Markus Friedl, BSc

Mobile Learning Applications for Android und iOS for German Language Acquisition based on Learning Analytics Measurements

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

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

Co-Advisor
Dipl.-Ing. Markus Ebner

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Ing. Dr. Stefanie Lindstaedt

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

Thanks

At the beginning I would like to thank my supervisor Priv.-Doz. Dipl.-Ing. Dr.techn. Martin Ebner and co-supervisor Dipl.-Ing. BSc Markus Ebner, who supported me during the development of the mobile applications and writing this Master's thesis. Thank you for organizing the numerous meetings and suggestions necessary for the implementation of the applications.

Special thanks to HS-Prof. Mag. Dr. Konstanze Edtstadler and Mag. Prof. Elisabeth Herunter from the University College of Teacher Education Styria for the excellent cooperation, without whose help this project would not have gone so well. Thank you for your efforts and support in considering the exercises and for your engagement during the development of the applications.

I would also like to thank the five children who took time for the user experience test and did it with full commitment and enthusiasm.

Finally, I would like to especially thank my family and friends, who have always actively supported and motivated me throughout my studies.

Kurzfassung

Der Einsatz digitaler Medien in ihren unterschiedlichsten Formen ist heutzutage in der Schule und im Unterricht nicht mehr wegzudenken. Bereits Schülerinnen und Schüler im jüngsten Volksschulalter beherrschen meist den Umgang mit Computern und Tablets. Dieser Umstand wird sich in Zukunft stetig weiterentwickeln und die klassischen Lehr- bzw. Lernmethoden werden immer mehr durch neue technologiebetriebene Methoden ergänzt oder ersetzt. Daher ist es wichtig, dass in Schulen dieser Fortschritt als Bereicherung gesehen wird und solche Medien im Unterricht aktiv eingesetzt werden.

Durch die Digitalisierung verändert sich auch die Zusammenarbeit und das Miteinander von Lehrerinnen und Lehrer und deren Schülerinnen und Schüler, was die direkte Kommunikation bzw. Hilfestellung betrifftt. Genau für diesen Aspekt soll im Zuge dieser Masterarbeit eine mobile Applikation für Android und iOS entwickelt werden, in der verschiedene Übungen verfügbar sind um die deutsche Sprache zu lernen.
Lehrerinnen und Lehrer haben über die IDeRBlog ii-Platform die Möglichkeit, individuell angepasste Übungen für einzelne Personen oder Personengruppen zu erstellen, welche die Schülerinnen und Schüler auf Tablets mit Hilfe der Applikaton lösen können. Durch die Protokollierung aller getätigten Interaktionen wie das Starten bzw. Beenden einer Übung und das Verursachen von Fehlern während dem Lösen einer Übung bekommen Lehrerinnen und Lehrer immer genaue Auswertungen und Feedback über den Lernfortschritt der Schülerinnen und Schüler. Mit der Hilfe solcher Statistiken kann der Lehrinhalt positiv beeinflusst werden, wodurch Lehrerinnen und Lehrer und Lernende profitieren können.
Die umgesetzten Übungsformate in der Applikation wurden von Professoren der Pädagogischen Hochschule Steiermark vorgegben und deren Design wurde in gemeinsamer Arbeit umgesetzt.

Abstract

The use of digital media in their various forms has become indispensable in the field of school teaching. Even pupils of the youngest elementary school age usually master the use of computers and tablets. This circumstance will constantly be developed further in the future and the classical teaching and learning methods will increasingly be supplemented or replaced by new technology-driven methods. Therefore, it is important that this progress is seen as enrichment in schools and is implemented in some way in the teaching process.

Digitization also changes the way teachers and their students work together and interact with each other, which affects the face-to-face communication and assistance. For exactly this aspect, a mobile application for Android and iOS will be developed in the context of this master's thesis, in which various exercises are available to learn the German language.

Teachers can use the IDeRBlog ii platform to create customized exercises for groups of individuals that pupils can solve them on tablets using the application. By logging all interactions such as starting or finishing an exercise and causing errors while solving an exercise, teachers always get accurate evaluations and feedback on pupils' learning progress. With the help of such statistics, teaching content can be positively influenced and teachers and learners can benefit.

The implemented exercise formats in the application were defined by professors of the University College of Teacher Education Styria and their design was implemented in joint work.

Contents

List of Figures

List of Figures

List of Tables

Abbreviations

API Application Program Interface

ART Android Runtime

DEX Dalvik Executable format

DM Data Mining

EDM Educational Data Mining

HAL Hardware Abstraction Layer

HMAC Keyed-Hash Message Authentication Code

ID Identification

IDE Integrated Development Environment

IV Information Visualization

JDK Java Development Kit

JSON JavaScript Object Notation

LA Learning Analytics

LMS Learning Management Systems

OS Operating System

SNA Social Network Analysis

SOAP Simple Object Access Protocol

TEL Technology Enhanced Learning

TU Graz Graz University of Technology

UI User Interface

USP Unique Selling Point

UX User Experience

VLE Virtual Learning Environments

1. Introduction

More and more digital media are being used in teaching instead of classical media, which can lead to new ways of teaching and learning. Already in the 1960s and 70s, the first attempts were made to use computers in teaching with more or less success. At that time, language laboratories and self-learning centers were set up with the help of mini-computers. In the late 1970s, the introduction of computer science education began in secondary schools. With the continuous development of computers, which became more and more widespread and used by more and more people, they were also increasingly integrated into the teaching of schools. (Eickelmann, 2010)

In the meantime, the computer has become an indispensable aid in teaching, and it is already being used in the first class of the elementary school, which already teaches beginners how to use computers. With the introduction of tablets, the use of mobile computers has further revolutionized, opening up completely new possibilities for the use of digital media in teaching. This process is now so far advanced that there are attempts for pure tablet classes. (Prasse et al., 2016) The use of tablets has also opened up a new market for the development of learning applications, as the use of touch gestures is easier to learn compared to mouse and keyboard on computers.

In the context of this master thesis, a mobile application was developed to support learning in a playful way. The IDeRBlog ii application should offer teachers the possibility to write individual and customized tasks for learning the German language, which should be solved by the pupils within the application. By storing and collecting data in a central database, it is possible to generate various statistics from the data and to get information about the performance level of the pupils. This also ensures that teachers always have an overview of the pupils' activities and receive feedback.

Chapter 2 of this master's thesis describes the scientific background and is divided into three categories. First, a possible effect of the use of such digital media is explained and the advantages that can be achieved. Due to the ongoing development of digitization in schools, the term Learning Analytics (LA) is increasingly used in this field. Therefore, in the following section its drivers, the process of LA and a reference model for LA are described to get a small overview of this emerging topic. At the end of the chapter, four mobile applications are listed, which can also be integrated in the area of language learning in order to obtain a comparison of existing applications to the IDeRBlog ii application.

Chapter 3 contains the complete technical implementation of the application, which was developed as a native application for Android and iOS. Since all data are loaded by Application Program Interface (API)s and processed within the application, first the entire user administration is described, which is used in all applications developed by Graz University of Technology (TU Graz). Then the API for loading the exercise data and the handling of the log and statistics data is described. At the end of the first section the different test types are described, which are used by default in the software development. The following two sections describe the development and test environment for the two mobile Operating System (OS)s Android and iOS. The last point of this chapter describes the developed application and gives more information about the different exercise formats and their implementation. At this point it should be mentioned that the application serves as first prototype, which should be extended in the future.

After the description of the application, chapter 4 deals with the evaluation of the application, which is subdivided into the preparation, the procedure and the result of the evaluation process. The evaluation was performed by five children between the ages of eight and eleven using a User Experience (UX) test.

2. State of the art

The use of digital media in schools is constantly increasing. (Robin, 2008) More and more branches in the field of teaching are being covered as well. (McLellan, 2007) The current use and possible influencing factors of digital media in teaching are described in this chapter. With the digitization process LA is becoming more and more important and different models are presented. In the last section of this chapter different existing mobile applications for German language acquisition are explained in more detail.

2.1. Digital Media in Teaching

This section deals with the effects of the use of digital media in teaching and their influencing factors. Furthermore, a possible added value and an advantage of digital media is discussed.

2.1.1. Effect of Digital Media

In order to answer the question of different effects of digital media in educational institutions, it is useful to determine potential factors which influence such effects. Teaching can be seen as an interaction between teachers and learners. The learning activities of pupils are simulated and supported by specific teaching activities by teachers. (Tulodziecki, Herzig, and Blömeke, 2017) Digital media can be integrated into the associated teaching and learning processes as a didactically instrument and an educationally object. From the didactically point of view digital media represent learning materials

or tools. The effect of digital media in school teaching can be divided into following factors (Herzig, 2014):

- The digital media and media offerings themselves
- The teaching process which the media offerings are integrated
- The actors directly involved in teaching (teachers and learners)

According to (Herzig, 2014) four categories can be extracted from this list above, whose context is shown in Figure 2.1 and then described in more detail.

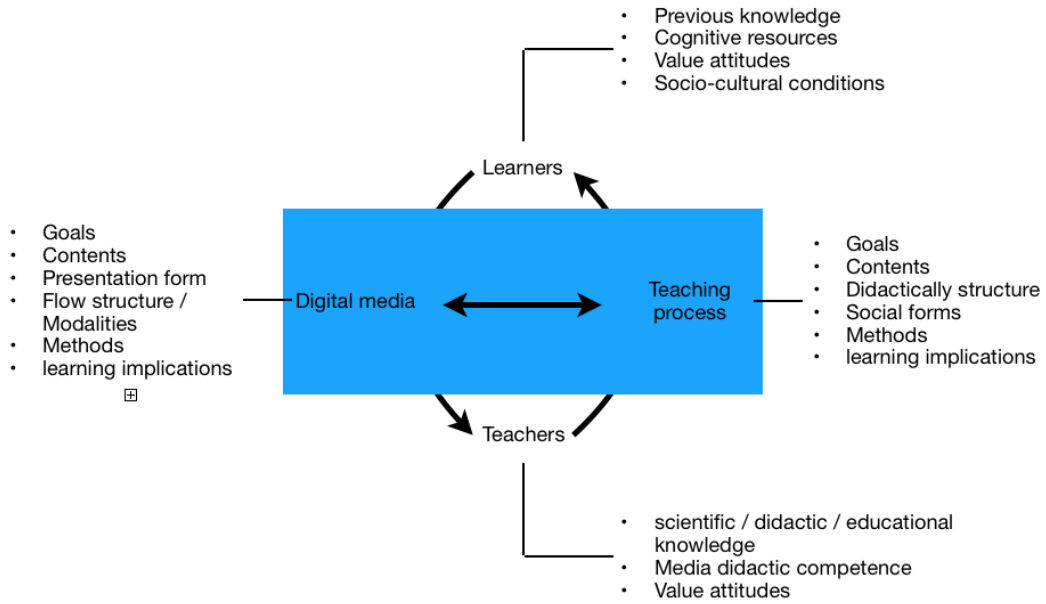


Figure 2.1.: Digital media: Influencing factors (own representation based on (Herzig, 2014))

- Digital media: A digital media offer can be described by various features and characteristics. The effect of an offer can have meaning individually or in interaction with others. An example for such a media offer can be a learning program which is characterized by specific content, objectives, forms of presentation, flow and navigation structures, characteristics of inactivity, types of coding, sensory modalities

addressed, design techniques or learning theoretical implications. Depending of the offer, some of these characteristics may have greater or lesser significance. The sequence, interaction structures or the forms of presentation would be particularly relevant for a learning program.

- Teaching process: A teaching process can be explained by various constitutive features like teaching objectives, contents discussed or edited, the didactic structure of the process, social forms used, methods applied or learning theoretical implications. The development of the mentioned aspects can also take different forms and sequences.

- Teachers: Teachers are distinguished by their expertise in the fields of science, didactics and educational science. In addition, they possess media didactic competence, a specific professional understanding, values and attitudes which can influence the design of teaching and learning situations in different ways.

- Learners: Pupils can also be described by specific qualities and characteristics. Such factors can be their subject-specific or interdisciplinary previous knowledge, their cognitive resources or intellectual capacities and their views of values and attitudes. Their socio-cultural backgrounds, such as the educational proximity or distance of their parental home also have an influence of these characteristics.

The differentiation into these four factors clarifies the question of the effects of digital media. It is not only the digital medium that can be considered as an influencing factor. The other factors must also be differentiated and considered in their interactions. It must also be taken into account no general statements can be made that digital media in teaching processes have this or that effect on pupils.

2.1.2. Benefit of Digital Media

New media formats can lead to an increase in motivation among learners. This is based on the so-called novelty effect, which is, however, of short duration. If such a teaching offer is perceived as easy or entertaining, this

can lead to a reduction in the effort required to solve the learning material. This results in a lower learning performance compared to conventional learning materials.
Since the introduction of computer-based learning methods, the question has arisen whether new forms of presentation in the field of teaching and learning are superior to others. Conventional teaching is often used as a comparison. It was discovered that the average learning success is relatively independent of the chosen media system or technology used. It is decisive which didactic method is used to prepare the learning material. A dependency has to be made in a number of parameters. Such parameters can be, for example, the target group, the teaching content or the learning objective.
While the effectiveness of learning depends on the success of learning, in practice the efficiency of educational offers is of particular relevance. The relationship between effort and results in different procedures is also an important factor. Higher learning efficiency occurs when

a) a higher learning success with constant effort

b) a constant learning success with less effort

is achieved.
The use of media can certainly lead to an increase in efficiency. However, it can also be the case that this goal is missed because the acceptance and use of media by learners is too low. This is often the case when learners are not sufficiently prepared for this type of learning or they are not adequately supervised. Nevertheless, a reduction of the average learning duration can be assumed for media-supported forms of learning.
The benefits of new technologies for education do not depend on the availability of equipment and technology in schools, but on the overall quality of the process chain of their utilization. This includes, for example, the quality of planning, conception, development, introduction, use, maintenance and quality management. An added value only arises when the technology is transformed into a problem solution for educational concerns. (Kerres, 2002)

2.2. Learning Analytics

> "Learning Analytics can be seen as an iterative cycle of hypothesis formation, testing and refinement. It means selective extraction of the kept data of large databases, their processing with the use of several educational techniques of Data Mining such as classification, clustering, statistics, regression etc. and acquiring the processed data that would improve the approach to larger groups of participants in the learning process." (Krikun, 2017)

The system of education has changed constantly over time, since several new techniques influence the system to improve it. For some time now, LA has increasingly been associated with Technology Enhanced Learning (TEL). LA has gained additional importance when it was predicted in 2012 as a new future trend in the field of education. (Johnson, Adams, and Cummins, 2012) This is associated with the fact that digital content is increasingly being used in education and teaching. This makes it possible to record and evaluate a lot of data and information about the various approaches to solving tasks and the behavior of individual users. This process is called Educational Data Mining (EDM), which raises the question which data and information should be collected, so that learning can be supported and improved by LA. (Duval, 2011)
On the one hand, teachers can benefit from this collected data and information by getting feedback on student activities and common mistakes. On the other hand, learners can improve their learning by getting recommendations of specified learning objects.
The following sections explain the different drivers for the development of LA. Afterwards a reference model for LA is briefly described.

2.2.1. Learning Analytics Drivers

That someone can understand the meaning of the use of LA in education, it is very helpful to describe the drivers and motivators of this process. These drivers cover many different areas, ranging from technological to political drivers. (Ferguson, 2012)

Thus one can recognize that the LA process is complex and not easy to implement.

Big Data

In many areas extreme amounts of data are collected nowadays, which can neither be processed by humans nor by simple database systems. Therefore, new systems must be developed that can handle these data sets with corresponding performance. Through the development and introduction of various Virtual Learning Environments (VLE) systems, such as Moodle, the education sector was also confronted with large amounts of data. In order to obtain useful information from these data sets to achieve improvements in teaching, the data must be analyzed and processed. (Ferguson, 2012)

Online Learning

In the field of education, more and more tasks or entire courses are solved or attended online, so that learners are never under the supervision of teachers who can help with problems of ambiguity. Therefore, it is important that teachers receive information and feedback from their students, conversely, it is also not negligible that pupils somehow get help from their teachers. In this way, frequently occurring problems can be solved and avoided in the future, so that both groups benefit from the system. (Ferguson, 2012)

Political concerns

Politics has a big influence on this whole process of LA, because this data preparation takes up many resources and therefore, requires high investments. Without this external support, LA can only be realized with great difficulty or not at all. (Ferguson, 2012)

2.2.2. Process

As already managed, LA is an iterative process which is shown in Figure 2.2. This process cycle is an example for the educational sector and has three steps which are explained in the following listing. (Chatti et al., 2013)

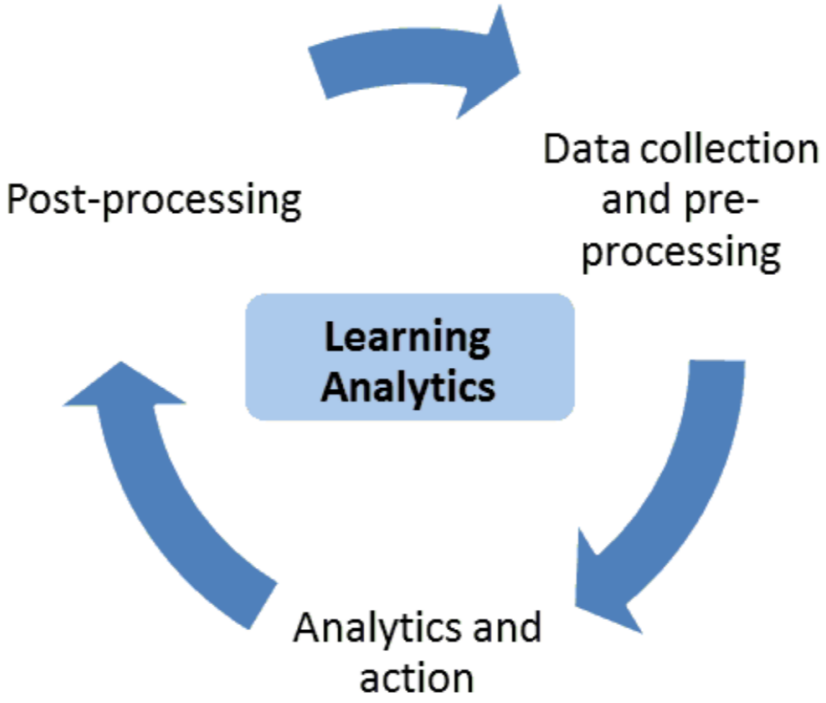


Figure 2.2.: Learning Analytics process (Chatti et al., 2013)

- **Data collection and pre-processing**: In the data collection step all the data are collected which quality influence the final result. The pre-processing step converts the collected data into the appropriate format and prepares it so that the further steps in the process can deal with the data. (Chatti et al., 2013)

- **Analytics and action**: The data from the previous step are analyzed with the help of different techniques and they are visually represented which should increase the learning effectiveness. In addition to analytics, various actions are performed on this information, such as

monitoring, forecasting or evaluation. (Chatti et al., 2013)

- **Post-processing**: In this step, the processed data can be extended or replaced with new data, or another analytic method can be chosen if the result was not satisfactory. The first step is then executed again with the help of these new data. (Chatti et al., 2013)

2.2.3. Reference Model

The described LA reference model is divided into four dimensions, which can be seen in Figure 2.3. The four dimensions are divided as follows and they are explained in more detail afterwards:

- **What?**: The first dimension describes the data which is processed and analyzed by the system.
- **Why?**: The purpose of the data analysis is described in the second dimension.
- **Who?**: The third dimension deals with the target group, which should benefit from the result of the analysis.
- **How?**: The last dimension explains the data analysis process.

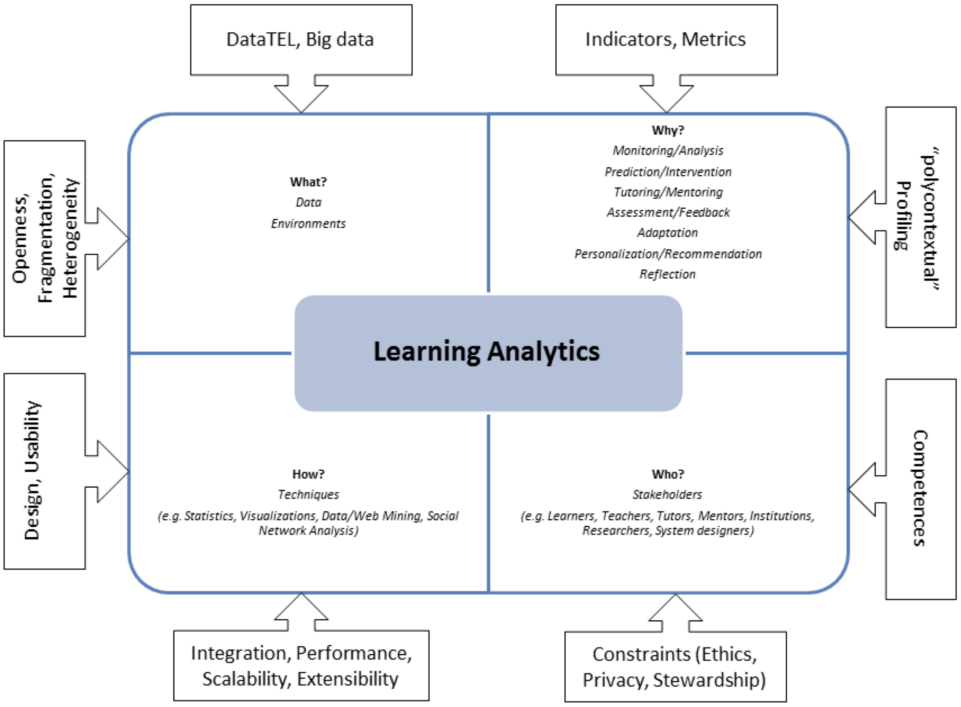


Figure 2.3.: Learning Analytics reference model (Chatti et al., 2013)

1. Dimension - What?

The most important component in the implementation of LA are the data from the field of education. The data are obtained from different sources, whereby the sources can be divided into the centralized educational systems and distributed learning environments categories. In the first category are Learning Management Systems (LMS) integrated, in which commercial systems like Blackboard (Martin, 2008) or WebCT (Lu, Yu, and Liu, 2003) and open source systems like Moodle (Dougiamas and Taylor, 2003) or groups in social networks (Wang et al., 2012) are included. LMS store a lot of data, which are generated by different activities of learners like writing or reading. These examples show that a lot of data is generated from different sources, so that LA from distributed learning environments becomes more and more important. For this reason, data is often collected

in different formats, which makes the conversion to a uniform format and the subsequent processing very difficult. (Chatti et al., 2013)

2. Dimension - Why?

This dimension includes objectives of the different stakeholders which are included in LA and some of these objectives are described in the following listing. (Chatti et al., 2013)

- **Monitoring/Analysis**: This point is especially important for teachers by recording and evaluating the behavior of their students. This gives them the opportunity to evaluate their teaching process in a better way and, as a result, to improve it and adapt it to the needs of the learners.

- **Prediction/Intervention**: In predication and intervention, the behavior of learners should enable their knowledge and skills to be predicted. This makes it possible for teachers to know in advance which of their students need additional help, which can improve learners' achievement.

- **Tutoring/Mentoring**: Tutoring is the help given to learners in a specific context, for example to a part of the material of a course. The help of a tutor mainly refers to the explanation of the approach and practical implementation of the subject matter. In contrast, mentoring refers to a longer period of time than tutoring and is not very contextual. It also focuses on the teaching process and the learner should be prepared by a mentor for new tasks and challenges in the future.

- **Assessment/Feedback**: The progress of the learning process will be measured and evaluated by the use of the assessment. Teachers and learners also receive feedback on their concern and the context of the learning material.

- **Adaption**: In this point the next steps of resource planning are proposed by teachers individually for each learner.

- **Personalization/Recommendation**: In personalization the learning methods and so on are explained to each learner so that everyone can achieve their personal learning goals. Furthermore, the learners should be trained not only to get the learning material from the teachers, but also to acquire knowledge through recommendations.

- **Reflection**: In this point teachers and learners should compare their teaching and learning principles with others and get improvements from them in order to enhance their methods in the future.

All these points are often difficult to evaluate, as collecting the necessary information is not trivial.

3. Dimension - Who?

LA can be useful for several groups of people who have different demands on the process. These stakeholders and there personal objectives are described in the following listing. (Chatti et al., 2013)

- **Teachers**: It is interesting for teachers that their teaching practices are analyzed by the system and adapted to the needs of their students.

- **Learners**: Learners want suggestions for improving their learning environment so they can improve their grades.

- **Educational institutions**: These institutions have expectations from the system to help them make decisions about their curriculum, finances and staff planing. In addition, learners with learning problems and correspondingly bad grades should to be identified and the general learning success including grades of all learners should be improved.

- **LA researchers**: These researchers should publish their findings and results so that others can see the effects of the implementation and use of such a LA system.

4. Dimension - How?

For this dimension exists many different techniques, but mainly four have established themselves in the last years. (Chatti et al., 2013)

- **Statistics**: In almost all LMS it is possible to export the interaction with the system in various statistical evaluations , where the average, median or standard deviation is displayed. Mostly the page views, the used time of the system and so on are evaluated.

- **Information Visualization (IV)**: In order to present the statistics as simple as possible, mostly diagrams, graphs or tables are used, because plain text is very difficult to understand. However, it is important to choose the correct form of presentation in order to achieve the desired goal of the analysis.

- **Data Mining (DM)**: In data mining, knowledge is gained from data sources such as texts, pictures, databases, and so on using different methods.

- **Social Network Analysis (SNA)**: The representation and analysis of networks has become more and more popular due to the success of social networks, as the graphic representation makes it relatively easy to understand the connection between two nodes. In addition, important nodes are recognizable at first glance since these central nodes have many connections to others.

This reference model and the division into four dimensions should serve as an example of how the LA process can be realized. (Chatti et al., 2013)

2.3. Mobile Language Learning Applications

The offer of various mobile learning applications to learn a language is very large and is constantly growing. In this section a few examples of such applications are shown and briefly described to get a short overview

of existing applications in this field. The selection of applications refers to mobile applications which are designed and made for smartphones and tablets. The selected applications were found in the Apple App Store using the German spelling search terms ”Sprache lernen” and ”Grammatik”, which means ”learn language” and ”grammar”, and these applications are tested on an iPad Air 2. Most applications require a registration and a login before use, which is free in most cases. However, in most cases a user have to pay for the full functionality of the application, which limits the amount of exercises in the free version. Therefore, often only single exercises are available or advertisements are shown during the usage of the application, which has a negative influence on the UX. The examples also show that the target group of the various applications is strongly differentiated, since some applications are made more for children, which have included many playful components, and others are designed more for adults, since the teaching content is more in the foreground. A possible use of the applications in schools is also addressed, since the IDeRBlog application is mainly intended for schools and therefore, this aspect should not be neglected. At this point it should also be mentioned that the descriptions of the applications are taken from the websites of the manufacturers and that the functionality of the applications has been tested by own testing.

2.3.1. Babbel

Babbel[1] is one of the most advertised and widely used application for language learning. When using the application, it is immediately noticeable that at the beginning the user should be introduced to a previously selected language through simple exercises. Step by step the exercises become more demanding and the aim of this application is to learn a language by means of listening, reading, writing and speaking tasks. In Figure 2.4 the overview of the different courses is shown. The gray lock next to the lines in the list indicates that theses courses are locked in the free version of the application. In order to make all functionalities and courses available, a user must activate them through a paid subscription model.

[1] **https://www.babbel.com/?locale=en [accessed on 14.08.2019]**.

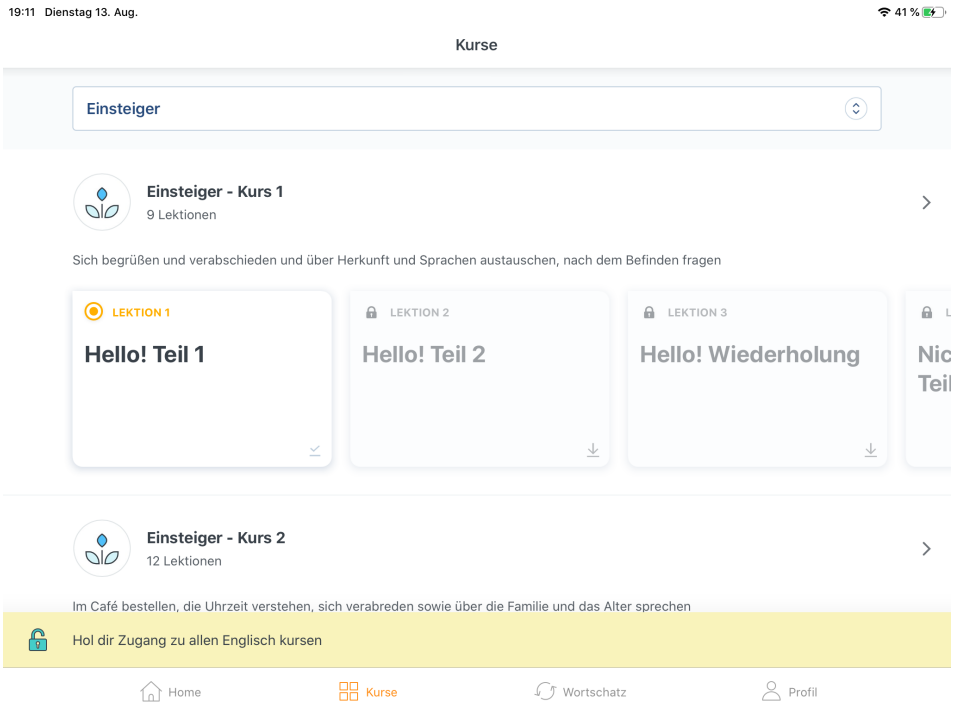


Figure 2.4.: Babbel: Exercise overview

When testing the application, it is also noticeable that the learning process is based on the constant repetition of words and short phrases, which creates a vocabulary for the user and makes it visible. However, this vocabulary is very limited in the free version, so that the use of this application without payment is not very effective and therefore, not suitable for schools.

2.3.2. Duolingo

The second well known application is *Duolingo*[2], which is similar to Babbel very common and widely advertised. The first time the application is started, a short test is displayed so that the user can be placed at a certain level of the previously selected language. Then the user can select the different exercises

[2] **https://www.duolingo.com [accessed on 15.08.2019]**.

in the course overview shown in Figure 2.5. In this application, a user account must be created in order to select the different exercises. During the exercises there are a certain number of lives available, which will be lost if mistakes are made during the exercises.. If all lives are wasted, the user have to wait a certain time until the lives are filled up again and the exercises can be solved further. This process with the limited lives or the removal of advertisements can be prevented by signing a paid subscription model, which also unlocks other functionalities. Unlike Babbel, however, all exercise content can be viewed and practiced in the free version. However, this application is also rather unsuitable for use in schools, because practicing in the free version is not possible indefinitely, since lives must always be available for it. In addition, the exercise content is very similar to Babbel, since the same phrases and words are always repeated, even if the scope of the exercise is relatively large.

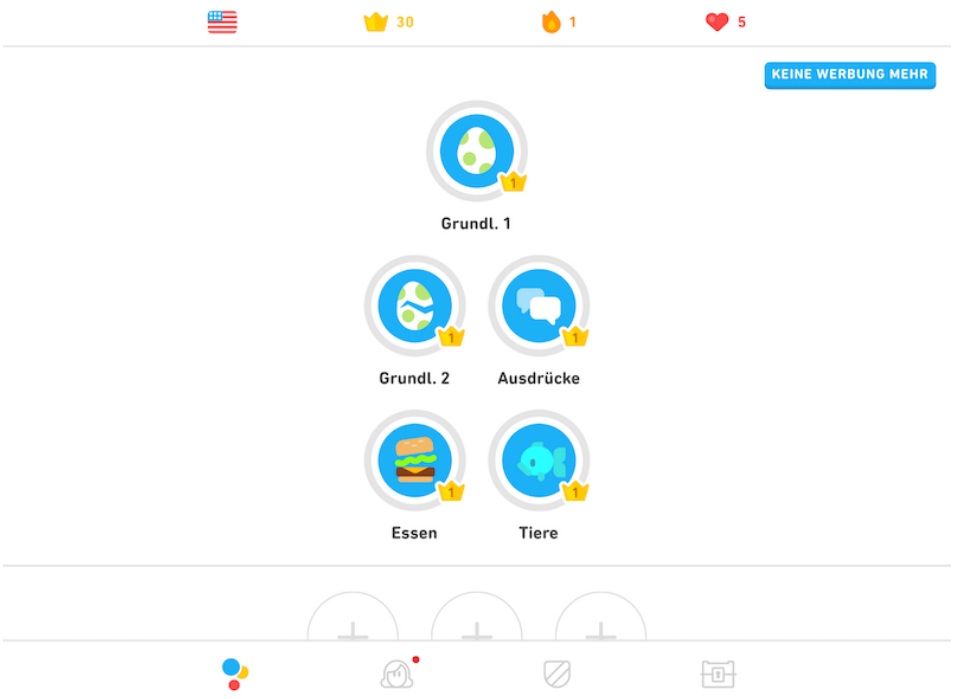


Figure 2.5.: Duolingo: Exercise overview

2.3.3. Lernspiele für Kinder – Lernkarten

The application *Lernspiele für Kinder – Lernkarten*[3] is designed for children because the exercises are relatively simple and understandable. When starting the application the user will get an overview of the exercises, which are divided into different categories shown in Figure 2.6. The teaching content of this application differs slightly from the two applications described above, as mainly listening and speaking exercises can be selected. For the full functionality of the application, an amount of money has to be activated, but this amount has to be paid only once in contrast to the above mentioned subscription models. Nevertheless, this application is not suitable for schools, because the content of the free and also the paid version is too small.

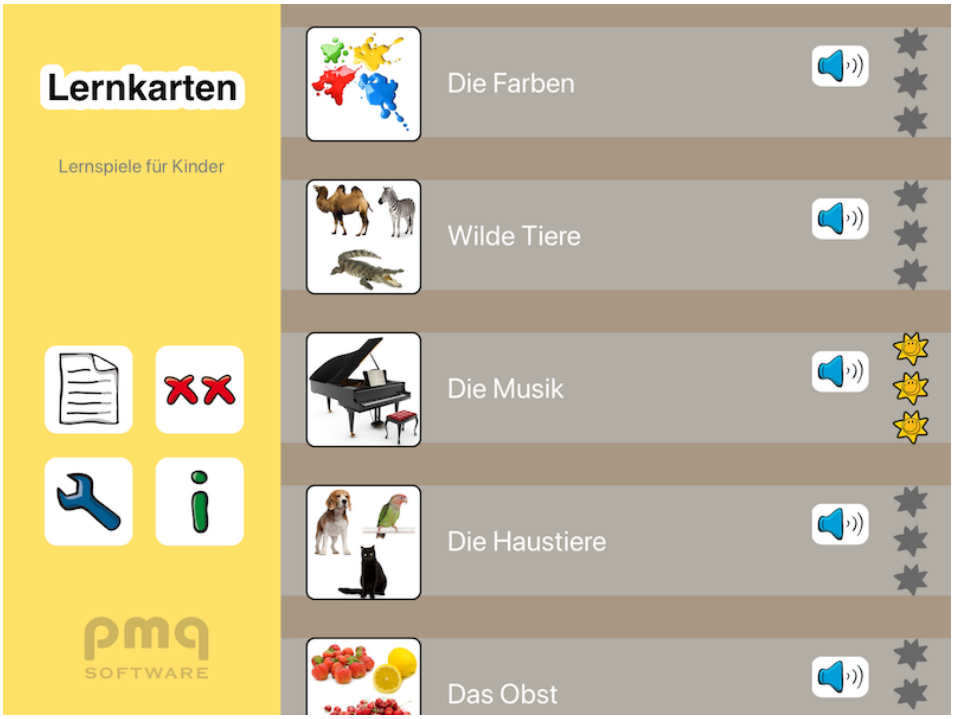


Figure 2.6.: Lernspiele für Kinder – Lernkarten: Exercise overview

[3] **http://www.pmq-software.com/sw/de/lernspiele/lernkarten/ [accessed on 15.08.2019]**.

2.3.4. 6000 Wörter - Deutsche Sprache Lernen

The application *6000 Wörter - Deutsche Sprache Lernen*[4] has a very large selection of exercises in the free version, as shown in Figure 2.7. In order to unlock all functionalities, either a one-time fee must be paid or points has to be redeemed which can be collected by completing exercises. Theoretically the full functional range of the application can be activated free of charge, which differs from all other described applications and can be regarded as very positive. As a result, this application can be used in schools because the content of the course is large and no costs are incurred. In order for the teacher to have a direct insight into the progress and statistics of the individual pupils, a separate user account must be created for each pupil.

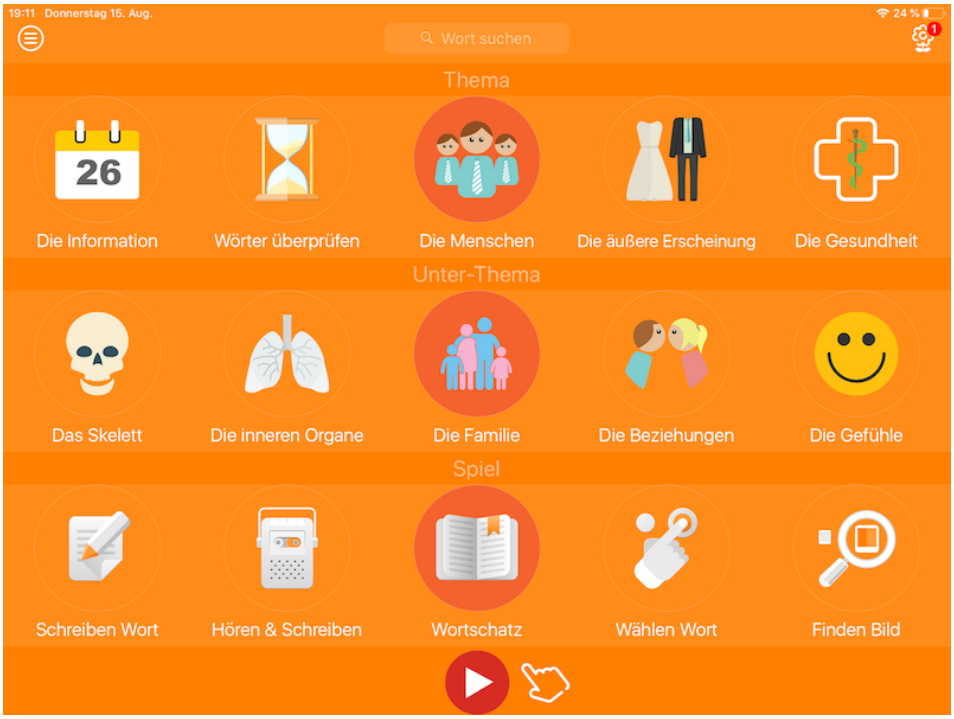


Figure 2.7.: 6000 Wörter - Deutsche Sprache Lernen: Exercise overview

[4] **https://www.funeasylearn.com/de/app/deutsch-lernen/ [accessed on 15.08.2019]**.

3. Technical Implementation

This chapter deals with the technical implementation of the application for the two platforms Android[1], which is developed by Google[2], and iOS[3], which is developed by Apple[4]. In the first part, the user authentication is explained that someone can log in to the application, which is a requirement for the use of the applications. In this section is also explained how the exercise data are loaded and saved in the application and how different exercise results are handled. Then the different types of tests used in software development are described. As already mentioned, an application was developed for Android and iOS. A small insight is given into the different used development environments, used frameworks and implemented tests. As last point in this chapter the two developed applications are described, which will serve as basis applications for the IDeRBlog ii project. From today's perspective, several or different exercise formats are to be added or supplemented in the applications in the future. The applications can be used universally, that means they can run on smartphones and tablets, but the design was conceived for tablets, as this use case mainly reflects the use in schools. As a note it should be mentioned here that the singular is used in the further description, although the Android and iOS versions of the application are meant.

[1] **https://www.android.com [accessed on 18.07.2019]**.
[2] **https://about.google/intl/en/ [accessed on 18.07.2019]**.
[3] **https://www.apple.com/ios/ios-12/ [accessed on 20.07.2019]**.
[4] **https://www.apple.com/contact/ [accessed on 20.07.2019]**.

3.1. Administration

Since the application can currently only be used by registered users, the following paragraph describes the existing user management of the TU Graz Graz in more detail and how it is used. All the exercise data is provided by an external server, which can be loaded form the server via a RESTful[5] API. Various data such as exercise results or user actions can be sent to the server, which is also realized via an API. It was a requirement during the development of the application that a user can also use it in offline mode, that means without an existing Internet connection. All these functions of the data exchange and the further processing of the the data and the possibility to store data locally are also explained.

3.1.1. User Management

Like many other applications of TU Graz Graz, this application also uses the user management for the authentication. At the moment there is no possibility to register as a new user in the application, so only already registered users with existing accounts have access to the content of the application. The reason for this is that the next version of the application will access a database with an intelligent dictionary that provides individual exercise data for each user. For this aspect to be realized, every user must be registered and authenticated. For the communication with the user management a Simple Object Access Protocol (SOAP) web service is available, which was used in the application. The following web service functions are implemented:

- **isUserAllowed**: For this function, the corresponding application must be entered in the user management. Otherwise it can not be checked at all whether a user can log in or not. The username and password are transmitted for authentication in the request and an object with various attributes is sent back in the response.

[5] **https://restfulapi.net/http-methods/ [accessed on 18.07.2019]**.

- **getName**: This function expects the user Identification (ID) int the request and returns the first name, last name and username of the corresponding user in the response.

An example for the request body of the isUserAllowed API call which is sent to the user management server[6]:

```
<soapenv:Body>
  <soap:isUserAllowed ... >
    <password xsi:type=\"xsd:string\">XX</password>
    <username xsi:type=\"xsd:string\">XX</username>
    <idApp xsi:type=\"xsd:string\">XX</idApp>
    <hmacClient xsi:type=\"xsd:string\">XX</hmacClient>
  </soap:isUserAllowed>
</soapenv:Body>
```

The response body of a successfully authenticated user:

```
<SOAP-ENV:Body>
  <ns1:isUserAllowedResponse>
    <return xsi:type="ns1:LoginCredentials">
      <accepted xsi:type="xsd:boolean">true</accepted>
      <hmac xsi:type="xsd:string">XX</hmac>
      <idUser xsi:type="xsd:int">XX</idUser>
      <message xsi:type="xsd:string">Everything fine :-)</message>
      <roles xsi:type="ns1:Roles">
        ...
      </roles>
    </return>
  </ns1:isUserAllowedResponse>
</SOAP-ENV:Body>
```

The unnecessary parts have been replaced with '...' and all values containing user information with 'XXX'. An example for the request and response body of the getName API call looks very similar to the example above, only the attributes are different.

[6] **https://schule.learninglab.tugraz.at/usermanager/soap [accessed on 18.07.2019]**.

For the login process, the user enters the username and password in input fields in a login screen. With the username, the hashed password, the app ID and a Keyed-Hash Message Authentication Code (HMAC) string, which is generated from all mentioned parameters and a fixed defined key, the isUserAllowed function of the user management is called. The returned value is checked and if the login process is successful, the user is redirected to the start page of the application. Otherwise, an error message is displayed and the user has the option to repeat the login process. If the user is logged in successfully, all parameters from the response body are stored locally. Afterwards the getName function with the user ID from the received response body is called and the user information are stored locally too.

3.1.2. Exercise Data

The exercise data are loaded from an external server via an API. This has the big advantage that new tasks of an exercise created on the web server are immediately available in the application without having to deploy and publish the application again. The only requirement is an internet connection to load the data which stored locally on a tablet. After loading the exercise data once, these exercises can also be used offline. In the application two different API calls for each exercise are used for receiving the exercise data, which are executed and processed. These two calls are using the GET method, so data is expected as response from the server. If a call or timeout problem occurs, an error is returned and displayed to the user. In this case the loading of the exercise data can be tried manually again. The following list describes the two API calls in detail.

- **Check update**: This API[7] returns a response in JavaScript Object Notation (JSON)[8] format containing the version number of the keyword exercise. Since this version number is stored locally, this call can be used to easily find out if new or changed exercise data is available on the server. If the version number is different, a new loading of the exercise data will be executed. The response body looks like this:

[7] **http://schule.learninglab.tugraz.at/iderblogexercises/merkwoerter/api/check-update**.
[8] **https://www.json.org [accessed 18.07.2019]**.

```
{
  "success": true,
  "data": "e1554f6200a3b8d284da..."
}
```

- **Get exercise data**: This API[9] returns a response body in JSON format containing the exercise data of the keyword exercise. The response body looks like this:

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "name": "Merkwörter (leicht)",
      "description": "Du bekommst jeweils ein... ",
      "config": {
        "words_count": 6,
        "words_per_round": 1,
        "chars_clickable": true,
        "show_wrong_input": true,
        "time_read": 0,
        "time_write": 0,
        "time_wrong_input": 0
      }
    }
  ]
}
```

These two examples show how the response from the server is structured. The `success` parameter indicates whether the API request could be processed successfully. The `data` parameter contains the required data, which is parsed in the application for further processing. Long values are abbreviated with '...' in the examples. For another exercise type the data in the `config` part looks a little bit different in comparison to this example because each exercise has its individual configuration parameters.

[9] **http://schule.learninglab.tugraz.at/iderblogexercises/merkwoerter/api/exercises**.

The error locations and insert exercise need in addition to the exercise also sentences data. This information is loaded for example for the insert exercise by an additional API[10] from the server. The response body contains the following values in JSON format:

```
{
  "success": true,
  "data": [
    {
      "id": 2,
      "exercise_id": 1,
      "text": "Am Dienstag fliegen wir nach New York.",
      "gaps": {
        "4": "Überlege, ob das i kurz...",
        "5": "",
        "14": "Überlege, ob das i kurz...",
        "15": ""
      }
    }
  ]
}
```

This data differ marginally depending on the exercise format too.

3.1.3. Exercise Logs

During the operation of the application, various user interactions are recorded and processed. Each interaction is uploaded to the server so that it is possible to generate various statistical evaluations from the collected data. For example, the data of some interaction can be uploaded within the keyword exercise via the API[11] using the POST method. For this functionality, the request body in JSON format must contain the following information:

[10] **http://schule.learninglab.tugraz.at/iderblogexercises/einsetzen/api/sentences**.
[11] **http://schule.learninglab.tugraz.at/iderblogexercises/merkwoerter/api/log**.

```
{
  "app": "iOS",
  "mode": "write",
  "exercise_id": 5,
  "user_role": "Student",
  "user_id": 2858,
  "session": 1560709957,
  "timestamp": 1560709957,
  "state": "checked",
  "input_errors": {
    "grapheme_clicked": null,
    "word": "",
    "words": [
      "ängstlich",
      "schreien"
    ],
    "user_inputs": [
      "engstlich"
    ]
  },
}
```


The response has only one key-value pair in JSON format with the information if the data upload was successful or not. If any error occurs during this process, the request object is cached locally. The next time a data upload will be executed, all buffered data is also uploaded. The data is not deleted locally until it has been successfully uploaded to the server. As already mentioned in the exercise data section, the various interactions also differ slightly in the data structure which are uploaded to the server.

The following user actions are tracked:

- **Start an exercise**: At starting a new exercise, the uploaded data object contains `"state": "started"`.

- **Interactions during an ongoing exercise**: If the user makes a mistake during an exercise, a data object which contains `"state": "checked"`

is uploaded.

- **Finish an exercise**: After all tasks of an exercise are completed by the user, the uploaded data object contains `"state": "finished"`. With the start and finish information it is possible to calculate the time a user needed to complete the exercise.

3.1.4. User Statistics

If tasks are solved correctly in the different exercises, points are awarded and credited to the corresponding user. The back-end server manages how many points are awarded for the tasks. The already achieved points of a user can be retrieved via the API[12] using the GET method by passing the users ID as the only parameter (`"id": "[USER ID]"`) in the request body.

3.1.5. Tests

Writing tests is an important part in the developing process of an applications. With the help of such tests, various programming errors can be prevented or found. The Figure 3.1 shows a distinction between three categories of tests, which pursue and fulfill different tasks. Furthermore, they differ greatly from the complexity of the creation and from the processing time. The number of tests of the different types is to be set in relation to their complexity. This means that many unit tests and few UI tests should be written. Some points to keep in mind when creating tests (Tam and Katz, 2019):

- **Duration time**: Tests should have a short duration time and run quickly.

- **Independent**: Every test should have the same state before starting and they should not influence each other.

[12] **http://schule.learninglab.tugraz.at/iderblogexercises/api/users/[USER-ID]**.

- **Repeatable**: It is necessary that each test have the same result every time the test is started. To make this point applicable the input data has to be always the same.

- **Self-validating**: The test result should be either successful or unsuccessful and the test should run fully automated without interactions from the outside.

- **Timing**: Tests should be written first and then the code which has to be tested. This programming procedure is called Test-Driven-Development.

All these points cannot always be complied, but they should be considered as a guideline in programming. The three test types are described in detail below:

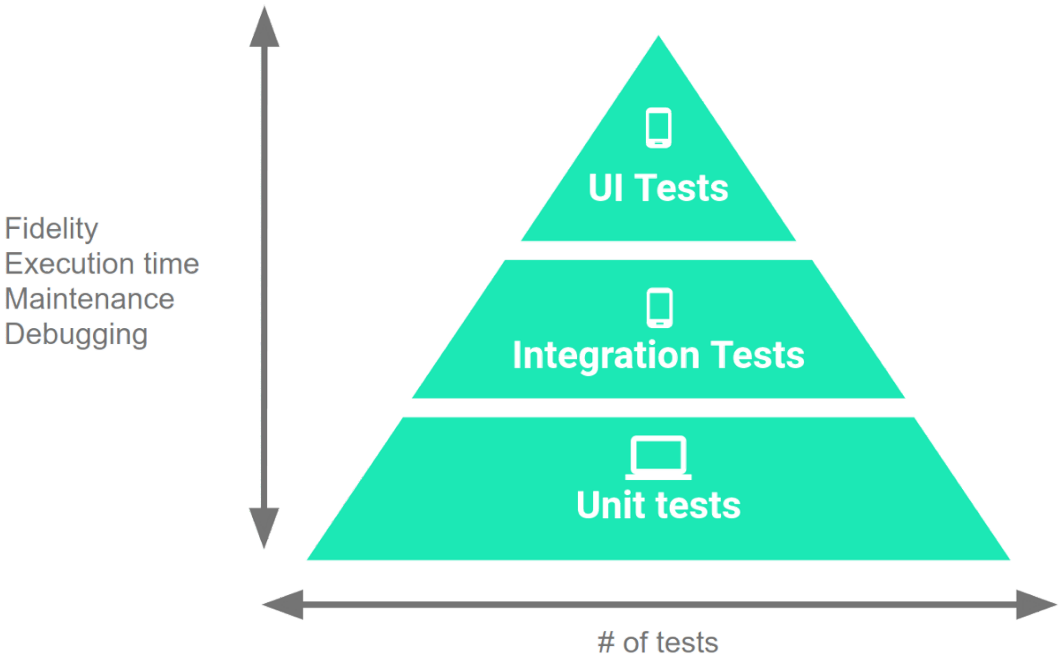


Figure 3.1.: Testing Pyramid (Google, 2019a)

- **Unit tests**: These tests can be created and written with little effort. In these tests, the behavior of components is validated. In object-oriented

programming languages, such components can be individual methods or entire classes. These have no or only small dependencies to other components in the program, so the processing time of the tests is short. The input parameters are preconfigured for such tests and the result is usually one output. By checking this output value errors can be found in the programming. An example could be the login function in a program. The input values would be a username and a password and the output could be true or false depending on correct or wrong input values.[13]

- **Integration tests**: In comparison to unit tests where only single components are tested individually, integration tests combine components to groups and test the collaboration and interaction between them. Such a test can be very useful to test the exchange of data or information between such modules. As a disadvantage it can be stated that if a test fails, it is not immediately apparent which module is faulty or where the error occurred, even if the individual components have already been tested. With this test, the login process can be extended as mentioned above and test whether a successful login forwards to the start page of the application. Additionally it can be checked if the entered username is also available in the new screen.[14]

- **User Interface (UI) tests**: The last test type is intended to test the design and handling of the UI, in which various user interactions are performed fully automatic. With the help of these tests, the behavior of an application is to be tested from a user's point of view using the graphical user interface. As an example, the login process can be used a further time. The UI test automatically fills in the username and password in the given input fields and then the login button is pressed. If the start screen appears, the test has run successfully, otherwise the test fails. At first glance it seems that UI tests are very similar to the integration tests and they can replace them, but these tests complement the integration tests. Although these tests are very important, in reality they are often neglected during development

[13] **http://softwaretestingfundamentals.com/unit-testing [accessed on 03.09.2019]**.
[14] **https://www.guru99.com/integration-testing.html [accessed on 23.07.2019]**.

because they have some disadvantages:

- The creation of such a test takes a lot of time because each interaction has to be programmed separately.
- UI tests are very difficult to read and therefore, it is often not quite clear what is being executed.
- Many applications get information via a network connection from a back-end server, which can make the tests very inconsistent because no control over the data flow has been achieved.
- It happens very often that the design of an application changes, which means that existing tests frequently have to be changed or supplemented.

Despite all these disadvantages, the advantages of UI tests outweigh and should therefore, also be considered when developing applications. (Tsadok, 2019)

The test environment and the applicable tools differ for each programming language, therefore, the sections 3.2.2 and 3.3.2 explain for each used OS, Android and iOS, the development environment independently.

3.2. Android

The Android OS for mobile devices is an open source project based on a Linux kernel. This OS is used on many devices with different display sizes from many various manufacturers. This is one reason why it is the most used OS[15] in the mobile sector worldwide, which also increases the development effort of an application. The current Android version is Android 9.0 Pie[16] (API level 28). In order to support the use fo the IDeRBlog ii application by many people, it was quickly decided to develop a version for the Android OS. Before a person can develop an application for an OS, it is important to know the structure of the system. A rough overview of this structure is shown in Figure 3.2 and explained briefly below. To ensure that the application can be used smoothly on the various devices with different hardware configurations, it is a good practice to use as few hardware dependencies as possible.

[15] **http://gs.statcounter.com/os-market-share/mobile/worldwide [accessed on 19.07.2019]**.
[16] **https://developer.android.com/about/versions/pie [accessed on 21.07.2019]**.

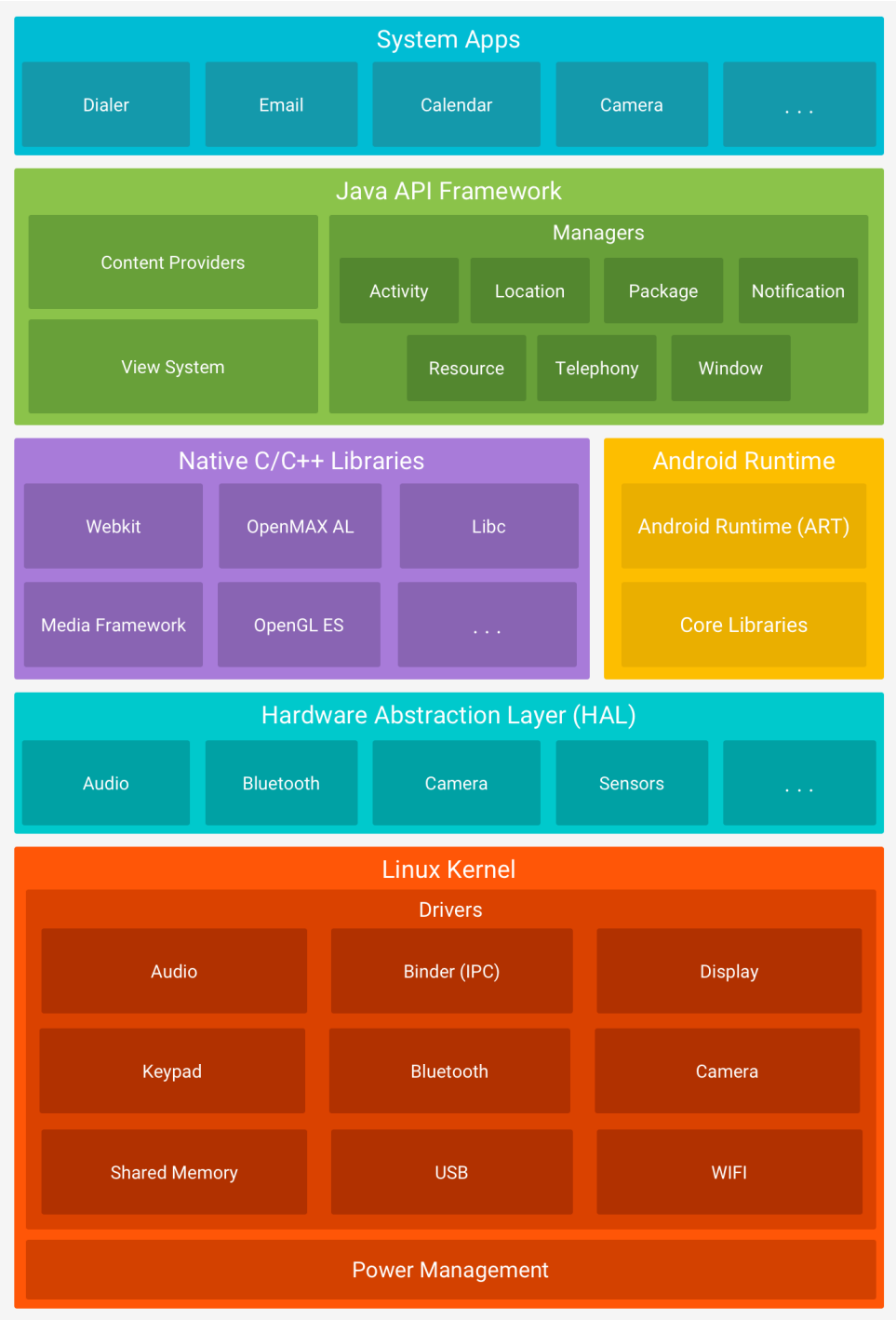


Figure 3.2.: Android structure (Google, 2019b)

The following list describes the individual layers of Android and their basic functionalities in more detail. (Google, 2019b)

- **Linux kernel**: The Linux kernel serves as connection of the hardware components and has all necessary drivers available. It offers hardware manufacturers the possibility to develop their own drivers so that the OS supports all functions of the different components. Examples are the screen or sound output, various connection modules like Bluetooth or WiFi and many more. The kernel is also responsible for the power management and threading, whose functions are accessed by many other layers.

- **Hardware Abstraction Layer (HAL)**: This layer is required for the use of the hardware components and provides a standard interface to expose hardware capabilities to the higher-level Java API framework.

- **Android Runtime (ART)**: Each app which is running on a device has its own container of the ART since Android version 5.0. The ART compiles source code into Dalvik Executable format (DEX) bytecode, which is executed on Android devices.

- **Native C/C++ Libraries**: Some of the Android system components are written in C and C++. This has the advantage that source code written in this programming language can also be easily used in applications.

- **Java API Framework**: All core, modular system components and services, which are available in the Android OS, are be accessed through the framework API and these components are easy to reuse. The View system provides many UI elements like buttons, grids, lists and text fields. This layer also includes managers for resources, notifications, activities and so one. The third big part in this layer is the content provider which handles the data sharing between applications.

- **System Apps**: The Android system has different system applications for email messaging, internet browsing, telephoning and so on which are can used and implemented in own applications.

3.2.1. Development Environment

For the development of Android applications, the official Integrated Development Environment (IDE) is Android Studio based on IntelliJ IDEA[17]. Android Studio is available on all popular platforms macOS[18], Windows[19] and Linux[20] and the current version is 3.4.2. There are some other IDEs available for the development of Android applications, which are not described here because Android Studio was used for this project. Additionally, a Java Development Kit (JDK)[21] must be installed before Android Studio can be used. To publish applications in the Google Play Store[22] someone needs a Google account, which has to be activated with a one-time amount of money. When the development of Android App began, Java[23] was used as the standard programming language. Since 2017 Kotlin[24] is officially supported by Google, which means that apps can also be developed in this programming language. This programming language is gaining more and more popularity, so since May 2019 this language is preferred by Google for the development of applications and Kotlin is also used in this project. (Lardinois, 2019)

3.2.2. Test Environment

As already mentioned, in the software development it is very important to check the developed program or application with different types of tests in order to detect or prevent errors. For this master thesis the testing of the API was required, therefore, in this project only unit tests were written. A working API is very important for this application, because otherwise no data can be exchanged with the back-end server, which would limit the

[17] **https://www.jetbrains.com/idea/ [accessed on 20.07.2019]**.
[18] **https://www.apple.com/macos/mojave/ [accessed on 20.07.2019]**.
[19] **https://www.microsoft.com/en-us/windows/ [accessed on 20.07.2019]**.
[20] **https://www.linux.org [accessed on 20.07.2019]**.
[21] **https://www.oracle.com/technetwork/java/javase/downloads/index.html [accessed on 20.07.2019]**.
[22] **https://play.google.com/store [accessed on 21.07.2019]**.
[23] **https://www.java.com/en/ [accessed on 20.07.2019]**.
[24] **https://kotlinlang.org [accessed on 20.07.2019]**.

functionality of the application. In the following paragraphs the functionality of a test for loading the exercise data of an exercise will be demonstrated.

Dependencies

For the Android unit tests three external test frameworks have been integrated so that the tests can be executed.

- **JUnit**[25]: This is a framework for writing unit tests. This framework is a open source project and is widely used, which guarantees continuous development of the framework.
- **AndroidX**[26]: This framework is developed by the Android team and is also an open source project. With this framework different libraries are provided and it is downward compatible.
- **Robolectric**[27]: Unit tests can be performed on a device or in a simulator, often resulting in long test lead times. Robolectric was used to minimize this long lead time and to avoid dependencies on devices or simulators. This framework sets up a freely configurable Android system in an isolated sandbox and executes unit tests in this environment.

Preparation

In Android Studio, you can create your own test files in which test methods are created.

[25] **https://junit.org/junit4/ [accessed on 30.07.2019]**.
[26] **https://developer.android.com/jetpack/androidx [accessed on 30.07.2019]**.
[27] **http://robolectric.org [accessed on 30.07.2019]**.

```
@RunWith(RobolectricTestRunner::class)
class ApiUnitTest {
    private lateinit var context: Context

    @Before
    fun setUp() {
        context = ApplicationProvider.getApplicationContext()
    }

    @Test
    fun testGetExercisesWords() {
        val urlSring :String  = Api.Url.property + ExerciseIdApi.Words.property + Api.AllExercises.property

        val apiManager :ApiManager  = ApiManager.Companion

        val result :String?  = apiManager.getServerData(urlSring)
        assertNotNull(result)
        val resultObject :ExerciseData!  = Gson().fromJson(result, ExerciseData::class.java)
        assertEquals(ExerciseData::class.java, resultObject.javaClass)
    }
```

Figure 3.3.: Android test preparation

Figure 3.3 shows a part of the code from the test file. Before the initialization of the test class, the Robolectric framework must be integrated with the command `@RunWith`. If `@Before` is written before a method within the test class, this method is executed before the test methods are executed. In this example, the variable `context` is initialized so that the test methods can access classes of the application. The final test methods are declared with `@Test`, which can be executed.

Result

If the test methods are executed, a result is returned indicating which tests were successful and which were executed with errors. In Figure 3.4 you can see such a result, in which successful tests are marked with a green tick and faulty tests are marked with an orange 'X'. Additionally the run time of each test method is displayed.

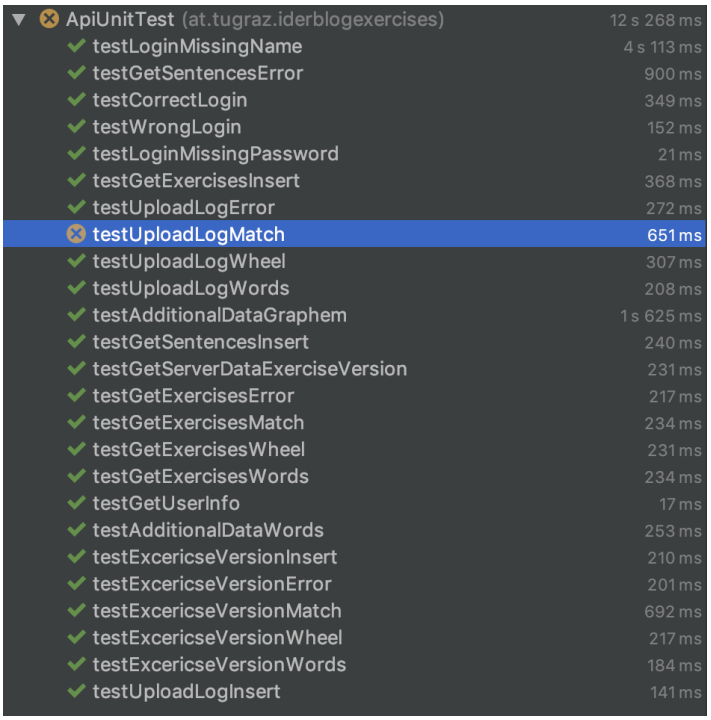


Figure 3.4.: Android test result

3.3. iOS

The OS for all Apple mobile devices is based on iOS, which was introduced to the public with the first iPhone. The current iOS version is 12.3.2 and it only runs on Apple devices, therefore, the number of supported devices is manageable, which greatly simplifies the design and hardware adaptations of the different devices compared to Android. Over time, the name of the OS was adapted to different devices. iOS runs on iPod Touch, iPhone, iPad and on Apple TV the name of the OS is tvOS. In June 2019 a new OS for the iPad was introduced, which will be called iPadOS. This OS also based on iOS and will be officially introduced in autumn 2019. Since the same base is always used, the development of applications for the different devices does not differ very much. In many elementary schools iPads are used in class to teach children how to handle digital media. Therefore, the decision was made to develop an iOS application for the IDeRBlog ii project in addition to Android. This OS differs in comparison to Android in many ways, so the structure of the system is also describe. Figure 3.5 shows a rough overview of the system, which is explained in more detail.

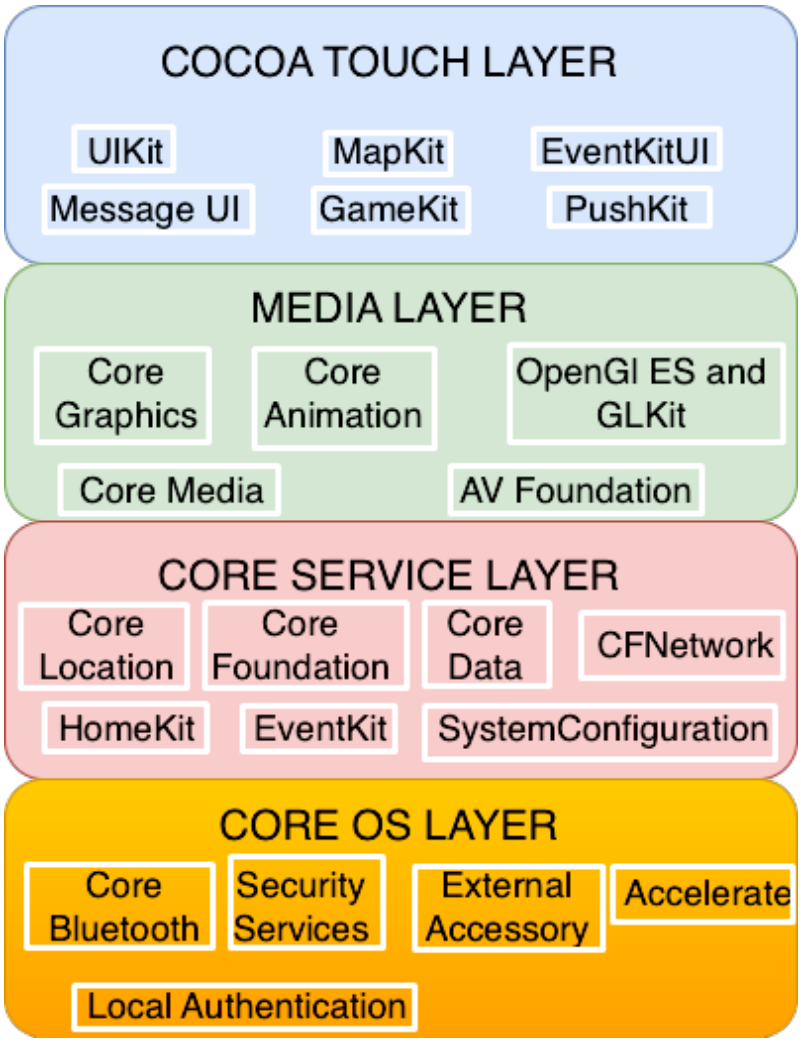


Figure 3.5.: iOS structure (Rupesh, 2017)

The following list describes the individual layers of iOS and their basic functionalities in more detail. (Rupesh, 2017)

- **Core OS**: This is the low-level layer and it has direct access to the hardware components and their functionalities. The Core OS contains many features as well which can be directly used by the developer

like the Security Services or Core Bluetooth Framework.

- **Core Services**: The Core Foundation and Foundation Frameworks are the main services in this layer, which supports system OS features that can be used in applications. For example the network availability, threading or file access are parts of the Core Services.

- **Media**: All graphic, video and sound outputs of an application are managed and created by this layer. These framework can be used by developers to beautify the design of an application.

- **Cocoa Touch**: The top of the iOS architecture stack overview is the Cocoa Touch layer. This layer contains all key frameworks of the OS and determine the look and feel of an application. On of the most important frameworks in this layer is the UIKit, which includes the basic application management and infrastructure, the multitasking support, many user action events and so on.

3.3.1. Development Environment

In order to develop applications for Apple devices and to publish them, you need to own a Mac computer and an Apple account. The registration for an account is free, so someone is already able to develop applications and install them on own Apple devices. However, if a developer wants to publish an application in the App Store[28] so that it is available to everyone, a paid subscription is needed which amount must be paid annually. For the development of applications for Apple devices, Xcode[29] is used as development tool, which already contains all functions for the developing process and it is only available for macOS. The current version is Xcode 10.2.1 and can be downloaded for free.

Similar to the development of the programming languages for Android, Objective C[30] was used as a supported programming language at the beginning

[28] **https://www.apple.com/ios/app-store/ [accessed on 21.07.2019]**.
[29] **https://developer.apple.com/xcode/ [accessed on 21.07.2019]**.
[30] Apple, 2014.

of the development of own iOS applications. Objective C was developed in 1984 and was an extension of C with SmallTalk style messaging and Object Orientation. In 2014 Apple introduced the new programming language Swift[31], which replaced Objective C as the standard used language. Nowadays all new developed applications have to be written in Swift, but Objective C libraries and code can still be integrated and used.

3.3.2. Test environment

In contrast to Android, writing and executing tests for iOS is a bit easier, since almost all necessary frameworks and libraries are already included in Xcode. Therefore, no external frameworks or dependencies have to be installed in addition to Xcode. The integrated test framework is called XCTest[32] which can run unit tests, performance tests and UI tests. As already mentioned above in the test section for the Android application 3.2.2, only the API for iOS is tested with the aim of unit tests. During the creation of the tests, care was naturally taken to ensure that the tests match for both platforms and deliver the same result. In order to see the difference between the tests for Android and iOS, the following sections will deal with the same test .

Dependencies

For the creation and execution of tests for iOS only Xcode is needed which is already necessary for the development of native iOS applications. Therefore, there are no dependencies for using other frameworks in comparison to the Android test environment.

Preparation

In Xcode, you can create your own test files in which test methods are created.

[31] **https://swift.org [accessed on 21.07.2019]**.
[32] **https://developer.apple.com/documentation/xctest [accessed on 31.07.2019]**.

```
import XCTest
@testable import IdeRBlog_Exercises

class IdeRBlog_ExercisesTests: XCTestCase {

  func testGetExercisesWords() {
    let exercisesExpectation = expectation(description: "Exercise data words")

    let urlString = Api.url.rawValue + ExerciseIdApi.words.description + Api.allExercises.rawValue
    Exercise.sharedInstance.getAllExerciseData(urlString: urlString,
                                               userDefaultKey: ExerciseUserDefaultKeys.words.description,
                                               store: false,
                                               completionHandler: { result in

      XCTAssertTrue(result)
      exercisesExpectation.fulfill()
      return()
    })
    waitForExpectations(timeout: 10, handler: nil)
  }
```

Figure 3.6.: iOS test preparation

Figure 3.6 shows a part of the code from the test file. Before the initialization of the test class, the application class has to be imported by `@testable import IdeRBlog_Exercises` to use classes of the application. The test methods do not differ from other methods, only the test class differs from other classes because it has to be derived from the `XCTestCase` class. With this derivation, the class is recognized as a test class and all methods in it are tested during execution.

Result

If the test methods are executed, a result is returned indicating which tests were successful and which were executed with errors. In Figure 3.7 such a result is shown, in which successful tests are marked with a green tick and faulty tests are marked with a red 'X'.

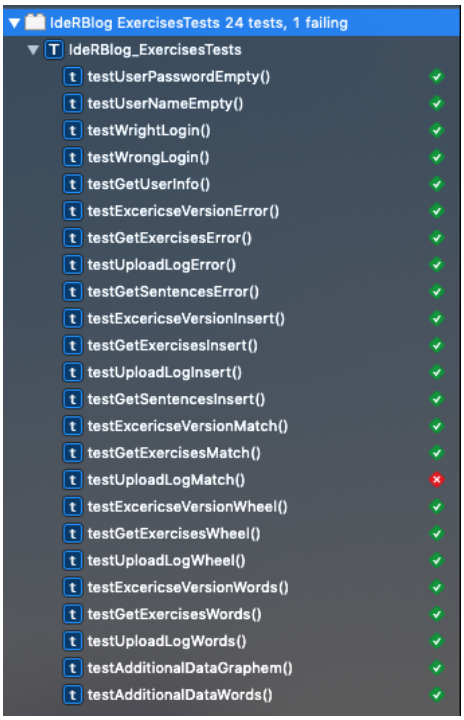


Figure 3.7.: iOS test result

3.4. Application

This chapter shows the developed prototype of the application and describes the operation and individual functionalities with the help of screenshots. There are always screenshots for the developed Android and iOS version of the application, because they differ marginally in design. The slightly different design is therefore, due to the fact that both applications were developed as a native version. This approach has the advantage that for each application standard functions and elements can be used, so the user is faced with a the familiar interface of the respective OS. In order to use the application in a meaningful way, a user account for the IDeRBlog platform is required that someone can log in to the application. As already mentioned there is no possibility to register a new user account in the application. For the illustration of the application an already existing test user account was used. At this point it should be mentioned that the application is currently only available in German, so the description always refers to the German text. Furthermore, all exercise formats including tasks have been designed and specified by professors of the University College of Teacher Education Styria.

3.4.1. Login Screen

When the application is started for the first time, the login screen in Figure 3.8 appears, where the username and password can be entered. The username input field is a standard input field without any validation of the inputted text. The inputted text in the password input field is invisible by default. The user has the possibility to toggle the visibility of the inputted text in the password input field by tapping the eye icon on the right side of the input field. The login data for the next application starts can be saved by activating the switch next to the label "Anmeldedaten merken" (remember login data). This functionality is only executed after a successful login and prevents the user to enter the login data each time the application is started. After entering the username and password in the respective input field, the button with the text "Anmelden" (login) is activated and the user can start the login process with a tap on it.

(a) Android

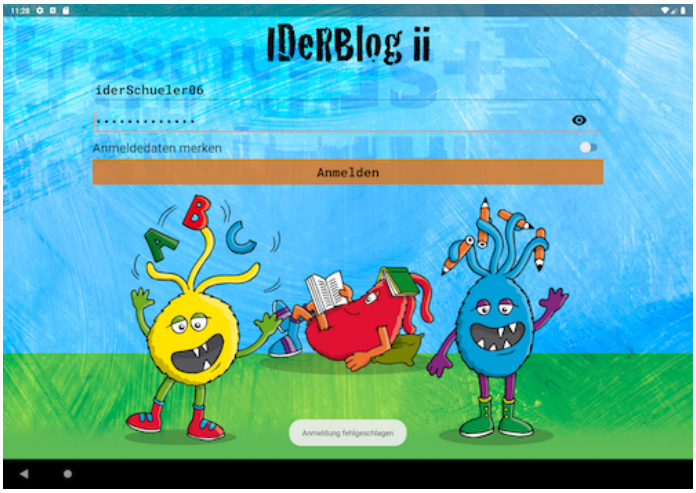
(b) iOS

Figure 3.8.: Login screen

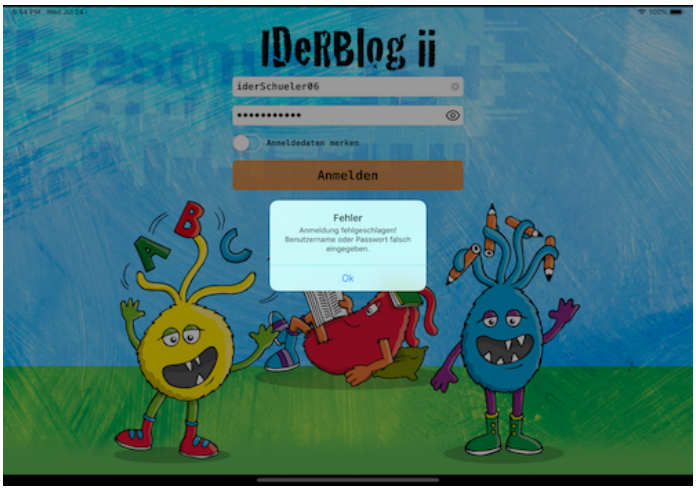
With the execution of this action a loading symbol is displayed so that the loading process is visually represented and a request with a HMAC[33] encrypted string which is generated from the username, the password which

[33] **https://bit.ly/2ypkp97 [accessed on 21.08.2019]**.

is hashed with the SHA-256[34] algorithm, an application key and a key,



(a) Android



(b) iOS

Figure 3.9.: Login screen - Error

is sent to the login server. If something goes wrong with this login process, such as there is no internet connection, the username or password was

[34] **http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf [accessed on 21.08.2019]**.

entered incorrectly or a wrong response is returned from the login server, then an error message is displayed, which can be seen in the Figure 3.9. The login screen will still be displayed and it is possible to restart the login process. If correct credentials have been entered and the server returns the response that the user has the permission to log in, the login screen will be hidden and the user will be redirected to the home screen which is explained in the next section.

3.4.2. Home Screen

After the user has successfully logged in, the so called home screen of the application is displayed. When the screen appears, the update process will be executed in the background. This process checks if new versions of the exercises are available on the server and these exercise data will be downloaded if it is necessary. In such a case, the new data is loaded from the server and a loading screen, which is shown in Figure 3.10, is displayed to the user until all data has been loaded successfully. If an unexpected error occurs during this loading process, a dialog box like in Figure 3.11 is displayed and the user has to confirm this with a tap on the ”Ok” button. In such a case, the user has always the opportunity to start this loading process manually by a tap on the button with the title ”Übungen laden” (download exercises). Since the application can also be used in an offline mode, that means no active internet connection is available, all exercise data are persisted and stored in the local memory. For this reason, all user actions during the completion of the exercises described in later sections will be saved locally if there is an error when uploading the data to the server. As soon as an Internet connection is available again, these stored data are uploaded to the server and deleted locally, so that a possible data loss can be excluded.

(a) Android

(b) iOS

Figure 3.10.: Home screen - Loading exercise data

(a) Android

(b) iOS

Figure 3.11.: Home screen - Loading exercise data error

After all exercise data have been loaded from the server, the user has the possibility on the home screen, which is shown in Figure 3.12, to select the different exercise formats by tapping on the corresponding field. If such a field is clicked, the user will be forwarded to the exercise list screen, which is described in section 3.4.3.

(a) Android

(b) iOS

Figure 3.12.: Home screen

The current user has the opportunity to log out by tapping on the button with the title "Abmelden" (logout). Before the logout process is started and the login screen Figure 3.8 is displayed again, the user has explicitly to confirm this process in a dialog box. This prevents the user from being logged out immediately if the button is tapped unintentionally. This function

is intended to allow users to log in with a different user account. The feature that a logged-in user is automatically logged out after a certain period of time is not yet available.

In addition to the selection of the various exercise formats, loading the exercise data and the possibility of logging out the current user, the first and last name of the currently logged in user are also displayed in the center bottom of the home screen. Below the name the points which are already reached by the user are also displayed. This score results from successfully completed tasks and exercises and is calculated from the server and transmitted back to the application. With this point score different tasks of an exercise can be unlocked by a user. This aspect should maintain the motivation for using the application and always new tasks will be available.

3.4.3. Exercise List

After selecting an exercise format on the home screen in Figure 3.12, a list of different exercises are shown. As an example, the list of all available key words exercises is shown in Figure 3.13. The representation of the exercise list for all other exercise formats is exactly the same, only the content displayed in the list is different. If there are too many entries in the list which can not be displayed on on page, the list entries can be scrolled.

At the top of the screen the title of the exercise format is displayed, so the user can see which exercise format was selected. In the lower left corner is a button with the title "Zurück" (back) that takes the user back to the home screen in Figure 3.12. In the list the exercise title is shown in a larger font at the top of each line. Below the title there is a short description of each exercise so the user can get an idea of the content of it. The number of exercises and their content is loaded from a server as already mentioned and that data is only shown without any possibilities to modify them.

(a) Android

(b) iOS

Figure 3.13.: Exercise list

The number of exercises and their activation changes with the already reached points as described in section 3.4.2. As shown in Figure 3.13, the unlocked exercises are not displayed in the list. A desired exercise can be selected and started by tapping on the list entry. As soon as this action has been executed, a request is sent to the server containing information about

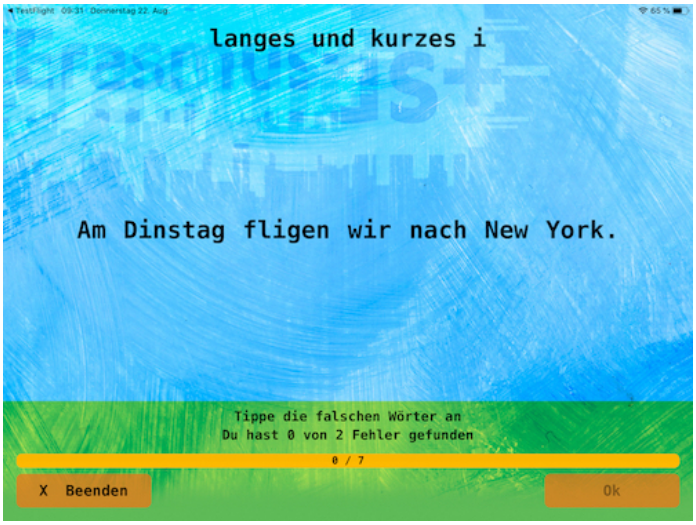
the start of the selected exercise.
The different exercise formats and their representation are explained in the following sections. The basic structure of the exercise screens is always the same, so this layout is only described once in the first section of all exercise descriptions. Therefore, the description of the remaining exercises does not go further into the basic structure of the screen.

3.4.4. Error Locations

In this section the first exercise format, the error locations exercise, is described and is reachable via the button with the title ”Fehlerstellen” (error locations) on the home screen in Figure 3.12. But before the exercise itself is described, the basic structure of the exercise screen is explained.

At the top of the exercise screen which is shown in Figure 3.14 is the title of the exercise which is selected in the previous exercise list screen in Figure 3.13. Also the elements in the lower area are always the same, so the yellow progress bar is showing the current progress within the exercise.As soon as a task of an exercise is completed, the progress bar is filled with a certain percentage of orange depending on the total number of tasks. In addition, the already completed tasks are displayed as text in the middle of the bar, so that the user knows immediately how many tasks still need to be done for the selected exercise. In the lower left corner is the exit button with the title ”Beenden” (exit), which can be used to cancel the current exercise. This operation has to be confirmed in an additional dialog box and if it is answered with yes, the user is redirected to the exercise list in Figure 3.4.3. The button in the lower right corner has more than one functions. With a tap on the button either a selection or a user interaction of a task is confirmed or the next task of an exercise is opened. This button is only clickable if all user interactions, which depend on the selected exercise format, have been performed. As mentioned above, this structure is the same for all exercise formats, so it is not described every time but only once. Therefore, this part is neglected for all other exercise formats.

(a) Android

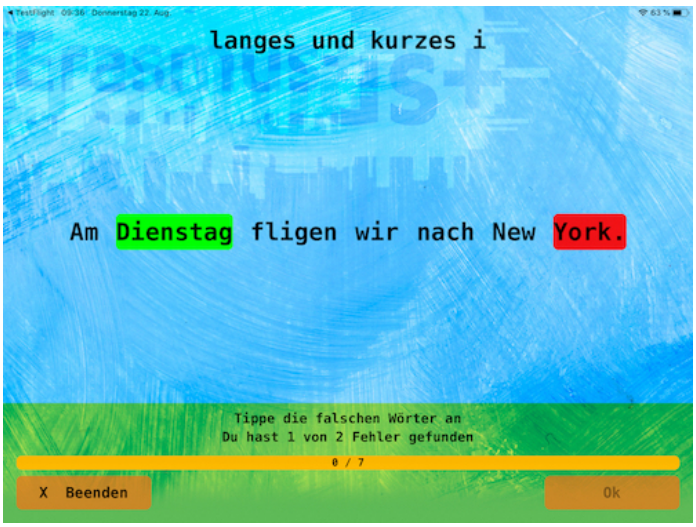


(b) iOS

Figure 3.14.: Exercise: Error Locations

The error locations exercise displays a sentence in the middle of the screen which contains one or more incorrect words. Each word is displayed as a button which can be clicked by the user and has different actions which are explained below.

(a) Android

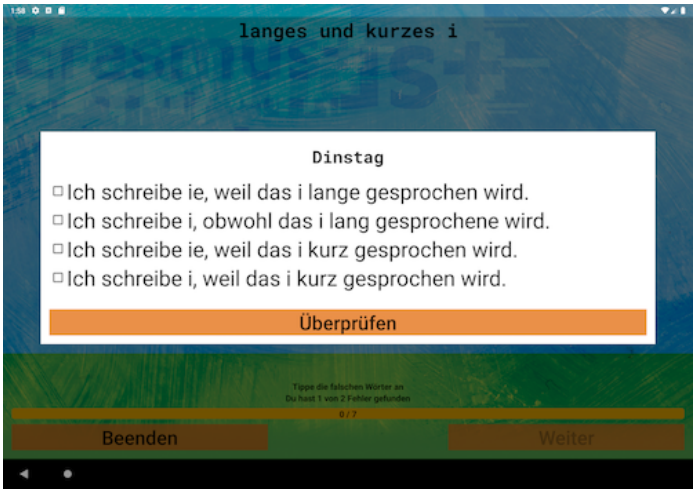


(b) iOS

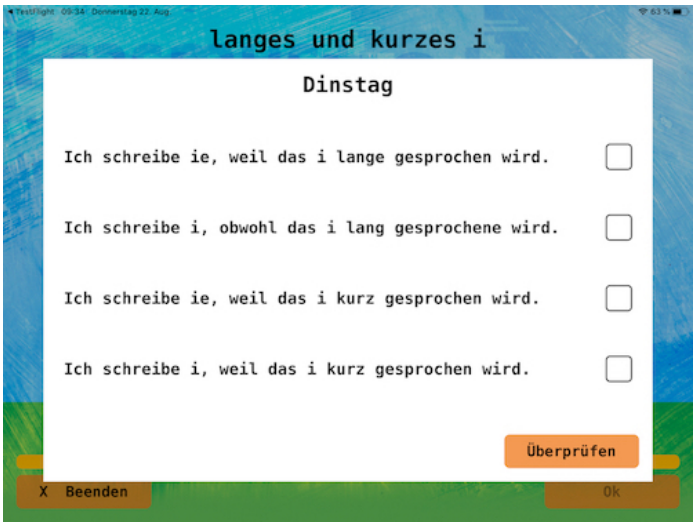
Figure 3.15.: Exercise: Error Locations - Mistake

If a correctly spelled word is mistakenly tapped, it flashes briefly red on as shown in Figure 3.15 and the user recognizes that a mistake has been made. This failure will be uploaded with different information by a request to the server and this influences in a negative sense the amount of winning points

for the user. When tapping a misspelled word which is in this exercise a correct user action, a popup window appears in the center of the screen which is shown in Figure 3.16.



(a) Android



(b) iOS

Figure 3.16.: Exercise: Error Locations - Evaluation

At the top of this window the word is shown as title which is tapped in the previous exercise screen. Below is a list of various justifications that should describe why the word in the title is misspelled. A checkbox is displayed next to each justification, which can be checked or unchecked by the user via a tap gesture. It is possible to mark neither, one or more answers. If the user is of the opinion that all correct answers have been selected, the selection can be checked with a tap on the button with the title "Überprüfen" (check). In Figure 3.17 is such a evaluation displayed where all correct answers are highlighted in green and all wrong answers in red. Here also all wrong answers with all necessary information are sent to the server by a request. After the check the popup window can be closed with a tap on the button "Schließen" (close) and the exercise screen in Figure 3.14 is shown again.

After an error has been found, the incorrectly written word is replaced with the correctly written word in the sentence and additionally highlighted with a green color. This allows the user to recognize which errors have already been found. In addition, a single word can only be selected once, so an incorrectly selected word is only recognized as an error once and sent to the server, even if it is selected several times by the user. To let the user know how many errors still need to be found, the number is displayed in a hint text above the progress bar. After all errors have been found, the button with the title "Weiter" (next) is enabled and the exercise can be continued with a tap on it.
If all tasks are completed and the user tap on the button with the title "Fertig" (finish), a window is shown which provides a summary of how all tasks of the exercise were solved.

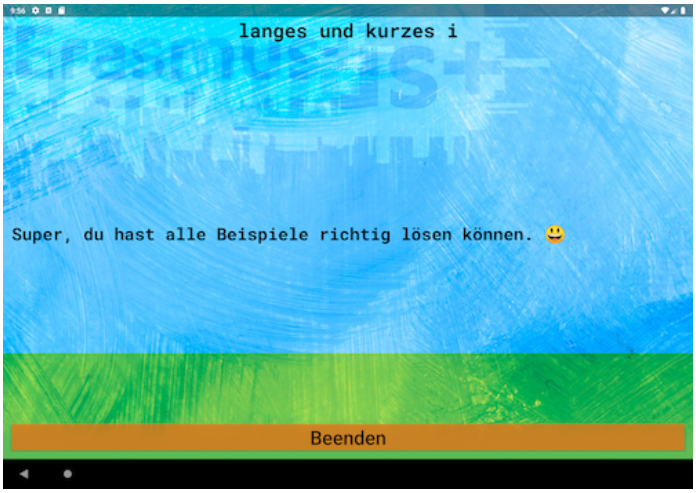
(a) Android

(b) iOS

Figure 3.17.: Exercise: Error Locations - Evaluation Result

When the text ”Super, du hast alle Beispiele ohne Fehler richtig lösen können. 😃” (Excellent, you were able to solve all examples correctly without any mistakes.) is displayed in the center of the screen like in Figure 3.18, then all tasks have been solved correctly. Otherwise the text ”Oh je, du hast nicht alle Beispiel fehlerfrei lösen können. Probiere es erneut, dass nächste

Mal gelingt es dir besser! 👍" (Oh dear, you haven't been able to solve all the examples without errors. Try it again, the next time you will succeed better!) will appear after at least one mistake has been made in a task of the exercise.



(a) Android

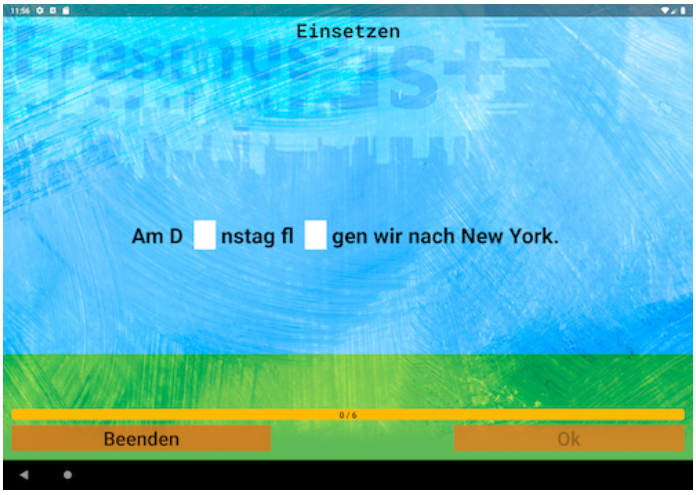
(b) iOS

Figure 3.18.: Exercise: Summary

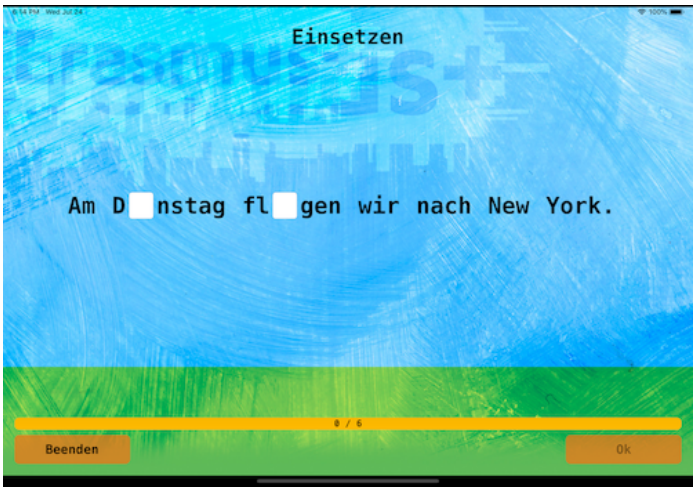
As with the exercise list, this screen is displayed after each exercise for

all exercise formats, therefore, this summary is only described within this exercise format and is not mentioned in detail in all other ones.

3.4.5. Insert



(a) Android



(b) iOS

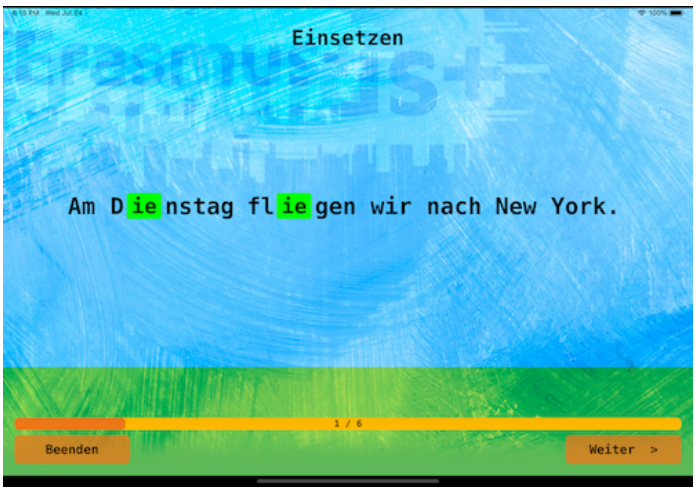
Figure 3.19.: Exercise: Insert

This section explains the next exercise format, which can be selected via the button with the title ”Einsetzen” (insert) on the home screen in Figure 3.4.2. The basic structure of the exercise screen is described in more detail in section 3.4.4.
After an exercise is selected in the exercise list in Figure 3.13, a sentence is displayed that contains text gaps in various places, as shown in Figure 3.19, which must be filled by the user. Each gap is a separate text input field in which a maximum of three characters can be entered. Depending on the configuration of the exercise, it may be possible to display a help in form of a text. This text is displayed above the gap text when the user taps on an input field to enter letters. As soon as the keyboard is hidden again, the help text is also hidden. Such a help text is displayed with gray text color and should give the user some hints which letters could be the wright on for the gap. This help text is set by a configuration parameter and can either never be displayed, only after a certain number of incorrect entries or immediately.

When all text gaps are filled in, the ”Ok” button is enabled and can be pressed. After this action the text in each text input field will be checked. If the gap is filled in correctly, the input field is highlighted in green and the field can no longer be edited by the user. If the entered word part is incorrect, the text field is highlighted in red and this error is sent to the server via a request. All incorrect input fields must be filled in and checked by the user as often as all gaps are correct. It should be noted that every error, even if it is the same gap more than once, is sent to the server. Figure 3.20 shows a correctly filled out sentence and the user can continue the exercise by clicking on the button with the title ”Weiter” (next) and solving the next gap text.

The end of the exercise and the display of the summary are identical to those described in the section 3.4.4.

(a) Android



(b) iOS

Figure 3.20.: Exercise: Insert - Evaluation

3.4.6. Keywords

The keyword exercise format is explained in this section. This exercise can be configured by different parameters in many ways, whereby two examples

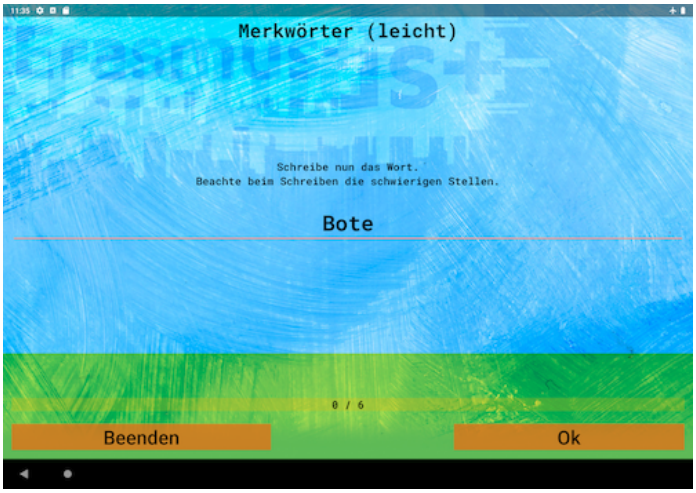
are shown in the following figures.
The first example can be seen in Figure 3.21. After selecting a exercise from the list which is explained in section 3.4.3, beneath the basic structure, which is described in the section 3.4.4, a short description of the task, what the user has exactly to do, is displayed first below the title.

(a) Android

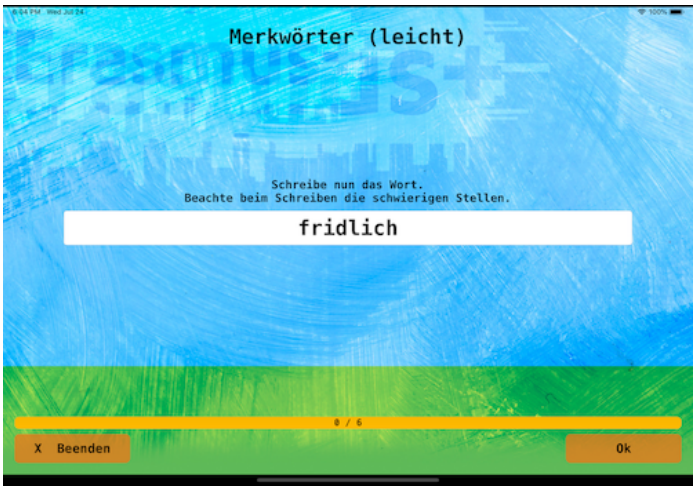
(b) iOS

Figure 3.21.: Exercise: Keywords 1

In the middle of the screen a word is displayed, which is divided into its individual letters.



(a) Android

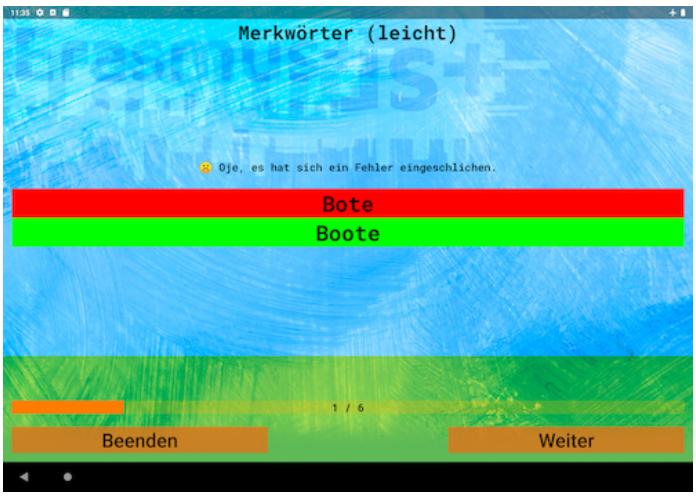


(b) iOS

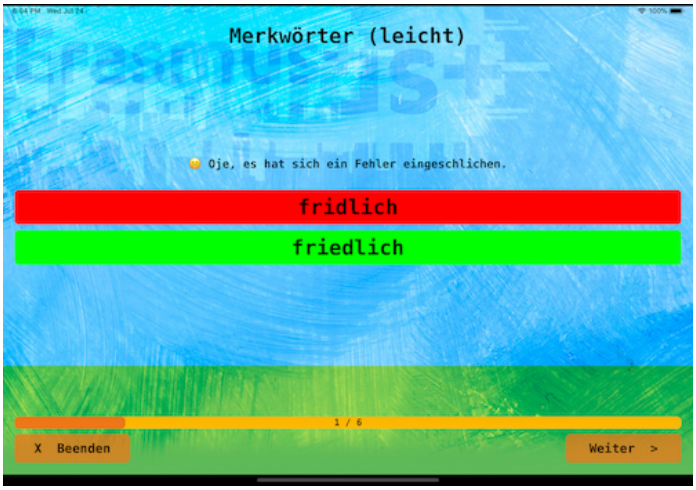
Figure 3.22.: Exercise: Keywords 1 - Input word

If the single letters are underlined, every letter can be tapped and the user is asked to mark one or more letters. A marked letter is colored in orange

and is to indicate possible difficult parts within the word. All marked letters are transmitted after continuing the task to the server by request. This log entry can be used for interesting statistical evaluations for teachers.
Below the word a further hint for the process of the task is displayed.



(a) Android



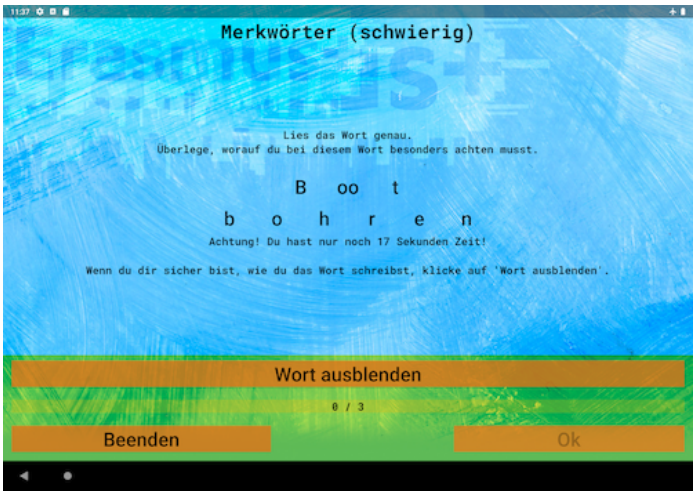
(b) iOS

Figure 3.23.: Exercise: Keywords 1 - Evaluation

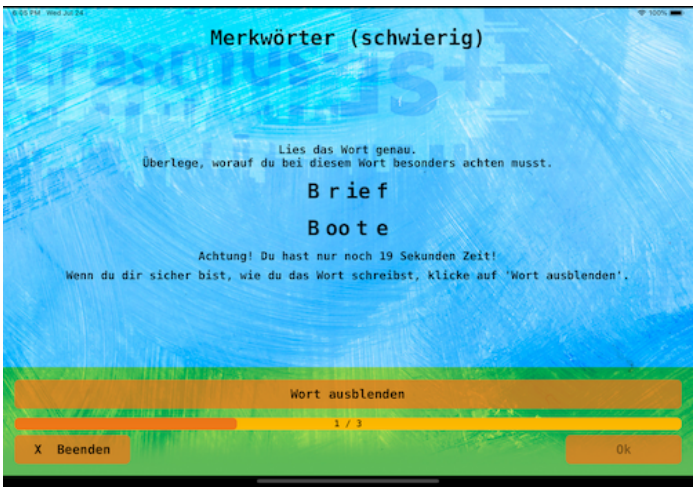
As soon as one or more letters are marked, the button with ”Wort ausblenden” (hide word) becomes active and can be pressed. After this action the screen in Figure 3.22 appears where a text input field is displayed in the middle. The previously shown word has to be entered into the input field by the user and by tapping on the button with the title ”Ok” the inputted word can be checked.
If the word was entered correctly, a screen will appear with the message that everything was correct. If the user has written the word incorrectly, the entered word and the word to be written are displayed one below the other, with the incorrect word highlighted in red and the correct word highlighted in green which is shown in Figure 3.23. This allows the user to see the mistakes made and compare both words. Furthermore, this incorrect entry is also sent to the server by a request. In both cases the exercise can be continued by tapping the button with the title ”Weiter” (next) until all words are written and the summary is shown.

The second example of the exercise which is shown in Figure 3.24 is slightly more difficult as the described version above. In this configuration, two words are displayed at the same time. To further increase the difficulty, there is also running a timer that is displayed below the words, which limits the time for remembering the words. After the timer has expired, the same action will be executed as for a tap on the button with ”Wort ausblenden” (hide word). Only the individual letters are not underlined, which means that no letters can or must be marked. In the next screen, two input fields are displayed accordingly for the input of the words, however, the order of the words does not have to be paid attention to. When the words are checked, the misspelled words and all specified words are displayed in the case of an error. All other steps of the exercise are identical to the first example.

These two examples show that by combining the different parameters, the exercises can be made easier or more difficult and individually adapted to the users.



(a) Android



(b) iOS

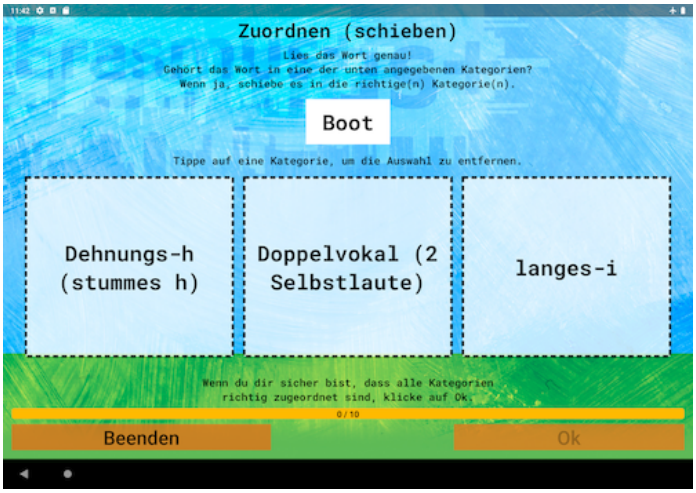
Figure 3.24.: Exercise: Keywords 2

3.4.7. Match

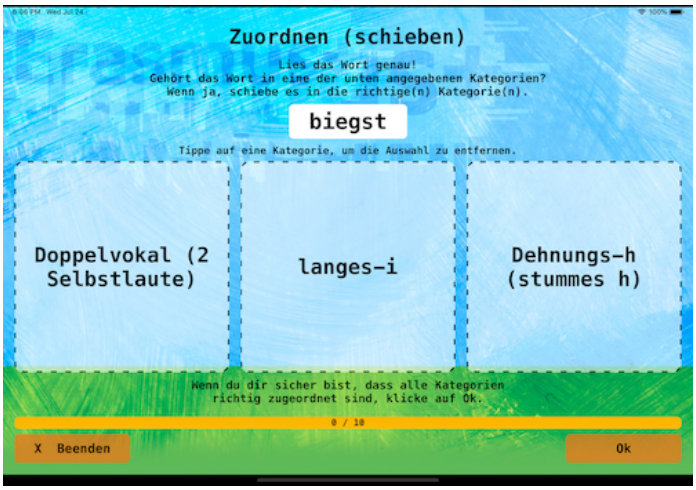
The next exercise format is the match exercise which is explained in this section. There are also several configurations for this format, whereby two

different exercise forms can be formed, which are described in more detail in the following illustrations.

In the first configuration, after starting an exercise from the list which is described in section 3.4.3, the screen in Figure 3.25 is displayed.



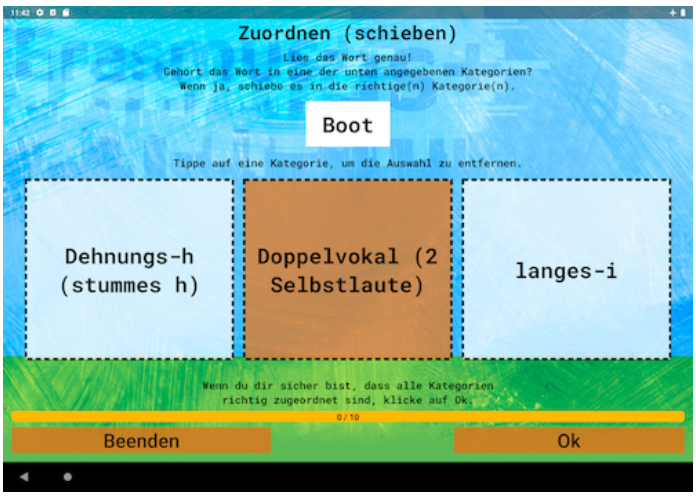
(a) Android



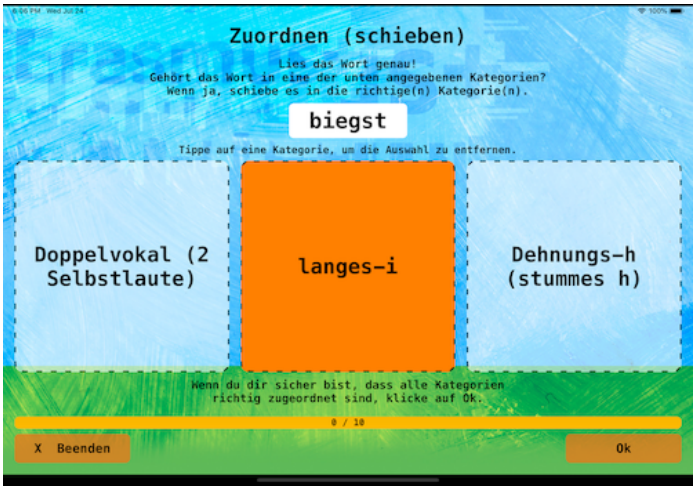
(b) iOS

Figure 3.25.: Exercise: Match Drag & Drop

Beside the basic structure of the screen, which is explained in detail in Section 3.4.4, the first paragraph shows a short description of the exercise, what the user has to do. Below the description is a white field containing the word to be practiced.



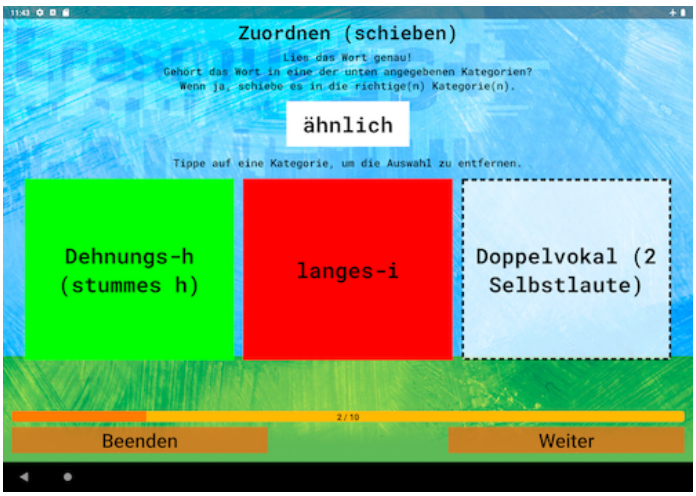
(a) Android



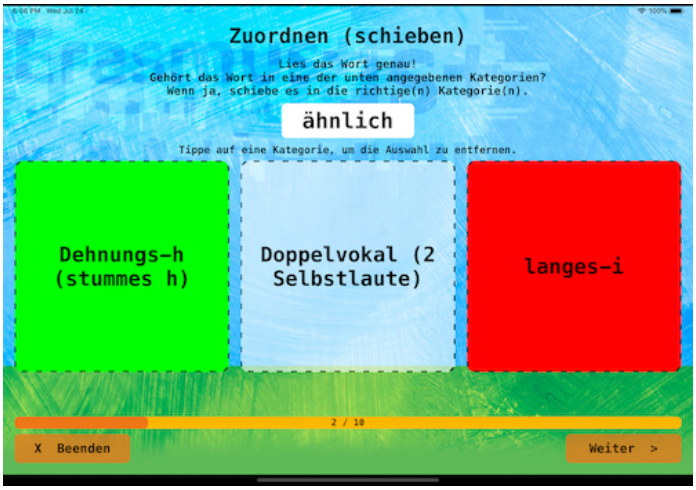
(b) iOS

Figure 3.26.: Exercise: Match Drag & Drop - Selection

This field can be moved arbitrarily by a drag and drop gesture and should be dragged into one or more large fields shown in the middle of the screen which serve as containers for different word categories.



(a) Android



(b) iOS

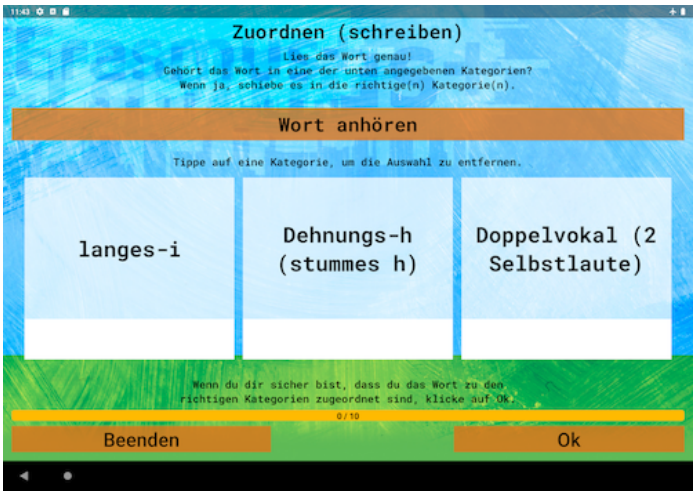
Figure 3.27.: Exercise: Match Drag & Drop - Result

If such a category is selected, it will be highlighted in orange as shown

in Figure 3.26. After a wrong category was selected by mistake, it can be deselected by a tap on the corresponding field. The selected categories can then be checked by tap on the button with the title ”Ok”. If the selection of the categories is correct, a text is displayed instead of the category fields that everything is correct. If the selection contains an error, all correct categories are highlighted in green and wrong ones in red as shown in Figure 3.27. In this exercise, incorrect selectors are also sent to the server by request.
After the check the exercise can be continued with a tap on the button with the title ”Weiter” (next) until all tasks are completed and the summary screen is shown.

The second type of this exercise is much more difficult than the previously described form. Instead of the white field with the displayed word from the first configuration, a button with the title ”Wort anhören” (listen to the word) is displayed which is shown in Figure 3.28. By a tap on it a word is spoken to the user, which belongs written into one or more text input fields. Each input field is located below a big white field in the middle of the screen, which stands for a word category. In this type of exercise, the user is asked to listen carefully which word is required and which category it could belong to, and to pay attention to the correct spelling of the word later on. The check of the selection and control on the correct spelling of the word can be executed again with a tap on the button ”Ok”.

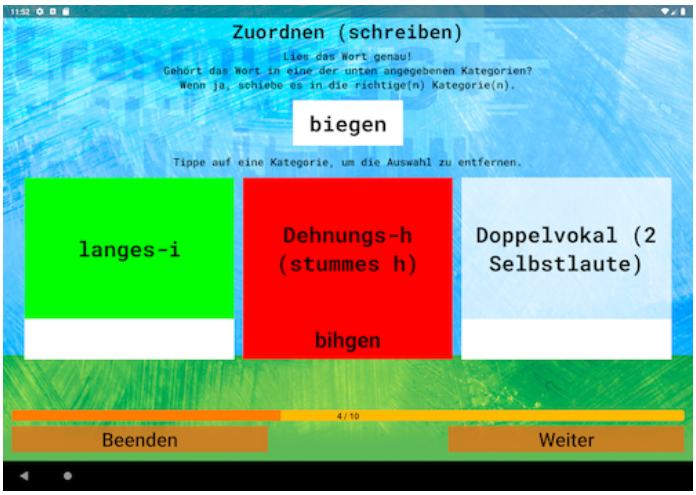
This verification validates two different things, whereby each error is also sent to the server via request. On the one hand the correct spelling of the inputted word is checked and on the other hand the correct assignment of the categories. If the word has been assigned to the correct category but written incorrectly, the large category field is highlighted in green and the text field in red. If the word is assigned to a wrong category, the field and the corresponding text field will be highlighted in red, regardless of whether the word was entered correctly or incorrectly, as shown in Figure 3.29.



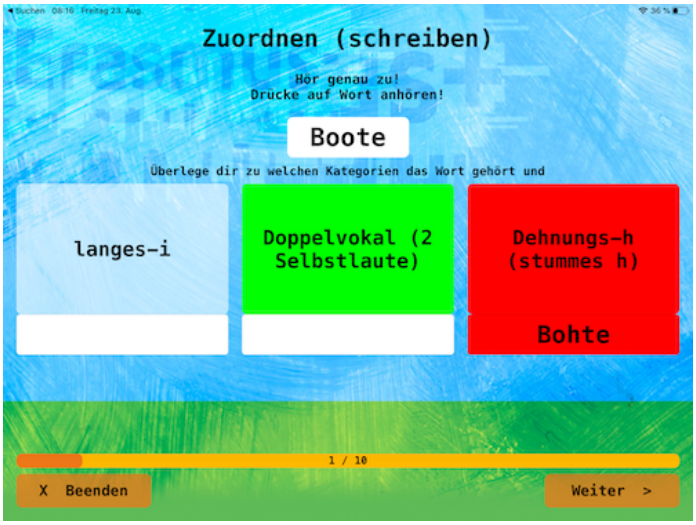
(a) Android

(b) iOS

Figure 3.28.: Exercise: Match Write



(a) Android



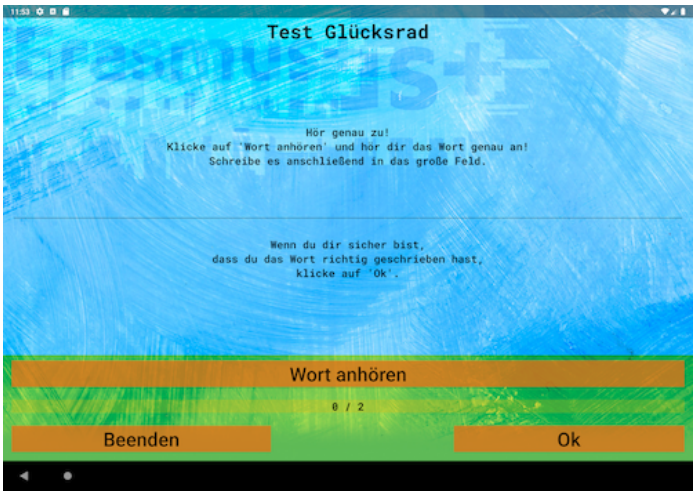
(b) iOS

Figure 3.29.: Exercise: Match Write - Result

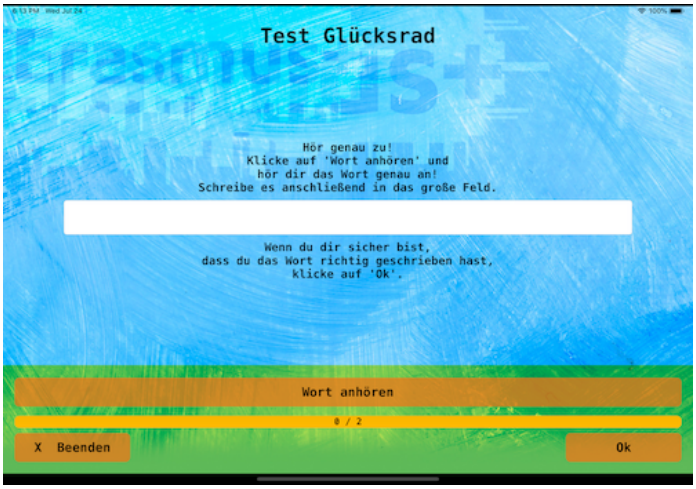
After the check the exercises can be continued again via tap on the button with ”Weiter” (next) until the summary screen is displayed.

3.4.8. Wheel of Fortune

The last exercise format is similar to second described configuration of the keywords exercise in section 3.4.7. After selecting a exercise in the list which is described in section 3.4.3 the screen in Figure 3.30 is shown.



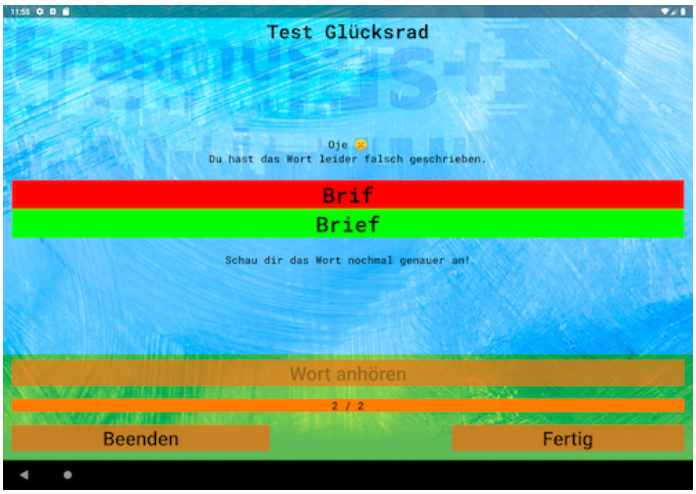
(a) Android



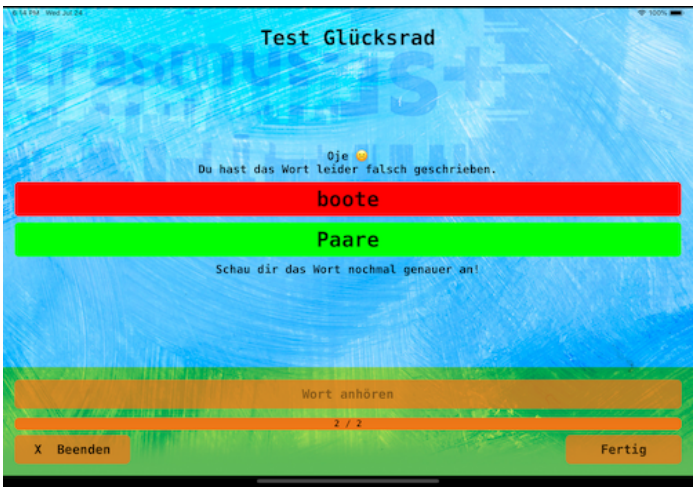
(b) iOS

Figure 3.30.: Exercise: Wheel of Fortune

In addition to the basic structure of the exercise screen, which is explained in detail in Section 3.4.4, a description text is displayed at the beginning to explain the task to the user. In the middle of the screen a text input field is shown and above the progress bar a button with the title "Wort anhören" (listen to the word) is to be seen.



(a) Android



(b) iOS

Figure 3.31.: Exercise: Wheel of Fortune - Result

By a tap on this button a word is spoken, which should be written into the input field by the user. The word can be listened to as often as needed by tapping the button. The check of the input can be executed by tapping the button ”Ok”. If everything is correct, a corresponding hint text is displayed in the middle of the screen that everything was done correctly. If an error occurs, the wrong word is displayed in red and the correct one in green among each other which is shown in Figure 3.31. So the user can compare what has been misspelled and by tap on the button ”Weiter” (next) the exercise can be continued until the summary screen is displayed. In the example shown, there is only one task in the whole exercise, so the button with ”Fertig” (finish) is displayed instead of ”Weiter” (next), so the summary is displayed immediately after the first completed task.

4. Evaluation

This chapter covers the evaluation process of the software. The process is subdivided into a preparation, procedure and result section.
The evaluation in form of a UX test is part of this Master's Thesis and an important aspect in the development of software, especially in the development of mobile applications. In addition, it must be noted that the usability of the application has to be designed for children since the approach and understanding of children is not comparable to that of adults. This requires a structure of the application which is as simple as possible.

It was planned that the UX test on iPads would take place in a third grade elementary school at the practical elementary school at the University College of Teacher Education Styria. Unfortunately there were time and administrative problems to install the mobile applications on the school's iPads. Therefore, alternative test subjects were needed. The UX test was done by five children between the ages of eight and eleven. The school level starts at the third elementary school level and reaches up to the first secondary school level.

4.1. Preparation

As the TU Graz has a cooperation with the University College of Teacher Education Styria the practical elementary school of the this university was chosen as test school for the UX test. Since a web application and a mobile application were programmed for the project, two different UX tests have to be carried out. Therefore, it was planned that about ten children test the web application and the same number of children test the mobile application. For the evaluation of each exercise, every child is asked to fill a

short feedback questionnaire before starting the next exercise. At the end of the UX test of one application, the two groups swap from web to mobile application and vice versa. A teacher of the practical elementary school has agreed to let us do the UX test in a third grade elementary school class. The computers in the school were tested, if the web application can be executed on the available web browser, so nothing could get wrong during the test. Unfortunately, the configuration and installing of the mobile application on the iPads of the elementary school turned out to be more difficult than expected. The responsible person, who determines which applications may be installed on the iPads, was not reached in time. Therefore, it was not possible to install the application either directly or via TestFlight[1] on the iPads. Due to these circumstances only the web application was tested in the elementary school on 25.06.2019. For the UX test of the mobile application it was agreed the application would be tested by five children in the private sector on three private iPads. As already mentioned, this test was done on 11.07.2019 by a girl and two boys who attended the third grade elementary school. The other two children, a girl and a boy, were already a little older and attended the first grade secondary school.

In order to carry out the UX test at all, access to the user management server for the application was provided on the production server in advance. This was necessary to have access to the user management service outside the local network of the Technical University of Graz and to be able to log into the application. Furthermore, the data server from which the exercise data is loaded had to be deployed from the development environment to the productive system. This would not have been absolutely necessary, since the exercise data can also be loaded from a separate server instance in the local network. However, this greatly simplified the configuration of the application as no additional tools were required.

[1] **https://developer.apple.com/testflight [accessed on 12.07.2019]**.

4.2. Procedure

First of all I introduce myself to the children, because not all children knew me before. Before the UX test was started, all five children were briefly explained why they should do this test at all. It was also very important to explain to the children that this test does not evaluate the children's performance, but the children should test and evaluate the usability and comprehensibility of the application. The feedback questionnaire, which had to be completed after each exercise, was presented too. For the UX test three iPads were used, so the children were divided into a group of three and a group of two children. The first group started with the UX test and after each child was assigned an iPad, they began solving the first exercise. After a child completed an exercise, the kid filled in the questionnaire. I also asked short questions about the exercise, such as how the exercise procedure was, whether there were any problems or suggestions for improvements and I documented them. Afterwards the child was allowed to start the next exercise. After about 40 minutes each child in the first group had finished all exercises and the same procedure was repeated with the second group, which took about 40 minutes too.

The entire procedure of the UX test took place without any incidents or problems and the application had no unexpected bugs or crashes. From the beginning all children were very attentive and interested in solving the exercises. They were able to cope very well with the given tasks and had much fun solving the exercises. To my astonishment I was even asked why there are not more different tasks for every exercise or when new exercise formats would be added to the existing ones. Looking back, I would not have thought that the evaluation would go so smoothly and it was fun to watch the children during practice.

4.3. Results

Since this project is the first one for me to address mainly primary school children as the target group, it was very exciting to observe how the pupils cope with the usability of the application. The UX test was made by five

children, the result of the evaluated questionnaires is not that representative. As already mentioned above, the group of testers would have been much larger if the availability of the application had worked in the practical elementary school. However, even with a small group of testers I got a good overview of whether the usability of the application was understandable and if the pupils were able to cope without external help. At this point it should be mentioned that all testers had some experience with the handling of tablets. The result of the questionnaires, which is illustrated in Table 4.1 and Table 4.2, would be probably different with testers who have less experience with tablets.

	I had understood the assignment definition.	I was able to solve the problems myself.
Error locations	1	1
Insert	1,2	1
Keywords - Easy	1	1
Keywords - Hard	1	1
Match - Drag & Drop	1	1
Match - Write	1	1
Wheel of fortune	1	1
Average	**1,03**	**1**

Table 4.1.: Result of the evaluation questionnaire (1)

If someone take a closer look at the individual result values or the average values of each exercise, one immediately notices that the test children rated the UX test very well overall. This result fully reflects my observations during the test. The first question in Table 4.1 refers to the understanding of the tasks. Since the Insert exercise was the very first exercise that was solved by the children, only one child was somewhat uncertain what was exactly

	I had fun to practice.	I would also like to do the exercise at home.
Error locations	1,2	1,2
Insert	1,2	1,2
Keywords - Easy	1	1
Keywords - Hard	1	1
Match - Drag & Drop	1	1,2
Match - Write	1	1,5
Wheel of fortune	1	1
Average	**1,06**	**1,16**

Table 4.2.: Result of the evaluation questionnaire (2)

required or how the usability of the app works. Therefore, the Insert exercise was rated slightly worse than the other exercises. The other four children had no problems with the usage of the applications or the understanding of the exercises from the beginning.

The second question in Table 4.1 was given the highest score by all children, as no child needed my help during the whole UX test. This result shows that all exercises were comprehensibly described and implemented.

The first question in Table 4.2 shows whether the children had fun during solving the different exercises or not. The result shows that the Insert and Error locations exercise still has potential for some improvements, whereby one could redesign these exercises and make them more interesting for the user. Finally, the second question in Table 4.2 was asked whether the children would also practice these exercises outside their school time, for example at home or in their free time. This question was rated worst of all, which means that the application is likely to be used mainly in school in

the teaching lessons. In order to increase the use of the application outside of school, a possible solution could be to force the game factor within the application. However, this can lead to a risk that the basic idea of a learning app will fade into the background.

After the children had filled out all questionnaires, I asked each child about possible improvements. In this final survey, the children mentioned several times that the voice output of the Wheel of Fortune and Match-Write exercise was very difficult to understand. This point should be improved in any case. Currently the application uses the speech output of the operating system, which someone has no influence on the quality of the speech output. In the future, it will be possible to have all words spoken by a natural person so that these sound recordings can be used for the speech output of the words.

5. Conclusion

The aim of this Master's thesis was to develop a prototype of mobile applications for the two OSs Android and iOS for German language acquisition based on LA measurements. The main target group of the application will be pupils of later primary school age, as it can be assumed that children of this age already have experience in using smartphones or tablets. The application developed in the context of this thesis will serve as a basis for the IDeRBlog ii project, which will be expanded in the future with further exercise content. In addition, the connection to the already available intelligent dictionary will be realized, so that the learners can practice their individual words and sentences, which are collected by analyzing already practiced tasks. (Aspalter, Edtstadler, and Martich, 2017)

The first chapter of this thesis deals with the use of digital media in schools. Since this aspect is seen very critically by many people, a possible effect and the resulting advantage of digital media in the teaching process is explained. The research has shown that there has been research in this area for some time on how to use digital media in the school. But I am of the opinion that the necessary technical equipment have recently been made available for everyone so that they can also be used meaningfully and effectively by pupils and the way they learn can change radically. However, a good compromise between classical and digital media should be used in teaching, which discussion is not part of this Master's thesis. Afterwards a short insight into LA is given and illustrated with the aim of a reference model. Through the digitization of exercise content, the collection and evaluation of the resulting data is enormously important for this process to be continuously improved, as it becomes more and more difficult for teachers to assist and control. At the end of the first chapter, four mobile applications are described, which also have to do with language learning and should serve as examples of already available applications in this field.

The search for free applications, which should have a similar teaching content as the IDeRBlog ii application, showed that the offer is very small. Therefore, the IDeRBlog ii application with the possibility of individual exercise content and the interaction of teacher and student in this area may have some Unique Selling Point (USP)s.

During the development of the application, the individual exercise formats were elaborated by professors from the University College of Teacher Education Styria and their technical implementation was implemented by me under constant mutual agreement. However, this process had the big disadvantage that the requirements changed several times during the development and therefore, often developed code had to be discarded. For the design of the application and the UI a simple and understandable handling was considered, so that the application can be used by children without problems. In addition, it should be noted that the design does not distract too much from the teaching content, but a certain gaming factor should be taken into account so that the use is fun for children over a longer period of time. Throughout the entire development phase, the cooperation with all persons involved worked perfectly and the API, which provides the exercise content and is responsible for logging data, could also be integrated without problems.

In summary, the evaluation of the application in the form of a UX test with children showed that the first version of the application was well accepted and that the children enjoyed practicing it. In the future, the application should be evaluated by several children, as a test with only five children cannot be considered very representative.

Bibliography

Apple (2014). *Platform Architecture*. URL: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html (visited on 07/21/2019) (cit. on p. 40).

Aspalter, Christian, Konstanze Edtstadler, and Susanne Martich (2017). "Recht-/schreibdidaktische Überlegungen für den Einsatz der Internetplattform IDeRBlog." In: (cit. on p. 83).

Chatti, Mohamed Amine et al. (2013). "A reference model for learning analytics." In: *International Journal of Technology Enhanced Learning* 4.5-6, pp. 318–331 (cit. on pp. 9–14).

Dougiamas, Martin and Peter Taylor (2003). *Moodle: Using Learning Communities to Create an Open Source Course Management System*. Ed. by David Lassner and Carmel McNaught. Honolulu, Hawaii, USA: Association for the Advancement of Computing in Education (AACE), pp. 171–178. URL: https://www.learntechlib.org/p/13739 (cit. on p. 11).

Duval, Erik (2011). "Attention please!: learning analytics for visualization and recommendation." In: *LAK* 11, pp. 9–17 (cit. on p. 7).

Eickelmann, Birgit (2010). *Digitale Medien in Schule und Unterricht erfolgreich implementieren: eine empirische Analyse aus Sicht der Schulentwicklungsforschung*. Vol. 19. Waxmann Verlag (cit. on p. 1).

Ferguson, R. (2012). "Learning analytics: drivers, developments and challenges." In: pp. 304–317. ISSN: 1753-5263. URL: http://oro.open.ac.uk/id/eprint/36374 (cit. on pp. 7, 8).

Google (2019a). *Fundamentals of Testing*. URL: https://developer.android.com/training/testing/fundamentals (visited on 07/22/2019) (cit. on p. 28).

Google (2019b). *Platform Architecture*. URL: https://developer.android.com/guide/platform (visited on 07/21/2019) (cit. on pp. 32, 33).

Herzig, Bardo (2014). "Wie wirksam sind digitale Medien im Unterricht." In: *Gütersloh: Bertelsmann Stiftung* (cit. on p. 4).

Johnson, L., S. Adams, and M. Cummins (2012). "The NMC Horizon Report: 2012 Higher Education Edition." In: p. 42 (cit. on p. 7).

Kerres, Michael (2002). "Bunter, besser, billiger? Zum Mehrwert digitaler Medien in der Bildung (Why? On the Benefit of Digital Media in Education)." In: *it-Information Technology* 44.4, pp. 187–192 (cit. on p. 6).

Krikun, I. (Nov. 2017). "Applying learning analytics methods to enhance learning quality and effectiveness in virtual learning environments." In: pp. 1–6. DOI: 10.1109/AIEEE.2017.8270532 (cit. on p. 7).

Lardinois, Frederic (2019). *Kotlin is now Google's preferred language for Android app development*. URL: https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development (visited on 09/03/2019) (cit. on p. 34).

Lu, June, Chun-Sheng Yu, and Chang Liu (2003). "Learning style, learning patterns, and learning performance in a WebCT-based MIS course." In: *Information and Management* 40.6, pp. 497–507. ISSN: 0378-7206. DOI: https://doi.org/10.1016/S0378-7206(02)00064-2. URL: http://www.sciencedirect.com/science/article/pii/S0378720602000642 (cit. on p. 11).

Martin, Florence (2008). "Blackboard as the learning management system of a computer literacy course." In: *Journal of Online Learning and Teaching* 4.2, pp. 138–145 (cit. on p. 11).

McLellan, Hilary (Sept. 2007). "Digital storytelling in higher education." In: *Journal of Computing in Higher Education* 19.1, pp. 65–79. ISSN: 1042-1726. DOI: 10.1007/BF03033420. URL: https://doi.org/10.1007/BF03033420 (cit. on p. 3).

Prasse, Doreen et al. (2016). "Lernen und Unterrichten in Tabletklassen. 1." In: *Zwischenbericht zur wissenschaftlichen Begleitforschung (Impact Measurement) der Smart Classrooms Switzerland* (cit. on p. 1).

Robin, Bernard R. (2008). "Digital Storytelling: A Powerful Technology Tool for the 21st Century Classroom." In: *Theory Into Practice* 47.3, pp. 220–228. DOI: 10.1080/00405840802153916. URL: https://doi.org/10.1080/00405840802153916 (cit. on p. 3).

Rupesh (2017). *iOS Layered Architecture*. URL: https://codeingwithios.blogspot.com/2017/09/ios-layered-architecture.html (visited on 07/21/2019) (cit. on p. 39).

Tam, Audrey and Michael Katz (2019). *iOS Unit Testing and UI Testing Tutorial*. URL: https://www.raywenderlich.com/960290-ios-unit-testing-and-ui-testing-tutorial (visited on 07/23/2019) (cit. on p. 27).

Tsadok, Avi (2019). *How To Solve Problems With UI Testing*. URL: https://medium.com/better-programming/having-troubles-with-ui-testing-heres-how-to-nail-it-cee2a90e5ce5 (visited on 07/24/2019) (cit. on p. 30).

Tulodziecki, Gerhard, Bardo Herzig, and Sigrid Blömeke (2017). "Gestaltung von Unterricht: eine Einführung in die Didaktik." In: 3311 (cit. on p. 3).

Wang, Qiyun et al. (2012). "Using the Facebook group as a learning management system: An exploratory study." In: *British Journal of Educational Technology* 43.3, pp. 428–438. DOI: 10.1111/j.1467-8535.2011.01195.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8535.2011.01195.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8535.2011.01195.x (cit. on p. 11).

Appendix

Appendix A.

UX-Feedback Sheet

ÜBUNG: USER:

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.	

Appendix B.

UX-Feedback Sheet - Evaluation

Einsetzen

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	2
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	2
Ich möchte die Übung auch zuhause machen.	2

Einsetzen

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Einsetzen

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Einsetzen

ÜBUNG: USER:

PC ☐
iPad ☑

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Einsetzen

ÜBUNG: USER:

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.	

Fehlerteufel

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	2
Ich möchte die Übung auch zuhause machen.	2

Tehlerstellen (langer und kurzes i)

ÜBUNG: USER:

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.	

Fehlerstellen

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Fehlerstellen langes & kurzes i

ÜBUNG: USER:

PC ☐
iPad ☑

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

ÜBUNG: Fehlerstellen USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Glücksrad

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Glücksrad

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Glücksrad

ÜBUNG: USER:

PC ☐
iPad ☑

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Glücksrad

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Glücksrad

ÜBUNG: USER:

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.	

Merkblätter (5)

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Merkwörter leicht

ÜBUNG: USER:

PC ☐
iPad ☑

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Merkwörter 1

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Merkwörter(L)

ÜBUNG: USER:

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.	

Merkwörter (fl)

ÜBUNG:

USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Merkwörter schwierig

ÜBUNG: USER:

PC ☐
iPad ☑

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Merkwörter (schwierig)

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Zuordnen (Schreiben)

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	2

Zuordnen (schr)
~~Zuordnen~~

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Zuordnen(Schreiben)

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	2

Zuordnen schreiben

ÜBUNG: USER:

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.	

Zuordnen schieben

ÜBUNG: USER:

PC ☐
iPad ☒

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Zuordnen schreiben

ÜBUNG: USER:

PC ☐
iPad ☑

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1

Zuordnen Schieben

ÜBUNG: USER:

PC
iPad

1 2 3 4 5

Ich wusste, was die Aufgabenstellung war.	1
Ich konnte die Aufgabe selber lösen.	1
Es hat mir Spaß gemacht zu üben.	1
Ich möchte die Übung auch zuhause machen.	1