaudio-Dokument und Annotationen analysiert und deren Zusammenhänge festgehalten. Diese Zusammenhänge können durch eine Schnittstellendatei abgebildet werden, deren Format in Abschnitt 4.2 definiert wird. Darauffolgend wird der Datenbankentwurf vorgenommen. Im Anschluss kann sich der Gestaltung der Benutzeroberfläche des Plugins gewidmet werden.

## 4.1 Zusammenhänge der Komponenten der Hyperaudio-Anwendung

Basierend auf der Definition eines Hyperaudio-Dokuments aus Abschnitt 2.3.3 und der in Abschnitt 3.2 erarbeiteten Anforderungen werden die Zusammenhänge der medialen Komponenten weiter analysiert. Hierbei soll vor allem geklärt werden, wie die einzelnen Komponenten von Hyperaudio-Dokument und Annotationen zusammenhängen und welche Möglichkeiten dadurch gegeben beziehungsweise nicht gegeben sind.

### 4.1.1 Komponenten

Im Mittelpunkt eines Hyperaudio-Dokuments steht eine Audio-Datei. Inhaltlich kann es sich hierbei beispielsweise um einen Vorlesungsvortrag handeln. Man könnte sich auch vorstellen, dass ein Hyperaudio-Dokument aus mehreren aneinander gereihten Audio-Dateien besteht. Dies würde an der grundsätzlichen Problemstellung jedoch nichts ändern und kann im Nachhinein jederzeit als Erweiterung umgesetzt werden. Aus diesem Grund wird in dieser Arbeit nur ein Plugin für ein Hyperaudio-Dokument bestehend aus einer Audio-Datei entwickelt.

Neben dieser zentralen Audio-Datei besteht das Hyperaudio-Dokument aus mehreren Zusatzinhalten, wobei es sich um Bilder, Graphen, Tabellen usw. handeln kann. Entscheidend ist aber, dass diese Zusatzinhalte immer nur eine rein grafische Darstellung verkörpern. Videos mit Ton sind somit beispielsweise nicht als Zusatzinhalt verwendbar, reine Animationen ohne Ton sind aber durchaus möglich.

Als besondere, nämlich externe Komponente, sind die Kommentare zu nennen. Diese gehören nicht zum eigentlichen Hyperaudio-Dokument, sollen aber mit diesem verknüpft werden. Es wird drei verschiedene Arten von Kommentaren geben, nämlich öffentliche Kommentare, persönliche Notizen und Lesezeichen. Innerhalb der öffentlichen Kommentare muss noch zwischen den Original-Kommentaren und den Antworten auf diese unterschieden werden.

### 4.1.2 Zusammenhänge

Die Zusammenhänge der soeben genannten Komponenten sind im UML-Diagramm in Abbildung 4.1 ersichtlich. Zunächst werden die Zusammenhänge zwischen der Audio-Datei und den Zusatzinhalten betrachtet. Nach dem Master-Slave-Prinzip werden Zusatzinhalte der Audio-Datei untergeordnet. Zu jedem beliebigen Zeitpunkt innerhalb der Abspieldauer der Audio-Datei kann maximal ein Zusatzinhalt gleichzeitig annotiert werden. Es sind also auch Phasen möglich, zu denen keinerlei Zusatzinhalt

dargestellt wird. Das Zeitfenster für die Annotation soll mittels einer Start- und Endzeit pro Zusatzinhalt definiert werden, wobei nur die Sekunden anzugeben sind. Bei dem Zeitfenster sollte natürlich bedacht werden, dass dieses nicht zu kurz sein sollte. Zwar soll, sobald ein Zusatzinhalt des Hyperaudio-Dokuments angezeigt wird, ein entsprechender *Audio Cue* abgespielt werden, dennoch können bereits einige Sekunden vergehen, bis der Studierende seinen Blick dem Zusatzinhalt zuwendet.

Auch die Kommentare stehen als externe Komponente in einer gewissen Art und Weise im Zusammenhang mit der Audio-Datei. Dies ergibt sich daraus, dass Kommentare zu einem bestimmten Zeitpunkt innerhalb der Audio-Datei erfasst werden. Während Antworten auf Original-Kommentare verfasst werden können, sind Antworten auf Antworten nicht möglich.

Zwischen Kommentaren und Zusatzinhalten gibt es jedoch keinen direkten Zusammenhang. Solche Zusammenhänge ergeben sich alleine aus den Zeitpunkten der Annotationen. Zusatzinhalte können wiederum in keinem Zusammenhang mit einem anderen Zusatzinhalt stehen.

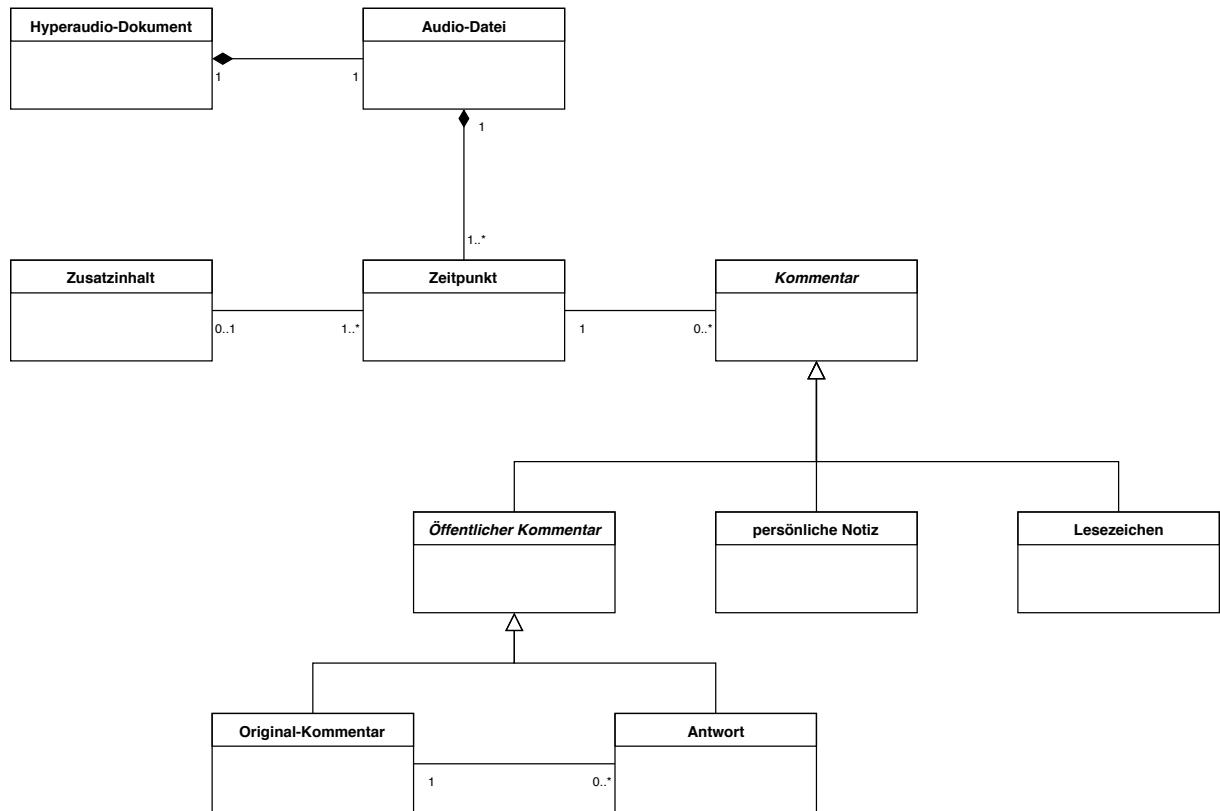


Abbildung 4.1: Zusammenhänge der Komponenten

## 4.2 Definition des Schnittstellenformats für Hyperaudio-Dokumente

Um ein Hyperaudio-Dokument zu erstellen, können mehrere Dateien hochgeladen werden. Verpflichtend ist das Bereitstellen einer Audio-Datei sowie einer Konfigurationsdatei. Darin wird festgehalten, zu welchem Zeitpunkt welcher Zusatzinhalt annotiert werden soll. Darüber hinaus ist es mittels der Konfigurationsdatei möglich Metadaten (Name und Beschreibung des Zusatzinhalts sowie die betroffene Kurseinheit und die zugehörigen Seiten) für die einzelnen Zusatzinhalte und das Hyperaudio-Dokument selbst anzufügen. Aufgrund des Einsatzes von PHP und JavaScript innerhalb der Moodle-Plugin-Entwicklung bietet sich der Einsatz von JSON (JavaScript Object Notation) an. JSON wird direkt durch JavaScript unterstützt und bietet im Vergleich zu XML (Extensible Markup Language) erhebliche Geschwindigkeitsvorteile (Nursetov et al., 2009). Mittels der JSON-Datei sollen die In-

## 4 Konzept

formationen zum Autor des Hyperaudio-Dokuments sowie folgende Informationen der Zusatzinhalte übertragen werden:

- Dateiname
- Name des Zusatzinhaltes
- Beschreibung des Zusatzinhaltes
- Kurseinheit
- Betroffene Seiten innerhalb der Kurseinheit
- Startzeitpunkt der Annotation
- Endzeitpunkt der Annotation

Entscheidend ist dabei vor allem der Dateiname. Anhand des Dateinamens kann anschließend die Zuordnung der weiteren Informationen zu der entsprechenden Audio-Datei in der Datenbank vorgenommen werden. Eine beispielhaft befüllte Konfigurationsdatei ist in Auflistung 4.1 dargestellt.

---

```
1 {
2   "author": "Dr. Niels Seidel",
3   "additional_contents": [
4     "additional_content": [
5       {"filename": "Abbildung_1_4.png",
6        "name": "Abbildung 1.4",
7        "course_unit": 1,
8        "page": "31",
9        "description": "Ein kooperativer Editor zur Visualisierung von gemeinsam zu
10       lernenden Vokabeln.",
11      "begin": "5",
12      "end": "10"}, {
13        "filename": "Abbildung_1_5.png",
14        "name": "Abbildung 1.5",
15        "course_unit": 1,
16        "page": "32",
17        "description": "Verschiedene Komponenten des Papierprototyps.",
18      "begin": "62",
19      "end": "220"}]
20   }
21 }
```

---

Auflistung 4.1: Beispielhafte Konfigurationsdatei

## 4.3 Datenbankentwurf

Um die dem Plugin zugrundeliegende Datenbank zu gestalten, wird auf die Erkenntnisse aus Abschnitt 4.1.2 zurückgegriffen. Das Ergebnis ist dem ER-Diagramm in Abbildung 4.2 zu entnehmen.

Jede der Datenbanktabellen verfügt über einen Primärschlüssel (*id*). Darüber hinaus wird zu jedem Eintrag gespeichert, wann dieser erstellt (*timecreated*) und zuletzt bearbeitet (*timemodified*) wurde. Für das Abspeichern von Dateien stellt Moodle die Tabelle *files* bereit (Moodle, 2018c). Dort kann die Datei abgelegt und an anderer Stelle darauf referenziert werden.

Im Mittelpunkt des Hyperaudio-Plugins steht die Tabelle *hyperaudio*. Diese repräsentiert das Hyperaudio-Dokument als Ganzes und die Audio-Datei aus Abbildung 4.1. Zunächst wird die ID des Moodle-Kurses (*course*) abgelegt, dem das Hyperaudio-Dokument zugeordnet ist. Während die Audio-Datei selbst in der bereits erwähnten Tabelle *files* zu finden ist, wird in der Tabelle *hyperaudio* deren Dateiname (*audiofile*) festgehalten. Zum Hyperaudio-Dokument können außerdem Name

und Ersteller in den Spalten *name* und *author* hinterlegt werden. Eine optionale Beschreibung kann entsprechend dem de-facto-Standard der Moodle-Plugin-Entwicklung über *introformat* und *intro* hinzugefügt werden (Moodle, 2016).

Bei der Tabelle *hyperaudio\_config* handelt es sich um eine Tabelle, in welcher die in Abschnitt 4.2 beschriebene Konfigurationsdatei gespeichert wird (*file* enthält den Dateinamen als Referenz auf die *files*-Tabelle). Daneben wird nur noch der Fremdschlüssel *hyperaudio\_id* auf die Tabelle *hyperaudio* als Zuordnung zum Hyperaudio-Dokument benötigt.

Zur Ablage der annotierten Zusatzinhalte dient die Tabelle *additional\_content*. Auch hier steht der Name der Datei in der Spalte *file* als Referenz auf die *files*-Tabelle. Ebenso dient der Fremdschlüssel *hyperaudio\_id* zur Verknüpfung des Zusatzinhalts mit der Audio-Datei. Des Weiteren werden folgende Metainformationen zum Zusatzinhalt abgespeichert:

- Name des Zusatzinhalts (*name*)
- optional: Beschreibung (*description*)
- optional: Kurseinheit (*course\_unit*)
- optional: Seitenangabe (*page*)
- Startzeitpunkt der Annotation innerhalb des Hyperaudio-Dokuments (*begin*)
- Endzeitpunkt der Annotation innerhalb des Hyperaudio-Dokuments (*end*)

Die Tabelle *hyperaudio\_comments* dient der Speicherung der vier in Abbildung 4.1 modellierten Kommentararten. Der Zusammenhang zum Hyperaudio-Dokument wird analog per Fremdschlüssel *hyperaudio\_id* hergestellt. Neben dem textuellen Kommentar (*commenttext*), werden auch der Verfasser in Form der *userid*, die Art des Kommentars (*comment\_type*)<sup>15</sup> und der Annotationszeitpunkt (*timeannotated*) gespeichert. Für den Fall, dass es sich um einen Antwortkommentar handelt, wird in der Spalte *comment\_id* die Referenz auf den Original-Kommentar festgehalten.

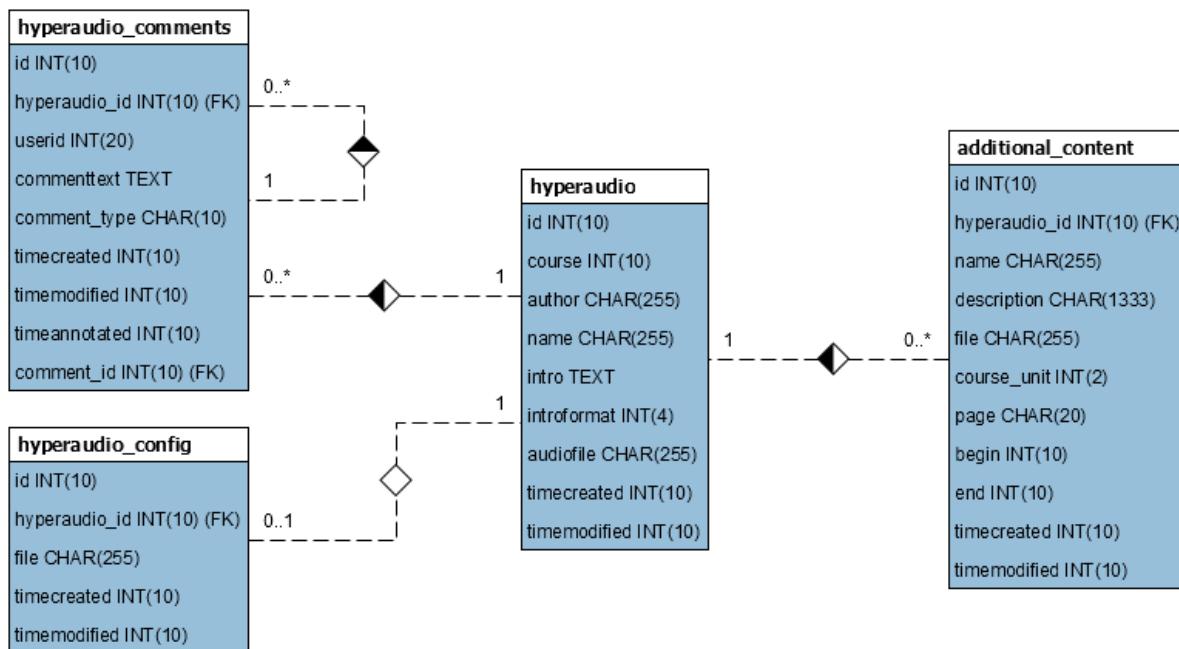


Abbildung 4.2: ER-Diagramm der Datenbank des Moodle-Plugins

<sup>15</sup>COMMENT, NOTE oder BOOKMARK

## 4.4 Gestaltung der Benutzeroberfläche

Damit den Lehrenden und Studierenden die im vorherigen Kapitel beschriebenen Nutzungsszenarien möglichst leicht fallen, wird sich nun der Gestaltung der Benutzeroberfläche zugewandt. „Das Design der Benutzeroberfläche stellt einen zentralen Aspekt für die Gebrauchstauglichkeit eines Softwareprodukts dar“ (Oppermann, 2002, S. 1). Ein entsprechend hoher Stellenwert soll der Benutzeroberfläche des Hyperaudio-Plugins zugeschrieben werden. Bei der Gestaltung der Benutzeroberfläche für die Darstellung der Hyperaudio-Aktivität wird die Ansicht in vier Bereiche aufgeteilt, welche in Abbildung 4.3 dargestellt sind.

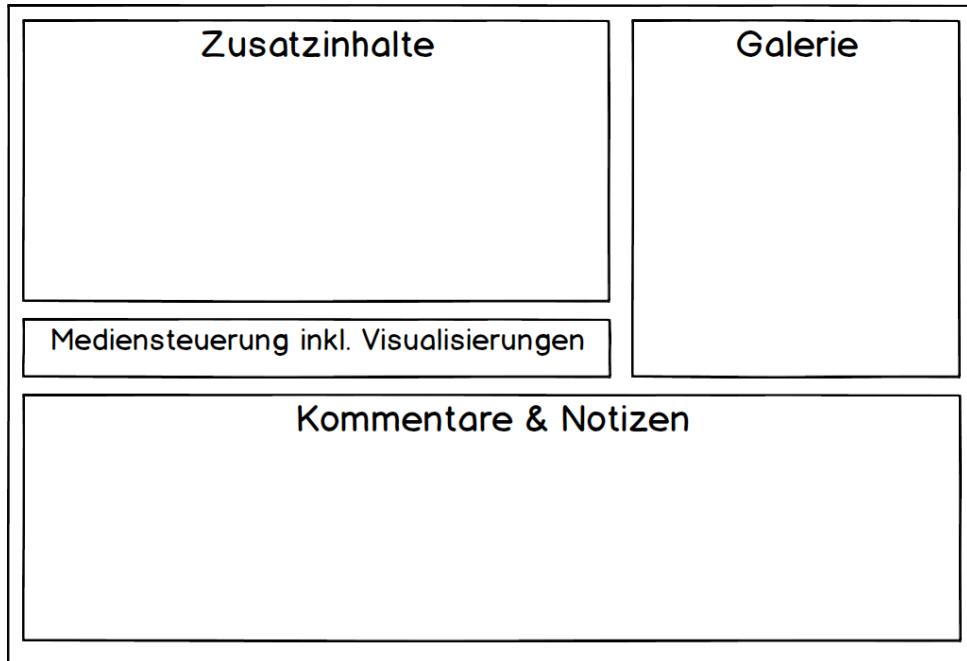


Abbildung 4.3: Bereiche einer Hyperaudio-Aktivität

Die Hyperaudio-Aktivität lässt sich demzufolge grob in die Bereiche „Zusatzinhalte“, „Mediensteuerung inklusive Visualisierungen der Annotationen“, „Kommentare und Notizen“ und „Galerie“ aufteilen. Zunächst werden nun für jeden dieser Bereiche verschiedene Designs erarbeitet und dann eines für die Verwendung im Hyperaudio-Plugin festgelegt.

Generell erfolgen alle Entscheidungen bezüglich der Oberfläche auf Basis von Mockups. Diese wurden mithilfe des Programms *Balsamiq Mockups 3.5.15<sup>16</sup>* erstellt. Anhand der Skizzen können Vor- und Nachteile der verschiedenen Designansätze schnell erkannt und auf Grund dessen sachliche Entscheidungen getroffen werden.

### 4.4.1 Zusatzinhalte

Im Bereich der Zusatzinhalte beschränkt sich die Diskussion auf die Darstellung jener im vorhergehenden Bereich. So ist zu entscheiden, inwiefern die darzustellenden Zusatzinhalte auf die Größe des Bereichs skaliert werden sollen. Dabei ergeben sich folgende Möglichkeiten:

1. Beibehalten der originalen Größe
2. Anpassen auf die Größe des Bereichs ohne Berücksichtigung des Seitenverhältnisses
3. Anpassen auf die Größe des Bereichs unter Berücksichtigung des Seitenverhältnisses

<sup>16</sup><https://balsamiq.com/>

Die verschiedenen Möglichkeiten illustriert Abbildung 4.4. Die erste Variante ist unpraktisch, da dies bei Inhalten, welche größer sind als der Bereich, dazu führt, dass nur ein Ausschnitt des Inhaltes dargestellt wird. Bei Inhalten, welche kleiner sind als der Bereich, wird hingegen nicht der volle Platz ausgenutzt und der Inhalt wird unnötig klein dargestellt. Bei der Variante 2 wird zwar stets der gesamte Bereich ausgenutzt und der Zusatzinhalt wird vollständig dargestellt, doch kann dies dazu zu unschönen Verzerrungen des dargestellten Inhaltes führen. Im Vergleich dazu bietet die dritte Variante den Vorteil, dass diese den Bereich möglichst gut ausnutzt ohne unschöne Verzerrungen zu verursachen.

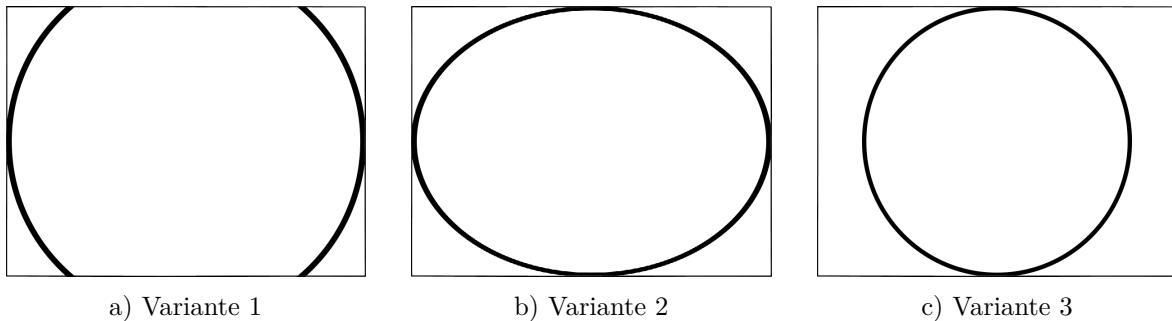


Abbildung 4.4: Verschiedene Möglichkeiten zur Darstellung von Zusatzinhalten

Neben der Entscheidung für Variante 3 muss auch die Entscheidung getroffen werden, wie sich der Bereich verhält, wenn gerade keine Zusatzinhalte dargestellt werden müssen. In diesem Fall sind verschiedene Szenarien denkbar. Zum einen wäre denkbar, dass der Bereich minimiert wird, sobald kein Zusatzinhalt dargestellt wird. Dies hätte aber den Nachteil, dass sich die Position der Kommentare mehrmals ändert und somit das Lesen erschwert wird. Zum anderen wäre auch die Darstellung eines Standardbildes oder einer einfarbigen Fläche vorstellbar. Es wäre aber auch möglich, innerhalb der Fläche die Metadaten des Hyperaudio-Dokuments darzustellen, sprich Titel, Autor und Beschreibung. Die letzte Variante stellt einen guten Kompromiss dar, da somit auch keine weitere Fläche für die Darstellung dieser Informationen benötigt wird.

Darüber hinaus muss auch über das Darstellungsformat der Zusatzinhalte entschieden werden. Hierbei bieten sich die Formate 16:9 und 4:3 für die Darstellung an. Nachdem vornehmlich vorhandene Abbildungen, Formeln und Bilder aus bestehenden Kurseinheiten dargestellt werden sollen, fällt die Wahl auf das 4:3-Format, welches mehr Ähnlichkeiten mit dem DIN A4-Format von Kurseinheiten aufweist als das 16:9-Format. Letzteres würde sich eher für die Darstellung von Videos eignen.

#### 4.4.2 Mediensteuerung inklusive Visualisierungen der Annotationen

Bei der Mediensteuerung für Hyperaudio-Dokumente sollen neben den üblichen Mediensteuerungselementen zusätzliche Informationen visualisiert werden. So sollen ähnlich wie bei *SoundCloud* auch die annotierten öffentlichen Kommentare, persönlichen Notizen und Lesezeichen dargestellt werden. Die Möglichkeit zum Erstellen von Lesezeichen soll ebenfalls in die Mediensteuerung integriert werden. Anhand dieser erweiterten Mediensteuerung kann sich der Nutzende schnell einen Überblick über die Annotationen verschaffen.

Der erste Entwurf der Mediensteuerung ist in Abbildung 4.5a dargestellt. Es ist zu erkennen, dass neben den üblichen Steuerungselementen die Visualisierung der Annotationen in der Timeline erfolgt. Hierzu wird für die drei verschiedenen Kommentar-Arten jeweils ein andersfarbiger Punkt (Kommentar: blau, Notiz: orange und Lesezeichen: grün) innerhalb der Timeline des Players dargestellt. Durch einen Klick auf einen Punkt, welcher einen Kommentar oder eine Notiz repräsentiert, soll dann zu der dazugehörigen Stelle innerhalb des Bereichs für Kommentare und Notizen gesprungen werden. Das Löschen und Anlegen von Lesezeichen soll durch eine Betätigung des Rechtsklicks auf den Punkt

## 4 Konzept

bzw. die Timeline möglich sein. Es wird ersichtlich, dass diese Variante den Nachteil hat, dass bei einer hohen Anzahl an Kommentaren und Notizen die Timeline unübersichtlich und schwer bedienbar wird.

In einem zweiten Entwurf der Mediensteuerung in Abbildung 4.5b wird die Visualisierung der Notizen und Kommentare in eine Box unterhalb der Timeline verschoben. Diese werden nun in einer Waveform dargestellt. Hierfür wird jedes Hyperaudio-Dokument in die gleiche fixe Anzahl an Zeitfenstern aufgeteilt. Diese Zeitfenster werden durch senkrecht orientierte Balken dargestellt, deren Höhe für die Anzahl der zu diesem Zeitfenster erfassten Kommentare und Notizen stehen soll. Die Balken werden anteilig nach Anzahl von Kommentaren und Notizen unterschiedlich eingefärbt. Dadurch wird nebenbei das Handling der Punkte in der Timeline vereinheitlicht, da es hier nur noch die Lesezeichen mit ihren Interaktionsmöglichkeiten gibt. Wenn mit der Maus über einen einzelnen Balken gefahren wird, soll jeweils eine Vorschau eines Kommentars mittels Tooltip dargestellt werden. Bei dem in der Vorschau angezeigten Kommentar handelt es sich um denjenigen Kommentar mit den meisten Antworten, also den meistdiskutierten Kommentar, innerhalb dieses Blockes. Wird hierbei kein eindeutiges Ergebnis gefunden, wird derjenige Kommentar mit dem frühesten Annotationszeitpunkt herangezogen. Wenn weiterhin kein eindeutiges Ergebnis erzielt ist, wird der neueste dieser Kommentare in der Vorschau angezeigt. Bei einem Klick auf den Block wird zu der dazugehörigen Stelle innerhalb des Bereichs für Kommentare und Notizen gesprungen. Diese Version bietet somit auch bei einer hohen Anzahl an Kommentaren und Notizen eine übersichtliche Darstellung.

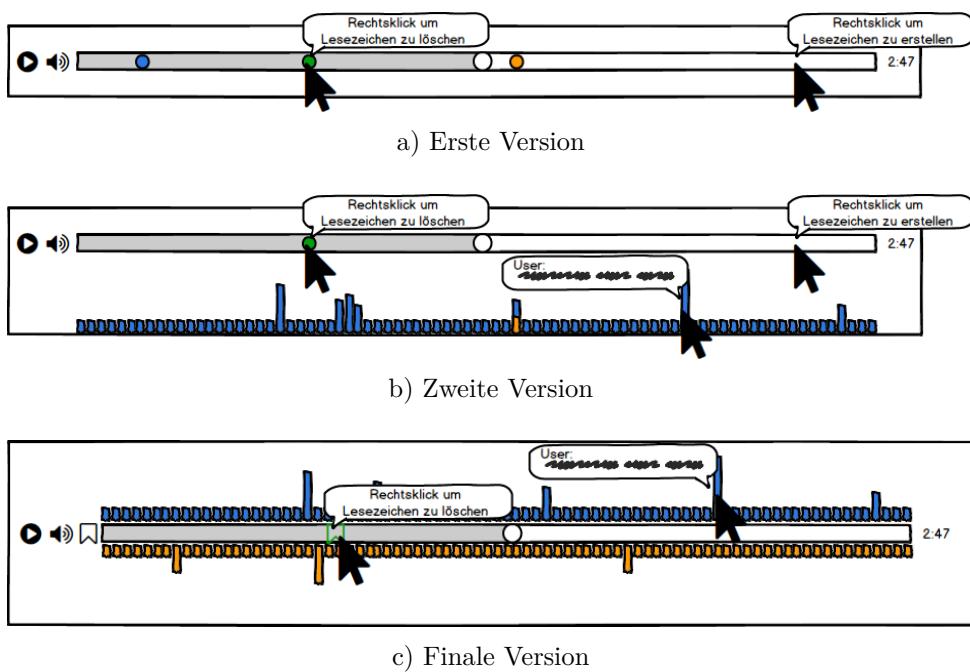


Abbildung 4.5: Benutzeroberfläche - Mediensteuerung

In der dritten und finalen Variante wird die Waveform an die Timeline gedockt und zweigeteilt (siehe Abbildung 4.5c). Der obere Teil repräsentiert die Kommentare, während der untere Teil die Notizen wiederspiegelt. Des Weiteren wurde ein Button zum Erstellen der Lesezeichen in der Mediensteuerung ergänzt. Somit ist nur noch zum Löschen der Lesezeichen die Verwendung des Rechtsklicks notwendig. Darüber hinaus wurde der Punkt für Lesezeichen durch ein dediziertes Symbol ersetzt. Diese Version bietet den Vorteil, dass die Interaktion, welche in der vorherigen Version ausschließlich Kommentaren vorbehalten war, ebenfalls für Notizen eingebettet werden kann. Demnach ist es auch möglich, in der Mediensteuerung mittels Mouseover eine Vorschau einer Notiz anzeigen zu lassen und mit Klick zu dieser zu springen. Durch die neue Art der Visualisierung von Kommentaren, Notizen und Lesezeichen

wird zudem die Barrierefreiheit der Anwendung verbessert, da die Kommentararten nicht nur anhand der Farbe zu unterscheiden sind, sondern auch anhand der Positionierung.

#### 4.4.3 Galerie

Die Galerie soll dazu dienen, einen Überblick über die vorhanden Zusatzinhalte zu bieten. Da die Zusatzinhalte alle einen grafischen Inhalt darstellen, kann jeder Zusatzinhalt durch ein kleines Vorschaubild repräsentiert werden. Hier wird in Fragen der Skalierung die gleiche Entscheidung getroffen wie bereits bei der Darstellung von Zusatzinhalten (vgl. Abschnitt 4.4.1). Eine weitere Grundfunktionalität einer Galerie ist die vergrößerte Anzeige der in der Vorschau dargestellten Inhalte, die auch in der Hyperaudio-Galerie zur Verfügung stehen soll. Um den Komfort zu erhöhen und den Hyperaudio-Gedanken zu stärken, ist eine Rückkopplung an die Mediensteuerung vorgesehen.

Die simpelste Umsetzung der Galerie ist eine Darstellung der Zusatzinhalte in einem einfachen Grid mit Scrollbalken, wie es in Abbildung 4.6a zu sehen ist. Zusätzlich wird das Grid um zwei Buttons für eine vergrößerte Ansicht des Zusatzinhalts sowie für die Rückkopplung zur Mediensteuerung ergänzt. Um eine dieser beiden Aktionen auszuführen, müsste also der gewünschte Zusatzinhalt markiert und der entsprechende Button betätigt werden.

Diese Variante hat den Vorteil, dass besonders viele Zusatzinhalte gleichzeitig angezeigt werden können. Auf der anderen Seite wird aber keinerlei Informationen zu den Zusatzinhalten geliefert. In der zweiten Variante, bei dem das Grid um einen Bereich für Details ergänzt wurde, kann der Nutzer zumindest die Details des ausgewählten Zusatzinhaltes einsehen. Diese in Abbildung 4.6b erkennbaren Details sind natürlich von den vorhandenen Metadaten abhängig. Nachteil ist in diesem Fall aber, dass durch den zusätzlichen Bereich für die Details bei gleicher Größe der Galerie weniger Zusatzinhalte zur selben Zeit dargestellt werden können. Das führt dazu, dass die Verwendung des Scrollbalkens häufiger notwendig wird.

Bei einer Darstellung der Zusatzinhalte als Kacheln, wie in Abbildung 4.6c zu sehen, können gleichzeitig für alle vorhandenen Zusatzinhalte Details angezeigt werden. Durch diese Art der Darstellung passen jedoch noch weniger Zusatzinhalte auf die gleiche Fläche.

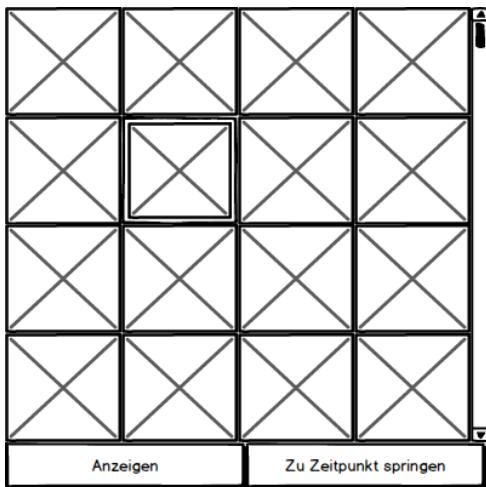
Eine besonders elegante Art der Darstellung wäre die des Cover Flows, bekannt aus verschiedenen Musikplayern, wie beispielsweise *iTunes*. In Abbildung 4.6d ist zu erkennen, dass auch hier ausschließlich Details des aktuell ausgewählten Zusatzinhaltes sichtbar sind. Des Weiteren hat diese Darstellungsweise den großen Nachteil, dass nicht auf einen Blick alle verfügbaren Zusatzinhalte ersichtlich sind, was das Durchsuchen der Zusatzinhalte ungemein erschwert. Somit ist diese Art der Darstellung zwar schön anzusehen, aber nicht sonderlich gebrauchstauglich im Zusammenhang mit Hyperaudio-Dokumenten.

Letztlich stellt sich die optimierte Variante der Kachel-Darstellung in Abbildung 4.6e als beste Lösung heraus. Die Optimierung besteht darin, dass die beiden Buttons obsolet gemacht werden. Dies kann zum einen dadurch erreicht werden, dass die vergrößerte Darstellung durch einen Klick auf das Vorschaubild ausgelöst wird. Zum anderen bietet der Bereich der Details noch ausreichend Platz, um hier die Funktion zur Rückkopplung an die Mediensteuerung einzufügen. Durch diese Verbesserungen wird nicht nur mehr Platz geschaffen, sondern auch die Benutzerfreundlichkeit erhöht, da der Vorgang zum Anzeigen der vergrößerten Ansicht beziehungsweise des Springens an den entsprechenden Zeitpunkt jeweils um einen Klick reduziert wurde.

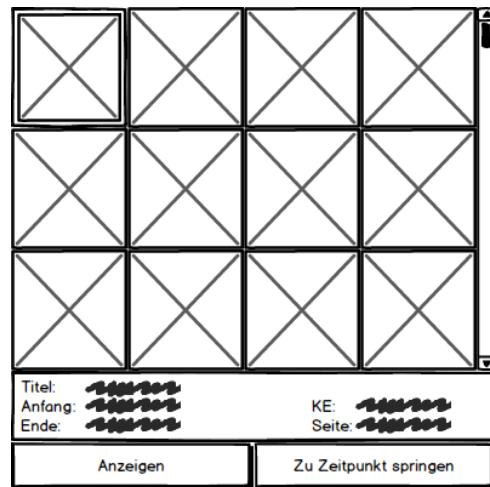
#### 4.4.4 Kommentare und Notizen

Der Kommentarbereich ist für das Erstellen und Anzeigen der öffentlichen Kommentare sowie der persönlichen Notizen zuständig. Zusätzlich müssen eine Suche, Filter- und Sortiermöglichkeiten auf Basis der Anforderungen aus Abschnitt 3.2.2 in die Oberfläche integriert werden.

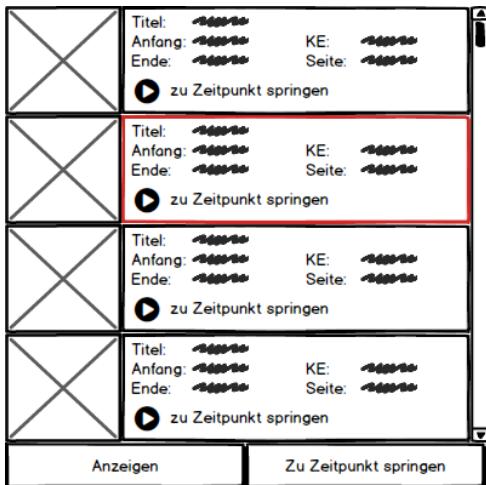
#### 4 Konzept



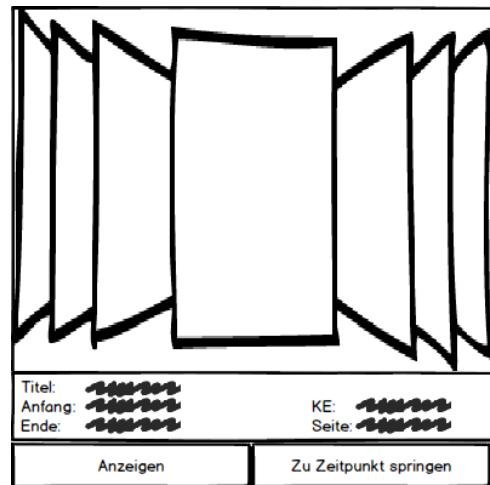
a) Galerie als einfaches Grid



b) Galerie als Grid mit Bereich für Details



c) Galerie mit Darstellung in Kachelform



d) Galerie als Cover Flow



e) Finale Version der Galerie

Abbildung 4.6: Benutzeroberfläche - Galerie

#### 4.4 Gestaltung der Benutzeroberfläche

Schreibe hier deinen Kommentar oder deine Notiz

Kommentare  Notizen Sortieren nach Zeitpunkt

Erstellt am 22.07.2018 - 16:13 Zeitpunkt: 00:15

Erstellt am 23.07.2018 - 18:32 Zeitpunkt: 00:15

Erstellt am 01.06.2018 - 09:34 Zeitpunkt: 01:27  
 Bearbeiten  Löschen

Erstellt am 24.08.2018 - 12:56 Zeitpunkt: 02:07

a) Erste Version

Schreibe hier deinen Kommentar oder deine Notiz

Kommentare  Notizen Sortieren nach Zeitpunkt

Erstellt am 22.07.2018 - 16:13 Zeitpunkt: 00:15

Erstellt am 23.07.2018 - 18:32 Zeitpunkt: 00:15

Erstellt am 01.06.2018 - 09:34 Zeitpunkt: 01:27  
 Bearbeiten  Löschen

Erstellt am 24.08.2018 - 12:56 Zeitpunkt: 02:07

b) Finale Version

Abbildung 4.7: Benutzeroberfläche - Bereich für Kommentare und Notizen

## 4 Konzept

Abbildung 4.7a zeigt eine erste Version des Kommentarbereichs. An oberste Stelle befinden sich eine Eingabemaske für Kommentare mit zwei Buttons, um den Inhalt der Eingabemaske als Kommentar oder Notiz abzuspeichern. Darunter befindet sich eine Suchmaske, neben dieser werden zwei Checkboxen dargestellt. Diese ermöglichen es dem Betrachter nach öffentlichen Kommentaren und persönlichen Notizen zu filtern. Im benachbarten Dropdown-Menü kann die Grundlage der Sortierung bestimmt werden. Die Sortierung kann nach Erstellungsdatum beziehungsweise nach Zeitpunkt der Annotation innerhalb des Hyperaudio-Dokuments erfolgen.

Unterhalb dieser Funktionen befindet sich die Anzeige der Kommentare und Notizen. Sowohl bei Kommentaren als auch bei Notizen wird neben dem Erstellungsdatum auch der Annotationszeitpunkt festgehalten. Dieser wird als Link umgesetzt, sodass analog zur Galerie bei einem Klick die Rückkopplung an die Mediensteuerung erfolgen kann.

Bei Kommentaren gibt es nach Betätigung der *Antworten*-Schaltfläche noch eine zusätzliche Eingabemaske zum Verfassen von Antworten. Damit die Kommunikation zwischen Studierenden und Lehrenden stets nachvollziehbar bleibt, werden weder eine Bearbeitungs- noch eine Löschfunktion für Kommentare und Antworten zur Verfügung gestellt.

Bei persönlichen Notizen hingegen wird das Bearbeiten und Löschen durch zwei zusätzliche Schaltflächen ermöglicht. Persönliche Notizen werden durch ein Schloss-Symbol hinter dem Erstellungsdatum gekennzeichnet.

Im nochmals verbesserten Design, welches in Abbildung 4.7b abgebildet ist, werden die Antworten auf Kommentare eingerückt dargestellt. Diese Darstellung führt zu einer verbesserten Übersicht und ist auch aus anderen modernen Anwendungen (wie zum Beispiel *SoundCloud* und *Youtube*, vgl. Abschnitt 3.4) bekannt.

### 4.4.5 Zusammenführen der Elemente

Im ersten Schritt werden die jeweils favorisierten Elemente in ein Layout zusammengeführt. Dabei wird sich zunächst an der groben Skizze aus Abbildung 4.3 orientiert. Wie nun in Abbildung 4.8a zu erkennen ist, wird der Anzeige der Zusatzinhalte und der Mediensteuerung eine große Fläche der Seite zugesprochen. Somit soll zugesichert werden, dass die Zusatzinhalte lesbar dargestellt werden. In dieser ersten Version ist der Bereich der Kommentare und Notizen so in die Breite gezogen, dass das Lesen der Inhalte auf großen Bildschirmen unangenehm werden kann. Aus diesem Grund wird in der finalen Version (siehe Abbildung 4.8b) die Breite dieses Bereichs auf die Breite der Zusatzinhalte und Mediensteuerung beschränkt. Dies hat zeitgleich zur Folge, dass der nun vorhandene freie Platz für die Galerie verwendet werden kann. Spätestens hiermit wird der Nachteil der gewählten Darstellungsweise der Galerie egalisiert, da nun ausreichend viele Zusatzinhalte ohne die Verwendung des Scrollbalkens eingesehen werden können.

### 4.4.6 Mobile Version

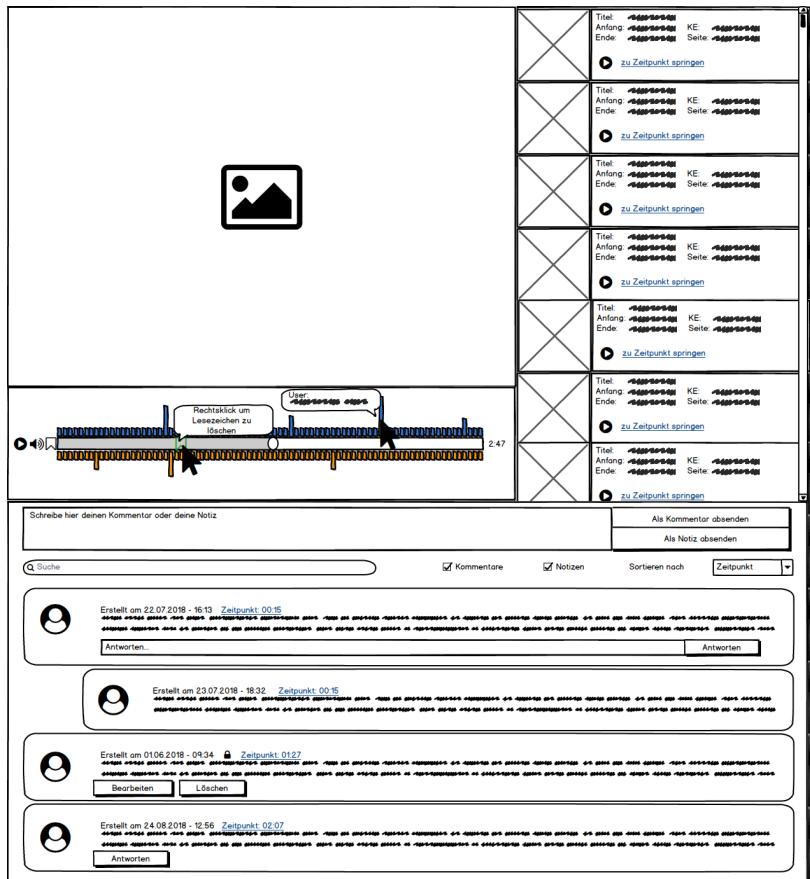
Für mobile Endgeräte wird keine neue Benutzeroberfläche entworfen. Stattdessen wird die vorhandene Oberfläche so angepasst (siehe Abbildung 4.9), dass diese für die Darstellung auf mobilen Geräten geeignet ist. Dazu wird die Strategie eines responsiven Grid-Layouts angewandt, das dafür sorgt, dass Inhalte bei kleineren Bildschirmgrößen anhand definierter Breakpoints umgebrochen werden (helllicht medien GmbH, o.D.).

So wird beispielsweise die Galerie auf kleinen Displays unterhalb des Bereichs für Kommentare und Notizen platziert. Dasselbe Vorgehen erstreckt sich auf weitere Bedienelemente, wie zum Beispiel die Filtermöglichkeiten. Zudem wird die Anzahl der Balken zur Visualisierung von Annotationen im Bereich der Mediensteuerung abhängig von der zur Verfügung stehenden Bildschirmgröße gewählt, sodass diese jeweils gut erkennbar bleiben und dabei möglichst viel Information liefern. Die Medien-

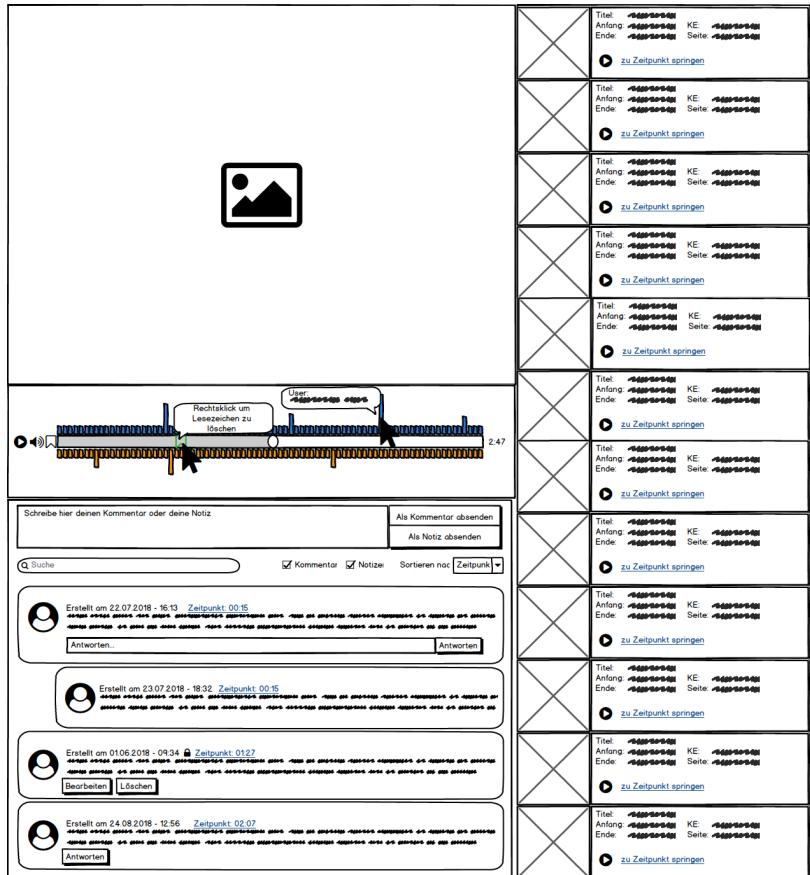
#### *4.4 Gestaltung der Benutzeroberfläche*

steuerung selbst sowie die Anzeige der Zusatzinhalte skaliert mit der Bildschirmbreite, jedoch nur bis zu einer definierten Maximalbreite.

## 4 Konzept



a) Erste Version



b) Finale Version

Abbildung 4.8: Benutzeroberfläche - Zusammengeführtes Layout

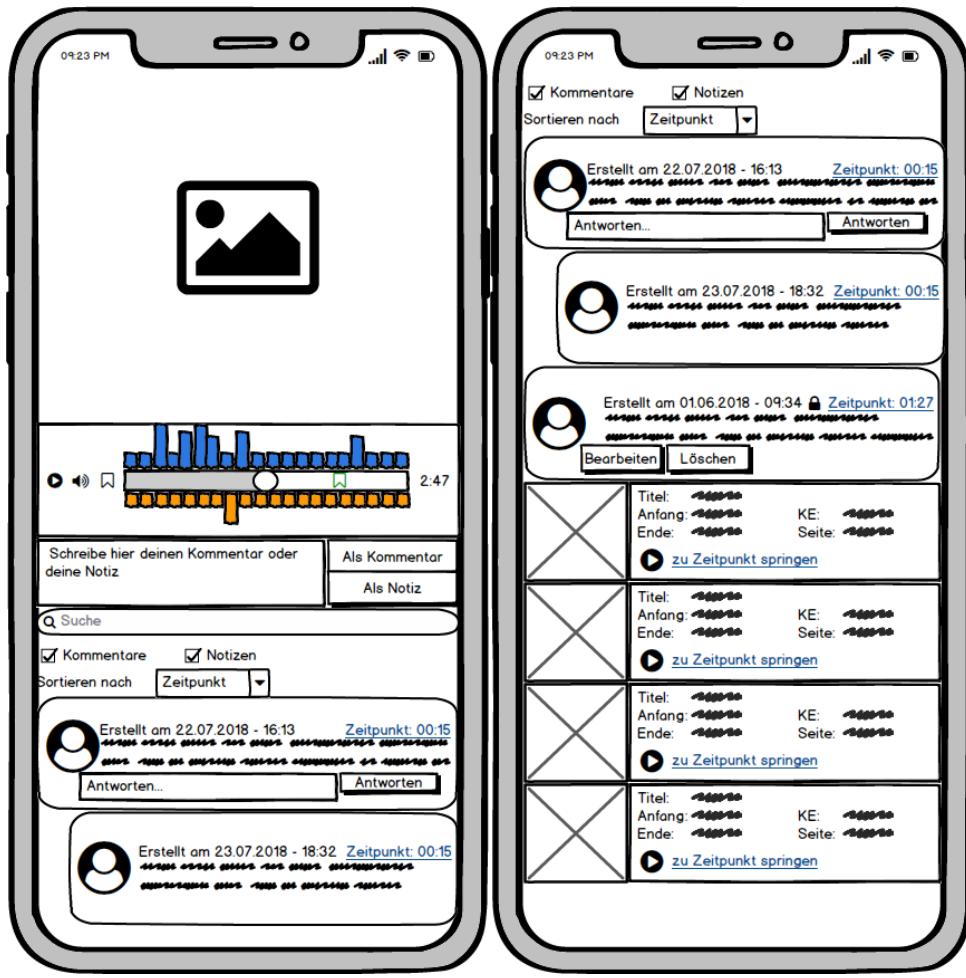


Abbildung 4.9: Benutzeroberfläche - Mobile Darstellung

## 4.5 Zusammenfassung

In diesem Kapitel wurden die grundlegenden konzeptuellen Entscheidungen getroffen. So wurden zu Beginn die Komponenten von Hyperaudio-Dokumenten im Sinne dieser Arbeit sowie deren Zusammenhänge bestimmt. Daraufhin wurde der Aufbau einer Konfigurationsdatei erarbeitet, welche diese Zusammenhänge abbilden kann. Auch der Datenbankentwurf wurde basierend auf diesen Ergebnissen vorgenommen. Nach dem Abschluss der technischen Konzeption wurde das Design des Plugins entworfen. Anhand verschiedener Mockups wurden Alternativen verglichen und ein Designkonzept beschlossen, das sowohl die Darstellung in Desktop-Umgebungen als auch auf mobilen Endgeräten umfasst.

# 5 Implementierung

Nachdem nun die konzeptionellen Vorarbeiten geleistet sind, wird in diesem Kapitel die Implementierung des Hyperaudio-Plugins erläutert. Zu Beginn wird zunächst die grundlegende Struktur des Moodle-Plugins festgehalten. Daraufhin werden das Vorgehen und die Ergebnisse der Implementierung beleuchtet.

## 5.1 Struktur des Hyperaudio-Plugins

Die Struktur des Hyperaudio-Plugins ergibt sich zum einen aus dem durch Moodle vorgegebenen Aufbau eines Plugins, zum anderen durch die gewählte Architektur des Plugins. Beide Bereiche werden in diesem Abschnitt beleuchtet.

### Aufbau eines Moodle-Plugins

Bei der Implementierung des Hyperaudio-Plugins ist die durch Moodle vorgegebene Aufbau von Plugins zu beachten (Moodle, 2016). Diese besteht stets aus vorgegebenen Dateien und Ordnern, wobei die jeweilige Anzahl von der Art des zu entwickelnden Plugins abhängig ist. Darüber hinaus bestimmt die Art des Plugins auch den zu wählenden Speicherort.

Bei Activity Plugins, wie dem Plugin für Hyperaudio-Dokumente, ist als Speicherort der Ordner **/mod** vorgeben. In diesem Ordner muss ein Unterordner mit dem Namen des Plugins angelegt werden, in diesem Fall **hyperaudio**, in welchem alle Plugin-Dateien abgelegt werden. Eine Übersicht über die Ordnerstruktur des Hyperaudio-Plugins findet sich in Abbildung 5.1.

Warum gibt es hier eine Datei ding.mp3?

Der Ordner **/hyperaudio/db** beherbergt die Dateien **access.php**, **install.xml** und **upgrade.php**. Die Datei **access.php** dient zur Steuerung der Berechtigungen innerhalb des Moodle-Plugins, wobei den verschiedenen Moodle-Rollen verschiedene Rechte für die einzelnen Funktionen zugewiesen werden können. Bei der Installation des Plugins wird die **install.xml** zur Erstellung der Datenbanktabellen für das Plugin verwendet. Es ist mindestens eine Tabelle mit dem Namen des Plugins anzulegen. Sollten die Datenbanktabellen nach Veröffentlichung des Plugins um Spalten erweitert werden, so kommt die Datei **upgrade.php** zum Einsatz. Hierin werden die notwendigen Schritte für einen Versionsabgleich definiert.

Im Ordner **/hyperaudio/lang** wird die Sprachlokalisierung vorgenommen. Für jede Sprache wird innerhalb des **lang**-Ordners ein eigener Unterordner angelegt. Darin befindet sich jeweils eine PHP-Datei, in welcher die Übersetzungen definiert werden. Der Name dieser Datei entspricht wiederum dem Namen des Plugins.

Das Icon, welches für das Plugin verwendet werden soll, muss im Ordner **/hyperaudio/pix** mit dem Dateinamen **icon.png** abgelegt werden und sollte eine Auflösung von 16x16 Pixel besitzen.

Im Ordner **/hyperaudio** liegen darüber hinaus die Dateien **lib.php**, **mod\_form.php**, **index.php**, **view.php** und **version.php**.

Die **lib.php** dient dazu, Standardfunktionen von Moodle zu überschreiben, wobei `<pluginname>_add_instance`, `<pluginname>_update_instance` und

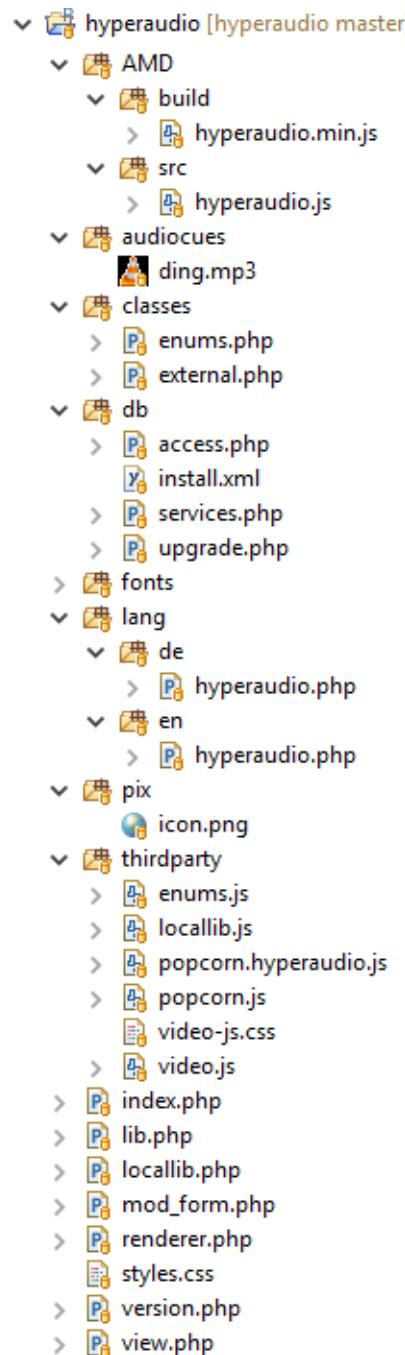


Abbildung 5.1: Ordnerstruktur des Hyperaudio-Plugins

<pluginname>\_delete\_instance als essenzielle Funktionen zu nennen sind. Mit diesen Funktionen wird das Anlegen, Aktualisieren und Löschen von Instanzen des Plugins ermöglicht.

Zum Anlegen und Aktualisieren wird in der **mod\_form.php** die dazugehörige Maske festgelegt.

Die **index.php** dient der Auflistung aller Instanzen eines Plugins innerhalb eines Kurses, worüber sich somit eine statistische Ansicht für den jeweiligen Kurs definieren lässt. Je nach Umsetzung kann der Inhalt dieser Auflistung unterschiedlich viele Informationen zu den Instanzen bereitstellen. Auch ist es beispielsweise anhand der Berechtigungen aus der **access.php** möglich, gewisse Informationen nur bestimmten Benutzern anzuzeigen.

Die erste Datei, die beim Öffnen der Aktivität geladen wird, ist die **view.php**, welche dementsprechend vornehmlich der Anzeige der Inhalte dient.

In der **version.php** wird die Version des Plugins gepflegt. Erhöht sich die Versionsnummer in der **version.php**, wird der automatische Upgradeprozess von Moodle für das Plugin ausgelöst.

Neben diesen vorgegebenen Dateien kommen üblicherweise noch weitere Dateien bei der Entwicklung eines Moodle-Plugins zum Einsatz (Wild, 2017). Dazu gehört beispielsweise die **locallib.php**, in welcher üblicherweise alle plugineigenen PHP-Funktionen deklariert werden. Auch ist es Usus, die eigentliche Darstellung der Plugin-Inhalte innerhalb eines Kurses von der **view.php** in eine **renderer.php** zu verlagern. Dort können verschiedene Renderer-Klassen, welche durch Moodle bereitgestellt werden, für die eigenen Bedürfnisse überschrieben werden. Anpassungen optischer Natur können durch CSS (Cascading Style Sheets) in der **styles.css** vorgenommen werden. Eigene JavaScript-Module, welche beispielsweise beim Laden der **view.php** automatisch aufgerufen werden, sind im Verzeichnis **/hyperaudio/AMD/src** (Asynchronous Module Definition) abzulegen. Für den produktiven Einsatz müssen JavaScript-Module noch minimiert werden. Moodle setzt hierbei auf *Grunt*<sup>17</sup> (Moodle, 2018d). Die minimierte Version der JavaScript-Module wird im Ordner **/hyperaudio/AMD/build** abgelegt.

### Architektur des Hyperaudio-Plugins

Basierend auf dem im vorherigen Kapitel beschriebenen Aufbau eines Moodle-Plugins kann nun die Architektur des Hyperaudio-Plugins dargelegt werden (siehe Abbildung 5.2). Das Plugin baut auf der um die für das Hyperaudio-Plugin benötigten Tabellen erweiterte Datenbank von Moodle auf. Die Renderer (**view.php**, **renderer.php**, **mod\_form.php** und **index.php**) dienen der Darstellung des User Interfaces und greifen hierfür auf die entsprechenden Daten aus der Datenbank zurück. Während die Renderer die statischen Inhalte des User Interfaces bereitstellen, dient das JavaScript-Modul (AMD) der Darstellung der dynamischen Inhalte. Hierfür bereitet das JavaScript-Modul ebenfalls Daten aus der Datenbank auf. Der Zugriff auf die Datenbank erfolgt in diesem Fall aber mittels Webservices. Neben dem Renderer und dem JavaScript-Modul wird das User Interface auch mittels CSS und Thirdparty JavaScript-Frameworks (wie zum Beispiel *Popcorn.js* und *VideoJS*) geformt. Abseits der Darstellung des User Interfaces dienen die Renderer sowie das JavaScript-Modul auch der Verarbeitung von Eingaben im User Interface. Die Core APIs von Moodle unterstützen dabei das Zusammenwirken der verschiedenen Komponenten. Sie ermöglichen beispielsweise die Darstellung von Inhalten sowie direkte Zugriffe auf die Datenbank.

lib aus Abbildung entfernen

## 5.2 Iterative Entwicklung

Die Entwicklung des Plugins wurde in iterativer Form durchgeführt. In jeder Iteration sollte das Plugin nur um einige wenige Funktionalitäten erweitert werden. Jede Iteration sollte mit einem lauffähigen Plugin abgeschlossen werden. So konnte direkt das Ergebnis betrachtet werden und gegebenenfalls in der nächsten Iteration nochmals angepasst werden (Stephan Augsten, 2018). Die Reihenfolge, in welcher die Funktionalitäten umgesetzt werden, leitete sich aus der Priorisierung der Anforderungen aus Abschnitt 3.2 ab. In den folgenden Abschnitten werden nun die verschiedenen Iterationsschritte vorgestellt. Dabei werden die Kernkonzepte mithilfe von Codeauszügen erläutert.

Ein Hinweis zum Stil: Da es sich bei dieser Textform nicht um ein Manual oder eine technische Dokumentation, sondern eher um einen Bericht der durchgeföhrten Arbeiten handelt, formuliert man den Text im Präteritum. Angesichts der Tatsache, dass die Iterationen bereits abgeschlossen sind und die Vorgehensweise (Issues, Iterationen, etc.) kaum auf andere Kontexte übertragbar und auch nicht wissenschaftlich relevant sind, muss der Text im Präteritum formuliert sein.

<sup>17</sup><https://gruntjs.com/>

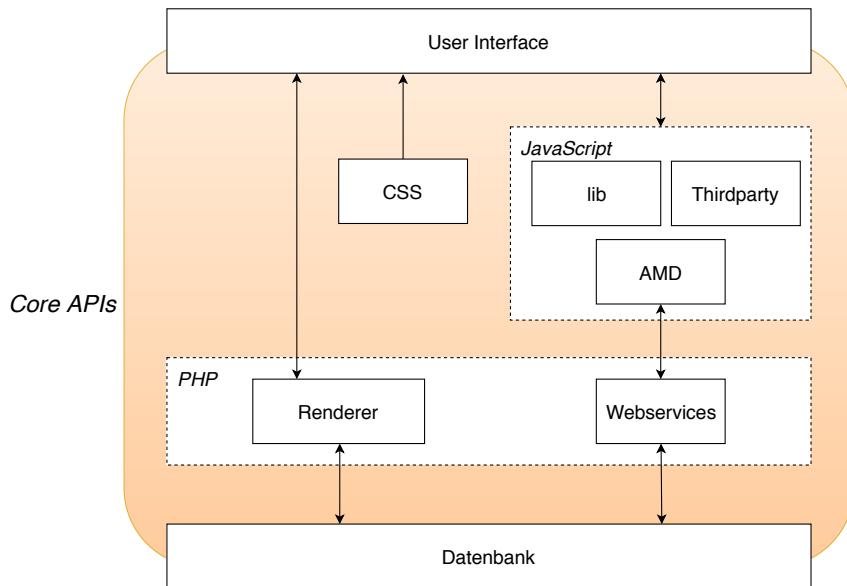


Abbildung 5.2: Architektur des Hyperaudio-Plugins

### 5.2.1 Speichern und Abspielen einer Audio-Datei

In der ersten Iteration wurde zunächst die grundlegende Struktur des Plugins erstellt (vgl. Abschnitt 5.1). Ziel der ersten Iteration sollte es sein, eine Audio-Datei speichern und wiedergeben zu können.

Dazu wurde in der Maske zum Anlegen und Aktualisieren von Instanzen des Hyperaudio-Plugins (**mod\_form.php**) neben dem obligatorischen Namens-Feld noch ein Element zum Hinzufügen einer Audio-Datei angelegt. Auflistung 5.1 zeigt einen Ausschnitt des Codes, der in der Funktion `definition` der Klasse `mod_hydraudio_mod_form`, die von der Klasse `moodleform_mod` erbt, ergänzt werden musste.

---

```

1 $mform = $this->_form;
2 $mform->addElement('text', 'name', get_string('hydraudio_mod_form_name',
3   'hydraudio'));
4 $mform->addHelpButton('name', 'hydraudio_mod_form_name', 'hydraudio');
5 $mform->setType('name', PARAM_TEXT);
6 $mform->addRule('name', get_string('error_wrong_hydraudio_name_input',
7   'hydraudio'), 'required');
8 $mform->addElement('filemanager', 'audiofile', get_string('hydraudiodata',
9   'hydraudio'), null,
10  array(
11    'subdirs' => 0,
12    'maxbytes' => 0,
13    'areamaxbytes' => 524288000,
14    'maxfiles' => 1,
15    'accepted_types' => array(
16      'audio'
17    )
18  ));
19 $mform->addHelpButton('audiofile', 'hydraudiodata', 'hydraudio');
20 $mform->addRule('audiofile', get_string('required', 'hydraudio'), 'required');
```

---

Auflistung 5.1: Ausschnitt der **mod\_form.php** in der 1. Iteration

Der `_form` der `moodleform_mod` konnten durch `addElement` neue Form-Elemente hinzugefügt werden. Mithilfe der Funktionen `setType`, `addHelpButton` und `addRule` konnten den Elementen Datentypen, Hilfe-Button und Regeln zugewiesen werden, die bei Auswertung der Form automatisch

## 5 Implementierung

validiert werden. Zum Hochladen von Dateien konnte der *filemanager* eingesetzt werden. Mithilfe eines Arrays konnten dabei Einschränkungen für Anzahl und Eigenschaften der hochzuladenden Dateien festgelegt werden. In diesem Fall darf maximal eine Datei hinzugefügt werden, die vom Typ *audio* sein muss. Die Funktion `get_string` dient im Allgemeinen der Darstellung der lokalisierten Bezeichnungen.

Um die Daten aus der Form in der Datenbank speichern und später wieder löschen zu können, musste auch die **lib.php** bearbeitet werden. Dazu dienen die bereits erwähnten Funktionen `<pluginname>_add_instance`, `<pluginname>_update_instance` und `<pluginname>_delete_instance`. Beispielhaft wird in Auflistung 5.2 die Funktion `hyperaudio_add_instance` zum Hinzufügen eines neuen Hyperaudio-Dokuments betrachtet.

---

```
1 function hyperaudio_add_instance($data) {
2     global $DB;
3
4     $cmid = $data->coursemodule;
5     $context = context_module::instance($cmid);
6
7     $draftitemid_audiofile = $data->audiofile;
8     unset($data->audiofile);
9
10    $now = time();
11    $data->timecreated = $now;
12    $data->timemodified = $now;
13
14    $data->id = $DB->insert_record('hyperaudio', $data);
15
16    hyperaudio_update_audiofile($data->id, $context, $draftitemid_audiofile);
17
18    return $data->id;
19 }
```

---

Auflistung 5.2: Ausschnitt der **lib.php** in der 1. Iteration

Der Parameter `$data` enthält bereits die in der Form eingegebenen Daten. Das Attribut `audiofile` enthält nicht die Audio-Datei selbst, sondern die ID der *draft file area* und sollte im ersten Schritt nicht in der Tabelle *hyperaudio* abgespeichert werden (vgl. Zeilen 7-8). Vor dem Speichern wird noch der aktuelle Zeitstempel hinterlegt (vgl. Zeilen 10-12). Mithilfe der Funktion `$DB->insert_record` kann das `$data`-Objekt mit seinen Attributen in der Tabelle *hyperaudio* abgelegt werden. Im Nachhinein sorgt die in der **locallib.php** definierte Funktion `hyperaudio_update_audiofile` dafür, dass die Audio-Datei in der *files*-Tabelle abgespeichert und in der *hyperaudio*-Tabelle korrekt referenziert wird (vgl. Auflistung 5.3).

---

```
1 function hyperaudio_update_audiofile($hyperaudiooid, $context, $draftitemid) {
2     global $DB;
3
4     file_save_draft_area_files($draftitemid, $context->id, 'mod_hyperaudio',
5         'audiofile', $hyperaudiooid);
6     $fs = get_file_storage();
7     $files = $fs->get_area_files($context->id, 'mod_hyperaudio', 'audiofile',
8         $hyperaudiooid, 'itemid, filepath, filename', false);
9
10    $file = reset($files);
11    $DB->set_field('hyperaudio', 'audiofile', $file->get_filename(), array(
12        'id' => $hyperaudiooid
13    ));
14 }
```

---

Auflistung 5.3: Ausschnitt der **locallib.php** in der 1. Iteration

Wie bereits in Abschnitt 5.1 angedeutet, übernimmt die **renderer.php** die Anzeige der Hyperaudio-Inhalte (siehe Auflistung 5.5). Die **view.php** dagegen reduziert sich auf wenige Zeilen (vgl. Auflistung 5.4). Der Plugin-Renderer wird hier benutzt, um Header, Hauptinhalte und Footer anzuzeigen.

---

```

1 $output = $PAGE->get_renderer('mod_hyperaudio');
2 echo $output->header();
3 echo $output->display($hyperaudio, $context);
4 echo $output->footer();

```

---

Auflistung 5.4: Ausschnitt der **view.php** in der 1. Iteration

In der Funktion `display` der Klasse `mod_hyperaudio_renderer`, die von der Klasse `plugin_renderer_base` erbt, werden die HTML-Inhalte erzeugt. Dabei handelt es sich in der ersten Iteration um einen Container, der ein `<audio>`-Element beinhaltet. Als Quelle wird im `<source>`-Element eine URL (Uniform Resource Locator) angegeben, die zuvor mit Moodle-Standardmitteln erzeugt wurde und auf die in der Datenbank abgelegte Audio-Datei verweist.

---

```

1 $audio_fileinfo = array(
2     'component' => 'mod_hyperaudio',
3     'filearea' => 'audiofile',
4     'itemid' => $hyperaudio->id,
5     'contextid' => $context->id,
6     'filepath' => '/',
7     'filename' => $hyperaudio->audiofile
8 );
9
10 $audiofileurl = moodle_url::make_pluginfile_url(
11     $audio_fileinfo['contextid'], $audio_fileinfo['component'],
12     $audio_fileinfo['filearea'], $audio_fileinfo['itemid'],
13     $audio_fileinfo['filepath'], $audio_fileinfo['filename']);
14 $audio_url = $audiofileurl->get_scheme() . '://'. $audiofileurl->get_host() .
15     $audiofileurl->get_path();
16 if ($audiofileurl->get_port()){
17     $audio_url .= ':' . $audiofileurl->get_port();
18 }
19 $output = '<div id="hyperaudio">'.
20     ' data-hyperaudio_id="'. $hyperaudio->id.'"'.
21     ' data-context_id="'. $context->id.'"'.
22     ' data-hyperaudio_titel="'. $hyperaudio->name.'"'.
23     ' data-hyperaudio_description="'. $hyperaudio->intro.'">'.
24 $output .= '<audio id="hyperaudio_audio" controls>'.
25     '<source src="'. $audio_url.'"/>'.
26     '</audio>';
27 $output .= '</div>';
28
29 echo $output;

```

---

Auflistung 5.5: Ausschnitt der **renderer.php** in der 1. Iteration

Auf die beschriebene Art und Weise lies sich ein Hyperaudio-Dokument, das vorläufig allein aus einer Audio-Datei bestand, speichern und mithilfe des HTML5-Audio-Players wiedergeben.

## 5.2.2 Speichern und Anzeige von Zusatzinhalten

Bei der zweiten Iteration wurde das Plugin um die Möglichkeit zum Speichern und zeitabhängigen Anzeigen der Zusatzinhalte erweitert. Für das Speichern wurde analog zu Abschnitt 5.2.1 vorgegangen. Es wurde ein *filemanager* (**mod\_form.php**) ergänzt, welcher das Hochladen von beliebig vielen Bilddateien erlaubt. Zusätzlich wurde die **locallib.php** um die Funktionen

## 5 Implementierung

hyperaudio\_update\_additional\_content und hyperaudio\_delete\_additional\_content erweitert. Letztere soll verhindern, dass beispielsweise beim Ergänzen von Zusatzinhalten die bereits vorhanden Zusatzinhalte doppelt abgespeichert werden. Dementsprechend wird diese Funktion, wie in Auflistung 5.6 zu sehen, zu Beginn der Funktion hyperaudio\_update\_additional\_content aufgerufen. Die Zeilen 22 bis 27 dienen dazu, die Metadaten des Zusatzinhaltes festzuhalten. In dieser Iteration wurden diese noch mit fixen Beispieldaten befüllt. Die Funktion hyperaudio\_update\_additional\_content wird nun von den Funktionen hyperaudio\_add\_instance und hyperaudio\_update\_instance aufgerufen.

---

```
1 function hyperaudio_update_additional_content
2   ($hyperaudiooid, $context, $draftitemid, $configfile) {
3     global $DB;
4
5     hyperaudio_delete_additional_content($hyperaudiooid, $context);
6
7     file_save_draft_area_files($draftitemid, $context->id,
8       'mod_hyperaudio', 'additional_content', $hyperaudiooid);
9     $fs = get_file_storage();
10
11    $files = $fs->get_area_files($context->id, 'mod_hyperaudio',
12      'additional_content', $hyperaudiooid, 'itemid, filepath, filename',
13      false);
14    $counter=1;
15    $begin=0;
16    $end=10;
17    foreach ($files as $file) {
18      $additional_content = new \stdClass();
19      $additional_content->file = $file->get_filename();
20      $additional_content->hyperaudio_id = $hyperaudiooid;
21
22      $additional_content->name = 'Name '.$counter;
23      $additional_content->course_unit = 'Kurseinheit '.$counter;
24      $additional_content->page = 'Seite '.$counter;
25      $additional_content->description = 'Beschreibung '.$counter;
26      $additional_content->begin = $begin;
27      $additional_content->end = $end;
28
29      $now = time();
30      $additional_content->timecreated = $now;
31      $additional_content->timemodified = $now;
32
33      $additional_content_id = $DB->insert_record('additional_content',
34        $additional_content);
35
36      $counter++;
37      $begin+=15;
38      $end+=15;
39    }
40 }
```

---

Auflistung 5.6: Ausschnitt der **locallib.php** in der 2. Iteration

Um die gespeicherten Zusatzinhalte zeitabhängig anzeigen zu können, kommt das JavaScript-Framework *Popcorn.js* zum Einsatz. Dazu wurden die JavaScript-Dateien von *Popcorn.js* und dessen *Images*-Plugin im Ordner **/hyperaudio/thirdparty** abgelegt und in die **view.php** eingebunden (siehe Auflistung 5.7).

---

```
1 $PAGE->requires->js('/mod/hyperaudio/thirdparty/popcorn.js', true);
2 $PAGE->requires->js('/mod/hyperaudio/thirdparty/popcorn.image.js', true);
```

---

---

Auflistung 5.7: Ausschnitt der **view.php** in der 2. Iteration

Im Renderer wurde ein weiteres `<div>`-Element ergänzt, in welchem die Zusatzinhalte angezeigt werden sollen. Diese werden aus der Datenbank geladen und in einer Schleife abgearbeitet. Dabei wird zunächst eine URL zur Bilddatei generiert. Über `popcorn.image` sorgt das *Popcorn.js Images*-Plugin dafür, dass diese URL zu den definierten Zeitpunkten abgerufen und das entsprechende Bild angezeigt wird (vgl. Zeilen 50-55).

---

```

1 $output = '<div id="hyperaudio">' .
2     ' data-hyperaudio_id="'. $hyperaudio->id.'"' .
3     ' data-context_id="'. $context->id.'"' .
4     ' data-hyperaudio_titel="'. $hyperaudio->name.'"' .
5     ' data-hyperaudio_description="'. $hyperaudio->intro.'">>' ;
6 $output .= '<div id="hyperaudio_additional_content"></div>';
7 $output .= '<audio id="hyperaudio_audio" controls>' .
8     '<source src="'. $audio_url.'" />' .
9     '</audio>';
10
11 $output .= '<script type="text/javascript">' .
12     'document.addEventListener("DOMContentLoaded", function() {' .
13         'var popcorn = Popcorn("#hyperaudio_audio");' ;
14
15 $files = hyperaudio_get_additional_content($hyperaudio->id, $context);
16 foreach ($files as $file) {
17     $additional_content = $DB->get_record('additional_content',
18         array(
19             'hyperaudio_id' => $hyperaudio->id,
20             'file' => $file->get_filename()
21         ), '*', IGNORE_MISSING);
22
23     if ($additional_content == false){
24         continue;
25     }
26
27     $additional_content_fileinfo = array(
28         'component' => 'mod_hyperaudio',
29         'filearea' => 'additional_content',
30         'itemid' => $hyperaudio->id,
31         'contextid' => $context->id,
32         'filepath' => '/',
33         'filename' => $additional_content->file
34     );
35
36     $additional_content_fileurl = moodle_url::make_pluginfile_url(
37         $additional_content_fileinfo['contextid'],
38         $additional_content_fileinfo['component'],
39         $additional_content_fileinfo['filearea'],
40         $additional_content_fileinfo['itemid'],
41         $additional_content_fileinfo['filepath'],
42         $additional_content_fileinfo['filename']);
43     $additional_content_url = $additional_content_fileurl->get_scheme() . '://'.
44     $additional_content_fileurl->get_host() .
45     $additional_content_fileurl->get_path();
46     if ($additional_content_fileurl->get_port()){
47         $additional_content_url .= ':' . $additional_content_fileurl->get_port();
48     }
49
50     $output .= 'popcorn.image({'.
51         ' start: '.$additional_content->begin.','.
52         ' end: '.$additional_content->end.','.

```

## 5 Implementierung

```
53      ' href: "javascript:void(0);",'.
54      ' src: "'.$additional_content_url.'",'.
55      ' target: "hyperaudio_additional_content""));';
56 }
57
58 $output .= '}, false);</script>';
59 $output .= '</div>';
60
61 echo $output;
```

---

Auflistung 5.8: Ausschnitt der **renderer.php** in der 2. Iteration

Mit diesen Erweiterungen wurde das Ziel der zweiten Iteration erreicht. Es war nun möglich, Zusatzinhalte abzuspeichern und diese unter Verwendung von *Popcorn.js* zeitabhängig zur Audio-Datei darstellen zu lassen.

### 5.2.3 Einbindung der Konfigurationsdatei

Damit die Beispieldaten durch tatsächliche Metadateninformationen ausgetauscht werden konnte, war in der dritten Iteration die Einbindung der Konfigurationsdatei notwendig. Zu Beginn dieser Iteration stand erneut die Erweiterung der **mod\_form.php** um einen weiteren *filemanager*, der maximal eine Konfigurationsdatei im JSON-Format entgegennimmt. Die Speicherung verläuft analog zur Speicherung der Zusatzinhalte. In diesem Zuge wurden also erneut die **lib.php** und **localib.php** entsprechend erweitert. Neben den Funktionen `hyperaudio_update_hyberaudio_config` und `hyperaudio_delete_hyberaudio_config` wurde, wie in Auflistung 5.9 zu erkennen, die Funktion `hyperaudio_update_additional_content` erweitert. Es ist zu entnehmen, dass zunächst die Inhalte der Konfigurationsdatei mithilfe der Funktion `hyperaudio_parse_hyberaudio_config` ausgelesen werden, um diese Metadaten dann beim Abspeichern der Zusatzinhalte zu verwenden.

```
1 hyperaudio_update_additional_content($hyberaudio_id, $context,
2     $draftitemid, $configfile){
3
4     //...
5
6     $additional_contents_data =
7         hyperaudio_parse_hyberaudio_config($hyberaudio_id, $context, $configfile)
8         ["additional_contents_by_filename"];
9
10    //...
11
12    foreach ($files as $file) {
13        //...
14
15        $additional_content_data =
16            $additional_contents_data[$additional_content->file];
17
18        $additional_content->name = $additional_content_data->name;
19        $additional_content->course_unit = $additional_content_data->course_unit;
20        $additional_content->page = $additional_content_data->page;
21        $additional_content->description = $additional_content_data->description;
22        $additional_content->begin = $additional_content_data->begin;
23        $additional_content->end = $additional_content_data->end;
24
25        //...
26    }
27 }
28
29 //...
```

```

31 function hyperaudio_parse_hyperaudio_config($hyperaudio_id, $context, $configfile) {
32
33     $fs = get_file_storage();
34     $file = $fs->get_file(
35         $context->id, 'mod_hyperaudio', 'hyperaudio_config',
36         $hyperaudio_id, '/', $configfile);
37     $jsontext = $file->get_content();
38     $json = json_decode ($jsontext);
39
40     $hyperaudio_author = $json->author;
41
42     $additional_contents = $json->additional_contents->additional_content;
43     $additional_contents_by_filename =
44         array_column($additional_contents, NULL, 'filename');
45
46     return array(
47         'additional_contents_by_filename' => $additional_contents_by_filename,
48         'hyperaudio_author' => $hyperaudio_author);
49 }

```

---

Auflistung 5.9: Ausschnitt der **locallib.php** in der 3. Iteration

Als Ergebnis dieser Iteration konnten nun Zusatzinhalte zu den Zeitpunkten dargestellt werden, welche in der Konfigurationsdatei definiert sind.

#### 5.2.4 Speichern und Anzeige von Kommentaren

Nachdem in den vorherigen Iterationen die Pflege und das Abspielen von Hyperaudio-Dokumenten im Fokus lag, konzentrierte sich diese Iteration auf die Kommentarfunktion. Zu diesem Zweck wurde im ersten Schritt der Renderer um entsprechende Elemente ergänzt, welche in Auflistung 5.10 zu sehen sind. Dabei handelt es sich zum einen um eine Textarea mit dazugehörigem Submit-Button, welche dem Erstellen von Kommentaren dienen. Das `<div>`-Element mit der ID *hyperaudio\_comments* soll der Anzeige der Kommentare dienen.

```

1 $output .=
2     '<div id="comment_creation_container" class="textarea_container">'.
3         '<textarea id="hyperaudio_comment" name="hyperaudio_comment"' .
4             ' class="hyperaudio_textarea" autocomplete="off"' .
5             ' placeholder=""'.
6                 get_string('mod_hyperaudio_renderer_comment', 'hyperaudio') .
7                 '....'">' .
8         '</textarea>' .
9         '<button type="button" class="comment_submit"' .
10            ' data-comment_type="'.CommentType::Comment.'">' .
11                get_string('mod_hyperaudio_renderer_submit_comment', 'hyperaudio') .
12            '</button>' .
13     '</div>';
14 $output .= '<div id="hyperaudio_comments"></div>';

```

---

Auflistung 5.10: Ausschnitt der **renderer.php** in der 4. Iteration

Das Speichern sowie das Anzeigen der Kommentare wurde unter Einsatz von Webservices, welche durch ein JavaScript-Modul ([/hyperaudio/AMD/src/hyperaudio.js](#)) angesprochen werden, realisiert. Hierfür wurden zusätzlich die Dateien **services.php** ([/hyperaudio/db](#)) und **external.php** ([/hyperaudio/classes](#)) für die Bereitstellung der Webservices benötigt.

In der **services.php** wurden zunächst die nötigen Webservices deklariert (siehe Auflistung 5.11). Hier wird lediglich auf die **external.php** verwiesen, die den eigentlichen Code der Webservices beinhalten.

## 5 Implementierung

---

```
1 $functions = array(
2     'mod_hyperaudio_save_comment' => array(
3         'classname'      => 'mod_hyperaudio_external',
4         'methodname'    => 'save_comment',
5         'classpath'     => 'mod/hyperaudio/classes/external.php',
6         'description'   => 'Save comment',
7         'type'          => 'write'
8     ),
9     'mod_hyperaudio_load_comments' => array(
10        'classname'     => 'mod_hyperaudio_external',
11        'methodname'   => 'load_comments',
12        'classpath'    => 'mod/hyperaudio/classes/external.php',
13        'description'  => 'Load comments',
14        'type'          => 'read'
15    )
16 );
```

---

Auflistung 5.11: Ausschnitt der **services.php** in der 4. Iteration

In der **external.php** wurden in der Funktion `mod_hyperaudio_external`, welche von der Klasse `external_api` erbt, die Webservice-Funktionen definiert. Dabei sind für einen Webservice jeweils vier Funktionen von Nöten:

- `<webservice_name>_parameters`: definiert die Eingabe-Parameter des Webservices
- `<webservice_name>_is_allowed_from_ajax`: definiert, ob der Webservice über AJAX (Asynchronous JavaScript and XML) aufgerufen werden kann
- `<webservice_name>`: Hauptfunktion, welche die Eingabe-Parameter verarbeitet und ein Ergebnis zurückliefert
- `<webservice_name>_returns`: definiert die Struktur der Rückgabewerte des Webservices

Auflistung 5.12 zeigt exemplarisch die Umsetzung des Webservices `load_comments`. Anhand der übergebenen `$hyperaudio_id` werden die Kommentare mittels SQL (Structured Query Language) aus der Datenbank gelesen und in einer aufbereiteten Struktur zurückgegeben.

---

```
1 class mod_hyperaudio_external extends external_api {
2
3     public static function load_comments_parameters() {
4         return new external_function_parameters(
5             array(
6                 'hyperaudio_id' => new external_value(PARAM_INT, 'hyperaudio_id')
7             )
8         );
9     }
10
11    public static function load_comments_is_allowed_from_ajax() {
12        return true;
13    }
14
15    public static function load_comments($hyperaudio_id) {
16        global $CFG, $DB;
17
18        require_once ($CFG->dirroot . '/mod/hyperaudio/locallib.php');
19        require_once ($CFG->dirroot . '/mod/hyperaudio/classes/enums.php');
20
21        $comments = $DB->get_records_sql(
22            'SELECT comments.id, comments.comment_type,
23            comments.commenttext, comments.timeannotated,
24            comments.timecreated, user.username,
25            FROM mdl_hyperaudio_comments comments'.
```

```

26      ' INNER JOIN mdl_user user ON comments.userid = user.id'.
27      ' WHERE comments.hyperaudio_id = ?'.
28      ' AND (comments.comment_type = ?)'.
29      ' ORDER BY comments.timecreated',
30      array($hyperaudio_id, CommentType::Comment)
31  );
32
33  $result_comments = array();
34  foreach ($comments as $comment) {
35      $result_comment = array(
36          'id' => $comment->id,
37          'username' => $comment->username,
38          'date' => date('d.m.Y H:i', $comment->timecreated),
39          'time' => format_time_annotated($comment->timeannotated),
40          'text' => $comment->commenttext,
41          'comment_type' => $comment->comment_type
42      );
43      $result_comments[] = $result_comment;
44  }
45
46  return $result_comments;
47 }
48
49 public static function load_comments_returns() {
50     return new external_multiple_structure(
51         new external_single_structure(
52             array(
53                 'id' => new external_value(PARAM_INT, 'id'),
54                 'username' => new external_value(PARAM_TEXT, 'username'),
55                 'date' => new external_value(PARAM_TEXT, 'date'),
56                 'time' => new external_value(PARAM_TEXT, 'time'),
57                 'text' => new external_value(PARAM_TEXT, 'text'),
58                 'comment_type' => new external_value(PARAM_TEXT, 'comment_type')
59             )
60         )
61     );
62 }
63 }

```

Auflistung 5.12: Ausschnitt der **external.php** in der 4. Iteration

Für den Zugriff auf die Webservices wurde das JavaScript-Modul **hyperaudio.js**, welches sich im Ordner **/hyperaudio/AMD** befindet, eingeführt. Das Modul dient dazu die im Renderer definierten Elemente zu befüllen beziehungsweise auszulesen. Die Funktion zum Speichern eines Kommentars ist exemplarisch in Auflistung 5.13 dargestellt. Zunächst werden die im Renderer erstellten Elemente zum Erstellen eines Kommentars ausgelesen. Um Funktionen die mehrfach benötigt werden, wie die Funktion `get_hyperaudio_id`, auszugliedern, wurde die Datei **locallib.js** im Ordner **/hyperaudio/thirdparty** implementiert. Nachdem die Elemente ausgelesen wurden, wird mittels eines AJAX-Calls der Webservice `mod_hyperaudio_save_comment` aufgerufen und die benötigten Parameter übergeben. Hierbei ist hervorzuheben, dass der Zeitpunkt, zu dem der Kommentar an das Hyperaudio-Dokument annotiert wird, über die Funktion `currentTime` des *Popcorn.js*-Frameworks ermittelt werden kann. Bei der Funktion `show_comments` zum Darstellen der Kommentare wurde analog vorgegangen, nur dass durch den Webservice die Kommentare für das betroffene Hyperaudio-Dokumente abgefragt und innerhalb des dazugehörigen Elements des Renderers dargestellt werden.

---

```

1 function save_comment(comment_type) {
2     var comment = $("#hyperaudio_comment").val();
3     var hyperaudio_id = get_hyperaudio_id();
4     var popcorn = Popcorn("#hyperaudio_audio");
5     var timeannotated = popcorn.currentTime();

```

## 5 Implementierung

```
6     timeannotated = parseInt(timeannotated);
7
8     var promises = ajax.call([
9         methodname: 'mod_hyberaudio_save_comment',
10        args:{
11            'comment': comment,
12            'hyberaudio_id': hyberaudio_id,
13            'timeannotated': timeannotated,
14            'comment_type': comment_type
15        }
16    });
17    promises[0].done(function(data) {
18        show_comments();
19    });
20 }
```

Auflistung 5.13: Ausschnitt der **hyberaudio.js** in der 4. Iteration

```
1 function get_hyberaudio_id(){
2     return $("#hyberaudio").attr("data-hyberaudio_id");
3 }
4 function jump_to_time(element){
5     var time = $(element).attr("data-annotated_time");
6     var popcorn = get_popcorn();
7     popcorn.currentTime(time);
8     popcorn.play();
9 }
```

Auflistung 5.14: Ausschnitt der **locallib.js** in der 4. Iteration

In der **locallib.js** ist darüber hinaus mit `jump_to_time` eine Funktion enthalten, mit welcher die Rückkopplung an die Mediensteuerung ermöglicht wurde.

Mit dem Abschluss dieser Iteration wurde die Möglichkeit geschaffen, Kommentare zu verfassen und dabei den Zeitpunkt innerhalb des Hyberaudio-Dokuments festzuhalten, zu dem der Kommentar erstellt wurde. Im gleichen Zuge wurde auch die Darstellung dieser Kommentare ermöglicht.

### 5.2.5 Antworten auf Kommentare

Im nächsten Schritt sollte nun eine Antwortmöglichkeit auf Kommentare geschaffen werden. Zu diesem Zweck wurde ein neuer Webservices zum Speichern von Antworten erstellt und der Webservice zum Darstellen der Kommentare um Antworten erweitert. Bei der **hyberaudio.js** bedurfte es ebenfalls nur kleinerer Anpassungen, um unter anderem die Anzeige eines Antworten-Buttons und eines Textfeldes bei dessen Betätigung zu ermöglichen (siehe Abbildung 5.3). Die hierfür notwendige Erweiterung innerhalb der Funktion `show_comments` ist in der Auflistung 5.15 zu sehen.



Abbildung 5.3: Darstellung der Antworten-Funktion

```
1 if (this.comment_type === CommentType.Comment && !this.comment_id) {
2     output = output + '<div' +
```

```

3      ' class="comment_answer comment_actions"' +
4      ' data-comment_id="' + this.id + '"';
5 output = output + '<a href="javascript:void(0);"' +
6      ' class="comment_answer_link comment_link active"' +
7      ' onclick="show_answer_textarea(this)"' +
8      ' data-comment_id="' + this.id + '">' + string_answer + '</a>';
9 output = output + '<div class="textarea_container">';
10 output = output + '<textarea' +
11      ' class="comment_answer_text comment_hidden_element hyperaudio_textarea"' +
12      ' data-comment_id="' + this.id + '" ' +
13      ' placeholder="' + string_answer + '..."/>';
14 output = output + '<button type="button"' +
15      ' class="comment_answer_button comment_hidden_element"' +
16      ' hyperaudio_comment_button"' +
17      ' data-comment_id="' + this.id + '">' + string_answer + '</button>';
18 output = output + '</div>';
19 output = output + '</div>';

```

Auflistung 5.15: Ausschnitt der **hyperaudio.js** in der 5. Iteration

Durch die Erweiterungen und Anpassungen innerhalb dieser Iteration war es nun möglich auf Kommentare zu antworten und diese Antwort beim zugehörigen Kommentar darzustellen.

### 5.2.6 Notizen

Nachdem durch die vorangegangenen Iterationen nun die Funktionen für die Kommunikation zwischen Studierenden und Lehrenden geschaffen wurden, zielte diese Iteration auf das Erstellen und Darstellen von Notizen ab.

Zum Speichern der Notizen konnten dieselben Webservices und Funktionen, wie zum Speichern der Kommentare verwendet werden. Dies röhrt daher, dass Notizen eine bestimmte Art von Kommentaren darstellen (vgl. Abschnitt 4.1.2).

Da Notizen im Gegensatz zu Kommentaren geändert und gelöscht werden können sollen, mussten hierfür entsprechende Webservices bereitgestellt werden. Gleichzeitig mussten analog zum Antworten auf Kommentare ein Button zum Löschen sowie Button und Textfeld für die Bearbeitung der Notiz ergänzt werden (siehe Auflistung 5.16).

Die Darstellung des Schloss-Symbols als Symbol für Notizen wurde mithilfe der schriftbasierten Icon-Sammlung *Font Awesome*<sup>18</sup> realisiert.



Abbildung 5.4: Darstellung der Notiz-Funktion

```

1 else if (this.comment_type === CommentType.Note) {
2     output = output + '<div' +
3         ' class="comment_note comment_actions" data-comment_id="' + this.id + '"';
4     output = output + '<a href="javascript:void(0);"' +
5         ' class="comment_note_edit_link comment_link active"' +
6         ' onclick="show_note_edit_textarea(this)"' +
7         ' data-comment_id="' + this.id + '">' + string_edit + '</a>';

```

<sup>18</sup><https://fontawesome.com/>

## 5 Implementierung

```
8     output = output + '<a href="javascript:void(0);"' +
9         ' class="comment_note_delete_link comment_link active"' +
10        ' data-comment_id="' + this.id + '">' + string_delete + '</a>';
11    output = output + '<div class="textarea_container">';
12    output = output + '<textarea' +
13        ' class="comment_note_edit_text comment_hidden_element"' +
14        ' hyperaudio_textarea"' +
15        ' data-comment_id="' + this.id + '">' + this.text + '</textarea>';
16    output = output + '<button type="button"' +
17        ' class="comment_note_edit_button comment_hidden_element"' +
18        ' hyperaudio_comment_button"' +
19        ' data-comment_id="' + this.id + '">' + string_edit + '</button>';
20    output = output + '</div>';
21    output = output + '</div>';
22 }
```

Auflistung 5.16: Ausschnitt der **hyperaudio.js** in der 6. Iteration

Entsprechend der Webservices wurden innerhalb der **external.php** Funktionen zum Editieren von Notizen und dem Löschen von Kommentaren erstellt. Damit waren alle nötigen Schritte durchgeführt um Notizen erstellen, anzeigen und löschen zu können.

### 5.2.7 Audio Cues

Das Abspielen der Audio Cues im Moment der Darstellung eines Zusatzinhaltes wurde durch eine Erweiterung des verwendeten *Popcorn.js*-Plugin *Images* erreicht. In diesem Zug wurde das *Popcorn.js*-Plugin *Hyperaudio* basierend auf dem *Images*-Plugin entwickelt.

Hierfür wurde in den Optionen ein zusätzlicher Parameter `cue` eingeführt. Dieser nimmt die ID eines `<audio>`-Elements entgegen, welches den abzuspielenden Audio Cue wiedergeben kann (vgl. Auflistung 5.17). Um den Ton zeitgleich mit der Anzeige des Zusatzinhaltes abzuspielen, wurde eine Anpassung in der Funktion `start` vorgenommen. Sofern keine Stummschaltung oder Pausierung des Hyperaudio-Dokuments vorliegt, wird die Wiedergabe des Audio Cues durch den Aufruf der `play`-Funktion des HTML5-Audioplayers angestoßen. Auflistung 5.18 zeigt die Erweiterungen des *Popcorn.js*-Plugins *Hyperaudio*.

Wenn ich das richtig verstehe, gibt es ein einziges Tonsignal für alle Arten von Annotationen.  
Wie soll der Nutzer erkennen, ob an der gegenwärtigen Abspielposition ein Komentar, eine Notitz oder ein Medienelement hinterlegt ist?

```
1 $output .= '<audio id="hyperaudio_audio_cue"' .
2     ' src="../hyperaudio/audiocues/ding.mp3"></audio>';
3
4 //...
5
6 $output .= 'popcorn.hyperaudio({
7     ' start: '.$additional_content->begin.','.
8     ' end: '.$additional_content->end.','.
9     ' href: "javascript:void(0);",'.
10    ' src: "'.$additional_content_url.'",'.
11    ' target: "hyperaudio_additional_content",'.
12    ' cue: "hyperaudio_audio_cue"});';
```

Auflistung 5.17: Ausschnitt der **renderer.php** in der 7. Iteration

```
1 options: {
2
3     //...
```

```

4
5      cue: "audio-cue",
6
7      //...
8
9 }
10 //...
11
12
13 start: function( event, options ) {
14     options.anchor.style.display = "inline";
15     if ( options.trackedContainer ) {
16         options.trackedContainer.start();
17     }
18     var audioElement = document.getElementById( options.cue );
19     if ( !this.media.muted && !this.media.paused) {
20         audioElement.play();
21     }
22 },

```

Auflistung 5.18: Ausschnitt der **popcorn.hyperaudio.js** in der 7. Iteration

Durch die Weiterentwicklung des *Images*-Plugins zum *Popcorn.js*-Plugin *Hyperaudio* wurde in dieser Iteration die Wiedergabe von Audio Cues beim Anzeigen von Zusatzinhalten implementiert.

### 5.2.8 Galerie der Zusatzinhalte

Der Anforderung nach einer Übersicht über annotierte Zusatzinhalte wurde mittels einer Galerie begegnet, welche in dieser Iteration umgesetzt wurde. Hierfür wurden in der **renderer.php** zunächst die Metainformationen zu allen Zusatzinhalten des Hyperaudio-Dokuments geladen, um anhand dieser im nächsten Schritt URLs zu den Zusatzinhalten zu generieren. Damit wurde eine Darstellung gemäß der Idee aus Abbildung 4.6e umgesetzt (siehe Abbildung 5.5). Die Rückkopplung zur Mediensteuerung erfolgte wie bereits bei den Kommentaren mithilfe der `jump_to_time`-Funktion.



Abbildung 5.5: Darstellung der Galerie

Um eine vergrößerte Darstellung der Zusatzinhalte zu realisieren, wurde ein versteckter Bereich angelegt. Beim Klick auf ein Vorschaubild wird mithilfe der Funktion `show_modal` der entsprechende Zusatzinhalt in diesen Bereich geladen und dargestellt (vgl. Auflistung 5.19).

---

```
1 function show_modal(source, caption){
```

## 5 Implementierung

```
2     $("#hyperaudio_modal_image").attr("src", source);
3     $("#hyperaudio_modal_caption").html(caption);
4     $("#hyperaudio_modal").css("display", "block");
5 }
```

---

Auflistung 5.19: Ausschnitt der **hyperaudio.js** in der 8. Iteration

Durch diese Implementierung wurde die Darstellung der Galerie erreicht, durch Verwendung der JavaScript-Funktion `jump_to_time` wurde zusätzlich noch die Rückkopplung an die Mediensteuerung geboten.

### 5.2.9 Zeitabhängige Visualisierung der Kommentare und Notizen

In dieser Iteration sollte sich nun der zeitabhängigen Visualisierung der Kommentare und Notizen zugewendet werden. An dieser Stelle muss festgehalten werden, dass dies beim Einsatz des Standard HTML5-Audioplayers schwer umsetzbar gewesen wäre, weshalb auf die Erkenntnisse bezüglich des *VideoJS Players* aus Abschnitt 3.4.2 zurückgegriffen wurde. Die umfassenden Anpassungsmöglichkeiten sollten sich nun in dieser Iteration zu Nutze gemacht werden.

Im ersten Schritt wurden zunächst die benötigten Informationen für die Darstellung der in Abbildung 4.5c beschriebenen Blöcke aus der Datenbank abgerufen. Hierfür wurde ein neuer Webservice eingeführt und unter anderem die **external.php** erweitert.

Um die ermittelten Informationen darzustellen, wird, wie in Auflistung 5.20 zu erkennen, der Webservice von der Funktion `show_timeline` der **hyperaudio.js** aufgerufen und darauf die Größe der Blöcke unter Berücksichtigung der Größe der Timeline berechnet. Dabei wird auf vorhandene HTML Elemente des *VideoJS Players* zurückgegriffen, um die gewünschte Darstellung der Blöcke ober- und unterhalb der Timeline zu erreichen.

```
1 var promises = ajax.call([
2   methodname: 'mod_hyperaudio_load_timeline_comments',
3   args: {
4     'hyperaudio_id': hyperaudio_id,
5     'duration': duration,
6     'number_of_slots': number_of_slots
7   }
8 ]);
9 promises[0].done(function(data) {
10   var container_height =
11     $(".hyperaudio_timeline_blocks_container.comment_blocks_container") .
12       height();
13
14   var pixel_width =
15     ((container_width - (number_of_slots - 1) * 2 * 2) / number_of_slots);
16   var width = (pixel_width / (container_width + 20)) * 100;
17   var height_1_comment = ((container_height - 5) / data.max_count);
18
19   var output_comments = '';
20   var output_notes = '';
21   var height;
22   var class_zero;
23
24   function set_height_and_class(annotations) {
25     height = annotations * height_1_comment;
26     if (height == 0) {
27       class_zero = 'block_zero';
28     }
29     else {
30       class_zero = '';
```

```

31      }
32      if (height < 5) {
33          height = 5;
34      }
35  }
36
37  function get_title(title, username) {
38      var result_title = '';
39
40      if (username) {
41          result_title = username + ':\\n';
42      }
43
44      if (title) {
45          if (title.length > 100) {
46              result_title = result_title + title.substr(0, 100) + " ...";
47          }
48          else {
49              result_title = result_title + title;
50          }
51      }
52
53      return result_title;
54  }
55
56  $(data.blocks).each(function() {
57      set_height_and_class(this.comments);
58      output_comments = output_comments +
59          '<div title="' +
60          get_title(this.title_comment_text, this.title_comment_username) + '" +
61          ' class="timeline_block comment_block' + class_zero + '" +
62          ' data-comment-id="' + this.title_comment_id + '" +
63          ' style="width: ' + width + '%; height: ' + height + 'px;"' +
64          ' onclick="jump_to_comment(this)"></div>';
65
66      set_height_and_class(this.notes);
67      output_notes = output_notes +
68          '<div title="' + get_title(this.title_note_text) + '" +
69          ' class="timeline_block note_block' + class_zero + '" +
70          ' data-comment-id="' + this.title_note_id + '" +
71          ' style="width: ' + width + '%; height: ' + height + 'px;"' +
72          ' onclick="jump_to_comment(this)"></div>';
73  });
74  $(".hyperaudio_timeline_blocks.comment_blocks").html(output_comments);
75  $(".hyperaudio_timeline_blocks.note_blocks").html(output_notes);
76 })();

```

---

Auflistung 5.20: Ausschnitt der **hyperaudio.js** in der 9. Iteration

Beim Klick auf einen Block wird mit der Funktion `jump_to_comment` zu dem Kommentar gesprungen, welcher im Tooltip dargestellt wird. Selbiges gilt für Notizen. Hierbei mussten, wie in Auflistung 5.21 zu sehen, auch Standard-Moodle-Elemente (Titelleiste) berücksichtigt werden, um die korrekte Stelle zu ermitteln, zu der gesprungen werden muss.

---

```

1  function jump_to_comment(element) {
2      var comment_id = $(element).attr("data-comment-id");
3
4      if (comment_id != "null") {
5          $('html, body').animate({
6              scrollTop:
7                  $('.comment[data-comment-id="' + comment_id + '"]').offset().top
8                  - $(".fixed-top.navbar").outerHeight()

```

## 5 Implementierung

```
9         }, 'slow');
10    }
11 }
```

---

Auflistung 5.21: Ausschnitt der **locallib.js** in der 9. Iteration

Als Ergebnis dieser Iteration ergab sich eine in die Mediensteuerung integrierte Visualisierung der Kommentare und Notizen, welche nicht nur der Übersicht sondern auch der Navigation dient. Um die Kommentare und Antworten anderer Nutzer einsehen zu können ohne das Hyperaudio-Dokument erneut öffnen zu müssen, wurde eine Echtzeitabfrage implementiert, die sich sowohl auf die Anzeige in der Mediensteuerung als auch auf den Kommentarbereich erstreckt.

### 5.2.10 Lesezeichen

Um die Vorstellung aus Abschnitt 4.4.2 zu vervollständigen mussten nun noch die Lesezeichen in die Mediensteuerung integriert werden. Dementsprechend wurde in dieser Iteration das Erstellen und Visualisieren von Lesezeichen behandelt.

In die vorhandenen Bedienelemente des *VideoJS Players* wurde eine weitere Schaltfläche zum Erstellen von Lesezeichen integriert (vgl. Auflistung 5.22 Zeilen 1-9).

```
1 $(".vjs-volume-panel").after(
2   '<button class="vjs-bookmark-control vjs-control vjs-button"' +
3   ' type="button" title="' + string_bookmark + '" aria-disabled="false">' +
4   '<span aria-hidden="true" class="far fa-bookmark vjs-icon-placeholder">' +
5   '</span>' +
6   '<span class="vjs-control-text" aria-live="polite">' + string_bookmark +
7   '</span>' +
8   '</button>'
9 );
10 //...
11
12 promises[1].done(function(data) {
13   var output = '';
14   $(data).each(function() {
15     var left = (this.timeannotated / parseFloat(duration)) * 100;
16     output = output + '<div title="' + string_delete_bookmark + '" ' +
17       ' class="bookmark" style="left: ' + left + '%;"' +
18       ' data-comment_id="' + this.id + '">' +
19       '<i class="far fa-bookmark"/></div>';
20   });
21   $(".hyperaudio_timeline_bookmarks_container").html(output);
22 });
23 };
```

---

Auflistung 5.22: Ausschnitt der **hyperaudio.js** in der 10. Iteration

Das Speichern der Lesezeichen erfolgt nach Betätigung der Schaltfläche analog zum Speichern der Kommentare (vgl. Abschnitt 5.2.4). Zur Darstellung der Lesezeichen wurde die Funktion `show_timeline` aus der **hyperaudio.js** erweitert sowie ein neuer Webservice für das Laden von Lesezeichen eingeführt. Der Webservice liefert der `show_timeline`-Funktion die Lesezeichen des aktuellen Benutzers. Ihre Position innerhalb der Timeline ist durch einen prozentualen Abstand von links festgelegt, wobei sich der Prozentsatz durch das Verhältnis des Annotationszeitpunktes zur Abspieldauer ergibt (vgl. Auflistung 5.22 Zeilen 12-22).

Mit einem Rechtsklick auf ein Lesezeichen wird die Löschung mit derselben Prozedur wie bei den Notizen abgearbeitet (vgl. Abschnitt 5.2.6). Somit war mit dieser Iteration auch das Erstellen, Anzeigen und Löschen von Lesezeichen möglich und die Implementierung der Mediensteuerung abgeschlossen.

Die visuelle Umsetzung ist Abbildung 5.6 zu entnehmen. In Anlehnung an den Designentwurf aus Abbildung 4.5c werden die Blöcke für Kommentare in blau und jene für Notizen in orange dargestellt. Zusätzlich werden Lesezeichen durch ein entsprechendes Symbol in der Timeline visualisiert und der Button zum Erzeugen von Lesezeichen in die Mediensteuerung links von der Timeline integriert.



Abbildung 5.6: Darstellung der Mediensteuerung inklusive Visualisierung der Kommentare, Notizen und Lesezeichen

### 5.2.11 Suche, Filter und Sortierung bei Kommentaren

Die elfte Iteration beschäftigte sich mit den gewünschten Such-, Filter- und Sortierungsmöglichkeiten innerhalb der Kommentare und Notizen.

Dafür wurde ein Eingabefeld, zwei Checkboxen und ein Dropdown-Menü in der **renderer.php** ergänzt. Die Filterung und Sortierung der Daten erfolgte mithilfe eines SQL-Statements, welches der Auflistung 5.23 zu entnehmen ist. Die einzelnen Funktionen wurden wie folgt umgesetzt:

- **Filter**

zusätzliche WHERE-Bedingung (Zeile 13):

```
comments.comment_type IN (:p_filter_comment, :p_filter_note)
```

Die Bindevariablen enthalten den jeweiligen Kommentar-Typ, falls dieser angezeigt werden soll und sind sonst leer (vgl. Zeilen 41-42).

- **Suche**

zusätzliche WHERE-Bedingung (Zeile 14):

```
$DB->sql_like('comments.commenttext', ':p_search_string')
```

Mithilfe der Funktion `sql_like` kann überprüft werden, ob der Kommentar den eingegebenen Suchbegriff enthält (vgl. Zeile 43).

- **Sortierung**

verschiedene ORDER BY-Clauses abhängig von der gewählten Sortierreihenfolge (Zeilen 16-35)

Die drei verschiedenen Kriterien werden in entsprechender Reihenfolge angewandt:

- Annotationszeitpunkte
- Erstellungszeitpunkte der Original-Kommentare
- Erstellungszeitpunkte der Antwort-Kommentare

---

```

1 $sql='SELECT comments.id, comments.comment_type, comments.commenttext,'.
2   ' comments.timeannotated, comments.timecreated,'.
3   ' user.username, comments.comment_id'.
4   ' FROM mdl_hydraudio_comments comments'.
5   ' INNER JOIN mdl_user user ON comments.userid = user.id'.
6   ' LEFT JOIN mdl_hydraudio_comments original_comment'.
7   ' ON original_comment.id = comments.comment_id'.
8   ' WHERE comments.hydraudio_id = :p_hydraudio_id'.
9   ' AND (comments.comment_type = :p_comment_type_c'.
10  ' OR (comments.comment_type = :p_comment_type_note'.
11  ' AND comments.userid = :p_userid))'.
12  ' AND comments.comment_type'.
13  ' IN (:p_filter_comment, :p_filter_note)'.
14  ' AND ' . $DB->sql_like('comments.commenttext', ':p_search_string');
15

```

## 5 Implementierung

```
16 switch ($order){
17     case timecreated_asc:
18         $sql.=
19             'ORDER BY'.
20             ' IFNULL(original_comment.timecreated, comments.timecreated) asc,'.
21             ' comments.timecreated asc';
22         break;
23     case timecreated_desc:
24         $sql.=
25             'ORDER BY'.
26             ' IFNULL(original_comment.timecreated, comments.timecreated) desc,'.
27             ' comments.timecreated asc';
28         break;
29     case timeannotated:
30         $sql.=
31             'ORDER BY comments.timeannotated asc,'.
32             ' IFNULL(original_comment.timecreated, comments.timecreated) asc,'.
33             ' comments.timecreated asc';
34         break;
35 }
36
37 $comments = $DB->get_records_sql($sql, array('p_hyperraudio_id' => $hyperraudio_id,
38     'p_comment_type_c' => CommentType::Comment,
39     'p_comment_type_note' => CommentType::Note,
40     'p_userid' => $USER->id,
41     'p_filter_comment' => $filter_comments ? CommentType::Comment : null,
42     'p_filter_note' => $filter_notes ? CommentType::Note : null,
43     'p_search_string' => '%' . $DB->sql_like_escape($search_string) . '%'));
```

Auflistung 5.23: Ausschnitt der `external.php` in der 11. Iteration

Die Umsetzung der Such-, Filter- und Sortierfunktion (siehe Abbildung 5.7) rundeten die Implementierung der Wiedergabe von Hyperraudio-Dokumenten mit Kommunikations- und Interaktionsmöglichkeiten ab.



Abbildung 5.7: Darstellung der Such-, Filter- und Sortierfunktion

### 5.2.12 Unterstützung mobiler Endgeräte

In der letzten Iteration wurden Anpassungen vorgenommen, um die Nutzung des Hyperraudio-Plugins auch auf mobilen Endgeräten zu ermöglichen. Dafür wurden wie bereits in Abschnitt 4.4.6 beschrieben Breakpoints eingeführt, um die Anordnung der Elemente für mobile Endgeräte zu optimieren.

Bei geringer Bildschirmbreite werden durch den ersten Breakpoint die Bereiche wie in Abbildung 5.8 sichtbar neu angeordnet, sodass statt einem zweispaltigen ein einspaltiges Layout verwendet wird. Das hat zur Folge, dass die Galerie nun unterhalb der Kommentare und Notizen angezeigt wird.

Ein weiterer Breakpoint wird bei der Such-, Filter- und Sortierfunktion angewandt. Für mobile Endgeräte werden diese Funktionen untereinander statt nebeneinander angeordnet (siehe Abbildung 5.9).

Um auch innerhalb der Galerie auf die Gegebenheiten von Endgeräten mit geringer Bildschirmbreite Rücksicht zu nehmen, wurde ein weiterer Breakpoint eingeführt. Dieser sorgt dafür, dass die Metainformationen der Zusatzinhalte unterhalb statt neben des Thumbnails dargestellt werden. Hieraus resultiert die in Abbildung 5.10 dargestellte Ansicht.

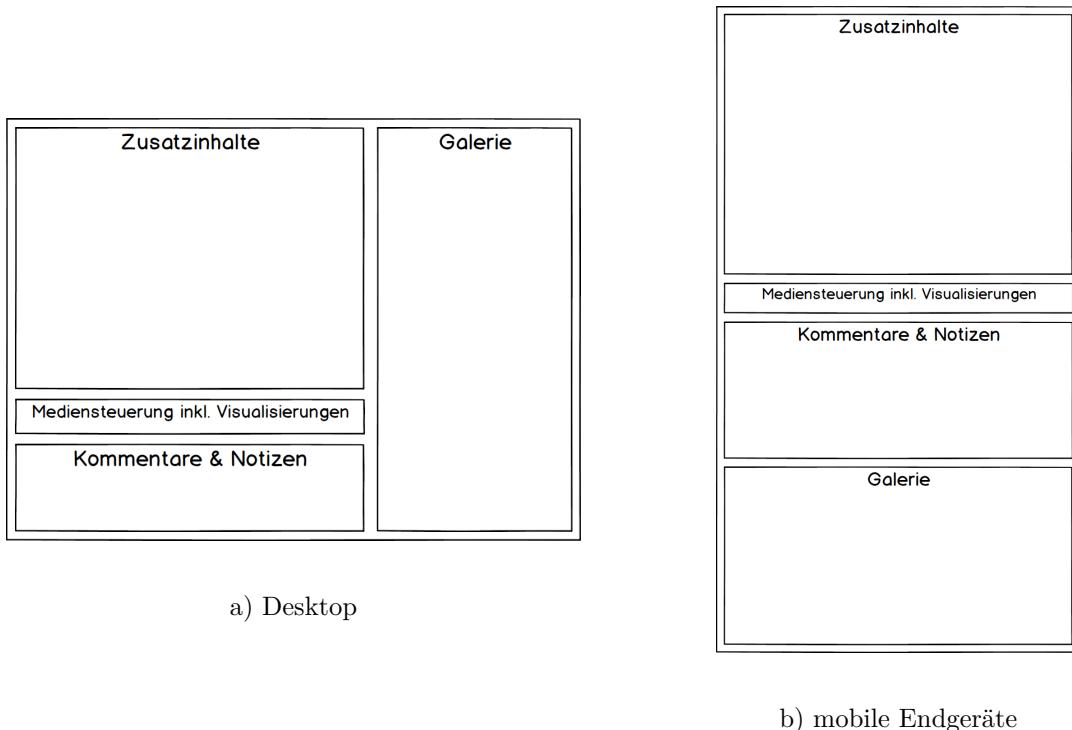


Abbildung 5.8: Darstellung der Bereiche



Abbildung 5.9: Darstellung der Such-, Filter- und Sortierungsfunktion

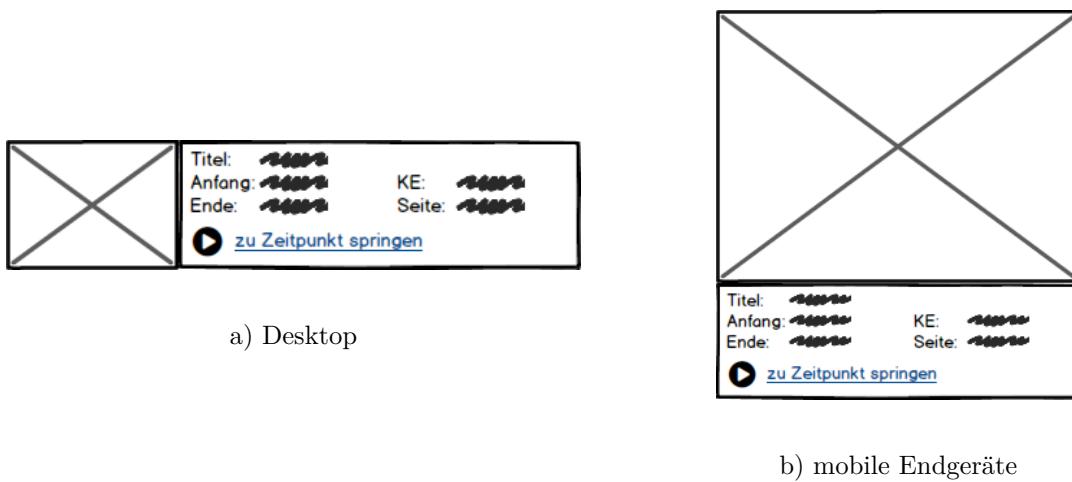


Abbildung 5.10: Darstellung der Galerie

Umgesetzt wurden diese Breakpoints anhand von CSS-Regeln innerhalb der **styles.css** (siehe Auflistung 5.24). Die verschiedenen Darstellungen werden mittels Medienabfragen gesteuert. Dabei wird

## 5 Implementierung

die Breite des Anzeigebereichs abgefragt und falls die angegebene Grenze überschritten wird, werden die innerhalb des @media-Blocks definierten Regeln angewandt. Diese erzeugen mittels Grid Layout die Desktop-Darstellungsvariante. Andernfalls gelten die „normalen“ CSS-Regeln, welche die Darstellung auf mobilen Endgeräten definieren.

---

```
1 @media (min-width: 100em) {  
2     div#hyperaudio div#hyperaudio {  
3         display: grid;  
4         grid-template-columns: 70em 1fr;  
5     }  
6     div#hyperaudio div#gallery_container {  
7         margin-left: 2em;  
8     }  
9 }  
10 @media (min-width: 40em) {  
11     div#hyperaudio div.gallery_additional_content {  
12         display: grid;  
13         grid-template-columns: 1fr 5fr;  
14     }  
15 }  
16 @media (min-width: 50em) {  
17     div#hyperaudio div#comment_tools_container {  
18         display: grid;  
19         grid-template-columns: 4fr 2fr 3fr;  
20     }  
21     div#hyperaudio input#hyperaudio_search {  
22         max-width: 35em;  
23         margin-top: 0em;  
24     }  
25     div#hyperaudio div#comment_tools_filter,  
26     div#hyperaudio div#comment_tools_order {  
27         display: flex;  
28         justify-content: flex-end;  
29         margin-left: 20px;  
30     }  
31 }  
32 div#hyperaudio div.gallery_additional_content {  
33     margin-bottom: 15px;  
34     padding: 5px 10px;  
35     border-radius: 10px;  
36 }  
37 div#hyperaudio div#comment_tools_container {  
38     align-items: baseline;  
39     margin-top: 1rem;  
40     max-width: 70em;  
41 }  
42 div#hyperaudio input#hyperaudio_search {  
43     font-family: Arial, FontAwesome;  
44     font-weight: normal;  
45     padding: 5px 10px;  
46     border: 1px solid lightgray;  
47     border-radius: 20px;  
48     width: 100%;  
49     margin-top: 0.3em;  
50 }  
51 div#hyperaudio div#comment_tools_order {  
52     align-items: baseline;  
53 }
```

---

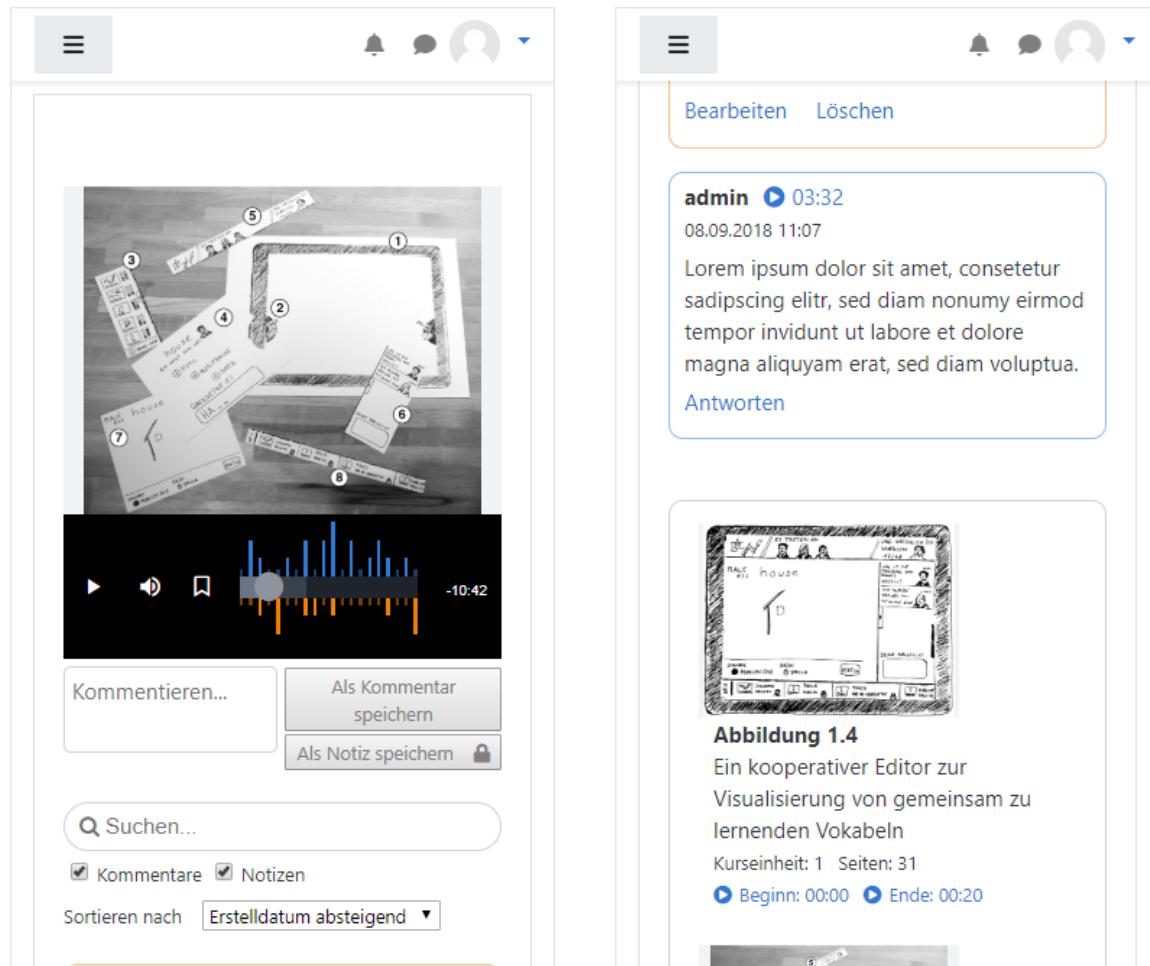
Auflistung 5.24: Ausschnitt der **styles.css** in der 12. Iteration

**Abbildung 1.4**  
Ein kooperativer Editor zur Visualisierung von gemeinsam zu lernenden Vokabeln  
Kursinhalt 1 Seiten: 31  
• Beginn: 00:00 • Ende: 00:20

**Abbildung 1.5**  
Verschiedene Komponenten des Paperprototyps  
Kursinhalt 1 Seiten: 32  
• Beginn: 01:02 • Ende: 01:40

Abbildung 5.11: Darstellung der finalen Desktop-Version

## 5 Implementierung



### 5.3 Zusammenfassung

Nachdem zu Beginn der Implementierung die grundlegende Moodle-Struktur erarbeitet wurde, wurden in den darauffolgenden Iterationen das Hyperaudio-Plugin Schritt für Schritt um die einzelnen Anforderungen erweitert. Somit ist am Ende die in Abbildung 5.11 sichtbar finale Repräsentation von Kurseinheiten als Hyperaudio-Dokumente innerhalb eines Moodle-Plugins entstanden. In der letzten Iteration wurde dann noch die Darstellung auf mobilen Endgeräten implementiert.

# 6 Evaluation

Nachdem nun die Implementierung des Hyperaudio-Plugins abgeschlossen ist, kann zunächst eine Validierung anhand der im Abschnitt 3.1.2 formulierten User Stories und den daraus abgeleiteten Anforderungen aus Abschnitt 3.2 erfolgen. Im nächsten Schritt erfolgt die Evaluation für die Rolle der Nutzenden anhand des User Experience Questionnaire (UEQ). Nachfolgend wird noch eine Möglichkeit zur Evaluation für die Rolle der Lehrenden beschrieben. Daraufhin werden mögliche Verbesserungen anhand von weiteren User Stories formuliert.

Die Frage nach der Umsetzbarkeit ist keine wissenschaftliche Frage, da es hierbei (trotz aller Mühen und fleißigen Programmierarbeit) da hier aus technischer Sicht keine herausragenden Problemstellungen bearbeitet wurden.

Es mag dennoch wichtig und richtig sein, hier eine Bilanz zu ziehen, welche User Stories bzw. Anforderungen in Ihrem Prototyp realisiert wurden. Eine solche Bilanz stellt jedoch nur eine Validierung dar. Eine Evaluation geht einen Schritt weiter.

Stellen Sie das doch als eine checkbox in einer Liste dar. Das umgesetzte erscheint so losgelöst.  
Zudem suggeriert der Abstand nicht die Zugehörigkeit.

## 6.1 Validierung der User Stories und Anforderungen

Nachfolgend wird untersucht, ob die User Stories mit der vorhandenen Implementierung umgesetzt werden können.

US-1: Als Administrierende möchte Prof. Dr. Karolin Schröder ein neues Hyperaudio-Dokument in ihrem Kurs „Einführung in die Wirtschaftsinformatik“ zur Verfügung stellen, um den Studierenden neue Lerninhalte bereitzustellen.

**umgesetzt**

Prof. Dr. Karolin Schröder ist in der Lage ein neues Hyperaudio-Dokument zu erstellen, indem sie Ihrem Kurs eine neue Aktivität *Hyperaudio* hinzufügt. Hierbei muss sie eine Audio-Datei, die gewünschten Zusatzinhalte und eine Konfigurationsdatei bereitstellen.

Sie fassen hier noch einmal das umgesetzte zusammen. Eine Evaluation würde eine kritische Analyse beinhalten. Idealer Weise würde Ihnen ein Experte oder eine Gruppe von potentiellen Anwendern Feedback zur Umsetzung der User Story geben. Fest steht, dass es keine richtige Lösung gibt, sondern viele alternative Lösungsmöglichkeiten. Warum ist Ihre Lösung eine gute Lösung?

US-2: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente aus ihrem Kurs „Einführung in die Wirtschaftsinformatik“ löschen können, um veraltete Informationen zu entfernen.

**umgesetzt**

Das Hyperaudio-Dokument kann Prof. Dr. Karolin Schröder löschen, indem sie die dazugehörige Aktivität *Hyperaudio* entfernt.

US-3: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente aus ihrem Kurs „Einführung in die Wirtschaftsinformatik“ im Sommersemester in den darauffolgenden Kurs im Wintersemester übernehmen, um diese nicht erneut erstellen zu müssen.

**teilweise umgesetzt**

Die Übernahme von Hyperaudio-Dokumenten in den darauffolgenden Kurs im Wintersemester ist Prof. Dr. Karolin Schröder nur über Umwege möglich. Hierzu müssen die einzelnen Dateien der Hyperaudio-Dokumente heruntergeladen werden und dann bei der Erstellung im neuen Kurs erneut hochgeladen werden. Der Titel und die Beschreibung des Hyperaudio-Dokuments muss manuell übernommen werden.

US-4: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente anderer Kurse in ihren Kurs „Einführung in die Wirtschaftsinformatik“ übernehmen, um auf die hervorragende Arbeit anderer Lehrender zurückgreifen zu können, da sich die Themen mit ihrem Kurs überschneiden.

**teilweise umgesetzt**

Die Hyperaudio-Dokumente anderer Kurse können Prof. Dr. Karolin Schröder analog zum Vorgehen bei US-3 nur anhand der einzelnen Dateien der Hyperaudio-Dokumente bereitgestellt werden. Diese müssen dann beim Erstellen der Hyperaudio-Dokumente verwendet werden.

US-5: Als Administrierende möchte Prof. Dr. Karolin Schröder Erkenntnisse daraus gewinnen, wie die Hyperaudio-Dokumente des Kurses „Einführung in die Wirtschaftsinformatik“ von Studierenden genutzt werden, um Verbesserungspotenzial auszumachen.

**umgesetzt** Prof. Dr. Karolin Schröder ist es möglich, über den Block *Aktivitäten* zu der Auflistung aller Hyperaudio-Aktivitäten ihres Kurses zu gelangen. In dieser Auflistung kann sie sowohl die Anzahl der erstellten öffentlichen Kommentare, persönlichen Notizen und Lesezeichen aller Nutzer entnehmen, als auch die Anzahl der Nutzer, die diese geniert haben.

US-6: Als Administrierender möchte Dr. Julian Schmidt ein vorhandenes Hyperaudio-Dokument in dem von ihm betreuten Kurs „Marketing“ überarbeiten, um einen Fehler zu beseitigen.

**umgesetzt** Das Bearbeiten von Hyperaudio-Dokumenten ist möglich. Es können Audio- und Konfigurationsdatei ausgetauscht werden. Auch Zusatzinhalte können getauscht, entfernt oder neue Zusatzinhalte hinzugefügt werden. Dabei ist zu beachten, dass einige dieser Änderungen auch die Anpassung der Konfigurationsdatei erfordern.

US-7: Als Nutzende möchte Prof. Dr. Karolin Schröder die bereits vorhandenen Hyperaudio-Dokumente aus ihrem Kurs „Einführung in die Wirtschaftsinformatik“ wiedergeben, um diese auf ihre Richtigkeit zu überprüfen.

**umgesetzt** Das Hyperaudio-Dokument kann nach dem Öffnen der *Hyperaudio-Aktivität* wiedergegeben werden.

US-8: Als Nutzende möchte Prof. Dr. Karolin Schröder die Kommentare zu einem Hyperaudio-Dokument lesen und beantworten können, um auf Fragen von Studierenden einzugehen.

**umgesetzt** Nach dem Öffnen einer *Hyperaudio-Aktivität* sind alle an das Hyperaudio-Dokument annotierten Kommentare sichtbar. Diese können nach einem Klick auf „Antworten“ beantwortet werden.

US-9: Als Nutzende möchte Prof. Dr. Karolin Schröder eine Notiz zu einem Hyperaudio-Dokument machen, um ihren Gedanken festzuhalten und später darauf zurückgreifen zu können.

**umgesetzt**

Das Erstellen einer Notiz wird durch das Befüllen des entsprechenden Textfeldes und der Betätigung

## 6 Evaluation

des „Als Notiz speichern“-Buttons erreicht. Die Notiz wird zum aktuellen Zeitpunkt der Wiedergabe annotiert.

**US-10:** Als Nutzender möchte Dr. Julian Schmidt eine gefundene Erklärungslücke in einem Hyperradio-Dokument durch einen Kommentar zum entsprechenden Zeitpunkt schließen, um eventuellen Fragen der Studierenden zuvorkommen.

**umgesetzt**

Dr. Julian Schmidt kann durch Befüllen des entsprechenden Textfeldes und der Betätigung des „Als Kommentar speichern“-Buttons einen Kommentar hinterlegen. Der Kommentar wird zum aktuellen Zeitpunkt der Wiedergabe annotiert.

**US-11:** Als Nutzende möchte Laura Ebert mittels Hyperradio-Dokument lernen, um die Zeit während Haushaltarbeiten, wie dem Bügeln, Kochen oder Putzen, und dem Pendeln sinnvoller zu nutzen.

**umgesetzt**

Durch die auditive Wiedergabeform der Hyperradio-Dokumente ist Laura Ebert in der Lage in den gewünschten Situationen mit Hyperradio-Dokumenten zu lernen. Falls ein visueller Zusatzinhalt dargestellt wird, so wird sie durch die Wiedergabe einer Audio Cue auf diesen hingewiesen.

**US-12:** Als Nutzende möchte Laura Ebert erfahren, welche Hyperradio-Dokumente in den von ihr belegten Kursen angeboten werden, um herauszufinden, mit welchen Mitteln sie sich auf die anstehenden Prüfungen vorbereiten kann.

**teilweise umgesetzt**

Laura Ebert kann sich mithilfe des Blocks *Aktivitäten* ausschließlich einen Überblick über alle Hyperradio-Dokumente eines jeweiligen Kurses verschaffen. Eine kursübergreifende Übersicht ist nicht vorhanden.

**US-13:** Als Nutzende möchte Laura Ebert einen Kommentar verfassen, um dem Kursbetreuer und den anderen Studierenden eine Frage zu stellen.

**umgesetzt**

Indem sie ihre Frage in das Kommentarfeld schreibt und dann mittels des entsprechenden Buttons speichert, kann Laura Ebert eine Frage an den Kursbetreuer und andere Studierende stellen.

**US-14:** Als Nutzende möchte Laura Ebert ein Lesezeichen setzen, wenn eine klausurrelevante Thematik erklärt wird. Bei der Prüfungsvorbereitung möchte sie anhand dieser Lesezeichen diejenigen Themen erkennen, mit welchen sie sich besonders intensiv beschäftigen möchte.

**umgesetzt**

Das Hinterlegen von Lesezeichen ist durch die Betätigung des Lesezeichen-Buttons links neben der Timeline möglich. Das Lesezeichen wird zum Wiedergabezeitpunkt erstellt und in Form eines Lesezeichen-Symbols in der Timeline dargestellt.

**US-15:** Als Nutzende möchte Laura Ebert ein Lesezeichen löschen, da sie den markierten Lerninhalt inzwischen beherrscht. Anhand der übrigen Lesezeichen möchte sie schnell erkennen, wo für sie noch Lernbedarf besteht.

**umgesetzt**

Lesezeichen können durch Rechtsklick auf das entsprechende Lesezeichen-Symbol in der Timeline gelöscht werden.

**US-16:** Als Nutzende möchte Laura Ebert eine Notiz erstellen, um ein Beispiel zu dem genannten Sachverhalt festzuhalten, sodass sie die Thematik beim nächsten Mal einfacher nachvollziehen kann.

**umgesetzt**

Das Erstellen einer Notiz wird durch das Befüllen des entsprechenden Textfeldes und der Betätigung des „Als Notiz speichern“-Buttons erreicht. Die Notiz wird zum aktuellen Zeitpunkt der Wiedergabe annotiert.

US-17: Als Nutzende möchte Laura Ebert die Wiedergabe eines Hyperaudio-Dokuments beenden und am nächsten Tag automatisch an derselben Stelle fortsetzen können, um das Lernen schnell wiederaufnehmen zu können.

### **nicht umgesetzt**

Eine automatische Wiedergabe des Hyperaudio-Dokuments an der Stelle, an der es beendet wurde, ist nicht möglich.

US-18: Als Nutzende möchte Laura Ebert die Hyperaudio-Angebote mit ihrem Smartphone in Anspruch nehmen, um auch die Zeit während des Pendelns zum Lernen nutzen zu können.

### **teilweise umgesetzt**

Durch die angepasste Darstellung auf mobilen Endgeräten kann Laura Ebert auch mit ihrem Smartphone das Hyperaudio-Angebot wahrnehmen. Dabei ergibt sich allerdings die Einschränkung, dass sie vorhandene Lesezeichen nicht löschen kann.

US-19: Als Nutzender möchte Max Lustig eine alte Notiz bearbeiten, um einen Schreibfehler zu korrigieren.

### **umgesetzt**

Durch Betätigen der „Bearbeiten“-Schaltfläche ist Max Lustig in der Lage seine Notiz zu bearbeiten und die Änderung zu speichern.

US-20: Als Nutzender möchte Max Lustig eine alte Notiz löschen, da er inzwischen Lernfortschritte gemacht hat und auf diese Notiz verzichten kann.

### **umgesetzt**

Durch Betätigen der „Löschen“-Schaltfläche kann die Notiz durch Max Lustig gelöscht werden.

US-21: Als Nutzender möchte Max Lustig schnell erkennen welche Inhalte im Hyperaudio-Dokument behandelt werden, um eine Erklärung eines bestimmten Themas zu finden.

### **umgesetzt**

Durch Betrachtung der annotierten Zusatzinhalte, welche gesammelt in der Galerie dargestellt werden, kann sich Max Lustig schnell einen groben Überblick über das Hyperaudio-Dokument verschaffen.

US-22: Als Nutzender möchte Max Lustig nach Textinhalten in Kommentaren suchen können, um schnell Erklärungen zu finden.

### **umgesetzt**

Durch Nutzung des Suchfeldes können sowohl Kommentare als auch Notizen durchsucht werden.

US-23: Als Nutzender möchte Max Lustig die Kommentare nach Erstellungsdatum sortieren können, um sich einen Überblick über die neuesten Aktionen zu verschaffen.

### **umgesetzt**

Wenn Max Lustig im Dropdown-Menü für die Sortierung „Erstelltdatum absteigend“ auswählt, kann er sich einen Überblick über die neuesten Aktionen verschaffen.

US-24: Als Nutzender möchte Max Lustig die Kommentare und persönlichen Notizen zu den Annotationszeitpunkten zuordnen können, um diese bei der Wiedergabe verfolgen zu können.

### **umgesetzt**

Die Kommentare und Notizen in der Timeline der Mediensteuerung visualisiert. Zusätzlich werden

## 6 Evaluation

die Kommentare und Notizen im Kommentarbereich entsprechend sortiert, wenn Max Lustig im Dropdown-Menü für die Sortierung „Annotationszeitpunkt“ auswählt.

**US-25:** Als Nutzender möchte Max Lustig öffentliche Kommentare und persönliche Notizen getrennt betrachten können, um die öffentliche Diskussion verfolgen beziehungsweise die eigenen Anmerkungen isoliert betrachten zu können.

### **umgesetzt**

Mittels zweier Checkboxen kann festgelegt werden, ob Kommentare und/oder Notizen dargestellt werden sollen.

**US-26:** Als Nutzender möchte Max Lustig auf Kommentare antworten können, um sich mit den Studierenden und Lehrenden auszutauschen.

### **umgesetzt**

Nach dem Öffnen einer *Hyperaudio*-Aktivität sind alle an das Hyperaudio-Dokument annotierten Kommentare sichtbar. Diese können nach einem Klick auf „Antworten“ beantwortet werden.

**US-27:** Als Nutzender möchte Max Lustig erkennen, welche Hyperaudio-Dokumente er zuletzt abgespielt hat, um seinen Lernfortschritt im Auge zu behalten.

### **nicht umgesetzt**

Eine Übersicht über die zuletzt abgespielten Hyperaudio-Dokumente ist nicht vorhanden.

**US-28:** Als Nutzender möchte Max Lustig besonders hilfreiche Hyperaudio-Dokumente als Favoriten speichern, um diese schnell als solche identifizieren zu können.

### **nicht umgesetzt**

Das Erstellen von Favoriten ist nicht möglich.

**US-29:** Als Nutzender möchte Max Lustig die Markierung als Favorit entfernen können, wenn der Inhalt für ihn nicht mehr von Interesse ist.

### **nicht umgesetzt**

Da bereits das Erstellen von Favoriten nicht umgesetzt wurde, ist auch das Löschen nicht möglich.

**US-30:** Als Nutzender möchte Max Lustig auf seinem Tablet Zugang zu Hyperaudio-Dokumenten haben, um die Zeit auf dem Laufband gleichzeitig zum Lernen nutzen zu können.

### **teilweise umgesetzt**

Durch die angepasste Darstellung auf mobilen Endgeräten kann Max Lustig auch mit seinem Tablet das Hyperaudio-Angebot wahrnehmen. Dabei ergibt sich allerdings die Einschränkung, dass er vorhandene Lesezeichen nicht löschen kann.

Wie zu erkennen ist, wurde mit 21 User Stories ein Großteil der User Stories umgesetzt. Vier User Stories wurden nicht und fünf nur teilweise bei der Implementierung berücksichtigt.

Nachdem das Hyperaudio-Plugin auf Basis der User Stories bewertet wurde, werden die Ergebnisse nun auf die Anforderungen aus Kapitel 3.2 übertragen und zusammengefasst. Die Resultate sind in Tabelle 6.1 und 6.2 aufgeführt.

Tabelle 6.1: Evaluierung der Anforderungen der Administrierenden

Nr.	Anforderung	Priorität	Erfüllungsgrad
1	Erstellen eines Hyperaudio-Dokuments	hoch	erfüllt
2	Bearbeiten eines Hyperaudio-Dokuments	hoch	erfüllt
3	Löschen eines Hyperaudio-Dokuments	hoch	erfüllt
4	Übernahme eines Hyperaudio-Dokuments in einen anderen Kurs	mittel	teilweise erfüllt
5	Statistische Auswertungen über die Nutzung der Hyperaudio-Dokumente	niedrig	erfüllt

Tabelle 6.2: Evaluierung der Anforderungen der Nutzenden

Nr.	Anforderung	Priorität	Erfüllungsgrad
1	Wiedergabe von Hyperaudio-Dokumenten	hoch	erfüllt
2	Hinweise auf die Darstellung von annotierten Zusatzinhalten	hoch	erfüllt
3	Übersicht über annotierte Zusatzinhalte	mittel	erfüllt
4	Kommentarfunktion bei Hyperaudio-Dokumenten		
4.1	Erstellen von Kommentaren	hoch	erfüllt
4.2	Anzeigen von Kommentaren	hoch	erfüllt
4.3	Antworten auf Kommentare	hoch	erfüllt
4.4	Suchfunktion innerhalb der Kommentare	mittel	erfüllt
5	Notizfunktion bei Hyperaudio-Dokumenten		
5.1	Erstellen von Notizen	hoch	erfüllt
5.2	Anzeigen von Notizen	hoch	erfüllt
5.3	Bearbeiten von Notizen	hoch	erfüllt
5.4	Löschen von Notizen	hoch	erfüllt
6	Lesezeichenfunktion bei Hyperaudio-Dokumenten		
6.1	Erstellen von Lesezeichen	mittel	erfüllt
6.2	Anzeigen von Lesezeichen	mittel	erfüllt
6.3	Löschen von Lesezeichen	mittel	erfüllt
7	Filter- und Sortermöglichkeiten	mittel	erfüllt
8	Favoritenfunktion für Hyperaudio-Dokumente		
8.1	Erstellen von Favoriten	niedrig	nicht erfüllt
8.2	Anzeigen von Favoriten	niedrig	nicht erfüllt
8.3	Löschen von Favoriten	niedrig	nicht erfüllt
9	Übersicht über alle Hyperaudio-Dokumente der belegten Kurse	niedrig	teilweise erfüllt
10	Übersicht über die zuletzt abgespielten Hyperaudio-Dokumente	niedrig	nicht erfüllt
11	Funktion zum Fortsetzen unterbrochener Wiedergaben bei folgenden Aufrufen in Moodle	niedrig	nicht erfüllt
12	Unterstützung von mobilen Endgeräten	mittel	teilweise erfüllt

## 6.2 Evaluation des Hyperaudio-Plugins

Nachdem im vorherigen Abschnitt ermittelt wurde welche Anforderungen erfüllt wurden, wird nun die Evaluation vorgenommen. Für die Rolle der Nutzenden werden UEQ und TAM bei der Evaluation eingesetzt. Anschließen wird ein Ausblick für die Möglichkeiten bei der Evaluation der Rolle der Lehrenden gegeben.

### 6.2.1 Rolle der Nutzenden

Für die Rolle der Nutzenden soll im Folgenden die User Experience (UX) anhand von des User Experience Questionnaire durchgeführt werden.

Die Verwendung von UEQ ermöglicht eine schnelle Evaluation der User Experience interaktiver Produkte und misst dabei nicht nur Aspekte der Usability, wie Effizienz, Verständlichkeit und Steuerbarkeit, sondern auch User Experience Aspekte, wie Stimulation und Originalität (Rauschenberger et al., 2013a). Neben der Möglichkeit zum Vergleich zweier Produkte, kann UEQ auch eingesetzt werden, um zu testen, ob ein Produkt eine ausreichende User Experience bietet (Schrepp, 2018). Bei UEQ wird jeweils ein Fragebogen von Testkandidaten ausgefüllt. „Der UEQ besteht . . . aus 26 bipolaren Items, die sich auf sechs Dimensionen verteilen. Die Items sind als siebenstufige Lickert-Skalen realisiert“ (Rauschenberger et al., 2013b). Im Folgenden sind die sechs Dimensionen aufgelistet:

- Attraktivität
- Durchschaubarkeit
- Effizienz
- Steuerbarkeit
- Stimulation
- Originalität

Die Fragebögen sollten durch den Testkandidaten möglichst direkt nach der Nutzung des Produktes ausgefüllt werden, damit die Testkandidaten nicht durch eine mögliche Diskussion beeinflusst werden (Schrepp, 2018). Nachdem die Fragebögen durch die Testkandidaten ausgefüllt wurden, werden diese ausgewertet. Hierbei wird für jede Dimension ein Wert zwischen -3 und 3 ermittelt. Standardmäßig werden Werte zwischen -0,8 und 0,8 als neutrale, Werte > 0,8 als positive und Werte < -0,8 als negative Evaluation gewertet (Schrepp, 2018). Mit dem UEQ können zwar nicht direkt Punkte abgeleitet werden, welche eine Steigerung der User Experience mit sich bringen, dennoch hilft UEQ dabei eine auf Sachkenntnis gestützte Vermutung über die Bereiche, welche den größten Einfluss haben, zu äußern (Schrepp, 2018).

Nachdem die Grundlage des UEQ erläutert wurden, sollen nun die Ergebnisse des UEQ zum Hyperaudio-Plugin betrachtet und interpretiert werden.

### 6.2.2 Rolle der Lehrenden

Auf die Evaluation für die Rolle der Nutzenden sollte nun auch eine Evaluation für die Rolle der Lehrenden durchgeführt werden. An dieser Stelle soll nun, aber nur eine mögliche Art der Evaluation beschrieben werden. Natürlich könnten hierfür ebenfalls die im vorherigen Abschnitt beschrieben Methode des UEQ durchgeführt werden, es soll nun aber eine weitere Art der Evaluation beschrieben werden. Bei dieser alternativen Methode handelt es sich um das sogenannte Think-Aloud-Protokoll beziehungsweise Think-Aloud-Test. Bei einem Think-Aloud-Test werden die Testkandidaten darum gebeten, das Produkt zu verwenden und währenddessen ihre Gedanken, die bei der Verwendung des Produktes auftreten, laut auszusprechen (Nielsen, 2012).

Sie sollten dann aber darlegen, wie eine Evaluation mit den Lehrenden aussehen sollte (Ablauf, Fragen, Testinstrumente, Vorgehensweise). Viele Studierende wenden in diesem Kontext das Think aloud protocol an.

### 6.3 Verbesserungsvorschläge

Die Verbesserungsvorschläge für das Hyperaudio-Plugin, die bei der Implementierung und Evaluation entstanden sind, sollen in Form von neuen User Stories festgehalten werden. Für diese User Stories werden die gleichen Personas aus Abschnitt 3.1.1 verwendet wie bereits bei den User Stories aus Abschnitt 3.1.2.

Ich finde es gut, dass Sie Vorschläge zur Verbesserung erbringen. Es ist jedoch unklar, warum Sie erst jetzt auf diese User Stories kommen und diese nicht schon vorher formuliert haben. Die Systematik, mit der Sie diese User Stories ermittelt haben, ist außerdem unklar. Hätten Sie eine Art von Evaluation mit potentiellen Nutzern durchgeführt, wäre Ableitungen dieser Art aus den Evaluationsergebnissen nachvollziehbar,

US-31: Als Administrierende möchte Prof. Dr. Karolin Schröder detailliert auf Fehler in der Konfigurationsdatei hingewiesen werden, um solche schneller zu identifizieren, wie beispielsweise Zeitüberschneidungen bei annotierten Zusatzinhalten.

US-32: Als Administrierende möchte Prof. Dr. Karolin Schröder Hyperaudio-Dokumente ganzheitlich innerhalb der Moodle-Umgebung erzeugen können, um nicht mühsam eine Konfigurationsdatei erstellen zu müssen.

US-33: Als Administrierende möchte Prof. Dr. Karolin Schröder mehrere Audio-Dateien für Hyperaudio-Dokumente verwenden können, um die Vertonung von Kurseinheiten in mehreren Abschnitten vornehmen zu können.

US-34: Als Administrierende möchte Prof. Dr. Karolin Schröder auch andere Dateiformate als Bilddateien als Zusatzinhalt annotieren, um auch PDF-Dokumente oder Videos einsetzen zu können.

US-35: Als Administrierende möchte Prof. Dr. Karolin Schröder noch detailliertere Auswertungen über das Nutzungsverhalten von Hyperaudio-Dokumenten erhalten, um noch genauere Rückschlüsse über die Stärken und Schwächen des Hyperaudio-Dokuments zu ermitteln.

US-36: Als Administrierende möchte Prof. Dr. Karolin Schröder verschiedene Audio Cues einsetzen können, um die Studierende auf die verschiedenen Arten von Zusatzinhalten hinzuweisen.

US-37: Als Nutzende möchte Laura Ebert zunächst nur noch die wichtigsten Antworten auf Kommentare angezeigt bekommen, um eine bessere Übersicht bei viel diskutierten Hyperaudio-Dokumenten zu erhalten.

US-38: Als Nutzende möchte Laura Ebert ihrer Notiz eine Datei anfügen, um ihre selbst angefertigte Zeichnung zu hinterlegen.

US-39: Als Nutzender wünscht sich Max Lustig eine intelligenter Suchfunktion, um Ergebnisse trotz Schreibfehler oder mit ähnlichen Wörtern zu finden.

US-40: Als Nutzender möchte Max Lustig nicht nur Zeitpunkte sondern auch Zeitfenster markieren, um Beginn und Ende interessanter Passagen kenntlich zu machen.

US-41: Als Nutzender möchte Max Lustig schnell vom Kommentarbereich zur Anzeige der Zusatzinhalte springen, sobald ein neuer Zusatzinhalt dargestellt wird, um diesen direkt ohne langes Scrollen einsehen zu können.

Diese elf neu formulierten User Stories stellen weiterhin nur einen Teil der möglichen User Stories dar. Sie können dennoch als Basis für die Optimierung des Hyperaudio-Plugins dienen.

## 6.4 Zusammenfassung

Bei der Evaluation wurde die vorliegende Implementierung des Hyperaudio-Plugins analysiert. Die Ergebnisse zeigen auf, dass bereits viele der formulierten User Stories ermöglicht und entsprechend viele Anforderungen erfüllt wurden. Dennoch gibt es sowohl im Bereich der Administration als auch im Bereich der Nutzung noch Verbesserungspotenzial, welches in Form von neuen User Stories formuliert wurde.

Was Sie beispielsweise nicht evaluiert haben:

nicht-funktionale Anforderungen wie Performanz, Lauffähigkeit auf unterschiedlichen Systemen/-Browsern, Usability, User Experience

Können Sie sich drei Leute aus Ihrem persönlichem Umfeld suchen, die einen SUS oder UE Test absolvieren?

UE ist einfach: Rauschenberger, M., Cota, M. P., Thomaschewski, J. (2013). Efficient Measurement of the User Experience of Interactive Products. International Journal of Artificial Intelligence and Interactive Multimedia, 2(1), 39–45. <http://doi.org/10.9781/ijimai.2013.215> Es gibt den Fragebogen auf der Webseite und die Werkzeuge zur Auswertung ebenso.

Außerdem könnten Sie die Technologie Akzeptanz evaluieren: UTAUT, TAM2, TAM ... das sind standardisierte Fragebögen.

Zudem können Sie die Umsetzung der Audio Cues evaluieren und Nutzer schlicht fragen, ob der Ton lauf genug, zur rechten Zeit und eindeutig unterscheidbar erscheint.

Hier sind viele Fragen offen, die in einer wissenschaftlichen Arbeit beantwortet werden sollten. In einer Bachelorarbeit geht es nicht nur darum eine Software zu bauen, sondern auch darum die Anwendung von systematischen Untersuchungsmethoden einzuüben, welche die Nutzbarkeit, Akzeptanz und Zufriedheit der Zielgruppe (zumindest) nahelegt.

Sie werden natürlich keine hochgradig signifikante Tests durchführen können, dennoch sollte der Wille zur Qualitätssicherung und Verbesserung im wissenschaftlichen Sinne erkennbar sein.

# 7 Fazit

Zum Abschluss dieser Arbeit sollen die Ergebnisse nochmals zusammengefasst und der Bezug zu den anfangs gestellten Forschungsfragen hergestellt werden. Abschließend soll ein Ausblick für die Weiterentwicklung des Hyperaudio-Plugins gegeben werden.

## 7.1 Zusammenfassung

Beginnend mit der Formulierung der Grundgedanken für die Repräsentation der Kurseinheiten im Hyperaudio-Format, wurde das Hyperaudio-Plugin nach einer Analyse der Anforderungen und des aktuellen Stands der Technik konzeptioniert und implementiert. Dabei stand die Beantwortung der Forschungsfragen und das Erreichen der daraus abgeleiteten Ziele im Mittelpunkt.

- Wie kann den Studierenden mithilfe einer Hyperaudio-Lernumgebung ermöglicht werden mehr Zeit zum Lernen nutzen zu können?

Es wurde eine zusätzliche Lernumgebung für Studierende geschaffen. Durch die Umsetzung als Moodle-Plugin ist diese nahtlos in die bestehende Infrastruktur der FernUniversität in Hagen integrierbar. Da diese Hyperaudio-Lernumgebung primär auditive Inhalte vermittelt, ist das Lernen in vielen Alltagssituationen möglich. Im Vergleich zur textuellen Repräsentation von Kurseinheiten ist nur selten die visuelle Aufmerksamkeit der Studierenden gefordert. Auf diese Notwendigkeit wird mithilfe akustischer Signale hingewiesen.

- Wie lassen sich auditive Inhalte verständlich gestalten?

In den Grundlagen wurden Wege aufgezeigt, wie die Vertonung von Kurseinheiten durchgeführt werden kann, sodass der Studierende den Lerninhalten erfolgreich folgen kann. Dies kann unter anderem durch den Einsatz der Prinzipien für Multimedia von Mayer (2009) oder unter Berücksichtigung der Erkenntnisse von Donker und Blenn (2007) erfolgen.

- Wie lassen sich Inhalte in der Hyperaudio-Lernumgebung darstellen, die nicht in auditiver Form abgebildet werden können?

Nicht auditiv repräsentierte Inhalte der Kurseinheiten, wie beispielsweise Tabellen oder Grafiken, können in Form von Bildern in die Hyperaudio-Repräsentation eingebunden werden.

- Wie können nicht-auditive Inhalte mit den auditiven Inhalten verknüpft werden?

Das Hyperaudio-Plugin bietet die Möglichkeit, Bilder zu definierten Zeitpunkten darzustellen. Die Zuordnung von Bildern zu Zeitpunkten erfolgt anhand einer Konfigurationsdatei im JSON-Format. Auf die eingebundenen Zusatzinhalte wird nach dem Vorschlag von Donker und Blenn (2007) durch Audio Cues hingewiesen.

- Wie lassen sich alle Interaktionen, die eine textuelle Darstellung der Lerninhalte bietet, in der Hyperaudio-Lernumgebung umsetzen?

In der Hyperaudio-Lernumgebung wurden Möglichkeiten geschaffen, um Interaktionen mit textuellen Kursrepräsentationen digital nachzubilden. So kann das klassische Lesezeichen in der Kurseinheit durch ein Lesezeichen in der Timeline des Hyperaudio-Dokuments ersetzt werden. Soll zudem noch

eine Anmerkung notiert werden, kann statt eines Lesezeichens eine persönliche Notiz angefertigt werden. Die Galerie des Hyperaudio-Dokuments repräsentiert gleichermaßen das Inhaltsverzeichnis einer Kurseinheit sowie die Möglichkeit, durch die Seiten zu blättern.

Sie sind hier in keiner Weise kritisch gegenüber Ihrer eigenen Arbeit. Es ist keinesfalls eine perfekte Lösung, die Sie hier hingelegt haben. Es wird Ihnen auch niemand vorwerfen, Ihr eigenes Werk kritisch zu hinterfragen. Es ist vielmehr eine Frage der Haltung, denn jede Kritik ist der Antrieb für eine Verbesserung und manchmal auch Innovation.

- Wie kann der Austausch zwischen Studierenden und Lehrenden umgesetzt werden?

Die im Hyperaudio-Plugin umgesetzte Kommentarfunktion ermöglicht den Austausch zwischen Studierenden und Lehrenden. Jeder Teilnehmer kann Kommentare zu bestimmten Zeitpunkten eines Hyperaudio-Dokuments verfassen oder auf bestehende Kommentare antworten.

Technisch ja, praktisch aus?

Und? Ist die Darstellung nun gut? Gibt es Daten, die das nahelegen oder sogar bestätigen?

Mit dem Moodle-Hyperaudio-Plugin wurde allen Forschungsfragen begegnet. Dennoch ist die Entwicklung einer Hyperaudio-Lernumgebung noch nicht abgeschlossen.

Warum? Was wurde beforscht?

Hatten Sie sich mit dem Design-Based-Research-Ansatz beschäftigt?

## 7.2 Ausblick

Im Folgenden soll nun ein Ausblick für mögliche Weiterentwicklungen der Hyperaudio-Lernumgebung gegeben werden.

In der Evaluation wurde festgestellt, dass noch nicht alle Anforderungen zu voller Zufriedenheit umgesetzt wurden. Zu diesen Themen (Favoritenfunktion, Übersichten, Unterstützung mobiler Endgeräte, Fortsetzen Unterbrochener Wiedergaben und Import- und Export-Funktion) sind Nachbesserungen erforderlich. Darüber hinaus wurden bereits in Abschnitt 6.3 Verbesserungsvorschläge in Form von User Stories festgehalten.

Im Bezug auf US-37 wäre die Entwicklung einer Bewertungsfunktion für Kommentare und Antworten denkbar, wie sie beispielsweise auf der Plattform *Stack Exchange*<sup>19</sup> zum Einsatz kommt.

Eine weitere Überlegung wäre es das Hyperaudio-Plugin so zu erweitern, dass auch das Streaming auf Fernsehgeräte, zum Beispiel durch *Google Chromecast* oder *Apple TV*, unterstützt wird. Hierdurch würde die Nutzung einer großen Bildschirmfläche abseits von Laptops und Desktops ermöglicht. Aber auch im Bereich der kleinen Displays, also der mobilen Endgeräte, besteht noch Potenzial zur Weiterentwicklung. Es wäre eine App vorstellbar, welche neben einem speziell für mobile Geräte ausgelegten Design auch über eine Offline-Funktionalität verfügt. Damit würden die Freiheiten des Studierenden nochmals erhöht, da er beim Lernen nicht mehr abhängig von einer Internetverbindung ist.

Neben diesen Punkten, die sich vor allem auf die Rolle der Nutzenden beziehen, wären auch einige Weiterentwicklungen für die Administratoren denkbar. So wäre auf lange Sicht eine Funktion von Nöten, um unerwünschte Nutzerkommentare (vgl. Reinmann-Rothmeier und Mandl (2002)) zu löschen.

Für Administratoren wäre ebenfalls eine Möglichkeit zur (halb-)automatischen Konvertierung der klassischen Kurseinheiten in Hyperaudio-Dokumente reizvoll, zum Beispiel durch den Einsatz von ScreenReadern (vgl. Donker und Blenn (2007)). Dies würde die Hürde zur Einführung von Hyper-

<sup>19</sup><https://stackexchange.com>

## 7 Fazit

audio-Dokumenten durch die Lehrenden enorm senken, da dies mit einem wesentlich geringeren Aufwand verbunden wäre.

So hätte man das vor 10 Jahren gemacht. Welche Möglichkeiten gibt es heute noch?

Eine weitere Fragestellung, die sich im Rahmen dieser Arbeit eröffnet hat, ist der Umgang mit Änderungen am Hyperaudio-Dokument. Besonders eine Änderung der zugrundeliegenden Audio-Datei zieht weitreichende Folgen nach sich. So müssen nicht nur Zusatzinhalte und deren Annotationszeitpunkte angepasst, sondern auch bestehende Kommentare, Notizen und Lesezeichen berücksichtigt werden. In diesem Rahmen ist über eine Versionierung beziehungsweise verschiedene Optionen zum Entfernen oder Migrieren der annotierten Inhalte nachzudenken.

Diese Arbeit hat gezeigt, wie eine Hyperaudio-Lernumgebung in Moodle gestaltet werden kann. Das ausbaufähige Hyperaudio-Plugin bietet Raum für zusätzliche Optimierungen für Nutzende als auch Administrierende.

# A Auswertungen

## A.1 Textanteil der Kurseinheiten

Bei der Auswertung des Textanteils der Kurseinheiten werden exemplarisch die Kurseinheiten der Pflichtmodule eines Bachelorstudienganges Wirtschaftsinformatik<sup>20</sup> an der FernUniversität in Hagen analysiert. Im ersten Schritt werden mithilfe des Tools *PDF24 Creator 8.6.0*<sup>21</sup> aus den PDF-Dokumenten alle Leerseiten, das eventuell enthaltene Glossar, alle Verzeichnisse und sonstige nicht zum eigentlichen Kurstext gehörenden Inhalte entfernt. Auch Kurseinheiten, welche selbst nur ein Glossar oder ähnliches darstellen, werden nicht berücksichtigt. Für die weitere Analyse der Kurseinheiten wird das Tool *PDF-Analyzer 5.0*<sup>22</sup> eingesetzt. Es werden die Seiten sowie die Wörter des Dokuments gezählt. Danach wird für jeden Kurs eine Seite mit möglichst großem Textanteil identifiziert und die Anzahl der Wörter auf dieser Seite bestimmt. Mithilfe dieser Kennzahl wird ein durchschnittlicher Textanteil pro Kurs errechnet. Durch das Verwenden eines Vergleichswerts für eine Seite, die somit einen Textanteil von 100% aufweist, sollen Unterschiede in Schriftgröße und verwendeter Seitenfläche ausgeglichen werden. Die Ergebnisse sind den Tabellen A.1 und A.2 zu entnehmen. Tabelle A.1 zeigt die Auswertung der einzelnen Kurseinheiten, während in Tabelle A.2 die zusammenfassende Analyse hinsichtlich des Textanteils für die Kurse dargestellt ist.

Kurs	Semester	KE	Seiten	Wörter	Wörter pro Seite
Algorithmische Mathematik	SS 2015	1	382	71.532	187
Anwendungssysteme und Geschäftsprozessmodellierung	WS 2015/2016	1	87	24.742	284
Anwendungssysteme und Geschäftsprozessmodellierung	WS 2015/2016	2	101	31.463	312
Anwendungssysteme und Geschäftsprozessmodellierung	WS 2015/2016	3	104	26.868	258
Betriebliche Informationssysteme	SS 2016	1	80	19.909	249
Betriebliche Informationssysteme	SS 2016	2	87	20.830	239
Betriebliche Informationssysteme	SS 2016	3	66	15.712	238
Betriebliche Informationssysteme	SS 2016	4	81	18.983	234
Betriebliche Informationssysteme	SS 2016	5	101	26.195	259
Betriebliche Informationssysteme	SS 2016	6	50	12.480	250
Betriebliche Informationssysteme	SS 2016	7	32	7.391	231
Buchhaltung	SS 2015	1	28	7.820	279
Buchhaltung	SS 2015	2	67	16.522	247
Buchhaltung	SS 2015	3	65	17.316	266
Buchhaltung	SS 2015	4	68	17.080	251
Buchhaltung	SS 2015	5	101	26.224	260
Datenbanken I	SS 2016	1	33	9.454	286
Datenbanken I	SS 2016	2	67	17.892	267
Datenbanken I	SS 2016	3	42	9.705	231
Datenmodellierung und Datenbanksysteme	WS 2015/2016	1	39	9.640	247
Datenmodellierung und Datenbanksysteme	WS 2015/2016	2	96	21.962	229
Datenmodellierung und Datenbanksysteme	WS 2015/2016	3	86	19.057	222
Datenmodellierung und Datenbanksysteme	WS 2015/2016	4	65	16.962	261
Einführung in das Marketing	SS 2016	1	194	41.166	212
Einführung in die Betriebswirtschaftslehre	WS 2014/2015	1	119	27.010	227
Einführung in die Betriebswirtschaftslehre	WS 2014/2015	2	102	26.850	263
Einführung in die Betriebswirtschaftslehre	WS 2014/2015	3	66	21.758	330
Einführung in die Betriebswirtschaftslehre	WS 2014/2015	4	79	21.412	271
Einführung in die objektorientierte Programmierung	SS 2015	1	65	12.723	196
Einführung in die objektorientierte Programmierung	SS 2015	2	80	12.079	151
Einführung in die objektorientierte Programmierung	SS 2015	3	73	12.146	166

<sup>20</sup>Hierbei handelt es sich um das Studium des Verfassers dieser Arbeit (WS 2014/2015 bis SS 2018), wobei die Kurseinheiten der Kurse „Grundlagen der Analysis und Linearen Algebra“ und „Grundzüge der Wirtschaftsinformatik“ nicht im PDF-Format vorlagen und dementsprechend bei der Analyse nicht berücksichtigt wurden.

<sup>21</sup><https://de.pdf24.org>

<sup>22</sup><https://www.is-soft.de>

## A Auswertungen

Kurs	Semester	KE	Seiten	Wörter	Wörter pro Seite
Einführung in die objektorientierte Programmierung	SS 2015	4	76	13.359	176
Einführung in die objektorientierte Programmierung	SS 2015	5	49	9.039	184
Einführung in die objektorientierte Programmierung	SS 2015	6	81	13.267	164
Einführung in die objektorientierte Programmierung	SS 2015	7	32	5.326	166
Einführung in die technische und theoretische Informatik	WS 2015/2016	1	85	24.792	292
Einführung in die technische und theoretische Informatik	WS 2015/2016	2	69	23.632	342
Einführung in die technische und theoretische Informatik	WS 2015/2016	3	73	25.157	345
Einführung in die technische und theoretische Informatik	WS 2015/2016	4	104	33.712	324
Einführung in die Volkswirtschaftslehre	WS 2014/2015	1	106	25.320	239
Einführung in die Volkswirtschaftslehre	WS 2014/2015	2	95	19.589	206
Einführung in die Volkswirtschaftslehre	WS 2014/2015	3	53	17.519	331
Finanzierung	WS 2015/2016	1	127	36.003	283
Finanzierung	WS 2015/2016	2	165	43.454	263
Grundbegriffe und Systeme der Kosten- und Leistungsrechnung	SS 2016	1	155	33.884	219
Grundbegriffe und Systeme der Kosten- und Leistungsrechnung	SS 2016	2	120	28.329	236
Grundlagen der Leistungserstellung	SS 2016	1	50	11.358	227
Grundlagen der Leistungserstellung	SS 2016	2	90	15.351	171
Grundlagen der Statistik	WS 2014/2015	1	153	19.647	128
Grundlagen der Statistik	WS 2014/2015	2	151	20.236	134
Grundlagen der Statistik	WS 2014/2015	3	110	15.995	145
Grundzüge der betrieblichen Steuerlehre	SS 2015	1	86	25.705	299
Informationsmanagement	WS 2017/2018	1	92	27.716	301
Informationsmanagement	WS 2017/2018	2	91	28.632	315
Informationsmanagement	WS 2017/2018	3	111	33.975	306
Informationsmanagement	WS 2017/2018	4	101	28.896	286
Informationsmanagement	WS 2017/2018	5	50	15.151	303
Informationsmanagement	WS 2017/2018	6	60	19.835	331
Investition	WS 2015/2016	1	72	18.779	261
Investition	WS 2015/2016	2	94	21.330	227
Investition	WS 2015/2016	3	63	13.407	213
Investition	WS 2015/2016	4	98	19.090	195
Investition	WS 2015/2016	5	99	21.603	218
Jahresabschluss	SS 2015	1	83	14.387	173
Jahresabschluss	SS 2015	2	98	19.007	194
Jahresabschluss	SS 2015	3	142	24.159	170
Jahresabschluss	SS 2015	4	115	26.581	231
Objektorientierte Systemanalyse	WS 2015/2016	1	103	32.447	315
Objektorientierte Systemanalyse	WS 2015/2016	2	185	48.088	260
Sicherheit im Internet I	SS 2016	1	47	16.321	347
Sicherheit im Internet I	SS 2016	2	62	19.917	321
Sicherheit im Internet I	SS 2016	3	54	16.421	304
Sicherheit im Internet I	SS 2016	4	48	16.011	334
Theorie der Marktwirtschaft	WS 2016/2017	1	54	19.168	355
Theorie der Marktwirtschaft	WS 2016/2017	2	214	57.132	267
Theorie der Marktwirtschaft	WS 2016/2017	3	141	31.494	223
Theorie der Marktwirtschaft	WS 2016/2017	4	185	47.198	255
Theorie der Marktwirtschaft	WS 2016/2017	5	69	19.168	278

Tabelle A.1: Auswertung der Kurseinheiten

Zusammenfassend wird ein nach Seitenzahl gewichteter Durchschnitt über den Textanteil gebildet. Dieser liegt bei 63,36%. Die Standardabweichung beträgt 7,51 Prozentpunkte. Getrennt nach Fakultät ergibt sich ein durchschnittlicher Textanteil von 61,08% an der Fakultät für Mathematik und Informatik mit einer Standardabweichung von 10,61 Prozentpunkten und ein durchschnittlicher Textanteil von 64,24% an der Fakultät für Wirtschaftswissenschaften mit einer Standardabweichung von 6,36 Prozentpunkten. Diese Ergebnisse gelten natürlich nur für die exemplarisch gewählten Kurseinheiten.

Es ist anzumerken, dass bei dieser Herangehensweise folgende Punkte die Auswertung verfälschen:

- Nicht jede Fläche, die nicht durch Text in Anspruch genommen wird, enthält automatisch einen anderen visuellen Inhalt. So sollte beispielsweise eine Seite mit wenigen Zeilen Text und einer großen Abbildung mit niedrigem Textanteil in die Auswertung eingehen, während die letzte Seite eines Kapitels, auf der nur noch wenige Zeilen Text stehen, mit einem hohen Textanteil in die Auswertung eingehen sollten.

Modul	Kurs	Wörter auf Seite mit maximalen Inhalt	Seiten	Wörter pro Seite	Textanteil
Algorithmische Mathematik	Fakultät für Mathematik und Informatik	316	382	71532	59,26%
Betriebliche Informationssysteme	Fakultät für Mathematik und Informatik	400	497	121500	61,12%
Einführung in die objektorientierte Programmierung	Fakultät für Mathematik und Informatik	370	456	77939	46,19%
Einführung in die technischen und theoretischen Grundlagen der Informatik	Fakultät für Mathematik und Informatik	409	331	107293	79,25%
Einführung in Internet-Technologien und Informationssysteme	Fakultät für Mathematik und Informatik	462	142	37051	261
Einführung in Internet-Technologien und Informationssysteme	Fakultät für Mathematik und Informatik	458	211	68670	325
Sicherheit im Internet I	Fakultät für Mathematik und Informatik	381	366	97030	265
Einführung in die Betriebswirtschaftslehre	Fakultät für Wirtschaftswissenschaft	398	254	62128	246
Einführung in die Volkswirtschaftslehre	Fakultät für Wirtschaftswissenschaft	374	329	84962	258
Extremes Rechnungswesen	Fakultät für Wirtschaftswissenschaft	445	86	25705	299
Extremes Rechnungswesen	Fakultät für Wirtschaftswissenschaft	301	438	84134	192
Grundlagen der Wirtschaftsmathematik und Statistik	Fakultät für Wirtschaftswissenschaft	239	414	55878	135
Informationsmanagement	Fakultät für Wirtschaftswissenschaft	429	505	154205	305
Internes Rechnungswesen und funktionale Steuerung	Fakultät für Wirtschaftswissenschaft	292	194	41166	212
Internes Rechnungswesen und funktionale Steuerung	Fakultät für Wirtschaftswissenschaft	385	275	62213	226
Internes Rechnungswesen und funktionale Steuerung	Fakultät für Wirtschaftswissenschaft	353	140	26709	191
Investition und Finanzierung	Fakultät für Wirtschaftswissenschaft	405	292	79457	272
Investition und Finanzierung	Fakultät für Wirtschaftswissenschaft	395	426	94209	221
Modellierung von Informationssystemen	Anwendungssysteme und Geschäftsprozessmodellierung	429	292	83073	284
Modellierung von Informationssystemen	Datenmodellierung und Datenbanksysteme	419	286	67621	236
Modellierung von Informationssystemen	Objektorientierte Systemanalyse	399	288	80535	280
Theorie der Marktwirtschaft (Mikroökonomik)	Theorie der Marktwirtschaft	403	663	174160	263

Tabelle A.2: Analyse des Textanteils der Kurse

## A Auswertungen

- Formeln, Tabellen und Code-Beispiele werden ebenfalls als Text interpretiert. Hier wäre im Einzelfall zu entscheiden, ob diese Bestandteile als Text gewertet werden sollen oder nicht.
- Die Formatierung der Inhalte beeinflusst die Auswertung dahingehend, dass beim Einsatz vieler Überschriften, Einrückungen oder Aufzählungen der ermittelte Textanteil geringer ausfällt.
- Die Ergebnisse hängen stark von der gewählten Vergleichsseite ab.

Anhand der durchgeführten Auswertung kann ein Eindruck über die Verteilung zwischen Text und nicht-textuellen Inhalten in Kurseinheiten gewonnen werden. Um verlässliche Zahlen zu generieren, wären intensivere Studien nötig.

## A.2 Einsatz von Moodle

Tabelle A.3 zeigt die Auflistung der Pflichtmodule des Bachelorstudiengangs Wirtschaftsinformatik mit der Information, ob Moodle in den Kursen eingesetzt wird.

Modul	Kurs	Moodle im Einsatz
Algorithmische Mathematik	Fakultät für Mathematik und Informatik	nein
Betriebliche Informationssysteme	Fakultät für Mathematik und Informatik	nein
Einführung in die objektorientierte Programmierung	Einführung in die objektorientierte Programmierung	nein
Einführung in die technischen und theoretischen Grundlagen der Informatik	Betriebssysteme und Rechnernetze für Wirtschaftsinformatiker	nein
Einführung in die technischen und theoretischen Grundlagen der Informatik	Einführung in die technische und theoretische Informatik	nein
Einführung in Internet-Technologien und Informationssysteme	Daten- und Dokumentenmanagement im Internet	nein
Einführung in Internet-Technologien und Informationssysteme	Datenbanken I	nein
Einführung in Internet-Technologien und Informationssysteme	Einführung in wissensbasierte Systeme	nein
Einführung in Internet-Technologien und Informationssysteme	Sicherheit im Internet I	nein
Einführung in die Wirtschaftsinformatik	Einführung in die Wirtschaftsinformatik	ja
Einführung in die Wirtschaftswissenschaft	Einführung in die Betriebswirtschaftslehre	ja
Externes Rechnungswesen	Einführung in die Volkswirtschaftslehre	ja
Externe Rechnungswesen	Buchhaltung	ja
Externe Rechnungswesen	Grundzüge der betrieblichen Steuerlehre	ja
Grundlagen der Wirtschaftsmathematik und Statistik	Jahresabschluss	ja
Informationsmanagement	Grundlagen der Analysis und Linearen Algebra	ja
Internes Rechnungswesen und funktionale Steuerung	Grundlagen der Statistik	ja
Internes Rechnungswesen und funktionale Steuerung	Informationsmanagement	ja
Internes Rechnungswesen und funktionale Steuerung	Einführung in das Marketing	ja
Investition und Finanzierung	Grundbegriffe und Systeme der Kosten- und Leistungsberechnung	ja
Investition und Finanzierung	Grundlagen der Leistungserstellung	ja
Makroökonomik	Finanzierung	ja
Makroökonomik	Investition	ja
Modellierung von Informationssystemen	Makroökonomik I (Dateikurs und Studienbrief)	ja
Modellierung von Informationssystemen	Makroökonomik II (Dateikurs und Studienbrief)	ja
Modellierung von Informationssystemen	Anwendungssysteme und Geschäftsprozessmodellierung	ja
Modellierung von Informationssystemen	Datenmodellierung und Datenbanksysteme	ja
Theorie der Marktökonomik	Grundlagen der Modellierung betrieblicher Informationssysteme	ja
Theorie der Marktökonomik	Objektorientierte Systemanalyse	ja
Theorie der Marktökonomik	Theorie der Marktwirtschaft	ja

Tabelle A.3: Verwendung von Moodle in den Pflichtmodulen des Bachelorstudiengangs Wirtschaftsinformatik (Sommersemester 2018)



# Literaturverzeichnis

- Mike Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- Larry Constantine. Users, Roles, and Personas. *The Persona Lifecycle*, pages 498–519, 2006.
- Larry L. Constantine. Usage-Centered Software Engineering: New Models, Methods and Metrics. In *Proceedings of the 1996 International Conference on Software Engineering: Education and Practice (SE:EP '96)*, SEEP '96, pages 2–, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7379-6. URL <http://dl.acm.org/citation.cfm?id=829500.829892>.
- Alan Cooper et al. *The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]*. Sams Indianapolis, 2004.
- Pierre Dillenbourg. *Collaborative learning: Cognitive and computational approaches. advances in learning and instruction series*. ERIC, 1999.
- Pierre Dillenbourg, Michael J Baker, Agnes Blaye, und Claire O'Malley. The evolution of research on collaborative learning, 1995.
- Hilko Donker und Norbert Blenn. Gestaltung von hyperlinks in einer hyperaudio-encyklopädie. In *Mensch & Computer*, pages 139–148, 2007.
- FernUniversität in Hagen. Das Kursportal – helpdesk.  
[https://wiki.fernuni-hagen.de/helpdesk/index.php/Das\\_Kursportal](https://wiki.fernuni-hagen.de/helpdesk/index.php/Das_Kursportal), 2014. abgerufen am 28.07.2018.
- FernUniversität in Hagen. Kurs: WS16/17 - Kurs 01697 Einführung in Mensch-Computer-Interaktion.  
<https://moodle-wrm.fernuni-hagen.de/course/view.php?id=949>, 2016. abgerufen am 15.08.2018.
- FernUniversität in Hagen. Zusammenarbeiten: Moodle, BSCW, Virtual Classroom, News, Chat - Leistungsangebot - ZMI - FernUniversität in Hagen. [http://www.fernuni-hagen.de/zmi/produkte\\_service/kommunikationstools.shtml](http://www.fernuni-hagen.de/zmi/produkte_service/kommunikationstools.shtml), 2018a. abgerufen am 28.07.2018.
- FernUniversität in Hagen. Moodle – helpdesk.  
<https://wiki.fernuni-hagen.de/helpdesk/index.php/Moodle>, 2018b. abgerufen am 28.07.2018.
- FernUniversität in Hagen. Daten-Zahlen-Fakten - Uni Intern - FernUniversität in Hagen.  
<http://www.fernuni-hagen.de/arbeiten/statistik/daten/index.shtml>, 2018c. abgerufen am 01.06.2018.
- FernUniversität in Hagen. Virtueller Studienplatz – helpdesk.  
[https://wiki.fernuni-hagen.de/helpdesk/index.php/Virtueller\\_Studienplatz](https://wiki.fernuni-hagen.de/helpdesk/index.php/Virtueller_Studienplatz), 2018d. abgerufen am 28.07.2018.
- Google. Videos mit einem Abspann versehen - Computer - Hilfe für YouTube.  
[https://support.google.com/youtube/answer/6388789?visit\\_id=1-636685303009655859-2954709875&p=end\\_screens&hl=de&rd=1](https://support.google.com/youtube/answer/6388789?visit_id=1-636685303009655859-2954709875&p=end_screens&hl=de&rd=1), 2018a. abgerufen am 28.07.2018.
- Google. Infokarten in Videos hinzufügen - Computer - Hilfe für YouTube.  
<https://support.google.com/youtube/answer/6140493?hl=de>, 2018b. abgerufen am 28.07.2018.
- H5P. Interactive Video | H5P. <https://h5p.org/interactive-video>, 2013. abgerufen am 15.08.2018.
- H5P. wavesurfer.js. <https://wavesurfer-js.org/>, o.D. abgerufen am 15.08.2018.
- helllicht medien GmbH. Code - Responsive Webdesign. <http://be-responsive.de/code/>, o.D. abgerufen am 27.08.2018.
- Michael Kerres und Thomas Jechle. Didaktische Konzeption des telelernens. *Information und Lernen mit Multimedia und Internet*, 3:267–281, 2002.
- R.E. Mayer. *Multimedia Learning*. Cambridge University Press, 2009. ISBN 9780521514125. URL <http://dl.acm.org/citation.cfm?id=829500.829892>.

- <https://books.google.de/books?id=5g0AM1CHysgC>.
- Marshall McLuhan. *Laws of the Media. ETC: A Review of General Semantics*, pages 173–179, 1977.
- Moodle. Aufbau einer Moodle-Site – MoodleDocs.  
[https://docs.moodle.org/35/de/Aufbau\\_einer\\_Moodle-Site](https://docs.moodle.org/35/de/Aufbau_einer_Moodle-Site), 2015a. abgerufen am 28.07.2018.
- Moodle. Block settings - MoodleDocs. [https://docs.moodle.org/35/en/Block\\_settings](https://docs.moodle.org/35/en/Block_settings), 2015b. abgerufen am 18.08.2018.
- Moodle. Was ist Moodle – MoodleDocs. [https://docs.moodle.org/35/de/Was\\_ist\\_Moodle](https://docs.moodle.org/35/de/Was_ist_Moodle), 2015c. abgerufen am 28.07.2018.
- Moodle. Activity Modules - MoodleDocs. [https://docs.moodle.org/dev/Activity\\_modules](https://docs.moodle.org/dev/Activity_modules), 2016. abgerufen am 15.08.2018.
- Moodle. Media players - MoodleDocs. [https://docs.moodle.org/dev/Media\\_players](https://docs.moodle.org/dev/Media_players), 2017a. abgerufen am 18.08.2018.
- Moodle. Plugin types - MoodleDocs.  
[https://docs.moodle.org/dev/Plugin\\_types#List\\_of\\_Moodle\\_plugin\\_types](https://docs.moodle.org/dev/Plugin_types#List_of_Moodle_plugin_types), 2017b. abgerufen am 28.07.2018.
- Moodle. About Moodle - MoodleDocs. [https://docs.moodle.org/35/en/About\\_Moodle](https://docs.moodle.org/35/en/About_Moodle), 2018a. abgerufen am 28.07.2018.
- Moodle. Blocks - MoodleDocs. <https://docs.moodle.org/35/en/Blocks>, 2018b. abgerufen am 18.08.2018.
- Moodle. File API - MoodleDocs. [https://docs.moodle.org/dev/File\\_API](https://docs.moodle.org/dev/File_API), 2018c. abgerufen am 15.08.2018.
- Moodle. Grunt - MoodleDocs. <https://docs.moodle.org/dev/Grunt>, 2018d. abgerufen am 08.09.2018.
- Moodle. Installing Moodle - MoodleDocs. [https://docs.moodle.org/35/en/Installing\\_Moodle](https://docs.moodle.org/35/en/Installing_Moodle), 2018e. abgerufen am 18.08.2018.
- Moodle. Moodle.org: Moodle Statistics. <https://moodle.net/stats/>, 2018f. abgerufen am 28.07.2018.
- Daniel C Moos und Elizabeth Marroquin. Multimedia, hypermedia, and hypertext: Motivation considered and reconsidered. *Computers in Human Behavior*, 26(3):265–276, 2010.
- Theodor H Nelson. Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100. ACM, 1965.
- Jakob Nielsen. Thinking aloud: The# 1 usability tool. *Nielsen Norman Group [online]*. January, 16, 2012.
- Jakob Nielsen. *Multimedia, Hypertext und Internet: Grundlagen und Praxis des elektronischen Publizierens*. Springer-Verlag, 2013.
- Donald A. Norman und Stephen W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986. ISBN 0898597811.
- Nurzhan Nurseitov, Michael Paulson, Randall Reynolds, und Clemente Izurieta. Comparison of json and xml data interchange formats: a case study. *Caine*, 9:157–162, 2009.
- Reinhard Oppermann. User-interface design. In *Handbook on information technologies for education and training*, pages 233–248. Springer, 2002.
- Rampl, Hansjörg. ISO 9241:10 Erwartungskonformität.  
<http://www.handbuch-usability.de/erwartungskonformitaet.html>, 2007. abgerufen am 30.07.2018.
- Maria Rauschenberger, Martin Schrepp, Manuel Pérez Cota, Siegfried Olschner, und Jörg Thomaschewski. Efficient measurement of the user experience of interactive products. how to use the user experience questionnaire (ueq). example: spanish language version. *International Journal of Artificial Intelligence and Interactive Multimedia*. 2013; 2 (1): 39-45, 2013a.
- Maria Rauschenberger, Martin Schrepp, und Jörg Thomaschewski. User experience mit fragebögen messen–durchführung und auswertung am beispiel des ueq. *Tagungsband UP13*, 2013b.
- Gabi Reinmann-Rothmeier und Heinz Mandl. Kooperation. lernen im team. *Grundlagen der Weiterbildung*, 6(2): 65–68, 1995.
- Gabi Reinmann-Rothmeier und Heinz Mandl. Analyse und förderung kooperativen lernens in netzbasierten umgebungen. *Zeitschrift für Entwicklungspsychologie und Pädagogische Psychologie*, 34(1):44–57, 2002.

- Maren Schmidt-Kassow, Marie Deusser, Christian Thiel, Sascha Otterbein, Christian Montag, Martin Reuter, Winfried Banzer, und Jochen Kaiser. Physical exercise during encoding improves vocabulary learning in young female adults: a neuroendocrinological study. *PloS one*, 8(5):e64172, 2013.
- Martin Schrepp. *User Experience Questionnaire Handbook*, 2018.
- Sheila Scutter, Ieva Stupans, Tim Sawyer, und Sharron King. How do students use podcasts to support learning? *Australasian journal of educational technology*, 26(2), 2010.
- SoundCloud. Victorious von Panic! At The Disco | Kostenlos hören auf SoundCloud.  
<https://soundcloud.com/panicatthedisco/panic-at-the-disco-victorious>, 2015.  
 abgerufen am 29.07.2018.
- SoundCloud. Info zu SoundCloud auf SoundCloud. <https://soundcloud.com/pages/contact>, o.D.  
 abgerufen am 28.07.2018.
- Statista. Videoplattformen - Reichweite in Deutschland 2016 | Statistik.  
<https://de.statista.com/statistik/daten/studie/209329/umfrage/fuehrende-videoportale-in-deutschland-nach-nutzeranteil/>, 2016. abgerufen am 28.07.2018.
- Statistisches Bundesamt. Wie die Zeit vergeht - Ergebnisse zur Zeitverwendung in Deutschland - 2012/2013.  
[https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2015/zeitverwendung/Pressebroschuere\\_zeitverwendung.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2015/zeitverwendung/Pressebroschuere_zeitverwendung.pdf?__blob=publicationFile), 2015. abgerufen am 27.07.2018.
- Stephan Augsten. Was ist iterative Entwicklung?  
<https://www.dev-insider.de/was-ist-iterative-entwicklung-a-707340/>, 2018.  
 abgerufen am 19.08.2018.
- Mark S Tremblay, Allana G LeBlanc, Michelle E Kho, Travis J Saunders, Richard Larouche, Rachel C Colley, Gary Goldfield, und Sarah Connor Gorber. Systematic review of sedentary behaviour and health indicators in school-aged children and youth. *International Journal of Behavioral Nutrition and Physical Activity*, 8(1):98, 2011.
- I. Wild. *Moodle 3.x Developer's Guide*. Packt Publishing, 2017. ISBN 9781786469540. URL  
<https://books.google.de/books?id=EHg5DwAAQBAJ>.
- Ralf Wiedemann. *Scrum mit User Stories*. Carl Hanser Verlag GmbH Co KG, 2017.
- Youtube. Panic! At The Disco: Victorious [OFFICIAL VIDEO] - YouTube.  
<https://www.youtube.com/watch?v=AUCHk0lxF44>, 2015. abgerufen am 28.07.2018.
- Joerg Zumbach und Christiane Kroeber. Learning with Hyperaudio: Cognitive Load and Knowledge Acquisition in Non-Linear Auditory Instruction. *Avoiding simplicity confronting complexity. Advances in studying and designing (computer-based) powerful learning environments*, pages 359–170, 2006.



# Abbildungsverzeichnis

Schematischer Aufbau einer Moodle Seite . . . . .	14
Kursseite des Kurses „Einführung in Mensch-Computer-Interaktion“ (FernUniversität in Hagen, 2016) . . . . .	15
Benutzerrollen . . . . .	21
Player der Musik- und Audio-Plattform <i>SoundCloud</i> (SoundCloud, 2015) . . . . .	27
Player der Video-Plattform <i>Youtube</i> (Youtube, 2015) . . . . .	27
Anzeige der Infokarten (Youtube, 2015) . . . . .	28
Auszug aus einem <i>Interactive Video</i> (H5P, 2013) . . . . .	29
Der Audio-Player auf der <i>wavesurfer.js</i> -Webseite(H5P, o.D.) . . . . .	30
Zusammenhänge der Komponenten . . . . .	33
ER-Diagramm der Datenbank des Moodle-Plugins . . . . .	35
Bereiche einer Hyperaudio-Aktivität . . . . .	36
Verschiedene Möglichkeiten zur Darstellung von Zusatzinhalten . . . . .	37
Benutzeroberfläche - Mediensteuerung . . . . .	38
Benutzeroberfläche - Galerie . . . . .	40
Benutzeroberfläche - Bereich für Kommentare und Notizen . . . . .	41
Benutzeroberfläche - Zusammengeführtes Layout . . . . .	44
Benutzeroberfläche - Mobile Darstellung . . . . .	45
Ordnerstruktur des Hyperaudio-Plugins . . . . .	47
Architektur des Hyperaudio-Plugins . . . . .	49
Darstellung der Antworten-Funktion . . . . .	58
Darstellung der Notiz-Funktion . . . . .	59
Darstellung der Galerie . . . . .	61
Darstellung der Mediensteuerung inklusive Visualisierung der Kommentare, Notizen und Lesezeichen . . . . .	65
Darstellung der Such-, Filter- und Sortierfunktion . . . . .	66
Darstellung der Bereiche . . . . .	67
Darstellung der Such-, Filter- und Sortierfunktion . . . . .	67
Darstellung der Galerie . . . . .	67
Darstellung der finalen Desktop-Version . . . . .	69
Darstellung der finalen mobilen Version . . . . .	70



# **Tabellenverzeichnis**

Anforderungen von Administrierenden . . . . .	23
Anforderungen der Nutzenden . . . . .	25
Evaluierung der Anforderungen der Administrierenden . . . . .	77
Evaluierung der Anforderungen der Nutzenden . . . . .	78
Auswertung der Kurseinheiten . . . . .	86
Analyse des Textanteils der Kurse . . . . .	87
Verwendung von Moodle in den Pflichtmodulen des Bachelorstudiengangs Wirtschaftsinformatik (Sommersemester 2018) . . . . .	89



# Verzeichnis der Auflistungen

4.1 Beispielhafte Konfigurationsdatei . . . . .	34
5.1 Ausschnitt der <b>mod_form.php</b> in der 1. Iteration . . . . .	49
5.2 Ausschnitt der <b>lib.php</b> in der 1. Iteration . . . . .	50
5.3 Ausschnitt der <b>locallib.php</b> in der 1. Iteration . . . . .	50
5.4 Ausschnitt der <b>view.php</b> in der 1. Iteration . . . . .	51
5.5 Ausschnitt der <b>renderer.php</b> in der 1. Iteration . . . . .	51
5.6 Ausschnitt der <b>locallib.php</b> in der 2. Iteration . . . . .	52
5.7 Ausschnitt der <b>view.php</b> in der 2. Iteration . . . . .	52
5.8 Ausschnitt der <b>renderer.php</b> in der 2. Iteration . . . . .	53
5.9 Ausschnitt der <b>locallib.php</b> in der 3. Iteration . . . . .	54
5.10 Ausschnitt der <b>renderer.php</b> in der 4. Iteration . . . . .	55
5.11 Ausschnitt der <b>services.php</b> in der 4. Iteration . . . . .	55
5.12 Ausschnitt der <b>external.php</b> in der 4. Iteration . . . . .	56
5.13 Ausschnitt der <b>hyperaudio.js</b> in der 4. Iteration . . . . .	57
5.14 Ausschnitt der <b>locallib.js</b> in der 4. Iteration . . . . .	58
5.15 Ausschnitt der <b>hyperaudio.js</b> in der 5. Iteration . . . . .	58
5.16 Ausschnitt der <b>hyperaudio.js</b> in der 6. Iteration . . . . .	59
5.17 Ausschnitt der <b>renderer.php</b> in der 7. Iteration . . . . .	60
5.18 Ausschnitt der <b>popcorn.hyperaudio.js</b> in der 7. Iteration . . . . .	60
5.19 Ausschnitt der <b>hyperaudio.js</b> in der 8. Iteration . . . . .	61
5.20 Ausschnitt der <b>hyperaudio.js</b> in der 9. Iteration . . . . .	62
5.21 Ausschnitt der <b>locallib.js</b> in der 9. Iteration . . . . .	63
5.22 Ausschnitt der <b>hyperaudio.js</b> in der 10. Iteration . . . . .	64
5.23 Ausschnitt der <b>external.php</b> in der 11. Iteration . . . . .	65
5.24 Ausschnitt der <b>styles.css</b> in der 12. Iteration . . . . .	68



# Abkürzungsverzeichnis

<b>AJAX</b>	Asynchronous JavaScript and XML
<b>AMD</b>	Asynchronous Module Definition
<b>API</b>	Application Programming Interface
<b>BAT</b>	BMAT Annotation Tool
<b>CSS</b>	Cascading Style Sheets
<b>DML</b>	Data Manipulation Language
<b>HTML</b>	HyperText Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>LMS</b>	Learningmanagementsystem
<b>LVU</b>	Lernraum Virtuelle Universität
<b>Moodle</b>	Modular Object-Oriented Dynamic Learning Environment
<b>PDF</b>	Portable Document Format
<b>PHP</b>	PHP: Hypertext Preprocessor
<b>SQL</b>	Structured Query Language
<b>TAM</b>	Technology Acceptance Model
<b>UEQ</b>	User Experience Questionnaire
<b>URL</b>	Uniform Resource Locator
<b>UX</b>	User Experience
<b>VU</b>	Virtuelle Universität
<b>XML</b>	Extensible Markup Language



**Eidesstattliche Erklärung**

Ich erkläre hiermit, dass ich diese Bachelorarbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Oberasbach, den .....

Datum

Michael Lämmermann