

Stephan Moser

Learning Analytics in Massive Open Online Courses

Diplomarbeit

Technische Universität Graz

Institut für Informationssysteme und Computermedien

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

Mitbetreuer: Dipl.-Ing. Behnam Taraghi

Graz, Mai 2015

This document is set in Palatino, compiled with pdfL^AT_EX₂ε and Biber.

The L^AT_EX template from Karl Voit is based on KOMA script and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

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

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 entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____

Datum

Unterschrift

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

Dedikation

Diese Diplomarbeit ist meiner Familie gewidmet.

Ohne ihrer Liebe und immerwährenden Unterstützung im Laufe meines Lebens und insbesondere meines Studiums, wäre es nicht möglich gewesen, diese Arbeit anzufertigen.

Danksagung

Ohne die Hilfe von einigen Personen wäre es nicht möglich gewesen diese Arbeit zu realisieren. Deswegen möchte ich mich bei diesen Personen für ihre Unterstützung und Hilfe hier bedanken.

Zuerst möchte ich mich bei Univ.-Doz. Dipl.-Ing. Dr.techn. Martin Ebner für seine Führung, Unterstützung und Geduld, im Zuge des Zeitraums dieser Arbeit, bedanken. Weiters möchte ich mich bei Karl Wiesenhofer bedanken, der bei technischen Fragen stets zur Verfügung stand.

Mein Dank gilt auch all meinen Freunden für ihre großartige Unterstützung und Freundschaft während dieser Arbeit.

Schlussendlich bedanke ich mich aber bei meiner Familie und meiner Freundin für die uneingeschränkte Unterstützung während meines Studiums.

Kurzfassung

Diese Arbeit befasst sich mit dem Thema "Learning Analytics" in "Massive Open Online Courses". Zu diesem Zweck wird eine von der Technischen Universität Graz in Kooperation mit der Karl Franzens Universität Graz 2014 erstellte Kursplattform mit dem Namen iMoox untersucht. Diese Plattform stellt eine immer größer werdende Vielfalt an unterschiedlichsten Kursen zur Verfügung, die zum Teil auf in den jeweiligen Universitäten angebotenen Lehrveranstaltungen basieren und jeder Kurs multimediale Lehrinhalte anbietet.

Die Forschungsarbeit fokussiert sich sowohl auf die Erfassung und Darstellung des Verhaltens des/der Studenten/in auf dieser Plattform, als auch auf den sinnvollen Einsatz von Lehrmitteln. Dazu wird das Lernverhalten bezüglich verschiedenster Aktivitäten analysiert.

Zum Zweck der Analyse und Auswertung der Aktivitäten wurde eine Software entwickelt, die dem/der BenutzerIn der Software automatisch Informationen darstellt. Weiters ist es möglich die Beliebtheit der Lehrmittel bezüglich ihrer Konsumationsquantität z. B. Häufigkeit eines Dateidownloads und auch ihrer Konsumationsqualität z. B. Zeitsprünge in Lehrvideos, zu untersuchen.

Die durch das Analysesystem angebotenen Informationen sind jeweils ta-

gesaktuell und werden dem/der BenutzerIn in Diagrammen und textueller Form über eine Internetplattform angeboten.

Durch diese Software ist es möglich die Auswirkungen von gut betreuten Kursen und vernünftig zusammengestellten Lehrmitteln auf die Kollaboration und Aktivität von StudentenInnen zu erkennen.

Weiters befasst sich diese Arbeit auch mit einigen detaillierteren Untersuchungen und ihren Auswirkungen auf die weitere Entwicklung der iMoox Plattform.

Abstract

In this paper we deal with the topic of Learning Analytics in Massive Open Online Courses. Therefore we have examined a platform which has been introduced by Graz University of Technology and University of Graz in 2014 named iMoox. This platform provides a growing number of courses partly based on "real" classes offered by the involved universities, whereby every course provides varying kinds of multimedial teaching aids. Our research focused on the students' behaviour on this platform, but also on the reasonableness of the teaching aids. We examined different students' habits.

For the purpose of analysis and examination of the students' activities we have built up an analysis platform, which automatically shows the user of this platform information .

Further it is possible to examine the popularity of the offered teaching aids regarding their consumption quantity, e.g. the frequency of a file download, but also by the quality of their consumption, e.g. the timesteps the students have jumped to in a teaching video.

The implemented system updates on a daily basis and presents the data mostly in a chart format on a web platform.

This analysis shows us the impact of well conducted courses and reasonable and well elaborated teaching aids on the students' activities and collaboration on the platform.

In this paper we concern ourselves with some detailed examinations and their implications for the further development of the iMoox platform.

Inhalt

Abstract	ix
1. Einleitung	1
1.1. Gliederung der Arbeit	2
2. Forschungsüberblick	5
2.1. Learning Analytics	5
2.1.1. Business Intelligence	7
2.1.2. Web Analytics	8
2.1.3. Academic Analytics	9
2.1.4. Educational Data Mining	9
2.2. Massive Open Online Courses	10
2.2.1. iMoox	12
2.2.2. Coursera	13
2.2.3. edX	14
2.2.4. Udacity	14
2.2.5. Stanford Online	15
2.2.6. iversity	17
2.2.7. OpenCourseWorld	17
2.2.8. openHPI	18
2.3. Analyse bestehender Learning-Analytics-Systeme	18
2.3.1. LASyM	19

Inhalt

3. Umsetzung	23
3.1. Funktionalität	23
3.2. Methodik	30
3.2.1. Log Daten	34
3.3. Architektur	38
3.4. Server	40
3.5. Datenbank	44
3.6. Web Application	44
4. Evaluation und Diskussion	47
4.1. Kollaboration	47
4.2. Intervention	52
4.3. Personalisierung & Adaptierung	54
4.4. Kritik	55
4.4.1. Datensicherheit	55
4.4.2. Informationsvisualisierung	56
5. Zusammenfassung	57
6. Ausblick	61
6.1. Integration von Learning Analytics in die MOOC-Plattform	61
6.2. Weiterentwicklung der Web-Plattform	62
6.3. Verbesserungen in der Datenbank	62
A. Quelltext	67
A.1. Verarbeitung am Server (Serialization)	67
A.2. Verarbeitung Server (Deserialization)	71
A.3. Datenbank Struktur	91
Bibliography	99

Abbildungen

2.1. Was ist ein MOOC?	12
2.2. Screenshot iMoox	13
2.3. Screenshot Coursera	14
2.4. Screenshot edX	15
2.5. Screenshot Udacity	16
2.6. Screenshot Stanford Online	16
2.7. Screenshot iversity	17
2.8. Screenshot Open Course World	18
2.9. Screenshot openhpi	19
2.10. LASyM	21
3.1. Landing Page LA	24
3.2. Logins LA	24
3.3. Beliebteste Downloads	25
3.4. Downloader LA	26
3.5. Forum Reading	26
3.6. Forum Writing	27
3.7. Quizzes Page 1	28
3.8. Quizzes Page 2	28
3.9. Quizzes Vergleich	29
3.10. Videos Page	30
3.11. Landing Page LA	31

Abbildungen

3.12. Parameter Dashboard	32
3.13. Kurs Dashboard	32
3.14. Applikationsablauf	33
3.15. Systemarchitektur	39
3.16. Datenbank	45
4.1. Datei Downloads GOL	48
4.2. Videokonsumation GOL	49
4.3. Forumaktivität GOL und LIN	50
4.4. Quiz vs. Downloads GOL	51
4.5. Quiz vs. Forum GOL	53

1. Einleitung

Zweifelsfrei nimmt das Internet bzw. Online-Plattformen bei der Gestaltung und Verwaltung des täglichen Lebens eines/r Einzelnen eine immer größer werdende Rolle ein. Aus eigener Erfahrung werden alltägliche Einkäufe über Online-Plattformen (E-Commerce) abgewickelt, mediale Inhalte werden über das Internet konsumiert, die Kommunikation verlagert sich immer mehr auf Online Dienste und sogar Verwaltungsangelegenheiten (E-Government) werden immer öfter online erledigt. Auch die Bildung (E-Learning) folgt diesem Trend. Der voranschreitende Einsatz von Massive Open Online Courses (MOOCs) ermöglicht es, Wissen auf Universitätsniveau einer breiten Masse zur Verfügung zu stellen. Der große Vorteil von MOOCs begründet sich in seiner Eigenschaft, beinahe unbegrenzt vielen Teilnehmern Bildung zukommen zu lassen. (Robes, 2012)

Nichtsdestotrotz bedarf es weiterer Forschung in diesem Bereich, da es Beobachtungen gibt die zeigen, dass trotz vielen Kursteilnehmern sich nur wenige an einem solchen Kurs auch bis zum Ende aktiv beteiligen und zu einem Abschluss kommen. (Jordan, 2015) Daraus lassen sich für die zukünftige MOOCs wichtige Verbesserungen ableiten:

- Senken der Abbruchrate
- Verbesserung der Lehrmethodik
- Verbesserung der Lehrmittel

1. Einleitung

- Erhöhung der Kollaboration von Studierenden

Dieser Thematik widmet sich diese Arbeit. Es wird ein Learning Analytics System für MOOCs (hier: imoox.at¹) vorgestellt, dessen Ziel es ist, die durch Studierende auf der Plattform generierten großen Datenmengen zu analysieren, um dadurch Rückschlüsse für Lehrende zur Gestaltung von Lehrmitteln und zum Kursablauf zu gewinnen. Weiters ist es möglich die Auswirkungen von gut betreuten Kursen und vernünftig zusammengestellten Lehrmitteln auf die Kollaboration und Aktivität von Studierenden zu erkennen.

Im folgenden Abschnitt 1.1 wird ein kurzer Überblick über die Gliederung der Arbeit gegeben.

1.1. Gliederung der Arbeit

In Kapitel 2 wird erklärt welche Bedeutung die Begriffe "Massive Open Online Course" und "Learning Analytics" haben, und welche Arbeiten es zu diesen Themen bereits gibt. Es werden beispielhaft einige der größten MOOC-Plattformen erwähnt und auch bestehende Arbeiten in der Verbindung von MOOCs und Learning Analytics untersucht.

Kapitel 3 beschreibt die detaillierte Umsetzung des Learning-Analytics-Systems auf Basis von imoox.at.

Eine Evaluation und eine anschließende Diskussion dieser Arbeit wird in Kapitel 4 vorgenommen.

¹<https://www.imoox.at> letzter Aufruf: 16.05.2015

1.1. Gliederung der Arbeit

Abschließend gibt es in Kapitel 5 eine nochmalige Zusammenfassung der Arbeit bevor in 6 zukünftige Entwicklungen und möglicher Optimierungen der Plattform analysiert werden.

2. Forschungsüberblick

Im Forschungsüberblick wird die Bedeutung der Begriffe Learning Analytics und MOOC untersucht, ob es bereits Projekte gibt, die diese zwei Begriffe zusammenführen und wie der aktuelle Stand der Technik hierzu aussieht. Learning Analytics wird in Abschnitt 2.1 erläutert. In Abschnitt 2.2 werden Massive Open Online Courses untersucht und in Abschnitt 2.3 wird ein Überblick über am "Markt" befindlichen Software-Lösungen für Learning Analytics und MOOCs gegeben.

2.1. Learning Analytics

"Learning analytics refers to the interpretation of a wide range of data produced by and gathered on behalf of students in order to assess academic progress, predict future performance, and spot potential issues. Data are collected from explicit student actions, such as completing assignments and taking exams, and from tacit actions, including online social interactions, extracurricular activities, posts on discussion forums, and other activities that are not directly assessed as part of the student's educational progress. Analysis models that process and display the data as-

2. Forschungsüberblick

sist faculty members and school personnel in interpretation. The goal of learning analytics is to enable teachers and schools to tailor educational opportunities to each student's level of need and ability. Learning analytics need not simply focus on student performance. It might be used as well to assess curricula, programs and institutions. It could contribute to existing assessment efforts on a campus, helping provide a deeper analysis, or it might be used to transform pedagogy in a more radical manner. It might also be used by students themselves, creating opportunities for holistic synthesis across both formal and informal learning activities." (Johnson et al., 2011, p. 28)

Diese, im Horizon Report 2011 des New Media Consortium erschienene Begriffsdefinition, war eine der ersten zu diesem Thema.

Learning Analytics ist ein aufstrebendes Forschungsfeld, in dem komplexe Analysesysteme dazu genutzt werden Lernen und Bildung zu optimieren, indem sie Daten von Studierenden sammeln, interpretieren und daraus Lernfortschritte erkennen, Probleme erfassen und Vorhersagen für die Zukunft treffen. Es leitet sich von einer Reihe anderer Forschungsbereiche ab, wie z.B. Business Intelligence, Web Analytics, Academic Analytics and Educational Data Mining. (Elias, 2011) Diese Begriffe werden in den Kapiteln 2.1.1, 2.1.2, 2.1.3, 2.1.4 beschrieben.

Anders als Educational Data Mining, welches auf systemgenerierter und automatisierter Rückmeldung für Studierende basiert, ermöglicht Learning Analytics "maßgeschneiderte" Rückmeldung, beispielsweise durch Adaptieren von Lehrmitteln, Eingreifen bei "Risikostudierenden" oder auch durch Feedback für Studierende und integriert die Lehrperson in die Entscheidungsfindung. (Ebner et al., 2014)

2.1. Learning Analytics

Eine Schlüsselanwendung von Learning Analytics ist das Beobachten und Vorhersagen des Lernfortschritts von Studierenden und das frühzeitige Erkennen von Problemen (durch die Lehrperson) zur Identifizierung von "Risikostudierenden", welche einen Kurs nicht bestehen könnten. (Johnson et al., 2011)

Durch den Ruf nach mehr Transparenz von, und Kontrolle über die Leistungen ihrer Studierenden, zeigen Universitäten immer größeres Interesse am Einsatz von Learning Analytics.

Zusammenfassend beschreibt Bienkowski, Feng, and Means (2012) Learning Analytics als System zur Beantwortung von Fragen wie:

- Wann sind Studierende bereit zum nächsten Thema voranzuschreiten?
- Wann fallen Studierende im Stoff zurück?
- Wie hoch ist das Risiko, dass Studierende ihre Kurse nicht abschließen?
- Welche Note würden Studierende ohne Intervention bekommen?
- Welchen anderen, nächsten Kurse sollten Studierende belegen?
- Sollte dem Studierenden Hilfe angeboten werden?

2.1.1. Business Intelligence

Unter dem Begriff Business Intelligence wird laut Rud (2009) eine Reihe von Techniken und Werkzeugen zur Überleitung von rohen Daten zu sinnvollen und aussagekräftigen Informationen beschrieben, die für die Analyse bei wirtschaftlichen Aufgaben eingesetzt werden.

Business-Intelligence-Technologien helfen dabei, große Mengen unstrukturierter Daten zu verarbeiten um dadurch neue Geschäftsmöglichkeiten

2. Forschungsüberblick

zu identifizieren und zu entwickeln. Das Ziel diese großen Datenmengen richtig zu interpretieren, hilft dabei, einen Marktvorteil und langfristige Geschäftsstabilität zu erreichen.

2.1.2. Web Analytics

“Web Analytics is the measurement, collection, analysis and reporting of Internet data for the purposes of understanding and optimizing Web usage.” (*Web Analytics Definitions* 2008, p. 3)

Web Analytics ist ein Sammelbegriff für den Einfluss von Inhalten auf Internetseiten auf ihre BenutzerInnen. Vor allem Unternehmen im Bereich von ECommerce, aber auch andere benutzen Web Analytics Software um konkrete Informationen über das Verhalten der BenutzerInnen zu erfahren. Folgend sind demonstrativ einige typische Fragestellungen aufgelistet.

- Wie viele BesucherInnen hat meine Webseite?
- Wie viele dieser BesucherInnen sind eindeutig (mehrere Besuche werden nicht gezählt)?
- Woher kommen meine BesucherInnen?
- Welche Suchbegriffe verwenden meine BenutzerInnen?
- Wie lange befinden sich BenutzerInnen auf der Internetseite?
- Welche Links wurden von den Benutzern angeklickt?

2.1.3. Academic Analytics

Academic Analytics wird definiert als der Prozess der Evaluierung und Analyse von Organisationsdaten aus universitären Systemen, zum Zweck der Entscheidungstreffung und Berichterstattung. (Campbell and Oblinger, 2007)

Akademikerquoten sind im Steigen begriffen. Dadurch steigt auch die Verantwortlichkeit in den betreffenden universitären Instituten. Academic Analytics können diesen Instituten dabei helfen, ihre Verantwortlichkeit im Bereich der Lehre wahrzunehmen und gleichzeitig ihre akademischen Aufgaben zu erfüllen, ohne dass die Kosten in diesem Bereich signifikant steigen. Akademische Systeme generieren eine große Bandbreite an Daten und durch Academic Analysis ist es möglich, diese Daten mit statistischen Techniken und Vorhersagemodellen zu verknüpfen, um zu ermöglichen, vorzeitig Maßnahmen zur verantwortungsvollen Lehre zu treffen.

2.1.4. Educational Data Mining

“Educational data mining exploits statistical, machine-learning, and data-mining algorithms over different types of educational data. EDM is concerned with developing methods to explore the unique types of data in educational settings and, using these methods, to better understand students and the settings in which they learn. EDM seeks to develop computational approaches that combine data and theory to transform practice.” (Romero and Ventura, 2010, p. 601)

2. Forschungsüberblick

Baker (2011, p. 8) betrachtet folgende Aufzählung als das Ziel seiner Forschung:

1. Vorhersage über das Lernverhalten von Studenten durch Etablierung von Modellen mit detaillierten Informationen über Studenten wie
 - Wissen
 - Motivation
 - Metakognition
 - Einstellung
2. Entdecken und Verbessern von Modellen zur Charakterisierung von Inhalten die gelernt werden sollen und optimal unterrichtlichen Abfolgen
3. Erlernen der Effekte von unterschiedlichen Arten der pädagogischen Unterstützung, die durch die Anwendung von Lernsoftware ermöglicht wird
4. Verbessern des wissenschaftlichen Wissens über "das Lernen" und über "Lernende" durch den Aufbau von Computermodellen

2.2. Massive Open Online Courses

"Ein MOOC (Massive Open Online Course) ist eine spezielle Form von Onlinekursen, bei der traditionelle Formen der Wissensvermittlung wie Videos, Lesematerial und Problemstellungen mit Foren, in denen Lehrende und Lernende miteinander kommunizieren können, und Quizzes, anhand derer die Lernenden ihren Wissenserwerb überprüfen können, kombiniert werden. MOOCs eignen sich optimal, um Lerninhalte einer großen Zahl von Interessierten zeit- und ortsunabhängig näher zu bringen. Eine Obergrenze betreffend die Teilnehmeranzahl gibt es

2.2. Massive Open Online Courses

bei dieser Form von Onlinekursen nicht. Generell stellen MOOCs aufgrund ihrer multimedialen Aufbereitung einen niederschweligen Zugang zu wissenschaftlich fundierten Informationen dar.“
(iMoox 2014)

Wie der Name es nahelegt, basiert das Konzept von MOOCs auf vier Grundprinzipien:

- **Massive:** Es gibt (ausgenommen technisch) keine Einschränkung der Studierendenzahl, was einen fundamentalen Unterschied zu Kursen in einem universitären Umfeld darstellt, da die Anzahl der Studierenden stark anwachsen kann.
- **Open:** Abbildung 2.1¹ regt dazu an, diesen Punkt zu diskutieren. In der Praxis stellen MOOC-Anbieter ihre Inhalte kostenlos zur Verfügung. "Open" bedeutet aber gleichzeitig die rechtliche Unbeschränktheit bei der Wiederverwendung und Verbreitung der angebotenen Inhalte. Ebner et al. (2014) befasst sich mit dem "O" in MOOCs.
- **Online:** Inhalte werden rein online vermittelt. Es gibt keinen physisch - persönlichen Kontakt zwischen Lehrenden und Studierenden. Online Kommunikation und Kollaboration ist jedoch eine Grundlage von MOOCs.
- **Course:** Es handelt sich dabei um einen Online-Kurs, der Wissen vermitteln, und sich vom Ergebnis nicht von einem konventionellen Kurs unterscheiden soll.

Bei MOOCs gilt es in erster Linie zwischen **xMOOCs** und **cMOOCs** eine Unterscheidung zu treffen. xMOOCs sind zumeist auf Video aufgezeichnete Massenkurse aus der Vergangenheit, die mit einer Prüfung abgeschlossen

¹<http://www.flickr.com/photos/mathplourde/8620174342/sizes/1/in/photostream/>
letzter Aufruf: 16.05.2015

2. Forschungsüberblick

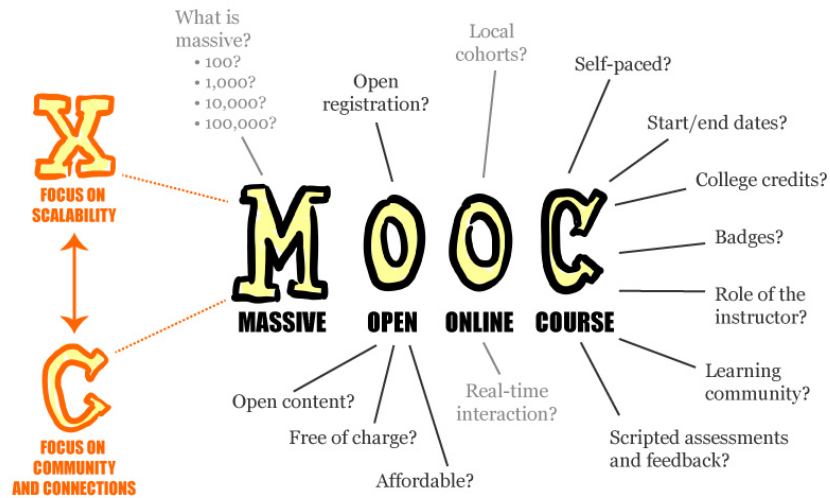


Abbildung 2.1.: Was ist ein MOOC? (Mathieu Plourde)

werden, wobei im Gegensatz dazu cMOOCs eher Seminarcharacter haben und auf Kooperation und Konnektivismus beruhen. (Schulmeister, 2013)

Im den folgenden Abschnitten werden ein paar bekannte MOOC Anbieter vorgestellt.

2.2.1. iMoox²

iMooX, im Dezember 2013 von der Karl-Franzens-Universität Graz und der Technischen Universität Graz gegründet, ist Österreichs erste und bisher einzige MOOC-Plattform. Auf iMooX werden freie Online-Kurse (Massive Open Online Courses) zu unterschiedlichen Themen angeboten, die allen Interessentinnen und Interessenten kostenlos, zeit- und ortsunabhängig zur

²<https://www.imoox.at> (letzter Aufruf: 11. 03. 2015)

2.2. Massive Open Online Courses



Abbildung 2.2.: Screenshot iMoox

Verfügung stehen (iMoox 2014). In Abbildung 2.2 ist ein Screenshot der Website zu sehen.

2.2.2. Coursera³

Coursera ist eine MOOC-Plattform aus den USA. Sie gilt als eine der führenden MOOC-Anbieter und arbeitet weltweit mit Universitäten zusammen, deren Lehre als überdurchschnittlich angesehen wird. In Abbildung 2.3 ist ein Screenshot der Website zu sehen.

³<https://www.coursera.org> (letzter Aufruf: 11. 03. 2015)

2. Forschungsüberblick

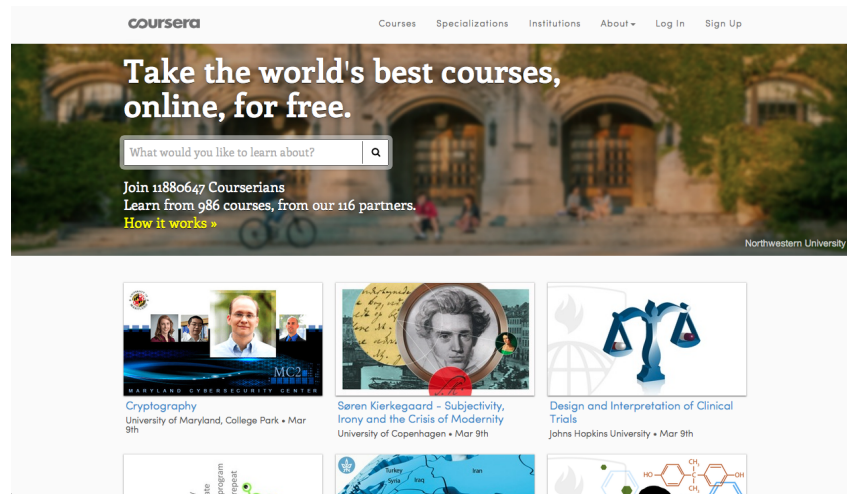


Abbildung 2.3.: Screenshot Coursera

2.2.3. edX⁴

edX ist ebenfalls eine nicht kommerzielle Online-Initiative, gegründet in Partnerschaft von der Universität Harvard und dem Massachusetts Institute of Technology. Es bietet interaktive Online-Kurse und MOOCs von MITx, HarvardX, BerkeleyX, UTx und vielen anderen Universitäten. In Abbildung 2.3 ist ein Screenshot der Website zu sehen.

2.2.4. Udacity⁵

US-amerikanische Plattform, die vor allem Kurse im Informatikbereich anbietet. Die Plattform ist darauf spezialisiert die Fähigkeiten zu lehren,

⁴<https://www.edx.org> (letzter Aufruf: 11. 03. 2015)

⁵<https://www.edx.org> (letzter Aufruf: 11. 03. 2015)

2.2. Massive Open Online Courses

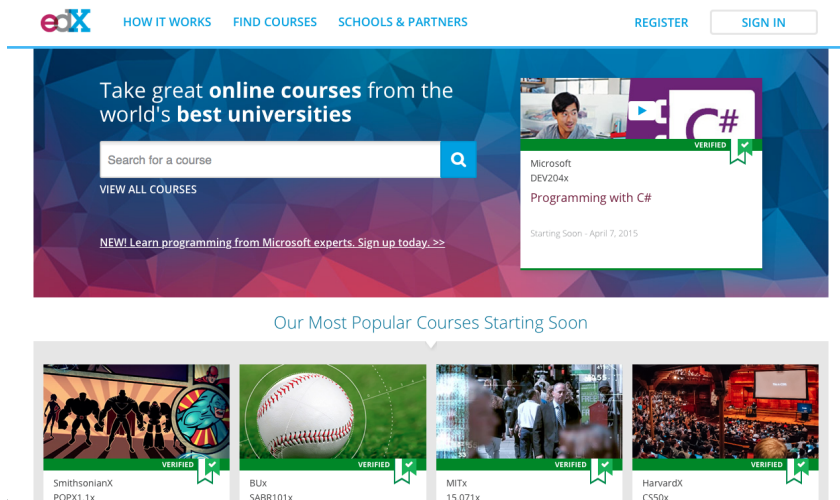


Abbildung 2.4.: Screenshot edX

die in der IT-Branche (vor allem im Silicon Valley) benötigt werden. Diese kooperiert mit den Branchengrößen Google, AT&T, Facebook, Salesforce, Cloudera, etc. In Abbildung 2.5 ist ein Screenshot der Website zu sehen.

2.2.5. Stanford Online⁶

Nicht kommerzielles MOOC-Angebot der Stanford University. Die Plattform basiert auf der OpenEdx platform, eine von edX angebotene Open-Source-MOOC-Basis. In Abbildung 2.6 ist ein Screenshot der Website zu sehen.

⁶<http://online.stanford.edu> (letzter Aufruf: 11. 03. 2015)

2. Forschungsüberblick

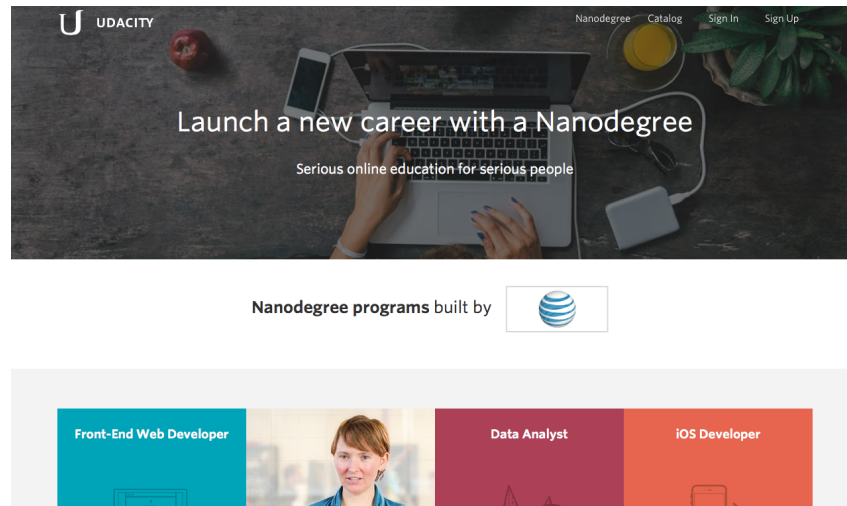


Abbildung 2.5.: Screenshot Udacity

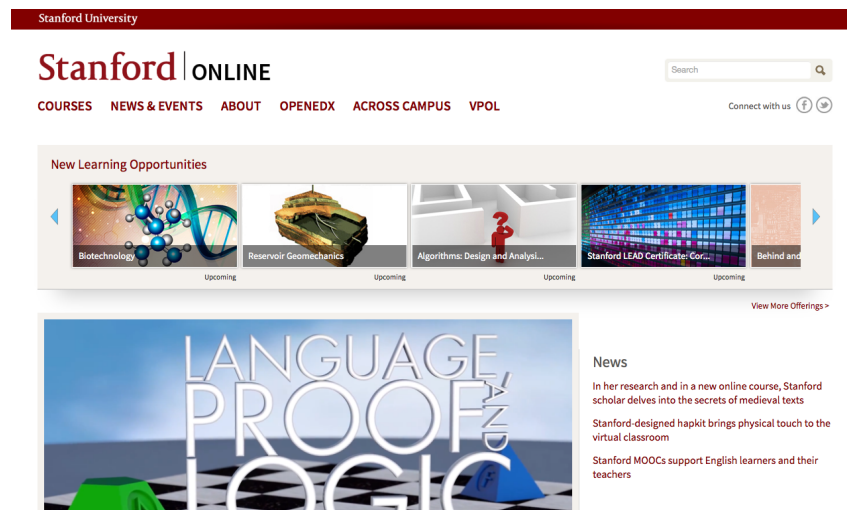


Abbildung 2.6.: Screenshot Stanford Online

2.2. Massive Open Online Courses

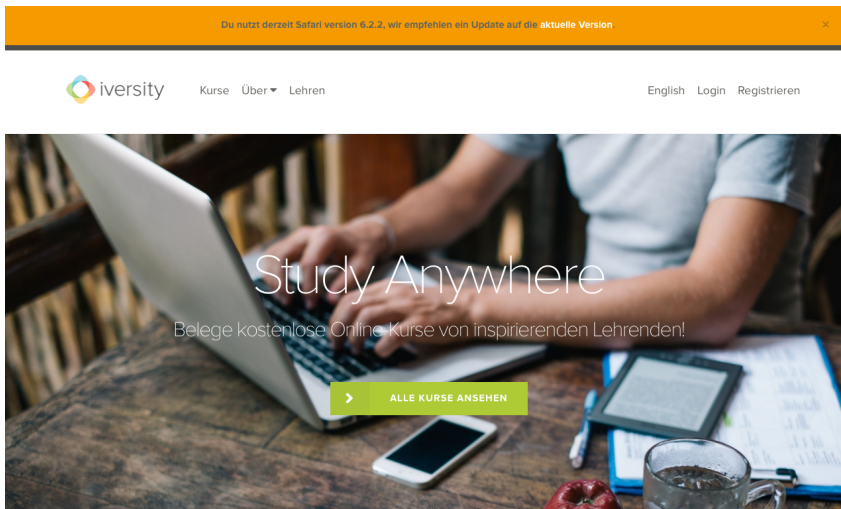


Abbildung 2.7.: Screenshot iversity

2.2.6. iversity⁷

Kommerzieller Anbieter von MOOCs aus Berlin. In Abbildung 2.7 ist ein Screenshot der Website zu sehen.

2.2.7. OpenCourseWorld⁸

Deutsche MOOC-Plattform, betrieben von der IMC AG, ein Softwareunternehmen im E-Learning-Bereich. In Abbildung 2.8 ist ein Screenshot der Website zu sehen.

⁷<https://www.iversity.org> (letzter Aufruf: 11. 03. 2015)

⁸<https://www.opencourseworld.de> (letzter Aufruf: 11. 03. 2015)

2. Forschungsüberblick



Abbildung 2.8.: Screenshot Open Course World

2.2.8. openHPI⁹

Eine von vom Hasso-Plattner-Institut betriebene MOOC-Plattform mit Schwerpunkt Informationstechnologie (IT). In Abbildung 2.9 ist ein Screenshot der Website zu sehen.

2.3. Analyse bestehender Learning-Analytics-Systeme

Learning Analytics in Massive Open Online Courses stellt ein sehr junges Forschungsgebiet dar. Der Fokus der meisten früheren Arbeiten bezüglich

⁹<https://open.hpi.de> (letzter Aufruf: 11. 03. 2015)

2.3. Analyse bestehender Learning-Analytics-Systeme

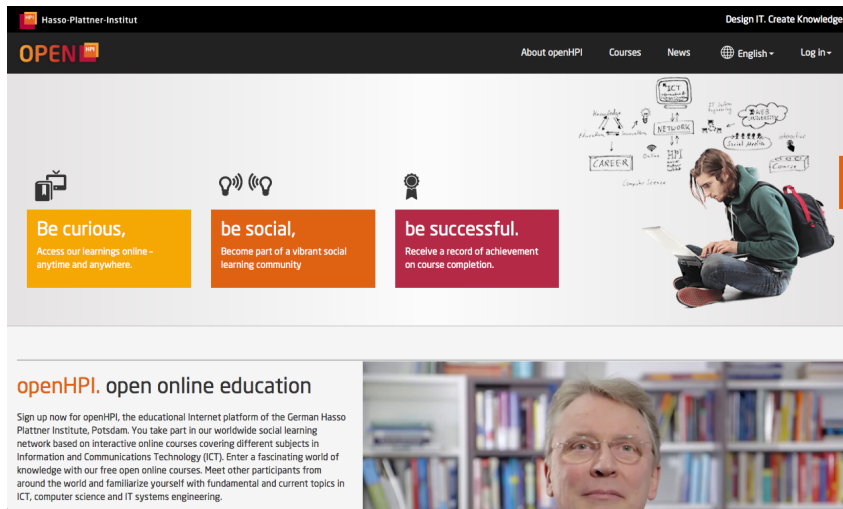


Abbildung 2.9.: Screenshot openhpi

Learning Analytics lag auf LMS (Learning Management Systems) bzw. auf der hohen Dropout Rate bei MOOCs.

Im Folgenden werden bereits bestehende Learning-Analytics-Systeme vorgestellt, die am ehesten mit dem hier verfolgten Ansatz verglichen werden können.

2.3.1. LASyM

LASyM steht für "A Learning Analytics System for MOOCs" und wurde von Yassine Tabaa und Abdellatif Medouri aus der Abdelmalek Essaadi University Tetouan, Morocco entwickelt. Der Fokus dieser Softwarelösung liegt darauf, ein Learning-Analytics-System zu entwickeln, welches "at-risk" Studierende erkennt und mit sehr großen Datenmengen umgehen kann. Aus

2. Forschungsüberblick

diesem Grund wurde **Apache's Hadoop MapReduce Library**¹⁰ genutzt.

Zur Erkennung von "at-risk" Studierenden, wurden zwei prinzipielle Charakteristiken eines Lernenden als Indikatoren verwendet: Interaktion und Ausdauer. Diese Indikatoren werden, wie bei dem in dieser Arbeit vorgestellten System, aus dem Verhalten und den Aktivitäten eines Studierenden abgeleitet. Diese Aktivitäten umschließen Forum-Aktivität, Anzahl der angesehenen Videos und Anzahl der abgeschlossenen Quizzes.

LASyM führt Learning Analytics an großen typischen MOOC-Datensätzen durch.

Abbildung 2.10 zeigt den Zusammenhang zwischen dem Verhalten von Studierenden und dessen Einfluss auf die Erkennung von "at-risk" Studierenden in einer schematische Darstellung in LASyM.

¹⁰<http://hadoop.apache.org> (letzter Aufruf: 11. 03. 2015)

2.3. Analyse bestehender Learning-Analytics-Systeme

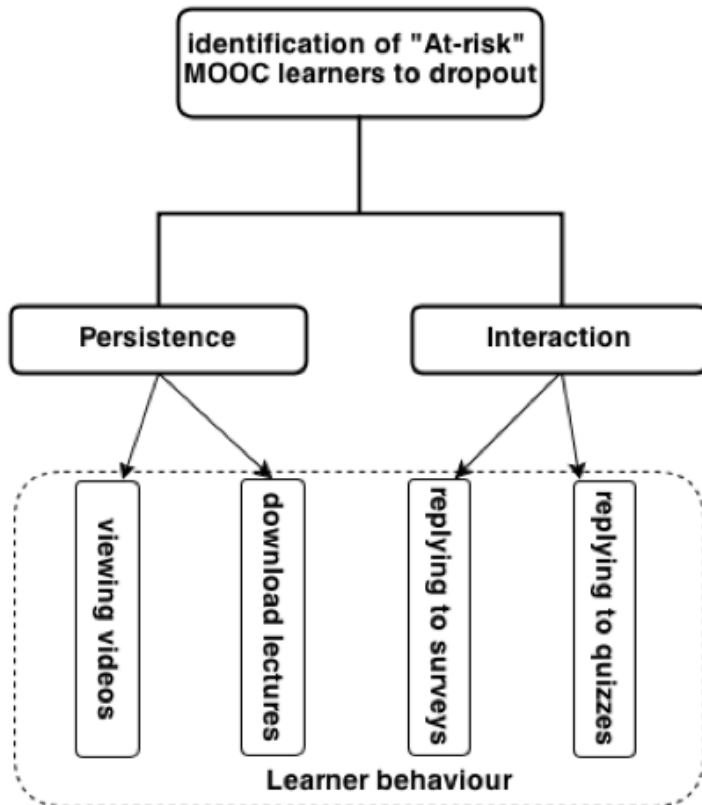


Abbildung 2.10.: Identifikation von "at-risk" Studierenden in LASyM (Tabaa and Medouri, 2013)

3. Umsetzung

In diesem Kapitel wird beschrieben, wie die Learning-Analytics-Plattform in technischer und technologischer Hinsicht umgesetzt wurde, wie die Methodik zur Generierung von wertvollen Informationen aus Rohdaten aussieht, welcher Funktionalitätsumfang implementiert wurde und wie die Architektur der Software aussieht.

3.1. Funktionalität

Abbildung 3.1 zeigt die Landing Page der Learning-Analytics-Plattform. Diese gibt auch zugleich einen ersten Überblick der implementierten Funktionalitäten. Die Plattform bietet die Möglichkeit, folgende Aktivitäten zu beobachten:

- **Logins:** Abbildung 3.2 zeigt einen Überblick wer sich am öftesten auf der Plattform imoox.at angemeldet hat. Es ist möglich, den zu untersuchenden Zeitraum einzuschränken, die Anzahl der angezeigten Logins zu limitieren und zwischen einer grafischen und einer textuellen Darstellungsform zu wählen.

3. Umsetzung



Abbildung 3.1.: Landing Page der Learning Analytics Web Application

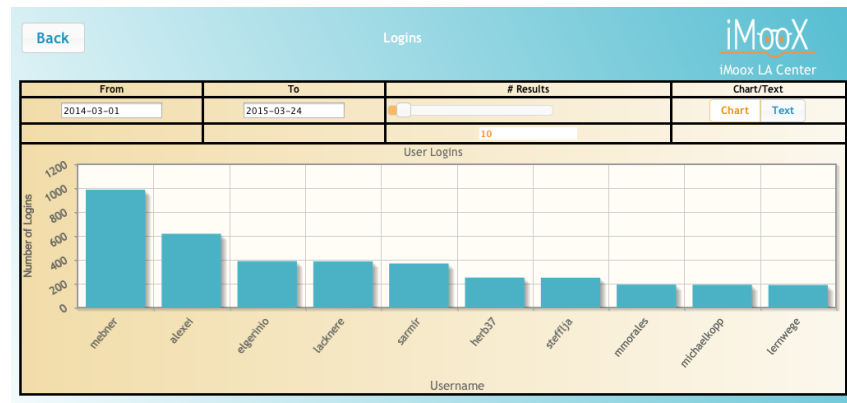


Abbildung 3.2.: Logins Page der Learning Analytics Web Application

3.1. Funktionalität

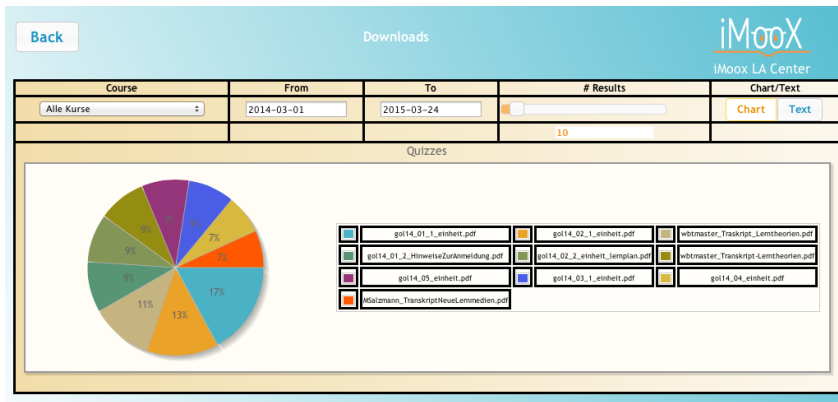


Abbildung 3.3.: Beliebteste Downloads Page Learning Analytics Web Application

- **Downloads:** 3.3 und 3.4 zeigen den Bereich Downloads. Es ist möglich die beliebtesten Downloads(Files) zu erkennen, sowie die aktivsten "Downloader" herauszufinden. Es besteht natürlich die Möglichkeit Downloads nur für einen bestimmten Kurs und/oder für einen bestimmten Zeitraum anzeigen zu lassen.
- **Forum Aktivität:** Aktivitäten in den Foren der jeweiligen Kurse zu untersuchen ist beispielhaft in den Abbildungen 3.5 und 3.6 zu sehen. Dabei wird die Anzahl der geschriebenen bzw. gelesenen Postings je BenutzerIn in den jeweiligen Kursforen untersucht.
- **Quizzes:** Ähnlich wie bei File Downloads ist es hier möglich, die beliebtesten Quizzes (bezüglich abgeschlossener Quizversuche) und die aktivsten QuizteilnehmerInnen zu filtern. Abbildungen 3.7 und 3.8 zeigen dies. Weiters ist es, wie in Abbildung 3.9 zu sehen, möglich, jedes einzelne Quiz genauer zu betrachten, wobei der Durchschnittscore aller QuizteilnehmerInnen für jeden einzelnen Quizversuch (jede/r BenutzerIn hat maximal 5 Versuche ein Quiz zu absolvieren) in Vergleich zu anderen Quizzes gestellt werden kann. Eine Datenreihe stellt jeweils ein Quiz dar und dazu wird auf der X-Achse der jeweilige Ver-

3. Umsetzung

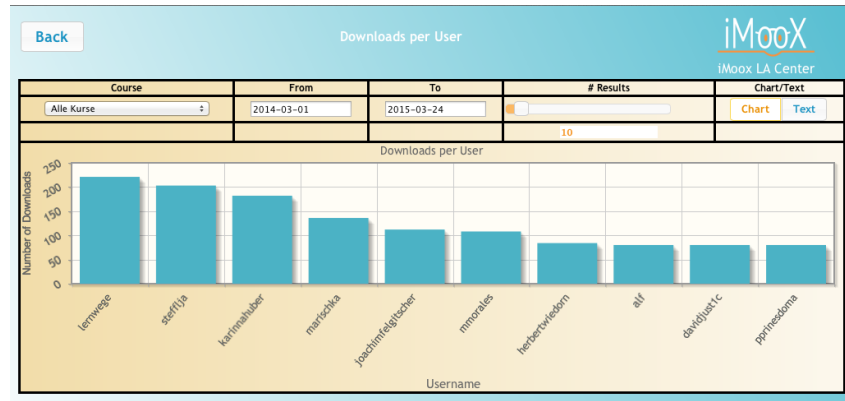


Abbildung 3.4.: "Downloader" Page der Learning Analytics Web Application

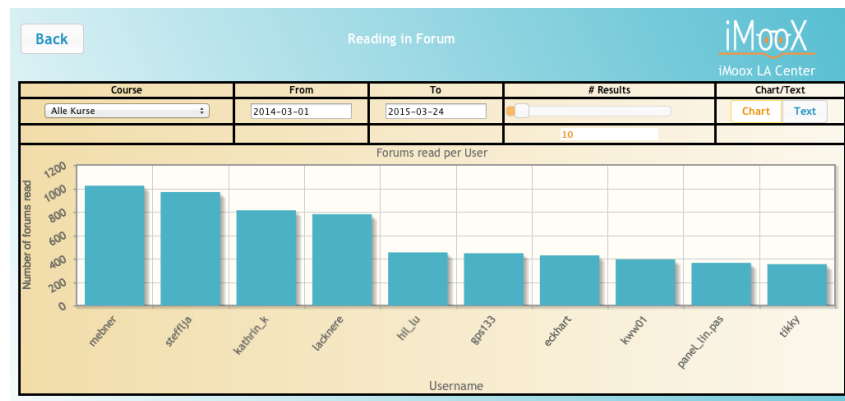


Abbildung 3.5.: Forum Reading Page Learning Analytics Web Application

3.1. Funktionalität

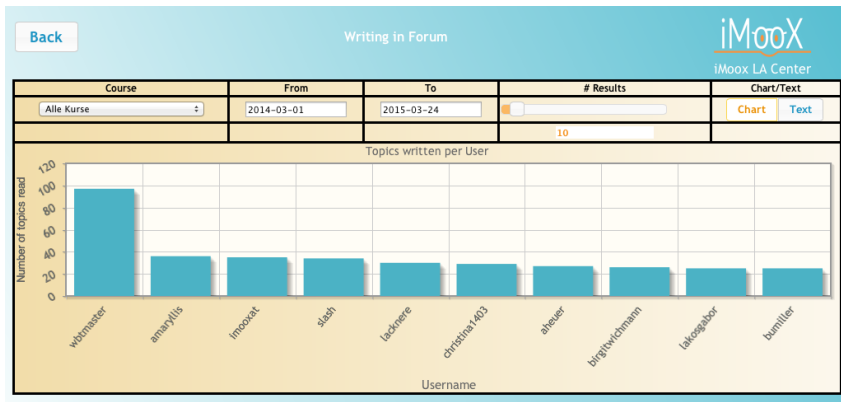


Abbildung 3.6.: Forum Writing Page der Learning Analytics Web Application

sich dem auf der Y-Achse aufgetragenen Quizerfolg gegenübergestellt.

- **Videos:** Es ist möglich, für jedes auf der Plattform imoox.at angebotene Video chronologisch darzustellen, wann und von wem das jeweilige Video gesehen wurde. Abbildung 3.10 zeigt einen Screenshot davon. Diese Darstellung ist scroll -und zoombar.
- **Dashboards:** Dashboards bieten die Möglichkeit, eine große Menge an Informationen schnell und auf einen Blick darzustellen. Auch hier wurden solche Dashboards eingesetzt um Informationen über Studierende, Kurse und weitere Kombinationen daraus darzustellen. Abbildung 3.11 zeigt einen Schnellüberblick mit Informationen über alle bisher genannten Funktionalitäten der Plattform für eine(n) gewählte(n) Studierende(n) an. Weiters ist es hier möglich zu sehen, welche Sprünge in der Zeit in einem von dem/der StudentIn angesehenen Video gemacht wurden. Das Parameter Dashboard (Abbildung 3.12) bietet die Möglichkeit, zwei unterschiedliche Parameter (Aktivitäten) eines/r Benutzers/in einander gegenüberzustellen, um Einflüsse von gewissen Aktivitäten auf der Plattform auf andere Aktivitäten zu erkennen.

3. Umsetzung

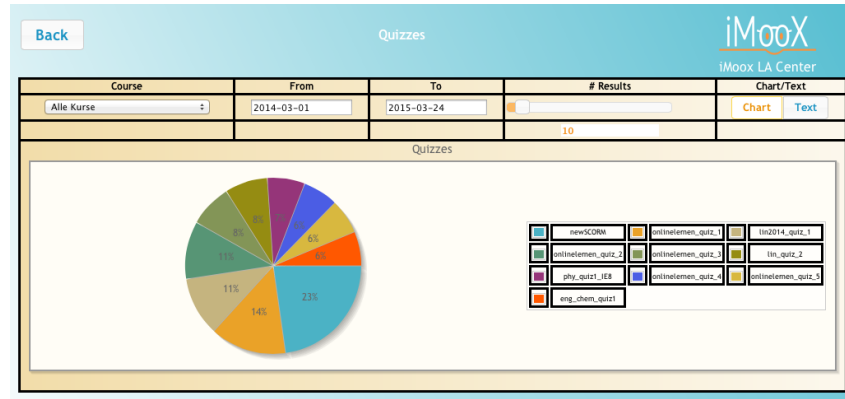


Abbildung 3.7.: Quizzes Page der Learning Analytics Web Application

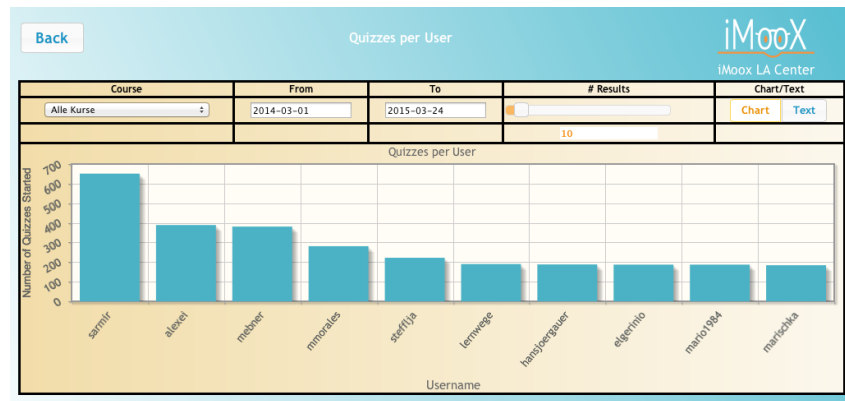


Abbildung 3.8.: Quizzes Page der Learning Analytics Web Application

3.1. Funktionalität

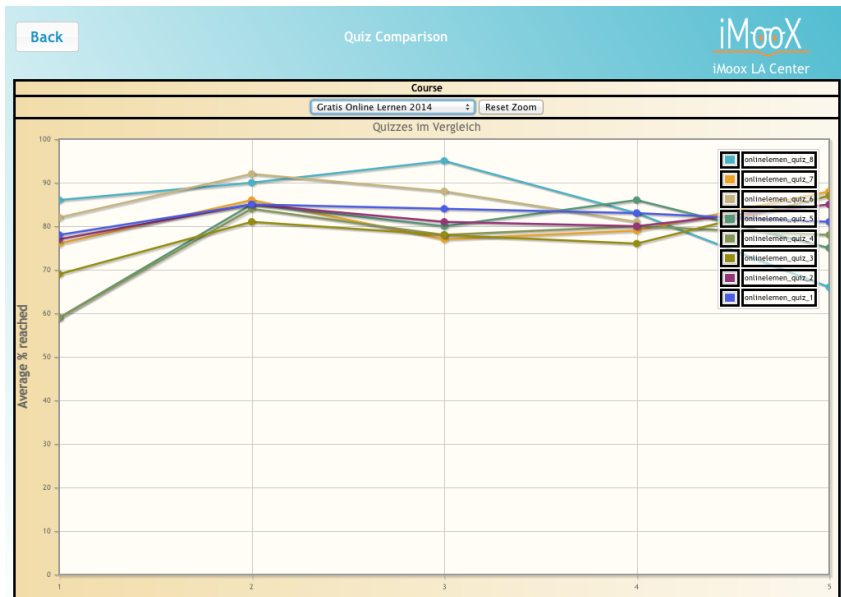


Abbildung 3.9.: Quizzes Vergleich Page der Learning Analytics Web Application

Hier stellt ein Datenpunkt im Diagramm eine(n) Studierende(n) dar. Im Kurs Dashboard (Abbildung 3.12) erhält man einen Überblick über Aktivitäten in einem jeweiligen Kurs. Es ist auch möglich einen Überblick über alle Kurse gesammelt zu bekommen.

Zur Veranschaulichung eines Applikationsablaufes zeigt Abbildung 3.14 einen Teil (hier: Dashboards) des gesamten Applikationsumfangs als Flussdiagramm. Die Trapeze im Diagramm stellen den Input und die Pfeile die Navigation des/der Benutzers/in dar.

3. Umsetzung

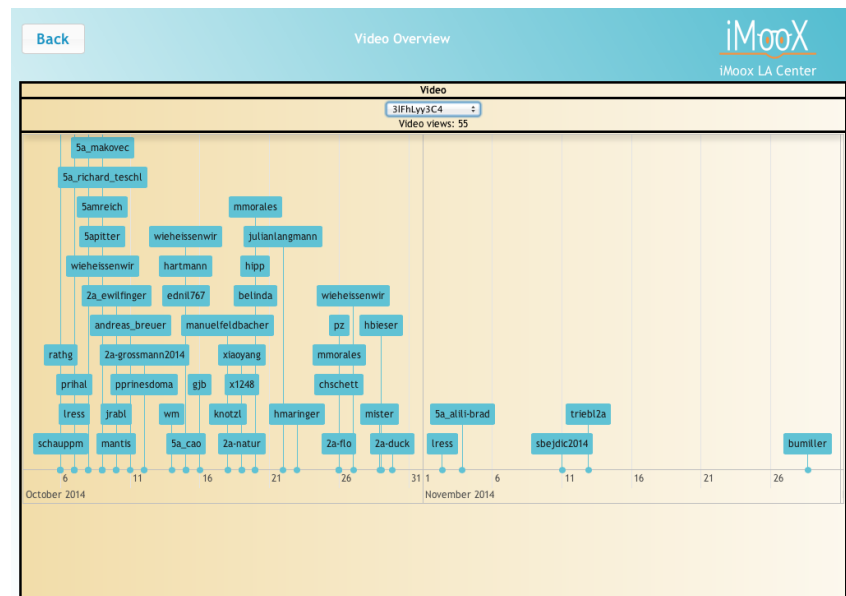


Abbildung 3.10.: Videos Page der Learning Analytics Web Application

3.2. Methodik

In diesem Abschnitt erläutern wir, woher die Rohdaten für die Analyse kommen, wie sie verarbeitet werden und wie sie zur Präsentation gelangen.

Wie bereits erwähnt, basiert die hier vorgestellte Learning-Analytics-Plattform auf Rohdaten der Plattform iMoox.at.¹ Die von dieser Plattform generierten Apache-Webserver²-Log-Daten werden dazu genutzt, Aktivitäten von Benutzern/innen der Plattform zu analysieren. Weiters kommen textbasierte, programmatisch erstellte Log Files zum Einsatz. Für das Handling der

¹<http://www.imoox.at/> letzter Aufruf: 16.05.2015

²<http://httpd.apache.org> letzter Aufruf: 16.05.2015

3.2. Methodik



Abbildung 3.11.: Landing Page der Learning Analytics Web Application

3. Umsetzung

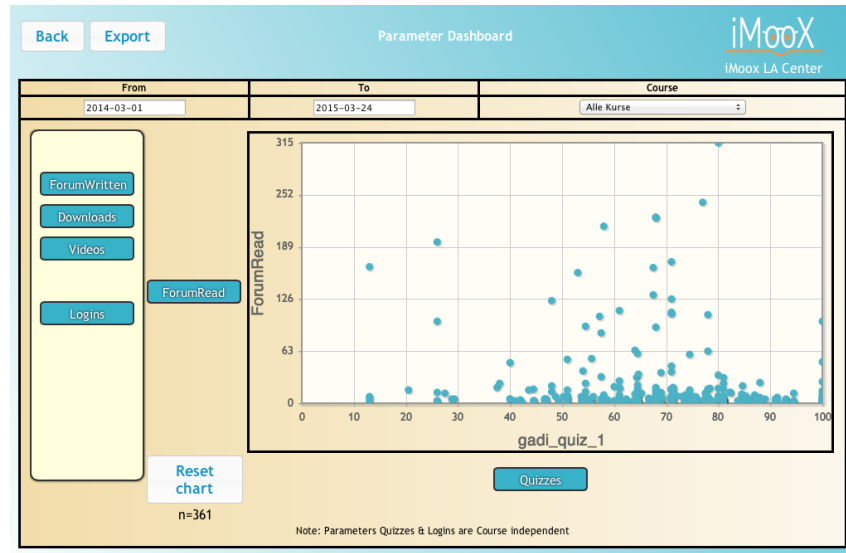


Abbildung 3.12.: Parameter Dashboard der Learning Analytics Web Application

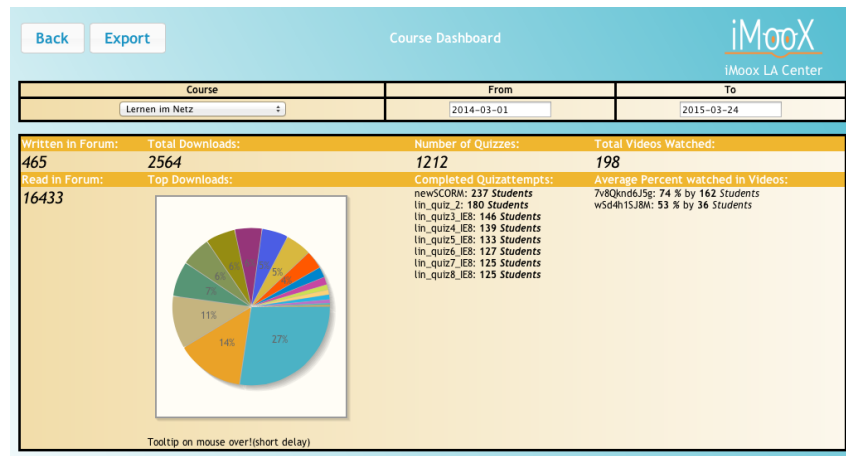


Abbildung 3.13.: Kurs Dashboard der Learning Analytics Web Application

3.2. Methodik

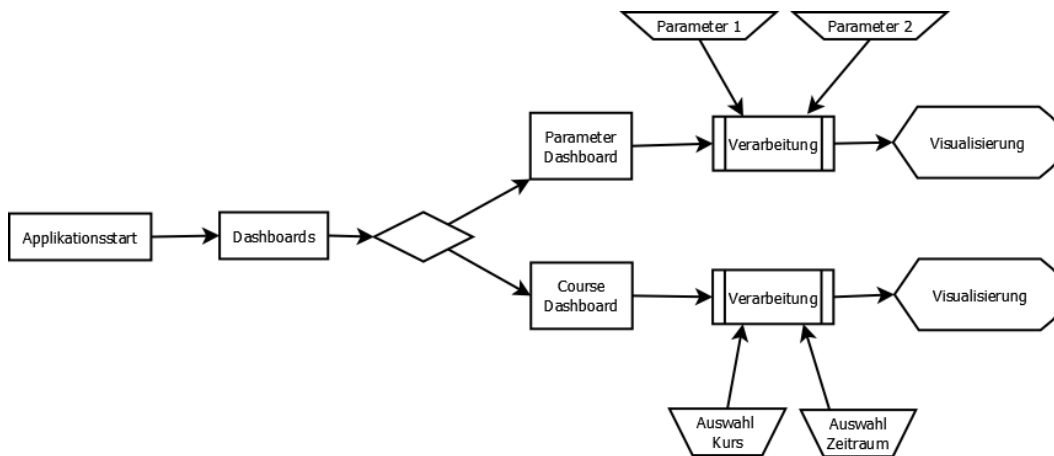


Abbildung 3.14.: Applikationsablauf

Log Files des Apache Web Servers wurde Splunk³ verwendet, ein Log- und Monitoring-Tool zur vereinfachten Aggregation und Filterung der Log Files.

Die Verarbeitung dieser Log Files passiert immer tagesaktuell, was gleichzeitig bedeutet, dass es nicht möglich ist, Ereignisse auf der MOOC-Plattform live zu analysieren. Weiters werden die Log Files chronologisch abgearbeitet und daraufhin in die Learning Analytics Datenbank gespeichert.

Basierend auf den nun in der Datenbank abgelegten Daten wird die Analyse und Präsentation durchgeführt. In Abschnitt 3.2.1 wird gezeigt, in welcher Form die Log Daten zur weiteren Verarbeitung vorliegen.

³<http://de.splunk.com/> letzter Aufruf: 16.05.2015

3. Umsetzung

3.2.1. Log Daten

Wie die hier verarbeiteten Logfiles aussehen wird in diesem Abschnitt erläutert. Das Tool Splunk bietet die Möglichkeit programmatisch über eine API (Application Programming Interface) auf die einzelnen Felder dieser Logfiles zuzugreifen. Das vereinfacht die Verarbeitung der Logfiles etwas, um sie dann in aufbereiteter Form in der Learning-Analytics-Datenbank abzuspeichern.

Login

Auflistung 3.1 zeigt exemplarisch die Log Einträge zur Erfassung einer Login Aktivität auf der MOOC-Plattform. Wichtig hierbei ist der Benutzername (hier: "wbtmaster"), und der zugehörige Token (hier: "3532D255A313C61E9DA272EB5710BFCB").

Die Kombination aus Benutzername und aktuellem Token wird in der Datenbank gespeichert, um zukünftige Aktivitäten nachverfolgen zu können. Hierbei sei erwähnt, dass ein(e) Benutzer(in) nach jedem erfolgreichen Login einen neuen Token zugeordnet bekommt.

```
1 129.27.116.22 -- [25/Mar/2014:10:57:20 +0100] "GET /wbtmaster/  
forum.Forum?action=get_user_card&login=wbtmaster&token=3532  
D255A313C61E9DA272EB5710BFCB&timestamp=25105723103 HTTP/1.1"  
200 209
```

Quelltext 3.1: Log File Login Aktivität

Dateidownload

Ein beispielhafter Log Eintrag für einen Dateidownload ist in Auflistung 3.2 zu sehen. Aufgrund des angegebenen Tokens im Log ist es möglich eine Zuordnung des Downloads zum Downloadenden zu treffen. (Hier: BenutzerIn mit Token "29F7A632C1C57C4056832538A3329433" hat am "14/Apr/2014:11:01:49" die Datei "Transkript-Lerntheorien.pdf" heruntergeladen).

```
1 129.27.116.22 -- [14/Apr/2014:11:01:49 +0200] "GET /wbtmaster/
  threads/lin/wbtmaster_Transkript-Lerntheorien.pdf?token=29
  F7A632C1C57C4056832538A3329433 HTTP/1.1" 200 230489
```

Quelltext 3.2: Log File Dateidownload Aktivität

Forumsbeitrag lesen

In Auflistung 3.3 ist ein Log zu sehen, der das Lesen eines Forumsbeitrages darstellt. Wichtig hierbei, wiederum ist der Token und in diesem Fall das Feld "file", aus welchem das konkrete Forum extrahiert werden muss.

```
1 129.27.116.22 -- [25/Mar/2014:10:23:46 +0100] "GET /wbtmaster/
  forum.Forum?action=get&file=lin/gateway/imooxat0903142150.11.
  htm&token=3532D255A313C61E9DA272EB5710BFCB&timestamp=2510234914
  HTTP/1.1" 200 450
```

Quelltext 3.3: Log File Forum Lesen Aktivität

3. Umsetzung

Forumsbeitrag schreiben

Auflistung 3.4 zeigt das Lesen eines Forumsbeitrages. Wiederum lässt sich über den Token auf den/die BenutzerIn schließen, und über das Feld "room" auf das jeweilige Forum.

```
1 129.27.116.22 -- [25/Mar/2014:10:33:41 +0100] "GET /wbtmaster/
groovy/addForum.groovy?room=lin&file=forumstring.lib&addon=
testuser12503141033_44%25%3A%25%25%3A%25Wurstsemmeln%25%3A%25
testuser12503141033_44.htm%25%3A%25%25%3A%25*2503141033%25%3A
%25testuser1%25%2C%25testuser1%25%3A%25forumstring.lib%25%3B
%25&token=3532D255A313C61E9DA272EB5710BFCB&timestamp=2510334440
HTTP/1.1" 200 1610
```

Quelltext 3.4: Log File Forum Schreiben Aktivität

Quiz

Bei Quizversuchen dienen nicht Webserver Logs als Grundlage zur Analyse, sondern programmatisch erstellte Logfiles (hier: "wbtmaster-linquiz2.scorm"), mit dem Usernamen und dem zugehörigen Quiznamen im Dateinamen. In diesem Fall hätte User "wbtmaster" im Quiz "linquiz2" die in Auflistung 3.5 gezeigte Leistung erbracht. Das Logfile, zeigt welche Frage richtig bzw. falsch beantwortet wurde und wie sich der/die StudentIn im Laufe der fünf Quizversuche verbessert (verschlechtert) hat. Vor allem die Verbesserung bzw. Verschlechterung im Laufe der fünf Quizversuche ist für die Analyse interessant.

```
1 %,%Question 0%:%4%:%not_checked%:%right%:%3%:%not_checked%:%right%
:%1%:%checked%:%right%:%2%:%not_checked%:%right%:%
2 %,%Question 1%:%4%:%not_checked%:%right%:%2%:%checked%:%right%:%1%
:%not_checked%:%right%:%3%:%not_checked%:%right%:%
```

3.2. Methodik

```
3 %;%Question 2%:%3%:%checked%:%right%:%2%:%not_checked%:%right%:%1%
   :%not_checked%:%right%:%4%:%not_checked%:%right%:%
4 %;%Question 3%:%1%:%checked%:%right%:%3%:%checked%:%right%:%2%:%
   checked%:%right%:%4%:%not_checked%:%right%:%
5 %;%Question 4%:%1%:%checked%:%right%:%2%:%checked%:%right%:%3%:%
   not_checked%:%right%:%4%:%not_checked%:%right%:%
6 %;%Question 5%:%1%:%checked%:%right%:%4%:%not_checked%:%right%:%2%
   :%checked%:%right%:%3%:%checked%:%right%:%
7 %;%Question 6%:%2%:%not_checked%:%right%:%1%:%checked%:%right%:%4%
   :%not_checked%:%right%:%3%:%not_checked%:%right%:%
8 %;%last_try%:%100.0%;%best_try%:%100.0%;%attempts_left%:%1%;%cmi.
   core.session_time%:%00:00:00.0%;%progress%:
   %42.9;57.1;71.4;100.0%;%;
```

Quelltext 3.5: Log File Quiz

Videos

Auch die Video Logs werden programmatisch erstellt und nicht anhand von Webserver Logs gelesen. Beispielhaft zeigt Auflistung 3.6 dass der/die BenutzerIn "wbtmaster" das Video mit der ID "j3b7oggyWjA" angesehen hat und welche Zeitsprünge im Video gemacht wurden (hier: Sprünge von 0.00s auf 146.50s auf 454.09s usw.). All diese Informationen werden in der Learning Analytics Datenbank zur späteren Analyse und Präsentation gespeichert.

```
1 Mon Mar 16 2015 20:30:19 GMT+0100%:%wbtmaster%:%2%:%j3b7oggyWjA%:
   %0.00%:%859.49
2 [|||]
3 Mon Mar 16 2015 20:30:19 GMT+0100%:%wbtmaster%:%2%:%j3b7oggyWjA%:
   %146.50%:%859.49
4 [|||]
5 Mon Mar 16 2015 20:30:22 GMT+0100%:%wbtmaster%:%1%:%j3b7oggyWjA%:
   %454.09%:%859.49
```

3. Umsetzung

```
6 [ ]
7 Mon Mar 16 2015 20:30:28 GMT+0100%:%wbtmaster%:%2%:%j3b7oggyWjA%:
  %801.51%:%859.49
8 [ ]
9 Mon Mar 16 2015 20:30:28 GMT+0100%:%wbtmaster%:%2%:%j3b7oggyWjA%:
  %851.51%:%859.49
```

Quelltext 3.6: Log File Video

3.3. Architektur

Das hier beschriebene Learning-Analytics-System beruht auf einer klassischen Server Infrastruktur. Abbildung 3.15 zeigt die Systemarchitektur. Der Learning Analytics Server (Application Server) verarbeitet die Log Files mit der Serverprogrammiersprache Python ⁴ und speichert die Ergebnisse in eine MySQL Datenbank. ⁵ Ein Apache Webserver mit PHP ⁶ bedient die BenutzerInnen der Learning-Analytics-Plattform mit den Daten vom Datenbankserver. Clientseitig werden diese Daten meist in Form von Diagrammen präsentiert.

⁴<https://www.python.org>

⁵<https://www.mysql.de>

⁶<http://php.net>

3.3. Architektur

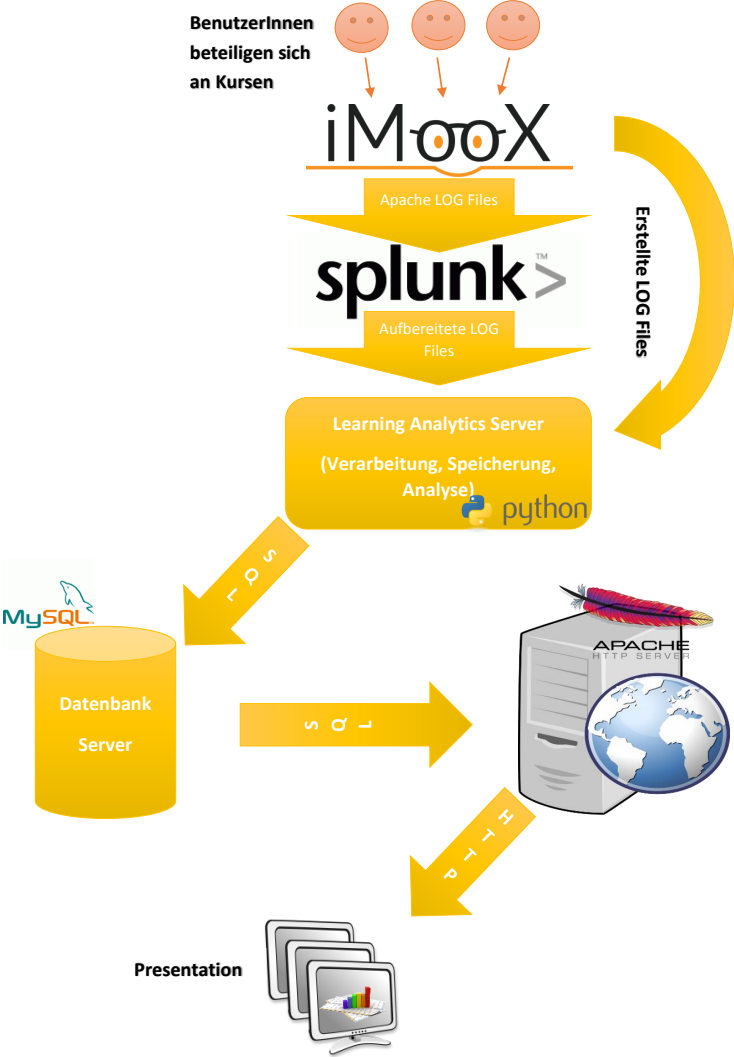


Abbildung 3.15.: Learning Analytics Systemarchitektur

3. Umsetzung

3.4. Server

Der Server auf dem die Daten verarbeitet werden nutzt derzeit die VMware-Infrastruktur⁷ des ZIDs⁸ der Technischen Universität Graz. Zur Verarbeitung stehen 1 vCPU, 1 GB RAM und 50 GB HDD zur Verfügung.

Die Datenverarbeitung am Server wurde in zwei Phasen getrennt und passiert jeweils einmal am Tag.

- **Serialization:** In der Serialization-Phase werden die durch Splunk bereitgestellten iMoox.at Log Files zuerst als Python Objekte serialisiert (gespeichert). Diese Zwischenspeicherung wurde eingeführt, um nicht unnötigerweise öfter auf Splunk zugreifen zu müssen, da dies ressourcenintensiv ist. Ressourcenintensivität bei einem Einsatz von Splunk hat den Grund, dass Splunk jedes mal Unmengen an Log-Daten voranalysieren muss, um relevante Informationen liefern zu können. Die Splunk-Daten werden somit als einzelne Dateien, bestehend aus Python-Objekten für jeden einzelnen Tag am Server gespeichert. Dadurch ist es möglich, auch ohne Splunk, auf ältere Daten zurückgreifen zu können. Auflistung A.1 im Appendix zeigt den zugehörigen Quelltext.
- **Deserialization:** In der Deserialization-Phase geschieht die eigentliche Verarbeitung der Python-Objekte aus der Serialization-Phase. Hier werden meist, aufgrund von Schlüsselwörtern in den Logs, Aktivitäten erkannt, bereinigt und gespeichert. Auflistung 3.7 zeigt die Erkennung von Aktivitäten.

```
1 if serializeData:  
2     for result in splunk_data:
```

⁷<http://www.vmware.com/at> letzter Aufruf: 16.05.2015

⁸<http://portal.tugraz.at/portal/page/portal/zid/> letzter Aufruf: 16.05.2015

3.4. Server

```
3  if "_time" in result.keys():
4      time = result['_time'].split("+")
5      time = time[0]
6      #cut the timezone correction, mysql datetime format
       doesn't like it
7  else:
8      time = 'not defined'
9
10
11  if 'uri' in result.keys() and result['uri'] !=
       previous_uri: #kick simple redundant logs
12      #LOGIN
13      if "get_user_card" in result['uri'] and 'token' in
           result.keys() and 'login' in result.keys():
14          parseLogin(result, time, previous_time_login,
                       previous_login_username)
15          previous_time_login = time
16          previous_login_username = result['login']
17      #LOGOUT
18      if "logout" in result['uri']:
19          parseLogout(result, time)
20      #QUIZ
21      if "method" in result.keys():
22          if result['method']=="GET" and ".scorm" in result['
           uri']:
23              parseQuizStarted(result, time,
                               previous_time_quiz_started,
                               previous_time_quiz_started_username)
24              previous_time_quiz_started = time
25              previous_time_quiz_started_username=filterString(
                   result['uri'], 'mainOwp/', '- ')
26          if result['method']=="POST" and ".scorm" in result['
           uri']:
27              parseQuizEnded(result, time, previous_time_quiz_ended
                               )
28              previous_time_quiz_ended = time
29      #DOWNLOAD
```

3. Umsetzung

```
30     if "/wbtmaster/threads" in result['uri'] and 'token' in
        result.keys():
31         parseDownload(result, time, previous_time_download,
            previous_time_download_username)
32         previous_time_download = time
33         if result['token'] in logins_tokens.keys():
34             previous_time_download_username = logins_tokens[
                result['token']]
35     #FORUM
36     if "/wbtmaster/groovy/addForum.groovy?room" in result['
        uri'] and 'token' in result.keys():
37         parseForumWritten(result, time,
            previous_time_forum_written,
            previous_time_forum_written_username)
38         previous_time_forum_written = time
39         if result['token'] in logins_tokens.keys():
40             previous_time_forum_written_username =
                logins_tokens[result['token']]
41     if "/wbtmaster/forum.Forum?action=get&file=" in result[
        'uri'] and 'token' in result.keys():
42         parseForumRead(result, time, previous_time_forum_read,
            previous_time_forum_read_username)
43         previous_time_forum_read = time
44         if result['token'] in logins_tokens.keys():
45             previous_time_forum_read_username = logins_tokens[
                result['token']]
46     previous_uri = result['uri']
```

Quelltext 3.7: Aktivitätenerkennung

Wurde eine konkrete Aktivität eines/r Benutzers/in erkannt, wird die jeweilige Funktion zur Verarbeitung aufgerufen. Beispielhaft wird in Auflistung 3.8 die Verarbeitung und Speicherung in der Datenbank von "Lesen eines Forumbeitrages" gezeigt. Der gesamte Quellcode wird in Auflistung A.2 gezeigt.

```
1 def parseForumRead(result, time, previous_time,
    previous_time_forum_read_username):
```

3.4. Server

```
2  if previous_time != None:
3      last_minutes = previous_time.split("T")[1].split(":")[1]
4  else:
5      last_minutes = "None"
6
7  recent_minutes = time.split("T")[1].split(":")[1]
8
9
10
11  if result['token'] in logins_tokens.keys():
12      if last_minutes != recent_minutes or (last_minutes ==
13         recent_minutes and (logins_tokens[result['token']] !=
14         previous_time_forum_read_username)): # if not in the
15         same minute it's OK, if in the same minute, usernames
16         must be different
17
18         room=filterString(result['uri'], "file=", "/")
19
20         if result['token'] in logins_tokens.keys() and room not
21         in ignored_rooms:
22             #saveCourseName(room)
23             if room in forum_read:
24                 forum_read[room] +=1
25             else:
26                 forum_read[room] = 1
27
28             if logins_tokens[result['token']] in
29                 forum_read_per_user:
30                 forum_read_per_user[logins_tokens[result['token']]]
31                 += 1
32             else:
33                 forum_read_per_user[logins_tokens[result['token']]]
34                 = 1
35         if screenOutput and printForumRead:
36             print "FORUM entry in course ", room, " read by ",
37                 logins_tokens[result['token']], " at ", time
38
39
```

3. Umsetzung

```
30         if room in courses_array:
31             if len(room) < 80:
32                 curA.execute("INSERT IGNORE INTO forums (fname,
33                             cname) VALUES (%s,%s)" ,(room,room))
34                 cnx.commit()
35             if len(room) < 80 and len(logins_tokens[result[ '
36                 token ' ]]) < 80:
37                 curA.execute("INSERT INTO forums_read (fname,
38                             uname,time) VALUES (%s,%s,%s)" ,(room,
39                             logins_tokens[ result[ 'token ' ] ],time))
40                 cnx.commit()
```

Quelltext 3.8: Verarbeitung von "Lesen eines Forumbeitrages"

3.5. Datenbank

Zur persistenten Speicherung der Learning Analytics Daten wurde eine MySQL-Datenbank verwendet. Das Datenbankmodell ist in 3.16 zu sehen. In detaillierter Form zeigt A.3 im Appendix ein Skript zur Initialisierung der Datenbank. Als Infrastruktur werden zentrale Datenbankserver des ZID⁹ genutzt.

3.6. Web Application

Die Web Plattform zur Präsentation der Learning-Analytics-Daten nutzt jQuery zur Darstellung. Als Bibliothek für Diagramme wurde jqplot¹⁰ ver-

⁹<http://portal.tugraz.at/portal/page/portal/zid/> letzter Aufruf: 16.05.2015

¹⁰<http://www.jqplot.com> letzter Aufruf: 16.05.2015

3.6. Web Application

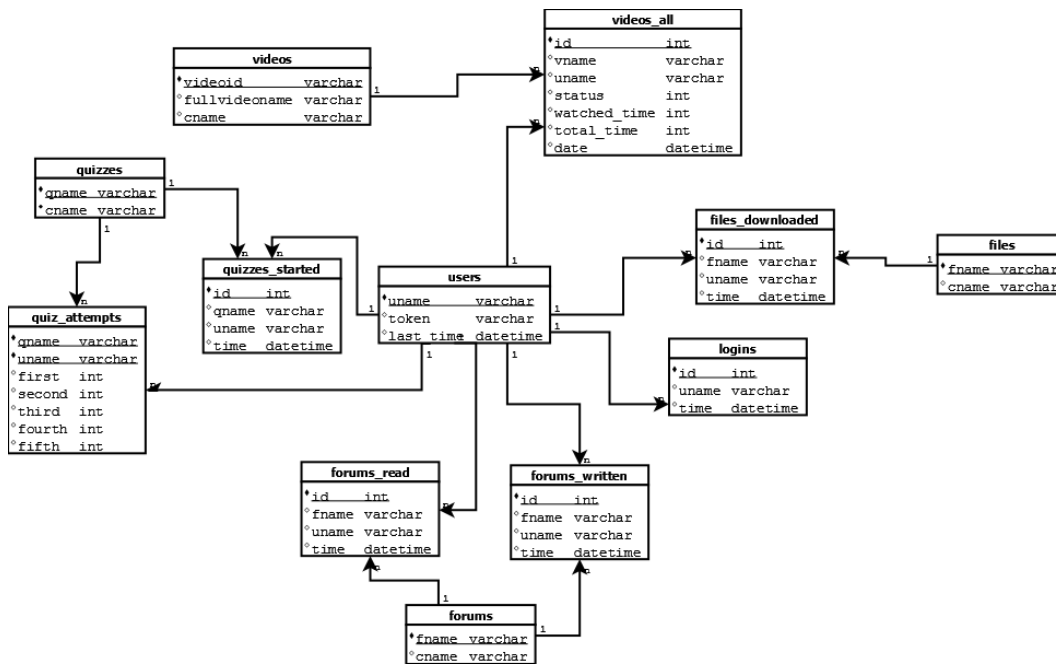


Abbildung 3.16.: Learning-Analytics-Database-Design

3. Umsetzung

wendet. Für den PDF Export wird jsPDF ¹¹ verwendet. Diese Web-Plattform ist zur Zeit als Prototyp innerhalb der Organisation in Verwendung und nicht öffentlich zugänglich.

¹¹<http://code.google.com/p/jspdf/> letzter Aufruf: 16.05.2015

4. Evaluation und Diskussion

In diesem Kapitel wird die vorgestellte Learning-Analytics-Plattform evaluiert und diskutiert. Es werden Beobachtungen in den einzelnen Kursen gezeigt, die für weitere Kurse auf der iMoox Plattform von Bedeutung sind und es wird diskutiert welche Verbesserungen an der Learning-Analytics-Plattform sowie an der MOOC-Plattform zukünftig durchgeführt werden könnten, bzw. welche Funktionalitäten noch nicht integriert sind. Einige der in diesem Kapitel gezeigten Grafiken sind freundlicherweise von Mohammad Khalil bereitgestellt worden, der bereits weitere Evaluationen zu der hier vorgestellten Arbeit durchgeführt hat.

4.1. Kollaboration

Die Vorhersage über die Entwicklung der Leistung und der Mitarbeit einer/s StudentIn ist ein wichtiger Teil von Learning Analytics. Dafür ist es notwendig, genau über die Aktivitäten eines Studierenden auf der MOOC-Plattform Bescheid zu wissen. Gleichzeitig ist es auch wichtig frühzeitig zu erkennen, ob der Kurs selbst stark an Kollaboration verliert. Learning Analytics bietet die Möglichkeit einen Aktivitätsabfall einerseits bei Studierenden und andererseits beim Kurs selbst zu erkennen.

4. Evaluation und Diskussion

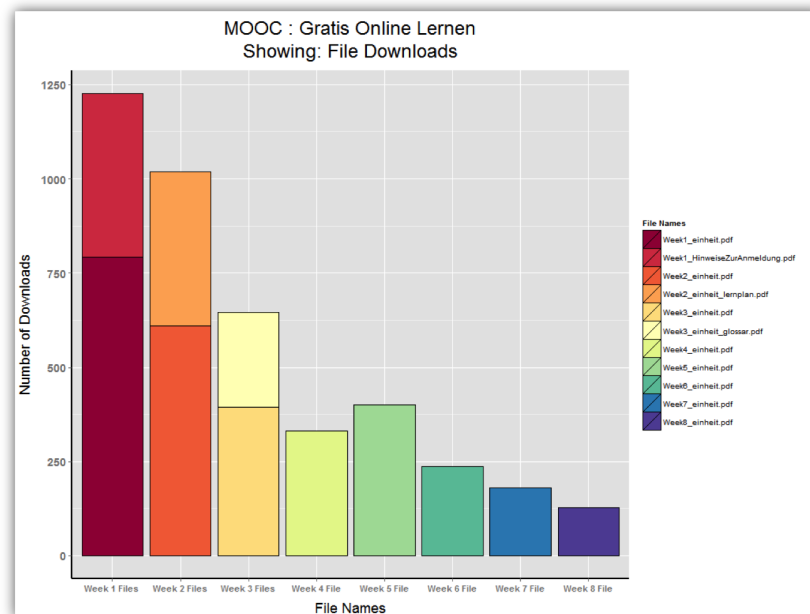


Abbildung 4.1.: Entwicklung der Datei Downloads im Kurs "Gratis Online Lernen"

4.1. Kollaboration

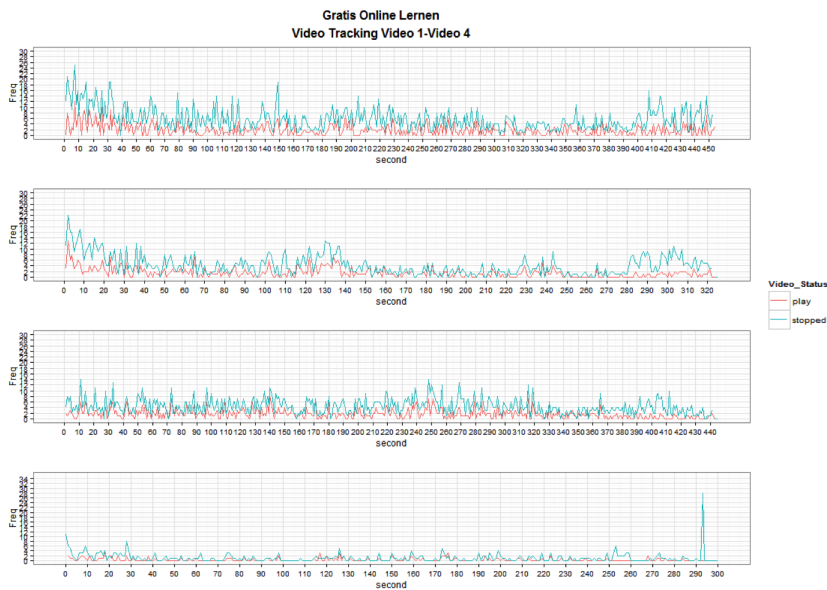


Abbildung 4.2.: Entwicklung der Videokonsumtion im Kurs "Gratis Online Lernen"

Abbildung 4.1 zeigt beispielhaft wie sich die Anzahl der Datei Downloads des Kurses "Gratis Online Lernen" über die Kursdauer verändert. Es ist hier eine deutliche Abnahme der Datei Downloads zu erkennen, was auch mit der Aktivität bei der Konsumation von Videos korreliert. Abbildung 4.2 zeigt hier die Konsumation der ersten vier Videos vom Kurs "Gratis Online Lernen". Dabei ist auf der y-Achse jeweils die Anzahl der Videoaufrufe und auf der x-Achse die Zeit aufgetragen. Je größer der Ausschlag der Kurve ist, desto mehr Studierende haben sich das Video angesehen. Es ist in Abbildung 4.2 außerdem ersichtlich, zu welchen Zeitpunkten die Videos von den Studierenden pausiert bzw. abgespielt wurden, was natürlich Rückschlüsse auf den Videoinhalt an den jeweiligen Zeitpunkten zulässt.

Die beobachtbare Abnahme in der Aktivität von Studierenden ist auch in Abbildung 4.3 ersichtlich. Dabei stellt jeder Punkt im Diagramm die

4. Evaluation und Diskussion

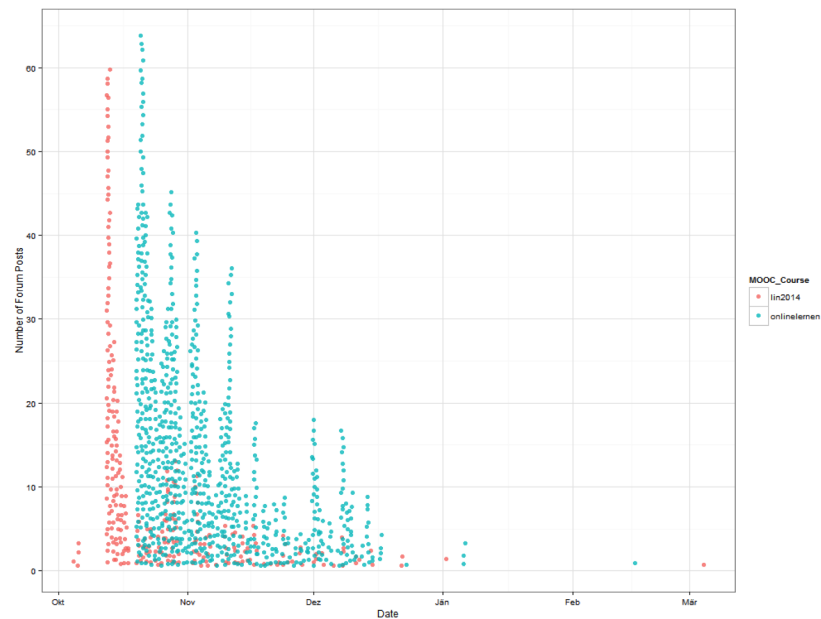


Abbildung 4.3.: Entwicklung der Forumsaktivität in den Kursen "Gratis Online Lernen" und "Lernen im Netz"

Gesamtanzahl der Forum-Postings pro Tag im Verlauf der Zeit dar. Es ist klar ersichtlich, dass die Kollaboration auch hier mit der Zeit abnimmt.

Einen weiteren Aspekt der Learning-Analytics-Plattform zeigt Abbildung 4.4. Dabei wird die Leistung der Studierenden in einem Quiz der Aktivität bei Dateidownloads gegenübergestellt. Ersichtlich ist hier, dass Studierende die sich eine der Kurswoche zugehörige Datei heruntergeladen haben, im Schnitt erkennbar besser beim zugehörigen Quiz abgeschnitten haben.

Abbildung 4.5 stellt Quizerfolg im ersten Quizversuch der Aktivität in den Foren für den Kurs "Gratis Online Lernen" gegenüber. Hierbei ist erkennbar, dass Studierende die im Forum aktiv waren, bei den Quizzes im

4.1. Kollaboration

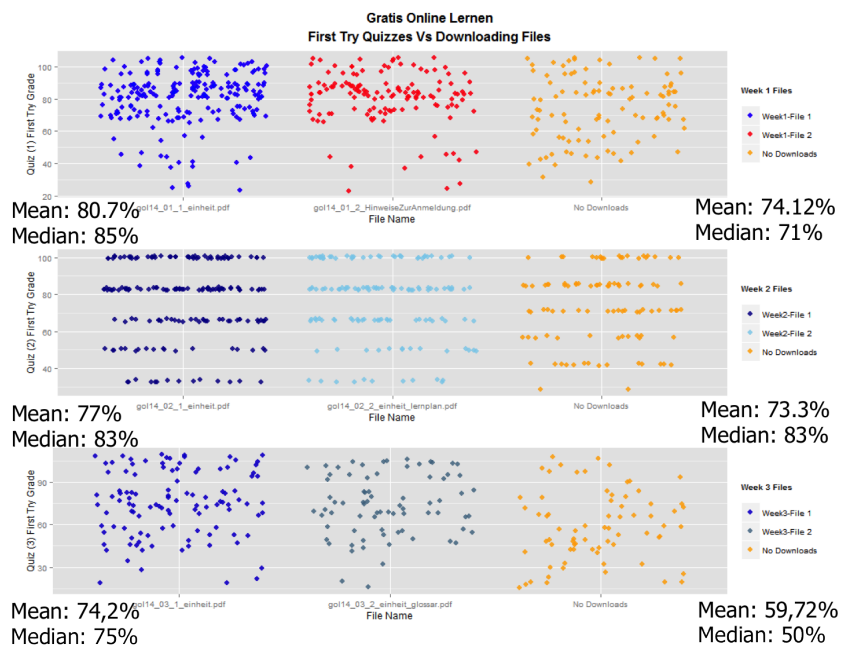


Abbildung 4.4.: Zusammenhang von Quizleistung und Datei Downloads im Kurs "Gratis Online Lernen"

4. Evaluation und Diskussion

Kurs seltener negativ abgeschnitten. Umgekehrt ist es jedoch nicht möglich die Aussage zu treffen, dass Studierende, die im Forum wenig aktiv waren deswegen erkennbar schlechter bei den Quizzes abgeschnitten haben. Abbildung 4.5 zeigt weiters, dass es nur eine relativ schwache Beziehung zwischen diesen zwei Faktoren gibt. Die x-Achse stellt das durchschnittliche Abschneiden in allen Quizzes durch den Studierenden dar. Die blaue Linie repräsentiert eine lineare Regression, während der graue Bereich um die Linie herum den Standardfehler darstellt. Studierende mit einem guten Abschneiden bei den Quizzes ($>90\%$), lasen im Schnitt 21 Forumsbeiträge. Andererseits gibt es auch Studierende, die mehr als 21 Forumsbeiträge gelesen haben, jedoch nicht alle Quizzes positiv abschließen konnten. Das bedeutet, der Standardfehler ist in diesem Fall größer, wenn das Abschneiden bei den Quizzes schlechter ist ($<60\%$). Nichtsdestotrotz gibt es aber, wie erwähnt, auch noch andere Faktoren, wie das Ansehen eines Videos und das Lesen der bereitgestellten Dokumente, die Einfluss auf das Abschneiden der Studierenden haben.

4.2. Intervention

Für zukünftige Arbeiten zum Thema Learning Analytics sollte es in Betracht gezogen werden, die LA-Plattform besser in die iMoox-Plattform zu integrieren. Ein wichtiger Punkt, die Intervention des Betreuenden/Lehrenden bei Risikostudierenden (Studierende die entweder nicht mehr mitarbeiten, oder deren Aussicht auf einen positiven Abschluss des Kurses nicht gegeben ist), ist in der jetzigen technischen Umsetzung einerseits der LA Plattform andererseits der iMoox Plattform nicht möglich. Hierzu müssten zukünftig Adaptionen vorgenommen werden, durch welche der Betreuende den Studierenden direkt ansprechen kann, wenn erkennbar ist, dass der

4.2. Intervention

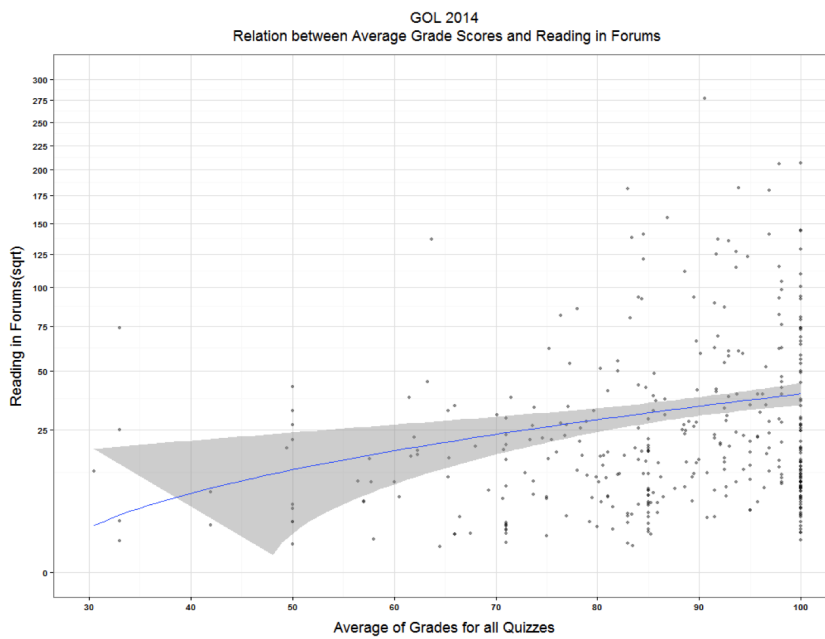


Abbildung 4.5.: Zusammenhang von Quizleistung und Forum Aktivität im Kurs "Gratis Online Lernen"

4. Evaluation und Diskussion

Studierende sich zu einem Risikostudierenden entwickelt. Wie Erkenntnisse in Abschnitt 4.1 zeigen, geht es in erster Linie darum, die Motivation von Studierenden hochzuhalten. Dies wäre durch einfache Hinweise, die personalisiert in der iMoox Plattform erscheinen, umsetzbar.

4.3. Personalisierung & Adaptierung

Ähnlich wie in Abschnitt 4.2 beschrieben, geht es hierbei um die individuelle Betreuung von Studierenden auf der MOOC-Plattform. Wird erkannt, dass der/die StudentIn Probleme mit den zur Verfügung gestellten Lehrmitteln hat, sollte es unter Anwendung der Learning Analytics Plattform möglich sein, den Studierenden gezielt zusätzliche individuelle Lehrmittel auf der MOOC-Plattform zur Verfügung zu stellen. Ein weiterer Aspekt für die zukünftige Entwicklung ist, dass Studierenden die Möglichkeit gegeben werden sollte, selbst ihr Feedback auf der MOOC-Plattform einzubringen, um zusätzliche Informationen über die bereitgestellten Inhalte zu erlangen. Vorstellbar wären auch Abstimmungen darüber, welche Art der Wissensvermittlung für die an einem Kurs teilnehmenden Studierenden am interessantesten ist.

4.4. Kritik

4.4.1. Datensicherheit

Gerade in einer Zeit, in der Datenschutzverletzungen beinahe an der Tagesordnung stehen (NSA PRISM¹), und dadurch das öffentliche Interesse am Schutz der persönlichen Informationen enorm gestiegen ist, müssen wir bei der Umsetzung der Learning-Analytics-Plattform mit diesen äußerst sensiblen Informationen über Studierende sehr gewissenhaft umgehen.

Einige zur Datensicherheit beitragende Maßnahmen auf der Learning-Analytics-Plattform sind:

- Speicherung der Learning-Analytics-Daten maximal 3 Monate
- Es werden keine IP-Adressen von Studierenden gespeichert
- Zugriff auf Learning-Analytics-Daten nur für BetreuerInnen
- Zugriff auf Learning-Analytics-Daten nur über eine gesicherte VPN Verbindung

Nichtsdestotrotz werden zur Zeit jedoch Benutzernamen der Studierenden in der Datenbank gespeichert. Da diese Benutzernamen häufig die realen Namen darstellen, muss in Zukunft über eine Überarbeitung dieser Speicherung nachgedacht werden. Vorstellbar wäre es, dass Benutzernamen bei Anmeldung am System eine zufällige Identifikationsnummer zugeordnet wird. Diese Identifikationsnummer sollte mit Hilfe einer Hash-Funktion ermittelt werden, um bei zukünftigen Anmeldungen und Aktionen eines/r

¹<http://www.theverge.com/2013/7/17/4517480/nsa-spying-prism-surveillance-cheat-sheet> letzter Aufruf: 13.04.2015

4. Evaluation und Diskussion

Benutzers/in wieder eine eindeutige Zuordnung zum/zur jeweiligen BenutzerIn in der Learning-Analytics-Datenbank machen zu können. Sollte es nun durch einen nicht berechtigten Zugriff auf die Datenbank zu einem Diebstahl von Learning-Analytics-Daten kommen, ist es für den Angreifer unmöglich, die jeweiligen Datensätze realen BenutzerInnen zuzuordnen.

4.4.2. Informationsvisualisierung

Obwohl in der Learning Analytics Datenbank beinahe alle auf der MOOC-Plattform erkennbaren Aktivitäten der Studierenden gespeichert sind, ist es noch nicht möglich, alle diese Informationen über das Web-Interface zu visualisieren. Hier bedarf es einer Weiterentwicklung dieses Web-Interfaces. Um sehr gezielt Informationen über Aktivitäten auf der Plattform zu erlangen, muss zur Zeit noch direkt auf die Datenbank zugegriffen werden.

5. Zusammenfassung

Diese Arbeit beschreibt einen Prototypen, wie Learning Analytics im Kontext von Massive Open Online Courses verwendet werden kann. Dazu wurde ein Learning-Analytics-System entwickelt, das auf den Daten der MOOC-Plattform imoox.at, initiiert von der Universität Graz und der Technischen Universität Graz, basiert.

Zum Zwecke der Analyse wurden Log-Dateien (protokollierte Aktivitäten der BenutzerInnen der MOOC-Plattform) herangezogen um daraus Verhaltensmuster und Aktivitäten von Studierenden zu erkennen und zu verarbeiten. Diese Log-Dateien wurden auf einem eigens dafür bereitgestellten Server analysiert und verarbeitet. Die Verarbeitung geschieht in der Regel anhand einer Schlüsselwortextraktion, auf Basis dieser eine Aktivität erkannt wird und in weiterer Folge zusätzliche Informationen zu dieser Aktivität extrahiert werden. Nach erfolgreicher Verarbeitung (zB. das Erkennen, dass der/die Studierende sich ein Video angesehen hat) wird diese Information zur weiteren Benutzung auf einem Datenbankserver gespeichert.

Als zweiter großer Teil dieser Arbeit wurde eine Web-Plattform erstellt, die die Daten vom Datenbankserver zur Visualisierung bringt. Der/die BenutzerIn dieser Plattform hat die Möglichkeit, sich individualisierte Informationen zu einem jeweiligen Kurs, zu den angebotenen Inhalten eines Kurses bzw. zu einem/r Studierenden auf der MOOC-Plattform anzeigen

5. Zusammenfassung

zu lassen. Diese Informationen werden meist in Form von Diagrammen präsentiert. Es ist jedoch auch möglich sich textuelle Informationen anzeigen zu lassen.

Diese Visualisierungen dienen als Entscheidungshilfe für Lehrende, um die angebotenen Kurse und deren Inhalte auf der MOOC-Plattform gezielt verbessern zu können. Es dient jedoch auch als Hilfe dafür, frühzeitig Studierende zu identifizieren, bei denen es absehbar ist, dass sie am Kurs nicht mehr aktiv teilnehmen bzw. den Kurs nicht erfolgreich abschließen werden können. Zu letzterem Zweck wäre es jedoch auch notwendig, die MOOC-Plattform selbst zu erweitern, was in Kapitel 6 erläutert wurde.

Als generelle Erkenntnis bei der Auswertung dieser Arbeit wäre zu erwähnen, dass durchaus signifikante Abhängigkeiten zwischen Inhaltskonsumation und Abschneiden in den angebotenen Kursen zu erkennen sind. Es lassen sich beispielsweise Zusammenhänge zwischen den im Kurs heruntergeladenen Dateien und dem zugehörigen Abschneiden bei den Quizzes in Mittelwert und Medianwert über die Gesamtheit der Studierenden ausmachen. Eine weitere Erkenntnis bei der Evaluation der angebotenen Kurse ist jedoch, dass die Kollaboration durch die Studierenden über die Kursdauer doch sehr deutlich abnimmt. Es war in fast allen Kursen zu beobachten, dass nach der zweiten Woche viele der eigentlich angemeldeten Studierenden sich nicht mehr aktiv am Kurs beteiligten. Diese Beobachtungen bestärken uns jedoch nur darin, die Wichtigkeit von Learning Analytics nicht hoch genug einzuschätzen, denn genau hier bietet Learning Analytics in MOOCs den Ansatzpunkt, deutliche Verbesserungen herbeiführen zu können.

Eine detaillierte Literaturanalyse in Kapitel 2 zeigt zusätzlich, dass das Thema Learning Analytics in Massive Open Online Courses jedenfalls ein Zukunftsträchtiges ist. Viele der großen amerikanischen Universitäten, hier seien stellvertretend Stanford University und Harvard University erwähnt,

bieten mittlerweile Teile ihres Studienangebots in Form von MOOCs an. Gleichzeitig entstehen auf diesen Universitäten zur Zeit Learning-Analytics-Systeme zur Unterstützung der angebotenen MOOCs, was uns bezüglich des Einsatzes und der Weiterentwicklung der hier vorgestellten Learning-Analytics-Plattform nur bestärken kann.

6. Ausblick

Dieses Kapitel diskutiert mögliche Weiterentwicklungen und Adaptierungen des Learning-Analytics-Systems. Die aktuelle Entwicklung, obwohl sie durchaus schon sehr umfangreich ist, lässt noch vieles an Verbesserungen zu. Dies birgt auch noch eine Menge Potenzial für weitere Arbeiten. Die folgenden Punkte sollten darin behandelt werden.

6.1. Integration von Learning Analytics in die MOOC-Plattform

Im jetzigen Entwicklungsstatus des Learning-Analytics-Systems ist es nicht vorgesehen, den Studierenden persönlich Feedback zu ihren Leistungen in den Kursen auf der MOOC-Plattform zu geben. Stattdessen haben nur berechnete Lehrende Zugriff auf die Daten des Learning-Analytics-Systems. Es wäre jedoch wichtig, die gewonnenen Informationen auch den Studierenden individuell mitzuteilen. Um ein solches persönliches Feedback umsetzen zu können, bedarf es aber einer technischen Erweiterung der MOOC-Plattform selbst, um Learning Analytics Informationen dort einbinden zu können. Aus meiner Sicht wäre es vorstellbar, den BenutzerInnen eine Leistungsübersicht anzubieten, auf der sie selbst sehen können, was sie im Kurs

6. Ausblick

bereits bzw. was sie noch nicht geleistet haben. Weiters könnten sie auch gewarnt werden, falls es absehbar ist, dass sie den Kurs nicht erfolgreich abschließen werden. Zur Motivation wäre es auch möglich, Ranglisten der aktivsten KursteilnehmerInnen anzuzeigen.

Um die Qualität des Learning-Analytics-Systems zu verbessern, wäre es auch vorstellbar, Studierende persönlich über die MOOC-Plattform nach ihrer Meinung zum Kurs bzw. zu Kursinhalten, mittels kurzen Zwischenfragen oder Formularen, befragen zu können.

6.2. Weiterentwicklung der Web-Plattform

Die Web-Plattform bietet derzeit bereits viele Möglichkeiten sich individuelle Informationen zu Kursen anzeigen zu lassen, doch haben wir aus Gesprächen mit Kursbetreuern erfahren, dass gewisse Informationen, obwohl sie am Datenbankserver vorhanden sind, noch nicht ausreichend präsentiert werden. Aus diesem Grund ist es zukünftig notwendig, auch die Web-Plattform zur Visualisierung der Learning-Analytics-Informationen weiterzuentwickeln. Dabei geht es rein um die Weiterentwicklung clientseitig. Serverseitig ist keine Adaption notwendig.

6.3. Verbesserungen in der Datenbank

Auf der Datenbank, die die Learning-Analytics-Daten beinhaltet, werden datenschutzrelevante Informationen derzeit im Klartext gespeichert. Der Zugriff auf diesen Datenbankserver ist natürlich auf wenige Personen

6.3. Verbesserungen in der Datenbank

eingeschränkt, und die Daten darauf sind auch nicht älter als 3 Monate, dennoch ist es notwendig einige Informationen zu schützen. Vorstellbar wäre, dass Benutzernamen gehasht werden. Dadurch bleiben Learning-Analytics-Daten natürlich weiterhin eindeutig zuordenbar, doch der/die BetreuerIn der Datenbank kann nicht mehr auf den ersten Blick nachvollziehen, welche Daten zu welchem/r BenutzerIn gehören. Weiters wäre es sinnvoll die gesamte Datenbank zu verschlüsseln.

Appendix

Appendix A.

Quelltext

A.1. Verarbeitung am Server (Serialization)

```
1 import splunklib.client as client
2 import splunklib.results as results
3 from time import sleep
4 import sys
5 import pickle
6 import os
7 from datetime import date, timedelta
8
9 import ConfigParser
10
11 config = ConfigParser.ConfigParser()
12
13 config.read('../config')
14
15 HOST = config.get('splunk', 'HOST')
16 PORT = config.get('splunk', 'PORT')
17 USERNAME = config.get('splunk', 'USERNAME')
18 PASSWORD = config.get('splunk', 'PASSWORD')
```

Appendix A. Quelltext

```
19 SOURCE = config.get('splunk', 'SOURCE')
20 DESTINATION_SERVER = config.get('splunk', 'DESTINATION_SERVER')
21
22 NUMBER_OF_RESULTS_PER_PAGINATION = config.get('splunk', '
    NUMBER_OF_RESULTS_PER_PAGINATION')
23
24 SERIALIZATION_PATH = config.get('DEFAULT', 'serializationpath')
25 SPLUNKLOGS = config.get('DEFAULT', 'splunklogspath')
26
27
28 OBJECTS_PER_PICKLE = 100
29 # Create a Service instance and log in
30 service = client.connect(
31     host=HOST,
32     port=PORT,
33     username=USERNAME,
34     password=PASSWORD)
35
36 # Run a blocking search
37 kwargs_normalsearch = { "exec_mode": "blocking",
38     "rf":["uri", "token", "login", "method"]}
39
40 today = date.today()
41 day_today = today.strftime('%d')
42 month_today = today.strftime('%m')
43 year_today = today.strftime('%Y')
44
45 print "today: "+day_today+" "+month_today+" "+year_today
46
47 yesterday = date.today() - timedelta(1)
48 day_yesterday = yesterday.strftime('%d')
49 month_yesterday = yesterday.strftime('%m')
50 year_yesterday = yesterday.strftime('%Y')
51
52 if len(sys.argv) == 1:
53     dayfrom = int(day_yesterday)
54     dayto = int(day_yesterday)
```

A.1. Verarbeitung am Server (Serialization)

```
55 month= str(month_yesterday)
56 year= str(year_yesterday)
57 else:
58     dayfrom = int(sys.argv[1])
59     dayto = int(sys.argv[2])
60     month = str(sys.argv[3])
61     year = str(sys.argv[4])
62
63
64 for i in range(dayfrom, dayto+1):
65
66     searchquery_normal = "search host=\""+DESTINATION_SERVER+"\\"
67         source=\""+SOURCE+"\" earliest="+month+"/"+str(i)+"/"+year+"
68         :00:00:00 latest="+month+"/"+str(i)+"/"+year+":23:59:59 |
69         reverse"
70
71     print searchquery_normal
72     job = service.jobs.create(searchquery_normal, **
73         kwargs_normalsearch)
74
75     print "Search job properties"
76     print "Search job ID:          ", job["sid"]
77     print "The number of events: ", job["eventCount"]
78     print "The number of results:", job["resultCount"]
79     print "Search duration:       ", job["runDuration"], "seconds"
80     print "This job expires in:   ", job["ttl"], "seconds"
81     print "_____\\n"
82     print "Search results:\\n"
83     resultCount= job['resultCount']
84     offset = 0
85     count = 10000
86     #for offsetting the search when too much searchresults: http://
87         dev.splunk.com/view/python-sdk/SP-CAAER5
88     filename = SERIALIZATION_PATH+"/"+month+"_"+year+"/"+month+
89         serialized_"+str(i)+"_"+month+"_"+year+".pickle"
90     dir = os.path.dirname(filename)
91     if not os.path.exists(dir):
92         os.makedirs(dir)
```

Appendix A. Quelltext

```
86
87
88 file = open(filename, "wb")
89
90
91 pickle.dump(job['resultCount'], file)
92
93 while (offset < int(resultCount)):
94     kwargs_paginate = {"count": count, "offset": offset}
95
96     blocksearch_results = job.results(**kwargs_paginate)
97
98
99     data = results.ResultsReader(blocksearch_results)
100
101     j = offset
102     for value in data:
103         if isinstance(value, dict):
104             pickle.dump(value, file)
105         elif isinstance(value, results.Message):
106             print value
107             if j % (int(job['resultCount']) / 100) == 0: #/100 is percent
108                 print float(j)/float(job['resultCount'])*100, " % done"
109             j = j + 1
110         offset += count
111     file.close()
112
113 #with open("serialized.pickle", "wb") as f:
114 #    pickle.dump(job['resultCount'], f)
115 #    for value in data:
116 #        pickle.dump(value, f)
117
118 f = open(SPLUNKLOGS, 'a+b')
119 f.write(day_yesterday+"_" +month_yesterday+"_" +year_yesterday+" OK\
120         n") # python will convert \n to os.linesep
```

A.2. Verarbeitung Server (Deserialization)

```
121 f.close()
```

Quelltext A.1: Python Quelltext Serialization

A.2. Verarbeitung Server (Deserialization)

```
1 from time import sleep
2 import sys
3 import pprint, pickle
4 import operator
5 import mysql.connector
6 import os
7 import re
8 from datetime import date, timedelta
9 from datetime import datetime
10
11 import ConfigParser
12
13 config = ConfigParser.ConfigParser()
14
15 config.read('../config')
16
17 DATABASE = config.get('mysql', 'DATABASE')
18 HOST = config.get('mysql', 'HOST')
19 USERNAME = config.get('mysql', 'USERNAME')
20 PASSWORD = config.get('mysql', 'PASSWORD')
21
22
23 SERIALIZATION_PATH = config.get('DEFAULT', 'serializationpath')
24 PROCESSINGLOGS = config.get('DEFAULT', 'processinglogspath')
25 videodir = config.get('DEFAULT', 'externalfilepath')+'/videos'
26 quizdir = config.get('DEFAULT', 'externalfilepath')+'/quizzes'
27
28
```

Appendix A. Quelltext

```
29
30 #screen Output options
31 screenOutput = True
32
33 printLogins = False
34 printLogouts = False
35 printQuizzesStarted = False
36 printQuizzesEnded = False
37 printDownloads = False
38 printForumRead = False
39 printForumWritten = False
40
41 printDeserializationProgress = True
42
43 serializeData = True
44
45
46
47 # Run a normal search—search everything, return 1st 10 events
48 ignored_rooms=["..", "mainOwp", "server"]
49 #splunk_jobs = service.jobs
50
51 #lookup token->username
52 logins_tokens = dict()
53
54 #dicts for displaying purposes
55 logins_number = dict()
56 quizzes_number = dict()
57 quizzes_per_user= dict()
58 downloads_number = dict()
59 downloads_per_user = dict()
60 forum_written = dict()
61 forum_read = dict()
62 forum_written_per_user = dict()
63 forum_read_per_user = dict()
64 courses_array = []
65 quizzes_array = []
```


A.2. Verarbeitung Server (Deserialization)

```
66
67 #mysql connection
68 cnx = mysql.connector.connect(user=USERNAME,password=PASSWORD,
    database=DATABASE, host=HOST)
69 curA = cnx.cursor(buffered=True)
70
71
72
73
74
75
76
77
78 #function to filter a specific string in between 2 given
    delimiters
79 def filterString(raw, first_delimiter ,second_delimiter):
80     result = raw.split(first_delimiter)
81     result = result[1]
82     result = result.split(second_delimiter)
83     result = result[0]
84     return result
85
86 def write2file(data, destination):
87     first_line = ''
88     second_line = ''
89     #print "NUMBER OF LOGINS PER USER: "
90     for item in data:
91         first_line = first_line+item[0]+" "
92         second_line = second_line+str(item[1])+" "
93         #print item
94     #print first_line
95     #print second_line
96
97     f = open(destination , 'w')
98     f.write(first_line.strip()+'\n') # python will convert \n to os.
        linesep
99     f.write(second_line.strip())
```

Appendix A. Quelltext

```
100     f.close()
101
102 def parseLogin(result,time,previous_time, previous_login_username)
103     :
104     if previous_time != None:
105         last_minutes = previous_time.split("T")[1].split(":")[1]
106     else:
107         last_minutes = "None"
108
109     recent_minutes = time.split("T")[1].split(":")[1]
110
111     if last_minutes != recent_minutes or (last_minutes ==
112         recent_minutes and (result['login'] !=
113         previous_login_username)): # if not in the same minute it's
114         OK, if in the same minute, usernames must be different
115         last_event = "LOGIN"
116     if screenOutput and printLogins:
117         print "LOGIN at ",time," by ", result['login']
118     if len(result['login']) < 50 and len(result['token']) < 45: #
119         too long usernames
120         logins_tokens[result['token']] = result['login']
121         insert_new_user = ("REPLACE INTO users (uname, token,
122             last_time) VALUES (%s,%s,%s)")
123         curA.execute(insert_new_user ,( result['login'],result['token
124             '],time))
125         insert_login = ("INSERT INTO logins (uname, time) VALUES (%s
126             ,%s)")
127         curA.execute(insert_login ,(result['login'], time))
128         cnx.commit()
129
130         if result['login'] in logins_number:
131             logins_number[result['login']] += 1
132         else:
133             logins_number[result['login']] = 1
134     else:
135         print "Username "+result['login']+ " too long, max 50 chars"
```

A.2. Verarbeitung Server (Deserialization)

```
129
130 def parseLogout(result, time):
131     if screenOutput and printLogouts:
132         if result['token'] in logins_tokens.keys():
133             print "LOGOUT by          ", logins_tokens[result['token']],
134                 at ", time
135         else:
136             print "LOGOUT by          ", result['token'], " at ", time
137
138 def parseQuizStarted(result, time, previous_time,
139                     previous_time_quiz_started_username):
140
141     if previous_time != None:
142         last_minutes = previous_time.split("T")[1].split(":")[1]
143     else:
144         last_minutes = "None"
145
146     recent_minutes = time.split("T")[1].split(":")[1]
147
148     username=filterString(result['uri'], 'mainOwp/', '- ')
149     quizname=filterString(result['uri'], username+'-', '.scorm')
150     coursename=quizname.split("-")[0]
151
152     if last_minutes != recent_minutes or (last_minutes ==
153         recent_minutes and (username !=
154         previous_time_quiz_started_username)): # if not in the same
155         minute it's OK, if in the same minute, usernames must be
156         different
157
158     if screenOutput and printQuizzesStarted:
159         print "QUIZ          ", quizname, "started by ", username,
160             " at ", time
161
162     if quizname in quizzes_number:
163         quizzes_number[quizname] += 1
164     else:
165         quizzes_number[quizname] = 1
166
167     if username in quizzes_per_user:
```

Appendix A. Quelltext

```
159     quizzes_per_user[username] +=1
160     else:
161         quizzes_per_user[username] = 1
162
163     if len(quizname) < 80 and len(username) < 80:
164         insert_quiz_started = ("INSERT INTO quizzes_started (qname,
165             uname, time) VALUES (%s,%s,%s)")
166         curA.execute(insert_quiz_started ,(quizname ,username ,time))
167         cnx.commit()
168 def parseQuizEnded(result ,time ,previous_time):
169
170     username=filterString(result['uri'],'mainOwp/','_')
171     quizname=filterString(result['uri'], username+'_', '.scorm')
172     if screenOutput and printQuizzesEnded:
173         print "QUIZ          ", quizname, "ended by ", username, " at ",
174             time
175 def parseDownload(result ,time ,previous_time ,
176     previous_time_download_username):
177     if previous_time != None:
178         last_minutes = previous_time.split("T")[1].split(":")[1]
179     else:
180         last_minutes = "None"
181     recent_minutes = time.split("T")[1].split(":")[1]
182
183
184     if result['token'] in logins_tokens.keys():
185         if last_minutes != recent_minutes or (last_minutes ==
186             recent_minutes and (logins_tokens[result['token']] !=
187                 previous_time_download_username)): # if not in the same
188             minute it's OK, if in the same minute, usernames must be
189             different
190
191     #if "token" in result.keys() and result['token'] in
192         logins_tokens.keys():
```

A.2. Verarbeitung Server (Deserialization)

```
188
189     filename = 'unknown'
190     for course in courses_array:
191         course_adapted = course + "/"
192         if course_adapted in result['uri']:
193             filename=filterString(result['uri'],course_adapted,'?
                token')
194             if len(filename) < 80 and len(course) < 80:
195                 curA.execute("INSERT IGNORE INTO files (fname,cname)
                    VALUES (%s,%s)",(filename ,course))
196                 cnx.commit()
197             if len(filename) < 80 and len(logins_tokens[result['
                token']]) < 80:
198                 curA.execute("INSERT INTO files_downloaded (fname,
                    uname,time) VALUES (%s,%s,%s)",(filename ,
                    logins_tokens[result['token']],time))
199                 cnx.commit()
200             #print filename
201
202             if screenOutput and printDownloads:
203                 print "DOWNLOAD of file ",filename, "by ",logins_tokens[
                    result['token']], "at ",time
204             if filename in downloads_number:
205                 downloads_number[filename] +=1
206             else:
207                 downloads_number[filename] = 1
208
209             if logins_tokens[result['token']] in downloads_per_user:
210                 downloads_per_user[logins_tokens[result['token']]] += 1
211             else:
212                 downloads_per_user[logins_tokens[result['token']]] = 1
213
214 def parseForumWritten(result ,time ,previous_time ,
    previous_time_forum_written_username):
215
216     if previous_time != None:
217         last_minutes = previous_time.split("T")[1].split(":")[1]
```

Appendix A. Quelltext

```
218 else:
219     last_minutes = "None"
220
221 recent_minutes = time.split("T")[1].split(":")[1]
222
223 if result['token'] in logins_tokens.keys():
224     if last_minutes != recent_minutes or (last_minutes ==
        recent_minutes and (logins_tokens[result['token']] !=
            previous_time_forum_written_username)): # if not in the
                same minute it's OK, if in the same minute, usernames must
                be different
225     #if "token" in result.keys() and result['token'] in
        logins_tokens.keys():
226     forumname=filterString(result['uri'], "room=", "&")
227     if screenOutput and printForumWritten:
228         print "FORUM entry written in ", filterString(result['uri']
            ], "room=", "&"), " by ", logins_tokens[result['token']],
            " at ", time
229     if forumname in forum_written:
230         forum_written[forumname] +=1
231     else:
232         forum_written[forumname] = 1
233
234     if logins_tokens[result['token']] in forum_written_per_user:
235         forum_written_per_user[logins_tokens[result['token']]] +=
            1
236     else:
237         forum_written_per_user[logins_tokens[result['token']]] = 1
238     #if forumname in courses_array:
239     if len(forumname) < 80 and len(logins_tokens[result['token']
        ]]) < 80:
240         curA.execute("INSERT INTO forums_written (fname,uname,time
            ) VALUES (%s,%s,%s)",(forumname, logins_tokens[result['
            token']], time))
241         cnx.commit()
242
```

A.2. Verarbeitung Server (Deserialization)

```
243 def parseForumRead(result ,time ,previous_time ,
    previous_time_forum_read_username):
244     if previous_time != None:
245         last_minutes = previous_time.split("T")[1].split(":")[1]
246     else:
247         last_minutes = "None"
248
249     recent_minutes = time.split("T")[1].split(":")[1]
250
251
252
253     if result['token'] in logins_tokens.keys():
254         if last_minutes != recent_minutes or (last_minutes ==
            recent_minutes and (logins_tokens[result['token']] !=
            previous_time_forum_read_username)): # if not in the same
            minute it's OK, if in the same minute, usernames must be
            different
255
256         room=filterString(result['uri'], "file=", "/")
257
258         if result['token'] in logins_tokens.keys() and room not in
            ignored_rooms:
259             #saveCourseName(room) #TODO: welche raeume gibts da alles
                ??
260             if room in forum_read:
261                 forum_read[room] +=1
262             else:
263                 forum_read[room] = 1
264
265             if logins_tokens[result['token']] in forum_read_per_user:
266                 forum_read_per_user[logins_tokens[result['token']]] += 1
267             else:
268                 forum_read_per_user[logins_tokens[result['token']]] = 1
269             if screenOutput and printForumRead:
270                 print "FORUM entry in course ", room," read by ",
                    logins_tokens[result['token']], " at ",time
271
```

Appendix A. Quelltext

```
272     if room in courses_array:
273         if len(room) < 80:
274             curA.execute("INSERT IGNORE INTO forums (fname,cname)
                VALUES (%s,%s)" ,(room,room))
275             cnx.commit()
276         if len(room) < 80 and len(logins_tokens[result['token'
                ]]) < 80:
277             curA.execute("INSERT INTO forums_read (fname,uname,
                time) VALUES (%s,%s,%s)" ,(room,logins_tokens[result
                ['token']],time))
278             cnx.commit()
279
280 def parseCourseNames():
281     #Scan existing quizzes and existing courses
282     for file in os.listdir(quizdir):
283         if file.endswith(".txt"):
284             # file
285
286             coursename = '_'.join(file.split("_")[0:len(file.split("_"))
                -1])
287             #print coursename
288
289             saveCourseName(coursename)
290             destination = quizdir+"/"+str(file)
291             f = open(destination, 'r')
292             for line in f:
293                 quizzes = line.split(";")
294                 for quiz in quizzes:
295                     if len(quiz) < 80 and len(coursename) < 80:
296                         if len(quiz) != 0:
297
298                             insert_quiz = ("INSERT IGNORE INTO quizzes (qname,
                cname) VALUES (%s,%s)")
299                             curA.execute(insert_quiz ,(quiz ,coursename))
300                             cnx.commit()
301                             if quiz not in quizzes_array:
302                                 quizzes_array.append(quiz)
```


A.2. Verarbeitung Server (Deserialization)

```
303         #else:
304             #print "undefined quizname: *"+quiz+"* for course "+
                coursename
305             #insert_quiz = ("INSERT IGNORE INTO quizzes (qname,
                cname) VALUES (%s,%s)")
306             #curA.execute(insert_quiz,(" undefined",coursename))
307             #cnx.commit()
308
309
310     f.close()
311
312 def parseQuizAttempts():
313     print "BEGIN PARSING QUIZ ATTEMPTS"
314     #Scan quiz attempts and progress and write to DB
315     #print quizzes_array
316     for file in os.listdir(quizdir):
317
318         progress_array = []
319         if file.endswith(".scorm"):
320             username = "undefined"
321             quizname = "undefined"
322             for quiz in quizzes_array:
323
324                 if quiz in file:
325                     stripstring = "_" + quiz + ".scorm"
326                     username = file.split(stripstring)[0]
327                     quizname = quiz
328
329             destination = quizdir + "/" + str(file)
330             f = open(destination, 'r')
331             for line in f:
332
333                 if "progress" in line: #progress not included in first
                    versions
334                     progress_array = ['o', 'o', 'o', 'o', 'o']
335                     progress = line.split("progress")[1]
336                     progress = progress.replace("%", "")
```

Appendix A. Quelltext

```
337     progress = progress.replace(":", "")
338     progress = progress.split(";")
339
340     i = 0
341     #print len(progress)
342     for item in progress:
343         if i < 5:
344             if item == '':
345                 progress_array[i]='o'
346             else:
347                 progress_array[i]=item
348             i+=1
349
350     else: #if progress was not yet included, fill with nulls
351         progress_array = ['o','o','o','o','o']
352     #print progress_array, username, quizname
353
354
355     first = get_int_secure(progress_array[0])
356     second = get_int_secure(progress_array[1])
357     third = get_int_secure(progress_array[2])
358     fourth = get_int_secure(progress_array[3])
359     fifth = get_int_secure(progress_array[4])
360
361
362     if len(quizname) < 80 and len(username) < 80 and quizname
363         != "undefined" and username != "undefined":
364         insert_attempt = ("REPLACE INTO quiz_attempts (qname,
365             uname, first, second, third, fourth, fifth) VALUES(%s
366             ,%s,%s,%s,%s,%s,%s,%s)")
367
368         insert_data = (quizname, username, first, second, third,
369             fourth, fifth)
370         curA.execute(insert_attempt, insert_data)
371         cnx.commit()
372
373     f.close()
```

A.2. Verarbeitung Server (Deserialization)

```
370 print "END PARSING QUIZ ATTEMPTS"
371
372 def parseVideosWatched():
373     print "BEGIN PARSING VIDEOS"
374     for subdir, dirs, files in os.walk(videodir):
375         for dir in dirs:
376             #print os.path.join(subdir, file)
377             #print dir
378             saveCourseName(dir)
379
380             destination = videodir+"/"+dir+"/youtube.log.arc"
381
382             try:
383                 f = open(destination, 'r')
384
385
386                 for line in f:
387                     if not re.match(r'^\s*$', line) and "[|]" not in line:
388                         # line is empty (has only the following: \t\n\r and
389                         # whitespace)
390                         if len(line.split("%:%")) == 6:
391                             date = line.split("%:%")[0]
392                             user = line.split("%:%")[1]
393                             status = line.split("%:%")[2]
394                             videoid = line.split("%:%")[3]
395                             watched_time = line.split("%:%")[4]
396                             total_time = line.split("%:%")[5]
397                         elif len(line.split("%:%")) == 5:
398                             date = line.split("%:%")[0]
399                             user = line.split("%:%")[1]
400                             status = line.split("%:%")[2]
401                             videoid = line.split("%:%")[3]
402                             watched_time = line.split("%:%")[4].split("%;%")[0]
403                             total_time = line.split("%:%")[4].split("%;%")[1]
404                         else:
405                             print "Format of Video Watched not readable"
```

Appendix A. Quelltext

```
405         #print date , dir
406
407     try:
408         converted_date = datetime.strptime(' '.join(date.
409             split(" ")[1:5]), '%b %d %Y %H:%M:%S')
410         #RESULT: wieviel prozent eines videos hat ein user
411             angeschaut
412         #SELECT uname, vname, MAX(watched_time/total_time *
413             100) as percent_watched FROM videos_all GROUP BY
414             uname, vname
415
416         if len(user) < 80:
417             insert_attempt = ("REPLACE INTO videos_all (vname,
418                 uname, status, watched_time, total_time, date)
419                 VALUES(%s,%s,%s,%s,%s,%s)")
420             insert_data = (videoid, user, status, watched_time
421                 , total_time, converted_date)
422             curA.execute(insert_attempt, insert_data)
423             insert_video = ("INSERT IGNORE INTO videos (
424                 videoid, fullvideoname, cname) VALUES (%s,%s,%s
425                 )")
426             curA.execute(insert_video ,(videoid ,videoid , dir))
427
428             cnx.commit()
429         except ValueError:
430
431             print "Wrong
432                 date format:
433                 "+date
434
435
436         f.close()
437     except IOError:
438         print "File "+destination+" not found"
439
440 print "END parsing Videos"
```

A.2. Verarbeitung Server (Deserialization)

```
430 #Delete videos which are in ignored_videos —> the ones that are
      not online
431 delete_video = ("DELETE FROM videos WHERE videoid IN ( SELECT
      videoid FROM ignored_videos)")
432 curA.execute(delete_video)
433 cnx.commit()
434
435 def saveCourseName(coursename):
436     if coursename not in ignored_rooms:
437         #print coursename
438         curA.execute("INSERT IGNORE INTO courses (cname, fullname)
      VALUE (%s,%s)", (coursename, coursename))
439         courses_array.append(coursename)
440         cnx.commit()
441
442
443 def get_int_secure(s):
444     try:
445         return int(float(s))
446     except ValueError:
447         return 0
448
449 splunk_data = []
450
451 #Deserialization
452
453
454 today = date.today()
455 day_today = today.strftime('%d')
456 month_today = today.strftime('%m')
457 year_today = today.strftime('%Y')
458
459 print "today: "+day_today+" "+month_today+" "+year_today
460
461 yesterday = date.today() - timedelta(1)
462 day_yesterday = yesterday.strftime('%d')
463 month_yesterday = yesterday.strftime('%m')
```

Appendix A. Quelltext

```
464 year_yesterday = yesterday.strftime( '%Y' )
465
466
467 if len(sys.argv) == 1:
468     dayfrom = int(day_yesterday)
469     dayto = int(day_yesterday)
470     month = month_yesterday
471     year = year_yesterday
472 else:
473     dayfrom = int(sys.argv[1])
474     dayto = int(sys.argv[2])
475     month = sys.argv[3]
476     year = sys.argv[4]
477
478 if serializeData:
479     for i in range(dayfrom-1,dayto):
480         if screenOutput and printDeserializationProgress:
481             print "File "+str(i+1)+" opened"
482             file = open(SERIALIZATION_PATH+"/"+month+"_"+year+"/"+"
                        serialized_"+str(i+1)+"_"+month+"_"+year+".pickle", "rb")
483             for _ in range(int(pickle.load( file ))):
484                 splunk_data.append(pickle.load( file ))
485             if screenOutput and printDeserializationProgress:
486                 print "File "+str(i+1)+" deserialized"
487
488             file.close()
489             if screenOutput and printDeserializationProgress:
490                 print "File "+str(i+1)+" closed"
491
492 parseCourseNames()
493
494 parseVideosWatched()
495
496 parseQuizAttempts()
497
498 previous_uri = None
499 previous_time_login = None
```

A.2. Verarbeitung Server (Deserialization)

```
500 previous_login_username = None
501 previous_time_download = None
502 previous_time_download_username = None
503 previous_time_forum_read = None
504 previous_time_forum_read_username = None
505 previous_time_forum_written = None
506 previous_time_forum_written_username = None
507 previous_time_quiz_started = None
508 previous_time_quiz_started_username = None
509 previous_time_quiz_ended = None
510
511 # iterate over all splunk data chronologically
512
513 if serializeData:
514     for result in splunk_data:
515         if "_time" in result.keys():
516             time = result['_time'].split("+")
517             time = time[0]
518             #cut the timezone correction, mysql datetime format doesn't
             like it
519         else:
520             time = 'not defined'
521
522
523     if 'uri' in result.keys() and result['uri'] != previous_uri: #
         kick simple redundant logs
524     #LOGIN
525     if "get_user_card" in result['uri'] and 'token' in result.
         keys() and 'login' in result.keys():
526         parseLogin(result, time, previous_time_login,
             previous_login_username)
527         previous_time_login = time
528         previous_login_username = result['login']
529     #LOGOUT
530     if "logout" in result['uri']:
531         parseLogout(result, time)
532     #QUIZ
```

Appendix A. Quelltext

```
533     if "method" in result.keys():
534         if result['method']=="GET" and ".scorm" in result['uri']:
535             parseQuizStarted(result,time,previous_time_quiz_started,
                    previous_time_quiz_started_username)
536             previous_time_quiz_started = time
537             previous_time_quiz_started_username=filterString(result[
                    'uri'],'mainOwp/','_')
538         if result['method']=="POST" and ".scorm" in result['uri']
                    :
539             parseQuizEnded(result,time,previous_time_quiz_ended)
540             previous_time_quiz_ended = time
541     #DOWNLOAD
542     if "/wbtmaster/threads" in result['uri'] and 'token' in
                    result.keys():
543         parseDownload(result,time,previous_time_download,
                    previous_time_download_username)
544         previous_time_download = time
545         if result['token'] in logins_tokens.keys():
546             previous_time_download_username = logins_tokens[result[
                    'token']]
547     #FORUM
548     if "/wbtmaster/groovy/addForum.groovy?room" in result['uri']
                    and 'token' in result.keys():
549         parseForumWritten(result,time,previous_time_forum_written,
                    previous_time_forum_written_username)
550         previous_time_forum_written = time
551         if result['token'] in logins_tokens.keys():
552             previous_time_forum_written_username = logins_tokens[
                    result['token']]
553     if "/wbtmaster/forum.Forum?action=get&file=" in result['uri']
                    and 'token' in result.keys():
554         parseForumRead(result,time,previous_time_forum_read,
                    previous_time_forum_read_username)
555         previous_time_forum_read = time
556         if result['token'] in logins_tokens.keys():
557             previous_time_forum_read_username = logins_tokens[result
                    ['token']]
```


A.2. Verarbeitung Server (Deserialization)

```
558     previous_uri = result['uri']
559
560
561
562
563
564
565 #save how many times a user has logged in
566 for key, value in logins_number.iteritems():
567     if len(key) < 80:
568         insert_stmt = ("REPLACE INTO number_of_logins (uname, times)
569             VALUES(%s,%s)")
570         insert_data = (key, value)
571         curA.execute(insert_stmt, insert_data)
572 #TODO: druber nachdenken das commit ueberall anders auch aus den
573     schleifen rauszugeben!!!!!!
574
575 curA.close()
576 cnx.close()
577
578
579 f = open(PROCESSINGLOGS, 'a+b')
580 f.write(day_yesterday+"_"+month_yesterday+"_"+year_yesterday+" OK\
581     n") # python will convert \n to os.linesep
582 f.close()
583
584
585 #sort dicts
586 #sorted_logins = sorted(logins_number.iteritems(), key=operator.
587     itemgetter(1))
588 #sorted_quizzes = sorted(quizzes_number.iteritems(), key=operator.
589     itemgetter(1))
590 #sorted_quizzes_per_user = sorted(quizzes_per_user.iteritems(),
591     key=operator.itemgetter(1))
```

Appendix A. Quelltext

```
589 #sorted_downloads = sorted(downloads_number.iteritems(), key=
    operator.itemgetter(1))
590 #sorted_downloads_per_user = sorted(downloads_per_user.iteritems()
    , key=operator.itemgetter(1))
591 #sorted_forum_written = sorted(forum_written.iteritems(), key=
    operator.itemgetter(1))
592 #sorted_forum_written_per_user = sorted(forum_written_per_user.
    iteritems(), key=operator.itemgetter(1))
593 #sorted_forum_read = sorted(forum_read.iteritems(), key=operator.
    itemgetter(1))
594 #sorted_forum_read_per_user = sorted(forum_read_per_user.iteritems
    (), key=operator.itemgetter(1))
595
596 #'/Users/Stephan/Sites/jqueryCharts/examples/logins.txt'
597 #write2file(sorted_logins, config.get('DEFAULT', 'basepath')+'data/
    logins.txt')
598 #rite2file(sorted_quizzes, config.get('DEFAULT', 'basepath')+'data/
    quizzes.txt')
599 #write2file(sorted_quizzes_per_user, config.get('DEFAULT', 'basepath
    ')+'data/quizzes_per_user.txt')
600 #write2file(sorted_downloads, config.get('DEFAULT', 'basepath')+'
    data/downloads.txt')
601 #write2file(sorted_downloads_per_user, config.get('DEFAULT', '
    basepath')+'data/downloads_per_user.txt')
602 #write2file(sorted_forum_written, config.get('DEFAULT', 'basepath')+'
    /data/forum_written.txt')
603 #write2file(sorted_forum_written_per_user, config.get('DEFAULT', '
    basepath')+'data/forum_written_per_user.txt')
604 #write2file(sorted_forum_read, config.get('DEFAULT', 'basepath')+'
    data/forum_read.txt')
605 #write2file(sorted_forum_read_per_user, config.get('DEFAULT', '
    basepath')+'data/forum_read_per_user.txt')
```

Quelltext A.2: Python Quelltext Deserialization

A.3. Datenbank Struktur

```

1 — phpMyAdmin SQL Dump
2 — version 4.0.0
3 — http://www.phpmyadmin.net
4 —
5 — Host: dbo2.tugraz.at
6 — Generation Time: Dec 10, 2014 at 09:25 AM
7 — Server version: 5.0.95
8 — PHP Version: 5.3.27
9
10 SET SQLMODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
    */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8 */;
18
19 —
20 — Database: 'vl_la'
21 —
22
23 —————
24
25 —
26 — Table structure for table 'courses'
27 —
28
29 CREATE TABLE IF NOT EXISTS 'courses' (
30   'cname' varchar(80) NOT NULL,
31   'fullname' varchar(80) default NULL,
32   PRIMARY KEY ('cname'),
33   UNIQUE KEY 'cname_UNIQUE' ('cname')
34 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Appendix A. Quelltext

```
35 |
36 |-----
37 |
38 |---
39 |--- Table structure for table 'files'
40 |---
41 |
42 |CREATE TABLE IF NOT EXISTS 'files' (
43 |  'fname' varchar(80) NOT NULL,
44 |  'cname' varchar(80) default NULL,
45 |  PRIMARY KEY ('fname'),
46 |  UNIQUE KEY 'fname.UNIQUE' ('fname')
47 | ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
48 |
49 |-----
50 |
51 |---
52 |--- Table structure for table 'files_downloaded'
53 |---
54 |
55 |CREATE TABLE IF NOT EXISTS 'files_downloaded' (
56 |  'id' int(11) NOT NULL auto_increment,
57 |  'fname' varchar(80) default NULL,
58 |  'uname' varchar(80) default NULL,
59 |  'time' datetime default NULL,
60 |  PRIMARY KEY ('id'),
61 |  UNIQUE KEY 'id.UNIQUE' ('id')
62 | ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO.INCREMENT=15616 ;
63 |
64 |-----
65 |
66 |---
67 |--- Table structure for table 'forums'
68 |---
69 |
70 |CREATE TABLE IF NOT EXISTS 'forums' (
71 |  'fname' varchar(80) NOT NULL,
```

A.3. Datenbank Struktur

```
72 'cname' varchar(80) default NULL,  
73 PRIMARY KEY ('fname'),  
74 UNIQUE KEY 'fname.UNIQUE' ('fname')  
75 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;  
76  
77 -----  
78  
79 ---  
80 --- Table structure for table 'forums_read'  
81 ---  
82  
83 CREATE TABLE IF NOT EXISTS 'forums_read' (  
84 'id' int(11) NOT NULL auto_increment,  
85 'fname' varchar(80) default NULL,  
86 'uname' varchar(80) default NULL,  
87 'time' datetime default NULL,  
88 PRIMARY KEY ('id'),  
89 UNIQUE KEY 'id.UNIQUE' ('id')  
90 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=103366 ;  
91  
92 -----  
93  
94 ---  
95 --- Table structure for table 'forums_written'  
96 ---  
97  
98 CREATE TABLE IF NOT EXISTS 'forums_written' (  
99 'id' int(11) NOT NULL auto_increment,  
100 'fname' varchar(80) default NULL,  
101 'uname' varchar(80) default NULL,  
102 'time' datetime default NULL,  
103 PRIMARY KEY ('id'),  
104 UNIQUE KEY 'id.UNIQUE' ('id')  
105 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2615 ;  
106  
107 -----  
108
```

Appendix A. Quelltext

```
109 —
110 — Table structure for table 'ignored_videos'
111 —
112
113 CREATE TABLE IF NOT EXISTS 'ignored_videos' (
114   'videoid' varchar(30) NOT NULL,
115   PRIMARY KEY ('videoid')
116 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
117
118 —————
119
120 —
121 — Table structure for table 'logins'
122 —
123
124 CREATE TABLE IF NOT EXISTS 'logins' (
125   'id' int(11) NOT NULL auto_increment,
126   'uname' varchar(80) default NULL,
127   'time' datetime default NULL,
128   PRIMARY KEY ('id')
129 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=72040 ;
130
131 —————
132
133 —
134 — Table structure for table 'number_of_logins'
135 —
136
137 CREATE TABLE IF NOT EXISTS 'number_of_logins' (
138   'uname' varchar(80) NOT NULL,
139   'times' int(11) default NULL,
140   PRIMARY KEY ('uname')
141 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
142
143 —————
144
145 —
```

A.3. Datenbank Struktur

```
146 — Table structure for table 'quizzes'
147 —
148
149 CREATE TABLE IF NOT EXISTS 'quizzes' (
150   'qname' varchar(80) NOT NULL,
151   'cname' varchar(80) default NULL,
152   PRIMARY KEY ('qname'),
153   UNIQUE KEY 'qname.UNIQUE' ('qname')
154 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
155
156 —————
157
158 —
159 — Table structure for table 'quizzes_started'
160 —
161
162 CREATE TABLE IF NOT EXISTS 'quizzes_started' (
163   'id' int(11) NOT NULL auto_increment,
164   'qname' varchar(80) default NULL,
165   'uname' varchar(80) default NULL,
166   'time' datetime default NULL,
167   PRIMARY KEY ('id'),
168   UNIQUE KEY 'id.UNIQUE' ('id')
169 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=54616 ;
170
171 —————
172
173 —
174 — Table structure for table 'quiz_attempts'
175 —
176
177 CREATE TABLE IF NOT EXISTS 'quiz_attempts' (
178   'qname' varchar(80) NOT NULL,
179   'uname' varchar(80) NOT NULL,
180   'first' int(11) default NULL,
181   'second' int(11) default NULL,
182   'third' int(11) default NULL,
```

Appendix A. Quelltext

```
183 'fourth' int(11) default NULL,  
184 'fifth' int(11) default NULL,  
185 PRIMARY KEY ('qname', 'uname')  
186 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

187

188

189

190

191 — Table structure for table 'users'

192

193

```
194 CREATE TABLE IF NOT EXISTS 'users' (  
195 'uname' varchar(80) NOT NULL,  
196 'token' varchar(80) default NULL,  
197 'last_time' datetime default NULL,  
198 PRIMARY KEY ('uname'),  
199 UNIQUE KEY 'uname_UNIQUE' ('uname')  
200 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

201

202

203

204

205 — Table structure for table 'videos'

206

207

```
208 CREATE TABLE IF NOT EXISTS 'videos' (  
209 'videoid' varchar(50) NOT NULL,  
210 'fullvideoname' varchar(150) NOT NULL,  
211 'cname' varchar(80) NOT NULL,  
212 PRIMARY KEY ('videoid')  
213 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

214

215

216

217

218 — Table structure for table 'videos_all'

219

A.3. Datenbank Struktur

```
220
221 CREATE TABLE IF NOT EXISTS 'videos_all' (
222   'vname' varchar(45) collate utf8_bin NOT NULL,
223   'uname' varchar(80) collate utf8_bin NOT NULL,
224   'status' int(11) NOT NULL,
225   'watched_time' int(11) NOT NULL,
226   'total_time' int(11) NOT NULL,
227   'date' datetime NOT NULL,
228   PRIMARY KEY ('vname', 'uname', 'status', 'watched_time', '
      total_time', 'date')
229 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
230
231 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
232 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
233 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Quelltext A.3: Struktur der MySQL Datenbank

Bibliography

- Baker, R. S. J. d. (2011). "Data Mining for Education." In *International Encyclopedia of Education*. 3rd (cit. on p. 10).
- Bienkowski, M. Feng, and M. Means (2012). *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: An Issue Brief*. URL: <http://www.cra.org/ccc/files/docs/learning-analytics-ed.pdf> (visited on 05/16/2015) (cit. on p. 7).
- Campbell, John P. and Diana G. Oblinger (2007). *Academic Analytics*. URL: <https://net.educause.edu/ir/library/pdf/PUB6101.pdf> (visited on 05/16/2015) (cit. on p. 9).
- Ebner, Martin et al. (2014). *Das O in MOOCs – über die Bedeutung freier Bildungsressourcen in frei zugänglichen Online-Kursen*. URL: http://link.springer.com/article/10.1365/s40702-014-0106-0?sa_campaign=email/event/articleAuthor/onlineFirst# (visited on 05/16/2015) (cit. on pp. 6, 11).
- Elias, Tanya (2011). *Learning Analytics: Definitions, Processes and Potential*. URL: <http://learninganalytics.net/LearningAnalyticsDefinitionsProcessesPotential.pdf> (visited on 05/16/2015) (cit. on p. 6).
- iMoox (2014). URL: <http://imoox.at/wbtmaster/startseite/about.html> (visited on 05/16/2015) (cit. on pp. 11, 13).
- Johnson, L. et al. (2011). *The Horizon Report*. URL: [http://www.nmc.org/sites/default/files/pubs/1316814265/2011-Horizon-Report\(2\).pdf](http://www.nmc.org/sites/default/files/pubs/1316814265/2011-Horizon-Report(2).pdf) (visited on 05/16/2015) (cit. on pp. 6, 7).

Bibliography

- Jordan, K. (May 16, 2015). *MOOC completion rates*. URL: <http://www.katyjordan.com/MOOCproject.html> (visited on 05/16/2015) (cit. on p. 1).
- Robes, Jochen (2012). *Massive Open Online Courses: Das Potenzial des offenen und vernetzten Lernens*. URL: http://www.weiterbildungsblog.de/wp-content/uploads/2012/06/massive_open_online_courses_robres.pdf (visited on 05/16/2015) (cit. on p. 1).
- Romero, C. R. and S. Ventura (2010). *Educational Data Mining: A Review of the State of the Art*. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews* 40 (6), pp. 601–628. URL: <http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5326> (visited on 05/16/2015) (cit. on p. 9).
- Rud, Olivia (2009). *Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy*. Wiley & Sons (cit. on p. 7).
- Schulmeister, Rolf (2013). *MOOCs – Massive Open Online Courses. Offene Bildung oder Geschäftsmodell?* URL: <http://www.waxmann.com/fileadmin/media/zusatztexte/2960Volltext.pdf> (cit. on p. 12).
- Tabaa, Yassine and Abdellatif Medouri (2013). *LASyM: A Learning Analytics System for MOOCs*. URL: http://thesai.org/Downloads/Volume4No5/Paper_16-LASyM_A_Learning_Analytics_System_for_MOOCs.pdf (visited on 05/16/2015) (cit. on p. 21).
- Web Analytics Definitions* (2008). URL: http://www.digitalanalyticsassociation.org/Files/PDF_standards/WebAnalyticsDefinitions.pdf (visited on 05/16/2015) (cit. on p. 8).