

Interactive Lecture Notes

A Computer-Aided Education System for Mastery Learning via Flipped Classrooms.

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering and Internet Computing

eingereicht von

B.Sc. Sven Sebastian Geiger

Matrikelnummer 1127054

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Dr. Florian Zuleger

Mitwirkung: Dr. Andreas Holzer, M.Sc.

Wien, 30. August 2018

Sven Sebastian Geiger

Florian Zuleger

Interactive Lecture Notes

A Computer-Aided Education System for Mastery Learning via Flipped Classrooms.

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

B.Sc. Sven Sebastian Geiger

Registration Number 1127054

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Florian Zuleger

Assistance: Dr. Andreas Holzer, M.Sc.

Vienna, 30th August, 2018

Sven Sebastian Geiger

Florian Zuleger

Erklärung zur Verfassung der Arbeit

B.Sc. Sven Sebastian Geiger
Einsiedlergasse 42/5a, 1050 Wien, Austria

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. August 2018

Sven Sebastian Geiger

Acknowledgements

I would like to thank Andreas Holzer and Florian Zuleger for providing a motivating and interesting topic, for their valuable input during reviews and for guiding me during this work.

I would like to thank my wife for supporting me during the writing of this thesis.

A special thanks goes to all students who evaluated my software and provided valuable feedback in the surveys.

This work is dedicated to my wife and my daughter.

Kurzfassung

In dieser Arbeit analysieren wir wie Studenten mit Hilfe von *Micro-Tests* und *dynamischen Modellen* direktes Feedback über ihren Lernfortschritt erhalten können. Zu Beginn geben wir eine Übersicht über unterschiedliche Terminologien und definieren häufig genutzte Begriffe, die für eine Diskussion heutiger E-Learning Systeme notwendig sind.

Wir vergleichen eine Reihe von bekannten E-Learning Systemen wie Khan Academy, Coursera, edX, Udacity und Moodle und untersuchen welche Methoden diese Systeme implementieren um direktes Feedback an Studenten zu liefern. Wir kommen zu dem Schluß, dass ein eigenes E-Learning System die beste Wahl zur Evaluierung von direkten Feedback mittels Micro-Tests und dynamischen Modellen darstellt.

Im Hauptteil der Arbeit beschreiben wir die Anforderungen eines solchen E-Learning Systems und stellen unsere eigene Implementierung vor. Unser *Interactive Lecture Notes* genanntes System implementiert ein interaktives Vorlesungsskriptum mit dynamischen Modellen und Micro-Tests und ermöglicht es Studenten direktes Feedback während des Lernprozesses zu erhalten.

Wir haben unsere Implementierung in der Praxis im Kontext einer Lehrveranstaltung in drei Semestern mit Umfragen und Benutzerstatistiken evaluiert. Die Ergebnisse liefern eine erste Bestätigung für unsere Konzepte und zeigen, dass Micro-Tests und dynamische Modelle effektive Mechanismen für direktes Feedback sind.

Abstract

The main goal of this thesis was to evaluate how *micro-tests* and *dynamic models* can provide instant feedback in e-learning systems. In the beginning of this thesis we present an overview of the terminology required for a thorough discussion of today's e-learning systems and platforms.

We then present a comparison of the most common e-learning platforms in use; including Khan Academy, Coursera, edX, Udacity and Moodle and we analyze which methods each of these platforms implements in order to support instant feedback. Based on this comparison we motivate our reasons for implementing our own e-learning system as the most suitable solution to evaluate micro-tests and dynamic models.

In the main part we present the requirements for such an e-learning system and introduce our own implementation. Our e-learning platform is called *Interactive Lecture Notes* and implements an interactive version of lecture notes that include dynamic models and micro-tests. It allows students to receive direct feedback during their learning process.

We conducted three evaluations of our implementation in the context of a lecture and collected feedback via surveys and statistical usage data. The results seem to confirm our hypothesis that micro-tests and dynamic models are effective tools to provide instant feedback.

Contents

Kurzfassung	ix
Abstract	xi
Contents	xiii
1 Introduction	1
1.1 Motivation	2
1.2 Preliminaries	3
1.2.1 E-Learning	3
1.2.2 Models for Teaching and Learning	4
1.2.3 Teaching and Learning Methodologies	6
1.2.4 Feedback Loops and Instant Feedback	11
1.2.5 Technical Aspects of E-Learning Systems	12
1.2.6 Traditional Methods in the Context of Mastery Learning	16
1.3 Aim of the Work	17
1.4 Structure of the Work	20
2 Related Work	23
2.1 Khan Academy	23
2.2 Coursera	28
2.3 EdX	31
2.4 Udacity	33
2.5 Moodle	36
2.6 Comparison	38
3 Interactive Lecture Notes	45
3.1 Requirements	45
3.2 Design Goals	46
3.3 Content Model	48
3.4 Application Roles	53
3.4.1 Student Role	54
3.4.2 Teacher Role	54
3.4.3 Admin Role	55

4	Software Architecture	57
4.1	Architectural Overview	57
4.2	Data Model	59
4.3	Backend Module	65
4.3.1	Persistence	65
4.3.2	Security	67
4.3.3	Endpoint Reference	68
4.4	Client Module	73
4.4.1	Frontend Application	74
4.4.2	Web Server and Reverse Proxy	75
5	Evaluation	77
5.1	Survey	78
5.1.1	Overall Experience	78
5.1.2	Detailed Feedback on E-Learning Tool	80
5.1.3	Lecture Context	84
5.2	Statistics	89
5.2.1	Course Progress	89
5.2.2	Feedback Loop Durations.	90
6	Conclusion	95
A	Build and Deployment	99
A.1	Build System Documentation	99
A.2	Deployment Documentation	102
B	Evaluation Results	107
	List of Figures	117
	List of Tables	118
	List of Algorithms	119
	Index	121
	Glossary	123
	Acronyms	125
	Bibliography	127

Introduction

The idea of using software to help students learn is as old as the invention of digital computers itself [BBL61, AB70, Gog13]. This idea has been continuously refined and advanced with each new technology that has become available [Kid09, Cro04]. Today the field of digital learning is commonly referred to as *e-learning* and various terms exist in this field, while with each new trend new terms are created.

In this thesis we apply a learning methodology called mastery learning [Blo68] that was first introduced by Benjamin S. Bloom in 1968 to the field of e-learning applications. In particular, we focus on the implementation of mastery learning strategies in e-learning software, with the goal to enable individual feedback and fast feedback times.

In this Chapter, we start with a discussion about the situation of e-learning in European universities and we motivate the need for interactive e-learning software with fast and individual feedback (see Section 1.1). In Section 1.2 we define the terminology that is most commonly used in discussions about e-learning. We then present the problem statement and our hypothesis in Section 1.3. Finally, we introduce the structure of this work in Section 1.4.

In Chapter 2, we introduce and compare several popular e-learning platforms such as Khan Academy, Coursera, edX and Udacity. We then introduce our own e-learning system, named *Interactive Lecture Notes* (see Chapter 3) and present the user experience considerations from an end user's point of view. A detailed technical discussion of our implementation is given in Chapter 4, where we present the various technical considerations of the design and architecture of our software as well as the data models and interface specifications of the different modules.

We have evaluated our software in the context of the lecture *Formal Methods in Computer Science* at the Technical University of Vienna in three different semesters and collected feedback from several students (see Chapter 5). We end this thesis with a conclusion and present some ideas for future work (see Chapter 6). Finally, in Appendix A, we explain

how to build and deploy our software and in Appendix B we include additional results from the evaluations.

1.1 Motivation

Public universities are often underfunded and resources for teaching are scarce [Tug16, LB06]. This situation leads to drastic suggestions like limiting the number of students to maintain an acceptable quality in teaching [APA, BK]. To maintain or even improve the current education quality without limiting access to education new education approaches must be found. In particular, new methods and tools that increase learning- and teaching-effectiveness as well as learning- and teaching-efficiency must be found. Such methods and tools would also lead to better cost-efficiency.

Situation in European Universities. We have observed, that the content provided to students in today's lectures is generally a combination of the following three artifacts, while not all of these may be available in each course:

- Slides that are used during the lectures.
- Screencasts that were recorded during the lecture and which are made available after each lecture.
- Lecture notes and scientific publications.

Digital content such as videos or even interactive teaching tools are not common to be found in today's lectures. This observation is also supported by the results of our evaluation (see Chapter 5, Questions 20 and 21). If video content is available at all in today's university courses, then mostly in the form of screencasts that were recorded during actual lectures. This is among the simplest methods of creating video content for a lecture. Watching those screencasts is time consuming, as each video usually extends through the full duration of a lecture (e.g. 90 minutes). In comparison, research has shown, that the optimal video length is around 6 minutes [Guo14, Guo13].

E-learning software, when available, is mostly used to manage course enrollment, support the download of additional learning materials or to support the submission of assignments. However, interactive software that supports the learning experience of students and which can provide fast and individual feedback about the learning progress is not in widespread use.

In addition to the provided course materials, students can usually receive support from teaching assistants and tutors. However, university departments often have limited resources for teaching assistants and tutors, and teachers have little time to create high quality teaching material. Additionally, the career of Professors depends to a large degree on the quality of published research papers and to a much lesser extent on the quality of

their teaching [Aga15, Lef16]. E-learning software, which provides fast and individual feedback to students and supports a cost-efficient way to create course content could make a significant contribution to improve this situation.

Note, that the above observations are based on the authors experience with state funded universities in Germany and Austria and may not apply for private universities or universities in other parts of the world.

1.2 Preliminaries

E-learning systems and platforms – in particular the so called *Massive Open Online Course* (MOOC) platforms – have been getting increasingly more attention in the recent years [Tho11, Ban15, Pap12]. They have started to complement the traditional teaching model, through forms of *blended learning* or the *flipped classroom* approach [Tuc12, BV13, GQJ⁺13, MRG⁺14]. We also see an increasing trend to use *mastery learning* rather than *scheduled learning*. In this section we will properly introduce each of these terms and discuss their impact on today's learning landscape.

1.2.1 E-Learning

Before the wide-spread availability of affordable and fast Internet access there had already been a wide range of digital learning software that was usually implemented as desktop software (distributed on CD-ROM) and did not have any communication features [Kid09, Cro04]. The rise of the Internet and the World Wide Web has caused a migration away from desktop software towards web applications and brought an increasing focus on communication features as a central aspect of the learning process [Kid09]. The term *e-learning* originates from the late 1990s. It has been defined several times by different authors and in various contexts and there is currently “no single agreed definition of eLearning” [Kid09]. As Garrison and Kanuka [Gar11] write “the term e-learning came into use in the mid-1990s along with developments in the World Wide Web and interest in asynchronous discussion groups”. Garrison and Kanuka formally describe e-learning as “electronically mediated asynchronous and synchronous communication for the purpose of constructing and confirming knowledge”. This rather constructivist definition focuses mostly on the communication aspect of digital learning.

Jay Cross described e-learning as “learning on Internet Time, the convergence of learning and networks. [...]” [Cro04, KA15]. And Cisco defines it as “eLearning is Internet-enabled learning. [...]” [Cro04, KA15]. For the purpose of this thesis we will use the following two definitions:

Definition 1 (Educational Technology). “*Educational technology is the study and ethical practice of facilitating learning and improving performance by creating, using, and managing appropriate technological processes and resources*” [RMR08].

Definition 2 (E-Learning Software / Platform). *An **e-learning software** is a software that provides specific implementations of educational technologies and learning methodologies to create a digital learning environment.*

*An **e-learning platform** is an e-learning software which provides a Service Provider Interface (SPI) or an Application Programming Interface (API), that can be used by third parties to extend the software with their own extensions, or which allows third parties to offer their own digital content through the software.*

In this thesis we are only interested in interactive web-based e-learning software/platforms. With these rather general definitions in place we will now look at more specialized terms in the context of e-learning.

1.2.2 Models for Teaching and Learning

In the following discussion about e-learning we usually want to compare the terms that we introduce with the model of teaching as it is still (to a large degree) conducted in universities and schools. The current form of teaching in schools and universities is what we call *traditional teaching* or depending on the context it is referred to as the *traditional classroom*, a *traditional lecture* or *traditional education* [BS13, OGI12, Min14]. This form is a teacher centered method that (among others) structures a course into lectures, includes the use of lecture notes or books and that has graded assignments and exams at fixed dates which students must pass.

Time Aspects of Teaching and Learning. An important aspect for the discussion of e-learning software is the timing between teaching and learning and the ability to scale the teaching process to many students.

When we look at the traditional teaching model, we see that it does not scale very well with an increasing audience size. The degree of interactivity is inversely proportional to the size of the audience. That is, the larger the audience the less interactive can the teaching be, because the teacher has to divide his or her attention between more students and has less time to answer individual questions. Thus, traditional teaching is most interactive, when it is either face-to-face or in small groups. However, modern e-learning software has the potential to provide students with personalized content and interactive feedback that meets the particular needs of each individual student even in a mass teaching context.

In traditional teaching the teacher decides what to learn and when to learn it. Whereas with e-learning the students are in control of what they learn, when they learn it and how long they learn [Kha13b]. This leads us to the following definition:

Definition 3 (Synchronous and Asynchronous Learning). *A form of learning where all students learn the same content at the same time is called **synchronous learning**. If each individual student can choose the content and the time to learn this content*

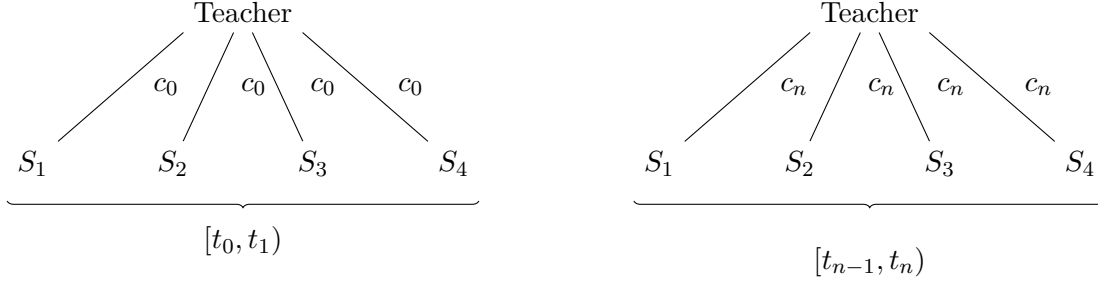


Figure 1.1: Example of Synchronous Learning: During each time interval $[t_a, t_b)$ a teacher is teaching the same concept c_i to all students S_i .

*independently, we call this form **asynchronous learning**. **Synchronous teaching** and **asynchronous teaching** are defined analogously from a teachers point of view.*

Let us look at an example for both variants. In the first example (see Figure 1.1) we see a traditional lecture, which takes place during the time interval $[t_0, t_n]$ and has a fixed content, which all students S_i will learn during that lecture. In the first part of that lecture (e.g. during $[t_0, t_1)$) all students learn concept c_0 , then the teacher moves on to the next concept in the next interval. This continues until the end of the lecture (e.g. interval $[t_{n-1}, t_n)$), where concept c_n is taught.

With an e-learning software, the content that is taught and the time when it is taught become decoupled, since each student that is using the software can choose the time and content individually. In the second example (see Figure 1.2) we no longer reason about time intervals but instead look at different moments in time, since at any particular moment each student might independently decide to switch to another concept. As we can see in the second example at the time t_0 the students S_1 and S_3 are learning concept c_1 from video V_1 , whereas student S_2 and S_3 already learn more advanced concepts. At a later time (e.g. t_n), student S_1 might still be reviewing concept c_1 , whereas student S_3 has already advanced to concept c_5 .

In the above example we have chosen video as a common example of e-learning content; additional types of e-learning content including interactive content types will be presented through the course of this thesis.

E-learning systems generally support a form of asynchronous learning, where as traditional teaching uses a form of synchronous learning. A notable exception are e-learning systems that use live conferencing technology to synchronously connect teachers and students. Later in this Chapter we will also encounter mixed forms of synchronous and asynchronous learning, when we discuss blended learning and the flipped classroom. The use of asynchronous learning gives e-learning systems a significant advantage over synchronous forms of learning and, in this thesis we are therefore not interested in systems that only support synchronous forms of learning.

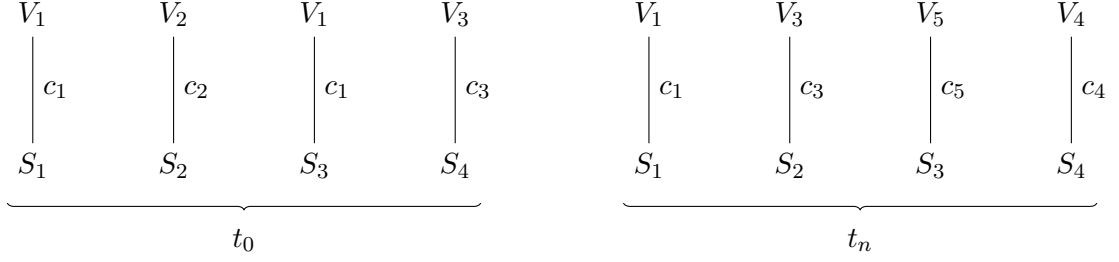


Figure 1.2: Example for Asynchronous Learning: At each time t_i the students S_j can learn concepts c_k independently from each other by watching the corresponding video V_k .

1.2.3 Teaching and Learning Methodologies

Scheduled Learning. In this paragraph we will discuss different methods of learning by analyzing the learning process over a longer period of time. In the traditional education model learning has always been *scheduled* in advance. A student has to attend a course in which lectures are laid out with a fixed schedule and the student is expected to learn new concepts for a fixed amount of time until the course advances to the next concept. At the end of that schedule an exam is conducted that determines the degree of mastery that the student has achieved in the set of concepts that were taught during the course. Obviously the amount of time to learn a set of concepts is always fixed with this method and the variable aspect is the degree of mastery that the student can achieve [Kha13b].

Definition 4 (Scheduled Learning). *In **scheduled learning** a schedule of lectures is defined, in which lectures are given with a fixed cadence and with fixed content. At the end (and possibly at intermediate points) of that schedule an examination is conducted to identify the varying degrees of understanding that each student has gained of the content.*

Mastery Learning. A method that was first proposed by Benjamin Bloom in 1968 in his thesis “Learning for mastery” [Blo68] is called *mastery learning*. It has recently gained much popularity due to its use on the popular Khan Academy platform. The fundamental assumption in mastery learning is that more then 95% of the students are generally able to achieve mastery in a complex topic but that the time required to achieve this mastery may vary between students depending on certain factors such as [Blo68]:

- Aptitude for particular kinds of learning.
- Quality of instruction.
- Ability to understand instruction.
- Perseverance.
- Time allowed for learning.

The important aspects of mastery learning are, that “[e]ach strategy must find some way of dealing with individual differences in learners through some means of relating the instruction to the needs and characteristics of the learners” [Blo68].

Bloom defines several important preconditions for the application of mastery learning as a learning method. In particular these are the “specification of the objectives and content of instruction” [Blo68] as well as the “translation of the specifications into evaluation procedures” [Blo68]. These evaluation procedures can then be used to measure the progress of the student and to give students the necessary advice if they have not yet mastered the learning objective.

The important aspect in Bloom’s definition of mastery learning is that the “teaching-learning process and the evaluation process [...] are *separate* processes” [Blo68]. This means that during the learning phase both the student and the teacher must be able to measure the progress towards the learning objective. As Bloom writes: “Both the teacher and the learner must have some understanding of what the achievement criteria are and both must be able to secure evidence of progress toward these criteria” [Blo68]. This process of intermediately conducted evaluations has been defined by Bloom as “formative evaluation” and is based on the distinction of formative and summative evaluation (see [Scr67]). For this discussion we will use the following two definitions which are based on Bloom and Scriven [Scr67, Blo68]:

Definition 5 (Formative Evaluation). *Formative evaluation is conducted at intermediate points of time during the learning process to assess the progress of the student. The results of the evaluation should not count towards the students final grade but serve as a diagnosis to determine whether the student has mastered the learning objectives and which aspects – if any – still require further study.*

In particular, the diagnoses attained through formative evaluation should be used to formulate very specific prescriptions that can help the student to master the concepts that were not well understood [Blo68]. In contrast to formative evaluation which is part of the teaching-learning process, there is also a need for graded evaluation that is conducted at the end of a period. This is called *summative evaluation*:

Definition 6 (Summative Evaluation). *Summative evaluation is a graded evaluation that is conducted at the end of a teaching period such as a semester or term. The goal is to perform a final assessment of the students that can serve as evidence for the degree of mastery that the students have achieved during the course.*

Based on Blooms definitions [Blo68] we will use the following definition of mastery learning in this thesis:

Definition 7 (Mastery Learning). *Mastery learning assumes that the majority of students is able to achieve mastery of the content that is being studied. However, different students require different amounts of time to achieve mastery. The goal of mastery*

learning is to minimize the amount of time that each student requires to achieve mastery, by emphasizing formative evaluation. In particular both student and teacher must be able to frequently evaluate their progress and clear instructions must be offered to the student if problems are revealed during that evaluation. Finally, there must be a clear separation between formative and summative evaluations.

Different strategies to implement mastery learning can be created. But the constant is always the degree of understanding that a student must achieve and the variable parts are the time when a student learns the concept and the amount of time that the learning process requires [Kha13b]. The focus of mastery learning is on formative evaluation and the definition emphasizes a clear separation of formative and summative evaluation. This gives rise to the question of whether and when a summative evaluation should be conducted. While the majority of students should be able to achieve mastery, there may still be students that do not achieve mastery. Reasons may include that students lack the aptitude for learning the subject or that the time required to master the subject still exceeds the available time of a student. A mastery learning strategy can improve the learning experience and reduce the time required to master a subject but it cannot guarantee success. Therefore, even in the context of mastery learning a summative evaluation may be conducted at the end of a term, in order to determine which students have mastered the subject.

The definitions for scheduled learning and mastery learning are to some degree connected via the definitions of synchronous and asynchronous learning. Both synchronous and asynchronous learning can be used in a scheduled learning environment; the students would learn for a certain amount of time either synchronously or asynchronously, and then afterwards a summative evaluation (e.g an exam) is conducted. Mastery learning on the other hand is most suited for asynchronous learning, because in a synchronous learning environment either the slowest student would define the speed of a course or the slowest students might not achieve mastery.

When Bloom wrote his definition of *mastery learning* in 1968 it was not yet common to use computers in classrooms, and so Blooms strategy to implement mastery learning does largely focus on suggestions for the teacher. The three most relevant suggestions are:

- The segmentation of a course into short units (Bloom mentions one or two weeks per block [Blo68]).
- The use of formative evaluation at the end of each unit.
- Giving students specific prescriptions that allow them to review concepts which they have insufficiently understood.

For each of the above suggestions partial solutions exist, but they are not fully automated and involve manual work. For example, e-learning software can facilitate the creation of

shorter units than the one or two week units suggested by Bloom, through the support of fine grained content models. Formative evaluation can be automated, but requires support for the respective exercise types in the e-learning software. However, in both cases the creation of the actual course and exercise content is still a manual task. Offering students specific prescriptions based on the results of formative evaluation is still a hard problem to solve. A possible solution is to manually create a set of predefined prescription candidates for each exercise and then use a heuristic to select one of these candidates based on the answers of a student. In Chapter 2, we include a further discussion of these aspects, when we introduce and compare various e-learning platforms.

We believe that improving the above mentioned aspects is where e-learning software currently offers the most potential. There is however one aspect that we must be aware of: The integration of formative evaluation into e-learning software and the possibility for feedback on a scale of minutes rather than days or weeks, allows us to make learning units arbitrarily small. But if we conduct an evaluation after each atomic concept that is taught, then we create a situation where we can only test if a student has understood a concept, not if he has sufficiently memorized and practiced the concept to ensure long term retention. A successful e-learning implementation will therefore have to find a way of dealing not only with testing for understanding but also for retention.

One way to increase retention is to repeatedly practice a certain concept in different contexts. This helps to improve the understanding of the concept, lets students quickly discover problems in their understanding, and helps the students to apply the concept faster and with more confidence. If an e-learning software applies mastery learning as the central method of learning, it is therefore essential that it includes not only exercises as formative evaluation, but also gives students the ability to repeatedly practice a certain concept; especially if formative evaluation shows that the student has not yet mastered the concept.

Blended Learning. There are generally two ways to use e-learning software in a teaching process. The first one is the use of e-learning software as a replacement for traditional teaching, such that the learning process happens only through the e-learning platform. MOOCs, which we will discuss later in this section, are an example of this form. The second one is a form where the e-learning software is integrated into the traditional lecture in order to support the teaching process. This form is called *blended learning*.

In a 2004 study by Garrison and Kanuka who have analyzed the potential of blended learning, the following definition for blended learning was given:

Definition 8 (Blended Learning). *Blended learning is the thoughtful integration of classroom face-to-face learning experiences with online learning experiences [GK04].*

However, as the edX¹ course on blended learning explains, “Blended learning is still a developing field and has no specific, agreed upon definition” [eI16]. This is consistent

¹See Section 2.3 for an introduction of the edX platform.

with the above definition, which does not exactly state how a “thoughtful integration” can be achieved.

One aspect that is emphasized by Garrison and Kanuka, is that an e-learning system is not just another offering that exists independent to the lecture, but instead is an essential part of the lecture. Another aspect described by Garrison and Kanuka is the asynchronous nature of communication in an e-learning platform, which allows students to communicate and exchange ideas more freely than they would otherwise do during the lecture. In a lecture there is limited time available for questions or to present own ideas, or students may be afraid to ask questions if they feel that they could embarrass themselves [GK04]. By using the communication tools of an e-learning platform (usually some kind of forum system), students can “confront questionable ideas and faulty thinking in more objective and reflective ways” [GK04].

Garrison and Kanuka emphasize the communication aspect of blended learning, but there is also the aspect of asynchronous learning that is important. It enables teachers to be less involved in the actual teaching of content and instead allows them to engage in more interactive forms of learning during the lecture.

Flipped Classroom. A special form of blended learning is the concept of the *flipped classroom* [Kne11, Edu]. At the core of this concept is a central change to the way teaching is conducted. On the one hand, activities which can be automated are moved out of the classroom and into an e-learning software. On the other hand, tasks which require more interaction with the teacher or other students are moved into the classroom. Like *blended learning* there still exists “a lack of consensus on what exactly the flipped classroom is” [LPT00]. Thus most definitions only describe the intended results of flipping a classroom but fail to explain how this can actually be achieved or what criteria can be used to assess the quality of a flipped classroom. For example Lage et. al. simply describe the flipped classroom in the following way: “Inverting the classroom means that events that have traditionally taken place inside the classroom now take place outside classroom and vice-versa” [LPT00].

In a flipped classroom new content is not learned from the teacher in a synchronous learning environment, but is rather learned from an e-learning software in an asynchronous learning environment. And instead of learning these new concepts while in class and then doing exercises as homework, the students are asked to learn new concepts from the e-learning software (e.g. by watching videos) outside of the class. The time during the lecture can then be used for exercises, group work or discussions with the teacher and as a result the lecture becomes much more interactive [LPT00]. This also means that there is more time for students to ask questions and discuss problems that arise from newly learned concepts.

As with other terms different definitions exist. A more formal definition that we will use in this thesis was given by Bishop and Verleger:

Definition 9 (Flipped Classroom). “*The flipped classroom [is] an educational technique that consists of two parts: interactive group learning activities inside the classroom, and direct computer-based individual instruction outside the classroom.*” [BV13]

Bishop and Verleger explicitly restrict this definition “to exclude designs that do not employ videos as an outside of the classroom activity” [BV13]. However in the context of higher education a limitation on video content may be too restrictive. For the purpose of this thesis we will thus use the above definition without this restriction. In particular, we want to evaluate designs which do not use videos as the main form of content delivery. Instead we aim to create a learning experience which includes a large amount of interactive content types that are directly integrated into the content. We will further elaborate on this decision at the end of this Chapter when we present the aim of this work (see Section 1.3) and in Chapter 3, when we introduce the design aspects of our own e-learning implementation.

Since in a flipped classroom the e-learning software is an integral part of the teaching process, and using the e-learning software is part of the students homework, this gives teachers a large amount of statistical data that teachers can use to plan activities in the classroom. For example teachers can monitor their students progress as the students learn and solve exercises on the e-learning platform. This gives teachers the ability to identify students who are struggling to understand a specific concept and specifically help those students [Kha11]. Or even better, they could identify students which have already understood that concept and pair them with those struggling students during group work to let the good students help them [Kha11].

1.2.4 Feedback Loops and Instant Feedback

In the context of formative evaluation, we are often interested in the duration that passes between the time when students learn a concept, and the time when the students receive feedback which shows them if, and how well, they have understood this concept. We refer to this process as a *feedback loop* and to the duration as a *feedback loop duration* and use the following formal definition:

Definition 10 (Feedback Loop). *A **feedback loop** consists of three steps:*

1. *A student learns a concept.*
2. *A student engages in formative evaluation for that concept.*
3. *A student receives feedback about the degree of understanding for that particular concept.*

*Students can learn multiple concepts before they engage in formative evaluation. However, the **feedback loop duration** refers to duration of this process with respect to one particular concept.*

Depending on the teaching methodology that is used the feedback loop duration may be on a scale of minutes, days or weeks. In traditional education the duration of such feedback loops is usually on a scale of days and weeks, where as with modern e-learning software much shorter feedback loop durations are possible. A further reduction of feedback loop durations is a goal of this thesis.

Confirmation Bias. Confirmation bias often happens, when we have already formed a hypothesis or gained a partial understanding of how a certain aspect in our world works. We then often tend to look for facts that confirm our hypothesis and subconsciously ignore other facts that might contradict our original hypothesis [Plo93, Nic98]. For example, when looking at a formula for a mathematical model, we might have a (wrong) intuition about what this formula represents. We then look for clues in the text that let us confirm our intuition and might overlook or ignore other information that contradicts our intuition. Shorter feedback loop durations should help to reduce confirmation bias.

1.2.5 Technical Aspects of E-Learning Systems

In this Section we discuss technical aspects that are relevant for the implementation of e-learning software. We introduce the terms *micro-tests* and *dynamic models*, which describe methods to provide fast individual feedback. We also discuss the terms Massive Open Online Course (MOOC), which describes a class of of massively scalable e-learning courses and *gamification*, which is used to increase user engagement [Ruh] and which is not limited to e-learning.

Micro-tests. *Micro-tests* are a method to implement formative evaluation with very short feedback loop durations by using exercises and questions that are directly embedded into a stream of content. Embedding micro-tests directly into the content distinguishes them from e-learning tools where content and exercises are separated into distinct sections, one for learning and one for practicing, and therefore require a switch of context to navigate between each section. By eliminating this context switch micro-tests allow shorter feedback loop durations and avoid potential distractions during the context switch.

Definition 11 (Micro-tests). *A **micro-test** is a small exercise or question that is directly embedded into a content stream.*

Micro-tests are only used for formative evaluation and are therefore not graded.

Micro-tests must provide immediate feedback about the correctness of an answer. Micro-tests must offer detailed explanations when the answer was incorrect and should also give a short explanation when the answer was correct.

Dynamic Models. With *dynamic models* students can interactively explore a concept in order to evaluate the concept and to test their understanding of particular aspects of the concept. Interactions with a dynamic model include the modification of its state such

as by adding or removing items, changing parts of the model via clicking or drag-and-drop, or through the activation of its functions. The dynamic model reacts to these changes and students can directly observe the effect of a change in the models representation.

Definition 12 (Dynamic Models). *A **dynamic model** is an interactive graphical or textual representation of a concept that changes its representation in response to actions from a user.*

A dynamic model should be limited in scope and only include the functionality that is necessary to demonstrate one particular aspect of a concept.

Dynamic models are particularly suited for interactive mathematical or theoretical models or executable pseudo-code algorithms. For example, an executable pseudo-code algorithm could help students understand the behavior of algorithms by visualizing the execution of individual steps similar to a debugger.

The second part of our definition for dynamic models is included to emphasize that it is better to include multiple dynamic models each with little complexity, rather than one big interactive tool that can be used to evaluate all aspects of a concept. For example, a dynamic model that shows the difference between cyclic and acyclic graphs does not need to support the full complexity of graph theory. Instead providing the ability to add or remove a few edges at particular positions in a graph would be enough to demonstrate the presence of cycles in a graph.

Massive Open Online Course. A term that is widely used in today’s discussions about e-learning platforms is the MOOC. The term is used to describe a kind of higher education course format that can be accessed online by anyone with an Internet connection. It provides the complete content of a *course* and makes it available to a *massive* amount of people – compared to the limited amount of students in a university lecture. The term *open* has different definitions, it may refer to the aspect of content being under an open license, but also to the aspect that registration for a course is open for any user or even to the aspect that the course is available free of charge. Another aspect of openness is the use of open standards. A MOOC is designed to reach a large amount of people and this is best achieved if there are no access limitations with respect to the dependence on a certain platform, technology, device type or application. Therefore the term open can also refer to the fact, that a MOOC is accessible through a web browser, rather than a particular desktop or mobile application. Not all definitions of *open* may apply to a given MOOC or the platform on which it is hosted.

An important aspect of MOOCs is that they try to replicate the format of a traditional university course in an online environment [Kha13a]. Just like a traditional university course a MOOC usually has a fixed start and end date. Students must enroll in the course before they can access the course material (although there are usually no limitations on who may enroll) and the course contains assignments and labs – with due dates – that must be completed by the students. The due dates of assignments ensure that students

generally follow a similar pace through the course – just like in the case of a traditional lecture.

The term MOOC was first coined by David Cormier in 2008 [Par13, SSH15] but varying definitions can be found. For the remainder of this discussion we will use the following definition:

Definition 13 (Massive Open Online Course). *A MOOC is a course format that is highly scalable to a large number of students, accessible under an open license to anyone independently of geographic location or time constraints and available online through the use of open standards.*

However the term MOOC has since been described as a misnomer by David Wiley, because “Almost every so-called MOOC violates at least one letter in the acronym.” [Wil12]. Wiley explains that: “Many MOOCs are massive but not open, [...] open but not massive, [or] try very hard not to be courses”. In fact some MOOCs may have partially offline content and thus violate the “online” part of the acronym. An example for this are the so called “blended MOOCs” as introduced by Yousef et al. [YCSW15].

The introduction of platforms such as Coursera, EdX, Udacity and others, has caused the term MOOC to receive widespread attention from students as well as newspapers around the world [Pap12, Swo14]. Therefore it is important to clarify that these websites are not MOOCs – but platforms – and the offered courses are what is commonly called a MOOC. Also not everything that is offered on e-learning platforms can be described by the term MOOC. In the remainder of this thesis, when we discuss different e-learning platforms, we will also analyze if the content that is served on these platforms can be categorized as a MOOC and which parts – if any – of the acronym are violated by the MOOC.

MOOC platforms offer the advantage that students who use these platforms no longer have to be physically present in the geographic region where a lecture is taking place. Instead, the digital nature of the lecture content allows students to consume it from any location in the world – as long as the students have Internet access. Where students previously had to attend a lecture at the time it was given, they can now view the recorded lecture at any time convenient, with the additional benefit that they can pause, rewind and repeat the recording of the lecture if necessary [Kha11].

MOOCs have the ability to reach large numbers of students all over the world (e.g. 155.000 students in the first edX MOOC [Aga13]), but most MOOCs also have very high dropout rates. Research by Katy Jordan shows that while the average size of MOOCs – in terms of the number of students – is around 43.000 students per course, only 6.5% of those students actually complete the course [Jor14]. It should be noted though, that there is a high variance in this data depending on factors such as course, topic and platform that offers a MOOC [Jor15]. However, even with such high dropout rates, due to the high number of students that enroll for MOOCs the number of students that pass such a course can still be significant. There were 7,200 students who passed the first edX MOOC [Aga13]. On the one hand this means that only 4.6% passed the course, but as

Anant Agarwal explains: “If I were to teach at MIT two semesters every year, I would have to teach for 40 years before I could teach this many students” [Aga13].

Technical Aspects of MOOCs. A MOOC usually consists of video content, text content, as well as interactive content such as exercises and quizzes that can be solved online, and which are automatically checked by the software to evaluate the students understanding of the content. The interactive content and the ability to automatically check the answers of students form a particularly important aspect of MOOCs. The interactive content allows the students to evaluate themselves and to check their understanding of the content, and the ability to automatically check the students answers are an important factor for scalability, since manual exercise evaluation is not feasible with thousands of students.

When we consider the above aspects from a technical point of view, the question if a software is suitable as a MOOC platform mainly depends on two factors. The first one is scalability of the content distribution itself, in particular the ability to support thousands of concurrently active users using the software and learning new content. The second is scalability of the evaluation methods used (including both formative and summative evaluation). This refers to the ability to provide automatic evaluation of exercises and assignments.

Gamification. Gamification is another term that can often be found in discussions about learning, and it is being used by several e-learning platforms such as Khan Academy, Coursera or EdX. Different definitions for gamification can be found, but for the purpose of this thesis we will use the following definition by Deterding et al. [DDKN11]:

Definition 14 (Gamification). *Gamification is “the use of game design elements in non-game contexts” [DDKN11].*

The authors define gamification with an emphasis on the word “game”, which they define as “playing structured by rules and competitive strife toward goals” [DDKN11] and explicitly contrast it to the word “*play* (or playfulness)” [DDKN11], which they define as “a more free-form, expressive, improvisational, even “tumultuous” recombination of behaviors and meanings” [DDKN11].

Another definition refers to gamification as the “integration of game mechanics into [applications] to motivate participation, engagement and loyalty” [Bun16]. Gamification is not specific to the domain of education and learning and can also be found in other fields such as work [Ruh] or health [LWC⁺14].

The basic idea of gamification is to introduce virtual goods into an application that can be awarded to the users for engaging in various activities during the use of the application. The virtual goods that are awarded to users are different for each application and can include points, virtual currencies, badges, avatars, reputation, or access to particular features that are otherwise unavailable.

A practical example for the use of gamification can be found on the *Stack Exchange* websites. Stack Exchange² is a network of question and answer websites, each of these websites rewards its users with *reputation* points [Excb] for adding and moderating content. Gaining reputation has a direct effect for users, since it results in more privileges (e.g. access to more features) such as the ability to post comments and it leads to a reduction of advertisement [Exca].

We are not using gamification in our own e-learning implementation, but we will discuss gamification in specific e-learning systems as part of the related work.

1.2.6 Traditional Methods in the Context of Mastery Learning

In traditional learning different methods exist to support the learning processes of students, to evaluate learning progress and to provide students with feedback about their learning progress. Common methods include exercise sheets, tutoring sessions and study groups. These methods are not specific to a particular learning methodology such as scheduled learning or mastery learning. However, differences in the application of each method can exist depending on their use in the context of different learning methodologies.

Exercise sheets can be used as a means for both formative and summative evaluation of students but also to help them practice a certain topic. However, the concept of formative evaluation as suggested in mastery learning goes beyond the use of simple exercise sheets. In particular, when students are unable to solve an exercise they should be able to obtain detailed individual feedback that helps them to identify problems and to show them a path towards mastering the exercises.

Sample solutions provided with exercise sheets provide a rather limited solution to this problem. On the one hand, sample solutions only offer a generic solution to a problem and therefore lack the aspect of individual feedback. On the other hand, with respect to the duration of feedback loops the time when sample solutions become available must be considered. Especially in scheduled learning environments, sample solutions may only become available after an exercise sheet is due for submission which can lead to feedback loop durations of several days or weeks.

Tutoring sessions are another means to provide individual feedback to students. The advantage of tutoring sessions is that they support individual feedback paired with potentially short feedback loops. However, tutoring sessions suffer from the lack of scalability. With increasing group sizes the duration of feedback loops increases and the chances of receiving individual feedback decrease as tutors may no longer have enough time for the needs of each student. On the other hand the cost for tutoring sessions is directly proportional to the group size and therefore tutoring sessions can be a significant cost factor for educational institutions.

²www.stackexchange.com

Study groups can provide a cheaper alternative to professional tutoring sessions. However, the effectiveness of study groups can vary significantly and may depend on factors such as the previous knowledge of students and the students' ability to understand the available instructions. Bloom writes that study groups are most efficient, if students meet in study groups after a formative evaluation has been conducted, such that students can “[...] review the results of their formative evaluation tests and to help each other overcome the difficulties identified on these tests” [Blo68].

1.3 Aim of the Work

An important recommendation of mastery learning is the frequent use of *formative evaluation* (see Definition 5). While Bloom suggests to perform these evaluations approximately once in two weeks [Blo68], the use of e-learning software makes it possible to achieve much shorter intervals. Ideally, *feedback loops* could be achieved that are on a scale of minutes, rather than hours, days or even weeks. However, Bloom's suggestions have neither been implemented widely in traditional education nor in modern e-learning software.

A popular exception is Khan Academy [Kha, Tho11], which has implemented many of Bloom's suggestions for mastery learning. On Khan Academy, new skills are learned by watching videos with a duration of 10 to 15 minutes. Feedback is then collected through exercises that are automatically evaluated. A simple feedback loop that consists of one video and a set of exercises takes on average 30 minutes to complete. Other popular e-learning platforms such as Coursera [Coua], edX [edX] or Udacity [Udab] have feedback loops with a similar or even longer duration. This is a significant improvement compared to the two-week intervals suggested by Bloom but not yet on a scale of minutes.

Drawbacks of Video-Based Content. With video-based content the duration of a feedback loop has a lower bound, which depends on the duration of the corresponding video. To create shorter feedback loops, for example with a duration of one minute it is necessary to limit the content duration to about 30 seconds. This requires that a video stream is either paused intermittently or split into individual segments. Both is unpractical and difficult to achieve with video-based content.

A video is an atomic unit, that has a tight coupling between the video and audio stream. In order to make changes to a video the audio-visual context must first be recreated before a new fragment can be recorded and edited into the original video. Often this can only be done by the original author of the video. As a consequence, videos are usually the result of a single author or company where as content with text-based sources can be created and updated by many different individuals. The implication is, that it is considerably harder to make incremental changes to a video or to reuse video content to create a new video, than with content that has text-based sources.

Video-based content can also be expensive to create (costs can be as high as “\$4,300 per hour of finished video” [HT14]). Furthermore, videos have high bandwidth requirements. While this is not an issue in most larger cities, it can be problematic in rural areas and

countries with less developed Internet infrastructure. For example for viewing HD and Full-HD video a bandwidth of 5 to 8 Mb/s is recommended [Net]. Automated search through video content is not possible unless appropriate meta-data has been created for the video. Manual search through a video is a time consuming task and generally requires $O(\text{video length})$ time.

Due to these drawbacks of videos, alternative content types that are more suitable to achieve instant feedback need to be evaluated.

Problem Statement. How to realize instant feedback, i.e. feedback loops on a scale of minutes in modern e-learning software is still an open question. Which content models are best to support feedback loops durations on a scale of minutes is also an open problem, in particular due to the drawbacks of video content discussed above.

Hypothesis. Our hypothesis is that *micro-tests* (see Definition 11) and *dynamic models* (see Definition 12) are an effective means to provide *instant feedback* in e-learning systems. *Micro-tests* are small exercises or questions that are integrated into the content to evaluate how well a student has understood a subject and to give *instant feedback* with the result of that evaluation to the student. With *dynamic models* students can interactively explore a concept in order to evaluate and test their understanding of the concept. Examples of dynamic models are interactive content and executable pseudo-code algorithms. The integration of micro-tests and dynamic models into the lecture content enables an individualized content stream that meets each student's specific needs. In such a content stream, content can be loaded dynamically based on evaluations via micro-tests or a student's interaction with a dynamic model and thus allows students to receive instant feedback.

Goals of this Thesis. We will investigate if implementing a mastery learning strategy that uses micro-tests and dynamic models to support individual feedback on a scale of minutes is technically feasible. As part of our mastery learning strategy we will implement an individualized content stream that uses micro-tests and dynamic models. For our implementation we will compare current e-learning software to decide if we can extend an existing e-learning system or build our own e-learning implementation.

We plan to evaluate our solution to test our hypothesis and to see if it can be an efficient alternative to existing video-based e-learning platforms.

Evaluation. We will evaluate the implementation of our mastery learning strategy, which uses *micro-tests* and *dynamic models*, in the context of the lecture *Formal Methods in Computer Science*. We have selected this lecture for several reasons. Formal Methods in Computer Science is a mandatory lecture for several computer science majors at the Technical University of Vienna and has therefore many students attending the course. This makes the lecture a good candidate to evaluate our results. The lecture has a strong focus on mathematical and logical concepts which provides a good basis to create

exercises and questions that have well-defined answers and which can be automatically evaluated. This makes it suitable for automatic formative evaluation such as through our micro-tests.

Finally, Formal Methods in Computer Science is considered a challenging subject by many students and therefore students would benefit from the additional support that an e-learning tool can provide.

Survey. Our evaluation will be based on a survey with we offer to students who attended the lecture and also used our online course. The survey will be conducted anonymously using a questionnaire on Google Forms [Goo18]. The questionnaire will contain a total of 32 questions and is structured into following three categories:

- **Overall Experience.** In this category we ask general questions about our e-learning tool and our design decision to prefer textual content over video.
- **Detailed Feedback on the E-Learning Tool.** This category is spit into four subcategories which ask more detailed questions about the e-learning implementation:
 - **Presentation of Content.** In this subcategory we collect feedback about our content model and the incremental presentation of content.
 - **Interactivity.** In this subcategory we ask questions about our content model and different aspects of our interactive content types such as exercises and interactive elements.
 - **General Feedback.** In this subcategory we give students the opportunity to provide feedback about aspects which they found helpful and which did not find helpful on our platform.
 - **Platform.** In this subcategory we ask questions about the type of device on which students used our implementation and how they perceived the speed of the tool.
- **Lecture Context.** In this third category we ask questions about the offline lecture such as lecture attendance or whether students had the necessary mathematical background before they attended the lecture. We also ask how often students have encountered interactive tools and video content at the Technical University of Vienna.

Statistics. We will support the results of our survey with statistical data from the e-learning implementation such as course progress of students and the average duration which students spend on exercises and content.

1.4 Structure of the Work

In the following paragraphs we describe the structure of our work.

Related Work. In Chapter 2 we evaluate and compare the e-learning platforms Khan Academy, Coursera, edX, Udacity, and Moodle (see Sections 2.1 through 2.5). Our comparison of these e-learning platforms (see Section 2.6) is focused on three aspects:

- We compare the teaching and learning methodologies of each platform such as mastery learning, blended learning and flipped classrooms and discuss to what degree each platform has implemented these methodologies.
- We discuss whether these platforms are MOOC platforms and which part of the MOOC acronym applies for each of these platforms.
- Finally, we present to which degree each of these platforms supports instant feedback and we discuss the possibilities of using each platform for our evaluation of instant feedback via micro-tests and dynamic models. We also explain our decision to evaluate our hypothesis about instant feedback by building our own e-learning implementation instead of reusing one of the evaluated platforms.

Interactive Lecture Notes. In Chapter 3 we present our own e-learning implementation called “Interactive Lecture Notes”. The e-learning implementation features a dynamic content model and is specifically designed to support mathematical and scientific content. Our dynamic content model is able to interleave a variety of interactive and non-interactive content types that include textual content (e.g. text or mathematical formulas) but also micro-tests and dynamic models.

We start with an introduction of functional and non-functional requirements (see Section 3.1) as well as user interface design goals (see Section 3.2). We finish the Chapter with a presentation of our content model (see Section 3.3) as well as an introduction to the different application roles such as students, teachers and administrators (see Section 3.4).

Implementation. In Chapter 4 we present the architectural design of the individual modules that are part of our implementation (see Section 4.1). We introduce our data model in Section 4.2 and present how our content model maps to the various entities of the data model. The backend module which is a RESTful web service that interacts with the database and offers functionality for the frontend is presented in Section 4.3. Last but not least the client module which we implemented as a Single Page Application (SPA) that provides a web-based frontend for our e-learning application is presented in Section 4.4.

Evaluation. In Chapter 5 we present the results of our evaluation. We conducted two evaluations of our e-learning implementation and collected results through user surveys and as well as statistical data from our e-learning system.

Conclusion. In Chapter 6 we summarize the results of this thesis and present ideas for future work.

Related Work

In this Chapter we will introduce the e-learning platforms Khan Academy, Coursera, Udacity, edX and Moodle. For each of these platforms we will classify them in the context of the terms and definitions that we introduced in the previous chapter.

2.1 Khan Academy

With over 10 million unique users per month Khan Academy is a widely used e-learning platform [MGK⁺14, Ban15, kha16]. While Khan Academy originally focused on high-school and undergraduate mathematics, nowadays it offers a wide variety of topics including finance, science, engineering, art, and history.

History of Khan Academy. Salman Khan, the founder of Khan Academy originally started to record math videos in 2004, when he was tutoring his 13 year old cousin Nadja, who – at that time – was having trouble with her math lessons. As they were living in different cities Khan had suggested to meet online and he used voice chat and screen sharing software to explain the math concepts to his cousin [Tho11]. When for various reasons an online meeting could not happen, Khan recorded the instructions on video so Nadja could watch them later. After a while other cousins of Khan became interested and Khan was soon tutoring several of his cousins in mathematics. It became quickly apparent to Khan that his cousins preferred his videos to the live online meetings, because this way his cousins were able to pause and rewind the videos and go over the math concepts as often as they required, without becoming embarrassed, if they did not know a certain concept that they should have already learned [Kha12, Kha11]. In addition to the videos, Khan started to program small exercises that his cousins could solve online. After they had solved several exercises for one concept the software would start to give them more complex problems. Khan paired these exercises with a database

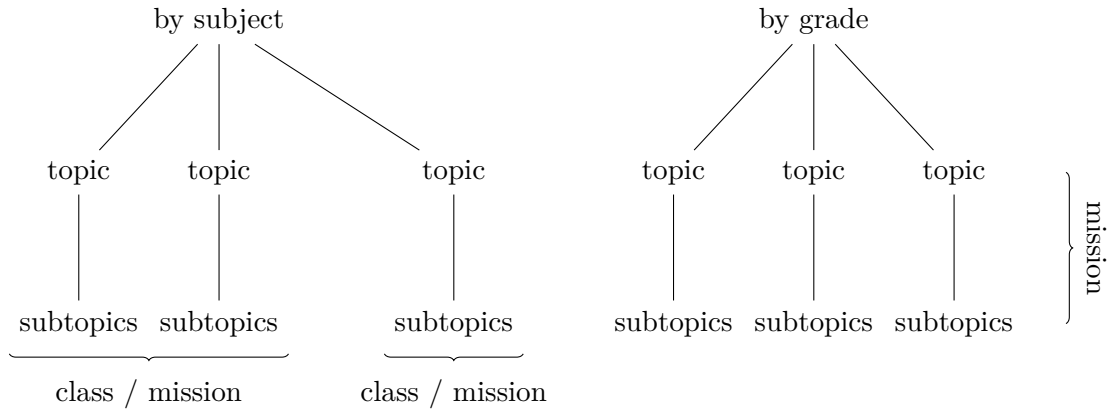


Figure 2.1: Khan Academy Content Structure

to be able to monitor his cousins progress, and to be able to focus on concepts that his cousins had the biggest problems with [Kha11, Kha12, Kha13b, Tho11].

After a friend of Khan had convinced him to upload the videos to Youtube, Khan soon discovered that except his cousins many other people were interested in his videos. In 2009 the interest in his videos had become so large that Khan was finding it hard to focus on his daily job, and so he decided to quit his job and founded Khan Academy as a non-profit organization [Kha11]. Since then, the organization has received a large amount of funding, which the organization used to continuously expand and improve their content. Today the Khan Academy website¹ is the central point of access for all students. At the time of this writing their content covers the areas of mathematics, science, engineering, computing, arts & humanities, economics & finance and more.

Content Structure. In the remainder of this Section we will focus on the mathematical content in order to introduce Khan Academy in more detail. Khan Academy uses several different logical views to structure its content (see Figure 2.1), at the highest level the views are structured *by subject* and *by grade* and each subject or grade is subdivided into *topics* and *subtopics*. A subtopic (see Figure 2.2) has a title and description and usually consists of videos and zero or more skills (e.g. exercises) that are used to practice concepts from videos. We will take a closer look at skills and exercises in the next paragraph (see Achieving Mastery). A subtopic can also contain zero or more reviews, which are text-based summaries of the content and which end with a short quiz. Icons are used to distinguish between the different content types (see Figure 2.2), a video is represented by a play icon, a skill is represented by a star icon and a review is represented by a document icon. Items that have not yet been accessed are shown with a gray color. Besides the structure of topics and subtopics each subject is also structured into classes, where each class contains a subset of topics of the respective subject. The main difference

¹<https://www.khanacademy.org/>

between classes and topics is, that classes offer an additional logical view called *mission*. The content structure by grade, also offers a mission view, which covers all topics and subtopics of the respective grade (e.g. 8th grade math). These two different mission views allow students to practice content either by class or by grade (see Figure 2.1).

A mission allows the students to track their progress as they practice the skills from topics of a particular class or grade (see Figure 2.3). The mission view is structured in the form of a dashboard that includes the mission progress, a skill breakdown table and an unsorted list of *skills* from which the students can select the next skill to practice. Each mission starts with a *warm up* phase, where the student must solve a sequence of exercises for various skills, until the software has determined the current level of the student. Once the warm up phase is completed, the software determines a set of skills from different topics and presents them in the mission view.

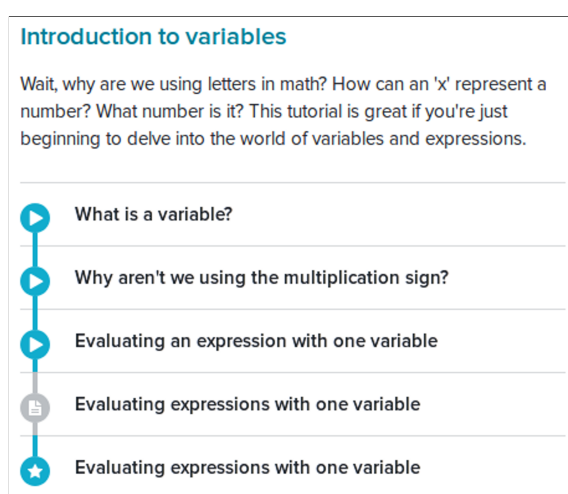


Figure 2.2: Subtopics

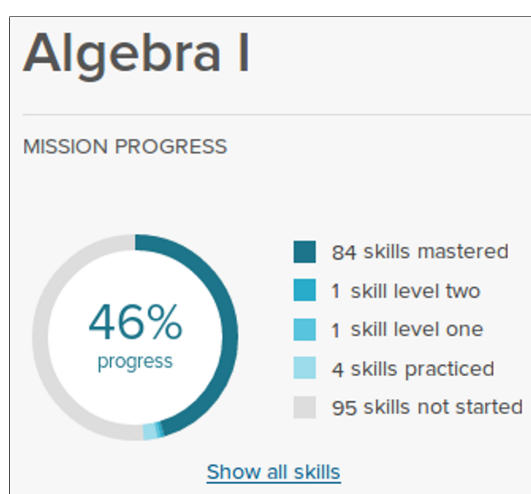


Figure 2.3: Mission Progress

Achieving Mastery. Skills and exercises in Khan Academy are a central part of the platform. Each skill corresponds to a particular type of exercise. Each type of exercise can generate an arbitrary number of exercise instances that can be used to practice the corresponding skill. For example the skill called *Interpret Quadratic Models* corresponds to a type of exercise that requires the student to solve quadratic formulas and determine the extremum or the zeros of a function.

Some exercises also make use of dynamic models that allow students to interactively experiment with the concept that is being learned. For example the exercises for the skill *Graph quadratics in vertex form* contain a graphical tool to manipulate quadratic functions (see Figure 2.4). The dynamic model allows students to manipulation the graph of a quadratic function through two green handles. Students solve the exercise by dragging each of the two green handles on the graph to positions that are valid solutions of the exercise equation.

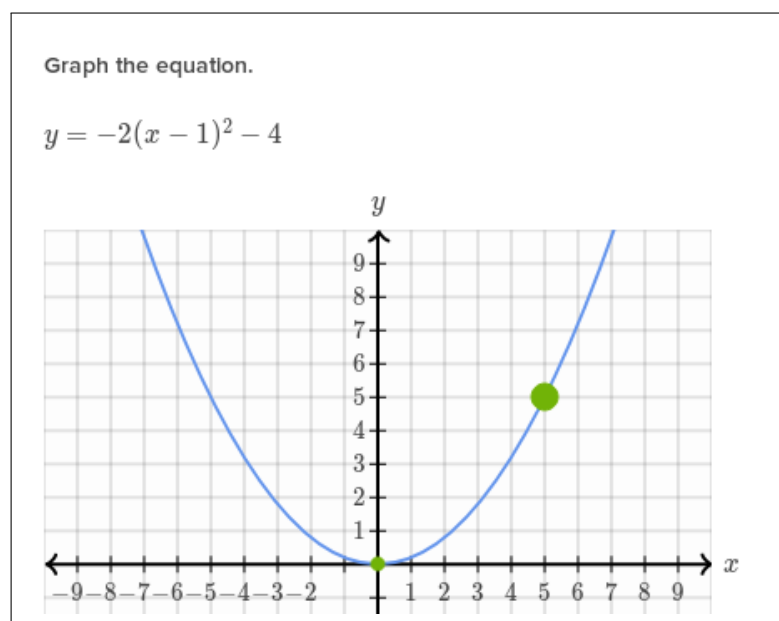


Figure 2.4: Exercise with a Dynamic Model for the Skill *Graph quadratics in vertex form*.

For each skill a five level scale determines the degree of mastery that a student has already achieved in a particular skill. The lowest level is *Needs Practice*, which at the same time is the initial level of each skill that has not been started by a student. The highest level is *Mastered*, which means that the student has achieved mastery of the skill. The following list shows the five possible levels for each skill:

- Needs Practice (Not Started).
- Practiced.
- Level One.
- Level Two.
- Mastered.

In order to pass through the above levels two steps are necessary. First a student must practice a skill and solve several exercises correctly in a row. The number of exercises varies between 1 and 5 depending on the current level and on the success rate of previously answered exercises from that skill. When a student succeeds and solves the right number of exercises without making a mistake in between, then the student reaches the *practiced* level. To reach higher levels the student must solve so called *mastery challenges*, each mastery challenge includes up to five exercises from skills that the student has previously practiced. If a particular exercise is solved correctly the next level is reached, otherwise

the level stays the same or if during a previous mastery challenge the exercise has already been solved incorrectly, then the level is reduced to the next lower level. A so called rest period [Acab], is enforced between mastery challenges for the same skill; the student can still practice the respective skill, but will only be able to engage in a mastery challenge and thus level up once the rest period (e.g. 16 hours) has passed. Mastery challenges also include exercises from skills that a student has already mastered, to ensure that the student is still able to use the respective skill.

Gamification. Khan Academy has integrated gamification (see Definition 14) aspects into its website to increase the motivation of students to watch videos and engage in exercise. The gamification aspects on Khan Academy include the following concepts:

- Energy points.
- Different types of badges.
- Skill levels.

Energy points are awarded to users for watching videos, practicing exercises and for completing mastery challenges. Energy points on Khan Academy unlock new (and visually more appealing) avatars, that can be used as profile icons. Badges are awarded to users for the completion of various tasks, such as the completion of a topic or subtopic, which requires to reach the mastery level for all skills in a particular topic or subtopic. Other badges are also awarded for reaching certain amounts of points or for watching a certain amount of videos. Different types of badges exist, and some types of badges are more rare and thus more difficult to achieve than others. The above mentioned level system for skills is another aspect of gamification on Khan Academy. Points and badges that a user has earned are shown on the users profile dashboard and are visible to other users by default, thereby creating some degree of competition between students.

Educational Aspects. The design and education concepts of the Khan Academy website enable a pure online experience as well as a blended learning environment [Kha11]. Khan Academy has been designed for mastery learning, rather than for scheduled learning. This design is visible in many areas of the website. For example the fact that each subject is broken down into topics and subtopics reflects Blooms suggestion to separate a course into short units. Another example is the concept of mastery challenges, which are an implementation of formative evaluation (see Definition 7), this allows the students to get immediate feedback when they practice a skill.

Due to the level system a student can easily identify the skills that require further study or practice and decide to watch the corresponding video again or to further practice the skill. If a student does not know the answer to a particular exercise, it is always possible to request a hint that includes detailed step-wise instructions to solve the exercise. The level system and the hints are both a step towards Bloom's suggestion to give students

specific prescriptions as a result of formative evaluation. Besides formative evaluation the exercise concept also ensures long term retention of the practiced concepts, since the student will have to practice a particular skill multiple times before mastery can be achieved. The fact that already mastered skills can reappear in mastery challenges is another factor towards ensuring long term retention.

Khan Academy provides a *coach dashboard* for use in blended learning and flipped classroom environments. The coach dashboard allows teachers to create classes and track their students progress. When a class is created content can be added by choosing from various categories of the mathematical content (e.g. subjects or grades, see Figure 2.1), or from many of the other subjects that Khan Academy offers. Once a class has been created, students can be added to the class and exercises or videos for particular skills can be assigned to students. Even though the focus of Khan Academy is not on scheduled learning it is also possible to assign due dates for assigned exercises and videos. Using various statistical tools, teachers can monitor their students progress and identify students who are struggling to master a skill. The statistical tools include a progress view that shows *progress by skill* and *progress by student* as well as an activity view that shows when and for how long each student was active. The activity also contains detailed information about the videos and exercises that each student has engaged. These statistical tools make Khan Academy a perfect tool for use in a flipped classroom environment [Gup12].

Additionally Khan Academy certainly qualifies as a tool that can be used for a purely online experience (see Section 2.6 and Table 2.1), in particular due to the mission concept and the gamification aspects of Khan Academy.

Instant Feedback. As discussed in Section 1.3 the duration of a feedback loop on Khan Academy is approximately 30 minutes due to the fact that students first watch one or more videos to learn a skill before they practice the skill in a separate practice section. While a duration of 30 minutes is much faster than the feedback times of two weeks that were anticipated by Bloom [Blo68] it is not yet on the scale of minutes that we want to achieve with our concept of micro-tests and dynamic models which are directly integrated into the content. While Khan Academy does make use of dynamic models these are tied to the exercises and therefore separated from the content. This means that there is a lower bound for the duration of feedback loops since students must always perform a context switch from content to exercises to complete a feedback loop.

2.2 Coursera

Another popular e-learning platform is Coursera. Coursera specifically targets students of higher education and its educational concept is very close to that of traditional teaching in universities. In particular it uses a course format that is very similar to that of a traditional university course. Coursera itself does not produce any content, instead it partners with universities and other educational institutions who create the content. The Coursera platform itself only provides the tools and interfaces to create the course

content, including assignments and quizzes, and the ability to integrate the course content into the platform.

Coursera supports two course formats, a session-based course format and an on-demand course format. Session-based courses are only available while the course is active and the course content becomes unavailable when the course ends [Sha15]. On demand courses on the other hand are always available for users. However, even on-demand courses have a virtual starting date, where courses keep restarting on a monthly schedule [Coub, Spre]. Students must always enroll in a course before they are able to access its content.

Coursera offers free of charge access to the course material of almost all courses [Car14, Couc], with the exception of some specialization courses. Depending on the course there can be up to three ways to enroll for a course, two are free and one is payed. The free modes are “*Audit Only*”, meaning that the students cannot submit assignments and will also not receive a certificate, and “*Full-Course, No Certificate*“, meaning that the student can access the whole course content including assignments. They will also receive a grade but no certificate. Finally in the payed mode all these options are available.

Educational Concepts. The content of a course is organized in weekly parts, rather than by topic. This shows that Coursera uses a *scheduled* learning model, rather than one which is aimed at *mastery learning*. The scheduled learning approach is also reflected by the fact that a course always has a start and end date. While the general learning model of Coursera is scheduled, Coursera claims to have some aspects of mastery learning build into their courses. For example they often give immediate feedback to questions and quizzes and they “provide randomized versions of the assignment so a learner can re-study and re-attempt until they master it” [Coua]. There are also universities who offer their courses on Coursera in addition to their on-campus lecture “to provide their on-campus students with an improved learning experience” [Coua]. This shows that – to some degree – Coursera can be used in blended learning environments.

Content Types. Since the course content is created by the partnering institutions of Coursera it is not possible to give a general description of content types, since the type and style of content heavily depends on the organization who creates it. Nevertheless the following list gives some examples of the content types that we encountered by looking at a few randomly selected courses:

- Video content.
- Textual content integrated into the Coursera website.
- Textual content hosted on other websites.
- PDF Files that have to be downloaded and read separately.
- Assignments, such as quizzes, exercises, homework, peer-graded assignments and exams.

Generally each course contains at least a few hours of video per week (on average 8-9 hours per month [KNDC13]). Where each video is around 15 minutes long and has embedded assignments to help the student verify their understanding of the course material [Sev12]. The videos often contain a transcript, that is shown below the video stream and the respective sentences from the video are automatically highlighted as the video stream progresses.

The weekly parts of a course often have assignments, which have due dates and must be completed in time for the assignment to be graded. The type of such assignments includes quizzes, exercises, homework, peer-graded assignments and exams and the required time to complete these tasks ranges from a few minutes for short quizzes to several hours for peer-graded assignments. There seems to exist no clear terminology to distinguish between properties of different types of assignments such as quizzes exercises or homework.

The scheduling of content into weekly parts is also part of a design to ensure that students follow the course at a similar pace. This scheme has two benefits. First, it avoids a fragmentation between students that rush ahead and those that lag behind. This ensures that more students can engage in discussions and help each other using the communication tools, such as *forums*, built into the Coursera platform. The second benefit concerns the assignments. Coursera has created a *peer-grading system* to grade those assignments, that are too complex to be graded automatically. Each student who submits an assignment has to review and grade a randomly selected assignment from some other student [Sev12]. Finding a suitable student for the peer-review is easier if all students engage on the same assignment, rather than having to pair students from different weekly parts.

Is Coursera a MOOC platform? Coursera has been designed for scalability right from the start and has been tested with course sizes of more than 100,000 students [Sev12]. So courses can certainly be classified as massive according to the MOOC acronym. However, since not all course content is available for free, the openness of the courses is limited. With respect to licensing we must separate between user generated content such as forum posts and the actual course content. User content is generally copyrighted by Coursera as stated in the “Terms of Use”².

With respect to course content, “Coursera [...] claims no intellectual property rights to the content hosted on its platform, and does not demand an exclusive relationship with university content providers.” [Bog15] [Cou12, p. 6, ‘Non-Exclusivity’]. This means that the openness of course content is determined by the license that was chosen by the partnering university.

Courses on Coursera certainly qualify for the online part of the MOOC acronym, as the platform has been designed to be used as a pure online solution. This observation is based not only on the fact that all course material is available online, but also on the

²<https://www.coursera.org/about/terms>

fact, that students can actually receive grades from pure online participation, due to the use of automatically graded or peer-graded assignments. Last but not least Coursera courses obviously satisfy the course acronym of the MOOC term.

Instant Feedback. Coursera is a platform where different authors can create courses and therefore it is not possible to make a general statement about the duration of feedback loops. The duration of feedback loops and the methods to give feedback depend on how authors create their courses and can thus differ from course to course. Coursera supports several feedback methods and thus has feedback loops of varying durations. The fastest feedback loop is achieved through optional "in-video" questions [Cen18] that appear intermittently during videos, which allows feedback loops on a scale of several minutes (see Section 1.3 for a discussion of this method). Other forms of feedback include automated assignments, peer-reviewed assignments and forums. Especially peer-reviewed assignments and forums can have feedback loops with a duration of several days, even though feedback times may sometimes be much faster.

2.3 EdX

EdX is in many aspects similar to Coursera. Like Coursera it is an e-learning platform that offers courses that are focused on higher education and which are often referred to as MOOCs. Courses take several weeks and students need to enroll in a course in order to get access to content. EdX offers two kinds of courses – *self-paced* courses and *instructor-paced* courses. Self-paced courses can be started by the students at any time, while the instructor-paced courses start at a specific point in time. Courses which have been held in the past may still be accessible to students as *archived* courses but certain aspects such as graded assignments, homework or the final exam will not be available.

The edX platform was founded by Harvard and Massachusetts Institute of Technology (MIT) in 2012, and according to its own description is the "only leading MOOC provider that is both nonprofit and open source" [edX, Shu13].

EdX is only the name of the official edX website, while the underlying software is named Open EdX. Since the open edX platform is open source³, interested organizations can use it to deploy their own version of the platform. For example in 2013 ten Chinese universities have partnered to create their own online learning portal called XuetangX [Inc13], as of 2016 several other countries and organizations have created their own online learning portals based on the open edX platform. This also gives course creators a choice to publish their courses on any e-learning portal that is based on the open edx platform. The edX website also provides a private staging environment called "edX edge" [Incd], that can be used to develop courses or create non-public courses for a limited audience.

Content Structure. Courses on the edX platform are directly integrated into the platform and can be built with an integrated editor. An official documentation [Incc, Incb]

³<https://github.com/edx/>

is available that has detailed explanations for building courses on the edX platform. The following paragraph will just give a brief overview of the most important aspects. Each edX course has a similar overall structure but the content types used to build the individual segments can vary from course to course. The overall structure consists of a segment navigator on the top, a table of contents on the right side and the content area in the center [Fis12].

The content in the content area is built from *components* of varying types such as:

- HTML components that can contain any kind of HTML.
- Videos.
- Discussions.
- Problems.

It is also possible to create a library of these components that can be reused in different courses and that can be shared with others [Incb, Chapter 6 and 7].

The above mentioned *problem* components can be created from a pool of more than forty different tools for interactive exercises, ranging from simple ones such as *checkbox* or *dropdown* problems to more complex ones such as a *circuit schematic builder* or a *chemical equation problem*. Some of the more complex problem types include dynamic models of a concept, such as the *circuit schematic builder*, which has a model of electrical circuits that allows students to interactively composition electrical circuits to solve a problem. Detailed documentation for the available problems can be found in the edX Documentation [Incb, Chapter 8].

Educational Concepts. Since edX is a MOOC platform it can be used as a pure online environment, but edX also encourages universities and teachers to use their MOOCs in a blended learning environment and there exist several reports where this has successfully been done [Aga13]. In fact edX even offers its own MOOC about blended learning that teaches how to apply edX in a blended learning environment [eI16].

When we look at the *time aspects* of edX courses we see that their general concept is a scheduled approach, rather than being focused on mastery learning. However edX does not place such a strong emphasis on the time aspect as Coursera does. While on Coursera courses have start and end dates and their courses are split into weekly parts, edX organizes their courses into units, which do not necessarily take a whole week to complete. EdX also has many different types of exercises and problems (see above), some of which have aspects of mastery learning. For example there exist repeatable exercise types that can be reset by the user in order to practice the problem again. When the exercise is reset, a new instance of the exercise is generated with different values.

Instant Feedback. Similar to Coursera, edX supports different feedback methods such as *problems*, a forum-like discussion section and peer-reviewed assignments. The duration of feedback loops on edX is again dependent on the individual design of each course and thus no general statement about feedback loops can be made. However, as a large degree of courses uses videos this also means that the duration of feedback loops depends on the length of those videos (see Section 1.3).

2.4 Udacity

Udacity is an e-learning platform similar to edX and Coursera but its focus is mainly in the field of computer science and mathematics. It was founded in 2011 by Sebastian Thrun, David Stavens and Mike Sokolsky [Inca] as a for-profit company. Like Coursera and edX it was originally a platform for MOOCs, but it suffered from the same symptomatic problem of high dropout rates [Aga13, Jor15, Jor14], where few students finish a course and even fewer get a good passing grade.

Thrun realized that the platform was not as successful as he had initially anticipated and even called the platform a “lousy product” [Cha13]. As a result he decided to shift the focus from MOOCs to the teaching of “vocational skills” [Cha13] that are of practical use for students and help them to increase their chances of finding employment. In order to ensure that skills which are taught in Udacity’s courses are demanded by companies, Udacity is partnering with several large IT companies such as Google, AT&T, Facebook, Salesforce, Cloudera and others [Udab].

Currently Udacity offers both individual courses and so called *nanodegree programs*⁴. A *nanodegree* is essentially a marketing term for online courses. Besides video lectures and quizzes each *nanodegree* includes comprehensive *portfolio projects*. Further more Udacity provides career support for students to enhance their chances of finding a job. Portfolio projects are a mandatory part of nanodegrees that must be completed in order to receive a certificate. The portfolio projects are also reviewed by instructors (including code review) and students receive personal feedback. Upon completion of the nanodegree, these portfolio projects can then be used as references in job applications. *Nanodegrees* are created in close collaboration with industry partners and are therefore tailored to teach skills which are in demand on the job market. The offering of individual courses includes free and paid courses and there are 21 nanodegree programs at the time of this writing, including web and mobile development and a nanodegree program for a “Self-Driving Car Engineer” [Ince].

Content Model. The Udacity content model is strongly focused on professionally produced high quality short video sequences, which are just 2-3 minutes long [Udaa]. Quizzes are the predominant type of testing the student’s understanding of the content. In the course which we evaluated [LSK], about 50% of the content was made up from

⁴The word “Nanodegree” is a registered trademark of Udacity.

quizzes, which were interleaved between the videos. There is usually a quiz after every two or three videos of content.

A quiz is a set of one or more exercises that requires students to reflect about the content that they currently learn. The quiz question itself is presented in the form of a short video sequence. After the quiz has been solved by the student another short video sequence explains the correct solution. This means that the whole content in a Udacity course except for the answer dialog of quizzes is in the form of short videos.

There exist two types of quizzes, simple form-based quizzes and programming quizzes. For simple quizzes, an answer form will be rendered on top of the last frame of the video (see Figure 2.5), such that form elements overlap with answer candidates or blank spaces that were drawn in the video. The quiz questions are explained in the video that precedes the quiz, and the answer form will only contain the form elements to enter the answers.

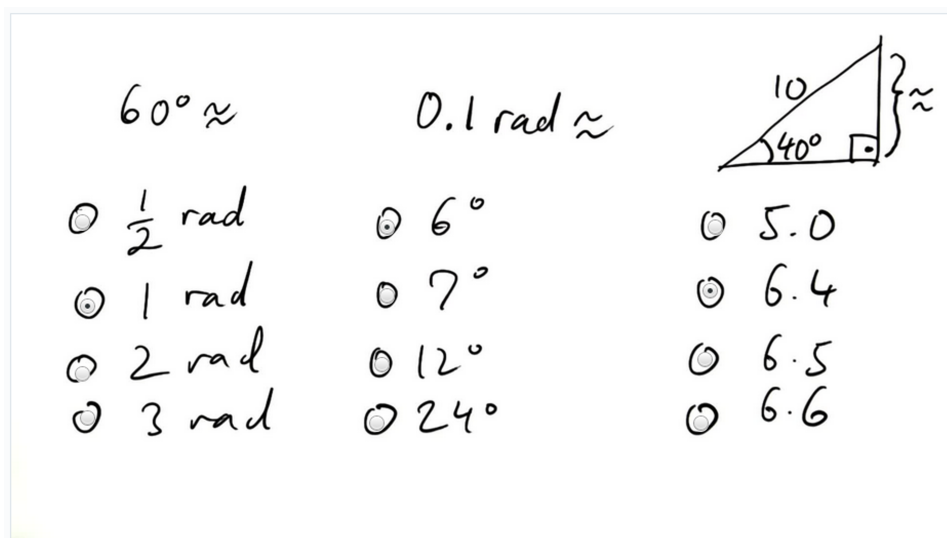
The programming quizzes require the student to write a short amount of Python code (e.g. 2-20 lines). In this case the video content is replaced by an integrated code editor. At the time of this writing the code editor offers syntax highlighting but no other features that are found in popular IDEs.

For all quizzes the students can submit their answer as many times as they want and they always have the option to skip the quiz and directly view the correct answer. Afterwards they can go back to the quiz and use their new knowledge to answer the quiz. It is the responsibility of the students to try and solve a quiz by themselves, rather than to skip ahead to the answer of a quiz. However, if a student has tried to solve the quiz and did not find an acceptable solution, then the ability to access the correct answer can provide valuable help.

For programming quizzes there is an additional option, that allows to test the code without submitting it, this will show the output of the code on the screen (e.g. both textual or graphical output depending on the code) and allows the students to experiment with the program before they submit their answer.

Educational Model. The educational goals that Udacity pursues are different from the other platforms that we have reviewed so far. The platform is designed for a pure online learning experience and we could not find references that it is being used for blended learning or in flipped classroom environments. It does have concepts for mastery learning built into the courses, but with a different approach than Khan Academy.

In particular, as we have seen earlier, there is usually only one exercise instance of every quiz or exercise type and students can always access the answer without any kind of penalty. A student, who is unable to complete the quiz and views the answer, will not achieve the same learning effect, when subsequent work on the exercise is done, since the student then already knows the answer. In this situation it would be useful if further exercises could be generated, in order to practice the concept in different contexts.



This Figure shows a screenshot of an integrated exercise. In the video several possible answers are provided. The video is then stopped and a mechanism to select an answer (see circles) is layed over the video.

Figure 2.5: Udacity Quiz with Answers Mechanism.

An important aspect of formative evaluation (see Definition 5) is to provide prescriptions that enable the students to resolve problems with their understanding of the content. As we have discussed above for each exercise there exists an answer in the form of a video that explains the correct solution. However, the usefulness of the answer depends on a large degree on how detailed the solution is explained, and in particular whether it includes the intermediate steps that were required to arrive at the solution. During our evaluation of Udacity we found, that these intermediate steps were not always included in the answer. Of course this may differ depending on the course. We also found, that there is no system that provides references between the concepts that are required in an exercise and the videos, which teach those concepts. Thus, it is left to the students, to browse through the course material and to find the videos that correspond to the concepts, which they need to review.

Another related aspect concerns the nature of quizzes that include programming tasks. The student may be unable to solve the quiz, not because of problems with the actual content, but due to problems with the programming language and with the required library functions, that are required to solve the quiz. In such a situation a students has three options:

- Post a question in the forum and ask other students.
- Review the previous course content to find more information.
- Skip to the answer and learn the solution.

Obviously skipping to the answer is both the easiest and fastest choice for the students. In this situation, it might be useful to distinguish between programming issues and content related issues, and allow the students to request hints only for the respective issue.

The fact that there exists only one instance for each type of quiz and the fact that students are not given thorough prescriptions after each quiz, leads us to the conclusion that Udacity has implemented Bloom's suggestions for mastery learning only to a small degree.

Instant Feedback. As we have seen before, a typical feedback loop on Udacity includes watching between one and two videos to learn a concept. These videos are followed by a quiz which starts with another video explaining the quiz question. Once a student has answered the quiz a follow-up video explains the correct solution. A feedback loop, therefore, consists of three to four videos. If we take into account that the average length of a video is 2-3 minutes then each feedback loop includes at least 6 minutes of video. However, for many cases, 10 minutes is a more realistic value.

The duration of a feedback loop also depends on the time that a student requires for solving the quiz which in-turn depends on the type of quiz. Programming quizzes are often more difficult to solve than form-based quizzes and therefore take longer to solve. This is because for programming quizzes a sufficient understanding of the programming language and the required library functions is required in addition to understanding the conceptional or algorithmic problem of the quiz. The complexity of quiz questions also varies and increases as a course progresses. Based on our experience it takes at least 10 minutes to solve a typical form-based quiz; again this might be much longer for many quizzes.

Taking these considerations into account the duration of a feedback loop on Udacity is at least 16 minutes - but may be significantly longer. An important consideration when comparing the duration of feedback loops with other platforms is that all videos on Udacity are professionally produced and potentially have a significant production cost. With a maximum cost of "\$4,300 per hour of finished video" [HT14], a three-minute Udacity video might cost as much as \$215 to produce.

2.5 Moodle

Moodle is different from other platforms discussed so far by the fact that it is not a single well known platform but rather a framework that can be used to build customized e-learning systems. In this respect, Moodle uses a rather general "one-size-fits-all" approach that tries to cover a broad variety of requirements that a typical school or university will have [Moog, Mooc]. Besides features that are directly related to course content and exercises, Moodle also supports a lot of features that concern the management of courses and students, communication, time management, knowledge management and several more.

Course Content. There are several ways to create and manage course content. With respect to multi-media Moodle supports adding audio and video content that can be embedded directly into a course. There is also file management support. Various types of files can be uploaded and linked to a course and can then be downloaded by the students.

Another way to create course content is by using the various activity modules such as *Assignments*, *Choice*, *Quiz*, and *Lesson*. The following list gives a brief overview of the features that each of these modules support:

- **Assignments:** The Assignments activity [Mood] allows teachers to grade and comment on various kinds of assignments.
- **Choice:** The Choice activity [Moof] allows the creation of one simple question as a multiple-choice type (both single answer and multiple answer are possible). It is intended for a quick collection of feedback from students and much less comprehensive than the quiz module (see below).
- **Lesson:** The Lesson activity [Mooj] allows to create HTML pages with content and choices.
- **Quiz:** The Quiz activity [Mool] allows the creation of quizzes, consisting of a set of questions of different types. A wide range of question types are available such as true/false, multiple-choice (single answer and multiple-answer), short answer, numerical, and several others.

Administration. Moodle has features to manage courses, users and groups, and comes with a built-in role and permission system. It provides reporting functions to evaluate student participation and comes with a plug-in system that allows the extension of Moodle by third parties. Moodle also has a high degree of interoperability and supports open standards to import and export courses.

Communication. Moodle comes with several modules for communication, such as a forum for asynchronous communication and a chat system that allows synchronous real-time communication between students.

Time Management. Moodle has a built-in calendar module [Moee] and a notification system [Mook]. The calendar keeps track of site events, course events as well as group and user events and allows the user to create own events. With the built-in notification system the user can be notified about deadlines, or events from the communication modules such as forum posts and messages.

Knowledge Management. Moodle comes with a built-in *Wiki* activity [Moon] for collaborative knowledge management as well as a Glossary activity [Mooi] to keep a list of definitions and terms.

Feedback. There are different ways to collect feedback and statistical data with Moodle. The first is the *Feedback* activity [Mooh], which is actually used to conduct surveys. It is described as “ideal for the likes of course or teacher evaluations” [Mooh]. There is another activity that is actually called *Survey* [Moom]. In contrast to the Feedback activity, the Survey activity already comes with predefined and standardized survey questions.

Moodle as a MOOC platform. Moodles has been described as a MOOC platform by various reports [Swol4], and there exists even a “Learn Moodle MOOC” [Moob], which is built with the Moodle platform. From a technical point of view Moodle seems to have the required features to create course content and exercises, but regarding scalability it is somewhat limited, for example the Moodle documentation lists a limit of 5.000 users per course and gives no numbers regarding the supported number of concurrent users per course. Given the fact that some of the more popular MOOC courses had more than 100.000 users this seems a little small.

The Moodle code base is also rather old and the current architecture has been in use since 2001. As such Moodle does still mainly rely on a mostly server-side architecture with frequent page reloads and therefore does not provide the look and feel of some more modern SPAs.

In summary Moodle can be classified as a suitable MOOC platform as long as the stakeholders are aware of its limitations.

Instant Feedback. Moodle is a framework which supports different ways to create course content and to collect feedback. Therefore, it is not possible to make a general statement about the duration of feedback loops for Moodle. As we have seen Moodle has several built-in modules to collect feedback such as the Assignment, Choice, Lesson or Quiz modules. In addition, Moodle also supports the creation of custom JavaScript modules that could be used to collect feedback.

2.6 Comparison

In this Section we compare the different platforms regarding their qualification for certain learning methodologies such as mastery learning, blended learning and flipped classrooms (see Table 2.1). We have already analyzed, which platforms are MOOC platforms and which parts of the MOOC acronym are not satisfied by the different platforms. In this Section we give a summary of this discussion. Finally, we relate the goals that we defined earlier (see Section 1.3) to the different platforms and explain why – instead of using one of these platforms – we decided to build our own e-learning implementation for the evaluation of micro-tests and instant feedback.

Mastery Learning. From the platforms we have introduced, currently the only platform that fully supports the concept of mastery learning is Khan Academy. Khan

Name	Platform	Flipped Class- room	Mastery Learning	Scheduled Learning	Video Content
Khan Academy	Yes	Yes	Yes	No	Yes
Coursera	Yes	No	No	Yes	Yes
EdX	Yes	possible	limited	Yes	Yes
Udacity	Yes	No	No	Yes	Yes
Moodle	Yes	possible	possible	Yes	possible
Our imple- mentation	Yes	possible (fu- ture work)	planned (fu- ture Work)	No	possible

Table 2.1: Comparison of E-Learning Platforms and Software

Academy has integrated mastery learning deeply into their software in the form of missions, repeatable exercises and their level concept which requires students to repeat each exercise a certain amount of time. This means that repeating exercises until the student has mastered a concept is a “first-class” element in the design of the learning experience of Khan Academy.

While Coursera and edX advertise that they support mastery learning, this actually depends a lot on the individual courses and whether they actually implement exercise types that are intended to be done multiple times by the students. In any case these platforms do not place the same emphasis on repeatable exercises as Khan Academy does and there is no comparable implementation of a level concept like the one Khan Academy has implemented. While there are repeatable exercise types on Coursera and edX, the majority of exercises are intended to be answered just once. This shows that the emphasis is more on giving the student the ability for self-evaluation rather than to practice a concept. In addition, due to the fact that exercises and assignments are often graded, it may not even be possible to reset an exercise to repeat it.

Udacity supports aspects of mastery learning, but it takes a different approach than Khan Academy. Khan Academy has a strong focus on repeating exercises until the student has mastered a concept, this includes passing through several levels and repeating already mastered exercises to ensure that students have not forgotten the content. Udacity has a lot of exercises (called quizzes) on their platform, but unlike Khan Academy there is generally only one instance for each quiz, and there is no built-in system to repeat the exercise multiple times. Since there is only one instance of each quiz, the students do not get the ability to practice an aspect in different contexts to ensure that they have completely mastered it.

Blended Learning and the Flipped Classroom. Whether a platform is suitable for blended learning and flipping the classroom depends on several factors. In general every platform could be used in a blended learning environment. The teachers would

simply ask the students to learn a particular concept via the platform and then during the lecture they would focus more on practice and interactive tasks rather than on the teaching of new content. However, there is more to blended learning and flipping the classroom than simply asking students to use a certain platform for learning. The most important aspect is that teachers have the ability to get statistical information about their students' progress. This includes that teachers know which concepts the students have already learned when they come to the lecture and whether they have had problems to apply these concepts in the exercises.

Unfortunately, for this thesis we had only access to freely available information, such as the information provided by the platforms themselves, information from scientific publications and from blogs or videos, but we did not have access to paid features or paid content on platforms such as Coursera, edX, and Udacity. We have been able to empirically evaluate the platforms from a students perspective, but except for Khan Academy we have no insight as to how the teachers can monitor their students' progress on the platforms. This means that with respect to blended learning and flipped classrooms we can only present a very limited comparison for the platforms Coursera, edX and Udacity.

From the five platforms which we reviewed, we have seen that Khan Academy is particularly suited for being used in a blended learning and flipped classroom environment. Khan Academy is currently the only platform for which it is well documented that it is being used in blended learning environments and flipped classrooms [Kha13b]. One aspect that makes Khan Academy useful for blended learning and flipped classrooms is the *coach dashboard* which allows teachers to create their own *classes* and to track their students' progress.

While Coursera advertises blended learning on their website [Coua], we were not able to find any definitive references that report whether Coursera was successfully used in blended learning environments. EdX has reported usages of their platform in flipped classroom environments [Aga13], but again we have no data on how widely that is being applied or how successful it is. Udacity does not advertise blended learning or flipping the classroom with their platform. The Moodle platform itself supports the use of both blended learning and flipped classroom concepts for courses [mooa, Edu, Moo13]. Just like with Coursera and edX we were not able to find any data about actual courses which make use of this.

MOOC Acronym Comparison. During our research, we found that it is possible to find sources for each of these platforms, which describe these platforms as MOOC platforms. However that is not always accurate, so we will take a closer look at each of the platforms to see if and how they qualify as MOOC platforms (see Table 2.2). In the preliminaries (see Section 1.2) the concept of openness was discussed. We saw the term *open* can be used in the context of both licensing, where it refers to the ability to change and redistribute a licensed work, as well as cost, where it refers to the fact that

something is available free of charge. Therefore we discuss these aspects separately in our the comparison.

- **Khan Academy** is not a MOOC platform [Aka13], simply because its main form of content structure is not organized in courses but by subjects and by grade. Typical concepts of courses such as the requirement to enroll in a course as well as start and end dates are also not implemented by Khan Academy. Khan Academy satisfies the massiveness aspect, since it does support millions of users. Its videos and exercises are licensed under the Creative Commons CC-BY-NC-SA 3.0 license [Acaa, Com], which is an open license. This means content on Khan Academy is free of charge and it is also open, as long as the content is not used for commercial purposes. Even though it is not a MOOC platform, we have still included it in the comparison table to allow easier comparison between the e-learning platforms.
- **Coursera** is clearly a MOOC platform. Its main form of content structure is the course. Coursera also supports millions of students on their platform. However Coursera is not very open. All courses are licensed under copyright licenses and many courses require payment to enable the full range of functionality.
- **EdX** is a MOOC platform. Like Coursera its main form of content structure is the course. EdX is built for millions of users so it clearly satisfies the massiveness aspect. The edX platform itself is open source, but the courses are not. So just like Coursera most courses are licensed under copyright licenses and many courses require a payment to enable the full range of functionality.
- **Udacity's** main focus is to offer vocational training for practical skills that are currently in demand on the job market. Udacity's focus is not on MOOCs although such courses are still being offered. The focus of the platform is rather on specialized training of skills in combination with practical project work (e.g. nanodegree). As these nanodegree are neither free, open or massive we argue that Udacity is not a MOOC platform. We have still included it in our comparison because it is a popular e-learning platform.
- **Moodle** is a software platform that can be used to build implementations of e-learning platforms. The question if a particular Moodle-based e-learning platform is a MOOC platform, depends to a large degree on the individual implementation. Just by itself Moodle is not a MOOC platform, however individual implementations can be developed in such a way that the courses can be considered MOOCs. As we have discussed earlier, the scalability aspect of Moodle is rather limited and so is the massiveness aspect of courses on a Moodle-based platform is certainly not on the same scale as the other platforms that we have discussed.

Name	Massive	Open (Free of Charge)	Open (License)	Online	Course
Khan Academy	Yes	Yes	No	Yes	limited
Coursera	Yes	partially	No	Yes	Yes
EdX	Yes	partially	No	Yes	Yes
Udacity	Yes	partially	No	Yes	limited
Moodle	limited	possible	possible	Yes	Yes

Table 2.2: MOOC Acronym Comparison

Results of this Comparison. In this section we have compared the different e-learning platforms that we introduced in this Chapter. Each of these platforms has different approaches with respect to the supported content types and exercise types. Each platform also uses different educational methods. In the remainder of this Section we explain why we decided against using an existing e-learning platform for our evaluation of micro-tests and dynamic models.

The common denominator for all platforms (with the exception of Moodle) is their strong focus on video content. Udacity exclusively uses video content and on the Khan Academy platform the majority of the content is also video-based, with the exception of review content which is text-based. Coursera and EdX take a mixed approach that uses both video and text-based content. Finally Moodle has support for a large variety of content types including videos but the actual usage depends on the individual implementation of a course and can hardly be generalized.

We have excluded Moodle due to its “one-size-fits-all” approach. Moodle is very generic, but also inflexible when it comes to adjusting the system to a particular user experience and design. We rather wanted to have the freedom to experiment with new usability concepts and designs that would have been difficult to achieve with the existing Moodle code base. Another reason is that we wanted to use the high degree of interactivity that is only possible with modern SPAs, an aspect which would have been difficult to achieve with Moodle as well. For our evaluation of instant feedback with micro-tests and dynamic models we will not require many of the features that Moodle supports, such as a calendar, grading tools, a chat system and others. Especially since the Technical University of Vienna already has a learning management system (e.g. TUWEL [Wie]), we do not want create a competitive system that covers all course related activities, but rather a specialized tool that has a focus on interactive learning.

We have excluded Khan Academy due to their extensive use of video content which does not provide a good match for our goal of largely text-based content. Coursera and Udacity are proprietary platforms and thus not available as a basis for our implementation. Last but not least there is edX. The edX platform is based on the open edX software, which is open source, has good documentation, a large developer community and a modern code

base. Nevertheless, the edX platform already provides a framework for the creation of content that fixes several design aspects (both algorithmically and visually) such as the overall course structure. Using the edX platform thus would have limited our possibilities for the evaluation of micro-tests and dynamic models, such as the possibility to implement and experiment with different forms of content presentation and workflows.

Due to the fact that we wanted to explore new possibilities for the presentation of our content and for the integration of micro-tests and dynamic models directly into the content stream we have decided that a new implementation offers the greatest flexibility to experiment with these ideas and to collect feedback from our users. If our approach proves successful and as the application grows we might require more features that have already been implemented in other e-learning tools. At that point we will reevaluate our choices and consider porting the implementation to a different platform. The most likely candidate for this would be the open edX platform as it seems to have the largest intersection of features to our implementation.

A course on our e-learning platform can to some degree be considered as a MOOC. We have a strong focus on the course concept, and the application has been implemented with a focus on scalability. The application is also suitable for a pure online use. However, our main goal was not to create a MOOC platform, but to evaluate micro-tests and dynamic models. Therefore, there are still some features missing before our platform can be considered suitable as a MOOC platform:

- There is currently no user registration implemented and thus user accounts need to be created by an administrator.
- There is currently no grading or exam functionality built into the software. However, this is an essential feature of MOOC platforms.
- We have designed the architecture to be scalable and to support a large number of users. However, since for the evaluation purpose of our implementation scalability was not our main concern, there are still some missing aspects in order to achieve a completely scalable application. In particular our client module does currently not implement any load balancing functionality.

Interactive Lecture Notes

In this chapter we will introduce our own e-learning implementation named “Interactive Lecture Notes”. More detailed explanations of the software architecture, the data model, implementation aspects, and the RESTful web interfaces are presented in Chapter 4. In Section 3.1 we summarize the requirements that lead to our e-learning implementation. In Section 3.2 we discuss our design goals with respect to the overall structure and presentation of the content stream, as well as the creation and import of content into our e-learning application. The detailed content model as well as the currently implemented content types that are available to create course content are presented in Section 3.3. An introduction of the different application roles for students, teachers and administrators is given in Section 3.4.

3.1 Requirements

In this section we briefly summarize the functional and non-functional requirements that lead to the design of our Interactive Lecture Notes e-learning platform.

Functional Requirements:

- Implementation of a content model supporting the content types defined in Section 3.3. This includes support for micro-tests (e.g. exercises) and dynamic models (e.g. interactive content).
- Implementation of a content stream with support for asynchronous loading of content elements.
- Implementation of a role-based system to separate application privileges for students, teachers and administrators (see Section 3.4) and the implementation of corresponding frontend dashboards for each role.

- Implementation of a data model to persist course content and course progress of students.
- Implementation of course statistics section in the teacher dashboard to present the course progress of students.
- Support for user management and batch import of users into the system via the administrator role.

Non-Functional Requirements. In addition to the functional aspects above we have considered several non-functional requirements that had an influence on the development of our software. These are:

- A flexible and lowly coupled architecture that allows for horizontal scalability and distributed deployment.
- A high degree of extensibility and maintainability.
- Build-in security from the beginning.
- Availability of documentation.
- Automated deployment (see Appendix A).

3.2 Design Goals

The main goal of this thesis was to evaluate if micro-tests and dynamic models are an effective means to provide instant feedback in e-learning systems. A related problem in this context was to investigate if video is a suitable type of content for the goal of short feedback loops. We have considered to evaluate micro-tests and dynamic models either in the context of an existing e-learning platform or by implementing our own solution. For reasons discussed in Sections 1.3 and 2.6 we decided to implement our own e-learning solution that does not use video as its main form of content delivery. Instead we have created a content model that uses text-based content types. This includes both marked-up text and interactive content types which are compiled from source code. Finally, we also wanted to use a presentation format that is unintrusive and does not distract our users from their actual goal, which is to focus on the content of a course.

Content Stream. We have designed the representation of the course content as a linear stream of content entries (see Figure 3.1). We refer to this representation as a stream, because new entries can be asynchronously loaded and appended to the stream. Therefore at any time the stream just represents a part of the total content of a course. We refer to this stream as being linear, because there exist no branches in the presentation of the content that would require the student to navigate away from the content, for example to access content on other pages. This way of presenting the content closely

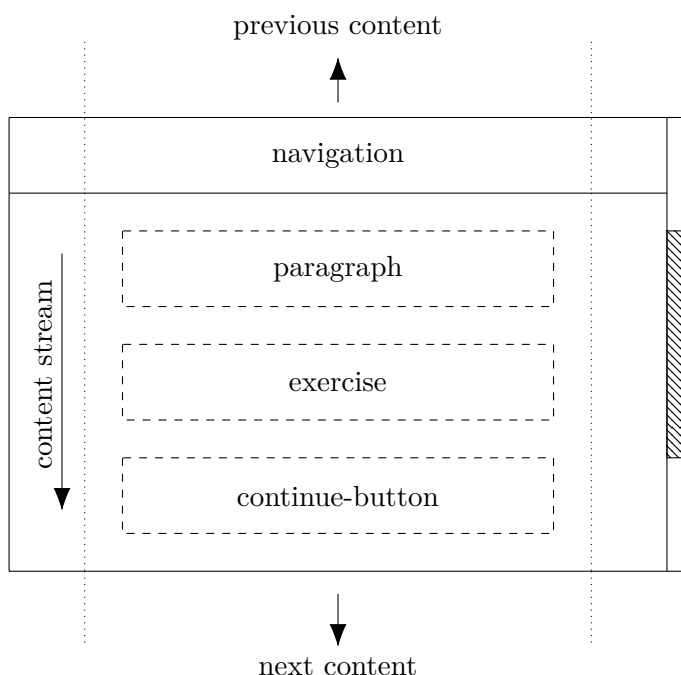


Figure 3.1: Content Stream

resembles that of a traditional book. Just as a book is usually read in a linear fashion from the beginning to the end, so is our content stream designed to be used in a linear way. In particular the user is not required to navigate away from the content stream as all content including interactive content and exercises are directly integrated into the content stream. We believe, that with this design students can more easily focus on the current content and that students are less likely to lose track of their learning objective.

It should be noted that internally the content of a course is not linear, instead it is stored as a tree with various branches (see Section 4.2). However, the presentation of a course to a particular student is linear as it reflects a specific path through the content tree. The path is determined by the answers which that student provided to our micro-tests.

Content Development. Currently the creation of new course content is API-driven rather than through a What You See Is What You Get (WYSIWYG) editor. This means that course content must be provided as JavaScript Object Notation (JSON) and can be imported via a RESTful interface. The decision creates a lot of flexibility for the creation of course content, that would not be easily achievable with a WYSIWYG editor. We have chosen this approach for two reasons, first it gives us greater flexibility with respect to the supported content types as courses must be “programmed” and not designed with a GUI. Second, it allowed us to focus our efforts on the implementation of the student frontend, because we did not have to create a graphical course editor. Since our software is currently designed to be used in the field of computer science (see Section 1.3), we

Content Type	Blocking	Interactive
Sections	No	No
Paragraphs	No	No
Definitions	No	No
Takeaway Points	No	No
Continue Buttons	Yes	Yes
Yes/No Exercise	Yes	Yes
Multiple Answer Exercise	Yes	Yes
Interactive Content	No	Yes

Table 3.1: Content Model

believe that having to “program” a course is an acceptable requirement.

3.3 Content Model

In order to achieve the above mentioned flexibility we have created a fine grained content model that provides a content type for each kind of element in the content stream. This allows us to add semantic meanings to the different content types, such as whether they are blocking the content stream or whether they are interactive.

Table 3.1 shows a list of content types and their different attributes. Each content type can be blocking or non-blocking, as well as interactive and non-interactive. Blocking vs. non-blocking refers to the fact that loading of additional entries is stopped if a blocking content type is encountered. The user must then interact with the element to advance in the content stream. Blocking elements therefore are always interactive. Interactive content types keep their own state, which gets updated when a user interacts with the element. The state of interactive elements is persisted in the user’s history as it changes and is automatically restored when the content is reloaded.

The root of this content model is a *Course* object. It contains information about the course itself, such as title, description and period, and it contains one or more section elements (see below). All other content types must be grouped into a section before they can be added to a course.

In this section we will only give a high level introduction of each content type and show how the different content types are rendered in the frontend. A more technical definition, that includes the individual data fields, can be found in the data model (see Section 4.2).

Sections. Sections are used to group related content together. Each section has a title and a description and can contain both nested sections as well as any other kind of content type. This allows us to build a hierarchy of content that is organized into sections and subsections. The top level sections of a course are presented in a table of contents to the user. Initially all top-level sections except the first section are disabled and the

CTL is a restricted subset of CTL* that permits only branching-time operators—each of the linear-time operators **G**, **F**, **X**, and **U** must be immediately preceded by a path quantifier. More precisely, CTL is the subset of CTL* that is obtained if the following two rules are used to specify the syntax of path formulas:

- If f and g are state formulas, then Xf and fUg are path formulas.
- If f is a path formula, then so is $\neg f$.

Example: $\mathbf{AG}(\mathbf{EF}p)$

Figure 3.2: Paragraphs with HTML Content and Math Mode Enabled

Definition 2 (Computation Tree)

Let $M = (S, R, L)$ be a Kripke structure. If we designate a state $s \in S$ as initial state, then M can be unwound into an infinite tree with s as its root. We will refer to this infinite tree as the **computation tree** of M . Paths in the tree represent possible computations or behaviors of M .

Figure 3.3: Paragraph as Widget of Type “Definition”

user must gradually work through the content to unlock access to further sections. This way we want to help the students to focus on the content that they are learning and not distract them with more advanced sections.

Paragraphs. Paragraphs (as shown in Figure 3.2) can be used either for normal text, HTML content, or as widgets with special markup. In both cases mathematical content can be rendered with a special math mode. Paragraphs are not interactive, they do not block the content stream and cannot be nested. To express mathematical content we support a simple L^AT_EX-like notation that can be embedded between $\$$ -signs [EA].

The two widget types we currently support for paragraphs are **definitions** (see Figure 3.3) and **takeaway points** (see Figure 3.4). A widget is like a normal paragraph, except that it has an additional title and a numbering that is unique per widget type. Widgets are rendered with a header and a body. The header has a highlighted background and contains the title, numbering and widget type. The widget content is shown in the widget’s body and supports the same content as a regular paragraph.

Continue Button. Continue buttons block the content stream and are used to request a manual interaction with the user in order to resume loading more elements. When the user clicks on a continue button it will remove itself from the content stream, such that it does no longer distract the user, and then causes the client to load additional content from the server. A continue button remembers its state, that is, whether it has already been clicked, in the user’s history. When the content stream is loaded again at a later

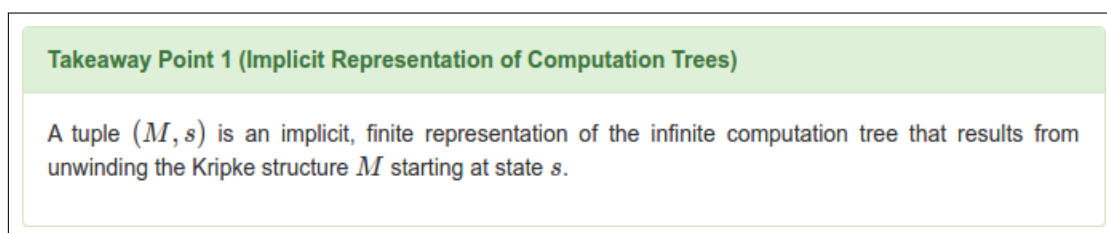


Figure 3.4: Paragraph as Widget of Type “Takeaway Point”

time, a continue button that has already been clicked will no longer block the content stream.

Exercises. The purpose of exercises in our content model is to test the students understanding of the content and to give feedback to both students and teachers. However, our exercises are not intended to be used for grading the student, but as a way of formative evaluation (see Definition 5). The exercises are therefore a part of our strategy to use the application for mastery learning.

Each exercise can be linked to a new stream of content depending on whether the exercise was answered correctly or incorrectly. For example, if the exercise was answered correctly a short confirmation text might be shown, and if the answer was wrong, a detailed explanation and possibly another exercise could be shown. The ability to show content in response to the user’s answer means that each student sees an individual content stream, as the stream content depends on the answers from each exercise.

We currently support two exercise types, these are yes/no exercises (see Figure 3.5) and multiple choice exercises (see Figure 3.6).

The yes/no exercise type consists of a title, and a simple question. It can be answered with yes or no and once it has been answered its background will be highlighted with a green or red color to give a visual feedback if the answer was correct. The title for the exercise is optional and will be hidden if it is empty.

The multiple choice exercise type consists of a title, a question and one or more answer candidates. A correct answer can consist of one or more of such answer candidates. After the exercise has been answered its individual answer candidates are highlighted with a green and red backgrounds to indicate which of the candidates was correctly selected. As with yes/no exercises the title of the exercise is optional and will be hidden if it is empty.

Once an exercise has been answered it stores this answer in the user’s history and cannot be answered again. We have deliberately implemented this feature to encourage students to think about their answers, instead of resorting to a trial-and-error way of answering exercises until they find the correct answer.

The exercises are directly embedded into the content stream and thus give the students a means to check their understanding of the content immediately after they learned

Exercise: Temporal Logic

Is an atomic proposition an ACTL* formula and an ECTL* formula?

Answer:

Yes ▼

Your answer is correct

Figure 3.5: Exercise Type with Yes/No Answer.

Exercise: Propositional Logic

Which of the following answers is syntactically right? Select all that apply.

Answers:

☐ $Q \wedge P$

☐ $Q \wedge \wedge P$

☐ $\wedge P$

☐ $(Q \wedge P)$

Check

Figure 3.6: Exercise Type with Multiple Possible Answers.

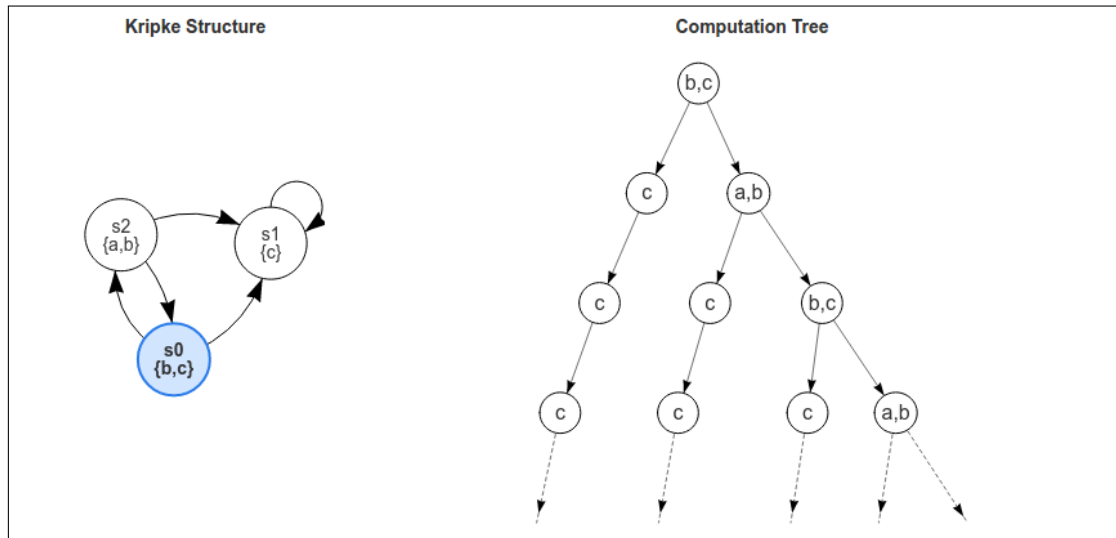


Figure 3.7: Interactive Kripke Structure with Corresponding Computation Tree [Hol16]

it. We believe that this is a way to give students early feedback and let them correct misunderstandings in their understanding, which could otherwise lead to confirmation bias [Plo93, Nic98] that can make it difficult for students to learn more content.

By embedding exercises as a way of formative evaluation directly into the content, we offer students a means to immediately check their understanding and thus a way to correct their understanding of a concept when they answer a question with a wrong answer. On the other hand, when a question is answered with a correct answer, it should give students more confidence and encourage them to continue with the next part of the content.

Interactive Content. The final element in our content model is the interactive content type. It allows course creators to integrate any kind of interactive content into their courses. The interactive content type is intended for small interactive “widget”-like content. For example, this could be an interactive graphic that reacts to user input, or an interactive source code listing that demonstrates how an algorithm works.

To integrate an interactive element, the course author must provide a library, which has to be integrated into the software at build time, such that it is available at run time. During the creation of the course content the author must then provide the name of an initialization function for each interactive type of content as well as some initialization data that can be used to customize the element. Interactive elements keep their own state in the user’s history, and restore their state when they are loaded. Examples of interactive content that has been integrated into a course through an external library are shown in Figure 3.7 and Figure 3.8.

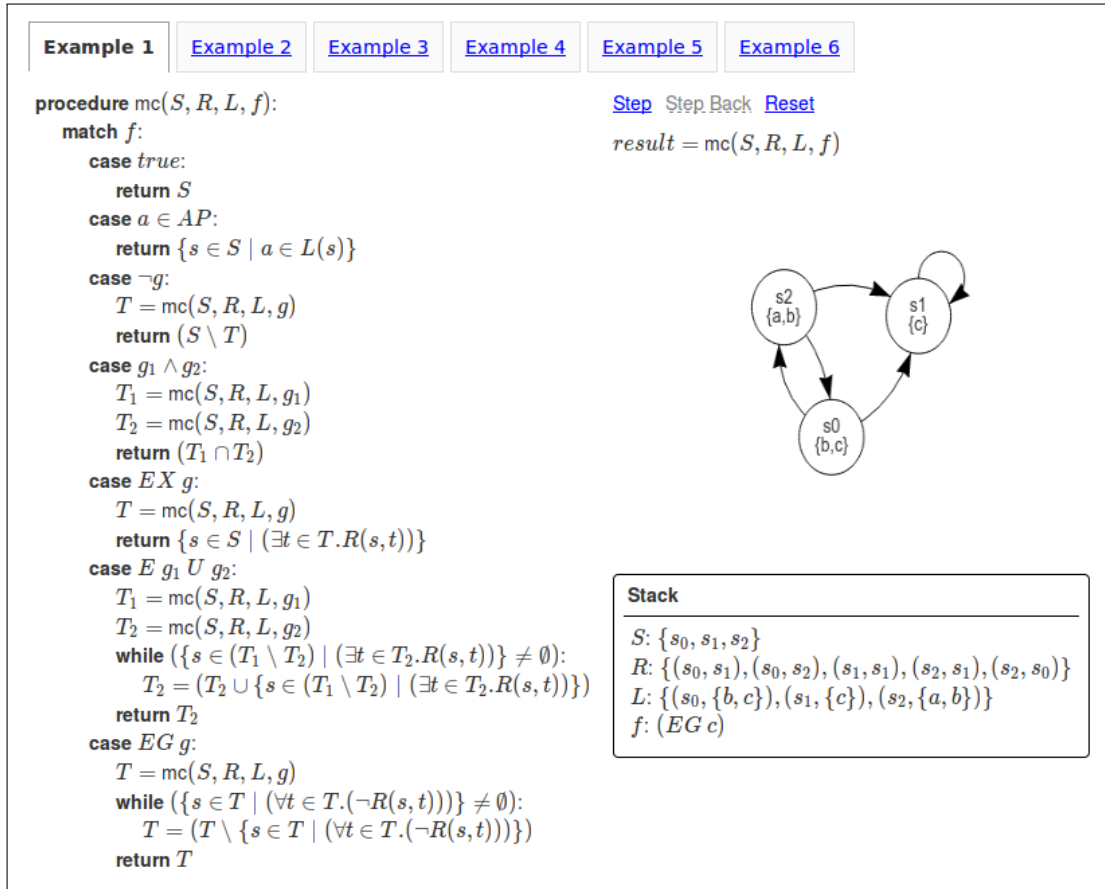


Figure 3.8: Interactive Pseudo Code Evaluator [Hol16]

3.4 Application Roles

The application has three distinct roles that show different content depending on the currently active role. By default each user has the *STUDENT* role, which is used to view and open courses. The second role is the *TEACHER* role, which is used by teaching assistants or professors to edit courses and view the statistics for each course. The third role is the *ADMIN* role, that is required to access the administration panel that can be used to manage users. Users have to login before they can use application. In the case that users forget their account password, we have integrated a password reset feature. This allows users to request a password reset email, which contains a link with a reset token. The link leads to a page where (if the token is valid) a new password can be chosen by the user.

On the top of the page a navigation bar can be found. It is always visible but contains different menu items and links depending on the state of the application. When no role is active (e.g. the user is not logged in) a home button, and an about page is shown.

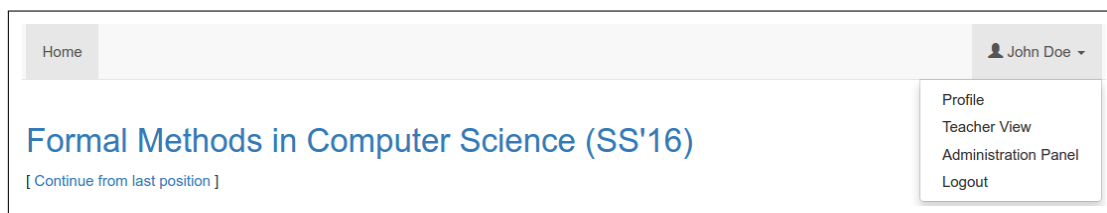


Figure 3.9: Home Screen with Navigation

When the user is logged in a home button is shown that leads to the home screen of the currently active role and a user menu is visible on the right side of the navigation bar (see Figure 3.9). The user menu allows access to the user’s profile where the password can be changed, and it contains a link to logout from the application. If the user has additional roles other than the student role, such as the teacher or the administrator role, then the user menu contains links to switch between these roles.

3.4.1 Student Role

After users have logged in, they are placed on the home screen of the student role (see Figure 3.9).

The content area of the students home screen is deliberately kept simple and only shows a list of currently available courses. For each course, the title and a link to the last position in the course is shown. Clicking on the course title itself opens the table of contents for that course. The table of contents is created from only the top level sections of a course (see Section 3.3), and allows the students to navigate to the respective content of each section, if it has already been unlocked. Below the title a “Continue from last position” link is displayed, that directly opens the content stream for the respective section and also positions the scroll buffer at the end of the content stream.

Currently we do not provide a way to sign up for a course. Instead any course in the system is available to each user.

3.4.2 Teacher Role

The teacher view shows an overview of the currently registered courses. It is possible to edit a course and view the statistics of each course.

Course editing is currently limited to changing the name, description and time period of the course. In the future we plan to extend this to the entries of a course, such that it is possible to add new entries and update existing ones.

The statistics view currently shows the progress that each user has made in the respective course, both in absolute numbers and in percentage. Progress is measured in entries seen compared to the total number of entries for the course. Due to the representation of courses as trees a student will never progress through the whole tree, but only through a

```
1 Example User 1;user1@example.com
2 Example User 2;user2@example.com
```

Listing 3.1: CSV Import Format without Header Line

particular path of the tree. For example when an exercise is answered correctly, then the student will never see entries that are linked with an incorrect answer and vice-versa. As a result students will never achieve 100% of progress of a given course.

3.4.3 Admin Role

We have implemented an administration panel, that can be used by users with the ADMIN role in order to configure the application. The currently supported features are:

- View the list of currently registered users.
- View the list of roles for each user.
- Add and remove the roles of users.
- Delete users.
- Add new users.
- Import multiple new users from a CSV file.

When the administration panel is opened the table with registered users is shown. This table shows the account details and roles of each user.

The import feature can be used to import multiple users from a CSV file. The CSV file must contain a list of the full name and email address of each user. Other details such as the username, the user id, the password and the roles will be configured automatically and cannot be provided in the CSV file. For each imported user the user id, a generic username and a random password will be generated. Generated user names are prefixed with “user” followed by a random number. Each user will receive the role STUDENT by default. If some user needs additional roles, these roles must be configured manually by editing the user after the import has finished.

The import dialog has a checkbox named (“Ignore header line”), by default this checkbox is unchecked and the import will assume that the CSV file does not contain a header line (see Listing 3.1). If the first line in the CSV file is a header line (see Listing 3.2), then this checkbox must be checked.

After a file has been selected for import through the “Browse” button, the file contents are shown in a table, where they can be reviewed before the actual import is started. The import is started with the “Add all” button. The import of each user requires the

3. INTERACTIVE LECTURE NOTES

```
1 COLUMN HEADER 1; COLUMN HEADER 2
2 Example User 1;user1@example.com
3 Example User 2;user2@example.com
```

Listing 3.2: CSV Import Format with Header Line

generation of a password hash, which can take up to one second. To avoid server timeouts the list of users is sent to the server in separate request of five users per request. The imported users are shown in another table, together with the generated user ids and usernames and roles. The table lists successfully imported users with a green color, while users for which an error occurred are shown in red. In the later case a message explaining the error is shown in the message column.

Software Architecture

In this Chapter we start with an overview of the software architecture and explain the impact of functional and non-functional requirements as presented in Section 3.1 on architectural choices and technical decisions in our software (see Section 4.1). Afterwards, we present the data model of our e-learning platform (Section 4.2), the architecture of the backend implementation (Section 4.3) and last but not least the architecture of the frontend implementation (Section 4.4).

4.1 Architectural Overview

The Interactive Lecture Notes e-learning platform has been implemented as a web-based application which is built as a distributed system with three separate modules. A *client module* that contains the frontend part of the application, a *backend module* that implements the business logic of the application and a *database module* which stores the application data in a relational database. An overview of our architecture is presented in Figure 4.1.

The database module stores all required data in a relational database schema that is currently implemented using MySQL. The database schema includes tables for the storage of our courses, the user accounts and the course history of each user (see Section 4.2). From an architectural and deployment point of view the database module is a separate module that contains the database schema for our application, however, it is not a buildable artifact because the database schema is implemented and managed by the backend module.

The backend handles database access, performs user authentication and authorization, implements the business logic and defines RESTful endpoints that are used by the client module. Communication between the database and the backend is handled via the

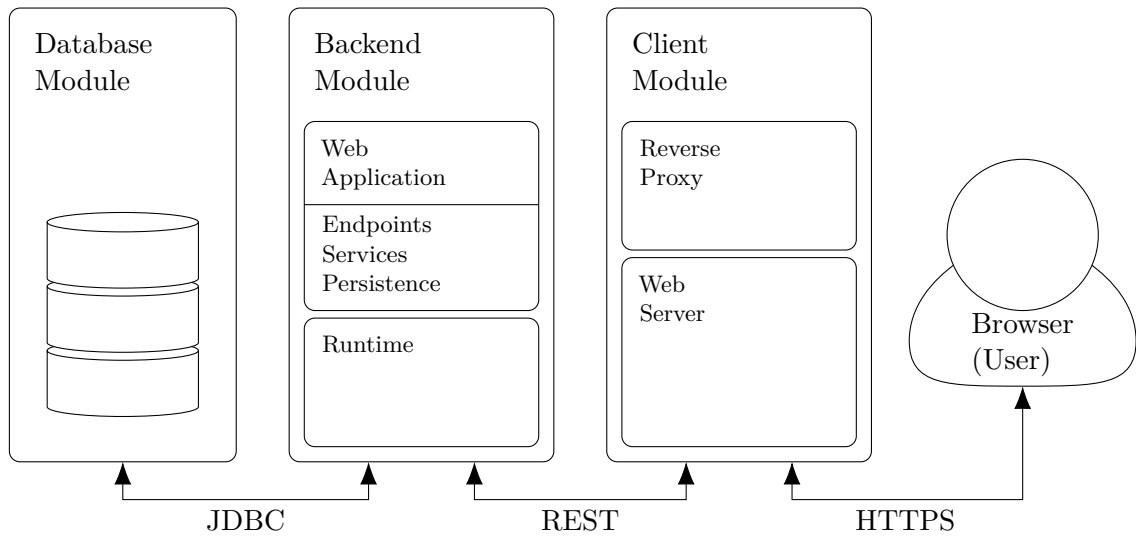


Figure 4.1: Architectural Overview

Java Database Connectivity (JDBC) interface and allows us to easily switch between various database providers should this become necessary in the future.

The client module consists of a web server that hosts the frontend part of the web application and additionally serves as a reverse proxy for requests that concern the backend module. The frontend part runs in the user's web browser and is implemented as a SPA. The frontend implements separate dashboards for each user role (e.g. student, teacher and administrator) as presented in Section 3.4.

The following list introduces design decisions that were influenced by the non-functional requirements presented in Section 3.1:

- **Scalability.** Each module is decoupled from other modules and modules communicate with each other through well defined interfaces. This has the advantage that each module can be packaged, deployed and updated independently from other modules and it enables better horizontal scalability of the whole application. It also allows us to deploy each module on separate hosts or even deploy multiple instances of the backend and client modules.
- **Maintainability.** By using established frameworks and build systems we lay the foundation for a high degree of extensibility and maintainability. Our backend (see Section 4.3) is built with Java¹. It uses the Java EE² development stack and is implemented as a web application using the Java Servlet specification as well as the Spring Framework.

¹Java (Version 8)

²Java Enterprise Edition (Version 7)

The client is implemented in JavaScript, HTML and CSS and built with the AngularJS framework. In addition the client module functions as a reverse proxy that forwards API request from the frontend to the backend module (see Section 4.4). This has the advantage that all the communication between browser, client and backend is routed through the client module. Therefore we only need to secure this public communication, while the internal communication can remain unsecured³. A second advantage of this design is that it is not necessary to configure Cross Origin Request Sharing (CORS) between the client and the backend, which would add additional complexity to the design.

- **Security.** Security aspects have been addressed in various parts of the software and concern all modules. In particular we are storing passwords in hashed and salted forms in the database. We are using a security framework (e.g. Spring Security) in the backend module that provides authentication and authorization features and sending all communication between the client and the backend over encrypted communication channels (e.g. HTTPS).
- **Documentation.** Last but not least we have provided documentation about building and deploying our software in Appendix A.
- **Deployment.** In addition to a manual installation of each module we also support a containerized deployment approach with Docker. In this Chapter we will focus on the architecture and implementation details of each module. Nevertheless, more information on the build system, containerization and deployment aspects can be found in Appendix A.

4.2 Data Model

The data model of the application can be categorized into two main data structures, each with several related tables; these are the `Course` table and the `User` table.

The `Course` table and its related tables contain all information that is required to represent a *course* in our application. The `User` table on the other hand stores information about user accounts and it has related tables, which store the history data that tracks the progress of the user through a course. Each table has an `id` field that uniquely identifies the rows of that table, for better readability, we have omitted this *id*-field from the diagrams below.

Course Data Model. The `Course` table as shown in Figure 4.2 is the root element for a course and contains the fields `title`, `period`, and `description`. Each course is linked to a tree of entries, which contain the actual course data. A course can have an arbitrary number of children, which again can have nested children thus forming a

³This assumes that all internal communication runs through private networks and is thus implicitly secure.

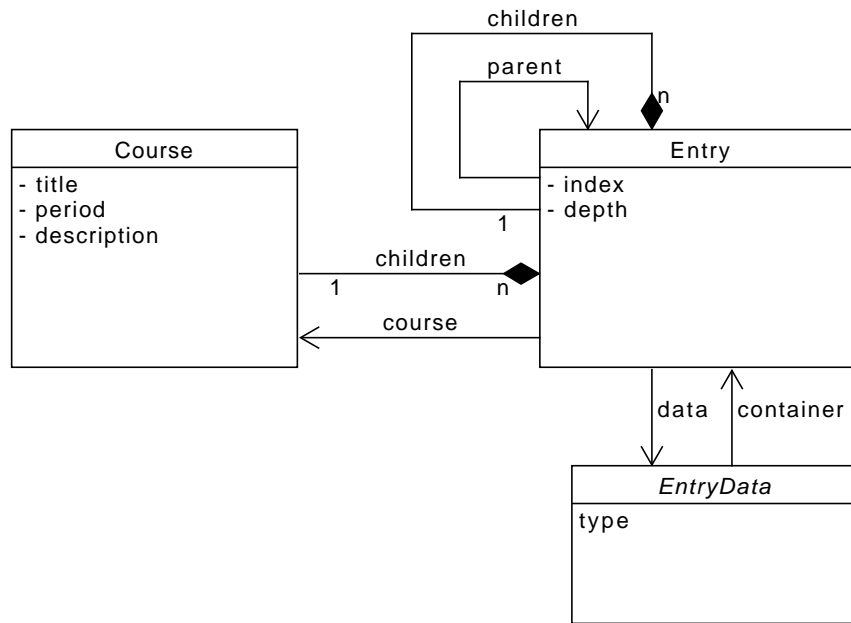


Figure 4.2: Data Model for Course and Entry Tree

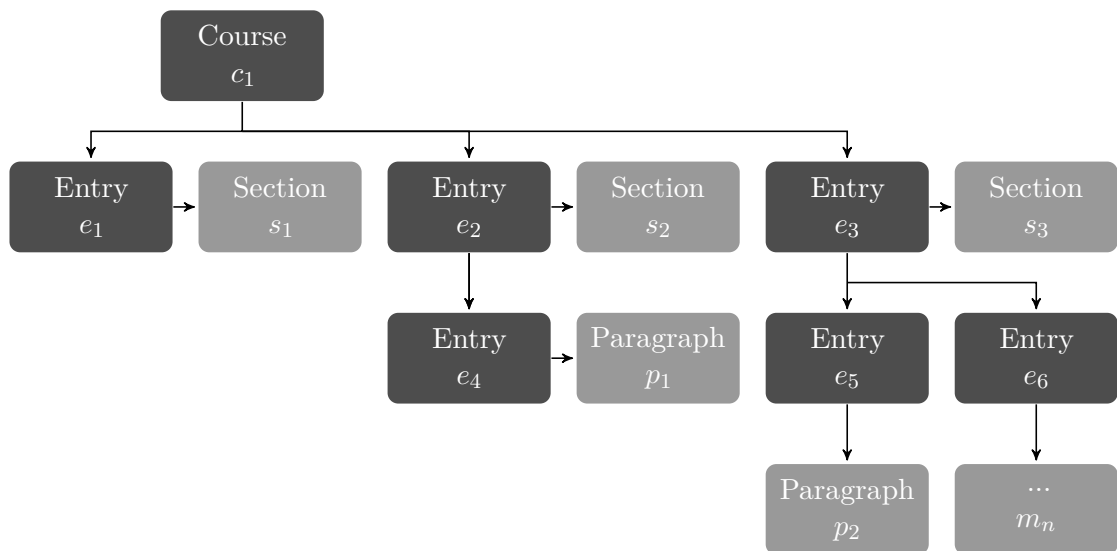


Figure 4.3: Example Course with Entry Tree (bright nodes are instances of **EntryData**)

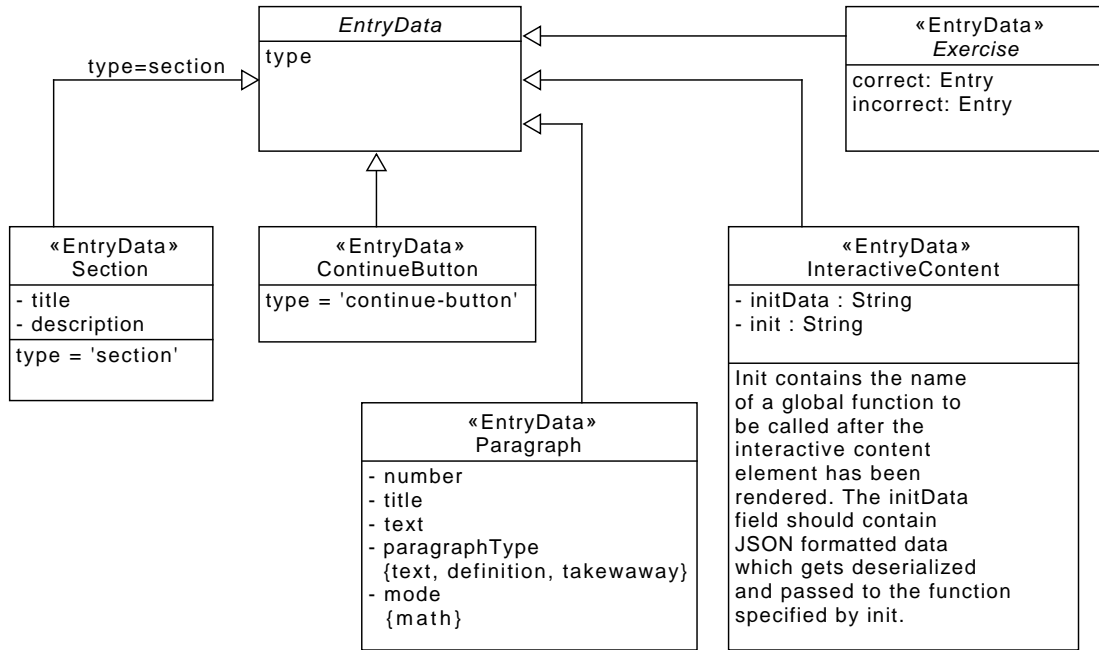


Figure 4.4: Data Model for Entry Data

tree of entries. This entry tree is stored in the `Entry` table as an adjacency list using a *parent-child* relationship (see Figure 4.2). To ensure a fixed order between entities that have the same parent entry, we store an index value for each entry. We also store the depth in the tree for each node, this information can be used by the frontend when the content is displayed in order to adjust the visual representation. When a course is loaded by the user its nodes are enumerated using a pre-order depth first search. We are aware that for large trees this can become a rather expensive operation. For future versions of our software we will consider other approaches such as moving the course information to a non-relational database or by using optimizations that can speed up the loading of entries.

An example of a course can be seen in Figure 4.3. The example shows a course c_1 , with several entries. When the tree of these entries is enumerated, the resulting order would be $e_1, e_2, e_4, e_3, e_5, e_6$. Each entry in turn links to an `EntryData` element whose type is one of the supported content types (see Chapter 3 in Table 3.1).

While the data model generally permits any entry type to contain additional children, the backend limits the creation of child entries to entries where the connected entry data is of type *section*. In addition a course can only contain entries where the connected entry data is of type *section*. This means for example that a paragraph cannot be an immediate child of a course.

The `EntryData` table (see Figure 4.4) is an *abstract* table and does not correspond to an actual table in the database. Each concrete subtype of the `EntryData` object has a separate table in the database, the `type` field is used to discriminate between the different subtypes.

The `Section` table serves as a grouping element for other `EntryData` elements and contains the fields `title` and `description`.

The `ContinueButton` table does not define any fields. Its only use is to serve as a blocking element for the content flow.

The `Paragraph` table stores the content for the different types of paragraphs, such as text paragraphs and widgets. The field `paragraphType` is used to discriminate between the different types. The currently supported values for this field are:

- `text`
- `definition`
- `takeaway`

The field `text` is used to store the paragraphs content. To enable the math mode, the `mode` field must be set to the value `math`. Two additional fields are `number` and `title`, which are only used for widgets to set the number and title of a widget.

The `InteractiveContent` table (see Figure 4.4) stores the initialization data for any kind of interactive content that is displayed in a course. The `init` field is used to store the name of a global JavaScript function that is called when the interactive content element is rendered. The function is responsible to return the HTML content for the interactive content. This function gets passed the data from the `initData` field as a parameter to allow for a customization of the generated content. It is the responsibility of the content creator to provide a corresponding JavaScript library and to ensure that it is deployed with the latest version of the application.

Finally, the `Exercise` table (see Figure 4.5) is again an *abstract* table that has two concrete subtypes. The first subtype is the `YesNoExercise` type, the second is the `MultiAnswerExercise` type. There are two common fields for each subtype, which are of type `Entry` and are used to link to a new subtree of entries. The field `correct` points to the subtree that is loaded when the exercise is answered correctly, similarly the field `incorrect` points to the subtree that is loaded when the exercise is answered incorrectly.

The `YesNoExercise` table has a field `title` for the title and a field `text` for the exercise question. An additional field `answer` contains a boolean value, which stores the correct answer. For `MultiAnswerExercises` there is an additional auxiliary table named `AnswerCandidate` that stores the possible answer candidates. Each answer candidate has a field `text` that stores the candidates question text and a field `correct` that stores whether that answer is correct.

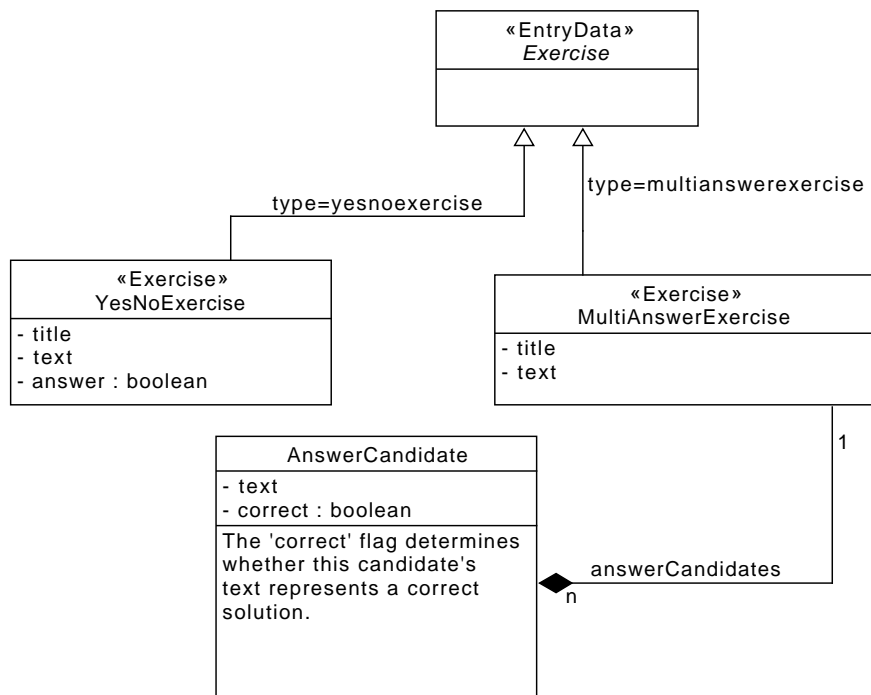


Figure 4.5: Data Model for Exercises

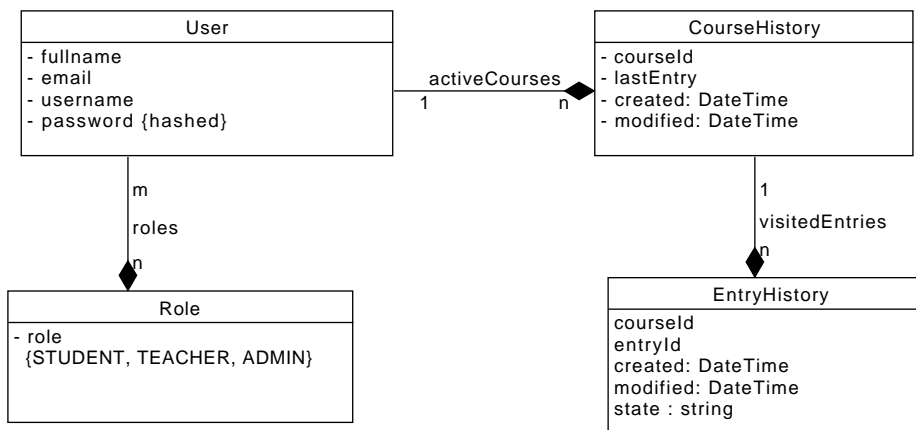


Figure 4.6: Data Model for User Data

User Data Model. The user accounts are stored in the `User` table. The fields `email` and `username` both uniquely identify a user and can both be used to login into the application. However the `username` is optional and a generic username will be generated if none is specified during user creation. In addition the `User` table has the fields `fullname` to store that actual users name, as well as a `password` field. We use the BCrypt password hashing function to securely store hashed versions of the passwords in the database. In order to authorize access to certain parts of the application, each user can be associated with one or more roles.

The `Role` table stores a list of currently supported roles in the application. The list of available roles is currently fixed and the supported roles are imported when the database is initialized during the first application start. The currently supported roles are `STUDENT`, `TEACHER` and `ADMIN`.

For each user we maintain a list of the currently active courses for that user in the `CourseHistory` table. These courses are shown to the user on the landing page, after the user has logged in. In order to display a shortcut to the last content that the user has seen (see Section 3), we also store a reference to the respective entry in the `lastEntry` field. Finally we store the `created` and `modified` timestamps for statistic evaluation.

There is currently no feature to enroll in a course, therefore we are automatically adding each available course to the course history of a user when the list of active courses is accessed.

The actual history information of a course is saved in the `EntryHistory` table. The fields `course` and `entry` are used to identify each `EntryHistory` entry with the corresponding entries of a course. The field `state` stores a string that encodes the state of the corresponding course's `EntryData` object. The format and contents of the state string depends on the actual `EntryData` object that corresponds to this `EntryHistory` object. Again the additional fields `created` and `modified` are used to store timestamps for statistic evaluation.

Schema Initialization and Versioning. Ensuring that the database and the backend have a common schema definition is important to ensure the reliable operation of our software. This is particularly important if schema changes are necessary during upgrades of the software. For this reason the database schema is managed by the backend (using the Flyway library⁴), which is responsible for performing the initial schema creation and for any further modifications of an existing schema. As a result, when the database schema needs to change, the necessary DDL statements must be added to the backend module and will subsequently be applied during runtime of the backend module. The backend module verifies the version of the database schema during application start, and performs an upgrade to the latest schema version if necessary. This information is stored in a special `schema_version` table in the database and ensures that the backend module

⁴<https://flywaydb.org/>

will only start if it can either find a matching schema version in the database or if it is able to upgrade the schema to the required version.

When a migration of the database schema is necessary it is the developers responsibility to create the necessary DDL statements. These DDL statements must be placed into a file named `V<number>_<description>.sql`, where `<number>` is an integer that is one higher then the currently active database schema version, and where `<description>` is a textual description of the changes. The file must be placed into the backend module under `src/main/resources/db/migration`, where it will be automatically picked up during runtime. A log message will be output at runtime indicating the current database schema version and whether any schema migration was performed.

4.3 Backend Module

The backend modules task is to handle persistence, provide RESTful services for the client and to manage the endpoint security and authorization for the different user roles in order to restrict access to certain endpoints for unauthorized users. The client application itself is decoupled from the backend and deployed as a separate module (see Section 4.4). For communication with the client the backend provides Representational State Transfer (REST) endpoints that are documented later in this section.

The backend module is implemented as a web application using Java 8 and is built using the Java EE Servlet API [jav]. We have used Spring Boot [sprb] and its associated libraries such as the Spring Framework [sprc], Spring Security [sprd], Spring Web MVC and others. By default Spring Boot generates an executable `jar` archive, which starts the application inside an embedded Tomcat servlet container. For readers who are not familiar with Spring, the Spring Boot documentation [spra] should serve as a good starting point to get familiar with these technologies. A detailed discussion of either the Java Servlet Specification, the Spring Framework or other libraries is outside the scope of this thesis and the reader is advised to refer to the respective reference documentations of these libraries for more detailed information.

4.3.1 Persistence

The persistence layer in the backend is implemented using an Object-Relational Mapping (ORM)-framework (e.g. Hibernate). This allows us to create the database schema from the class structure automatically by using Java annotations from the Java Persistence API (JPA). Annotations such as `@Entity`, `@OneToMany` and others are used to declare the mapping between Java classes and database tables (for more details we refer to the JPA specification). During development Hibernate can automatically create the database schema based on these Java annotations, but as was explained in the last section, for production deployments we use the Flyway framework to manage and version the database schema. All Java classes that correspond to tables in the data model can be found in the `org.masterylearning.domain` package.

```
1 @Entity
2 public class Course
3 {
4     @Id
5     @GeneratedValue
6     public Long id;
7
8     public String title;
9     public String period;
10    public String description;
11
12    @OneToMany (mappedBy = "course",
13               cascade = CascadeType.ALL)
14    public List<Entry> children = new ArrayList<>();
15 }
```

Listing 4.1: Hibernate Annotations in Java for Course Entity

A simple example that demonstrates the use of Hibernate annotations is shown in Listing 4.1. The `@Entity` annotation declares the `Course` class as a database entity and maps it to the corresponding database table named `course`. Primitive types (such as `Long` and `String`) do not need any additional annotations. The `id` field is annotated with `@Id`, which marks it as the primary key for this entity, we also use `@GeneratedValue` to ensure that Hibernate automatically generates a new identifier for each new course entity. Referenced entities and lists require the use of special annotations to setup the foreign keys between entity relations. In this case the list of `Entry` entities named `children` is annotated with the `@OneToMany` annotation. It declares through the `mappedBy` attribute, that each `Entry` has a reverse mapping named `course` that references back to the `Course` entity. It also specifies the `cascade` option, which declares that cascading Structured Query Language (SQL) operations (such as `DELETE`) cascade to the `children` list. For further information we refer to the Hibernate Manual [Hib].

Database access under Java EE is generally handled by the JPA. In addition to this API we also use the Spring Data library, which offers several features for convenient data access. In particular it is not necessary to write Data Access Object (DAO) classes, instead the Spring Data library provides an interface named `JpaRepository`, that can be used to generate common DAO functionality. By extending from this interface a DAO repository is automatically created to the respective domain object. Additional method signatures can be added (if required), in order to perform more specialized data queries. The `UserRepository` interface (see Listing 4.2) defines such a repository for the `User` entity and defines two additional methods named `getUserByUsername` and `getUserByEmail`. Spring Data then generates the implementation for this interface automatically.

All repository interfaces can be found in the `org.masterylearning.repository` pack-


```
1 @Repository
2 public interface UserRepository extends JpaRepository<User, Long> {
3
4     User getUserByUsername (String username);
5
6     User getUserByEmail (String email);
7 }
```

Listing 4.2: Spring Data Repository with Additional Methods

age.

4.3.2 Security

The backend module implements a security concept, which by default requires each request to the backend to be authenticated. Authentication must be performed via the HTTP Basic authentication method. Requests that do not need any kind of authentication, must be explicitly added to a *whitelist* in the security configuration. Authentication only ensures that the user has provided valid credentials for a user account. In addition to authentication we have implemented a simple authorization scheme that is based on application roles (see Section 3.4). When a request is made, the security implementation checks the role requirements for the particular endpoint and rejects the request if the authenticated user does not possess all of the required roles.

Endpoints which are whitelisted in the security configuration, include the password reset endpoints and a special bootstrap endpoint (see Section 4.3.3).

Password Reset. Users who have forgotten their password, can request a reset link via email. The link includes a secure token, that can be used in place of a password to authenticate the respective user. Both the endpoint to request the email (e.g. `/password/resetToken/`), as well as the endpoint that is used to provide the token and change the password (e.g. `/password/resetToken/:token`), must be accessible to unauthenticated users and are therefore whitelisted in the security configuration.

Bootstrapping Security. When the backend is started with a newly created empty database no users exist yet to authenticate incoming requests. But in order to create a user we must first authenticate somehow. It would be possible to solve this issue by creating a default username and password combination, but this can easily lead to security issues if the administrator forgets to change the default password. To avoid this class of security issue, we provide a special endpoint named `/bootstrap/users/`. If no user accounts exist in the database this endpoint can be used to create the first user account with a custom username and password. After the first user has been created any further invocation of this endpoint will be ignored.

4.3.3 Endpoint Reference

Access to resources is provided via REST endpoints. The endpoints are grouped by resource and implemented in several controller classes. This section contains the detailed specifications for each endpoint. The documentation of each endpoint includes a short description, the endpoint URL and Method (such as GET, POST, etc.) and the respective Java class and method of the endpoint implementation. If the method accepts any parameters, then the name and the respective type of the parameter is documented. Unless otherwise indicated, specified parameters are always required. Some endpoints accept data in the body of the HTTP request. For such endpoints the Body specification lists the name of a Java class that can be passed in the body of the request as a serialized JSON string.

An overview of all currently implemented endpoints and their corresponding controller classes can be found at the end of this section in Table 4.1.

Course Endpoints

Endpoint 1: Get Course List

Description:	Get the list of all courses in the database.
URL:	/courses
HTTP Method:	GET
Params:	none
Implementation:	CourseController::getCourseList

Endpoint 2: Get Course Overview

Description:	Get the table of contents of a course by its course ID.
URL:	/courses/:courseId
HTTP Method:	GET
Params:	courseId=[Integer]
Implementation:	CourseController::getCourseOverview

Endpoint 3: Update Course

Description:	Update one or more fields of a course object.
URL:	/courses/:courseId
HTTP Method:	POST
Params:	courseId=[Integer]
Implementation:	CourseController::updateCourse
Body:	{ CourseUpdateDto }

Endpoint 4: Get Full Course

Description:	Get a fully serialized course object including its entry tree.
URL:	/courses/:courseId/full
HTTP Method:	GET
Params:	courseId=[Integer]
Implementation:	CourseController::getFullCourse

Endpoint 5: Create Course

Description:	Save a complete course including its entry tree.
URL:	/courses
HTTP Method:	POST
Params:	none
Implementation:	CourseController::createCourse
Body:	{ Course }

Endpoint 6: Delete Course

Description:	Delete a course. This also cascades into the entry tree.
URL:	courseId/courses/c
HTTP Method:	DELETE
Params:	courseId=[Integer]
Implementation:	CourseController::deleteCourse

User Endpoints**Endpoint 7: Get Single User**

Description:	Get the currently authenticated user object.
URL:	/users/current
HTTP Method:	GET
Params:	none
Implementation:	UserController::getUser

Endpoint 8: Get User List

Description:	Get the list of all users in the system.
URL:	/users
HTTP Method:	GET
Params:	none
Implementation:	UserController::getUserList

Endpoint 9: Create User

Description:	Save a new user into the database.
URL:	/users
HTTP Method:	POST
Params:	none
Implementation:	UserController::createUser
Body:	{ CreateUserDto }

Endpoint 10: Import Multiple Users

Description:	Import multiple users into the database. Note that the body of this endpoint accepts a <code>CreateUsersDto</code> object, while the <code>createUser</code> endpoint accepts a <code>CreateUserDto</code> object.
URL:	<code>/users/import</code>
HTTP Method:	POST
Params:	none
Implementation:	<code>UserController::createUsers</code>
Body:	<code>{ CreateUsersDto }</code>

Endpoint 11: Delete User

Description:	Delete a user from the database.
URL:	<code>/users/:username</code>
HTTP Method:	DELETE
Params:	<code>username=[String]</code>
Implementation:	<code>UserController::deleteUser</code>

Endpoint 12: Update Roles

Description:	Update the list of roles that are assigned to a user.
URL:	<code>/users/:username/roles</code>
HTTP Method:	POST
Params:	<code>username=[String]</code>
Implementation:	<code>UserController::updateRoles</code>
Body:	<code>{ RolesDto }</code>

Endpoint 13: Change Password

Description:	Change the password of the currently authenticated user.
URL:	<code>/users/current/password</code>
HTTP Method:	POST
Params:	none
Implementation:	<code>UserController::changePassword</code>
Body:	<code>{ ChangePasswordDto }</code>

User History Endpoints

Endpoint 14: Get Active Courses

Description:	Get the list of active courses for the current user.
URL:	<code>/userHistory/getActiveCourses</code>
HTTP Method:	GET
Params:	none
Implementation:	<code>HistoryController::getActiveCourses</code>

Endpoint 15: Get Tables of Contents

Description:	Get the detailed course overview.
URL:	/userHistory/courses/:courseId
HTTP Method:	GET
Params:	courseId=[Integer]
Implementation:	HistoryController::getTableOfContents

Endpoint 16: Enumerate Entries

Description:	Enumerate the entry tree of a course.
URL:	/userHistory/courses/:courseId/enumerate
HTTP Method:	POST
Params:	courseId=[Integer]
Implementation:	HistoryController::enumerateEntries
Body:	{ EnumerationInDto }

Endpoint 17: Set Entry States

Description:	Save the current state of an entry in the users history.
URL:	/userHistory/courses/:courseId/entries/:entryId
HTTP Method:	POST
Params:	courseId=[Integer], entryId=[Integer]
Implementation:	HistoryController::setEntryState
Body:	{ EntryStateDto }

Statistics Endpoints**Endpoint 18: Get Statistics**

Description:	Get the progress statistic of all users for a course.
URL:	/statistics/courseHistory/:courseId
HTTP Method:	GET
Params:	courseId=[Integer]
Implementation:	StatisticsController::getStatistics

Password Endpoints**Endpoint 19: Get Password Token Per Mail**

Description:	Request a password reset token by mail.
URL:	/password/resetToken
HTTP Method:	POST
Params:	none
Implementation:	PasswordController::getPasswordTokenPerMail
Body:	{ ResetPasswordDto }

Implementation Class	Base URL	Description
CourseController	/courses	Base endpoint for course management.
UserController	/user	Base endpoint for user account management.
HistoryController	/userHistory	Base endpoint for user history.
StatisticController	/statistics	Base endpoint for statistics.
PasswordController	/password	Base endpoint for password resets.
InitializationController	/bootstrap	Base endpoint for bootstrapping the system.

Table 4.1: Endpoint Overview

Endpoint 20: Reset Password

Description:	Change the password by providing a password reset token.
URL:	/password/resetToken/:token
HTTP Method:	POST
Params:	token=[String]
Implementation:	PasswordController::resetPassword
Body:	{ ResetPasswordDto }

Initialization Endpoints**Endpoint 21: Create Initial User**

Description:	Create the initial user. This endpoint is only enabled, when no users exist in the database.
URL:	/bootstrap/users
HTTP Method:	POST
Params:	none
Implementation:	InitializationController::createInitialUser
Body:	{ CreateUserDto }

Endpoint Usage Example. As an example for the usage of the REST endpoints we demonstrate the creation of a new user account. The relevant endpoint for the creation of a new user account is *Create User* (see Endpoint 9). The endpoint accepts a *CreateUserDto* object in the request body that must contain the details for the new user account such as *full name*, *email address*, *username* and *password*. The username is optional and a generic username will be created if no username is provided.

Assuming that the backend is deployed at `localhost:8080`, we would sent an HTTP POST request to `https://localhost:8080/users` with a request body as shown in Listing 4.3. Assuming that the user was successfully created, the returned response would be similar as shown in Listing 4.4. If any error occurs during the request, the response

```
1 {  
2   "email": "john@doe.com",  
3   "fullname": "John Doe",  
4   "password": "secret"  
5 }
```

Listing 4.3: Request Body for Create User Endpoint.

```
1 {  
2   "userId": 5,  
3   "username": "user12345",  
4   "fullname": "John Doe",  
5   "email": "john@doe.com",  
6   "roles": ["STUDENT"]  
7 }
```

Listing 4.4: Response Body for Create User Endpoint.

```
1 {  
2   "userId": null,  
3   "message": "The username does not match the username requirements",  
4 }
```

Listing 4.5: Error Response for Create User Endpoint.

will contain an error message in the message property of the response body as shown in Listing 4.5.

4.4 Client Module

The client module contains the frontend application which runs in the browser, as well as the web server which is responsible to serve the various files for the frontend application and also serves as a reverse proxy to route asynchronous requests from the browser to the backend module.

The frontend application has been designed as a SPA in order to minimize requests to the web server and to avoid unnecessary page reloads. Traditional server-side web applications required a full round trip between the server and the user's browser for each request. A modern SPA however runs inside the user's browser without the need to refresh the page after the first initialization. When the page is first opened in the browser all necessary resources such as JavaScript and template files are downloaded and cached in the client. Any subsequent interaction that a user performs on the client website then only causes asynchronous JavaScript requests that fetch additional data

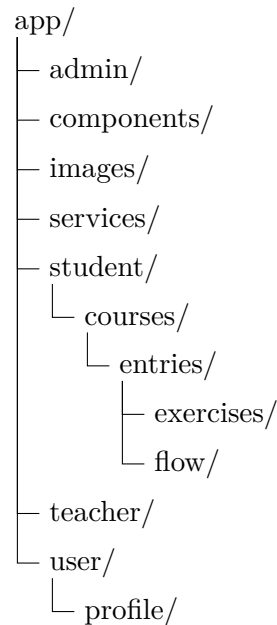


Figure 4.7: Folder Structure of the Client Module

from the backend, but that do not cause a reload of the initial web page.

4.4.1 Frontend Application

The frontend application is written with JavaScript, HTML and CSS. The JavaScript parts are implemented with the AngularJS 1 framework, while the HTML and CSS parts are implemented with the Bootstrap 3 framework and rendered using AngularJS directives and templates.

Code Structure. The client application is structured with respect to the different application roles of a user (see 3.4). The folders `admin/`, `teacher/` and `student/`, correspond to the *ADMIN*, *TEACHER* and *STUDENT* role. User specific functionality that is shared between different roles is placed in a separate `user/` folder. Each folder contains several HTML template files as well as JavaScript files that implement the respective parts of the application. Specific features of each role are placed into a folder that is named after the respective feature. For example, the student role, contains a subfolder named `courses/`, that contains the implementation to load and render the course content. Reusable components and services have been placed into the `components/` and `services/` folder respectively.

Navigation and Routing. In a traditional server-side rendered web application, navigation was performed via links to individual pages. Each page had the navigation parts such as menus and sidebars embedded into the HTML code that was returned by

the browser in addition to the requested content. In a SPA however, such traditional links are undesired since they cause a full reload of the web application. A SPA therefore requires an implementation that allows client-side navigation between different pages but that does not cause page reloads. Naive implementations could just use JavaScript to replace parts of the Document Object Model (DOM) with new content. However, such an implementation has several drawbacks. In particular, the location of different pages is not reflected in the address bar of the browser and therefore navigation with the browsers history buttons (e.g. back and forward) will not work. Additionally, such a design is not search engine friendly because search engines can only index the parts that can be statically determined, while JavaScript based navigation is only evaluated at run-time.

To solve this problem client-side navigation can either use HTTP fragment identifiers (e.g. `#some-identifier`) together with JavaScript or the more modern HTML5 history API (which may not be supported in all browsers). In the default configuration, AngularJS uses hash-based fragment identifiers. Two popular implementations exist for the AngularJS framework that are configurable to use either solution. These are the AngularJS native `ngRoute` and the third party plugin `ui-router`. We have chosen to use the more modern `ui-router` in our implementation because its state-based routing model offers more flexibility and makes navigation between nested pages easier.

Each page in our SPA is implemented as a template and is registered as state under a unique path and state name in the internal routing service of the application. A state can be activated when the user navigates to the URL that matches the corresponding states path, or via JavaScript by using the unique state name. When a state is activated the corresponding template is rendered and placed into the websites DOM replacing the content of the previous page.

4.4.2 Web Server and Reverse Proxy

As our application consists of separate modules that must communicate with each other the aspect of service discovery requires additional consideration. Ideally, the client module should not need to know specific information such as port or IP address of the backend and instead access the backend's services only via a generic URL. This avoids, that the client module must be configured specifically for each deployment in order to reach the backend service. Such a generic URL could be DNS based (e.g. `api.example.com`) or path based (e.g. `example.com/api`). A reverse proxy can be used to map such generic URLs to the actual location of the backend module. The term reverse proxy refers to the fact, that the proxy handles incoming rather than outgoing requests.

We have chosen to use nginx [Ngi] as both web server and reverse proxy in our client module. A path based mapping is used to resolve requests. Incoming requests that match the path `/api` are routed to the backend module. The reverse proxy functionality is configured with the `proxy_pass` directive and must be setup in the nginx configuration, as shown in Listing 4.6.

```
1     location / {
2         root /var/www;
3     }
4
5     location /api {
6         rewrite /api/(.*) /$1 break;
7         proxy_pass http://backend:8080;
8     }
9
10    location /user/password/resetToken {
11        rewrite /user/password/resetToken/(.*)
12            /#/user/password/resetToken/$1 redirect;
13    }
```

Listing 4.6: Nginx Reverse Proxy Configuration

In the configuration three locations are necessary. The root location (/) points to a local folder on the server, that contains the source files of the frontend application. The backend location (/api) is needed to redirect all requests which arrive on URLs starting with /api/ to the backend module. In addition a rewrite of that URL is performed to drop the /api prefix, as this prefix is only required for the process of matching URLs in the reverse proxy. Finally, the third location rewrites requests to /user/password/resetToken to the fragment identifier based URL format that is required for routing in the frontend application. This additional rule allows to include links without a fragment identifier in password reset e-mails. Without the last rule these links would not open the correct page in the browser.

Evaluation

We evaluated our e-learning implementation in the context of the course “Formal Methods in Computer Science” at the Technical University of Vienna. The course consists of four individual parts that cover different aspects of formal methods. A total of three evaluations were conducted in the context of the fourth part of the lecture which deals with model checking. The first evaluation took place during the summer term of 2016. A second evaluation was conducted in the winter term of 2016. Finally, a third evaluation was conducted in the summer term of 2018.

To avoid ambiguities in the remainder of this section we will refer to the physically held lectures as the *offline course* and we will refer to the course that was available in our e-learning system as the *online course*. For the online course we used course material created by Andreas Holzer [Hol16] that covers the topic of *temporal logic*. Temporal logic is one of several topics that are covered in the model checking part of the offline course.

During the first evaluation in the summer term of 2016 students were offered to volunteer for an evaluation of our implementation and were provided with user accounts on request. During the second and third evaluations in the winter term 2016, and summer term 2018, we automatically imported all users who had enrolled in the offline course into our e-learning platform. During each evaluation we collected student feedback via anonymous questionnaires as well as statistical usage data through our e-learning implementation.

In Section 5.1 we will first present the questions and result from the three surveys together with our interpretation of the collected results. In Section 5.2 we will then present statistical data of the students course progress that was collected on the e-learning platform itself.

Evaluation Period	Number of Submissions
Summer Term (ST) 2016	4
Winter Term (WT) 2016	9
Summer Term (ST) 2018	17
Total Responses	30

Table 5.1: Number of Submissions per Evaluation.

5.1 Survey

The survey, which we conducted during each evaluation, contained questions with respect to usability of our implementation as well as the content and structure of the online course. We also asked questions about the context of each student with respect to the offline course, such as whether a student had taken the course in a previous semester.

We conducted the survey anonymously using Google Forms [Goo18]. The number of submissions for each survey is shown in Table 5.1. Due to the small amount of feedback in the first two evaluations, we decided to conduct a third evaluation in the summer term of 2018. For this third evaluation we offered students the chance to win five Amazon vouchers worth 20,- Euro each. This helped to increase the number of submissions during the third evaluation.

To keep the results between the evaluations comparable, we used the same set of questions in each survey. The questions were structured into the following three categories:

- Overall Experience.
- Detailed Feedback on E-learning Tool.
- Lecture Context.

For the third evaluation, we added two additional questions to the Section *Lecture Context*.

In the remainder of this Section we will present the questions in each category and give an interpretation of the results. There were a total of 32 questions (30 in the first two evaluations). For each question we will state the number of the question in the questionnaire, the type of question (e.g. ordinal, percentage, multiple choice, etc.), the text of the question and the results of each evaluation. For questions of type ordinal we also include a definition of the minimum and maximum values in the value range. All columns which contain average values (avg.) were computed using an arithmetic mean.

5.1.1 Overall Experience

In the category *Overall Experience* we asked four questions to collect general feedback about our e-learning implementation (see Table 5.2 and Table 5.3).

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
1	Ordinal (1-5), higher is better	I have an overall positive impression of the e-learning tool: (1=“I don’t agree“, 5=“I fully agree“)	4.00	4.22	4.76
2	Ordinal (1-5), higher is better	I would like to see such an e-learning tool to be used in more courses (including other parts of FMI): (1=“I don’t agree“, 5=“I fully agree“)	5.00	4.22	4.76
3	Ordinal (1-5), higher is better	If I had to choose between a lecture that is completely supported by such an e-learning tool (i.e., content and interactive elements for the whole course are available) and a lecture that is supported by video recordings of the lecture, I would chose the lecture with the e-learning tool: (1=“I don’t agree“, 5=“I fully agree“)	3.00	3.89	4.00

Table 5.2: Overall Experience.

The answers in Table 5.2 show that in all three evaluations the students had an overall positive impression of our e-learning implementation and that they would like to see the use of such a tool in more courses.

The responses to the third question vary between the evaluations. In the first evaluation we received an average result of 3.0, which shows that students had no clear preference between either a video-based content model or our dynamic and interactive content model. In the second and third evaluation the results shifted to 3.89 and 4.0 respectively. This means that students now showed a slight preference for our content model rather than for a video-based content model. In any case these results confirm our assumption that not relying on video-based content types is a valid approach for modern e-learning systems in the context of higher education.

In the fourth question, we asked students who did not complete the online course for their main reason to quit the course. This question was designed as a multiple-choice question. Besides four default choices students could also enter a custom reason. In the first evaluation (see Table 5.3) one point of feedback was, that the end of the course was not explicitly marked in the course material, and thus it was not clear if the student had reached the end of the online course or if loading further content had failed. We have changed this aspect for the third evaluation of the course. Now an explicit statement is shown when students have reached the end of the online course.

For our second and third evaluations, we did not receive any answers to this question.

Question 4: *If you did not complete the whole e-learning course, please state your main reasons.*

Type	Reason	ST 2016	WT 2016	ST 2018
Boolean	The software did not meet my expectations	0	0	0
Boolean	I had problems understanding certain parts	0	0	0
Boolean	I expected more interactive content	0	0	0
Boolean	The content was not helpful for the exercise sheet	1	0	0
Text	I am not sure if there was no more content available (last topic was 1.3.5) or if it was a bug in the system that denied to continue?	1	0	0

Answers of type *Text* show additional answers that were entered by students and which were not present in the original set of answers.

Table 5.3: Student Reasons for Aborting the Online Course.

5.1.2 Detailed Feedback on E-Learning Tool

In the second part of the evaluation, we asked students for feedback about our e-learning implementation. The questions were split into the following four subcategories:

- Presentation of Content.
- Interactivity.
- General Feedback.
- Platform.

Presentation of Content. We were interested in individual feedback about different aspects of our implementation. In particular, we wanted to know how users think of our content model and how they perceived the incremental presentation of the content.

As Table 5.4 shows our content model was particularly well received. Students liked the incremental aspect of the content presentation and felt that it helped them to concentrate on the current content (see questions 5 and 6). They also confirmed our hypothesis that the interleaving of content and exercises helped them to check if their understanding of the content was correct (see question 7). These results are a strong indicator that micro-tests and dynamic models are indeed an effective means to achieve instant feedback.

Question 8 in Table 5.4 asked about the length of *content chunks*, which were separated by continue buttons. An answer of 1 indicates that chunks are too short, while an answer of 5 indicates that chunks are too long. The results have an average of 3.0, 2.67, 2.82 respectively, which is very close to the medium value of the range (e.g. 3.0). These results show that our approach to let the students control the loading of new content is well

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
5	Ordinal (1-5), higher is better	I liked the incremental presentation of content: (1=“I don’t agree“, 5=“I fully agree“)	5.00	3.56	4.35
6	Ordinal (1-5), higher is better	The incremental presentation helped me to concentrate on the currently visible content: (1=“I don’t agree“, 5=“I fully agree“)	4.75	3.89	4.59
7	Ordinal (1-5), higher is better	Course material followed by exercises helped me to quickly check my understanding of the presented material: (1=“I don’t agree“, 5=“I fully agree“)	4.75	4.11	4.59
8	Ordinal (1-5), medium is better	We use "continue"-buttons to incrementally load more content. The sections between each continue button were: (1=“Too short; too many continue buttons“, 5=“Too long; could have been more continue buttons“)	3.00	2.67	2.82

Table 5.4: Detailed Feedback on ‘Presentation of Content’

received and that we have found a good balance for the length of content chunks and the amount of continue buttons.

Interactivity. When it comes to interactivity and exercises we see additional confirmation for our approach. The results in Table 5.5 show that students found the interactive elements helpful to understand the content, but would have liked more such interactive elements and exercises.

In questions 9 and 11 we asked about the amount of interactive elements and exercises. For both questions we received average values below 3.0, which show that the amount of interactive elements and exercises was slightly to low. However, the results of question 10 with average results of 4.5, 4.44 and 4.41 respectively show that the interactive elements clearly helped students to understand the course content.

This interpretation is supported by the results of question 13, which shows that students clearly favor the ability to evaluate their understanding of the content through the interleaved exercises, and would prefer more such opportunities. In addition, when asked in question 12 about a possible separation of content and exercises students show strong disagreement. These results show that our approach to interleave exercises and content was the correct decision. Together these results are another indicator that micro-tests and dynamic models are an effective means to provide instant feedback.

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
9	Ordinal (1-5), medium is better	The amount of interactive elements (not including exercises) was: (1=“Too low“, 5=“Too much“)	2.50	2.38	2.65
10	Ordinal (1-5), higher is better	The interactive elements (e.g., the interactive computation tree, or the CTL model checking algorithm) helped me understand the course material: (1=“I don’t agree“, 5=“I fully agree“)	4.50	4.44	4.41
11	Ordinal (1-5), medium is better	The amount of exercises was: (1=“Too few“, 5=“Too many“)	2.25	2.38	1.94
12	Ordinal (1-5), higher value equals more agreement	I would prefer if exercises were separated from the content: (1=“I don’t agree“, 5=“I fully agree“)	1.25	1.67	1.65
13	Ordinal (1-5), higher is better	I would like more exercises interleaved in the content, so I can check myself more frequently: (1=“I don’t agree“, 5=“I fully agree“)	4.00	3.33	4.25
14	Boolean (Yes/No), percentage shows yes answers	I would like the ability to reset the answers to my exercises so I can try again:	100%	100%	70.6%

Table 5.5: Detailed Feedback on ‘Interactivity’

Finally, question 14 shows a conflicting result between our design choice to lock the answer of an exercise and the students wish to reset their answers. In the first two evaluations all students showed a strong preference for the ability to reset their answers for exercises. However, during the third evaluation this value dropped to 70.6% which shows that a minority of students agrees with our design choice.

Nevertheless, we want to emphasize that this decision was made deliberately. As discussed in the context of mastery learning the results to such questions should be seen as formative evaluation (see Definition 5) and does not count towards a student’s grade. By locking the answer to a question the feedback that students receive for their answer will remain visible throughout the course and gives students the opportunity to review their mistakes and to see where they need further practice.

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
15	Free-form (Number of Answers)	I found the following aspects of the tool helpful (please provide the most helpful aspect first): (see Appendix B for individual answers)	1	4	14
16	Free-form (Number of Answers)	I found the following aspects of the tool NOT helpful (please provide the most unhelpful aspect first): (see Appendix B for individual answers)	1	4	8

Table 5.6: General Feedback

General Feedback. In this subcategory we asked students to provide general feedback about aspects of our implementation which they found most helpful and not helpful. The full text of all answers can be found in Appendix B. Below we summarize the most relevant responses of both questions.

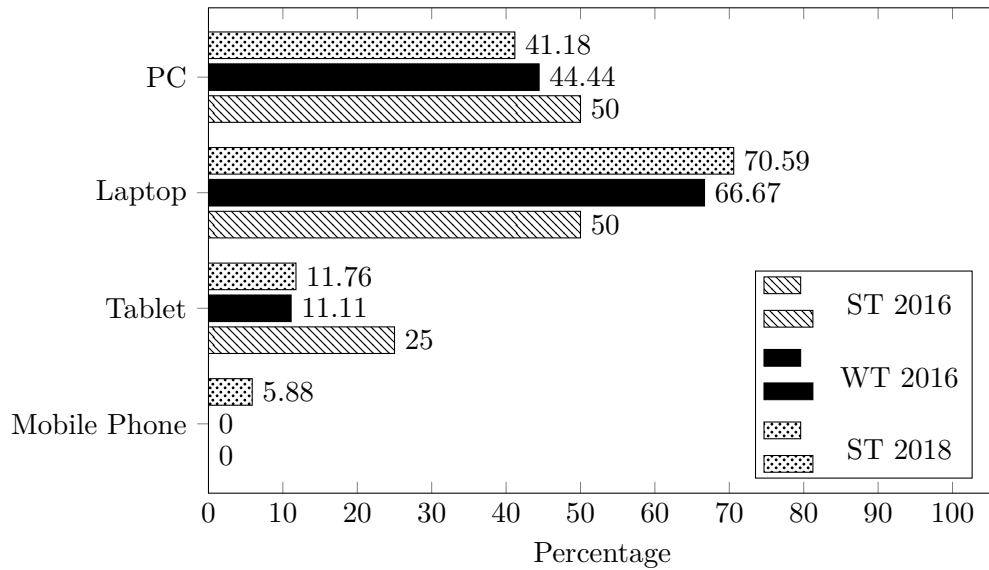
With respect to helpful aspects, many students stated that they found the interactive content types such as the pseudo-code evaluator helpful (7 responses), followed by exercises (6 responses) and the incremental presentation of content (4 responses). Other positive feedback included the content and presentation in general (5 responses). One student mentioned the *continue-from-last-position* feature as helpful and another one the takeaway points and definitions. The mastery learning aspect of the application, which allows learning at the students preferred speed, was also mentioned as helpful by one student.

With respect to aspects that were not helpful we received very heterogeneous feedback. Some students criticized the lack of a progress indicator (2 responses) and missing explanations for correctly answered exercises (2 responses). Some students criticized aspects of the content (2 responses) as well as the lack of a more detailed table of contents (1 response). Another response criticized the scrolling behavior when loading new content.

Finally, in the first two evaluations some exercises did not offer any explanation when an incorrect answer was given. This lack of explanation was criticized by two students. Due to this critique all exercises were reviewed and extended to provide a detailed explanation if the answer to an exercise was wrong [Hol16]. As a result of this change, explanations for incorrectly answered exercises were mentioned as helpful during the third evaluation (2 responses).

Platform. In this subcategory we asked students on which kind of platform they used our tool. In this context platform refers to the type of device or form factor, rather than

Question 17: *On which devices did you use the tool?*



This question type allowed to give multiple answers. Therefore the total percentage of all all answers is higher than 100%.

Figure 5.1: Platform Distribution

the operating system or browser vendor. We were interested in the different platforms that each student used to access our tool as well as the primary platform from which students mainly used our tool.

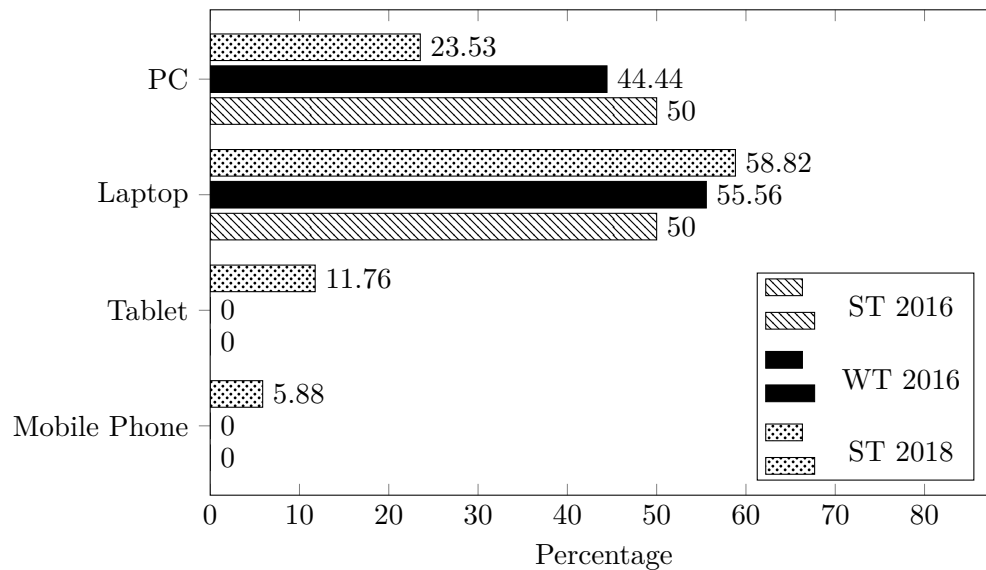
The answers in Figure 5.1 and 5.2 show that most students used the tool on one device only. For most students this device was therefore the primary platform on which they used the tool. The majority of users used the tool on their Laptop, closely followed by PCs, and rarely on a tablet. Mobile phone usage was only reported during the last evaluation in the summer term 2018 by one student.

We also asked how students perceived the speed of our tool. Feedback with respect to the speed of our implementation is very important to be able to judge if further optimizations are necessary in the future, to avoid unnecessary and premature optimizations. The responses as shown in Table 5.7 show that all students were very satisfied with the speed and that for the time being further optimizations are not necessary.

5.1.3 Lecture Context

In the third part of the evaluation (see Table 5.8) we asked questions about the context of the offline course rather than our e-learning implementation. We were interested in the different kinds of resources that students used to prepare for both the lecture and the exam and how they rated the quality of the available material. We also asked whether or

Question 18: *On which devices did you primarily use the tool?*



This question type only allowed to give one answer.

Figure 5.2: Primary Platform Distribution

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
19	Ordinal (1-5), higher is better	How did you perceive the speed of the tool: (1=“Slow; laggy“, 5=“Fast; responsive“)	4.50	4.89	4.76

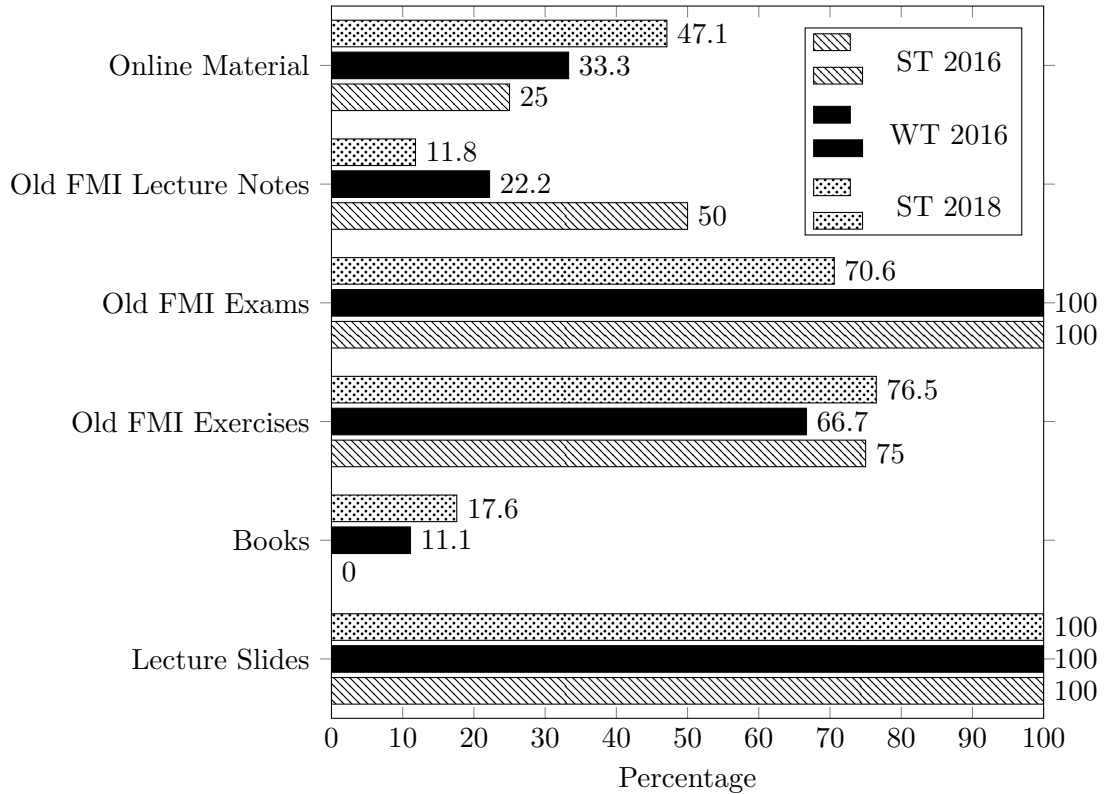
Table 5.7: Implementation Speed (Subjective User Experience)

not students already had the required logical and mathematical background when they attended the lecture, which is an important factor to succeed in the lecture.

In the third evaluation during the summer term 2018 we added two questions to determine the availability of video content and interactive tools at the Technical University of Vienna. With average values of 2.18 for video content (see question 20) and 1.41 for interactive study tools (see question 21) the results show that students did not have many encounters with such tools in other lectures at the Technical University of Vienna.

The answers to questions 22 and 23 show that students relied mostly on the official lecture slides, as well as old exams (see Figure 5.3 and Table 5.8), and that they rarely used books. The use of other materials includes the Kripke Builder [Pan] and official course materials, a student maintained Wiki called “VorlesungsWiki“ [Fac], as well as Google, Wikipedia

Question 22: *When preparing for the exercises and the exam of part 4 of FMI (i.e., model checking), I used the following materials:*



This question type allowed to give multiple answers. Therefore the total percentage of all all answers is higher than 100%.

Figure 5.3: Lecture Materials used by Students to Prepare for Exam

and StackExchange. One student mentioned the books *Model Checking* [CGP99] and *Principles of Model Checking* [BK08].

We also asked about the quality of the available lecture material. The responses from the evaluations have average results of 3.5, 3.25, 3.82 respectively. This shows that students are generally satisfied with the available lecture materials but there is still room for improvement.

Finally students were asked if they had the required prerequisites in mathematical proofs. The answers to this question have a very high variance. Some students state that they did not have the required prerequisites at all, while other students state that they had all of the necessary prerequisites.

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
20	Ordinal (1-5), higher is better	How often have you encountered video content (such as lecture recordings) in any of your lectures at the TU Vienna? (1="rarely", 5="very often")	N/A	N/A	2.18
21	Ordinal (1-5), higher is better	How often have you encountered interactive study tools or e-learning software in lectures at the TU Vienna? (1="rarely", 5="very often")	N/A	N/A	1.41
23	Free-form (Number of Answers)	If you used material other than lecture slides please provide more information (e.g., book title, or URL of online material): (see Appendix B for individual answers)	2	1	6
24	Ordinal (1-5), higher is better	How good would you rate the quality of the material made available for part 4 of FMI (excluding the e-learning tool): (1="Very bad", 5="Very good")	3.50	3.33	3.82
25	Ordinal (1-5), higher is better	I had the required prerequisites in logic and mathematical proofs before I attended the FMI lectures: (1="I don't agree", 5="I fully agree")	3.25	2.67	3.12

Table 5.8: Lecture Context

Lecture Attendance. In this subcategory we asked several questions related to lecture attendance in the offline course (see Table 5.9). The questions 26 and 27 asked about the lecture attendance for both the whole lecture in general, and the lectures of part four in particular. Question 28 asked about reasons for not attending the lecture and question 29 asked whether students had attended part four of the lecture in earlier semesters.

During the first evaluation attendance rates were rather high with 91.67 and 79.17 percent and students cited time constraints as their main reason for not attending all lectures. During the second evaluation attendance rates dropped to 59.26 percent but at the same time 66.67 percent of all students had taken part four of the lecture in previous semesters. During the third evaluation the attendance rates were at 69.61 and 67.65 percent respectively. Only 5.88 percent of the students had attended part four of the lecture in previous semesters, but most students cited time constraints as their main reason. In addition, three students cited the availability of lecture recordings as their

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
26	Percentage (0-100), higher is better	How often did you attend the FMI lectures:	91.67%	59.26%	69.61%
27	Percentage (0-100), higher is better	How often did you attend the lectures of part 4:	79.17%	59.26%	67.65%
28	Free-form (Number of Answers)	If you did not attend all lectures of FMI, what were the most important reasons for that? (see <i>Appendix B</i> for individual answers)	2	5	15
29	Boolean (Yes/No), percentage shows yes answers	Did you attend part 4 of FMI (model checking) in earlier semesters?	50.00%	66.67%	5.88%

Table 5.9: Lecture Attendance

reason for not attending the lecture.

The answers to question 30 (see Table 5.10) show that students already had some initial understanding of model checking when they first used the e-learning tool. The reason for this is likely that the e-learning tool was presented to students towards the end of the fourth part of the offline lecture, when students had already learned about most of the model checking content.

Finally, in question 31 we asked if students had attended the lecture *Computer Aided Verification (CAV)* which covers the topics of model checking in more detail (see Table 5.10). During the first two evaluations no students had attended CAV. During the third evaluation 17.64 percent of the students had attended or was attending CAV.

Further Comments. In this final subcategory (see Table 5.10, Question 32) we gave students the chance to provide an informal comment to address aspects that were not covered in other questions of the survey.

There were four responses in the summer term 2016, one response in the winter term 2016 and six responses in the summer term 2018. The full text of the original answers can be found in Appendix B. Below we present a short summary of the answers and refer to individual answers by their number in the Appendix.

The responses were very positive and most students expressed their appreciation for

#	Type	Question	ST 2016 (avg.)	WT 2016 (avg.)	ST 2018 (avg.)
30	Ordinal (1-5), higher is better	How well did you understand the Model Checking content of part 4 before you started using the e-learning tool? <i>(1="I did not understand the content", 5="Everything was completely understood")</i>	3.25	3.22	3.18
31	Boolean (Yes/No), percentage shows yes answers	Do you or did you attend the lecture "Computer Aided Verification (CAV)"?	0.00%	0.00%	17.64%
32	Free-form (Number of Answers)	Further comments: <i>(see Appendix B for individual answers)</i>	4	1	6

Table 5.10: Prerequisites and Further Comments

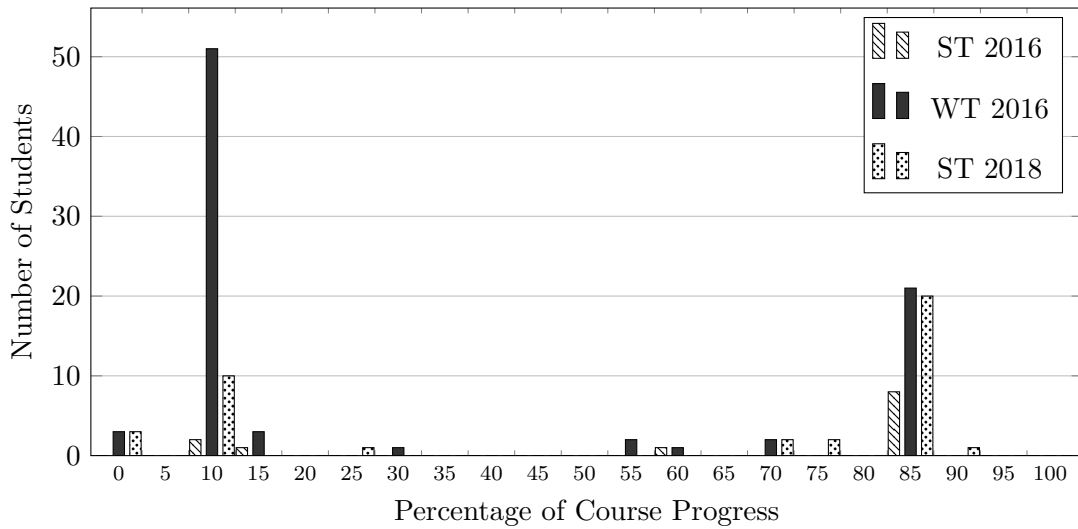
our e-learning platform. One student liked the idea of the platform, but criticized that the content was not helpful to prepare for the exam (see Quote 67). Several students included requests for additional features such as a table of contents (see Quote 73) or the ability to add their own notes to the text (see Quote 69). Several students asked for additional content such as *bi-simulation* (see Quote 72 and 74) as well as video-content (see Quote 69).

5.2 Statistics

We collected and analyzed usage statistics from the e-learning tool. On the one hand the statistics show how much course progress the students have achieved allowing us to see how many students worked through the complete course content. On the other hand the result shows the different feedback loop durations that the students achieved with our tool while studying the course content. The statistical data was collected directly in our e-learning tool and is available for each of the three evaluations.

5.2.1 Course Progress

The bar chart in Figure 5.4 shows how much progress students achieved in the online course. For better readability we have filtered the chart to exclude users who never logged into the system. The chart shows the percentage of course content that each student has



For implementation reasons it is not possible for students to achieve a completion rate of more than 90%. The reason for this is that students cannot view both the correct and incorrect branches of an exercise.

Figure 5.4: User Progress

completed on the x-axis and the number of students who reached that percentage on the y-axis.

We emphasize that for implementation reasons it is not possible to achieve 100% progress in the course. This is due to the fact that we count the number of content entries that each student has visited; currently it is not possible to visit both the correct and incorrect nodes of an answer.

The maximum percentage that students can practically achieve in our online course is therefore 85% to 90% depending on the exact path through the tree of content entries. Indeed, we see that a total of 50 students has achieved 85% or more progress during all evaluations and therefore completed the whole course. Another peak occurs at 10% of the course content, which is the amount of content that is loaded when first opening the course. These results show most students who did not abandon the online course immediately where able study the online course until completion.

5.2.2 Feedback Loop Durations.

The collected statistics also support our claim that our integrated content model enables short feedback loop durations that are on a scale of minutes. We extracted two kinds of duration values for each of the three evaluations. The first one is the feedback loop duration as defined in Definition 10. The second duration measures the time students needed to answer the exercises.

To compute the feedback loop duration, we have grouped the content into sequences of content entries where each sequence ends with an exercise. We then measured the

duration starting from the first entry in the sequence (e.g. the time when a student has first seen the entry) until an exercise (e.g the last entry in the sequence) was answered by the student. The histogram in Figure 5.5 shows the distribution of all feedback loop durations on a logarithmic scale for all three evaluations.

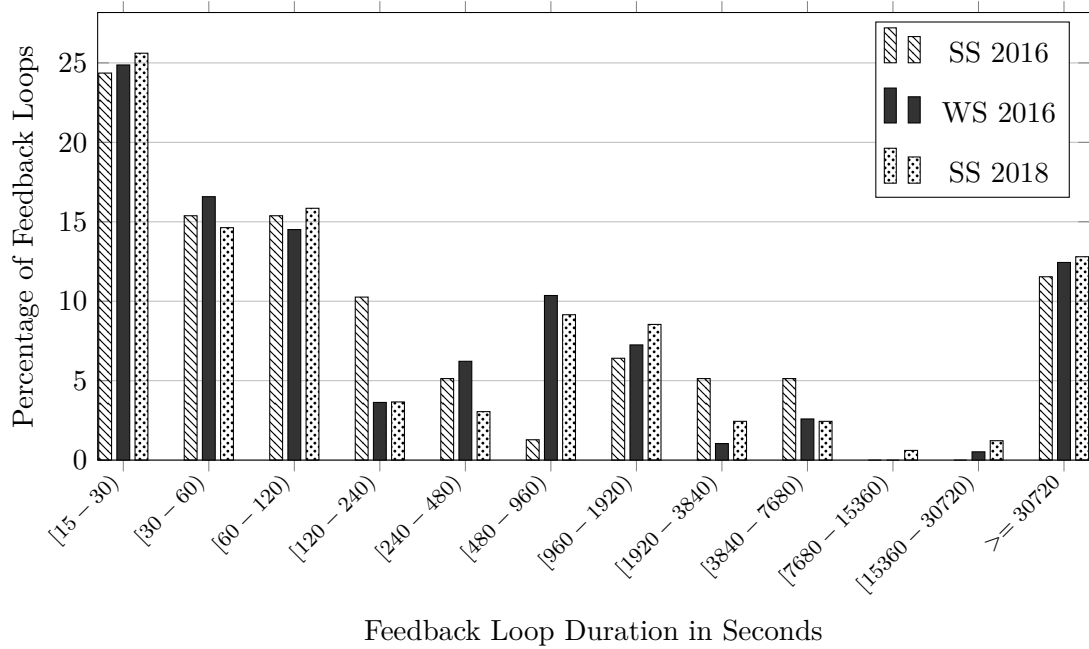
For the interpretation of the feedback loop durations we need to consider several properties of the data. First, it is possible that some students randomly clicked through the course without carefully studying the content. Second, it is possible that students loaded some content but then went offline and came back at a later time thereby significantly increasing the feedback loop duration. We assume that feedback loop durations above 8 hours include times where the students did not actively use the e-learning tool.

With respect to the first case, we have excluded entries with a duration of less than 15 seconds because these values likely correspond to students who quickly clicked through the whole course by randomly selecting answers. The first interval shown in the histogram (e.g. 15-30 seconds) should be interpreted with caution for the same reason. However, sometimes multiple exercises which can be answered quickly appear in close succession and could also explain these low durations.

With respect to the second case, we see that the histogram shows a trend of decreasing number of feedback loop durations with increasing duration intervals with the exception of the last interval. This last interval corresponds to cases where students were offline between starting and completing the feedback loop.

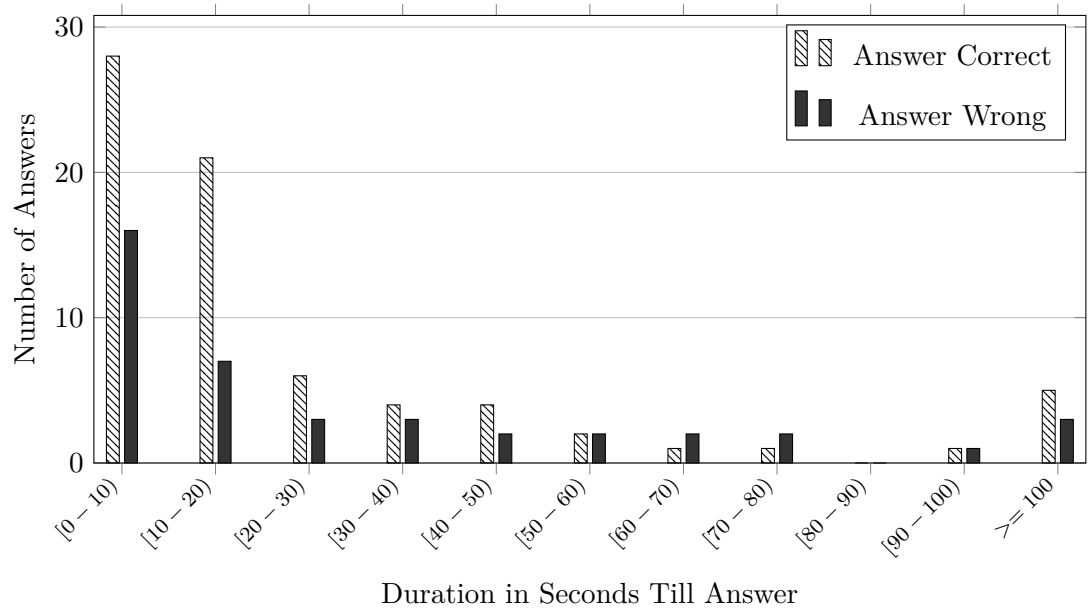
Independent, of whether we include or exclude the first interval (e.g. 15-30 seconds) we see that the majority of feedback loop durations is well below one hour and therefore on a scale of minutes.

The distribution of exercise answer times (see Figures 5.6, 5.7, and 5.8), shows that most students answered the exercises in less than two minutes. This shows that exercises were not a limiting factor for the duration of feedback loops and that feedback loop durations either depend on the length of the content between exercises or the times that students spend offline between studying the content. This also supports our claim that our integrated content model is suitable to achieve feedback loop durations on a scale of minutes.



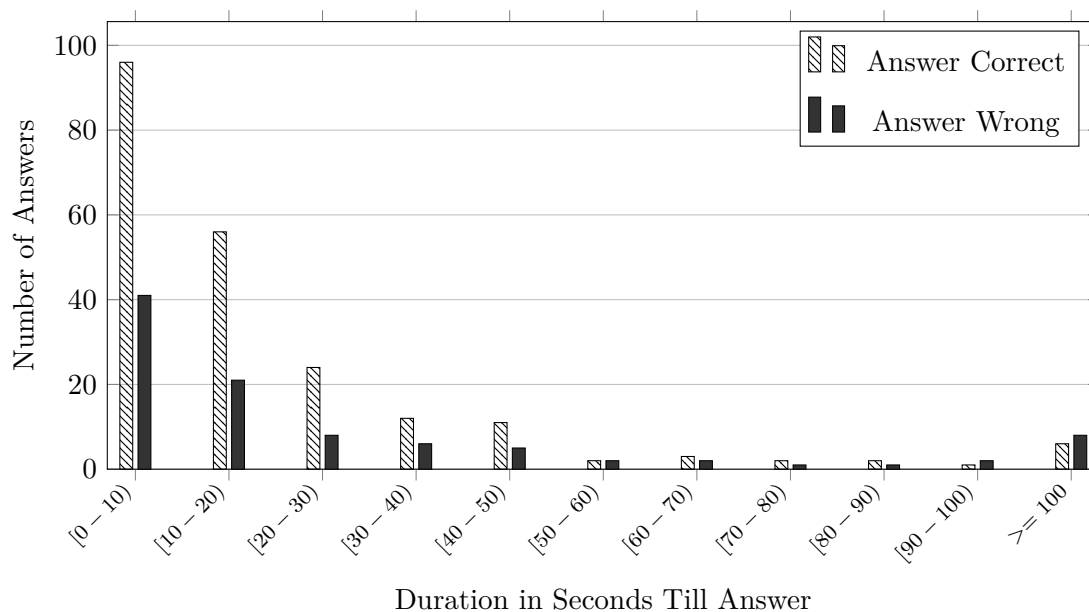
The histogram shows the distribution of feedback loop durations on a logarithmic scale.

Figure 5.5: Distribution of Feedback Loop Durations



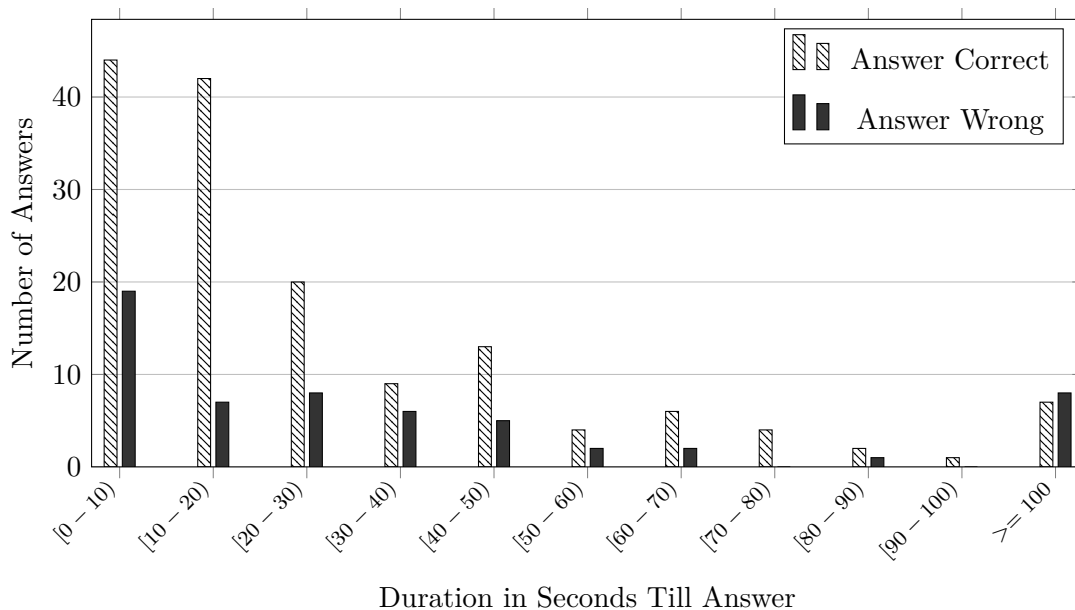
The histogram shows the number of correctly and incorrectly answered questions grouped by the duration between reading the question and giving an answer.

Figure 5.6: Distribution of Answer Times (SS 2016)



The histogram shows the number of correctly and incorrectly answered questions grouped by the duration between reading the question and giving an answer.

Figure 5.7: Distribution of Answer Times (WS 2016)



The histogram shows the number of correctly and incorrectly answered questions grouped by the duration between reading the question and giving an answer.

Figure 5.8: Distribution of Answer Times (SS 2018)

Conclusion

A key aspect of mastery learning is *formative evaluation*, i.e. the collection of early feedback through evaluations during the teaching process together with subsequent feedback for students that includes instructions which assist the learning process (see Definition 5). Bloom suggested to conduct these evaluations repeatedly in units of one or two weeks length [Blo68]. An open problem in this context was how modern e-learning software can be used to collect instant feedback, e.g. feedback loops on a scale of minutes rather than days or weeks, and how this feedback can be used to personalize the learning experience of students. In this thesis we have focused specifically on the question of how micro-tests and dynamic models can be used in modern e-learning software to achieve instant feedback.

Motivation and Preliminaries. In Chapter 1 we have presented our motivations for this thesis (see Section 1.1). We introduced a theoretical foundation and provided definitions for important terms in the field of e-learning such as for mastery learning, MOOC or flipped classroom (see Section 1.2). We saw that in mastery learning based education a strong distinction exists between *formative* and *summative* evaluation. In particular, formative evaluation turned out to be an important foundation for our micro-tests. Finally, we presented the aim of this work and discussed our goal to evaluate how micro-tests and dynamic models can support instant feedback in e-learning systems (see Section 1.3).

Related Work. In Chapter 2 we have introduced popular e-learning platforms and analyzed the content models, teaching methods and technical aspects of these platforms. We saw that from among these platforms only Khan Academy has implemented a mastery learning based teaching method (see Section 2.1). Other e-learning platforms such as Coursera and edX (see Sections 2.2 and 2.3) position themselves as MOOC platforms with a focus on scalability in order to provide access to higher education for a much

larger audience than in traditional universities. As such these platforms make teaching more efficient since a massive audience can be reached. But they offer few solutions for making learning more efficient as can be seen from the high dropout rates. We compared these platforms to each other and came to the conclusion that creating our own implementation will offer us the most flexibility to evaluate micro-tests and dynamic models (see Section 2.6).

Architecture and Implementation. In Chapters 3 and 4 we introduced our own e-learning implementation which we called “Interactive Lecture Notes”. This e-learning implementation features a dynamic content model and is specifically designed to support mathematical and scientific content. Our dynamic content model is able to interleave a variety of interactive and non-interactive content types that include textual content (e.g. text or mathematical formulas) but also micro-tests and dynamic models. We started with a high level introduction of the user interface design goals, a description of our content model and a presentation of our role system in Chapter 3. In Chapter 4 we then presented the architectural design of the individual modules that are part of our implementation. We have developed a flexible data model (see Section 4.2), that is able to model our dynamic content model as a tree-based data structure and that is able to store the course progress of individual students. Based on this data model we have developed a backend module that interacts with the database and offers its functionality through a RESTful web service (see Section 4.3). Finally we have implemented a client module in the form of a SPA that provides a web-based frontend for our e-learning application (see Section 4.4). The client module supports different roles to distinguish between student, teacher and administration functionality. Our main focus was on the student role and the implementation of micro-tests and a dynamic content stream. In addition, the teacher role provides access to course statistics and the administration role provides access to a small administration panel that can be used to (batch) import users and manage account details and roles of existing users.

Evaluation. In Chapter 5 we presented the results of our evaluation. We conducted three evaluations of our e-learning implementation and collected results through user surveys and from usage data. In total we received 30 responses from the surveys and a total of 50 users completed the online course on our platform.

Based on the collected results we see confirmation for the mastery learning approach. In particular, the answers from our surveys show that formative evaluation helps students during their learning process to identify problems more easily. We can also conclude that micro-tests and dynamic models seem to be a valid approach to provide instant feedback. Especially the integration of our micro-tests and the dynamic models into the content stream received a lot of positive feedback from students. Finally, our analysis of statistical data from the e-learning tool shows that feedback loop durations are on a scale of minutes.

Future Work. While we conducted three evaluations, we have seen that students are not particularly motivated to submit their feedback after they have used our system. We therefore need methods to collect better feedback on the e-learning system itself but also on the course content. Collecting such feedback would help us to improve both the implementation as well as the course content and it would allow us to make more statistically representative statements about formative evaluation, micro-tests and our dynamic content stream.

We have collected several ideas for further research in this direction, these include:

- Integration of “feedback elements” into the content stream.
- A voting and comment mechanism for individual content elements.
- A more thorough integration between the offline course and our e-learning system.

Feedback Elements. One way to collect more feedback could be the integration of *feedback elements* into the content stream that would be implemented in the form of a new *feedback* content type. Such a feedback content type could be used to ask questions about the e-learning system in general or about a specific part of the content. By integrating such questions directly into the content stream, students might be more willing to answer questions, than through a voluntary survey at the end of the whole course.

Voting and Comment System. A different approach could be through the implementation of a voting and comment system similar to that on the StackExchange [Excb] network. Such a system would mainly be useful to collect feedback about the content itself and not about the e-learning implementation in general. Students could use such a system to cast up or down votes on individual elements of our dynamic content stream to indicate the quality of the content. By adding comments to individual content elements it could also be possible to leave specific feedback for individual content elements.

Integration between e-learning system and offline course. During the evaluations the use of our e-learning system participation was voluntary and not all students made use of the e-learning system. A tighter integration of the e-learning system into the offline course or the introduction of mandatory aspects could significantly increase our user base. If mandatory aspects are introduced, they should still be in the form of formative rather than summative evaluation. For example, it could be required that a certain amount of micro-tests must be answered or that students achieve a certain amount of progress in the online course. However, we recommend that results from micro-tests themselves are not used to determine the students grades, as this would be contrary to the methods suggested by the mastery learning approach [Blo68].

Build and Deployment

In Section A.1 we will introduce the build system and containerization details. This should give the reader the necessary knowledge to make modifications in the build system and the docker images. We assume that the reader has a solid understanding of both Maven and Docker, which are used in the build system and for containerization, respectively. In Section A.2 we discuss how the different modules can be deployed and maintained in a test or production environment. Our software is open source and available on Github. It can be downloaded from <https://github.com/lanoxx/masterylearning>. The build instructions described in this Appendix apply to version 0.6 of our software.

A.1 Build System Documentation

In this section we start with a description of the build system aspects of the backend and client module that are important to build the docker images from these modules. For the database module we simply use the official MySQL docker image, which does not need to be explicitly built.

For a complete build several software dependencies must have already been installed with at least the versions shown below:

- Maven 3
- Java Development Kit (JDK) 8
- Node 4.0
- Node Package Manager (NPM) 3.5
- Bower 1.7

Attribute	Value
Hostname	localhost
Port	4001
Database Name	masterylearning

Table A.1: Test Database Configuration

- Docker 1.11

The build system has been configured in such a way, that the whole software can be built, tested and released with Maven using a single command (e.g. `mvn install`).

This includes running the unit tests, packaging the software artifacts, building the docker images for backend and client, running the integration tests and finally installing the docker images to the local docker registry.

Running the integration tests requires that a test database is available, which should be configured as specified in Table A.1. To make the docker images available on a public registry they must be manually pushed using docker's `push` command.

Dockerization. The install phase of the build system produces two docker images – the backend image and the client image. Building the docker images requires two things, a *Dockerfile* with build instructions and the software artifacts that have been produced during the packaging phase.

The backend module's docker image is built from the official Java 8 docker image¹. This image already includes the Java 8 runtime environment. We only need to add the executable jar archive to the image, specify the necessary environment variables and the start command that runs our Java application. The whole Dockerfile for the backend module is shown in Listing A.1. To make the database hostname of the docker image configurable at container creation time, we have specified it using the `ENV` directive. The default value (`mysql`) is specified with the `ARG` directive and can be overridden at image build time. Last but not least then `ENTRYPOINT` directive specifies that necessary command to execute our Java Application.

The Dockerfile of the client module (see A.2) is slightly more complex than the backend module. The client module's built process does not produce a single executable but rather a set of HTML, CSS and JavaScript files that must be served on a web server. We use NGINX as our web server in the docker image, therefore the Dockerfile uses the NGINX image from the official docker repository².

The docker image contains two essential parts, the first one is a template file named `default.template` that contains the nginx configuration file, the second part are the

¹https://hub.docker.com/_/java/

²https://hub.docker.com/_/nginx/

```
1 # backend/src/main/docker/Dockerfile
2 FROM java:8
3 VOLUME /tmp
4 ADD backend-@project.version@.jar app.jar
5 RUN bash -c 'touch /app.jar'
6 ARG DB_HOST=mysql
7 ENV DB_HOST $DB_HOST
8 ENTRYPOINT [
9     "java",
10    "-Djava.security.egd=file:/dev/./urandom",
11    "-Dspring.datasource.url=jdbc:mysql://${DB_HOST}/masterylearning",
12    "-jar",
13    "/app.jar"
14 ]
```

Listing A.1: Docker File for Backend Module.

```
1 # client/docker/Dockerfile
2 FROM nginx:stable
3 VOLUME /tmp
4 ADD nginx/default.template /etc/nginx/conf.d/default.template
5 RUN mkdir -p /var/www
6 ADD app /var/www
7 # Set a default value for our nginx configuration in case the user
8 # does not override this with '--env' when the container is started.
9 ENV NGINX_HOST localhost
10 ENTRYPOINT [
11     "/bin/bash",
12     "-c",
13     "envsubst '$NGINX_HOST' \
14     < /etc/nginx/conf.d/default.template \
15     > /etc/nginx/conf.d/default.conf \
16     && exec nginx -g 'daemon off;'"
17 ]
```

Listing A.2: Docker File for Client Module.

actual source files for the web site, which are located in the source tree in a folder named `app`. The reason that we use a template file for the nginx configuration is, that the hostname of the deployment machine is only known at deployment time and thus must always be specified when the container is created. As it can be seen in the `ENTRYPOINT` directive, before the nginx web server is started, the `envsubst` command is invoked. This will replace the `NGINX_HOST` variable in the template and create the final configuration file.

A.2 Deployment Documentation

For the deployment of our software we have used Amazon Web Services (AWS). The software itself however is built in such a way, that there are no specific dependencies on AWS, and thus a deployment on any other environment is equally possible. In fact there are two ways to deploy the software. One can either manually deploy the artifacts or use the docker images that are build from each module.

Requirements. Except for the built artifacts themselves, there is one additional requirement for the deployment of this software. In order for the password reset feature to work, a mail server and email address must be available that can be used by the backend module to send the password reset emails to the users. Also if the docker deployment option is used, then a recent enough docker installation must be available³.

Manual Deployment. For a manual deployment, the database, backend and client modules must each be manually installed and configured in the target environment. We do not provide detailed instructions for a manual deployment in this sections, since the docker deployment is much easier to setup and maintain.

Nevertheless, if a manual deployment is required, the following steps must be performed:

1. Setup a MySQL database instance and create a database named `masterylearning`.
2. Install a Java runtime, and run the backend module. Setup the necessary configuration options for the database as environment parameters, or pass them as system properties to the `java` command.
3. Install a web server such as `nginx` and configure it such that the frontend application files will be served. Also configure the reverse proxy appropriately as described in Section 4.4.2.

Docker Deployment. Using the docker images is considerably easier, as it just requires to create the necessary containers in a docker environment. In this section we will focus on a deployment using Docker. A simple minimum deployment requires at least three docker containers - one for each of the modules database, backend and client.

While it is possible to use separate hosts to the deploy the docker containers, for simplicity we assume that they are all deployed on the same docker host. In this case the containers use a separate docker network for inter-container communication (see Listing A.3). As a first step when creating a new deployment, one must this create such a network. We assume the network will be called `mlnet` (e.g. mastery learning network), it can be created with the following command:

³At the time of this writing Docker 1.11.2 was used

```
1 docker network create mlnet
```

Listing A.3: Creation of a Container Network with Docker

```
1 docker run --name masterylearning-mysql --net mlnet \  
2         --net-alias mysql \  
3         --env MYSQL_ROOT_PASSWORD=masterylearning \  
4         --env MYSQL_DATABASE=masterylearning \  
5         -v /srv/masterylearning:/var/lib/mysql -d mysql:5.7
```

Listing A.4: Create and Run Database Container.

Database Container. After a container network has been created, each of the three containers must be started. For the database module is is enough to just start a new container using the MySQL docker image⁴ from the official docker registry as shown in Listing A.4.

The reader should pay attention to several aspects of this command. The container will be named ‘masterylearning-mysql’ and it has an internal network alias that is named `mysql`. This alias is important later on, when we create the backend container, as it will by default assume that the database is available from a hostname named `mysql`, if a different name is specified, then this must also be configured correctly when the backend container is created. The environment variable for the password specifies the default root user’s password and a database named `masterylearning` is created by default if none exists yet.

The `-v` option directs docker to use a storage location for the MySQL database files that lies outside of the container in the folder on the host machine that is named `/srv/masterylearning`. This is important to ensure, that the persistent database content is not lost if we replace the database container, for example if a new MySQL release (e.g. with security fixes) is available. Last but not least we indicate with the `-d` option that we want to use the latest version of MySQL from the official docker repository.

Backend Container. The backend container can be setup with the command shown in Listing A.5. When running the backend container, the placeholders values in Listing A.5 for the configuration of the mail server must be replaced with values of a valid mail server. Listing A.6 shows additional environment variables that can be added to the command in Listing A.5, to override the connection details of the database. The values used for the environment parameters in listing A.6 are default values and thus only need to be specified if different values are required. Note, that the `--net-alias` parameter creates a network alias in the specified docker network. Since the commands to start the database and the backend both specify the same docker network (`mlnet`), the database will be reachable via its network alias from the backend container.

⁴https://hub.docker.com/_/mysql/

```
1 docker run \  
2   --name masterylearning-backend --net mlnet \  
3   --net-alias backend \  
4   --env 'spring.mail.host=<hostname>' \  
5   --env 'spring.mail.port=<port>' \  
6   --env 'spring.mail.username=<username>' \  
7   --env 'spring.mail.password=<password>' \  
8   --env 'spring.mail.properties.mail.smtp.auth=true' \  
9   --env 'spring.mail.properties.mail.smtp.starttls.enable=true' \  
10  --env 'spring.mail.test-connection=true' \  
11  --env 'email.hostname=<hostname>' \  
12  --env 'email.from=<email-address>' \  
13  -d masterylearning/backend
```

Listing A.5: Create and Run Backend Container.

```
1 --env 'spring.datasource.url=jdbc:mysql://mysql/masterylearning'  
2 --env 'spring.datasource.username=root'  
3 --env 'spring.datasource.password=masterylearning'
```

Listing A.6: Additional Parameters for Database Configuration.

```
1 docker run --name masterylearning-client --net mlnet --net-alias client \  
2           -p 80:80 \  
3           -p 443:443 \  
4           --volume /etc/letsencrypt:/etc/letsencrypt \  
5           --volume /etc/ssl/certs:/etc/ssl/certs \  
6           --env NGINX_HOST=<web-server-hostname> \  
7           -d masterylearning/client
```

Listing A.7: Create and Run Client Container.

Client Container. The client container can be setup with the command shown in Listing A.7. To run the client container, the `NGINX_HOST` environment variable must be set to the FQDN of the host on which the client container has been deployed. In addition two volumes must be mounted into the client container, the `/etc/letsencrypt` volume should contain SSL certificates for the host, created via “Let’s Encrypt”.

The second mapping mounts the `/etc/ssl/certs` folder into the container. A `dhparam` key-pair must be present in this folder under the name `dhparam.pem`. The key-pair can be created with the command shown in Listing A.8.

Backup Creation The client and backend containers are stateless and can be easily recreated with the above commands. The only part which requires to be backed up is the database content. The command in Listing A.9 shows how we can use a *MySQL* Docker

```
1 sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

Listing A.8: Create Key Pair for Additional Security in Client Container.

```
1 docker run --rm --interactive --tty --net mlnet \  
2     --user 'id -u': 'id -u' \  
3     --volume <local_destination>:/backup mysql \  
4     bash -c "mysqldump -u root -p -h mysql masterylearning \  
5         > /backup/backup-`date +%F_%H-%M-%S`.sql"
```

Listing A.9: Backup the MySQL Database Using a Docker Container

container which will use `mysqldump` in order to create the database backup. Note, that the placeholder `<local_destination>` has to be replaced by the folder where the final backup is supposed to be stored, such as `/home/myuser/backup/`. Also note, that when the script is being executed it will ask for the root password of the database on the shell. In order to automate the above script (e.g. by a cron job) the database password for the root user will have to be integrated into the script.

An interesting aspect of the command in Listing A.9 is, that we will use a docker container based on the `mysql` image to actually perform the backup. The advantage of this is that we do not need to have any MySQL dependencies installed on the host system. Also after the backup is complete the container will automatically be deleted. Let us take a closer look at the options that are being passed to docker. The `--net` option ensures that our backup container runs in the same network as the database container which we want to backup. Otherwise we would not be able to access the database. The `--user` option ensures that the container is being executed in the context of the currently logged in user. This has the effect that after the script finishes, the backup files on the host will be owned by the user who executed the script. The `--volume` option mounts the backup location under `/backup` inside the container and then the name of the docker image (e.g. `mysql`) is specified. Finally we execute `bash` with a custom command inside the container that performs the actual backup.

Evaluation Results

This Appendix contains results from the three evaluations for questions with free-form entries. The remainder of this Appendix presents the five questions for which free-form answers were possible, followed by a list of quotes with the answers from each evaluation. For an interpretation of the answers please refer to our evaluation (see Chapter 5. All quotes are numbered to allow referencing of individual quotes from other Chapters.

Question 15: *I found the following aspects of the tool helpful (please provide the most helpful aspect first):*

Answers in the summer term 2016:

“The incremental presentation
The questions/exercises” (1)

Answers in the winter term 2016:

“exercises” (2)

* * *

“I think for me the LTL simulation was the most important part because with it I could test any formula on any model.” (3)

* * *

“It only provided a piece of new information, so one could stay concentrated and did not feel overwhelmed when comparing to presenting everything at once. I think I would have immediately left, if I had seen the whole page at once.” (4)

* * *

“Visualization and self-checking the understanding; the continue from last point feature;” (5)

Answers in the summer term 2018:

“explanations to the yes/no questions; clicking through the model checking algorithms” (6)

* * *

“exercises” (7)

* * *

“Interactive elements (especially step by step execution of algorithms)” (8)

* * *

“Algorithm of CTL” (9)

* * *

“It explains some aspects of the lecture again with other words which is quite helpful in order to understand the concepts.” (10)

* * *

“The explanations as to why my answer was wrong. This really helped understand the material.” (11)

* * *

“very nice, easy to read, web design” (12)

* * *

“Specific exercises and interactive tools are very good to understand the concepts” (13)

* * *

“Learning at my own speed” (14)

* * *

“presentation of content one below the other, exercises to check knowledge after a section” (15)

* * *

“The frequent questions were very helpful to recap the content, nice interactive examples” (16)

* * *

“Explanation from CTL* to CTL and LTL.” (17)

* * *

“Takeaway/Definition blocks, exercises” (18)

* * *

“Well written descriptions and content, especially explanations of errors in exercises/questions. I also really liked the presentation of the algorithm and the interactive graphs. I really found the continuity and the pattern of alternating text descriptions/task descriptions/definitions, then followed by graphs/illustrations, then followed by examples (and after the example the next chapter starting from descriptions/definitions again and so on) very helpful and compelling.” (19)

Question 16: *I found the following aspects of the tool NOT helpful (please provide the most unhelpful aspect first):*

Answers in the summer term 2016:

“The lack of a "current position" - how much is to yet to come? When answering a question wrong, often no explanation was given (since the reasons why can be manifold I can understand, but a 'best guess' would still help. Or maybe a reference to the source material, even within the tool itself? Sometimes it was unclear to which parts of the content the questions belonged (the answer was not deducible from the presented content)” (20)

Answers in the winter term 2016:

“when an answer was wrong, there was no explanation/justification why” (21)

* * *

“It would have been nice to have the whole course outlined in a tabel of contet so that it would be possible to jump to any section instead of the continue button.” (22)

* * *

“It was not obvious how much the course would cover and what the overall progress was.” (23)

* * *

“not being able to reset the exercises; some (can’t remember which) example was not immediately clear to me (explanation was not enough) - maybe do a supervised run with students to find such issues;” (24)

Answers in the summer term 2018:

“some yes/no questions are too obvious” (25)

* * *

“Please also show the solution of correctly answered questions. More questions would be useful. Please also include other parts, e.g. bisimulation.” (26)

* * *

“Unfortunately there were exercises only for some parts of the content and not all of them.” (27)

* * *

“Loading additional content sometimes would scroll things around, so i had to scroll back to where i was.” (28)

* * *

“interactive CTL model checking algorithm - i still haven’t grasped the purpose of it” (29)

* * *

“Please provide an explainaiton why something is the correct answer!” (30)

* * *

“One question was missleading” (31)

* * *

“Sections 1.1 and especially 1.3.5 contained irregularly much text, which hurt the pattern of this e-learning page - section 1.3.5 lost a bit of my interest because of the wall-of-text appearance.” (32)

Question 23: *If you used material other than lecture slides please provide more information (e.g., book title, or URL of online material):*

Answers in the summer term 2016:

”Vowi, official FMI website (previous exams)” (33)

* * *

”<http://forsyte.at/kripke/>” (34)

Answers in the winter term 2016:

”wikipedia; google;” (35)

Answers in the summer term 2018:

”vowi” (36)

* * *

”Exams from here: <https://www.logic.at/lvas/fminf/>” (37)

* * *

"wikipedia" (38)

* * *

"exercises from the instructors provided via tuwel (incl. supplementary material)" (39)

* * *

"*) Model Checking by Clarke, Grumberg, and Peled
*) Principles of Model Checking by Baier and Katoen" (40)

* * *

"Stackexchange, Stackoverflow" (41)

Question 28: *If you did not attend all lectures of FMI, what were the most important reasons for that?*

Answers in the summer term 2016:

time (42)

* * *

Conflicting schedule (43)

Answers in the winter term 2016:

"I attended ~80% of the FMI lectures in a previous semester but I did not take the exam. Therefore I thought that I do not have to attend the lectures again and instead I focused on learning for the exam." (44)

* * *

"work and i attended the lecture in previous semesters" (45)

* * *

“Took the course in a previous semester” (46)

* * *

“too much other stuff to work on” (47)

* * *

“timing conflicts; Uwe;” (48)

Answers in the summer term 2018:

“overlap with another lecture with only one date per week” (49)

* * *

“I often skipped Lectures because watching the Video Recordings afterwards
by myself I could concentrate better” (50)

* * *

“scheduling problems” (51)

* * *

“other lectures” (52)

* * *

“Collision with other appointments” (53)

* * *

“Full time job” (54)

* * *

“Conflicting appointments” (55)

* * *

“That recordings are present so i can watch them on demand. I wish the recordings would include the black board too. ” (56)

* * *

“lack of time - i am fully employed with a 40hour job to feed my family” (57)

* * *

“Lectures are available online.” (58)

* * *

“overlap with other lectures“ (59)

* * *

“urgent stuff to do at my job, feeling that I don’t understand FMI stuff anyway so that it will possibly not make a difference whether I attend or not” (60)

* * *

“Work” (61)

* * *

“Some dates overlapped with other lectures” (62)

* * *

“Other lectures and concurrent lecture dates but I chose FMI over the other ones most of the time + Deadline pressure of other lectures (especially (other) exercises)” (63)

Question 32: *Further comments*

Answers in the summer term 2016:

“I applaud the effort. Such interactive tools are incredibly useful. I would love to see more of these in the future - even outside of university. For some people the slide & talk lecturing style (especially in this course) is a sub-optimal way of studying. Interactive tools can provide much more insight since hypothesis can be quickly tested and advanced/disregarded.” (64)

* * *

“Thank you for your work.” (65)

* * *

“Thank you for the invite to participate” (66)

* * *

“I really like the idea of your platform. Since my only goal is to pass the exam, i have to say though that for this purpose its not really necessary using the platform. The exam or in general the whole course focuses on proving things. The platform might be helpful in the beginning to understand the content, but it doesnt really prepare for the exam.” (67)

Answers in the winter term 2016:

“I think providing that additional tool is a great idea, and I would love if there were more exercises (and content). Although, I have already passed the exam, I think it will help students in the future. So keep the work up! Thanks for providing the tool and good luck with your master thesis. A dedicated exercises part with additional exercises would be great” (68)

Answers in the summer term 2018:

“Including video material along text would be great. And maybe the possibility of adding my notes (I ended up printing some parts of the material).” (69)

* * *

“Well done, more tools like this would enhance the "special" experience learning FMI.” (70)

* * *

“Usability/presentation was done very good (i find it very important that I can easily look back without clicking back 20 times - here it’s CTRL-F, done). Please continue the work, in contrast to the materials provided by the lecture it feels that this e-learning tool is better in explaining the content (slides are full of stuff, due to the nature of slides they miss natural language explanations). I had difficulties with the word "eventually" because I am bad at English and confused it with the German word "eventuell" until I looked it up and could finally rest in peace because the formal definition was true after all. I don’t know how much people have these types of problems but it may be good to include some box explaining the word (similar to the Amazon Kindle Word Wise feature in their e-books). Thank you, I think I could really profit from your tool.“ (71)

* * *

“It was very easy to follow and well explained. Some more content (Bisimulation etc.) would have been great :-)” (72)

* * *

“Maybe add an additional menu-like overview of the individual content parts in, for example, the side bar. The overview could help the user to navigate back to previous content parts“ (73)

* * *

“I just would like to see more usage of the e-learning tool, especially other blocks and extend block4 content to (bi-)simulation. The e-learning tool should not replace lecture recordings but complement them. FMI is one of the hardest courses in the computer science master’s programmes at VUT (TU Wien) and, unfortunately, lacks exercises with proper feedback (unfortunately I did not receive any individual feedback for this semester). To sum it up: this e-learning tool is very useful, especially for courses like FMI.” (74)

List of Figures

1.1	Example of Synchronous Learning: During each time interval $[t_a, t_b)$ a teacher is teaching the same concept c_i to all students S_i .	5
1.2	Example for Asynchronous Learning: At each time t_i the students S_j can learn concepts c_k independently from each other by watching the corresponding video V_k .	6
2.1	Khan Academy Content Structure	24
2.2	Subtopics	25
2.3	Mission Progress	25
2.4	Exercise with a Dynamic Model for the Skill <i>Graph quadratics in vertex form</i> .	26
2.5	Udacity Quiz with Answers Mechanism.	35
3.1	Content Stream	47
3.2	Paragraphs with HTML Content and Math Mode Enabled	49
3.3	Paragraph as Widget of Type “Definition”	49
3.4	Paragraph as Widget of Type “Takeaway Point”	50
3.5	Exercise Type with Yes/No Answer.	51
3.6	Exercise Type with Multiple Possible Answers.	51
3.7	Interactive Kripke Structure with Corresponding Computation Tree [Hol16]	52
3.8	Interactive Pseudo Code Evaluator [Hol16]	53
3.9	Home Screen with Navigation	54
4.1	Architectural Overview	58
4.2	Data Model for Course and Entry Tree	60
4.3	Example Course with Entry Tree (bright nodes are instances of EntryData)	60
4.4	Data Model for Entry Data	61
4.5	Data Model for Exercises	63
4.6	Data Model for User Data	63
4.7	Folder Structure of the Client Module	74
5.1	Platform Distribution	84
5.2	Primary Platform Distribution	85
5.3	Lecture Materials used by Students to Prepare for Exam	86
5.4	User Progress	90

5.5	Distribution of Feedback Loop Durations	92
5.6	Distribution of Answer Times (SS 2016)	92
5.7	Distribution of Answer Times (WS 2016)	93
5.8	Distribution of Answer Times (SS 2018)	93

List of Tables

2.1	Comparison of E-Learning Platforms and Software	39
2.2	MOOC Acronym Comparison	42
3.1	Content Model	48
4.1	Endpoint Overview	72
5.1	Number of Submissions per Evaluation.	78
5.2	Overall Experience.	79
5.3	Student Reasons for Aborting the Online Course.	80
5.4	Detailed Feedback on 'Presentation of Content'	81
5.5	Detailed Feedback on 'Interactivity'	82
5.6	General Feedback	83
5.7	Implementation Speed (Subjective User Experience)	85
5.8	Lecture Context	87
5.9	Lecture Attendance	88
5.10	Prerequisites and Further Comments	89
A.1	Test Database Configuration	100

List of Algorithms

3.1	CSV Import Format without Header Line	55
3.2	CSV Import Format with Header Line	56
4.1	Hibernate Annotations in Java for Course Entity	66
4.2	Spring Data Repository with Additional Methods	67
4.3	Request Body for Create User Endpoint.	73
4.4	Response Body for Create User Endpoint.	73
4.5	Error Response for Create User Endpoint.	73
4.6	Nginx Reverse Proxy Configuration	76
A.1	Docker File for Backend Module.	101
A.2	Docker File for Client Module.	101
A.3	Creation of a Container Network with Docker	103
A.4	Create and Run Database Container.	103
A.5	Create and Run Backend Container.	104
A.6	Additional Parameters for Database Configuration.	104
A.7	Create and Run Client Container.	104
A.8	Create Key Pair for Additional Security in Client Container.	105
A.9	Backup the MySQL Database Using a Docker Container	105

Index

- architecture, 38, 43, 57, 59
- assignment, 13, 29, 37
- asynchronous learning, 5, 8, 10
- badge, 27
- blended learning, 3, 9, 27, 32
- build system, 99
- confirmation bias, 52
- content model, 33, 48
- content stream, 18, 46
- content type, 29, 42, 48
 - blocking content, 48
 - continue button, 49
 - exercise, 50
 - interactive content, 48, 52
 - paragraph, 49
 - section, 48
- course
 - instructor-paced, 31
 - on-demand, 29
 - self-paced, 31
 - session-based, 29
- dynamic models, 12
- e-learning platform, 4, 10, 13
 - Coursera, 28
 - edX, 31
 - Khan Academy, 23
 - Moodle, 36
 - Udacity, 33
- e-learning software, 3, 9
- educational technology, 3
- energy point, 27
- entry tree, 61, 62
- exercise, 32
 - repeatable, 39
 - skill, 25
- feedback loop, 11
 - duration, 11, 90
- flipped classroom, 3, 10, 28, 39
- formative evaluation, 7, 9, 27, 35, 52, 95
- gamification, 15, 27, 28
- higher education, 28
- instant feedback, 18, 28, 31, 33, 36, 38, 42, 80, 81
- interactive
 - content, 15, 47
- Massive Open Online Course, 3, 9, 13, 30, 31, 38
- mastery challenge, 26, 27
- mastery learning, 3, 6, 7, 27, 29, 32, 36, 38, 95
- math mode, 49
- micro-tests, 12
- mission, 25
- quiz, 29, 33, 35, 37
- rest period, 27
- retention, 9, 28
- scheduled learning, 3, 6, 8, 27, 29, 32
- summative evaluation, 7, 8
- synchronous learning, 4, 8, 10

- traditional teaching, 3, 4, 28
 - traditional education, 6
 - traditional lecture, 5, 9
 - traditional university course, 13, 28
- video, 5, 10, 15, 29, 32, 33, 42
 - drawbacks, 17
 - optimal length, 2
 - screencast, 2
 - transcript, 30

Glossary

BCrypt A secure hashing algorithm developed by Niels Provos and David Mazières [PM99]. It includes a salt and a cost value that is used to adjust the computational cost of the hashing algorithm. 63

DDL The DDL is a subset of the SQL and is used to define the schema of relational databases. 64, 123

FMI *Formale Methoden der Informatik* (FMI) is the German title of the lecture *Formal Method in Computer Science*. 79

RESTful A web service architecture that uses REST concepts to implement its service endpoints. 20, 45, 47, 57, 65, 96

Acronyms

API Application Programming Interface. 4, 59, 65, 66, 75

CORS Cross Origin Request Sharing. 59

DAO Data Access Object. 66

DOM Document Object Model. 75

JDBC Java Database Connectivity. 58

JPA Java Persistence API. 65, 66

JSON JavaScript Object Notation. 47

MIT Massachusetts Institute of Technology. 31

MOOC Massive Open Online Course. 12–14, 31, 95

ORM Object-Relational Mapping. 65

REST Representational State Transfer. 72, 123

SPA Single Page Application. 20, 38, 42, 58, 73, 75, 96

SPI Service Provider Interface. 4

SQL Structured Query Language. 66, 123

WYSIWYG What You See Is What You Get. 47

Bibliography

- [AB70] Daniel Alpert and Donald L Bitzer. Advances in computer-based education. *Science*, 167(3925):1582–1590, 1970. <https://stacks.stanford.edu/file/druid:gq846pp5099/gq846pp5099.pdf>.
- [Acaa] Khan Academy. Can I use Khan Academy’s videos/name/materials/link on my site? <https://khanacademy.zendesk.com/hc/en-us/articles/202262954-Can-my-website-or-company-use-Khan-Academy-materials-or-its-name-> (Last accessed: 19.12.2016).
- [Acab] Khan Academy. Glossary of Khan Academy terms. <https://www.khanacademy.org/educator/reference-for-coaches/how-to/a/glossary-of-ka-terms> (Last accessed: 15.12.2016).
- [Aga13] Anant Agarwal. Anant Agarwal: Why massively open online courses (still) matter. https://www.ted.com/talks/anant_agarwal_why_massively_open_online_courses_still_matter?language=en, 2013.
- [Aga15] Anant Agarwala. Professor zweiter Klasse. <http://www.zeit.de/2015/28/lehre-forschung-universitaet-lehrprofessur> (Last accessed: 06.12.2016), 2015.
- [Aka13] Anuli Akanegbu. Does the Khan Academy Pass the MOOC “Duck Test”? <http://www.edtechmagazine.com/higher/article/2013/04/does-khan-academy-pass-mooc-duck-test> (Last accessed: 19.12.2016), 2013.
- [APA] APA. TU Wien: Aufnahmetest soll Studentenzahl drastisch senken. <http://derstandard.at/2000033932130/TU-Wien-Neue-Aufnahmetest-sollen-Studentenzahlen-drastisch-senken>.
- [Ban15] Sudarshana Banerjee. Each month, about 250,000 unique learners visit Khan Academy from India: Salman Khan. <http://www.business-standard.com/article/companies/>

each-month-about-250-000-unique-learners-visit-khan-academy-from-india-salman-khan-115031201372_1.html (Last accessed: 09.12.2016), 2015.

- [BBL61] D. Bitzer, P. Braunfeld, and W. Lichtenberger. PLATO: An Automatic Teaching Device. *IRE Transactions on Education*, 4(4):157–161, Dec 1961. <http://dx.doi.org/10.1109/TE.1961.4322215>.
- [BK] Lisa Breit and Lisa Kogelnik. Widerstand gegen Kürzung der IT-Plätze an TU Wien. <http://derstandard.at/2000043178457/Widerstand-gegen-Kuerzung-der-IT-Plaetze-an-TU-Wien>.
- [BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.
- [Blo68] Benjamin S Bloom. Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment*, 1(2):n2, 1968. <http://files.eric.ed.gov/fulltext/ED053419.pdf>.
- [Bog15] Miranda Bogen. Coursera flipped the classroom, but can it turn a profit? <https://openforum.hbs.org/challenge/understand-digital-transformation-of-business/business-model/coursera-flipped-the-classroom-but-can-it-turn-a-profit>, 2015.
- [BS13] Helen Beetham and Rhona Sharpe. *Rethinking pedagogy for a digital age: Designing for 21st century learning*. routledge, 2013.
- [Bun16] Bunchball. What Is Gamification? <http://www.bunchball.com/gamification> (Last accessed: 16.12.2016), 2016.
- [BV13] Jacob Lowell Bishop and Matthew A Verleger. The flipped classroom: A survey of the research. In *ASEE National Conference Proceedings, Atlanta, GA*, volume 30, 2013. <http://www.studiesuccesho.nl/wp-content/uploads/2014/04/flipped-classroom-artikel.pdf>.
- [Car14] Erin Carson. How MOOCs are flattening corporate training and education. <http://www.techrepublic.com/article/how-moocs-are-flattening-corporate-training-and-education/> (Last accessed: 16.12.2016), 2014.
- [Cen18] Coursera Help Center. Answer in-video questions. <https://learner.coursera.help/hc/en-us/articles/209818663-Answer-in-video-questions> (Last accessed: 20.01.2018), 2018.

- [CGP99] Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model checking*. MIT press, 1999.
- [Cha13] Max Chafkin. Udacity’s Sebastian Thrun, Godfather Of Free Online Education, Changes Course. 2013. <https://www.fastcompany.com/3021473/udacity-sebastian-thrun-uphill-climb>.
- [Com] Creative Commons. Attribution-NonCommercial-ShareAlike 3.0 United States. <https://creativecommons.org/licenses/by-nc-sa/3.0/us/> (Last accessed: 19.12.2016).
- [Coua] Coursera. About Coursera. <https://about.coursera.org/> (Last accessed: 16.12.2016).
- [Coub] Coursera. Coursera Blog Coursera Update: Striking a Balance with Start Dates and Deadlines. <http://coursera.tumblr.com/post/130838910317/coursera-update-striking-a-balance-with-start> (Last accessed: 16.12.2016).
- [Couc] Coursera. Courses and Specializations. <https://www.coursera.org/courses/?languages=en&query=free+courses> (Last accessed: 16.12.2016).
- [Cou12] Coursera. Online Course Hosting and Service Agreement. <https://assets.documentcloud.org/documents/400864/coursera-fully-executed-agreement.pdf> (Last accessed: 16.12.2016), 2012.
- [Cro04] Jay Cross. An informal history of eLearning. *On the Horizon*, 12(3):103–110, 2004.
- [DDKN11] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From Game Design Elements to Gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek ’11, pages 9–15, New York, NY, USA, 2011. ACM.
- [EA] Emily Eisenberg and Ben Alpert. KATEX - The fastest math typesetting library for the web. <https://khan.github.io/KaTeX/> (Last accessed: 17.01.2017).
- [Edu] Edudemic. The Teacher’s Guide To Flipped Classrooms. <http://www.edudemic.com/guides/flipped-classrooms-guide/> (Last accessed: 19.12.2016).
- [edX] edX. About Page of edX. <https://www.edx.org/about-us> (Last accessed: 16.12.2016).

- [eI16] edX Inc. Blended Learning with edX. <https://www.edx.org/course/blended-learning-edx-edx-blendedx-1>, 2016.
- [Exca] Stack Exchange. StackOverflow Help Center: Privileges. <http://stackoverflow.com/help/privileges> (Last accessed: 16.12.2016).
- [Excb] Stack Exchange. StackOverflow Help Center: Reputation & Moderation. <http://stackoverflow.com/help/whats-reputation> (Last accessed: 16.12.2016).
- [Fac] Fachschaft Informatik (TU Wien). Vowie (VorlesungsWiki). <http://vowi.fsinf.at/> (Last accessed: 07.07.2018).
- [Fis12] Lyla Fischer. <https://www.youtube.com/watch?v=OEoXaMPEzfM>, 2012.
- [Gar11] D Randy Garrison. *E-learning in the 21st century: A framework for research and practice*. Taylor & Francis, 2011.
- [GK04] D Randy Garrison and Heather Kanuka. Blended learning: Uncovering its transformative potential in higher education. *The internet and higher education*, 7(2):95–105, 2004.
<http://www.anitacrawley.net/Resources/Articles/GarrisonKanuka2004.pdf>.
- [Gog13] Roberta Gogos. A brief history of elearning (infographic). <https://www.efrontlearning.com/blog/2013/08/a-brief-history-of-elearning-infographic.html> (Last accessed: 06.12.2016), 2013.
- [Goo18] Google. Google forms. <https://docs.google.com/forms> (Last accessed: 08.07.2018), 2018.
- [GQJ⁺13] Khosrow Ghadiri, Mohammad H Qayoumi, Ellen Junn, Ping Hsu, and Sutee Sujitparapitaya. The transformative potential of blended learning using MIT edX’s 6.002 x online MOOC content combined with student team-based learning in class. *environment*, 8:14, 2013. <https://www.edx.org/sites/default/files/upload/ed-tech-paper.pdf>.
- [Guo13] Philip Guo. Optimal Video Length for Student Engagement. <http://blog.edx.org/optimal-video-length-student-engagement>, 2013.
- [Guo14] Philip J. Guo. How video production affects student engagement: An empirical study of MOOC videos. 2014. https://www.researchgate.net/profile/Juho_Kim3/publication/262393281_How_video_production_affects_student_engagement_an_empirical_study_of_MOOC_videos/links/53d51e170cf220632f3d49f5.pdf.

- [Gup12] Sanjay Gupta. Khan Academy: The future of education? <https://www.youtube.com/watch?v=9kh60v5PxMk&t=5m58s>, 2012.
- [Hib] Hibernate. Hibernate Manual (Version 4.3). http://docs.jboss.org/hibernate/orm/4.3/manual/en-US/html_single/ (Last accessed: 13.02.2017).
- [Hol16] Andreas Holzer. Formal Methods in Computer Science (SS'16). Evaluation Course for our Interactive Lecture Notes application, published on elearning.forsyte.at (Last accessed: 18.01.2017), 2016.
- [HT14] Fiona M Hollands and Devayani Tirthali. MOOCs: Expectations and Reality. Full report. *Online Submission*, 2014. <http://files.eric.ed.gov/fulltext/ED547237.pdf>.
- [Inca] Crunchbase Inc. Crunchbase Organization Details for Udacity. <https://www.crunchbase.com/organization/udacity> (Last accessed: 16.12.2016).
- [Incb] EdX Inc. Building and Running an Open edX Course: Eucalyptus Release. <http://edx.readthedocs.io/projects/open-edx-building-and-running-a-course/en/open-release-eucalyptus.master/> (Last accessed: 16.12.2016).
- [Incc] EdX Inc. Documentation for edx.org and the Open edX Community. <http://docs.edx.org/> (Last accessed: 16.12.2016).
- [Incd] EdX Inc. EdX Edge. <https://edge.edx.org> (Last accessed: 16.12.2016).
- [Ince] Udacity Inc. Self-Driving Car Engineer. <https://www.udacity.com/drive> (Last accessed: 16.12.2016).
- [Inc13] EdX Inc. EdX Open Source Platform Chosen to Power China's New Online Education Portal. <https://www.edx.org/press/edx-open-source-platform-chosen-power> (Last accessed: 16.12.2016), 2013.
- [jav] Java Servlet Specification (Version 3.0). http://download.oracle.com/otn-pub/jcp/servlet-3.0-fr-eval-oth-JSpec/servlet-3_0-final-spec.pdf (Last accessed: 13.02.2017).
- [Jor14] Katy Jordan. Initial trends in Enrolment and Completion of Massive Open Online Courses. *The International Review of Research in Open and Distributed Learning*, 15(1), 2014. <http://www.irrodl.org/index.php/irrodl/article/viewFile/1651/2813> (Last accessed: 2017-06-15).

- [Jor15] Katy Jordan. MOOC completion rates: The data. *Available at: <http://www.katyjordan.com/MOOCproject.html>*, 2015.
- [KA15] B.H. Khan and M. Ally. *International Handbook of E-Learning Volume 1: Theoretical Perspectives and Research*. Number v. 2 in Routledge Handbooks Online. Taylor & Francis, 2015.
- [Kha] Khan Academy. About Khan Academy. <https://www.khanacademy.org/about> (Last accessed: 06.06.2017).
- [Kha11] Saman Khan. Salman Khan, TED Talk. https://www.ted.com/talks/salman_khan_let_s_use_video_to_reinvent_education?language=en#, 2011.
- [Kha12] Salman Khan. MIT Talk with Salman Khan. <https://www.youtube.com/watch?v=z9JCpMCQ5qM>, 2012.
- [Kha13a] Salman Khan. Questionsfor AACC. <https://youtu.be/NTVOMifZ-Rk?t=11s> (Last accessed: 06.12.2016), 2013.
- [Kha13b] Salman Khan. Salman Khan talking about Mastery Learning. <https://www.youtube.com/watch?v=Q7VrfzxLxuk&feature=youtu.be&t=1115>, 2013.
- [kha16] khanacademy.org Traffic Statistics. <http://www.alexa.com/siteinfo/khanacademy.org> (Last accessed: 09.12.2016), 2016.
- [Kid09] Terry T Kidd. A brief history of eLearning. *Online Education and Adult Learning: New Frontiers for Teaching Practices: New Frontiers for Teaching Practices*, page 46, 2009. http://www.academia.edu/download/30802459/1605668303_20Online_20Education.pdf#page=69.
- [KNDC13] Daphne Koller, Andrew Ng, Chuong Do, and Zhanghao Chen. Retention and Intention in Massive Open Online Courses: In Depth. 2013. <http://er.educause.edu/articles/2013/6/retention-and-intention-in-massive-open-online-courses-in-depth>.
- [Kne11] Knewton. Flipped Classroom. <https://www.knewton.com/infographics/flipped-classroom/> (Last accessed: 06.12.2016), 2011.
- [LB06] Richard Lambert and Nick Butler. The future of European universities. *Renaissance or decay*, 2006.
- [Lef16] Jochen Leffers. Ein Herz für die Lehre. <http://www.spiegel.de/lebenundlernen/uni/exzellenz-initiative-ein-herz-fuer-die-lehre-a-460824.html> (Last accessed: 06.12.2016), 2016.

- [LPT00] Maureen J Lage, Glenn J Platt, and Michael Treglia. Inverting the classroom: A gateway to creating an inclusive learning environment. *The Journal of Economic Education*, 31(1):30–43, 2000.
http://www.academia.edu/download/2604611/Inverted_Classroom_Paper.pdf.
- [LSK] Jörn Loviscach and Miriam Swords Kalk. Differential Equations in Action. <https://www.udacity.com/course/differential-equations-in-action--cs222> (Last accessed: 16.12.2016).
- [LWC⁺14] Cameron Lister, H. Joshua West, Ben Cannon, Tyler Sax, and David Brodegard. Just a Fad? Gamification in Health and Fitness Apps. *JMIR Serious Games*, 2(2):e9, Aug 2014.
- [MGK⁺14] Robert Murphy, Larry Gallagher, Andrew E Krumm, Jessica Mislevy, and Amy Hafter. Research on the use of Khan Academy in schools: Research brief. 2014.
- [Min14] Steven Mintz. Five Ways that 21st and 20th Century Learning Will Differ. <https://www.insidehighered.com/blogs/higher-ed-beta/five-ways-21st-and-20th-century-learning-will-differ> (Last accessed: 09.12.2016), 2014.
- [mooa] Flipping Your Classroom With Moodle. <http://moodle.oakland.k12.mi.us/os/course/view.php?id=1361> (Last accessed: 19.12.2016).
- [Moob] Moodle. Learn Moodle. <https://learn.moodle.net/> (Last accessed: 19.12.2016).
- [Mooc] Moodle. Moodle Activities. <https://docs.moodle.org/31/en/Activities> (Last accessed: 19.12.2016).
- [Mood] Moodle. Moodle Assignments Activity. https://docs.moodle.org/31/en/Assignment_activity (Last accessed: 19.12.2016).
- [Mooc] Moodle. Moodle Calendar. <https://docs.moodle.org/31/en/Calendar> (Last accessed: 19.12.2016).
- [Moof] Moodle. Moodle Choice Activity. https://docs.moodle.org/31/en/Choice_activity (Last accessed: 19.12.2016).
- [Moog] Moodle. Moodle Features. <https://docs.moodle.org/31/en/Features> (Last accessed: 19.12.2016).
- [Mooh] Moodle. Moodle Feedback Activity. https://docs.moodle.org/31/en/Feedback_activity (Last accessed: 19.12.2016).

- [Mooi] Moodle. Moodle Glossary Activity. https://docs.moodle.org/31/en/Glossary_activity (Last accessed: 19.12.2016).
- [Mooj] Moodle. Moodle Lesson Activity. https://docs.moodle.org/31/en/Lesson_activity (Last accessed: 19.12.2016).
- [Mook] Moodle. Moodle Messaging. <https://docs.moodle.org/31/en/Messaging> (Last accessed: 19.12.2016).
- [Mool] Moodle. Moodle Quiz Activity. https://docs.moodle.org/31/en/Quiz_activity (Last accessed: 19.12.2016).
- [Moom] Moodle. Moodle Survey Activity. https://docs.moodle.org/31/en/Survey_activity (Last accessed: 19.12.2016).
- [Moon] Moodle. Moodle Wiki Activity. https://docs.moodle.org/31/en/Wiki_activity (Last accessed: 19.12.2016).
- [Moo13] Moodlerooms. Flip your classroom with Moodle Tools. <https://www.moodlerooms.com/2013/05/01/flip-your-classroom-moodle-tools/> (Last accessed: 19.12.2016), 2013.
- [MRG⁺14] Jacqueline E McLaughlin, Mary T Roth, Dylan M Glatt, Nastaran Gharkholonarehe, Christopher A Davidson, LaToya M Griffin, Denise A Esserman, and Russell J Mumper. The flipped classroom: a course redesign to foster learning and engagement in a health professions school. *Academic Medicine*, 89(2):236–243, 2014. <http://dx.doi.org/10.1097/ACM.0000000000000086> (Last accessed: 09.12.2016).
- [Net] Netflix. Internet Connection Speed Recommendations. <https://help.netflix.com/en/node/306> (Last accessed: 26.05.2017).
- [Ngi] Nginx. Nginx. <http://nginx.org/> (Last accessed: 28.05.2017).
- [Nic98] Raymond S Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175, 1998.
- [OGI12] Peter F Oliva and William R Gordon II. *Developing the curriculum*. Pearson Higher Ed, 2012.
- [Pan] Thomas Pani. Kripke Builder. <http://forsyte.at/kripke/> (Last accessed: 07.07.2018).
- [Pap12] Laura Pappano. The Year of the MOOC. <http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html>, 2012.

- [Par13] Chris Parr. Times Higher Education. <https://www.timeshighereducation.com/news/mooc-creators-criticise-courses-lack-of-creativity/2008180>. article, 2013.
- [Plo93] Scott Plous. *The psychology of judgment and decision making*. McGraw-Hill Book Company, 1993.
- [PM99] Niels Provos and David Mazières. A Future-Adaptable Password Scheme. *USENIX Annual Technical Conference*, 1999. https://www.usenix.org/legacy/publications/library/proceedings/usenix99/provos/provos_html/index.html.
- [RMR08] Rhonda Robinson, Michael Molenda, and Landra Rezabek. Facilitating learning. *Educational technology: A definition with commentary*, pages 15–48, 2008. http://www.aect.org/publications/EducationalTechnology/ER5861X_C002.pdf.
- [Ruh] Umar Ruhi. Level Up Your Strategy: Towards a Descriptive Framework for Meaningful Enterprise Gamification. <http://timreview.ca/article/918> (Last accessed: 16.12.2016).
- [Scr67] Michael S Scriven. The methodology of evaluation (Perspectives of Curriculum Evaluation, and AERA monograph Series on Curriculum Evaluation, No. 1). *Chicago: Rand McNally*, 1967. <http://www.comp.dit.ie/dgordon/Courses/ILT/ILT0005/TheMethodologyOfEvaluation.pdf>.
- [Sev12] C. Severance. Teaching the World: Daphne Koller and Coursera. *Computer*, 45(8):8–9, August 2012. <http://dx.doi.org/10.1109/MC.2012.278>.
- [Sha15] Abhinav Sharma. Will the material in Coursera be available after the class is over? <https://www.quora.com/Will-the-material-in-Coursera-be-available-after-the-class-is-over> (Last accessed: 16.12.2016), 2015.
- [Shu13] Daniel Shumski. MOOCs by the numbers: How do EdX, Coursera and Udacity stack up? <http://www.educationdive.com/news/moocs-by-the-numbers-how-do-edx-coursera-and-udacity-stack-up/161100/> (Last accessed: 16.12.2016), 2013.
- [spra] Spring Boot Reference Documentation (Version 1.3.3). <http://docs.spring.io/spring-boot/docs/1.3.3.RELEASE/reference/pdf/spring-boot-reference.pdf> (Last accessed: 13.02.2017).

- [sprb] Spring Boot, Version 1.3.3.
<http://projects.spring.io/spring-boot/> (Last accessed: 13.02.2017).
- [sprc] Spring Framework, Version 4.1.2.
<http://projects.spring.io/spring-framework/> (Last accessed: 13.02.2017).
- [sprd] Spring Security, Version 4.1.2.
<http://projects.spring.io/spring-security/> (Last accessed: 13.02.2017).
- [Spre] Stephen Sproule. On demand courses. <https://learner.coursera.help/hc/en-us/community/posts/205780125-On-demand-courses> (Last accessed: 16.12.2016).
- [SSH15] Christian M Stracke and Tatiana Shamarina-Heidenreich. *The Need for Change in Education: Openness as Default?* 2015.
http://dspace.ou.nl/bitstream/1820/6286/1/LINQ_2015_Proceedings.pdf.
- [Swo14] John Swope. A Comparison of Five Free MOOC Platforms for Educators. <http://www.edtechmagazine.com/higher/article/2014/02/comparison-five-free-mooc-platforms-educators>, 2014.
- [Tho11] Clive Thompson. How Khan Academy is changing the rules of education. *Wired Magazine*, 126:1–5, 2011.
- [Tuc12] Bill Tucker. The flipped classroom. *Education next*, 12(1), 2012. http://www.msuedtechsandbox.com/MAETELy2-2015/wp-content/uploads/2015/07/the_flipped_classroom_article_2.pdf (Last accessed: 09.12.2016).
- [Tug16] Alina Tugend. How Public Universities Are Addressing Declines in State Funding. <http://www.nytimes.com/2016/06/23/education/how-public-universities-are-addressing-declines-in-state-funding.html>, 2016.
- [Udaa] Udacity. 5 study tips for Udacity students. <http://blog.udacity.com/2012/05/start-off-on-right-foot-5-tips-for.html> (Last accessed: 16.12.2016).
- [Udab] Udacity Inc. About Us. <https://www.udacity.com/us> (Last accessed: 16.12.2016).
- [Wie] Technische Universität Wien. <https://tuwel.tuwien.ac.at> (Last accessed: 19.12.2016).

- [Wil12] David Willey. The MOOC Misnomer.
<http://opencontent.org/blog/archives/2436>, 2012.
- [YCSW15] Ahmed Mohamed Fahmy Yousef, Mohamed Amine Chatti, Ulrik Schroeder, and Marold Wosnitza. A usability evaluation of a blended MOOC environment: An experimental case study. *The International Review of Research in Open and Distributed Learning*, 16(2), 2015.
<http://www.irrodl.org/index.php/irrodl/rt/prINTERfriendly/2032/3270>.

Thank you for your attention.

Last but not least, I would like to thank the university for the chance to write this thesis. I thank my supervisors for their support and advice. Finally, I would like to thank my past and current employers for providing a flexible working environment in my job that gave me the required time to write this thesis.