Karin Martina Pichler

Relevant Information and Information Visualizations for Lecturers in Web-Based Audience Response Systems

Master's Thesis

Graz University of Technology

Institute of Information Systems and Computer Media
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Frank Kappe

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

Graz, May 2013

Statutory Declaration

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

Graz, _____ _____

Date Signature

Eidesstattliche Erklärung[1]

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

Graz, am _____ _____

Datum Unterschrift

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

Abstract

Audience Response Systems (ARSs) are a way of interacting with students in large lecture halls. Since the rise of web-based devices, such as smartphones, tablets and laptops, the technology of ARSs has changed. ARSs do not use hardware clickers anymore. Many of today's ARSs are web-based systems, which can be accessed with web-enabled devices. The systems use the web-based devices of students for voting. Therefore, more students and lecturers can use this technology during lectures without buying expensive hardware.

This work covers the information retrieval and information visualizations, which are relevant to the lecturers interface, of a web-based ARS called *RealFeedback*. Existing web-based ARSs are analysed and evaluated to gather a deeper understanding of essential functions of an ARS. Lecturers who already used RealFeedback during their lectures are interviewed. Through the interviews, a deeper understanding of the usage of RealFeedback is gathered. The results are filtered and visualization possibilities are worked out.

The results of the research show, that comparing results of previous lectures is an essential function of an ARS. RealFeedback does not provide this function. Therefore, sessions are introduced and implemented in RealFeedback. An appropriate visualization for comparing sessions is chosen and implemented in the lecturers' interface of RealFeedback.

During research, different topics for future work came up, which are covered in the last section of this thesis. These topics and features will have an impact on RealFeedback.

Kurzfassung

Audience Response Systeme (ARSs) bieten die Möglichkeit mit allen Studierenden zu interagieren, unabhängig von der Größe des Auditoriums. Durch das Aufkommen von web-basierten Geräten wie Smartphones, Tablets und Laptops hat sich auch die Technologie von ARSs geändert. Die web-basierten Geräte der Studenten können nun dazu verwendet werden um mit dem System zu interagieren.

Diese Arbeit beschäftigt sich mit der Informationsgewinnung und Informationsvisualisierung von einem web-basierten ARS. Bei diesem ARS handelt es sich um *RealFeedback*. Im Speziellen wird das Administrationsinterface behandelt. Dazu werden bereits existierende ARSs evaluiert und analysiert um die wichtigsten Funktionen herauszufiltern. Danach werden Lehrende die RealFeedback bereits in ihrer Vorlesung eingesetzt haben befragt. Die Interviews geben ein besseres Verständnis, wie RealFeedback eingesetzt wird und welche Funktionen fehlen. Die Resultate des Vergleichs sowie der Interviews werden gefiltert und Visualisierungsmöglichkeiten ausgearbeitet.

Die Forschungsergebnisse zeigen, dass es für Lehrende wichtig ist, die Ergebnisse von vorhergehenden Lehrveranstaltungen zu vergleichen. Diese Funktion gibt es in RealFeedback nicht. Daher werden “sessions” in RealFeedback eingeführt und implementiert. Eine “session” kann als eine vorhergehende Lehreinheit gesehen werden, in welcher die Resultate dieser Einheit gespeichert sind. Eine Visualisierungsmöglichkeit wurde ausgewählt und in das Administrationsinterface des Lehrenden integriert.

Acknowledgements

First, I want to thank my boyfriend Christian who supported me all the time. Without you, I would not have got that far in such a short amount of time.

In addition, I want to thank my supervisor Martin Ebner who supported me during the last stage of my studies.

I would also like to thank my friends. Sorry for being kind of unsocial during writing my thesis. It will change, I promise!

Last but not least, I would like to thank my family who helped me wherever they could. Thank you for supporting me on the entire way of my journey!

Contents

List of Figures

List of Tables

1. Introduction

"Audience Response Systems (ARSs) or clickers, as they are commonly called, offer a management tool for engaging students in the large classroom." [Caldwell, 2007]. Very often ARSs are called Student Response System (SRS), Personal Response System (PRS) or Classroom Communication System (CCS). This thesis uses the term ARS when talking about this kind of systems. The aim of ARSs is to interact with every person in the audience, even when the audience is large. The presenter or lecturer asks a question with the ARS, and the whole audience can respond immediately to this question via clickers or presently via their internet-enabled devices. Afterwards the lecturer and the audience can immediately see the results of the question.

The results are presented to the lecturer. It can be seen how many people voted for a single answer, most commonly this is represented in the form of a bar chart. However, depending on the number of people in the audience a huge amount of data is collected during a presentation or a lecture. This data might be useful in many different ways to the lecturer or the presenter. A finding of Beatty and Gerace, 2009 is that the lecturers who use an ARS must feel comfortable with the system, and they must be self-reflective. The data, which is presented to the lecturers helps them to reflect on their lectures.

Therefore, it is important to collect relevant data and to provide this information to the lecturer. This thesis deals with the information collection and the presentation of this information in the lecturers interface of a web-based ARS. To find the features and information, which are relevant to the lecturer it is important to analyse the usage of ARSs and to review and compare existing ARSs. This work covers both, the research of the usage as well as the comparison of different existing ARSs.

To get a broad view on the possibilities of ARSs, different systems are analysed, independent of their target group (lectures, conferences, presentations, etc.). The question arises how the information, which is retrieved from the ARS, can be presented or visualized for the lecturer. Therefore, another focus of this work is the visualization of the data and the integration into an existing ARS. The system, which is analysed and improved during this work is *RealFeedback*, a web-based ARS developed in 2012 at Graz University of Technology.

1.1. Structure

This thesis consists of three main parts. The first part describes ARSs in general and the usage of ARSs. Afterwards, the research on the existing system is described and the last part is regarding the implementation of the changes in RealFeedback.

Chapter 2 covers the emerging of the different ARSs beginning from the 1960s until now. The next chapter, chapter 3 refers to the usage of ARSs. This chapter covers the findings and suggestions of the literature.

In chapter 4 RealFeedback is described in detail. The concepts behind the system as well as the technology of RealFeedback are analysed. Chapter 5 is a state-of-the-art analysis of existing web-based ARSs. Twelve web-based systems are analysed, evaluated and compared against each other.

Interviews with lecturers are held who use RealFeedback in their lectures. In chapter 6 the interviews and the findings are explained. This chapter together with the findings of chapter 5 builds the basis for the next chapter.

In chapter 7 the findings are summarized. For these findings, visualizations are presented and analysed. One concept and a visualization for that concept is chosen. This concept and the visualization are implemented into RealFeedback.

Chapter 8 describes the implementation of the concept and the visualization.

The work concludes with chapter 9 where the findings are described and the problems, which emerged, are discussed. Topics for future work are described in chapter 10.

2. History of Audience Response Systems

Audience Response Systems (ARSs) are mentioned in literature which can be dated back to the 1960s [Froehlich, 1963; Boardman, 1968].The first systems were used in military scenarios. Later universities used ARSs for interacting with students in lecture rooms [Judson and Sawada, 2002]. At this time, an ARS was extremely expensive and complex. To use the system appropriately a whole lecture room must have been wired. Each seat or a group of seats was equipped with a device for voting. This device was connected via cable to the main computer. The main computer was responsible for collecting and calculating the results. The instructor presented the results to the audience.

The early ARSs were based on voltmeters. The students pressed a button at their seats and the instructor could see the results on a voltmeter. At the lecturers side a voltmeter was assigned to each answer possibility. Depending on the volts, which showed up at each voltmeter the lecturer could identify which answer got the most votes [Abrahamson, 2006]. IBM also developed an ARS in 1988 [Horowitz, 1988] and did research on how such a system improves the learning process of students. The results show, that the system had a positive effect on the performance of students.

One successful ARS was Classtalk 2.0. The system was created by *Better Education Inc.* [Dufresne et al., 1996]. Each student had an input device. The input devices were graphing calculators produced by Texas Instruments. Classtalk 2.0 was fully compatible with the TI-83, TI-83+, TI-85, and TI-86 graphing calculators of Texas Instruments [Better Education Inc, 1996]. The instructor asked multiple choice questions and the students voted for the correct answer with their graphical calculators. After the voting had been

finished, the results were shown as histogram on the screen of the lecturer. The lecturer presented the results to the students via the projector. The system could be used for grading. Depending on the number of correct responses of a student the grade improved. The lecturer could see the results for each student in the system.

One serious drawback of Classtalk 2.0 was the costs. It was extremely expensive to install Classtalk 2.0. In order to use the system, the entire lecture room must have been wired and the system was not portable. Every lecture room had to be equipped with its own Classtalk 2.0 system [Beatty, 2005; Cue, 1998].

Besides the usage of graphical calculators as input devices, clickers were developed. A clicker is a small hardware device which is used by the audience for voting. The audience votes for an answer by pressing the related button on the clicker [Caldwell, 2007]. At first clickers were connected via cable to the main computer, where the software is installed. Later wireless systems evolved, because it was extremely expensive to equip each classroom with a wired network [Roschelle, 2003]. The wireless devices could be carried from one room to another and could therefore be used in more than just one lecture room. Roschelle introduced the term Wireless Internet Learning Device (WILD) in 2003 [Roschelle, 2003]. WILDs were able to connect a whole classroom via wireless technology. For example, a peer-to-peer or a LAN connection could be established. The devices were also able to create an internet-connection. Everything, which was needed to use this system, were the WILDs, a computer where the WILDs connected to and a software, which was able to process the responses of the WILDs. The first WILDs were mobile clicker systems, which were connected via infrared.

The next generation of ARSs using wireless technology allowed the students to bring their own devices. The first systems were based on Short Message Service (SMS). The students send their answer via their mobile phones as text message to the system [Roschelle, 2003; Scornavacca and Marshall, 2007; Tremblay, 2010]. Afterwards web-based systems evolved. The lecturer and the students use a web-service for asking and answering questions. The students open the web site with their browser of their smartphones, tablets or laptops and vote for an answer. Many of today's systems are based on web-technology [Wong, Mohan, and Lam, 2011].

3. Usage of Audience Response Systems

Audience Response Systems (ARSs) have a long tradition. Different types of usage evolved over time. Lectures should be student-driven and the student should be engaged to active learning [Bransford, Brown, and Cocking, 2000, pp 23; Bruff, 2009]. An ARS fulfils both needs for lectures.

Active learning includes writing, talking with other students, describing and explaining concepts and reflecting [Dufresne et al., 1996; Bonwell and Eison, 1991]. Interactive lectures are a way to achieve active learning in lecture halls [Dufresne et al., 1996] and ARSs provide interaction between the lecturer and the students [Gauci et al., 2009; Beatty, Leonard, et al., 2006; Cue, 1998; Lowery, 2006; Beatty and Gerace, 2009; Beatty, 2005; Liu et al., 2003].

The counter part of active learning on the students side is agile teaching. "Agile teaching refers to the practice of teaching with a very tight feedback loop, almost continually probing students to ascertain and monitor their learning progress and difficulties." [Beatty, Leonard, et al., 2006]

Due to these requirements, different usage types evolved over the years. This chapter covers the most important usage types of ARSs. The following approaches are described in this chapter:

- Question Cycle [Dufresne et al., 1996]
- Question Driven Instruction (QID) [Beatty, Leonard, et al., 2006]
- Peer Instruction [Mazur, 1997]
- TEFA - Technology Enhanced Formative Assessment [Beatty and Gerace, 2009]

3.1. Question Cycle

In 1996, Robert J. Dufresne et al. at the University of Massachusetts defined the term "question cycle". At this time an ARS called Classtalk was in use for interacting with students during lectures. The learner should play an active role in the whole process of the lecture and the construction of knowledge [Dufresne et al., 1996].

Therefore, they invented the question cycle to activate students during lectures. They tried to build a student-centered classroom instead of a teacher-centered class. The students are involved during lecture by answering questions alone or in groups and discussing the answers with their colleagues. Afterwards, a class-wide discussion is started where the entire class talks about the results. As soon as the students understand the correct answer, the next question comes up. The instructor spends just one third of the class period to present information to the class. The rest of the time is used for questioning the students. See figure 3.1 for a detailed description of the question cycle.

The instructor prepares the questions before the lecture starts. Depending on the answers of the students, the instructor changes the questions during the lecture in order to fit the students needs. Changing the lecture style and the questions according to the students needs is part of agile teaching.

3.2. Question Driven Instruction

The term Question Driven Instruction (QDI) describes a method which uses the question cycle for teaching. Encouraging students to active learning and the teacher to agile teaching is a main focus of this method [Beatty, Leonard, et al., 2006] compared to the question cycle which was described before. Figure 3.2 on page 3.2 shows how the ARS is embedded in the process of teaching.

QDI moves away from the "transmit and test" paradigm and moves to a repeating process of asking questions, allowing students to get to an

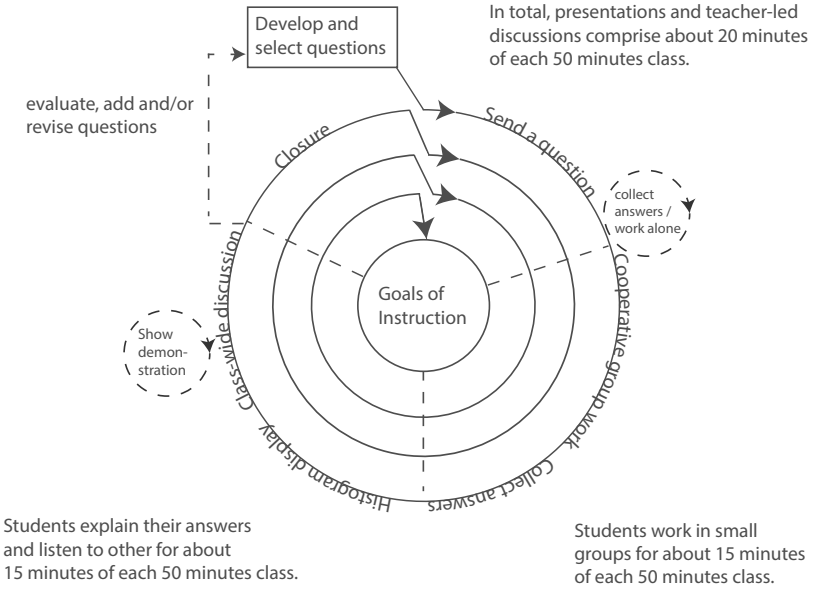

Figure 3.1.: This figure describes the question cycle which was fist mentioned by Dufresne et al., 1996. Adapted from [Dufresne et al., 1996]. The cycle starts with asking a question. Afterwards, the students discuss and answer the questions. The results are shown as soon as the question stops and a class-wide discussion starts. When the question-cycle has finished the lecturer goes on with the course material or with the next question.

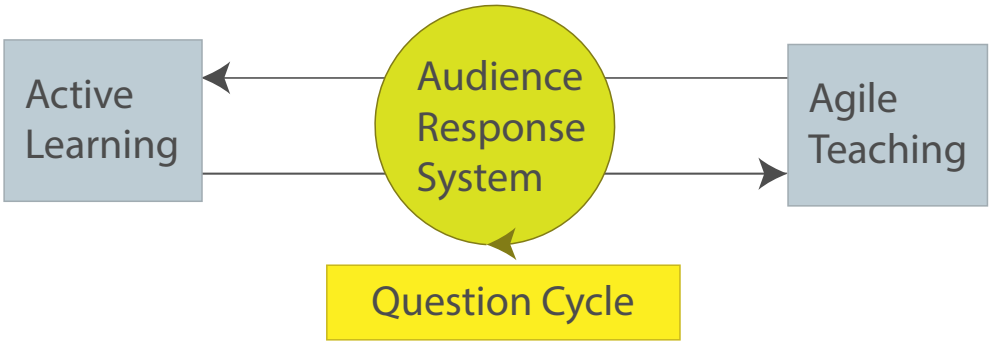

Figure 3.2.: Question driven instruction includes the question cycle. Agile teaching on the lecturer's side and active learning on the student's side are influenced by the question cycle. Adapted from [Beatty, Leonard, et al., 2006].

answer, and discussing the answer with other students and/or the teacher. This approach is the same as with the question cycle.

The approach of QDI is that students work on the topics for the lecture at home and go through the material in the lecture. In the lecture, the students work with the concepts they learned at home. They engage with the topic and get a deeper understanding. The lecturer uses an ARS to ask questions during the lecture. When the lecturer asks the question the question cycle starts and the students begin to talk and discuss the topic. Then they commit to an answer and the class-wide discussion follows. The lecturer always tries to react and adjust the lecturing method according to the results of the question.

With QDI, it is essential, that the teacher allows the students to discuss. It may seem that the teacher is losing time, but according to Beatty, Leonard, et al., 2006 it is very essential to the learning of the students to work with the material, to discuss different aspects and argue with others. In this way, the students see different perspectives, get deeper into the topic, and get a deeper understanding of concepts.

3.3. Peer Instruction

Eric Mazur - a physics professor at Harvard University - defined peer instruction in 1997 [Mazur, 1997]. He wanted the students to understand the material and the topics instead of learning them by hard. Mazur thought that the only way to get into a topic and to understand it is to discuss it with the students. It requires teachers to get into an interactive teaching style. Questions shall be asked by the lecturer during lectures and students have to think about the basic concept. The focus is on understanding and not only problem-solving and memorizing the content without understanding the concept [Mazur, 1997].

Students work on the course material on their own before the class starts. The lecturer starts the class with a question regarding the topics and the material the students had to learn. The students answer, and the results are presented. Afterwards, students form small groups and discuss the answers in these groups. Then the whole group gives the answer again to the same question. After each group has given an answer, the results are presented to the class again.

A lecture consists of short presentations of key facts followed by a concept test. The aim of the concept test is to get students think about a concept more deeply. Concept tests take up one third of the lecture time. The concept test delivers quick feedback to the students and the lecturer whether the students understood the concept right or if there are misconceptions. This helps the students to find misconceptions earlier and prevents confusion as well as misunderstandings when moving on to topics that are more complex.

The concept test consists of the following steps [Mazur, 1997, p10]:

- Question posed (1 minute)
- Students given time to think (1 minute)
- Students record individual answer (optional)
- Students convince their neighbours (peer instruction) 1-2 minutes
- Students record revised answers (optional)
- Feedback to teacher: Tally of answers
- Explanation of the right answer 2+ minutes

Since concept tests take up a lot of time during the lecture Mazur sees two ways to cope with this problem [Mazur, 1997, p14]:

- "discuss in lecture only part of the material to be covered over the span of the semester"
- "reduce the number of topics covered during the semester"

Mazur prefers the first choice: students have to do pre-class reading and during the lecture, he does not cover all topics which are written down in the lecture notes. Mazur only covers topics, which were not understood by the students during their self-study and which must be covered during the lecture.

Back in 1993, there were not many ARSs on the market and in most cases they were expensive. Mazur does not rely on a technical solution in order to get feedback. He describes the following methods:

- Show of hands
- Scanning forms
- Hand-held computers (for example Classtalk)

In his opinion, a technical solution makes a class more complicated. An advantage of a technical solution is that the results are shown immediately after the answers are given and that the results are more precise than, for example, with the "show of hands"-method [Mazur, 1997, pp 16].

3.4. TEFA - Technology-Enhanced Formative Assessment

Beatty and Gerace, 2009 first mentioned Technology-Enhanced formative assessment (TEFA). As all approaches mentioned in this chapter TEFA is based on active learning and agile teaching too. It consists of four core principles:

- Question Driven Instruction (QDI)
- Dialogical Discourse
- Formative Assessment

- Meta-level Communication

Question Driven Instruction: QDI in the context of TEFA is again based on the concept of the question cycle which is shown in figure 3.1. When using the question cycle the lecturer can achieve the following objectives according to Beatty and Gerace, 2009, p. 13:

- learn about student's knowledge, thinking and perceptions
- help students do become more aware of their own knowledge, thinking and perceptions
- help students become cognizant of other students' knowledge, thinking and perceptions
- set up subsequent instruction
- catalyze small-group discussion and peer learning
- provoke, open, motivate, ground and contextualize whole-class discussions of a topic
- precipitate student insights and realizations

The question cycle builds the basis for TEFA.

Dialogical Discourse: The dialogical discourse focuses on the discussion phase of the question cycle. Small and class-wide discussions are an essential part of the dialogical discourse.

The ideas behind the dialogical discourse in TEFA are [Beatty and Gerace, 2009, p. 15]:

- to clarify thoughts through the process of articulation and externalization
- to expose students to different points of view and lines of thinking
- to promote analysis and resolution of disagreements
- to supply stimuli, context and tools for individual sense making
- to provide practice speaking the social language of science.

Formative Assessment: Formative assessment is defined by [Bransford, Brown, and Cocking, 2000] as follows: "Ongoing assessments designed to make student's thinking visible to both teachers and students. [...] They permit the teacher to grasp the students' preconceptions, understand where the students are in the developmental corridor from informal to formal thinking and design instruction accordingly. [...] formative assessments help both teachers and students monitor progress."

The questions, which are asked during lecture, give feedback about the knowledge and understanding of the students as well as knowledge about how their colleagues understand the material.

Lecturers learn about the prior-knowledge, the understanding, the perceptions and misconceptions of their students. This knowledge helps the lecturer with agile teaching. This means to change the lecturing style according to the needs of the students.

Meta-level Communication The principle of meta-level communication is defined by [Beatty and Gerace, 2009] as follows: "Help students cooperate in the learning process and develop metacognitive skills with meta-level communication".

Meta-level communication in TEFA aims to "help the students develop expertise in the science subject being taught, and help to prepare them for future learning" [Beatty and Gerace, 2009].

4. RealFeedback

RealFeedback is an ARS developed in 2012 at Graz University of Technology. The aim of RealFeedback is to provide a basic system for asking questions to a large audience with a user-friendly and easy-to-use interface. Every year more and more students start studying in Austria, and the interaction between the lecturer and the students gets more difficult. Figure 4.1 on page 16 shows the growth of students from 1971 to 2012 in Austria. The aim of RealFeedback is to allow interaction between lecturers and large groups of students in lecture halls.

RealFeedback is a web-based system. No additional software must be installed to use RealFeedback. A web-browser and an internet connection are needed to start asking questions. The students or the audience can vote via web-enabled devices such as laptops, tablets or smartphones. For smartphones, a mobile website is accessible to guarantee a good usability even on small-screen devices. The results are shown to the lecturer as well as to the students in the form of bar charts and numbers. The lecturer also has the option to download the results of his entire project as comma-separated values (CSV) report.

RealFeedback is reduced to the basic functionality such a system must provide. It is specialized on asking questions and letting the audience vote for answers. No further functions are added. Everyone can use the system without registering to RealFeedback. One can easily start asking questions. As long as the user is not registered, the questions will be lost as soon as the browser-tab or the browser-window is closed. If the user wishes to save the project for later use, she has to register. All questions can be reused if the user has registered and the project is stored. In order to get new results when reusing questions, the old results must be cleared. As soon as the results are cleared the user has no possibility of viewing them again. The results are deleted irrecoverably.

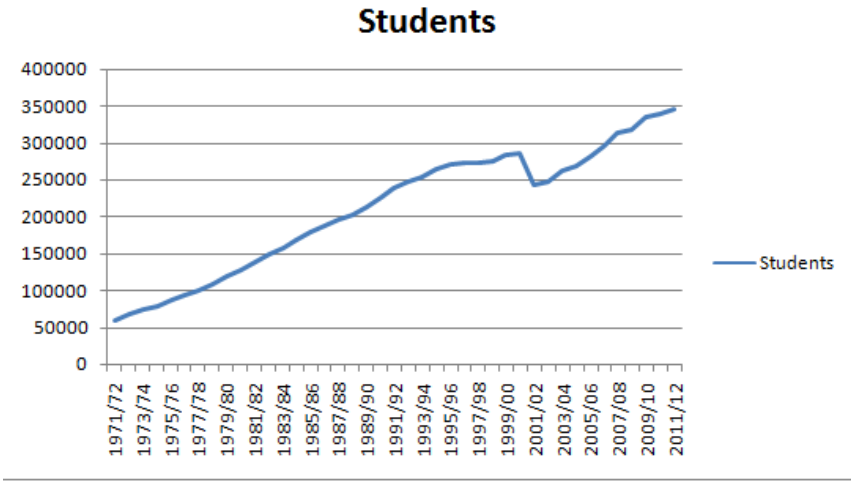

Figure 4.1.: This diagram shows the growth of students in Austria from 1971 to 2012. The data for this diagram is gathered from *Statistik Austria* accessed on 10/04/2013.

Answering to questions is anonymous and the system provides no possibility for the students or the audience to register. Because of this reason RealFeedback cannot be used for summative assessment or as a grading system where students get graded based on the correct answers they give.

4.1. Description of the Concepts

In this work, RealFeedback is defined as a group of concepts, which all work together. The concepts that are used in order to describe RealFeedback, are explained in this section.

User: The user is the person who uses RealFeedback for asking questions to the audience. A user can be a registered user as well as a user, who uses RealFeedback only once without registering.

Students and Audience: The students and the audience are the people who use the system for voting. There is no need for students or the audience to register at RealFeedback.

Project: As soon as the lecturer starts generating questions in RealFeedback a project is automatically created. A project consists of one or more questions and a project can be stored if the user is registered at the system. When comparing a project to university it can be seen as a single lecture. All answers the students give during the lecture via RealFeedback are stored within the project.

Question: A question always belongs to a project. One question consists of a question text, and as many answers the user would like to add. When a question is created, it is not visible for the audience. The lecturer has to activate it and as soon as the question is activated, the students or the audience can vote. It can be voted as long as the question is activated. As soon as the lecturer deactivates the question, the audience cannot vote any longer for this question. When the question is deactivated, the results are displayed. The results for one question are stored as long as the question is not cleared. To ask the question again to another audience the results should be emptied. If not, the new votes will be added to the old results.

Answer: An answer belongs to one question and consists of an answer text. The audience can vote for one answer and the number of votes for each answer is stored. As soon as the question where the answer belongs to is deactivated, the results are calculated. The results are shown as bar chart beside each answer. Besides the bar chart, the total number and the percentage of votes for this answer are displayed.

Session: The term session refers to the point of time when the lecturer uses RealFeedback for asking the students. For example, a lecture could be a session.

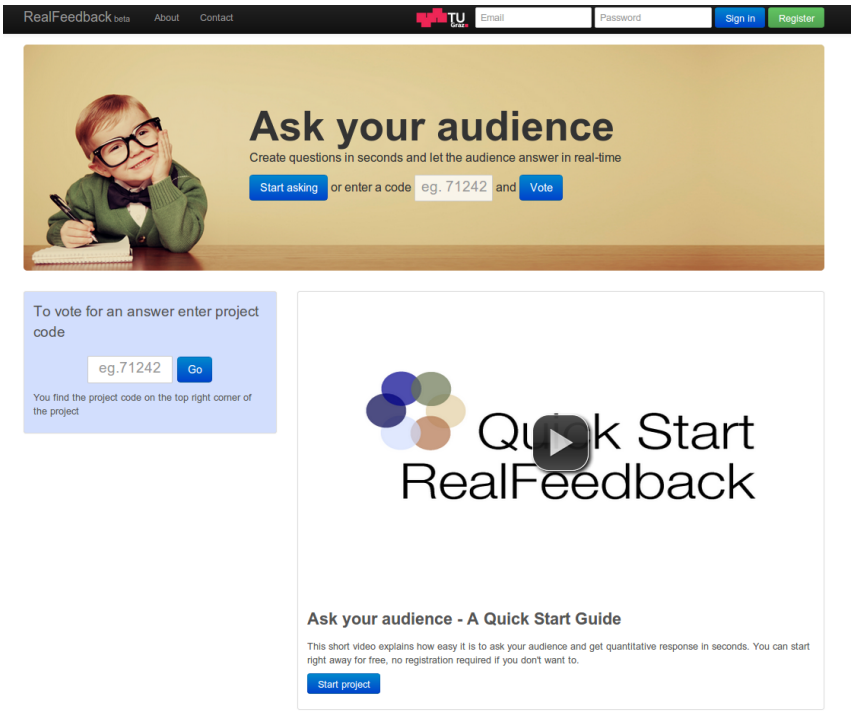

Figure 4.2.: This figure shows the start page of RealFeedback. On the start page the lecturer can start asking questions and the students can enter the session-keys to start voting. The session key can be seen at the top right corner of the figure.

4.2. The Process of Asking Questions

This section covers the different steps a lecturer has to take for asking questions. It is described how questions can be generated, how they can be activated and deactivated, how the audience can vote for an answer and how the statistics and the results look like.

4.2.1. Generating Questions

Questions can be generated easily without registration. If the lecturer wants to save a project, she has to register at RealFeedback. As soon as the lecturer

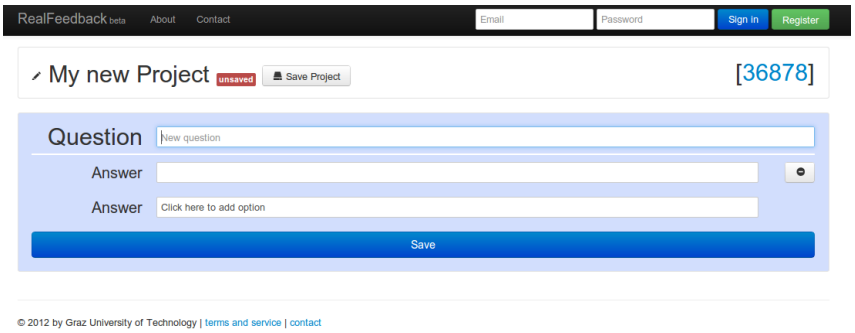

Figure 4.3.: This figure shows the input mask for generating a question. The lecturer can add new answers by clicking in the next empty answer field or by pressing the "tab" key. When the lecturer clicks on the "save" button the question is saved.

is registered, projects can be reused for later lectures.

Creating questions can be easily done at the main page of RealFeedback. How the main page looks like can be seen in figure 4.2 on page 18. A button who says "Start asking" leads the lecturer to the interface where she can generate questions. Figure 4.3 on page 19 shows how the interface looks like. The lecturer can add as many answers to one question as she likes. Answers can be added by clicking into the next empty answer field or by pressing the "tab" key. The question will be saved when the lecturer clicks on the "Save" button. The questions as well as the answers can be edited at any point of time. As soon as the answer is saved, the lecturer can ask the question to the audience.

4.2.2. Asking Questions

As soon as the question is generated, the lecturer can start asking her students. Each project has a session-number, which shows up in the top right corner of the administration interface of a project. The session-number can be seen in figure 4.3 on page 19. This number must be provided to the audience. With the session-number, the students join the session. The session-number must be typed into the field at the main page of RealFeedback

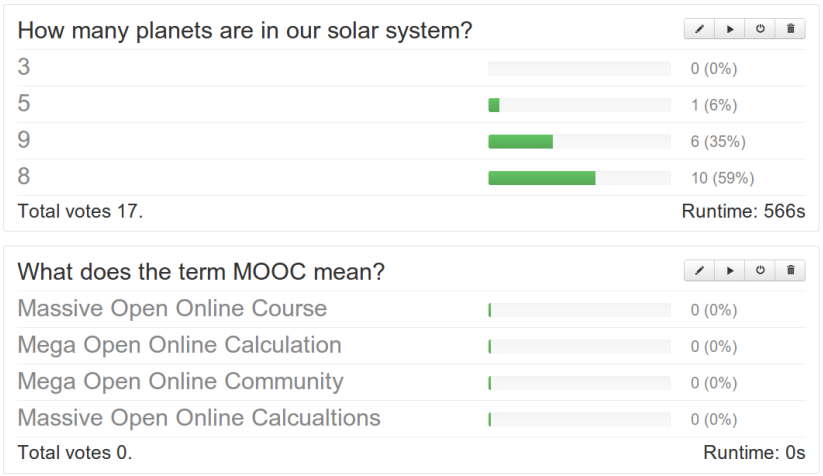

Figure 4.4.: This figure shows how the generated questions look like in the lecturers interface. The first question has already some votes whereas the second questions is not started at all. The lecturer can start the question by pressing on the "play" button in the top right corner of each question.

(figure 4.2 on page 18). After pressing the button "Go" or "Vote" the students are registered to the session.

The administration interface for the lecturer can be seen in figure 4.4 on page 20. All questions which were ever created for this project are shown to the lecturer. When the lecturer activates a question, the audience can see the question at the interface on their personal devices. Figure 4.5 on page 21 shows the interface for students. They can vote for the correct answer at the moment the question shows up. No results are shown during the process of voting. This has the effect that the students are not influenced by the votes of their colleagues and that they are not afraid of voting for the wrong answer [Bruff, 2009, pp. 198]. The audience can vote for the question as long as the lecturer keeps the question active. Before the vote can be submitted, the students have to solve a captcha [*Captcha Definition*]. This prevents RealFeedback from attacks and automated votes of robots. To stop the voting process the lecturer deactivates the question. After the question is stopped, the results are displayed at the lecturers interface and on the devices of the students. A bar chart represents the number of votes

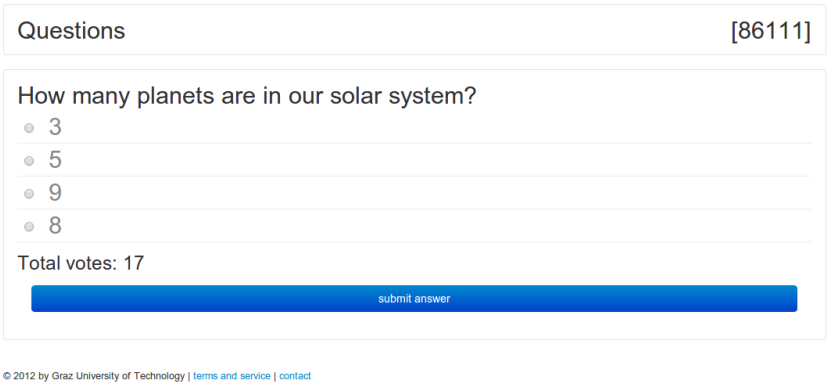

Figure 4.5.: This figure shows the interface of RealFeedback which can be used by the students. By clicking on an answer and pressing the "submit"-button students can vote.

for each answer of a question. The total number of votes and the percentage of people who voted for a specific answer are shown as well.

The lecturer can review the results of a session at any time. To view the results the lecturer must open the project, in the administration interface. Users, who are not registered, have no possibility of reviewing the results as soon as they have closed the tab or the browser-window.

4.2.3. Reusing Questions

The lecturer can reuse projects in later lectures. The lecturer can change questions and answers of a project at any point of time. To reuse questions it is reasonable to clear the results of the previous session. If the results are not cleared the new votes are summed up to the results of the previous session. As soon as the results of the previous session are cleared they cannot be reviewed anymore because they were deleted irrecoverably from the system. The only possibility for the lecturer to keep the results is to download a report before clearing the results. This functionality is only available for registered users.

4.3. Database Structure and Data Retrieval

This section describes the database architecture of RealFeedback and the functionality, which is provided by the server to gather this data. First, the existing database structure of RealFeedback is analysed and described. The next section covers the data retrieval.

4.3.1. Database Structure

This section describes the database architecture of RealFeedback. All data, which is needed for RealFeedback to work is stored in a NoSQL database [*NoSQL Database*]. The server-side of RealFeedback is implemented in python, and as database framework MongoDB is used.

The definition of MongoDB is as follows: "MongoDB (from "humongous") is an open source document database, and the leading NoSQL database. Written in C++" [*MongoDB*].

Figure 4.6 on page 23 shows the collections of the database as well as the relations between the collections. It can be seen that the concepts, which were defined before (user, project, question, answer) are mainly the collections, which are stored in the database.

In MongoDB a collection is defined as follows: "Collections are groupings of BSON documents. Collections do not enforce a schema, but they are otherwise mostly analogous to RDBMS tables. The documents within a collection may not need the exact same set of fields, but typically all documents in a collection have a similar or related purpose for an application." [*MongoDB*]

The database structure of RealFeedback consists of the following collections:

- users
- projects
- questions
- answers
- project_user

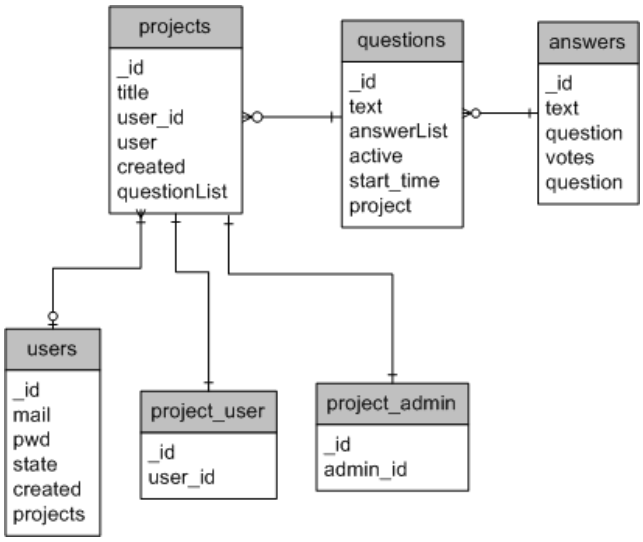

Figure 4.6.: The database structure of RealFeedback is realized as NoSQL database [*NoSQL Database*]. The structure of the database is shown in this figure. MongoDB is used as framework for the NoSQL database [*MongoDB*].

- model_changed
- captcha

How these collections are related to each other is shown in figure 4.6. The following paragraphs describe each collection and the data structure of the collection.

users: The user collection holds all data regarding the administrator or the lecturer who registers at RealFeedback. For each user, the data described below is stored.

_id unique ID for each user
created time when the user registered at RealFeedback
mail E-mail address of the user
pwd password hash
projects list of all projects which belong to a user
state indicates whether the user is active or not

Projects are linked to a user via the "projects"-list of a user.

projects: In this collection, the data for each project ever created is stored independent of whether the user registered at RealFeedback or not. If the project belongs to an user the ID of the project is stored in the "projects"-list of the user-collection. The link to a user is missing if the project is not created by a registered user. Each project holds the following data:

_id unique ID for each project
title title of the project
created time when the project was created
user the ID of the user who created the project
questionList list of the question IDs belonging to this project
user_id session-key for the project

Questions are linked to a project via the "questionList" of a project.

questions: The data for each question is stored in the this collection. A question is always linked to a project. The following information is stored for a question.

_id unique ID for each question
text the question text
project the unique ID of the project where the question belongs to
answerList a list of unique IDs of answers which belong to this question
start_time the time when the question was activated
active this field indicates whether the question is active or not

Answers are linked to a question via the "answerList" of a question.

answers: In this collection, the data for an answer is stored. An answer can never exist without a question. The following data is stored for each answer.

_id unique ID for each answer
text the text of the answer
votes the number of votes for the answer

question the unique ID of the question the answer belongs to

project_user: The relation between a session-number and a project is stored in this collection. Each project gets assigned a session-number ("user_id").

_id unique project ID
user_id the session-key.

model_changed: In this collection, the points of time are stored when a project has changed.

captcha This collection stores data which is relevant to finish a captcha challenge.

4.3.2. Data Retrieval

The data is retrieved via RESTful Hypertext Transfer Protocol (HTTP) calls [Fielding and Taylor, 2002]. This section covers the RESTful HTTP calls, which are already defined for RealFeedback. These RESTful HTTP calls deliver the data from the server to the web-interface. The prefix of the RESTful HTTP request, which stays the same for every call is "realfeedback.tugraz.at/v1/". The structure of the following part of the URL changes according to the information which must be retrieved.

Get Project-ID: This call returns the ID of a project where the session-number (user_id) belongs. The session-number must be provided to the RESTful HTTP call.

`realfeedback.tugraz.at/v1/project/userid/{user_id}`

The parameter, which must be provided to this RESTful HTTP call, is the `user_id`. This is the session-number, which must be entered by the audience to register to a session.

Method: `GET`

Returns the following data as JSON:

_id The `project_id`, which belongs to the provided session number.
user_id The session-number for the project.

Get Questions of a Project This RESTful HTTP call returns all questions which belong to the specified project.

`realfeedback.tugraz.at/v1/project/{project_id}/questions`

The parameter which must be provided to this RESTful HTTP call is the `project_id`.

Method: `GET`

The return-value are the questions for this project as JSON-array. Each question is a JSON object in the array:

_id The `question_id` of a single question, which belongs to the project.
answerList An array of `answer_ids`, which belong to this question.
text The text of the question.
start_time The start-time of the question.
active This field indicates whether the question was activated or not.

Get Answers for one Question The answers, which belong to a single question, are retrieved with this RESTful HTTP call.

`realfeedback.tugraz.at/v1/question/{question_id}/answers`

This call needs the `question_id` of the question for which the answers must be retrieved.

Method: `GET`

The return-value of this call are the answers, which belong to the question as JSON-array. Each answer is a JSON object.

_id The `answer_id` of a single answer which belongs to the question.
text The text of the answer.

votes The number of votes for this answer.
question The `question_id` of the question where the answer belongs to.

There are some more RESTful HTTP requests concerning data retrieval, which are provided by RealFeedback, but they are not relevant for this work.

5. Comparison of Web-Based Audience Response Systems

Many web-based Audience Response Systems (ARSs) emerged due to the change of technology. This section describes and compares twelve web-based ARSs. To gain a deeper understanding of existing web-based ARSs different systems where tested, evaluated and compared to each other. The different systems, which were tested, can be found in table 5.1 on page 30.

In section 5.1 on page 29 the systems are described in a short way and what complications appeared while testing and evaluating. In the next section, 5.2 on page 43 the systems are compared against each other based on different categories, which can be measured. Afterwards in section 5.3 on page 52 the strengths and weaknesses of RealFeedback are discussed in comparison to the evaluated ARSs.

5.1. Description of the Audience Response Systems

Different ARSs were compared against each other. For comparison, only web-based systems were evaluated. The reason for this decision is that RealFeedback is also based on web technology. The twelve tools, which were evaluated, can be found in table 5.1 on page 30.

In this section, the impressions during testing are described for each tool, with special focus on usability and unique features. The following tasks must have been accomplished in the test scenario:

System	Website
Socrative	www.socrative.com
Top Hat Monocle	www.tophatmonocle.com
SMSPoll	www.smspoll.net
mQlicker	www.mqlicker.com
Clicker School	www.clickerschool.com
Poll Everywhere	www.polleverywhere.com
understoodit	www.understoodit.com
Pinnion	www.pinnion.com
Free Mobile Polls	www.freemobilepolls.com
LectureTools	www.lecturetools.com
PINGO	www.pingo.ubp.de
Mentimeter	www.mentimeter.com

Table 5.1.: The ARSs listed in this table were compared against each other. Each tool was last visited on 12th of April 2013.

- Create a lecturer account for the tool, if necessary.
- Create questions. If different types of questions are available, a question is created for each question type. Each question type should be evaluated and tested.
- Creating a student account for the tool, if necessary.
- Starting one or more questions with the lecturer account.
- Answering questions with the student account.
- Review the results of the questions in the lecturers interface.
- Review reports if there are any available.
- Test further functionality, if available.

All systems, which were tested during this analysis, are described in the following paragraphs. The special features of each system, as well as the usability of the system are explained. A paid version is available for some of the tested ARSs. All tests and comparisons are made with the free version of the systems. Some functions and features are only available in the paid version. This is the reason why not each functionality of every system could be tested. If a paid version is available, it is mentioned in the description of the ARS.

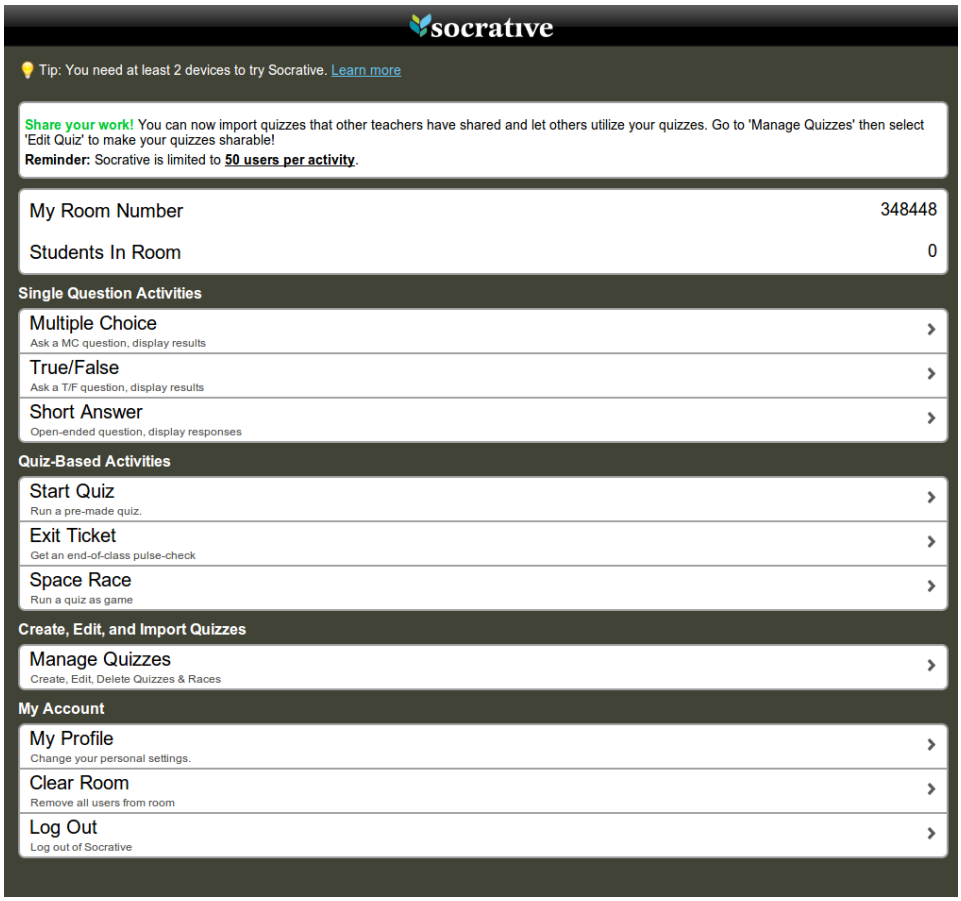

Figure 5.1.: This figure shows the administration interface of Socrative.

Socrative: This ARS is available as web-service and as mobile application, for Android and iOS. The mobile application is available for the students as well as for the lecturer.

Socrative provides an easy way of asking questions to the students instantly during lecture. This feature allows the lecturer to ask questions without writing down the answers into the system. When an instant question is started the students can vote for a specific answer by clicking on the letter (A, B, C, D, E). The lecturer only has to provide the information which answer belongs to which letter. The lecturers interface of Socrative can be seen in figure 5.1 on page 31.

The lecturer can also prepare questionnaires before the lecture starts. The results of the different sessions of a questionnaire are saved can be downloaded as CSV report. If a lecturer uses the same questionnaires every year, she has the option to compare the results of each year by downloading the CSV reports and comparing them manually.

Socrative is not anonymous to use by the students except when answering instant questions. All students have to provide their names before they can start answering prepared questionnaires. The lecturer gets an overview, how students responded. These results are presented to the lecturer as table. The students have no possibility to look at the results on their devices. Socrative also provides gamification [Deterding et al., 2011, Groh, 2012, pp. 39] in the form of a space race where students work together in teams. The teams have to answer as many questions as possible correct in order to win the space race.

Top Hat Monocle: Not all functions of Top Hat Monocle could be tested. The usage for the lecturer is free. However, the students have to pay in order to use Top Hat Monocle. This is a unique concept among the tested ARSs. When registering for the lecturers account a school in the US must be chosen. Therefore, it can be assumed that Top Hat Monocle is a tool designed for the US only. The lecturers interface can be seen in figure 5.2 on page 33. Besides the website, Top Hat Monocle provides a mobile application for Android and iOS devices.

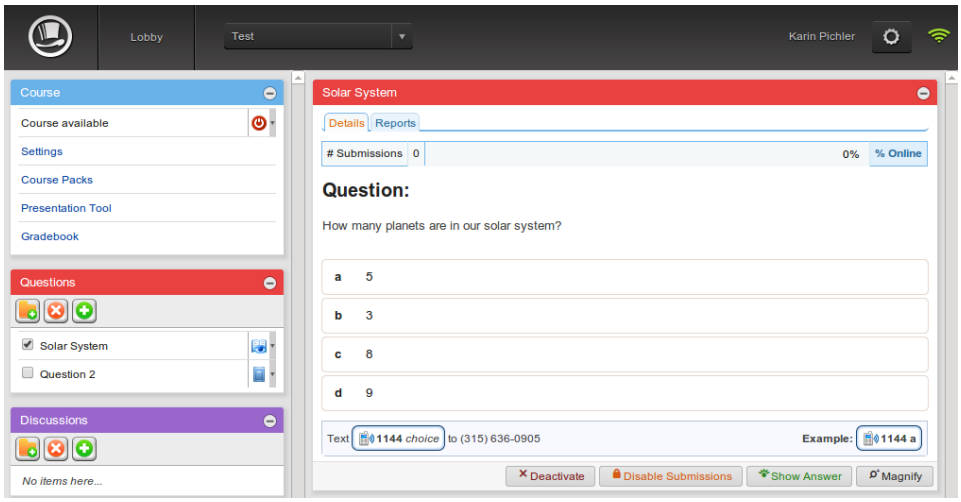

Figure 5.2.: This figure shows the administration interface of Top Hat Monocle.

The administration interface of Top Hat Monocle offers many features and functions. As far as it could be tested, Top Hat Monocle provides the functionality of reviewing the results. It also offers a feature where different sessions of a questionnaire could be compared against each other in the backend. Fulfilling the tasks, which were defined for the testing scenario, was difficult because Top Hat Monocle offers a lot of features. Therefore, it is more complex and harder to learn. The student side of answering questions could not be tested. One notable feature of Top Hat Monocle is the student chat. However, the chat also could not be tested.

SMSPoll: This ARS provides a web-interface or SMS for responding to questions. The web-interface can be seen in figure 5.3 on page 34. When registering for an administration account for SMSPoll one out of four countries must be chosen (Australia, New Zealand, UK & Europe and United States of America). If a participant wants to answer a question, she has to choose the correct country for which the poll was created. Otherwise, the question cannot be answered. Answering to a poll means for the participant that she has to enter a five-digit number instead of letters (for example 12345).

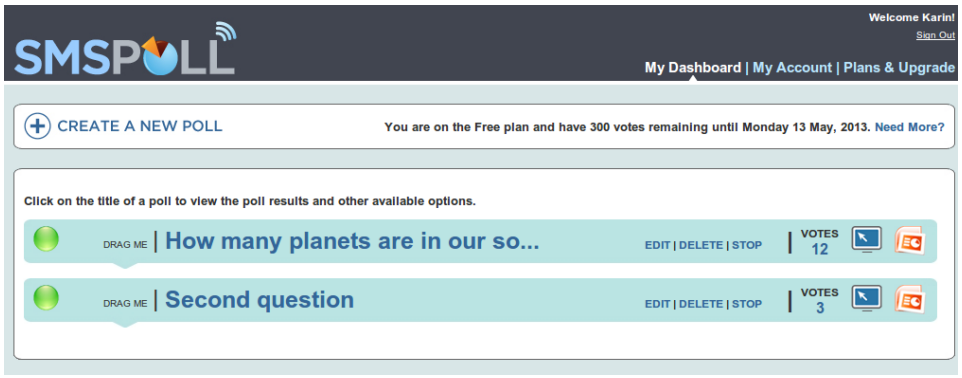

Figure 5.3.: This figure shows the administration interface of SMSPoll.

The participation is completely anonymous. The results of the poll are visualized for the lecturer or the administrator of the poll as bar chart. The participants do not get any feedback about the results. After the questions are answered the instructor can download a CSV report, which contains the results of the questions. Different sessions of same questions cannot be compared against each other within the system. If the instructor wants to use the same question more than once, the results must be cleared. The cleared results are not longer available for review.

mQlicker: This ARS offers many features and the interface for the lecturer looks extremely complicated at a first glance. When first confronted with mQlicker it takes a lot of time to get into the system. Fulfilling the tasks, which were defined for the testing scenario, is difficult when using it the first time. mQlicker provides an extremely enhanced user interface with ribbons like the menus in the Microsoft Office Suite are designed. How the administration interface of mQlicker looks like can be seen in figure 5.4 on page 35.

The lecturer has the possibility to structure the questionnaires by building hierarchies, which help to keep the polls and questionnaires organized. mQlicker offers quite a lot of ways - compared to the other systems - of visualization. Results can be visualized in the form of bar charts, pie charts and tables. Furthermore results can be integrated into a Microsoft Power

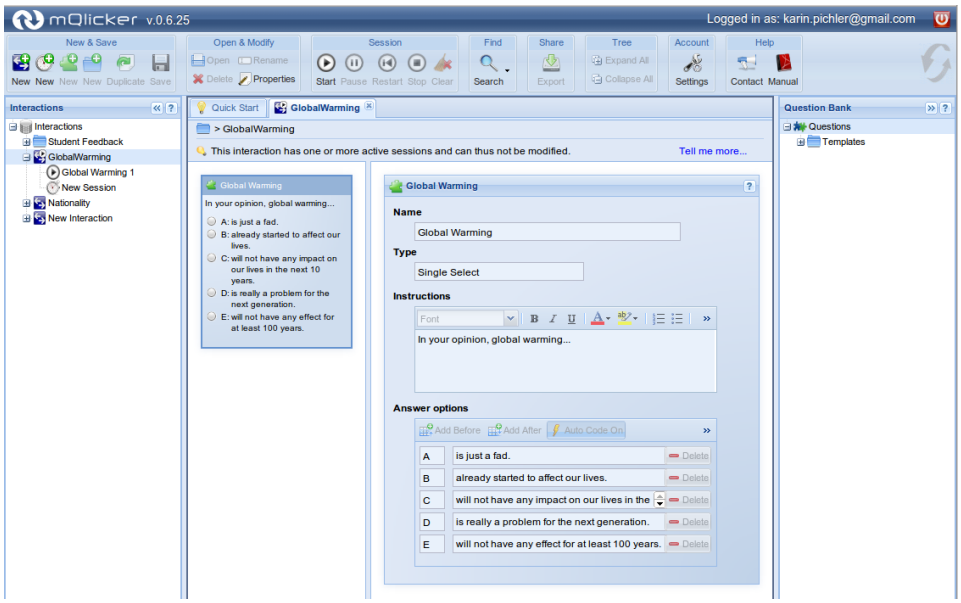

Figure 5.4.: This figure shows the administration interface of mQlicker.

Point presentation. mQlicker provides the possibility of asking the same questionnaire more than once without losing the results of previous sessions. The system does not provide functionality for comparing different sessions of a questionnaire within the system. However, for each session a CSV report can be downloaded and the lecturer has the option to compare the results manually. Answering questions is completely anonymous for the students. In contrast to some other ARSs mQlicker provides a visualization of the results for the students too.

Clicker School: Clicker School offers the possibility of voting via the web-interface and it provides mobile applications for Android and iOS devices. The web-interface for the lecturer can be seen in figure 5.5 on page 36. The mobile applications are built for students and lecturers. Clicker School also provides a way of responding with hardware clicker systems.

Fulfilling the tasks for the test scenario in Clicker School is extremely difficult to achieve the first time. When students respond to the questions,

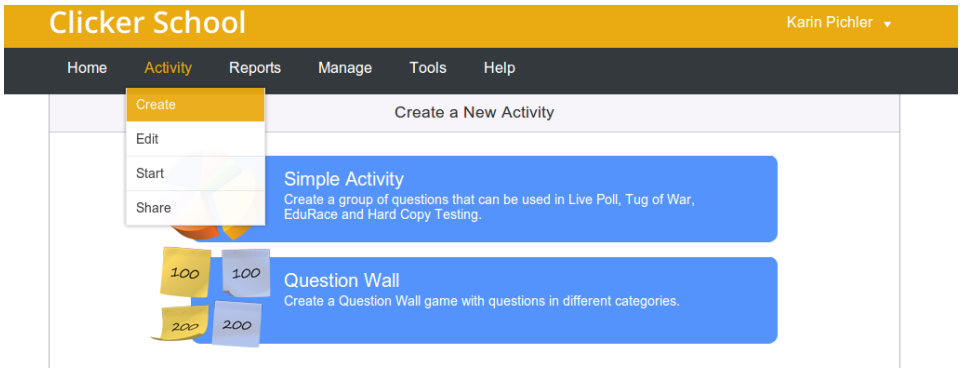

Figure 5.5.: This figure shows the administration interface of Clicker School.

they have to fill in their names before they can start answering. Students have no possibility to vote anonymously. The results are not shown on the student's devices. Only the lecturer gets the results visualized as bar chart or pie chart in her administration-interface. The results cannot be downloaded, but the lecturer has the option to look at the results of each student individually. For this reason, Clicker School is suitable for summative assessments where the correct answers of each student are added to the grade.

Clicker School also integrates some kind of gamification [Deterding et al., 2011, Groh, 2012, pp. 39] aspects. Different games for students are integrated. These games are EduRace, QuestionWall and Tug of war. The students also have the possibility to chat with their colleagues and the lecturer via the mobile application and the web-interface.

Poll Everywhere: One unique feature of Poll Everywhere is that the lecturer has the option to schedule when the questionnaire should start. The questionnaire starts automatically at the scheduled time. The system is easy to understand, and the tasks of the test scenario could be fulfilled easily. Figure 5.6 on page 37 shows the interface for the lecturer. In contrast to other systems, Poll Everywhere offers the functionality to insert math equations into a question. Poll Everywhere also offers integration into Microsoft

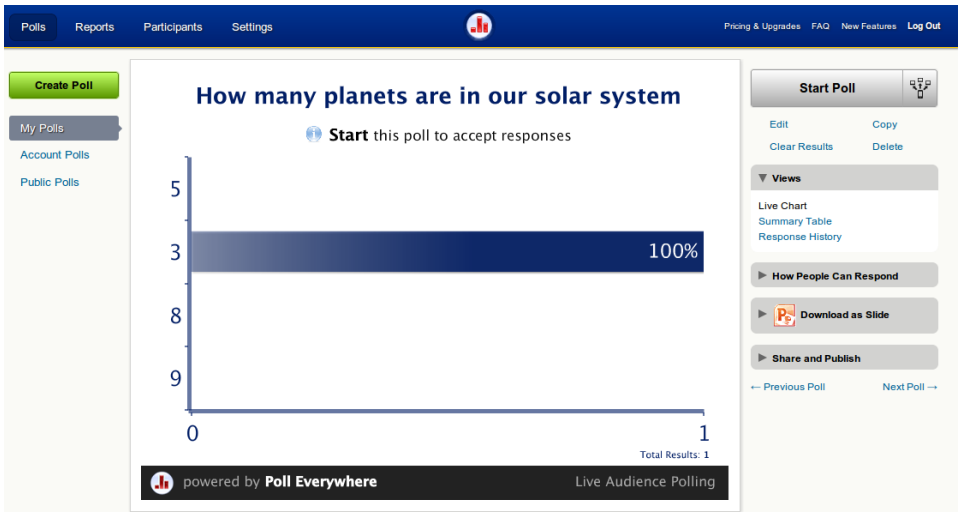

Figure 5.6.: This figure shows the administration interface of Poll Everywhere.

Power Point presentations.

The audience can respond to questions via SMS, via the web-interface or via Twitter. When the audience wants to answer a question, they have to fill in the five-digit number, which is provided for each answer. Therefore responding is not very intuitive. The audience can answer the questions completely anonymous. The results of the questionnaire are only shown to the lecturer. Students do not get the results on their devices. The reporting functions of Poll Everywhere could not be tested because these are only available in the paid version.

understoodit: This ARS offers two main functions to the lecturer. One function is asking questions. The other function is getting feedback from the audience whether they understood everything or not. The lecturer can create polls, which can be reused. The results of the different sessions of these polls can be reviewed at any time. The students can answer questions via the web-interface of understoodit. Answering questions is anonymous, and the results are immediately shown - as soon as a student votes - at the lecturers and at the students interface. This is uncommon compared

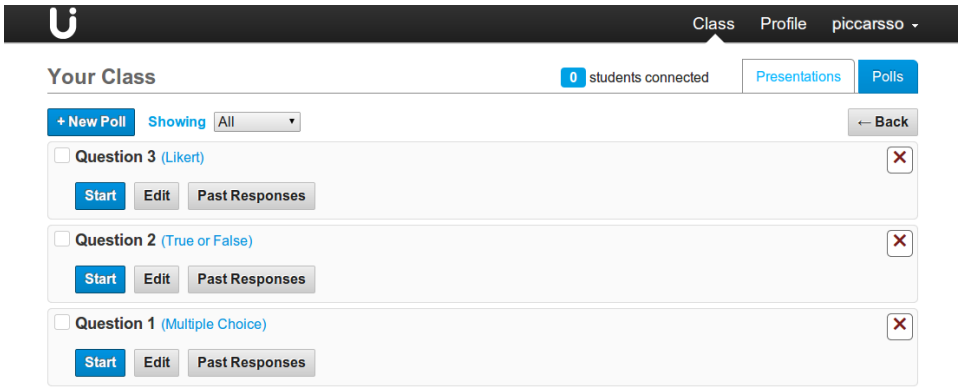

Figure 5.7.: This figure shows the administration interface of understoodit.

to other systems. The other systems show the results after the lecturer has closed the question for voting. Figure 5.7 on page 38 shows the administration interface for the lecturer.

Pinnion: In Pinnion, a questionnaire (called Pinion) can be scheduled. A start date and an end date can be provided, and the questionnaire automatically starts. The participants have the possibility to vote for an answer via SMS and the web-interface. Pinnion also provides native mobile applications for Android and iOS devices. These mobile applications are developed for each customer individually and therefore they are not for free. The participants can answer a questionnaire anonymously. The results are shown to the lecturer as well as to the participants or students who vote for a questionnaire. The results cannot be downloaded and different sessions cannot be compared against each other. Figure 5.8 on page 39 shows the administration interface for the lecturer.

Free Mobile Polls: Free Mobile Polls does not offer many features to the user. As soon as a question is generated, it can be answered all the time. There is no functionality to activate or deactivate questions. To participate at a questionnaire the students must know the session-ID. With this

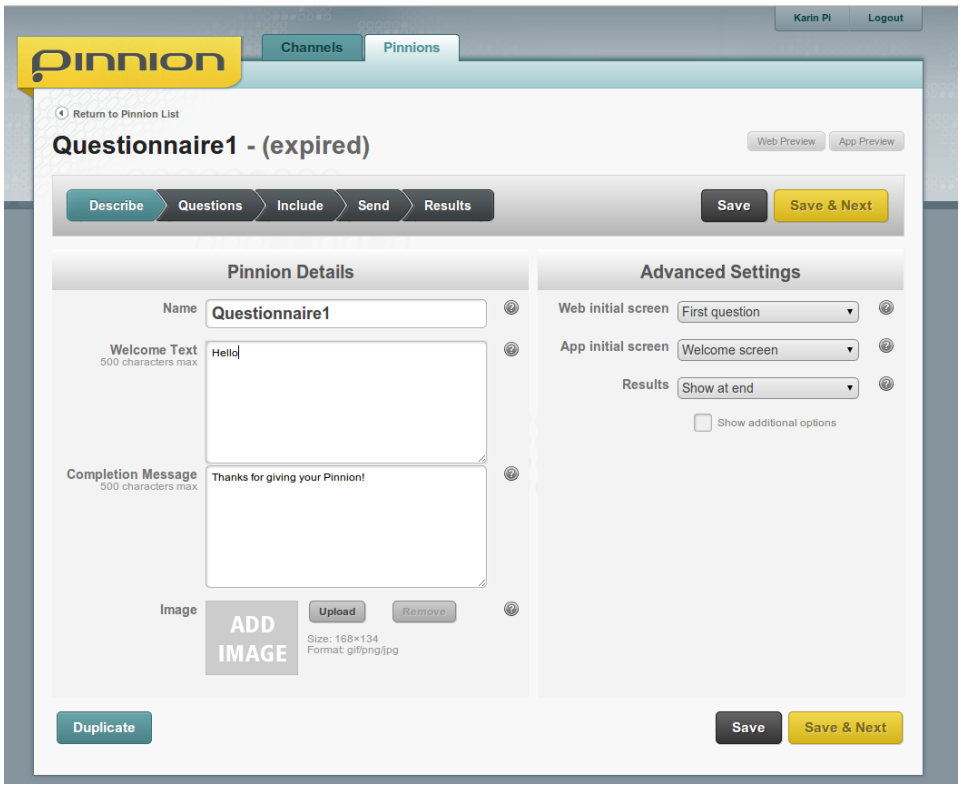

Figure 5.8.: This figure shows the administration interface of Pinnion.

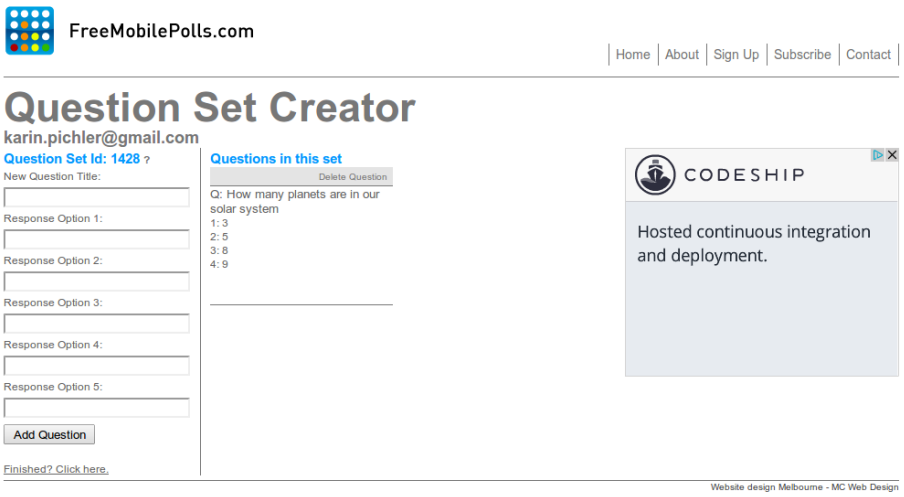

Figure 5.9.: This figure shows the administration interface of FreeMobilePolls.

session-ID, the students can register to a questionnaire, and they can start voting. The only way of answering a question is via the web-interface. The results are shown immediately to the students as well as to the lecturer. The results of the questionnaire are visualized as bar charts. The lecturer has no possibility of downloading the reports. There is also no possibility to store different sessions of a questionnaire. The results are lost as soon as the questionnaire is cleared. Figure 5.9 on page 40 shows the administration interface for the lecturer.

LectureTools: This ARS provides a complete environment for lecturing. The slides as well as the questionnaires can be created within the system. Figure 5.10 on page 41 shows the administration interface for the lecturer. If the lecturer wants to ask questions to the audience, a presentation must be created in the system. The students track the entire lecture via the web-interface. They can take notes, mark slides as important, mark slides as not understandable and chat with the lecturer as well as with their colleagues. To accomplish the tasks of the test scenario is extremely difficult with this system if one uses the system the first time. There are many functions and features provided which make the interface complex and difficult to

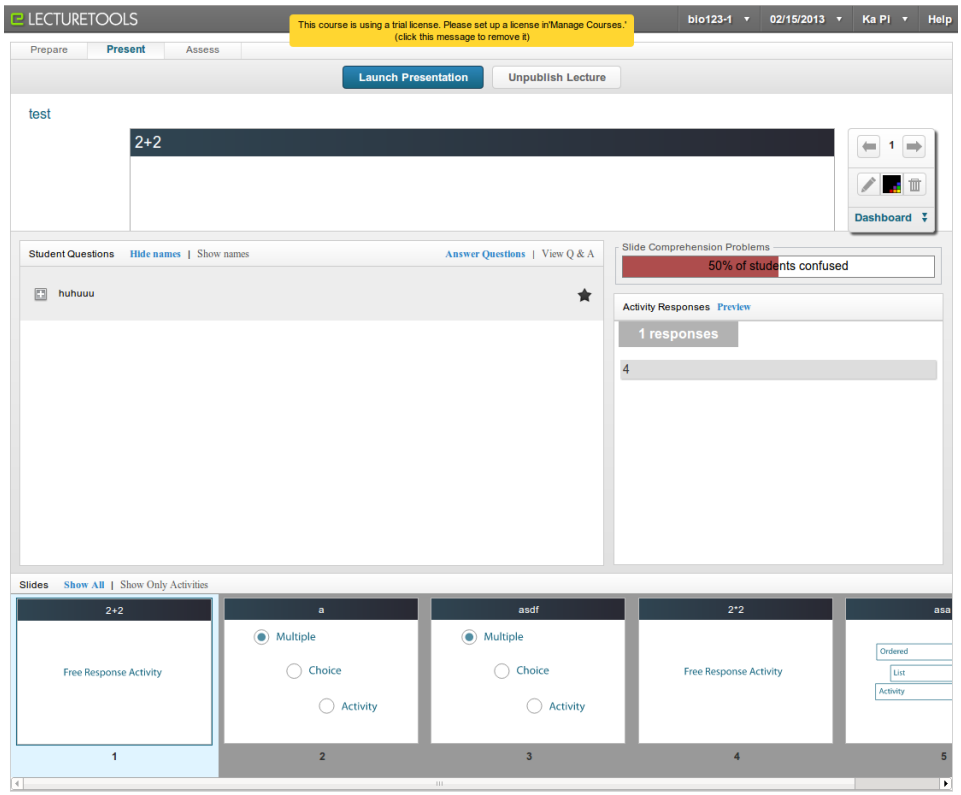

Figure 5.10.: This figure shows the administration interface of LectureTools.

understand. As soon as the students voted for an answer, the results are shown at the lecturers interface. Students do not get a visualization or presentation of the results. Compared to the other systems, LectureTools offers the most question types.

PINGO: Pingo is a very lightweight tool, which is easy to use even if the system is used the first time. The web-interface for the lecturer can be seen in figure 5.11 on page 42. Questions can be easily generated. The audience can respond anonymously to the questions. Answering works by clicking on the correct answer in the web-interface. The results of the questions are shown at the lecturers interface. The students do not see any results on

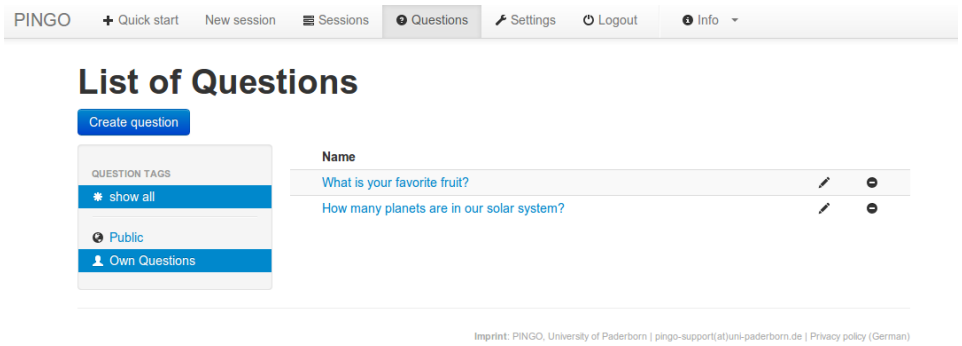

Figure 5.11.: This figure shows the administration interface of Pingo.

their devices. This system offers the possibility of looking at the results of previous sessions. The web site offers all information which are stored to the lecturer, there is no possibility of downloading the results.

Mentimeter: This ARS is similar to Pingo an extremely lightweight system. Figure 5.12 on page 43 shows the web-interface of Mentimeter. The lecturer can start immediately asking questions, without registering. The tasks of the test scenario could be accomplished very comfortable with this system. The single questions can be structured and summed up to questionnaires which are called "sessions" in this system. The audience can answer questions anonymously without providing their names. The results are shown at the lecturers interface as soon as the audience answers the question. The audience does not receive a visualization or presentation of the results at their interface.

Mentimeter offers premium features. In the paid version reports can be generated and reviewed. Therefore, the functionality of reports in Mentimeter could not be tested.

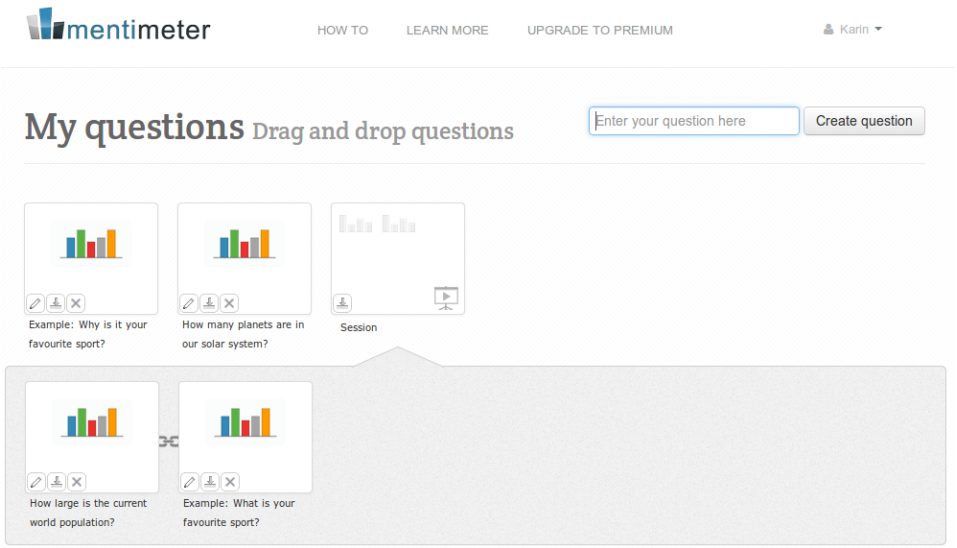

Figure 5.12.: This figure shows the administration interface of Mentimeter.

5.2. Comparison

In this section, the different tools are compared against each other regarding measurable and defined features. The features are summed up to groups. The features for comparison and the groups evolved during the analysis of the tools. Some of the features are based on the comparison which is provided by the *Poll Everywhere Comparison*. The groups are described shortly and then the comparison of the tools follows.

General Features of Questions and Questioning: This group includes all features, which are important for creating and asking questions and questionnaires. The following features are included in this group:

- Ways to respond (SMS, web site, mobile applications)
- Group questions to questionnaires
- Sort questions
- Copy questions
- Start and stop questions automatically

- Clear the results of a question without loosing previously collected data
- Bulk start / stop / clear / delete
- Anonymous voting

Question Types: Each ARS provides different types of questions. During the analysis of the ARSs, which were compared in this chapter the following question types could be distinguished.

- Multiple choice
- Yes / No questions
- Open questions, free text answer
- Sorting problem
- Matching problem
- Image quiz

Features of Questions: In this group, the different features of single questions are analysed. The following features are compared within this group.

- Hide votes until the question is closed for voting
- Show the results (number of votes) as totals and/or percentage
- Definition of correct answers
- Math equations

Visualization: This group includes all features regarding visualization of results with special focus on the lecturers interface.

- Visualization types (bar chart, pie chart, etc.)
- Visualization for students (Yes / No)

Reporting and Statistics: Some of the compared ARSs provide a reporting functionality. This section covers this functionality.

- Downloadable reports
- Output (csv, xls, etc)

- Individual results for each student, grading within the system
- Results from previous sessions

5.2.1. General Features for Questions and Questioning

The comparison of this group listed in table 5.2 on page 47. During the analysis of the different features for generating and asking questions the following key findings could be identified:

Some of the systems are especially developed for student interaction and summative assessment. The comparison of this group of features shows, which of the compared ARSs are focused on student interaction or summative assessment. It can be assumed that every tool, where the audience has to provide their names, has a focus on summative assessment. The system can track the votes of all students, and the lecturer gets an overview of the performance of each student.

Only four out of twelve systems offer mobile applications. It can be assumed that the most important way of responding to a question is via a web-interface. Each of the compared systems provides the functionality of voting via the website. Only four of the twelve evaluated systems provide mobile applications. Five of the evaluated systems offer the possibility of voting via SMS.

Ten out of twelve systems provide a way to organize questions as questionnaires. Grouping the questions into questionnaires is provided by ten of the twelve evaluated systems.

Six out of twelve systems offer the functionality of keeping the data of the old session when the questions are cleared. Nearly every system with focus on student interaction and summative assessment provides the functionality of storing data of old sessions and to review them. It can

be assumed that this functionality is provided to the lecturer to reflect on previous lectures.

5.2.2. Question Types

The comparison of the different question types, which are available in each system, is evaluated in this section. The results of the comparison can be seen in table 5.3 on page 48. No system offers all question types. The main findings of the comparison of the different question types are described below.

Every system offers multiple/single-choice questions. It can be seen that multiple/single-choice questions are the most important question type because every system offers this question type.

Seven out of twelve systems offer open questions. Open questions are provided by seven systems. It can be assumed that open questions have importance as well.

Systems, which are focused on student interaction and summative assessment, offer the most different question types. The systems with the most different question types are focused on student interaction. These tools are Top Hat Monocle and Lecture Tools. However, the other tools with focus on students, offer different question types as well. Whereas the systems, which are not focused on student interaction, provide only one or two question types.

5.2.3. Features of Questions

In this section, the different features are compared, which are offered for a single question. The main findings of this comparison are listed below.

System	Socrative	Top Hat Monocle	SMS Poll	mQlicker	Clicker School	Poll Everywhere	understoodit	Pinnion	Free Mobile Polls	LectureTools	PINGO	Mentimeter
Ways to respond												
SMS		●	●			●		●		●		
Web site	●	●	●	●	●	●	●	●	●	●	●	●
Mobile application	●	●			●			●				
Twitter						●						
Group questions	●	●		●	●	●	●	●	●	●		●
Sort questions	●	●	●		●	●		●		●		●
Copy questions		●				●		●				
Start / stop questions automatically						●						
Clear data of questions without loosing previous data	●	●		●	●	●	●					
Bulk start / stop / clear / delete	●	●			●	●		●	●	●		●
Anonymous voting		-	●	●		●	●	●	●		●	●

Table 5.2.: The general features for generating questions and questioning of the different systems are compared in this table. It can be seen that the voting functionality via a web site is provided by every system. Therefore, it can be assumed that this is a very important feature.

System	Socrative	Top Hat Monocle	SMSPoll	mQlicker	Clicker School	Poll Everywhere	understoodit	Pinnion	Free Mobile Polls	LectureTools	PINGO	Mentimeter
Multiple/Single-choice	•	•	•	•	•	•	•	•	•	•	•	•
Yes/No Question	•						•					
Open Question	•	•		•	•	•	•			•		
Sorting Problem		•								•		
Matching Problem		•										
Image Quiz										•		

Table 5.3.: Comparison of the different question types which are provided by the systems. It can be seen that the most important question type is the multiple/single-choice question.

System	Socrative	Top Hat Monocle	SMSPoll	mQlicker	Clicker School	Poll Everywhere	understoodit	Pinnion	Free Mobile Polls	LectureTools	PINGO	Mentimeter
Hide votes until the question is closed for voting		-						•	•		•	
View totals number of votes / percentage		-	•	•	•	•	•	•	•	•	•	•
View percentage of votes		•	•	•	•	•	•	•			•	
Definition of correct answers	•	•			•					•	•	
Math equations		•			•	•						

Table 5.4.: Comparison of the different features for a single question.

Four out of twelve systems hide the answers until the answer time stops. According to Bruff, 2009 it is important for students that the results cannot be seen immediately. This has the effect that students must think about the correct answer on their own. Another effect of this feature is that students can vote anonymously, without worrying that the other students immediately see how they have voted.

Five out of twelve provide the option for the lecturer to define which answer is the right one. This feature could be used for graded questionnaires too. The ARSs with focus on student interaction offer this kind of functionality. It can be assumed, that students may use this feature for learning and preparation for exams.

5.2.4. Visualization

This section shows the results for comparing the different visualization features of each system. Table 5.5 on page 51 shows the results of the comparison. Due to the results of this comparison it can be assumed that not much focus is laid on the visualization of results and statistics. The key findings are listed below.

Five out of twelve systems provide visualization on the student's interface. Seven out of twelve systems display the results to the lecturer only. Students cannot see the results at their interface or on their devices. If the lecturer does not present the results to the students, they have no feedback how the question is being answered.

Every system provides visualization of the results as bar chart for the lecturer. Each system represents the results as bar chart, at least to the lecturer. It can be assumed that this is the easiest way to display and interpret the results.

Two out of twelve systems provide visualization as pie chart. Only two systems offer another type of visualization. All other systems stick to the bar chart.

5.2.5. Reporting and Statistics

Table 5.6 on page 52 shows the results of the comparison of the reporting features. Not all systems could be evaluated regarding the report features because some tools offer reporting for premium members only. It can be assumed that the reporting function is something extremely valuable because some of the systems only offer reports for premium members. The main findings of this comparison are described below.

System	Socrative	Top Hat Monocle	SMSPoll	mQlicker	Clicker School	Poll Everywhere	understoodit	Pinnion	Free Mobile Polls	LectureTools	PINGO	Mentimeter
Visualization for lecturer												
Bar chart	●	●	●	●	●	●	●	●	●	●	●	●
Pie chart				●	●							
Visualization for students		-		●	●		●	●	●			

Table 5.5.: Comparison of the different visualizations provided by the systems.

Four out of ten systems provide downloadable reports. Only a view systems provide downloadable CSV reports. The two missing ARSs do not provide any information whether they offer downloadable reports or not.

Two out of ten systems produce reports for individual students. The systems with a focus on student interaction provide reports for individual students. This builds the basis for grading students with an ARS. The missing ARSs do not provide any information about this feature and it could not be tested because of premium membership.

Five out of eleven systems offer the results of previous sessions as data report. It can be assumed that accessing previous sessions is not essential to the systems which were compared. Only one third of all systems provide this feature. For the missing tool no information could be found whether this feature is available or not.

System	Socrative	Top Hat Monocle	SMSPoll	mQlicker	Clicker School	Poll Everywhere	understoodit	Pinnion	Free Mobile Polls	LectureTools	PINGO	Mentimeter
Downloadable report	•		•	•		•						
Output (filetype)	csv		csv	xls		csv						
Results from previous sessions	•			•		•	•				•	

Table 5.6.: Comparison of the different report features of the systems.

5.3. Strengths and Weaknesses of RealFeedback

The previous section compared different web-based ARSs. This section compares the findings with the features which are provided by RealFeedback. The strengths and weaknesses of RealFeedback are discussed.

5.3.1. Strengths

This section covers the strengths of RealFeedback.

Clear Interface: The different ARSs are tested according to the test scenario described in section 5.1 on page 29. With some systems, it was difficult to accomplish these tasks because the systems provided an extraordinarily feature-rich interface. Because of the many features, the systems got complex and hard to understand. RealFeedback offers a remarkably clear interface compared to these ARSs. The user-friendly interface of RealFeedback can be assumed as strength because the lecturer and students can easily work with the system.

Ease of Voting for Answers: Students can start voting easily compared to some of the other systems. The students do not have to register in order to vote for an answer. The students can see the questions and the whole answer text. Some of the other ARSs do only show numbers or letters as answer and do not show the text itself. The student can vote by simply clicking on the right answer. For some of the other systems, especially those who offer to voting via SMS, voting for an answer is not that easy to achieve. The audience has for example to type in a five-digit number, which has nothing to do with the answer itself.

Votes are Hidden Until Question is Deactivated: RealFeedback hides the results from the audience as long as the audience can vote for an answer. According to literature, this feature has the effect that students have to think on their own and are not influenced by the answers of other students. The students have no fear of being wrong because other students cannot see their votes immediately [Bruff, 2009].

Downloadable Report: A downloadable report as csv or xls file is only provided by four systems. RealFeedback offers a basic download of the statistics for all projects of a lecturer. Reflecting on the results is a crucial part of introducing an ARS in a lecture or to a lecturer [Beatty and Gerace, 2009, p 26]. When reports are provided, the lecturer has the possibility to reflect on the session. The lecturer can do manual analysis.

Anonymous Voting: Anonymous voting is also strength of RealFeedback. Anonymity engages shy or introverted students into the process of participating at the lecture. According to literature it is important to allow anonymity within ARSs. Especially the systems with a focus on student interaction often do not provide anonymous voting [Feldman and Capobianco, 2007; Bruff, 2009, p 103; Collins, Tedford, and Womack, 2008; Hoffman and Goodwin, 2006; Gauci et al., 2009; Abrahamson, 1999; Roschelle, 2003].

Free to Use: RealFeedback is free to use for the lecturer as well as for the students. Each feature is available for everyone and there are no premium features. Some ARSs are not free to use or offer premium features. In many cases, the lecturer or the university has to pay license fees, and in some cases even the student has to pay a fee per semester in order to use the system.

5.3.2. Weaknesses

RealFeedback does also have some weaknesses compared to the other ARSs. The weaknesses of RealFeedback are discussed in this section.

Question Types: RealFeedback provides only single choice questions. There are many systems, which provide more than just multiple or single choice questions especially the ARSs with focus on student interaction. Some of them even provide the functionality of uploading images for a question.

Mark Answers as Correct: When creating questions the lecturer has no possibility to define which answer is the correct one. If the lecturer could mark an answer as correct, students can use the questions for further learning process. For example, they could prepare for exams when they can see which answer is the right one.

Response Possibilities: Some of the systems offer many ways of responding to questions whereas RealFeedback provides voting only via a web-service. If RealFeedback would provide a mobile application, it would be easier and faster for students to vote. If one must open the browser on the smartphone, type in the URL in order to vote for a question it is more effort than opening an installed mobile application.

Strengths	Weaknesses
Clear Interface	Question Types
Ease of Voting for Answers	Mark Answers as Correct
Votes are Hidden Until Question is Deactivated	Response Possibilities
Downloadable Report	Keeping Data of Previous Sessions
Anonymous Voting	Visualization of Results and Statistics
Free to Use	

Table 5.7.: Strengths and weaknesses of RealFeedback compared to other systems.

Keeping Data of Previous Sessions: Some ARSs offer the option to view the data of previous sessions because the data is stored. The lecturer can use the data to reflect on previous lectures or to compare the performance of the different semester. RealFeedback only stores the current session for every question. The data is lost, as soon as a question is cleared.

Visualization of Results and Statistics: RealFeedback visualizes the results a bar chart for each question. Other systems have more ways of visualizing the results. For example, pie charts or different table views are used for visualization and presentation of the data. Overall statistics for an entire project cannot be reviewed in RealFeedback because there is no visualization for this kind of data.

6. Interviews

To get feedback of the current functionality of RealFeedback lecturers who currently are using RealFeedback are interviewed. During the quantitative interview, some questions, which are relevant to the lecturer, emerged. These questions are used to improve the questions of the following interviews.

6.1. Structure of the Interview

The interview is designed regarding the guidelines described by Turner, 2010. This section describes the structure of the interview. The interview consists of groups of questions. These groups are described in detail.

General Questions Regarding the Lecturer: This part of the interview covers general questions regarding the person who was interviewed. It is asked how long the person holds lectures at university, if the person has ever used an ARS before and what kind of technology the person uses during lecture. The intent of this part of the interview is to get a general overview of the interview partner. If the interview partner has already used other ARSs, she has a feeling for this kind of systems. It is asked what technology is used for lecturing and presenting the material.

General Questions Regarding RealFeedback: The intent of this part of the interview is, to get an overview how the lecturers use RealFeedback. It is asked how often the lecturers are using RealFeedback in their lectures and what are their main goals. Another interesting point which is asked in

this section, are the experiences they had during lecturing with RealFeedback.

Question Generation in RealFeedback: This group of questions covers the topic of question generation. This includes the time and effort the lecturer needs to generate questions before the lecture starts. Another important topic of this group is, if the lecturer can imagine to ask questions instantly during the lecture without preparing them.

Usability of the System: This group contains questions regarding the usability of RealFeedback. The aim of this group is to get an overview how the usability of the systems appears to the lecturer. The usability of the system is an important point when developing or changing a system. The system must be save, effective and efficient to use, easy to learn and easy to remember [Rogers, Sharp, and Preece, 2011].

Statistics and Reports: The intent of this section is to get an idea of how valuable statistical overviews are helpful for the lecturer. It is asked which statistics are useful to the lecturer and which one are not needed within the system. The interviewees are asked if they prefer some kind of statistics or reports. This section also covers the topics of reusing question and comparing sessions against each other.

Other: This group contains general feedback from the interviewees. Every feedback, which is not covered in the groups mentioned before, is covered in this group. One question of this group is regarding the impression of the students and what the students dislike.

6.2. Results of the Interview

This section covers the results of the interview. The results are dedicated to the groups, which were described in the previous section 6.1 on page

Lecturer	Experience in teaching	Knows other ARSs
A	lecturing since 2001	No
B	first lecture in 2013	No
C	internal training courses since 2011	No
D	lecturing since 2000	Yes

Table 6.1.: Four lecturers who use RealFeedback were interviewed. This table shows some general information about the lecturers.

57. In general, the lecturers who were interviewed have a positive attitude towards RealFeedback. All of them had the impression that students like to use RealFeedback during lecture. The lecturers do not use the system for summative assessment because the votes of the students are anonymous in RealFeedback. The lecturers use RealFeedback to keep the attention of the students high and to provide students a method for self-assessment. The questions are chosen to make the students think and not straightforward questions, which the students know immediately.

6.2.1. Results: General Questions Regarding the Lecturer

The results for this group of questions are summed up in table 6.1 on page 59. Only one of the interviewed lecturers has used ARSs before. All of the interviewees use different kind of technology during their lectures. The range goes from Power Point, to web sites, coding examples and the usage of the blackboard.

6.2.2. Results: General Questions Regarding RealFeedback

All of the interviewed lecturers used RealFeedback at least once. The overall impression of RealFeedback is high, and all of the interviewed lecturers are going to use RealFeedback in future. Three of the four interviewed lecturers have the ambition to activate students during lectures and to keep

their attention high when using RealFeedback. Another goal of the lecturers is to gain information about the student's level of knowledge in a certain topic. One of the interviewed people used the system during her internal training courses where she asks questions regarding the attendees. Depending how the students responded to a question the lecturer can see whether the students understood the material or not. If many students did not understand the material, the lecturer can explain it again.

6.2.3. Results: Question Generation in RealFeedback

All of the four interview-partners prepare questions before the lecture starts. The time, which is needed to generate the questions, is different, depending on the lecturer and the material she covers. The times the lecturers use for question generation are listed in table 6.2 on page 61. The results in this table include the thinking process as well as the generation of the questions.

The aim of the questions is to make the students think in depth about the correct answer and to keep their attention. One lecturer mentions that the attention of the students is extremely high when the question is going to be solved by the lecturer. Therefore, the learning effect of the students is also huge at this point of time. The students can benefit from this moment as long as a relevant question is being asked. Three of the four interview-partners think that it is hard to ask live questions during lecture because they must think of original wrong answers, which are not too easy nor to hard.

6.2.4. Results: Usability of the System

All of the interview-partners like the usability of the system. Each of them said that the system is easy to use for generating questions and working with it during lecture. Three of the interview-partners mentioned that they enjoyed the clear interface of RealFeedback and the ease to use the system.

Lecturer	Time for creating a question
A	30 minutes
B	4 minutes
C	5 minutes
D	10 minutes

Table 6.2.: This table lists the time, which is needed by each interviewed lecturer to generate a question. This includes the thinking process as well as the generation of the question.

6.2.5. Results: Statistics and Reports

All interviewees liked the visualization, which is offered by RealFeedback. The bar charts are easy to read and to understand. From their perspective the information, which is provided for each question is enough. Three of the four interviewed lecturers would like to get a comparison of the different sessions of a project or of single questions. With this feature the lecturers expected to see the improvement of the students or of the lecturing style over the years.

The reports, which are provided as CSV file are mentioned as beneficial by three of the four interviewed lecturers. The reason is that lecturers can use the data for doing their own analysis of the results. All interviewed lecturers would like to see how many of the students, who are registered to a session, have already voted. This helps them to estimate when the question can be stopped.

6.2.6. Results: Other

All interviewed lecturers say that the feedback of the students is positive. Some of the students would like to have a mobile application. This would make voting faster. One interview partner mentioned that it would be helpful to her when notes can be added to questions. With this information, reusing questions at a later point of time is easier.

Important features to the lecturer	Mentioned by interviewees
RealFeedback provides a clear interface and is easy to use	4
Starting questions during lecture is simple	4
Students like to use the system during lecture	4
RealFeedback is used to gather information about the knowledge of students	3
Asking live questions during lecture cannot be done easily	3
Questions are prepared before lecture starts	3
The reports which are provided by RealFeedback are valuable	3

Table 6.3.: This table lists the functions which are useful to the lecturer and positive statements.

The visualization of the mobile web site can be improved. One of the interview partners suggested making the font smaller because now it is too large. With this improvement, more information could be displayed.

6.2.7. All Results Summarized

This section presents all results. The results are split in the facts that the interviewed lecturers mentioned as valuable and useful, and features which are missing. The facts and useful features of RealFeedback are listed in table 6.3 on page 62.

Table 6.4 on page 63 lists the features which are missing. These features are relevant to the lecturers as well. When implementing new features in RealFeedback, it must be considered to keep the concept of the clear and easy to use interface.

Missing features	Mentioned by interviewees
A visualization of how many students have already voted for the question is missing	4
Defining which answer is the correct one is missing	3
Comparison of the results over different sessions is missing	3
A mobile application is missing	1
A function for taking notes for a question is missing	1
Visualizing how fast the students responded to a question is missing	1

Table 6.4.: This table shows the missing features which are mentioned by the interview-partners.

7. Solutions and Visualization Possibilities

RealFeedback is described in chapter 4. It is explained how RealFeedback works, as well as the technical background and the database structure. In chapter 5 the different web-based ARSs are evaluated and compared against each other. In chapter 6 the results of the interviews are presented.

With this knowledge and the findings, the different visualization possibilities can be explored. This section compares different possibilities of presenting statistics and reports to the lecturer. The different visualization possibilities are described, and the strengths and weaknesses of each approach are discussed. The last section of this chapter 7.5 defines the findings and the solution, which is implemented in RealFeedback.

7.1. Summary of the Findings

The findings of chapter 5 and chapter 6 are summarized in this section. The results of each chapter are connected to each other in table 7.1 on page 66. The first column represents the findings of the comparison of the different ARSs and the second column represents the findings of the interview which are relevant. The first four rows are the results which are relevant for this chapter. The features which are mentioned in these rows are:

- Comparison of the results over different sessions is missing.
- Defining which answer is the correct one is missing.

Weaknesses of RealFeedback based on the comparison	Results Interview
Keeping data of previous sessions is analysed as missing	Comparison of the results over different sessions is missing
Mark answers as correct is a weakness of RealFeedback	Defining which answer is the correct one is missing
Visualization of results and statistics is a weakness of RealFeedback	A visualization of how many students have already voted for the question is missing
Visualization of results and statistics is a weakness of RealFeedback	Visualizing how fast the students responded to a question is missing
Response possibilities	A mobile application is missing
	A function for taking notes for a question is missing

Table 7.1.: The mayor findings of the previous chapters are connected in this table. The first four rows are relevant for the question, which features should be added to RealFeedback.

- A visualization of how many students have already voted for the question is missing.
- Visualizing how fast the students responded to a question is missing.

7.2. Keeping Data of Previous Sessions

Keeping data of previous sessions would be a valuable improvement for the interviewees. The comparison of the different ARSs let assume that this feature is extremely valuable to users of such systems. Very often detailed reports and statistics over sessions are only offered to premium members who pay for the system. These facts let assume that sessions and comparisons between the different sessions is a beneficial feature. The interviewees mentioned that the most valuable point of this feature is that they can use the data to reflect on previous sessions. Literature mentions the importance of reflecting over the results of the ARS [Beatty and Gerace, 2009, p. 26].

When sessions are implemented in RealFeedback, the lecturer can measure the in- or decrease of knowledge of each student over time. The different sessions can be compared against each other, and the lecturer gets feedback of the lecture.

7.2.1. Changes in RealFeedback

This section describes how RealFeedback must be changed to introduce sessions into the system. RealFeedback does not store the data of individual sessions. The data is lost as soon as a question is cleared and cannot be recovered. Therefore, it is necessary to introduce the concept of sessions into the system. In section 4.3 the database structure of RealFeedback is described. The structure does not provide a way to store different sessions. Sessions must be integrated into the database structure. Three additional collections must be added to the database:

- sessions
- session_questions
- session_answers

session: Each project consists of zero or more sessions. As soon as the questions are cleared, they must be stored as a new session. The structure of the questions and answers stays the same. The data must be saved into a new session before clearing the data. The session collection can be seen as a container into which the questions and answers of a project are copied. A session must provide the following information:

_id Unique ID of the session.
date The time when the session was created.
question_list A list of all question-IDs which belong to this session.

session_question: The collection session_question holds single questions. Each question, which belongs to a session is stored in the question_list of the session. In the session_question collection the following information must be stored:

_id Unique ID of the question.
answer_list A list which contains all answer-IDs which belong to this question.
text The text of the question.
start_time The time when the session was started.
duration This field stores the duration of how long the question was active.
votes The number of total votes for this question.

session_answer: In this collection the answers are stored which belong to a question. An answer must provide the following data:

_id Unique ID of an answer.
text Answer text.
votes Number of votes for a single answer.

7.3. Marking Answers as Correct

Many of the compared web-based ARSs support the feature of marking which answer is the correct one. With a good visualization, this information can be beneficial to the lecturer. This information could be useful to the lecturer when she analyses and reviews the performance of her students. She can see quickly how the students performed and which topics do the students not understand. Students can take a look at the questions if they know the session-number. They can use the system for learning or when they are preparing for an exam, because the correct answer can be identified.

The lecturer may not necessarily want to ask only right/wrong questions. For example, some questions might be regarding the demographics of the students. Therefore, it is useful that the lecturer is not forced to provide information about which answer is correct. However, she should have the option to provide this information.

7.3.1. Changes in RealFeedback

To include this feature into RealFeedback a system must be provided to store which answer is the correct one. This requires a change in the database structure of RealFeedback. The collections "answers" and "session_answer" must change. A field must be added which indicates whether the answer is correct or not.

As a second step, the interface of RealFeedback must change. The lecturer must have the possibility to provide this information when the question is generated. The easiest way for achieving this would be a checkbox beside each answer. The answer, which is marked with the checkbox, is the correct one.

7.4. Visualizations

The comparison of the different systems and the interviews let assume that it is essential to get a quick overview over different results and statistics. According to literature, it is easier for people to read information out of visualizations than numbers. This also depends on the type of visualization [Yau, 2011, p. 2]. During the analysis and research, different visualization possibilities for RealFeedback emerged. These visualizations are described and explained in this section.

The findings of the interviews reveal that visualizations of statistics and results are useful to the lecturer. Besides the findings described in section 7.2 on page 66 and section 7.3 on page 68, which were named by the interviewees one could think of other visualizations, which could be useful and helpful to the lecturer.

This section covers the missing visualizations mentioned by the interviewees as well as other results and information, which could be visualized in RealFeedback. Every visualization has strengths and weaknesses, which are described and analysed.

To think of different types of visualization RealFeedback is split into different hierarchy levels. The hierarchy of the single elements of RealFeedback

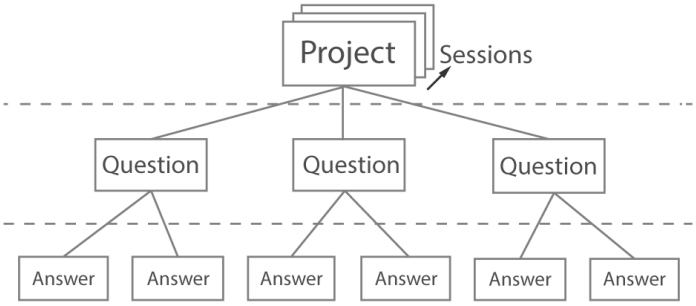

Figure 7.1.: This figure shows the hierarchy which can be created for RealFeedback. The basic element is the project, which consists of one or more questions, and each question consists of one or more answer. The sessions, which are shown in this diagram, are not part of RealFeedback now but they are introduced as a new feature. For every level of the hierarchy visualization is possible.

is shown in figure 7.1 on page 70. The top most element is the project. A project consists of questions and each question consists of answers. Visualizations can be created for every level of the hierarchy. The findings demonstrate the need for sessions, which are also included in this hierarchy. Each project has zero or more sessions.

It can be assumed that, in every level of the hierarchy, visualization of statistics and information is possible. The visualizations possibilities are covered in the next sections:

- visualizations for single answers
- visualizations for single questions
- visualizations for single projects
- visualizations of different sessions of a project

For every level of the hierarchy, theoretical solutions are explained and evaluated. The strengths and weaknesses of every visualization are discussed based on the findings of the interviews and based on literature. For every level of the hierarchy, one visualization is chosen as a solution.

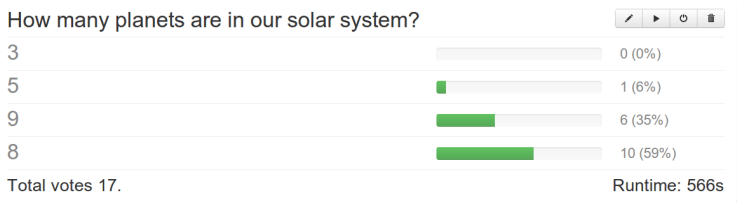

Figure 7.2.: This figure shows how the results of a single question are visualized now. The results for the question can be seen as bar chart for each answer. The number of total votes, as well as the number in percent is provided on the right side of each bar chart.

7.4.1. Visualizations for Single Answers

The lowest levels of the hierarchy in RealFeedback are answers. Even this level offers possibilities for different visualizations and additional data, which could be gathered. At the moment, the system stores the text of an answer, the number of votes and to which question the answer belongs. RealFeedback visualizes the number of votes for the lecturer and the students. The votes are visualized as bar charts with information about the total number of votes and the percentage of votes for each question. This is the visualization, which is provided by RealFeedback now. How this visualization looks like can be seen in figure 7.2 on page 71.

During the analysis of the database structure and the server structure of RealFeedback the possibility of saving the time-entries for each voting event emerged. This does mean that as soon as a student votes for an answer the time needed for voting is stored. With this information, the lecturer can retrieve how fast students answered. This could be a measure for the difficulty of a question or if students could answer it quickly because they know the right (or wrong) answer immediately. When the information of the different voting times is stored, the mean voting time for an answer could be provided to the lecturer as well. The presentation of these two measures, voting time for each vote and mean voting time must be integrated into RealFeedback.

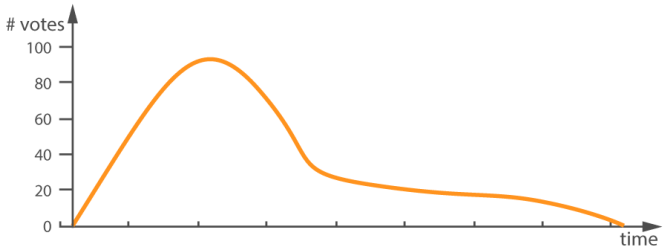

Figure 7.3.: This figure shows the visualization of the voting time entries for single answers. The visualization could appear as soon as the lecturer moves the mouse over the result bar chart of an answer.

Voting Time Of All Votes

When the system gathers the time of each voting event, this data should be visualized. Providing the time entries as numbers makes it difficult for the lecturer to read and to evaluate. The more students vote for an answer the more time-entries the lecturer would have to read. Visualizing the time-entries as line chart, which is presented to the lecturer when she clicks, for example, on an answer is easier to read and interpret. The visualization of such a representation can be seen in figure 7.3 on page 72. On the y-axis, the number of votes is plotted and the x-axis views the different points of time. The lecturer can see quickly whether the students answered immediately or if they had to think about the right answer.

Since the lecturer knows which answer is the correct one, she can reason about how fast students can find the correct answer. Measuring the time how long it takes students to vote, allows to reason about the difficulty of the question. It can be also used as an indicator if the question should be changed before reusing it in the next lecture.

To implement this feature in RealFeedback, the time entries of each voting event must be stored for each answer. The database structure must be changed to save the time entries. Each answer must contain information about how fast students voted. The easiest way to include the time-entries

is a list for each answer, which holds the time entries. The interface of the lecturer must provide a way of showing the diagram. For example, by clicking on a button or the bar chart for one answer. This solution is located at the lowest level of the hierarchy in figure 7.1. For reviewing the diagram, the lecturer must navigate deep into the hierarchy. It can be assumed that it offers specialized information. During the interviews, one interview partner mentioned this feature as valuable. To prove whether this feature is useful to the lecturers or not, it must be implemented. After the implementation, further interviews and usability studies must be organized.

Mean Voting Time

The mean voting time can be calculated for each answer. The aim of this solution is that, the lecturer can get a quick overview how quickly the students decided on average for an answer. To provide this information the same data must to be collected as described in the previous section 7.4.1. Each time entry of the voting events must be saved for every answer. The mean voting time for an answer is calculated as the mean over all voting time entries for this answer. Since this is only one single number a visualization is not required. How the integration of the mean voting time could look like, is shown in figure 7.4 on page 74. The dashed line marks the changes in the interface of the lecturer. Each answer provides the mean voting time as additional information.

Since this information is also located at the lowest level of the hierarchy of RealFeedback, (figure 7.1) it must be carefully analyzed before it is added. It can be assumed that this kind of information is not relevant. The basis for this assumption is that none of the interview-partner mentioned the feature and none of the compared systems provide this information.

This information can be included easily into the system. Although, effort must be laid on keeping the interface clear. All interview-partners like the simplicity and clearness of RealFeedback. Any additional information makes the interface a bit more complicated.

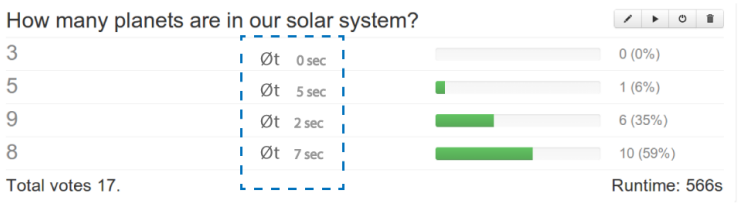

Figure 7.4.: This figure shows the visualization of the changes in the interface if the mean voting time is added. The dashed line in this figure marks what must be changed.

7.4.2. Visualizations for Single Questions

The next level in the hierarchy is the one of questions. Presently RealFeedback provides a brief overview of the results for a single question. The results for each answer of a question are visualized as bar chart, and information about the total votes and the percentage of votes are included. The presentation of the results of RealFeedback can be seen in figure 7.2 on page 71. The number of votes and the duration of the voting time are presented.

During the research and the comparison of the other ARSs some other visualization possibilities for this level of the hierarchy emerged. These visualization methods are described in this section.

Voting time of all votes

As described in section 7.4.1 on page 71 the time-entries for each voting event can be collected. The possibility for visualizing the voting time entries for a single answer is the line chart in figure 7.3. The voting time entries of all answers can be summarized in a single line chart. Figure 7.5 on page 75 shows the visualization of all voting time entries for a single question. Each line represents the voting time entries for a single answer. The lecturer can get a quick overview, how fast the students choose the different answers.

The same changes are needed, which are already described in section 7.4.1. Instead of drawing just one line, a line for each answer is drawn in the line

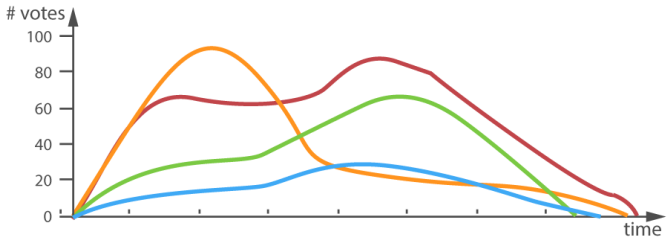

Figure 7.5.: This figure visualizes the voting time entries for a single question. Each line represents an answer. The lecturer can see quickly which answers the students choose immediately.

chart. The visualization must be integrated into the interface. This integration can be realized with a button. Each question has its own button, which opens up the visualization when clicking on it.

This feature is mentioned by one interview-partner. None of the compared ARSs offer this feature. To verify if this feature is useful or not, further research is needed.

Mean Voting Time

The option of calculating a mean voting time is already mentioned in section 7.4.1. This calculation can be applied to a single question as well. To calculate an average voting time for a single question the voting time entries for the question must be collected. The mean voting time can be calculated as average over all time entries for this question.

To include this feature into the system the time entries for each voting event must be saved. Contrary to the mean voting time for an answer, it is sufficient to save the time entries for the question instead of an answer.

The representation of the mean voting time can be added to the interface as a single number. As already mentioned, more information makes the system more complicated.

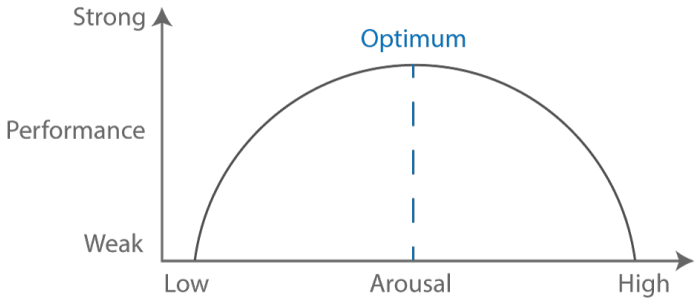

Figure 7.6.: This figure shows the Yerkes-Dodson law [Yerkes and Dodson, 1908] in a simplified version. The optimal performance can be reached when the arousal or the stimulus is between low and high.

Visualizing Correct Answers

As already mentioned in section 7.3 on page 68 the feature for marking answers as correct is implemented in some web-based ARSs. If this feature is added to RealFeedback, the lecturer has the option to specify which answer is correct. When this information is provided the lecturer must not read all answers or remember which answer is the correct one. When the correct answer is marked the lecturer can get a quick overview how much of the students answered the question correct. When this information is provided to the students as well, they can use the system when they prepare for exams. Two of the four interview-partners also mentioned this feature as useful.

It is necessary to mention, that according to literature it is not always the best way to inform the students about the correct answer immediately after the voting has finished [Dufresne et al., 1996; Beatty, Leonard, et al., 2006]. Students benefit from discussions after the voting has finished. The different answers are analysed and discussed together with the lecturer. Another theory is the Yerkes-Dodson law [Yerkes and Dodson, 1908]. It shows the relation between the stimulus and the performance. Figure 7.6 on page 76 shows this relation. When the arousal of students is in the middle of the scale, the optimal performance could be achieved. When students are ex-

Figure 7.7.: This figure shows how the correct answer can be visualized. It is important to keep color-blindness in mind [Ware, 2004]. The first column shows how non-color-blind people can see the visualization. The second column shows how people with Deuteranopia color-blindness can see the visualization. It can be seen that it is essential to choose colors, which can be differentiated by everyone.

cited before the correct answer is revealed this effect could be reached.

This different fact let assume, that it may be useful, when the lecturer can decide if she provides the information or not.

Figure 7.7 on page 77 shows a method how to visualize the information. This figure shows two visualization alternatives. The first alternative colors the correct answer green and the second alternative colors the correct answer blue. This figure also shows how color-blind people - with Deuteranopia color-blindness - would see the different visualizations. The colors in the first alternative are hard to distinguish. The second option is clearer to see for people with Deuteranopia color-blindness. It is essential to choose colors, which can be distinguished by everyone [Ware, 2004, pp. 99].

How the database structure must change, is already discussed in section 7.3 on page 68. To present the information to the lecturer the visualization must be changed. How this change looks like is discussed in figure 7.7.

Different Visualizations

RealFeedback visualizes the results of the answer as bar chart. There are some ways to visualize the results. Some of the other ARSs support pie charts as well. However, during the research and analysis some different approaches of visualizing the results of a question emerged. The different types of visualization are listed below and a brief overview of how these visualizations look like can be seen in figure 7.8 on page 79.

- Bar chart
- Pie chart
- Stacked bar chart
- Bubble chart

The comparison on the different ARSs let assume that the bar chart is the most common way to visualize the data. RealFeedback does already provide the visualization of the results as bar chart. To evaluate if the other visualizations have other advantages, further research is necessary.

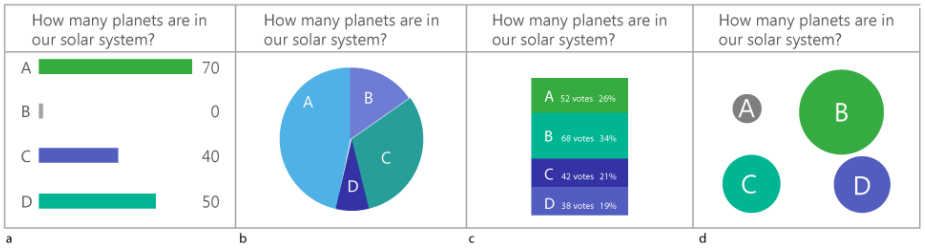

Figure 7.8.: This figure shows the different possibilities of visualizing results for a single question. Figure a) shows a bar chart, b) shows the visualization as pie chart, c) shows a stacked bar chart and figure d) shows the visualization as bubble chart.

Visualizing Amount of Students Who Have Voted

The interview-partners mentioned that they cannot decide easily when to deactivate a question for voting. They cannot estimate the time students need for voting. Some of the existing ARSs provide the information how many students are registered to a session. This information shows up in the lecturers interface as a single number. To include this feature into RealFeedback the number of registered students who have entered a session-ID must be counted. When the lecturer starts a question, the information can be offered how many students of the registered students have already voted for the question. The visualization as pie chart of this information makes it easier for the lecturer to see. This visualization can be seen in figure 7.9 on page 80. The lecturer gets a quick overview by looking at this pie chart. It can be seen how many votes are missing and the lecturer can decide easier when to deactivate the question for voting. Another way of visualizing the number of students who have already voted is a stacked bar chart. This bar chart can be seen in figure 7.9 too.

7.4.3. Visualizations for Single Projects

If the lecturer wants to get an overview of the results of a project, she has to open the project and look at each question separately. As soon as the

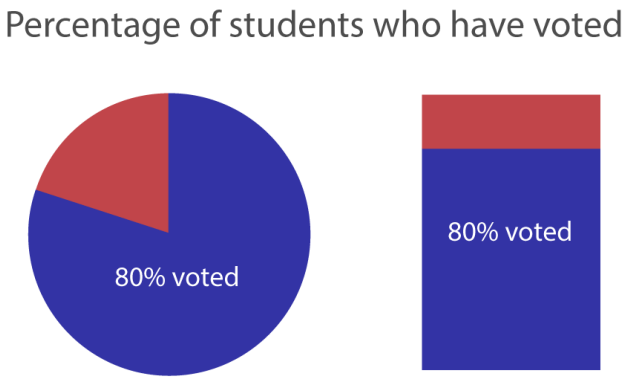

Figure 7.9.: This figure shows the visualization of how many students have already voted for an answer and how many votes are missing. Two possibilities of visualizing this information are shown in this figure. One visualization is a pie chart and the other a stacked bar chart.

project contains many questions, the lecturer cannot get a quick overview of the results. The only possibility, which is offered by RealFeedback now is to download a CSV file. This CSV file contains the results of all projects of one lecturer. There is no way to get the results for one project separately as CSV report. If a lecturer has many projects the CSV report contains much data and it is hard to find the relevant data.

The results of the interview show that it is important to the lecturers to have a possibility to reflect on the session of a project. When looking at the hierarchy in figure 7.1 again, this feature is located at the top most level. This section covers the approaches for visualizing the results of the project in a quick way. The strengths and weaknesses are discussed, and at the end one visualization is chosen which fits the needs best.

No matter which visualization is chosen it must be integrated into the lecturers interface of RealFeedback. The integration does not depend on the type of visualization. It can be realized for all visualizations in the same way. The most common way to provide access to the visualizations of the results of the whole project is by adding a button to the lecturers interface.

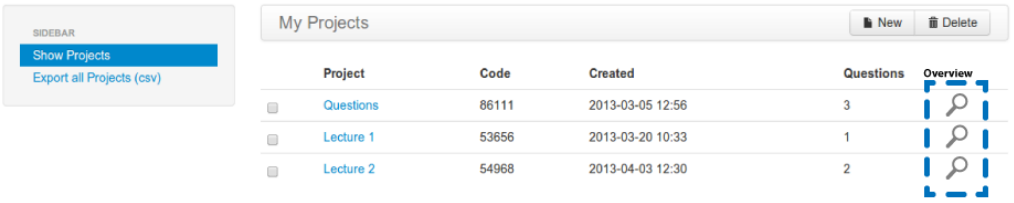

Figure 7.10.: The integration of the different visualization types for getting a quick overview of one project in the project-overview is shown in this figure. The dashed-lines indicate what must be added to RealFeedback. When the user clicks on the magnifying glass, the visualization appears. This integration can be used for the visualization of different sessions too.

As soon as the lecturer clicks on the button, the visualization appears and can be reviewed and analysed. Although a button is a simple and common element, it must be kept in mind that it is crucial to keep the system as easy as possible without adding too much complexity. How the button could be integrated into the project overview of RealFeedback is shown in figure 7.10 on page 81. The integration into the detailed view of a project can be seen in figure 7.11 on page 82.

The different visualization possibilities for reviewing the results of a entire project are covered in this section. For each alternative, an example is provided and the strengths and weaknesses are discussed. The following visualizations emerged during researching the topic:

- Bar chart
- Pie chart
- Stacked bar chart
- Bubble chart

Bar chart

The bar chart is probably one of the most common ways to display data. [Yau, 2011]. A bar chart is already used for displaying the results for a question in RealFeedback. For each answer, a bar appears which shows the total number of votes for the answer. The same visualization can be

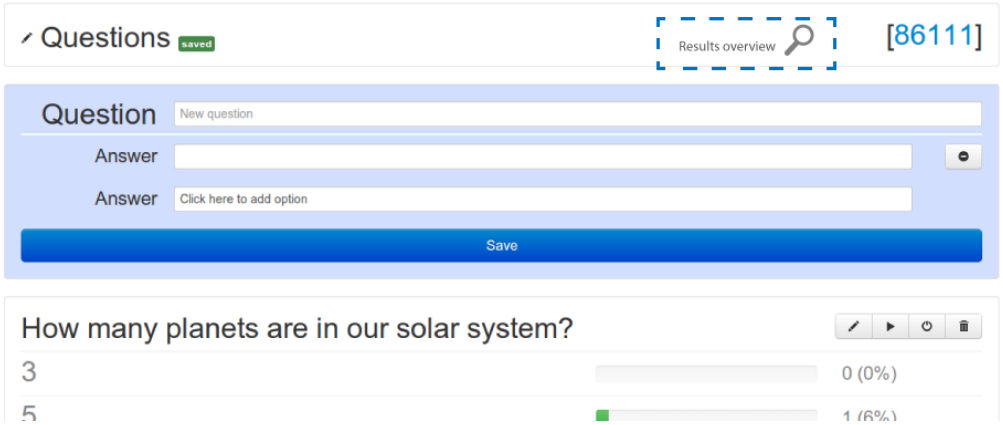

Figure 7.11.: The integration of the different visualization types for getting a quick overview of a project in the single project view is shown in this figure. The dashed-lines indicate what must be added to RealFeedback. When the user clicks on the magnifying glass, the visualization appears. This integration can be used for the visualization of different sessions too.

used to display the overall results of a project. When using the bar chart for visualizing the overall results of a project the main difference is that, the visualization is more compact than the visualization for the results of a single question.

One strength of the bar chart visualization is that the lecturer can quickly see how many votes each answer has. The lecturer can compare the results of the answers easily against each other. Therefore, a bar chart provides a quick overview of the results in this case. Figure 7.12 on page 83 shows how the overall visualization of a project could look like when the results are visualized as bar chart.

The visualization of the overall results of a project can be interpreted as a table. In each cell of the table, a question with all its details and results is displayed. One problem of this visualization is that the lecturer cannot see the text of the different answers. The text is removed to keep the general overview compact. An alternative is to provide the answer text when the lecturer moves the mouse over an answer. This requires the chart to be interactive.

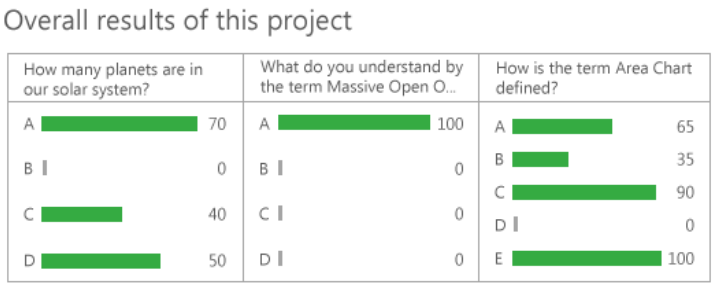

Figure 7.12.: The overall results of a project visualized as bar chart. RealFeedback does already use bar charts for presenting the results for one question. The main difference is that this visualization does compact the results so that more than one result can be seen at a time without scrolling through the different questions of the project.

The bars of the bar chart have all the same color. The aim of this visualization is to keep the overview simple. It is not necessary to color the charts different because the answers can be identified easily. Using too much color is confusing and color-blind people might have problems when reviewing the diagram [Lidwell, Holden, and Butler, 2010, pp. 48; Rogers, Sharp, and Preece, 2011, p18].

In section 7.3 on page 68 the function to mark answers as correct during the generation of the question. As soon as RealFeedback provides this functionality, more information can be visualized. The lecturer can get a quick overview whether the students answered the question right or not. Moreover, the lecturer can see how many students answered the question right. In this case, it makes sense to color the bar, which belongs to the right answer differently. As mentioned earlier, it is essential to take care of color-blindness. Color-blind people should be able to distinguish the colors [Lidwell, Holden, and Butler, 2010, pp. 48; Rogers, Sharp, and Preece, 2011, p18]. Figure 7.13 on page 84 shows an example how color-blind people would perceive the visualization. If the correct answer is colored green and the wrong answers are colored red, it would look quite similar to a color-blind person with Deuteranopia color-blindness.

Figure 7.13.: This figure shows how visualization looks like for a Deuteranopia color-blind person. When choosing the color red for visualizing the wrong answer and green for the correct answer (picture A) color-blind people can not clearly recognize the colors as can be seen in picture B.

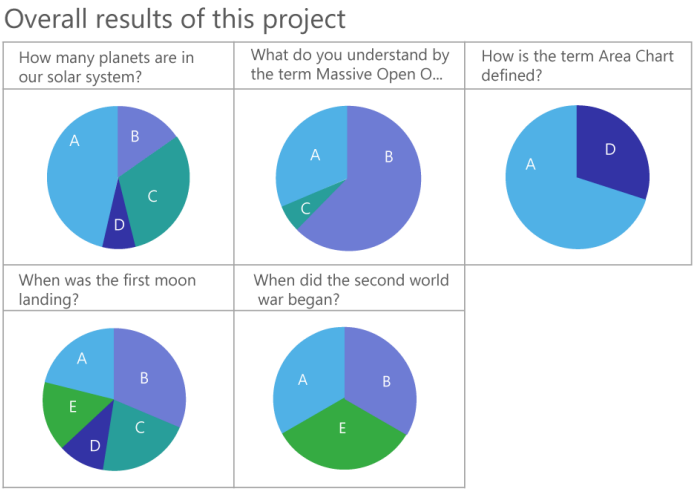

Figure 7.14.: The overall results of a project visualized as pie chart. Different colors must be used for each answer.

Pie chart

Another alternative for visualizing the overall results of a single project is a pie chart. Again, the results can be interpreted as table. Each cell of the table contains the entire information of a question including the results. Figure 7.14 on page 85 shows how the visualization as pie chart looks like. The lecturer gets a quick overview how the votes are distributed between the different answers. This is an advantage of this type of visualization compared to the bar chart in figure 7.12. The whole pie is 100% and the size of the sector which represents an answer shows how much percent voted for the answer. Whereas with the bar chart one cannot see that easy, how many percent voted for one answer by just looking at the bars. One disadvantage over the bar chart is that many different colors are needed to present the results. This adds distraction or makes it harder for color-blind people to see. Since the lecturer can add as many answers to a question as she likes, a pie chart might get complicated or unclear the more answers are added.

When looking at the functionality of marking an answer as correct during

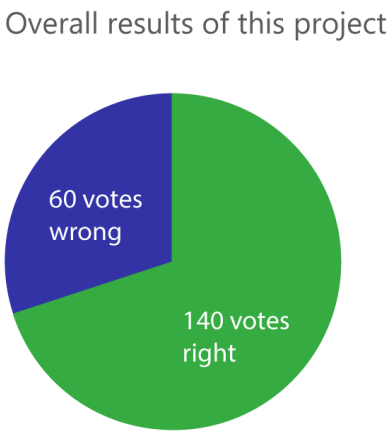

Figure 7.15.: Pie chart showing the overall results of correct answers and wrong answers of a project. The number of correct answers and wrong answers for the entire project is summed up and presented in the pie chart.

the generation of the question it is not easy to present this information in the pie chart. In the bar chart, only one color changes and the user can understand quickly which answer is the correct one. To present this information in the pie chart the correct answer must be marked differently for example, by adding the information besides the pie chart. However, this adds distraction to the chart, and it is harder to identify the correct answer when comparing it to the bar chart.

Another option of visualizing the overall results of a project is shown in figure 7.15 on page 86. This visualization needs the information which answer is correct. All votes of the correct answers and all votes of the wrong answers are summed up. These numbers are visualized as pie chart. Compared to the more detailed pie chart in figure 7.14 much of the information gets lost. The lecturer cannot see how the students voted for single answers or questions. The visualization of right and wrong answers over the whole project as pie chart provides a brief overview of the performance to the lecturer. The advantage for the lecturer is that she can see how the students performed and how well they understand the entire material, which was covered in the lecture. If the lecturer wants to go into more detail, she

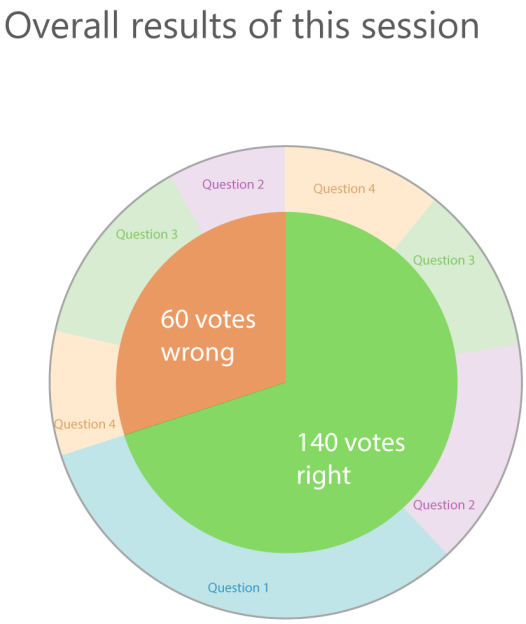

Figure 7.16.: Pie chart showing the overall results of correct answers and wrong answers. It is also shown which questions were answered correct and which questions were answered wrong.

has to analyse the results for each question individually. This feature is not requested by any interview-partner, and none of the ARSs, which were compared provide this functionality. To evaluate if this feature adds value for the lecturer more research on this topic is necessary.

A third approach of presenting all results of a project in a single pie chart is shown in figure 7.16 on page 87. This visualization shows a stacked pie chart. Again all correct votes and all wrong votes are summed up. These two numbers represent the inner circle of the stacked pie chart. The outer ring contains information about the questions. In this example, the lecturer gets an overview how the students (in percent) voted for each question. In the figure 7.16, one can see that all students voted correctly for “Question

1". It can be assumed that half of the student voted "Question 4" right and so on. One weakness of this representation is that many different colors are needed, and the chart cannot be reviewed quickly. Even the comparison of how many students voted correct and how many students voted wrong is hard to see. If the chart is implemented interactive then the lecturer can get some more details out of the results.

Stacked Bar Chart

Visualizing the results as stacked bar chart is another option. The visualization is shown in figure 7.17 on page 89. It can be seen that the results are presented in a compact way. The lecturer can see the distribution of the votes for a single question easily. However, when using a stacked bar chart for visualization, the same things must be considered as with the pie chart. The different stacks of the bar represent one answer. To distinguish the different stacks from each other, it is necessary to color them differently. This adds distraction and complexity to the interface of the lecturer, and it makes it harder to read for color-blind people. If the lecturer uses the feature of marking the correct answer (as soon as this feature is available) when generating a question, the same problems arise as with the pie chart in figure 7.14. Some additional information must be added in order to display which answer is the correct one.

Figure 7.18 on page 90 is another example of visualizing the overall results of a project. This option needs the information which answer is correct. The lecturer must provide this information when generating the question (as soon as the feature is available in RealFeedback). Instead of showing the distribution of the votes over the different answers, the number of right and wrong votes is shown for each question. The advantage of this approach compared to figure 7.17 is that only two colors are needed for the visualization. However, there is also some information lost. The lecturer cannot distinguish the votes for each answer.

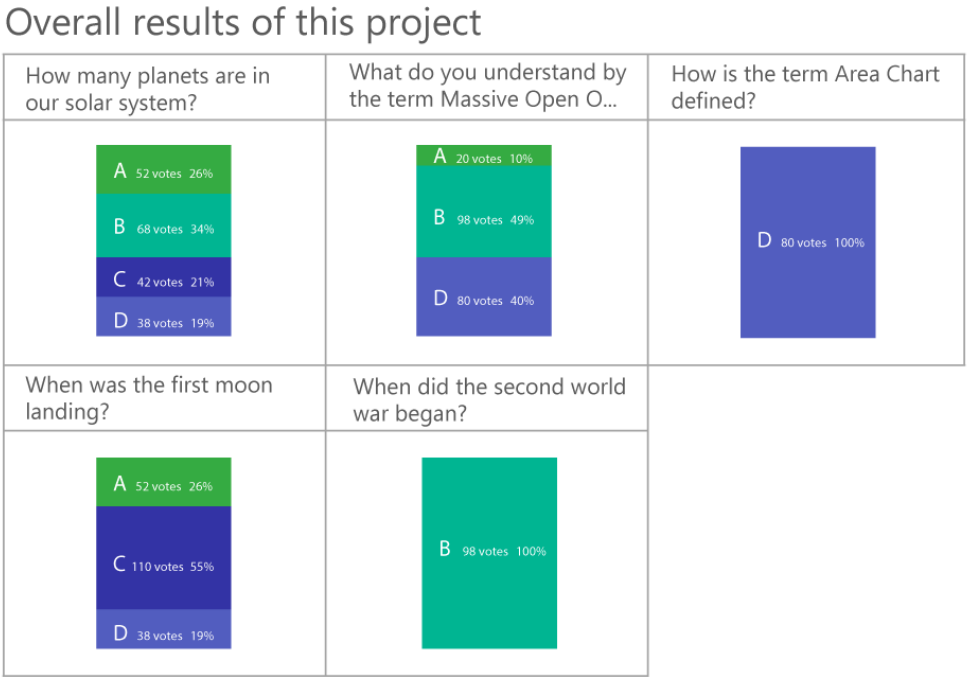

Figure 7.17.: This figure shows the visualization of the results of a project as stacked bar chart. Different colors must be used for visualizing the different answer possibilities.

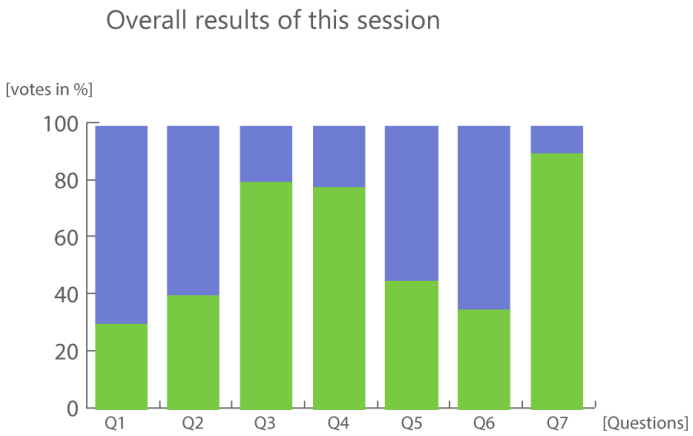

Figure 7.18.: This figure shows the visualization of the results of a project as stacked bar chart. The chart shows the number of correct votes and the number of wrong votes of a question. Each bar represents a question. The green stack represents the correct answers and the blue stack represents the wrong answers.

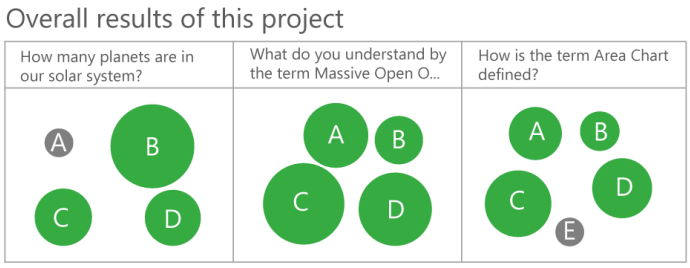

Figure 7.19.: This figure shows the visualization of the results of a project as bubble chart. Each bubble represents an answer. The size of the answer indicates how many students have voted for this answer.

Bubble Chart

The last approach for visualizing the overall results of all questions for a project in a compact way is the bubble chart. Figure 7.19 on page 91 shows the visualization of the results as bubble chart. One bubble represents an answer. The information how many students have voted for each answer is mirrored in the size of each bubble. The bigger the bubble the more votes the answer has got. Each answer has its own bubble. Therefore, one color is enough for the visualization. One downside compared to the bar chart is that one cannot see the different sizes of the bubbles easily. When using the bar chart each bar is below the other bar, which does provide a good overview, and the bars can easily be compared against each other. The bubbles are not located below each other. Therefore, it is much harder to see the difference, especially when the differences in the number of votes are small. This can be seen as a drawback of this visualization.

If a lecturer marks an answer as correct during the generation of the question, the solution for visualizing this information is the same as for the bar chart. The correct answer is colored differently. Again, one must take care of choosing colors that can be distinguished by color-blind people as well. Figure 7.13 shows how a color-blind person would see the difference between green and red.

Summary

The strengths and weaknesses of each representation are discussed in the previous sections. To decide on which visualization to select each representation is compared against each other. Table 7.2 on page 93 gives a quick overview of the results.

The comparison includes only those diagrams, which do not visualize correct answers because this functionality is not provided at the moment by RealFeedback. In general, it can be said that when visualizing the correct answers in the pie chart (figure 7.14) and in the stacked area chart (figure 7.17) additional information must be added which indicates which answer is the correct one. This information cannot visualized via the colors of the different sections because each answer already has a different color.

The comparison does also not include the diagrams, which visualize the sum of correct votes and wrong votes (figure 7.15 and 7.16). These diagrams are not compared because the result of the interview as well as the findings from the comparison of the other ARSs does not mark this kind of diagrams as important. For implementing these visualizations, the functionality of marking answers as correct must be provided by RealFeedback first. For a first implementation, it makes sense to provide a more detailed insight into the results.

Table 7.2 compares the following aspects of the diagrams:

Simplicity This indicates how simple the diagram can be read.

Comparing results The ease of comparing the results against each other. For example, how many students voted for answer A and how many for answer B.

Color-blindness This represents how easy it is for color-blind people to see the results and how much effort is needed to build the diagram in a way that it can be viewed easily by a color-blind person.

These aspects are compared against each other in table 7.2 and the bar chart visualization gets the most points. This also fits to the results of the interview. The interviewees like the visualization of the results as bar chart because it is straightforward and easy to read.

	simplicity	comparing results	color-blindness	**Results**
Bar chart	⋆⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆
Pie chart	⋆⋆	⋆⋆⋆	⋆	⋆⋆
Stacked area chart	⋆⋆	⋆⋆⋆	⋆	⋆⋆
Bubble chart	⋆⋆	⋆	⋆⋆⋆	⋆⋆

Table 7.2.: This table shows the comparison of the different chart types. The diagrams are rated regarding its simplicity to read, the simplicity of comparing results and factors regarding color-blindness.

The bar chart is the simplest chart to read compared to the others. Not much distraction is added because only one color is needed for the representation of the results. The pie chart and the stacked are chart add some distraction due to the different colors which must be used in order to represent the different answers. The bubble chart makes it hard to see the relations between the votes for each answer because the circles are not in a row and they have each a different size. Small differences are hard to see.

When comparing the different number of votes of each answer the best overview is provided with the pie chart or the stacked are chart. With this type of representation, one can easily see how many votes out of all votes belong to one single answer. With a bar or a bubble chart, it is not that easy to see distribution of the votes that easily and quickly.

Color-blind people might have a problem with the pie or a stacked are chart because for this visualization different colors must be used. The effort for developing a diagram, which is easy to read, by color-blind people is high compared to the bubble or the pie chart.

7.4.4. Visualizations of Different Sessions of a Project

The results of the interview show that it is essential to the lecturers to reuse the questions in future lectures. RealFeedback does offer the feature of reusing question in general. However, as soon as a question is reused

the data of the previous lecture must be cleared and gets lost. The lecturers mentioned this as a disadvantage. They would like to review the results of previous lectures and compare the different results. The comparison of the different results is an indicator for the lecturers if the knowledge of students increase. This features are very often premium features of other ARSs.

These facts let assume that reusing questions and keeping the data of previous sessions is an essential feature that should be implemented. Therefore, sessions must be introduced to RealFeedback. A session is created as soon as the lecturer starts one question and stops when the last question of the project is stopped. As soon as the lecturer wants to reuse the questions a new session is created, and the previous data is stored. Each project has one or more sessions, and in each session the results of a previous run of the project is stored. Introducing this structure means that the lecturer has the option to compare different sessions.

Because sessions are a new construct in RealFeedback they must be included and integrated into the interface of the lecturer. The results of the interviews point out that it is useful to compare sessions against each other. Therefore, this section covers the different visualization possibilities of comparing the sessions against each other. The integration of this feature into the interface of the lecturer could be done by introducing a button. How this looks like is shown in figure 7.10 and 7.11. In the first iteration of this feature, all sessions are compared against each other. The lecturer cannot select single sessions, which she would like to compare.

Visualizing different sessions is located on the top-most level of the hierarchy shown in figure 7.1 and it is a complex visualization because a lot of information is involved. This information must be visualized in a single diagram where the individual sessions can be compared against each other.

During the research of the different ARSs and the interviews some theoretical constructs emerged for visualizing sessions. The most powerful constructs are described and discussed in the next sections.

Visualizing Sessions in a Bar Chart

The different sessions of one project can be visualized as bar chart. The chart consists of groups of bars. Each bar represents a year, and each group of bars represents one question. The visualization of this bar chart is shown in 7.20 on page 96. The y-axis represents the number of correct votes. Therefore, the height of each bar represents the correctness of the votes for one question. To implement this kind of visualization the information is needed, which answer is correct. Without this information, the correctness of the question cannot be visualized. In this bar chart, the lecturer can see quickly how the students performed over different sessions.

Because only the correctness of the question is represented, some information gets lost. The lecturer cannot see how the votes rearranged between the wrong answers. This kind of visualization requires that the lecturer always provides the information which answer is the correct one when she generates a question. The feature of marking answers as correct is not implemented in RealFeedback now.

Visualizing Sessions in a Stacked Bar Chart

The previous section discussed the visualization of the comparison as bar chart where the y-axis represents the number of correct votes. With this visualization, some information is lost for example the distribution of the votes over the answers for a question. Therefore, another approach evolved which represents the results in more detail. Figure 7.21 on page 97 shows an alternative to the bar chart, which was discussed earlier in figure 7.20. This visualization uses all the data, which can be retrieved for one question and represents it. The y-axis does not show the correctness of the answer as described in the bar chart visualization. Instead, it shows the percentage of votes distributed over the different answers. One question consists of a group of bars, and each bar of this group represents a different year. Every bar is split into the number of answers that got votes. This visualization offers the lecturer the information how many students voted for each answer. The distribution of the answer can be seen easily. The lecturer can also see how the votes rearranged between the answers over the sessions.

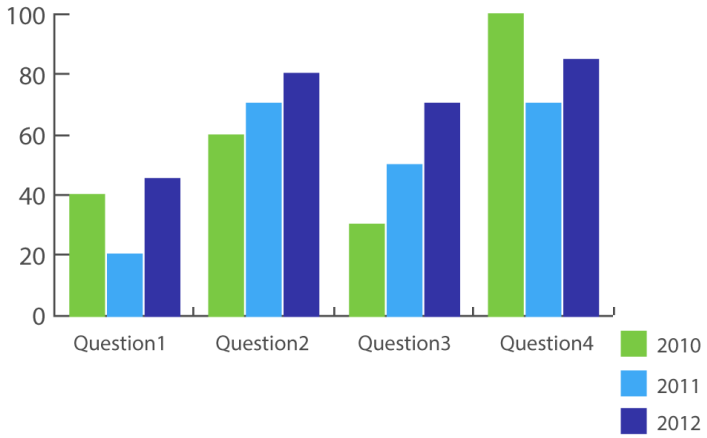

Figure 7.20.: This figure shows the visualization of different sessions in a bar chart. Groups of bars showing the changes of the answers for one question over time. Each group of bars belongs to one question, and each bar belongs to one year. The y-axis represents the number of correct votes.

The order of the answers on each stack stays always the same. Otherwise, it would be difficult for the lecturer to compare the results. This information is useful to the lecturer. The lecturer immediately sees how the change of the lecturing style affects the performance of the students.

Visualizing Sessions in a Stacked Pie Chart

Another approach for visualizing a comparison of the different sessions of a project is a stacked pie chart. Figure 7.22 on page 99 shows a stacked pie chart which presents different sessions. The stacked pie chart consists of rings. Each ring of the pie chart represents a year. The inner ring represents the oldest session, and the outer ring represents the newest session. The pie is split up into main sections. Each main section of the pie represents a question. A pie does have so many main sections as different questions are available in the project. This sections are split up again into small sections, which are called “answer-sections” in this work. The size of each “answer-

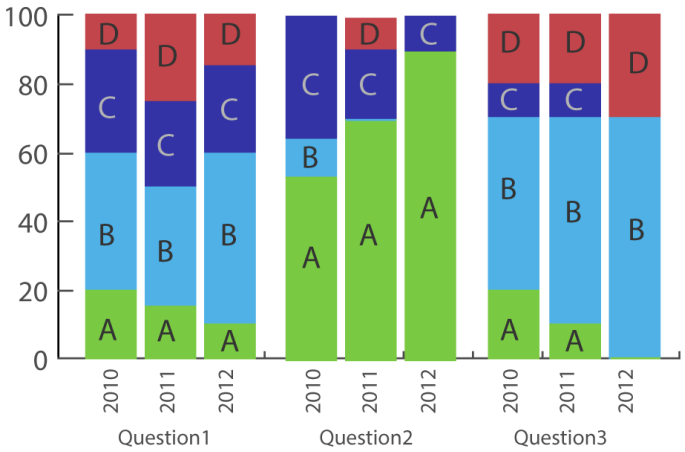

Figure 7.21.: This figure shows the visualization of different sessions of a project as stacked bar chart. Groups of bars showing the changes of the answers for one question over time. Each group of bars belong to one question, and each bar belongs to one year. The rearrangement of votes between the answers can be seen easily in this visualization.

section" represents how many students voted for each answer. Comparing the results of the sessions against each other is difficult to achieve with this kind of visualization. The reason therefore is that the inner ring has much smaller sections than the outer ring. Therefore, the sections, which are located in the inner ring, are much smaller than the sections of the outer ring. Because of this fact, the sizes of the different sections do not allow the lecturer to compare the different results simply against each other.

Another approach of comparing the different sessions against each other as stacked pie chart is shown in figure 7.23 on page 100. In this visualization each ring represents one question and the main sections of a pie represent the years now. This has the effect that the questions of the sessions are comparable against each other because the results for one question over more sessions (years) are located in the same ring. This makes it easier to compare the results of a question against each other because the sizes of the sections are in relation to each other. If the inner ring would represent "Question 3" and the results over the years are all in the same ring, the answers - which are split into sections - are comparable because the ratio is correct. However, comparing the different results for one question is difficult either because the lecturer must compare the different sections of one ring that are not beside each other. Otherwise, the lecturer cannot analyse the answers and she cannot see how the votes rearranged from one answer to another.

Visualizing Sessions in a Bubble Chart

A bubble chart can also be used for visualizing the results of different sessions. The visualization is shown in figure 7.24 on page 101. In this visualization, a group of bubbles represents a question. Each session is represented as row of vertical bubbles. Each horizontal row of bubbles represents an answer. In this visualization, the answers can be distinguished by the different colors as well. Using different colors is not essential for this kind of visualization because the answers can be distinguished by the position in the chart. The size of the bubble depends on the number of votes each answer has. If there are only minor differences in the number of votes, the sizes of the bubbles cannot be distinguished easily.

Figure 7.22.: This figure shows the visualization of different sessions of a project as stacked pie chart. Each ring represents a year, and each section of the pie chart represents one question. This sections are split into the answers. The size of the sections within the "question-section" represents the number of votes for an answer.

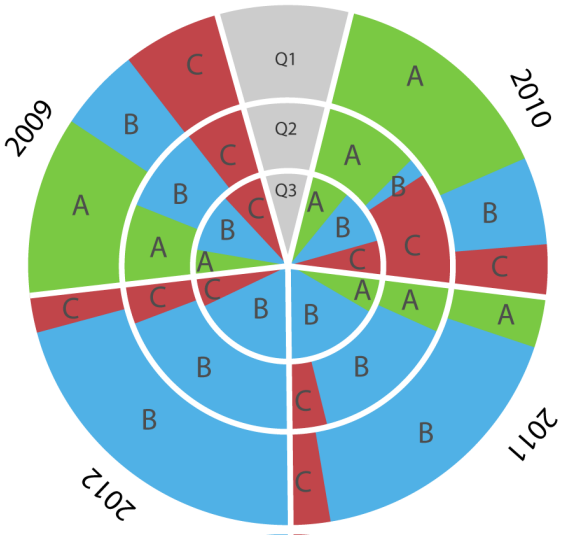

Figure 7.23.: This figure shows the visualization of different sessions of a project as stacked pie chart. Each ring represents a question, and each main section represents a year. This sections are split into the answers. The size of the sections within the "year-section" represents the number of votes for an answer.

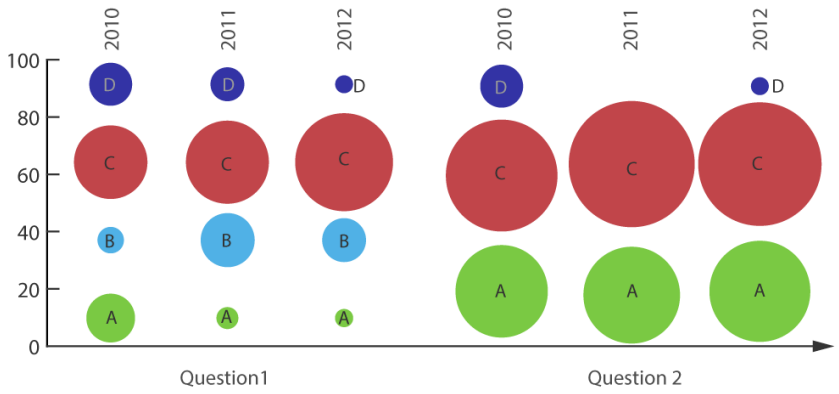

Figure 7.24.: In this figure, a bubble chart is used for presenting the results of different sessions of a project. Each group of bubbles belong to a question, each vertical row of bubbles belong to a session. Each bubble represents an answer.

Summary

Not all visualizations can be included into RealFeedback. Therefore, the different visualization types are analysed and compared against each other, and the best visualization is evaluated. The different visualization methods are compared according to the following criteria:

Simplicity The simplicity of reading the diagram.
Comparing results The ease of comparing results of different sessions against each other.

The factor of color-blindness was not taken into account for this comparison. The reason therefore is that all visualizations, which were mentioned in this section, need different colors for visualizing the results. However, such colors are chosen which can be distinguished by color-blind people.

Table 7.3 on page 102 shows the results of the comparison.

The analysis of the diagrams show that a bar chart and the stacked bar chart are the simplest charts to read compared to the other visualizations.

	Simplicity	Comparing results	**Results**
Bar chart	⋆⋆⋆	⋆	⋆⋆
Stacked bar chart	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆
Stacked pie chart	⋆	⋆	⋆
Bubble chart	⋆⋆	⋆⋆	⋆⋆

Table 7.3.: This table shows the comparison of the different chart types, which visualize the different sessions of a project. The diagrams are rated regarding its simplicity to read and the simplicity of comparing results. The stacked bar chart is the chart which is easy to read and easy to compare.

The stacked pie chart as well as the bubble chart is more difficult to read. The most difficult chart to read is the stacked pie chart. The reasons for the bad readability of the chart is described in section 7.4.4 on page 96.

For comparing the results between the different sessions the stacked bar chart is the most easiest to read chart. The reason therefore is that one can easily see the differences in the number of votes for each question. The bar chart gives no information about how many votes each answer got because it only shows the correct votes. Comparing the results in the stacked pie chart is difficult because the relations between the different answers cannot be seen clearly. The lecturer must search the correlating sectors in order to get an overview of the different results of a section. When comparing the bubble chart to the stacked bar chart the stacked bar chart is easier to read than the bubble chart. The relations of the overall votes to the number of votes for each answer can be easier.

According to the evaluation the stacked bar chart is the most easy to read and most easy to compare chart. This chart type is used for further analysis in this work.

7.5. Results

The results and findings of this section are summarized. First the results are presented together with the findings of the comparison of the different ARSs in chapter 5 and the results of the interview in chapter 6. Afterwards, the main points, which should be implemented in RealFeedback, are discussed.

The solutions to the single features which are analysed during the comparison of the ARSs and the findings of the interview are shown in table 7.4 on page 104. For each feature that is part of this work, a solution is provided. The following sections describe the different solutions. One of these solutions is chosen for implementation in RealFeedback.

7.5.1. Introducing Sessions and Visualization of the Sessions

The interview and the comparison of the different ARSs show that it is essential for the lecturer not to lose data of previous sessions. This feature builds a foundation for a variety of features that can be added to RealFeedback. Therefore, this feature is considered as the most notable feature to add. The hierarchy shown in figure 7.1 explains how the concept of a session fits into RealFeedback. A project has zero or more sessions. Each session stores data of a previous lecture or run.

The most important points of this feature are

- The lecturer does not lose the data of previous sessions.
- Questions can be reused easily.
- The sessions can be analysed and reviewed by the lecturer if an appropriate visualization is provided.

To evaluate and compare the different sessions it is essential to provide a suitable visualization. The different diagram types for visualizing sessions of a project are compared in table 7.3 on page 102 against each other. The results of this table show, that the best way to go is the stacked bar chart shown in figure 7.21 on page 97. It is easy to read and the results can be compared quickly.

Weaknesses of RealFeedback based on the comparison	Results Interview	Solution
Keeping Data of Previous Sessions is analysed as missing	Comparison of the results over different sessions is missing	Solution provided in section 7.4.4 on page 93
Mark Answers as Correct is a weakness of RealFeedback	Defining which answer is the correct one is missing	Solution provided in section 7.3 on page 68.
Visualization of Results and Statistics is a weakness of RealFeedback	A visualization of how many students have already voted for the question is missing	Solution provided in section 7.4.2 on page 79
Visualization of Results and Statistics is a weakness of RealFeedback	Visualizing how fast the students responded to a question is missing	Solution provided in section 7.4.1 on page 72 and section 7.4.2 on page 74
Response Possibilities	A mobile application is missing	Not a topic of this work
	A function for taking notes for a question is missing	Not a topic of this work

Table 7.4.: The findings of the previous sections are listed again in this table. For each problem, which is relevant for this work a solution is provided.

Because of the high impact on the system, this feature is added to RealFeedback in this thesis.

7.5.2. Mark Answers as Correct

It is already discussed in section 7.3 that this is a valuable feature. Students can use this information for exam-preparation. For the lecturer, the data presentation can change. Instead of comparing votes for each answer without knowing immediately which answer is the correct one, the correct answer can be easily visualized. Compared to the feature of introducing sessions this one does not have such a high impact on the system.

7.5.3. Visualize the Amount of Students who have Voted

Another feature which is identified as valuable, is the visualization of how many students already have voted for an answer. This information can be used as indicator how long a question should be open for voting and when the lecturer can close the question. How the visualization could look like is shown in figure 7.9 on page 80. The results of the comparison of the different ARSs and the interviews show, that this feature is not as valuable as introducing sessions into RealFeedback.

7.5.4. Visualize the Voting Time

This feature is described in section 7.4.1 on page 72 and in section 7.4.2 on page 74. None of the ARSs which are compared against each other in chapter 5 provide this feature. However, one of the interview-partners mentioned that this is a valuable feature for her. Therefore, it is necessary to do further research on this feature. According to the results of the previous research, this feature is not as important as introducing sessions.

8. Implementation

This section describes the implementation of the features which are chosen for implementation in chapter 7. The implementation details for accomplishing these features are discussed in this chapter as well as the architecture of the design after the features are added. According to the analysis of the interview, the comparison of the different ARSs and the discussion of the different visualization types the following features are added to RealFeedback:

- Add the concept of sessions to RealFeedback
- Visualize the results of different sessions for one project

Before the implementation is explained the frameworks of RealFeedback which are already in use are described. Afterwards, the implementation details are discussed.

8.1. Frameworks Used in RealFeedback

This section describes the technology and the frameworks that were already used to create RealFeedback. It is essential to analyse, which frameworks are in use because the implementation of the new features must correlate with these frameworks. The frameworks can be distinguished into server-side frameworks and client-side frameworks.

8.1.1. Server-Side Frameworks

The server is implemented in python [*Python Programming Language*]. There are frameworks for building a server in python.

Pyramid: For the implementation of RealFeedback Pyramid is used as server-side framework. Pyramid is defined as follows: "Pyramid is a small, fast, down-to-earth Python web application development framework. It is developed as part of the Pylons Project. It is licensed under a BSD-like license." [*Pyramid*]. Pyramid builds the foundation for the server-side implementation of RealFeedback. The functionality, which is added for implementing this feature, must integrate with Pyramid.

MongoDB: The database of RealFeedback is implemented as *NoSQL Database*. The framework which is used for the implementation of the NoSQL database is MongoDB. MongoDB is described as follows "MongoDB (from "humongous") is an open source document database, and the leading NoSQL database. Written in C++" [*MongoDB*].

8.1.2. Client-Side Frameworks

The client-side is written in *HTML5* and *JavaScript*. The main frameworks, which are used for the creation of the user-interface of RealFeedback are described in the following paragraphs.

jQuery: "jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers." [*jQuery*].

Backbone.js: "Backbone.js gives structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions, views with declarative event handling, and connects it all to your existing API over a RESTful JSON interface." [*Backbone.js*].

Bootstrap: The web-site is based on responsive design "A website that responds to the device that accesses it and delivers the appropriate output for it uses responsive design. Rather than designing multiple sites for different-sized devices, this approach designs one site but specifies how it should appear on varied devices." [*Responsive Web Design*]. The framework, which is used to implement the responsive web design behaviour is *Bootstrap*.

8.2. Implementing Sessions

RealFeedback does not support the feature of sessions at the moment. When the lecturer wants to reuse the questions, in later runs the old data is lost. Therefore, the concept of sessions is introduced into RealFeedback. Each run of a project (asking questions) is defined as a session. A session holds all the data which is collected during a run. The user has to define explicitly when a new session starts by clicking on a button in the lecturers interface. The lecturer should have the possibility to compare the different sessions against each other.

The following steps are necessary to integrate sessions into the existing system:

1. Add sessions to the database structure.
2. Provide the functionality of creating a session on the server.
3. Update the Backbone.js model on the client-side.
4. Provide a RESTful HTTP call, which delivers session-data from the server.
5. Integrating sessions into the lecturers interface.

8.2.1. Add Sessions to the Database Structure

At the moment, each project consists of questions that belong to the project. Every question consists of answers. To add sessions into the system, the database structure must be changed, because there is nothing foreseen for saving sessions. The data structure of the whole project can be seen

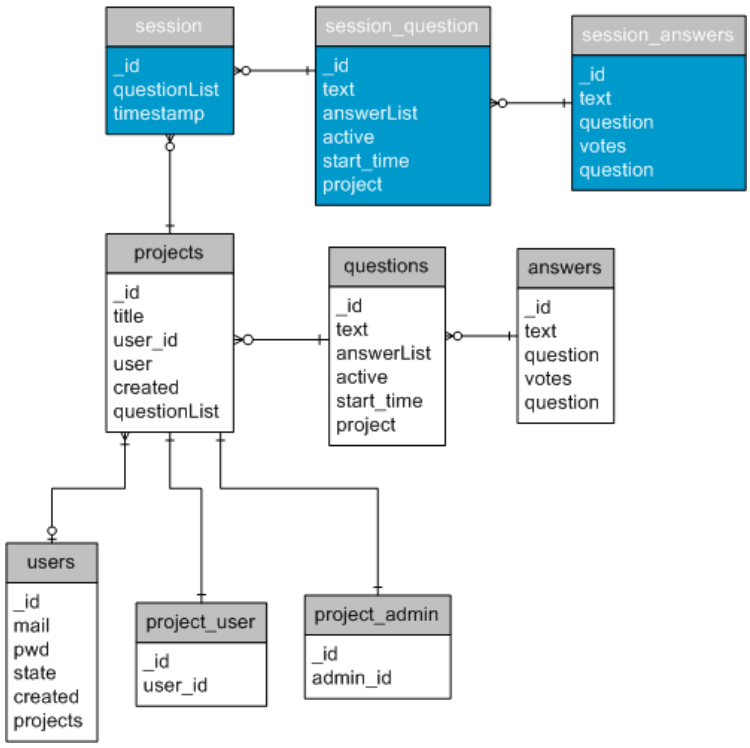

Figure 8.1.: A session consists of questions and questions consist of answers. The data of the current run is moved to a new session, as soon as a new session is created by the user. The questions and answers, which are connected to the project, are being cleared.

in figure 4.6 on page 23. To add sessions to project the following new collections are added to the database: sessions, session_question and session_answer. Each session consists of one or more session_question. Each session_question consists of one or more session_answers. How these collections are integrated in the existing database structure, is shown in figure 8.1 on page 110. Every project has a list of session-IDs. With this ID, the session is linked to the project. Besides the list of questions, each project holds a list of sessions. Each session holds the data of prior runs. The newest or the actual session is represented by the questions that are linked directly to the project.

8.2.2. Creating a Session on the Server

The server must provide the functionality of creating a session. At the client-side a new button is introduced ("Create New Session"). On the sever a new RESTful HTTP call is provided for creating a new session for the project.

`realfeedback.tugraz.at/v1/project/{project_id}/createsession`

When the user clicks on the button, the RESTful HTTP call is called, whereas the project_id must be provided. On the server, a new session is created and the data of the questions that belong to the project are moved into this new session. Then the questions and answers for the project are cleared. The texts of the question and answers as well as the information which answers belong to the questions are kept. It is crucial that the only one who can create a new session is the user to whom the project belongs. Before the session is created, the authentication of the user is checked. The model must be updated on the client-side as well. At first, the values are cleared manually on the client. And after the RESTful HTTP call has finished the new data is fetched again from the server.

8.2.3. Fetching Sessions from the Server

As soon as the lecturer wants to review or compare the previous sessions of a project the data must be fetched from the server. The server must provide a function for fetching the sessions from the server. Therefore, a new RESTful HTTP call is introduced.

`realfeedback.tugraz.at/v1/project/{project_id}/sessions`

In order to retrieve the data from the server the project_id must be provided to the RESTful HTTP call. The call returns a *JSON* result. The JSON result of the REST call is shown in listing 8.1 on page 112. Each session that was ever created for this project is delivered by this RESTful HTTP call. The oldest session is the first element in the list and the newest session is the last element in the list.

Listing 8.1: Listing of the JSON result which is returned by the server when the RESTful HTTP call for fetching sessions of a project is called.

```
[
    {
        questionList:[
            "bb57976e-f379-44b2-b0a3-95fdd0b03531"
        ],
        timestamp:1365247410,
        _id:"1db21a4e-6e3a-4e2c-b26e-4d601ac08bb8",
        questions:[
            {
                answerList:[
                    "9db39d47-3459-424f-95ff-9c68180db7c0",
                    "567324a7-0cff-4b97-8f80-d55bf55f575a"
                ],
                text:"Question 1",
                start_time:7,
                answers:[
                    {
                        text:"Answer 1",
                        votes:0,
                        _id:"9db39d47-3459-424f-95ff-9c68180db7c0",
                        question:"bb57976e-f379-44b2-b0a3-95fdd0b03531"
                    },
                    {
                        text:"Answer 2",
                        votes:1,
                        _id:"567324a7-0cff-4b97-8f80-d55bf55f575a",
                        question:"bb57976e-f379-44b2-b0a3-95fdd0b03531"
                    }
                ],
                project:"f6e87179-82f5-4f86-987c-7b7a134d3965",
                active:false,
                _id:"bb57976e-f379-44b2-b0a3-95fdd0b03531"
            }
        ]
    },
    {
        questionList:[
            "696e226a-5aed-4d62-b59b-98bbffc5a865"
        ],
        timestamp:1365247535,
        _id:"af27a0f5-b200-421a-bf4b-9bf5ffb93cbf",
        questions:[
            {
                answerList:[
                    "b61b50e5-1059-441c-84a8-e8df44bb9341",
```

```
                "7965658c-9a10-4b72-9137-ac0173b4ff32"
            ],
            text:"Question 1",
            start_time:13,
            answers:[
                {
                    text:"Answer 1",
                    votes:1,
                    _id:"b61b50e5-1059-441c-84a8-e8df44bb9341",
                    question:"696e226a-5aed-4d62-b59b-98bbffc5a865"
                },
                {
                    text:"Answer 2",
                    votes:0,
                    _id:"7965658c-9a10-4b72-9137-ac0173b4ff32",
                    question:"696e226a-5aed-4d62-b59b-98bbffc5a865"
                }
            ],
            project:"f6e87179-82f5-4f86-987c-7b7a134d3965",
            active:false,
            _id:"696e226a-5aed-4d62-b59b-98bbffc5a865"
        }
    ]
}
]
```


8.2.4. Integrating Sessions Into the Lecturers Interface

After the information is fetched from the server, it must be integrated into the lecturers interface. The feature for showing old sessions is included in the projects detail view and in the list of projects where all projects can be seen. When the lecturer clicks on the magnifying glass, the window for viewing the session data appears as overlay.

For the visualization of the diagram, different tools are analysed and compared. The main comparing criteria were the different chart types that can be realized with the framework, how they are included into HTML (canvas, svg) and how active the framework is in development. It was tried to choose a framework, which can be reused for other types of visualization in RealFeedback and with a large community and support. In total 36 different JavaScript, frameworks for visualizing data in are evaluated. The entire

Tool	Website	HTML integration
D3.js	d3js.org/	svg
flotr2	humblesoftware.com/flotr2/index	canvas
Google Chart Tools	developers.google.com/chart/	svg
JavaScript InfoVis Toolkit	philogb.github.io/jit/	canvas
jqPlot	www.jqplot.com/	canvas

Table 8.1.: The five JavaScript visualization libraries which are chosen for a deeper comparison.

list of the different frameworks can be found in appendix A. The five most promising frameworks are chosen and compared. The frameworks, which are chosen for the comparison, can be seen in table 8.1 on page 114.

Chart Types: All of these tools support the most common chart types:

- bar chart
- pie chart
- stacked bar chart
- area chart
- line chart

D3.js is the most flexible tool. Compared to the other tools many different chart types can be drawn with this library. It is a tool, which is in remarkably active development and the community around D3.js is extremely large, supporting each other. With D3.js, some basic chart types are supported but also individual charts and visualizations can be generated. The other tools do also support different chart types, but they are not as flexible to use as D3.js.

Integration Into HTML: The integration into HTML is also an essential point. Two ways of integrating diagrams into HTML are common. One approach is the integration via the HTML svg-tag and another approach is the integration via the HTML canvas-tag. The canvas-tag is supported by more mobile-browsers compared to the svg-tag. The svg-tag cannot be used, for

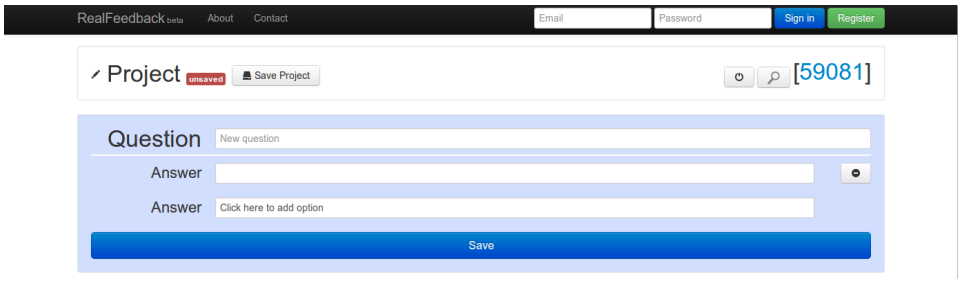

Figure 8.2.: This figure shows the integration of the visualization into the lecturers interface. When the lecturer clicks on the magnifying class the visualization appears.

example with Android versions smaller than 3.0 whereas the canvas-tag is supported by all mobile browser versions.

One considerable drawback of the canvas-tag is that charts and diagrams do not provide interactive content to the user. The svg-tag provides interactivity in diagrams.

Choosing a Framework: For the implementation of the diagrams in RealFeedback, the framework D3.js is chosen. It is the most flexible library. The fact that it is not supported on old Android devices is not taken as a serious problem. Old versions of Android disappear slowly from the market.

Integrating the comparison into the lecturers interface: The integration of this functionality was realized with a button. As soon as the lecturer clicks this button the session-data is fetched from the server. A dialogue opens up which visualizes the results of the different sessions. The values for displaying the sessions are normalized. When the values are normalized they can be compared better against each other.

Figure 8.3.: This figure shows the visualization of the sessions in RealFeedback after the implementation.

9. Conclusion

The focus of this work is how to improve the statistics and reports of *RealFeedback* - a web-based ARS - for the lecturers. Twelve web-based ARSs are compared against each other. The major-findings of this comparison are listed below.

- Downloadable reports are a common feature. When the data is provided as downloadable CSV file, the lecturer can do analysis on his own.
- Defining which answer is the correct one. Some of the compared ARSs offer the feature of defining which answer is the correct one during question generation.
- Reviewing and comparing results of previous lectures is a feature, which is often only available for premium members who pay for the service. Therefore, it was assumed that this is a valuable feature.

Four lecturers who use RealFeedback during their lecture and who are experienced with this tool were interviewed. Due to the analysis of the interviews, the following functions and features are currently missing in RealFeedback:

- The feature of comparing the results from previous lectures against each other was mentioned as an important feature.
- The lecturer cannot define which answer is the correct one. Some of the interviewees mentioned this as missing feature.
- For a better estimation, when to stop a question the interviewees would like to have a diagram, which shows how many of the registered students already have voted for an answer.
- One interviewee mentioned that it would be an interesting feature when the time of how fast the students responded to an answer is tracked and visualized.

The findings of the comparison and the interviews were correlated to each other and solutions for different visualization types emerged. Each solution was discussed and the most relevant solutions were summarized:

- Introducing the concept of sessions is crucial to compare the results of previous sessions (lectures) against each other.
- Marking answers as correct was also mentioned as an important feature.
- Visualizing the amount of students who already have voted for an answer.

In this work, sessions were introduced and implemented in RealFeedback. As soon as a new lecture starts, the lecturer can define a new session. The comparison of the sessions against each other was also implemented into RealFeedback. A button was integrated which visualizes the results of the sessions over time.

10. Future Work

During the research in this work, new topics were discovered. Some types of visualizations, which are presented in chapter 7 on page 65, are important for future work. In this work, there was no finding which could prove that those features and visualizations are important. However, further research can prove whether the visualizations are relevant or not.

During the implementation of sessions in *RealFeedback* some questions came up which are also relevant to cover in future work. The topics are mentioned below.

How should RealFeedback react on changes of questions or answers when comparing the sessions against each other. RealFeedback offers the function of changing the text of questions or answers. One critical situation, which must be addressed, is what happens when a question or an answer changes over time. The comparison of the session with the new question or answer must be changed because the data differs. The system must provide a way to identify which answer has changed. If there are only small changes which do not affect the semantic of the question, the questions or answers can be compared independent of the change. However, for this solution different learning algorithms and semantic analyses are necessary.

The function of clearing results of a single question. Until now, there was no functionality for clearing the data of the whole project. Only single questions could be cleared. During the implementation of the new feature, the possibility of clearing a single question was kept. The difference between clearing a question and clearing a project is that when clearing a project a new session is generated whereas clearing a single question only

deletes the data of the specific question. The old data of this question is totally lost. Due to the analysis of the system we suggest doing a research on the feature of clearing a single question. For later implementations, one must think of whether the feature of clearing a single question is necessary or not.

Interactivity of visualizations None of the compared ARSs provided interactivity of visualizations. Today's technology makes it easy to create charts or visualizations in an interactive way. The big advantage of interactivity is that more information can be added to a chart. However, it is not visible until the user interacts with the chart. When not interacting with the chart the basic information is shown anyway. As soon as the user starts interacting, she can get more detailed information about the results. Implementing interactive charts is also a topic which should be addressed in future work.

Appendix

Appendix A.

JavaScript Visualization Frameworks

JavaScript libraries for data visualization. The libraries are discussed in chapter 8 on page 107. Table A.1 on page 124 lists the tools.

Tool	Website
D3.js	d3js.org
Flotr2	humblesoftware.com/flotr2/index
Google Chart Tools	developers.google.com/chart
Javascript Infovis	philogb.github.io/jit
jqPlot	www.jqplot.com
amChart	amcharts.com
arcardiacharts	www.arcadiacharts.com
awesomecharts	cyberpython.github.io/AwesomeChartJS/
canvasXpress	canvasxpress.org
Chart.js	chartjs.org
dhmtlx	dhmtlx.com
dygraphs	dygraphs.com
ejschart	ejschrat.com
dojo	dojotoolkit.org
elycharts	elycharts.com
Flot	flotcharts.org
Envision.js	www.humblesoftware.com/envision/index
fusioncharts	fusioncharts.com
gRaphael	g.raphaeljs.com/
Highcharts	www.highcharts.com/products/highcharts
jqChart	www.jqchart.com/jquery/chart
jsPlumb	jsplumbtoolkit.com/jquery/demo.html
JSXGraph	jsxgraph.uni-bayreuth.de/wp/examples/
KendoUI DataViz	demos.kendoui.com/dataviz/overview/index.html
MilkChart	mootools.net/forge/p/milkchart
Morris.js	www.oesmith.co.uk/morris.js/
nvd3	nvd3.com
Protovis	mbostock.github.io/protovis/ex/
RGraph	rgraph.net
Rickshaw	code.shutterstock.com/rickshaw/examples/
Sencha Touch Charts	http://www.sencha.com/products/complete
TeeChart	www.steema.com/files/public/teechart/html5/-jscript/demos/
Wijmo	wijmo.com
Zingchart	zingchart.com

Table A.1.: List of javascript visualization frameworks.

Acronyms

ARS Audience Response System. v, vii, xvii, 1, 2, 5–8, 10, 12, 15, 29, 30, 32–35, 37, 40, 42, 44, 45, 49, 51–55, 57, 59, 65, 66, 68, 74–76, 78, 79, 87, 92, 94, 103, 105, 107, 117, 120

CCS Classroom Communication System. 1
CSV comma-separated values. 15, 32, 34, 35, 51, 61, 80, 117

HTTP Hypertext Transfer Protocol. 25–27, 109, 111

PRS Personal Response System. 1

QDI Question Driven Instruction. 8, 10, 12, 13

SMS Short Message Service. 6, 33, 37, 38, 43, 45, 53
SRS Student Response System. 1

TEFA Technology-Enhanced formative assessment. 12–14

WILD Wireless Internet Learning Device. 6

Bibliography

Abrahamson, Louis (1999). *Teaching with a Classroom Communication System - What it Involves and Why it Works* (cit. on p. 53).

Abrahamson, Louis (Feb. 2006). "A Brief History of Networked Classrooms: Effects, Cases, Pedagogy, and Implications." English. In: *Audience Response Systems in Higher Education*. Ed. by David Banks. IGI Global. ISBN: 9781591409472. DOI: 10.4018/978-1-59140-947-2 (cit. on p. 5).

Backbone.js. [Online. Accessed 14-April-2013]. URL: http://backbonejs.org/ (cit. on p. 108).

Beatty, Ian D. (2005). "Transforming student learning with classroom communication systems." In: *Educause Center for Applied Research (ECAR) Research Bulletin ERB0403* (cit. on pp. 6, 7).

Beatty, Ian D. and William J. Gerace (Jan. 2009). "Technology-Enhanced Formative Assessment: A Research-Based Pedagogy for Teaching Science with Classroom Response Technology." In: *Journal of Science Education and Technology* 18.2, pp. 146–162. ISSN: 1059-0145. DOI: 10.1007/s10956-008-9140-4 (cit. on pp. 1, 7, 12–14, 53, 66).

Beatty, Ian D., William J. Leonard, et al. (2006). "Question Driven Instruction: Teaching science (well) with an audience response system." In: *Audience Response Systems in Higher Education Applications and Cases*. Ed. by David A Banks. Idea Group Inc. Chap. 7, pp. 96–115 (cit. on pp. 7, 8, 10, 76).

Better Education Inc. [Online. Accessed 14-April-2013]. URL: http://www.bedu.com/ (cit. on p. 5).

Better Education Inc (1996). *Classtalk® 2.0 - the Calculator Version*. Tech. rep. (cit. on p. 5).

Boardman, Dorris E. (1968). "The use of immediate response systems in junior college." PhD thesis. University of California (cit. on p. 5).

Bonwell, Charles C. and James A. Eison (1991). "Active Learning : Creating Excitement in the Classroom ." In: *J-B ASHE Higher Education Report Series (AEHE)* (cit. on p. 7).

Bootstrap. [Online. Accessed 14-April-2013]. URL: http://twitter.github.com/bootstrap/ (cit. on p. 109).

Bransford, John D., Ann L. Brown, and Rodney R. Cocking (2000). *How People Learn: Brain, Mind, Experience, and School*. Ed. by John D Bransford, Ann L Brown, and Rodney R Cocking. Vol. Expanded E. 4. National Academy Press. Chap. 319, x, 374 p. ISBN: 0309070368. DOI: 10.1016/0885-2014(91)90049-J (cit. on pp. 7, 14).

Bruff, Derek (2009). *Teaching with Classroom Response Systems: Creating Active Learning Environments*. Vol. 33. Jossey-Bass higher and adult education series. Jossey-Bass, p. 240. ISBN: 0470288930 (cit. on pp. 7, 20, 49, 53).

Caldwell, Jane E. (2007). "Clickers in the Large Classroom : Current Research and Best-Practice Tips." In: *CBE Life Sci Educ. 2007*, pp. 9–20. DOI: 10.1187/cbe.06 (cit. on pp. 1, 6).

Captcha Definition. [Online. Accessed 14-April-2013]. URL: http://www.techterms.com/definition/captcha (cit. on p. 20).

Clicker School. [Online. Accessed 14-April-2013]. URL: http://www.clickerschool.com/Pages/SignedOutHome.aspx (cit. on pp. 30, 35).

Collins, Bobbie L., Rosalind Tedford, and H. David Womack (2008). "Debating the merits of clickers." In: *North Carolina Libraries (Online)* 66.1/2, p. 20 (cit. on p. 53).

Cue, Nelson (1998). "A Universal Learning Tool for Classrooms ?" In: *First Quality in Teaching and Learning Conference*. December (cit. on pp. 6, 7).

D3.js. [Online. Accessed 14-April-2013]. URL: http://d3js.org/ (cit. on p. 114).

Deterding, Sebastian et al. (2011). "From Game Design Elements to Gamefulness : Defining " Gamification "." In: *MindTrek conference*, pp. 9–15. ISBN: 9781450308168 (cit. on pp. 32, 36).

Dufresne, Robert J. et al. (Mar. 1996). "Classtalk: A classroom communication system for active learning." In: *Journal of Computing in Higher Education* 7.2, pp. 3–47. ISSN: 1042-1726. DOI: 10.1007/BF02948592 (cit. on pp. 5, 7–9, 76).

Feldman, Allan and Brenda M. Capobianco (Nov. 2007). "Teacher Learning of Technology Enhanced Formative Assessment." In: *Journal of Science*

Education and Technology 17.1, pp. 82–99. ISSN: 1059-0145. DOI: 10.1007/s10956-007-9084-0 (cit. on p. 53).

Fielding, Roy T. and Richard N. Taylor (May 2002). "Principled design of the modern Web architecture." In: *ACM Transactions on Internet Technology* 2.2, pp. 115–150. ISSN: 15335399. DOI: 10.1145/514183.514185 (cit. on p. 25).

flotr2. [Online. Accessed 14-April-2013]. URL: http://humblesoftware.com/flotr2/ (cit. on p. 114).

Free Mobile Polls. [Online. Accessed 14-April-2013]. URL: http://www.freemobilepolls.com/ (cit. on pp. 30, 38).

Froehlich, Herbert P. (1963). "What about classroom Communicators?" en. In: *Audiovisual communication review* 11.3, pp. 19–26. ISSN: 1556-6501. DOI: 10.1007/BF02768402 (cit. on p. 5).

Gauci, Sally et al. (Mar. 2009). "Promoting student-centered active learning in lectures with a personal response system." In: *Advances in physiology education* 33.1, pp. 60–71. ISSN: 1522-1229. DOI: 10.1152/advan.00109.2007 (cit. on pp. 7, 53).

Google Chart Tools. [Online. Accessed 14-April-2013]. URL: https://developers.google.com/chart/ (cit. on p. 114).

Groh, Fabian (2012). "Gamification: State of the Art Definition and Utilization." In: *Research Trends in Media Informatics*, pp. 39–46 (cit. on pp. 32, 36).

Hoffman, Christina and Susan Goodwin (2006). "A clicker for your thoughts: technology for active learning." In: *New Library World* 107.9/10, pp. 422–433. ISSN: 0307-4803. DOI: 10.1108/03074800610702606 (cit. on p. 53).

Horowitz, Harold M (1988). "Interactivity in a Classroom Environment." In: *Sixth Conference on Interactive Instruction Technology for the Society of Applied Learning Technology*. IBM Corporate Education Center. New York (cit. on p. 5).

HTML5. [Online. Accessed 14-April-2013]. URL: http://www.w3schools.com/html/html5_intro.asp (cit. on p. 108).

JavaScript. [Online. Accessed 14-April-2013]. URL: http://www.w3schools.com/js/default.asp (cit. on p. 108).

JavaScript InfoVis Toolkit. [Online. Accessed 14-April-2013]. URL: http://philogb.github.io/jit/ (cit. on p. 114).

jqPlot. [Online. Accessed 14-April-2013]. URL: http://www.jqplot.com/ (cit. on p. 114).

jQuery. [Online. Accessed 14-April-2013]. URL: http://jquery.com/ (cit. on p. 108).

JSON. [Online. Accessed 14-April-2013]. URL: http://www.json.org/ (cit. on p. 111).

Judson, Eugene and Daiyo Sawada (2002). "Learning from Past and Present: Electronic Response Systems in College Lecture Halls." In: *Journal of Computers in Mathematics and Science Teaching* 21.2, pp. 167–181 (cit. on p. 5).

LectureTools. [Online. Accessed 14-April-2013]. URL: http://www.lecturetools.com/ (cit. on pp. 30, 40).

Lidwell, William, Kritina Holden, and Jill Butler (2010). *Universal Principles of Design*. Rockport Publishers (cit. on p. 83).

Liu, Tzu-chien et al. (2003). "The Features and Potential of Interactive Response System Overview of Interactive Response System." In: *Proceedings of International Conference on Computers in Education (ICCE)*. Hong Kong, pp. 315–322 (cit. on p. 7).

Lowery, Roger C. (2006). "Clickers in the Classroom : A Comparison of Interactive Student-Response Keypad Systems." In: *National Social Science Association's National Technology and Social Science Program*. April, pp. 5–7 (cit. on p. 7).

Mazur, Eric (1997). "Peer Instruction." In: *Peer Instruction*. Vol. 30. 4. Prentice Hall. Chap. 2, p. 278. ISBN: 9780135654415 (cit. on pp. 7, 11, 12).

Mentimeter. [Online. Accessed 14-April-2013]. URL: https://www.mentimeter.com/ (cit. on pp. 30, 42).

MongoDB. [Online. Accessed 14-April-2013]. URL: http://www.mongodb.org/ (cit. on pp. 22, 23, 108).

mQlicker. [Online. Accessed 14-April-2013]. URL: http://www.mqlicker.com/ (cit. on pp. 30, 34).

NoSQL Database. [Online. Accessed 14-April-2013]. URL: http://nosql-database.org/ (cit. on pp. 22, 23, 108).

PINGO. [Online. Accessed 14-April-2013]. URL: http://wiwi.uni-paderborn.de/dep3/winfo2/forschung/projekte/peer-instruction-for-very-large-groups/ (cit. on pp. 30, 41).

Pinnion. [Online. Accessed 14-April-2013]. URL: http://www.pinnion.com/ (cit. on pp. 30, 38).

Poll Everywhere. [Online. Accessed 14-April-2013]. URL: http://www.polleverywhere.com/ (cit. on pp. 30, 36).

Poll Everywhere Comparison. [Online. Accessed 14-April-2013]. URL: http://www.polleverywhere.com/vs (cit. on p. 43).

Pyramid. [Online. Accessed 14-April-2013]. URL: http://www.pylonsproject.org/ (cit. on p. 108).

Python Programming Language. [Online. Accessed 14-April-2013]. URL: http://www.python.org/ (cit. on p. 107).

RealFeedback. [Online. Accessed 14-April-2013]. URL: https://realfeedback.tugraz.at/ (cit. on pp. v, vii, 2, 15, 117, 119).

Responsive Web Design. [Online. Accessed 14-April-2013]. URL: https://itservices.stanford.edu/service/web/mobile/about/terminology (cit. on p. 109).

Rogers, Yvonne, Helen Sharp, and Jenny Preece (2011). "Interaction Design: Beyond Human-Computer Interaction (3rd ed)." In: *Lecture Notes in Computer Science including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics* 5623 LNCS, pp. 229, 230 (cit. on pp. 58, 83).

Roschelle, Jeremy (2003). "Unlocking the learning value of wireless mobile devices." In: *Journal of Computer Assisted Learning* 19.May, pp. 260–272 (cit. on pp. 6, 53).

Scornavacca, Eusebio and Stephen Marshall (2007). "TXT-2-LRN: improving students' learning experience in the classroom through interactive SMS." In: *40th Annual Hawaii International Conference on. IEEE* (cit. on p. 6).

SMSPoll. [Online. Accessed 14-April-2013]. URL: http://www.smspoll.net/ (cit. on pp. 30, 33).

Socrative. [Online. Accessed 14-April-2013]. URL: http://www.socrative.com/ (cit. on pp. 30, 32).

Statistik Austria. [Online. Accessed 14-April-2013]. URL: https://www.statistik.at/web_de/statistiken/bildung_und_kultur/formales_bildungswesen/universitaeten_studium/index.html (cit. on p. 16).

Top Hat Monocle. [Online. Accessed 14-April-2013]. URL: https://www.tophatmonocle.com/ (cit. on pp. 30, 32).

Tremblay, Eric A. (2010). "Educating the Mobile Generation – using personal cell phones as audience response systems in post- secondary science teaching ." In: *Journal of Computers in Mathematics and Science Teaching* 29, pp. 217–227 (cit. on p. 6).

Turner, Daniel W. (2010). "Qualitative Interview Design : A Practical Guide for Novice Investigators." In: *The Qualitative Report* 15.3, pp. 754–760 (cit. on p. 57).

understoodit. [Online. Accessed 14-April-2013]. URL: https://understoodit.com/ (cit. on pp. 30, 37).

Ware, Colin (2004). *Information Visualization: Perception for Design*. Ed. by S K Card et al. Vol. 22. The Morgan Kaufmann series in interactive technologies / ed. Stuart Card. - San Francisco : Morgan Kaufmann. Morgan Kaufmann, p. 486. ISBN: 1558608192 (cit. on pp. 77, 78).

Wong, Kevin, Joseph Mohan, and Paul Lam (2011). "Classroom communication on mobile phones – first experiences with web-based ' clicker ' system." In: *ascilite Conference*. 1993, pp. 763–777 (cit. on p. 6).

Yau, Nathan (2011). *Visualize this: the FlowingData guide to design, visualization, and statistics.* Wiley Publishing, Inc. (cit. on pp. 69, 81).

Yerkes, Robert M. and John D. Dodson (Nov. 1908). "The relation of strength of stimulus to rapidity of habit-formation." In: *Journal of Comparative Neurology and Psychology* 18.5, pp. 459–482. ISSN: 0092-7015. DOI: 10.1002/cne.920180503 (cit. on p. 76).