Burazer Marko, BSc

Implementation of Interactive Learning Objects for German Language Acquisition in Primary School based on Learning Analytics Measurements

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Priv.-Doz. Dipl.-Ing. Dr.techn. Ebner Martin

Institute of Interactive Systems and Data Science
Head: Priv.-Doz. Dipl.-Ing. Dr.techn. Ebner Martin

Co-Supervisor: Dipl.-Ing. Ebner Markus

Graz, September 2019

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

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

Affidavit

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 indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Danksagung

Als erstes möchte ich mich bei meinen Betreuern Herrn Priv.-Doz. Dipl.-Ing. Dr. techn. Martin Ebner sowie Dipl.-Ing. Markus Ebner bedanken, ohne sie hätte diese Arbeit nicht realisiert werden können. Aber auch HS-Prof. Mag. Dr. Konstanze Edtstadler und Prof. Mag. Elisabeth Herunter müssen erwähnt werden, nicht nur durch ihre Unterstützung bei der Umsetzung der einzelnen Übungen sondern auch bei der Evaluierung selbst.

Ein besonderer Dank gilt natürlich meinen Eltern, die immer an mich geglaubt haben sowie meiner Schwester, die mir mit ihren perfekten Englischkenntnissen wichtige Tipps und Tricks für das Schreiben geben konnte.

Die aller besten Freunde der Welt müssen natürlich auch erwähnt werden, wobei manche in gewissen Momenten wie eine Art Lebenshilfe sein können.

Meinen Chef Dipl.-Ing. Simon Jimenez, MA darf ich nicht vergessen, der mir meinen Arbeitsplatz in der Firma zu Verfügung gestellt hat um rund um die Uhr an meiner Masterarbeit arbeiten zu können.

Abstract

Obviously, reading and writing are vital qualities nowadays, likely more so than ever before. Whether that be in school, work or everyday life, it is a skill set that is omnipresent. This is also evident by the countless contributions that are created and published on various online platforms such as Facebook, Twitter, Youtube or WhatsApp. In order to avoid being misunderstood, it is crucial to have the ability to express one's written thoughts in a structured and error-free manner. The platform IDeRBlog tries to reach children at an early age with the goal to simplify their German spelling learning process. Blog entries can be created by children and corrected by teachers before they are finally published. Any mistakes made by the children are stored in a database. On the basis of these errors, exercises are then being recommended to the children, so that they can improve their spelling. This master's thesis will present these exercises, which should enable the practice of writing, reading and listening carefully. The evaluation of the exercises as well as the results will be presented in the last chapter.

Kurzfassung

Schreiben und Lesen ist heutzutage noch immer sehr wichtig, vielleicht sogar wichtiger als jemals zuvor. Nicht nur in der Schule sondern auch im Alltag und im Beruf. Täglich werden unzählige Beiträge im Internet auf den verschiedensten Plattformen wie Facebook, Twitter, YouTube oder WhatsApp erstellt und veröffentlicht. Um nicht missverstanden zu werden, wäre es daher wichtig seine Gedanken auch in schriftlicher Form strukturiert und fehlerfrei von sich geben zu können. Die Plattform IDeRBlog setzt dabei sehr früh bei Kindern an um ihnen das Erlernen der deutschen Rechtschreibung zu erleichtern. Blogeinträge können auf der Plattform von Kindern verfasst und von Lehrern korrigiert werden bevor diese schlussendlich veröffentlicht werden. Fehler, die von den Kindern gemacht wurden werden dabei in einer Datenbank gespeichert. Anhand dieser Fehler sollen den Kindern anschließend Übungen vorgeschlagen werden damit sie an ihrer Rechtschreibung arbeiten können. In dieser Masterarbeit werden genau diese Übungen vorgestellt, welche das Üben von Schreiben, Lesen als auch genaue Zuhören ermöglichen sollen. Auch die Evaluierung der Übungen sowie die resultierenden Ergebnisse werden vorgestellt und im letzten Kapitel zusammenfassend erläutert.

Contents

Abstract v

1 Introduction 1

2 State of the Art 3
- 2.1 Learning Objects . . . 3
  - 2.1.1 Terminology . . . 5
- 2.2 Learning Analytics . . . 6
  - 2.2.1 Five Steps of Analytics . . . 7
  - 2.2.2 Learning Analytics Cycle . . . 8
  - 2.2.3 Learning Analytics Life Cycle . . . 9
  - 2.2.4 Limitations . . . 10
- 2.3 Gamification . . . 11
  - 2.3.1 Implementing Gamification . . . 14
- 2.4 E-Learning Tools . . . 16

3 About the IDeRBlog Platform 19
- 3.1 Overview . . . 20
- 3.2 Intelligent Dictionary . . . 22

4 Implementation of Interactive Learning Objects 23
- 4.1 Exercises Overview . . . 24
- 4.2 Exercise Einsetzen (*insert*) . . . 26
- 4.3 Exercise Fehlerstellen (*fault locations*) . . . 28
- 4.4 Exercise Merkwörter (*cue words*) . . . 33
  - 4.4.1 Easy . . . 35
  - 4.4.2 Moderate . . . 35
  - 4.4.3 Hard . . . 35

4.5 Exercise Zuordnen (*assign*) 35
4.5.1 Drag & Drop 36
4.5.2 Writing 38
4.6 Exercise Glücksrad (*wheel of fortune*) 41

5 Implementation of Gamification 45

6 Technical Implementation 49
6.1 Technologies and Tools 49
6.1.1 Web Application 49
6.1.2 HTML 50
6.1.3 CSS 51
6.1.4 Javascript 53
6.1.5 Text-To-Speech (TTS) 54
6.1.6 PHP 54
6.2 Development Tool (IDE) 55
6.3 API and Documentation 56
6.3.1 API 56
6.3.2 Documentation with Apiary 57
6.3.3 IDeRBlog Exercises API 57
6.4 Code Structure 60

7 Evaluation and Results 63
7.1 Evaluation Setup and Procedure 64
7.2 Questionnaire Evaluation and Data Analysis 67
7.2.1 Einsetzen (*insert*) 67
7.2.2 Fehlerstellen (*fault locations*) 69
7.2.3 Merkwörter (*cue words*) (Easy) 73
7.2.4 Merkwörter (*cue words*) (Hard) 76
7.2.5 Zuordnen (*assign*) (Drag&Drop) 78
7.2.6 Summary 80

8 Summary and Future Work 81
8.1 Summary 81
8.2 Future Work 83

Appendix 85

Bibliography 103

List of Figures

2.1 Metadata elements . . . 5

2.2 The 3 Vs of big data (Russom et al., 2011) . . . 6

2.3 Learning Analytics Cylce (Clow, 2012) . . . 9

2.4 Learning Analytics Life Cycle proposed by Khalil and Martin Ebner, 2015 . . . 10

2.5 Learning Analytics constraints (Khalil and Martin Ebner, 2015) 11

2.6 Gamification Science (DNNCorp, 2019) . . . 12

2.7 The process for applying gamification in education (Huang and Soman, 2013) . . . 14

3.1 The homepage of the IDeRBlog website and its design . . . 19

3.2 Workflow of the IDeRBlog platform (Edtstadler, Markus Ebner, and Martin Ebner, 2015) . . . 21

4.1 Starting page of every exercise (example from Einsetzen (*insert*) exercise) . . . 25

4.2 Last page after every exercise is finished (example from Einsetzen (*insert*) exercise) . . . 25

4.3 Einsetzen (*insert*) exercise with cloze text example . . . 26

4.4 Example for a tooltip in the Einsetzen (*insert*) exercise . . . 27

4.5 Example for displaying user errors in the Einsetzen (*insert*) exercise . . . 27

4.6 Example for managing the Einsetzen (*insert*) exercise . . . 28

4.7 Mockups for the Einsetzen (*insert*) exercise . . . 29

4.8 Example the Fehlerstellen (*fault locations*) exercise with one fault location found . . . 29

4.9 Example when user has to justify decision in the Fehlerstellen (*fault locations*) exercise . . . 30

4.10 Example when user selected a wrong justification in the Fehlerstellen (*fault locations*) exercise 31
4.11 Example for managing the Fehlerstellen (*fault locations*) exercise 31
4.12 Mockups for the Fehlerstellen (*fault locations*) exercise 32
4.13 Example for marking letters/grapheme of a word in the Merkwörter (*cue words*) exercise 33
4.14 Example for feedback if the user misspelled a word in the Merkwörter (*cue words*) exercise 34
4.15 Settings example for the 'easy mode' of an Merkwörter (*cue words*) exercise. One word, letters clickable, showing feedback to the user if word was written wrong and no time limitations 36
4.16 Mockups for the Merkwörter (*cue words*) exercise 37
4.17 Example of the Drag&Drop variant in the Zuordnen (*assign*) exercise 37
4.18 Example of the Drag&Drop variant with highlighted errors in the Zuordnen (*assign*) exercise 38
4.19 Example of the Write variant in the Zuordnen (*assign*) exercise 39
4.20 Example of the Write variant in the Zuordnen (*assign*) exercise 39
4.21 Mockups for the Zuordnen (*assign*) exercise 40
4.22 Example of the Glücksrad (*wheel of fortune*) exercise 41
4.23 Example of the Glücksrad (*wheel of fortune*) exercise with errors shown 42
4.24 Mockups for the Glücksrad (*wheel of fortune*) exercise 43

5.1 Homepage showing how many points the user has collected. 45
5.2 Settings page where points can be set for each exercise. 46
5.3 Homepage with locked and unlocked exercise. 47

7.1 A computer of the classroom, used for the usability test . . . 63
7.2 Smileys used for ratings (Agarwal, 2019) 65
7.3 Usability Questionnaire 66

List of Tables

2.1 Learning Object terminology (McGreal, 2004a) 5
2.2 Game Mechanics Examples (self-elements vs. social-elements) (Huang and Soman, 2013) 16

6.1 Overview of API calls for 'Einsetzen' 58
6.2 Overview of API calls for 'Fehlerstellen' 58
6.3 Overview of API calls for 'Merkwörter' 58
6.4 Overview of API calls for 'Zuordnen' 59
6.5 Overview of API calls for 'Glücksrad' 59
6.6 Overview of general API calls 59

7.1 Amount of errors made in each sentence of the Einsetzen (*insert*) exercise 68
7.2 Questionnaire results of the Einsetzen (*insert*) exercise 69
7.3 Questionnaire results of the Einsetzen (*insert*) exercise 70
7.4 Words in Fehlerstellen (*fault locations*) exercise which have been justified wrongly. 72
7.5 Words in Fehlerstellen (*fault locations*) exercise which have been false clicked on. 72
7.6 Questionnaire results of the Fehlerstellen (*fault locations*) exercise 73
7.7 Differnet types of spelling mistakes in Merkwörter (*cue words*) Easy exericse 75
7.8 Questionnaire results of the Merkwörter (*cue words*) (Easy) exercise 75
7.9 Questionnaire results of the Merkwörter (*cue words*) (Hard) exercise 76
7.10 Words used in the Merkwörter (*cue words*) (Hard) exercise and how often they were misspelled. 77

7.11 Words used in the Merkwörter (*cue words*) (Hard) exercise with wrong user inputs. 77
7.12 Questionnaire results of the Zuordnen (*assign*) (Drag&Drop) exercise 78
7.13 Differnet types of spelling mistakes in Merkwörter (*cue words*) Easy exericse 79
7.14 Overall Summary of Questionnaires 80

Listings

6.1 HTML example code 50
6.2 CSS example code 51
6.3 BEM example code 53
6.4 Text-To-Speech example code 54
6.5 PHP example code 55

1 Introduction

Mastering orthography is more important than ever in today's world. More and more people are communicating online through various applications; Facebook, Twitter, YouTube, WhatsApp, and many more. It is necessary to learn how to spell since nobody wants to be publicly criticized or even laughed at for spelling mistakes (Aspalter, Edtstadler, and Martich, 2017). According to Lehky, 2016, correct spelling is important and should not to be underestimated in the workforce nowadays, considering how spelling errors can be the cause for not being hired for a position.

The digital age allows almost unlimited possibilities for learning more productively: On a PC or on mobile devices, individually or together in a group, at home, on the way to work or at school.

The second chapter of this master's thesis covers the current state of the art. Among other things, it deals with the question of what learning objects are and what can be done with them. Learning analytics is a quite complex but also interesting topic, which offers many possibilities to not only increase one's own productivity but also that of others. Although gamification sounds like a playful triviality, when it comes to e-Learning, there are several aspects to consider when implementing it. These are described in detail in the fifth chapter.

IDeRBlog is a platform where young children have the opportunity to create and publish their own texts. Afterwards, any mistakes that were made will be corrected. On this basis, the children are thus offered further exercises in order to improve their spelling. The practical part of this master's thesis will refer to exactly these aforementioned exercises, or so-called learning objects. Chapter four describes all types of exercises in detail; how they are created to how they work in practice. The chapter further explains how a teacher can create different units of these exercises for various topics of the German orthography.

The fifth chapter will underline how gamification parts have been implemented and how these can be used so the children have fun learning

What type of technologies and tools have been used to create the interactive learning objects, is covered in the sixth chapter. What is also discussed in this section is the documentation on how external applications, such as smartphones or tablets, can interact with the provided exercises. Further, it offers a small description on how data can be accessed for analytics measurements

The IDeRBlog platform offers numerous exercises for children regarding German language acquisition. However, the learning process of a user cannot be measured with scientific accuracy. In the evaluation chapter of this thesis, the main goal therefore is to explore how the gathered user data can be used to improve the learning process. More precisely, the aim is to highlight how children react to different forms of exercises and how their willingness to study outside of the class room can be increased.

2 State of the Art

The first section will provide an overview of learning objects and explain what they are. Learning analytics and its different viewpoints and processes will be discussed in the second subchapter. Another topic that has been discussed a lot in recent years, as evident by the fact that it is gaining more and more relevance in education nowadays, is gamification. This topic will be covered in the third section of this chapter. Finally, the last chapter will provide some examples for online e-Learning courses.

2.1 Learning Objects

An important element of this master's thesis is the topic *learning objects*. This topic is not unfamiliar, since it was already mentioned by Wayne Hodgins in the year 1994 (Polsani, 2006). Also, there are several definitions from different persons and organizations regarding what learning objects really are.

The Committee, 2019 describes the learning object metadata standard as:

> "Learning Objects are defined here as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning."

They also mention examples of technology supported learning such as computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, etc. and examples of learning objects include multimedia content, instructional content, learning objectives and so on.

In the book *The Instructional Use of Learning Objects* (Wiley et al., 2002) the author describes how the idea of learning objects is based on the object-oriented development in computer science. Small objects can be easily created and then put together (but also reused several times) to form a much larger and more complex system. Therefore, learning objects should be seen as digital entities which are supplied only over the internet. In contrast to the traditional instructional media (eg. overhead, video tape, etc.) the use of learning objects is also not limited to a specific place at a certain time. It can be used concurrently by several people at the same time all over the world. In addition, an important difference to the existing teaching aids is that it allows people to benefit immediately if there are new versions of learning objects.

According to Hodgins, 2002 learning objects are developed to change the way people learn and will initiate new efficiency for content design, development and delivery. Furthermore, it has the potential to enhance human performance for effective learning.

Generally, there is an understanding for the learning objects community about the basic requirements (Polsani, 2006):

- *Accessibility*: learning objects should be labeled with metadata.
- *Reusability*: learning objects should work in different instructional contexts.
- *Interoperability*: learning objects should be independent of delivery media and knowledge management systems.

McGreal, 2004a outlines the importance of metadata and points out that it is fundamental for addressing learning objects. Metadata is a way of describing other data. For this specific case with metadata it is possible to search and find learning objects much easier and faster. To gain a better understanding an example is provided by McGreal, 2004a too: When a house is an learning object then the house number or street number would be a form of metadata.

Seghroucheni, Achhab, and mohajir, 2014 summarize learning objects as:

> ”A learning object is combination of the content and the metadata, ...”

Anything	**Anything Digital**	**Anything for Learning**	**Specific Learning Environment**
Asset	Content Object	Educational Object	Reusable Learning Object (RLO)
Component	Information Object	Learning Object	Unit of Learning
Learning Resource	Knowledge Object		Unit of Study
	Media Object		
	Raw Media Element		
	Reusable Information Object (RIO)		

Table 2.1: Learning Object terminology (McGreal, 2004a)

Figure 2.1 shows the key elements of metadata (Seghroucheni, Achhab, and mohajir, 2014).

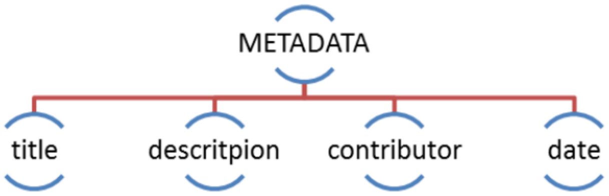


Figure 2.1: Metadata elements

2.1.1 Terminology

According to McGreal, 2004a there are still no general accepted definitions for learning objects, because obviously there are to many opinions about what learning objects should be (McGreal, 2004b). Nonetheless, McGreal distinguishes between four different groups, as shown in Table 2.1.

The entries in the first column describe objects that could be anything, whereas the second one entails any objects that are digital. The third column is about digital learning objects but allegedly with the aim for learning only. Finally, the last column is specifically for learning.

2.2 Learning Analytics

The origins of learning analytics can be traced back to the twentieth century, to the 1970's even; a time before online learning and big data have become an essential part of today's world (Ferguson, 2012). The following three elements are significantly speeding up the evolution of learning analytics:

1. Big Data
2. Online Learning
3. Political Concerns

Big Data

It is hard to find a clear definition what big data exactly is. However, the term *big data* was first used by John Mashey in the 1990s (Kitchin and McArdle, 2016). In 2001 Doug Laney described it according to the following three features: *volume*, *velocity* and *variety*. These features are also known as the three Vs and show clearly that big data is not only about data volume (Russom et al., 2011). Figure 2.2 shows the branching of every single V.

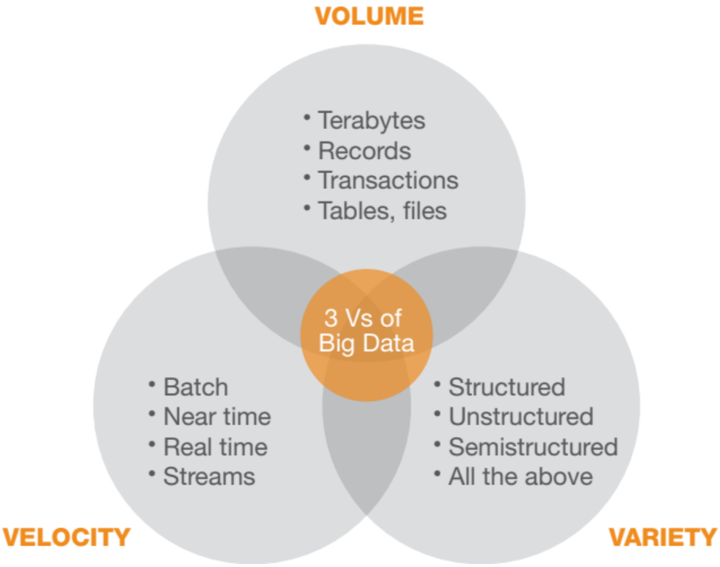


Figure 2.2: The 3 Vs of big data (Russom et al., 2011)

The big question that arises here is:

> ”How can we extract value from these big sets of learning-related data?”

Online Learning

According to Ferguson, 2012 learning online does not only have benefits. One of the biggest obstacles is that online learning isolates students, because they do not have physical proximity to their teachers. Students may get demotivated easily due to technical problems or disorientation in the online world. He (Ferguson, 2012) sees the second feature online learning as an educational challenge:

> ”How can we optimise opportunities for online learning?”

Political Concerns

Every economy in this world needs a well-educated population, therefore many countries are reforming their educational systems (Campbell and Oblinger, 2007). Ferguson, 2012 sees the third driver for speeding up the evolution of learning analytics as a political/economic challenge:

> ”How can we optimise learning and educational results at national or international levels?”

In the year 2011, the first International Conference on Learning Analytics and Kowledge describes this topic as following:

> ”Learning analytics is 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.” (LAK11, 2019)

2.2.1 Five Steps of Analytics

Additionally Campbell and Oblinger, 2007 mention the five steps for making decisions that act as an engine:

1. capture
2. report
3. predict
4. act
5. refine

Capture: Managing data by collecting and organizing it is very important, because decisions are based on the quality and integrity of the given data.

Report: That captured data and included information must be examined before it is transformed into proper statistics. Nowadays dashboards are being used to show data with eye-catching graphics instead of reports with tables full of data.

Predict: Depending on the gathered data, which is analysed using statistics, there are numerous possibilities to make predictions. For example, data can show if students have not prepared enough for an upcoming mathematics exam if the student did not go to lectures regularly. Therefore the prediction would be that the student's risk for failing is high.

Act: Predictions and probabilities must be used to make it possible for an institution to act on. There are many ways to act, one could be to provide information for students so they can monitor their learning progress with some additional instructions for improvement. Those predictions could also created automated steps if the probabilities of dropping out of school are too high, ie. contacting the student by phone call or e-mail or even provide additional help sessions.

Refine: Every project must be improved and updated continuously, either by adding new data or different actions to the refinement.

2.2.2 Learning Analytics Cycle

Clow, 2012 builds on the mentioned ideas above to shape a learning analytics cycle that closes the feedback loop. Figure 2.3 shows that this cycle always starts with **learners**, directly followed by **data**. This data is collected from the learners and can include demographic information, phone call information, login and clickstream or assessment results. The **metrics or analytics** is the processed data from the previous step with some insight to the learning process. This can, for example, entail visualisations, dashboards or a list of 'at risk' students. **Interventions** are closing this cycle. An intervention can be a dashboard for comparing activities with other students or enabling a teacher to make contact with a student who might be at risk of dropping out.

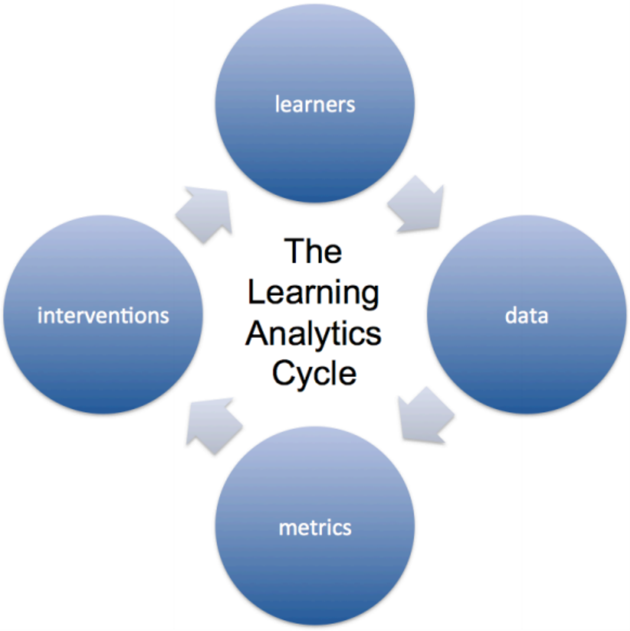


Figure 2.3: Learning Analytics Cylce (Clow, 2012)

Not all four steps are necessary for learning analytics but it makes it very effective.

2.2.3 Learning Analytics Life Cycle

Khalil and Martin Ebner, 2015 present a learning analyics life cycle that considers four parts (as shown in Figure 2.4):

1. **Learning environment**: stakeholders creating data
2. **Big Data**: huge amounts of data
3. **Analytics**: includes various analysis techniques
4. **Act** achieving targets to improve learning environment

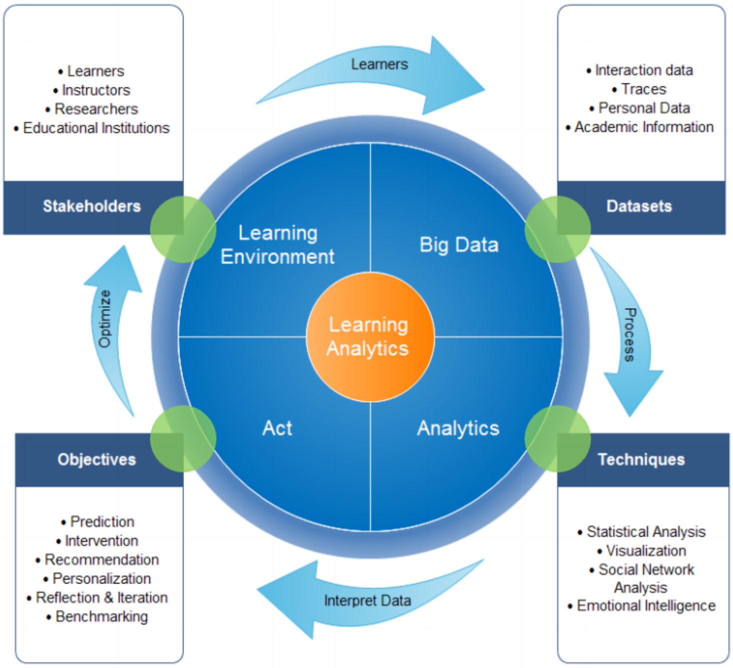


Figure 2.4: Learning Analytics Life Cycle proposed by Khalil and Martin Ebner, 2015

2.2.4 Limitations

There are some points regarding learning analytics that require consideration first. For a successful implementation of learning analytics it must be carefully formulated. Greller and Drachsler, 2012 identify six critical dimensions of learning analytics which are divided into soft and hard issues. Examples for soft issues are competences or ethics, whereas hard issues would be the challenges of data and algorithms.

Thereon Khalil and Martin Ebner, 2015 additionally introduce eight constraints that limit the use of learning analytics processes. These constraints are demonstrated in Figure 2.5

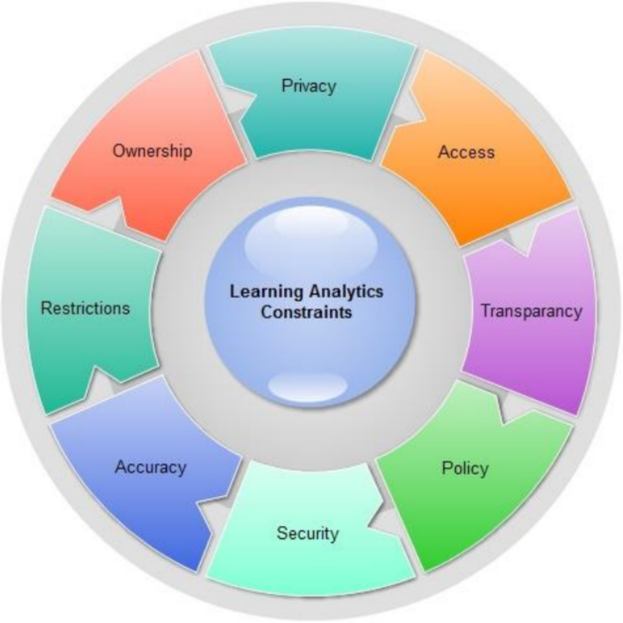


Figure 2.5: Learning Analytics constraints (Khalil and Martin Ebner, 2015)

2.3 Gamification

Historically, the roots of the term gamification can be traced back to the 1890s where marketers sold stamps to retailers (TechnologyAdvice, 2019).

However, the wide-spread distribution of the term was only possible after a generation of gamers had grown up and become active in their working lives. At the same time, it must also be mentioned that digital media, social networks and other Web 2.0 apps helped to spread this term even more. Gamification should not be seen as something where people are simply rewarded with points and badges. Moreover, it is about understanding and influencing human behaviours to animate them for the good of all. Building up on human psychology and behavioural science there are three primary factors which can be identified: **motivation**, **ability level** and **triggers**. These three factors can be seen in Figure 2.6. (Dale, 2014)

In order to animate people to perform well all of the three factors must be present. Two of those are trade-offs: *motivation* and *ability*. If there is little ability, more motivation is needed and if there is little motivation the behaviour steps need to be small.

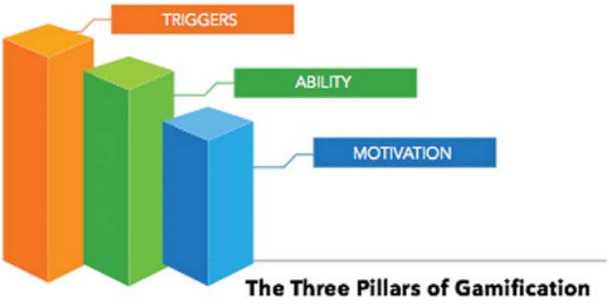


Figure 2.6: Gamification Science (DNNCorp, 2019)

There are also typical components or elements which are mentioned by several sources. According to Steve Dale there are eight components for gamified applications (Dale, 2014):

1. **Points** ... are rewarded for achievements or certain quality behaviors.
2. **Achievements** ... improve user behavior positively.
3. **Levels** ... act as door for new challenges.
4. **Missions** ... create user behaviors to unlock more rewards.
5. **Contests** ... combination of mission to reward the quickest and most effective ones.
6. **Leaderboards** ... allow learning to see how other perform and raises some competition.
7. **Notifications** ... encourage engagement.
8. **Anti-Gaming Mechanics** ... needed for setting limits, rewards should not be infinitely.

Nah et al., 2014 also mentions eight game design elements for education which are slightly different to the previously mentioned ones:

1. **Points**
   ... serve as measurement of success, can also be used as rewards. But also as credits in an academic environment (Kumar and Khurana, 2012).
2. **Levels/Stages**
   ... are useful to get the feeling that one is making progress.
3. **Badges**
   ... are inspiring to work harder for more achievements. (O'Donovan, Gain, and Marais, 2013)
4. **Leaderboards**
   One of the best motivational help. (O'Donovan, Gain, and Marais, 2013)
5. **Prizes and Rewards**
   Character upgrades is an good example for in-game rewards. (Raymer, 2011)
6. **Progress bars**
   The overall progression can be tracked and therefore also provide important feedback. (Raymer, 2011)
7. **Storyline**
   According to Kapp, 2013 it is the most effective but underused method for learning.
8. **Feedback**
   The effectiveness and learner engagement heavily depends on frequent and immediate feedback.

Suzanne Holloway (Holloway, 2018) provides another example for what Gamification is about:

> "Gamification is the use of game design and mechanics to enhance non-game contexts by increasing participation, engagement, loyalty and competition." (Holloway, 2018)

Furthermore, Holloway offers four examples for how education in the classroom can be gamified. He explains that there are tested theories with positive results for that:

1. **Gamifiy grades**
   Lee Sheldon demonstrates in his book that instead of grades, ”experience points” should be introduced and then graded depending on the student’s accomplishment (Sheldon, 2011).
2. **Badges for students**
   Just like in Kindergarten, students should be awarded with badges for completed assignments. To provide an example: Khan Academy (Academy, 2019) uses points and badges for their online courses to motivate students.
3. **Educational video games**
   The integration of educational video games gives students control over the choices they make. Also, instant feedback and rewards give them further motivation to learn more.
4. **Competition**
   TopHat, 2019 includes tournament modules in their platform because professors point out that tournaments excite students to learn and practice more. Moreover, it may provide students with a sense of accomplishment to see their names on the leaderboard.

2.3.1 Implementing Gamification

Even though the implementation of gamification is not very difficult, to ensure its effectiveness, is is relevant to consider the five-step process by Huang and Soman, 2013 (see Figure 2.7).

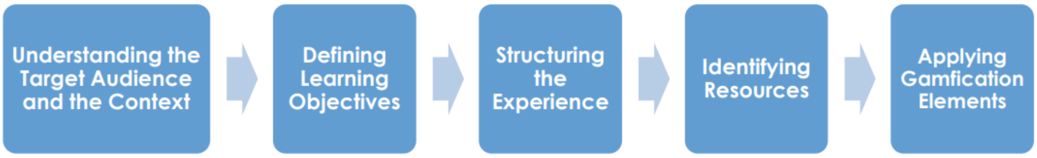

Understanding the Target Audience and the Context
Defining Learning Objectives
Structuring the Experience
Identifying Resources
Applying Gamfication Elements

Figure 2.7: The process for applying gamification in education (Huang and Soman, 2013)

Understanding the Target Audience and the Context

It is essential to determine factors about the target audience (such as age, group or learning abilities) and the context (i.e. group size, environment or sequencing of skills). Above these, there are a number of questions that need consideration. How long should the duration of the program be? Where will the program take place (classroom, office or somewhere else)? Which is more suitable, a group or one-on-one setting? And finally: What would be the ideal group size? Questions like these help to determine 'pain points' for the learning program. Such common pain points include:

- focus
- motivation
- skills
- pride
- physical, mental, emotional factors
- learning environment and nature of the course

Defining Learning Objects

Objectives should not only be specific learning goals, but also behavioural goals. Every success of a program is dependent a clear definition of the learning objects.

Structuring the Experience

Stages and milestones are important and should be inserted in a learning program, because it makes the final goal seem more tangible to achieve and also measurable.

Identifying Resources

After defining stages and milestones, the teacher has to think about which of those have to be gamified. The following questions need to be taken into

Self-Elements (Complete Stage)	Social Elements (Push Stage)
Points	Leaderboards
Levels	Virtual Goods
Trophies / Badges	Interactive Cooperation
Virtual Goods	Storyline
Storyline	
Time Restrictions	
Aesthetics	

Table 2.2: Game Mechanics Examples (self-elements vs. social-elements) (Huang and Soman, 2013)

account: Should a tracking mechanism be used? When is a level successfully completed? Are there rules that should be followed? But also, should there be a feedback? If so, how does it look like?

Applying Gamification Elements

Huang and Soman, 2013 mention that gamification also mean the addition of game-like elements or so-called game mechanics. These mechanics can be devided into self-elements and social elements. Examples for the mentioned elements can be seen in Table 2.2

2.4 E-Learning Tools

Nowadays there are so many e-Learning platform that it would probably be impossible to create an list of them all. However, Lecturio, 2016 mentions that classroom teaching plays an increasingly minor role in continuing education and provides a list of ten e-Learning tools for all different sorts of needs.

1. **For learners**
   Lecturio E-Learning-App[1] provides paid videos course for smartphones, tablets and PC.

[1] https://www.lecturio.de/ (last visited September 09 2019)

2. **For repeat offenders**
   With brainyoo[2] it is possible to create digital flashcards.
3. **For cosmopolitans**
   Babbel[3] is a popular app for learning different langauges.
4. **For teamplayer**
   Already a classic called Moodle[4]. With digital classrooms students have access to lots of learning materials and collaborate with others.
5. **For organization freaks**
   With Wunderlist[5] one can digitalize its To-Do lists.
6. **For inspiration seeker**
   Pinterest[6] is not only on 14th place of the top 100 tools for learning[7], it is well suited to gather inspiration for learning material in everyday life.
7. **For eccentric collectors**
   According to pocket[8] you can now put knowledge into your Pocket.
8. **For time savers**
   Feedly[9] lets you keep up with what you are interested in.
9. **For writers**
   Handwritten notes can be digitalized with Notability[10].
10. **For player**
    duolingo[11] is a free language acquisition tool.

[2] https://www.brainyoo.de/ (last visited September 09 2019)
[3] https://www.babbel.com/ (last visited September 09 2019)
[4] http://moodle.de/ (last visited September 09 2019)
[5] https://www.wunderlist.com/de/ (last visited September 09 2019)
[6] https://www.pinterest.com/ (last visited September 09 2019)
[7] http://c4lpt.co.uk/top100tools/ (last visited September 09 2019)
[8] https://getpocket.com/ (last visited September 09 2019)
[9] https://feedly.com/ (last visited September 09 2019)
[10] http://www.gingerlabs.com/ (last visited September 09 2019)
[11] https://de.duolingo.com/ (last visited September 09 2019)

3 About the IDeRBlog Platform

The following chapter describes the online platform IDeRBlog[1] (see also Figure 3.1) which has set the goal to motivate pupils form the age of 8 to 12 to practice their spelling skills, both in reading and writing. (Markus Ebner, Martin Ebner, and Edtstadler, 2016)

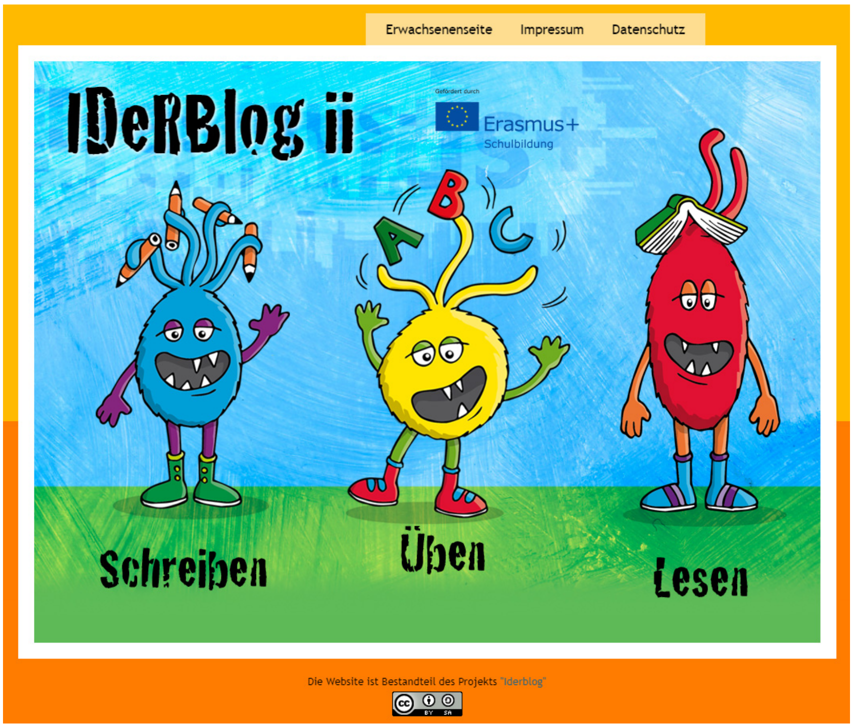


Figure 3.1: The homepage of the IDeRBlog website and its design

[1] http://iderblog.eu/ (last visited September 09 2019)

3.1 Overview

First of all, the name IDeRBlog is a German acronym for "***I**ndividuell **D**ifferenziert **R**ichtig schreiben mit **Blog**s*" (Aspalter, Edtstadler, and Martich, 2017) which, translated to English, means: "*Individually differentiated spelling with blogs*" (Markus Ebner, Edtstadler, and Martin Ebner, 2018)

It is part of a Erasmus+[2] project and consists of a team of three countries: Germany, Belgium and Austria (Aspalter, Edtstadler, and Martich, 2017, P.13).

As mentioned by Gros et al., 2015 writing, reading and calculating are not only important social qualities, but also a requirement in the work force nowadays. Writing can be split into three fields:

1. having writing skills
2. creating texts
3. writing correctly

All of the aforementioned points are acknowledged on the IDeRBlog website[3], with the main focus being on the last point, *writing correctly*, and especially the improvement of the student's orthography. On the website, pupils are given a platform where they can create and publish their own texts in the form of blog entries. The resulting blogs can not just be created on computers, but also on tablets and other mobile devices. Pupils can also created class diaries and make them available to the wider public. As a result, the motivation for writing increases automatically. (Gros et al., 2015)

After writing and uploading their texts, they are analysed by the intelligent dictionary first (see next section). Consequently, the students receive immediate feedback. Now, they have the choice to either correct their errors before submitting their answers, or submit them directly, without correcting their mistakes. It is necessary for a teacher to proofread the answers, before

[2] Erasmus+ https://ec.europa.eu/programmes/erasmus-plus/ (last visited September 09 2019)

[3] http://iderblog.eu/ (last visited September 09 2019)

finally publishing them on the platform. Figure 3.2 shows the workflow on how the process works in detail.

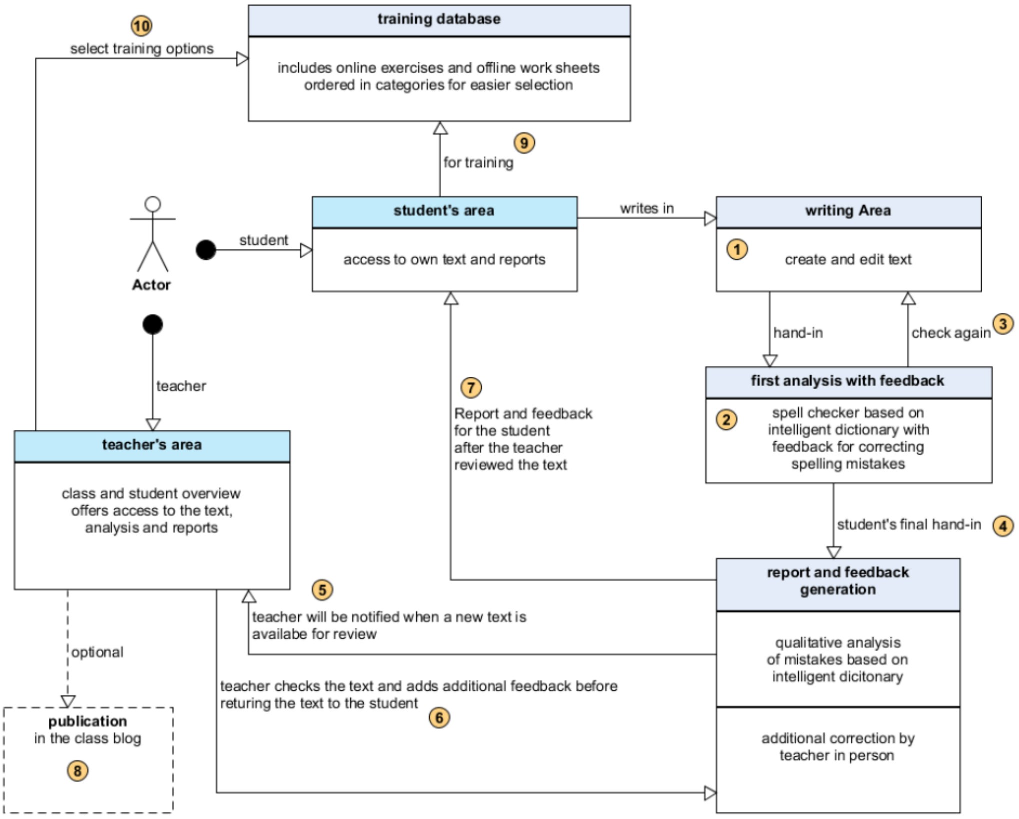

10
select training options
training database
includes online exercises and offline work sheets ordered in categories for easier selection
9
for training
student's area
access to own text and reports
writes in
writing Area
1
create and edit text
Actor
student
teacher
hand-in
check again
3
first analysis with feedback
2
spell checker based on intelligent dictionary with feedback for correcting spelling mistakes
7
Report and feedback for the student after the teacher reviewed the text
teacher's area
class and student overview offers access to the text, analysis and reports
student's final hand-in
4
report and feedback generation
qualitative analysis of mistakes based on intelligent dicitonary
additional correction by teacher in person
5
teacher will be notified when a new text is availabe for review
optional
teacher checks the text and adds additional feedback before returing the text to the student
6
publication
in the class blog
8

Figure 3.2: Workflow of the IDeRBlog platform (Edtstadler, Markus Ebner, and Martin Ebner, 2015)

Since the platform has younger students (age eight to twelve) as its target audience, the design and its process had to be coordinated accordingly, so the webinterface is appealing enough for that specific target group (Liebal and Exner, 2011).

The resulting design can be seen in Figure3.1. Students were also integrated in the creation process of the graphics, which was then finalized by the artist Bettina Weyland[4] (Gros et al., 2015).

[4] http://www.grafikdesign-weyland.de (last visited September 09 2019)

3.2 Intelligent Dictionary

As described by Markus Ebner, Martin Ebner, and Edtstadler, 2016 the main idea of the intelligent dictionary is to offer pupils immediate feedback by providing them with hints regarding their spelling mistakes. However, this does not work like a standard autocorrection system; where correct words are suggested to the users so they can correct the mistake with just one click, without actually thinking about the mistake they have done. The intelligent dictionary shows hints by asking the user to read the word carefully again, and think about what the right spelling could be. For example, it delivers feedback in the form of telling students to think about possible variants of a word, or to contemplate a longer variant before choosing a solution.

4 Implementation of Interactive Learning Objects

This chapter will provide a detailed description of all interactive learning objects, which were also mentioned in the previous chapter. Basically, all of the interactive learning objects are structured into two different types of categories:

1. default exercises
2. database-based exercises

Although the main purpose of the exercises is to help students to improve their spelling mistakes, sometimes it can be more effective to present students with an exercise that does not primarily focus on their own mistakes. Thus, exercises with whole sentences and diverse examples can help to comprehend how and why a word is written in a certain way.

All 'default exercises' consist of sentences where students either have to find mistakes and correct them, or for example, receive a cloze text where they can practice the correct spelling of all included words.

'Database-based exercises' are exercises that are more focused on the specific mistakes of students, which were created and published by the user on the IDeRBlog website[1]. Compared to 'default exercises', the main focus here are not sentences but words students have misspelled at least once. With a few exceptions, most words are provided by IDeRBlog's intelligent dictionary (described in Section 3.2).

[1] http://iderblog.eu/ (last visited September 09 2019)

4.1 Exercises Overview

As already mentioned on the previous page, there are essentially two different categories of interactive learning objects[2].

The first category, ′**standard exercises**′ includes two exercises:

1. Einsetzen (insert)
2. Fehlerstellen (fault locations)

The second category ″**database-oriented exercises**″ consists of three different types of exercises:

1. Merkwörter (cue words)
2. Zuordnen (assign)
3. Glücksrad (wheel of fortune)

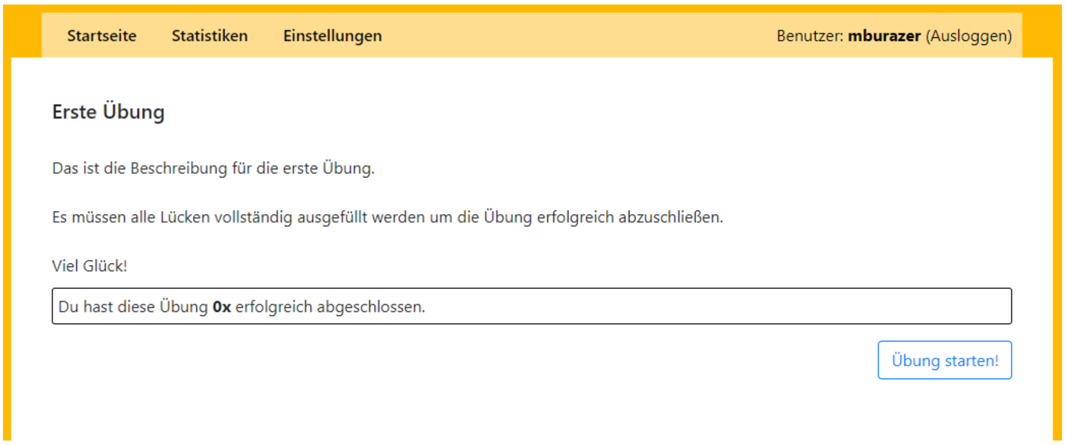
As one can see, the names of all the exercises are German words, which is due to the fact that the IDerBlog exercises are about German language acquisition.

Essentially, every exercise is structured in the same way. There is always a starting page, which has the purpose to explain the current exercise so that the users always have some information on how to complete each one of them (see Figure 4.1). Furthermore, if the user has already completed an exercise at least once, there is also some statistical information of the previous sessions available. This way, students can track their own progress which might trigger their motivation even further (see Figure 4.2).

In addition, after completing an exercise, one can always find a summary of the currently completed session. This way, the user is immediately made aware of the amount of mistakes that were made (if any). The exercises can then be repeated as often as the user wishes.

Before implementing each exercise, so-called mockups were created and used to make the implementation process easier. The Viking Code School (School, 2019) describes mockups as sketches that illustrate how a web page

[2] It should be mentioned that from now on, the term ‘exercises’ will be used instead of ‘learning objects”



Erste Übung

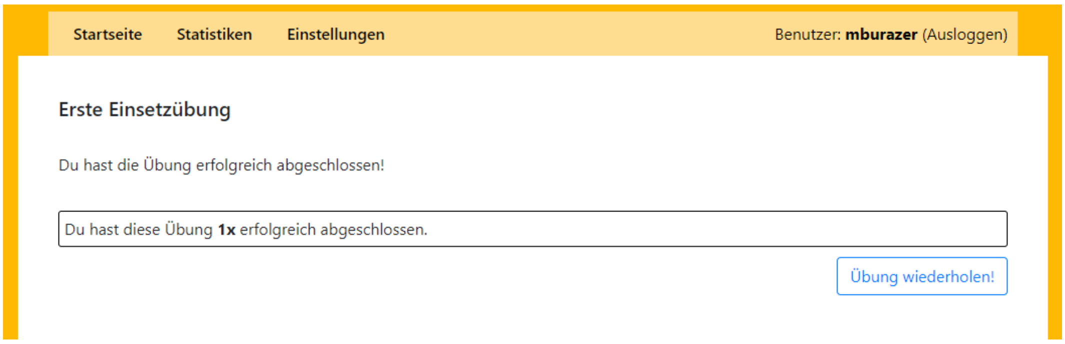
Das ist die Beschreibung für die erste Übung.

Es müssen alle Lücken vollständig ausgefüllt werden um die Übung erfolgreich abzuschließen.

Viel Glück!

Du hast diese Übung **0x** erfolgreich abgeschlossen.

Figure 4.1: Starting page of every exercise (example from Einsetzen (*insert*) exercise)



Erste Einsetzübung

Du hast die Übung erfolgreich abgeschlossen!

Du hast diese Übung **1x** erfolgreich abgeschlossen.

Figure 4.2: Last page after every exercise is finished (example from Einsetzen (*insert*) exercise)

will look like. All of the implemented learning objects were implemented based on these mockups, which were made with a trial version of a software program called balsamiq[3].

The following next pages will give an in depth description of the mentioned exercises.

[3] https://balsamiq.com/ (last visited on September 09 2019)

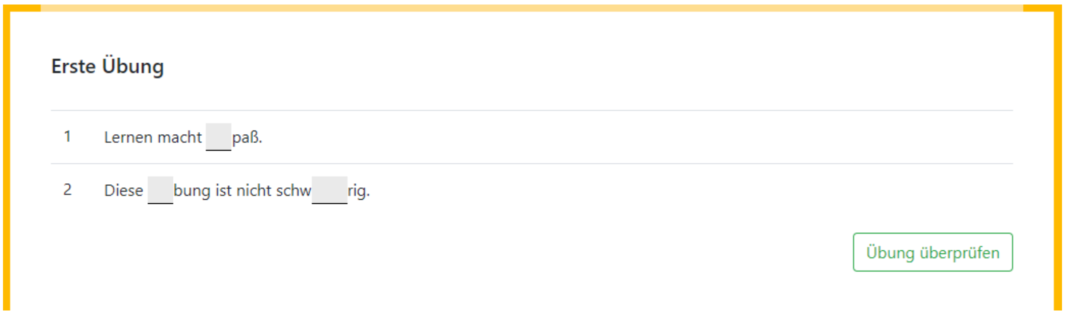
4.2 Exercise Einsetzen (insert)

The Einsetzen (*insert*) exercise is a typical cloze text exercise where students are presented with several sentences with gaps. The task for the students is to fill in the gaps with the correct words or letters.

This exercise is the first one of the 'standard exercises' category, which is unrelated to the specific spelling mistakes a user makes.

Before starting the exercise, a page appears, informing the student how to complete the exercise. In case the user has already completed this exercise before, some additional statistics are displayed as well. The statistics may show how many times the exercise has been completed or the progression level of the student.

Once the exercise has begun, all of the gap sentences that have to be filled with the correct words or letters, are shown (see Figure 4.3). Depending on the settings, which can be managed by an administrator (or a teacher with rights to edit the exercise settings), there is also a tooltip (see Figure 4.4) for every gap. This tooltip gives the user a hint as to what the correct word or letter might be.



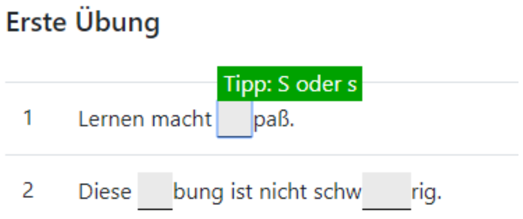
Erste Übung

1 Lernen macht ___paß.

2 Diese ___bung ist nicht schw___rig.

Figure 4.3: Einsetzen (*insert*) exercise with cloze text example

To complete the exercise and find out if all gaps are filled in correctly, the user has to click on the "check exercise" button first. If everything is inserted correctly, the exercise is finished and the user receives some statistics. In case there is at least one error, the same page is loaded again. All gaps which were filled in correctly are no longer displayed as input fields. However,



Erste Übung

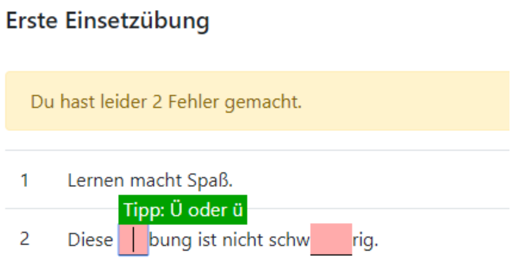
Tipp: S oder s

1 Lernen macht paß.

2 Diese bung ist nicht schw rig.

Figure 4.4: Example for a tooltip in the Einsetzen (*insert*) exercise

those gaps which contained errors are still input fields and highlighted in a red color instead. Additionally, at the top of the page there is a text showing how many mistakes the user has made (see Figure 4.5).



Erste Einsetzübung

Du hast leider 2 Fehler gemacht.

1 Lernen macht Spaß.

Tipp: Ü oder ü

2 Diese bung ist nicht schw rig.

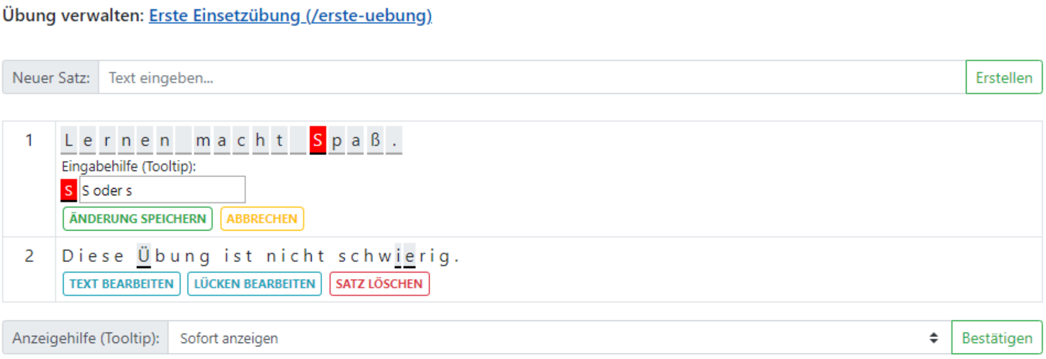
Figure 4.5: Example for displaying user errors in the Einsetzen (*insert*) exercise

Administrators and teachers can create different types of Einsetzen (*insert*) exercises. For example, there might be one exercise for capitalization[4], one for the many different ways of the German *"ss, s or ß"* rules or even a mix of many different types.

The number of sentences and gaps can be set by teachers, there are no limitations for that. Additionally, the tooltip should help the pupils in situations where it is not clear what word or letter should be used to fill in the gap.

[4] Detailed comparison between English and German capitalization rules: ThoughtCo, 2019 (`https://www.thoughtco.com/capitalization-in-german-4069437`) (last visited on September 09 2019)

Figure 4.6 shows how an administrator or teacher can manage an exercise. Every letter, or even spaces, can be set as a gap. When a gap is selected, there is an input field beneath. In this field, a textual note can be added so the user does not have to guess the word or letter that has to be filled in the gap.



Übung verwalten: Erste Einsetzübung (/erste-uebung)

Neuer Satz: Text eingeben...

1 Lernen macht Spaß.

Eingabehilfe (Tooltip): S S oder s

2 Diese Übung ist nicht schwierig.

Anzeigehilfe (Tooltip): Sofort anzeigen

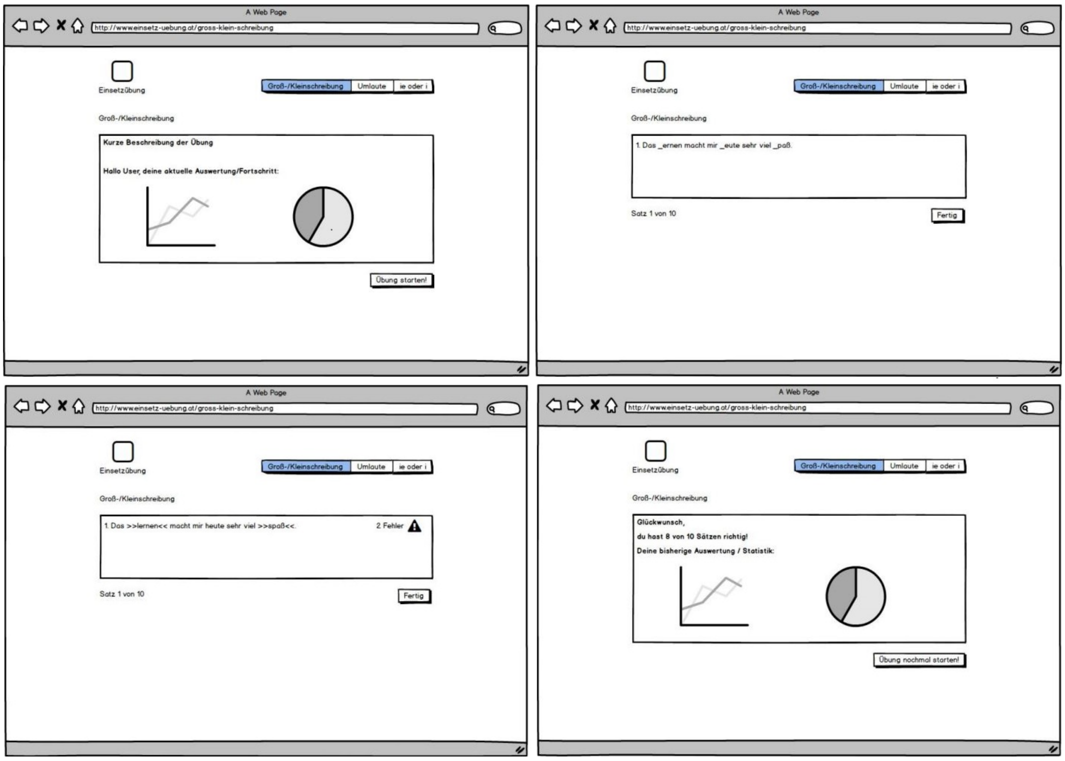
Figure 4.6: Example for managing the Einsetzen (*insert*) exercise

In Figure 4.7 one can see how the exercise was planned to be implemented with the help of mockups.

4.3 Exercise Fehlerstellen (fault locations)

The second exercise of the ”standard exercises” is the so called Fehlerstellen (*fault locations*) exercise. Looking at its basic structure, it is comparable with the previously explained exercise. Similarly, it does not entail user-specific spelling mistakes, but also contains several sentences with errors.

The challenge here is to locate all fault locations in every sentence. Further, it has to be noted that there can be more than one error in each sentence. Depending on the settings, every sentence can contain an error, or every word can be an error even. The user has to identify every fault location by clicking a word. In case an incorrect word was selected, it will be marked thick and green for a short period of time (can be seen in Figure 4.8 with



Kurze Beschreibung der Übung

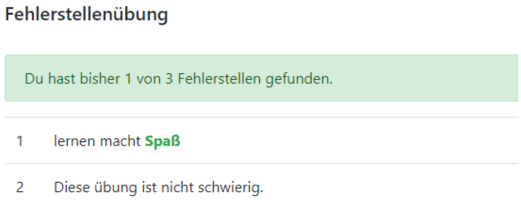
Hallo User, deine aktuelle Auswertung/Fortschritt:

1. Das _ernen macht mir _eute sehr viel _paß.

1. Das >>lernen<< macht mir heute sehr viel >>spaß<<.

Glückwunsch,
du hast 8 von 10 Sätzen richtig!
Deine bisherige Auswertung / Statistik:

Figure 4.7: Mockups for the Einsetzen (*insert*) exercise



Fehlerstellenübung

Du hast bisher 1 von 3 Fehlerstellen gefunden.

1 lernen macht **Spaß**

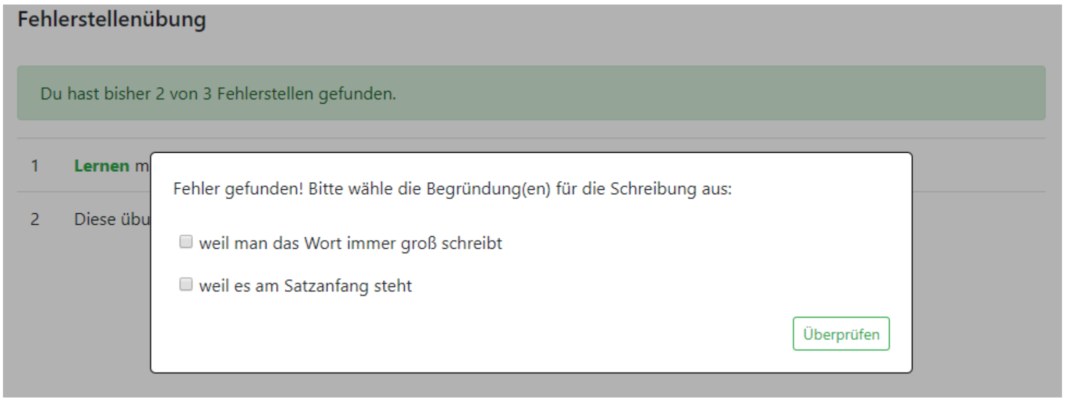
2 Diese übung ist nicht schwierig.

Figure 4.8: Example the Fehlerstellen (*fault locations*) exercise with one fault location found

the word ”Spaß” (*fun*)). If the user identifies a fault location, a small pop-up dialog appears after which the user has to justify his decision.

Due to the fact that the exercise can also be completed when the user clicks on every word, it would be difficult for the teaching staff to verify whether the students completed the exercise based on their own knowledge or cheating. Accordingly, every time a faulty location is selected, it must be justified (see Figure 4.9). Based on those justifications, the teacher can thus observe if the pupil did the exercise thoughtfully or not.

When the justification dialog appears a list of all possible answers is displayed. These answers can be selected by clicking on a checkbox (see Figure 4.9). If the user does not select all right answers, he will get immediate feedback. All correct answers will be highlighted green and the incorrect ones red (see Figure 4.10). Afterwards, when all answers are given, the dialog disappears and the user can continue with the task to find all the remaining fault locations.



Fehlerstellenübung

Du hast bisher 2 von 3 Fehlerstellen gefunden.

1 **Lernen** m

2 Diese übu

Fehler gefunden! Bitte wähle die Begründung(en) für die Schreibung aus:

□ weil man das Wort immer groß schreibt

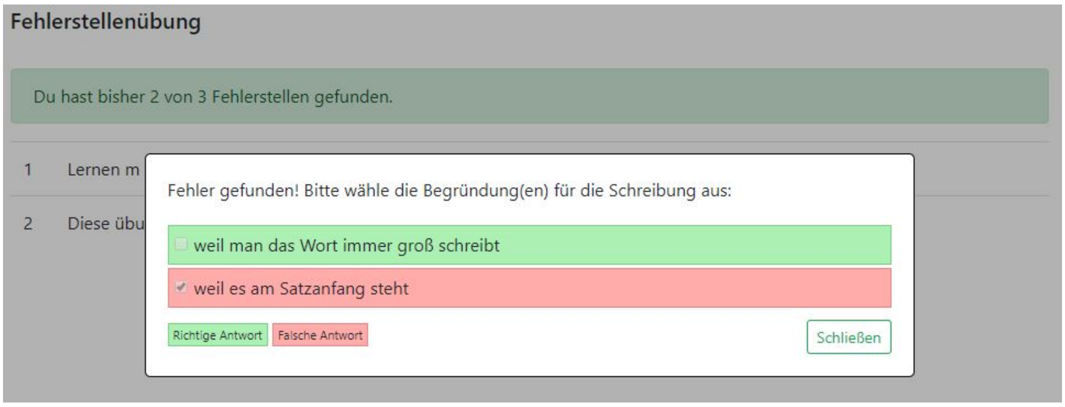
□ weil es am Satzanfang steht

Figure 4.9: Example when user has to justify decision in the Fehlerstellen (*fault locations*) exercise

From the administrator's point of view, after creating an Fehlerstellen (*fault locations*) exercises, first all sentences have to be created. Afterwards the fault locations must be defined. A fault location is at least one word, which means it is possible to have multiple words defined as one fault location. The interface for managing an exercise can be seen in Figure 4.11.

After creating all sentences and the corresponding fault locations the administrator must define justifications for every single fault location. There are two types of justifications:

1. default justifications
2. custom justifications



Fehlerstellenübung

Du hast bisher 2 von 3 Fehlerstellen gefunden.

1 Lernen m

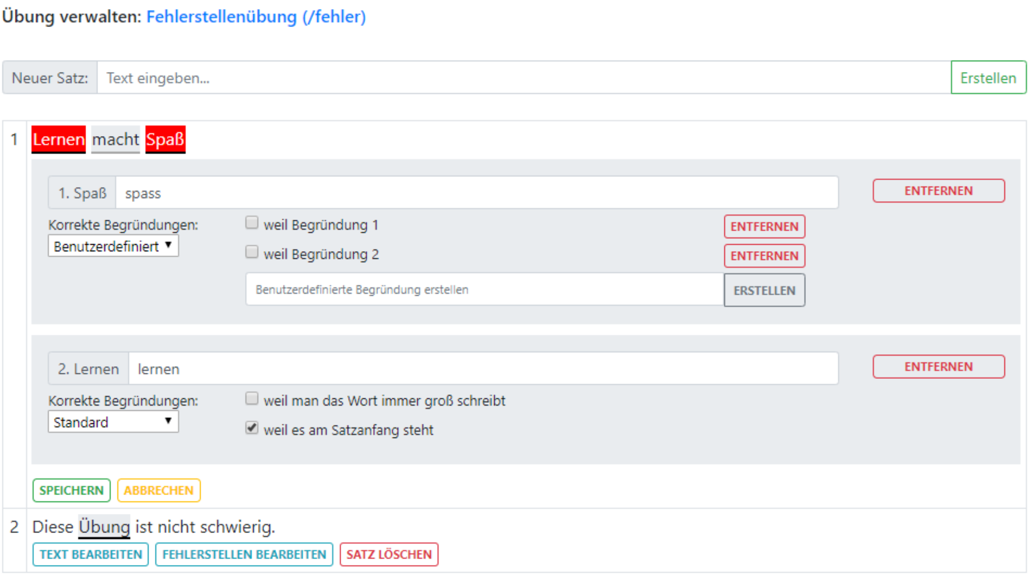
2 Diese übu

Fehler gefunden! Bitte wähle die Begründung(en) für die Schreibung aus:

□ weil man das Wort immer groß schreibt

☑ weil es am Satzanfang steht

Figure 4.10: Example when user selected a wrong justification in the Fehlerstellen (*fault locations*) exercise



Übung verwalten: Fehlerstellenübung (/fehler)

Neuer Satz: Text eingeben...

1 Lernen macht Spaß

1. Spaß spass

Korrekte Begründungen: Benutzerdefiniert ▾

□ weil Begründung 1

□ weil Begründung 2

Benutzerdefinierte Begründung erstellen

2. Lernen lernen

Korrekte Begründungen: Standard ▾

□ weil man das Wort immer groß schreibt

☑ weil es am Satzanfang steht

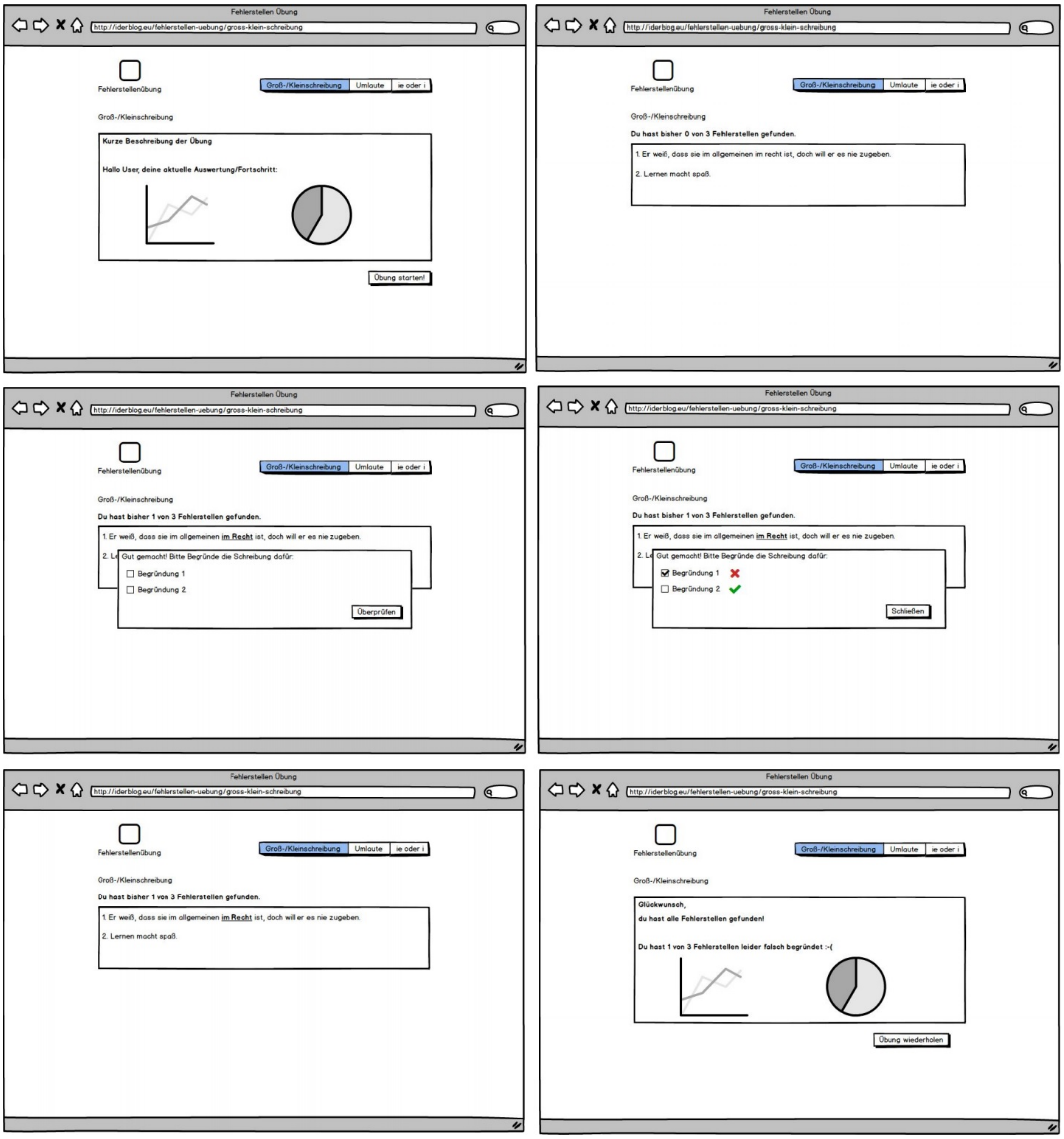
2 Diese Übung ist nicht schwierig.

Figure 4.11: Example for managing the Fehlerstellen (*fault locations*) exercise

Default justifications are automatically applied to all fault locations, the administrator must only define which answers of a default justification are assigned to a specific fault location.

Custom justifications can be used if a fault locations cannot or should not be justified with the default justifications.

Figure 4.12 show the mockups for the Fehlerstellen *fault locations* exercise.



Groß-/Kleinschreibung

Kurze Beschreibung der Übung

Hallo User, deine aktuelle Auswertung/Fortschritt:

Du hast bisher 0 von 3 Fehlerstellen gefunden.

1. Er weiß, dass sie im allgemeinen im recht ist, doch will er es nie zugeben.

2. Lernen macht spaß.

Du hast bisher 1 von 3 Fehlerstellen gefunden.

1. Er weiß, dass sie im allgemeinen **im Recht** ist, doch will er es nie zugeben.

Gut gemacht! Bitte Begründe die Schreibung dafür:

☐ Begründung 1

☐ Begründung 2

☑ Begründung 1 ✗

☐ Begründung 2 ✓

Glückwunsch,

du hast alle Fehlerstellen gefunden!

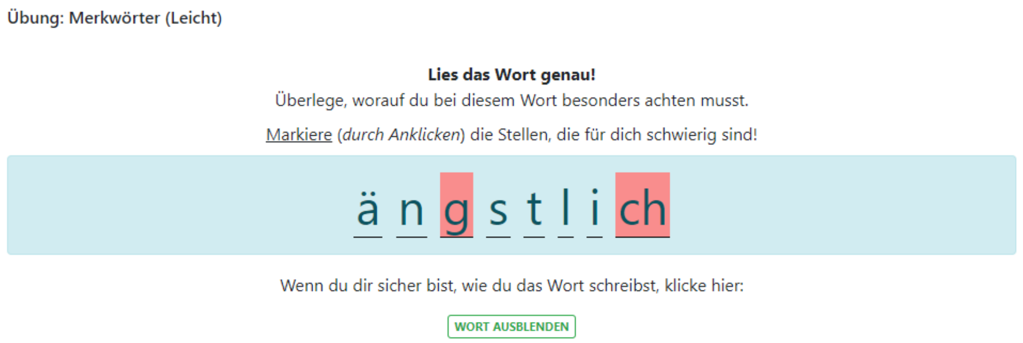
Du hast 1 von 3 Fehlerstellen leider falsch begründet :-(

Figure 4.12: Mockups for the Fehlerstellen (*fault locations*) exercise

4.4 Exercise Merkwörter (cue words)

Unlike the case with the first two exercises (the default exercises), the main focus for the Merkwörter (*cue words*) exercises are the user-specific spelling errors. Consequently, what this means is that the user does not have to find and correct errors in a sentence. Rather, the focus is on single words only, words which are provided by IDeRBlog's intelligent dictionary (described in Section 3.2).

Depending on the exercise settings, after beginning the exercise a single word could be displayed to the user first (see Figure 4.13 or an example). As a result, the user has to take a closer look at that word, thinking about possible problems that could potentially arise when using it. For example, a problem could be that a word is always written in lowercase instead of uppercase, or that some specific letters of a word are forgotten or swapped with another letter. Additionally, settings could also be adjusted in a way so that multiple words, instead of a single word only, are shown to the user.


Übung: Merkwörter (Leicht)

Lies das Wort genau!

Überlege, worauf du bei diesem Wort besonders achten musst.

Markiere (*durch Anklicken*) die Stellen, die für dich schwierig sind!

ä n g s t l i ch

Wenn du dir sicher bist, wie du das Wort schreibst, klicke hier:

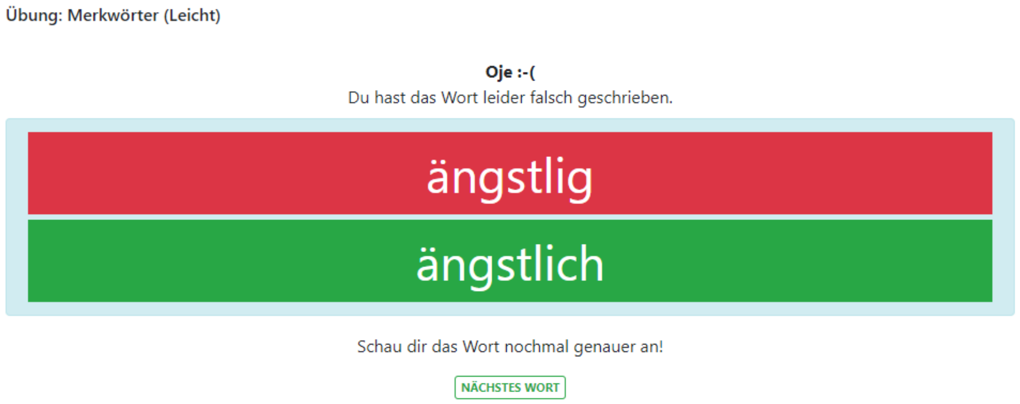
Figure 4.13: Example for marking letters/grapheme of a word in the Merkwörter (*cue words*) exercise

It is not only a good practise for pupils to think about a specific word, it is also beneficial for teachers to know how students view their own weaknesses. By having a greater understanding of the students' flaws, teachers have a more effective impact on the creation of future exercises by enhancing them accordingly.

In order for teachers to be aware where those weaknesses are, the Merkwörter (*cue words*) exercises allow the user to click on every letter or grapheme of the given word.

After having a closer look and marking graphemes, the user can then click on a button for the next step of the exercise to appear. The word disappears and an input field shows up. Now, the word has to be written without seeing it. Writing the word again not only helps to better memorize it, but it also increases the likelihood of writing the word correctly the next time. If there are more words displayed at the same time, the user is also presented with several input fields. In such case, the word order is not important, because the program will automatically recognize the corresponding words.

When the word has been written and the user is sure of its accuracy, he can then click on the next button to see the next word. Again, the student has to analyze the word and set all the markings where he thinks the possible issues could be. In case the word is written wrong, both the wrong input as well as the correct spelling is displayed. The wrong input is highlighted red and the correct word green (see Figure 4.14).


Übung: Merkwörter (Leicht)

Oje :-(
Du hast das Wort leider falsch geschrieben.

ängstlig

ängstlich

Schau dir das Wort nochmal genauer an!

Figure 4.14: Example for feedback if the user misspelled a word in the Merkwörter (*cue words*) exercise

To make the exercise more interesting and challenging an administrator can change the settings so the exercise can be created with different difficulty levels (see Figure 4.15).

Difficulty levels could look like this:

4.4.1 Easy

In the **easy mode** for example, a word could be displayed without any time limitations, i.e. there is no pressure to complete the exercise as fast as possible.

4.4.2 Moderate

The **moderate mode** is supposed to make the whole exercise more interesting and fun. Instead of one word, there could be three words that are shown. Additionally, a time limitation could be set with, for example 20 seconds.

4.4.3 Hard

For the ambitious pupils, the **hard mode** would be the perfect exercise. Here, noticeably more words could be presented coincidently. Also, time limitations could be determined for reading as well as for writing the words. The ultimate goal would be to awake the ambition of the student.

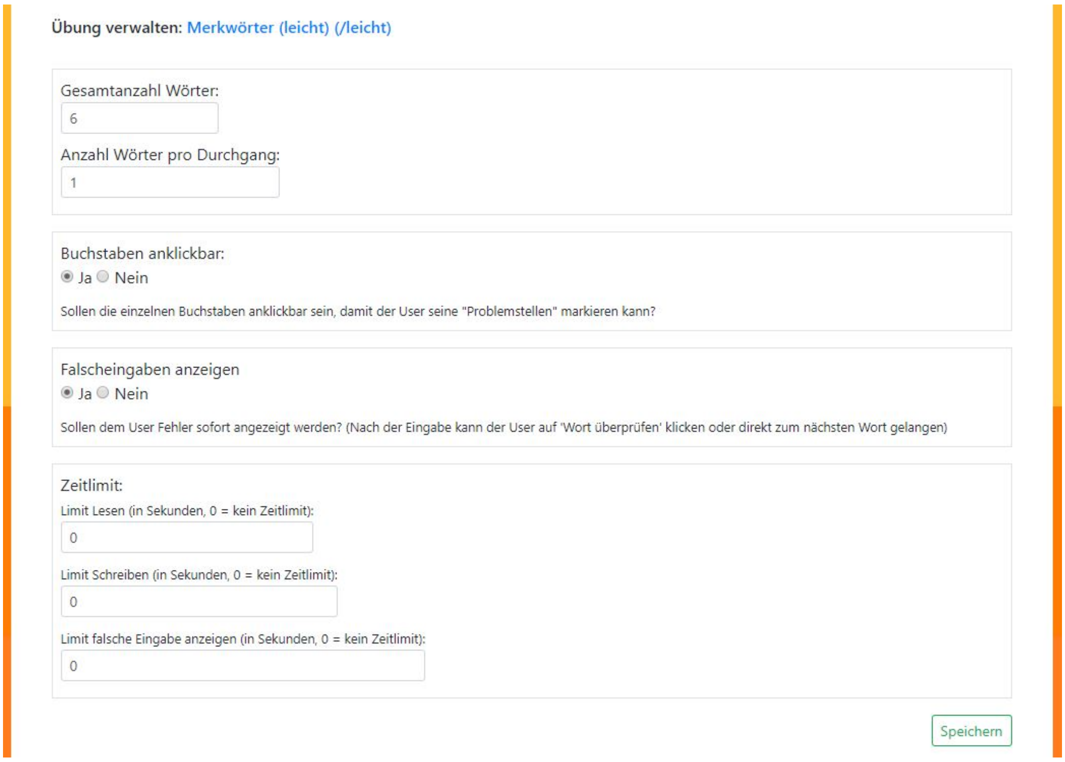
Mockups were also created for this exercise (see Figure 4.16).

4.5 Exercise Zuordnen (assign)

Just like in the previous exercise, in the Zuordnen (*assign*) exercise, the main focus are user-specific spelling mistakes from IDeRBlog's intelligent dictionary (described in Section 3.2).

First, the user is presented with a single word that has to be analysed. The difference in this exercise, however, is that the word has to be assigned to a certain category. It further differs significantly from the others, because it is divided into two different variants:

1. Drag&Drop
2. Write



Übung verwalten: Merkwörter (leicht) (/leicht)

Gesamtanzahl Wörter:
6

Anzahl Wörter pro Durchgang:
1

Buchstaben anklickbar:
● Ja ○ Nein

Sollen die einzelnen Buchstaben anklickbar sein, damit der User seine "Problemstellen" markieren kann?

Falscheingaben anzeigen
● Ja ○ Nein

Sollen dem User Fehler sofort angezeigt werden? (Nach der Eingabe kann der User auf 'Wort überprüfen' klicken oder direkt zum nächsten Wort gelangen)

Zeitlimit:

Limit Lesen (in Sekunden, 0 = kein Zeitlimit):
0

Limit Schreiben (in Sekunden, 0 = kein Zeitlimit):
0

Limit falsche Eingabe anzeigen (in Sekunden, 0 = kein Zeitlimit):
0

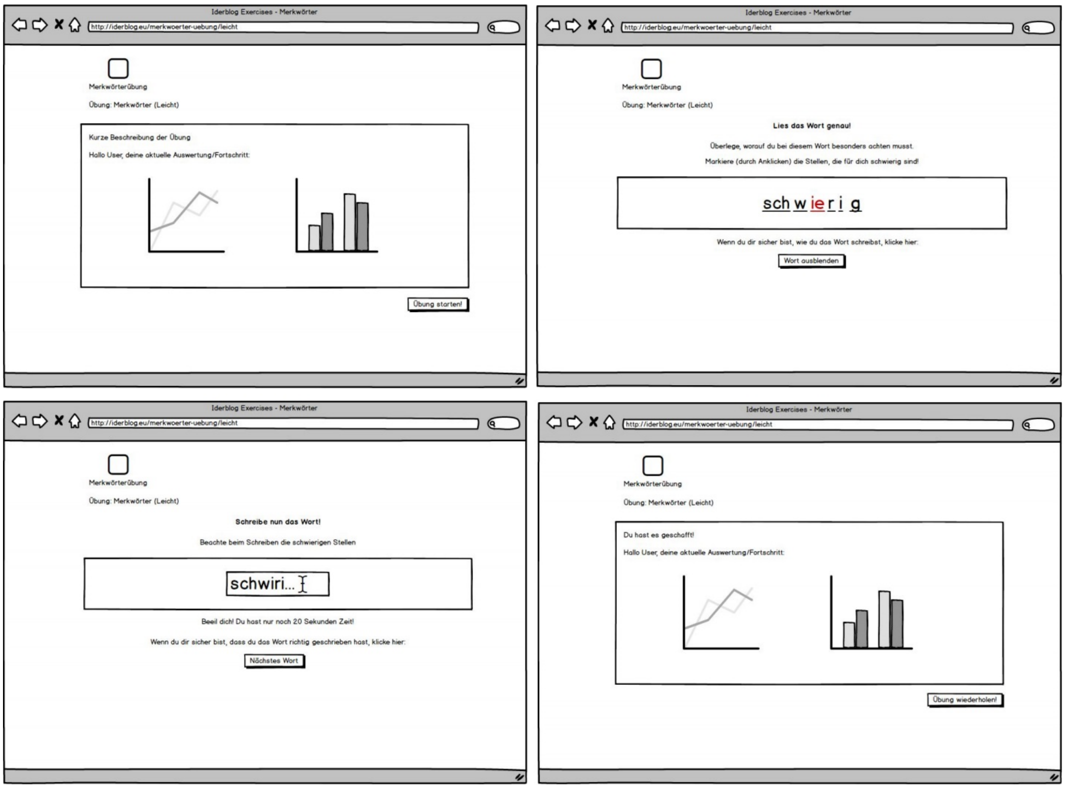
Figure 4.15: Settings example for the 'easy mode' of an Merkwörter (*cue words*) exercise. One word, letters clickable, showing feedback to the user if word was written wrong and no time limitations

4.5.1 Drag & Drop

In the Drag&Drop variant the word is displayed at the same time as the categories (see Figure 4.17). Depending on the currently shown word, there can be two to three different categories. The task for the user is to decide to which category the word belongs to. Also, the shown word can belong to all categories or non of the presented ones.

To successfully complete the exercise, the user has to click on the word and move it, by dragging and then dropping it, to the right category or categories (if there are more than one correct).

By clicking the next button, either the next word is displayed to be assigned



Merkwörterübung

Übung: Merkwörter (Leicht)

Kurze Beschreibung der Übung

Hallo User, deine aktuelle Auswertung/Fortschritt:

Merkwörterübung

Übung: Merkwörter (Leicht)

Lies das Wort genau!

Überlege, worauf du bei diesem Wort besonders achten musst.

Markiere (durch Anklicken) die Stellen, die für dich schwierig sind!

sch w ie r i g

Wenn du dir sicher bist, wie du das Wort schreibst, klicke hier:

Merkwörterübung

Übung: Merkwörter (Leicht)

Schreibe nun das Wort!

Beachte beim Schreiben die schwierigen Stellen

schwiri...

Beeil dich! Du hast nur noch 20 Sekunden Zeit!

Wenn du dir sicher bist, dass du das Wort richtig geschrieben hast, klicke hier:

Merkwörterübung

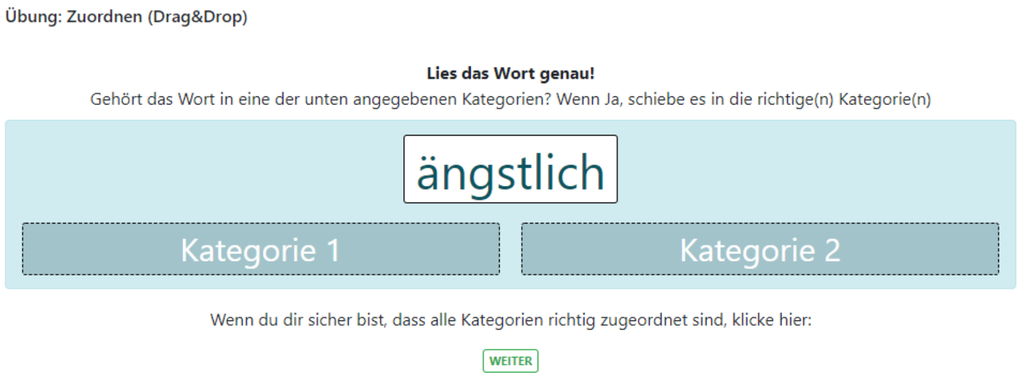
Übung: Merkwörter (Leicht)

Du hast es geschafft!

Hallo User, deine aktuelle Auswertung/Fortschritt:

Iderblog Exercises - Merkwörter
http://iderblog.eu/merkwoerter-uebung/leicht
Übung starten!
Wort ausblenden
Nächstes Wort
Übung wiederholen!

Figure 4.16: Mockups for the Merkwörter (*cue words*) exercise



Übung: Zuordnen (Drag&Drop)

Lies das Wort genau!

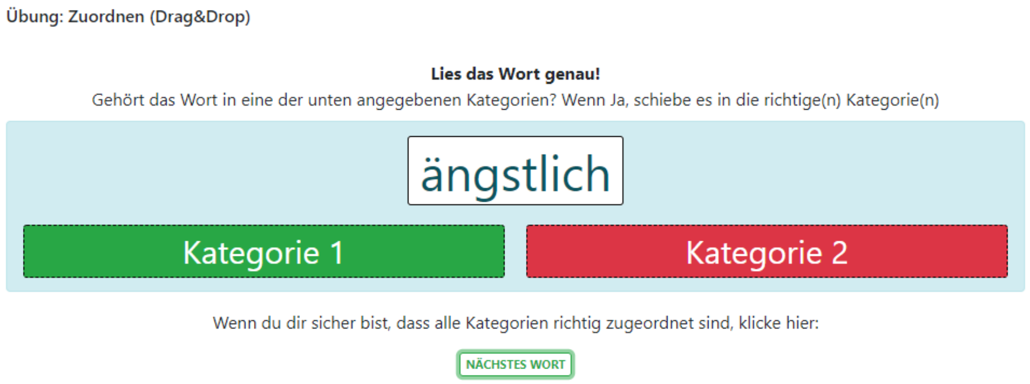
Gehört das Wort in eine der unten angegebenen Kategorien? Wenn Ja, schiebe es in die richtige(n) Kategorie(n)

ängstlich

Wenn du dir sicher bist, dass alle Kategorien richtig zugeordnet sind, klicke hier:

Figure 4.17: Example of the Drag&Drop variant in the Zuordnen (*assign*) exercise

or the wrong selected categories are highlighted to give the user feedback. Wrongly selected categories are outlined red, while correctly selected categories are outlined green (see Figure 4.18).



Übung: Zuordnen (Drag&Drop)

Lies das Wort genau!

Gehört das Wort in eine der unten angegebenen Kategorien? Wenn Ja, schiebe es in die richtige(n) Kategorie(n)

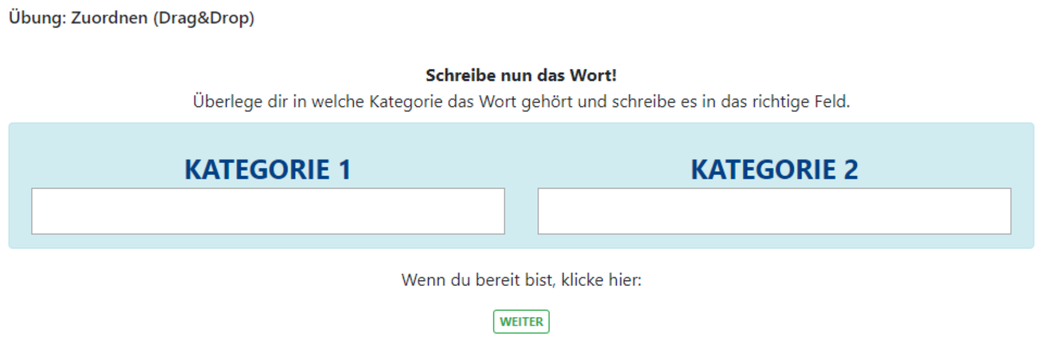
Wenn du dir sicher bist, dass alle Kategorien richtig zugeordnet sind, klicke hier:

Figure 4.18: Example of the Drag&Drop variant with highlighted errors in the Zuordnen (*assign*) exercise

4.5.2 Writing

Compared to the Drag&Drop version, here the user is shown the correct word first (similar to the Merkwörter (*cue words*) exercise). In addition, the student has to take a closer look at the word before it disappears after clicking the next button. Subsequently, after the user has clicked on the next button and the word disappeared, all associated categories are shown. Just beneath every word, an input field becomes visible where the correct word has to be written again. The same word has to be written in all categories which might be correct (see Figure 4.19).

If the user is sure he or she has written the word not only in the right category but also without any spelling errors, he or she can move on to the next word. In case either a category was not chosen correctly or a word is misspelled in any of the available input fields, the user will be provided with immediate feedback. All correct categories are marked as green and all wrong categories are marked as red (see Figure 4.20).



Übung: Zuordnen (Drag&Drop)

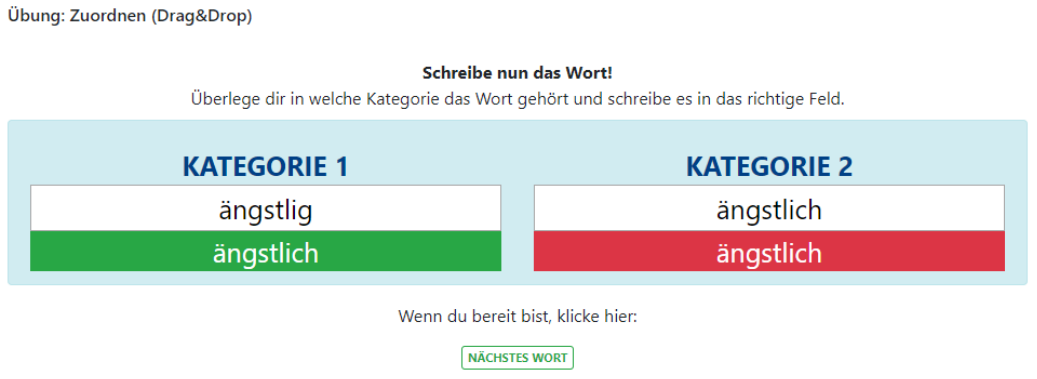
Schreibe nun das Wort!

Überlege dir in welche Kategorie das Wort gehört und schreibe es in das richtige Feld.

Wenn du bereit bist, klicke hier:

Figure 4.19: Example of the Write variant in the Zuordnen (*assign*) exercise

Additionally, all words are also shown beneath every category, falsely written words are highlighted in red, whereas correctly written words are highlighted in green.



Übung: Zuordnen (Drag&Drop)

Schreibe nun das Wort!

Überlege dir in welche Kategorie das Wort gehört und schreibe es in das richtige Feld.

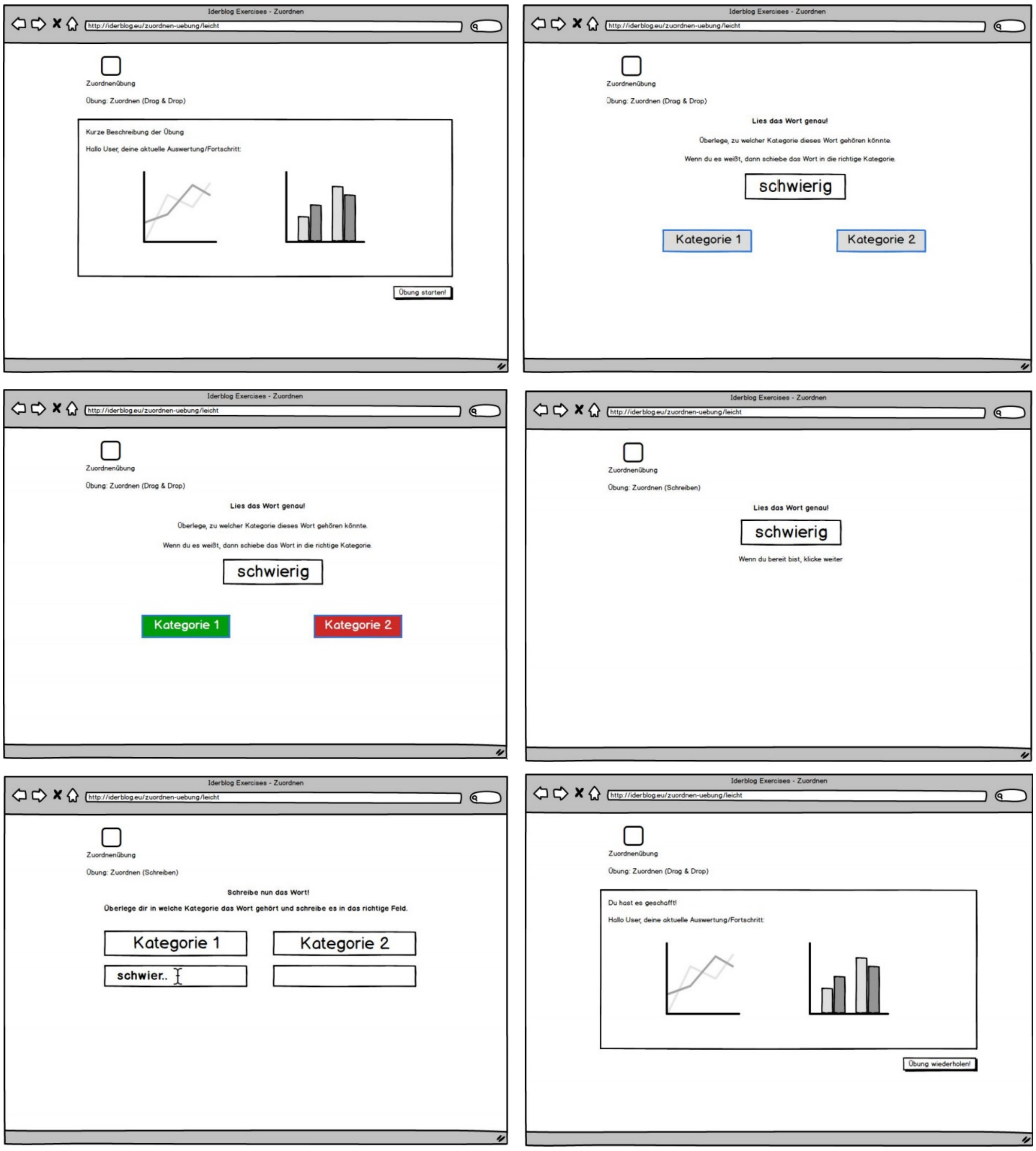
KATEGORIE 1	KATEGORIE 2
ängstlig	ängstlich
ängstlich	ängstlich

Wenn du bereit bist, klicke hier:

Figure 4.20: Example of the Write variant in the Zuordnen (*assign*) exercise

From the administrator's point of view there is only a settings page, which looks almost identical to the previous exercise Merkwörter (*cue words*) (see Figure 4.15).

Also, mokups were created for this exercise too, which can be seen in Figure 4.21.



Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Drag & Drop)
Kurze Beschreibung der Übung
Hallo User, deine aktuelle Auswertung/Fortschritt:
Übung starten!

Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Drag & Drop)
Lies das Wort genau!
Überlege, zu welcher Kategorie dieses Wort gehören könnte.
Wenn du es weißt, dann schiebe das Wort in die richtige Kategorie.
schwierig
Kategorie 1
Kategorie 2

Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Drag & Drop)
Lies das Wort genau!
Überlege, zu welcher Kategorie dieses Wort gehören könnte.
Wenn du es weißt, dann schiebe das Wort in die richtige Kategorie.
schwierig
Kategorie 1
Kategorie 2

Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Schreiben)
Lies das Wort genau!
schwierig
Wenn du bereit bist, klicke weiter

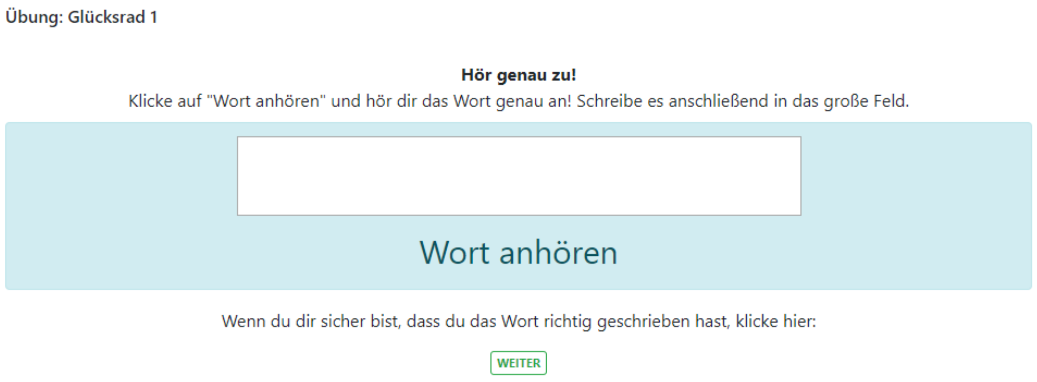
Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Schreiben)
Schreibe nun das Wort!
Überlege dir in welche Kategorie das Wort gehört und schreibe es in das richtige Feld.
Kategorie 1
Kategorie 2
schwier..

Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Drag & Drop)
Du hast es geschafft!
Hallo User, deine aktuelle Auswertung/Fortschritt:
Übung wiederholen!

Figure 4.21: Mockups for the Zuordnen (*assign*) exercise

4.6 Exercise Glücksrad (wheel of fortune)

The Glücksrad (*wheel of fortune*) exercise is the third and last one in the category of the database-oriented exercises. In fact, it is rather similar to the first Merkwörter (*cue words*) exercise; though with the important distinction that the user does not see the word that has to be analysed. Instead, the word is being spoken to the user by the computer by using Text-To-Speech technology. This will be explained in more detail in the chapter for the technical implementation of this master's thesis.



Übung: Glücksrad 1

Hör genau zu!

Klicke auf "Wort anhören" und hör dir das Wort genau an! Schreibe es anschließend in das große Feld.

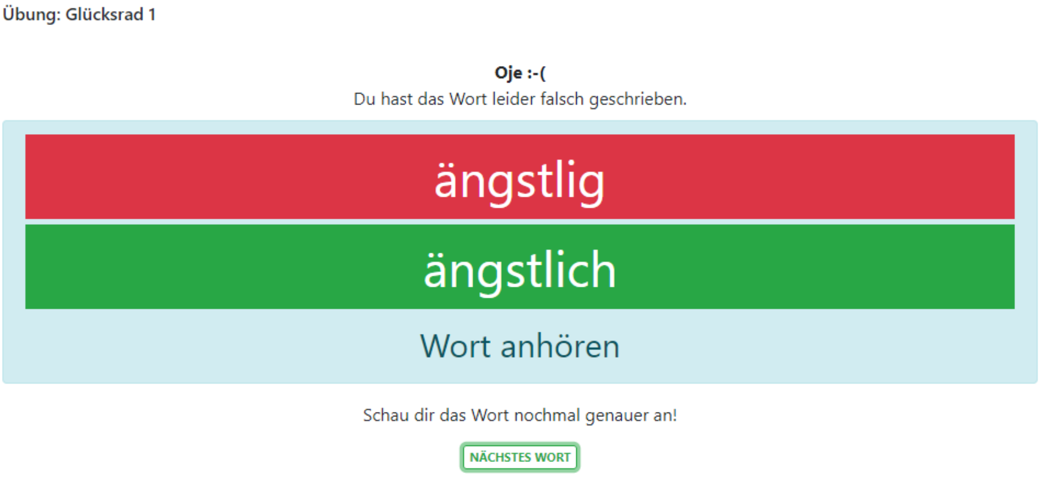
Wenn du dir sicher bist, dass du das Wort richtig geschrieben hast, klicke hier:

Figure 4.22: Example of the Glücksrad (*wheel of fortune*) exercise

Apart from the information that the user has to carefully listen to the spoken word, there is also a button that allows the user to play the word as often as necessary (see Figure 4.22). On the same page, beneath the "play word" button, there is an input field that requires the user to insert the correct word. If the user misspells it, the error feedback is similar to that of the Merkwörter (*cue words*) exercise; the only difference here being that it is possible to listen to the word again (see Figure 4.23).

Althougth the Glücksrad (*wheel of fortune*) exercise is very similar to the Merkwörter (*cue words*) exercise, also for this one mokups were created, which can be seen in Figure 4.24.

The interface for managing this exercise does not differ much from the interface of the Merkwörter (*cue words*), which can be seen in Figure 4.15.



Übung: Glücksrad 1

Oje :-(

Du hast das Wort leider falsch geschrieben.

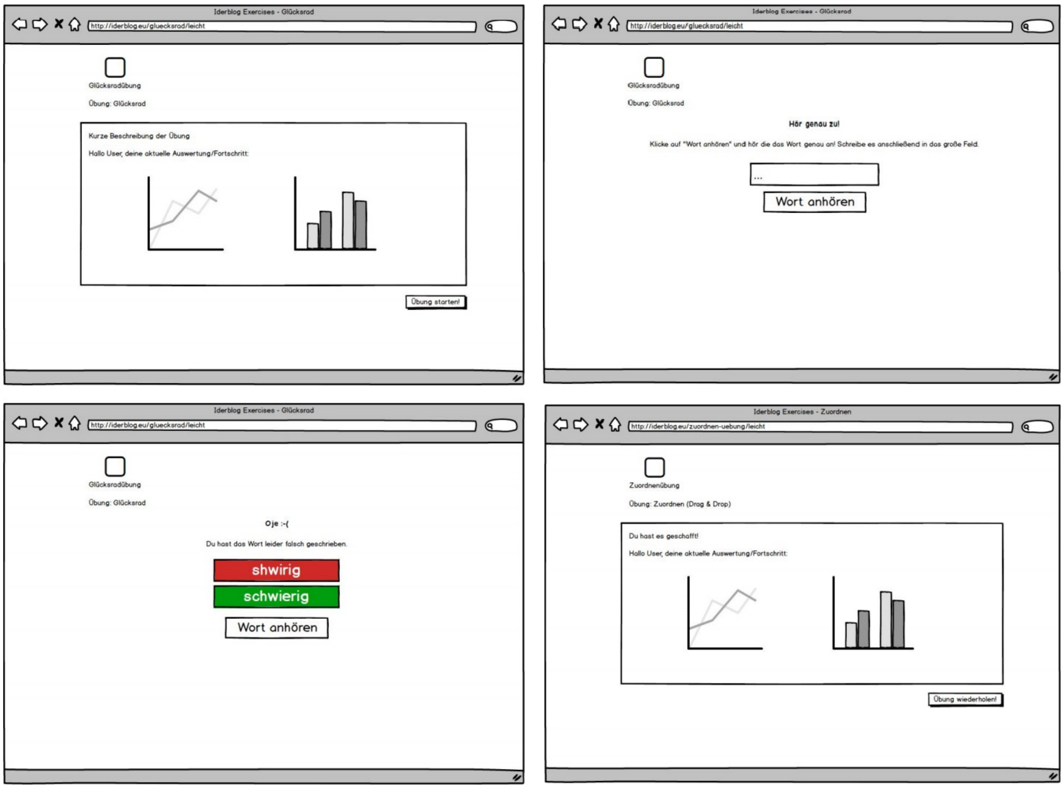
ängstlig

ängstlich

Wort anhören

Schau dir das Wort nochmal genauer an!

Figure 4.23: Example of the Glücksrad (*wheel of fortune*) exercise with errors shown


Iderblog Exercises - Glücksrad
http://iderblog.eu/gluecksrad/leicht
Glücksradübung
Übung: Glücksrad
Übung starten!
Iderblog Exercises - Glücksrad
http://iderblog.eu/gluecksrad/leicht
Glücksradübung
Übung: Glücksrad
...
Wort anhören
Iderblog Exercises - Glücksrad
http://iderblog.eu/gluecksrad/leicht
Glücksradübung
Übung: Glücksrad
shwirig
schwierig
Wort anhören
Iderblog Exercises - Zuordnen
http://iderblog.eu/zuordnen-uebung/leicht
Zuordnenübung
Übung: Zuordnen (Drag & Drop)
Übung wiederholen!

Kurze Beschreibung der Übung

Hallo User, deine aktuelle Auswertung/Fortschritt:

Hör genau zu!

Klicke auf "Wort anhören" und hör die das Wort genau an! Schreibe es anschließend in das große Feld.

Oje :-(

Du hast das Wort leider falsch geschrieben.

Du hast es geschafft!

Hallo User, deine aktuelle Auswertung/Fortschritt:

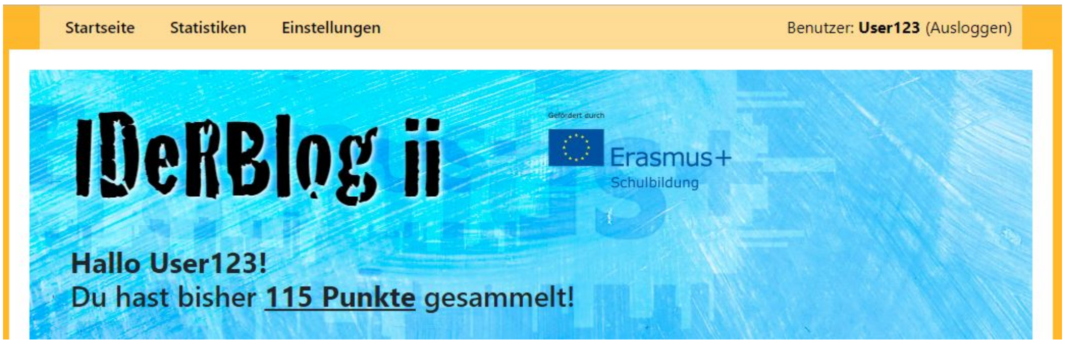
Figure 4.24: Mockups for the Glücksrad (*wheel of fortune*) exercise

5 Implementation of Gamification

The last chapter provided an detailed explanation for all learning objects which were implemented for this master's thesis. However, this chapter will explain how gamification was included and implemented for all exercises.

Learning and improving its own spelling by only executing all exercises might not be always the best way in the long run. Therefore, students must be motivated in a playful way, so they do not give up too fast. They also should have fun repeating the one or other exercise several times so they can benefit from that. Not only with knowledge but also with some sort of visible rewards.

Therefore, to make exercising more fun, the user receives points for all exercises he or she completes. Depending on the exercise, the student can receive different numbers of points. For the Einsetzen (*insert*) exercise, each correctly filled in gap means one point. In the Fehlerstellen (*fault locations*) exercise it is for every correct justification. All other exercises (Merkwröter (*cue words*), Zuordnen (*cue words*) and Glücksrad (*wheel of fortune*)) will reward the user with points for each correct word.

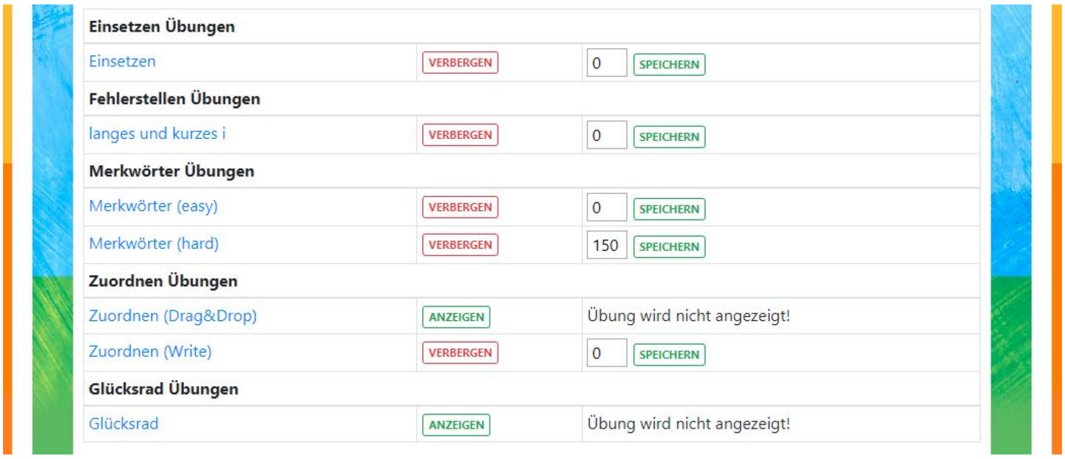


Hallo User123!
Du hast bisher **115 Punkte** gesammelt!

Figure 5.1: Homepage showing how many points the user has collected.

On the homepage, every user can see the obtained points (see Figure 5.1).

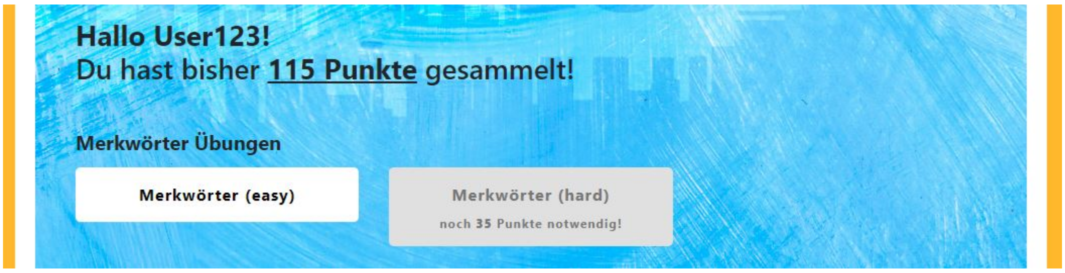
To keep the kids interested and to learn more by exercising more, specific exercises can also be unlocked if there are enough points gathered. For that to happen, there is a separate settings page where an administrator or teacher can decide how many points are needed for each exercise to be unlocked for the users (see Figure 5.2). All exercises and it's different variations can be enabled and disabled at any time.



Einsetzen Übungen		
Einsetzen	VERBERGEN	0 SPEICHERN
Fehlerstellen Übungen		
langes und kurzes i	VERBERGEN	0 SPEICHERN
Merkwörter Übungen		
Merkwörter (easy)	VERBERGEN	0 SPEICHERN
Merkwörter (hard)	VERBERGEN	150 SPEICHERN
Zuordnen Übungen		
Zuordnen (Drag&Drop)	ANZEIGEN	Übung wird nicht angezeigt!
Zuordnen (Write)	VERBERGEN	0 SPEICHERN
Glücksrad Übungen		
Glücksrad	ANZEIGEN	Übung wird nicht angezeigt!

Figure 5.2: Settings page where points can be set for each exercise.

Figure 5.3 shows an example how an unlocked and locked exercise could look like. The white rectangle with black text means the exercise is unlocked and available so the students can immediately see what exercises they can do. Gray rectangle means the exercise is locked. If an exercise is locked, there is always a little text beneath the name which highlights how many points need to be gathered to unlock it.



Hallo User123!
Du hast bisher **115 Punkte** gesammelt!

Merkwörter Übungen

noch 35 Punkte notwendig!

Figure 5.3: Homepage with locked and unlocked exercise.

6 Technical Implementation

This chapter is about the technical implementation of the interactive learning objects, which were explained in the previous chapters. It will give an overview of the used tools and technologies. Further, this chapter will show and explain an interface that can be accessed by other devices to obtain more information; not only about all of the exercises, but also about the logged data. That information can be used to either implement the same or additional exercises on other devices, such as smartphones or tablets, or for analytics measurements.

6.1 Technologies and Tools

6.1.1 Web Application

The target platform for the interactive learning objects are all modern web browsers. Not only because the IDeRBlog platform is build on that, but also because web applications are the easiest and fastest way to reach as many users as possible. Apart from having an internet connection, all that is needed to use a web application is a computer with a browser. No additional programs need to be installed, as it works from the start, regardless of the operating system (PC, Mac or mobile devices like Android and iOS). Moreover, every web application is divided into a client and server. A client, in this case a browser, retrieves all data from the server where the main logic lies.

Gibb, 2016 summarizes the benefits of a web application:

1. They run on multiple platforms.
2. There are compatibility issues, because the users access the same source.
3. No installation is needed, therefore no hard drive limitations occur.
4. Reduces potential software piracy.
5. Reducing costs, because there is less support and maintenance required.

6.1.2 HTML

HTML is a markup language for the representation of web pages and is used by all modern browsers. It is an acronym and stands for Hyper Text Markup Language[1].

Listing 6.1 shows a typical html example file. Almost everything is defined by keywords or tags that are not visible on a page. Tags mostly consist of two parts, an opening and closing tag (Example: <body> is an opening tag and </body> is the closing tag). Moreover, there are always two major tags defining every page: *head* and *body*. The head tag, for example, is used for setting the title of the page or the meta information. Everything that is contained within a head tag is not visible to the user. Unlike everything inside a body tag, which is visible[2].

Listing 6.1: HTML example code

```html
<!DOCTYPE html>
<html>
  <head>
    <title>example</title>
  </head>
  <body>

    Example text!

  </body>
```

[1] https://www.w3schools.com/html/html_intro.asp (last visited on September 09 2019)

[2] Simple description on what HTML is and how it is used: http://www.simplehtmlguide.com/whatishtml.php (last visited on September 09 2019)

```
</html>
```

6.1.3 CSS

The acronym CSS stands for Cascading Style Sheets. HTML is used for structuring web pages, whereas CSS is used to define the styling of every page[3]. While HTML can be employed without CSS, the opposite is not possible. CSS is used to define how a specific HTML tag should look like. Furthermore, with CSS, it makes sense to mark html tags with IDs or class-names. Listing 6.2 shows a little example of how CSS can be used to style HTML.

Listing 6.2: CSS example code

```
<head>
  <style>
    body {
      background-color: black;
      color: black;
      font-size: 20px;
    }
    .my-class-name {
      color: yellow;
      font-size: 15px;
    }
  </style>
</head>
<body>

  <div class="my-class-name">
    This text will appear yellow.
  </div>

</body>
```

[3] The Ultimate Guide to CSS: https://skillcrush.com/2012/04/03/css/ (last visited on September 09 2019)

Bootstrap

At a certain point or size of a project, it is not always advisable to rewrite everything over and over again. Therefore, it is quite helpful to use frameworks as it makes software development not only faster but also easier. Essentially, a framework is a set of tools that are frequently used so that they are not rewritten from the ground up for every new project. A CSS framework comes with a collection of CSS stylesheets, ready to be used immediately[4].

For this master's thesis, the popular Bootstrap[5] framework (Version 4.1.3) was used.

BEM

In contrast to Bootstrap, BEM is not a framework or library, it cannot be downloaded or installed. It is a naming convention which is one of many ways to make developing and reading code easier. The best description is provided by the BEM webpage[6] itself:

> "BEM - Block Element Modifier is a methodology that helps you to create reusable components and code sharing in front-end development."

Everyone who starts programming will, at some point, experience that working with others can be rather difficult. This is why naming conventions help to make the whole development process easier for everyone involved, not only for beginners but also for experienced developers. BEM works as the following: Every component of a webpage receives a name, also known as Block, while every sub-component is referred to as an element. With BEM, every sub-component consists of the name of the component itself. Therefore, it is easier to see which components belong to each other and what their relations are. Another purpose of BEM is that components should

[4] https://medium.com/html-all-the-things/what-is-a-css-framework-f758ef0b1a11 (last visited on September 09 2019)

[5] https://getbootstrap.com/ (last visited on September 09 2019)

[6] http://getbem.com/ (last visited on September 09 2019)

be easily reused in other projects without much effort. Listing 6.3 shows how a login component could look like if BEM is applied correctly.

Listing 6.3: BEM example code

```
<head>
  <style>
    .login {
      ...
    }
    .login__username {
      ...
    }
    .login__password {
      ...
    }
  </style>
</head>
<body>

  <div class="login">
    <input class="login__username" ...>
    <input class="login__password" ...>
  </div>

</body>
```

6.1.4 Javascript

While HTML and CSS are used for static web pages, JavaScript can be used for dynamically changing the structure by adding, moving or deleting elements. Or in other words: JavaScript allows the creation of interactive elements (King, 2019).

Stack Overflow's 2019 Developer Survey Results considers JavaScript as the most popular programming language[7]. This is likely due to the fact that it is compatible with all popular browsers and can be used for both, front-end and back-end development with Node.js (vidushi_dwivedi, 2019).

[7] https://insights.stackoverflow.com/survey/2019#most-popular-technologies (last visited on September 09 2019)

jQuery

jQuery is a JavaScript library, which makes the use of JavaScript much easier[8]. One of the biggest advantages of using it is the cross-browser compatibility, which is why it was chosen for the implementation of the learning objects.

6.1.5 Text-To-Speech (TTS)

Text-To-Speech is a technology that reads text on computers. This technology can be used on almost every browser on a computer, smartphone and tablet. Bidelman, 2019 provides an example on how to use this technology. According to him, three lines of code are enough to let the computer read a text as shown in Listing 6.4

Listing 6.4: Text-To-Speech example code

```
var msg = new SpeechSynthesisUtterance();
msg.text = 'Hello world!';
window.speechSynthesis.speak(msg);
```

6.1.6 PHP

PHP, the acronym for Hypertext Preprocessor, is an open source scripting language created for web development, which can also be inserted into HTML[9].

Listing 6.5 shows an typical example for a PHP script and its use within a HTML file.

[8] https://www.w3schools.com/jquery/jquery_intro.asp (last visited on September 09 2019)

[9] https://www.php.net/manual/en/intro-whatis.php (last visited on September 09 2019)

Listing 6.5: PHP example code

```
<!DOCTYPE html>
<html>

  <head>
    <title>example</title>
  </head>

  <body>
    <?php
      echo "Hello world!";
    ?>
  </body>

</html>
```

Zend-Framework

A popular PHP framework for the development of web applications is Zend Framework (For the creation of the learning objects, version 3 was used). The framework already has a number of tools that do not have to be developed from the ground up. Among others, it entails tools such as authentication, encryption or translations, all of which can be found on the Zend web page[10].

6.2 Development Tool (IDE)

To develop a web application, all that is needed is a simple text editor. But nowadays, there are numerous tools to increase productivity and simplify the programming process. One of such is a so-called IDE[11], which is more than just a text editor. Solutions, 2019 describes it as a ”software which provides a set of necessary tools in one package”.

According to Technostacks, 2018 the ten best PHP Development tools are:

[10] https://framework.zend.com/learn (last visited on September 09 2019)

[11] Integrated Development Environment

1. PHPStorm[12]
2. Eclipse PDT[13]
3. NetBeans[14]
4. Sublime Text 3[15]
5. NuSphere[16]
6. Zend Studio[17]
7. Atom[18]
8. Komodo[19]
9. Aptana Studio[20]
10. Cloud 9[21]

PHPStorm was also used in the development of the interactive learning objects.

6.3 API and Documentation

6.3.1 API

The essential purpose of an Application Program Interface (short API) is to allow applications to talk to each other. APIs are used by almost every app on a smartphone such as Facebook, WhatsApp or any weather app (MuleSoft, 2019).

An API was created for this master's thesis too, which is an important part of every exercise. In the long term, the implemented learning objects should not only be available on the web but also on smartphones and tablets. Future

[12] https://www.jetbrains.com/phpstorm/ (last visited on September 09 2019)
[13] https://www.eclipse.org/pdt/ (last visited on September 09 2019)
[14] https://netbeans.org/ (last visited on September 09 2019)
[15] https://www.sublimetext.com/3 (last visited on September 09 2019)
[16] http://www.nusphere.com/ (last visited on September 09 2019)
[17] http://www.zend.com/en/products/studio (last visited on September 09 2019)
[18] https://atom.io/ (last visited on September 09 2019)
[19] https://www.activestate.com/komodo-ide (last visited on September 09 2019)
[20] http://www.aptana.com/ (last visited on September 09 2019)
[21] https://aws.amazon.com/cloud9/ (last visited on September 09 2019)

implementation on mobile devices of the same learning objects can use the API to retrieve settings for each exercise and send user statistics to the server. This way, learning analytics measurements can be improved. Nowadays there are more children who have access to mobile devices than computers. Therefore, it is useful to also implement these exercises on mobile devices. As a result, pupils can learn and improve their skills wherever and whenever they wish to.

6.3.2 Documentation with Apiary

Every API is hardly usable if it is not documented. Thus, Apiary[22] was used during the implementation process to document all of the API's end points.

The documentation contains information for all learning objects. For example, how to retrieve the names of all exercises, obtain settings of a specific exercise or how to retrieve logged data of a specific exercise session.

6.3.3 IDeRBlog Exercises API

The following gives an overview of all API calls for each exercise:

API calls for Einsetzen (*insert*) exercise: see Table 6.1

API calls for Fehlerstellen (*fault locations*) exercise: see Table 6.2

API calls for Merkwörter (*cue words*) exercise: see Table 6.3

API calls for Zuordnen (*assign*) exercise: see Table 6.4

API calls for Glücksrad (*wheel of fortune*) exercise: see Table 6.5

General API calls: see Table 6.6

[22] `https://apiary.io` (last visited on September 09 2019)

URL	Description
/einsetzen/api/exercises	Get data of all exercises
/einsetzen/api/exercises/:id	Get data of a specific exercise
/einsetzen/api/exercises/:id/sentences	Get data of all sentences of an exercise
/einsetzen/api/sentences	Get data of all sentences (of all exercises)
/einsetzen/api/sentences/:id	Get data of a specific sentence
/einsetzen/api/check-status	Information if exercise is online and accessible
/einsetzen/api/log	Get and Set logging information
/einsetzen/api/log/:user-id	Get and Set logginf information of an user

Table 6.1: Overview of API calls for 'Einsetzen'

URL	Description
/fehlerstellen/api/exercises	Get data of all exercises
/fehlerstellen/api/exercises/:id	Get data of a specific exercise
/fehlerstellen/api/exercises/:id/sentences	Get data of all sentences of an exercise
/fehlerstellen/api/exercises/:id/justifications	Get data of all default justifications of an exercise
/fehlerstellen/api/sentences	Get data of all sentences (of all exercises)
/fehlerstellen/api/sentences/:id	Get data of a specific sentence
/fehlerstellen/api/check-status	Information if exercise is online and accessible
/fehlerstellen/api/log	Get and Set logging information
/fehlerstellen/api/log/:user-id	Get and Set logginf information of an user

Table 6.2: Overview of API calls for 'Fehlerstellen'

URL	Description
/merkwoerter/api/exercises	Get data of all exercises
/merkwoerte/apir/exercises/:id	Get data of a specific exercise
/merkwoerter/api/check-status	Information if exercise is online and accessible
/merkwoerter/api/log	Get and Set logging information
/merkwoerter/api/log/:user-id	Get and Set logginf information of an user

Table 6.3: Overview of API calls for 'Merkwörter'

URL	Description
/zuordnen/api/exercises	Get data of all exercises
/zuordnen/api/exercises/:id	Get data of a specific exercise
/zuordnen/api/check-status	Information if exercise is online and accessible
/zuordnen/api/log	Get and Set logging information
/zuordnen/api/log/:user-id	Get and Set logginf information of an user

Table 6.4: Overview of API calls for 'Zuordnen'

URL	Description
/gluecksrad/api/exercises	Get data of all exercises
/gluecksrad/api/exercises/:id	Get data of a specific exercise
/gluecksrad/api/check-status	Information if exercise is online and accessible
/gluecksrad/api/log	Get and Set logging information
/gluecksrad/api/log/:user-id	Get and Set logginf information of an user

Table 6.5: Overview of API calls for 'Glücksrad'

URL	Description
/api/check-status	Information which exercises are online and accessible
/api/words	Get a list of all default words
/api/grapheme	Get a list of all graphemes

Table 6.6: Overview of general API calls

6.4 Code Structure

As mentioned earlier in this chapter, this web application was created with the Zend Framework[23], which uses modules for organising a project[24]. To implement all five learning objects, the web application for this master's thesis was divided into fives modules with additional two modules for organisational purposes:

1. ApplicationInterface
2. ApplicationLibrary
3. Einsetzen *(insert)*
4. Fehlerstellen *(fault locations)*
5. Merkwoerter *cue words*
6. Zuordnen *(assign)*
7. Gluecksrad *wheel of fortune)*

There are five modules for each individual learning object (or exercise), which are also named the same. All of these modules are built in the same way and have a similar code structure. For example, there are services for managing what sentences or words are displayed to the user, services for retrieving and logging data as well as services for displaying some statistical information.

The ApplicationLibrary was mainly used to avoid code duplication. For instance, user points are managed in this modules (see description in Chapter 5), as well as adding and removing exercises for each module.

While all modules can be used separately, the ApplicationInterface was needed to hold all learning objects together. The homepage of this web applicaiton is managed in this module, all settings can be set here by administrators or teachers. Also, authentication is handled here, which is often an important part of any application.

The following shows how a basic stricture looks like for each module:

[23] https://framework.zend.com/learn (last visited on September 11 2019)

[24] https://framework.zend.com/manual/2.4/en/user-guide/modules.html (last visited on September 11 2019)

```
Modulename
├── src
│   ├── ApiException
│   │   └── ApiException.php
│   ├── Controller
│   │   ├── Factory
│   │   │   ├── ExerciseControllerFactory.php
│   │   │   └── ...
│   │   ├── ExerciseController.php
│   │   └── ...
│   ├── Entity
│   │   ├── Exercise.php
│   │   └── ...
│   ├── Helper
│   │   ├── ExerciseHelper.php
│   │   └── ...
│   └── Service
│       ├── Factory
│       │   ├── ExerciseManagerFactory.php
│       │   └── ...
│       ├── ExerciseManager.php
│       └── ...
├── test
│   └── Service
│       ├── ExerciseManagerTest.php
│       └── ...
└── view
    ├── modulename
    │   ├── exercise
    │   │   └── index.phtml
    │   └── index
    │       └── index.phtml
    └── layout
        └── layout.phtml
```

7 Evaluation and Results

This chapter describes the evaluation process of the interactive learning objects. Apart from providing the evaluation and interpretation of the given questionnaire, it also gives an insight to the data analysis. The first subchapter outlines the environment of the evaluation and the questions that were asked on the questionnaire to find out how the children got along. The second subchapter explains the results of the questionnaires, but also gives an insight into the analysed data.

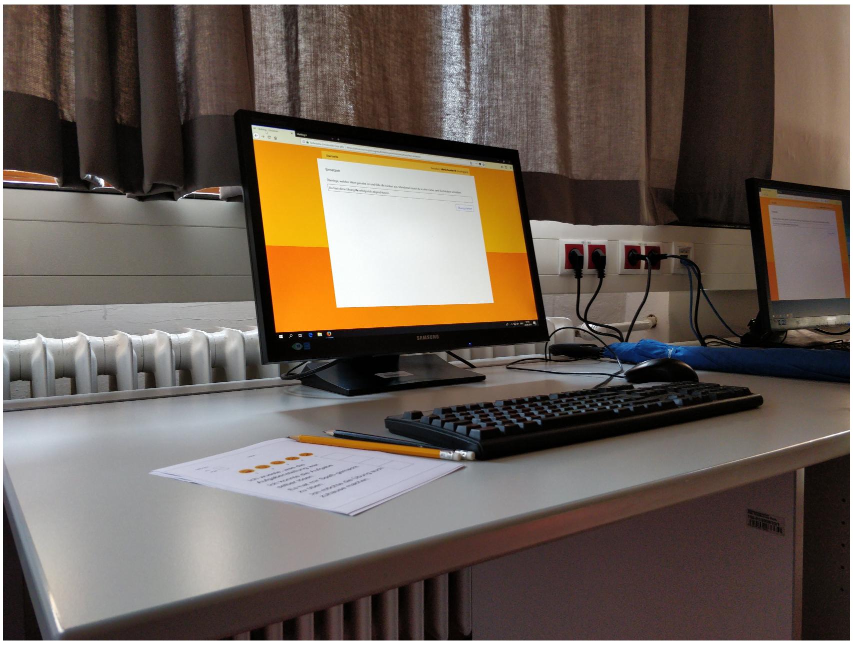


Figure 7.1: A computer of the classroom, used for the usability test

7.1 Evaluation Setup and Procedure

The evaluation took place on the 25th of June 2019 in an Austrian elementary school called ”Praxisvolksschule der Pädagogischen Hochschule Steiermark”[1]. In total, 21 pupils of a 3rd grade school class have been present to participate in the evaluation. There was a distribution of 11 girls and 10 boys. This information, however, had no impact on the evaluation since there were no distinctions made between the sexes. A time window of 08:00 to 09:30 on the evaluation day was agreed upon with Angelika Kornberger, BEd MA[2] with an extra hour in the afternoon if needed, which, as it turned out, was not necessary in the end. Furthermore, the available computer room of the school only had 15 computers (see Figure 7.1). As a consequence, some students had to work together in groups of two. Additionally, a questionnaire was created which had to be completed after each exercise. Due to the narrow time window and also to avoid chaos when completing the questionnaires, pencils were provided in addition, which the students ultimately were allowed to keep as a small reward for participating.

Unfortunately, not all of the five exercises could be evaluated due to some errors that were discovered shortly before the evaluation had started. Some features were only tested in Google's Chrome browser[3], but the school only had two browsers available, Mozilla Firefox[4] and Microsoft Edge[5]. The exercise Glücksrad (*wheel of fortune*) could not be evaluated because of a wrong configuration of the Text-To-Speech feature, which had the side effect that the speech output was in English. The Text-To-Speech functionality only worked in Google's Chrome browser. All exercises were evaluated in the Mozilla Firefox browser with the exception of the ”Zuordnen (*assign*) (Drag&Drop)” exercise, which was evaluated in the Microsoft Edge browser.

Also to test the different levels of difficulty two instances with different

[1] https://pvs.phst.at/ (last visited on September 09 2019)

[2] Teacher of the 3A school class https://pvs.phst.at/team/ (last visited on September 09 2019)

[3] https://www.google.com/intl/de/chrome/ (last visited on September 09 2019)

[4] https://www.mozilla.org/de/firefox/new/ (last visited on September 09 2019)

[5] https://www.microsoft.com/de-at/windows/microsoft-edge (last visited on September 09 2019)

settings of the Merkwörter (*cue words*) exercise were created. An easy mode without any time pressure and a hard mode with time pressure.

The questionnaire (see Figure 7.3), which was prepared a few days in advance, contained a total of four questions[6] (see Figure 7.2):

1. **I knew what the task was.**
   Ich wusste, was die Aufgabenstellung war.

2. **I could solve the task myself.**
   Ich konnte die Aufgabe selber lösen.

3. **It was fun to practice.**
   Es hat mir Spaß gemacht zu üben.

4. **I also want to do the exercise at home.**
   Ich möchte die Übung auch zuhause machen.

Each question should have been rated by a student from one to five after completing each exercise. Alternatively, smileys were also allowed to be drawn. Figure 7.2 shows the smileys used for this purpose.

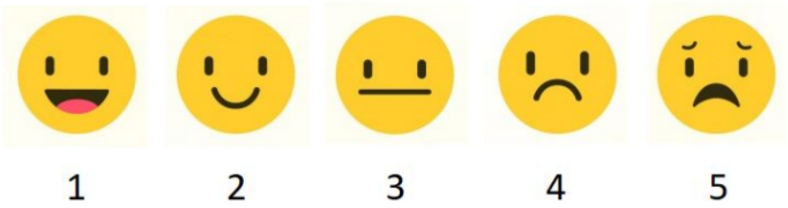


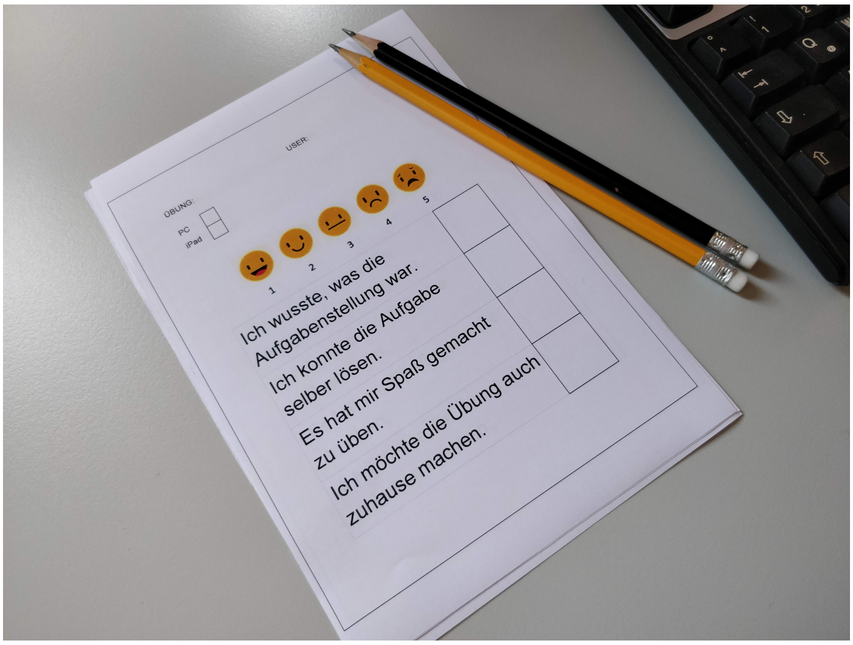
Figure 7.2: Smileys used for ratings (Agarwal, 2019)

To ensure a smooth evaluation process, all exercises needed to be completed within the 1.5 hour time frame, where all pupils had to be kept busy at the same time. Therefore, it was agreed upon to run each exercise on three computers simultaneously; except for the Merkwörter (*cue words*) (hard mode) exercise, which had run on four devices. The children had to move to

[6] This four questions will be also be referred to as statements in the following pages.

another device after completing each exercise, since explaining to them how to switch to another exercise on every single computer would have taken too much time. Also, considering how all exercises took the same amount of time, in the meantime, those who finished earlier could do additional exercises of their choice from the IDeRBlog website[7].

All sentences and words for every single exercise on the evaluation day have been prepared in advance by HS-Prof. Mag. Dr. Edtstadler Konstanze and Prof. Mag. Herunter Elisabeth.



USER:

ÜBUNG:

PC □

iPad □

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war. □

Ich konnte die Aufgabe selber lösen. □

Es hat mir Spaß gemacht zu üben. □

Ich möchte die Übung auch zuhause machen. □

Figure 7.3: Usability Questionnaire

[7] https://iderblog.lpm.uni-sb.de/typo3/fuer-erwachsene/uebungsdatenbank-online/ (last visited on September 09 2019)

7.2 Questionnaire Evaluation and Data Analysis

7.2.1 Einsetzen (insert)

The total rating for all four statements can be seen in Table 7.3. Almost all of the students knew immediately what they had to do to complete the exercise successfully. Also, they were able to do everything by themselves. Only one student stated that the exercise could only be completed with the help of someone else. With the exception of a few, everyone gave the first three statements a rating of one and two. Although, most students had fun during the evaluation, not all wanted do repeat it at home. Only just over half of the participants would like to do the exercise at home too, two of them would not consider doing this exercise at home.

The 'Einstzen' exercise consisted of six sentences with at least one gap per sentence. Th following list shows all six sentences, which have been prepared in advance.

1. Am Dienstag fliegen wir nach New York.
   Am D__nstag fl__egen wir nach New York.

2. Am Nachmittag beginnt das Spiel mit dem Anpfiff.
   Am Nachm_ttag beg_nnt das Sp__l mit dem Anpf_ff.

3. Die Kinder finden den Bären gefangen in einem Käfig.
   Die Kinder finden den B_ren gefangen in einem K_fig.

4. Er fährt mit dem Auto.
   Er f_hrt mit dem Auto.

5. Im Sommer schwimmen viele im See.
   Im So__er schwi__en viele im See.

6. Wir hatten am Wochenende viel Spaß im Zoo.
   _ir hatten am _ochenende viel _paß im _oo.

Sentence	Errors Total	Unique Sessions
Am Dienstag fliegen wir nach New York.	10	6
Am Nachmittag beginnt das Spiel mit dem Anpfiff.	10	8
Die Kinder finden den Bären gefangen in einem Käfig.	10	7
Er fährt mit dem Auto.	4	3
Im Sommer schwimmen viele im See.	2	1
Wir hatten am Wochenende viel Spaß im Zoo.	8	5

Table 7.1: Amount of errors made in each sentence of the Einsetzen (*insert*) exercise

Each points of the list above shows the correct German sentence and below it again the same sentence with gaps.

Data Analysis

The average session time was 5.43 minutes, the longest 9.18 minutes and the quickest 2.58 minutes. On average 1.47 sentences had grammar and typing errors.

Out of all 15 sessions, there were only two students (or groups of students) who did not made any mistake at all. Nine of those who made a mistake recognized and corrected it immediately. Two made an error twice. The remaining two repeated several mistakes for at least four times, either because they were guessing the correct spelling or did not really understand how to write a particular word correctly.

Combined together, all six sentences contained fifteen words and of all these words there were only two which the students did not misspelled: *fliegen (fly)* and *schwimmen (swim)*. The most misspelled words were *Anpfiff (kick-off)*, *beginnt (begins)*, *Dienstag (tuesday)*, *Käfig (cage)* and *Wochenende (weekend)*, all of which were mispelled fives five times in total. For more detail, see Table 7.2

Table 7.1 shows all sentences with the total amount of errors.

Word	Amount misspelled
Anpfiff	5
beginnt	5
Dienstag	5
Käfig	5
Wochenende	5
Bären	4
Nachmittag	4
Spaß	4
Wir	4
fährt	3
Zoo	3
Sommer	1
Spiel	1
fliegen	0
schwimmen	0

Table 7.2: Questionnaire results of the Einsetzen (*insert*) exercise

Interpretation

While the children were doing the exercise, many of them thought that it was more about a specific grammar exercise. Therefore, some of the mistakes are for example "Bieren" *(beers)* instead of "Bären" *(bears)*.

The data also shows that some errors should also be seen as typical typing errors. Some kids knew how to write a word but added for example a white space character or entered a character twice where it should not be.

There was also an instance where a child used the past tense instead of present tense. In hindsight, these kind of errors are understandable. The description was, on the one hand not clear enough and on the other hand, the meaning of the sentence itself could be ambiguous.

7.2.2 Fehlerstellen (fault locations)

Even though a description was displayed at the beginning, some children did not immediately know what the exact meaning of the exercise was.

Statement	**Rating**
I knew what the task was.	1.19
I could solve the task myself.	1.57
It was fun to practice.	1.38
I also want to do the exercise at home.	2.14

Table 7.3: Questionnaire results of the Einsetzen (*insert*) exercise

Nevertheless, they stated that they could do the exercise by themselves (see Table 7.6). The results further demonstrate that it was less fun than the previous exercise. The majority would repeat this exercise at home too. By contrast, four children expressed that they would not want to do this exercise at home.

In total, seven sentences have been prepared for this exercise, which was also the most time-consuming of all. All errorneous sentences (errors marked with a thick font) and the corrected versions are listed here:

1. Am **Dinstag fligen** wir nach New York.
   Correct: Am Dienstag fliegen wir nach New York.
   (On Tuesday we fly to New York.)

2. Im Hochsommer müssen **wier** täglich unsere Blumen **gißen**.
   Correct: Im Hochsommer müssen wir täglich unsere Blumen gießen.
   (In midsummer we have to water our flowers every day.)

3. Der **Iegel libt** die **Wise**.
   Correct: Der Igel liebt die Wiese.
   (The hedgehog loves the meadow.)

4. Zu **Miettag giebt** es zu **trienken Mielch**.
   Correct: Zu Mittag gibt es zu trinken Milch.
   (For lunch there will be milk.)

5. Die **Binen fligen** von Blüte zu Blüte.
   Correct: Die Bienen fliegen von Blüte zu Blüte.)
   (The bees fly from flower to flower.)

6. Hast du schon jemals einen **Brif geschriben**?
   Correct: Hast du schon jemals einen Brief geschrieben?
   (Have you ever written a letter?)

7. Ihm **fil** ein, dass er noch ein **Übungsbeispil** lösen muss.
   Correct: Ihm fiel ein, dass er noch ein Übunsbeispiel lösen muss.
   (He remembered that he had to solve another exercise.)

Data Analysis

Out of 15 sessions, the average session time was 8.12 minutes, the longest 12.9 minutes and the quickest 5.02 minutes. There was no child who did everything correctly. At least two mistakes have been made by each pupil. *Brief* (letter) was the only word where the misspelling was explained correctly.

Two types of errors have been recorded for data analysis: Clicking correctly on a wrong word, but giving a wrong justification, and clicking on words which are actually correct. The words which were correctly identified as misspellings and the amount of wrong justifications can be seen in Table 7.4. Some of the wrongly selected words are listed in Table 7.5 (a complete list of all wrong clicks can be seen in the appendix).

Interpretation

From the children's point of view, at the beginning it was not clear enough what the purpose of this exercise was and whether all mistakes were found. Some of them thought there was one mistake per sentence, which is why they were surprised when they realized exercise had not been indicated as being finished yet. Some students did not see or read the text at the top, which explained how many mistakes were found and how many there are still to be found. Two children, who did very well and had fun during the evaluation, also had some proposals for this exercise. They mentioned that it was slightly boring, because all you had to do in this exercise was to click. They would have preferred an option that made it possible for the user to

Words	Amount justifications	Sentence (Nr.)
Igel	13	3
gibt	12	4
wir	10	2
Milch	8	4
trinken	6	4
Dienstag	5	1
Mittag	5	4
Bienen	5	5
fliegen	5	5
gießen	3	2
Übungsbeispiel	3	7
liebt	2	3
fiel	2	7
fliegen	1	1
Wiese	1	3
geschrieben	1	6

Table 7.4: Words in Fehlerstellen (*fault locations*) exercise which have been justified wrongly.

Words	Wrong clicks	Sentence (Nr.)
Blüte	21	8
täglich	15	5
jemals	11	9
Hochsommer	9	5
Ihm	6	10
schon	4	9
müssen	3	5
wir	2	2
New	2	2
Blumen	2	5
du	2	9

Table 7.5: Words in Fehlerstellen (*fault locations*) exercise which have been false clicked on.

Statement	**Rating**
I knew what the task was.	1.47
I could solve the task myself.	1.53
It was fun to practice.	1.68
I also want to do the exercise at home.	2.21

Table 7.6: Questionnaire results of the Fehlerstellen (*fault locations*) exercise

write something, which would thereby add more variety to the exercise.

Reviewing the logged data, it becomes evident that some of the children did not take this exercise too seriously, and simply clicked through all the words to find the misspelled ones. In Table 7.5 it can be seen that the word *Blüte (blossom)* was selected 21 times, 20 of those being from the same user.

7.2.3 Merkwörter (cue words) (Easy)

Compared to the first two exercises, this one was very well received by the children, which can be seen in Table 7.8. All ratings range from 1 to 3, there was no one who struggled with the given task or did not wish do the same exercise at home again.

Unfortunately one questionnaire had to be removed, because a child falsely completed the questionnaire before the actual exercise was done.

Data Analysis

The data shows the average session time was 3.29 minutes, the longest 5.6 minutes and the quickest was 1.78 minutes. Out of 17 sessions, there were seven sessions without any typing errors at all, making this exercise the one with the least amount of mistakes.

Additionally, this exercise had two modes: *read* and *write*. In the read mode the children had to mark all graphemes that they thought would cause problems when writing. Graphemes were only marked in eight different sessions.

The most marked graphemes are listed as the following:

1. **h** in *bezahlen* (marked in 7 sessions)
2. **h** in *ähnlich* (marked in 5 sessions)
3. **h** in *Bahn* (marked in 5 sessions)
4. **h** in *ähnlicheres* (marked in 4 sessions)
5. **ie** in *biegst* (marked in 4 sessions)
6. **ie** in *biegen* (marked in 3 sessions)
7. **ch** in *ähnlicheres* (marked in 3 sessions)
8. **ch** in *ähnlich* (marked in 3 sessions)
9. **ä** in *ähnlich* (marked in 2 sessions)
10. **ä** in *ähnlicheres* (marked in 2 sessions)
11. **s** in *biegst* (marked in 2 sessions)
12. **t** in *biegst* (marked in 2 sessions)
13. **B** in *Bahn* (marked in 1 sessions)
14. **g** in *biegst* (marked in 1 sessions)
15. **z** in *bezahlen* (marked in 1 sessions)
16. **g** in *biegen* (marked in 1 sessions)

From the underlying data one can see that most selected grapheme was **h**, or, as the case with the words that were provided in the exercise, also known as the 'silent h'. This was directly followed by **ie** and **ch**.

As mentioned in the beginning of the data analysis section, there were ten sessions where spelling mistakes were made. Table 7.7 shows a list of all the different types of spelling errors in the write mode of this exercise.

Interpretation

The simplicity behind the task is probably the main reason why this exercise was received quite well by the students. There were no long sentences or too many words that could have potentially overwhelmed the children. Also, the exercise manages to avoid monotony by offering a balance between typing and clicking.

Word	User Input
ähnlicheres	ähnlichen ähnlicher ähnlichern ähnlicherres ähnliches änlicheres änlicheres CH
ähnlich	ähndlieh ähnlicheres änhlich (2x) änlich
Bahn	bahn BAHN
bezahlen	bezahalen
biegst	Biegst
biegen	beginnen bieken
Bahn	bahn

Table 7.7: Differnet types of spelling mistakes in Merkwörter (*cue words*) Easy exericse

Statement	Rating
I knew what the task was.	1.41
I could solve the task myself.	1.41
It was fun to practice.	1.35
I also want to do the exercise at home.	1.71

Table 7.8: Questionnaire results of the Merkwörter (*cue words*) (Easy) exercise

Statement	Rating
I knew what the task was.	1.74
I could solve the task myself.	1.53
It was fun to practice.	1.89
I also want to do the exercise at home.	2.47

Table 7.9: Questionnaire results of the Merkwörter (*cue words*) (Hard) exercise

7.2.4 Merkwörter (cue words) (Hard)

Although the *hard* mode had the same amount of words as the easy mode, it was not as well received in comparison. In Table 7.9 one can see that even though the task description was somewhat unclear for the students, many of them could complete the exercise regardless. Additionally, it was considered as the exercise that was the least fun for everyone and also the exercise that many of the students would not repeat at home.

The slightly worse ratings can likely be justified by the fact that this exercise had a time limitation, 20 seconds for *reading* and 40 seconds for *writing*. Also, unlike the case with the previous exercise, this time the children did not have to mark graphemes which they believed to cause problems when writing. However, instead of displaying one word after another, now two words have been displayed at the same time to slightly increase the difficulty level.

Data Analysis

For this exercise, 24 sessions have been recorded in total, because four computers have been used simultaneously and therefore some of the kids did this exercise more than once. Each of the other exercises has been tested on three computers at the same time.

The average session time was 1.94 minutes, the longest 3.1 minutes and the quickest 0.68 minutes. There was not a single session where all words were spelled correctly, since everyone made at least two spelling mistakes. These can also be seen based on the average error rate of 4.29 out of 6 words for every session. The total amount of wrongly typed words can be seen in

Word	Amount misspelled
ähnlicheres	20
bezahlen	20
ähnlich	18
Bahn	16
biegst	15
biegen	14

Table 7.10: Words used in the Merkwörter (*cue words*) (Hard) exercise and how often they were misspelled.

Word	Wrong user inputs
ähnlicheres	*nothing (10)*, ähnlicheres (3), ä *(2)*, ähnlich, ähnlichere, aniichere, änl
bezahlen	*nothing (7)*, be *(3x)*, beh, betzahl, Betzahl, beza *(3x)*, bezahealen, bezahl, bnez
ähnlich	*nothing (7)*, ählich, ähnlichere, Än, änhlich, Änlhich, änli, änlich *(4)*, Blege
Bahn	*nothing (5)*, Ba *(2)*, bahn *(5)*, bananih, banh, BANH, bezahlen
biegst	*nothing (5)*, beginst, beh, Bie, bie *(2)*, bieg, biegen, biegs, biegst bezahlen, bigst
biegen	*nothing (6)*, begien, bezahlen, bi, biegen, Biegen *(2)*, biegst, biehge, bigen

Table 7.11: Words used in the Merkwörter (*cue words*) (Hard) exercise with wrong user inputs.

Table 7.10, while the different user inputs can be found in Table 7.11

Interpretation

From the questionnaires, one can read that this exercise has not been received as well as all the others, but even more can be read from the analysed data. In both exercises (Merkwörter (*cue words*) easy and hard mode) both spelling errors and typos were made, but the number of errors differs significantly. The average error rate of 4.29 out of 6 words shows clearly how difficult this exercise was for the kids. At first glance, if one does not consider the time limitations, the given data makes little sense comparing

Statement	Rating
I knew what the task was.	1.33
I could solve the task myself.	1.06
It was fun to practice.	1.00
I also want to do the exercise at home.	1.22

Table 7.12: Questionnaire results of the Zuordnen (*assign*) (Drag&Drop) exercise

this exercise with the one before. Many input fields have been left blank and most words were not even spelled out completely.

This shows that the time limitations of 20 seconds for reading and 40 seconds for writing was to short for the children at that age, since most are not used to keyboards and need a bit more practice. Nevertheless, during the evaluation, some of the females students stated that it was too hectic for them, whereas the male students remarked that they had enjoyed it.

7.2.5 Zuordnen (assign) (Drag&Drop)

This was by far the most popular exercise of all. Every single student appeared to have had fun, as they all gave the statement "It was fun to practice", which was the best grade. With the exception of one, all stated that they could solve everything by themselves. Also, almost everyone wanted to do this exercise at home as well. On the basis of these ratings, one can see that there is still room for some improvements so that the task instructions become more clear to the user (see Table 7.12).

All words could be assigned to the following three categories:

1. Dehnungs-h (stummes h)
2. langes-i
3. Doppelvokal (2 Selbstlaute)

Data Analysis

19 sessions have been recorded with an average session time of 2.35 minutes, 3.95 minutes was the longest and 0.63 minutes the shortest recorded session.

Word	Wrong assignments
biegst	7
ähnlich	6
Boote	5
biegen	5
bohren	4
Boot	3
Bahn	3
bezahlen	3
Brief	2
ähnlicheres	2

Table 7.13: Differnet types of spelling mistakes in Merkwörter (*cue words*) Easy exericse

On average, every child made 2.11 mistakes, though five of them did not make any mistakes at all.

For this drag & drop exercise, ten words had to be allocated to the correct categories. It was possible to either select all or none categories and each word was displayed together with all the possible categories. A word could be assigned as well as unassigned to each category. Furthermore, there were also no time limitations, meaning that the children had enough time to think about the right categories.

Table 7.13 shows all words and how often they have been mapped incorrectly.

Interpretation

Simplicity and user interaction were probably the key factors that made this exercise the best rated one. One cannot only detect from the questionnaire that the exercise was well received, but also from the data which shows an average error rate of 2.11 (out of ten total words).

	Statement 1	Statement 2	Statement 3	Statement 4
Einsetzen (*insert*)	1.19	1.57	1.38	2.14
Fehlerstellen (*fault locations*)	1.47	1.53	1.68	2.21
Merkwörter (Leicht) (*cue words easy*)	1.41	1.41	1.35	1.71
Merkwörter (Schwer) (*cue words hard*)	1.74	1.53	1.89	2.47
Zuordnen (Drag&Drop) (*assign*)	1.33	1.06	1.00	1.22
Average	**1.43**	**1.42**	**1.46**	**1.95**

Table 7.14: Overall Summary of Questionnaires

7.2.6 Summary

An overall summary and comparison of questionnaires for all five exercises is displayed in Table 7.14. The data shown in this chapter is only a small part of what could be read out from the logged user inputs. Usability is a key factory for the success of these implemented learning objects. The better the children understand what they have to do to complete the tasks, the better they will perform and the more likely they will enjoy an exercise. When analyzing this data, it should be noted that obviously not all children are familiar with the use of a keyboard. Many of the students seem to like more interactive tasks, as opposed to solely clicking around. Also, exercises under time pressure should be well thought out by the teaching staff.

The logged data shows different numbers of sessions, for which there are two reasons. First, not all exercises take the same amount of time to complete and some students were faster than others. This explains why some of them simply repeated an exercise. Second, due to the fact that the classroom had an odd number of computers and all children had to be kept busy, one exercise was evaluated on four and all others on three computers simultaneously.

8 Summary and Future Work

This chapter is divided into two parts, Summary and Future Work. First, the summary section will give an abstract of the implemented learning objects, what the differences are and how they can be used in different variations for language acquisition. Next, the evaluation process will be outlined and what was learned from it. Finally, the section future work will give an outlook on what learning objects can do and what can be learned from the evaluation for future work.

8.1 Summary

Five different learning objects have been implemented, which were split into two categories: ’standard exercises’ and ’database-oriented exercises’.

There are two standard exercises:

1. Einsetzen *(insert)*
2. Fehlerstellen *(fault locations)*

and three database-oriented exercises:

1. Merkwörter *(cue words)*
2. Zuordnen *(assign)*
3. Glücksrad *(wheel of fortune)*

Einsetzen *(insert)* is a typical cloze text exercise. Sentences can be created where the user has to fill in all gaps. A gap can be a letter or a whole word, which can be set by the teacher.

Fehlerstellen *(fault locations)* is a slightly more complicated exercise. Sentences can be created too but instead of gaps, there are misspelled words that have to be identified by the user. If a misspelled word was found, the user then has to justify why it is spelled a certain way.

Merkwörter *(cue words)* is a very simple exercise. First one or more words are displayed for a certain period of time before they disappear and one or more input fields appear, where the user has to write the same word(s) again.

Zuordnen *(assign)* is split into two variations: 'Drag&Drop' and 'Writing'. In the Drag&Drop version, one word is displayed together with two to three categories. With the help of drag & drop, the word needs to be assigned to the correct category. However, in the 'Writing' option, the words are displayed first for a period of time before disappearing again. Next, two to three input fields appear, each labelled with a category. Finally, the word needs to be written into the corresponding input field of the correct category.

Glücksrad *(wheel of fortune)* is structured similarly to the Merkwörter *(cue words)* exercise, though there is one difference: The word is hidden and spoken by the computer, after which the user is expected to write it in the displayed input field.

During the evaluation it became apparent that descriptions are essential for certain exercises, considering how not all children knew immediately what the given task description for each exercise was. This can be seen in Table 7.14 where some children stated that they didn't need any help although they did not know what the task was in the beginning. It could be interpreted that if the task had been clear from the beginning, perhaps the motivation for doing the exercise at home would also be higher. In addition, thorough feedback is important during an exercise. If you take the Fehlerstellen *(fault locations)* exercise as an example, some kids wondered why the exercise was not finished yet, despite them having found an error in all of the sentences. Here, the problem was that there was more than one error in each sentence. For example, some sentences had one error, whereas others had four in total. In conclusion, not only better descriptions but also visual feedback is needed during an exercise.

Two kids stated during the evaluation that the Fehlerstellen *(fault locations)* exercise was too 'boring'. They would like to do a task where they can type or drag something, or in other words: They would like a more interactive task.

There are also some aspects that were not apparent enough during the evaluation, but became clear through analysis of the logged data. One of these is, for example, the Merkwörter *(cue words)* exercise. In order to test the different difficulty levels, versions with and without a time limit were created in advance. From the given data, it became clear that even though choosing the right timing is difficult, it is also very important. When comparing the easy mode (no time limitations) with the hard mode, which has time limitations[1], it becomes evident that everyone had troubles completing all tasks when there was, in fact, a time limitation. Concerning the degree of errors, the majority of those were minor ones in the easy mode. However, in the hard mode, many words were completely incorrect. Additionally, there were considerable more empty input fields. At this age, children are probably not used to the keyboard yet, and rather slow at typing which might be the reason for the amount of errors.

Something that needs to be mentioned is that some kids tried to help each other or looked what the seat neighbor was doing.

8.2 Future Work

After the evaluation, it became clear that there are some usability issues that need to be improved in the future. For example, what might be helpful is more feedback duration but also a small introduction tutorial. Instead of displaying how many mistakes were found, the feedback messages could be more positive, telling the children how well they are progressing so far. By doing so, this might motivate them to continue and do even better.

Further improvements that could be done include better highlighting when a correct word is selected in the 'Fehlerstellen' *(insert)* exercise. Also, there could be a number listed for each sentence, that indicates the amount of errors. For the database-oriented exercises, it might also be interesting to

[1] 20 seconds for reading and 40 seconds for writing before a word disappears or the next appears

allow the user to decide in advance how long the time limitation should be.

Since these learning objects were made for a browser, making them available on mobile devices might also be of interest for potential users. Expanding the ways in which these exercises could be accessed, the desire and curiosity for learning at home on one's smartphone or tablet might be increased.

By analyzing the given user data, teachers can create more exercises which can be added to this web application.

One aspect, which was not actually the main goal of the implementation but is nonetheless of relevance for future work, is the usability of the settings interfaces for each exercise. Before the evaluation took place, the teachers were instructed on how to create individual exercises. Meanwhile, it became clear that not only the execution of the exercise was important, but also the creation of the exercises for teachers. Aiding teachers with their work by creating better usability and design will simultaneously help the students.

Appendix

Analysis documents of the individual exercises (German)

Übung Einsetzen

Beschreibung

Die erste Übung für den Iderblog wird eine Einsetzübung werden. Lehrer / Administratoren können für jede Übung mehrere Sätze erstellen bzw. bearbeiten und bei jedem Satz ein oder mehrere Buchstaben festlegen, die für den User/Schüler bei der Übung nicht sichtbar sind. Bei der Übung müssen die korrekten Buchstaben, in ein Eingabefeld, eingetragen werden. Der User/Schüler bekommt nach jedem Satz sofort Feedback und weiß ob die Eingabe richtig oder falsch war. Auch eine Fortschrittsanzeige soll informieren wieviele Sätze pro Übung zu lösen sind bzw. bisher gelöst wurden. Nach jeder absolvierten Übung wird dem User/Schüler angezeigt ob er sich verbessert hat (Kleine Statistik, welche Wörter/Sätze waren falsch, bei welchem Wort hat man die meisten Fehler gemacht, etc.) Da die Buchstaben einzeln festgelegt werden können, ist man somit auch in der Lage mehrere Varianten von Einsetzübungen zu erstellen. Varianten könnten z.B. “Groß- und Kleinschreibung”, “Umlaute”, “s, ss oder ß”, etc. sein.

Funktionen

Nur für Admins bestimmt:

- createExercise()

Lehrer/Admins können Übungen mit einem Namen erstellen. Bsp: “Groß- und Kleinschreibung” - editExercise():
Übungen können umbenannt werden.

- createSentence():
  Einen Satz erstellen. z.B. “Das Lernen macht Spaß”
- editSentence()
  Einen bestehenden Satz bearbeiten. Ein Satz darf nur dann bearbeitet werden, wenn dieser noch nicht freigegeben wurde.
- releaseSentence()
  Einen bestimmten Satz für Übungen freigeben. Nach Freigabe kann dieser nicht mehr bearbeitet werden, weil sonst nachträgliche Änderungen die Auswertung/Statistik verfälschen könnte.
- setSentenceGaps()
  Setzt die Position(en) fest, an der eine Lücke sein soll. Bsp.: Satz: “Das Lernen macht Spaß”, Gaps: 4, 17 Ergebnis: “Das _ernen macht _spaß” Diese Funktion kann nur aufgerufen werden solange der Satz nicht freigegeben wurde.
- getStats()
  Gibt Auswertungen/Statistiken aller User / Übungen zurück.

Für User:

- getExercises()
  Gibt eine Liste von allen Übungen zurück (zB “Groß- und Kleinschreibung”, “Umlaute”, . . .), in der sowohl Name als auch ID der Übung enthalten ist.
- getExercise()
  Gibt eine bestimmte Übung mit allen enthaltenen Sätzen zurück (mit Positionsangabe der Buchstaben, die eine Lücke sein soll). Text mit Lücke wird im Frontend zusammengestellt.
- logSession()
  Speichert Informationen über die absolvierte Übung, welche Sätze richtig oder falsch ausgefüllt wurden, ob User die Übung abgebrochen oder komplett gemacht hat.
- geUsertStats()
  Gibt Userspezifische Auswertung/Statistik zurück.

Übung Fehlerstellen

Beschreibung

Bei den Fehlerstellenübungen bekommt der User/Schüler mehrere Sätze angezeigt, bei denen alle Fehler gefunden und begründet werden müssen. Ein Lehrer bzw. Administrator kann mehrere verschiedene Übungen für Fehlerstellen erstellen, wobei eine dieser Übungen zB für die "Groß- und Kleinschreibung" und eine Andere für zB "Umlaute" erstellt werden kann. Für jede dieser Übungen können Sätze erstellt und pro Satz Fehlerstellen definiert werden. Eine Fehlerstelle besteht aus mindestens einem Wort. Als Beispiel könnte im Satz "Lernen macht sehr viel Spaß" die Fehlerstelle "viel Spaß" definiert und das Wort "Spaß" klein geschrieben werden. Zusätzlich zu den definierten Fehlerstellen müssen auch Begründungen für alle Fehlerstellen erstellt werden. Dabei kann man Standardbegründungen erstellen, die für alle Fehlerstellen gelten, oder benutzerdefinierte Begründungen für bestimmte Fehlerstellen. Wenn der User/Schüler eine Übung startet, sieht er wie viele Fehlerstellen insgesamt zu finden sind. Bei jedem Klick auf ein korrektes Wort, wird dieses kurz etwas dicker dargestellt und nach kurzen Zeit wieder auf normalen Zustand gestellt werden um dem User/Schüler zu signalisieren, dass dieses Wort korrekt ist. Wenn aber ein falsch geschriebenes Wort angeklickt wird, erscheint ein kleines Pop-Up Fenster mit einer Liste von richtigen und falschen Begründungen. Nachdem alle korrekten Begründungen angeklickt wurden verschwindet das Fenster und der User/Schüler kann die restlichen Fehlerstellen suchen. Sollte aber eine falsche Begründung angeklickt werden, wird dies dem User/Schüler sofort angezeigt. Nach jeder absolvierten Übung wird dem User/Schüler angezeigt ob er sich verbessert oder verschlechtert hat (Kleine Statistik, wie viele Begründungen falsch/richtig waren)

Entities

Exercise:

- Name

Name der Übung
- Url
Teilpfad zur Übung (Bsp. für 'abc' wäre ganze URL '.../fehlerstellen/abc'
- Description
Beschreibung der Übung für die Schüler, damit sie wissen, was genau zu tun ist.
- Sentences
Alle Sätze, die zur Übung gehören.
- Justifications
Begründungen, die für alle Fehlerstellen standardmäßig verwendet werden.

Sentence:

- Text
Text/Satz selbst
- FaulLocations
Positionen aller Fehlerstellen.
- Exercise
Die dazugehörige Übung

Justification:

- Text
Text für die Begründung
- Exercise
Die dazugehörige Übung, bei der die Begründung bei allen Sätzen standardmäßig verwendet werden soll.

Log:

- UserId
User ID
- UserRole
Userrole zum Zeitpunkt des Logs (Lehrer, Schüler, Admin)
- App
Web, iOS oder Android
- State

‘started’ Übung wurde von User gestartet, ‘finished’ User hat die Übung abgeschlossen oder ‘check’ User hat Fehler gemacht, diese werden in diesem Log genau beschrieben.

- Session
  Zeitstempel des Starts der Übung
- Timestamp
  Zeitstempel von aktuellen Log
- ExerciseId
  ID der dazugehörigen Übung
- InputErrors
  Informationen mit Fehlern, die der User gemacht hat als JSON.

ExerciseManager Methoden:

- getExercise(s)
- deleteExercise
- createExercise
- editExercise

SentenceManager Methoden:

- getSentence(s)
- deleteSentence
- createSentence
- editSentence
- isAlreadyCreatedInExercise
  Überprüft ob der gleiche Satz bereits in der Übung xistiert
- setSentenceFaultLocations
  Fehlerstelle(n) festlegen
- checkManageSetFaultLocations
  Überprüfung ob in der Administrationsoberfläche alleEingaben (zB Position der Fehlerstellen, Begründungen, ...) richtig sind
- isFaultyLocation
  überprüft ob das angeklickte Wort falsch oder richtiggeschrieben ist
- getFaultyWords
  gibt eine Liste aller fehlerhaften Wörter zurück

JustificationManager Methoden:

- getJustification(s)

- deleteJustification
- createJustification
- editJustification
- isAlreadyCreatedInExercise
  Überpüft ob die gleiche Begründung bereits in der Übung existiert

LogManager Methoden:

- getLogs
  Liefert LogDaten, je nach Parameter
- setLogs
  Speichert gegebene LogDaten
- log
  Speichert einen einzigen Log
- logStartFinished
  Start bzw. Ende der Übung loggen
- isAlreadyLogged
  Überprüft ob der gleiche Log bereits geloggt wurde

StatisticsManager Methoden:

API Calls:

- listExercises
  Gibt eine Liste von Übungen zurück
- listExerciseSentences
  Gibt eine Liste aller Sätze einer Bestimmten Übung zurück.
- getExercise
  Gibt alle Informationen einer Übung zurück
- listSentences
  Gibt eine Liste von Übungen zurück
- getSentence
  Gibt all Informationen eines Satzes zurück (zB Position der Fehlertsellen)
- checkUpdate
  Gibt einen Hash-Wert zurück um zu überprüfen ob sich eine Übung(bzw. die Sätze einer Übung) geändert hat.
- getLog
  Gibt Logs zurück (abhängig von gegebenen Parametern)

- postLog
  Speichert gegebene Logs

Übung Merkwörter

Beschreibung

Bei den Merkwörter-Übungen sieht der User/Schüler zunächst ein richtig geschriebenes Wort, wo er die Möglichkeit hat alle Buchstaben (bzw. Grapheme) zu markieren bei denen er glaubt, dass es Schwierigkeiten beim Schreiben geben könnte. Nachdem alle Buchstaben/Grapheme markiert wurden verschwindet das richtig geschrieben Wort entweder nach Zeitablauf oder durch das Klicken auf einen Weiter-Button. Anschließend erscheint ein Input-Feld, wo das Wort nochmal richtig geschrieben werden muss. Wenn das Wort falsch geschrieben wurde, wird sowohl das falsche als auch das richtig geschriebene Wort angezeigt. Ein Administrator hat bei den Einstellungen die Möglichkeit einzustellen, ob falsch geschriebene Wörter erneut angezeigt werden sollen oder nicht.

Entities

Exercise:

- Name
  Name der Übung
- Url
  Teilpfad zur Übung (Bsp. für 'abc' wäre ganze URL '.../merkwoerter/abc'
- Description
  Beschreibung der Übung für die Schüler, damit sie wissen, was genau zu tun ist.
- WordsCount
  Anzahl der Wörter in einer Übung
- CharsClickable

Gibt an ob die einzelnen Buchstaben/Grapheme eines Wortes anklickbar sein sollen.

- ShowWrongInput
  Gibt an ob falsche Usereingaben angezeigt werden sollen.
- TimeReadSeconds
  Dauer in Sekunden wie lang ein Wort angezeigt werden soll.
- TimeWriteSeconds
  Dauer in Sekunden wie lang der User Zeit zum Schreiben hat.
- TimeShowWrongInput
  Dauer in Sekunden wie lange falsche Usereingaben angezeigt bleiben sollen.

Log

- UserId
  User ID
- UserRole
  Userrole zum Zeitpunkt des Logs (Lehrer, Schüler, Admin)
- App
  Web, iOS oder Android
- State
  'started' Übung wurde von User gestartet, 'finished' User hat die Übung abgeschlossen oder 'check' User hat Fehler gemacht, diese werden in diesem Log genau beschrieben.
- Session
  Zeitstempel des Starts der Übung
- Timestamp Z
  eitstempel von aktuellen Log
- ExerciseId
  ID der dazugehörigen Übung
- InputErrors
  Informationen mit Fehlern, die der User gemacht hat als JSON.
- Mode
  Wenn "read" dann sind in InputErrors Angaben über die angeklickten Grapheme, wenn "write" dann steht im InputErrors das falsch eingegebene Wort.

ExerciseManager Methoden:

- getExercise(s)
- deleteExercise
- createExercise
- editExercise

LogManager Methoden:

- getLogs
  Liefert LogDaten, je nach Parameter
- setLogs
  Speichert gegebene LogDaten
- log
  Speichert einen einzigen Log
- logStartFinished
  Start bzw. Ende der Übung loggen
- isAlreadyLogged
  Überprüft ob der gleiche Log bereits geloggt wurde
- getSettings
  Liefer Einstellungen (wieviele Wörter sollen pro Übung angezeigt werden, die Dauer für Zeitablauf, etc.)
- setSettings
  Einstellungen speichern

API Calls:

- listExercises
  Gibt eine Liste von Übungen zurück
- getExercise
  Gibt alle Informationen einer Übung zurück
- checkUpdate
  Gibt einen Hash-Wert zurück um zu überprüfen ob sich eine Übung(bzw. die Sätze einer Übung) geändert hat.
- getLog
  Gibt Logs zurück (abhängig von gegebenen Parametern)
- postLog
  Speichert gegebene Logs

Übung Zuordnen

Beschreibung

Die Zuordnen-Übungen bestehen aus zwei verschiedenen Varianten:

1. Drag&Drop
2. Schreiben

Bei beiden Varianten sieht der User/Schüler zunächst einmal ein Wort, welches zu einer Kategorie zugeordnet werden muss. In der Drag&Drop Variante werden zusätzlich auch die Kategorien angezeigt, wo das Wort hingeschoben werden soll. Je nachdem welches Wort gerade angezeigt wird, ist es möglich, dass mehrere Kategorien richtig sind. Nachdem das Wort zu allen Kategorien zugeordnet wurde, muss man auf "Weiter" klicken um zu sehen ob die Auswahl korrekt war oder nicht. Wenn alles korrekt ist, wird der User direkt zum nächsten Wort weitergeleitet. Sollte die Auswahl aber nicht korrekt sein, wird dem User grafisch die richtigen Kategorien angezeigt. Die Schreiben Variante unterscheidet sich daran, dass zunächst das Wort allein dasteht und der User/Schüler sich das Wort genau ansehen muss. Erst durch einen Klick auf "Weiter" erscheinen alle möglichen Kategorien mit Eingabefeldern. Um die Übung erfolgreich zu lösen, muss das Wort in das/die richtige(n) Eingabefeld(er) eingetragen werden.

Entities

Exercise:

- Name
  Name der Übung
- Url
  Teilpfad zur Übung (Bsp. für 'abc' wäre ganze URL '.../merkwoerter/abc'
- Description
  Beschreibung der Übung für die Schüler, damit sie wissen, was genau zu tun ist.

- WordsCount
  Anzahl der Wörter in einer Übung
- Type
  “drag&drop” oder “write”
- ShowWrongInput
  Gibt an ob falsche Usereingaben angezeigt werden sollen.
- TimeReadSeconds
  Dauer in Sekunden wie lang ein Wort angezeigt werden soll.
- TimeWriteSeconds
  Dauer in Sekunden wie lang der User Zeit zum Schreiben hat.
- TimeShowWrongInput
  Dauer in Sekunden wie lange falsche Usereingaben angezeigt bleiben sollen.

Log:

- UserId
  User ID
- UserRole
  Userrole zum Zeitpunkt des Logs (Lehrer, Schüler, Admin)
- App
  Web, iOS oder Android
- State
  ‘started’ Übung wurde von User gestartet, ‘finished’ User hat die Übung abgeschlossen oder ‘check’ User hat Fehler gemacht, diese werden in diesem Log genau beschrieben.
- Session
  Zeitstempel des Starts der Übung
- Timestamp
  Zeitstempel von aktuellen Log
- ExerciseId
  ID der dazugehörigen Übung
- InputErrors
  Informationen mit Fehlern, die der User gemacht hat als JSON.
- Type
  “drag&drop” oder “write”, je nachdem von welchem Typ die Übung ist, steht in InputErrors etwas anderes drin.

ExerciseManager Methoden:

- getExercise(s)
- deleteExercise
- createExercise
- editExercise

LogManager Methoden:

- getLogs
  Liefert LogDaten, je nach Parameter
- setLogs
  Speichert gegebene LogDaten
- log
  Speichert einen einzigen Log
- logStartFinished
  Start bzw. Ende der Übung loggen
- isAlreadyLogged
  Überprüft ob der gleiche Log bereits geloggt wurde
- getSettings
  Liefer Einstellungen (wieviele Wörter sollen pro Übung angezeigt werden, die Dauer für Zeitablauf, etc.)
- setSettings
  Einstellungen speichern

API Calls:

- listExercises Gibt eine Liste von Übungen zurück
- getExercise Gibt alle Informationen einer Übung zurück
- checkUpdate
  Gibt einen Hash-Wert zurück um zu überprüfen ob sich eine Übung(bzw. die Sätze einer Übung) geändert hat.
- getLog
  Gibt Logs zurück (abhängig von gegebenen Parametern)
- postLog
  Speichert gegebene Logs

Übung Glücksrad

Beschreibung

Bei der Glücksrad-Übung werden dem User/Schüler Wörter vorgesprochen (mittels Text-To-Speech Funktionalität), die dann in ein Eingabefeld geschrieben werden müssen. Zuerst erscheint immer ein Eingabefeld wo gleichzeitig das Wort vorgesprochen wird. Der User/Schüler hat jederzeit die Möglichkeit sich das Wort nochmal anzuhören (durch ein Klick auf "Wort anhören"). Nach Eingabe und Klick auf "Weiter" erscheint entweder das nächste Wort oder eine grafische Darstellung, dass das Wort falsch geschrieben wurde.

Entities

Exercise:

- Name
  Name der Übung
- Url
  Teilpfad zur Übung (Bsp. für 'abc' wäre ganze URL '.../merkwoerter/abc'
- Description
  Beschreibung der Übung für die Schüler, damit sie wissen, was genau zu tun ist.
- WordsCount
  Anzahl der Wörter in einer Übung
- ShowWrongInput
  Gibt an ob falsche Usereingaben angezeigt werden sollen.
- TimeReadSeconds
  Dauer in Sekunden wie lang ein Wort angezeigt werden soll.
- TimeWriteSeconds
  Dauer in Sekunden wie lang der User Zeit zum Schreiben hat.
- TimeShowWrongInput
  Dauer in Sekunden wie lange falsche Usereingaben angezeigt bleiben sollen.

Log:

- UserId
  User ID
- UserRole
  Userrole zum Zeitpunkt des Logs (Lehrer, Schüler, Admin)
- App
  Web, iOS oder Android
- State
  'started' Übung wurde von User gestartet, 'finished' User hat die Übung abgeschlossen oder 'check' User hat Fehler gemacht, diese werden in diesem Log genau beschrieben.
- Session
  Zeitstempel des Starts der Übung
- Timestamp
  Zeitstempel von aktuellen Log
- ExerciseId
  ID der dazugehörigen Übung
- InputErrors
  Informationen mit Fehlern, die der User gemacht hat als JSON.
- Type
  "drag&drop" oder "write", je nachdem von welchem Typ die Übung ist, steht in InputErrors etwas anderes drin.

ExerciseManager Methoden:

- getExercise(s)
- deleteExercise
- createExercise
- editExercise

LogManager Methoden:

- getLogs
  Liefert LogDaten, je nach Parameter
- setLogs
  Speichert gegebene LogDaten
- log
  Speichert einen einzigen Log

- logStartFinished
  Start bzw. Ende der Übung loggen
- isAlreadyLogged
  Überprüft ob der gleiche Log bereits geloggt wurde
- getSettings
  Liefer Einstellungen (wieviele Wörter sollen pro Übung angezeigt werden, die Dauer für Zeitablauf, etc.)
- setSettings
  Einstellungen speichern

API Calls:

- listExercises
  Gibt eine Liste von Übungen zurück
- getExercise
  Gibt alle Informationen einer Übung zurück
- checkUpdate
  Gibt einen Hash-Wert zurück um zu überprüfen ob sich eine Übung(bzw. die Sätze einer Übung) geändert hat.
- getLog Gibt
  Logs zurück (abhängig von gegebenen Parametern)
- postLog
  Speichert gegebene Logs

Bibliography

Academy, Khan (2019). *Khan Academy - Free Online Courses, Lessons & Practice*. URL: https://www.khanacademy.org/ (visited on 05/29/2019) (cit. on p. 14).

Agarwal, Vikrant (2019). *Best Rating Smileys*. URL: https://www.smileysymbol.com/2018/03/best-rating-smileys.html (visited on 07/31/2019) (cit. on p. 65).

Aspalter, Christian, Konstanze Edtstadler, and Susanne Martich (2017). “Recht-/schreibdidaktische Überlegungen für den Einsatz der Internetplattform IDeRBlog.” In: *R&ESOURCE. Bd*, pp. 1–15 (cit. on pp. 1, 20).

Bidelman, Eric (2019). *Web apps that talk - Introduction to the Speech Synthesis API*. URL: https://developers.google.com/web/updates/2014/01/Web-apps-that-talk-Introduction-to-the-Speech-Synthesis-API (visited on 08/26/2019) (cit. on p. 54).

Campbell, John P and Diana G Oblinger (2007). “Academic analytics.” In: *EDUCAUSE* (cit. on p. 7).

Clow, Doug (2012). “The learning analytics cycle: closing the loop effectively.” In: (cit. on pp. 8, 9).

Committee, IEEE Learning Technology Standards (2019). *The Learning Object Metadata standard. Retrieved February 11, 2009, from IEEE Learning Technology Standards Committee Web site*. URL: https://www.ieeeltsc.org/working-groups/wg12LOM/lomDescription (visited on 05/02/2019) (cit. on p. 3).

Dale, Steve (2014). “Gamification: Making work fun, or making fun of work?” In: *Business information review* 31.2, pp. 82–90 (cit. on p. 12).

DNNCorp (2019). URL: https://www.dnnsoftware.com/ (visited on 05/29/2019) (cit. on p. 12).

Ebner, Markus, Martin Ebner, and Konstanze Edtstadler (2016). “Learning Analytics and Spelling Acquisition in German - A First Prototype.” In:

vol. 9753, pp. 405–416. ISBN: 978-3-319-39482-4. DOI: 10.1007/978-3-319-39483-1_37 (cit. on pp. 19, 22).

Ebner, Markus, Konstanze Edtstadler, and Martin Ebner (2018). "Tutoring writing spelling skills within a web-based platform for children." In: *Universal Access in the Information Society* 17.2, pp. 305–323. ISSN: 1615-5297. DOI: 10.1007/s10209-017-0564-6. URL: https://doi.org/10.1007/s10209-017-0564-6 (cit. on p. 20).

Edtstadler, Konstanze, Markus Ebner, and Martin Ebner (2015). "Improved German Spelling Acquisition through Learning Analytics." In: *eLearning Papers* 45, pp. 17–28 (cit. on p. 21).

Ferguson, Rebecca (2012). "Learning analytics: drivers, developments and challenges." In: *International Journal of Technology Enhanced Learning* 4.5/6, pp. 304–317 (cit. on pp. 6, 7).

Gibb, Robert (2016). *What is a Web Application?* URL: https://blog.stackpath.com/web-application/ (visited on 08/26/2019) (cit. on p. 49).

Greller, Wolfgang and Hendrik Drachsler (2012). "Translating learning into numbers: A generic framework for learning analytics." In: (cit. on p. 10).

Gros, Michael et al. (2015). "Schreiben – Rechtschreiben lernen und Lesen mit der Plattform Individuell Differenziert Rechtschreiben mit Blogs – kurz IDeRBlog (www.iderblog.eu)." In: *LA-Multimedia* 2015, pp. 22–24 (cit. on pp. 20, 21).

Hodgins, H Wayne (2002). "The future of learning objects." In: (cit. on p. 4).

Holloway, Suzanne (2018). *Gamification in Education: 4 Ways To Bring Games To Your Classroom.* URL: https://tophat.com/blog/gamification-education-class/ (visited on 05/28/2019) (cit. on p. 13).

Huang, Wendy Hsin-Yuan and Dilip Soman (2013). "Gamification of education." In: *Research Report Series: Behavioural Economics in Action, Rotman School of Management, University of Toronto* (cit. on pp. 14, 16).

Kapp, Karl M (2013). *The gamification of learning and instruction fieldbook: Ideas into practice.* John Wiley & Sons (cit. on p. 13).

Khalil, Mohammad and Martin Ebner (2015). "Learning analytics: principles and constraints." In: *EdMedia+ Innovate Learning.* Association for the Advancement of Computing in Education (AACE), pp. 1789–1799 (cit. on pp. 9–11).

King, Ray (2019). *What Is JavaScript Used For And Why You Should Learn It.* URL: https://www.bitdegree.org/tutorials/what-is-javascript-used-for/ (visited on 08/26/2019) (cit. on p. 53).

Kitchin, Rob and Gavin McArdle (2016). "What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets." In: *Big Data & Society* 3.1, p. 2053951716631130 (cit. on p. 6).

Kumar, Balraj and Parul Khurana (2012). "Gamification in education-learn computer programming with fun." In: *International Journal of Computers and Distributed Systems* 2.1, pp. 46–53 (cit. on p. 13).

LAK11 (2019). *Learning Analytics & Knowledge: February 27-March 1, 2011 in Banff, Alberta*. URL: https://tekri.athabascau.ca/analytics/ (visited on 05/24/2019) (cit. on p. 7).

Lecturio (2016). *10 E-Learning-Tools, die Lehrer alt aussehen lassen*. URL: https://www.lecturio.de/magazin/e-learning-tools/ (visited on 08/27/2019) (cit. on p. 16).

Lehky, Andrea (2016). *Bewerbungsfehler, die den Job kosten können*. URL: https://diepresse.com/home/karriere/5112591/Bewerbungsfehler-die-den-Job-kosten-koennen (visited on 08/26/2019) (cit. on p. 1).

Liebal, Janine and Markus Exner (2011). *Usability für Kids*. Springer (cit. on p. 21).

McGreal, Rory (2004a). "Learning objects: A practical definition." In: *International Journal of Instructional Technology and Distance Learning (IJITDL)* 9.1 (cit. on pp. 4, 5).

McGreal, Rory (2004b). *Online education using learning objects*. Psychology Press (cit. on p. 5).

MuleSoft (2019). *What is an API? (Application Programming Interface)*. URL: https://www.mulesoft.com/resources/api/what-is-an-api (visited on 08/26/2019) (cit. on p. 56).

Nah, Fiona Fui-Hoon et al. (2014). "Gamification of education: a review of literature." In: *International conference on hci in business*. Springer, pp. 401–409 (cit. on p. 12).

O'Donovan, Siobhan, James Gain, and Patrick Marais (2013). "A case study in the gamification of a university-level games development course." In: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. ACM, pp. 242–251 (cit. on p. 13).

Polsani, Pithamber (2006). "Use and Abuse of Reusable Learning Objects." In: *Journal of Digital Information* 3.4. ISSN: 1368-7506. URL: https://journals.tdl.org/jodi/index.php/jodi/article/view/89 (cit. on pp. 3, 4).

Raymer, Rick (2011). *Gamification: Using Game Mechanics to Enhance eLearning*. URL: https://elearnmag.acm.org/featured.cfm?aid=2031772 (visited on 05/29/2019) (cit. on p. 13).

Russom, Philip et al. (2011). "Big data analytics." In: *TDWI best practices report, fourth quarter* 19.4, pp. 1–34 (cit. on p. 6).

School, Viking Code (2019). *What are Mockups?* URL: https://www.vikingcodeschool.com/web-design-basics/what-are-mockups (visited on 08/25/2019) (cit. on p. 24).

Seghroucheni, Yassine Zaoui, Mohammed AL Achhab, and Badr Eddin El mohajir (2014). "An Approach to Create Multiple Versions of the Same Learning Object." In: *International Journal of Emerging Technologies in Learning (iJET)* 9.5, pp. 17–21. ISSN: 1863-0383. URL: https://online-journals.org/index.php/i-jet/article/view/3762 (cit. on pp. 4, 5).

Sheldon, Lee (2011). *The multiplayer classroom: Designing coursework as a game*. Nelson Education (cit. on p. 14).

Solutions, SaM (2019). *Top Ten Embedded Software Development Tools*. URL: https://www.sam-solutions.com/blog/top-ten-embedded-software-development-tools/ (visited on 08/26/2019) (cit. on p. 55).

TechnologyAdvice (2019). *The History of Gamification Infographic: From Stamp to Space*. URL: https://technologyadvice.com/blog/marketing/history-of-gamification-infographic/ (visited on 05/29/2019) (cit. on p. 11).

Technostacks (2018). *10 Best PHP Development Tools That Every Developer Must Explore*. URL: https://technostacks.com/blog/php-development-ides-tools/ (visited on 08/26/2019) (cit. on p. 55).

ThoughtCo (2019). *German Capitalization Rules*. URL: https://www.thoughtco.com/capitalization-in-german-4069437 (visited on 05/14/2019) (cit. on p. 27).

TopHat (2019). *Education Software For Professors & Educators — Top hat*. URL: https://tophat.com/ (visited on 05/29/2019) (cit. on p. 14).

vidushi_dwivedi (2019). *Top 10 Programming Languages of the World – 2019 to begin with...* URL: https://www.geeksforgeeks.org/top-10-programming-languages-of-the-world-2019-to-begin-with/ (visited on 08/26/2019) (cit. on p. 53).

Wiley, David A et al. (2002). *The instructional use of learning objects*. Vol. 1. Agency for instructional technology Bloomington, IN (cit. on p. 4).