

Josef Wachtler

Echtzeit-Interaktionsformen für Web-Inhalte

Masterarbeit

Technische Universität Graz

Institut für Informationssysteme und Computer Medien

Leiter: Univ.-Prof. Dipl.-Ing. Dr.techn. Frank Kappe

Betreuer: Univ.-Doz. Dipl.-Ing. Dr.techn. Martin Ebner

Graz, Dezember 2012

Dieses Dokument wurde mit `kate`, dem erweiterten Texteditor des KDE Projekts geschrieben, ist gesetzt in Palatino, kompiliert mit `pdfLATEX2e` und `Biber`.

Die L^AT_EX Vorlage von Karl Voit basiert auf `KOMA Script` und kann hier gefunden werden: <https://github.com/novoid/LaTeX-KOMA-template>

Eidesstattliche Erklärung¹

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am _____

Datum

Unterschrift

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____

Date

Signature

¹Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Kurzfassung

Diese Arbeit dokumentiert die Entwicklung und die Evaluation eines Web-Portals zur Bereitstellung von verschiedenen Interaktionsformen während der Präsentation von beliebigen Echtzeit Web-Inhalten zu festgelegten Zeiten. Da die Interaktion und die Kommunikation als wichtige Bestandteile des E-Learnings angesehen werden, zielt das entwickelte Web-Portal auf eine Verwendung in diesem Bereich, hauptsächlich zur Unterstützung von Vorlesungsübertragungen über das Internet, ab. Somit wird es ermöglicht, dass die bzw. der Vortragende, Fragen an die Teilnehmerinnen und Teilnehmer stellen kann, ähnlich wie mit einem sogenannten Audience Response System. Außerdem ist auch die umgekehrte Richtung der Fragestellung möglich. Dadurch soll die Erhaltung bzw. Erhöhung der Mitarbeit und der Aufmerksamkeit der Teilnehmerinnen und Teilnehmer erreicht werden. Ebenfalls wird eine Berechnung des Aufmerksamkeitsgrads auf Basis der Mitarbeit an den Interaktionsformen, zur Bereitstellung eines Feedbacks, durchgeführt. Die Evaluation zeigt, dass das Potential zur Verbesserung der Aufmerksamkeit durch das Web-Portal gegeben ist, jedoch hängt der erfolgreiche Einsatz stark von der aktiven Unterstützung sowohl durch die Vortragenden als auch von der Mitarbeit durch die Teilnehmerinnen und Teilnehmer ab.

Abstract

This master thesis aims the development and the evaluation of a web-application which offers different synchronous methods of interaction to support any kind of real-time web-content at a fixed time. Due to the fact that communication is seen as an important part of E-Learning the developed web-application addresses mainly the broadcasting of lectures on the web. It provides the possibility to lecturers to ask questions to students and vice versa. So an occupation of the students should be reached to sustain and enlarge the contribution and the attention. In addition an algorithm is shown to gain an attention-level which is based on the contribution to the interaction-methods. The evaluation shows that the web-application has the power to enhance the contribution and the attention. However the key to a successful usage is the active support from both the lecturer and the students.

Danksagung

An dieser Stelle möchte ich mich beim Betreuer meiner Masterarbeit, Herrn Dr. Martin Ebner, bedanken. Ich danke für die Chance diese Arbeit umsetzen zu dürfen, sowie für die professionelle Betreuung und Unterstützung.

Ebenfalls gilt mein Dank allen Angehörigen der Technischen Universität Graz, welche mir, aufgrund meiner körperlichen Einschränkungen, immer unterstützend entgegengekommen sind. Hierbei seien vor allem die Verantwortlichen der verschiedenen Lehrveranstaltungen dankend erwähnt, denn organisatorische Angelegenheiten, wie zum Beispiel das Ersetzen einer schriftlichen Prüfung durch eine Mündliche, konnten immer problemlos an meine Bedürfnisse angepasst werden.

Ein ganz besonderer Dank gebührt meinen Eltern, Ingrid und Josef Wachtler. Ohne ihre tatkräftige Unterstützung in allen Bereichen wäre das Absolvieren des Studiums oder die Erstellung dieser Masterarbeit für mich nicht möglich gewesen.

Schließlich danke ich auch allen Korrekturleserinnen und Korrekturlesern dieser Masterarbeit für ihre sorgfältige und gewissenhafte Arbeit.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziele	1
1.2	Forschungsfrage	3
1.3	Überblick	3
2	Stand der Technik	5
2.1	Allgemeines	5
2.2	Audience Response Systems: Vor- und Nachteile	7
2.3	Ähnliche Projekte	13
2.4	Online-Assessment	15
3	Design und Implementation	17
3.1	Überblick	17
3.2	Das Django-Web-Framework	20
3.2.1	Geschichte	20
3.2.2	Prinzipien und Features	21
3.2.3	MVC vs. MTV Entwurfsmuster	23
3.3	Das Grundgerüst	26
3.3.1	Benutzeroberfläche	28
3.3.2	Internationalisierung und Lokalisierung	31
3.3.3	Benutzermanagement	32
3.3.4	Veranstaltungsverwaltung	34
3.4	Die Interaktionen-Architektur	39
3.4.1	Die Schnittstelle	39
3.4.2	Der Ablauf	46
3.5	Das Analyse-Framework	50
3.5.1	Der Ablauf	51
3.5.2	Die Schnittstelle	54

Inhaltsverzeichnis

3.5.3	Berechnung des Aufmerksamkeitsgrads auf Basis der Reaktionszeiten	55
3.6	Die Interaktionsformen	58
3.6.1	Einfache Fragen	59
3.6.2	Löse Captcha	61
3.6.3	Befrage Teilnehmer	63
3.6.4	Befrage Vortragenden	66
3.6.5	Aufmerksamkeit einstellen	69
3.6.6	Technisches Problem melden	71
3.7	Die Administrationsoberfläche	72
4	Evaluation	77
4.1	Der Erste Test - »Proof of Concept«	77
4.2	Erste Verwendung bei GADI 2012	81
4.3	Regelmäßiger Einsatz bei ESP 2012	83
5	Diskussion	87
5.1	Erfahrungen zum Ablauf	87
5.2	Bemerkungen zu den Aufmerksamkeitsgraden	89
5.3	Einschränkungen	90
5.4	Vor- und Nachteile	92
6	Ausblick	95
6.1	Allgemeine Verbesserungen	95
6.2	Zusätzliche Interaktionstypen und -formen	96
6.3	Weitere Forschungen und Einsatzmöglichkeiten	98
7	Zusammenfassung	101
	Literatur	105

Abbildungsverzeichnis

3.1	Der grundlegende Aufbau	18
3.2	Zusammenspiel der Django-Komponenten	25
3.3	Die Startseite des Web-Portals	28
3.4	Ein neues Profil erstellen	33
3.5	Typischer Ablauf einer Veranstaltung	35
3.6	Eine neue Veranstaltung erstellen	36
3.7	Aufmerksamkeit nach der Veranstaltung	38
3.8	Detailliertere Aufmerksamkeitsanalyse	38
3.9	Ablauf aus Sicht der/des Vortragenden	46
3.10	Ablauf aus Teilnehmersicht	47
3.11	Screenshot während einer Veranstaltung bei der bzw. dem Vortragenden	48
3.12	Screenshot während einer Veranstaltung bei einer Teilnehmerin bzw. einem Teilnehmer	49
3.13	Verlauf des Aufmerksamkeitsgrads auf Basis der Reaktionszeit	56
3.14	Eine einfache automatische Frage	59
3.15	Lösen eines einfachen Captchas	62
3.16	Eine Frage an alle Teilnehmerinnen und Teilnehmer	64
3.17	Die Teilnehmerinnen und Teilnehmer beantworten die Frage	64
3.18	Die bzw. der Vortragende erhält die Antworten	65
3.19	Die bzw. der Vortragende gibt eine Referenzantwort	66
3.20	Eine Frage an die Vortragende bzw. den Vortragenden richten	67
3.21	Anzeige der Frage bei allen Teilnehmerinnen und Teilnehmern	68
3.22	Die bzw. der Vortragende kann die Frage beantworten	68
3.23	Die Antwort wird bei allen Teilnehmerinnen und Teilnehmern dargestellt	68
3.24	Selbstständiges einstellen des Aufmerksamkeitsgrads	69
3.25	Melden eines technischen Problems	71

Abbildungsverzeichnis

3.26	Bearbeiten eines Benutzerprofils	73
3.27	Hinzufügen einer Interaktionsform	73
3.28	Such- und Filterfunktionen der Veranstaltungen	74
3.29	Bearbeiten einer Veranstaltung	75
4.1	Aufmerksamkeitsgrade der Teilnehmerinnen und Teilnehmer	82
4.2	Vergleich der Anzahl der gesamten Teilnehmerinnen bzw. Teilnehmer mit der Anzahl der Aktiven	85
5.1	Der Vollbildmodus verdeckt das Web-Portal	91

Tabellenverzeichnis

2.1	Ergebnisse einer Grundlagenstudie über ARS	8
2.2	Ergebnisse einer ausführlichen ARS-Studie	9
3.1	Einstellungen einer Interaktionsform	41
3.2	Einstellungen der Analyse einer Interaktionsform	57
4.1	Spezifikationen der Testnotebooks	78

1 Einleitung

Kommunikation nennen wir ein Beziehungsgeschehen (Interaktion) zwischen Menschen, das auf Verständnis abzielt.

(Urheber unbekannt¹)

Viele Web-Inhalte wurden bzw. werden erstellt, um ihren Konsumentinnen und Konsumenten Information bereitzustellen. In den meisten Fällen wird der Inhalt gut aufbereitet dargestellt, jedoch gibt es meist wenig Möglichkeiten zur Interaktion und Kommunikation mit den Autorinnen bzw. Autoren. Das Verständnis und vor allem die Aufmerksamkeit könnte aber durch den Einsatz von Interaktions- und Kommunikationsmöglichkeiten, besonders durch sogenannte Audience Response Systeme, kurz ARS, (vgl. z.B.: Latessa und Mouw [LM05]), gesteigert werden.

Dieses Problem wurde durch den in letzter Zeit stark steigenden Einsatz von verschiedenen sozialen Netzwerken verbessert. Solche Technologien sind aber für die Verwendung im Echtzeit-Bereich, vor allem zur Interaktion zwischen den Studentinnen bzw. Studenten und der bzw. dem Vortragenden während Live-Übertragungen von Vorlesungen an Universitäten, kaum geeignet. Außerdem sind die damit verbundenen rechtlichen Bedenken bzw. Probleme zu beachten.

1.1 Ziele

Die einleitend erwähnte Problematik soll im Rahmen dieser Arbeit durch den Entwurf und die Entwicklung eines Web-Portals zur Echtzeit-Interak-

¹ <http://www.zitate.de/db/ergebnisse.php?sz=2&stichwort=&kategorie=Kommunikation&autor=>, besucht am 04.10.2012

1 Einleitung

tion und -Kommunikation verbessert werden. Als primäres Einsatzgebiet des Portals sei die Unterstützung von Vorlesungsübertragungen an Universitäten genannt. Grundsätzlich sollte es aber auch für jede andere Art von Web-Inhalten geeignet sein.

Anforderungen
des Grundsystems

Das Grundsystem dieses Web-Portals soll folgende Funktionalitäten bereitstellen:

- Benutzerverwaltung: Nur registrierte und angemeldete Benutzerinnen und Benutzer sollen das Web-Portal verwenden können.
- Veranstaltungsverwaltung: Die Echtzeit-Interaktionen sollen nur zu bestimmten Zeiten im Rahmen einer Veranstaltung möglich sein (z.B.: während der Übertragung einer Vorlesung).
- Eine einfache Schnittstelle soll die Integration von verschiedenen Interaktionsformen ermöglichen.
- Während einer Veranstaltung soll den Teilnehmerinnen und Teilnehmern der eigentliche Inhalt (z.B. die Vorlesungsübertragung) dargestellt werden. Zusätzlich sollen aber auch Interaktionsformen bereitgestellt werden, ohne den Inhalt wesentlich zu stören.
- Der bzw. dem Vortragenden stehen während der Veranstaltung ebenfalls Interaktionsformen bereit und außerdem soll ein aktueller Aufmerksamkeitsgrad zur Darstellung kommen.
- Nach der Veranstaltung soll der bzw. dem Vortragenden eine Auflistung aller Teilnehmerinnen und Teilnehmer mit deren jeweiligen Aufmerksamkeitsgrad angeboten werden, ebenso den Teilnehmerinnen und Teilnehmern.

Formen der
Interaktion

Folgende Interaktionsformen sollen im Rahmen dieser Arbeit entwickelt und über die oben erwähnte Schnittstelle ins Grundsystem eingebunden werden:

- Die Teilnehmerinnen und Teilnehmer sollen eine Frage an die Vortragende bzw. den Vortragenden stellen können.
- Umgekehrt soll auch die bzw. der Vortragende eine Frage an alle Teilnehmerinnen und Teilnehmer richten können.

1.2 Forschungsfrage

- In beiden Fällen soll es natürlich möglich sein, die Fragen auch zu beantworten.
- Den Teilnehmerinnen und Teilnehmern soll auch eine Möglichkeit angeboten werden, selbstständig ihren Aufmerksamkeitsgrad einzustellen.
- Kleinere Aufgaben (z.B.: Lösen eines Captchas) sollen automatisch den Teilnehmerinnen und Teilnehmern gestellt werden, um ihre Aufmerksamkeit aufrecht zu erhalten bzw. zu erhöhen.
- Die Teilnehmerinnen und Teilnehmer sollen in der Lage sein, ein technisches Problem (z.B.: Probleme mit der Vorlesungsübertragung) zu melden.

Neben diesen funktionellen Zielen ist es aus technischer Sicht auch erforderlich, dass das Web-Portal auf der Seite der Benutzerinnen und Benutzer mit einem aktuellen Web-Browser ohne zusätzliche Software bedient werden kann.

1.2 Forschungsfrage

Entwicklung eines Web-Portals zur Erhaltung bzw. Erhöhung der Aufmerksamkeit und zur Feststellung des Aufmerksamkeitsgrads der Teilnehmerinnen und Teilnehmer durch verschiedene Echtzeit-Interaktionsformen.

1.3 Überblick

Die vorliegende Arbeit beginnt nun nach diesen einleitenden Betrachtungen mit einem Überblick zum Stand der Technik (siehe Kapitel 2) auf diesem Gebiet und gibt dabei ebenfalls einen Einblick in ähnliche und bestehende Projekte.

1 Einleitung

Danach wird im Kapitel 3 die Entwicklung des Web-Portals dokumentiert. Neben einer detaillierten Beschreibung der Funktionalitäten und ihrer Umsetzung, wird an wichtigen Stellen auch ein Einblick in die Implementation gegeben.

Weiters beschreibt Kapitel 4 die Evaluation des Web-Portals anhand der bisherigen Einsätze (z.B. bei der Vorlesung »Einführung in die strukturierte Programmierung«). Die Ergebnisse der Evaluation werden im Kapitel 5 diskutiert und es werden aus den daraus gewonnenen Erkenntnissen die Vor- und Nachteile des Web-Portals abgeleitet.

Abschließend gibt Kapitel 6 einen Ausblick auf zukünftige Möglichkeiten der Verwendung und beschreibt auch einige Verbesserungs- und Erweiterungsvorschläge, bevor Kapitel 7 alle wichtigen Punkte dieser Arbeit zusammenfasst.

2 Stand der Technik

Nach einigen allgemeinen Betrachtungen und Fragestellungen (siehe Abschnitt 2.1) zum Thema Interaktion und Kommunikation im Bereich E-Learning, welche sich häufig in der Literatur finden, behandelt Abschnitt 2.2 das Gebiet der Audience Response Systeme. Darauf aufbauend wird danach auf Projekte mit ähnlichen Zielen und Inhalten eingegangen (siehe Abschnitt 2.3). Schließlich wird im Abschnitt 2.4 auch noch der Bereich des Online-Assessment betrachtet.

2.1 Allgemeines

Als wesentlichen Bestandteil eines Online-Kurses nennen Carr-Chellman und Duchastel [CDoo] das Bestehen der Möglichkeit zur Kommunikation und Interaktion im Rahmen eines solchen Kurses. Es wird zum Einen unterschieden in synchrone und asynchrone Kommunikation und zum Anderen ist auch eine Unterscheidung nach den kommunizierenden Gruppen möglich.

Bei der asynchronen Kommunikation handelt es sich vor allem um Online-Diskussionsforen. Der Vorteil dabei liegt neben dem zeitunabhängigen Einsatz auch in der Tatsache, dass sich in solchen Foren leicht Lerngruppen (»learning communities«) bilden, in denen oftmals ehemalige und aktuelle Studentinnen und Studenten über das Forum kommunizieren und somit die Basis für einen guten Lernerfolg erzielt werden kann. [CDoo]

Im Vergleich dazu steht im synchronen Fall die Echtzeit-Kommunikation im Vordergrund (z.B.: Internet-Chat). Der Vorteil liegt in diesem Bereich

2 Stand der Technik

neben der direkteren und kollegialeren Interaktion auch in der Möglichkeit zur unmittelbaren und schnellen Reaktion auf Fragen und Aufgaben. [CDoo]

Die Unterscheidung nach den kommunizierenden Gruppen bringt folgende Klassifikation mit sich: [CDoo]

- Kommunikation der Studentinnen und Studenten untereinander
- Beschäftigung der Studentinnen und Studenten mit dem Inhalt
- Kommunikation zwischen den Studentinnen bzw. Studenten und der bzw. dem Vortragenden

Wie aus den in Abschnitt 1.1 genannten Zielen entnommen werden kann, adressiert die vorliegende Arbeit vor allem den Bereich der synchronen Kommunikation der Studentinnen bzw. Studenten mit der bzw. dem Vortragenden.

Diese Art der Kommunikation sollte laut Helmerich und Scherer [HS07] von der bzw. dem Vortragenden dazu genutzt werden, folgende Fragestellungen für sich zu beantworten:

- Ist es den Studentinnen und Studenten möglich, den Ausführungen Folge zu leisten?
- Wird die Vorlesung in einem angemessenen Tempo abgehalten?
- Haben die Studentinnen und Studenten ausreichend Vorwissen?
- War es den Teilnehmerinnen und Teilnehmern möglich, die Erklärungen inhaltlich zu verstehen?
- Ist die Aufbereitung der Lerninhalte zielgruppengerecht?
- Sind die Lerninhalte für die Zielgruppe geeignet?

Auf der Seite der Studentinnen und Studenten soll mit dieser Form der Interaktion dem sogenannten »Berieselungseffekt«, welcher Helmerich und Scherer [HS07] zufolge vor allem durch den vermehrten Einsatz von Multimedia in der Vorlesung entsteht, entgegengewirkt werden. Die bzw. der

2.2 Audience Response Systems: Vor- und Nachteile

Vortragende kann mit Verständnisfragen oder mit anderen, ähnlichen Aufgaben dazu beitragen, diesen Effekt zu minimieren und somit versuchen, die Aufmerksamkeit und die Motivation, beides wesentliche Parameter erfolgreichen Lernens, zu steigern [Holo0].

Mit dieser Arbeit wird versucht, diese Art der Kommunikation, welche in Präsenzvorlesungen erfolgreich angewandt wird, auch bei Liveübertragungen von Vorlesungen ins Internet bereitzustellen (siehe Abschnitt 1.1). Dadurch soll es auch hier ermöglicht werden, die Aufmerksamkeit über die gesamte Vorlesung aufrecht zu erhalten und so auch ein Beitrag zum besseren Verständnis geleistet werden. Dies ist wichtig, denn die Behaltensleistung hängt davon ab, wie stark sich die bzw. der Lernende mit dem Material beschäftigt oder wie sehr es den Vortragenden gelingt Aufmerksamkeit bei den Teilnehmerinnen und Teilnehmern zu erzeugen [Ebn10; Jam90].

2.2 Audience Response Systems: Vor- und Nachteile

Ein Audience Response System (kurz ARS) [Tob05] ist ein Soft- und Hardwaresystem, welches es ermöglicht, im Rahmen eines Vortrags ARS Definition

- Fragen oder andere kleine Aufgaben an die Zuhörerinnen und Zuhörer zu richten,
- welche diese mittels eines passenden Endgeräts beantworten können und
- die Antworten entsprechend aufbereitet darzustellen.

Üblicherweise finden ARS im Rahmen von Präsenzvorlesungen an Universitäten Verwendung. Diese Arbeit versucht, wie in Abschnitt 1.1 erwähnt, die Methodik von ARS auch bei Online-Übertragungen von Vorlesungen nutzbar zu machen. Dadurch sollen die positiven Auswirkungen auf die Aufmerksamkeit, welche von den folgenden Studien bescheinigt werden, auch in diesen Bereich zur Geltung kommen.

2 Stand der Technik

	Hoch	Normal	Niedrig	Null
Zu welchem Grad machte diese durch ARS unterstützten Vorlesung mehr Spaß als eine klassische Vorlesung?	84 %	16 %	0 %	0 %
Wie hoch war der Grad Ihrer Aufmerksamkeit in dieser durch ARS unterstützten Vorlesung, im Vergleich zu traditionellen Vorlesungen?	67 %	33 %	0 %	0 %
Zu welchem Grad hat das ARS geholfen mehr zu lernen als bei klassischen Vorlesungen?	22 %	63 %	15 %	0 %
Ohne auf finanzielle Überlegungen Rücksicht zu nehmen: Wie hoch ist die Wahrscheinlichkeit, dass Sie ein ARS einsetzen werden?	44 %	33 %	15 %	8 %

Tabelle 2.1: Diese Fragen wurden von Latessa und Mouw [LM05] an die Teilnehmerinnen und Teilnehmer an einer durch ARS unterstützten Vorlesung, im Rahmen einer Grundlagenstudie, gestellt. Die Ergebnisse zeigen, dass ein ARS zur Erhöhung der Aufmerksamkeit und auch des Lernerfolgs beiträgt.

»Use of an audience response system to augment interactive learning.« von Latessa und Mouw [LM05]

Eine grundlegende Studie von Latessa und Mouw [LM05] hatte zum Ziel festzustellen, ob ein ARS in der Lage ist, einen Beitrag zu einer besseren Lernerfahrung zu leisten. Dazu wurde eine Gruppe von 46 Teilnehmerinnen und Teilnehmer im Rahmen einer Vorlesung mit Fragen durch ein ARS konfrontiert.

Diese Befragung fand nach einer kurzen Einleitung statt und wurde gegen Ende der Veranstaltung mit den gleichen Fragen wiederholt. Nach jeder Fragerunde wurden die Ergebnisse in anonymisierter und zusammenfassender Form den Teilnehmerinnen und Teilnehmern präsentiert. [LM05]

Schlussendlich wurden die Teilnehmerinnen und Teilnehmer gebeten, den Einsatz von ARS anhand von vier Fragen, welche sich auf den Vergleich zu klassischen Vorlesungen beziehen, zu bewerten. Die Fragen und Antwort-

2.2 Audience Response Systems: Vor- und Nachteile

Gruppe	% Formelle Teilnahme	% Richtig: Fragen	% Richtig: Quiz
Klassische Vorlesung	-	-	57 %
Handzeichen	76 %	98 %	60 %
Antwortkarten	97 %	92 %	52 %
ARS	100 %	82 %	60 %

Tabelle 2.2: Diese Ergebnisse einer ARS-Studie von Stowell und Nelson [SN07] zeigen, dass verschiedene Interaktionsformen einen Beitrag zur Aufmerksamkeitssteigerung leisten

ten können aus der Tabelle 2.1 entnommen werden. Es ist ein deutlicher Trend zu einem stark erhöhten Aufmerksamkeitsgrad und Unterhaltungswert bei den Teilnehmerinnen und Teilnehmern erkennbar. Weiters hatten die Teilnehmerinnen und Teilnehmer das Gefühl mehr zu lernen als bei klassischen Vorlesungen. [LM05]

Eine weitere Studie zum Thema ARS wurde von Stowell und Nelson [SN07] durchgeführt. Dabei wurden die Auswirkungen des Einsatzes von ARS in Vorlesungen auf die Mitarbeit, den Lernerfolg und die Emotionen der Teilnehmerinnen und Teilnehmer untersucht.

Es wurde eine 30-minütige Psychologie-Vorlesung viermal abgehalten. Einmal als klassische Präsenzvorlesung, ein anderes Mal wurden dabei Fragen gestellt, welche per Handzeichen beantwortet werden konnten. Ein weiteres Mal bestand die Möglichkeit, Fragen mit je vier möglichen Antworten durch Hochhalten der passenden Antwortkarte zu beantworten. Schließlich wurde auch noch ein ARS, wo die Antwort mittels eines elektronischen Endgeräts gegeben wurde, eingesetzt. Um den Lernerfolg zu messen, mussten die Teilnehmerinnen und Teilnehmer auch ein Quiz mit zehn Fragen einige Zeit nach der Vorlesung absolvieren. [SN07]

In Tabelle 2.2 ist ersichtlich, dass die Teilnahme an den Fragen während der Vorlesung bei der Gruppe, welche diese per Handzeichen beantwortete, um Einiges geringer ist, als bei der Gruppe mit Antwortkarten oder der ARS Gruppe. Dies mag daran liegen, dass der Grad der Anonymität beim Antworten per Handzeichen geringer ist. Die Anzahl der richtigen

»Benefits of Electronic Audience Response Systems on Student Participation, Learning, and Emotion« von Stowell und Nelson [SN07]

2 Stand der Technik

Antworten auf diese Fragen ist generell sehr hoch, nur bei der ARS Gruppe war sie etwas niedriger (82%). [SN07]

Bei den richtigen Antworten auf die Fragen des Quiz sind keine gravierenden Unterschiede zwischen den vier Gruppen zu erkennen. Es fällt jedoch auf, dass die Anzahl der richtigen Antworten beim Quiz in der ARS Gruppe (60%) am ehesten mit der Anzahl der richtigen Antworten bei den Fragen während der Vorlesung (82%) übereinstimmt. Dies bedeutet, dass der Einsatz eines ARS, im Vergleich zu den anderen Methoden die besten Prognosen über den Lernerfolg der Teilnehmerinnen und Teilnehmer liefert. [SN07]

Um die Emotionen zu messen, wurde der sogenannte »Academic Emotions Questionnaire« (AEQ, [Pek+02]) verwendet. Dabei wurde zum Beispiel der Grad der Freude, der Hoffnung, des Ärgernisses oder der Langeweile gemessen und zwar vor, während und nach der Vorlesung. Die Auswertung dieses Fragebogens ergab keine großen Unterschiede der Emotionen in den verschiedenen Gruppen, jedoch ist eine gewisse Erhöhung der positiven Emotionen während der Vorlesung, welche durch ARS unterstützt wurde, erkennbar. [SN07]

»Observations on Benefits / Limitations of an Audience Response System« von Silliman und McWilliams [SM04]

Eine weitere, sehr ausführliche Studie zum Thema ARS wurde von Silliman und McWilliams [SM04] durchgeführt. Dabei wurden die Stärken und Schwächen eines ARS analysiert und es wurden auch die Meinungen der teilnehmenden Studentinnen und Studenten ausgewertet. Um ein möglichst gutes Ergebnis zu erzielen, wurde der Einsatz eines ARS in vier verschiedenen Situationen getestet: [SM04]

1. In einem sehr großen Anfängerkurs für Ingenieurinnen und Ingenieure,
2. bei einer Vorlesung für erfahrene Technikerinnen und Techniker,
3. um die Lernziele und Lernerfolge einer großen Gruppe von Studentinnen und Studenten zu erfragen und zu bewerten und
4. in einer Schulklasse.

Für die Studie wurden die Kursevaluationen der Studentinnen und Studenten und deren Grad der Zufriedenheit zur Bewertung des ARS-Einsatzes

2.2 Audience Response Systems: Vor- und Nachteile

herangezogen. Ebenso wurde eine Umfrage zum Thema ARS durchgeführt und für die Bewertung wurde auch der Lernerfolg der Studentinnen und Studenten berücksichtigt. [SMo4]

Die Studie brachte einige Vor- und Nachteile zu Tage. Die wichtigsten Vorteile sind: [SMo4]

- Lange und intensive Vorlesungen gestalten sich interessanter.
- Die Mitarbeit und auch die Aufmerksamkeit wird deutlich gesteigert.
- Hilfreiches Feedback sowohl für die Studentinnen und Studenten als auch für die Vortragende bzw. den Vortragenden.
- Umfangreiche Möglichkeiten zur Auswertung und Analyse der erzielten Ergebnisse.

Unter anderem wurden auch folgende Nachteile identifiziert: [SMo4]

- Die Vorlesung wird durch ARS unnötig verlangsamt.
- Der Zwang, schnell eine Antwort auf die mit ARS gestellten Fragen zu geben, wird oftmals als störend empfunden.
- Es besteht eine größere Gefahr von technologischen Problemen.
- Schwierig und zeitaufwändig gestaltet sich unter Umständen die Verteilung der Endgeräte.

Zusammenfassend kommen Silliman und McWilliams [SMo4] zum Schluss, dass ein ARS die Lernumgebung stark verbessern kann und auch in der Lage ist, den Lernerfolg der Studentinnen und Studenten positiv zu beeinflussen. Es sollte jedoch beachtet werden, dass das ARS nicht übermäßig eingesetzt wird.

In einer anderen Studie, welche von Cutrim [Cuto8] durchgeführt wurde, sind die pädagogischen Möglichkeiten eines ARS-ähnlichen Abstimmungssystems, in Verbindung mit einem elektronischen Whiteboard, analysiert worden. Diese Studie basiert auf Daten, welche im Rahmen eines Vorbereitungslehrgangs für internationale Studentinnen und Studenten an einer Britischen Universität gewonnen wurden. Dabei kamen unter anderem folgende Instrumente der Forschung zum Einsatz: [Cuto8]

»Using a voting system in conjunction with interactive whiteboard technology to enhance learning in the English language classroom« von Cutrim [Cut08]

2 Stand der Technik

- Beobachtungen des Klassenraums
- Interviews von Studentinnen und Studenten
- Fragebögen vor und nach dem Vortrag
- Aufzeichnungen und Feedback der bzw. des Vortragenden

Die Ergebnisse der Studie zeigen, dass die Mitarbeit der Teilnehmerinnen und Teilnehmer erhöht wurde und eine gute Möglichkeit zur Erhaltung von Feedback gegeben ist. Dies wird vor allem dadurch ermöglicht, dass die Mitarbeit durch ein ARS einen hohen Grad an Anonymität bietet und auch ein gewisser Unterhaltungswert gegeben ist. Trotz allem ist der Grad der Interaktivität noch immer gering. Dies könnte jedoch durch einfache Ansätze, wie zum Beispiel der Möglichkeit zur Diskussion vor und nach einer Fragerunde, verbessert werden. [Cuto8]

»Interaktion
zwischen
Lehrenden und
Lernenden in
Medien
unterstützten
Veranstaltungen«
von Helmerich
und Scherer
[HS07]

Erfahrungen zum Thema ARS wurden auch von Helmerich und Scherer [HS07] gemacht. Dabei wurde der Einsatz eines eigens entwickelten ARS (siehe Abschnitt 2.3), welches sich im produktiven Betrieb bei einer Vorlesung mit 60 bis 70 Teilnehmerinnen und Teilnehmer befindet, analysiert. Es wurden folgende Beobachtungen gemacht und Erkenntnisse abgeleitet: [HS07]

- Der Großteil der teilnehmenden Studentinnen und Studenten erwarteten sich vor Beginn der Vorlesung eine positive Auswirkung des ARS auf die Qualität der Vorlesung.
- Die Anzahl der Studentinnen und Studenten, welche auf Fragen die über das ARS gestellt wurden antworteten, lag konstant über 50 %.
- Die Interaktion ist im Vergleich zu der herkömmlichen Fragebeantwortung per Handzeichen somit deutlich höher.
- Die bzw. der Vortragende erhält durch das ARS umgehend ein qualitativ hochwertiges Feedback und ist dadurch in der Lage, Defizite bei den Studentinnen und Studenten rascher zu erkennen, um darauf reagieren zu können.
- Es wird dem in Abschnitt 2.1 erwähnten Berieselungseffekt entgegengewirkt.

2.3 Ähnliche Projekte

Die im vorherigen Abschnitt dargestellten Studien wurden unter anderem mit den nachfolgend präsentierten Audience Response Systemen durchgeführt.

Von Helmerich und Scherer [HS07] wurde ein System namens Ping-Pong vorgestellt, welches es ermöglicht, im Rahmen von Vorlesungen Fragen an die Studentinnen und Studenten zu richten. Der Unterschied zu den meisten ARS liegt darin, dass anstatt eines separaten Endgeräts zur Beantwortung der Fragen, die Notebooks der Teilnehmerinnen und Teilnehmer genutzt werden.

Dieses System basiert auf einem sogenannten LAMP¹ Server, welcher die komplette Programmlogik zur Verfügung stellt. Zur Bedienung stehen neben Clients für verschiedene Plattformen auch Zugänge über ein Web-Interface zur Verfügung. Der Einsatz von Ping-Pong kann in drei Phasen gegliedert werden. [HS07]

In der ersten Phase werden zur Vorbereitung von den Vortragenden relevante Fragen mit den dazugehörigen Antworten in einem sogenannten Fragenpool gesammelt. Daraus werden nun Fragen, welche zur Vorlesung passen, ausgewählt und in entsprechender Reihenfolge zu sogenannten Collections zusammengefasst. Ein wichtiger Punkt dabei ist die Wiederverwendbarkeit der Fragen. [HS07]

Die zweite Phase erstreckt sich auf die Vorlesung selbst. Dabei kann die bzw. der Vortragende eine Frage aus der gewählten Collection auswählen und diese wird dann durch Ping-Pong den Teilnehmerinnen und Teilnehmern auf ihren Notebooks zur Beantwortung angeboten. Die bzw. der Vortragende erhält nach der Fragerunde einen Überblick über die Antworten, um diese mit den Studentinnen und Studenten zu diskutieren. Ebenfalls können die Ergebnisse in die inhaltliche Gestaltung der restlichen Vorlesung einbezogen werden. [HS07]

¹kurz für »Linux, Apache, MySQL und PHP«

2 Stand der Technik

In der letzten Phase steht die Nachbearbeitung im Vordergrund. Ping-Pong bietet der bzw. dem Vortragenden eine Vielzahl von Statistiken und Analysen an. Die daraus gewonnenen Erkenntnisse können somit zur Verbesserung des Vorlesungsaufbaus und zur Anpassung der Lerninhalte genutzt werden. [HS07]

Im Abschnitt 2.2 wurden die Erkenntnisse und Erfahrungen, welche durch den Einsatz von Ping-Pong gesammelt wurden, bereits näher erläutert.

Ein weiteres ARS mit dem Namen »RealFeedback«² wurde von Haintz und Pichler [HP12] an der Technischen Universität Graz entwickelt. Mit diesem können Fragen in einer einfachen Art und Weise an die Teilnehmerinnen und Teilnehmer gestellt werden. Die Bedienung dieses Systems erfolgt ausschließlich über eine Web-Oberfläche und somit kommen zur Beantwortung der Fragen zum Beispiel die Smartphones oder Notebooks der Teilnehmerinnen und Teilnehmer zum Einsatz. [HP12]

Neben diesen, an Universitäten entwickelten Softwarelösungen, gibt es eine Reihe von kommerziellen Audience Response Systemen. Diese Systeme verwenden meist ein eigenes Endgerät zur Beantwortung der Fragen und benutzen dabei ein proprietäres Protokoll zur Kommunikation mit dem dazugehörigen Server [HS07]. Dadurch ist der Betrieb solcher Systeme mit größeren Kosten verbunden und somit nur in eigens adaptierten Hörsälen möglich [Tob05]. Der Vorteil von solchen Systemen ist jedoch deren oftmals sehr gute Integration in verbreitete Präsentationssoftware (z.B.: LibreOffice Impress³).

Die folgende Auflistung zeigt ohne Anspruch auf Vollständigkeit einige bekannte Audience Response Systeme:

- Turning Technologies: Assessment Delivery and Data Collection Solutions ⁴
- PowerComARS: Audience Response System ⁵

² <https://realfeedback.tugraz.at/>, besucht am 29.11.2012

³ <http://www.libreoffice.org/>, besucht am 13.10.2012

⁴ <http://www.turningtechnologies.com/>, besucht am 13.10.2012

⁵ <http://www.powercomars.com/>, besucht am 13.10.2012

- IML: Harnessing Audience Insight ⁶
- uvm.

2.4 Online-Assessment

Unter Online-Assessment versteht man die computer- und vor allem internetunterstützte Beurteilung der Leistungen der Teilnehmerinnen und Teilnehmer eines Kurses. Dazu können unter anderem folgende Techniken angewandt werden: [GM07]

- Beurteilung der Nachrichten und Beiträge der Studentinnen und Studenten in Chat-Umgebungen, Foren und ähnlichen Kommunikationseinrichtungen.
- Online-Prüfungen oder Quiz
- Wöchentliche Verständnisfragen
- Selbsttests
- usw.

Ein weiterer Punkt, der zu mehr Erfolg führen kann, ist laut Gaytan und McEwen [GM07] das Feedback zum Assessment. Dabei ist es wichtig, dass das Feedback aussagekräftig ist und zeitnah angeboten wird.

Im Bereich des Online-Assessments ist jedoch zu beachten, dass es sehr schwierig ist, die Identität der Teilnehmerinnen und Teilnehmer zu verifizieren. Diese Tatsache wird von vielen E-Learning-Plattformen nur rudimentär gehandhabt, denn es werden oftmals lediglich grundlegende Sicherheitsmechanismen, wie zum Beispiel die Authentifizierung mittels Benutzername und Passwort, eingesetzt. [Her+08]

Aus diesem Grund ist eine eigene Form der Interaktion nötig, um eine ausreichende Sicherheit gewährleisten zu können, da ansonsten das Assessment an speziell adaptierten Orten, wo eine Feststellung der Identität

⁶ <http://imlworldwide.com/>, besucht am 13.10.2012

2 Stand der Technik

möglich ist, stattfinden müsste [SHBo4]. So können zum Beispiel Aufzeichnungen der Webcam und des Bildschirms der Teilnehmerin bzw. des Teilnehmers zur Sicherstellung der Identität herangezogen werden [BJ09]. Außerdem ist eine Auswertung von biometrischen Daten, dazu zählt auch die Überprüfung der Fingerabdrücke, möglich [Her+08].

Neben diesen allgemeinen Betrachtungen ist laut Ricketts und Wilks [RW02] auch die Benutzeroberfläche der angebotenen Online-Assessment-Möglichkeiten von großer Bedeutung. So ist zum Beispiel der Erfolg der Studentinnen und Studenten wesentlich größer, wenn bei einer Online-Prüfung jede Frage auf einer einzelnen Seite dargestellt wird und nicht alle Aufgaben auf einer Seite, durch welche dann gescrollt werden muss, erscheinen.

3 Design und Implementation

Dieses Kapitel dokumentiert die Entwicklung eines Web-Portals, mit welchem die im Abschnitt 1.1 beschriebenen Anforderungen und Ziele umgesetzt bzw. erreicht werden sollen. Dazu werden die entwickelten Funktionalitäten beschrieben und es wird deren Design erläutert. Ebenfalls wird an wichtigen Stellen auf Details der Implementation eingegangen. Zur besseren Übersicht wird zuvor im Abschnitt 3.1 der grundlegende Aufbau des Web-Portals auf einer sehr hohen Abstraktionsebene beschrieben.

Das Web-Portal wird in der Lage sein, die Interaktion und Kommunikation der Konsumentinnen und Konsumenten von Web-Inhalten mit deren Autorinnen und Autoren in Echtzeit zu ermöglichen. So soll zum Beispiel die Kommunikation der bzw. des Vortragenden mit den Studentinnen und Studenten im Rahmen einer Liveübertragung einer Vorlesung möglich sein, so als ob sich alle im Hörsaal befinden.

Aufgrund des erwähnten Beispiels, welches als das primäre Einsatzgebiet des Web-Portals angesehen wird, wurde dieses auf den Namen LIVE getauft. Bei dieser Abkürzung handelt es sich um ein rekursives Akronym mit folgender Bedeutung:

LIVE Interaction in Virtual learning Environments

Dieser Titel transportiert neben dem Bekenntnis zur Interaktion in Echtzeit auch den starken Bezug zum E-Learning.

3.1 Überblick

Um einen besseren Überblick über die grundlegende Architektur des Web-Portals zu erhalten, zeigt dieser Abschnitt den allgemeinen Aufbau des

3 Design und Implementation

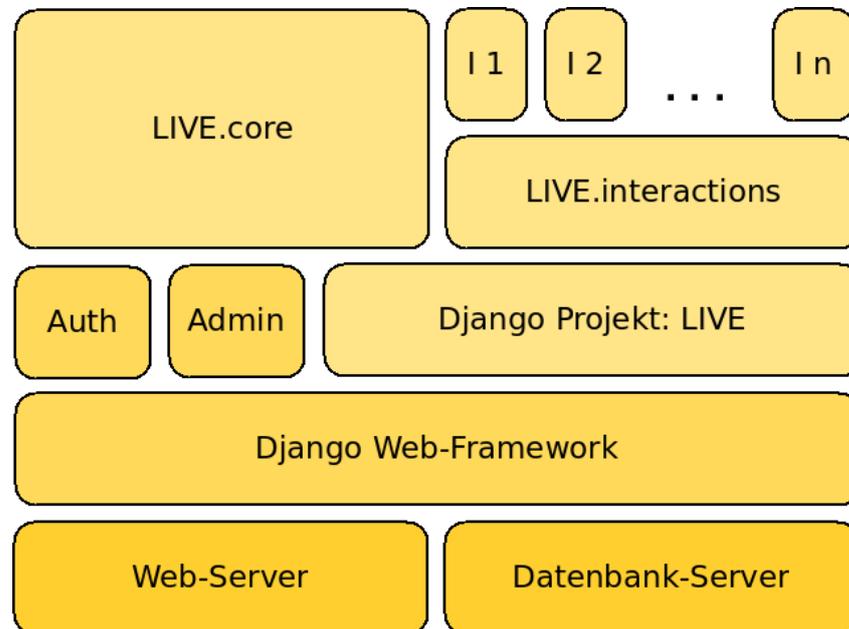


Abbildung 3.1: Der grundlegende Aufbau des Web-Portals

Systems.

Wie in Abbildung 3.1 ersichtlich, basiert LIVE auf einem Web- und einem Datenbank-Server. Darauf baut das Django Web-Framework auf, um eine einheitliche Schnittstelle für das gesamte Projekt zu bieten.

Das Framework bietet mit ihren Komponenten zur Authentifizierung und zur Administration neben der Projektdefinition von LIVE die Basis für die Implementation des Portals. Diese gliedert sich in das grundlegende Paket *LIVE.core* und das Paket *LIVE.interactions*. Das zuletzt genannte Paket bildet die Schnittstelle für die verschiedenen Interaktionsformen.

Der Web-Server stellt den Zugang zum Portal bereit. Für den ordnungsgemäßen Betrieb ist ein Web-Server nötig, welcher für das Django-Web-Framework geeignet ist. Dazu zählt neben dem Apache HTTP-Server¹ auch

¹ <http://httpd.apache.org/>, besucht am 15.10.2012

der Unicorn Web-Server².

Die Aufgabe des Datenbank-Servers ist die Speicherung aller anfallenden Daten des Portals, wie zum Beispiel den Benutzerprofilen oder den Interaktionen. Wiederum ist ein zu Django kompatibler Datenbank-Server, wie zum Beispiel MySQL³ oder PostgreSQL⁴ erforderlich.

Wie bereits erwähnt, kommt auf den beiden Servern aufbauend das Django-Web-Framework (siehe Abschnitt 3.2) zum Einsatz. Ebenfalls wird das Django *Auth*-Modul zur Unterstützung des Benutzermanagements und das Django-Modul *Admin* zur vereinfachten Administration des Web-Portals verwendet.

Die grundlegenden Funktionen des Portals, wie zum Beispiel die Veranstaltungsverwaltung, werden vom Paket *LIVE.core* bereitgestellt. Dieses Paket wird im Abschnitt 3.3 detailliert erläutert.

Die verschiedenen Interaktionsformen, welche vom Portal angeboten werden, benötigen zur einfachen Integration eine einheitliche Schnittstelle. Mit dem Paket *LIVE.interactions* wird nun eine solche angeboten (siehe Abschnitt 3.4). Dieses Paket stellt auch ein Framework zur Auswertung und Analyse der Mitarbeit der Studentinnen und Studenten an den Interaktionen bereit. Das Analyse-Framework, welches im Abschnitt 3.5 erläutert wird, stellt somit eine Abschätzung des Aufmerksamkeitsgrads der Teilnehmerinnen und Teilnehmer zur Verfügung.

Auf Basis der erwähnten Schnittstelle und des Analyse-Frameworks wurden nun die einzelnen Interaktionsformen implementiert. Dabei handelt es sich neben der Möglichkeit der bzw. des Vortragenden eine Frage an die Studentinnen und Studenten zu richten, auch um verschiedene kleine automatische Aufgaben, um die Aufmerksamkeit der Teilnehmerinnen und Teilnehmer zu erhalten bzw. zu steigern. All diese Interaktionsformen werden im Abschnitt 3.6 vorgestellt und deren Umsetzung beschrieben.

Schließlich wird im Abschnitt 3.7 auch noch die Administrationsoberfläche erläutert. Diese wird vom Modul *LIVE.core*, mit Hilfe des Django *Ad-*

² <http://unicorn.org/>, besucht am 15.10.2012

³ <http://www.mysql.com/>, besucht am 15.10.2012

⁴ <http://www.postgresql.org/>, besucht am 15.10.2012

3 Design und Implementation

min-Moduls, bereitgestellt und ermöglicht zum Beispiel die Verwaltung der Benutzerinnen und Benutzer, sowie deren Berechtigungen auf dem Web-Portal.

3.2 Das Django-Web-Framework

Beim »Django-Web-Framework«⁵ handelt es sich um ein hoch entwickeltes Framework für Web-Entwickler. Es ist mit und für die Programmiersprache »Python«⁶ geschrieben und scheint neben »Ruby on Rails«⁷ zurzeit das populärste Framework in diesem Segment zu sein. [HK12]

3.2.1 Geschichte

In den Kindertagen des World-Wide-Webs wurden die einzelnen Seiten einer Web-Präsenz noch in reinem HTML geschrieben. Wollte man etwas an diesen Seiten verändern, mussten die jeweiligen Dateien direkt bearbeitet werden. Im Falle einer kompletten Erneuerung des Designs, mussten alle Seiten entsprechend angepasst werden. [HK12]

Dieser Zustand war bald nicht mehr tragbar und führte zur Entwicklung eines Protokolls für dynamische Web-Anwendungen. Somit war das Common Gateway Interface (kurz CGI) als die erste Generation der dynamischen Web-Anwendungen geboren. Dabei wurde vom Web-Server bei einer Anfrage ein entsprechendes Programm über das CGI ausgeführt, welches den gewünschten Inhalt dynamisch erzeugte und zurück lieferte. Diese Programme waren im Vergleich zu den statischen Seiten eine große Erleichterung, jedoch kam es vor, dass ähnliche Code-Fragmente an vielen verschiedenen Stellen Verwendung fanden und somit ein schwer wartbares Programm entstand. [HK12]

⁵ <https://www.djangoproject.com/>, besucht am 16.10.2012

⁶ <http://www.python.org/>, besucht am 16.10.2012

⁷ <http://rubyonrails.org/>, besucht am 16.10.2012

3.2 Das Django-Web-Framework

Als zweite Generation im Bereich der dynamischen Web-Anwendungen wird die Einführung der Programmiersprache PHP⁸ angesehen. Diese ist bis heute eine der am weitest verbreiteten Techniken in diesem Gebiet. Dies liegt daran, dass PHP direkt in HTML-Dokumente eingebettet wird und somit sehr leicht zu erlernen ist. Diese Einfachheit führte jedoch dazu, dass schlecht und unsicher gemachte Web-Anwendungen eingesetzt wurden, da bei PHP der Entwickler selbst für die Sicherheit des Projekts verantwortlich ist. [HK12]

Diese Probleme führten zur Entwicklung von sogenannten Web-Frameworks der dritten Generation. Solche Frameworks stellen dem Entwickler eine einheitliche Programmierumgebung auf einer hohen Abstraktionsebene zur Verfügung und kümmern sich um eine Vielzahl von häufig verwendeten Funktionen. Dazu zählt zum Beispiel die Kommunikation mit einer Datenbank oder auch viele Mechanismen zur Erhöhung der Sicherheit. Das »Django-Web-Framework« ist ein Vertreter dieser Generation. [HK12]

Die Geschichte von Django reicht zurück in das Jahr 2003, in welchem das Web-Entwicklerteam der Zeitung »Lawrence Journal-World« begann, Python für Web-Anwendungen zu verwenden. Im Laufe der Zeit erreichte dieses Projekt frameworkartige Dimensionen. Dies führte schließlich zur Veröffentlichung von Django als Open-Source-Software im Sommer 2005. Als Namensgeber fungierte der Jazz-Gitarrist Django Reinhardt⁹. [HK12]

3.2.2 Prinzipien und Features

Die Entwicklung von Django erfolgt entlang der nachfolgenden Prinzipien, welche auch bei auf Django basierenden Projekten beachtet bzw. befolgt werden sollen: [Dja12a]

Prinzipien

- **Lose Zusammenhänge:** Ein Modul des Frameworks bzw. der Anwendungen soll nicht mehr über andere Module wissen als unbedingt notwendig.

⁸ <http://www.php.net/>, besucht am 16.10.2012

⁹ <http://www.redhotjazz.com/django.html>, besucht am 16.10.2012

3 Design und Implementation

- **Wenig Code:** Zur besseren Übersicht soll so wenig Programmcode wie nur möglich geschrieben werden.
- **Schnelle Entwicklung:** Das Framework und die damit entwickelten Anwendungen sollen den Anforderungen des schnelllebigen Web-Entwicklungsbereichs Rechnung tragen.
- **DRY-Prinzip:** Das »Don't repeat yourself« Prinzip besagt, dass alle Teile nur einmal gemacht werden sollen, und zwar in einer einfach wiederverwendbaren Form.
- **Explizit ist besser als Implizit:** Wo immer es möglich ist, sollten explizite Definitionen bevorzugt werden. Implizite Ableitungen sollen nur dann verwendet werden, wenn diese einen enormen Vorteil mit sich bringen.
- **Konsistenz:** Alle Teile des Frameworks und der damit entwickelten Anwendungen sollen auf allen Ebenen konsistent und einheitlich gestaltet werden.

Features Darauf aufbauend bietet Django unter anderem folgende Funktionen und Features an: [Dja12b; HK12]

- Lose Orientierung am MVC-Entwurfsmuster (siehe Abschnitt 3.2.3).
- Eine mächtige Datenbank-API inklusive eines objektrelationalen Mappers (kurz ORM)¹⁰.
- Explizite Konfiguration der URLs¹¹ mit regulären Ausdrücken.
- Ein umfangreiches Template-System
- Internationalisierung und Lokalisierung
- Benutzer- und Rechteverwaltung, sowie Authentifizierung

¹⁰Ein objektrelationaler Mapper ermöglicht, im Rahmen einer objektorientierten Programmiersprache, das Speichern der konkreten Objekte von Klassen in einer relationalen Datenbank. Umgekehrt betrachtet, handelt es sich um eine Abbildung der Datenbank in Form von Klassen und Objekten

¹¹ kurz für »Uniform Resource Locator«

3.2 Das Django-Web-Framework

- Verschiedene Sicherheitsmechanismen wie zum Beispiel der Schutz vor
 - SQL-Injections oder
 - Cross-Site-Scripting.
- Eigenständiger und leichtgewichtiger Web-Server für die Entwicklungsphase.
- Viele weitere nützliche Funktionalitäten, wie etwa eine API zur Erstellung und Auswertung von Formularen.

3.2.3 MVC vs. MTV Entwurfsmuster

Wie bereits erwähnt, folgt man bei der Entwicklung eines Projekts mit Django lose dem MVC-Entwurfsmuster. Dies hat im Rahmen von Django folgende Bedeutung: [HK12] Model, View, Controller

- **Model**¹²: Dient zur Speicherung und Bereitstellung aller anfallenden Daten. Dazu ist die Datenbank-API mit dem ORM verantwortlich.
- **View**¹³: Übernimmt die Auswahl und bestimmt die Form der Darstellung der angeforderten Daten. Dies wird durch die Django-Views und Templates bereitgestellt.
- **Controller**¹⁴: Leitet die Anfrage an die entsprechende View weiter. Dies wird vom Framework, auf Basis der URL-Konfiguration weitgehend automatisch erledigt.

Da der Punkt Controller annähernd automatisch abläuft, scheint eine Klassifikation nach dem MTV-Entwurfsmuster sinnvoller [HK12]. Dieser Umstand ist auch in Abbildung 3.2 ersichtlich. Model, Template, View

- **Model**: Entspricht der Darstellung beim MVC-Muster.

¹² englisch für »Modell«

¹³ englisch für »Präsentation«

¹⁴ englisch für »Steuerung«

3 Design und Implementation

- **Template**¹⁵: Enthält alle für die Darstellung relevanten Anweisungen.
- **View**: Stellt die eigentliche Programmlogik bereit. Üblicherweise werden Daten mit Hilfe der Modelle verarbeitet bzw. geladen und mit Hilfe eines Templates dargestellt.

Ablauf einer Anfrage

Daraus lässt sich ein gewisses Zusammenspiel der einzelnen Komponenten erkennen. Dieser, in Abbildung 3.2 dargestellte Ablauf, gestaltet sich folgendermaßen: [Dja12b]

1. Am Beginn steht immer eine Anfrage eines Clients (z.B. eines Web-Browsers) an den Web-Server.
2. Dieser leitet die Anfrage an das Django-Framework weiter, wo anhand der URL-Konfiguration,
3. die Anfrage zur passenden View geleitet wird.
4. Nun wird gegebenenfalls eine Abfrage an die Datenbank mit der entsprechenden API gestellt.
5. Die Ergebnisdaten dieser Abfrage werden wieder an die View retourniert.
6. Nach einer etwaigen Verarbeitung dieser Daten, werden diese mit Hilfe eines Templates gerendert. Dabei kommen verschiedene Tags und Filter zur Steuerung der Erzeugung der Antwort zum Einsatz.
7. Die Antwort wird nun im gewünschten Format (z.B. als HTML) an die View zurück gegeben und
8. über den Web-Server
9. an den Client zurück geliefert.

¹⁵ englisch für »Schablone«

3.2 Das Django-Web-Framework

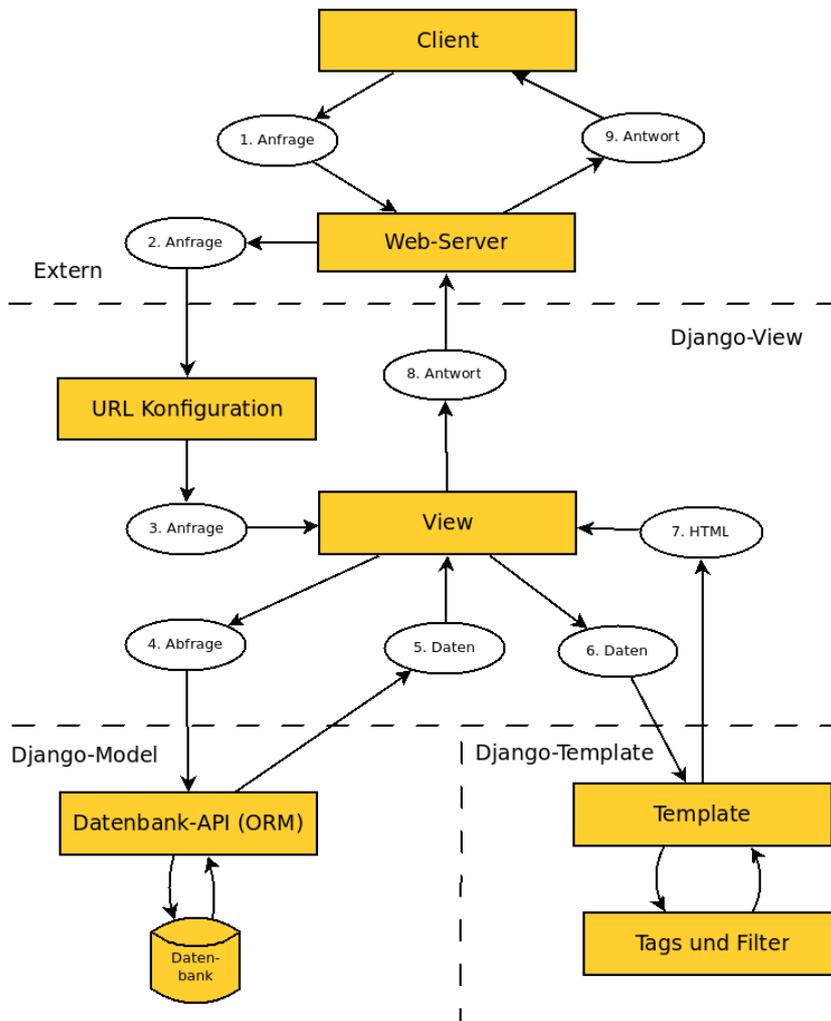


Abbildung 3.2: Ein Überblick über das Zusammenspiel der Django-Komponenten (abgeleitet aus der *Django Documentation* [Dja12b; Pro12])

3.3 Das Grundgerüst

Wie bereits im Abschnitt 3.1 erwähnt, bildet das Django-Projekt *LIVE* die Basis für die Implementation des Web-Portals. Bei einem Django-Projekt handelt es sich um ein Python-Package, welches im Rahmen dieses Web-Portals, unterhalb des Wurzelordners *LIVE*, folgenden Aufbau besitzt:

Aufbau des
Projekts

- *core*
Dieses Paket bildet das Grundgerüst für das Web-Portal und stellt wichtige Funktionen, wie zum Beispiel die Veranstaltungsverwaltung bereit.
- *interactions*
Die Schnittstelle für die Interaktionsformen und einige dazugehörige Funktionalitäten, wie zum Beispiel das Analyse-Framework, werden von diesem Paket angeboten.
- *templates*
Dieser Ordner beinhaltet alle zur Darstellung des Web-Portals nötigen Templates.
- *files*
Alle statischen Dateien des Web-Portals befinden sich in diesem Ordner. Dazu zählen vor allem Javascript- und css-Dateien.
- *locale*
In diesem Ordner befinden sich die Übersetzungen des Web-Portals.
- *__init__.py*
Dies ist die Initialisierungsdatei für das Python-Package.
- *manage.py*
Hiermit wird ein einfacher Zugriff auf verschiedene Funktionen von Django bereitgestellt. So kann zum Beispiel der Entwickler-Web-Server gestartet werden.
- *settings.py*
Diese Datei enthält alle globalen Einstellungen des Web-Portals. Unter anderem werden hier die Zugangsdaten für den Datenbank-Server hinterlegt.

3.3 Das Grundgerüst

- *settings_production.py*
Da manche Einstellungen für den produktiven Betrieb des Web-Portals geändert werden müssen, können in dieser Datei die globalen Einstellungen aus *settings.py* überschrieben bzw. erweitert werden.
- *urls.py*
Hier erfolgt die explizite URL-Konfiguration. Dabei wird hauptsächlich auf weitere Konfigurationen in den Paketen *core* und *interactions* verwiesen.

Bei den Paketen *core* und *interactions* handelt es sich um sogenannte Django-Apps. Diese sind im allgemeinen folgendermaßen aufgebaut: [Dja12b] Aufbau einer Django-App

- *__init__.py*
Diese Initialisierungsdatei ist nötig, da eine Django-App ein Python-Package ist.
- *models.py*
Enthält alle Modelle der App.
- *views.py*
Stellt die Views der App bereit.
- *urls.py*
Hier erfolgt die explizite URL-Konfiguration der App.
- Gegebenenfalls weitere Dateien wie zum Beispiel *admin.py* zur Konfiguration der Administrationsoberfläche.

Um eine Django-App nun im Rahmen eines Projekts zu verwenden, muss diese in den Einstellungen aktiviert werden. Allenfalls ist auch die URL-Konfiguration der App, in die des Projekts, zu integrieren. [Dja12b]

In den folgenden Abschnitten werden nun die einzelnen Funktionen und Komponenten des Grundgerüsts umfangreich erläutert.

3 Design und Implementation



Abbildung 3.3: Die Startseite des Web-Portals fungiert als Beispiel für die Benutzeroberfläche der Hauptseiten

3.3.1 Benutzeroberfläche

Wichtige Teile des Grundgerüsts sind das Design und die Bedienkonzepte der Benutzeroberfläche. Dabei wurde vor allem auf die einfache und intuitive Benutzbarkeit geachtet, denn wenn laut Norman [Nor92] »einfache Dinge Bilder, Beschriftungen oder Anleitungen benötigen, ist deren Design fehlgeschlagen«.

Grundsätzlich gibt es zwei verschiedene Typen der Benutzeroberfläche. Zum Einen existiert die Oberfläche der Hauptseiten des Web-Portals und zum Anderen kommt eine angepasste Oberfläche für die Teilnehmerinnen und Teilnehmer während einer Veranstaltung zum Einsatz. Die Letztgenannte wird im Abschnitt 3.4 näher betrachtet.

Wie in Abbildung 3.3 ersichtlich, befindet sich unter einer Titelleiste die Hauptnavigation der Benutzeroberfläche der Hauptseiten. Über diese erreicht man unter anderem, wie einfach erkennbar, die Benutzerverwaltung und die Veranstaltungsverwaltung. Unterhalb befindet sich der Inhalt der jeweiligen Seite. Am unteren Ende kann auch noch die Sprache des Web-Portals geändert werden.

Die Benutzeroberfläche wurde mit Hilfe von »jQuery UI«¹⁶, einer Sammlung von verschiedenen Komponenten für Benutzeroberflächen auf Basis

¹⁶ <http://jqueryui.com/>, besucht am 19.10.2012

3.3 Das Grundgerüst

der Javascript Bibliothek »jQuery«¹⁷, umgesetzt. Dabei kommt das Thema namens »UI-lightness« der ThemeRoller-API zum Einsatz.

Ein weiterer wichtiger, oft vergessener Teil der Benutzeroberfläche ist die URL. Beim Design der URLs sollten folgende Punkte beachtet werden: [And12]

- URLs sollten kurz sein.
- Alle Teile der URL sollten aussagekräftig sein.
- Die URL sollte leicht zu tippen sein.
- Die logische Struktur des Web-Portals sollte von der URL wiederspiegelt werden.
- Das Entfernen des Endes der URL sollte das Nach-Oben-Wandern in der Hierarchie des Portals ermöglichen.

Diese Anforderungen an die URLs wurden bei der expliziten URL-Konfiguration beachtet. In der folgenden Auflistung, welche einen Überblick über die URL-Struktur des Web-Portals gibt, ist die Anwendung dieser Anforderungen erkennbar:

- /
- */accounts/*
 - */accounts/newuser/*
 - */accounts/my/*
 - * */accounts/my/edit/*
 - * */accounts/my/changepw/*
 - * usw.
 - usw.
- */event/*
 - */event/create/*

¹⁷ <http://jquery.com/>, besucht am 19.10.2012

3 Design und Implementation

- * */event/(event_id)/edit/*
- * */event/(event_id)/start/*
- * */event/(event_id)/join/*
- * usw.
- usw.
- */interactions/*
 - */interactions/...*
- */admin/*
 - */admin/...*
- */legalnotice/*
- */help/*

Anhand dieser vereinfachten Darstellung der URL-Struktur erkennt man den Zusammenhang mit der Hauptnavigation. Die Benutzerverwaltung ist über die URLs unter */accounts/* und die Veranstaltungsverwaltung unter */event/* erreichbar. Die Interaktionsformen werden durch die URLs unterhalb von */interactions/* gesteuert. Diese werden aber von den Benutzerinnen und Benutzern nicht direkt angesurft.

Das hierarchische Design der URLs zeigt sich deutlich an den URLs zur Veranstaltungsverwaltung. Auf der obersten Ebene (*/event/*) ist eine Auflistung der Veranstaltungen erreichbar. Die darunterliegende Ebene ermöglicht das Arbeiten mit den Veranstaltungen. So kann zum Beispiel eine Veranstaltung erstellt werden (*/event/create/*). Wird eine zu einer Veranstaltung gehörende Aktion (z.B. ändern der Veranstaltungsdetails) aufgerufen, befindet sich eine eindeutige Identifikationsnummer der Veranstaltung als Parameter auf der zweiten Ebene der URL und erst auf dritter Ebene steht die gewünschte Aktion (z.B. */event/(event_id)/edit/*).

Somit ist eine Bedienung des Web-Portals nicht nur über die angebotenen Schaltflächen und Links möglich, sondern auch durch die direkte Eingabe der gewünschten URL.

3.3.2 Internationalisierung und Lokalisierung

Das Django-Web-Framework ermöglicht die Entwicklung von vollständig internationalisierten und lokalisierten Web-Portalen [Dja12b]. Dies bedeutet, dass neben den Texten des Web-Portals auch die Darstellung von Datum und Uhrzeit, sowie das Aussehen der Zahlen von der gewählten Sprache abhängig ist. Dazu kommt die Python Bibliothek namens »gettext«^{18,19} zum Einsatz. [Dja12b]

Unter Internationalisierung versteht man die Entwicklung eines Web-Portals, welches in der Lage ist, alle ihre Komponenten nicht nur in einer Standardsprache, sondern auch in beliebigen anderen Sprachen darzustellen. Darauf aufbauend, werden bei der Lokalisierung die Texte der Komponenten von der Standardsprache in eine beliebige andere Sprache, in welcher das Web-Portal angeboten werden soll, übersetzt. [Dja12b]

Im Rahmen der Internationalisierung werden nun also bei der Implementation alle Texte des Web-Portals, welche in der Standardsprache geschrieben sind, speziell gekennzeichnet. Dadurch ist es möglich, diese Texte zu extrahieren, um diese für die Lokalisierung bereitzustellen und sie dann durch die lokalisierte Version zu ersetzen. Die erwähnte Kennzeichnung der Texte wird in allen Python-Dateien und in den Templates vorgenommen. [Dja12b]

Die extrahierten Texte werden in speziellen Dateien im Ordner *locale* gespeichert und bei der Lokalisierung in die gewünschte Sprache übersetzt. Die Standardsprache dieses Web-Portals ist Englisch und es existiert eine deutsche Lokalisierung.

Django wurde so konfiguriert, dass es anhand der Reihenfolge der folgenden Richtlinien die Lokalisierung auswählt: [Dja12b]

1. Es wird versucht, die gewünschte Lokalisierung aus der Session des aktuellen Benutzers herauszufinden. Dies ist der Fall, wenn die Möglichkeit zur manuellen Auswahl der Sprache genutzt wurde (siehe Abbildung 3.3).

¹⁸ <http://docs.python.org/library/gettext.html>, besucht am 19.10.2012

¹⁹ <https://docs.djangoproject.com/en/1.4/topics/i18n/translation/>, besucht am 19.10.2012

3 Design und Implementation

2. Scheitert dies, wird nach einem entsprechenden Cookie gesucht.
3. Schlägt auch dies fehl, wird der Accept-Language HTTP Header ausgewertet.
4. Als letzte Konsequenz wird die Standardsprache verwendet.

3.3.3 Benutzermanagement

Einer der wichtigsten Teile des Grundgerüsts ist das Benutzermanagement. Es stellt folgenden Funktionsumfang zur Verfügung:

- **Verwaltung der Benutzerprofile**
Die Benutzerinnen und Benutzer können ein neues Profil erstellen und dieses verändern.
- **Verwaltung der Berechtigungen**
Zusätzlich zu normalen Benutzerprofilen gibt es auch solche mit Rechten für Lehrende und Administratoren.
- **Authentifizierung und Zugangskontrolle**
Das Web-Portal steht nur registrierten und authentifizierten Benutzerinnen und Benutzern zur Verfügung. Diese können es laut ihren Berechtigungen benutzen.

Das gesamte Benutzermanagement wurde mit Hilfe des Django-*Auth*-Moduls²⁰ umgesetzt. Dadurch steht ein mächtiges Benutzermanagement mit einer auf Cookies basierenden Benutzer-Session und einer Infrastruktur zur Verwaltung der Benutzerprofile sowie der Rechte und Gruppen bereit [Dja12b].

Normale Benutzerprofile berechtigen zur Teilnahme an Veranstaltungen und zur Nutzung der dabei angebotenen Interaktionsformen. Mit zusätzlichen Rechten für Lehrende ist es auch möglich, Veranstaltungen zu erstellen und mit diesen zu arbeiten (siehe Abschnitt 3.3.4). Mit Administrations-Rechten ist auch ein Zugriff auf die Administrationsoberfläche möglich, wo

²⁰ <https://docs.djangoproject.com/en/1.4/topics/auth/>, besucht am 19.10.2012

3.3 Das Grundgerüst

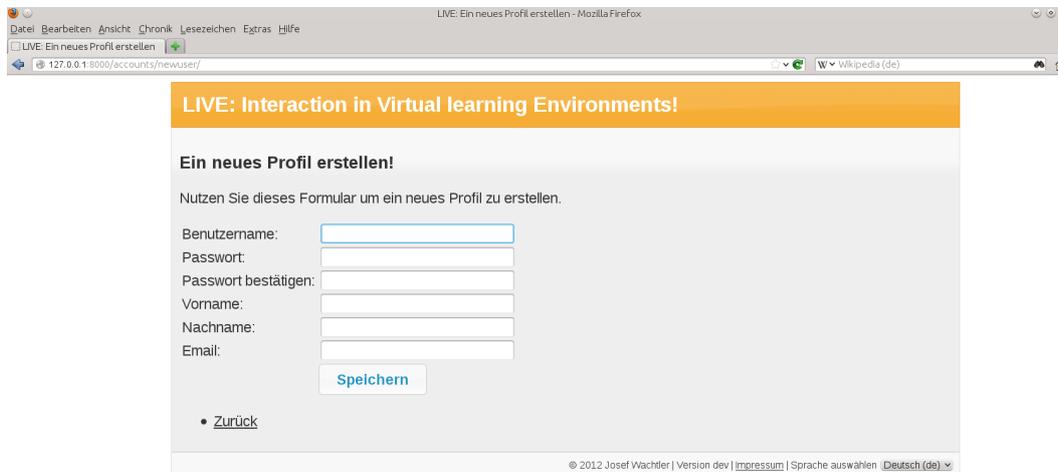


Abbildung 3.4: Ein neues Profil erstellen

zum Beispiel die Rechte der Benutzerinnen und Benutzer geändert werden können (vgl. Abschnitt 3.7).

Für viele Standardfunktionen, wie zum Beispiel für das Ein- und Ausloggen von Benutzern, bietet das Django-Auth-Modul bereits vorgefertigte Views an. Es mussten nur die zum Web-Portal passenden Templates implementiert werden. Des Weiteren wurden Views und Templates zur Erstellung (siehe Abbildung 3.4) und Bearbeitung eines Benutzerprofils umgesetzt.

Damit nun die Authentifizierung und die Berechtigungen Anwendung finden, musste bei der Implementation der Views auf ein paar Besonderheiten Rücksicht genommen werden. Deshalb wurden alle Views, welche nur für eingeloggte Benutzerinnen und Benutzer zugänglich sein sollen, mit einem entsprechenden Decorator²¹ gekennzeichnet. Ein weiterer Decorator kommt zum Einsatz, wenn die View nur mit bestimmten Berechtigungen (z.B. Rechte für Lehrende) aufrufbar sein soll.

Weiters wurde bei der Implementation der Views darauf geachtet, dass die Benutzerinnen und Benutzer nur auf Daten Zugriff haben, welche auch

²¹ <http://wiki.python.org/moin/PythonDecorators>, besucht am 19.10.2012

3 Design und Implementation

```
1 @login_required
2 @user_passes_test(is_in_teacher_group)
3 def edit_event(request, event_id):
4     event = get_object_or_404(Event, id=event_id,
5         teacher=request.user)
6     ...
```

Quellcode 3.1: Der Beginn der View zur Bearbeitung einer Veranstaltung zeigt die Anwendung der Maßnahmen des Benutzermanagements

für sie bestimmt sind. So sollen sie zum Beispiel nur ihr eigenes Profil verändern können.

Diese beschriebenen Maßnahmen werden im Quellcode 3.1 anhand der View zum Bearbeiten einer Veranstaltung dargestellt. Man sieht in der Zeile 1 den Decorator, welcher die View nur für eingeloggte Benutzerinnen und Benutzer zugänglich macht. Darunter befindet sich der Decorator, mit dem das Vorhandensein von Rechten für Lehrende überprüft wird. Nach der Definition der View in der dritten Zeile, wird die zu bearbeitende Veranstaltung geladen. Dabei wird darauf geachtet, dass der Lehrende der Veranstaltung auch die aktuelle Benutzerin bzw. der aktuelle Benutzer ist (Zeile 6). Danach folgt der restliche Code der View.

3.3.4 Veranstaltungsverwaltung

Der Funktionsumfang der Veranstaltungsverwaltung kann einmal aus Sicht der normalen Benutzerinnen und Benutzer und ein weiteres Mal aus der Perspektive solcher mit Rechten für Lehrende gesehen werden. Mit normalen Rechten ist es nur möglich an Veranstaltungen teilzunehmen und danach kann noch eine Auswertung der eigenen Aufmerksamkeit während der Veranstaltung eingesehen werden.

Im Vergleich dazu ist mit Rechten für Lehrende ein weitaus größeres Funktionsangebot vorhanden. So können Veranstaltungen nicht nur erstellt, sondern auch bearbeitet und gelöscht werden. Weiters ist es möglich, Veranstaltungen zu starten sowie diese wieder zu beenden.

3.3 Das Grundgerüst



Abbildung 3.5: Der typische Ablauf einer Veranstaltung

Der typische Ablauf einer Veranstaltung ist in Abbildung 3.5 ersichtlich. Nachdem eine Veranstaltung erstellt wurde, kann diese gestartet werden und die Studentinnen und Studenten können daran teilnehmen. Dabei interagieren und kommunizieren sie mit der bzw. dem Vortragenden mit Hilfe der angebotenen Interaktionsformen. Schließlich erfolgt nach der Beendigung der Veranstaltung eine Analyse der Aufmerksamkeit. Mehr dazu kann den folgenden Abschnitten entnommen werden.

Ablauf einer
Veranstaltung

Die Erstellung einer Veranstaltung erfolgt mit dem in Abbildung 3.6 dargestellten Dialog. Dazu muss ein Titel für die Veranstaltung sowie der geplante Beginn und das geplante Ende eingegeben werden. Optional ist auch die Eingabe einer URL zum Web-Inhalt der Veranstaltung (z.B. zur Liveübertragung) möglich. Zusätzlich müssen auch noch die bei der Veranstaltung angebotenen Interaktionsformen ausgewählt werden. In der zugrunde liegenden View wird der Dialog nun ausgewertet und mittels eines Modells (siehe Quellcode 3.2) gespeichert. Somit werden also die Felder *title*, *scheduled_start*, *scheduled_end*, *teacher*, *interactions* sowie gegebenenfalls das Feld *url* befüllt.

Erstellen einer
Veranstaltung

Nun kann die Veranstaltung gestartet werden. Dazu wird in der entsprechenden View die tatsächliche Beginnzeit (*real_start*) des Modells der Veranstaltung gesetzt.

Ab diesem Zeitpunkt können die Benutzerinnen und Benutzer an der Veranstaltung teilnehmen. Dazu wird in der View das Modell *JoinedUser* (siehe Quellcode 3.3) zur Herstellung einer Verbindung zwischen der Veranstaltung und dem Benutzerprofil eingesetzt. Dieses Modell zeichnet auch den Zeitraum der Teilnahme (von *joined_at* bis *leaved_at*) auf.

Teilnehmen an
einer
Veranstaltung

3 Design und Implementation

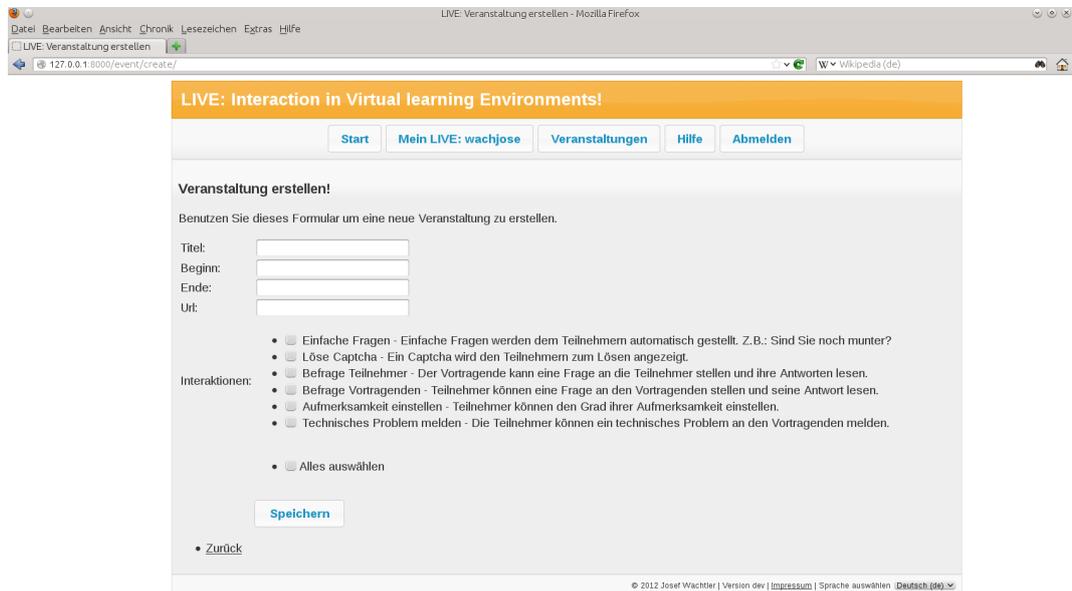


Abbildung 3.6: Dieser Screenshot zeigt den Dialog zum Erstellen einer neuen Veranstaltung

```
1 class Event(models.Model):
2     title = models.CharField(max_length=128)
3
4     scheduled_start = models.DateTimeField()
5     real_start = models.DateTimeField(blank=True, null=True)
6
7     scheduled_end = models.DateTimeField()
8     real_end = models.DateTimeField(blank=True, null=True)
9
10    teacher = models.ForeignKey(User)
11
12    url = models.URLField(blank=True)
13
14    interactions = models.ManyToManyField(InteractionMethod)
```

Quellcode 3.2: Das Modell der Veranstaltungen

3.3 Das Grundgerüst

```
1 class JoinedUser(models.Model):
2     student = models.ForeignKey(User)
3     event = models.ForeignKey(Event)
4
5     joined_at = models.DateTimeField()
6     leaved_at = models.DateTimeField(blank=True, null=True)
```

Quellcode 3.3: Dieses Modell verbindet ein Benutzerprofil mit einer Veranstaltung

Während der Veranstaltung kümmert sich nun die Interaktionen-Architektur um die Abwicklung der Interaktionsformen. Dabei findet auch das Analyse-Framework zur Bestimmung des Aufmerksamkeitsgrads der Teilnehmerinnen und Teilnehmer Verwendung. Mehr dazu kann den Abschnitten 3.4, 3.5 und 3.6 entnommen werden.

Das Ende der Veranstaltung muss von der bzw. dem Vortragenden eingeleitet werden. Dazu wird durch die zuständige View das tatsächliche Ende (*real_end*) der Veranstaltung eingetragen. Dies führt auch bei den Teilnehmerinnen und Teilnehmern zum Ende der Veranstaltung und sie werden um eine Schätzung ihres durchschnittlichen Aufmerksamkeitsgrads gebeten.

Beenden einer
Veranstaltung

Nach einer Veranstaltung erhalten die Vortragenden eine Auflistung aller Teilnehmerinnen und Teilnehmer mit deren jeweiligen Aufmerksamkeitsgrad (siehe Abbildung 3.7). Durch Anklicken einer Teilnehmerin oder eines Teilnehmers aus der Liste, wird eine detailliertere Analyse, welche auch von diesem selbst abgerufen werden kann, dargestellt. In Abbildung 3.8 sieht man, dass neben der verspäteten Teilnahme und dem zu frühen Verlassen auch ein Vergleich des geschätzten Aufmerksamkeitsgrads mit dem tatsächlichen angezeigt wird. Für die Analyse kommt das Framework, welches in Abschnitt 3.5 näher behandelt wird, zum Einsatz.

Die in diesem Abschnitt beschriebenen Funktionen der Veranstaltungsverwaltung decken den gesamten Lebenszyklus einer solchen ab. Dieser reicht von der Erstellung über die Durchführung bis zur Analyse danach.

3 Design und Implementation



Abbildung 3.7: Eine Analyse der Aufmerksamkeit nach der Veranstaltung



Abbildung 3.8: Die detailliertere Aufmerksamkeitsanalyse einer Teilnehmerin oder eines Teilnehmers

3.4 Die Interaktionen-Architektur

Die Interaktionen-Architektur stellt für die Erfüllung der Aufgaben des Web-Portals zwei existenzielle Funktionalitäten bereit. Dabei handelt es sich zum Einen um die Definition einer Schnittstelle für die Interaktionsformen und zum Anderen um deren geregelten Ablauf während einer Veranstaltung. Dieser Ablauf kann sowohl aus Sicht der Vortragenden als auch aus der Perspektive der Teilnehmerinnen und Teilnehmer betrachtet werden.

Grundsätzlich gibt es drei verschiedene Typen von Interaktionsformen:

1. **Automatisch ausgelöste Interaktionsformen**

Diese werden den Teilnehmerinnen und Teilnehmern automatisch angeboten. Dabei handelt es sich hauptsächlich um kleinere Aufgaben, wie zum Beispiel das Lösen eines Captchas.

2. **Von Teilnehmern ausgelöste Interaktionsformen**

Die Teilnehmerinnen und Teilnehmer können Interaktionen starten, um zum Beispiel eine Frage an die Vortragende bzw. den Vortragenden zu richten.

3. **Von Vortragenden ausgelöste Interaktionsformen**

Umgekehrt können auch die Vortragenden Interaktionen in Gang setzen.

3.4.1 Die Schnittstelle

Um nun die Interaktionsformen in einer einheitlichen und flexiblen Art und Weise implementieren zu können, stellt die Django-App *interactions* einige Anforderungen und bietet zusätzlich nützliche Werkzeuge an. Diese bilden in ihrer Gesamtheit die Schnittstelle für die Interaktionen und werden nachfolgend erklärt.

Eine Interaktionsform wird als eigenständige Django-App umgesetzt. Diese muss sich als Sub-App innerhalb der *interactions*-App befinden.

3 Design und Implementation

Zusätzlich zu den üblichen Modulen (z.B. *views*) muss auch ein Modul namens *settings* vorhanden sein. Dieses Modul enthält die grundlegenden Einstellungen einer Interaktionsform (siehe Tabelle 3.1). Je nach Typ der Interaktionsform stehen neben den allgemeinen Einstellungen (z.B. Titel, Beschreibung) unterschiedliche Einstellungsmöglichkeiten bereit. Außerdem sind die meisten Einstellungen zwingend vorgeschrieben. Manche jedoch sind optional. Durch das Analyse-Framework kommen gegebenenfalls noch weitere optionale Einstellungsmöglichkeiten hinzu (siehe Abschnitt 3.5).

Manche Interaktionsformen lassen es zweckmäßig erscheinen, Details über ihren Einsatz bei einer Veranstaltung im Nachhinein darzustellen. So können zum Beispiel alle Fragen der Teilnehmerinnen und Teilnehmer auch nach der Veranstaltung noch abgerufen werden (siehe Abschnitt 3.6.4). Soll dieses Feature von einer Interaktionsform verwendet werden, muss eine entsprechende View implementiert und deren Name in den Einstellungen (`DETAIL_VIEW`) bekannt gegeben werden.

Handelt es sich bei der Interaktionsform um eine automatisch ausgelöste (Typ 1) ist es notwendig anzugeben, wie oft diese pro Stunde gestartet werden soll (`TRIGGERS_PER_HOUR`). Dadurch ist es möglich, einen Plan für die automatischen Interaktionsformen zu berechnen, welcher aussagt, wann welche Interaktionsform ausgeführt wird. Diese Berechnung der Startzeiten erfolgt im Rahmen dieser Einstellung auf zufälliger Basis und ist für jede Teilnehmerin bzw. jeden Teilnehmer anders.

Bei den Interaktionsformen der Typen 2 und 3 ist die Bereitstellung eines Templates mit den zum Auslösen der Interaktionsform notwendigen Bedienelementen zwingend vorgeschrieben. Dieses Template wird somit dazu benutzt, um die genannten Bedienelemente den Benutzerinnen und Benutzern anzuzeigen. Über diese Bedienelemente muss eine View aufgerufen werden, welche die gewünschten Interaktionen nun auslöst.

Um eine Interaktion auszulösen, ist es notwendig, ein entsprechendes Modell zu instanziiieren. Wie bereits erwähnt, geschieht dies bei Interaktionsformen vom Typ 1 automatisch und bei Typ 2 oder 3 muss dies in einer eigenen View erledigt werden. Dazu gibt es drei verschiedene Modelle:

3.4 Die Interaktionen-Architektur

Einstellung	zwingend
Beschreibung	
Allgemeine Einstellungen	
TITLE	ja
Ein aussagekräftiger Titel der Interaktionsform.	
DESCRIPTION	ja
Eine detaillierte Beschreibung der Interaktionsform.	
INTERACTION_TYPE	ja
Typ der Interaktionsform. 1: Automatisch ausgelöst; 2: vom Teilnehmer gestartet; 3: vom Vortragenden initiiert	
DETAIL_VIEW	nein
Name einer View, welche detailliertere Informationen über ihren Einsatz bei einer Veranstaltung bereitstellt.	
Einstellungen bei Typ 1	
TRIGGERS_PER_HOUR	ja
Gibt an, wie oft die Interaktion in einer Stunde ausgeführt werden soll.	
Einstellungen bei Typ 2	
STUDENT_TEMPLATE	ja
Pfad zu einem Template, welches die Bedienelemente zum Auslösen der Interaktion durch die Teilnehmerin bzw. den Teilnehmer bereitstellt.	
Einstellungen bei Typ 3	
TEACHER_TEMPLATE	ja
Pfad zu einem Template, mit dem die Bedienelemente zum Starten der Interaktion durch die Vortragende bzw. den Vortragenden bereitgestellt werden.	

Tabelle 3.1: Die Einstellungen für eine Interaktionsform

3 Design und Implementation

```
1 class Interaction(models.Model):
2     joined_user = models.ForeignKey(JoinedUser)
3     interactions = models.ForeignKey(InteractionMethod)
4
5     scheduled_start = models.DateTimeField()
6     real_start = models.DateTimeField(blank=True, null=True)
7     responded_at = models.DateTimeField(blank=True, null=True)
```

Quellcode 3.4: Dieses Modell löst eine Interaktion bei einer Teilnehmerin oder einem Teilnehmer aus

1. Ein Modell zum Auslösen einer Interaktion bei einer Teilnehmerin oder einem Teilnehmer.
2. Ein Weiteres, um eine Interaktion bei der bzw. dem Vortragenden auszulösen.
3. Mit einem anderen Modell kann eine Fehlermeldung bei der bzw. dem Vortragenden angezeigt werden.

Interaktionen
für die
Teilnehmerinnen
und Teilnehmer

Quellcode 3.4 zeigt das Modell zum Auslösen einer Interaktion bei einer Teilnehmerin oder einem Teilnehmer. Dabei muss neben der gewünschten Teilnehmerin bzw. neben dem gewünschten Teilnehmer (*joined_user*) auch die aufrufende Interaktionsform (*interactions*) angegeben werden. Weiters ist es notwendig, die geplante Startzeit anzugeben. Soll die Interaktion mehrere Teilnehmerinnen bzw. Teilnehmer betreffen, so muss für jede bzw. jeden ein Modell instanziiert werden.

Das Feld *real_start* wird automatisch gesetzt und enthält die tatsächliche Startzeit der Interaktion. Sobald die Teilnehmerin bzw. der Teilnehmer auf die Interaktion antwortet, muss der Zeitpunkt der Antwort im Feld *responded_at* eingetragen werden.

Interaktionen
für die
Vortragende bzw.
den Vortragenden

Betrifft die Interaktion die Vortragende bzw. den Vortragenden, ist das Modell aus Quellcode 3.5 zu verwenden. Dieses ähnelt sehr stark dem Modell für die Interaktionen der Teilnehmerinnen bzw. der Teilnehmer. Der einzige Unterschied liegt darin, dass anstelle der Teilnehmerin bzw. des Teilnehmers die Veranstaltung (*event*) angegeben wird.

3.4 Die Interaktionen-Architektur

```
1 class TeacherInteraction(models.Model):
2     event = models.ForeignKey(Event)
3     interactions = models.ForeignKey(InteractionMethod)
4
5     scheduled_start = models.DateTimeField()
6     real_start = models.DateTimeField(blank=True, null=True)
7     responded_at = models.DateTimeField(blank=True, null=True)
```

Quellcode 3.5: Mit diesem Modell wird eine Interaktion beim Vortragenden der Veranstaltung ausgelöst

```
1 class ErrorMessage(models.Model):
2     event = models.ForeignKey(Event)
3     joined_user = models.ForeignKey(JoinedUser,
4         blank=True, null=True)
5
6     triggered_at = models.DateTimeField()
7     responded_at = models.DateTimeField(blank=True, null=True)
8
9     subject = models.CharField(max_length=128)
10    message = models.TextField()
```

Quellcode 3.6: Für eine Fehlermeldung bei der bzw. dem Vortragenden wird dieses Modell verwendet

3 Design und Implementation

```
1 @login_required
2 def index(request, interaction_id):
3     ...
```

Quellcode 3.7: Die Signatur einer View zum Anzeigen einer Interaktion bei einer Teilnehmerin bzw. einem Teilnehmer

```
1 @login_required
2 @user_passes_test(is_in_teacher_group)
3 def teacher_index(request, tinteraction_id):
4     ...
```

Quellcode 3.8: Die Signatur einer View zum Darstellen einer Interaktion bei der bzw. dem Vortragenden

Fehlermeldung anzeigen	Um eine Fehlermeldung bei der bzw. dem Vortragenden einzublenden, kann das Modell aus Quellcode 3.6 verwendet werden. Es muss zusätzlich zur betreffenden Veranstaltung (<i>event</i>) und den Zeitpunkt des Fehlers (<i>triggered_at</i>) natürlich auch ein Betreff (<i>subject</i>) und eine Nachricht (<i>message</i>) des Fehlers eingetragen werden. Wird die Fehlermeldung von einer Teilnehmerin bzw. einem Teilnehmer ausgelöst, ist diese bzw. dieser im Feld <i>joined_user</i> anzugeben.
Anzeigen der Interaktionen	Zur geplanten Startzeit einer Interaktion wird diese, je nach verwendetem Modell, bei der Teilnehmerin bzw. dem Teilnehmer oder bei der bzw. dem Vortragenden angezeigt. Dazu muss die Interaktionsform spezielle Views mit bestimmten Signaturen anbieten (siehe Quellcode 3.7 und 3.8). Bei beiden ist vor allem der Name (<i>index</i> oder <i>teacher_index</i>) und der zweite Parameter von Bedeutung. Dieser Parameter enthält die Identifikationsnummer der Interaktion, welche durch die Modelle zur Verfügung steht. Die View sollte nun die Interaktion mit dieser Identifikationsnummer laden und deren Gültigkeit prüfen. Danach kann die Implementation der Darstellung der Interaktion folgen.
Interaktionen überprüfen	Um die genannte Überprüfung der Interaktion durchzuführen, steht sowohl für die Interaktionen der Teilnehmerinnen und Teilnehmer als auch für die der Vortragenden eine entsprechende Funktion bereit. Diese Funktionen führen unter anderem folgende grundlegende Überprüfungen durch:

3.4 Die Interaktionen-Architektur

- Wird die aufrufende Interaktionsform überhaupt von der Veranstaltung angeboten?
- Wurde die Interaktion von der richtigen Interaktionsform aufgerufen?
- Ist die Interaktion von der richtigen Person gestartet worden? Im Falle einer Interaktion für die Vortragende bzw. den Vortragenden wird überprüft, ob die Aufruferin bzw. der Aufrufer der Interaktion auch wirklich die bzw. der Vortragende der Veranstaltung ist. Auf der Seite der Teilnehmerinnen und Teilnehmer wird untersucht, ob die Interaktion auch für die Aufrufende bzw. den Aufrufenden bestimmt ist.
- Ist der Zeitpunkt des Aufrufens der Interaktion plausibel? Dies bedeutet, dass eine Interaktion nur gültig ist, wenn sie nicht schon vorher einmal aufgerufen wurde oder noch nicht beantwortet wurde. Weiters kann der tatsächliche Startzeitpunkt einer Interaktion nur nach dem Geplanten liegen.

Wenn schließlich eine Django-App mit all den genannten Anforderungen und Erweiterungen umgesetzt wurde, ist die Implementation einer Interaktionsform abgeschlossen. Nun muss diese nur mehr für das Web-Portal verfügbar gemacht werden. Dazu sind die folgenden Schritte zu erledigen.

1. Aktivieren der Django-App in den globalen Einstellungen des Projekts.
2. Einbinden der URL-Konfiguration der App in die der *interactions*-App.
3. Eintragen der App in die Datenbank. Dazu kann eine Django-Fixture²² oder die Administrationsoberfläche (siehe Abschnitt 3.7) verwendet werden.

²² <https://docs.djangoproject.com/en/1.4/howto/initial-data/>, besucht am 23.10.2012

3 Design und Implementation

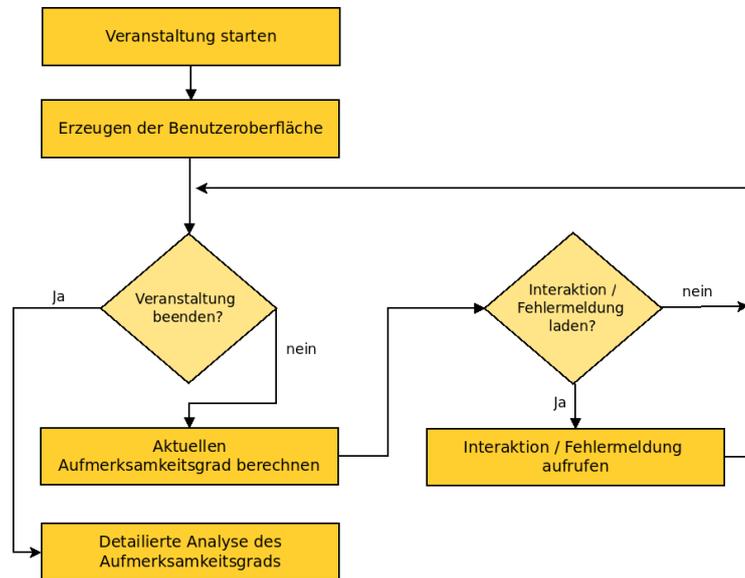


Abbildung 3.9: Aus der Perspektive der bzw. des Vortragenden zeigt sich dieser Ablauf einer Veranstaltung mit ihren Interaktionen

3.4.2 Der Ablauf

Wie bereits erwähnt, übernimmt die Interaktionen-Architektur die Steuerung des geregelten Ablaufs der Interaktionsformen. Dieser Ablauf ist einmal aus der Sicht der Vortragenden und ein anderes Mal aus der Perspektive der Teilnehmerinnen bzw. der Teilnehmer zu betrachten.

In Abbildung 3.9 ist der Ablauf aus Sicht der Vortragenden dargestellt. Nachdem die bzw. der Vortragende die Veranstaltung gestartet hat, wird eine passende Benutzeroberfläche erzeugt. Danach wird immer wieder der aktuelle Aufmerksamkeitsgrad der Teilnehmerinnen und Teilnehmer berechnet und falls Interaktionen anstehen, werden diese geladen und angezeigt. Dies geschieht solange, bis die bzw. der Vortragende die Veranstaltung beendet. Anschließend wird eine ausführliche Analyse der Aufmerksamkeit zur Verfügung gestellt.

Im Vergleich dazu zeigt Abbildung 3.10 den Ablauf aus der Sicht der Teilnehmerinnen und Teilnehmer. Auch hier wird mit der Teilnahme ei-

3.4 Die Interaktionen-Architektur

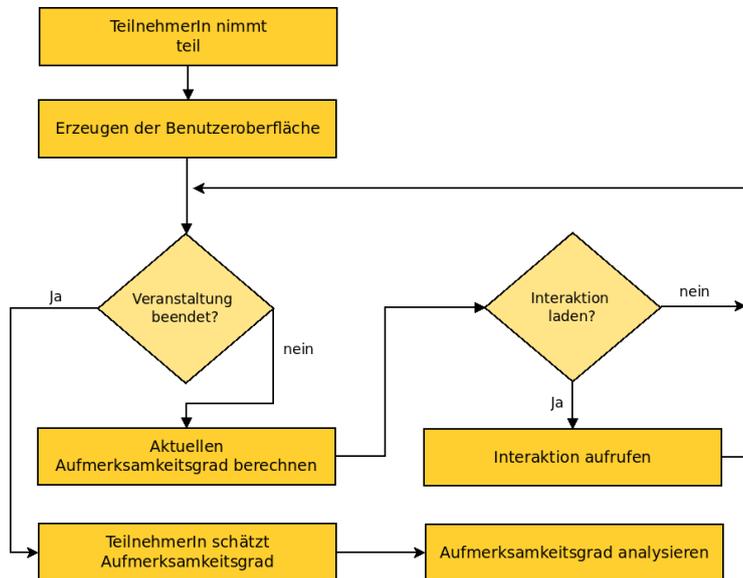


Abbildung 3.10: Der Ablauf der Interaktionen während einer Veranstaltung aus Sicht einer Teilnehmerin oder eines Teilnehmers

ne entsprechende Benutzeroberfläche generiert, und solange die Veranstaltung nicht beendet ist, wird ebenfalls nach anstehenden Interaktionen gesucht. Außerdem wird der Aufmerksamkeitsgrad der Teilnehmerin bzw. des Teilnehmers berechnet. Am Ende der Veranstaltung wird noch nach einer Schätzung des Aufmerksamkeitsgrads gefragt, bevor eine Analyse des tatsächlichen Aufmerksamkeitsgrads angezeigt wird.

Die Benutzeroberfläche der bzw. des Vortragenden integriert sich in die Hauptoberfläche (siehe Abbildung 3.11). Eine Auflistung einiger Daten zur Veranstaltung und eine Möglichkeit die Veranstaltung zu beenden (1) werden angezeigt. Daneben sieht man den aktuellen Aufmerksamkeitsgrad aller Teilnehmerinnen und Teilnehmer (2). Unterhalb befindet sich links (3) der Bereich in dem anfallende Interaktionen angezeigt werden. Auf der rechten Seite (4) werden die Templates, welche die von der bzw. dem Vortragenden auslösbaren Interaktionsformen bereitstellen, eingebunden, um die dafür notwendigen Bedienelemente anzuzeigen.

Eine komplett eigene Benutzeroberfläche gibt es für die Teilnehmerinnen

3 Design und Implementation

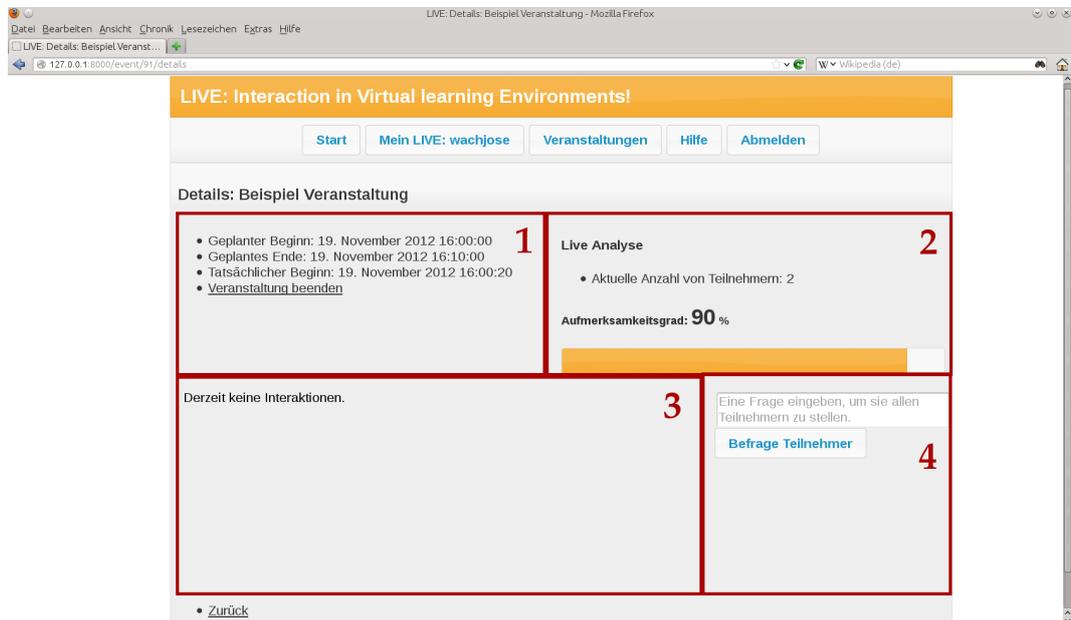


Abbildung 3.11: Ein Screenshot während einer Veranstaltung bei der bzw. dem Vortragenden zeigt das Design der Benutzeroberfläche

und Teilnehmer während einer Veranstaltung (siehe Abbildung 3.12). Der Großteil der Oberfläche dient zur Darstellung des fremden Web-Inhalts (1), welcher bei der Erstellung der Veranstaltung angegeben wurde und LIVE tritt nur mehr als Seitenleiste in Erscheinung. Dabei werden oben (2) einige Informationen zur Veranstaltung und eine kurze ampelartige Anzeige des eigenen aktuellen Aufmerksamkeitsgrads angezeigt. Darunter folgt ein Bereich zur Darstellung der unter Umständen anfallenden Interaktionen (3) sowie eine Möglichkeit zum vorzeitigen Verlassen der Veranstaltung. Wie auch bei der Oberfläche für die Vortragende bzw. den Vortragenden werden auch hier die Templates zum Auslösen einer Interaktionsform integriert (4).

Nach der Erstellung der Benutzeroberfläche startet in beiden Fällen ein immer wiederkehrender Ablauf, in dem der aktuelle Aufmerksamkeitsgrad berechnet wird und nach anstehenden Interaktionen gesucht wird. Für die Berechnung des Aufmerksamkeitsgrads kommt das Analyse-Framework

3.4 Die Interaktionen-Architektur

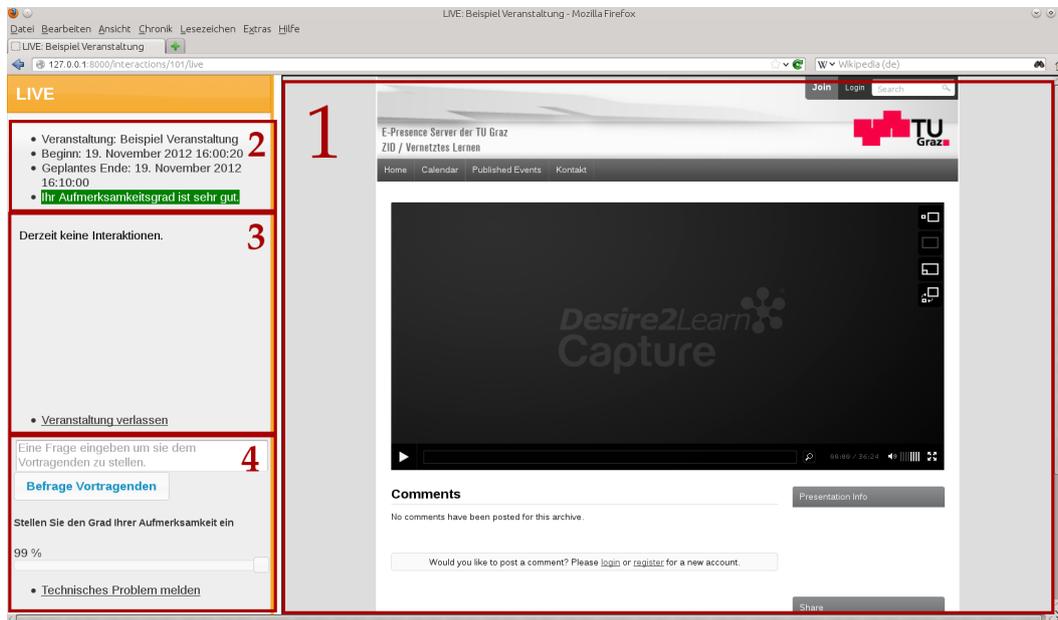


Abbildung 3.12: Das Aussehen der Benutzeroberfläche während einer Veranstaltung bei einer Teilnehmerin oder einem Teilnehmer zeigt dieser Screenshot

(siehe Abschnitt 3.5) zum Einsatz.

Im vorigen Abschnitt wurde erläutert, dass eine Interaktion durch Instanzieren von speziellen Modellen ausgelöst wird. Um nun nach anstehenden Interaktionen zu suchen, wird nach Interaktionsmodellen, welche eine der aktuellen Zeit entsprechende geplante Startzeit haben, gesucht. Gibt es einen Treffer, wird die entsprechende Interaktion geladen. Dazu wird eine View, welche die Interaktionsform speziell dafür anbieten muss, genutzt. Der einzige Unterschied bei dieser Suche nach Interaktionen zwischen der Seite der Vortragenden und der Teilnehmerinnen bzw. Teilnehmer besteht darin, dass unterschiedliche Modelle abgefragt werden (siehe Abschnitt 3.4.1).

Wurde die Veranstaltung von der bzw. dem Vortragenden beendet, werden die Teilnehmerinnen und Teilnehmer gebeten ihren durchschnittlichen Aufmerksamkeitsgrad während der Veranstaltung zu schätzen. Da während der Veranstaltung nur der aktuelle Aufmerksamkeitsgrad in Form ei-

3 Design und Implementation

ner Ampel angezeigt wird und somit keine genauen Zahlen preisgegeben werden, ist eine Schätzung des durchschnittlichen Aufmerksamkeitsgrads trotzdem möglich. Anschließend folgt eine ausführliche Analyse der Aufmerksamkeit während der Veranstaltung. Diese ist bereits wieder Teil der Veranstaltungsverwaltung (siehe Abschnitt 3.3.4) und wird vom Analyse-Framework bereitgestellt.

3.5 Das Analyse-Framework

Wie in den vorherigen Abschnitten bereits mehrfach erwähnt, kommt das sogenannte Analyse-Framework zum Einsatz, um eine Auswertung der Aufmerksamkeit während und nach einer Veranstaltung bereitzustellen. Dieses Framework wird vom Paket *interactions*, durch das Modul *analysis*, angeboten und setzt sich aus folgenden Komponenten zusammen:

- Eine Schnittstelle, welche die Interaktionsformen nutzen können, um in die Berechnung des Aufmerksamkeitsgrads einbezogen zu werden.
- Durchführung und Steuerung der Berechnung des Aufmerksamkeitsgrads während und nach einer Veranstaltung, sowohl für eine Teilnehmerin bzw. einen Teilnehmer als auch für alle.

Beim Aufmerksamkeitsgrad handelt es sich um eine Prozentzahl, welche den rechnerisch ermittelten Grad der Aufmerksamkeit ausdrückt.

Im Allgemeinen basiert die Berechnung des Aufmerksamkeitsgrads auf einer Auswertung der Reaktionszeiten auf Interaktionen. Dies bedeutet, dass die Aufmerksamkeit einer Teilnehmerin bzw. eines Teilnehmers sinkt, wenn auf eine Interaktion verspätet oder gar nicht geantwortet bzw. reagiert wird. Eine Interaktionsform kann jedoch auch eine andere Berechnungsmethode für ihre Interaktionen bereitstellen.

3.5.1 Der Ablauf

Das erwähnte Modul *interactions.analysis* stellt eine Klasse namens *AnalysisManager* bereit, um auf alle Funktionen des Analyse-Frameworks zuzugreifen, mit welcher die verschiedenen Aufmerksamkeitsgradberechnungen durchgeführt werden. Dazu bietet sie unter anderem folgende Methoden an:

- *__init__(event, ref_datetime=None)*
Mit dem Konstruktor wird die zu analysierende Veranstaltung und optional ein Referenzzeitpunkt übergeben.
- *get_attention_level(joined_user=None)*
Liefert während der Veranstaltung den aktuellen Aufmerksamkeitsgrad entweder für alle Teilnehmerinnen und Teilnehmer, oder, falls der Parameter übergeben wurde, den Aufmerksamkeitsgrad einer Teilnehmerin bzw. eines Teilnehmers zurück.
- *get_analysis()*
Berechnet nach der Veranstaltung den durchschnittlichen Aufmerksamkeitsgrad für jede Teilnehmerin und jeden Teilnehmer.

Während einer Veranstaltung wird nun also immer wieder die Methode *get_attention_level(...)* aufgerufen, um den aktuellen Aufmerksamkeitsgrad zu berechnen. Der Ablauf dieser Berechnung ist in Quellcode 3.9 schematisch dargestellt. Aufmerksamkeit während der Veranstaltung

Zu Beginn des Pseudocodes wird eine Variable (a) zum Akkumulieren des Aufmerksamkeitsgrads auf 0 initialisiert und die Variable n enthält die Anzahl der Interaktionsformen der Veranstaltung. Danach iteriert eine Schleife (Zeilen 5-10) über diese Interaktionsformen und lädt die Klasse *LiveAnalysis* der Interaktionsform, welche eine entsprechende Schnittstelle implementiert (siehe Abschnitt 3.5.2). Diese übernimmt die tatsächliche Berechnung für ihre Interaktionen und liefert einen Wert zwischen 0% und 100% zurück, welcher auf die Variable a addiert wird. Kommt es dabei zu einem Fehler und es wurde nichts addiert, wird n dekrementiert.

Am Ende wird nun noch der akkumulierte Aufmerksamkeitsgrad aller Interaktionsformen (a) durch die Anzahl der gültigen Interaktionsformen (n)

3 Design und Implementation

```
1 FUNKTION Berechnung_des_Live_Aufmerksamkeitsgrads
2     a = 0 // der Aufmerksamkeitsgrad
3     n = Anzahl der Interaktionsformen der Veranstaltung
4
5     FUER ALLE Interaktionsformen der Veranstaltung
6         VERSUCHE
7             k = LiveAnalysis der Interaktionsform
8             a = a + k->berechne_aufmerksamkeitsgrad
9         SONST
10            n = n - 1
11
12     WENN n > 0 DANN
13         a = a / n
14     SONST
15         a = 0
16     RETURN a
```

Quellcode 3.9: Ablauf der Berechnung des Aufmerksamkeitsgrads während einer Veranstaltung in Pseudocode

dividiert. Diese geschieht natürlich nur, wenn eine Division ($n > 0$) überhaupt möglich ist.

Aufmerksamkeit
nach der
Veranstaltung

Wie bereits mehrfach erwähnt, erhält die bzw. der Vortragende nach der Veranstaltung eine Auflistung der Aufmerksamkeitsgrade aller Teilnehmerinnen und Teilnehmer. Diese Auflistung wird von `get_analysis()` generiert. Die pseudocodeartige Darstellung des Ablaufs der Berechnung in Quellcode 3.10 lässt Gemeinsamkeiten mit jener während der Veranstaltung erkennen.

Die Variable `n` enthält wieder die Anzahl der Interaktionsformen der Veranstaltung und `a` ist nun ein Python-Dictionary, welches jeder Teilnehmerin und jeden Teilnehmer ihren Aufmerksamkeitsgrad zuordnet. In den Zeilen 5 und 6 wird jeder Aufmerksamkeitsgrad nun auf 0 initialisiert. Anschließend wird für jede Interaktionsform der Veranstaltung die Klasse *Analysis* geladen (siehe Abschnitt 3.5.2), um mit ihrer Hilfe die Berechnung durchzuführen. Tritt dabei ein Fehler auf, wird auch hier die Variable `n` um 1

```

1 FUNKTION Berechnung_der_Aufmerksamkeitsgrade
2   a = Dict(Teilnehmer_ID, Aufmerksamkeitsgrad)
3   n = Anzahl der Interaktionsformen der Veranstaltung
4
5   FUER ALLE Teilnehmer
6     a[Teilnehmer_ID] = 0
7
8   FUER ALLE Interaktionsformen der Veranstaltung
9     VERSUCHE
10      k = Analysis der Interaktionsform
11      k->berechne_aufmerksamkeitsgrade(a)
12    SONST
13      n = n - 1
14
15  FUER ALLE Teilnehmer
16    WENN n > 0 DANN
17      a[Teilnehmer_ID] = a[Teilnehmer_ID] / n
18      WENN Teilnehmer zu spaet teilnahm
19        KORRIGIERE a[Teilnehmer_ID]
20    SONST
21      a[Teilnehmer_ID] = 0
22  RETURN a

```

Quellcode 3.10: Ablauf der Berechnung des Aufmerksamkeitsgrads jeder Teilnehmerin bzw. jedes Teilnehmers nach einer Veranstaltung in Pseudocode

verringert.

Schließlich wird noch einmal über die Aufmerksamkeitsgrade der Teilnehmerinnen und Teilnehmer (a) iteriert, um diese jeweils durch n zu dividieren. Dabei wird, falls die Teilnehmerin bzw. der Teilnehmer zu spät zur Veranstaltung gekommen ist, der Aufmerksamkeitsgrad entsprechend korrigiert. Der Fall, dass die Veranstaltung zu früh verlassen wurde, muss nicht gesondert behandelt werden, da dabei ohnehin Interaktionen versäumt wurden und somit ein entsprechend niedriger Aufmerksamkeitsgrad berechnet wurde.

3 Design und Implementation

Zusammenfassend lässt sich der Ablauf der Berechnung des Aufmerksamkeitsgrads auch in einer allgemeinen Form durch eine Formel beschreiben. Diese spiegelt auch die Darstellung der Berechnung in den beiden Pseudocodes wieder:

$$a = \frac{\sum_{i=0}^n t_i}{n} =$$

$$\text{Aufmerksamkeitsgrad} = \frac{\sum \text{Aufmerksamkeitsgrad einer Interaktionsform}}{\text{Anzahl der gültigen Interaktionsformen}}$$

3.5.2 Die Schnittstelle

Im vorherigen Abschnitt wurde erläutert, dass die eigentliche Berechnung des Aufmerksamkeitsgrads von den Interaktionsformen selbst durchgeführt wird. Dazu müssen sie bestimmte Schnittstellen implementieren. Diese Implementationen sollen die Interaktionsformen in einem eigenen Modul namens *analysis* bereitstellen.

Aufmerksamkeit während der Veranstaltung Um nun eine Implementation der Berechnung des Aufmerksamkeitsgrads während der Veranstaltung anzubieten, muss eine Klasse mit dem Namen *LiveAnalysis*, welche von der Klasse *GenericLiveAnalysis* des Frameworks abgeleitet ist, zur Verfügung gestellt werden.

Zur Berechnung des Aufmerksamkeitsgrads muss zumindest die Methode *compute_attention_level(joined_user=None)* implementiert werden. Diese soll eine Prozentzahl zwischen 0% und 100% zurück liefern und Auskunft geben, wie hoch die Aufmerksamkeit aller Teilnehmerinnen und Teilnehmer im Kontext der Interaktionsform ist. Wird allerdings der Parameter übergeben, so ist die Berechnung nur für diese Teilnehmerin bzw. diesen Teilnehmer durchzuführen.

Aufmerksamkeit nach der Veranstaltung Ähnlich gestaltet sich auch das Vorgehen für die Analyse nach einer Veranstaltung. Dazu muss die Interaktionsform eine von der Klasse *GenericAnalysis* des Frameworks abgeleitete Klasse, namens *Analysis*, anbieten. Diese

Klasse erhält durch den Konstruktor unter anderem das oben erwähnte Dictionary zum Speichern der Aufmerksamkeitsgrade aller Teilnehmerinnen und Teilnehmer.

In der Methode *analyze()* wird nun diese Berechnung durchgeführt. Dabei sollen die Aufmerksamkeitsgrade der Teilnehmerinnen und Teilnehmer im übergebenen Dictionary auf die jeweilige Position addiert werden. Schlussendlich ist dieses Dictionary von der Methode zu retournieren.

Anstatt eine eigene Berechnungsmethode zu implementieren, kann auch die im folgenden Abschnitt erläuterte Variante eingesetzt werden.

3.5.3 Berechnung des Aufmerksamkeitsgrads auf Basis der Reaktionszeiten

Die meisten Interaktionsformen stützen die Berechnung des Aufmerksamkeitsgrads auf die Auswertung der Reaktionszeiten der Interaktionen. Aus diesem Grund stellen die Klassen *GenericLiveAnalysis* und *GenericAnalysis* jeweils eine Methode bereit, welche diesen Ansatz implementieren.

Im Abschnitt 3.4.1 wurde bereits das Modell zum Speichern einer Interaktion vorgestellt. Dieses enthält neben der geplanten und der tatsächlichen Startzeit der Interaktion auch den Zeitpunkt der Antwort bzw. Reaktion. Auf diesen Daten basiert nun die Berechnung des Aufmerksamkeitsgrads.

Für eine einzelne Interaktion bedeutet dies, dass für eine Reaktion bis zu einem bestimmten Zeitpunkt (*SUCCESS_UNTIL*) nach der Startzeit eine Aufmerksamkeit von 100% angenommen wird. Weiters wird bei einer Reaktion ab einem späteren Zeitpunkt (*FAILED_AFTER*) von einer Aufmerksamkeit von 0% ausgegangen. Im Bereich dazwischen wird der Aufmerksamkeitsgrad linear interpoliert. Dieser Sachverhalt ist in Abbildung 3.13 dargestellt. Somit wird ab dem tatsächlichen Start der Interaktion, welcher maximal 60 Sekunden nach der geplanten Startzeit liegen darf, der Aufmerksamkeitsgrad der Interaktion, ausgehend vom tatsächlichen Startzeitpunkt, berechnet.

3 Design und Implementation

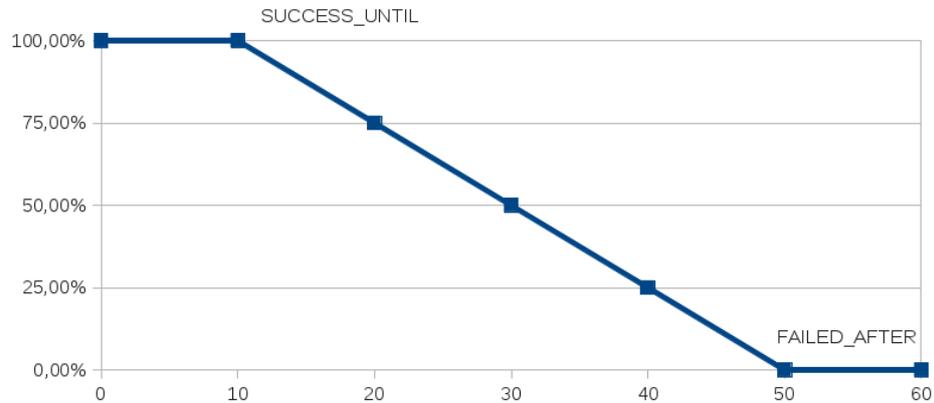


Abbildung 3.13: Verlauf des Aufmerksamkeitsgrads einer Interaktion auf Basis der Reaktionszeit

Um nun den durchschnittlichen Aufmerksamkeitsgrad für einen bestimmten Zeitraum zu berechnen, werden für die hineinfließenden Interaktionen die Aufmerksamkeitsgrade nach diesem Schema berechnet und aufsummiert. Eine Interaktion fällt in einen Zeitraum, wenn die geplante Startzeit in diesem liegt. Schließlich erfolgt noch die Bildung des Mittelwerts durch die Division durch die Anzahl der Interaktionen des Zeitraums. Dies führt zur folgenden allgemeinen Formel zur Berechnung des Aufmerksamkeitsgrads einer Interaktionsform auf Basis der Reaktionszeiten:

$$\text{Aufmerksamkeitsgrad} = \frac{\sum \text{Aufmerksamkeitsgrad einer Interaktion}}{\text{Anzahl der gültigen Interaktionen}}$$

Im Falle der Analyse nach der Veranstaltung erstreckt sich der genannte Zeitraum auf die gesamte Veranstaltungsdauer. Handelt es sich hingegen um die Berechnung des aktuellen Aufmerksamkeitsgrads während einer Veranstaltung, so wird nur ein kurzer Zeitraum (*MAX_LOOKBACK_TIME*) vor der aktuellen Zeit berücksichtigt.

Soll diese Art der Berechnung des Aufmerksamkeitsgrads von einer Interaktionsform verwendet werden, so muss in der Implementation der Be-

3.5 Das Analyse-Framework

Einstellung	Standard
Beschreibung	
MAX_LOOKBACK_TIME	300
Gibt an, wie viele Sekunden die Berechnung des Aufmerksamkeitsgrads während einer Veranstaltung zurückblicken soll.	
LIVE_SUCCESS_UNTIL und SUCCESS_UNTIL	10
Bestimmt, wie viele Sekunden Reaktionszeit auf eine Interaktion noch 100% Aufmerksamkeit bringen.	
LIVE_FAILED_AFTER und FAILED_AFTER	50
Legt fest, nach wie vielen Sekunden Reaktionszeit auf eine Interaktion nur mehr 0% Aufmerksamkeit verrechnet wird.	

Tabelle 3.2: Die Analyse-Einstellungen für eine Interaktionsform

rechnung nur die entsprechende Methode aufgerufen werden (siehe Abschnitt 3.5.2). Dabei können die in Tabelle 3.2 aufgelisteten Einstellungsoptionen zur Parametrisierung genutzt werden. Auf die verschiedenen Einstellungsmöglichkeiten wurde im Laufe dieses Abschnitts bereits Bezug genommen. Das Präfix *LIVE_* mancher Einstellungen bedeutet, dass dieser Parameter für die Berechnung des Aufmerksamkeitsgrads während der Veranstaltung Verwendung findet.

Quellcode 3.11 zeigt die Verwendung am Beispiel der Analyse während der Veranstaltung (*compute_attention_level(...)*). Man sieht, dass dabei nur eine vorgefertigte Methode (*time_based_attention_level(...)*) aufgerufen wird und die entsprechenden Einstellungsoptionen als Parameter übergeben werden.

Die im nächsten Abschnitt beschriebenen Interaktionsformen setzen teilweise die Methode zur Berechnung der Aufmerksamkeitsgrade ein. Dabei werden die jeweiligen Werte der Einstellungsmöglichkeiten näher erläutert. Grundsätzlich kamen diese Werte dadurch zustande, dass sie zuerst geschätzt und im Laufe der verschiedenen Einsätze zur Evaluation durch die Analyse der Reaktionszeiten auf die Interaktionen verfeinert wurden.

3 Design und Implementation

```
1 class LiveAnalysis(GenericLiveAnalysis):
2
3     def compute_attention_level(self, joined_user=None):
4         return self.time_based_attention_level(
5             max_lookback=MAX_LOOKBACK_TIME,
6             success_until=LIVE_SUCCESS_UNTIL,
7             failed_after=LIVE_FAILED_AFTER,
8             joined_user=joined_user
9         )
```

Quellcode 3.11: Beispielhafter Einsatz der Aufmerksamkeitsgradberechnung auf Basis der Reaktionszeiten

3.6 Die Interaktionsformen

Nachdem nun alle grundlegenden Funktionen des Web-Portals, die Interaktionen-Architektur und das Analyse-Framework detailliert erläutert wurden, dokumentiert dieser Abschnitt die auf die dadurch bereitgestellten Schnittstellen aufsetzenden Interaktionsformen. Es existieren zurzeit folgende automatisch ausgeführte Interaktionsformen:

- **Einfache Fragen**
Kurze allgemeine Fragen werden den Teilnehmerinnen und Teilnehmern automatisch gestellt.
- **Löse Captcha**
Ein einfaches Captcha wird den Teilnehmerinnen und Teilnehmern zum Lösen angezeigt.

Dazu kommen einige Interaktionsformen, welche von den Teilnehmerinnen und Teilnehmern gestartet werden können:

- **Befrage Vortragenden**
Die Teilnehmerinnen und Teilnehmer können eine Frage an die Vortragende bzw. den Vortragenden stellen und gegebenenfalls die Antwort lesen.

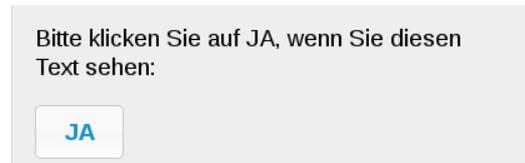


Abbildung 3.14: Eine einfache automatische Frage

- **Aufmerksamkeit einstellen**

Der Grad der Aufmerksamkeit kann von den Teilnehmerinnen und Teilnehmern auch selbst mitbestimmt werden.

- **Technisches Problem melden**

Die Teilnehmerinnen und Teilnehmer erhalten die Möglichkeit, ein technisches Problem an die Vortragende bzw. den Vortragenden zu berichten.

Außerdem gibt es auch eine Interaktionsform, die von der bzw. dem Vortragenden in Gang gesetzt werden kann:

- **Befrage Teilnehmer**

Die bzw. der Vortragende erhält die Möglichkeit, eine Frage an die Teilnehmerinnen und Teilnehmer zu richten und kann die Antworten lesen.

3.6.1 Einfache Fragen

Diese erste und zugleich auch einfachste Interaktionsform stellt den Teilnehmerinnen und Teilnehmern eine einfache Frage und wird automatisch ausgelöst. Diese Frage soll durch einen Klick auf einen Button beantwortet werden. Die Interaktionsform präsentiert sich den Teilnehmerinnen und Teilnehmern in der in Abbildung 3.14 gezeigten Form. Aufgrund ihrer Einfachheit kann diese Interaktionsform auch als beispielhafte Implementation der Schnittstellen angesehen werden.

3 Design und Implementation

```
1 @login_required
2 def index(request, interaction_id):
3     im = get_object_or_404(Interaction,
4         id=interaction_id,
5         joined_user__student=request.user,
6         real_start__isnull=False,
7         responded_at__isnull=True
8     )
9     verify_interaction(
10        interaction=im,
11        interaction_method='Simple_Questions',
12        user=request.user
13    )
14
15    questions = [
16        (_('Please_confirm_your_attendance:'),
17         _('Confirm')),
18        (_('Please_click_Yes_if_you_see_this_text:'),
19         _('Yes'))
20    ]
21    rn = random.randint(0, (len(questions)-1))
22    params = {
23        'interaction' : im,
24        'question' : questions[rn][0],
25        'button' : questions[rn][1]
26    }
27    return render_to_response(
28        'interactions/simple/ask.html', params,
29        context_instance=RequestContext(request))
```

Quellcode 3.12: Diese spezielle View übernimmt die Darstellung der Frage

3.6 Die Interaktionsformen

Da es sich um eine automatisch ausgelöste Interaktionsform handelt, werden, wie im Abschnitt 3.4.1 beschrieben, die Interaktionen zu einem zufälligen Zeitpunkt gestartet. Es wurde festgelegt, dass die Interaktionsform zehnmal pro Stunde ausgeführt wird. Damit die Fragen nun angezeigt werden können, ist die Implementation einer speziellen View erforderlich (siehe Quellcode 3.12). Es wird zuerst die entsprechende Interaktion geladen und mit der zur Verfügung stehenden Funktion verifiziert. Danach wird zufällig eine Frage ausgewählt und zur Darstellung mit einem Template kombiniert. Diese Implementation kann repräsentativ für eine Interaktion angesehen werden, da das Laden und Verifizieren der Interaktion sowie das Einbinden eines Templates von den meisten solchen Views durchgeführt werden muss.

Wird die Interaktion durch Anklicken des Buttons bestätigt, kommt eine weitere View zur Ausführung. Diese lädt und verifiziert wieder die Interaktion und setzt den Zeitpunkt der Antwort im entsprechenden Modell.

Da diese Interaktionsform auch Teil der Analyse der Aufmerksamkeit ist, wurden die entsprechenden Module und Klassen implementiert (siehe Abschnitt 3.5). Dabei kommt auch die vorgefertigte Implementation der auf den Reaktionszeiten basierenden Analyse zum Einsatz. Diese wurde so konfiguriert, dass die Interaktionen der letzten 600 Sekunden bei der Analyse während einer Veranstaltung miteinbezogen werden. Weiters führt eine Reaktionszeit von 10 Sekunden noch zu keinen Abzügen bei der Aufmerksamkeit. Liegt die Antwort jedoch über 50 Sekunden nach der Startzeit, wird nur mehr ein Aufmerksamkeitsgrad von 0% verrechnet.

3.6.2 Löse Captcha

Diese Interaktionsform ist der zuvor beschriebenen sehr ähnlich, denn es werden ebenfalls automatisch kleine Aufgaben an die Teilnehmerinnen und Teilnehmer gestellt. Im Gegensatz zu den Fragen, müssen hier aber einfache Captchas²³ gelöst werden. Auch diese Interaktionsform tritt in zufälligen Abständen zehnmal pro Stunde in Erscheinung.

²³ kurz für »Completely Automated Public Turing Test to Tell Computers and Humans Apart«

3 Design und Implementation

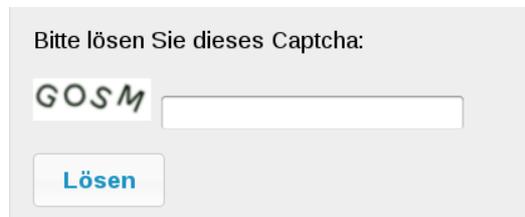


Abbildung 3.15: Lösen eines einfachen Captchas

Unter dem Begriff Captcha versteht man ein Programm, welches Aufgaben erstellt, die nur von einem Menschen gelöst werden können und nicht durch einen aktuellen Computer [YEAo8]. Üblicherweise handelt es sich dabei um in ein Bild gerenderte Buchstaben, welche in ein Formularfeld eingegeben werden müssen. Diese Methode wird vor allem von Web-Formularen eingesetzt, um zu verhindern, dass dieses Formular missbräuchlich von einem Computerprogramm verwendet wird.

In Abbildung 3.15 ist das Captcha im Rahmen einer Interaktion zu sehen. Die Captchas bestehen immer aus vier einfachen Buchstaben, welche nur leicht gedreht dargestellt werden. Es wird kein Hintergrundrauschen oder eine ähnliche Erschwerung verwendet, da das Captcha in diesem Fall keine Sicherheitsfunktion hat, sondern zur Erhaltung und Erhöhung der Aufmerksamkeit eingesetzt wird. Diese Vereinfachungen wurden gewählt, damit das Captcha nicht einen gegenteiligen und somit negativen Einfluss auf die Aufmerksamkeit ausübt.

Es wurde eine spezielle View zur Generierung und Überprüfung des Captchas implementiert. Diese lädt und verifiziert die Interaktion und benutzt das Django-Modul »Simple-Captcha«²⁴. Dieses Modul wurde entsprechend den oben genannten Anforderungen konfiguriert.

Als Grundlage für die Analyse der Aufmerksamkeit bei dieser Interaktionsform dient wiederum die Auswertung der Reaktionszeiten auf die Interaktionen. Dazu wurde festgelegt, dass Interaktionen der letzten 600 Sekunden Teil der Analyse während einer Veranstaltung sind. Im Allgemeinen gilt, dass eine Interaktion bis zu einer Reaktionszeit von 15 Sekunden

²⁴ <https://github.com/mbi/django-simple-captcha>, besucht am 01.11.2012

einen Aufmerksamkeitsgrad von 100% erzeugt und ab 60 Sekunden wird von einer versäumten Interaktion ausgegangen.

3.6.3 Befrage Teilnehmer

Diese Interaktionsform ermöglicht es der bzw. dem Vortragenden eine Frage an die Teilnehmerinnen und Teilnehmer zu stellen, so wie im Rahmen einer Präsenzvorlesung in einem Hörsaal. Die gestellte Frage wird den Teilnehmerinnen und Teilnehmern durch eine Interaktion dargestellt und bietet die Möglichkeit, diese zu beantworten. Die Antworten kommen unmittelbar nachdem sie gegeben wurden bei der bzw. dem Vortragenden zur Darstellung, welche bzw. welcher nach der Fragerunde selbst eine Referenzantwort geben kann.

Damit die bzw. der Vortragende die Fragerunde starten kann, wird ein Template zur Darstellung der erforderlichen Bedienelemente bereitgestellt und durch die entsprechende Einstellungsmöglichkeit in die Benutzeroberfläche integriert. Bei den Bedienelementen handelt es sich um ein Textfeld zum Eingeben der Frage und um einen Button zum Absenden (siehe Abbildung 3.16).

Frage stellen

Durch das Absenden der Frage wird eine View aufgerufen, welche Interaktionen für alle Teilnehmerinnen und Teilnehmer erstellt und diese mit dem in Quellcode 3.13 dargestellten Modell verknüpft (*interactions*). Außerdem wird eine Interaktion für die Vortragende bzw. den Vortragenden erstellt und ebenfalls in das Modell eingetragen (*teacherinteraction*), um die View zum Anzeigen der Antworten aufzurufen. Neben diesen Verknüpfungen zu den Interaktionen enthält das Modell natürlich die Frage (*question*) und gegebenenfalls auch die Referenzantwort (*answer*).

Durch die Interaktionen wird die Frage bei den Teilnehmerinnen und Teilnehmern angezeigt (siehe Abbildung 3.17). Dadurch wird auch die Möglichkeit zur Beantwortung der Frage bereitgestellt. Erfolgt eine Antwort, wird in der entsprechenden View die Antwort (*answer*) mit Hilfe eines Modells (siehe Quellcode 3.14) gespeichert. Dabei wird natürlich auch eine Verbindung zur Frage (*question*) und zur Teilnehmerin bzw. zum Teilnehmer (*joined_user*) hergestellt. Diese Antworten werden nun von einer sich

Frage
beantworten

3 Design und Implementation



Abbildung 3.16: Eine Frage an alle Teilnehmerinnen und Teilnehmer stellen

```
1 class QuestionToStudents(models.Model):  
2     teacherinteraction = models.OneToOneField(  
3         TeacherInteraction)  
4     interactions = models.ManyToManyField(Interaction)  
5  
6     question = models.TextField()  
7     answer = models.TextField(blank=True, null=True)
```

Quellcode 3.13: Dieses Modell speichert eine Frage der bzw. des Vortragenden an die Teilnehmerinnen und Teilnehmer

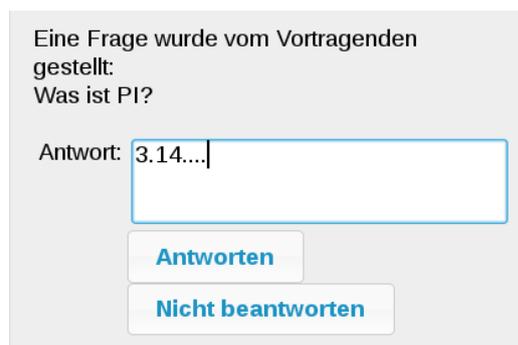


Abbildung 3.17: Die Teilnehmerinnen und Teilnehmer können die Frage beantworten

3.6 Die Interaktionsformen

```
1 class AnswerFromStudents(models.Model):  
2     question = models.ForeignKey(QuestionToStudents)  
3     joined_user = models.ForeignKey(JoinedUser)  
4     answer = models.TextField()
```

Quellcode 3.14: Die Antwort einer Teilnehmerin bzw. eines Teilnehmers wird mit diesem Modell gespeichert

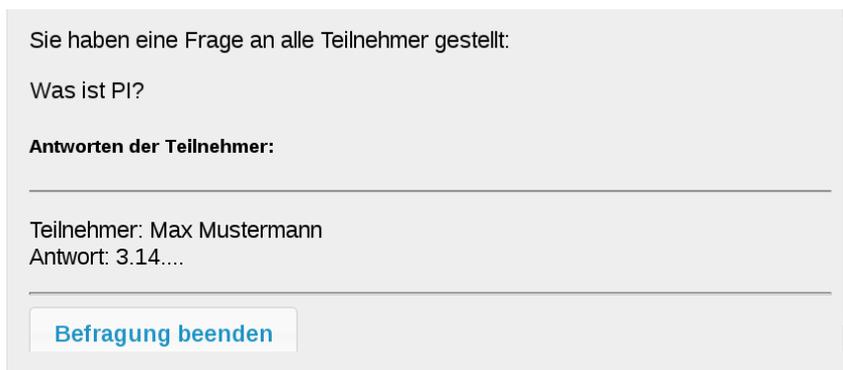


Abbildung 3.18: Die bzw. der Vortragende erhält die Antworten

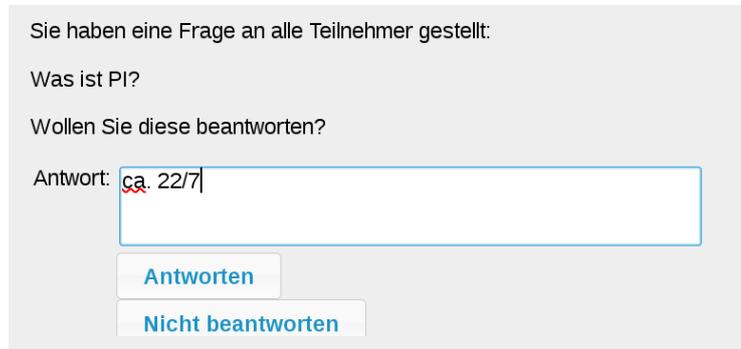
immer wieder aktualisierenden View, welche durch die beim Stellen der Frage erstellte Interaktion aufgerufen wurde, bei der bzw. dem Vortragenden dargestellt (siehe Abbildung 3.18).

Nachdem die bzw. der Vortragende die Fragerunde beendet hat, kann sie bzw. er mit dem in Abbildung 3.19 dargestellten Formular eine Referenzantwort geben. Diese wird im oben vorgestellten Modell der Frage gespeichert und durch Interaktionen den Teilnehmerinnen und Teilnehmern in der gleichen Form angezeigt wie die Frage vorhin.

Damit die während einer Veranstaltung gestellten Fragen und erhaltenen Antworten auch nach der Veranstaltung noch einsehbar sind, wurde eine entsprechende View implementiert. Diese wird in den Einstellungen der Interaktionsform als Detail-View gekennzeichnet und somit auf der Analyseseite der bzw. des Vortragenden verlinkt.

Auch bei dieser Interaktionsform basiert die Berechnung der Aufmerksam-

3 Design und Implementation



Sie haben eine Frage an alle Teilnehmer gestellt:

Was ist PI?

Wollen Sie diese beantworten?

Antwort:

Abbildung 3.19: Die bzw. der Vortragende kann auch eine Referenzantwort geben

keit auf den Reaktionszeiten. Neben dem 600 Sekunden großen Analyseintervall während der Veranstaltung wird in den Einstellungen auch festgelegt, dass bis zu 25 Sekunden Reaktionszeit noch eine Aufmerksamkeit von 100% und ab 90 Sekunden Reaktionszeit 0% Aufmerksamkeit verrechnet werden.

3.6.4 Befrage Vortragenden

Im Gegensatz zur vorhin beschriebenen Interaktionsform, deckt diese den umgekehrten Fall ab. Dies bedeutet, dass durch diese Interaktionsform eine Möglichkeit geschaffen wird, durch welche die Teilnehmerinnen und Teilnehmer eine Frage an die Vortragende bzw. den Vortragenden stellen können. Die Frage kann natürlich auch beantwortet werden.

Da diese Interaktionsform von den Teilnehmerinnen bzw. Teilnehmern ausgelöst wird, bietet ein Template die dafür notwendigen Bedienelemente (ein Textfeld für die Frage und einen Button zum Absenden) an. Dieses Template wird durch eine entsprechende Erwähnung in den Einstellungen in die Benutzeroberfläche eingebunden (siehe Abbildung 3.20).

Frage stellen Nachdem die Frage abgesendet wurde, speichert eine View diese (*question*) mit dem in Quellcode 3.15 dargestellten Modell. Dabei wird auch die fragende Teilnehmerin bzw. der fragende Teilnehmer (*asking_joined_user*)

Abbildung 3.20: Die Teilnehmerinnen und Teilnehmer können eine Frage an die Vortragende bzw. den Vortragenden richten

```

1 class Question(models.Model):
2     asking_joined_user = models.ForeignKey(JoinedUser)
3     teacherinteraction = models.OneToOneField(
4         TeacherInteraction)
5     interactions = models.ManyToManyField(Interaction)
6
7     question = models.TextField()
8     answer = models.TextField(blank=True, null=True)

```

Quellcode 3.15: Dieses Modell speichert eine Frage einer Teilnehmerin bzw. eines Teilnehmers an die Vortragende bzw. den Vortragenden

eingetragen. In der View wird zusätzlich auch eine Interaktion (*teacherinteraction*) erstellt und in das Modell eingetragen, um die Frage bei der bzw. dem Vortragenden anzuzeigen. In der gleichen Art und Weise werden auch Interaktionen für die Teilnehmerinnen und Teilnehmer erstellt und eingetragen (*interactions*), damit auch sie von der Frage Notiz nehmen (siehe Abbildung 3.21).

Durch die erwähnte Interaktion wird nun eine View aufgerufen, welche sich um die Darstellung der Frage und das Anbieten eines Formulars zur Beantwortung bei der bzw. dem Vortragenden kümmert (siehe Abbildung 3.22). Dazu lädt und verifiziert sie die Interaktion und stellt die Frage dar oder speichert die Antwort (*answer*), falls das Antwortformular abgeschickt wurde, im Modell und erstellt weitere Interaktionen, mit welchen diese Antwort bei allen Teilnehmerinnen und Teilnehmern zur Darstellung kommt. Diese Darstellung der Antworten wird in Abbildung 3.23 illustriert und von der gleichen View, welche auch das Anzeigen der Frage übernimmt, durchgeführt.

Frage
beantworten

3 Design und Implementation

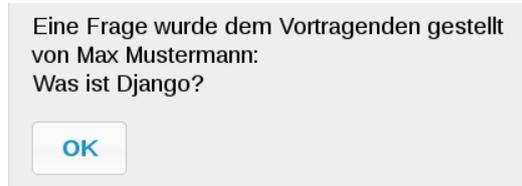


Abbildung 3.21: Die Frage wird bei allen Teilnehmerinnen und Teilnehmern angezeigt

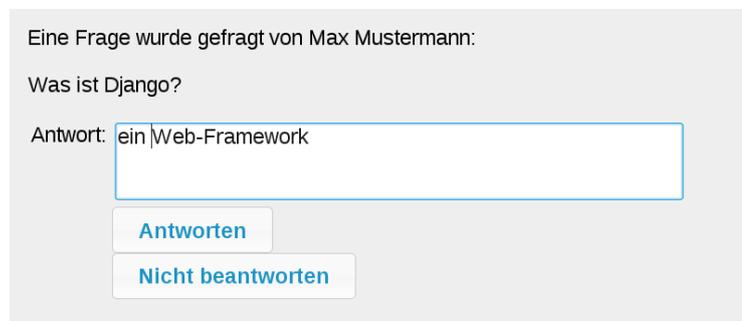


Abbildung 3.22: Die bzw. der Vortragende kann die Frage beantworten

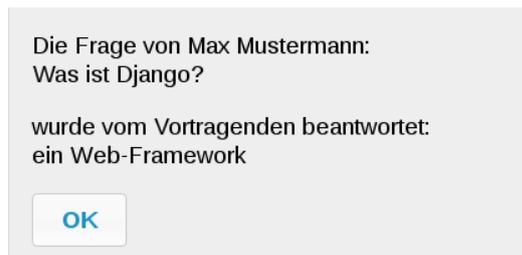


Abbildung 3.23: Die Antwort wird ebenfalls bei allen Teilnehmerinnen und Teilnehmern dargestellt

3.6 Die Interaktionsformen

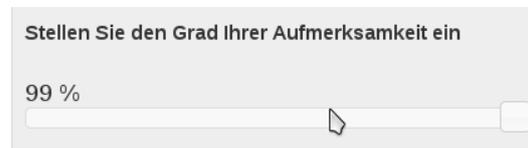


Abbildung 3.24: Die Teilnehmerinnen und Teilnehmer können auch selbst ihren Aufmerksamkeitsgrad einstellen

Wie auch bei der vorigen Interaktionsform ist es auch hier sinnvoll, dass die Fragen und Antworten auch nach der Veranstaltung noch von der bzw. dem Vortragenden eingesehen werden können. Dazu wurde eine entsprechende View implementiert und diese in den Einstellungen als Detail-View angegeben.

Auch diese Interaktionsform ist Teil der Analyse der Aufmerksamkeit. Da sie sehr große Ähnlichkeit mit der vorherigen Interaktionsform besitzt, wurde auch hier die Analyse auf Basis der Reaktionszeiten eingesetzt und mit den gleichen Werten parametrisiert (siehe Abschnitt 3.6.3).

3.6.5 Aufmerksamkeit einstellen

Eine weitere Interaktionsform, welche von den Teilnehmerinnen und Teilnehmern ausgelöst werden kann, bietet die Möglichkeit, mit Hilfe eines Schiebereglers den aktuellen Grad der Aufmerksamkeit selbst einzustellen. Dieser Schieberegler ist in Abbildung 3.24 dargestellt und soll von den Teilnehmerinnen und Teilnehmern dazu genutzt werden, auch selbst einen Wert für die Aufmerksamkeitsanalyse bereitzustellen, damit die bzw. der Vortragende eine genauere Abschätzung über die Aufnahmefähigkeit erhält.

Der Schieberegler wird durch ein Template, welches in den Einstellungen der Interaktionsform bekannt gemacht wird, in die Benutzeroberfläche der Teilnehmerinnen und Teilnehmer eingebunden. Bei jeder Verschiebung des Schiebereglers wird eine View aufgerufen, mit der das in Quellcode 3.16

3 Design und Implementation

```
1 class SetAttention(models.Model):
2     joined_user = models.ForeignKey(JoinedUser)
3
4     attention_level = models.IntegerField()
5     triggerd_at = models.DateTimeField()
```

Quellcode 3.16: Dieses Modell speichert einen eingestellten Aufmerksamkeitsgrad einer Teilnehmerin bzw. eines Teilnehmers

präsentierte Modell verwendet wird, um die Veränderung des Aufmerksamkeitsgrad zu speichern. Dabei wird neben dem eingestellten Aufmerksamkeitsgrad (*attention_level*) und der Teilnehmerin bzw. dem Teilnehmer (*joined_user*) auch der aktuelle Zeitpunkt (*triggerd_at*) eingetragen.

Da diese Interaktionsform auf keinen Interaktionen beruht, kommt im Gegensatz zu den bisher beschriebenen Interaktionsformen auch nicht die vorgefertigte Analyse der Aufmerksamkeit auf Basis der Reaktionszeiten zum Einsatz. Stattdessen fließt bei der Analyse während der Veranstaltung der jeweils letztgesetzte Aufmerksamkeitsgrad in die Berechnung ein. Dies bedeutet im Fall der Analyse für eine Teilnehmerin bzw. einen Teilnehmer, dass dieser letztgesetzte Aufmerksamkeitsgrad auch das Ergebnis ist. Im Fall der Analyse für alle Teilnehmerinnen und Teilnehmer, werden all ihre letztgesetzten Aufmerksamkeitsgrade addiert und durch Division durch die Anzahl der Teilnehmerinnen und Teilnehmer, welche somit der Anzahl der gesetzten Aufmerksamkeitsgrade entspricht, der Mittelwert gebildet.

Bei der Analyse nach der Veranstaltung werden alle gesetzten Aufmerksamkeitsgrade berücksichtigt. Diese werden ebenfalls addiert und durch die Anzahl der gesetzten Aufmerksamkeitsgrade dividiert. Dieser Mittelwert wird für jede Teilnehmerin bzw. jeden Teilnehmer gebildet.

Somit ist der von den Teilnehmerinnen und Teilnehmern jeweils manuell gesetzte Grad Teil der Aufmerksamkeitsanalyse. Das bedeutet, dass diese Werte, im Rahmen der erwähnten Berechnungen, Teil des gesamten Aufmerksamkeitsgrads sind (siehe Abschnitt 3.5).

3.6 Die Interaktionsformen

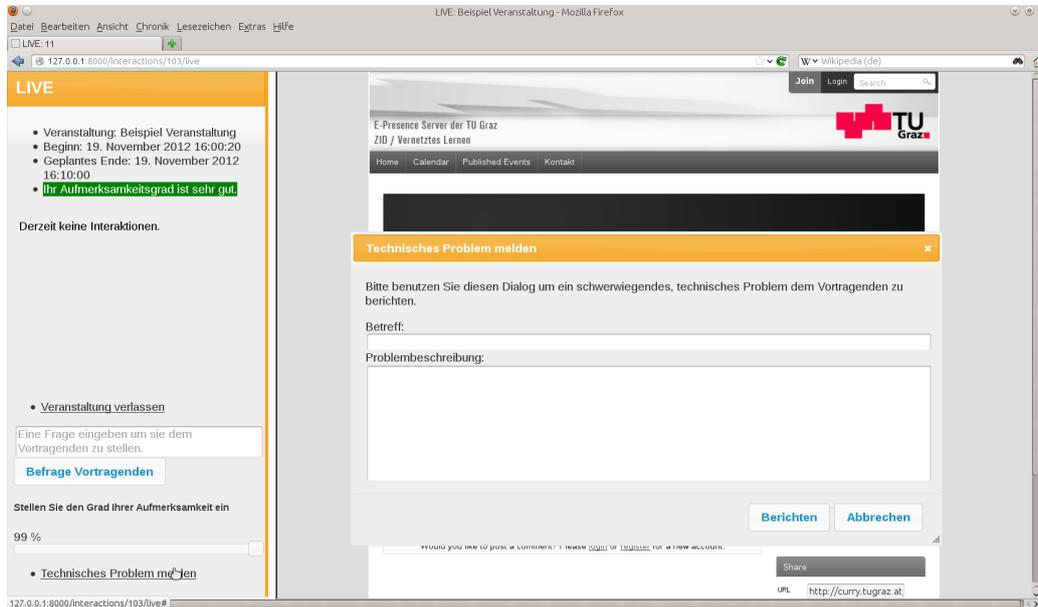


Abbildung 3.25: Melden eines technischen Problems

3.6.6 Technisches Problem melden

Mit dieser Interaktionsform erhalten die Teilnehmerinnen und Teilnehmer die Möglichkeit, ein technisches Problem, wie zum Beispiel Probleme mit der Übertragung der Vorlesung, an die Vortragende bzw. den Vortragenden, zu melden. Dazu wurde ein Template erstellt, welches einen Dialog zum Melden des Problems in die Benutzeroberfläche integriert (siehe Abbildung 3.25).

Dieser Dialog erfordert die Eingabe eines Betreffs und einer Beschreibung des Problems. Nach dem Absenden der Problembeschreibung erstellt eine View eine Instanz des Modells für Fehlermeldungen. Dies bewirkt, wie in Abschnitt 3.4.1 erwähnt, die Darstellung des Problembereichs bei der bzw. dem Vortragenden an einer prominenten Stelle.

Da diese Interaktionsform normalerweise nicht regelmäßig verwendet wird, ist sie nicht Teil der Analyse der Aufmerksamkeit. Aus diesem Grund wurde die entsprechende Schnittstelle nicht implementiert, wodurch diese In-

3 Design und Implementation

Interaktionsform bei der Berechnung des Aufmerksamkeitsgrads ignoriert wird.

3.7 Die Administrationsoberfläche

Es wurde bereits mehrfach erwähnt, dass das Web-Portal auch eine Administrationsoberfläche anbietet. Diese ermöglicht zum Beispiel das Vergeben der Benutzerrechte oder das Aktivieren von Interaktionsformen.

Das Django Web-Framework liefert dazu eine vorgefertigte Implementation mit [Dja12b; HK12]. Diese basiert auf den Modellen des Projekts und ermöglicht es im einfachsten Fall, Daten dieser Modelle anzusehen, zu bearbeiten oder auch Daten hinzuzufügen. Damit diese Administrationsoberfläche die gewünschten Funktionen übernimmt, wurde im Paket *core* ein Modul namens *admin* erstellt, welches die entsprechende Konfiguration der Administrationsoberfläche enthält.

Profile verwalten Eine der wichtigsten Funktionen der Administrationsoberfläche ist die Verwaltung der Benutzerprofile. Dadurch sind die Administratoren nicht nur in der Lage Benutzerprofile zu bearbeiten und zu löschen, sondern können diese auch deaktivieren. Außerdem ist die Vergabe von erweiterten Rechten nur hier möglich. Abbildung 3.26 zeigt den Dialog zum Bearbeiten eines Benutzerprofils. Dabei erkennt man neben den Feldern für die allgemeinen Daten (z.B.: Benutzername, email, usw.) auch jene zum Erwerben der Administrations-Rechte. Zusätzlich kann auch die Zugehörigkeit zur Gruppe der Lehrenden festgelegt werden (siehe Abschnitt 3.3.3).

Interaktionsform aktivieren Wie im Abschnitt 3.4.1 erwähnt, muss eine Interaktionsform in die Datenbank eingetragen werden, bevor sie verwendet werden kann. Dazu stellt die Administrationsoberfläche den in Abbildung 3.27 gezeigten Dialog zur Verfügung. Dazu gibt es neben dem Feld für den Namen des Moduls der Interaktionsform auch welche zur Eingabe eines Titels und einer Beschreibung. Weiters steht ein Feld bereit, um den Typ der Interaktionsform festzulegen.

Erweiterte Verwaltung der Veranstaltungen Schließlich wird auch eine erweiterte Version der Veranstaltungsverwaltung angeboten, welche wesentlich umfangreichere Funktionen bereitstellt

3.7 Die Administrationsoberfläche

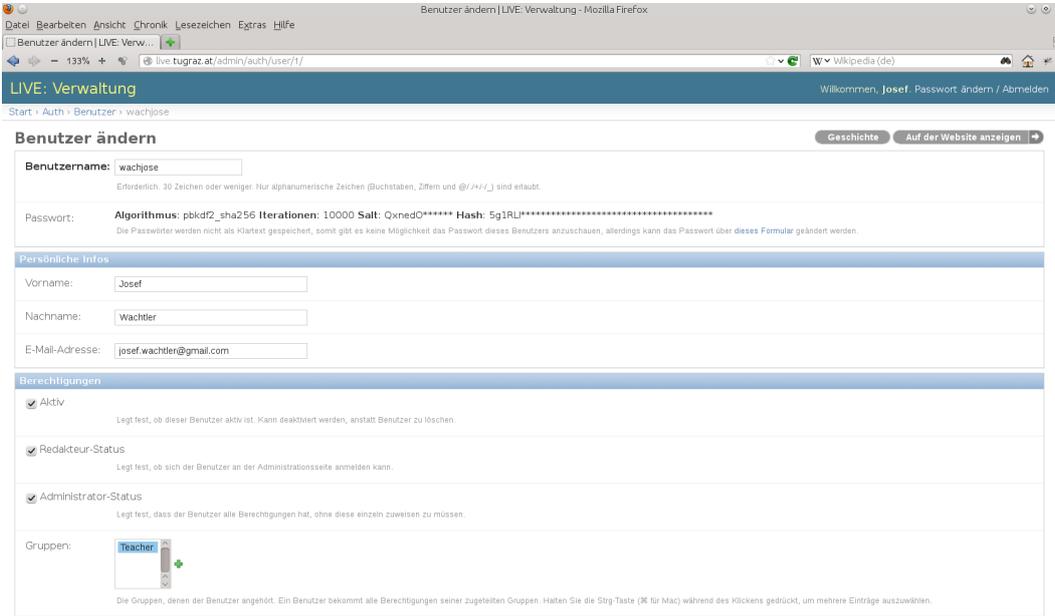


Abbildung 3.26: Bearbeiten eines Benutzerprofils

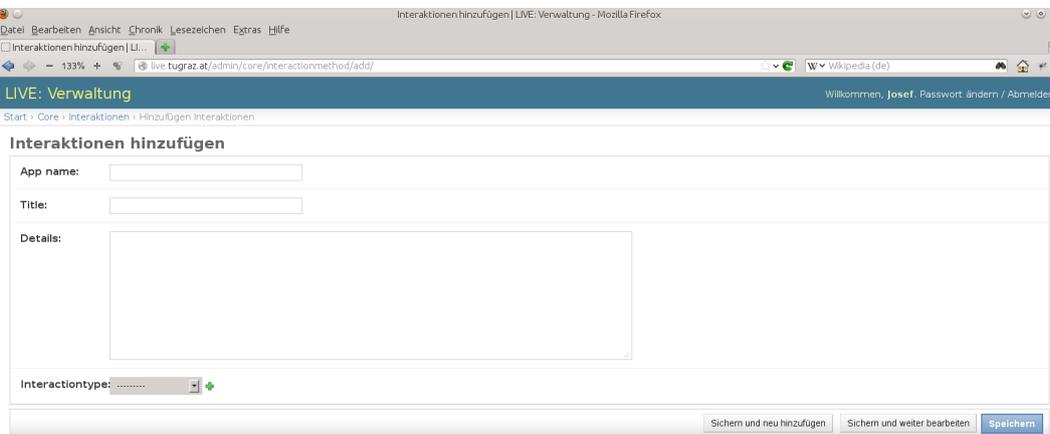


Abbildung 3.27: Hinzufügen einer Interaktionsform

3 Design und Implementation

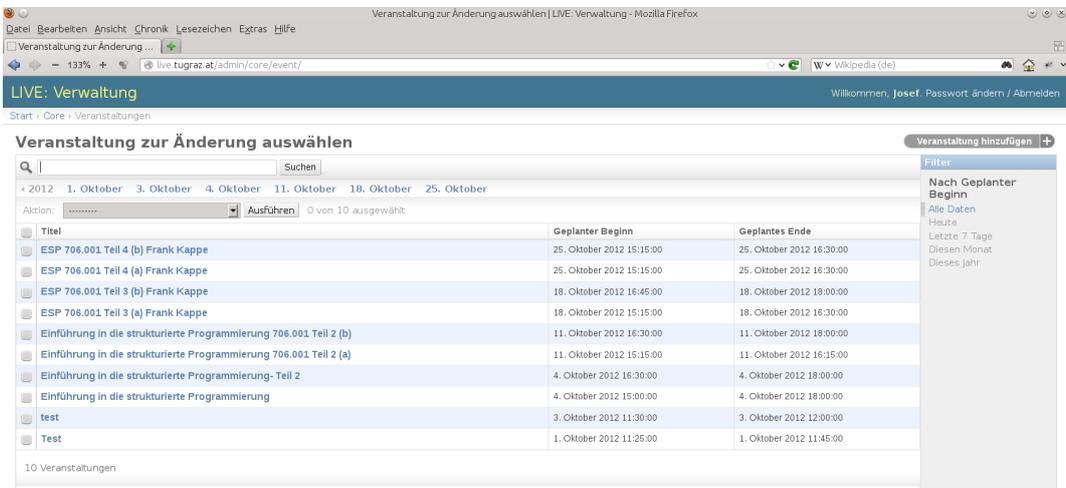


Abbildung 3.28: Such- und Filterfunktionen der Veranstaltungen

als die der Benutzerinnen bzw. Benutzer mit Rechten für Lehrende. Dabei handelt es sich um folgende Funktionen:

- Umfangreiche Such- und Filterfunktionen (siehe Abbildung 3.28).
- Bearbeiten aller Daten einer Veranstaltung.
So können zum Beispiel auch die tatsächlichen Beginn- und Endzeiten oder die bzw. der Vortragende verändert werden.
- Löschen einer Veranstaltung auch im Nachhinein.
- Entfernen einer Teilnehmerin bzw. eines Teilnehmers aus einer Veranstaltung (siehe Abbildung 3.29).

3.7 Die Administrationsoberfläche

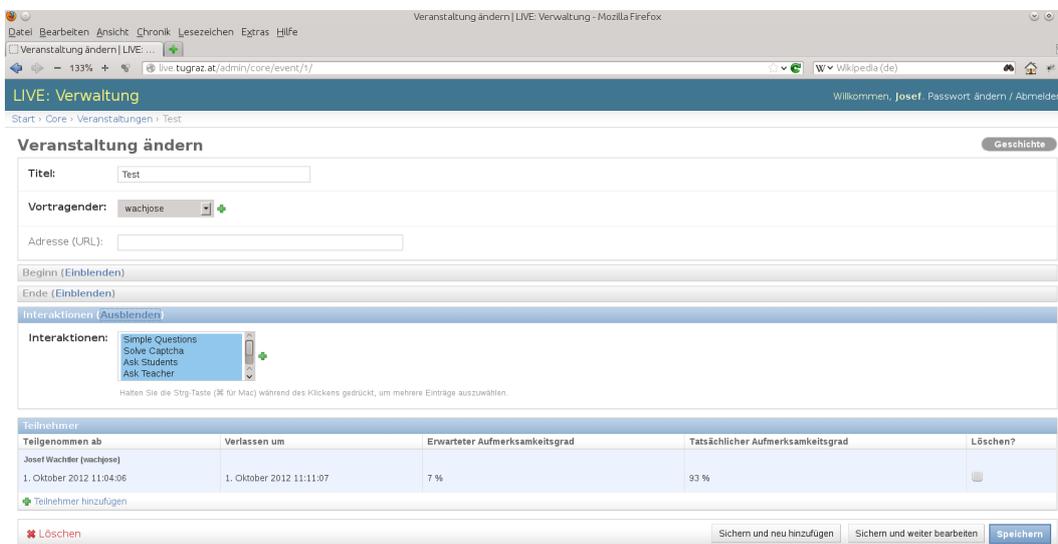


Abbildung 3.29: Bearbeiten einer Veranstaltung mit deren Teilnehmerinnen und Teilnehmern

4 Evaluation

Nachdem im vorangegangenen Kapitel alle Funktionen des Web-Portals erläutert wurden und deren Implementation vertiefend erklärt wurde, dient dieses Kapitel zur Dokumentation der Evaluation des Web-Portals. Dazu wird neben einem ersten Testlauf auch auf die Einsätze bei verschiedenen Vorlesungsübertragungen eingegangen. Für jeden Einsatz werden die Ziele und der Ablauf sowie die Resultate beschrieben. Die daraus gewonnenen Erkenntnisse, inklusive der Vor- und Nachteile, werden im Kapitel 5 diskutiert.

4.1 Der Erste Test - »Proof of Concept«

Dieser Abschnitt dokumentiert die ersten Testläufe des Web-Portals mit fünf freiwilligen Testpersonen mit ihren Notebooks. Dabei wurden hauptsächlich die folgenden Ziele verfolgt:

- Erprobung aller Komponenten des Web-Portals unter realitätsnahen Bedingungen.
- Finden von passenden Werten für die Auftrittshäufigkeit der automatischen Interaktionen (siehe Abschnitt 3.4.1).
- Testen mit verschiedenen ausgestatteten Geräten.
- Gewinnung von realistischen Werten für die Parameter der Berechnung des Aufmerksamkeitsgrads auf Basis der Reaktionszeiten (siehe Abschnitt 3.5.3).

4 Evaluation

Modellname		
Prozessor	RAM	Display
Betriebssystem	Web-Browser	
Acer Aspire One Happy		
Intel Atom Dual Core N550, 1,5GHZ	1 GB	10,1", 1024x600
Ubuntu 10.04 i386	Mozilla Firefox 12.0	
Fujitsu Amilo M1425		
Intel Pentium-M 735, 1.70GHZ	512 MB	15,4", 1280x800
Microsoft Windows XP Home	Mozilla Firefox 9.0.1	
Apple MacBook Pro		
Intel Core i7-2820QM, 2.30GHZ	4 GB	15,6", 1680x1050
Mac OS X 10.7	Safari 5.0.5	
Lenovo ThinkPad W510		
Intel Quad Core i7-820QM, 1.73GHZ	8 GB	15,6", 1920x1080
Kubuntu 12.04 amd64	Chromium 18.0	
Lenovo ThinkPad W520		
Intel Quad Core i7-2820QM, 2.3GHZ	8 GB	15,6", 1920x1080
Microsoft Windows 7	Internet Explorer 9	

Tabelle 4.1: Die Spezifikationen der Testnotebooks beim ersten Testlauf

Um diese Ziele zu erreichen, wurden mehrere Testläufe mit freiwilligen Testpersonen durchgeführt. Dabei wurde vor allem darauf geachtet, dass deren Notebooks sich in der Leistung und der verwendeten Software unterscheiden. In Tabelle 4.1 sind die verschiedenen Merkmale und Konfigurationen der Notebooks dargestellt. Man erkennt, dass die Hardware von alten oder kleinen und somit schwächeren Notebooks bis hin zu modernen und leistungsfähigen Geräten reicht. Softwareseitig sind drei der bekanntesten Betriebssysteme in unterschiedlichen Versionen im Einsatz und darauf laufen verschiedene Web-Browser. Somit wurde versucht, trotz einer kleinen Anzahl von Testgeräten, eine möglichst breit gefächerte Zusammensetzung zu erreichen.

Im Rahmen dieser Testreihe wurden die Testpersonen gebeten, an vier halbstündigen Veranstaltungen teilzunehmen. Dabei wurde eine Videoan-

4.1 Der Erste Test - »Proof of Concept«

leitung¹ für die Programmiersprache »Scratch«² vom Web-Portal eingebunden und dadurch mit Interaktionen unterstützt. Während den Veranstaltungen wurden mit der entsprechenden Interaktionsform (siehe Abschnitt 3.6.3) zum Thema passende Fragen an die Teilnehmerinnen und Teilnehmer gestellt.

Bei der ersten Veranstaltung erhielten die Teilnehmerinnen und Teilnehmer, außer dem Hinweis sich normal zu verhalten, keinerlei Instruktionen. Bei der zweiten Veranstaltung hingegen wurden die Teilnehmerinnen und Teilnehmer gebeten, nach bestimmten Verhaltensweisen zu agieren:

- Eine Teilnehmerin bzw. ein Teilnehmer sollte pünktlich zur Veranstaltung erscheinen und auf alle Interaktionen schnell und korrekt reagieren.
- Im Vergleich dazu sollte eine andere bzw. ein anderer zwar pünktlich kommen, jedoch auf die Interaktionen etwas verzögert reagieren.
- Weiters wurde eine Teilnehmerin bzw. ein Teilnehmer gebeten, mit einer zehnminütigen Verspätung an der Veranstaltung teilzunehmen und die Interaktionen aber zeitnah und richtig zu erledigen.
- Umgekehrt sollte eine Teilnehmerin bzw. ein Teilnehmer die Veranstaltung um zehn Minuten zu früh verlassen.
- Schließlich wurde eine Teilnehmerin bzw. ein Teilnehmer dazu veranlasst, während der Veranstaltung für einen Zeitraum von zehn Minuten auf keine Interaktion zu reagieren.

Danach wurden die berechneten Aufmerksamkeitsgrade analysiert und auf ihre Plausibilität geprüft. Auf Grund dessen war es möglich, die Parameter der Funktion zur Berechnung der Aufmerksamkeitsgrade auf Basis der Reaktionszeiten (siehe Abschnitte 3.5.3 und 3.6) realitätsnah anzupassen.

Außerdem wurden die Teilnehmerinnen und Teilnehmer um ihre Erfahrungen, vor allem um ihre Meinung zur Häufigkeit der Interaktionen und deren Auswirkungen auf die Aufmerksamkeit, gebeten. Dabei wurde grundsätzlich eine positive Auswirkung auf die Aufmerksamkeit bescheinigt, da

¹ <http://www.youtube.com/watch?v=qrits8nDWzE>, besucht am 06.11.2012

² <http://scratch.mit.edu/>, besucht am 06.11.2012

4 Evaluation

hauptsächlich die zum Inhalt passenden Fragen zur Mitarbeit motivieren. Allerdings wurden die automatischen Interaktionen, welche 20 mal pro Stunde auftraten, als zu häufig angesehen. Diese Anregungen fanden für die dritte und vierte Veranstaltung ihre entsprechende Berücksichtigung und deshalb wurden die Auftrittshäufigkeiten der automatischen Interaktionen auf 10 mal pro Stunde reduziert.

Mit dieser veränderten Konfiguration des Web-Portals wurden die Testpersonen gebeten, erneut an zwei Veranstaltungen teilzunehmen, welche in der gleichen Art und Weise absolviert werden sollten. Am Ende gaben nun wieder alle Teilnehmerinnen und Teilnehmer einen Erfahrungsbericht ab, welcher die Veränderungen der Auftrittshäufigkeit der automatischen Interaktionen jetzt für passend befand. Außerdem wurde wiederum die erhöhte Motivation zur Mitarbeit und die somit entstehende größere Aufmerksamkeit lobend erwähnt. Die erneute Evaluation der Berechnung der Aufmerksamkeitsgrade zeigte nun eine ausreichend genaue Parametrisierung der Aufmerksamkeitsanalyse.

Die meisten Komponenten des Web-Portals zeigten während dieser Testläufe ein zufriedenstellendes Verhalten. Es wurde jedoch auch festgestellt, dass das zu späte Erscheinen zu einer Veranstaltung bei der Analyse des Aufmerksamkeitsgrads nach der Veranstaltung nicht entsprechend berücksichtigt wurde. Deshalb musste die im Abschnitt 3.5.1 erwähnte spezielle Berücksichtigung dieses Falles eingeführt werden.

Alle eingesetzten Testgeräte waren ausreichend leistungsfähig und konnten die Anforderungen erfüllen. Es zeigte sich, dass wenn der fremde Web-Inhalt Adobe Flash³ verwendet, es zu Problemen mit WebKit⁴ basierten Web-Browsern (z.B. Safari) kommen kann. Dabei handelt es sich um ein bekanntes Problem^{5,6} und dieses fällt auch nicht in den Bereich des Web-Portals. Da jedoch die meisten Web-Inhalte, welche zur Übertragung von Vorlesungen dienen, Flash verwenden, ist es empfehlenswert, einen entsprechenden Web-Browser zu verwenden.

³ <http://www.adobe.com/de/products/flashplayer.html>, besucht am 06.11.2012

⁴ <http://www.webkit.org/>, besucht am 06.11.2012

⁵ <https://discussions.apple.com/thread/3081384?start=0&tstart=0>,
besucht am 06.11.2012

⁶ <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=519854>, besucht am 06.11.2012

4.2 Erste Verwendung bei GADI 2012

Im Rahmen der Lehrveranstaltung »Gesellschaftliche Aspekte der Informationstechnologie« (kurz GADI)⁷ an der Technischen Universität Graz wurde das Web-Portal LIVE zum ersten Mal in der Praxis eingesetzt. Bei dieser Lehrveranstaltung des Bachelorstudiums Informatik werden verschiedene Vorträge abgehalten, bei welchen Anwesenheitspflicht herrschte. Da die Vorträge vom E-Presence Server⁸ live ins Internet übertragen wurden, kam das Web-Portal zur Unterstützung dieser Übertragungen bei zwei Vorträgen zum Einsatz. Dies geschah aus Sicht der Evaluation zur Erreichung der folgenden Ziele:

- Verwenden und testen des Web-Portals in der Praxis
- Gewinnen von Feedback der Teilnehmerinnen und Teilnehmer
- Verfeinern der Parameter der Berechnung des Aufmerksamkeitsgrads

Für diese erste Verwendung in der Praxis wurde in der Newsgroup der Lehrveranstaltung nach freiwilligen Teilnehmerinnen und Teilnehmern gesucht. Daraufhin fanden sich neun Freiwillige, welche vor den Vorträgen ein kurzes Einführungsmail erhielten. Darin wurde hauptsächlich auf die Hilfe des Web-Portals verwiesen. Da es sich bei den Vorträgen um Gastvorträge handelte, war es nicht möglich, dass das Web-Portal auf Seite der Vortragenden von diesen selbst bedient werden konnte. Somit übernahm diese Aufgabe ein Studienassistent. Dieser überwachte den Ablauf während der Vorträge und reichte einige Fragen über die entsprechende Interaktionsform an die Teilnehmerinnen und Teilnehmer weiter.

Während der Vorträge zeigte die Analyse des Aufmerksamkeitsgrads einen hohen Wert, welcher durchwegs über 75% lag. Dies wurde auch durch die detaillierte Analyse nach den Vorträgen bestätigt. Dabei war ersichtlich, dass der Aufmerksamkeitsgrad der meisten Teilnehmerinnen und Teilnehmer über 75% gelegen war und nur bei wenigen war ein niedriger Wert

⁷ https://online.tugraz.at/tug_online/lv.detail?sprache=1&clvnr=162241, besucht am 06.11.2012

⁸ <http://curry.tugraz.at>, besucht am 06.11.2012

4 Evaluation

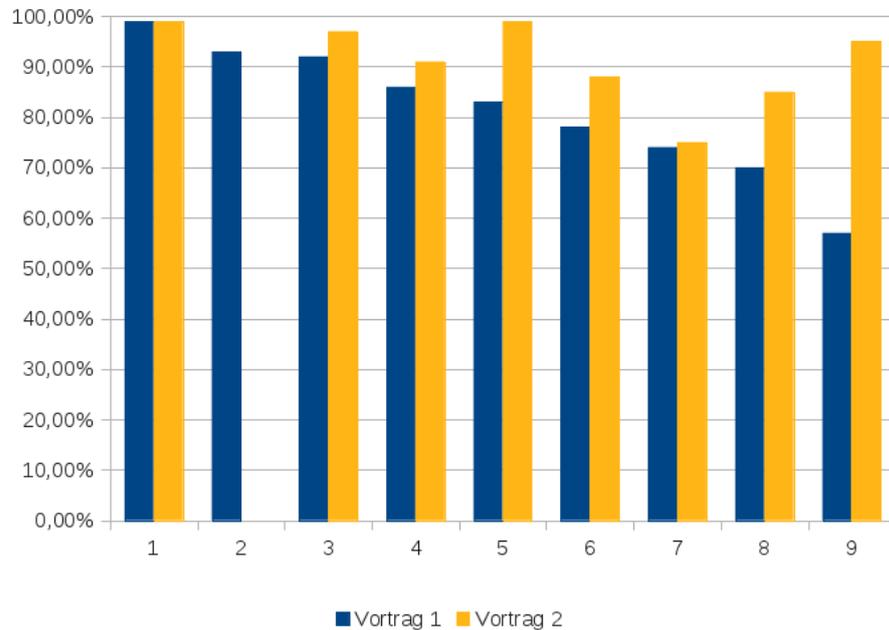


Abbildung 4.1: Die durchschnittlichen Aufmerksamkeitsgrade der Teilnehmerinnen und Teilnehmer im Vergleich

berechnet worden. Weiters ließ sich beim zweiten Vortrag ein höherer Aufmerksamkeitsgrad als beim Ersten ablesen. In Abbildung 4.1 sind diese Daten zusammenfassend dargestellt, indem für jede Teilnehmerin bzw. jeden Teilnehmer die durchschnittlichen Aufmerksamkeitsgrade vergleichend gegenübergestellt werden.

Zur Gewinnung eines Feedbacks der Teilnehmerinnen und Teilnehmer wurden diese um eine Auflistung der Vor- und Nachteile des Systems aus Ihrer Sicht sowie um einen Erfahrungsbericht gebeten. Dies brachte zusammenfassend folgende positive Aussagen bzw. Meinungen:

- Verbesserung des Verständnisses und der Aufmerksamkeit durch die gezielten zum Inhalt passenden Fragen während der Vorträge.
- Positiv ist, dass auch Fragen an die Vortragende bzw. den Vortragenden gestellt werden können.

4.3 Regelmäßiger Einsatz bei ESP 2012

- Die verschiedenen Interaktionsformen bieten eine interessante Möglichkeit zur Anwesenheitskontrolle.

Im Gegensatz dazu wurden folgende negative Punkte geäußert:

- Anstatt der einfachen Fragen (siehe Abschnitt 3.6.1) wären mehr zum Inhalt passende Fragen vorteilhaft.
- Das Lösen eines Captchas wurde als sinnlos und somit etwas störend empfunden.
- Es war nicht möglich, die Liveübertragung der Vorträge im Vollbildmodus zu betrachten ohne Interaktionen zu versäumen.
- Nach den Vorträgen waren keine Interaktionsformen mehr zugänglich.

Im Allgemeinen überwog jedoch der positive Eindruck, denn die berichteten weniger guten Erfahrungen wurden hauptsächlich nur wenig störend empfunden. Für eine ausführliche Diskussion dieser Rückmeldungen sei auf das Kapitel 5 verwiesen.

Aus technischer Sicht verlief dieser erste Praxiseinsatz ohne nennenswerte Schwierigkeiten. Zusätzlich wurden die in der ersten Testreihe (siehe Abschnitt 4.1) gewonnenen Parameter für die Analyse der Aufmerksamkeit auf Basis der Reaktionszeiten größtenteils bestätigt, denn es zeigten sich nur marginale Abweichungen bei den automatisch ausgelösten Interaktionsformen.

4.3 Regelmäßiger Einsatz bei ESP 2012

Im Wintersemester 2012 war das Web-Portal zur Unterstützung der Liveübertragungen der Vorlesungseinheiten der Lehrveranstaltung namens »Einführung in die Strukturierte Programmierung« (kurz ESP)⁹ regelmäßig

⁹ https://online.tugraz.at/tug_online/lv.detail?sprache=1&clvnr=162268, besucht am 06.11.2012

4 Evaluation

bei allen sechs Einheiten im Einsatz. Diese Lehrveranstaltung ist unter anderem Teil des Bachelorstudiums Informatik und laut Studienplan im ersten Semester angesiedelt. Demzufolge wird diese Lehrveranstaltung von einer großen Anzahl von Studentinnen und Studenten besucht. Aus der Sicht der Evaluation seien für diese Einsätze folgende Ziele genannt:

- Regelmäßiger Einsatz mit unerfahrenen Teilnehmerinnen und Teilnehmern
- Erprobung des Web-Portals mit einer größeren Anzahl von Teilnehmerinnen und Teilnehmern
- Gewinnung von Feedback der Benutzer der Vortragendenoberfläche

Wie auch schon bei der im vorigen Abschnitt beschriebenen Verwendung des Web-Portals kam auch hier der E-Presence-Server der Technischen Universität Graz zur Übertragung der Vorlesungseinheiten zum Einsatz. In der ersten Einheit wurde erklärt, dass die Vorlesungen nicht nur im Hörsaal besucht werden können, sondern auch ins Internet übertragen werden. Dabei wurde auf das Web-Portal verwiesen. Auch hier wurde die Betreuung des Web-Portals auf der Seite der Vortragenden von einem Studienassistenten übernommen.

Es zeigte sich, dass das Web-Portal durchschnittlich von 14 Teilnehmerinnen und Teilnehmern genutzt wurde. Die Analyse der Aufmerksamkeitsgrade brachte jedoch zu Tage, dass nur eine kleine Anzahl (2 bis 4) auch aktiv an den Interaktionen teilnahm. Die restlichen Teilnehmerinnen und Teilnehmer benutzten das Web-Portal nur zum Betrachten der Übertragung und ignorierten somit alle angebotenen Interaktionen. Dieser Sachverhalt ist auch in Abbildung 4.2 dargestellt. Dabei ist ebenfalls ersichtlich, dass die Teilnehmeranzahl ab der zweiten Einheit zurückging.

Im Vergleich zum Einsatz bei GADI, konnten hier durch die erhöhte Anzahl an Teilnehmerinnen und Teilnehmern, ebenfalls keine technischen Probleme festgestellt werden. Der betreuende Studienassistent bestätigte diesen reibungslosen Ablauf, berichtete jedoch von einer fehlenden Möglichkeit zum Pausieren der Veranstaltung, denn die Einheiten der Lehrveranstaltung waren jeweils durch eine 15 minütige Pause geteilt. Aus diesem Grund war es notwendig, für jede Einheit zwei Veranstaltungen im Web-

4.3 Regelmäßiger Einsatz bei ESP 2012

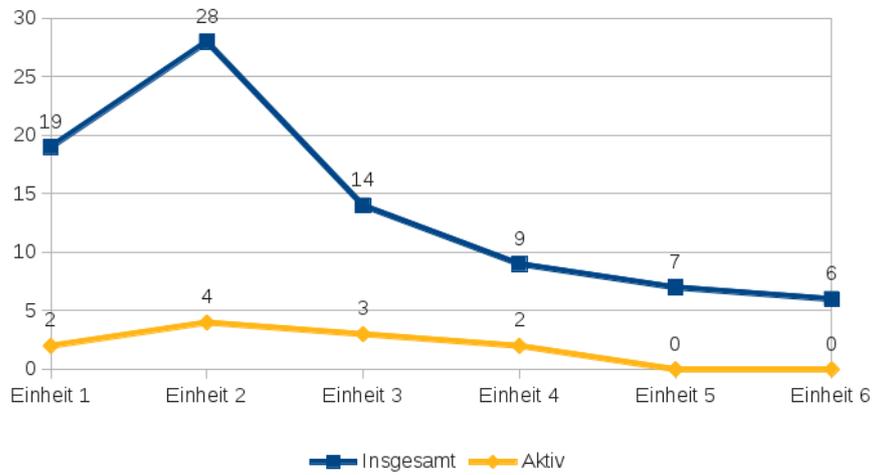


Abbildung 4.2: Vergleich der Anzahl der gesamten Teilnehmerinnen bzw. Teilnehmer mit der Anzahl der Aktiven (durchschnittlicher Aufmerksamkeitsgrad über 50%) für jede Vorlesungseinheit

Portal anzulegen, da ansonsten die Teilnehmerinnen und Teilnehmer auch während der Pause mit Interaktionen konfrontiert geworden wären. Diese unter Umständen versäumten Interaktionen hätten somit eine fehlerhafte Berechnung der Aufmerksamkeitsgrade zur Folge gehabt.

5 Diskussion

Dieses Kapitel dient zur Diskussion der Erkenntnisse und Ergebnisse der Evaluation. Dabei wird neben einigen Bemerkungen zu den ermittelten Aufmerksamkeitsgraden und deren Veränderungen auch auf die Erfahrungen und Meinungen der Teilnehmerinnen und Teilnehmer eingegangen. Weiters kommen auch Einschränkungen zur Diskussion, welche hauptsächlich aufgrund der Zielsetzungen dieses Web-Portals entstehen. Daraus ergibt sich eine zusammenfassende Auflistung der Vor- und Nachteile des Web-Portals.

5.1 Erfahrungen zum Ablauf

In den einleitenden Bemerkungen in den Kapiteln 1 und 2 wurde erläutert, dass das entwickelte Web-Portal verschiedene Methoden zur synchronen Interaktion und Kommunikation während einer Veranstaltung des E-Learnings, vor allem bei Liveübertragungen von Vorlesungen an einer Universität ins Internet, bereitstellen wird. Dies soll einen wesentlichen Beitrag zur Erhöhung der Mitarbeit leisten und somit auch die Aufmerksamkeit erhalten bzw. fördern. Der Schwerpunkt bei der Evaluation lag somit bei der Gewinnung von Feedback, welches Aufschluss gibt, ob die Teilnehmerinnen und Teilnehmer tatsächlich positive Effekte auf die Mitarbeit und Aufmerksamkeit feststellen konnten. Es zeigte sich, dass dies zwar grundsätzlich der Fall ist, dafür jedoch einige Punkte beachtet werden müssen:

- Die Häufigkeit der Interaktionen muss passend sein.
- Interaktionen mit zum Inhalt passenden Aufgaben bzw. Fragen sind zu bevorzugen.

5 Diskussion

- Andere Interaktionen (z.B. das Lösen eines Captchas) bedürfen einer Erklärung und eines sparsamen Einsatzes.
- Interaktionen, welche von den Teilnehmerinnen bzw. Teilnehmern ausgelöst werden können (z.B. das Stellen einer Frage an die Vortragende bzw. den Vortragenden), sind wichtig für die Teilnehmerinnen und Teilnehmer.
- Besprechen und Erklärung des Einsatzes des Web-Portals am Beginn einer Veranstaltung.

Laut Silliman und McWilliams [SMo4] ist der Erfolg des Einsatzes von solchen Interaktionsformen stark von der Häufigkeit der Interaktionen abhängig, denn eine zu häufige Verwendung führt zu einer unnötigen Verlangsamung der Veranstaltung und somit auch zu einer sinkenden Aufmerksamkeit. Dieser Effekt wurde bei der Evaluation des Web-Portals bestätigt.

Zusätzlich trägt auch die Art der Interaktionen wesentlich zum Erfolg oder Misserfolg bei. So zeigte sich, dass verschiedene Aufgaben bzw. Fragen, welche zum Inhalt der Veranstaltung passen, den größten Nutzen bringen. Dies ist wahrscheinlich der Fall, da es dadurch zu einer erneuten Beschäftigung mit dem Inhalt und zu keiner unnötigen Ablenkung kommt.

Im Gegensatz dazu wurden zu häufig auftretende allgemeine Interaktionen, welche keine inhaltsbezogenen Fragen anbieten, als etwas störend empfunden. Der Hauptgrund für dieses Empfinden als Störung lag hauptsächlich darin, dass die Teilnehmerinnen und Teilnehmer nicht wussten warum auch solche sinnlos erscheinenden Interaktionen vorkommen. Daher ist es empfehlenswert, den Einsatz dieser Art von Interaktionen zu minimieren und einleitend zu begründen.

Ein weiterer von den Teilnehmerinnen und Teilnehmern häufig positiv empfundener Punkt ist die Möglichkeit, auch selbst Interaktionen auszulösen. Besonders förderlich für die Mitarbeit wurde dabei die Funktion zum Stellen einer Frage an die Vortragende bzw. den Vortragenden angesehen. Im Kontrast dazu steht jedoch die Tatsache, dass diese Funktionen bei den Einsätzen zur Evaluation kaum verwendet wurden. In diesem Fall führt anscheinend alleine das bloße Bestehen der Möglichkeit zur Kommunikation mit der bzw. dem Vortragenden schon zu einem positiveren Gefühl.

5.2 Bemerkungen zu den Aufmerksamkeitsgraden

Wie bereits erwähnt, ist den Teilnehmerinnen und Teilnehmern oft nicht klar, warum sie sich mit den angebotenen Interaktionen auseinandersetzen sollen. Dies führte in einigen Fällen zur Irritation und in weiterer Folge auch zur Ignorierung der Interaktionen. Aus diesem Grund ist es empfehlenswert, dass die bzw. der Vortragende am Beginn der Veranstaltung den Teilnehmerinnen und Teilnehmern kurz die positiven Auswirkungen der Interaktionen auf die Mitarbeit und die Aufmerksamkeit erläutert. Dies sollte in beidseitigem Interesse geschehen, denn nicht nur die Teilnehmerinnen und Teilnehmer profitieren von einer erhöhten Aufmerksamkeit, sondern auch die bzw. der Vortragende erhält im Falle der aktiven Benutzung des Web-Portals ein wertvolles Feedback über die Aufmerksamkeit und bei gezielten Fragestellungen auch einen Überblick über das Verständnis der Teilnehmerinnen und Teilnehmer.

Zusammenfassend lässt sich also ableiten, dass die erfolgreiche Nutzung des Web-Portals sowohl von der aktiven Unterstützung durch die Teilnehmerinnen und Teilnehmer, als auch von der aktiven Verwendung durch die Vortragende bzw. den Vortragenden abhängig ist. Wird das Web-Portal bei der Liveübertragung einer Lehrveranstaltung ins Internet genutzt, so ist diese erwähnte Unterstützung vor allem bei kleineren Veranstaltungen erreichbar.

5.2 Bemerkungen zu den Aufmerksamkeitsgraden

Bei den beiden Einsätzen im Rahmen der Evaluation bei Vorlesungen an der Technischen Universität Graz (siehe Abschnitte 4.2 und 4.3) wurde auch eine Analyse der berechneten durchschnittlichen Aufmerksamkeitsgrade durchgeführt. Diese zeigte jeweils ein unterschiedliches Bild.

Im Fall der beiden Vorträge bei der Lehrveranstaltung GADI wurde ein durchwegs hoher Aufmerksamkeitsgrad bei nahezu allen Teilnehmerinnen und Teilnehmern errechnet. Dies mag hier daran liegen, dass die Teilnehmerinnen und Teilnehmer eine Vorabinformation erhalten hatten und sich aus der aktiven Nutzung einen Vorteil erhofften. Ebenfalls spielte die herrschende Anwesenheitspflicht, welche durch die errechneten Aufmerksamkeitsgrade kontrolliert werden kann, eine entscheidende Rolle.

5 Diskussion

Weiters ist durch die höheren Aufmerksamkeitsgrade beim zweiten Vortrag ersichtlich, dass der effiziente Umgang und somit die erfolgreiche Benutzung auch von der Erfahrung mit dem Web-Portal abhängig ist. Diese Tatsache führt die Notwendigkeit der bereits erwähnten kurzen Einführung in die Zielsetzungen des Einsatzes dieses Web-Portals am Beginn einer Veranstaltung vor Augen.

Im Gegensatz dazu präsentiert sich bei den Einheiten der Vorlesung ESP ein komplett anderes Bild. Die Aufmerksamkeitsgrade liegen bei den meisten Teilnehmerinnen und Teilnehmern auf einem sehr niedrigen Niveau, nur bei wenigen erreichte der errechnete Aufmerksamkeitsgrad eine akzeptable Höhe. Dies kam wahrscheinlich dadurch zustande, dass die Teilnehmerinnen und Teilnehmer durch fehlende einleitende Erklärungen keinen Sinn in den Interaktionen sahen. Auch dürfte die fehlende Anwesenheitspflicht eine Rolle gespielt haben, was zum Ignorieren der Interaktionen führte.

Ebenfalls wurde ein beträchtlicher Rückgang der Anzahl der Teilnehmerinnen und Teilnehmer festgestellt. Diese Tatsache liegt unter Umständen daran, dass die Übertragungen auch aufgezeichnet wurden, wodurch die Notwendigkeit zur Teilnahme in Echtzeit entfallen ist.

5.3 Einschränkungen

Die im Abschnitt 1.1 aufgelisteten Zielsetzungen bringen naturgemäß auch einige Einschränkungen mit sich. Diese Einschränkungen wurden im Rahmen der Evaluation von den Teilnehmerinnen und Teilnehmern erkannt und berichtet und sollen hier diskutiert werden.

Da das Web-Portal ja für alle Arten von Web-Inhalten verwendbar sein soll und somit vollständig unabhängig von diesen arbeitet, ergibt sich daraus unter bestimmten Umständen ein Problem. So bieten zum Beispiel viele Systeme zur Übertragung einer Vorlesung ins Internet die Möglichkeit an, diese Übertragung auch im Vollbildmodus zu betrachten. Wird diese Funktion von den Teilnehmerinnen und Teilnehmern verwendet, so ist es für sie nicht mehr möglich, von den durch das Web-Portal angebotenen Interaktionen Notiz zu nehmen. Dieses störende Problem entsteht dadurch, dass

5.3 Einschränkungen

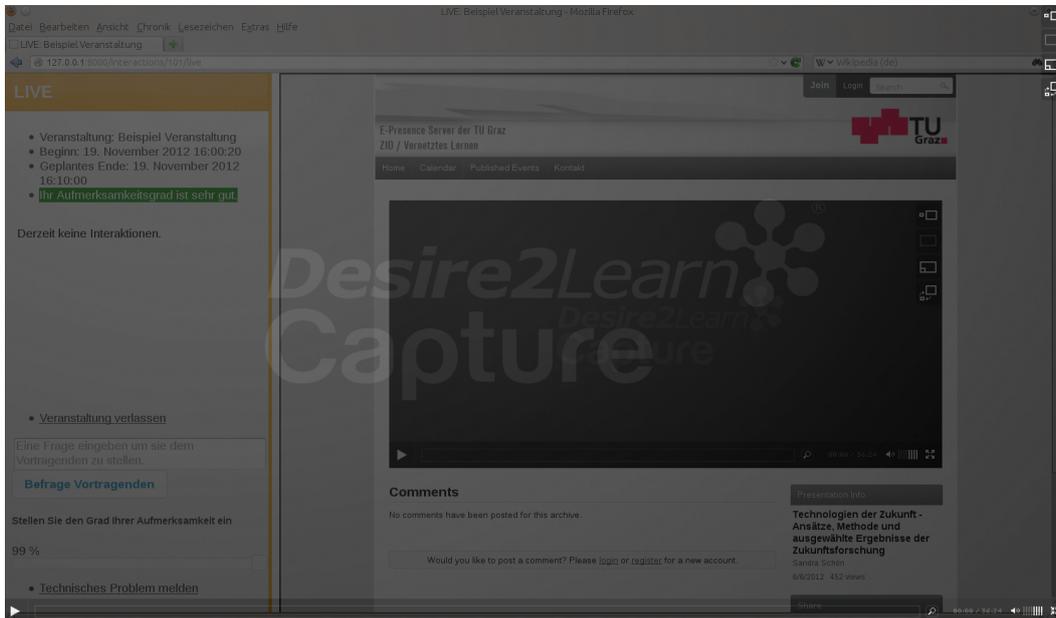


Abbildung 5.1: Der Vollbildmodus der Vorlesungsübertragung (hier transparent dargestellt) verdeckt das Web-Portal

das Web-Portal keine Möglichkeit hat auf den fremden Web-Inhalt Einfluss zu nehmen um den Vollbildmodus bei auftauchenden Interaktionen zu beenden.

Dieser Sachverhalt ist in Abbildung 5.1 dargestellt. Dabei ist ersichtlich, dass die Darstellung der Übertragung im Vollbildmodus, welcher hier zu Demonstrationszwecken transparent angezeigt wird, zu einer vollständigen Überblendung des Web-Portals führt.

Zusätzlich wurde von manchen Teilnehmerinnen und Teilnehmern bemängelt, dass nach einer Veranstaltung keine Möglichkeiten mehr zur Interaktion und Kommunikation bestehen. Dieser Umstand ist einer grundlegenden Entscheidung geschuldet, welche es als oberste Priorität ansah, Interaktionen während einer Veranstaltung zu ermöglichen und somit außerhalb der Zielsetzungen dieser lag.

5.4 Vor- und Nachteile

Aus den in diesem Kapitel diskutierten Ergebnissen der Evaluation, den grundlegenden Zielsetzungen und deren Umsetzung lassen sich nun einige Vor- und Nachteile des Web-Portals zusammenfassend aufzählen. So können die folgenden positiven Aspekte genannt werden:

- Bereitstellung von verschiedenen Formen der Interaktion und Kommunikation während einer Veranstaltung.
- Stellen von Fragen an die Teilnehmerinnen und Teilnehmer durch die Vortragende bzw. den Vortragenden.
- Möglichkeiten zur Kommunikation der Teilnehmerinnen und Teilnehmer mit der bzw. dem Vortragenden (z.B. durch das Stellen einer Frage).
- Positive Auswirkungen auf die Mitarbeit und die Aufmerksamkeit der Teilnehmerinnen und Teilnehmer unter den genannten Voraussetzungen.
- Feedback über die Höhe der Aufmerksamkeit sowohl für die Vortragende bzw. den Vortragenden als auch für die Teilnehmerinnen und Teilnehmer.
- Verwendbar für alle Arten von Web-Inhalten bei welchen Interaktionen in Echtzeit sinnvoll erscheinen.

Im Gegensatz dazu müssen jedoch auch die folgenden nachteiligen Punkte erwähnt werden:

- Finden von passenden Auftrittshäufigkeiten der Interaktionen
- Notwendigkeit von sinnlos erscheinenden automatischen Interaktionen zur Berechnung der Aufmerksamkeitsgrade
- Erklärungsbedarf am Beginn einer Veranstaltung über den Sinn der Interaktionen und deren positiven Auswirkungen
- Fehlende Kontrolle des Web-Portals über den fremden Web-Inhalt

5.4 Vor- und Nachteile

Der letzte Punkt sowohl bei den Vorteilen als auch bei den Nachteilen behandelt dasselbe Feature. Hierbei zeigt sich, dass die mögliche Verwendbarkeit mit allen Arten von Web-Inhalten nicht nur den großen Vorteil der Flexibilität mit sich bringt, sondern auch zu Problem führen kann (siehe Abschnitt [5.3](#)).

6 Ausblick

Die vorliegende Arbeit dokumentiert in den bisherigen Kapiteln die Entwicklung und die Evaluation einer ersten Version eines Web-Portals zur Interaktion und Kommunikation in Echtzeit zur Unterstützung von anderen Web-Inhalten, vor allem von Übertragungen von Vorlesungen ins Internet. Dabei ergeben sich naturgemäß auch einige Ansätze zur Verbesserung und Erweiterung. Diese werden in diesem Kapitel erläutert und es wird auch auf weitere Einsatzmöglichkeiten zur detaillierteren Erforschung eingegangen.

6.1 Allgemeine Verbesserungen

Wie durch den Testbetrieb im Laufe der Evaluation gezeigt wurde, wären die folgenden Verbesserungen begrüßenswert:

- Pausieren einer Veranstaltung
- Steuerungsmöglichkeiten für die fremden Web-Inhalte
- Umfangreichere Analyse der Aufmerksamkeit

Bei der Nutzung des Web-Portals zur Unterstützung von Vorlesungsübertragungen zeigte sich, dass Vorlesungen typischerweise durch eine Pause getrennt sind. Aus diesem Grund wäre eine Funktion zum Pausieren einer Veranstaltung wünschenswert, damit während dieser Pause keine Interaktionen auftreten, welche somit die Analyse der Aufmerksamkeit verfälschen könnten.

Außerdem entpuppte sich der Vorteil, dass das Web-Portal mit jeder Art von Web-Inhalten betrieben werden kann, auch gleichzeitig als Nachteil,

6 Ausblick

denn aus diesem Grund hat das Web-Portal keine Möglichkeit zur Steuerung des fremden Web-Inhalts. Wie erwähnt, führt dies zu Problemen bei der Übertragung von Vorlesungen, da die eingesetzte Übertragungslösung auch einen Vollbildmodus anbietet und somit keine Möglichkeit mehr besteht, von den auftauchenden Interaktionen Notiz zu nehmen. Deshalb wäre eine Schnittstelle, welche die Implementation von Steuerungsmöglichkeiten für fremde Web-Inhalte standardisiert, sinnvoll. Durch so eine Schnittstelle könnte dann das Web-Portal ermächtigt werden, zum Beispiel bei auftauchenden Interaktionen den erwähnten Vollbildmodus zu überblenden.

Ein weiteres hilfreiches Feature bei der Analyse des Aufmerksamkeitsgrads wäre eine detailliertere Darstellung des Verlaufs der Aufmerksamkeit während einer Veranstaltung. Hierfür würde sich ein Liniendiagramm anbieten.

6.2 Zusätzliche Interaktionstypen und -formen

Im Rahmen der Evaluation wurde oftmals der Wunsch geäußert, dass es auch nach Veranstaltungsende noch möglich sein sollte, bestimmte Interaktionen zu nutzen. Aus diesem Grund wäre die Einführung eines vierten Typs notwendig. Mit diesem Typ müsste eine Infrastruktur zum Abwickeln der Interaktionen außerhalb von Veranstaltungen bereitgestellt werden. Auf Basis dieser Infrastruktur erscheint es sinnvoll, zum Beispiel die folgenden Interaktionsformen zu implementieren:

- **Chatfunktionalität**
Bereitstellen der Möglichkeit zum Chatten sowohl in einem öffentlichen Chatraum als auch in geschlossenen Räumen für jede Veranstaltung.
- **Diskussionsforum**
Ein Diskussionsforum könnte neben öffentlichen Bereichen auch Geschlossene für jede Veranstaltung bereitstellen, um auch asynchrone Kommunikationsmöglichkeiten anzubieten.

6.2 Zusätzliche Interaktionstypen und -formen

Unabhängig von einem neuen Typ sind auch weitere Interaktionsformen auf Basis der bestehenden Infrastruktur möglich. So wäre zum Beispiel eine Interaktionsform, mit welcher die bzw. der Vortragende eine Abstimmung unter den Teilnehmerinnen und Teilnehmern vornehmen kann, wünschenswert. Dazu soll die bzw. der Vortragende eine Fragestellung und passende Antwortmöglichkeiten eingeben können. Nachdem die Teilnehmerinnen und Teilnehmer abgestimmt haben, sollen die Ergebnisse in Form eines Diagramms in anonymer Form angezeigt werden.

Da bei der Evaluation viele Teilnehmerinnen und Teilnehmer die automatischen Interaktionsformen störend empfanden, wäre die Entwicklung einer weiteren Interaktionsform, welche automatisch zum Inhalt passende Aufgaben anbietet, empfehlenswert. Dazu müsste eine Infrastruktur zum Einpflegen der Aufgaben vor einer Veranstaltung durch die Vortragende bzw. den Vortragenden geschaffen werden.

Eine weitere Möglichkeit besteht auch in der Umsetzung der oben erwähnten Chatfunktionalitäten für den Betrieb auch während einer Veranstaltung. Dazu könnte diese Interaktionsform die gleiche Grundlage nutzen, jedoch nur den zur Veranstaltung passenden Chatraum bereitstellen. Dieser Chatraum könnte dann durch die oben erwähnte Interaktionsform vom neuen Typ auch nach der Veranstaltung zugänglich gemacht werden.

Aufgrund der einfachen Implementationsweise der Interaktionsformen ist es auch denkbar, Interaktionsformen für spezielle Einsatzgebiete zu entwickeln. Grundsätzlich gibt es hier keine Beschränkungen und der Fantasie kann freien Lauf gelassen werden. Dies sollen die folgenden Beispiele illustrieren:

- **Programmieraufgaben lösen**

Im Rahmen der Vorlesung ESP (siehe Abschnitt 4.3) könnten die Teilnehmerinnen und Teilnehmer zum Beispiel dazu aufgefordert werden, sehr kleine Programmieraufgaben zu lösen. Der Quellcode in der Antwort könnte auch gleich automatisch überprüft werden.

- **Lerngruppen bilden**

Eine spezielle Interaktionsform könnte auch zur Bildung von Lerngruppen umgesetzt werden. Dabei könnte zum Beispiel die Möglichkeit angeboten werden, dass eine Teilnehmerin bzw. ein Teilnehmer

6 Ausblick

eine Anfrage ausschicken kann, worauf interessierte Teilnehmerinnen und Teilnehmer reagieren können.

Bei all diesen Beispielen ist darauf zu achten, dass die Ablenkungen gering gehalten werden. Dies ist wichtig, da ein übermäßiger Einsatz solcher Technologien zu unerwünschten und somit negativen Auswirkungen auf die Aufmerksamkeit führen kann [SM04].

6.3 Weitere Forschungen und Einsatzmöglichkeiten

Die grundlegende Evaluation des entwickelten Web-Portals bestand im Laufe dieser Arbeit hauptsächlich aus Einsätzen bei Vorlesungen zur Unterstützung der Übertragung ins Internet. Dabei waren die obersten Ziele zum einen die Verifikation der Funktionsweise aller Komponenten des Web-Portals und zum anderen die Gewinnung von Informationen über die Auswirkungen der Interaktionen auf die Aufmerksamkeit, welche hauptsächlich durch das Feedback der Benutzerinnen und Benutzer gewonnen wurden.

Zur umfangreicheren Erforschung der Auswirkungen auf die Aufmerksamkeit wären weitere Testreihen erforderlich. Dazu könnten zum Beispiel folgende Methoden zur Anwendung kommen:

- Gezielte Fragebögen vor und nach einer Veranstaltung
- Messen der Emotionen mit den sogenannten »Academic Emotions Questionnaire« (AEQ, [Pek+02])
- Analysieren der Ablenkung durch die Interaktionen mit »Eye-Tracking«
- Überprüfen der Auswirkungen auf das Verständnis mit gezielten Fragen einige Zeit nach der Veranstaltung
- Messen des Verlaufs der Aufmerksamkeit während einer Veranstaltung zum Beispiel mittels EEG¹

¹kurz für »Elektroenzephalografie«

6.3 Weitere Forschungen und Einsatzmöglichkeiten

Bis jetzt wurde das Web-Portal hauptsächlich zur Unterstützung von Vorlesungsübertragungen eingesetzt. Alternativ wäre auch der Einsatz direkt im Hörsaal in Form eines ARS denkbar, denn das Web-Portal ist auf vielen Endgeräten benutzbar, so zum Beispiel auch auf einem modernen Smartphone.

Außerdem wäre auch die Anwendung in Kombination mit statischen Web-Inhalten eine interessante Option. Dabei könnte das Web-Portal für ein zu lernendes Dokument Interaktions- und Kommunikationsmöglichkeiten in Echtzeit mit der Betreuerin bzw. dem Betreuer zu gewissen Zeiten zur Verfügung stellen. Dies entspricht dann in etwa einer Fragestunde bzw. Sprechstunde.

7 Zusammenfassung

Im Rahmen dieser Arbeit wurde ein Web-Portal namens LIVE¹ entwickelt und evaluiert. Das entwickelte Web-Portal ist in der Lage, verschiedene Kommunikationsmöglichkeiten in Echtzeit zur Unterstützung von beliebigen Web-Inhalten bereitzustellen. Grundsätzlich adressiert das Web-Portal alle Arten von Web-Inhalten, vor allem wird jedoch von einer Verwendung im Bereich des E-Learnings, hier hauptsächlich zur Unterstützung von Übertragungen von Vorlesungen ins Internet, ausgegangen, denn die Kommunikation ist laut Carr-Chellman und Duchastel [CDoo] eine der wichtigsten Komponenten des Lernens.

Das entwickelte Web-Portal stellt also zu festgelegten Zeiten einer Veranstaltung den fremden Web-Inhalt dar und unterstützt diesen mit verschiedenen Kommunikationsmöglichkeiten, den sogenannten Interaktionsformen. Dadurch wird die synchrone Kommunikation zwischen den Konsumentinnen und Konsumenten und der Anbieterin bzw. dem Anbieter des Web-Portals ermöglicht. Im Falle der Verwendung zur Unterstützung von Vorlesungsübertragungen bedeutet dies zum Beispiel, dass die bzw. der Vortragende die Möglichkeit hat, eine Frage an die Teilnehmerinnen und Teilnehmer zu stellen. Natürlich ist es umgekehrt auch möglich, eine Frage an die Vortragende bzw. den Vortragenden zu richten.

Wie im Kapitel 1 einleitend erwähnt, soll das Web-Portal einen Beitrag zur Erhaltung bzw. Erhöhung der Aufmerksamkeit bei der Teilnahme an durch das Web-Portal unterstützten Veranstaltungen leisten. Dazu stellt das Web-Portal neben den bereits erwähnten Möglichkeiten zur Fragestellung auch verschiedene automatische Interaktionsformen bereit. Mit deren Hilfe wird eine gewisse »Beschäftigung« der Teilnehmerinnen und Teilnehmer erreicht, damit es nicht zum sogenannten »Berieselungseffekt« mit

¹ kurz für LIVE Interaction in Virtual learning Environments!

7 Zusammenfassung

seinen negativen Auswirkungen auf die Aufmerksamkeit kommt [HS07]. Ebenfalls wird auf Basis dieser Interaktionsformen ein Aufmerksamkeitsgrad ermittelt, welcher sowohl der bzw. dem Vortragenden als auch den Teilnehmerinnen und Teilnehmern als Feedback zur Verfügung steht.

Diese Ausführungen lassen einen Zusammenhang mit sogenannten Audience Response Systems (kurz ARS) erkennen, denn diese ermöglichen ebenfalls das Stellen von Fragen an die Teilnehmerinnen und Teilnehmer einer Vorlesung direkt im Hörsaal [Tob05]. Aus diesem Grund werden im Kapitel 2 einige Studien zum Thema ARS präsentiert. Diese Studien bescheinigen einem ARS weitgehend positive Auswirkungen auf die Mitarbeit und die Aufmerksamkeit (vgl. z.B. Silliman und McWilliams [SM04] oder Helmerich und Scherer [HS07]). Dabei wird jedoch auch darauf hingewiesen, dass der Einsatz solcher Technologien nicht übertrieben werden sollte, da dadurch unter Umständen ein gegenteiliger Effekt erreicht wird.

Die Entwicklung des Web-Portals ist im Kapitel 3 ausführlich dokumentiert. Dabei zeigt am Beginn ein Überblick den grundlegenden Aufbau. Es ist ersichtlich, dass das Web-Portal mit Hilfe eines Web-Frameworks entwickelt wurde (siehe Abschnitt 3.2). Darauf baut eine Komponente zur Bereitstellung der Verwaltung der Benutzerprofile und der Veranstaltungen, das sogenannte Grundgerüst, auf. Dieses wird im Abschnitt 3.3 näher erläutert. Daneben existiert eine Schnittstelle auf deren Basis die bereits erwähnten Interaktionsformen implementiert sind (siehe Abschnitte 3.4 und 3.6). Diese Schnittstelle stellt auch ein Framework, auf welches im Abschnitt 3.5 näher eingegangen wird, zur Berechnung des Aufmerksamkeitsgrads zur Verfügung.

Nach der ausführlichen Dokumentation der Entwicklung des Web-Portals beschreibt Kapitel 4 die Evaluation anhand einiger Einsätze zur Unterstützung von Vorlesungsübertragungen und Kapitel 5 diskutiert die bei der Evaluation gewonnenen Erkenntnisse. Dabei zeigt sich, dass das Web-Portal durchaus das Potential zur erfolgreichen Unterstützung der Mitarbeit und der Aufmerksamkeit hat. Es müssen jedoch dazu einige Punkte beachtet werden, um nicht den gegenteiligen Effekt zu erreichen. So ist zum Beispiel die Möglichkeit zum Stellen von zum Inhalt passenden Fragen sehr wohlwollend empfunden worden. Probleme zeigten sich jedoch bei der Verwendung von Interaktionsformen, welche in keinem Zusammenhang

mit dem Web-Inhalt stehen. Hier erzeugt eine einführende Erklärung, dass diese Interaktionsformen zur Erhaltung der Aufmerksamkeit dienen, größere Akzeptanz und ermöglicht dadurch positive Auswirkungen auf die Mitarbeit und die Aufmerksamkeit.

Abschließend lässt sich die gestellte Forschung insofern beantworten, dass das Web-Portal verschiedene Web-Inhalte, vor allem Vorlesungsübertragungen, unterstützen kann und somit in der Lage ist, einen Beitrag zur verbesserten Mitarbeit und zur Erhaltung bzw. Erhöhung der Aufmerksamkeit zu leisten. Dies geschieht durch die Bereitstellung von verschiedenen synchronen Interaktions- und Kommunikationsmöglichkeiten. Ein wichtiger Schlüssel zum erfolgreichen Einsatz ist dabei die aktive Unterstützung sowohl von den Vortragenden als auch von den Teilnehmerinnen und Teilnehmern.

Literatur

- [And12] Keith Andrews. *Information Architecture and Web Usability Lecture Notes*. 2012. URL: <http://courses.iicm.tugraz.at/iaweb/iaweb.pdf> (besucht am 18. 10. 2012) (siehe S. 29).
- [BJ09] Jeffrey L. Bailie und Michael A. Jortberg. »Online Learner Authentication: Verifying the Identity of Online Users«. In: *MERLOT Journal of Online Learning and Teaching* 5.2 (Juni 2009) (siehe S. 16).
- [CD00] Alison Carr-Chellman und Philip Duchastel. »The ideal online course«. In: *British Journal of Educational Technology* 31.3 (2000), S. 229–241. ISSN: 1467-8535. URL: <http://dx.doi.org/10.1111/1467-8535.00154> (siehe S. 5, 6, 101).
- [Cuto8] Euline Schmid Cutrim. »Using a voting system in conjunction with interactive whiteboard technology to enhance learning in the English language classroom«. In: *Comput. Educ.* 50 (1 Jan. 2008), S. 338–356. ISSN: 0360-1315. URL: <http://dl.acm.org/citation.cfm?id=1296335.1296517> (siehe S. 11, 12).
- [Dja12a] Django Team. *Django: Design philosophies*. 2012. URL: <https://docs.djangoproject.com/en/1.4/misc/design-philosophies/> (besucht am 09. 10. 2012) (siehe S. 21).
- [Dja12b] Django Team. *Django Documentation*. 2012. URL: <https://docs.djangoproject.com/en/1.4/> (besucht am 09. 10. 2012) (siehe S. 22, 24, 25, 27, 31, 32, 72).
- [Ebn10] Martin Ebner. *Technology Enhanced Learning - Lecture Notes*. 2010. URL: http://tugtc.tugraz.at/wbtmaster/courses/TEL_starte.htm (besucht am 02. 12. 2012) (siehe S. 7).

Literatur

- [GM07] Jorge Gaytan und Beryl C. McEwen. »Effective Online Instructional and Assessment Strategies«. In: *American Journal of Distance Education* 21.3 (2007), S. 117–132. URL: <http://www.tandfonline.com/doi/abs/10.1080/08923640701341653> (siehe S. 15).
- [Her+08] J.A. Hernandez u. a. »Biometrics in Online Assessments: A Study Case in High School Students«. In: *18th International Conference on Electronics, Communications and Computers CONIELECOMP 2008*, März 2008, S. 111–116 (siehe S. 15, 16).
- [HK12] Adrian Holovaty und Jacob Kaplan-Moss. *The Django Book*. 2012. URL: <http://www.djangobook.com/> (besucht am 09.10.2012) (siehe S. 20–23, 72).
- [Holo00] Andreas Holzinger. *Basiswissen Multimedia. Band 2: Lernen*. Würzburg: Vogel-Verlag, 2000 (siehe S. 7).
- [HP12] Christian Haintz und Karin Pichler. *RealFeedback*. 2012. URL: <https://realfeedback.tugraz.at/> (besucht am 29.11.2012) (siehe S. 14).
- [HS07] Jürgen Helmerich und Jürgen Scherer. »Interaktion zwischen Lehrenden und Lernenden in Medien unterstützten Veranstaltungen«. In: *Neue Trends im E-Learning*. Hrsg. von Michael H. Breitner, Beate Bruns und Franz Lehner. Physica-Verlag HD, 2007, S. 197–210. ISBN: 978-3-7908-1922-9 (siehe S. 6, 12–14, 102).
- [Jam90] William James. *The Principles of Psychology*. 1890. URL: <http://psychclassics.yorku.ca/James/Principles/prin11.htm> (besucht am 02.12.2012) (siehe S. 7).
- [LM05] R. Latessa und D. Mouw. »Use of an audience response system to augment interactive learning.« In: *Family Medicine* 37.1 (2005), S. 12–14 (siehe S. 1, 8, 9).
- [Nor92] Donald Norman. *The Design of Everyday Things*. Basic Books, 1992 (siehe S. 28).
- [Pek+02] Reinhard Pekrun u. a. »Academic Emotions in Students' Self-Regulated Learning and Achievement: A Program of Qualitative and Quantitative Research«. In: *Educational Psychologist* 37.2 (2002), S. 91–105 (siehe S. 10, 98).

- [Pro12] Stefano Probst. *Django Model-Template-View(MTV) - CC-by-sa 3.0/at*. 2012. URL: http://de.wikipedia.org/w/index.php?title=Datei:MTV_Django.svg&page=1&filetimestamp=20120208222359 (besucht am 18. 11. 2012) (siehe S. 25).
- [RW02] C. Ricketts und S. J. Wilks. »Improving Student Performance Through Computer-based Assessment: Insights from recent research«. In: *Assessment & Evaluation in Higher Education* 27.5 (2002), S. 475–479. URL: <http://www.tandfonline.com/doi/abs/10.1080/0260293022000009348> (siehe S. 16).
- [SHB04] Gavin Sim, Phil Holifield und Martin Brown. »Implementation of computer assisted assessment: lessons from the literature«. In: *Research in Learning Technology* 12.3 (2004) (siehe S. 16).
- [SM04] SE. Silliman und L. McWilliams. »Observations on Benefits / Limitations of an Audience Response System«. In: *Proceedings of the American Society for Engineering Education Annual Conference*. Salt Lake City, Utah, Juni 2004 (siehe S. 10, 11, 88, 98, 102).
- [SN07] Jeffrey R. Stowell und Jason M. Nelson. »Benefits of Electronic Audience Response Systems on Student Participation, Learning, and Emotion«. In: *Teaching of Psychology* 34.4 (2007), S. 253–258. URL: <http://www.tandfonline.com/doi/abs/10.1080/00986280701700391> (siehe S. 9, 10).
- [Tob05] Brian Tobin. *Audience Response Systems*. Stanford University School of Medicine. 7. Nov. 2005. URL: http://med.stanford.edu/irt/edtech/contacts/documents/2005-11_AAMC_tobin_audience_response_systems.pdf (besucht am 09. 10. 2012) (siehe S. 7, 14, 102).
- [YEA08] Jeff Yan und Ahmad Salah El Ahmad. »Usability of CAPTCHAs or usability issues in CAPTCHA design«. In: *Proceedings of the 4th symposium on Usable privacy and security*. SOUPS '08. Pittsburgh, Pennsylvania: ACM, 2008, S. 44–52. ISBN: 978-1-60558-276-4. URL: <http://doi.acm.org/10.1145/1408664.1408671> (siehe S. 62).