

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTTLICHE ERKLÄRUNG

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

Graz, am

.....
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

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

.....
date

.....
(signature)

Acknowledgement

To begin with, I would like to thank my supervisor Martin Ebner for his support, advice, and confidence in me during the development of the software prototype and writing this thesis. I would also wish to thank my colleague Gerald Geier for his support with the layout of the software and the ideas he put forward for the improvement of the layout and design.

Furthermore, I would like to thank the students and the teacher who consent to help me with the evaluation of the software. I would also like to thank all my colleagues who supported me with ideas and discussions for this thesis.

In particular I am grateful to my parents Ursula and Helmut, who supported me financially and emotionally during all the years of my studies. Finally, I would like to thank Julia, one of the most important persons in my life, for her understanding, patience, and emotional support in busy times during my studies.

Graz, May 2015

Robert Gröbl

Abstract

The constantly increasing Internet usage and the resulting amount of user generated data bears great opportunities for progress in various fields of research. Education sectors are no exception and researchers of educational fields have tried to derive meaningful information about learning processes from collected learner produced data over the last decades. The research area learning analytics is primarily concerned with the collection and analysis of learner related data and the development of tools which aim at informing and supporting learners and teachers by comprehensible analysis models. Learning analytics tries to understand learning processes as a whole. This often leads to the development of very complex and incomprehensible analysis tools which are useless in a school context. Therefore, a significant aspect regarding the development of analysis tools with focus on learning analytics is the simple representation of analysed data to allow meaningful interpretation by stakeholders.

The aim of this thesis is the conception and implementation of an information system for pupils of the first grade in secondary school. The application named “VOCABTrainer” offers learners the opportunity of quickly test their vocabulary knowledge from English to German and vice versa. The software supports students in the acquisition of vocabulary by immediate feedback about validity of their answers, and in addition, provides tools for students and their teachers which allow long-term analysis of students’ test performances and consequently reveal detailed insights into the learning progress and process of learners.

Basically this thesis consists of two parts. The first part provides an overview of learning analytics and points out the aims of this research area. Furthermore, this part elucidates theoretical approaches to language teaching, which try to shed light on the process of vocabulary acquisition and constitute the basis for the development of the information system. Furthermore, a number of available vocabulary training software was reviewed and evaluated. The second part is concerned with the development of the software and describes the conception, design and implementation of the vocabulary trainer. Moreover, the developed prototype was tested in an Austrian secondary school, which resulted in a first set of student produced data for analysis and the validation of the prototype, which is discussed in the last part of this thesis.

Kurzfassung

Die ständig steigende Nutzung des Internets und die daraus resultierende Menge an generierten Daten durch Benutzer und Benutzerinnen stellt für viele Forschungsbereiche eine große Chance zur Weiterentwicklung dar. Der Bildungssektor ist dabei keine Ausnahme und es wird seit einigen Jahren versucht nützliche Informationen über Lernprozesse aus gesammelten Daten zu gewinnen. Der Forschungsbereich Learning Analytics beschäftigt sich genau mit diesem Thema und versucht durch die Sammlung und Analyse von Lerner bezogenen Daten Werkzeuge zu entwickeln, welche Lernende und Lehrende durch die Auswertung dieser Daten unterstützen sollen. Learning Analytics versucht den Lernprozess als Ganzes zu verstehen, was oftmals zur Entwicklung von sehr komplexen Analysewerkzeugen führt die im Schulbereich nicht angewendet werden können. Ein wichtiger Aspekt in der Entwicklung von Werkzeugen die auf Learning Analytics abzielen ist deshalb die einfache Darstellung von Analysen, damit Stakeholder einen Nutzen daraus ziehen können.

Das Ziel dieser Arbeit ist die Konzeption und Implementierung eines Informationssystems für Schülerinnen und Schüler der ersten Klasse AHS und NMS. Die Software mit dem Namen „VOCABTrainer“ bietet Lernenden die Möglichkeit der raschen Vokabelüberprüfung von Englisch auf Deutsch und umgekehrt. Das Programm unterstützt Lernende einerseits durch eine sofortige Rückmeldung über die Richtigkeit ihrer Antworten und bietet zusätzlich Werkzeuge für Lernende und Lehrende die eine langzeit Analyse der Testleistungen erlauben und somit detaillierte Einblicke in den Lernverlauf und Lernprozess von Lernenden erlauben.

Die Arbeit besteht grundsätzlich aus zwei Abschnitten. Der erste Teil liefert einen Überblick über Learning Analytics und zeigt die Ziele dieses Forschungsbereichs. Weiters erläutert dieser Teil theoretische Ansätze, welche versuchen Aufschlüsse über den Prozess des Vokabellernens zu liefern, und eine Basis für die Entwicklung des Informationssystems darstellen. Außerdem wurden bereits erhältliche Vokabeltrainer untersucht und evaluiert. Der zweite Teil beschäftigt sich mit der Entwicklung der Software und beschreibt die Konzeption, Planung und Implementierung des Vokabeltrainers. Außerdem wurde die Software an einer österreichischen AHS getestet. Diese Evaluation lieferte erste Daten für die Analyse und die Bewertung der Software, was im letzten Abschnitt der Arbeit diskutiert wird.

Contents

ACKNOWLEDGEMENT	III
ABSTRACT	IV
KURZFASSUNG	V
1 INTRODUCTION	1
1.1 STRUCTURE OF THIS THESIS	2
2 LEARNING ANALYTICS	4
2.1 LAK AND EDM	4
2.2 LEARNING ANALYTICS CYCLE	8
3 VOCABULARY ACQUISITION	11
3.1 THE AUSTRIAN CURRICULUM FOR FOREIGN LANGUAGES	11
3.2 LEXICAL APPROACH	13
3.3 COMPETENCE AND PERFORMANCE	15
3.4 MENTAL LEXICON	16
3.5 VOCABULARY TESTING	17
4 LANGUAGE LEARNING SOFTWARE	19
4.1 BABEL	19
4.2 LANGUAGECOURSE.NET – VOCABULARY TRAINER	21
4.3 LEO DICTIONARY – LANGUAGE TRAINER	22
4.4 LINGOVERSITY	22
4.5 SUMMARY	23
5 CORE VOCABULARY	25
6 SOFTWARE CONCEPT	27
6.1 REQUIREMENTS AND ASSUMPTIONS	27
6.2 TEXTUAL DESCRIPTION OF THE CONCEPT	29
6.2.1 <i>Manage/Browse vocabulary mode</i>	30
6.2.2 <i>Test mode</i>	31
6.2.3 <i>Analyze mode</i>	31
6.3 USE CASES	33
7 TECHNICAL IMPLEMENTATION	34
7.1 USED WEB TECHNOLOGIES	35
7.1.1 <i>PHP</i>	35
7.1.2 <i>MySQL</i>	36
7.1.3 <i>jQuery</i>	37
7.1.4 <i>CSS and HTML</i>	37
7.2 BASIC COMPONENT OVERVIEW	37
7.3 DESIGN	39

7.3.1	<i>The MVC architecture</i>	39
7.3.2	<i>Database</i>	40
7.3.3	<i>Class diagram</i>	43
7.4	IMPLEMENTATION.....	45
7.4.1	<i>Basic system</i>	46
7.4.2	<i>Browse and manage vocabulary modes</i>	48
7.4.3	<i>Test mode</i>	51
7.4.4	<i>Analyze mode</i>	56
7.5	LAYOUT	61
7.5.1	<i>Basic layout</i>	61
7.5.2	<i>Browse and manage vocabulary modes</i>	63
7.5.3	<i>Test mode</i>	65
7.5.4	<i>Analyze mode</i>	66
8	EVALUATION	70
8.1	PREPARATION	70
8.2	PROCEDURE.....	71
8.3	RESULTS	72
8.4	FEEDBACK.....	74
9	DISCUSSION AND OUTLOOK	75
10	CONCLUSION	81
	BIBLIOGRAPHY	83
	LIST OF FIGURES	86
	LIST OF SOURCE CODE	87
	APPENDIX	88
A.	FEEDBACK SHEET	88
B.	USER ACCOUNTS	89

1 Introduction

Vocabulary learning is an integral component in the acquisition of foreign languages and is considered especially important at the beginning of the learning process. In the context of Austrian schools students usually start learning English as a second language in first grade at the age of about ten years. At this age many students face issues with time management or simply lack interest in learning the language, which often results in very irregular and infrequent vocabulary learning. Often vocabulary is learned only shortly before exams when it is absolutely necessary, and forgotten immediately after the exam. In many cases this behavior stays undetected in lower grades and ultimately leads to serious communication issues in higher grades, when students face more difficult tasks, but completely lack basic vocabulary. One reason for this behavior could be that all other forms of vocabulary testing do not or only very little influence the students' grading as vocabulary testing, written or orally, must not be graded by teachers but only serves the purpose of giving feedback to their students. Thus, a very commonly employed practice among language teachers is to take notes of vocabulary tests and take these results to assess the students' cooperation in lessons, although this practice is illegal. Vocabulary learning is not only a very demanding task for students, but also for teachers as it is their task to keep track of their students' vocabulary development. This demands teachers to take notes of all of their students' contributions regarding vocabulary learning to keep track of individual issues and to detect general issues concerning many students of a class, which is utterly time consuming inside and outside of the classroom.

The research area learning analytics focuses on the computer based gathering and analysis of great amounts of data produced by students and provides features for teachers and learners which allow meaningful interpretations. Literature on the research area learning analytics points out that a crucial aspect of learning analytics is the easy readability of gained data, which allows teachers, students and other stakeholders to draw immediate conclusions. Active scientists in this field of research already developed a number of tools, which try to provide a useful overview of students' performances regarding various school subjects. In recent years a number of programs, which aim at assessing students performances in mathematics, history and other subjects appeared on the internet. Learning analytics as it is defined is a teacher centered approach and mainly aims at informing teachers about strengths and weaknesses of

their students. Thus, it is important to highlight that learning analytics is not meant to actively support students in the actual learning process, but rather tries to inform teachers (as well as students) about the learners' current state of knowledge, and therefore, allows early intervention. In addition to these programs there appeared other programs closely related to the research area of learning analytics, which aim at assessing students' behaviour while learning and capture useful information about the students' learning environment. Such programs basically have the same aim as they try to improve learning by collecting great amounts of data, but differ from learning analytics in so far as they follow a different approach by focusing on context information rather than actual content. This research area is called educational data mining.

The aim of this thesis is to develop a functional prototype of a vocabulary trainer for beginners of English with focus on learning analytics. Vocabulary learning is a highly complex process, which is emphasised by the fact that literature on vocabulary learning provides various different theories on how human brains store vocabulary in the so called mental lexicon, but none of these theories seems to provide neurological evidence. However, studies which followed the lexical approach to language teaching tend to describe the mental lexicon as an interconnected net or web of semantics and word syntax or form and provide empirical data to underpin these theories. (cf. Coady and Huckin 1997, Aitchison 2012) Although, these studies also lack neurological evidence they seem very plausible and are most likely to be true. For this reason, the lexical approach to language teaching is taken as a basis for the conception and implementation of the software. As the focus of the software is on learning analytics a main aspect of the trainer is the reasonable processing and representation of collected data, to point out possible problem areas of learners.

1.1 Structure of this thesis

This thesis basically consists of three parts. The first part provides theoretical background which forms the basis for the development of the software. The second part is concerned with the concept, design and implementation of the software. The third part consists of an evaluation of the developed prototype, a discussion of the software and the results of the evaluation, and provides an outlook for further development. This thesis concludes with a brief summary of the main findings.

Chapter 2 of this thesis provides a brief overview of learning analytics by comparison of several definitions. Furthermore, this chapter briefly introduces the research area educational data mining and highlights the differences to learning analytics.

Chapter 3 is concerned with vocabulary acquisition in general. Thus, it points out the status of vocabulary learning in Austrian schools, explains the main features of vocabulary learning according to the lexical approach, describes the mental lexicon and touches upon the topic of vocabulary testing.

Chapters 4 and 5 are rather short chapters. The former provides a brief review of free-to-use vocabulary learning software and points out the main difference to the software developed for this thesis. The latter discusses the term core vocabulary that is often used to refer to vocabulary seen as crucial for beginners of a language. This chapter also explains the choice of vocabulary available in the software.

Chapter 6 is concerned with the concept of the software. This chapter starts with a brief description of the conceptualization process in general. Then the requirements and assumptions, which developed through several iterations of refining the concept, are listed and explained. The core of this chapter is a textual description of the final concept which links the theory to the requirements and assumptions.

Chapter 7 of this thesis is the largest chapter as it is concerned with all technical aspects of the software. Thus, starting with an explanation of used web technologies, then the design of the software is explained, then some main technical features and important functions of the software are described in greater detail and finally the layout of the finished prototype is presented.

Chapter 8 explains the preparation and procedure of the evaluation and summarizes the feedback, results and problems. The results of the evaluation are further discussed in chapter 9 and several ideas for further development are discussed.

2 Learning analytics

The aim of this chapter is to briefly describe the relatively new discipline of learning analytics and knowledge (LAK) by contrasting it to the very closely related research area of educational data mining (EDM). This chapter is mainly based on Neuhold (2013) as his theoretical findings on learning analytics closely correlate with the findings of the research done for this thesis.

2.1 LAK and EDM

Internet usage is rising daily and so is the amount of data produced by individual users. Data is shared between institutions of various branches and places all around the globe and educational institutions are no exception. (cf. Piety 2013) Regarding language learning there are endless examples of websites which offer some kind of learning resource. These resources range from simple textual grammar explanations, over online dictionaries, to language learning course platforms, etc. Such learning resources in most cases rely on dynamic user interaction and by doing so they produce a great amount of very useful data. If effectively analyzed and comprehensibly visualized this data could bear a great chance for education, as it could be used by teachers and positively influence their decision making with individual students. However, according to Siemens and Long educational institutions have failed to use this data effectively in the past:

“Higher education, a field that gathers an astonishing array of data about its ‘customers’, has traditionally been inefficient in its data use, often operating with substantial delays in analyzing readily evident data and feedback. [...] Something must change.” (Siemens and Long 2011)

A result of the heavy use of the internet for learning is an enormous growth of data about learners’ behaviour. In contrast to traditional forms of learning, as reading a book, or listening to a teacher, learning on the internet leaves traces of every interaction. Thus, every click and all other interactions between students and their computers can be captured while they are learning and retraced for later analysis. The learners’ produced data can be merged and analysed to gain insights into the learners’ learning process. These insights can lead to an

early detection of problems in students' learning processes and enable teachers to actively intervene in their learning processes to solve these problems effectively. (cf. Siemens and Long 2011)

Siemens and Baker (2012) state that the two research areas of learning analytics and knowledge, and educational data mining specifically focus on BigData in education as the demand for effective and meaningful analysis of this data increases. Although, both research areas are concerned with BigData in educational contexts and share the same overall aim of improving learning, they slightly differ in their approaches. The International Educational Data Mining Society defines their research area EDM as follows:

“Educational Data Mining is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in.” (International Educational Data Mining Society 2015)

LAK is the measurement, storage, analysis and visualisation of data produced by learners and the context in which learning takes place. Thus, the goal of LAK is to explore, understand and improve the processes underlying learning and the contexts in which these processes happen. In the approach taken by LAK especially teachers are foregrounded as it adopts a teacher centered approach. Both research areas, LAK and EDM, take an approach which depends on great amount of data and try to positively influence educational institutions. The analysis of BigData to provide teachers with help in their planning, intervention and decision making is a very new and innovative approach, which demands new technologies and methods to process the data. (cf. Neuhold 2013, Siemens and Baker 2012)

Although, LAK and EDM are very similar in many aspects they still show slight but significant differences in their approaches. Educational data mining focuses on automatic detection of patterns, whereas LAK is concerned with supporting the decision making of stakeholders, especially teachers. For this reason, EDM puts great effort in the development of intelligent computer systems with the goal of automatized decision making by these systems. In contrast, LAK aims at creating models by analysis of learner produced data to inform teachers and learners, and thus, support them in their decision making process. Another significant difference can be found in the exploration and understanding of systems (underlying learning). While EDM in its approach splits large and complex systems in smaller

components and analyses these components, LAK seeks to analyse and understand the process as a whole in its full complexity. A third important difference can be found in the origin of both research areas. EDM has its roots in educational software and student modeling software and is rather concerned with analysis of students relations to their educational institutions to predict overall course developments as for example student drop outs, etc. In contrast, LAK developed from research areas such as semantic web and intelligent curriculum, and thus, focuses on the analysis of students' learning processes regarding single subjects. (cf. Neuhold 2013)

Siemens and Baker (2012) argue that especially these slight differences of EDM and LAK constitute a great advantage for both research areas. The different focus of both fields of research offers scientists the opportunity to learn from each other. The authors emphasize that it is vitally important to improve the communication between both communities to maximize the gains for educational institutions, science and stakeholders. According to Baker et al. (2012) LAK and EDM should become best friends, to ensure the exchange of valuable findings and knowledge between these research areas and consequently accelerate and push progress in the study of learning processes.

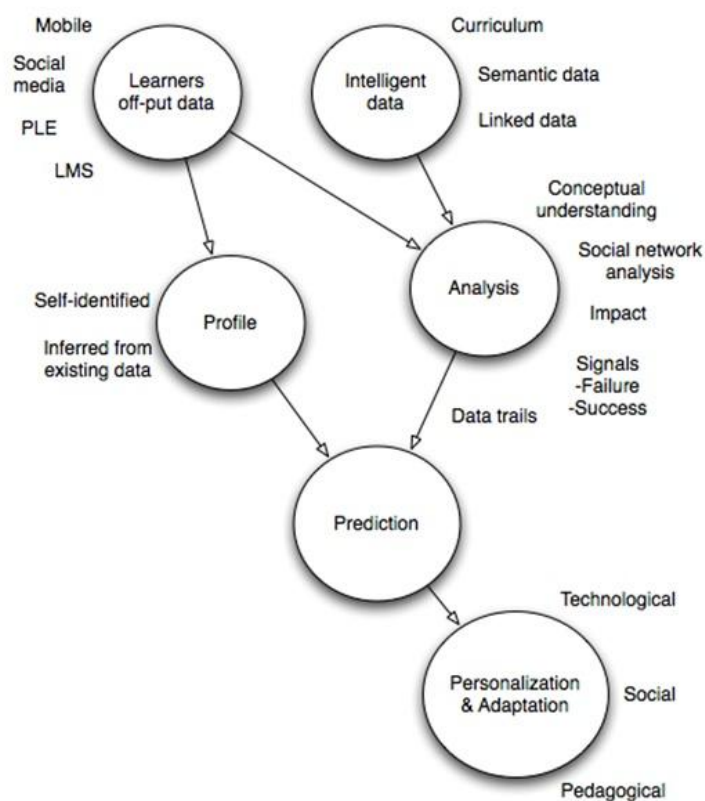


Figure 1: The process of learning analytics (Siemens 2010)

Siemens defines learning analytics (LA) in another article as follows:

“Learning analytics is the use of intelligent data, learner-produced data, and analysis models to discover information and social connections, and to predict and advise on learning.” (Siemens 2010)

Figure 1 illustrates a possible approach towards the process of learning analytics as Siemens defines it. (cf. Siemens 2010) Learners constantly produce data during their learning processes. This data includes their use of social media, learning management systems, etc. LA tries to capture and collect the data and stores it in a database. The gained information is extended by an individual profile of the user, which is created by data from various sources across the internet (Facebook, Twitter, etc.). Neuhold states that all the collected data is then transformed into intelligent data through semantic processes. The explanation that Siemens provides for intelligent data in his picture is from my point of view a bit vague, but I had rather interpreted it as follows. Based on figures 1 and 2 and the explanations Siemens provides for both images, intelligent data is a set of data which is based on and retrieved through subjects (and their underlying theories) of a curriculum on the one hand, and on the other hand also includes student data gathered by educational institutions. Then this intelligent data is used as a basis for comparison to an individual students’ off-put and his current learning objective, which allows a kind of competence measurement and ultimately could lead to useful predictions and feedback for teachers and learners. After the comparison the data gathered about the students’ performance is added to the set of intelligent data. Furthermore, the gained information could be used for personalization and adaption of the context in which learning takes place. Educational institutions could detect individual learning difficulties earlier and chances to successfully solve possible problems increase. (cf. Neuhold 2013, Siemens 2010)

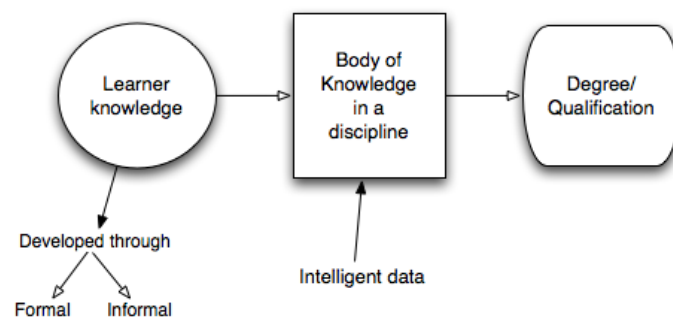


Figure 2: Application of learning analytics (Siemens 2010)

Baker et al. (2012) describe learning analytics as the collection of traces which learners leave. These traces help learning analytics to understand and improve learning by providing feedback and informing teachers and learners about the current state of knowledge on the side of the learners. According to the authors, it is significantly important to keep visual representations and feedback simple as too much insignificant information might rather result in confusion on the side of the stakeholders and does not allow reasonable interpretation. (cf. Neuhold 2013)

The 1st International Conference on Learning Analytics & Knowledge defines learning analytics as follows:

“The measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs.” (1st International Conference on Learning Analytics & Knowledge 2011)

This definition covers a large area of research and emphasizes the many aspects and facets learning analytics is concerned with.

2.2 Learning analytics cycle

Learning analytics is a teacher centered approach and seeks to support teachers in their decision making processes and provide an overview of possible interventions. In order to reach this goal analysis and interpretation of student data is necessary. Campell and Oblinger (2007) provide a model to logically describe the analysis process in five steps: capture, report, predict, act and refine. Clow (2012) takes these five steps as a basis to define the learning analytics cycle, which is illustrated in figure 3.

Clow (2012) describes the learning analytics cycle as a closed and iterative process consisting of four main components. The first of these components are learners. Whether these learners are university students, pupils or other learners is not significant, as long as they use any form of computer based learning environment.

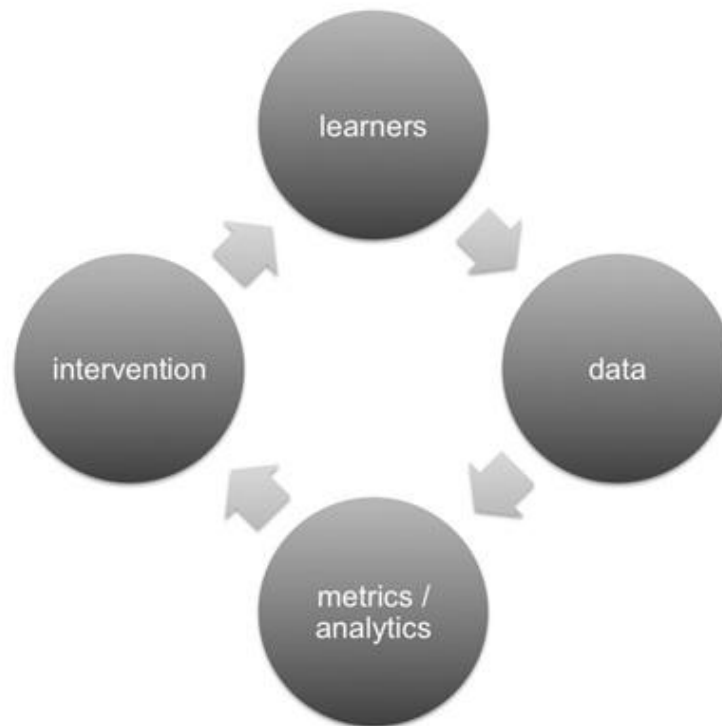


Figure 3: Learning analytics cycle (Clow 2012)

The second component is data, which is concerned with the production and collection of learner data. The collected data might include information about the learners, geographical data, logdata of learning environments, activities on community boards, results of online exams, etc. Learning analytics tries to produce and collect all relevant data about learners effectively and automatically.

The third component in the learning analytics cycle is metrics/analytics which is the core of learning analytics. In this step the collected data from the former step is used as a basis for analysis to gain insights into the learning processes of individual students. This step includes the creation of meaningful statistics and visual representations of the collected data to inform teachers. Researchers from this area have already developed very useful tools as for example, network analysis tools, prediction systems, dashboards, etc. Such tools are significantly useful for teachers as they might reveal information about their students which they might not access through conventional teaching methods.

The fourth and last component is intervention. On the one hand this step could be realised through statistics and visual representations directly provided for learners. On the other hand such statistics and visual representations could be provided for teachers, who then interpret

them, and eventually inform their students and intervene in their learning processes. As it is utterly important that assumptions gained through this process really inform and help students it seems more efficient and meaningful that interventions are performed by teachers, as it might be very difficult, especially for young learners, to draw conclusions about their own learning processes.

A tool that focuses on learning analytics not necessarily has to consist of all of these four components, as for example the production and collection of data might be done by another tool. However, a tool which compiles statistics and other visual representations about learners' learning processes, but does not provide this data to stakeholders will probably not be successful. (cf. Neuhold 2013, Clow 2012)

3 Vocabulary acquisition

The aim of this chapter is to summarize theoretical findings on vocabulary acquisition and point out significant theoretical findings which are used as basis for the concept and design of the vocabulary trainer developed for this thesis. Thus, a first necessary step is to assess the importance, methods of and approaches towards teaching vocabulary in Austrian schools, which will be done by taking a closer look at the Austrian curriculum for foreign languages (“Lehrplan für lebende Fremdsprachen”). This step is important as it not only helps to classify and choose a set of vocabulary for the target group of learners, but it also helps to narrow down the focus on specific linguistic approaches to foreign language acquisition used as a basis for the concept of the program.

After examining the Austrian curriculum, this chapter will briefly describe the lexical approach to language acquisition and teaching. By doing so, the difference between recognizing words and knowing words will be point out. Then this chapter tries to explain how linguistic theory can help to assess whether learners really know (have learned) a specific lexical unit. Furthermore, this chapter will provide theoretical findings on the mental lexicon, testing of vocabulary and finally relate these findings to learning analytics to sum up the theoretical basis for the development of the vocabulary trainer.

3.1 The Austrian curriculum for foreign languages

Basically, the Austrian curriculum for foreign languages follows the communicative approach to language teaching. According to Newby (2011), the communicative approach prefers the acquisition of grammatical meanings in context to the acquisition of rules. Thus, this approach introduces the concept of notions to refer to grammatical concepts. Overall the communicative approach is centered on grammar, and pays only very little attention to vocabulary. For this reason, the communicative approach alone is not sufficient enough for the development of software which focuses on vocabulary. Regarding the teaching of vocabulary the curriculum states the following:

“Kontextualisierung von Wortschatz und Grammatik

Der Vermittlung von Wortschatz und Grammatik in vielfältig kontextualisierter und vernetzter Form ist größtes Gewicht beizumessen, zB ist Vokabular, wo immer möglich, in Kollokationen, Redewendungen und Phrasen mit impliziter Grammatik einzubetten.

[...] Wo es sinnvoll ist, sind grammatische Strukturen besser ohne Regelformulierung als lexikalische Einheiten zu vermitteln.” (BMUKK 2015)

This definition clearly reflects the communicative approach as it demands teachers to teach grammar in combination with vocabulary in various contexts and pay little attention to grammar rules. Regarding vocabulary teaching the curriculum does not give any other guidelines as to teach vocabulary if possible in collocations and phrases with implicit grammar. In addition, the curriculum does not state a single word about testing vocabulary. From my point of view, as a future language teacher, the guideline given by the curriculum is insufficient as the acquisition of vocabulary is seen as solely happening in combination with the acquisition of grammar. Especially, for beginners of foreign languages the acquisition of words in isolation is according to Thornbury (2012) important and even necessary as the embedding into realistic contexts is difficult because of very limited options when starting with a new language. The author further states that even in the communicative approach the “plentiful exposure” to “a quantity of vocabulary items” is considered as a means to speed language acquisition up, which are core tenors of the lexical approach. As the software in its first functional prototype focuses on isolated items of language – words – it is mainly based on theoretical principles underlying the lexical approach, which will be further explained in the next section.

However, before moving on to the next section I want to refer back to the introduction where I mentioned the dilemma of testing vocabulary in schools. As teachers have to ensure that their students acquire vocabulary the majority of teachers tend to so called graded “Lernzielkontrollen” or “Schriftliche/Mündliche Mitarbeitüberprüfungen”. The reason for these tests is that teachers have to ensure that their learners do their job and learn vocabulary. In an interview I conducted a while ago a teacher made me aware of the fact that this method is actually illegal and referred to a letter from the Landesschulrat für Steiermark (2013) which clearly states that such methods are only means to gather information about a learner’s current

knowledge level, but must not be graded. As this letter only refers to written tests I asked the teacher about oral tests and she told me that there is actually no difference between written and oral tests. From her own experience she told me that a big problem with this policy is that unmotivated students often stop learning vocabulary as soon as they recognize that their performance on such “Informationsfeststellungen” has no influence on their grade.

No matter whether this method is applied illegally (by grading) or legally (for purposes of information gathering) another aspect, namely time, should be considered. By applying this method regularly a lot of valuable class time is spent on ensuring that learners do their task of learning vocabulary. The aspects mentioned in the preceding paragraphs should be kept in mind as the concept for the software was partly influenced by them.

3.2 Lexical approach

The term lexical approach was coined and defined by Michael Lewis in 1993. The basic idea behind this approach to language acquisition is that the most important aspect of language learning is the learners’ development of the skill to acquire and apply words and longer lexical units. This is in stark contrast to the communicative approach which points out the acquisition of grammatical notions as most important. However, supporters of the lexical approach also argue that through their approach learners’ also develop the skill to recognize, understand and apply grammatical patterns and structures. (cf. Moudraia 2001, Thornbury 2012) Coady and Huckin (1997) argue that the learning and acquisition of idiomatic lexical units is a highly important aspect of learning vocabulary in a foreign language, as the language use of native speakers to great parts also consists of idiomatic phrases and sentences.

The following four aspects are the basis of the lexical approach:

- “Lexis (the vocabulary of a language, as distinct from its grammar)¹ is the basis of language.

¹ <http://dictionary.reference.com/browse/lexis> [accessed on 29.04.2015]

- Lexis is misunderstood in language teaching because of the assumption that grammar is the basis of language and that mastery of the grammatical system is a prerequisite for effective communication.
- The key principle of a lexical approach is that ‘language consists of grammaticalized lexis, not lexicalized grammar.’
- One of the central organizing principles of any meaning centered syllabus should be lexis.” (Moudraia 2001)

Coady and Huckin (1997) explain the significance of theoretical findings on the lexical approach for teaching methods and point out the following aspects:

- “Early emphasis on receptive skills.
- De-contextualized vocabulary learning is a fully legitimate strategy.” (Coady and Huckin 1997)

The six points from the quotations above demonstrate that the basic principles underlying the lexical approach are not too different from what the curriculum dictates, as both highlight the importance of acquiring words, phrases, collocations and lexical units. The only aspect which might be considered different is the significance of context. While the curriculum states that diverse contexts are important and necessary, the lexical approach is not as strict and sees the learning and eventual acquisition of words and lexical units in isolation still as valuable activities, as these activities are said to speed things up. However, many aspects stated in the curriculum and the lexical approach match which makes the lexical approach a valuable theory, and therefore, it will be further investigated in this chapter and used as a basis for the development of the software. This chapter will also explain theoretical findings of the lexical approach regarding the mental lexicon and its implications for classifying vocabulary to allow effective learning. Another aspect which will be discussed in this chapter is vocabulary testing. Before moving on to the next section I shall be forced to say that the majority of well-known linguists consider the learning of words, phrases and lexical units, similar to the learning of grammar rules, as too artificial activities for the classroom which do not lead to the expected outcome (cf. Yule 2006). This, however, is irrelevant for the development of a vocabulary trainer which aims at learning analytics, as the intended aim of the software is not to replace teaching, but to support teaching.

Before this thesis explains findings of the lexical approach in greater detail the more general

linguistic question of what knowledge about language related concepts (words, grammar, etc.) means should be clarified. The next section will therefore briefly explain two basic linguistic concepts found in most approaches to language acquisition and teaching, namely competence and performance.

3.3 Competence and performance

Competence and performance are very basic linguistic concepts, which describe the difference between recognizing and knowing language related concepts. Regarding vocabulary and the software developed in accordance with this thesis, these two concepts are very important, as they define when learners have really learned a word, phrase or lexical unit. For the software, these concepts are insofar important as they define when a word can be marked as learned.

Competence: This term describes the knowledge about a language and can be further divided into sub-categories such as grammatical competence, cognitive competence, lexical competence, etc. Lexical competence describes the knowledge about lexical items of a language: words, phrases, lexical units and their linked semantic and syntactic concepts. (cf. Newby 2011)

Performance: This term describes the ability to actively apply the knowledge about a language. Performance and competence are closely related and it is said that performance usually is less distinct than competence. This means that learners might already know semantic meanings and syntactic forms of words when they hear or read them (competence), but might not be able to use these words in their natural language use (performance). Thus, performance defines whether a learner has learned a lexical unit or not. (cf. Newby 2011, Yule 2006)

For the design and development of a vocabulary trainer the linguistic concepts of competence and performance imply that it is actually not enough to test word recognition as this only allows conclusions regarding competence, but do not indicate whether a student has really learned a word. In order to test performance it is necessary that learners are confronted with tasks which demand their own application of lexical units, or in other words they have to

undergo the process of speaking or writing. However, keeping in mind that the vocabulary trainer aims at complete beginners of English it can be assumed that learners have no or only very little lexical knowledge about the language. For this reason, competence and performance can be seen as almost similar at this learning level, which makes testing of vocabulary recognition absolutely meaningful and legitimate.

3.4 Mental lexicon

The linguistic term mental lexicon refers to the organisation of words, phrases and lexical units in the human brain. Thornbury (2012) describes the mental lexicon as a net or web of word meanings or semantics which are connected to each other in various ways. In addition, these mental concepts are connected to symbols or word forms which again have connections to other syntactic units. Aitchison (2012) states “[...] the human word-web – the way in which humans link words together in their minds. We noted that words seem to be organized in semantic fields, and that, within these fields, various types of relations exist.” The author describes numerous types of relations between these fields and single items within these fields, as for example coordinates, synonyms and opposites, etc. In summary it can be said that the mental lexicon is believed to be a highly complex web of semantic fields containing lexical units which are also related to other lexical units. Significant is the assumption that the mental lexicon, against former theories, does not resemble the structure of a dictionary. Empirical surveys on the mental lexicon revealed that orthographically similar lexical units with very different semantics seem to be a lot harder to acquire for most learners, than semantically similar words. Thornbury (2012) suggest learning words not in alphabetical order, but according to their belonging to a specific semantic field. He also compares semantic fields to topics and states that both are basically inseparable as topics mostly consist of several semantic fields, which are assumed to be closely related in the mental lexicon.

Figure 4 shows a simple representation of how the mental lexicon is believed to look like. It shows weekdays, months, feast days and adverbs of time, which could be labels for the semantic fields to which the words in the image belong to. Thus, this image could be reduced to four connected bubbles with respective labels. It is important to notice that this is basically the simplest possible representation of the mental lexicon as it does not include syntactic connections. Furthermore, it must be said that the theory of the mental lexicon is solely based

on empirical evidence, and thus, the validity of the theory is debatable. However, as long as scientists do not falsify the theory by neurological evidence it should be regarded as highly possible because of empirical evidence. Software based on this theory could provide even more convincing evidence.

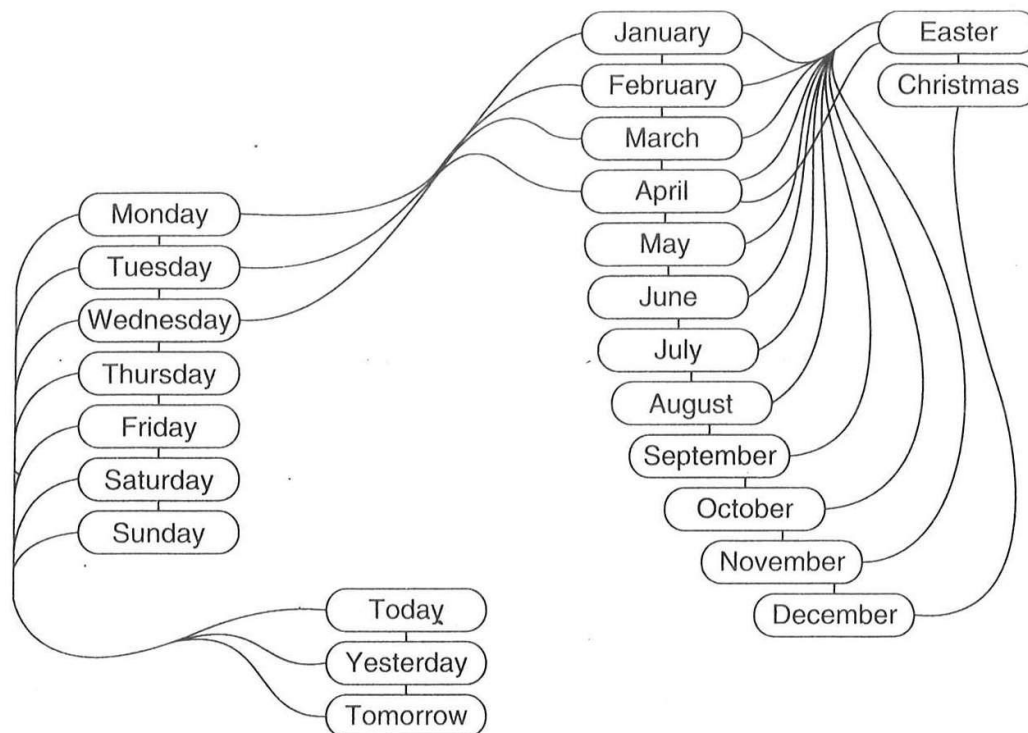


Figure 4: A semantic word-web – clusters of coordinates (Aitchison 2012).

3.5 Vocabulary testing

Thornbury (2012) points out that testing vocabulary is vitally important in the process of learning a language. He argues that testing vocabulary is not a simple one dimensional task, as there are many aspects to consider. He provides a list of important aspects to consider when testing vocabulary and suggest testing the following:

1. “the word’s form - both spoken and written
2. the word’s meaning (or meanings)
3. any connotations the word might have
4. whether the word is specific to a certain register or style

5. the word's grammatical characteristics - e.g. part of speech
6. the word's common collocations
7. the word's derivations
8. the word's relative frequency" (Thornbury 2012)

With regard to the Austrian curriculum and the lexical approach, especially aspects 1, 2, 5, 6 and 7 are significant. A closer look at these aspects reveals that it is impossible to cover all of them in a single exercise, and Thornbury (2012) suggest to pick one, two or at a maximum three of these aspects to create meaningful exercises and tests. Multiple choice matching tasks for instance usually aim at testing the recognition of word form and meaning, but not at grammatical characteristics.

Another crucial aspect regarding vocabulary testing and the above mentioned criteria is the language level of students. Not all of these eight aspects are meaningful for beginners of a language. Aspects 4 and 6, word connotations and common collocations, for example are not applicable with beginners as their language level is simply too low and they are not yet expected to know many connotations and collocations. Vocabulary testing with beginners of a language should from my point of view focus on word forms (syntax), word meanings (semantics) and partly on word derivations.

4 Language learning software

This chapter provides a short review of free-to-use language (English) learning software available on the internet. A simple Google search for “vocabulary trainer” or “vocabulary learning software” provides a great number of results of commercial and free-to-use web-based vocabulary trainer applications and downloadable computer programs. These applications follow various approaches to language teaching and offer a number of different tasks which might help to improve learners’ language skills and expand their mental lexica.

The majority of free online vocabulary trainers seem very insufficient regarding learning analytics as they only provide a record of correct and wrong answers immediately after a test is completed, but do not offer long-term analysis. The number of applications which provide analysis tools is very limited and also their analysis options are mostly very limited. For this review the four vocabulary trainers which seem to offer the most extensive analysis tools were chosen. Three of these programs are web-based and one is only available as client software. Two of the web-based trainers offer free contents and additional charged contents. With these trainers only the free contents are reviewed.

4.1 Babbel²

Babbel is one of the best-known and, considering the fact that they even can afford advertisement on TV, one of the most successful language learning applications on the internet. In terms of learning opportunities and variety of offered tasks Babbel is probably the most extensive application available. Learners are exposed to multiple tasks, which the program suggests to its users automatically. When trying out the application it becomes clear, however, that Babbel does not actually aim at learning analytics, or in other words its aim is not to inform teachers or learners about their language competence/performance level, but to provide a full language course for users, which supports them in the actual acquisition process.

² <https://www.babbel.com/> [accessed on 29.04.2015]

Also the format of Babbel's tasks clearly shows the aim of the software. Students first get explanations of words and then have to show their ability to apply these words in various tasks. Figure 5 illustrates Babbel's analysis mode. In the free version of Babbel the statistics provided for learners are very limited and only consist of a list of "learned" words with some additional information as for example the number of revisions. Although, Babbel states that they offer free course materials, these materials are very limited and can be completed within ten to fifteen minutes. Therefore, their free course materials rather seem to be a means to lure learners into paying for more extensive courses. Altogether, Babbel has a very different aim than the software developed in accordance with this thesis. In addition, the small amounts of free contents makes the program irrelevant for a school context as the charged contents are unaffordable for Austrian schools.

Dein Wortschatz (6) [Wie funktioniert der Wiederhol-Manager?](#)

Heute wiederholen:
0 Vokabeln

Jetzt wiederholen

Verteilung der Begriffe über die Lernstufen:

1 2 3 4 5 6

Wie lernst du eigentlich am besten? Diskutiere auf dem [Babbel-Forum](#) mit Nutzern und Machern von Babbel deine Erfahrungen.

Filtern: [+ Eigene Liste anlegen](#) [Mehr](#)

	Deutsch	Englisch		Anzahl Wiederholungen	Zuletzt wiederholt [▲]	Lernstufe
	hallo	hello		1	29.04.2015	x
	auf Wiedersehen	goodbye		1	29.04.2015	x
	danke	thank you		1	29.04.2015	x
	guten Morgen	good morning		1	29.04.2015	x
	guten Abend	good evening		1	29.04.2015	x
	gute Nacht	goodnight		1	29.04.2015	x

Vokabeln pro Seite

Figure 5: Analysis mode of Babbel

4.2 Languagecourse.net – vocabulary trainer³

In contrast to Babbel the vocabulary trainer on languagecourse.net only offers free contents, which is a great advantage for schools. Also the structure of the tool is not like an entire language course, but rather seems to be intended as a tool to be used in addition to a language course or language education in school.

The vocabulary trainer is easy to understand and does not vary the task format. Thus, the procedure is always the same: Learners choose their native language, the language they intend to learn and choose a course. The different courses offered are not literal courses, but various different vocabulary lists. When learners start the trainer the program chooses a number of lexical items and shows these items to the learners. The speed by which the lexical items are shown can be adjusted by the learners. After these words and phrases were shown to the students the same lexical units are tested in a multiple choice format. When a learner has answered all questions a record of the test result is provided as can be seen in figure 6.

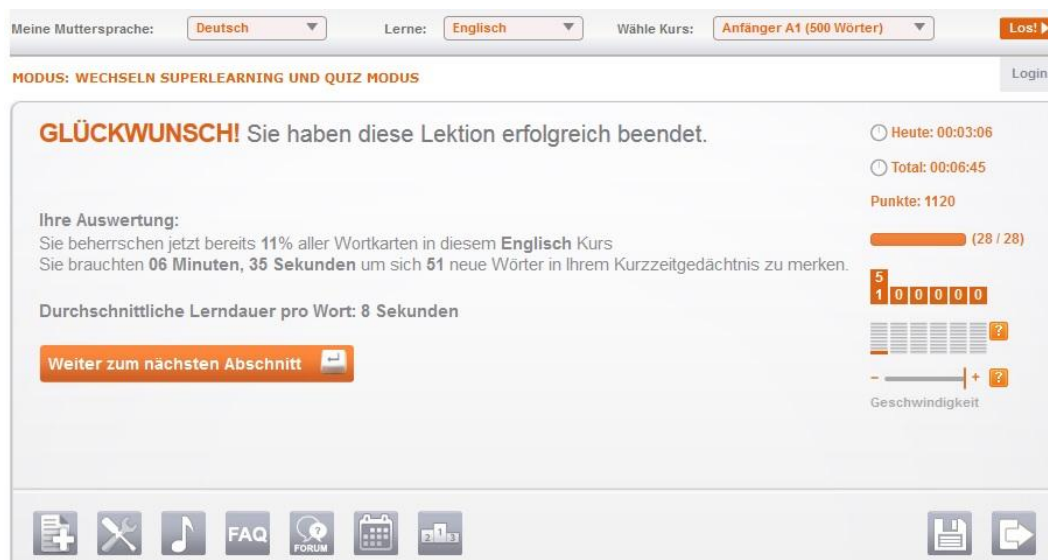


Figure 6: Analysis mode of languagecourse.net

This test record is the only means of analysis offered by the vocabulary trainer and is from my point of view not only very limited, but also very incomprehensible as it does for example not

³ <http://www.languagecourse.net/vocabulary-trainer.php> [accessed on 29.04.2015]

explain the meaning of the gained points. Furthermore, the aim of this vocabulary trainer again is different to the aim of the software developed for this thesis. The trainer of languagecourse.net is based on the theory of spaced repetition and aims at long-term memorization of lexical units by repeating words in progressively increasing time intervals. This trainer might be very useful for language learners and could help them to memorize words more effectively. However, in terms of learning analytics the software is also insufficient as it does not inform stakeholders about learners' current language levels and possible problem areas.

4.3 LEO dictionary – language trainer⁴

LEO is already famous for their extensive dictionary and over the last few years they also developed language trainers for various languages. Regarding their English vocabulary trainer it is to say that it is very similar to Babbel. Both Babbel and LEO provide free as well as charged contents. While Babbel's free contents are very limited, LEO offers a greater amount of free contents. However, their trainer is very similar to Babbel with regard to task format and purpose. Thus, its purpose is to provide an entire language course for students, which becomes clear when considering the format of their tasks as many of them do not test vocabulary but try to explain lexical items. Furthermore, LEO's analysis mode is nearly identical to Babbel's analysis mode, which only provides very limited data. For this reason, their trainer does not provide enough meaningful data to draw conclusions regarding a student's learning process.

4.4 Lingoversity⁵

Lingoversity is a client program, which is not web-based and has no interface to the internet. In terms of language learning the software offers a variety of tasks and compared to the other web-based applications a rather extensive analysis mode. Figure 7 shows a screenshot of

⁴ <https://dict.leo.org/trainer/index.php> [accessed on 29.04.2015]

⁵ <http://www.lingoversity.com/> [accessed on 29.04.2015]

Lingoversity's analysis mode, which provides information about learning frequency in the form of a calendar, etc. On the first glimpse their realisation of analysis tools seems to be very useful. However, when taking a closer look, their representations of learner produced information lacks clarity. The diagrams do not include headings or keys, which makes it very difficult or even impossible to interpret them as it is not clear which data they actually represent. Also the overview in the middle is rather incomprehensible and I cannot tell what it actually represents. However, compared to the analysis tools offered by the other reviewed applications, I would say that Lingoversity is closest towards learning analytics, but as a client software with no interface to the Internet its great disadvantage is that data can only be collected about single users, and cannot be compared to other users' data nor can it be accessed by teachers.

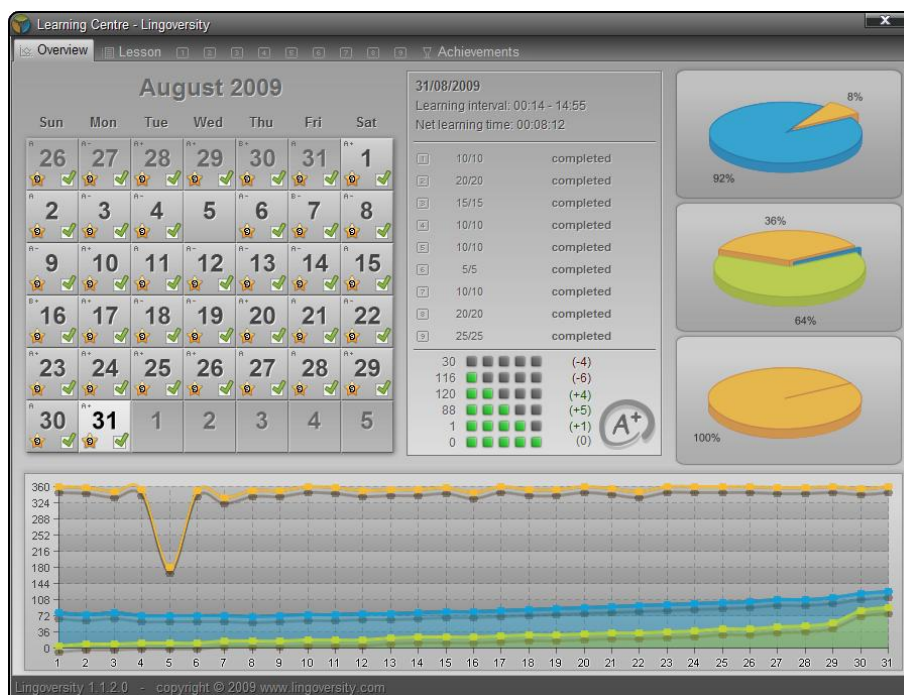


Figure 7: Analysis mode of Lingoversity

4.5 Summary

In conclusion, the internet offers a broad variety of different vocabulary trainers with specific aims and purposes. Regarding learning analytics none of the reviewed vocabulary trainers seems to be sufficient. The main reason for this can be found in the different aims of most of

the trainers as they do not seem to focus on learning analytics but rather at offering entire language courses. Especially, Babbel's and LEO's tools focus on providing whole language courses, which do not aim at informing stakeholders about learners' language competence and performance, but aim at expanding a learners knowledge of a language. Languagecourse.net's vocabulary trainer is in sharp contrast to the other applications as its mere purpose is to support learners in memorizing lexical units. As mentioned above Lingoversity's tool seems to come closest to learning analytics in the way they represent learner produced data, but somehow fail to represent this data in a comprehensible way.

A great disadvantage of all of the reviewed applications is that learner produced data is only available to learners themselves, but does not inform teachers. As learning analytics is a teacher centered approach tools that inform teachers about learners' language skills or at least allow access to learner produced data are at the core of the approach and of significant importance.

The difference between the reviewed vocabulary trainer applications and the software developed in accordance with this thesis will become clear in chapter 6 where the concept of the developed software is introduced.

5 Core vocabulary

The term core vocabulary⁶ is often used to refer to a list or corpus of vocabulary containing the most important words, phrases or lexical units of a language. Research on core vocabulary reveals that linguists and other scientists who are concerned with defining a corpus of core or basic vocabulary in English do not entirely agree on which lexical items should be included in such a corpus. The reason for a debate about which lexical items to define as core vocabulary is not only because of different perspectives on language, but mainly because language is not static, but constantly changing. Thus, new words are invented or adopted from other languages constantly and enter the corpus of commonly used words, while older words become out of fashion and disappear from the corpus.

A majority of researchers refers to the Oxford English Corpus (OEC)⁷ which is considered as one of the largest available corpora of English. Their constant long-term research on language and collection of language in use reveals interesting facts about existing language in general and more specific language use. By analysing real language in use the OEC is able to provide lists which rank words according to their frequency of appearance, which indicates the importance of a word. Figure 8 shows how many different lemmas a speaker has to know in order to understand a certain percentage of written and spoken English. A lemma refers to the base form of a word.

Vocabulary size (no. lemmas)	% of content in OEC	Example lemmas
10	25%	the, of, and, to, that, have
100	50%	from, because, go, me, our, well, way
1000	75%	girl, win, decide, huge, difficult, series
7000	90%	tackle, peak, crude, purely, dude, modest
50,000	95%	saboteur, autocracy, calyx, conformist
>1,000,000	99%	laggardly, endobenthic, pomological

Figure 8: The OEC – word usage (Oxford Dictionaries 2015)

⁶ http://en.wikipedia.org/wiki/Basic_English [accessed on 30.04.2015]

⁷ <http://www.oxforddictionaries.com/words/the-oxford-english-corpus> [accessed on 30.04.2015]

According to Oxford Dictionaries (2015), a speaker only needs to know ten different lemmas to understand 25% of used language. A speaker who has acquired a thousand different lemmas is able to understand 75% of language produced by other speakers. The average English native speaker is said to know about seven thousand to ten thousand lemmas which accounts for about 90% of produced language.

Thornbury in his book explores vocabulary lists, coursebooks, vocabulary books, dictionaries and corpus data. Regarding coursebooks he states that they “select vocabulary on the grounds of: usefulness, frequency, learnability and teachability”. Compared to the Oxford English Corpus, usefulness and frequency seem to play a crucial role in both, coursebooks and the OEC. Learnability and teachability are no obvious aspects in the OEC, but one might argue that learnability is important as a hidden parameter in the OEC as easier words might be considered as more likely to be used by a greater amount of speakers. Coursebooks vary widely in the number of words they introduce and according to Thornbury they range from a thousand words up to four thousand words. (cf. Thornbury 2012)

Usefulness, frequency and learnability are all factors which influence both, vocabulary in coursebooks and the OEC. The only parameter which only seems to influence the vocabulary choice in coursebooks is teachability. A quick manual comparison of vocabulary from three different first grade coursebooks used in Austrian schools showed that all three books contain about a thousand to a thousand five hundred words. A further comparison of the vocabulary from the books to the Oxford 3000^{TM8}, which is a vocabulary list based on the OEC containing the three thousand most important English words, revealed that all words used in the coursebooks also appear in the Oxford 3000TM.

Altogether, I came to the decision to base the vocabulary database used by the software on Austrian schoolbooks, as the software’s intended area of application is first and foremost in a school context in Austria and other German speaking countries. Thus, the vocabulary database for the software is a compilation of words from Gerngross et al. (2009), Gaynor and Hübner (2007), Harmer (2007) and Gerngross et al. (2005), and contains about a thousand one hundred words.

⁸ <http://www.oxfordlearnersdictionaries.com/wordlist/english/oxford3000/> [accessed on 30.04.2015]

6 Software concept

The concept for the vocabulary trainer developed over several iterations of meetings with my supervisor (and at times other researchers from the field of learning analytics) and refinement of the concept. The main reason for this process on the one hand was lack of theory regarding vocabulary acquisition, and on the other hand was a rather open-ended definition of the task. The task was defined as the title of this thesis states and no other requirements were given to narrow the scope. However, over the course of several meetings the concept was constantly refined and requirements as well as some assumptions emerged. In a software development process, requirements are usually defined by customers before the conception and if necessary they are refined after specific stages of the development process. As this was not the case with this software the development of the concept took about two and a half months, which was longer than expected.

In the following sections the concept will be explained in detail. Thus, the basic structure of the software is pointed out and the main functions of the different components are explained. This chapter does not provide information about how various components and functions were realized, but describes the basic structure in terms of which functionality various components and functions should provide. Although, the requirements were not defined at the beginning of the conception process the next section will first of all provide an overview of the requirements and assumptions as this allows a more logical description of the concept. Then the concept will be explained textual and finally a use case diagram summarizes this chapter.

6.1 Requirements and assumptions

This section provides a list of requirements and assumptions, which derive from the requirements. The list is divided into several sections starting with general requirements, over requirements regarding testing, analysis and the management of vocabulary.

General:

- The final software should be a functional and stable web application.
- The layout of the software should be kept simple and not too infantile.
- The target group of users are pupils of the first grade in secondary school and their teachers, or complete beginners of English.
- The software is used by three groups of users: students, teachers and administrators.
- The purpose of the software is not to provide a language course, but to assess students' vocabulary and provide analysis tools which allow meaningful conclusions about possible means of intervention.
- Students are not meant to learn new vocabulary when using the program. Thus, it can be assumed that students have already learnt the tested vocabulary.

Testing:

- The first functional prototype only tests a student's knowledge regarding recognition of word forms and meanings.
- Students do not enter words or produce language, but only choose correct answers. Thus, the program does not assess performance but competence.
- Students should be able to run the test mode "endlessly", meaning that there is no fixed number of tested items, but students can stop whenever they want.

Analysis:

- Students can only access analysis of their own produced data for reasons of privacy.
- Teachers can only access analysis of data produced by their own students, but not of other teachers' students.
- Administrators can access analysis of all student produced data.
- Analysis tools must not include diagrams or other complex visual representations, but must be as simple as possible.

Management of vocabulary:

- Students have no other access to the vocabulary used for testing than in actual tests.
- Teachers can browse and search through the vocabulary database. They cannot add, delete or modify topics or word entries, but can enter suggestions for topics and words to be added to the database in a form which forwards their suggestions to administrators via e-mail.
- Administrators can add and delete topics, add and delete words, add and delete words to/from topics, edit word entries, and browse through the vocabulary database.

6.2 Textual description of the concept

The basic concept for the vocabulary trainer prototype was mainly shaped by the requirements listed in the section above. Significant is the requirement that the software will be used by three different groups of users, which also shaped the other requirements. The more specific requirements basically suggest four different modes: a mode for administrators to manage the vocabulary database, a mode for teachers to browse the vocabulary database, a mode for students to test vocabulary and a mode to analyze learner produced data with different permissions according to a users' role. The first two modes are very similar, and therefore, were merged into one mode with different permissions for teachers and administrators. The final concept consists of three different modes: the manage/browse vocabulary mode, the test mode and the analyze mode. These three modes were implemented in the same order as stated before and will be explained in further detail in the following sections.

Another aspect which heavily influenced the conception and implementation of the software was the theory on language acquisition mentioned in chapter 3. Especially, the theory on the lexical approach and the mental lexicon were used as a basis for the concept. Thus, the software aims at analyzing students' vocabulary in terms of the mental lexicon. Although, this theory lacks neurological evidence, empirical evidence in a way suggests that vocabulary is stored in the form of semantically linked and ordered word-webs with further connections according to word syntax. This leads to the assumptions that problems with vocabulary manifest within such semantic fields, and problem areas might be found among closely related semantic fields. For this reason, the vocabulary database was designed in terms of

topics, which in its definitions can also be seen as semantic fields.

With regard to English words, phrases and other lexical units, such a design constitutes several problems. First, English words can be divided into two larger groups, namely content words and function words. Content words are unproblematic and can be assigned to semantic fields, function words, however, are of grammatical nature and do not convey meaning themselves but serve to shape the overall meaning of sentences. (cf. Coady and Huckin 1997) Therefore, function words are not assigned to semantic fields, nor they are simply left out, but they constitute a separate category which is utterly important as among the top one hundred words in the Oxford 3000TM one mostly finds function words. Furthermore, phrases and longer lexical units must be treated differently than single words and were not included in this prototype.

6.2.1 Manage/Browse vocabulary mode

This mode is used by teachers and administrators with respective permissions. Teachers can only browse through the vocabulary database and suggest new words and topics to add to the database. This is necessary as teachers often introduce vocabulary to their students which is not found in their coursebooks. However, they are not allowed to add, delete or edit vocabulary and topics on their own, as this could lead to a very unstructured and useless collection of vocabulary. Another option for teachers, which was included in a former version of the concept, was that teachers should be able to create individual vocabulary pools from the existing vocabulary database. This function, however, was rejected for reasons of complexity and additional expenditure of time for teachers. As I see such functionality as necessary for the use of the software in a school context this aspect will be further addressed in the discussion in chapter 9.

Administrators have full access to the vocabulary database which allows them to: add or delete topics, add new words to the database or delete existing words from the database, add words to specific topics or delete words from topics, edit word entries or simply browse through the vocabulary database. In order to allow easy and fast management of the vocabulary database this mode must be intuitive and representations of the existing contents of the database must be simple and clear.

6.2.2 Test mode

In a first concept of the test mode it was intended to incorporate the theory on language acquisition in the test mode to assess whether it is easier for students to learn vocabulary structured in semantic fields or otherwise (alphabetically order lists, etc.). However, with the introduction of the requirement that the program should not help students in their learning of new words, but only test words and implicitly help students through analysis, this approach shifted and the theory was used as a basis for the analysis, which also better corresponds to the theory underlying learning analytics.

Furthermore, the requirements regarding testing mentioned above, led to the conception of a very simple test mode. The testing of word recognition in terms of form and meaning can be easily achieved through simple multiple choice tests. For the prototype of the vocabulary trainer only one test format was implemented, which is a multiple choice (single choice) task. The application provides one word in either English or German and four words in the other language. Students are asked to click on the correct equivalent in the other language. Furthermore, students are asked to rate the question regarding difficulty on a six-level scale, when a word is tested for the first time. The ratings represent values between zero and one in steps of 0,2 where zero means very easy and one means very difficult. These ratings are stored in the database for single users in the form of tested word entries, and also influence the rating stored for single words, which is the average value of a base rating and all users' ratings of a specific word. The base rating is attached to topics and defined by administrators when they add new topics to the database. A word's base rating is the average value of base ratings of the topics it is assigned to. The ratings of words could reveal useful information about a learners' self-assessment and more generally about the difficulty of specific words for a majority of learners. Furthermore, the ratings attached to single words are used to compile homogenous tests in terms of difficulty, which is explained in more detail in chapter 7.4.3.

6.2.3 Analyze mode

The conception of the analyze mode and its means of analysis was the most difficult part, as theoretical findings on language acquisition provide no or only very little information about how to effectively assess students' vocabulary and come up with possible interventions for improvement. As mentioned above the theory on the mental lexicon was used as a basis for

the design of the database and also serves as basis for the development of the analyze mode. Thus, one tool for analysis should highlight students' possible problems with specific semantic fields. As the representation of analysis must not contain diagrams or other complex visual representations, this was planned in the form of a table using colours to highlight problems. Another feature planned in the conception was a self-assessment scale which should on the one hand inform teachers about their students' self-assessment, but also encourage students to honest ratings of words. Other features planned for this mode were: general statistics regarding errors, progress and overall performance, a list of all errors, and information about past test results. At this stage the software does not provide suggestions for intervention, but leaves it to teachers to interpret the data and draw conclusions.

6.3 Use cases

Figure 9 shows a use case diagram, which can be seen as a translation of the textual requirements into an image. It illustrates the three different user roles and associations to their respective use cases.

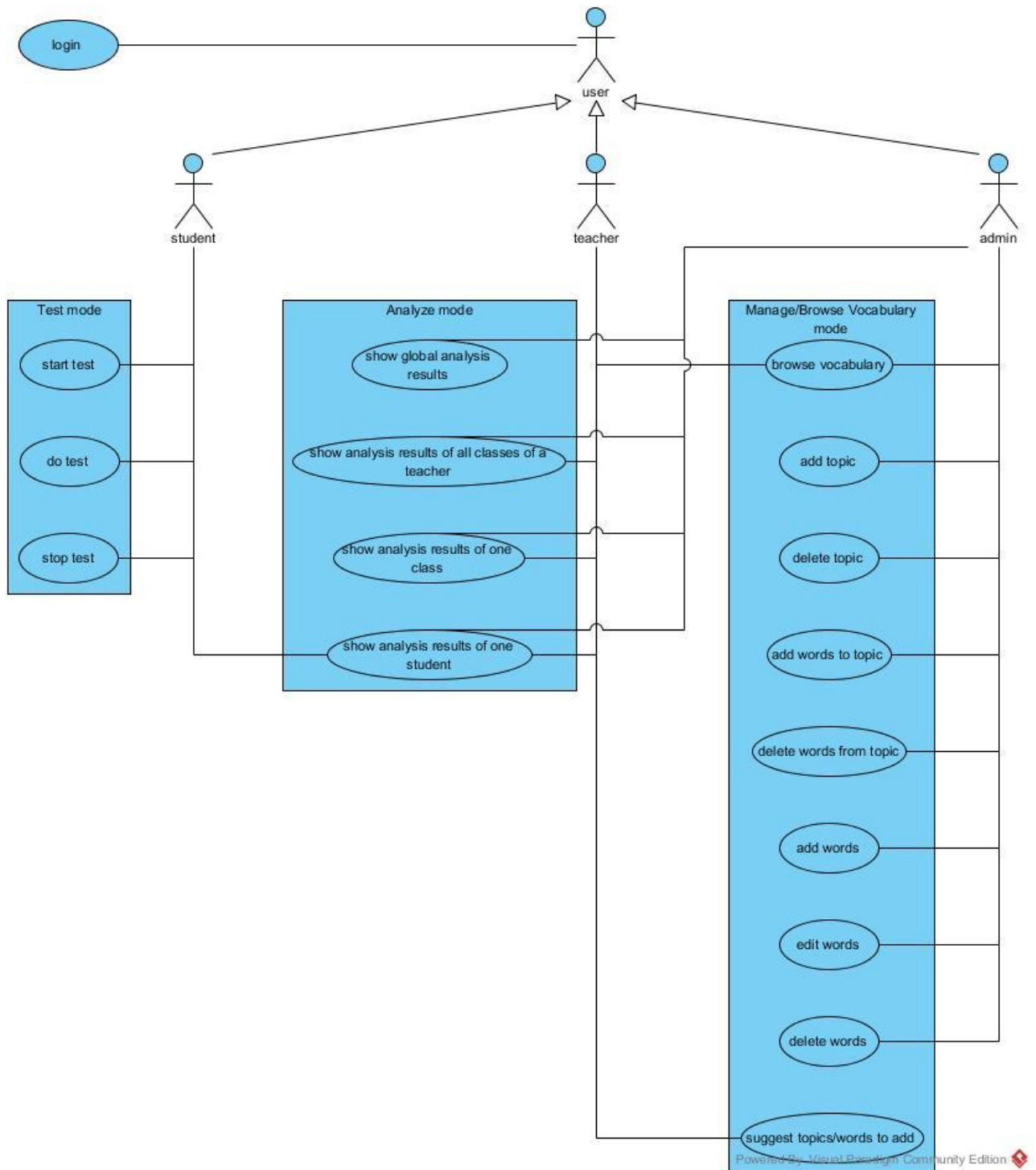


Figure 9: Use case diagram

7 Technical implementation

This chapter is concerned with the technical implementation of the first functional prototype of the software. The software in its current version is named “VOCABTrainer”. The name was chosen, because it is short, significant and informative as it states the program's purpose in a single word. The development of the prototype took about three months and slightly overlapped with the conception of the program as the requirements and assumptions slightly changed during the implementation of the first component. The result of the implementation is a stable web application in version 1.0. This version should not be seen as a fully functional vocabulary trainer to use in school, but rather as a first approach towards a vocabulary trainer with focus on learning analytics. Thus, it is important to realize the complexity of vocabulary learning with its various aspects, and the numerous opportunities this complex topic bears for analysis and the development of a vocabulary trainer.

The prototype was developed as a web application and runs in all well-established web browsers. This ensures easy access to the program as students can even use the software on their mobile devices. The software is hosted by a server of Graz University of Technology, which also hosts all other learning apps, and can be accessed at the URL <http://schule.learninglab.tugraz.at/vocabtr/>. Similar to all other learning apps provided by the TU Graz, the VOCABTrainer uses the central usermanagement system, which demands new users to sign up before accessing the software. This is a necessary step to ensure an exact allocation of students, teachers, classes and schools. Furthermore, the software was developed under the Apache Licence, Version 2.0⁹ which allows other developers to use and adapt the source code as they wish and paves the way for an unproblematic continuation of the project by other developers.

All learning apps offered by the TU Graz, as well as the VOCABTrainer, are completely free-to-use. Hyperlinks to the various trainers can be accessed at the URL <http://schule.learninglab.tugraz.at/>. This chapter further elaborates on used web technologies, the basic structure of the software, the final implementation and the layout of the software to provide insights into the functionality of the application.

⁹ <http://opensource.org/licenses/Apache-2.0> [accessed on 01.05.2015]

7.1 Used web technologies

To ensure easy accessibility and platform independence the software was developed by technologies holding an open source license¹⁰. All used web technologies are free to access and can be easily obtained. The following section provides a brief overview of the used technologies.

7.1.1 PHP¹¹

For the implementation of the prototype the programming language PHP was used. PHP is a backronym for PHP Hypertext Preprocessor and was mainly developed for purposes of dynamic web development. For this prototype PHP 5 in the version 5.4 was used. The introduction of PHP 5 in 2005 had brought the greatest changes since the introduction of PHP, as PHP 5 introduced a revised and improved object model. The introduction of the Zend Engine 2, which is at the core of the programming language, enables programmers to apply a more object oriented programming approach similar to C++ or Java, which was not possible with former PHP versions. (cf. Reimers and Thies 2012)

There are various reasons for choosing PHP as programming language for the development of the vocabulary trainer. First, PHP is provided under an open source licence, namely the PHP Licence v3.01, and therefore, is free-to-use (cf. PHP Group 2015a). Second, PHP runs on almost all platforms and web servers: "PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache¹², IIS, and many others." (PHP Group 2015b) Third, PHP supports a wide range of databases. Fourth, PHP is very simple for beginners, but at the same time offers a great number of possibilities for advanced programmers. (cf. PHP Group 2015b)

¹⁰ <http://opensource.org/licenses> [accessed on 01.05.2015]

¹¹ <http://php.net/> [accessed on 01.05.2015]

¹² <http://httpd.apache.org/> [accessed on 01.05.2015]

A huge advantage of PHP is that it contains HTML with embedded code. This is in sharp contrast to programming languages such as C or Perl where a lot of commands are needed to produce simple HTML output. (cf. PHP Group 2015b) In contrast to Javascript for example which is interpreted on the clients' computer, PHP is a server side programming language. This means that clients send requests to a web server. The web server hands the relevant PHP files over to the PHP-parser, which interprets the files, requests possible data from some kind of database and generates HTML output to show to the client. (cf. Reimers and Thies 2012)

7.1.2 MySQL¹³

MySQL is a widespread relational databasemanagementsystem (DBMS) provided by the Oracle Corporation. The Oracle Corporation states: "Many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, Alcatel Lucent and Zappos rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software." (Oracle Corporation 2015b) MySQL is provided under an open source licence (cf. Oracle Corporation 2015a) and can be downloaded directly from the MySQL website. The decision to use MySQL as a DBMS for the development of the prototype was not only made because of its popularity and advantages, but also because the server provided by TU Graz had already been set up for the use of MySQL. The software developed for this thesis uses MySQL as back-end to store and retrieve learner produced data, and to store the vocabulary used by the program. In combination with PHP MySQL becomes a very powerful tool as PHP provides plenty of useful functions to access and manipulate data stored in MySQL databases.

Another tool, namely phpMyAdmin¹⁴, is closely related to MySQL and many development environments as for example XAMPP¹⁵ ship their products including this tool. PhpMyAdmin allows easy configuration and maintainability of MySQL databases. Thus, it provides an intuitive userinterface to access and manipulate databases, their tables and data.

¹³ <https://www.mysql.com> [accessed on 01.05.2015]

¹⁴ <http://www.phpmyadmin.net> [accessed on 01.05.2015]

¹⁵ <https://www.apachefriends.org> [accessed on 01.05.2015]

7.1.3 jQuery¹⁶

jQuery is a small and very performant open source JavaScript (JS) library (cf. jQuery Foundation 2015), which offers an easy way to animate website elements. jQuery was developed to provide a means for developers to easily and quickly develop interactive websites, which was very complex and time-consuming with JavaScript in the past. The VOCABTrainer uses jQuery in version 1.11.2 to provide a more interactive experience for its users. The library supports developers in finding, manipulating and animating document object model (DOM) elements and contents. Furthermore, it offers means for event handling and asynchronous JavaScript and XML (AJAX) requests. (cf. Neuhold 2013, Reimers and Thies 2012)

7.1.4 CSS and HTML¹⁷

Cascading Style Sheets (CSS) and Hypertext Markup Language (HTML) are two of the core technologies for web development. While CSS is responsible for describing the layout of a website, HTML describes its structure. Thus, CSS is used to define colors, layouts, fonts and other visual parts of a website, whereas HTML is used to define headings, paragraphs, lists, and other structural parts of a website. CSS and HTML are a means to separate content and presentation of a website (cf. Reimers and Thies 2012)

7.2 Basic component overview

This section of the thesis constitutes the transition from the concept to the implementation. Thus, figure 10 illustrates the textual concept and use cases in the form of a basic state diagram. The image provides an overview of possible states according to a users' role in very general terms. It is not meant to give a detailed description of the software and does not include all use cases, but serves the purpose of providing a rough overview of the different

¹⁶ <http://jquery.com/> [accessed on 01.05.2015]

¹⁷ <http://www.w3.org/standards/webdesign/htmlcss> [accessed on 01.05.2015]

modes a user can access. The use cases pointed out in figure 9 are included in the respective states, but not all of the use cases are explicitly stated in the diagram below. The diagram was designed after the conception stage at the very beginning of the design stage and all other diagrams produced in the design stage are in a way based on figure 10.

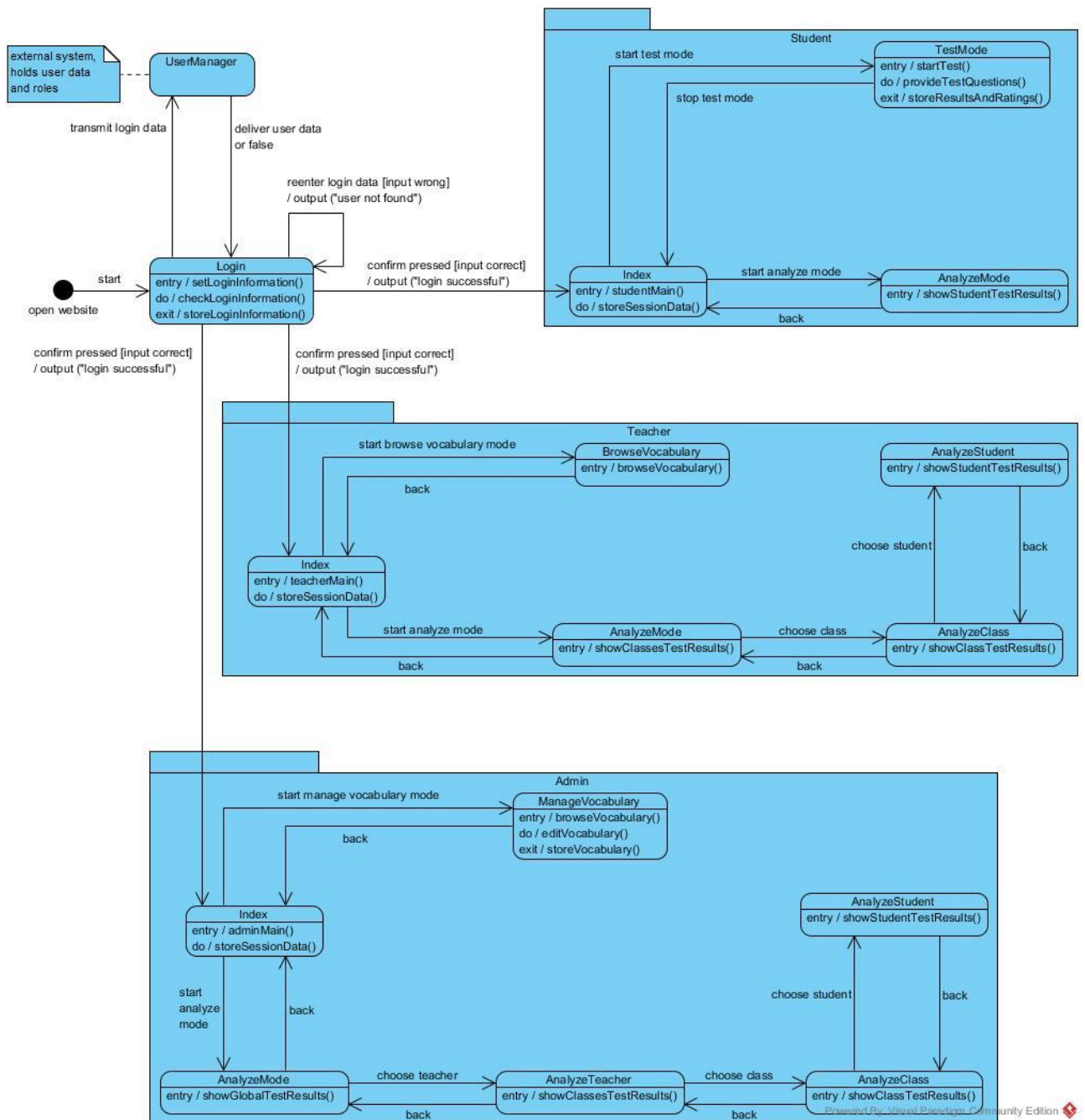


Figure 10: State diagram

7.3 Design

The design stage is crucial in the development of software as this stage defines the further progress of the implementation stage. The design stage often includes decisions regarding the architecture of the software, planning of databases, decisions about used technologies and many more. (cf. Reimers and Thies 2012) Regarding the design stage of the VOCABTrainer it is to say that the conception and design stages overlapped to a certain degree. Thus, decisions regarding used technologies were partly made during the conception stage and slightly changed in the design stage, which is the reason for placing the section on used technologies inbetween the sections on conception and design in this thesis. The design stage for the vocabulary trainer prototype included the decision on a model-view-controller (MVC) architecture, the design of the database and a detailed design of the software under considerations of object oriented principles, modularity, extensibility, stability and performance in the form of an extensive class diagram. The following sections will address these aspects in greater detail with the aid of a number of diagrams.

7.3.1 The MVC architecture

The model-view-controller software architectural pattern is a widespread design pattern for software development and is used by developers of various branches and programming languages. The MVC pattern divides software into three parts which is displayed in figure 11. One part is the model which is responsible for the communication with the database and retrieves or transmits data. Another part is the view which is responsible for data representation and the communication with users of an application. The third part is the controller which sits inbetween the model and the view and contains the businesslogic of a program. The controller is at the core of the MVC architecture and communicates with the model and view. (cf. Reimers and Thies 2012)

Reimers and Thies (2012) state that the MVC pattern is found in various different forms, especially in web development. In many forms views communicate with controllers, these controllers then communicate with models, and finally models communicate with views. Such realization of the MVC pattern, however, means that bussinesslogic is not only contained in controllers but also in models, as their direct communication with views demands preparation

of data to be used by views. In order to restrict the implementation of businesslogic to controllers, which allows a complete separation of presentation, businesslogic and database access, this software adheres to a form of the MVC pattern, where the controller is at the center. Thus, views and models only communicate with controllers and controllers communicate with views and models. As can be seen in figure 11 in this way models do not communicate with views anymore. Such implementation of the MVC pattern allows keeping views and models very simple and understandable as all complex code is contained in controllers.

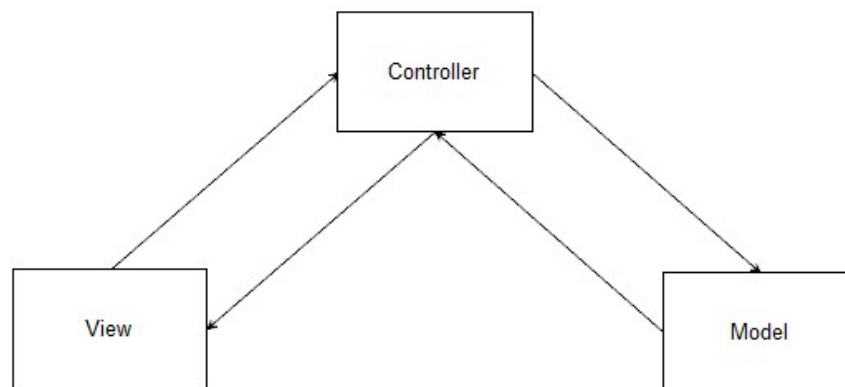


Figure 11: MVC architecture

7.3.2 Database

Database design is probably the most important aspect in the design stage. Changes in the design of the program during the implementation are very common and usually these changes can be done in short time. Changes in the design of the database, especially if it already contains data, are considered as not so easy in most cases, and can become utterly time-consuming. To avoid later changes the basic database design should be definite and precise. This does not mean that each parameter of a table has to be planned from the beginning, but at least the tables and their primary key plus the most important parameters should be fixed. A precise design of the database before implementation also helps a lot with the planning of other aspects of the program as developers make concrete decisions about which data to store and retrieve and also get a better impression of how the program has to deal with the data. (cf. Reimers and Thies 2012) When sticking to an MVC software architectural pattern for example the design of the database also heavily influences the design of model classes.

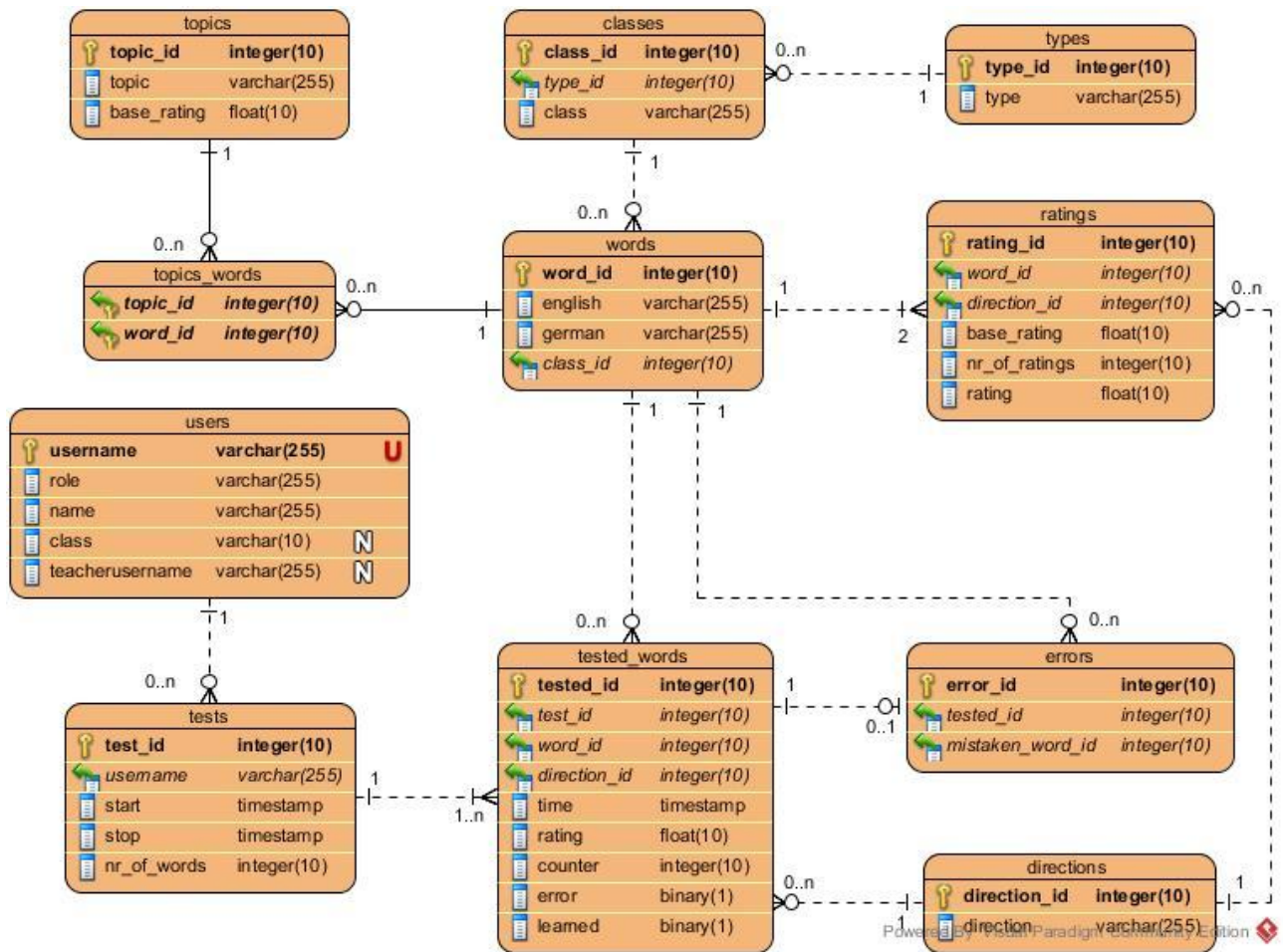


Figure 12: Entity relationship (ER) diagram

Figure 12 shows the entity relationship diagram of the database used by the VOCABTrainer. Each entity in the diagram above represents a table of the actual MySQL database with its attributes. The image was slightly updated after the implementation stage, because the *users* table changed during the linking of the prototype to the external usermanager. All other entities in the image were planned in the way they are represented in the image and did not change over the course of implementation.

In order to reduce data redundancy, and increase data integrity and extensibility the database was mostly normalized to the third normal form. (cf. Riemers and Thies 2012) Only the *users* table was not normalized as it has no influence on other tables. In addition, the software has no influence on the data stored in the *users* table as it is retrieved from the external usermanager during the login and instantly stored in the prototype’s database.

Basically the database consists of three parts: vocabulary, tests and users. The part concerning

the users has only one table which is named similarly. The table's attributes are shown in figure 12 and together constitute a user's profile. Although the same data is also stored in the external usermanager for reasons of performance and structure I considered it better to use an internal table.

The upper part of the ER diagram, more specifically the tables: *topics*, *topics_words*, *words*, *classes*, *types* and *ratings* hold the vocabulary used by the software. The tables *types* and *classes* are "static" in this version of the prototype. The table *types* defines a word as either content word or function word. The type of a word is linked to a word's word class, as for example noun, verb or adjective, which information is stored in the table *classes*. The table *words* stores all available word pairs and is linked to the table *classes* by a word's *class_id*, which attaches more information to a word. The table *topics* stores all topics/semantic fields entered by an administrator, plus a topic's *base_rating*. The *base_rating* defines a topic's difficulty and serves to assess a word's difficulty. Actually, topics would not need to be rated, but I decided to attach a *base_rating* to topics so that administrators do not have to rate every single word they add to the database. A word's *base_rating* is the average value of *base_ratings* of the topics it is assigned to. A word can be assigned to several different topics. This information is stored in the table *topics_words*, where *word_ids* are assigned to *topic_ids*. The table *ratings* stores the average value of students' ratings of words. Each *word_id* has two entries, one for each direction. In the current version of the software a word can be tested either from English to German or the other way around. These directions are stored in a separate "static" table which is in relation to the table *ratings* and *tested_words*.

The lower part of the ER diagram including the tables: *tests*, *tested_words*, *errors* and *directions* hold the learner produced data from the test mode which is used for analysis. The table *tests* stores general information about tests and assigns tests to individual users. Thus, each test entry has an attribute *username* which allocates tests and in further consequence tested words to individual users. Entries in this table are created when a learner starts a test, and they are updated when a learner stops a test. Each tested word is stored in a separate entry in the table *tested_words*. These entries have several attributes, which indicate whether a question was answered correctly, a word has been learnt, how often a word was tested, in which direction a word was tested, the timestamp of the answer and the student's rating of a word. If a word was answered incorrectly the attribute *error* is set to true (1) and an entry, which links the tested word by its id to the wrong answer given by the student is created in the

table errors.

This structure of the database allows flexible access of specific data, reduces the amount of redundant data and most importantly allows easy extension of the database by other developers.

7.3.3 Class diagram

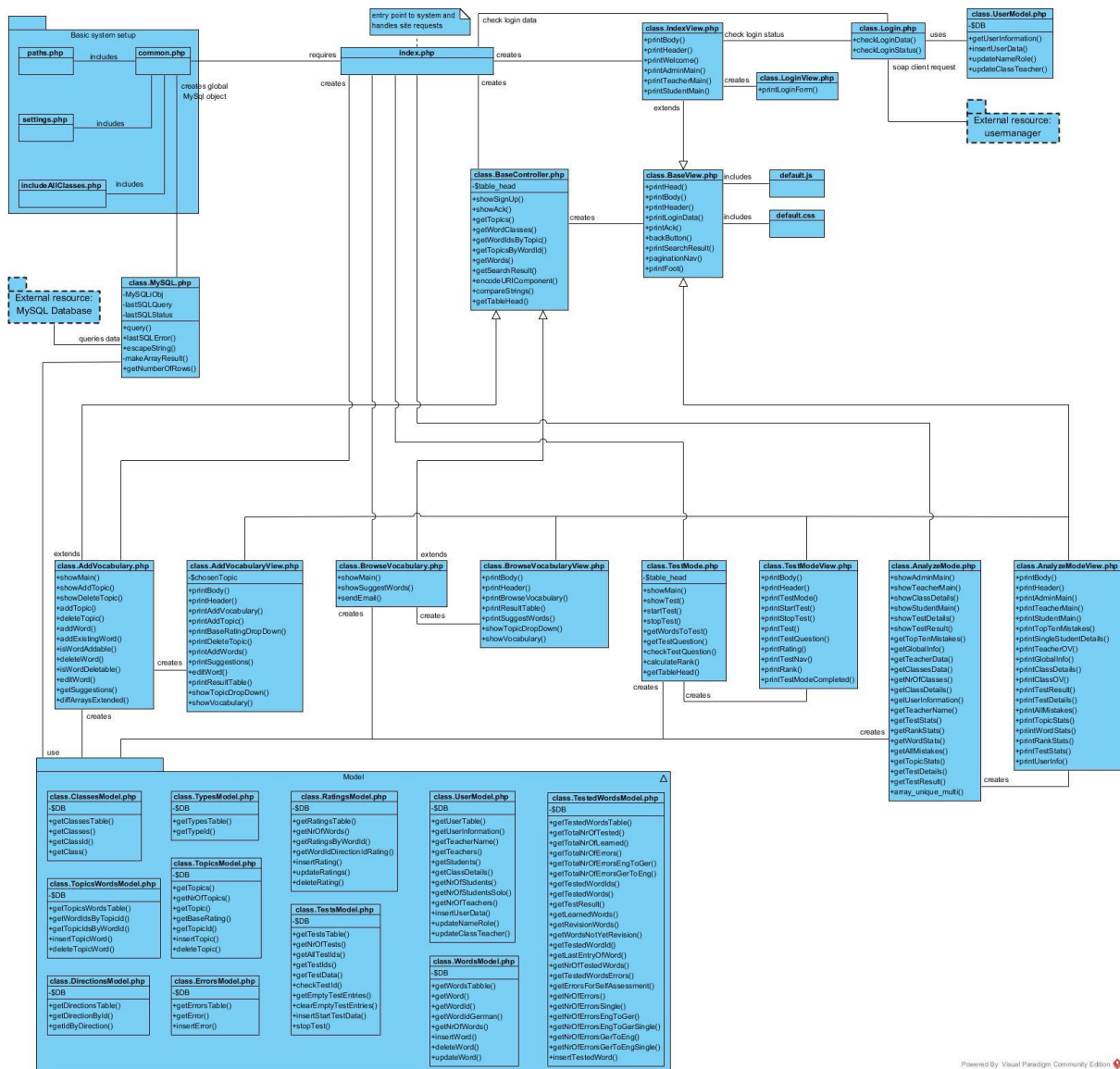


Figure 13: Complete classdiagram

Figure 13 shows a complete classdiagram of the developed prototype. As the diagram is rather

extensive this image is hardly readable and at this place is intended to just give you an overview of the software architecture, as single parts of the diagram are illustrated and explained in greater detail in section 7.4, and therefore, figure 13 should help you to put these single parts together.

However, an experienced eye might notice that some of the classes illustrated in figure 13 are very extensive and provide numerous methods. This in a way indicates a violation of object oriented principles as the software does not strictly conform to the single responsibility principle¹⁸, which states that each class should be responsible for a single part of functionality provided by a program. If and to what extent the design of the software violates this principle is debateable and is discussed in the following paragraph.

Considering the extent and complexity of most PHP frameworks learning them can become utterly time-consuming. In addition, the subject of vocabulary acquisition is highly complex and the process of planning and implementing software, which examines this learning process, also takes quite some time. Therefore, I decided to not learn a PHP framework, but to focus on improving my PHP skills and implement a simple basic system on my own. This decision was consolidated during the research on Zend Framework 2¹⁹ because Eggert (2013) recommends the framework only for very experienced PHP programmers. In retrospect I believe this decision was right, as I would probably not have mastered to implement this prototype in the same amount of time when learning a framework. For this reason, the basic system was implemented following the example given by Reimers and Thies (2012) which will be explained in further detail in section 7.4.1. Although, I did not use a framework in this version, I still consider it necessary, for reasons of extensibility and performance, to embed the software into a framework, as for example Zend Framework 2, in later versions when the software grows bigger. Thus, I considered it important to structure this version in a comprehensible manner to pave the way for a fast migration of the prototype into a framework. Considering the size of the prototype a fully optimized object oriented realization would have resulted in a great number of small classes and a very complex and incomprehensible structure of the program. In addition, the MVC pattern mentioned above

¹⁸ http://en.wikipedia.org/wiki/Single_responsibility_principle [accessed on 02.05.2015]

¹⁹ <http://framework.zend.com/> [accessed on 02.05.2015]

already suggests a structure for the software which led to the decision of defining objects in larger terms of offered components/modes providing separate public methods for single functionalities of the components.

Thus, each mode (manage vocabulary, browse vocabulary, test and analyse) was realized as a module providing separate classes for controllers and views. The model classes were not structured in terms of the offered modes, but each model class represents one table of the database. The view classes are only responsible for data representation and the forwarding of user requests and data input to their respective controllers. This means that view instances create instances of controllers and vice versa to communicate data and requests. Controllers create instances of models to retrieve data which is then presented to the users by instances of views, or they hand processed data over to model instances and in further consequence store the data in the database. Model classes were kept very simple and complex MySQL queries using joins and other complex MySQL statements were avoided to increase comprehensibility of these classes. Furthermore, this restricts the implementation of bussineslogic to controllers.

This architecture of the prototype means that smaller units of the software, as for example single words, were not implemented as single classes. Instead, the models use a global instance of a MySQL object for queries, which returns data from the database by default as arrays or as MySQL resultsets if indicated by a flag. Models have no other purpose than providing these data arrays for controllers or store data transmitted by controllers in the database. The data arrays retrieved by controllers are then manipulated by methods of the controllers, prepared for presentation to the user and finally delivered to views. The MySQL class is a kind of layer between the models and the database and provides a means to optimize object orientation in later versions of the program as it provides a method for MySQL queries and defines the result of queries, which could be changed so that query results are no arrays anymore but different objects or collections of objects are returned.

7.4 Implementation

This section of the thesis aims at providing insights into the functional principle of the developed software and focuses on technical aspects. Thus, several smaller parts of the complete classdiagram from figure 13 are presented and explained in detail. The most

significant parts of the program will be explained in greater detail and complemented with illustrations and explanations of source code. This section is structured in four sub-chapters: starting with a description of the basic system, then a brief overview of the browse and manage vocabulary modes are given, then insights into the functionality of the test mode are provided and as last part the analyze mode is presented. The focus will mainly be on controllers as they include the main functionality. The model classes will not be explained in further detail, except of one short paragraph when the test mode is explained.

7.4.1 Basic system

As already mentioned above, the basic system of the prototype is based on a basic system provided by Reimers and Thies (2012). In their system Reimers and Thies do not apply a MVC architecture which was adapted for this software. However, they provide a very simple, but useful basic system setup including the files `common.php`, `index.php`, `paths.php` and `settings.php`. Furthermore, their system includes a stylesheet, a class which provides database connectivity and a simple login. This basic system was borrowed, adapted and extended for the VOCABTrainer. Figure 14 shows the filesystem of the prototype, which clearly illustrates the simple and comprehensive structure of the program.

The file `index.php` is the entry point to the system and also handles the navigation through the program. Furthermore, this script checks a user's permissions to access sites according to their role.

The `index.php` includes the `common.php` script, which is a central script to include all other files (classes and scripts). Furthermore, the `common.php` creates the before mentioned global database object and starts a session. The `paths.php` is another short script which is included by the `common.php`, and sets the project path, the projectname and the server protocol. These files are responsible for basic system settings and are not intended to be changed.

Apart from the `index.php`, the `settings.php` file is probably the most important file to mention with regard to the basic system. The `settings.php` is a central place for setting error reporting, database connection data, the HTML title, the timezone, data to communicate with the

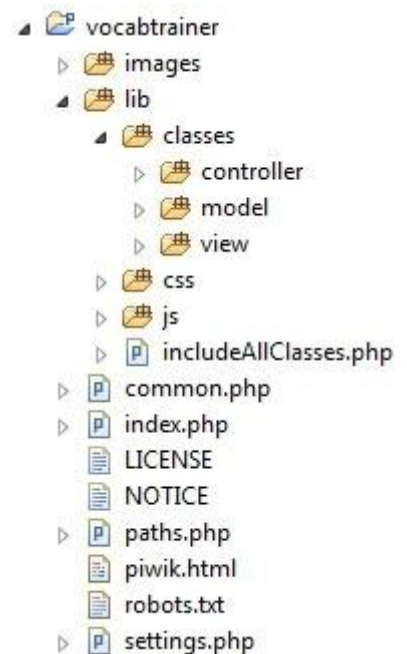


Figure 14: Filesystem

external usermanager (soap url, etc.), e-mail addresses for word suggestions and the revision interval for the test mode.

The two folders css and js contain the files default.css (stylesheet) and the default.js (JavaScript file containing jQuery code). In the current version of the prototype these two files contain all CSS code and JS code. The files LICENCE and NOTICE are simple text files containing licence information regarding the Apache Licence, Version 2.0.

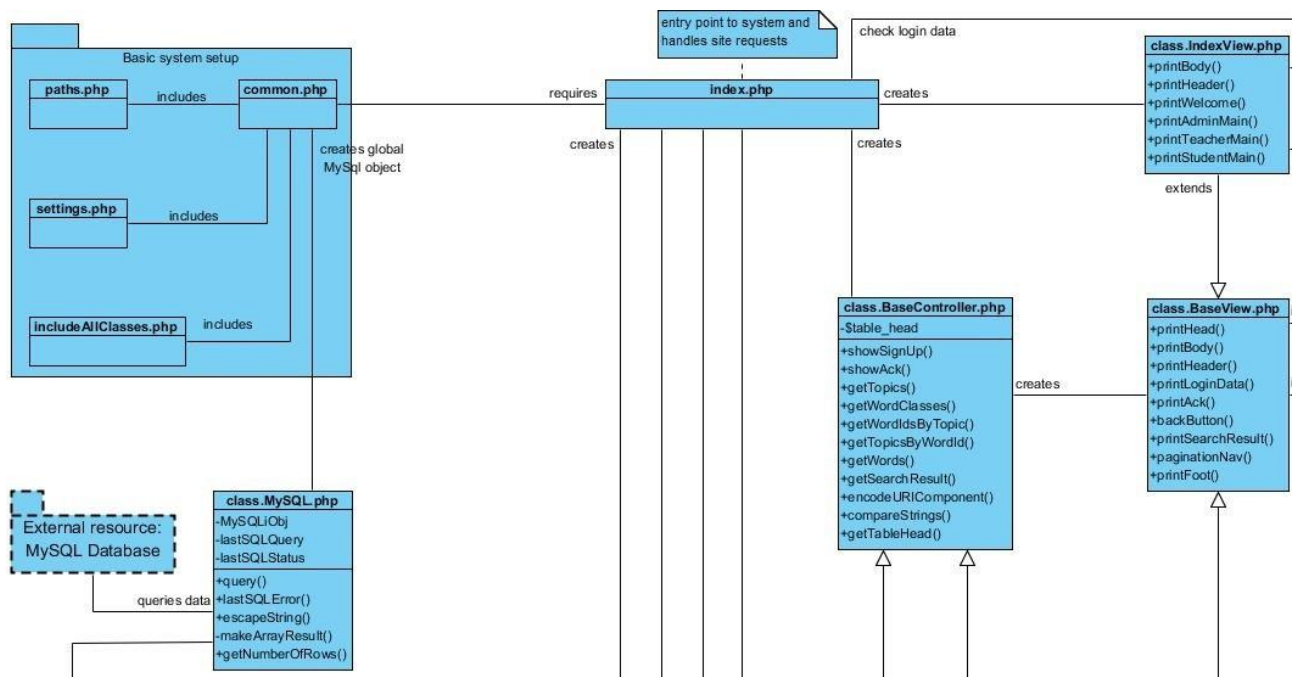


Figure 15: Classdiagram basic system

Figure 15 shows the left upper part of the complete classdiagram, which describes the basic system. This image shows three classes which were not mentioned so far: the BaseController, the BaseView and the IndexView. The BaseController is not too important for the basic system, but the BaseView and IndexView are. Especially, the BaseView is a very important class, because it defines the default structure of the HTML pages and provides methods used by all other views. Thus, all other views of the software extend the BaseView class. The IndexView is created by the index.php script and is responsible for printing a welcome screen for not logged in users, which includes a login form and a hyperlink to a sign up page. Furthermore, this class is also responsible for printing the main screen for logged in users according to their role.

Another part of the basic system is the login which communicates with an external

usermanager. The usermanager provides a SOAP (simple object access protocol) web service as interface to other programs. PHP offers a library to create SOAP clients²⁰ and servers. This library makes communication with the usermanager pretty simple. After a user has entered username and password an instance of the login class is created, which takes these two parameters. The login class prepares the necessary values (username, password, id and HMAC), creates a SoapClient instance and calls the isUserAllowed() method of the usermanager with the prepared values. Then the returned values are verified and if the login was successful the users table of the MySQL database is updated. Figure 16 shows the right upper part of figure 13 which illustrates the classdiagram of the login.

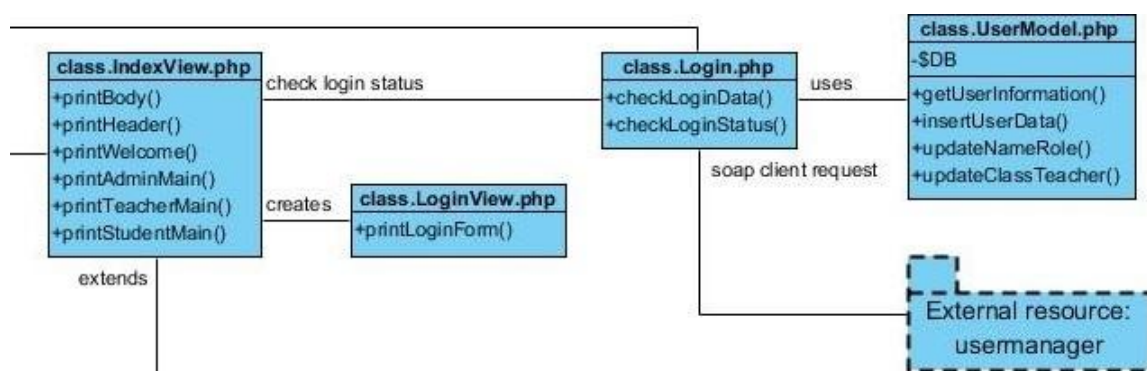


Figure 16: Classdiagram login

7.4.2 Browse and manage vocabulary modes

The two modules browse vocabulary and manage vocabulary are very similar regarding their controller classes. Thus, browse vocabulary is the same module providing less functionality. As both classes have very much in common, both of them extend the BaseController, which offers a number of methods to retrieve words, topics and word classes. The BaseController covers most functionality of the browse vocabulary component and great parts of the manage vocabulary component. At this point I should clarify that whenever I speak of the manage vocabulary component I refer to the class AddVocabulary or the file class.AddVocabulary.php illustrated in figure 17.

²⁰ <http://php.net/manual/de/class.soapclient.php> [accessed on 04.05.2015]

This section of the thesis is kept short and no source code is given as these modes of the program are not directly concerned with learning analytics. To start with, the main differences of the two modes are functions provided for administrators which allow the modification of the vocabulary database. As these functions influence both, the controller and view of the manage vocabulary mode, these two modes were not directly merged as planned in the conception stage but the BaseController was introduced as a parent class for both the BrowseVocabulary class and the AddVocabulary class. Although, the presentation of both classes looks very similar, the slight differences have a great impact on the implementation of the classes, and therefore, also the views were implemented separately.

Figure 17 shows the part of the full classdiagram which describes the manage vocabulary mode. Most methods of the controller class are very simple and need no further detailed explanation. Basically, the methods do what their name says and the source code of these functions is rather simple. The most complex part of source code is probably the updating of the ratings table. Thus, whenever a new word is added or deleted, or existing words are added to or deleted from a topic, the base_rating and other parameters in the ratings table have to be updated. Furthermore, the methods concerned with adding, deleting and manipulating data restrict user action in certain ways. Thus, to avoid double entries words can only be added if no entry with the same combination of English word and word class exists in the database. The German word entered can already exist in the database as some English words have the same German meaning. Deleting word entries is only allowed if a word has never been tested, as the removal of already tested words would lead to program errors and unusable data for analysis. Content words are not deleted instantly, but when they are not attached to a topic anymore they appear under the topic “unassigned words” where they can be deleted. Function words, however, are not assigned to a “real” topic and words which are deleted from this category are instantly deleted from the database. To indicate whether a word is only removed from a topic or deleted from the database, different symbols are used in the presentation. The red minus indicates that words can be removed from topics, whereas the red dustbin indicates that words can be removed from the database.

The most complex method of the AddVocabulary class is getSuggestions(). This method is called asynchronously by the help of jQuery and provides suggestions for existing words to be added to a certain topic as soon as the administrator starts typing. To do so the method first fetches all words from the database. Then the single word entries are compared to the user

input and if a word entry equals the user input it is pushed onto a temporary array. As a last step all words which are already part of the currently edited topic are removed from the array of words which contains suggestions of words. The word suggestions are then handed over to the view which presents the information to the administrator who can add them to the currently edited topic by clicking on a green plus symbol.

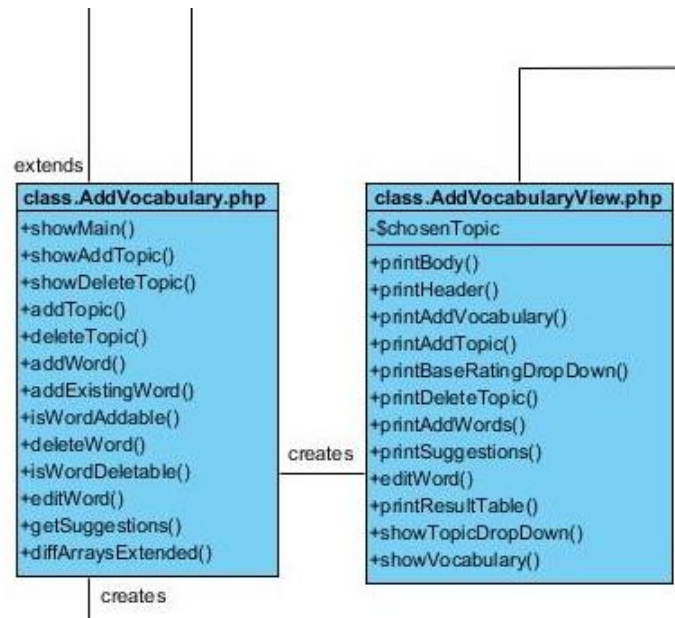


Figure 17: Classdiagram manage vocabulary

The browse vocabulary component is visually very similar to the manage vocabulary component, but does not provide symbols for modification of data. The most complex method of the BrowseVocabulary class is `getSearchResult()`, which is also used by the AddVocabulary class. Therefore, this method is member of their parent class BaseController. The method's functionality is similar to the `getSuggestions()` method, but is insofar different as it searches for word entries of the chosen topic. Figure 18 shows the classdiagram representing the browse vocabulary mode. A comparison of figures 17 and 18 clearly shows the difference of both components in terms of implementation, which one might not so easily identify when comparing the visual presentation or layout of both components.

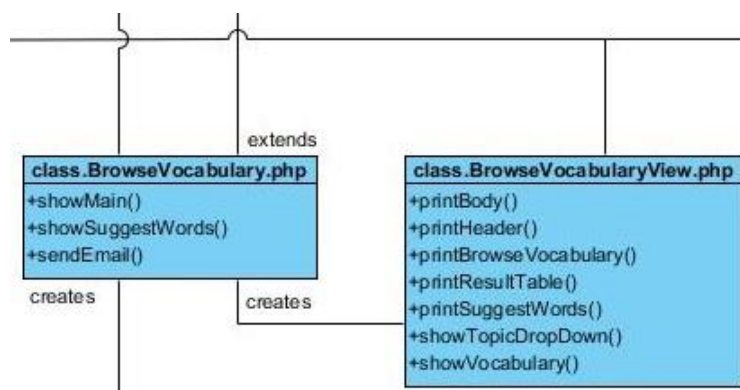


Figure 18: Classdiagram browse vocabulary

7.4.3 Test mode

In terms of learning analytics the test mode is central as this component collects all subject related data for analysis. The test mode has several rather short methods, which use different model classes to store general test related data in the database. These methods will not be explained in this section as they are simple and can easily be understood through comments in the source code. The most significant and extensive methods are `getWordsToTest()`, `getTestQuestion()` and `checkTestQuestion()`. Especially, the first two methods which compile the test questions are important and will be explained in greater detail in the following paragraphs. Figure 19 shows the classdiagram which describes the test mode. Similar to all other components of the software it also consists of a controller, which uses several model instances, and a view to present data to users.

At this point I shall refer back to chapter 6 where the concept of the software was explained. On the one hand the software is intended to be used in a school context and on the other hand one requirement demands that students should be able to use test mode “endlessly”. These two statements are insofar contradictory, as the use in a school context would demand teachers to choose vocabulary which is used in the test mode for their students. This is necessary as teachers often skip vocabulary provided by coursebooks or do not work through coursebooks in the suggested order. For this reason it would be important that teachers have a possibility to define the words to test in order to ensure that their students have already learnt these words. Such option was not implemented in the current version of the VOCABTrainer, because it was rejected in the conception stage. Regarding the order by which test questions

are chosen the requirements left two options open. Either questions are chosen because of a fixed order of topics, or tests are compiled according to the difficulty of words. A fixed order of topics was considered as no good solution, and therefore, I decided to choose test questions by a word's ratings/difficulty to compile homogenous tests, which are similarly difficult. This ensures that easy, medium and hard words are equally distributed over tests despite the fact that test have no fixed number of tested words. In addition to the three categories of difficulty a fourth category, which consists of words that should be revised, has to be taken into account. This category consists of words, which have already been tested at least once, but have not been tested for a certain period of time. This time interval is set to a minimum of three days in the current version of the software to ensure that during the evaluation period some words are revised to test this feature. This time interval can be changed in the settings.php script and should be increased to about two weeks or more after the evaluation to test students' long-term memory vocabulary repertoire. (cf. Thornbury 2012)

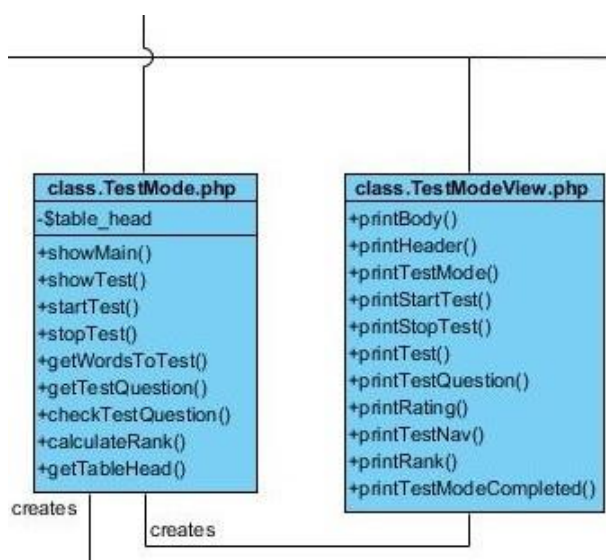


Figure 19: Classdiagram test mode

Source code 1 shows the last lines of the `getWordsToTest()` method, which returns all words available for testing in both languages separately. This function only returns words, which have not been tested yet, are not marked as learned (words are marked as learned if they were answered correctly twice in a row) and those words which fulfil the criteria for revision. The method assigns these words to the four categories revision, easy, medium and hard. The category revision was already explained above. All other categories contain words according to their rating. Words with ratings under 0,34 are in the category easy, words with ratings

between 0,34 and 0,66 are assigned to the category medium and all words with ratings higher 0,66 are assigned to the category hard. Furthermore, the percentage of words contained in each category with reference to the total number of words to test is calculated and rounded to full tens. All these values are stored in an array and forwarded to the method `getTestQuestion()`, which compiles the actual test questions.

```

1 <?php
2 class TestMode {
3     // [...]
4
5     public function getWordsToTest() {
6         // [...]
7         $percentages = array('revision_percentage' => $rev_perc,
8                             'easy_percentage' => $easy_perc,
9                             'medium_percentage' => $medium_perc,
10                            'hard_percentage' => $hard_perc);
11         $words_to_test = array('revision' => $revision,
12                               'easy' => $easy,
13                               'medium' => $medium,
14                               'hard' => $hard,
15                               'percentages' => $percentages);
16         return $words_to_test;
17     }

```

Source code 1: Returned multidimensional array from `getWordsToTest()`

```

20 public function getTestQuestion($nr_tested) {
21     // [...]
22     // Choose a word to test.
23     $difficulty_class = $nr_tested % 10;
24     $revision_span = $words_to_test['percentages']['revision_percentage'] / 10;
25     $easy_span = $words_to_test['percentages']['easy_percentage'] / 10;
26     $medium_span = $words_to_test['percentages']['medium_percentage'] / 10;
27     $hard_span = $words_to_test['percentages']['hard_percentage'] / 10;
28     // Get the class from which to choose a word: revision, easy, medium, hard.
29     if($difficulty_class < $revision_span)
30         $test_class = "revision";
31     elseif($difficulty_class >= $revision_span &&
32           $difficulty_class < ($revision_span + $easy_span))
33         $test_class = "easy";
34     elseif($difficulty_class >= ($revision_span + $easy_span) &&
35           $difficulty_class < ($revision_span + $easy_span + $medium_span))
36         $test_class = "medium";
37     elseif($difficulty_class >= ($revision_span + $easy_span + $medium_span) &&
38           $difficulty_class < ($revision_span + $easy_span + $medium_span + $hard_span))
39         $test_class = "hard";
40     // Randomly choose a word to test from the test class.
41     $test_word_rand = array_rand($words_to_test[$test_class], 1);
42     $test_word = $words_to_test[$test_class][$test_word_rand];
43     // [...]
44     // Finally, get the word for the question and the word for the answer.
45     $question_word = $word_to_test[0][$question_lang];
46     $answer_word = $word_to_test[0][$answer_lang];

```

Source code 2: Compilation of test questions (1/2)

Source code 2 illustrates the actual process of choosing test questions. The basic principle

behind this algorithm is to distribute words from the four categories mentioned above over ten test questions according to the percentage of words each category contains. For example: Assuming that the number of words to test is exactly one hundred, and words are already assigned to the four categories, so that the category revision contains twenty words, easy contains forty words, medium contains thirty words and hard contains ten words. This would lead to percentages equal to the number of words of each category. With such constellation every ten questions of a test would start with two revision words, followed by four easy words, three medium words and one hard word. In this way tests are always equally difficult and it is avoided that some tests might only contain very easy words, while others might only contain very hard words.

In the image above the first thing in line 23 is to calculate from which difficulty class a word is chosen. This is done by calculating the number of already tested words of the current test modulo ten, which results in a number between zero and nine. In lines 24 – 27 the span of each category is calculated by simply dividing the rounded percentages by ten, which results in numbers between zero and ten. Lines 29 – 39 then provide the category from which to choose a word. In lines 41 and 42 first a random array index is calculated and then the word at this index is chosen. Then the word's translation is fetched from the database, which was left out in source code 2. In lines 45 and 46 the question word and the correct answer in the other language are assigned to variables.

After the test question and its correct answer have been chosen three wrong answers must be added, which is illustrated by source code 3. First of all, two empty arrays are initialized. In lines 53 – 64 an array of all words is passed through in a loop and each word is compared to the correct answer by use of the PHP function `similar_text()`²¹ which returns the similarity of two strings and a value in percentage. If the words are not equal, the correct answer string is not contained in the compared string or vice versa, and the compared word is not already in the array, then the compared word and its percentage of similarity are pushed onto an array. The rest of the source code illustrated above finds the three most similar words in the array by comparison of the percentages returned by the `similar_text()` function.

²¹ <http://php.net/manual/en/function.similar-text.php> [accessed on 05.05.2015]


```

47 // Get the wrong answers. Calculate the similarity of the answer word and all
48 // other words and save to array. Don't save if it is exactly the same (100),
49 // or the correct answer contains the same string as the wrong answer or viceversa.
50 // Don't save same words twice.
51 $similarity_array = array();
52 $class_array = array();
53 foreach($_SESSION['all_words'] as $row) {
54     similar_text($answer_word, $row[$answer_lang], $percent);
55     $temp = array('wrong_answer' => $row[$answer_lang],
56                 'percent' => $percent);
57     $temp_class = array('class' => $row['class']);
58     if($percent != 100 && strpos($answer_word, $row[$answer_lang]) === false &&
59         strpos($row[$answer_lang], $answer_word) === false &&
60         in_array($temp, $similarity_array) === false) {
61         array_push($similarity_array, $temp);
62         array_push($class_array, $temp_class);
63     }
64 }
65 // Save the three most similar words to array.
66 $wrong_answers = array(
67     array('wrong_answer' => "", 'percent' => 0, 'class' => ""),
68     array('wrong_answer' => "", 'percent' => 0, 'class' => ""),
69     array('wrong_answer' => "", 'percent' => 0, 'class' => ""));
70 for($i = 0; $i < 3; $i++) {
71     for($j = 0; $j < count($similarity_array); $j++) {
72         if($similarity_array[$j]['percent'] >= $wrong_answers[$i]['percent']) {
73             $wrong_answers[$i]['wrong_answer'] = $similarity_array[$j]['wrong_answer'];
74             $wrong_answers[$i]['percent'] = $similarity_array[$j]['percent'];
75             $wrong_answers[$i]['class'] = $class_array[$j]['class'];
76             $k = $j;
77         }
78     }
79     unset($similarity_array[$k]);
80     $similarity_array = array_values($similarity_array);
81     unset($class_array[$k]);
82     $class_array = array_values($class_array);
83 }
84 //[...]
85 }
86 }
87 ?>

```

Source code 3: Compilation of test questions (2/2)

The complete test question is then passed to the view method `printTestQuestion()` which presents the question to the user. The actual user interaction is handled by jQuery and also the visual validation of the test answer is implemented in jQuery. When a user clicks the next button to get the next question the learners' produced data (answer and rating) are passed to the controller which enters the data into the database. This process is a bit sensitive as all users, who run the test mode at the same time, modify data in the ratings table. Although probability is very low, it could happen that two users answer the same question word at nearly the same time and want to update the rating of a word simultaneously. As this could

lead to mistakes in the calculation of a word's rating the model uses MySQL transactions²², which allow the temporary locking of single rows of a database table.

Another feature presented to learners is their rank, which is an indicator for their overall performance. This indicator is also shown in the analyze mode and will be explained in detail in the next section.

7.4.4 Analyze mode

The analyze mode similar to all other components provides its functionality by a controller class, a view class and several model classes which are illustrated as classdiagram in figure 20. This component is the largest and regarding learning analytics the most important one. The view class again is only reliable for the presentation of data and mostly consists of straightforward HTML. The controller class is pretty large as it provides methods for all three user groups. However, as one requirement was to not provide diagrams or other complex representations many of the methods below are rather simple as they provide general statistics and information about users, tests, words (learned, errors, progress) and mistakes in the form of tables. Most interesting with regard to learning analytics are probably the topic statistics as they point out the learning progress with regard to specific topics or semantic fields. Other interesting tools are the rank and self-assessment indicators. These indicators inform students about their overall performance and their self-assessment according to their ratings on erroneous words. These tools are also used as a means to inform teachers about the general performance of their class in a table showing all their classes, and further in a more detailed table which lists all students of a class to point out strong and weak students. This allows teachers to quickly identify weak students. The following paragraphs will focus on a more detailed description of the topic statistics, and the rank and the self-assessment indicators. All other features will be briefly explained in section 7.5, which provides some screenshots of the final prototype.

²² <http://dev.mysql.com/doc/refman/5.6/en/commit.html> [accessed on 05.05.2015]

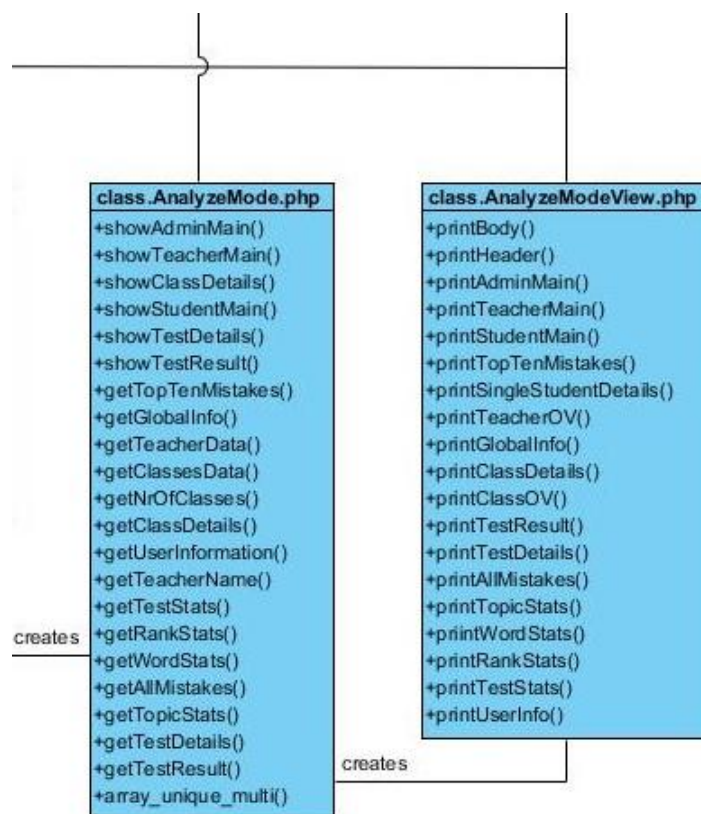


Figure 20: Classdiagram analyze mode

Topic statistics are collected and prepared by the method `getTopicStats()` in the class `AnalyzeMode`. Theory on language acquisition states that it is considered easier for students to learn words ordered in semantic fields, but does not provided answers to questions about problems that students might have in vocabulary acquisition, and if or how these problems are related. However, if it is easier for students to learn semantically related lexical items, and theory on the mental lexicons is correct, it could be assumed that specific semantic fields are more difficult to acquire for individual students than others. Thus, any problems in the acquisition process could manifest within specific semantic fields. The topic statistics try to provide an analysis of mistaken words in relation to their assigned semantic fields, by highlighting those semantic fields where a learner’s proportion of mistakes is greater than 40%. This allows early discovery of problematic areas and allows teachers to intervene in a student’s learning process by directing a student’s attention towards such areas.

Technically, this method is rather complex as it has to fetch a great amount of data from the database. Thus, the method gathers all tested words of a learner and all errors of a learner and assigns these data to their respective topics. Then the total number of words assigned to a

topic, the number of tested words of each topic, the number of learned words of each topic and the number of errors of each topic are assessed. All collected and sorted data is then forwarded to the view by calling its `printTopicStats()` method, which presents the data in the form of a table to users.

Two other important features of the analyze mode are the rank and self-assessment indicators. In this version of the software the calculation of the values to indicate a student's rank and self-assessment are kept pretty simple and probably need refinement when a greater amount of learner produced data is available.

The ranks basically represent a grade, which students would receive on their test performances in a school context. This feature indicates a student's overall performance and is presented to learners in the test mode and in the analyze mode, and to teachers in the class overview provided in the analyze mode. Ranks are represented by symbols in the form of three stars, which can either be unfilled, half-filled or filled. Thus, one filled and 2 unfilled stars indicate a very weak performance on tests, whereas three filled stars indicate a very strong performance on tests. On the one hand the purpose of this indicator is to provide a means for teachers to quickly identify weak students. On the other hand it should motivate weak students to increase their poor ranks and strong students to keep their good ranks.

Source code 4 illustrates the technical realisation of the rank and self-assessment indicators. The method `getRankStats()` is responsible for the calculation of values represented by both features. In lines 8 – 16 the overall rank and the rank of a student's latest test are calculated. Thus, the method fetches the number of tested words of a user from the database, as well as the total number of errors a student made and calculates the percentage of correct answers in line 10. The calculation of the latest test rank is done in a similar way, but only takes into account the performance of the latest test. The calculated values are delivered to the view, which presents a combination of unfilled, half-filled and filled stars to the users. The lowest possible rank, one filled star and two unfilled stars, signifies that a user has answered less than 60% of questions correctly. The highest possible rank, three filled stars, means that a user has answered more than 96% of questions correctly. All other ranks indicate percentages inbetween.

```
1 <?php
2 class AnalyzeMode {
3     // [...]
4
5     public function getRankStats($username) {
6         // [...]
7         // Calculate overall rank.
8         $nr_of_tested_words = $tested_words_model->getNrOfTestedWords($test_ids);
9         $nr_of_errors = $tested_words_model->getNrOfErrors($test_ids);
10        $overall_rank = 100 - ($nr_of_errors * 100 / $nr_of_tested_words);
11        // Calculate latest test rank.
12        $nr_of_test_ids = count($test_ids);
13        $latest_test_id[0] = $test_ids[$nr_of_test_ids - 1];
14        $nr_of_tested_words = $tested_words_model->getNrOfTestedWords($latest_test_id);
15        $nr_of_errors = $tested_words_model->getNrOfErrors($latest_test_id);
16        $latest_rank = 100 - ($nr_of_errors * 100 / $nr_of_tested_words);
17        // Calculate self-assessment
18        $self_assessment = 0;
19        $error_ratings = $tested_words_model->getErrorsForSelfAssessment($test_ids);
20        $error_ratings = array_filter($error_ratings);
21        if(!empty($error_ratings)) {
22            foreach($error_ratings as $row) {
23                if($row['rating'] > 0.5)
24                    $self_assessment -= -(0.6 - $row['rating']);
25                else
26                    $self_assessment += (0.4 - $row['rating']);
27            }
28        }
29        // [...]
30    }
31 }
32 ?>
```

Source code 4: Calculation of rank and self-assessment indicators

The source code above also illustrates the calculation of the self-assessment indicator. The purpose of the self-assessment scale is to raise the students' awareness of their, as the name of this feature indicates, self-assessment. Furthermore, it should implicitly encourage students to honest ratings of words in the test mode. Thus, it analyzes a learner's ratings of wrongly answered questions. The feature follows the assumption that correctly answered words are most likely rated as rather easy and wrongly answered words are most likely perceived as rather difficult by learners, and therefore, they should be rated as rather difficult. Thus, the rating of words can be seen as reflecting a student's self-assessment of their knowledge.

The class `AnalyzeModeView` presents the self-assessment value using the same symbols used to rate words in the test mode – smileys. The calculation of the self-assessment value is illustrated in source code 4 in lines 18 – 26. In line 18 the value is initialized to zero, which means that self-assessment is assumed to be very good at the beginning. Then the method fetches all ratings of a student's mistakes. In lines 21 – 26 each of these values is compared to 0,5 which indicates whether a word was rated as rather easy (lower than 0,5) or as rather

difficult (greater than 0,5). If a wrongly answered question was rated as rather difficult (0,8 or 1) the self-assessment value is decreased by 0,2 or 0,4 which indicates that self-assessment was good. If a wrongly answered question was rated as rather easy (0 or 0,2) the self-assessment value is increased by 0,2 or 0,4 which indicates that self-assessment was poor. If a wrongly answered question was rated neutral (0,4 or 0,6) the self-assessment value is not changed. Thus, the self-assessment value is either not changed, or increased/decreased by 0,2 or 0,4 for each error, which overall results in a value below zero, between zero and one, or greater than one. If this value is below zero or greater than one it is rounded to zero or one respectively, which was left out in the source code above in line 29, and presented to the learners and teachers by the view. It should also be mentioned that the self-assessment value has no influence on the rating/difficulty of a word, but this feature has a mere informative purpose.

This section explained the main structure of the developed software and pointed out the most important technical aspects by detailed descriptions of its operating principles. In order to fully understand the technical aspects and gain a clear impression of the actual functionality of the software the following sub-chapter provides screenshots of the final prototype which illustrate the layout. In addition to the mostly self-explanatory screenshots, brief textual descriptions should provide a clear impression of the VOCABTrainer's functionality.

7.5 Layout

The layout of a website is crucial for its success and the probably most decisive factor in the planning and creation of a layout are a website's intended users. Developers and webdesigners should try to create a user-friendly layout, and the same time they should try to relate the design of a website to its contents. This process is difficult and demands some experience in webdesign as the risk of overdoing it visually is high. From my own internet experience I know that visually elaborate and sophisticated websites often lack usability as layout and design hinder easy access of contents. For this reason, simplicity had top priority in the planning of the layout.

CSS and HTML offer possibilities to create different layouts for different groups of users. At the beginning of the project I intended to create different layouts for students and teachers. However, as one requirement was to create a not too infantile layout, because students at this age are in the middle of crossing the border from childhood to adulthood and the website should not be classified as too immature by them, I decided to stick to one layout for all user groups, which is kept neutral.

7.5.1 Basic layout

Figure 21 depicts the welcome screen of the software. The layout of the prototype is exactly the same on all pages, consisting of the same header, a content area in the middle and footer at the bottom. For logged-in users the header provides a logout button in the form of a red "X" in the top right corner of the page next to the name of a user. The footer provides several links to other pages and default pages which can be accessed by even not logged-in users. The hyperlink *Home* on the left directs users back to the first screen after the login, which provides only two possibilities to proceed: enter the analyze mode, or enter the test mode/browse vocabulary mode/manage vocabulary mode depending on the user's role. The hyperlinks *Contact*, *About*, *Learn-Apps for Students* and *Learning Apps TU Graz* are links to official pages of the TU Graz, which open in a new blank tab of the browser. The last hyperlink *Acknowledgement* shows a page containing text that states the source of the symbols used for the prototype. The content area in figure 21 shows the login form and the text above this form provides a hyperlink to the sign up page of the external usermanager.

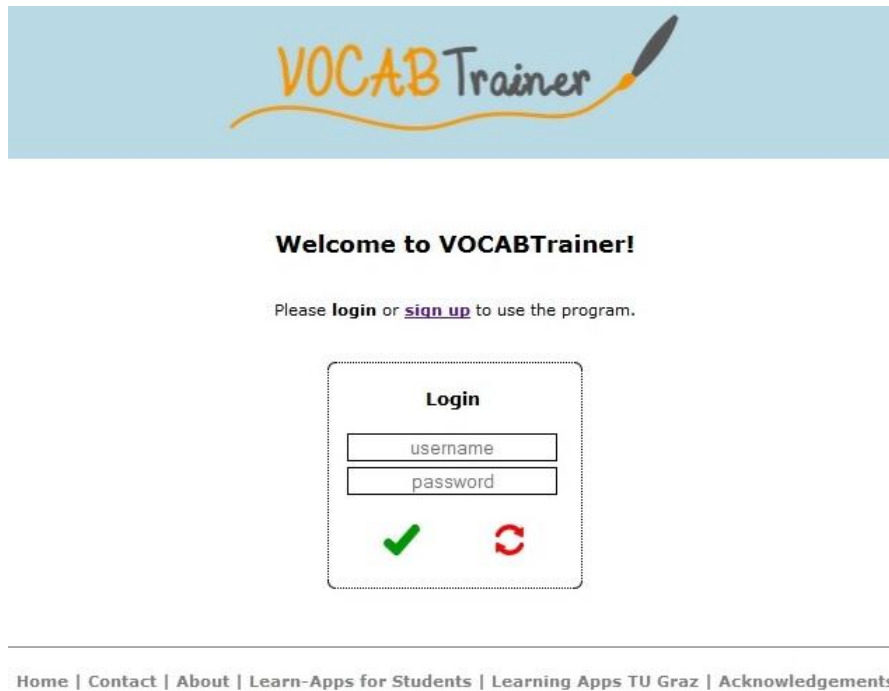


Figure 21: Basic layout of the prototype

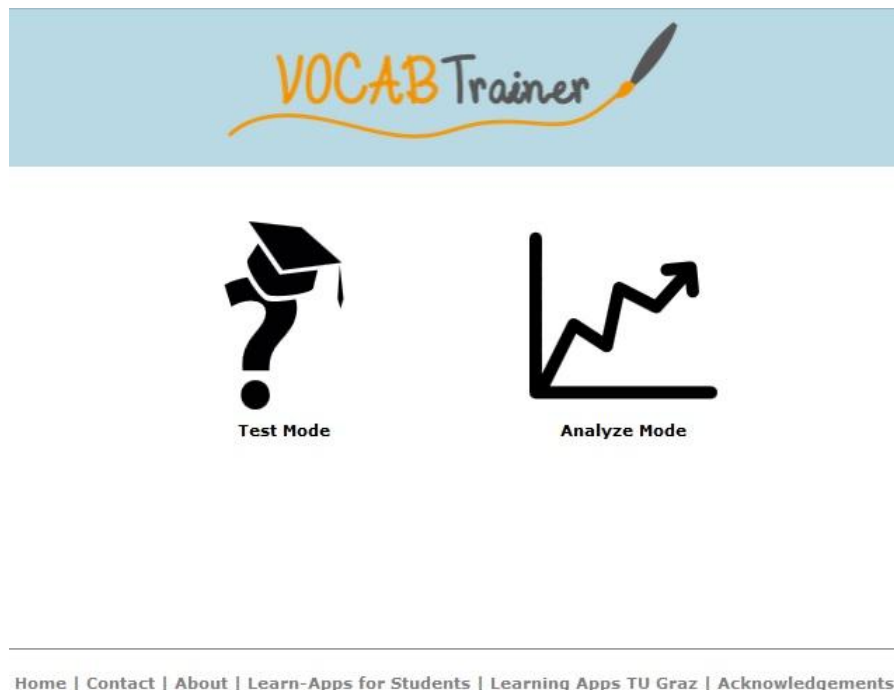


Figure 22: First screen for logged-in users (student)

Figure 22 above shows the first screen presented to logged-in users. This screen is nearly similar for all users. Only the symbol on the left is different depending on a user’s role.

7.5.2 Browse and manage vocabulary modes

When administrators or teachers enter the manage or browse vocabulary modes the content area looks similar to figure 23. Figure 24 shows the manage vocabulary mode. The only difference between the displayed content for administrators and teachers is the small box on the left side. In the upper part of the box users of both groups can choose a topic to display in the larger content area. While teachers find a hyperlink below the box, which directs them to a form to suggest words, administrators find two links at the same position, where one link leads to a form to add topics and the other to a form to delete topics.

The main part of the content area, which shows the vocabulary assigned to a topic allows the searching of the table by input of words in either English or German, and/or by choosing a word class from the drop down menu. The search function filters words in the table as soon as the user starts typing. For teachers this screen looks similar with all topics, whereas for administrators this view differs for the topic “all” and the other topics (compare figures 23 and 24). Another difference of the topic “all” to all other topics is the table column “Topic(s)”, which lists all topics a word is assigned to. Users can also browse the table manually by the pagination below the table.

Choose a topic:

- All
- Unassigned words
- Actions/Movement
- Animals
- Body
- Clothes

Suggest words to add...

Topic: All

English	German	Class	Type	Topic(s)
a	ein, eine	determiner	function word	Function words
a bit	ein wenig, ein Stück	adverb	content word	Universal words
a lot (of)	viel, viele, sehr, viel, sehr viel	adjective	content word	Universal words
about	über, von	preposition	function word	Function words
above	über	preposition	function word	Function words
activity	die Tätigkeit, die Betätigung	noun	content word	Universal words
add	hinzufügen	verb	content word	Universal words
address	die Adresse	noun	content word	Universal words
adult	der/die Erwachsene, ein Erwachsener	noun	content word	People
adventure	das Abenteuer	noun	content word	Universal words
after	nach, danach, nachher	adverb	content word	Universal words
afternoon	der Nachmittag	noun	content word	Time/Duration
ago	vor, her, vorher	adverb	content word	Universal words
alien	der Alien, der Außerirdische	noun	content word	Universal words
all	alle	adjective	content word	Universal words, People
alone	allein	adjective	content word	Universal words
aloud	laut	adjective	content word	Universal words
alright	in Ordnung, schön, na gut	adverb	content word	Universal words
also	auch, ebenfalls	adverb	content word	Universal words
always	immer	adverb of frequency	function word	Function words

< prev **1** 2 3 ... 56 next >

Figure 23: Browse vocabulary mode

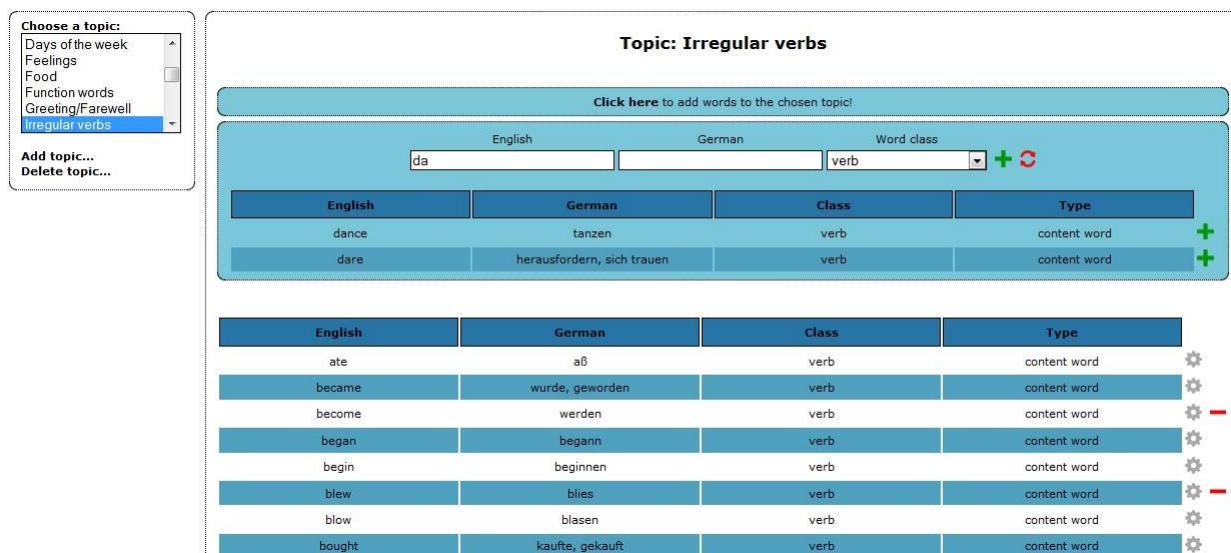


Figure 24: Manage vocabulary mode

The content area of figure 24 shows the manage vocabulary mode. The view of this mode adds some features to the table. The most obvious feature is the blue slide box below the heading which allows administrators to add words to the chosen topic. In order to add a word all three input fields must contain content and several other criteria, which were explained in the section “Implementation”, must be met. As soon as administrators start typing or choose a word class from the drop down menu, a table with suggestions of words from other topics appears below the input fields. This allows them to quickly add individual words to several different topics.

Another difference between the manage and browse vocabulary modes are the symbols on the right side of the tables. The green plus in the slide box allows the quick adding of words to a topic. The grey gearwheel allows administrators to edit word entries and the red minus removes words from a topic, but does not delete words permanently. If a content word is not assigned to any topic, it is listed in the topic list under “Unassigned words”. In this category and in the category “Function words” the red minus is replaced by a red dustbin symbol, which indicates that words are deleted permanently.

Altogether the use of both modes is very intuitive from my point of view, and the features for managing the vocabulary make the extension and maintenance of the database simple and fast. This statement is based on my own experience from filling the database with about a thousand one hundred word pairs within two days.

7.5.3 Test mode

When learners start the test mode they are directed to a page which provides instructions on how to use the test mode. These instructions are given in English and German. This page also shows a large start button.

After clicking on the start button learners are provided with the first test question illustrated in figure 25. This part of the software is the most interactive part realized to great parts using jQuery. At first, users only see the test question and their overall rank on the right side. The smileys for the rating, the green tick to submit the answer and rating, and the next and stop buttons are hidden. As soon as learners choose their answer by clicking on one of the four light blue boxes the smileys to rate the question appear. Only after learners have rated a question the green tick below the rating appears. As long as students do not submit their answers and ratings they can change both. When clicking on the green tick the answer is submitted and the correct answer is highlighted. Furthermore, the next and stop buttons on the right appear which either lead to the next test question or stop the test. Figure 25 below shows a correctly answered question, which is highlighted by changing the color of the chosen box to green.

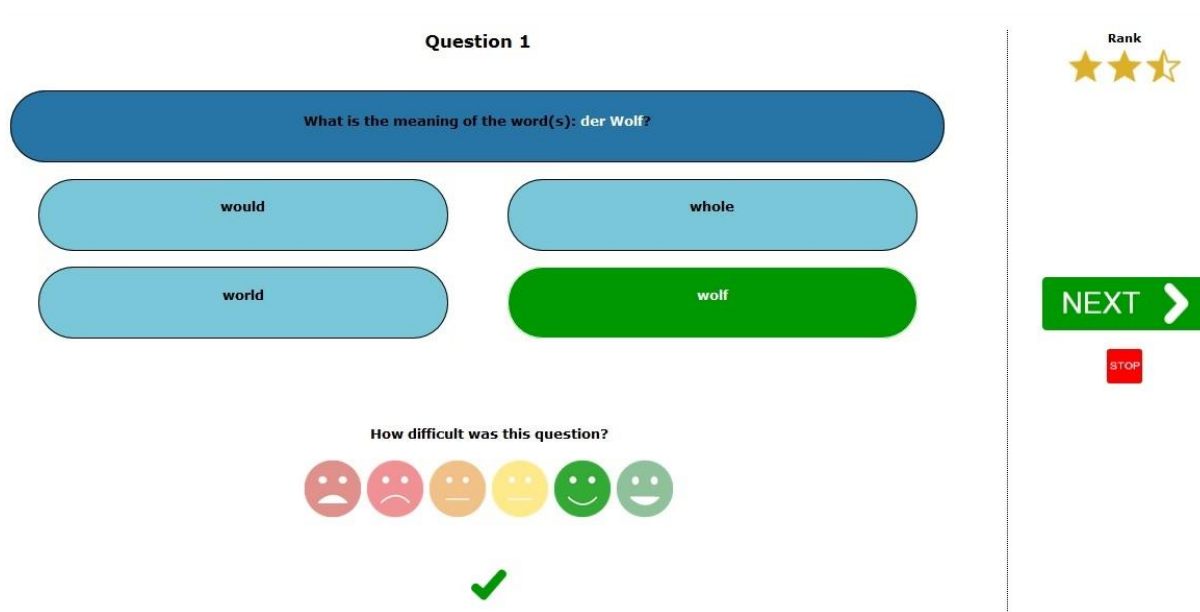


Figure 25: Test mode – correct answer

In contrast to the image above, figure 26 below shows a wrongly answered test question,

where the color of the box with the student's answer was changed to red and the color of the box containing the correct answer was changed to green.

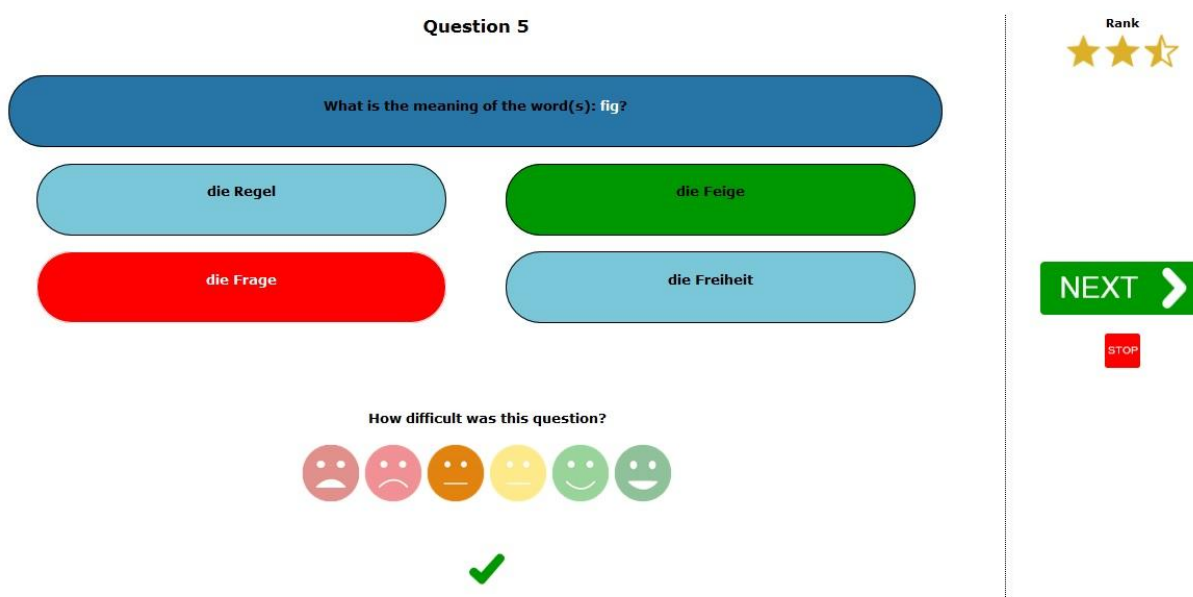


Figure 26: Test mode – wrong answer

When a test is stopped a record of the test result in form of a table is presented to the student. On top the record shows the number of questions, correct answers and wrong answers. Below the single questions are listed in a table showing the question word, the answer given by the student and the correct answer. Errors are highlighted in red.

7.5.4 Analyze mode

The analyze mode consists of several different pages. Permission to access these pages depends on the role of a user. Administrators are provided with some general global information about people, who use the program, the number of words and topics stored in the database, and a summary of test statistics of all users. Furthermore, they can access the analyze modes of all teachers and students. Another box lists all independent students, who are not assigned to a teacher. This box looks almost similar to figure 28 and allows access to the students' detailed analyze modes illustrated in figure 29. Administrators find another analysis tool at the bottom of their analyze mode view, which lists the ten most common mistakes made by all users.

The teachers' analyze mode consists of three boxes. The first box on top left shows a teacher's profile information, which is actually not important for analysis but rather constitutes a kind of means for navigation for administrators. Another box at the bottom of this page, lists the top ten errors of students assigned to a teacher. The most important analysis tool for teachers is contained in the third box, which lists all classes of a teacher as illustrated in figure 27. This overview of classes helps teachers to gain a clear impression of the overall performance of their classes. The box provides the number of students of a class, who have already logged-in once, the average overall rank, the best student's rank and the weakest student's rank. Furthermore, a hyperlink to a detailed class overview is provided in this table.

Class	Nr. of students	Ø Rank	Best rank	Worst rank	Details
...	10	☆☆☆	☆☆☆	☆☆☆	Details...

Figure 27: Analyze mode (teacher) – list of classes

The detailed class overview is depicted in figure 28 below. This view presents a list of all students of a class in the form of a table. The table shows the students' names, number of tested words, number of mistakes, their overall rank, their self-assessment indicator and provides hyperlinks to the detailed analyze modes of single students. The purpose of this view is to offer teachers a means to quickly assess the general performance of students of a single class and to identify weak and strong students of a class. Furthermore, it allows teachers to assess the extent to which the program is used by individual students. The numbers of tested words and mistakes in a way reflect a student's effort put into learning vocabulary.

Name	Nr. of tested words	Nr. of mistakes	Rank	Self-Assessment	Details
...	115	5	☆☆☆	●●●●●	Details...
...	57	4	☆☆☆	●●●●●	Details...
...	111	7	☆☆☆	●●●●●	Details...
...	178	15	☆☆☆	●●●●●	Details...
...	15	1	☆☆☆	●●●●●	Details...
...	18	0	☆☆☆	●●●●●	Details...
...	126	10	☆☆☆	●●●●●	Details...
...	339	13	☆☆☆	●●●●●	Details...
...	17	0	☆☆☆	●●●●●	Details...
...	92	1	☆☆☆	●●●●●	Details...

Figure 28: Analyze mode (teacher) – class details

Figure 29 shows the students' analyze mode, which offers the most detailed analysis of all views. The first box in the top left corner of the image shows a user's profile information. Depending on whether a student is assigned to a class/teacher this box shows more or less information. Test statistics include general statistical data about a user's completed tests. The most interesting data included in the test statistics are probably the separate error percentages for the two test directions. Aitchison (2012) explains that scientists, who are involved in research on the mental lexicon, believe that different languages can be seen as individual language systems in the human brain, which have connections to each other. If these statistics are heavily onesided this might indicate a problem in a student's learning process, as these links might only be developed in one-way. This enables teachers to intervene in the student's learning process by making them aware of this problem and highlighting the importance of learning new words equally well in both directions. Moreover, the test statistics provide a link to test details, which lists all completed tests in a table and allows access of detailed test records of single tests.

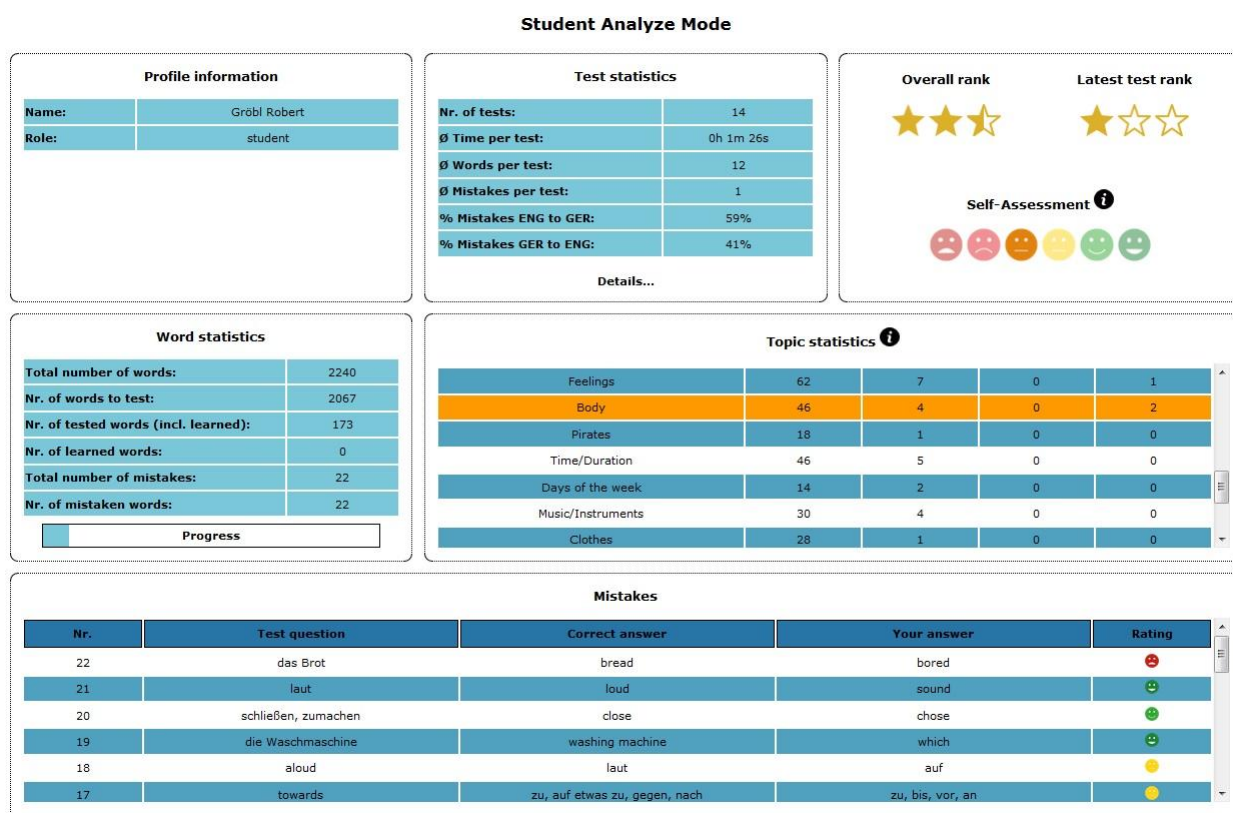


Figure 29: Analyze mode (student)

The box containing the word statistics aims at informing learners about their progress by providing statistical data in the form of a table. The table includes the total number of words in the database, the number of words which have not been tested yet, the number of tested words, the number of words which are marked as learned, the total number of errors which also counts multiple mistakes of the same word, and the number of mistaken words which does not separate multiple mistakes of the same word. Furthermore, this box provides a progressbar to visualize a student's progress. The number of tested words is indicated by a blue bar which can be seen in figure 29 and the number of learned words is shown as a green bar, which covers the blue bar. In figure 29 the green bar is not visible yet as no tested words have been marked as learned so far.

At the bottom of the page learners and their teachers are provided with a list of all mistakes of a student. This table enables students to quickly revise difficult words. However, the two most important features from my point of view are the rank and self-assessment indicators, and the topic statistics. These features were already explained in great detail in the section "Implementation" of this thesis and will not be explained in too much detail at this point. As can be seen in figure 29 the top right box in the analyze mode holds the rank and self-assessment indicators. In this image the self-assessment scale indicates that the self-assessment of the learner is mediocre. A closer look at the mistakes table reveals that the learner often rated wrongly answered questions as rather easy which results in a not too good self-assessment value. The feature topic statistics provides a list showing all topics/semantic fields from which words have been tested so far. The columns of this table show the topic names, the number of words assigned to a topic, the number of tested words, the number of learned words and the number of errors. Summing these numbers up results in different values than in the table word statistics, because words might be assigned to several topics. Topics with a percentage of mistakes higher than 40% and lower than 60% are highlighted in orange. Topics where the percentage of errors is higher than 60% are highlighted in red. Thus, this feature informs teachers and learners about problem areas of individual students in terms of semantic fields. From my point of view this features has great potential for extension which could lead to valuable findings regarding learning analytics, language acquisition in general and more specifically about the structure of the mental lexicon. My vision for extension of this feature will be explained in further detail in chapter 9 of this thesis.

8 Evaluation

This chapter is concerned with the preparation, procedure, results and feedback of the evaluation of the software. The evaluation is an integral part in the development process of software as it often reveals flaws of computer programs and provides inspiration for improvement.

The evaluation of the VOCABTrainer was conducted with a second grade in secondary school. Although, the software is actually intended for first graders I decided to ask a second grade for their help as with learners of their language level I could be sure that they already learnt the tested words. For reasons of school infrastructure and exams the evaluation did not proceed as expected, which will be further explained in the following sections.

8.1 Preparation

The first thing in the planning of the evaluation was to contact an English teacher, who was my mentor during the school practice. I already asked her if it was possible to conduct the evaluation with her class when I did my second school practice. The class consists of seventeen students, which I considered enough for a first evaluation of the program.

Originally I planned to start with the evaluation by introducing the program to the students in the computer room of their school in one lesson. I planned to use a few minutes of the lesson to explain how students can access the software. The rest of the lesson was intended as time for the students to use the VOCABTrainer and to gather some feedback of the students. Furthermore, I planned to ask the students to use the application for another week at home to gather more learner produced data.

However, as a matter of poor infrastructure in Austrian schools it was not possible to find a date on which I could use the computer room of the school with the students. Thus, I had no other options as to either find another school for the evaluation, or to stick with the class and change the procedure of the evaluation. I decided on the latter and changed the procedure in so far as I visited the class and introduced the software to them. I had no other choice as to ask them to use the program on a daily basis for one week at home. In order to make the whole procedure as comfortable as possible for the learners I prepared user accounts for them, and

handed out sheets, which included the URL to access the VOCABTrainer, their user data and brief instructions for the use of the program (see appendix). Furthermore, I kindly asked the teacher to inform me about possible problems the students had with accessing or using the program. In addition, I provided a simple feedback sheet for the students, and asked them to take notes of possible errors they encountered. I also asked the teacher to take a look at the program at the end of the evaluation and give me some feedback about the usefulness of the analysis tools for her teaching.

The necessary change of the procedure implied that I had only very little control over the evaluation process as the success of the evaluation fully depended on the students' motivation and computer skills. The evaluation started on 8th of April 2015 and, after a one week extension, ended on the 22nd of April.

8.2 Procedure

The evaluation started on 8th of April 2015, when I visited the class and introduced the software to them. Although, I had originally intended to only demonstrate how to access the application and login into the program, I changed this plan and also explained the use of the test and analyze modes to the students, because I had to ensure that each student understood how to use the program. Thus, the introduction of the software took about twenty minutes. After the explanation I handed out the sheets containing the learners' login data and the feedback sheet, which they returned after the evaluation.

I checked the evaluation progress on a daily basis through the software and noticed from the beginning that the evaluation only proceeded very slowly. After four days of evaluation only seven students had used the program regularly. Thus, I contacted the teacher and asked her whether her students had problems to access the program. She assured me that there were no problems and she would encourage the other students to also use the program. As half of the intended evaluation period had been over at this point, I decided to extend the evaluation for another week.

However, after two weeks of evaluation only ten students had used the software and three of them only used it once. On the 22nd of April the evaluation ended and I visited the class again to gather some oral feedback and collect the feedback sheets. When I asked the class why

only half of them participated in the evaluation, most of the students who did not participate told me that they could not find the website. The reason for this was that some of them had put “www” in front of the URL, as they told me that they thought that every URL would start with the “www” tag. Another reason for the students’ minor participation were exams for which they had to study.

Altogether the evaluation of the program was not sufficient and I expected to gain a greater amount of data. Also the communication with the teacher was rather poor and I had the impression that she had only very little interest in the program, as she did not provide any feedback and told me that she had no time to use the software. However, even if the evaluation proceeded as expected I doubt that the gathered data would be more meaningful than the now available data as two weeks of evaluation with one class from my point of view are not enough to assess the usefulness of this software. I will further elaborate on the importance of a long-term evaluation of the software in chapter 9.

8.3 Results

As mentioned above the results of the evaluation are not all that significant as the amount of data is insufficient. In spite of that I will still provide a brief discussion of the results in the following paragraphs.

Figures 27 and 28 on page 67 illustrate the class overview and class details after the evaluation. The ranks in these images clearly point out that all learners performed pretty well during the evaluation. The self-assessment indicator is interesting as most students’ self-assessment was either very good or very bad, which however is not too informative considering the low amount of mistakes.

Figures 30 and 31 on the next page show the analyze mode of two students. Student 1 produced most learner data with 329 tested words and only 13 mistakes. Thus, her overall rank is the highest rank possible. Her self-assessment scale indicates a very good self-assessment, which is reflected by her honest ratings with mistaken words, as she mostly rated wrongly answered questions as rather difficult. Her topic statistics are insignificant and mistakes are equally distributed over various semantic fields. Her performance clearly reflects the teacher’s impression of her as the best student of the class.

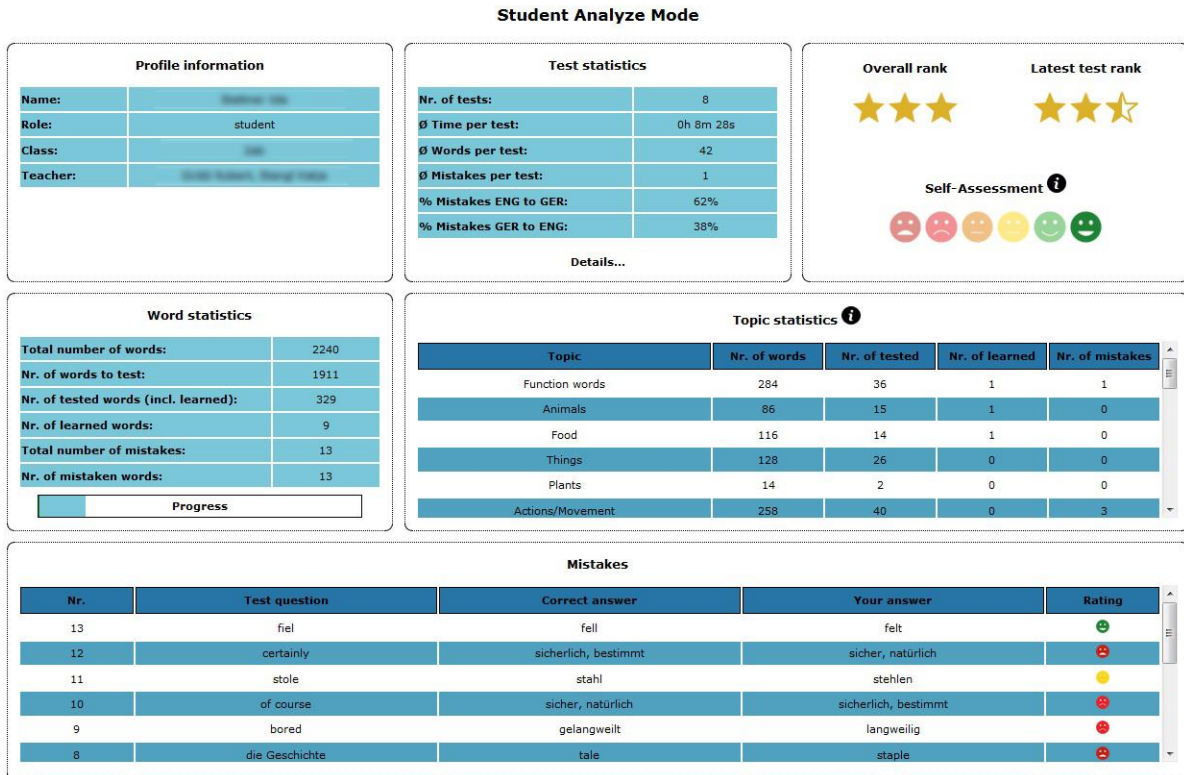


Figure 30: Analyze mode of student 1 after the evaluation period

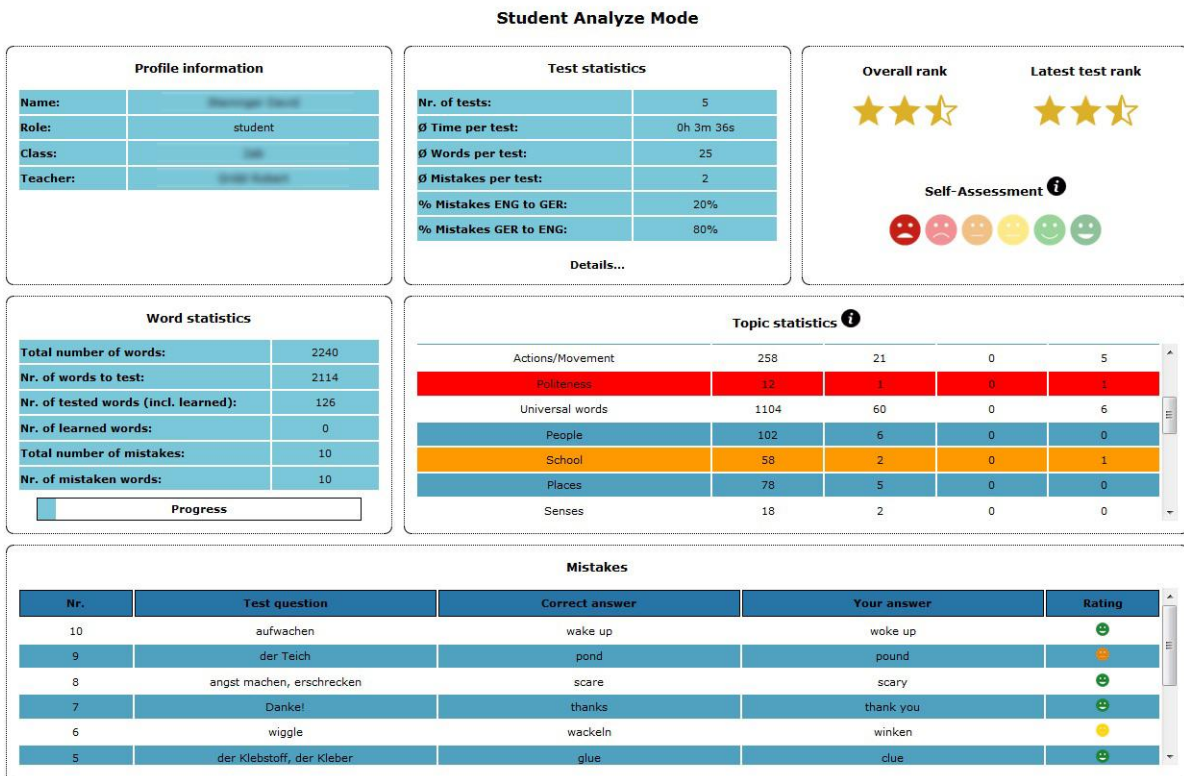


Figure 31: Analyze mode of student 2 after the evaluation period

Student 2 was less productive than student 1 with 126 tested words, but he made 10 mistakes which are almost as many as student 1 made. Thus, student 2's rank is lower than student 1's. Significant is student 2's self-assessment scale which indicates a poor self-assessment. This is also reflected by his ratings of wrongly answered questions. A closer look at the mistakes he made shows that he often rated wrongly answered questions as rather easy. In student 2's topic statistics one line is highlighted in red and another in orange. These lines refer to the semantic fields of politeness and school, but the information that these are two of his weak topics are not meaningful as only a few words were tested from these topics.

Although the results were not sufficient enough to draw meaningful conclusions about students' learning processes or to identify problem areas of individual students, the evaluation still revealed some flaws of the software, which will be further discussed in chapter 9.

8.4 Feedback

Student feedback was obtained written by a feedback sheet and orally when collecting the feedback sheets. In order to gain useful feedback of eleven to twelve years old adolescents it was important to keep the questions simple and the whole feedback sheet short. The sheet included three yes/no questions, three sentence beginnings to express what they liked, what they did not like and what they would wish the program would do, and a last section to note down errors they encountered.

The first yes/no questions intended to discover whether using the program in general was perceived difficult, which was answered with "no" by all learners. The second and third questions asked whether it was difficult to understand the test and analyze modes. Both questions were also uniformly answered with "no" by the students. The oral feedback was basically similar to the written feedback, but implicitly the wish for explanations to some of the analysis tools became apparent.

The students' comments on the sentence beginnings revealed that the software was easy to use and some described it as a useful program for practising vocabulary. However, most of the students felt that they had to click too many times to get to the next test question. The only wish that was expressed was to add determiners/articles to English nouns because their German equivalents also have articles attached.

9 Discussion and outlook

This chapter discusses several aspects addressed in the former chapters of this thesis and by doing so provides an outlook for further development. Thus, this chapter starts with a discussion of the evaluation process, subsequently discusses several features of the software and provides ideas for improvement and extension of the prototype. Then this chapter discusses the basic system and elaborates on improvement of performance and modular expandability. This part also summarizes some ideas for improvement of the layout which emerged during a final meeting with my colleague Gerald Geier, who supported me in the creation of the layout throughout this project. Furthermore, necessary extensions for the use of the VOCABTrainer in higher grades are addressed. As mentioned in preceding chapters the topic statistics are seen as most significant in terms of learning analytics and I have a clear vision for the extension of this analysis tool, which will be explained at the end of this chapter.

In general, the introduction of new software is considered a critical and tedious process as it often takes quite some time for new applications to be fully accepted by users. Especially, software which depends on user generated data is critical as the purpose and meaningfulness of such applications often only becomes apparent when a certain amount of data has been gathered. This applies for the VOCABTrainer and became explicit after the evaluation of the prototype. As already mentioned in the preceding chapter the evaluation did not proceed as intended and the gained learner data was less than expected. However, even if the evaluation proceeded as expected the amount of learner produced data would still be not enough and probably not allow meaningful conclusions about learning processes. In order to draw significant conclusions a much greater amount of data and the refinement of some analysis tools is necessary, as all analysis tools were implemented on the basis of assumptions derived from theory and not on the basis of learner produced data. Thus, it was difficult to adjust certain parameters of analysis tools which heavily depend on learner data, as for example the colored highlighting of the topic statistics or the algorithm to calculate the self-assessment value. In order to optimize certain parameters of the prototype a more extensive evaluation of the software with a group of students that is even closer to the intended target group is necessary. The software in its current version lacks one important feature to evaluate it in a real school setting with students and teachers of a first grade in secondary school during a schoolyear, which will be explained in the next paragraph. For this reason the ideal test users

would be a group of students, who just finished first grade of secondary school and the best time to conduct a more extensive evaluation would be during their summer holidays.

In order to use the software supplementary to teaching in school it needs to be extended by a feature for teachers, which enables them to select the tested vocabulary for their students. Such feature was not implemented in the prototype developed in accordance with this thesis, because it was rejected in the conception stage and would have gone beyond the scope of this thesis. However, to use the software supplementary to school such feature would be absolutely necessary as it is the only possible way by which teachers could ensure that students are only tested on words which they have already learnt. Moreover, such feature could also be useful for defining the order by which vocabulary of independent users is tested. Thus, the vocabulary lists created by teachers could be analyzed and merged to define the arrangement of vocabulary for independent learners. This could replace the currently used algorithm of the prototype, which chooses vocabulary based on students' ratings of single words to compile homogenous test in terms of difficulty.

Another feature provided for teachers which could be easily improved and result in more comfortable use of the software for teachers and administrators, is the form for topic/word suggestions. In the current version of the software this feature was implemented in the easiest possible way and provides only a text box to enter suggestions, which are sent to administrators via e-mail. In its current form this feature is not really comfortable for teachers, nor is it comfortable for administrators. Especially for administrators this function means quite some work, as they have to copy every single suggestion separately and insert it into the database using the manage vocabulary mode. The word suggestion form could be improved by providing single input fields including a drop down for choosing word classes. Furthermore, this functionality could be improved by implementing a simple messaging system, which forwards suggestions to administrators directly in the software and allows the quick adding of topics and words. Improving this functionality could tremendously increase the usability of the software for administrators and teachers.

Regarding the currently implemented analysis tools it is not yet clear whether all tools provide the intended meaningful information. The tools need to be assessed according to their usefulness when enough learner produced data is available. Not so useful tools could then be replaced by other features as the collected data bears great opportunities for further, perhaps more useful, tools. Thus, I could imagine the development of a tool, which illustrates the most

difficult words for students according to their ratings. Such feature could be easily implemented in a short amount of time and could reveal useful information. Another tool which I actually intended to implement is a pie chart which illustrates the distribution of errors by word classes. This feature, however, was discarded with the introduction of the requirement that no diagrams, charts or other complex visual representations should be used as means of analysis. This feature could help teachers to put the focus of their vocabulary teaching on word classes, which seem especially difficult for most of their students. Altogether, the opportunities for the implementation of new analysis tools and the adaption of already existing features are enormous.

On the whole, the basic system used for the implementation of the prototype is very comprehensible and its simple structure allows easy extension. Also the existing components can easily be extended by adding methods to controllers and views. The performance of the software was tested together with the administrator of the TU Graz webserver and was basically classified as good. However, there is still space for improvement especially regarding database access. Comparisons of PHP and MySQL reveal that using more complex MySQL queries, instead of simple queries which demand further data processing by PHP, can boost the performance of software.²³ As the data volume is expected to increase quickly when the software is released for public access, changes from simple to more complex MySQL queries in model classes could increase performance and at some point in future such changes might be necessary. The reason for using simple MySQL queries in this version was comprehensibility of source code, which implies easier extensibility of the software. In general it might be a good idea to embed the existing source code into a larger framework, as for example Zend Framework 2, in future versions of the software. The migration into a PHP framework could increase modularity, extensibility, reusability of code and also performance of the software, because well-known frameworks especially aim at these aspects.

With regard to the layout of the software the following aspects might be considered for improvement in terms of design and usability. Concerning the design some minor changes of colours were suggested which could make the software more attractive for young learners and

²³ <http://www.onextrapixel.com/2010/06/23/mysql-has-functions-part-5-php-vs-mysql-performance/> [accessed on 07.05.2015]

at the same time improve the readability of certain information in tables. The suggested colours are mainly a matter of taste and I will not state the exact colour codes in this thesis. However, it should be noted that the design was kept very simple in the prototype, and therefore, the improvement of visual features in later versions is advisable to approach a greater range of users.

Regarding usability it was suggested to replace several symbols, which were considered ambiguous, by more distinct buttons. Especially, the red symbol used to reset input fields (see figures 21 and 23) was considered ambiguous and should be replaced by buttons. Furthermore, the explanations of input fields and drop-down lists were pointed out to be inconsistent and should be completed so that all of these fields provide explanations. The most crucial aspect pointed out for the improvement of usability was the missing error message at the login. Thus, users are not provided with an error message when login data was wrong, which could lead to confusion about what went wrong in the login process. Given the architectural design pattern of the software the implementation of these error messages is simple and can be realized in a minimum of time. However, before it makes sense to implement this error message an error on the side of the usermanager, which appears with newly created teacher accounts, has to be fixed by the developers responsible for the usermanager. Interestingly, this error only applies to the VOCABTrainer and no other learning applications which also use the usermanager and is probably connected to the application id of the vocabulary trainer. The responsible developer has already been informed about this error, but the error has not been fixed yet, which is the reason why the error message with the login was not implemented in this version of the software.

Altogether, changes in the layout of the software are simple and can be completed in a short amount of time. This is possible because of the architectural design pattern of the software. The suggested aspects for improvement of layout and design are all minor changes and should be realized with the next version of the software.

Chapter 3 of this thesis was concerned with language acquisition and clearly pointed out that the mere testing of word recognition (meaning and form) is only useful in early stages of language acquisition. This implies that for further more extensive and meaningful analysis of learning processes the development of various different tasks, including the input of words by students, is essential. Thus, a future version of the VOCABTrainer, which is intended for a greater target group including students of higher grades, has to test further dimensions of

lexical competence and performance.

Finally, I want to discuss the analysis tool topic statistics which is intended to highlight a student's problem areas in terms of semantic fields. From my point of view the actual representation of these statistics in a table is not very convincing and informative. Therefore, I want to use the rest of this chapter to introduce a vision I have had in mind since I started my research on this thesis.

In a first concept of the prototype I referred to a tool called DISCO²⁴ (extracting DIstributionally related words using CO-occurrences) – compute semantic similarity between words, which provides a Java API to retrieve semantically similar words to an input word, retrieve a value of semantic similarity between two words and retrieve collocations for words. By the help of this tool it should be possible to represent tested words, erroneous words and learned words in the form of a word web, which resembles the mental lexicon explained in chapter 3. Representing learner produced data in a similar way as the mental lexicon is believed to look like could reveal utterly valuable information about learning processes underlying language acquisition. Furthermore, the analysis and discovery of common error patterns within such word webs could help to better understand language learning and errors which occur in this process. Ultimately, this information could be used for intervention and improvement of language learning.

The representation of learner produced data in the form of a word web could look similar to figure 4 on page 17, and additionally, provide a zoom function to reduce the displayed information to bubbles containing labels for semantic fields. Mistakes and learned words could be highlighted by the use of colors to emphasize strong and weak areas of a learner. During my further research for this thesis I discovered another tool called WordNet²⁵ – a lexical database for English, which is a project of Princeton University that started in the mid 1980s and is still further developed. This tool is similar to DISCO, but seems to be even more powerful.

²⁴ <http://www.linguatools.de/disco/> [accessed on 07.05.2015]

²⁵ <http://wordnet.princeton.edu/wordnet/> [accessed on 07.05.2015]

An application which is based on WordNet is Visuwords^{TM26} online graphical dictionary, which provides a dictionary in the form of a word web. VisuwordsTM realization of a dictionary in the form of a neural web is close to my idea of visualization of learner produced data, but is way more complex because of numerous different relations to aspects such as connotations and derivations of words. VisuwordsTM is mentioned at this point to demonstrate the potential of WordNet for the further development of the VOCABTrainer.

²⁶ <http://www.visuwords.com/> [accessed on 07.05.2015]

10 Conclusion

To conclude, the aim of this thesis was to develop a prototype of a vocabulary trainer for beginners of English with focus on learning analytics. The first part of this thesis briefly summarizes theory on learning analytics to define the aim and purpose of this thesis in greater detail. The process of learning analytics is explained and core aspects for the development of software in this research area are pointed out. The teacher centered process of learning analytics is explained based on the learning analytics cycle.

The next chapter provides a theoretical basis for the development of the software prototype by a summary and discussion of theory on vocabulary acquisition. To relate the thesis to a school context the Austrian curriculum is explored and findings on vocabulary learning are pointed out. The curriculum is mainly based on the communicative approach to language teaching which is a rather grammar centered approach that highlights the importance of language use in a real context. However, for the development of a vocabulary trainer the lexical approach to language teaching, which sees lexis as the basis of language, provides more useful information. The definition of the mental lexicon according to the lexical approach is solely based on empirical evidence and has not been underpinned by neurological evidence so far. Nevertheless, the theory on the mental lexicon is very plausible, is accepted by many renowned scientists and is very likely to be true. In literature the concept of the mental lexicon is described as a complex web of words and meanings with further relations to forms. Language acquisition and more specifically vocabulary acquisition are complex topics, which leave many questions unanswered. Language in general is a complex field of research, which is highlighted by the fact that language is constantly changing.

A review of available vocabulary trainers and learning software, which provide means for analysis, reveals that the choice of programs with focus on learning analytics is very limited. Furthermore, the evaluation of four vocabulary trainer applications, which have components for analysis, shows that the aim of these applications is different to the VOCABTrainer as they either tend to provide full language courses or aim at systematic memorization of vocabulary. The analysis tools of all evaluated applications seem insufficient to draw meaningful conclusions about learning processes.

A detailed description of the software concept and technical implementation provide insights

into the structure and functionality of the prototype. Extensibility, which includes comprehensibility and simplicity, is of main importance for the implementation of the software, which is given through a modular architecture of the prototype following a model view controller software architectural pattern. In terms of usability I tried to keep the prototype as simple as possible for all user groups. The software is intended to be used supplementary to English lessons in school, but at the same time it should not bear additional expenditure of time for teachers. For this reason, a feature that allows teachers to choose vocabulary for their students to be tested was rejected in the conception stage. However, in order to be usable complementary to school I see such feature as essential. Additionally, it should be noticed that choosing vocabulary for their students is a rather short task for teachers, in comparison to checking written vocabulary tests, conducting oral revisions and taking endless notes of learners' vocabulary knowledge, which would become unnecessary tasks when using the software. However, before the software is ready for usage in a school context a longer evaluation should be carried out and analysis tools should be refined, tweaked and perhaps even replaced.

In order to assess the usefulness of analysis tools provided in the current version of the prototype it is of great importance to collect a greater amount of data. The ideal target group for the gathering of data with the current version would be students, who just finished the first grade of secondary school and use the software to revise the learnt vocabulary. Altogether, I consider the developed prototype as a successful attempt towards a vocabulary trainer with focus on learning analytics, as the analysis tools provide a general overview of students' vocabulary competence and allow teachers to quickly identify weaker students. The rating of words by learners, which is central to the concept of the software, is in the current version used for the compilation of homogenous tests, and to gain impressions of learners' self-assessment. With the increasing amount of learner generated data the ratings of single words could also be used for clear statements about the difficulty of specific words for a majority of learners. In combination with the feature that provides the top ten mistakes made by students of a teacher such analysis tool could help teachers to purposefully use difficult words in their teaching. Furthermore, the test mode, which provides immediate feedback on the validity of answers and points out the correct word if an answer was wrong, might implicitly support students in the acquisition of new words.

Bibliography

- 1st International Conference on Learning Analytics & Knowledge (2011). *Call for Papers – 1st International Conference on Learning Analytics and Knowledge 2011*. [Online]. <https://tekri.athabascau.ca/analytics/call-papers> [accessed on 18.04.2015].
- Aitchison, Jean (2012). *Words in the mind: An introduction to the mental lexicon*. 4th ed. Singapore: Ho Printing Singapore Pte Ltd.
- Baker, R., Duval, E. , Stamper, J. , Wiley, D. and Buckingham Shum, S. (2012). *Educational data mining meets learning analytics*. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. LAK '12. Vancouver, British Columbia, Canada: ACM, p. 20 – 20.
- BMUKK (2015). *Österreichischer Lehrplan für Lebende Fremdsprachen*. [Online] https://www.bmbf.gv.at/schulen/unterricht/lp/ahs8_782.pdf [accessed on 18.04.2015].
- Campell, J. P., and Oblinger, D. G. (2007). *Academic Analytics*. In: EDUCAUSE Quarterly October 2007.
- Clow, Doug (2012). *The learning analytics cycle: closing the loop effectively*. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. LAK '12. Vancouver, British Columbia, Canada: ACM, p. 134 – 138.
- Coady, J. and Huckin, T. (1997). *Second Language Vocabulary Acquisition*. Cambridge: Cambridge University Press.
- Eggert, Ralf (2013). *Zend Framework 2: Das Praxisbuch*. Bonn: Galileo Press.
- Gaynor, R. and Hübner, L. (2007). *Your Turn: Workbook 1*. Vienna: Langenscheidt ELT.
- Gerngross, G., Puchta, H., Holzmann, C., Stranks, J. and Lewis-Jones, P. (2007). *More!: Student's Book 1*. 2nd ed. (2009). Helbling Languages.
- Gerngross, G., Puchta, H., Holzmann, C., Stranks, J. and Lewis-Jones, P. (2007). *More!: Workbook 1*. 2nd ed. (2009). Helbling Languages.

- Gerngross, G., Puchta, H., Davis, R. L. and Holzmann, C. (2005). *The new you&me Textbook 1*. Munich: Langenscheidt ELT.
- Gerngross, G., Puchta, H., Davis, R. L. and Holzmann, C. (2005). *The new you&me Workbook 1*. Munich: Langenscheidt ELT.
- Harmer, Jeremy (2007). *Your Turn: Textbook 1*. Vienna: Langenscheidt ELT.
- International Educational Data Mining Society (2015). *Home – International Educational Data Mining Society*. [Online] <http://www.educationaldatamining.org/> [accessed on 18.04.2015].
- jQuery Foundation (2015). *License – jQuery Foundation*. [Online] <https://jquery.org/license/> [accessed on 01.05.2015].
- Landesschulrat für Steiermark (2013). *Abgrenzung zwischen (schriftlichen) Arbeitsleistungen und Tests; Erläuterungen und Hinweise*. [Online] [http://www.lsr-stmk.gv.at/cms/dokumente/10074385_430538/1c5cd67e/Erledigung\[1\].pdf](http://www.lsr-stmk.gv.at/cms/dokumente/10074385_430538/1c5cd67e/Erledigung[1].pdf) [accessed on 28.04.2015].
- Moudraia, Olga (2001). *Lexical Approach to Second Language Teaching*. Washington DC: Center for Applied Linguistics.
- Newby, David (2011). *Applied Linguistics for Language Teachers: Competence and Performance*. Graz: Karl-Franzens University.
- Neuhold, Benedikt (2013). *Adaptives Informationssystem zur Erlernung mehrstelliger Addition und Subtraktion*. Graz: Graz University of Technology.
- Oracle Corporation (2015a). *MySQL :: FOSS Licence Exception* [Online] <http://www.mysql.com/about/legal/licensing/foss-exception/> [accessed on 01.05.2015]
- Oracle Corporation (2015b). *MySQL :: Why MySQL?* [Online] <http://www.mysql.com/why-mysql/> [accessed on 01.05.2015].
- Oxford Dictionaries (2015). *Oxford Dictionaries: Language matters*. [Online] <http://www.oxforddictionaries.com/> [accessed on 29.04.2015]

- PHP Group (2015a). *PHP: Licence Information*. [Online] <http://www.php.net/license/> [accessed on 01.05.2015].
- PHP Group (2015b). *PHP: PHP Manual*. [Online] <http://php.net/manual/en/> [accessed on 01.05.2015].
- Piety, Philip J. (2013). *Assessing the Educational Data Movement*. New York: Teachers College Press.
- Reimers, S. and Thies, G. (2012). *PHP 5.4 und MySQL 5.5: Das umfassende Handbuch*. Bonn: Galileo Press.
- Siemens, G. and Baker, R. (2012). *Learning analytics and educational data mining: towards communication and collaboration*. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. LAK '12. Vancouver, British Columbia, Canada: ACM, p. 252 – 254.
- Siemens, George (2010). elearnspace - What are Learning Analytics? [online] <http://www.elearnspace.org/blog/2010/08/25/what-are-learning-analytics/> [accessed on 18.04.2015].
- Siemens, G. and Long, P. (2011). *Penetrating the Fog: Analytics in Learning and Education*. In: EDUCAUSE Review 46.5, p. 30 – 32.
- Thornbury, Scott (2012). *How to teach vocabulary*. Essex: Pearson Education Limited.
- Yule, George (1985). *The Study of Language*. 3rd ed. (2006). Cambridge: Cambridge University Press.

List of figures

Figure 1: The process of learning analytics (Siemens 2010)	6
Figure 2: Application of learning analytics (Siemens 2010)	7
Figure 3: Learning analytics cycle (Clow 2012)	9
Figure 4: A semantic word-web – clusters of coordinates (Aitchison 2012).	17
Figure 5: Analysis mode of Babel	20
Figure 6: Analysis mode of languagecourse.net	21
Figure 7: Analysis mode of Lingoversity	23
Figure 8: The OEC – word usage (Oxford Dictionaries 2015)	25
Figure 9: Use case diagram	33
Figure 10: State diagram	38
Figure 11: MVC architecture	40
Figure 12: Entity relationship (ER) diagram	41
Figure 13: Complete classdiagram	43
Figure 14: Filesystem	46
Figure 15: Classdiagram basic system	47
Figure 16: Classdiagram login	48
Figure 17: Classdiagram manage vocabulary	50
Figure 18: Classdiagram browse vocabulary	51
Figure 19: Classdiagram test mode	52
Figure 20: Classdiagram analyze mode	57

Figure 21: Basic layout of the prototype62

Figure 22: First screen for logged-in users (student)62

Figure 23: Browse vocabulary mode.....63

Figure 24: Manage vocabulary mode64

Figure 25: Test mode – correct answer65

Figure 26: Test mode – wrong answer66

Figure 27: Analyze mode (teacher) – list of classes.....67

Figure 28: Analyze mode (teacher) – class details.....67

Figure 29: Analyze mode (student)68

Figure 30: Analyze mode of student 1 after the evaluation period73

Figure 31: Analyze mode of student 2 after the evaluation period73

List of source code

Source code 1: Returned multidimensional array from getWordsToTest()53

Source code 2: Compilation of test questions (1/2)53

Source code 3: Compilation of test questions (2/2)55

Source code 4: Calculation of rank and self-assessment indicators.....59

Appendix

A. Feedback sheet

Feedback

Bitte beantworte folgende 3 Fragen mit JA oder NEIN:

1. Hattest du Schwierigkeiten dich im Programm zu Recht zu finden? _____
2. Hattest du Schwierigkeiten den Test Modus zu verstehen? _____
3. Hattest du Schwierigkeiten den Analyse Modus zu verstehen? _____

Hast du positive/negative Rückmeldungen?

Ich fand gut, dass...

Ich fand schlecht, dass...

Ich würde mir wünschen, dass...

Hast du einen Fehler entdeckt? Wenn ja, bitte kurze Beschreibung hier, damit ich den Fehler beheben kann 😊

B. User Accounts

Zugangsdaten zum Vokabeltrainer

Website: <http://schule.learninglab.tugraz.at/vocabtr/>

Name: Forename Surname

Username: forename_surname

Password: password