Christian Haintz

# Quantitative Digital Backchannel: Developing a Web-Based Audience Response System for Measuring Audience Perception in Large Lectures

**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

While Audience Response Systems (ARS) are in use since the 1950s for military training purposes, their technique and usage as pedagogical tools to support the learning and teaching have changed significantly over the years. This thesis focuses on a specific type of ARS, the so-called quantitative digital backchannel system in the context of agile teaching. ARS of this type provide a continuous background feedback channel from the audience directly to the lecturer. An example quantitative backchannel feedback from the audience regarding the presentation speed could be "too fast". According to agile teaching the lecturer should react to this feedback and slow down the presentation. Although quantitative backchannel systems are far less researched than their qualitative counterparts, three services were identified, which support the quantitative aspect of backchanneling. This thesis analyzes these systems. Based on this analysis combined with the research of qualitative backchannel systems the requirements for a new quantitative digital backchannel system are derived. The new backchannel system is implemented as a web application using state of the art web technology such as HTML5, AngularJS, Responsive Design, and WebSockets. The auditor user interface is designed using an image-based approach, which tries to utilize the Tamagotchi effect for motivating the audience to participate. The first iteration of the new system is implemented focusing on the auditor interface and its usability. The implementation is evaluated during a lecture unit with 133 students. The acquired data during the live test are analyzed and discussed. It can be summarized that the participation during the live test of the new system was 75% of the audience. More than 80% of the participants voted more than three times during a 40-minutes lecture unit. Feedback shows that the visualization of the audience perception is crucial. The number of different device types during the live test supports the decision to implement the system as a web application.

# Kurzfassung

Audience Response Systeme (ARS) sind schon seit Mitte der 1950er Jahre zu militärischen Trainingszwecken im Einsatz. Die Technik und der Einsatz von ARS als pädagogisches Hilfsmittel, hat sich seit damals mehrfach verändert. Diese Arbeit behandelt eine Untergruppe der ARS im Kontext von Agile Teaching – die sogenannten quantitativen digitalen Backchannel Systeme. Diese Systeme arbeiten im Gegensatz zu den weit verbreiteten fragegesteuerten ARS im Hintergrund. Diese Art der Backchannelsysteme bietet den Studierenden die Möglichkeit quantitatives Feedback sofort und unmittelbar während der Vorlesung an den Vortragenden zu übermitteln. Ein mögliches Feedback in so einem quantitativen digitalen Backchannel wäre z. B. die Präsentationsgeschwindigkeit. Empfinden die Studierenden diese als zu schnell, können sie das mithilfe eines derartigen Systems schnell und einfach dem Vortragenden mitteilen. Der Vortragende kann dann sofort reagieren und die Vortragsgeschwindigkeit verringern. Obwohl quantitative Backchannel Systeme wenig untersucht sind im Vergleich zu den qualitativen, konnten im Rahmen dieser Arbeit drei existierende Systeme gefunden werden, die den quantitativen Aspekt unterstützen. Basierend auf der Analyse dieser Systeme und den wissenschaftlichen Untersuchungen von qualitativen Systemen, wurden Anforderungen abgeleitet die als Grundlage für das neu implementierte System dienen. Das neue quantitative Backchannel System wurde als Web-Applikation unter Verwendung aktueller Technologien wie HTML5, AngularJS, Responsive Design und WebSockets umgesetzt. Ein spezielles Augenmerk wurde dabei auf die Benutzerschnittstelle der Studierenden gelegt. Ein neuer bildhafter Ansatz wurde dabei verwendet, der sich den Tamagotchi Effekt zunutze macht um die Motivation der Benutzer zu steigern. Die Implementierung der ersten Iteration wurde in einer Vorlesungseinheit eingesetzt. Die aufgezeichneten Daten dieser Vorlesung wurden analysiert um die Zweckmäßigkeit der Anforderungen zu prüfen und das Benutzerverhalten besser zu verstehen. Es kann gezeigt werden, dass 75% von 133 Studierenden das neue Backchannelsystem verwendet haben. Über 80% davon haben in den 40 Minuten Präsentation abgestimmt. Rückmeldungen von Benutzern haben gezeigt, dass die Visualisierung der Informationen für die Benutzerfreundlichkeit entscheidend ist. Die Anzahl an unterschiedlichen Gerätetypen während des Tests unterstützt die Entscheidung das System als Webapplikation umzusetzen.

# Acknowledgments

I want to thank my girlfriend Karin for being with me on this journey. Without you, the writing of this thesis would not have been that much fun!

Big thanks to my supervisor Martin Ebner, for the great support during the whole masters program.

I would also like to thank my family who is standing behind me all the time.

Last but not least thanks to my friends. Sorry for being kind of unsocial during the writing of my thesis. This will get better now!

# Contents

# List of Figures

# 1 Introduction

"Audience Response System (ARS) comprise hardware and software which is used in conjunction with face-to-face educational processes to support, deepen, and enhance learning by promoting greater interaction between all those engaged in a learning activity." [Banks, 2006, p. vii]

ARSs can be used as tools in a broad range of strategies. One usual setup is based on multiple-choice questions asked to the audience during the lecture [Kay and LeSage, 2009]. Students can answer through the ARS and results are instantly transferred to the lecturer. The vision behind ARS is much broader than that. Besides multiple-choice questions, open questions, homework assignments, and learning games are also suggested as a use case scenario for ARS [Abrahamson, 2006].

This work focuses on a certain topic of the vision of ARSs: the continuous interaction from students to lecturer. While answering multiple-choice questions with the support of ARS must be well integrated in the lecture, this work focuses on the development of a web-based system which directly supports the students-to-lecturer channel. The system runs in the background of the lecture and provides continuous communication from students to the lecturer. This so-called *digital backchannel* [Bruff, 2009, p. 62; Yardi, 2006, p. 1] should give the lecturer the opportunity to get more insights how the audience incorporate with the lecture and to react according to it just in time. From the students' perspective, it should give them more influence how they get taught.

There are two main types of backchannel systems: qualitative and quantitative backchannel systems. While the qualitative backchannel is gaining focus in research [Yardi, 2006; Ebner, 2009; Purgathofer, 2008; Atkinson, 2009; Gehlen-Baum, Pohl, and Bry, 2011; Pohl, Gehlen-Baum, and Bry, 2011; Baumgart and Pohl, 2011; Gehlen-Baum, Pohl, Weinberger, et al., 2012; Pohl, Gehlen-Baum, and Bry, 2012] this work focuses on the quantitative backchannel system. Although less attention is paid in this field in research, existing systems and services which supports quantitative features are evaluated. Combined with findings from literature focused on qualitative digital backchannel a new approach to quantitative digital backchannel systems is evaluated and implemented.

An example scenario for such a continuous interaction from students to the lecturer in a quantitative backchannel system could be the lecturer's presentation speed. With a backchannel system, students can submit if the presentation is too fast to understand or if it is too slow to keep attention for the topic. The ARS instantly informs the lecturer via a simple non-distracting visualization about the current situation in the lecture hall.

This work addresses the filtering of the requirements needed for a quantitative backchannel system based on the existing products and services. Further, the technical approach is discussed. A first development iteration is implemented and live-tested in a lecture. Feedback and acquired data are analyzed to evaluate if the requirements, the technical approach and the usability assumptions are reasonable or need to be adapted for the next iteration.

## 1.1 Structure

In chapter 2 the terms and the pedagogical environment of ARS for this work are explained. In chapter 3 the history of ARS with the focus on digital quantitative backchanneling is covered. Chapter 4 discusses and evaluates three state of the art services, which support quantitative backchanneling.

The implementation of the new quantitative backchannel system is covered in chapter 5. This chapter starts with the requirements for the new system. Based on the requirements the system architecture is derived. A technology stack is built which supports this architecture. Further, implementation specific decisions such as user interface design are discussed.

The implemented system is evaluated in a lecture unit to acquire data for evaluation. The findings based on this live test are discussed in chapter 6. The work closes with the conclusion provided in chapter 7 and gives an outlook of future perspectives in chapter 8.

# 2 Terms and Definition

In this work, the digital backchannel is primarily seen as a technical system in the context of an ARS used in formative assessments to support agile teaching and active learning. A distinction is made between qualitative and quantitative backchannel systems. Additionally, an explanation of the Bring Your Own Device (BYOD) policy and the integration of a digital backchannel system to a lecture is discussed in this chapter.

## 2.1 Agile Teaching and Active Learning

"The instructor continually probes for and adjusts to the students' learning needs — a practice we call agile teaching." [Beatty, Gerace, et al., 2006]

Agile teaching concisely means the lecture is led by students' feedback that the teacher transforms into a proper learning environment for the students' need. In contrast, the ballistic approach, where the lecture unit is planned and launched. In the ballistic approach, the next exam will show if the learning goal is reached or not [Beatty, Leonard, et al., 2006, p. 4]. In agile teaching, a feedback loop should give instant feedback how the teaching is affecting the students.

Active Learning is the equivalent methodology for the learner's perspective. The learner mind is not like an empty vessel which is waiting to get filled by knowledge [McManus, 2001, p. 424]. Instead, "students engage in effortful, directed cognitive activity in order to assimilate and refine new ideas and structure their knowledge" [Beatty, Leonard, et al., 2006, p. 2].

Agile teaching similar to active learning must use a tight feedback loop, which continuously monitor student behavior. The lecturer must be capable to react instantly based on the continuous stream of feedback from the audience. A commonly used strategy for supporting agile teaching and active learning is formative assessment.

## 2.2 Formative Assessment

ARS as part of formative assessment can support agile teaching [Bruff, 2009, p. 41]. Formative assessment provides feedback for the learner and the teacher to improve learning and teaching [Bransford, Brown, and Cocking, 2000, p. 24]. The way to achieve the learning goal is depending on the learning stage of the learner and the flexibility of the teacher. The learner should adapt the learning approach, for example, based on existing knowledge on this topic and give feedback to the teacher about the learning progress. The teacher should adapt the lecture according to the learner's feedback minute by minute. To validate if formative assessment has improved learning and teaching, the feedback loop between learner and teacher must be closed [Boud, 2000, p. 158]. As illustrated in figure 2.1 ARSs can be used to close the feedback loop and get this information about the current stage of thinking and understanding which are central elements of formative assessment.

Another assessment strategy is summative assessment. It is often referred as distinct to formative assessment [Gedye, 2003, p. 40]. Summative assessment focus on the summation of all student's assessments to conclude the performance of the student, for example, reflected in the grading. Black and Wiliam [1998] talk about the formative and summative functions of assessments. A formative and a summative function can be in one assessment. For example, multiple-choice questions during lecture provided by an ARS can be formative if the lecturer gives good quality feedback. At the same time, the student's answers can be used as a factor of a summative grading.

ARSs can be used in both assessment strategies, this work focuses on the formative aspects.

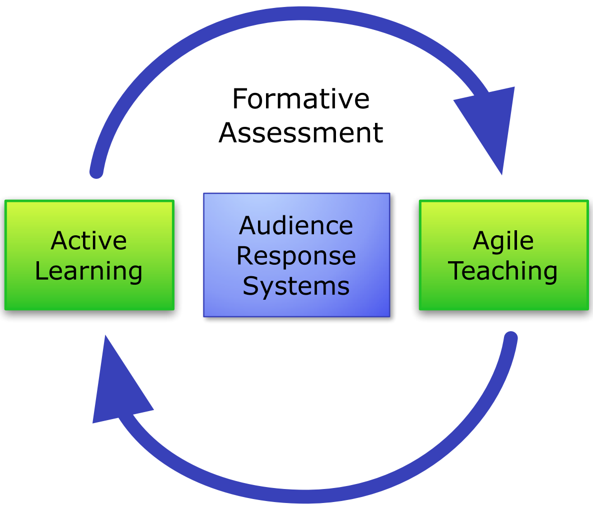


Figure 2.1: ARS as a tool for formative assessment to support active learning and agile teaching. Based on figure by Beatty, Leonard, et al., 2006

## 2.3 Digital Backchannel

A digital backchannel in the e-learning context uses communication technology to provide a backchannel between an audience and a lecturer. The definition of backchannel itself varies in the context and usage. The main intention of a backchannel is to provide an additional channel, which complements the frontchannel, which is represented by a teacher, lecturer, professor, or speaker. Whereas the frontchannel is the main communication channel, the backchannel usually is a secondary background channel which supports the frontchannel [Yardi, 2006].

Bruff [2009, p. 62] explains the backchannel in the context of ARS as a continuous formative feedback channel for the lecturer.

Besides the technical challenges of such a system, user experience is a main goal. For the audience, the backchannel system needs to be non-distracting and easy to use. In the best case, it should be fun to use acting as a motivator. Gamification might be one way to foster motivation.

For the Lecturer, it should additionally be non-distracting. The system should give the lecturer important information about the current situation of the audience. The measurement and reduction of the information from the audience is crucial for the lecturer. Backchannel as a type of ARSs can be seen as a technical system. This work focuses on backchannel systems in the context of agile teaching.

## 2.4 Audience Response Systems

Backchannel systems are a type of Audience Response System (ARS). ARSs are systems which support the interaction of audience and lecturer. A distinction can be made in backchannel and frontchannel ARS. Frontchannel systems are actively integrated in the lecture presentation while backchannel systems are intentionally in the background trying not to distract from the lecture itself. At the same time, the system supports the interaction or communication besides the main lecture presentation [Yardi, 2006]. The term ARS is often used instead of question-driven ARS, which can be categorized as a frontchannel ARSs. The term digital backchannel is also influenced by social media services like Twitter [2013], which can also act as a backchannel when used for that certain purpose. Digital backchannel systems are, therefore, not tightly related to education which neither are ARS [Alexander et al., 2009; Atkinson, 2009]. Besides education, ARSs including the digital backchannels are used at conferences, presentations, TV interviews, and other situations where the audience interacts via technical systems [Atkinson, 2009].

While distinctions between types of ARSs are historically established, backchannel systems have no well-established distinctions. Therefore, two categories of digital backchannel systems are introduced. The qualitative backchannel systems and the quantitative backchannel systems as seen in figure 2.2.

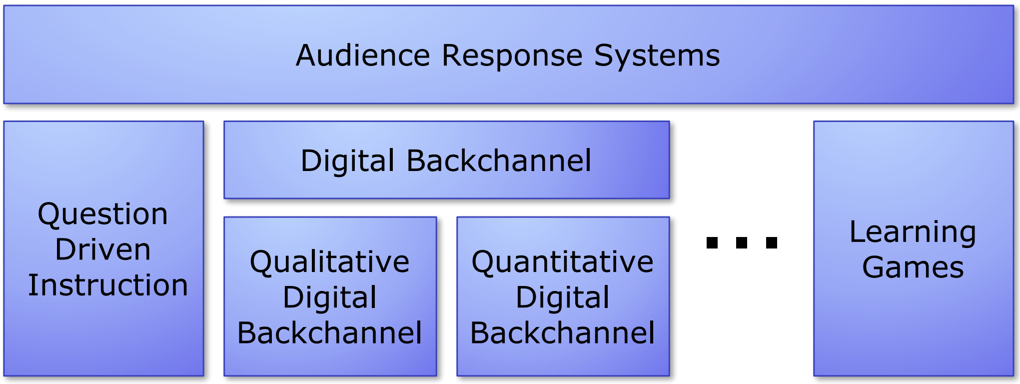


Figure 2.2: Quantitative digital backchannel in the context of ARS

## 2.5 Qualitative Backchannel Systems

Qualitative backchannel systems are often implemented as commenting and open feedback systems for lectures. The term qualitative in the context of backchannel systems is related to the input method for the auditors. In qualitative backchannel systems, auditors can typically write comments in the system that are sent to the lecturer. The lecturer might read it besides lecturing and integrate the comments instantly in the current presented topic if applicable. This type of backchanneling can also be achieved with the micro blogging service Twitter [2013] with the effect that all comments are public and visible worldwide. Twitter is often used for this purpose in conferences with varying success [Atkinson, 2009; Reinhardt et al., 2009]. Twitter can be used as a backchannel even if the lecturer is not integrated in the feedback loop. This form can get distracting to the lecturer [Atkinson, 2009].

To use qualitative backchanneling to support agile teaching the lecturer needs to integrate the system and the method in the lecture. If the audience uses backchannel tools introduced by the lecturer, the audience generally expects the lecturer to react to the feedback given by the audience. Otherwise, the audience might stop giving feedback or use the backchannel to criticize the lacking reaction to the backchannel. To satisfy the audience and to support agile teaching, the lecturer has additional tasks to do besides lecturing [Ebner, 2011]. The lecturer needs to follow the backchannel continuously, filter the information, for example, in constructive and non-constructive comments, and integrate or react to it seamlessly in the ongoing lecture. It can be assumed that these additional tasks to perform agile teaching with qualitative digital backchannel systems are not easily manageable by non-trained lecturer.

There are techniques, which address the added workload for the lecturer. For example, a lecture, or presentation can be divided into smaller chunks. At the end of each chunk, the backchannel is reviewed by the lecturer, which is followed by a time-boxed discussion about the feedback and comments. After that the next lecture chunk is continued based on the discussion outcome.

In contrast *Backstage* is a digital backchannel system especially developed for educational purposes, which also focuses on qualitative feedback [Pohl, Gehlen-Baum, and Bry, 2011; Gehlen-Baum, Pohl, and Bry, 2011]. Backstage fully integrates the lecturer and helps him to filter the information transferred on the backchannel. Backstage uses a weighting algorithms for the pre-filtering of information communicated over the Backstage tool. Backstage expands the backchannel term in a way that Backstage is not only used to transfer feedback and comments from the audience over a backchannel to the lecturer. Backstage actively supports the inter-audience communication and even collaboration. The communication channel is extended to a virtual communication room where every auditor can participate, similar to a chat room. The lecturer gets only pre filtered information from this communication room based on filtering algorithms, which should reduce the "information noise" for the lecturer.

While qualitative systems obviously have advantages based on the qualitative data they provide, they have several disadvantages in live operation. Qualitative data are hard to filter, summarize, and interpret for present algorithms. The lecturer must do the final interpretation ideally instantly, which needs active training and adaption of the lecture style to support agile teaching.

Distraction resulting from qualitative backchannel systems is also a risk for the audience. Entering qualitative data needs concentration, which might lead to loss of attention for the lecture. However, this is a risk of all backchannel systems based on their definition of working in parallel to the frontchannel. Human brain is very limited in multi-tasking capabilities [Miller, 1994] therefore distraction by the backchannel system should be reduced to a minimum to not countervail the lecture.

## 2.6 Quantiative Backchannel Systems

Quantitative backchannel systems compared to qualitative backchannel systems are transferring quantitative data over the backchannel. This can be achieved by giving the audience predefined answers, which they can submit. The quantitative aspect is related to the audience input data. Similar to question-driven frontchannel ARSs which are usually also a quantitative feedback systems. The main differences are the continuous feedback by the audience and the usage in the background of the lecture. While in a typical question-driven ARS, votes for a specific question can only be submitted once by one auditor, in a quantitative backchannel system multiple votes are accepted for a question over a defined time. A reasonable example regarding agile-teaching for a quantitative backchannel question would be "Do you understand?" with the two answer choices *yes* and *no*. If this question is open during the whole lecture, and auditors can submit multiple votes due to the fact that their opinion can change, it is a valid quantitative backchannel system. On the lecturer screen of the system, the current result of the voting should be displayed. This information can give the lecturer an indicator if the audience comprehends the lecture.

While the previous example is a simple example, the core advantage of quantitative backchannel systems is simplicity. Quantitative data can typically entered faster than qualitative data. Therefore, participating and submitting data is less distracting for the audience. The system itself can preprocess, filter and interpret quantitative data effectively and present the lecturer only the important information and filtering out the noise automatically. This leads to less work for the lecturer compared to qualitative backchannel systems. This should also reduce the distraction caused by the backchannel system. For example, if the lecturer gets the information that most of the audience does not understand what is lectured the lecturer could recap the topic or ask the audience what exactly is not understandable. This workflow already supports agile teaching. Assuming the lecturer is experienced in direct questioning by the audience this approach needs less attention from the lecturer for the system during lecturing. No qualitative information needs to be processed; the auditor is signaled only on important changes on the audience perception.

The difference in the work-flow, and effort between qualitative backchannel systems and quantitative backchannel systems can be roughly compared to a conventional lecture where two different questions are asked for, one quantitative and one qualitative. Each question represents one type of backchannel. The question for the quantitative system is “Hands up please who have not understood this topic.” while the qualitative question is “Everybody who has not understood this topic, please tell me what exactly is not understandable.”. While these questions are clear frontchannel interactions, the activity flow and the result they generate are similar to the corresponding types of backchannel systems. It can be assumed that the quantitative question can provide an immediate trend based on the number of hands, which are raised, compared to the whole audience size. Based on this information the lecturer can decide how to react. For the qualitative question, it can be assumed that the answers are more individual, and the lecturer has to process each information individually, for example, if different comments address the same misunderstood part of the lectured topic. Based on the different comments from the auditors the lecturer can decide how to react.

This example should illustrate the strengths and weaknesses of the two distinct approaches. It can be assumed that the quantitative question and the interpretation of its result is less time-consuming and challenging compared to the qualitative approach. However, there is also less acquired information with the quantitative approach. Even if, the lecturer knows that 80% have not understood the topic, the lecturer still has no clue what exactly is not clear to the audience. In contrast, the qualitative approach can be assumed more time-consuming and challenging for the lecturer. The result can also be expected to be more precise. Based on the individual answers in the qualitative approach, the lecturer will know which parts the audience did not understand. Based on this qualitative result the lecturer can react more targeted than on the quantitative approach.

## 2.7 Bring Your Own Device (BYOD)

BYOD is a policy which appreciates or at least allows the use of devices owned by the employees for business related applications and to access the business network. In the academic world, this can be transferred to the usage of students own devices during lecture to access university related services, for example ARSs [Logicalis, 2012].

The usage of mobile Internet devices like smartphones and tablets increased significantly in the last years [Ebner, Nagler, and Schön, 2012; Emery, 2012]. While the BYOD policy brings obvious advantages compared to hardware provided by the universities, there are also risks when BYOD is used [Thomson, 2012]. While most of the disadvantages are focusing on companies, some can be transferred to academic scenario.

Focusing on the academic scenario, an advantage of BYOD is the reduced hardware costs for the universities. Early ARS which were often realized as wired systems or involves custom clicker equipment, which had a significant amount of expenses related to installation and maintenance of the hardware besides purchasing the clickers. Distribution, collection maintenance, and loss management need to be considered as a cost factor for devices owned by the university. With mobile Internet devices brought by the students, these management tasks and costs can be prevented.

Disadvantages of the BYOD policy are the lack of control what devices are used for. Besides laptops, many of these devices are typically based on a mobile Operating System (OS) which allows the installation of so-called mobile apps which enhance these devices to general purpose computers with some limitations. The large range of application of these mobile devices also leads to a disadvantage for the academic use. Besides using the devices for the intended function during lecture, they could also be used for other purposes, which are distracting the auditor from participating in the lecture.

The requirements for the use of BYOD are a stable Internet connection which is available for all auditors. The Internet connections should also be available to all devices independent of the used OS. Further, the Internet connection should be fast and reliable.

Because most of the BYOD devices should be capable of executing the ARS, the ARS needs to support the vast majority of devices and platforms used by students. A modern browser can be assumed to be on board on most of the modern mobile Internet devices. Therefore, ARS which are implemented as web applications for modern browsers considering the needs of a mobile user interface, can be assumed to suit the BYOD policy.

## 2.8 Integration of a Digital Backchannel System in a Lecture

What differences can be assumed between conventional lecture feedback? Which technical requirements need to be fulfilled?

One aspect in conventional feedback is the lack of anonymity. For example on the previous scenario in 2.6 where the following question was asked "Hands up please, who have not understood this topic." People tend to be shy in a large group and will unlikely admit that they have not comprehended the topic. This retention will increase the larger the audience size is [Abrahamson, 2006]. Providing anonymity will lower the barrier to show the current perception. This can also become a problem in qualitative feedback systems because the tone of the feedback can also get worse.

A disadvantage of the digital backchannel is the need of devices, which allow participating in the backchannel process. Earlier ARSs used custom devices called clickers to collect votes from the audience. The current trend goes to BYOD [Logicalis, 2012]. Even if the saturation of mobile Internet devices like smartphones, tablets, and laptops are increasing, not every auditor might have one. This locks some auditors out of the interaction process. This can be addressed by providing spare devices for auditors who have no mobile Internet device. This will become more and more needless when saturation of mobile Internet devices continuous increasing.

The lecture hall needs to be equipped with adequate fast Internet access which is reliable and designed to provide Internet for a full lecture hall. This is a requirement for all Internet-based ARS systems. Providing access to the Internet for every auditor leads to another possible disadvantage: the potential distraction by the Internet itself if access is not limited to the ARS tool. While this might seem problematic, finding the reason why auditors prefer surfing the Internet instead of participating in the lecture could probably be identified with the digital backchannel.

The lecturer also needs an Internet device to consume the information provided by the backchannel system. This device needs not necessarily be a laptop. A tablet or a smartphone might be a suitable choice depending on the backchannel system.

# 3 History

Backchannel systems have evolved from ARSs. ARSs have a long history starting in the 1950s by the military. Electronic devices were used to respond to multiple-choice questions during training films [Froehlich, 1963]. In the 1960s ARSs were used for marketing research and unreleased motion pictures [Collins, 2008]. The first ARS installed and in use for academic purposes was in Stanford University in 1966. The wired system in the lecture hall was using a voltmeter as the lecturer interface to display the number of votes for an answer [Abrahamson, 2006]. Garg, 1975 already used an ARS to provide a quantitative backchannel to the lecturer by letting students vote for the pace of the lecturer. Students could react to the presentation speed by voting for "go faster" and "go slower" continuously. Nevertheless, using ARSs as a backchannel tools are not intensively researched at that time. The focus was mainly on the question driven frontchannel.

In 1998 when the TV show "Who Wants to Be a Millionaire" started, ARSs are getting well known to a broad audience. This quiz show uses a lifeline named "Ask the audience" where an ARS is used.

Over the years, the wired systems were replaced or modified to wireless systems. The systems started using electronic mobile devices already present in education such as phones, handheld organizers, and graphic calculators [Roschelle, 2003; Alexander et al., 2009]. At the same time, web-based systems are also evolving in the e-education [Chun, 2004]. This leads to solutions which introduced the use of smartphones and tried to combine their possibilities of mobile Internet with web-based ARSs [Lam et al., 2011; Elliman, 2006].

While the quantitative backchannel is rarely found in history, the qualitative approach is getting more attention in the recent past [Yardi, 2006; Ebner, 2009; Purgathofer, 2008].

# 4 State of the Art

The focus in this research work is on quantitative backchannel systems which are supporting BYOD. While many different systems on the Internet [Shambles.net, 2013] can be found with the term "backchannel" tools, many of them are collaboration systems with an academic focus. Qualitative feedback systems including question driven ARSs are also often categorized as backchannel systems. The academic literature research revealed only systems and methods which can be categorized as qualitative backchannel systems [Pohl, Gehlen-Baum, and Bry, 2011; Ebner, 2009; Purgathofer, 2008; Atkinson, 2009; Yardi, 2006] considering the definition in section 2.5. Three systems could be identified which supports quantitative auditor input to the backchannel [MyTU, 2013; NK Labs Inc, 2013; Lecture Tools, 2013].

## 4.1 MyTU

MyTU is a smartphone application focused on students of the TU Bergakademie Freiberg [TU Freiberg, 2013], which provides access to the library catalog, lecture schedule, and other systems and information from the university. The application is available for the smartphone platforms iOS and Android. There is also a backchannel module in the app. The student's interface in figure 4.1 on page 20 provides a feedback channel for the speed of the lecturer and a comprehension dimension to indicate if the lectured topic is understandable or not.

The speed dimension offers three different feedback values *too slow*, *good* and *too fast*[1]. To address the comprehension dimension there is a *stop* but-

[1] Text of the German MyTU app is translated to English by the author.


Figure 4.1: Auditor interface (german) of MyTU app [MyTU, 2013]

ton. The button can be pressed by the auditor to indicate that the current topic was not understood. Additionally since the recent version 2.1.0 there is also a qualitative feedback channel, which allows sending text questions directly to the lecturer. While the smartphone app is available for free and can be installed by everyone, access to the backchannel functionality is technically limited to courses and students on the university. Therefore, this app was evaluated by reading through the version history and description about the application on the website [MyTU, 2013]. Because of this limitations, the behavior of the lecturer interface could not be evaluated.

## 4.2 Lecturetools

Lecturetools.com [Lecture Tools, 2013] states to provide an active learning platform realized as an ARS. While the main system interface is web-based, there is also an iPad app for auditors. For those auditors who do not have an Internet device, votes can also be sent via text message from cell phones. Leturetools is not focused on quantitative backchannel, but there is a quantitative backchannel feature available in the system. On figure 4.2 on page 22 on the upper left there is a button with a red flag. If this button is pressed, the flag, which is by default colored gray, will turn into red. This way the auditor can indicate the lecturer that the current slide was confusing. The auditor can toggle the state of the flag. This is a one-dimensional backchannel for comprehension. The possible values are binary (*understood* when flag is gray, *confused* when the flag is red). The lecturer interface (see figure 4.3 on page 23) visualizes the number of confused auditors compared to all auditors in a bar chart in percentage. To use the Lecturetools, in a real situation, costs arise. To use the web-based tool, auditors and lecturers need to register online and sign-in. There is a free trial license for the first two lectures after registering online.

## 4.3 Understoodit

Understoodit [NK Labs Inc, 2013] has started as a pure quantitative backchannel system in May 2012. Meanwhile, it also integrates question driven ARS features. For this work, the focus is on the quantitative backchannel aspects. The system is implemented as a web application. Lecturers need to register to use it, auditors only need to open a link pointing to the lecture in a browser. Figure 4.4 on page 25 shows the auditor interface. The interaction possibilities can be seen on the upper right part of the screenshot. Two buttons are available: one for the state *understood* and one for the opposite state *confused*. This can be assumed as one comprehension dimension with binary values. The diagram on the lower part indicates the perception in the audience.

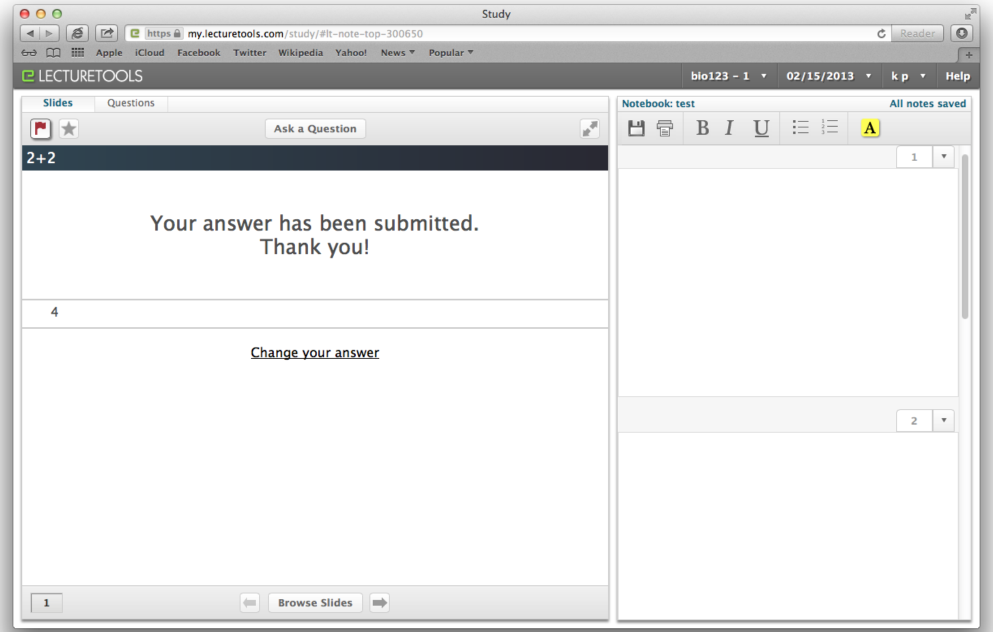


Figure 4.2: Auditor interface of Lecturetools [Lecture Tools, 2013]

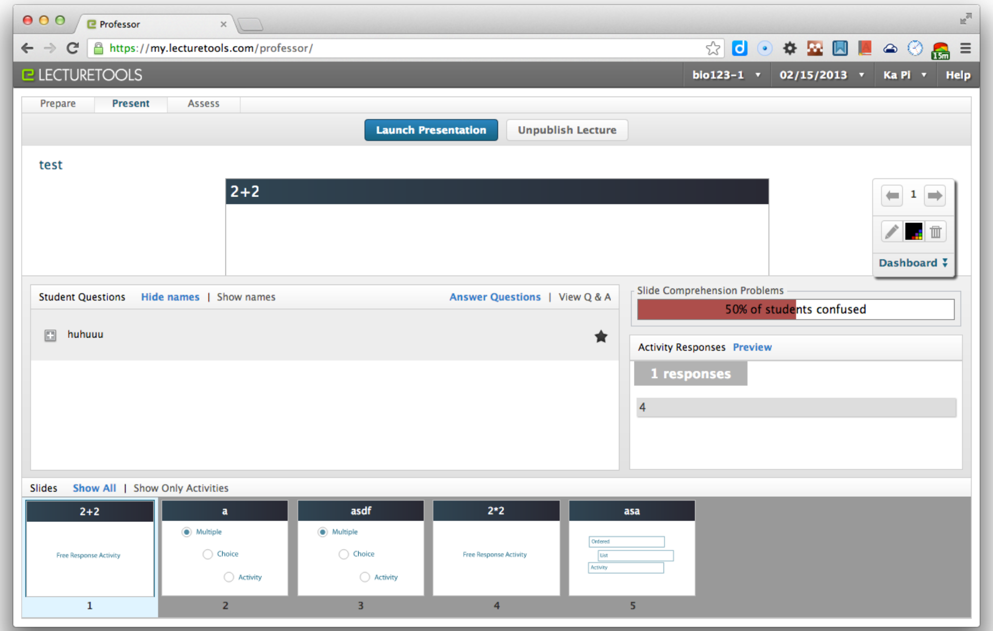


Figure 4.3: Lecturer interface of Lecturetools [Lecture Tools, 2013]

The buttons can be pressed multiple times. The only limitation is 15 seconds (15 seconds is the default value) must elapse between two clicks. The vote value while only binary during voting expands to a discrete value by including the time as additional information. Observations have shown that the system handles a state for every auditor. When the auditor presses *confused* the state goes to *100% confused*. Over time the state is decreasing until it reaches its neutral state. When *confused* is pressed by the auditor, and 15 seconds later the *understood* button is clicked, the state of the auditor changes instantly from *100% confused* to *100% understood*. This leads to the assumption that each auditor has one state which is in between *100% confused* and *100% understood*. The auditors can change their state by pressing the buttons. This method collects additional information for the lecturer. Recent activities have more impact on the audience perception compared to votes which are in more distant past. The lecturer interface is shown in figure 4.5 on page 26. The interface for the lecturer shows the same backchannel information as the auditor interface.

Understoodit is free of charge for the smallest plan, which includes the full working quantitative backchanneling module. Limitations on the smallest plan are related to the question-driven ARS features and uploading of presentations.

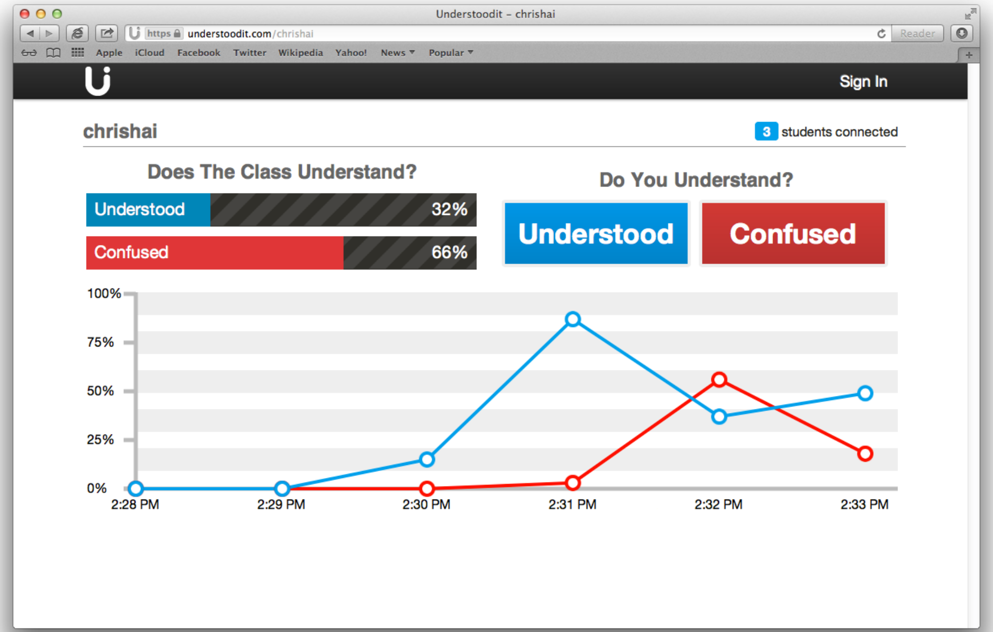


Figure 4.4: Auditor interface of Understoodit [NK Labs Inc, 2013]

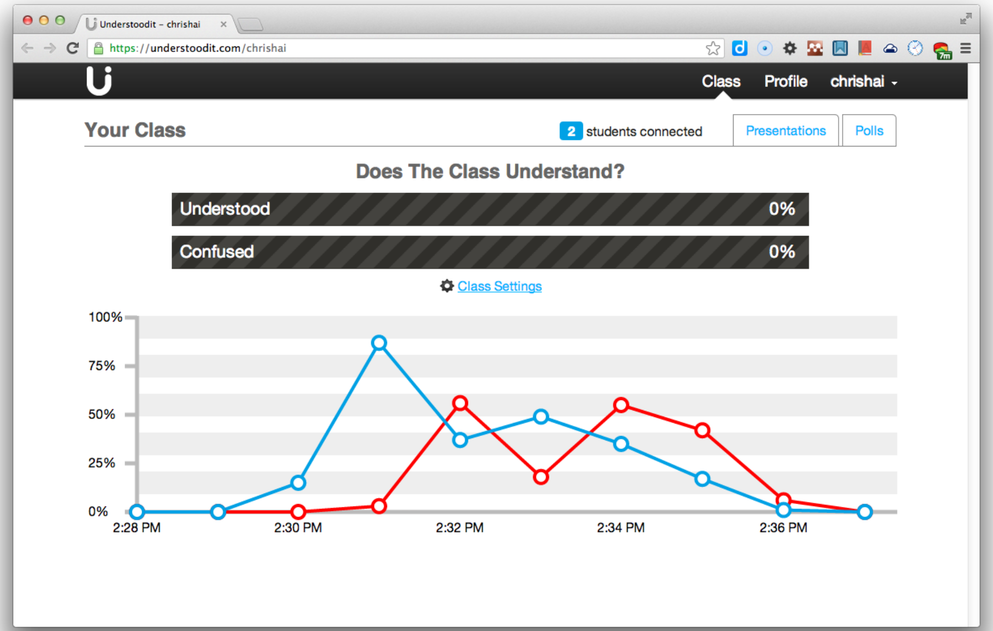


Figure 4.5: Lecturer interface of Understoodit [NK Labs Inc, 2013]

# 5 A New Quantitative Backchannel System

While there are many ways to interact via a backchannel system in lecture situations most of the available systems are qualitative backchannel solutions such as Pigeonhole Live [Pigeon Lab Pte Ltd, 2013], Twitter [Twitter, 2013], and Backstage [Pohl, Gehlen-Baum, and Bry, 2011]. There is one system named LectureTools [Lecture Tools, 2013], which provides support for quantitative backchannel votes integrated in a complete ARS solution. Two systems could be identified which are focused on quantitative backchanneling during lecture: Understoodit [NK Labs Inc, 2013] and MyTU [MyTU, 2013].

Based on the literature research quantitative backchannel systems are less researched compared to qualitative systems. Research results from qualitative backchannel systems will be integrated in the design and implementation of the new quantitative backchannel system.

Based on the state of the art research and experience which was acquired by designing, developing, and evaluating of a question driven ARS named RealFeedback [Pichler, 2013] a different approach compared to the existing quantitative backchannel systems is implemented and evaluated. Question driven ARSs are different to backchannel ARSs and not all findings from RealFeedback can be applied to backchannel systems. Nevertheless, many findings are not tightly related to question driven ARSs but can be stated useful for different web-based ARSs using BYOD in the lecture.

The realization of the new system [Carrot & Company GmbH, 2013] is focusing on the strengths (see 2.6 on page 11) a quantitative backchannel system can provide while at the same time trying to reduce its weaknesses.

## 5.1 Development Process

The development process is structured as an iterative agile development process. The evaluation of the system is a core component of this process. The process is realized with the agile manifesto in mind.

| **Infobox 5.1.1 Agile Manifesto [Beck et al., 2001]** |
| --- |
| "Individuals and interactions over processes and tools<br>Working software over comprehensive documentation<br>Customer collaboration over contract negotiation<br>Responding to change over following a plan" |

The iterative cycle is modeled as following:

1. Planning
2. Requirements
3. Analysis & Design
4. Implementation
5. Testing and Evaluation

In this work, the first iteration cycle is realized.

## 5.2 Requirements

The following requirements address features, which are assumed to be important in such a system and might be missing in current systems. Additional requirements are stated which addresses general properties of ARSs and are assumed to be important. The requirements are also influenced by previous experience in designing and development of web services for different use cases.

Based on these long-term requirements, requirements for the first iteration are derived. Goals for the first iteration are the auditor user experience and the overall performance of the system. Lecturer perspective, while kept in mind during the development of the iteration, is not in the focus of the first iteration.

**Requirement 1** *Constant and continuous backchannel activity should be aspired to produce meaningful output*

Requirement 1 is based on the feedback loop which should be established to support agile teaching and active learning [Flint, 2012]. By aspiring active participation of the audience in the backchannel, the system can produce meaningful information for the lecturer to react appropriately. Especially in quantitative backchannel systems the amount of data is a vital factor for the quality of the generated output for the lecturer. To address this requirement, the focus is laid on the user interface and the user experience for the audience. If the motivation can be generated by the system itself to actively use the backchannel system, no additional effort during lecture is necessary. There are different ways to conquer this challenge. Gamification is a trending method to improve motivation [Groh, 2012]. This requirement is relevant for the first iteration.

**Requirement 2** *Distraction for both auditors and lecturer should be reduced to a minimum*

Requirement 2 can be assumed as a core requirement to all backchannel systems. If the backchannel system is too distracting for the audience or the lecturer, the use of the system is abolished and might even sabotage the lecture. To conquer distraction the user interface must consider the environment in a lecture hall during the lecture for both, the auditors and the lecturer. It can be assumed, the better the user experience the less distracting the system will be mentioned by the users. The more different things an auditor or a lecturer has to manage simultaneously, the higher the chance to get distracted [Miller, 1994]. This requirement is relevant for the first iteration.

**Requirement 3** *The system should be developed with usability goals considered from the beginning.*

Requirement 3 states the consideration of usability goals such as effectiveness, efficiency, safety, utility, learnability, and memorability [Sharp, Rogers, and Preece, 2007]. This goes hand in hand with requirement 2. The better the usability goals can be achieved the less distracting the system will be. This requirement is relevant for the first iteration.

**Requirement 4** *The system itself should be as simple as possible to achieve the required tasks*

Requirement 4 can be assumed as a requirement for system development in general. The architecture, code, and design should be following this requirement. This is also a value of agile software development following the question: "What is the simplest thing that could possibly work?"[Beck, 2006]. This requirement is relevant for the first iteration.

**Requirement 5** *Bring your own device policy should be supported by the system*

Requirement 5 supports the increasing trend of BYOD which reduces the costs for the university or host of the lecture by supporting devices owned by the auditors. This requirement leads to easy or even no installation of the backchannel system. Another implication of this requirement is cross-platform support which is also addressed in requirement 9. This requirement is relevant for the first iteration.

**Requirement 6** *The User Interface (UI) should adapt across devices and make the best of the available resource each device type provides*

Requirement 6 is important regarding the BYOD policy stated in requirement 5. While it is possible to design and develop different versions of the application for every device available, it is assumed less effort to make the application responsive to the device properties itself. The system should choose the optimum design based on the device characteristics. This makes the system more stable against new devices, which were not available during development. Because the system reacts on the properties of the new devices, it can be assumed that the system adapts according to the available resources, even if the system does not know the device itself. While responsive design is a term known from web design [Marcotte, 2011] the principal can be applied generically to cross-device development. This requirement is relevant for the first iteration.

**Requirement 7** *Actions by auditors need to generate visible impact*

Requirement 7 assumes that the auditors who participate in the backchannel system want to see the reaction of their actions. For example, if an auditor tells the backchannel system about not comprehending the current topic, the auditor expects to see the collective perception changes. This assumption is based on the collective action principle that an individual auditor wants to be a member of the group who can change the feedback to the lecturer (the advantaged group) [Wright, Taylor, and Moghaddam, 1990]. Therefore, the individual auditor needs to get informed of being a member of this group when participating in the backchannel. The auditor's action needs to have an impact on the collective action, which is presented to the auditor. This requirement is relevant for the first iteration.

**Requirement 8** *The information provided to the lecturer should be reduced to the most essential meaningful information to give a good overall impression about the audience without distracting the lecturing.*

Requirement 8 addresses the lecturer perspective. Although information in a backchannel system is transferred from the audience to the lecturer the representation of this information target different goals. While it can be assumed that auditors want information regarding their impact on the collective perception and about the recent activities (see also requirement 7), this is not assumed suitable for the lecturer. The lecturer wants to get information regarding significant changes in perception of the audience. It can be assumed that the current activity, as long as it does not change the overall perception, is of minor interest to the lecturer or is even just distracting. Too many parallel tasks cannot be handled proper by the lecturer [Miller, 1994]. Only relevant information should be presented to the lecturer in a clear and non-distracting way. This requirement is not relevant for the first iteration as the first iteration focuses on the auditor perspective and the overall performance.

**Requirement 9** *Cross-platform capabilities should be considered from the beginning*

This requirement 9 assumes that based on the BYOD requirement 5 the auditors have different devices which support different platforms. Cross-platform capabilities are therefore essential to support most of the auditor devices. Cross-platform capabilities are also a fundamental requirement for the lecturer interface. It cannot be assumed that different lecturers are using the same platform. Nowadays OS platform fragmentation is significant and is further increasing [Statcounter.com, 2013]. If the target group of users cannot be narrowed down to a specific target group, which correlates with a single platform, cross-platform capabilities should be considered a core requirement from the beginning. This requirement is relevant for the first iteration.

**Requirement 10** *Internationalization should be considered from the beginning.*

This requirement 10 can also be assumed as valid for a broad range of software development projects. Internationalization is the first step to prepare the system or application to make it accessible in different locations worldwide while supporting possible future locations with their typical language, date formats, etc. Generally, the expenses for introducing internationalization in the first iteration are much lower compared to introducing internationalization to a system in an advanced stage [Freij, 2012; Larman, 2003]. Internationalization should be part of the architecture from the beginning. This requirement is also added as a requirement for the first iteration.

**Requirement 11** *Maximizing the meaningful information provided by the auditors while keeping it simple*

Requirement 11 assumes that even if the system is quantitative and not qualitative the maximum possible amount of meaningful information should be requested from the auditor. While the inspected quantitative systems

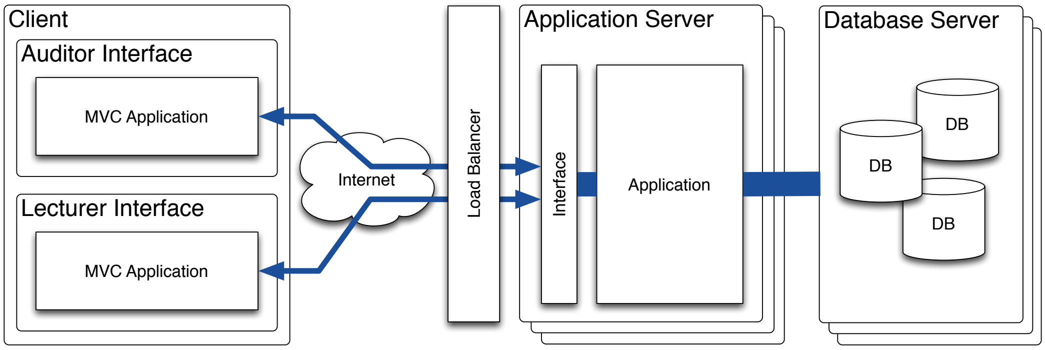


Figure 5.1: Architecture for the new quantitative backchannel system

[Lecture Tools, 2013; MyTU, 2013; NK Labs Inc, 2013] are focusing on binary vote aggregation (for example *confused* and *understood*) a larger result set might be an option to gain more information. The requirement which should be preserved is usability (3). While maximizing the meaningful information can be assumed a useful requirement, proofing that this information is truly meaningful to the system and to the lecturer is the difficult part. This can only be addressed with the evaluation of the system in a real situation. Because the lecturer perspective is not the focus in the first iteration, evaluating this requirement from the lecturer perspective is not addressed in this work. However, the requirement is relevant for the first iteration because the evaluation of other requirements might improve with a larger dataset.

## 5.3 Architecture

The architecture shown in figure 5.1 on page 33 is a classical web service architecture. The architecture is divided into two parts: the client part and the server part. The communication is realized over the Internet. The server side architecture is a generic cloud architecture including a load-balancer, horizontal scaling application servers, and access to horizontal scaling database servers. While this server side architecture is working with one application server and one database server, the technology stack should be chosen with horizontal scaling in mind.

The client side is also generically designed as one or more applications, which implement the auditor interface and the lecturer interface. The Internet is used as the communication channel between the interfaces and the application server.

To fulfill the requirements from section 5.2 some aspects should be considered. The latency from the client side to the server needs to be low, to achieve instant data communication and visualization. Low latency also applies for the database setup. Latency is an important factor of user experience. If the data could not be cached on the client side, the communication channel must be of low latency to support the user experience. For the database system, in-memory systems could be of adequate performance for this requirement. The server side interface should be based on the RESTful [Richardson and Ruby, 2008] paradigm where possible and reasonable.

### 5.3.1 Cross-Platform Approach

Based on the architecture design in figure 5.1 on page 33 and on the requirement 5 and 9 building a system which supports multiple platforms and is ready for BYOD requires several design decisions. Both requirements require a system, which supports a large amount of heterogeneous mobile Internet devices. The decision which cross-platform technology the ARS is based will influence several characteristics of the system and the number of supported mobile platforms. Four main categories of cross-platform approaches for mobile devices could be distinguished: the native approach, the native cross-platform approach, the web-based approach, and the hybrid approach.

#### Native Approach

The native approach is similar to conventional application development on desktop OS. The developed application uses the provided Application Programming Interface (API) directly from the OS to build the application.

To make the application work on a different OS, which can be assumed to have a different API, the application must be adapted or even rewritten to support the other OS API. Considering the different mobile OS, which have a significant market share on mobile devices, the application needs to be rewritten or adapted for at least two widely used mobile OS (iOS and Android) to support 90.6% of the devices sold in Q4 2012 [Gartner, 2012]. Considering the lecturer interface, which might not be suitable for mobile devices, at least the main used desktop OS needs to be considered, which is currently Microsoft Windows [Statcounter.com, 2013]. Assuming in the academic world alternative OS such as OS X and Linux are more widespread than in the global average, additional versions for these OS needs to be developed. The advantages of this approach, while time intensive, are high execution speed and the possibility of using device specific features through the native OS API. The speed improvement originates from the direct execution of the API of the OS. Another factor, which can be assumed to improve performance, is the fact that usually native applications are compiled instead of interpreted.

### Native Cross-platform Approach

There are frameworks which address the writing of the same application code for different OS by abstracting the OS based API. These frameworks introduce an additional layer of abstraction. This approach comes at the cost of reduced flexibility. To support multiple OS API, the framework must map a generic API to the corresponding native API interface. Therefore, the framework can only support the set of common API calls. All OS specific API calls are unusable unless the abstraction framework itself implements their functionality. This is not possible in all cases. The advantage of such frameworks is single API approach. Applications only need to support one API, the generic API provided by the framework. Dependent on the OS where the application is running, the framework library maps the generic API calls to the native API calls. This reduces the development effort while maintaining high performance. While these frameworks such as QT, Marmelade, and Murlengine [Digia, 2013; Ideaworks3D Limited, 2013; Spraylight GmbH, 2013] are providing cross-platform development for the major platforms, fewer platforms are supported compared to the third approach which uses web application to provide cross-platform capabilities.

### Web-based Approach

Writing an application entirely executing in a web browser can support the largest number of devices. Almost every Internet device can be assumed to has a browser installed. The number of supported devices is heavily dependent on the installed browser and the needed functionality of the web application. While the native approach uses the API of the OS, the web application can only rely on the functionality of the web browser.

Comparing the native approach with the web-based approach it can be assumed that the browser has the role of an OS for the web application. The browser provides a type of API to the web application. Unfortunately, there are also many different browsers in use. The fragmentation of browsers is much higher compared to OS Statcounter.com, 2013. However, it can be assumed that most of today's most used modern browsers support a standardized set of features like an API for the application. Using this approach often involves additional JavaScript libraries, which add an additional layer of abstraction to support even more functionality as the standardized set of cross-browser features. With this in mind web applications can be developed using javascript and HTML. The new standard HTML5 [W3C, 2013] has improved the possibilities in developing rich web applications.

While the device support can be stated good, there are other disadvantages compared to other approaches. The performance is heavily dependent on the browser performance. However compared to native applications, which are normally compiled and optimized to the hardware, JavaScript is interpreted in the browser. The browser itself runs as an application in the OS. If JavaScript libraries are used to maximize compatibility, an additional layer is introduced. With every abstraction layer the performance decreases. While browsers, which support HTML5, are providing a large range of functionality to the web application, many tasks especially hardware related features cannot be realized as web applications. Web applications have larger limitations in functionality compared to native applications because of the more restrictive security policies needed in browsers. Therefore, web applications cannot address all kinds of software.

Compared to native applications an advantage of web applications is the deployment. While the deployment cycle for native applications is highly dependent on the end users install-and-update behavior, web applications can be assumed up-to-date instantly. There is no end user interaction necessary to update a web application. The browser loads the recent version of the application whenever the user is browsing to the web application (dependent of the browsers caching settings). Recent mobile OS also support automatic updates of native applications when the application is distributed over the OS application deployment system.

Another difference from native application is the user interface. This can be stated as an advantage or disadvantage depending on the perspective. While native applications can use native UI elements, web applications can only use HTML markup to display the user interface. This is especially relevant for the user experience. If the user experience is tightly coupled with the native UI of the OS, a web application might not be reasonable. If the user is used to address this task via a web-based interface, the user experience might not be reduced by a web application.

As already explained for the UI one clear difference of web applications is the missing integration in the OS. While this is relevant for the UI at first sight, it is also noticeable for the user when the application needs to be started. While native applications are started intuitively through the OS, web applications needs to be started in the browser by browsing to the URL of the application. While the latter can be slightly simplified using bookmarks, the user experience is still different, and this introduces an additional acceptance barrier.

While these limitations and disadvantages exist, a broad range of applications is using the web-based approach. In many cases, the advantages outpace the disadvantages. This can be assumed the case when

- the applications requirements can be realized with the given functionality by recent web browsers,
- the target user group is using recent web browsers,

- the target user groups user experience is not negatively influenced by the shortcomings of web application UIs, and
- the target user group is comfortable to open the browser to access the application.

Under these conditions, web applications are a reasonable choice and can be stated to support the widest range of devices compared to other cross-platform solutions.

#### Hybrid Approach

The hybrid approach tries to combine the native cross-platform approach with the web application approach. The hybrid approach wraps a web application in a native application container. This lets the web application access the native API calls of the OS through the native application container. Assuming a web application exists; a wide range of devices is already supported. However, the typical disadvantages of a web application exist. Using hybrid frameworks such as Phonegap [Adobe Systems, 2013] can make a native application container with the web application inside. For end-users, such an application is well integrated in the OS like any other native application. It is installed and started like native applications. Although the web application usually cannot access the native OS API, when packaged in a native application container the framework can provide native API calls to the web application. However, the interface is still limited to HTML the same way as it is for web applications, native UI elements are still not available in this approach. Hybrid applications can only run on OS which are supported by the hybrid framework. To provide a larger platform coverage, a conventional web application can be provided as a fallback for the unsupported OS. For example, Phonegap can deploy to iOS, Android, Windows Phone, Blackberry, WebOS and Symbian. One disadvantage besides the lack of native UI elements is the performance. From user experience side, responsiveness is noticeably slower than in pure native applications. This can be argued by the additional layer, which is introduced by the framework. The native application container generated by the framework is a native specialized web browser with the web application included.

#### Conclusion

Evaluating these approaches based on the requirements. The web application approach was chosen with the hybrid approach as an option for further development. Reasons for the decision of the web application approach can be summarized as following.

Regarding the cross-platform requirement (9) many different devices and platforms should be supported. A high device support should lead to a potential large user base to collect enough data for evaluation of the system. Based on the evaluation of the approaches it can be assumed that the web application approach can address the most number of different devices and platforms compared to other approaches. By carefully designing the architecture of the system and the use of JavaScript libraries, the browser support could be further increased. While the performance trade-off might be an issue, it can further be assumed that feedback will indicate if the possibly poor performance will conquer the user experience. Requirement (5) BYOD could also be satisfied with this approach. The web application does not need to be installed. Concerning the responsive design requirement 6 for the web application approach, responsive design is already a well tested approach and several libraries and framework already supports it with less coding overhead [Twitter Bootstrap, 2013; JQuery Foundation, 2013]

### 5.3.2 Technology Stack

The technology stack is discussed based on the architecture in figure 5.1 on page 33 and based on the decision from section 5.3.1.

Beginning on the server side the application server provides a pseudo RESTful server interface [Richardson and Ruby, 2008]. The server interacts with the persistence layer, which is provided by several socket-based databases (Redis, MongoDB) to be prepared for horizontal scaling on future iterations. The server handles the whole communication from the audience and the lecturer. The server itself is realized as a Web Server Gateway

Interface (WSGI) server. The whole server architecture is designed to support dynamic horizontal scaling. Therefore, the first application server and database server are also running in the cloud. The application server itself is implemented in python using the pyramid framework [Pylons Project, 2013].

The client side of the backchannel system consists of the lecturer interface and the auditor interface. Both are realized as HTML5 front-ends with fully integrated responsive design. The design is optimized for desktop view as well as optimized for mobile devices. The optimizations are regarding UI and performance. Especially on the mobile platforms performance and responsiveness are still a large factor for user experience.

The client side business logic is implemented using a Model View Controller (MVC) [Krasner and Pope, 1988] JavaScript framework called AngularJS [Google, 2013b]. AngularJS uses a new approach for JavaScript frameworks which addresses innovative ways of improving modularization, separation of concerns, testability, HTML templating, etc.

The real-time interaction between auditors and lecturer is implemented using HTML5 WebSockets. In the RealFeedback project WebSockets were not used, instead a pure RESTful interface combined with polling was used. For this use-case, a RESTful interface combined with a relative high polling rate is an adequate approach. The problem that arose where due to many single asynchronous REST calls. The capturing of edge conditions resulting from timing was getting quite complex on the large number of parallel requests.

This is the reason HTML5 WebSockets are integrated for the near real-time communication in the new backchannel system from the beginning. A wrapper library is used for implementing WebSocket communication because of the lack of WebSockets support in older browsers. Socket.IO provides several fallback mechanisms like long polling and flash sockets. To support Socket.IO on the server side, gevent-socket.io is used. Gevent-socket.io is a greenlet based non-blocking asynchronous library, which implements the Socket.IO protocol. Using the gevent-socket.io library a gunicorn server is used to serve the application. This gunicorn [Chesneau, 2013] server was hidden behind a Nginx [Sysoev, 2013] instance which is responsible for serving all conventional Hypertext Transfer Protocol (HTTP) requests including the static client side HTML5. Nginx was installed behind varnish [Varnish Software, 2013] an HTTP accelerator which redirects all WebSocket requests directly to gunicorn and other requests to nginx. The redirecting of WebSocket calls is necessary because nginx does not support WebSockets in the current stable version.

While WebSockets provide an instant communication channel which is both scaling and performing well, the server needs to manage the data, which are sent through the WebSockets. Depending on the audience size and the number of simultaneous held lectures, the data management on the server can be a performance bottleneck. To address this issue, the publish-subscribe pattern is used which is implemented in the redis database server. The publish-subscribe pattern itself is not a performance improvement but a complexity reduction by decoupling publisher and subscriber regarding requirement 4 of keeping the system simple.

The publish-subscribe pattern states that the sender of a message needs not to know who is the receiver of the message. Receivers are subscribing to channels. Receivers are therefore called subscribers in the pattern. Publishers are publishing messages to channels, and subscribers can subscribe to these channels. There even might be no subscriber on a channel or there may be multiple subscribers on one channel. The publisher does not care about the number of subscribers. With this pattern, publisher logic can be separated from subscriber logic, which reduces complexity and foster flexibility.

Combining this pattern with a high performance in-memory database like Redis will give the needed performance for the near real-time communication between audience and lecturer. Redis is an in-memory key/value store which is in the category of NoSQL databases [Strauch, 2011]. Redis can optionally be made persistent by writing a change history to a file on a persistent medium such as a hard disk.

### 5.3.3 User Interface Design

Starting on the main page there should be no need to sign-up or to register. The easiest possible way with as less interaction as possible, should

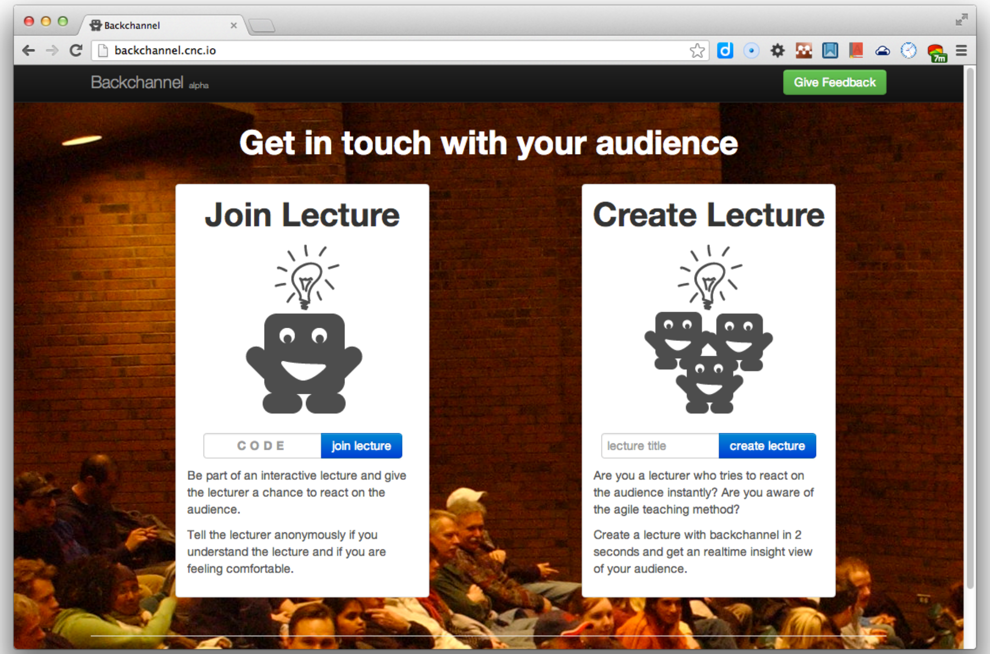


Figure 5.2: Screenshot of the implemented main page for the first iteration

be implemented to create a backchannel session. Every barrier that might be a reason to stop the process of creating a backchannel session should be omitted. Regarding requirement 4 no options and settings should be needed or made available which are unnecessary to fulfill the most basic use case. Additional settings for advanced users should not be irritating the user who only wants to fulfill the default use case. A similar principle should be applied to the lecturer and to the auditor use case. The implementation of the main page of the web application is shown in figure 5.2 on page 5.2.

For the lecturer interface, special attention should be paid to the circumstance that the lecturer cannot pay much attention to the backchannel screen. Considering the requirement 2 the information provided on the lecturer interface must be clearly stated and easy to recognize even when only seen peripheral. Another challenge is the currentness of the presented data. However, lecturers should not get distracted with data, which are not relevant for them. The data displayed to the audience, and the data displayed to the lecturer need to be prepared and presented differently. Although this will not be implemented in the first iteration, it should be argued for future work. It is necessary for the audience to see the influence of their interaction (regarding requirement 7) and the activity of other auditors. For the lecturer, that might be distracting. Data needs to be filtered by relevance for the lecturer. Only if relevant changes happen in the audience perception the lecturer should be informed clearly and non distracting regarding requirement 8.

There are two actors in the system the lecturer and the auditor. The lecturer wants to create a backchannel session for the lecture where the audience could participate.

The lecturer workflow is as following.

1. Lecturer browses to the backchannel web application
2. Lecturer creates session
3. Lecturer invites audience
4. Lecturer observes audience backchannel and tries to react accordingly

The auditor workflow is as following.

1. Auditor browses to the backchannel web application
2. Auditor accepts the invitation and joins the session
3. Auditor gives feedback via the backchannel web application

The invite is realized with a code the lecturer sees when the session is created. This 5-letter code needs to be communicated to the audience. The audience can join the session by entering the session code in the auditor interface. After joining the session, the auditor is connected to the session of the lecturer. The auditor can now start to vote, and the lecturer will see the collective perception of the audience instantly.

## 5.4 Quantitative Feedback Dimensions

A quantitative feedback dimension represents a type of auditor feedback. For example, dimensions in the state of the art evaluation of quantitative backchannel systems were *comprehension* and *speed*. Auditors can vote for one or multiple dimensions during the lecture by interacting with the backchannel system. For example, when an auditor presses the *confused* button she votes in the comprehension dimension. The same dimension would be addressed when the auditor presses the *understood* button.

Dimensions in the context of quantitative backchannel systems are the types of information, which can be transmitted from the audience to the lecturer through the system. The backchannel system can further filter and process this information to give a more meaningful representation to the lecturer.

Implementing multiple dimensions in the system can bring more information to process, which can lead to a better understanding of the current audience perception. It can also be assumed that if the audience has too much effort to enter the data the overall participation might decrease. This leads to fewer data. Less information can be extracted from fewer data, which might lead to less meaningful information. Therefore, fewer dimensions can lead to getting more data. Therefore deciding which dimensions are collected and how many dimensions is crucial and will influence the whole system.

Based on the evaluation of the state of the art systems the following criteria are derived, which quantitative backchannel dimensions should fulfill.

**Criterion 1** *Understandable to the auditor*

**Criterion 2** *Meaningful to the lecturer*

**Criterion 3** *Clear extremums*

**Criterion 4** *Values should be expectable to change over the lecture*

Based on these criteria three dimensions for the new backchannel system were chosen.

- Happiness
- Comprehension
- Presentation Speed

Happiness is a new dimension which none of the evaluated systems collected. With this additional dimension, the system might add more context to the other two dimensions in future iterations. Comprehension and speed related values are not necessarily related to the happiness of an auditor. For example, an auditor might appreciate that the lecturer presentation speed is slow and stays happy. An auditor might be unhappy even if the speed is ok and the auditor understood the topic, but maybe the handwriting on the blackboard is too small.

Comprehension is used in all evaluated systems in section 4. The comprehension dimension should inform the lecturer if the topic is understood or if the lecturer might need to recap some parts.

The speed dimension is the third dimension it is also in use in the MyTU system [MyTU, 2013] presented in section 4. The speed dimension should indicate the presentation speed. The extremums of this dimension are "slow down presentation speed" and "speed up presentation". There is also a need for a neutral value that states that the presentation speed is convenient.

## 5.5 Auditor Interface

The auditor interface needs to satisfy several requirements. Besides the technical requirements (see requirements 5, 6, 9) which are addressed in section 5.3.1 on page 34 there are other requirements, which need to be considered.

Usability (requirement 3) is a requirement which addresses the whole system and should also be considered for the auditor interface. However, there are three requirements, which are of significant importance for the auditor's interface.

- Auditor impact (Requirement 7)
- Maximize the collection of meaningful information (Requirement 11)
- Continuous interaction (Requirement 1)

### 5.5.1 Maximize the Collection of Meaningful Information

Given the three dimensions *happiness*, *comprehension*, and *speed*, which were evaluated in section 5.4, the system needs to collect the maximum possible information of each dimension from the auditor. Similar to Understoodit [NK Labs Inc, 2013] the time information could be taken into account. To provide the lecturer with meaningful information, time can be assumed as an important factor. Every vote of any dimension will be tagged with a timestamp by the system for further filtering and processing.

Another property, which is taken into account by the system, is the auditor. While the system should be easily accessible in turns of usability, regarding the requirement (3) usability, no registration is necessary. However, the system registers the auditor automatically. Auditors are registered only with a random ID and no further information of the auditor is transferred during registration. This approach improves the relevance of votes, which can be related to auditors. Different information can be concluded if the system can differ if one auditor votes multiple times or multiple auditors vote once each. Further, the system can interpret activity on a per auditor base. This information can only be extracted if the system maintains auditors. The auditor interface web application maintains the `auditor_id` in the browser cookies. The server system saves the `auditor_id` to every vote from an auditor. Auditors are preserved as long as the browser cookies are not deleted. The registration process is as following.

1. If no `auditor_id` is present or `auditor_id` is unknown, then create a new auditor and send it to the auditor
2. Sign in with the `auditor_id`

While the introduction of auditors to the system enriches the vote data, additional information regarding the auditor can be retrieved. Even if auditors never vote, significant information of them can be extracted by the system. The number of auditors who are accessing the web application may be different to the auditors who are actively voting. If the system only captures votes with the `auditor_id`, the system only gets the number of active voting auditors. This might not be an adequate foundation for generating information for the lecturer. For example, imagine a situation where most of the auditors are so excited about the lecture that they forget to vote for their excitement. In the same time, a few are unhappy and vote their mood. When the system sees only the auditors, which are active in such a situation, the system will interpret the data wrong. The system only sees that most of the audience is unhappy because the foundation for the audience perception is only the active auditors. This interpretation of the situation by the system is wrong. To prevent this case auditor connections are also recorded. The time is recorded when an auditor joins a lecture session. When the auditor closes the browser or switches off the smartphone, the auditor is disconnected from the session automatically. With this method, the system can also count auditors who are only passively watching the backchannel without further user interaction.

### Maximizing Dimension Resolution

There are different ways of entering quantitative information to the backchannel system. While the evaluated systems are all using buttons to collect specified predefined values of the dimensions the new system tries to improve the resolution of the dimensions. For example, the buttons in Understoodit [NK Labs Inc, 2013] represents the states *confused* and *understood*. Even though, confused auditors might not admit themselves that they are confused about the lecture. Eventually, they are more willing to say that they have not understood it to 100% instead of admitting that they are confused. The system should address this possible case by collecting intermediate values. The live test evaluates this assumption by counting the number of votes, which have intermediate values.

Buttons, as used in the evaluated systems, are good for predefined actions

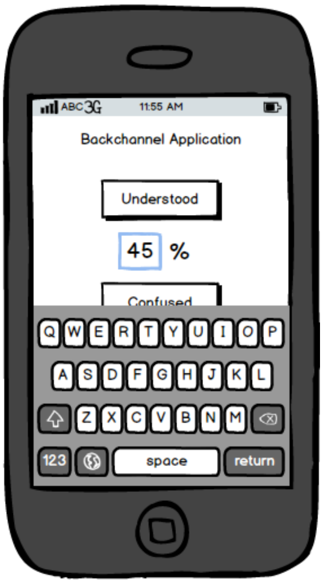


Figure 5.3: Evaluation of the text input element for the auditor to fulfill the usability requirements

or in this case dimension values. A different UI element is needed which provides a higher dimension resolution to support intermediate voting values.

**Text Input** A first approach to maximize the resolution would be a text input element (see figure 5.3 on page 48). With the text input element auditors can enter their confusion percentage directly. However, entering a value in a text input element would be cumbersome especially on a touch device with an onscreen keyboard. This would violate the requirements for usability (3) and BYOD (5).

**Buttons** A list of different value buttons (see figure 5.4 on page 50) is evaluated. Buttons have the advantage that they are fast to activate. Every button has a different predefined value, which increases the resolution compared to the two-button approach. Although the number of buttons, which can be displayed in a proper way, limits the resolution. However, buttons with *100% confused*, *50% confused*, *10% confused* and *understood* might be reasonable. The problem, which occurs here, is the asymmetric presentation of the dimension, which might influence the auditor in the voting.

To address this problem, the multi-button approach can be used with a different distribution. For example *100% confused*, *50% confused*, *50% understood* and *100% understood*. The dimension is now symmetrically presented. Nevertheless, it cannot be assumed that it is easy to decide for the auditor if the comprehension is *50% confused* or *50% understood*. Possibly these two buttons might be interpreted as similar by the auditor. This can lead to confusion of the auditor, which is not acceptable regarding the usability requirement (3). Instead of buttons, drop down element or a list element can be used. Nevertheless, all of these UI elements will struggle with the same disadvantages in this use case.

**Numerical Stepper** Another UI element, which should be discussed, is the numerical stepper also known as spinner (see figure 5.5 on page 51). While it was considered a text input is to distracting to use, a numerical stepper can be a compromise. It changes its value by clicking on buttons labeled with an up and down arrow. Additionally, the value could be edited directly like on a text input element. For example, every click on the *up* button will increase the value by one step. With one numerical stepper, the auditor could set the percentage for the vote. Two buttons, *confused* and *understood*, will send the percentage from the numerical stepper and the direction of the comprehension. Values can be achieved from *100% confused* to *100% understood* with a resolution for example of 1%. This would maximize the resolution to 200 individual values in one dimension. It needs to be evaluated in a real situation application of the system if this high resolution is reasonable. According to the requirement (11) "maximizing meaningful data collected from the auditor" such an approach can be suitable. From the usability perspective (3) this approach might not be intuitive to use.

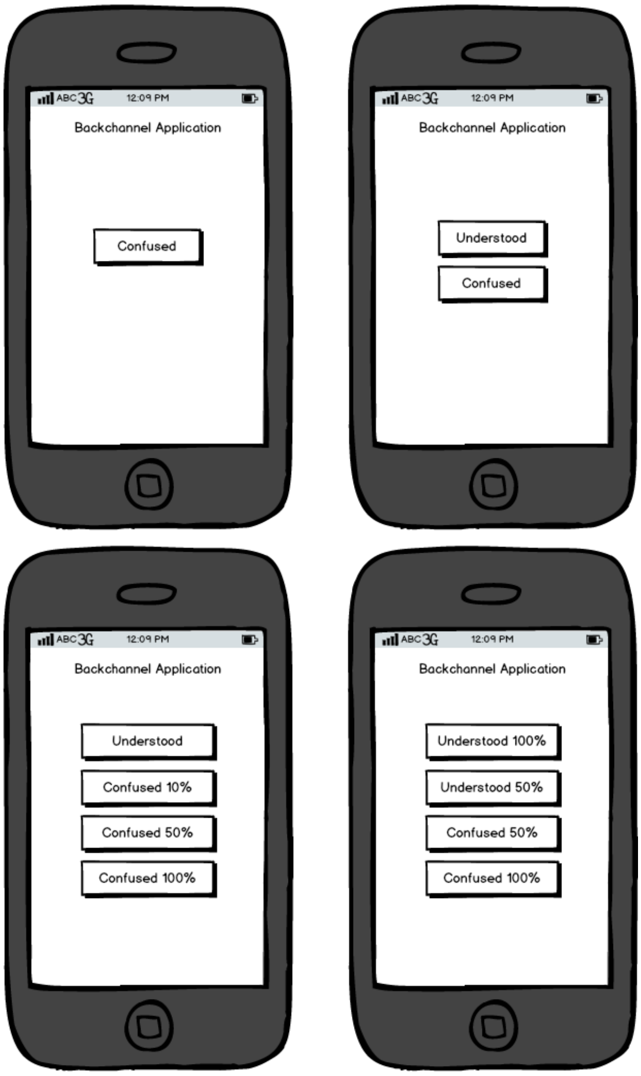


Figure 5.4: Evaluation of button input elements for the auditor to fulfill the usability requirements

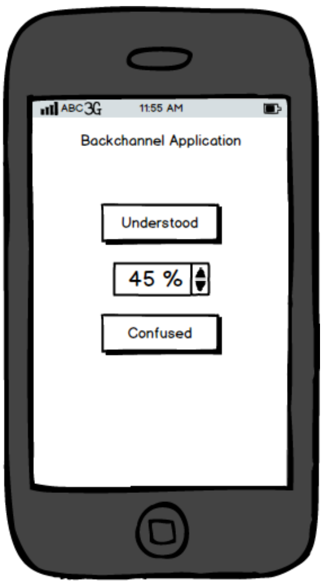


Figure 5.5: Evaluation of the numerical stepper input element for the auditor to fulfill the usability requirements

**Slider** Slider elements are used to indicate a value in a given range (see figure 5.6 on page 53). The indicator element on the slider bar can be moved by the user while dragging it with a left mouse click or by touch. While this element supports fast changes by the auditor, accurate values are more difficult to set (depending on the resolution of the slider) compared to a numerical stepper. A numerical stepper increments or decrements exactly one step on each click. A slider indicator can be moved arbitrarily over the full range with one user interaction. An advantage of the slider is the possibility to react on the event when the slider indicator is dropped. From the usability side, this is the moment where the user has finished the positioning process of the indicator and is satisfied with the value of the slider. Using this assumption no additional buttons are needed. The vote with the value and the direction of the comprehension can be sent when the auditor drops the indicator element. This also implies that swaying of the slider without dropping the indicator will not influence the voting. A slider fulfills the usability requirements assuming the system does not need a high precision input element.

### Conclusion and Implementation

The different UI element variants are evaluated in table 5.1 on page 54 regarding resolution, number of needed UI elements to fulfill a vote, number of interactions needed to send a vote with 100% *confused* value, and the number of interactions needed to send a vote with 50% *confused*.

"Resolution in steps" means how many different states can be created. The number of needed elements to prepare a vote is counted as the number of different UI elements the user must interact with. The number of interactions for a 100% *confused* state is counted as the number of clicks and keystrokes an auditor needs to apply to send a vote with the requested values.

[1] Numerical steppers allow the adjustment of the values by clicks and by entering the value directly. The first value in the table cell is the number of interactions needed to adjust the value by incrementing the stepper with clicks. The second value is the number of interactions by directly entering the value in the numerical stepper.

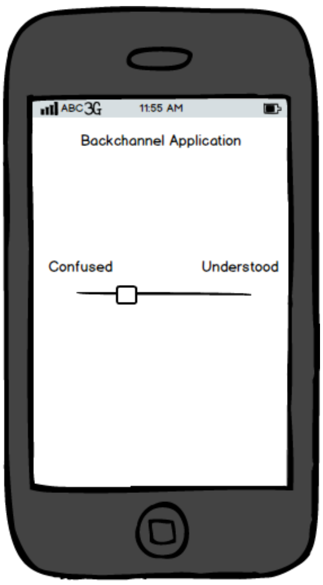


Figure 5.6: Evaluation of the slider element for the auditor to fulfill the usability requirements

| UI Element Variant | resolution in steps | # of needed elements | # interactions for 100% confused | # interactions for 50% confused |
|---|---|---|---|---|
| Text Input with 2 Buttons | 200 | 3 | 4 | 3 |
| 2 Buttons (*confused* & *understood*) | 2 | 2 | 1 | 1 |
| 4 Buttons (*understood* and *confused* in 50% and 100%) | 4 | 4 | 1 | 1 |
| Numerical Stepper with 2 Buttons 1% Resolution | 200 | 3 | 101 / 4[1] | 51 / 3[1] |
| Slider with 1% Resolution in Each Direction | 200 | 1 | 1 | 1 |

Table 5.1: Key parameters of the UI element variations

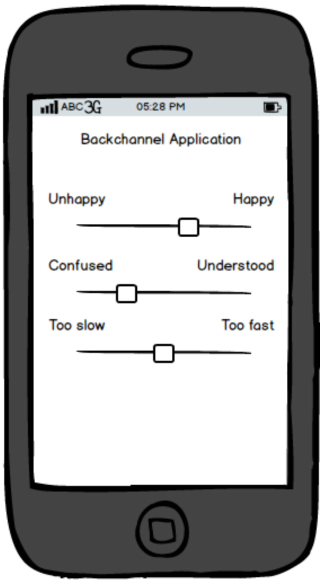


Figure 5.7: Mockup of the dimensions *happiness*, *comprehension* and *presentation speed* using sliders to maximize the collected information from the auditor

Comparing the approaches for the auditor UI element considering the requirements *continoues interaction* 1, *non-distraction* 2, *usability* 3, *simplicity* 4, *BYOD* 5, and the *maximization of the meaningful information* 11 the slider approach is chosen. Combining the slider approach with the chosen backchannel dimensions from section 5.4 on page 44 the mockup looks like figure 5.7 on page 55.

To implement this approach, a JavaScript library *jQuery UI* [JQuery, 2013] for cross-browser UI elements is used. This library offers a slider element which is suitable for the new backchannel application, see listing 5.1 on page 56. To provide extended mobile touch support, the jQuery UI Touch Punch [Furfero, 2013] JavaScript library is used.

Listing 5.1: jQuery UI Slider Element

```
<script>
        $(function() {
                $( "#slider" ).slider();
        });
</script>
```

Based on this code an AngularJS directive is needed to add slider elements to the AngularJS application. Therefore, the call to `slider()` needs to be wrapped in an AngularJS directive to keep the separation from business logic and Document Object Model (DOM) manipulation.

Listing 5.2: Angularjs jQuery UI slider directive

```
app.directive('slider', function() {
    return {
        restrict:'E',
        scope: {
          ngModel: '=',
          onChanged: '&'
        },
        replace: true,
        template:'<div></div>',
        require: 'ngModel',
        link:function(scope,element,attrs){
           //watch the ngModel to set slider when val in ngModel var changes
           scope.$watch('ngModel', function(newVal, oldVal){
             //check when ngModel is not initialized
              if (newVal !== undefined){
               element.slider("value", parseInt(newVal,10));
             }
           });
           //create jQuery UI Slider
           element.slider({
              min: parseInt(attrs.min,10),
              max: parseInt(attrs.max, 10),
              value: scope.ngModel,
              step: parseInt(attrs.step, 10)
            });

            //bind the slide function to update the ngModel
            element.bind( "slide",function( event, ui ) {
              scope.ngModel = ui.value;
              scope.$apply();
```

Figure 5.8: The AngularJS slider directive rendered in the browser

```
            });

            //execute onChanged function when slider value is changed
            //(not executed during dragging the slider)
            if ('onChanged' in attrs){
              element.bind( "slidechange", function(event, ui){
                  scope.onChanged()(ui.value);
              });
            }
        }
    };
});
```

This directive is used in the HTML code as following.

Listing 5.3: Angularjs jQuery UI slider directive

```
<slider ng-model="sliderValue" on-changed="slideChangedFunction" max="100" min="0"></slider>
```

The slider directive is realized to support the needed event as discussed in the section 5.5.1. The function `slideChangedFunction` is called when the slider indicator is dropped. This slider directive is rendered in a browser as shown in figure 5.8 on page 57.

## 5.5.2 Continuous Interaction

The requirement to aspire continuous interaction (1) by the auditor is addressed by using the Tamagotchi effect [Holzinger et al., 2001] on the auditor interface. It is assumed that if the auditor interface can somehow establish a relationship between the auditor and the interface, the auditor might feel responsible for the interface and might actively maintains it through participating in the backchannel during the whole lecture.

For the auditor interface, an interactive avatar[1] is developed to address the Tamagotchi effect. This avatar, which is an interactive image on the auditor interface, symbolically illustrates the auditor. This interactive image is a stickman with a large face to express facial expressions clearly. This avatar should represent the auditor and the auditor's current mood. If the mood of the auditor changes, the avatars mood should be changed accordingly. For example, if the auditor is satisfied with the lecture style the avatar should also be in the happy state and changes its facial expression to smile. Adjusting the avatar to the personal feelings of the auditor should be motivated by the Tamagotchi effect. The auditor can change the avatars facial expressions by voting to the backchannel. For example, if the slider of the happiness dimension is moved to the right the avatar is smiling. If the happiness slider is moved to the middle, the avatar should look like neutral. Moving the happiness slider far left meaning unhappy will make the avatar frown as shown on figure 5.9 on page 59.

The avatar as shown in the mockups in figure 5.9 can only illustrates three different states. This might not be enough to let the auditor think that all the different values in between have a meaning. To address this potential problem, five different states per dimension were created. Five states are chosen to improve the resolution while not confusing the auditor with too many different illustrations. While there are only five different states, the votes, which are sent to the server, are including the exact value with a resolution of 200 steps per dimension.

Assuming the dimension happiness can be illustrated with a smile and a frown as facial expressions two other dimensions need to be visualized via the avatar simultaneously. Each dimension has five states. Every state of every dimension can be combined. This results in $5 \times 5 \times 5 = 125$ different combinations of states. To optimize the visual appearance, 125 images need to be designed. Each avatar image represents a specific combination of the three dimensions via facial expressions. The manual design of 125 different images is not assumed to be an efficient way to generate 125 states. Another approach which was finally chosen, is the splitting up of the dimensions in separate parts of facial expressions. These facial expressions must not

[1] "an icon or figure representing a particular person in a computer game, Internet forum, etc." [Oxford University Press, 2013]

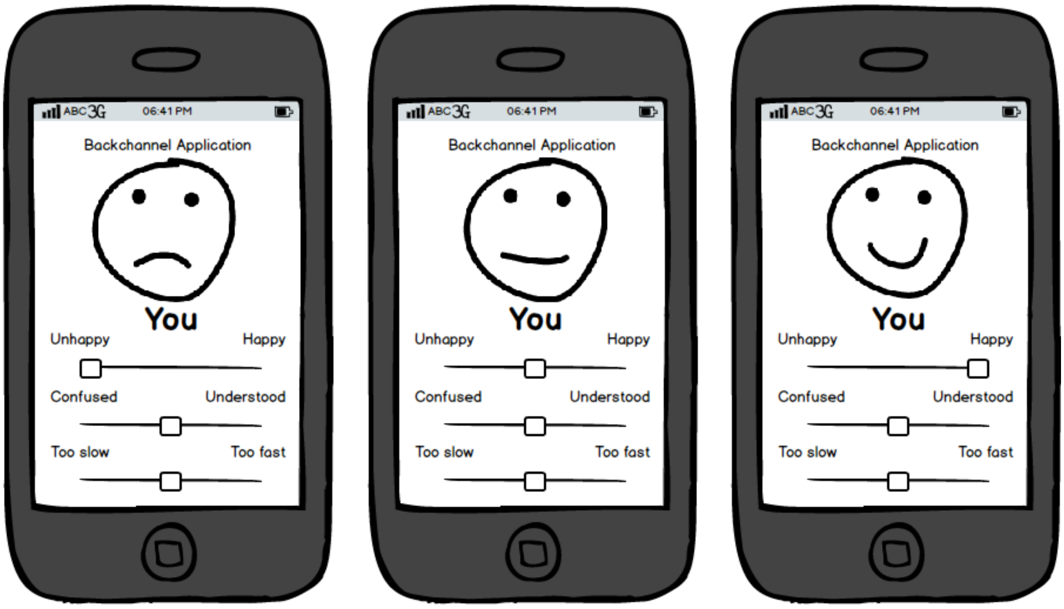


Figure 5.9: Illustrating the effect, changes of the happiness slider have to the avatar

significantly influence each other. For example, the happiness dimension focuses on changes to the mouth of the avatar. The comprehension dimension let question marks appear over the head of the avatar when the auditor votes confused, and will show enlightening light bulbs over the head when the auditor votes understood. The speed dimension will focus on the eyes of the avatar. The eyes are closing symbolizing falling asleep when the presentation speed is too slow and looking overstrained when the speed is too fast. By separating these dimensions in the avatar visualization, the 125 different visualizations can be generated by combining only 15 different images (three dimensions with 5 states each).

The design of the avatar followed the principle of simplicity and easy to understand facial expressions. While the face is the dominant part of the avatar, the arms and legs are added to support the illusion of a complete being. Figure 5.10 on page 60 illustrates the happiness dimension of the avatar from left to right from *unhappy* to *happy*. The happiness dimension focuses on the mouth and uses the arms to support the emotional state.

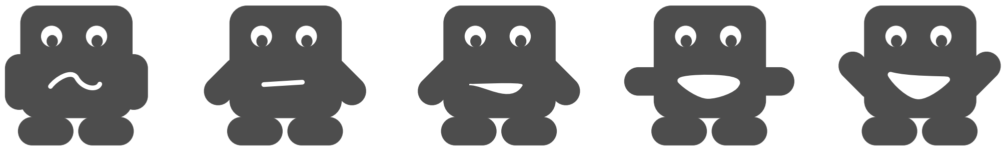

Figure 5.10: Illustrating the happiness dimension while the other dimensions are neutral

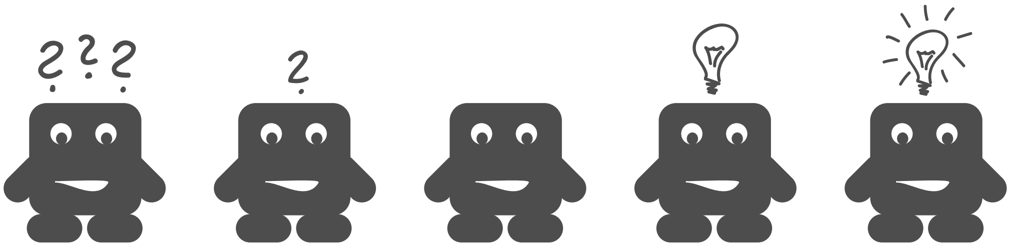

Figure 5.11: Illustrating the comprehension dimension while the other dimensions are neutral

Figure 5.11 on page 60 illustrates the comprehension dimension from left to right from *confused* to *understood*. This dimension is using question marks and light bulbs over the head of the avatar to communicate comprehension. Figure 5.12 on page 61 illustrates the presentation speed dimension from left to right from *too slow* to *too fast*. This dimension uses sleeping eyes on the *too slow* direction. On the *too fast* direction, the impression of overstraining should be illustrated with eyes rolling, sweating, and drooling.

The emotional states are intentionally exaggerated illustrated to give also the negative direction a humorous touch. This should encourage the auditor to vote even if the voting value is usually negatively afflicted.

The implementation in the web application is based on Cascading Style Sheets (CSS) Sprites [CSS Tricks, 2013]. The avatar illustrations are separated by their dimensions and combined in one image (see figure 5.13 on page 61), which will be the image for the CSS Sprite. In the browser only the relevant section from the sprite image will be shown, the other parts are

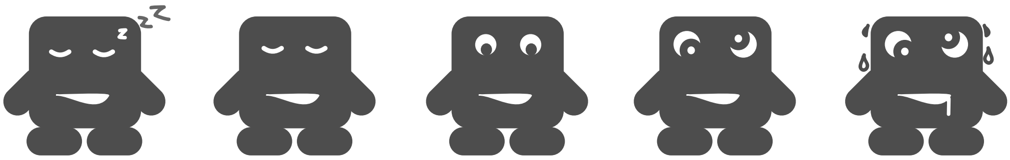

Figure 5.12: Illustrating the presentation speed dimension while the other dimensions are neutral

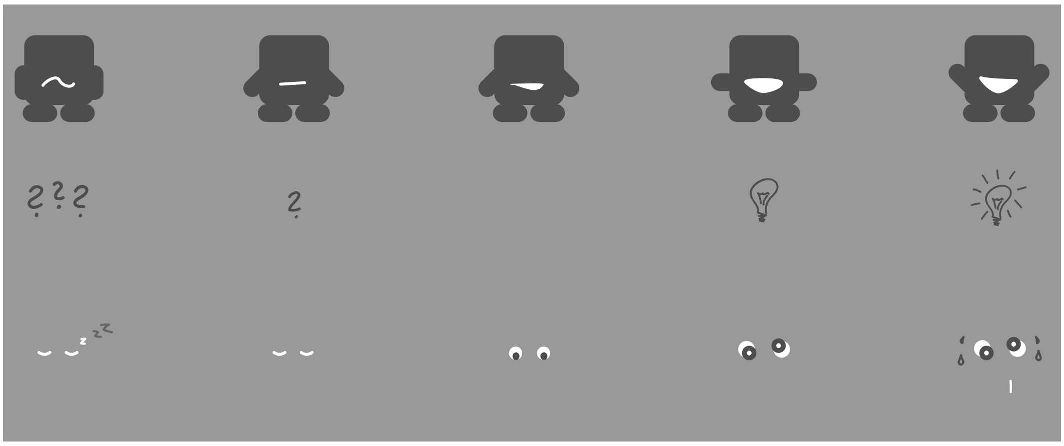

Figure 5.13: Illustrating the three dimensions of facial expression separation as image sprites

invisible. By moving one of the sliders the image behind a DIV[2] element, which represents a rectangular clipping box, is moved. Therefore different parts of the images appear in the DIV element when moving the slider. The actual moving of the image is not visible. To the auditor it appears as if there is a new image, which replaced the old one. By combining this technique with the overlay of images, 125 different avatar illustrations can be simulated with only 15 base images. According to AngularJS, a directive (see listing 5.4) is written for the needed DOM manipulation.

[2]DIV is a HTML element used to structure HTML content

Listing 5.4: Angularjs avatar directive using CSS sprites and image overlay to generate 125 different images based on 15 image elements

```
app.directive('avatar',function () {
        return {
                restrict: 'E',
                scope: {
                        happiness: '=',
                        comprehension: '=',
                        speed: '='
                },
                replace: true,
                template: '<div class="avatar">' +
                        '<div class="avatar-img layer1"></div>' +
                        '<div class="avatar-img layer2"></div>' +
                        '<div class="avatar-img layer3"></div></div>',
                link: function (scope, element, attrs) {
                        var layer1 = $(element).find('.layer1');
                        var layer2 = $(element).find('.layer2');
                        var layer3 = $(element).find('.layer3');
                        var max = parseInt(attrs.max, 10) || 100;
                        var min = parseInt(attrs.min, 10) || -100;
                        var range = max - min;
                        var num_steps = parseInt(attrs.numSteps, 10)
                            || 5;
                        var x = Math.ceil(num_steps / 2);
                        var y = Math.ceil(num_steps / 2);
                        var z = Math.ceil(num_steps / 2);
                        var icon_width = parseFloat(layer1.css('width'
                            ));
                        var icon_height = parseFloat(layer1.css('
                            height'));

                        function setAvatar() {
                                layer1.css('background-position', -((x
                                     - 1) * 2 * icon_width) + 'px 0px'
                                    );
                                layer2.css('background-position', -((y
                                     - 1) * 2 * icon_width) + 'px ' +
                                    -icon_height * 2 + 'px');
                                layer3.css('background-position', -((z
                                     - 1) * 2 * icon_width) + 'px ' +
                                    -icon_height * 4 + 'px');

                        }
                        function interpolateImg(value) {
                                var interval = range / (num_steps - 1)
                                    ;
```

```
                                var res = Math.ceil((value - min - (interval / 2)) / interval) + 1;
                                return res;
                        }
                        scope.$watch('happiness', function (newVal, oldVal) {
                                if (newVal !== undefined) {
                                        x = interpolateImg(newVal);
                                        setAvatar();
                                }
                        });
                        scope.$watch('comprehension', function (newVal, oldVal) {
                                if (newVal !== undefined) {
                                        y = interpolateImg(newVal);
                                        setAvatar();
                                }
                        });
                        scope.$watch('speed', function (newVal, oldVal) {
                                if (newVal !== undefined) {
                                        z = interpolateImg(newVal);
                                        setAvatar();
                                }
                        });
                }
        };
});
```

While the directive code might seem complex the actual use of the avatar is one line in the HTML code, see listing 5.5.

Listing 5.5: Angularjs avatar directive used in HTML

```
<avatar class="my-avatar hidden-phone" happiness="myVotes.happiness"
    comprehension="myVotes.comprehension" speed="myVotes.speed"
    num_steps="5"></avatar>
```

### 5.5.3 Internationalization

An advantage for the new backchannel system regarding internationalization is the relatively simple interface for the auditor and the lecturer. There is not much information, which must be transferred textually because all

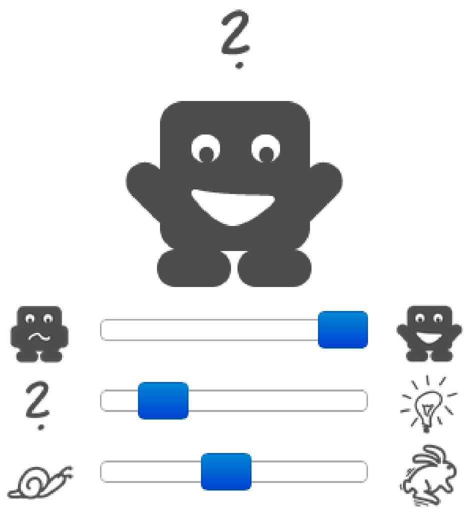

Figure 5.14: Internationalization of the slider legends by using symbols instead of text

communicated data are of quantitative type. This also reduces the work needed for internationalization. For the first iteration internationalization is considered on the core interaction elements for the auditor and the lecturer. Even though, internationalization is considered in the first iteration for the whole system, no localization is made in the first iteration. All text elements are in English to address a potential international user base.

The avatar in the system is communicating emotional states visually through an illustration. This can be assumed internationally understandable assuming cultural differences will not lead to different interpretations of the avatar. Therefore, the avatar element as a core UI element can be assumed internationalized. Based on this assumption the dimension legend could also be communicated visually instead of textually. Assuming sliders are intuitive to handle for users of mobile Internet devices, all core interaction elements for the auditor are internationalized by using images see figure 5.14 on page 64.

The symbols for the legend of the dimensions *happiness* and *comprehension* are used from the avatar visualizations. The presentation speed dimension was not considered easily understandable by extracting the extremum facial expressions for this dimension from the avatar. The issue with the speed dimension is the perspective, which can inverse the understanding of the dimension extremums. For example, the jumping rabbit should symbolize *too fast*. However, it could also be understood as faster. The same applies for the snail. Although this issue was already known during implementation, these symbols are implemented. These misunderstandable legend symbols might result in interesting feedback during testing and might lead to improved symbols for future iterations.

### 5.5.4 Calculating the Audience Perception

The audience perception is the overall emotional state of the audience, which should be prepared for presentation to the lecturer, and the audience. It might not be adequate to choose the same preparation and presentation for the lecturer and the audience. However, this approach was chosen because of the focus on the audience in the first iteration regarding the requirements in section 5.2.

The dataset, which is available for the preparation, is consisting of the auditor votes (see table 5.2) and the auditor connection state. The first approach that was implemented is the calculation of the arithmetic mean. The arithmetic mean is calculated for every dimension separately. The input values for the calculation are the dimension values from the last vote of the auditors. Therefore, only the last vote per dimension of an auditor is in the calculation of the collective perception. Dimensions are separated and do not influence each other. The result of this approach consists of one average value per dimension and per point in time.

This approach is chosen because a simple approach might be more understandable to the auditors when they are testing how much impact they can achieve by voting. This approach does not introduce aging like UnderstoodIt [NK Labs Inc, 2013] uses it. This representation assumes that auditors take care of their states and changes the sliders over time according to their personal perception.

| name | description | type |
|---|---|---|
| timestamp | unix timestamp | float |
| value | between -100 and 100 | integer |
| type | dimension | string |
| auditor id | unique id of the auditor (UUID4) | string |
| lecture id | unique id of the lecture (UUID4) | string |

Table 5.2: Data included in every vote by the auditor

This averaged data per dimension is sent back to the auditor interface of each connected auditor to be visualized. This averaged data is also sent to the lecturer interface. Because the lecturer interface is not the focus in the first iteration, no additional data preparation was implemented for the lecturer interface.

### 5.5.5 Visualizing the Auditors Impact

Considering requirement 7 for the first iteration a visualization is needed to show the impact of the auditor to the collective audience perception. Every time an auditor gives feedback via the backchannel system it is assumed that the auditor needs to see the impact of this action to gain motivation out of this action. This motivation should lead to continuous voting during the lecture. Although motivation is also fostered by using the Tamagotchi effect (see section 5.5.2 on page 57) it is not known whether this effect can generate enough motivation to achieve continuous maintaining of the auditor perception. While the voting result is already visualized with the avatar, the collective perception also needs a visualization to represent the auditor impact.

For the first prototypes (see figure 5.15 on page 67) it is assumed that it is important for the auditors to know the perception of their virtual neigh-

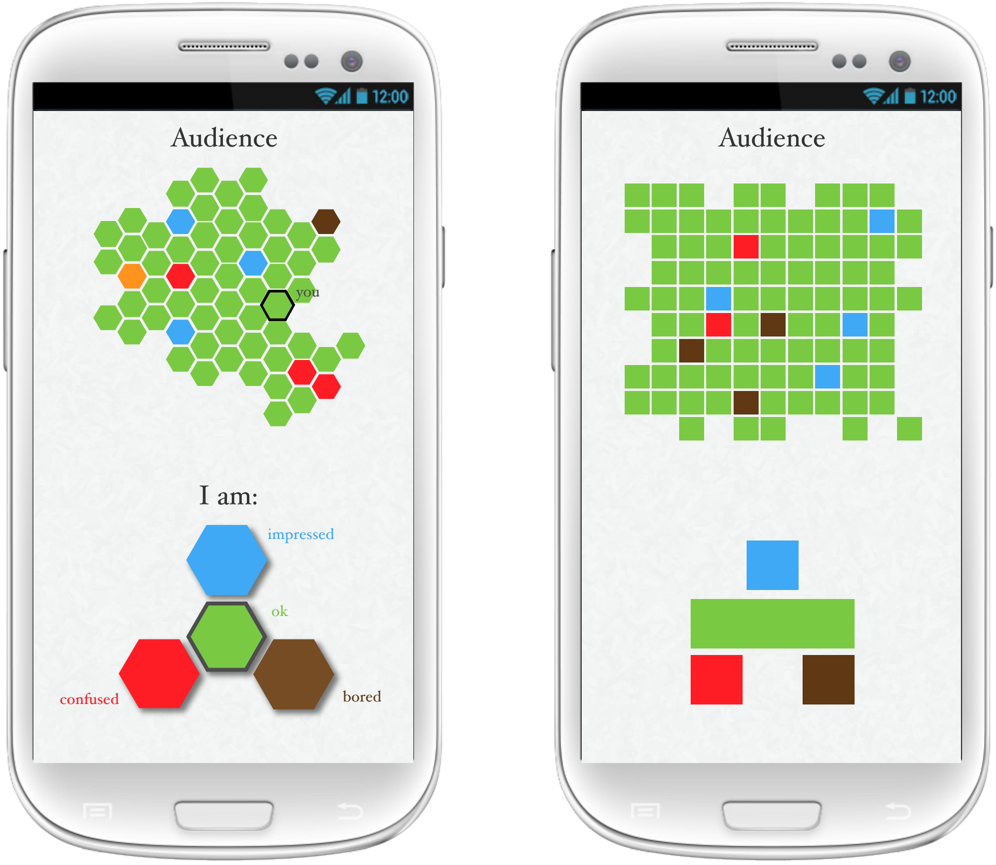


Figure 5.15: Protoypes for visualizing the auditor impact on the collective perception of the audience

bors. In this visualization, every element in the upper screen represents an auditor who is also in the same lecture hall. While the virtual position of the other auditors in the visualization is not related to the real position in the lecture hall the feeling of *being part of a group* should be communicated. One of the elements is representing the auditor. The impression of being in the middle of many others, which are all anonymous, should also encourage voting.

In these early prototypes, the dimensions in the lower part of the mockup were not selected based on the criteria from section 5.4 and are all only binary. The two values are differentiated as *voted* and *not-voted*. These prototypes are focusing on the visualization of the auditor impact. The dimensions are color-coded. When the auditor votes for one dimension, the auditor element changes its color corresponding to the vote. The same applies to the elements of the other auditors. When an auditor votes, the color of the auditor's element instantly changes to the dimension color on all other auditor devices. It is assumed that this visualization can address two goals, the overall trend of the collective audience is recognizable and every individual vote is visualized. It is assumed that this approach limits the suitable size of the audience to a certain degree. The more auditors are present, the smaller the individual virtual auditor elements are scaled. It is assumed that audience sizes up to 500 auditors can be visualized accordingly on current mobile Internet devices. More auditors cannot be displayed while individual auditor elements stay distinguishable without magnification. Magnification will also increase the interaction complexity for the auditor.

Another potential disadvantage is the limited visualizable information per auditor. If the dimensions are of higher resolution, the color-coding will be hard to distinguish. The visibility of an overall trend will also decrease with the increasing number of colors. Another potential disadvantage is the unclear and unintuitive presentation of the own element. While it is highlighted with a black border on the left mockup in figure 5.15, usability interviews with this mockup showed that it needs an additional text label to not confuse users. The fact that the distribution of the virtual neighbors is not related to the real position in the lecture hall was also confusing.

A possible discussed enhancement of this visualization was the stacking of votes. Every click on a dimension should add a disc with the color of the dimension on the top of the own element. Therefore, elements are a three dimensional stack of disks. This would imply that auditors, which are voting more often, would have a larger stack. This could motivate other auditors to vote more regularly to increase the size of their stacks. The downside of this motivation could be the gaming effect of this visualization. If the sole motivation is to get a higher stack of discs, the voting might be just a means to an end, and the voting values are not anymore representing the perception of the auditor. The usability of this three dimensional view could be similar to the well known Google Earth app [Google, 2013e] on mobile devices. These navigational touch gestures and mouse gestures have already reached a broad range of users worldwide it is assumed that these gestures are user-friendly.

While many of these disadvantages could possibly be solved by further investigation, the main reason this approach is discontinued are performance issues. Considering the number of auditor elements should be able to increase up to 1000. It can be assumed that using DOM elements for representing this number of auditors will be too performance and memory consuming even for non 3D visualization. Therefore, graphic libraries where evaluated and tested on different mobile devices using 1,000 elements to simulate a basic implementation. Performance tests were executed with Three.js [Ricardo Cabello, 2013] and paper.js [Lehni and Puckey, 2013] on Android, Windows Phone, iOS. The overall performance of test implementations and demos is not supporting good usability. This might change in the near future as device performance and performance of the canvas element increases. For the first iteration, this is not suitable because this will make the user experience worse on older devices. This is not acceptable regarding requirement BYOD 5.

While the discussed approaches might be further developed in future work, for the first iteration a less complex approach is used. The avatar visualization, which is in use to represent a single auditor, is extended to represent the whole audience. While the dimension value representations are kept the same, three avatars are representing the audience. An example state of the audience representation is shown in figure 5.16 on page 70. All three auditor illustrations are acting simultaneously. The three stickmen indicate

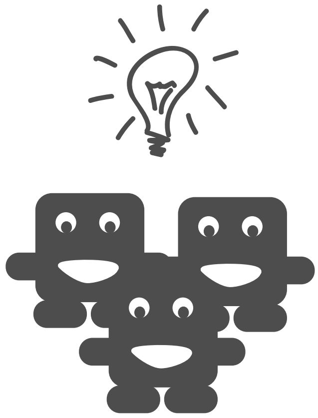

Figure 5.16: Audience avatar

a group of many auditors. The visualization representing the audience is called the audience avatar.

While the audience avatar already gives an indicator how the perception of the collective audience is, it might be hard to translate this visualization back to dimensions. Assuming an auditor wants to know whether the audience also thinks the presentation speed is too fast. The auditor needs to know the different stages of visualization for the dimensions. To make it easier for the auditor to recognize the dimensions separately, dimension indicators are introduced.

Dimension indicators have the purpose to indicate the audience perception on a dimension level. Using already known elements for the auditor the design of the dimension indicators is closely following the slider paradigm. While sliders are used to be controlled by users, the dimension indicator is a slider where the sliders indicator position is controlled by the system. The backchannel system is calculating the collective perception of the audience based on the individual votes of auditors see section 5.5.4. Then, the system sends the results back to the auditor interface to accordingly set the dimension indicators. The interaction by the user is disabled for the dimension indicators. A user cannot control a dimension indicator by touching or clicking on it. A screenshot of the dimension indicators is shown on figure

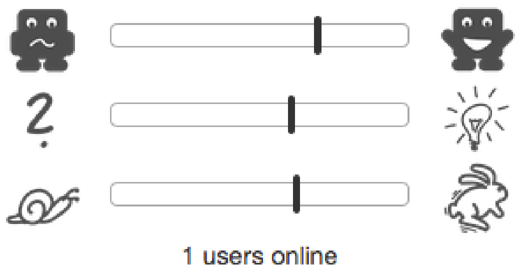


Figure 5.17: Three dimension indicators are visualizing the perception of the collective audience

5.17 on page 71.

The audience avatar has only five states per dimension. This resolution is low compared to the 200 steps resolution on each dimension the system is collecting from the auditors. An auditor can see only an impact of the auditors vote in the auditor avatar if the vote has enough impact to change from one of the five states to another. Then, the auditor can see a change in the audience avatar. All other auditors also see this change because the audience avatar is changing simultaneously on all auditor interfaces. This also applies to the dimension indicators, which also represent the audience perception but have a much higher resolution of 200 steps per dimension. Therefore, even if the audience avatar is not visually affected by an individual vote of an auditor the impact on the dimension indicator can be seen because of the higher resolution. How much the indicator is moving is depending on the algorithm which also influences how much an individual vote is affecting the collective perception, see section 5.5.4 for details.

Figure 5.18 on page 72 shows the final implementation of the auditor interface for the first iteration. The auditor can vote on the left side via the slider elements, and the overall audience perception can be seen on the right side. When the auditor drags the sliders on the left side, the dimension indicators on the right side are changing almost instantly according to the applied processing algorithms.

While the audience avatar and the dimension indicators are representing only the current perception in the lecture hall, it is assumed that this is

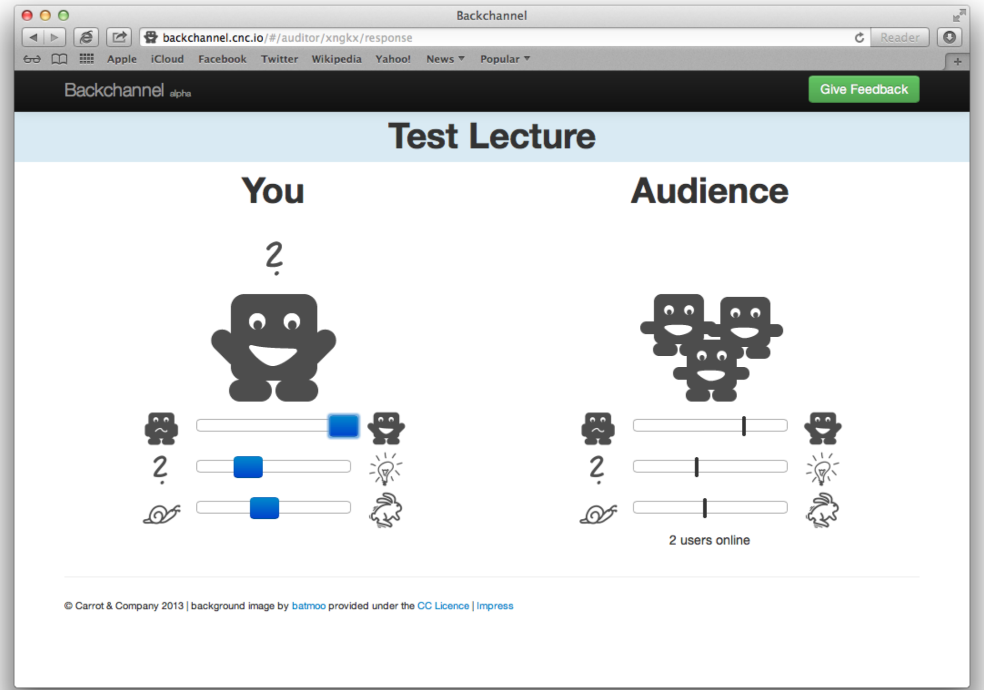


Figure 5.18: A screenshot of the auditor interface of the first iteration illustrating the impact of the auditor voting on the collective perception of the audience illustrated on the right side.

enough information to motivate, not distract, and not influence the auditors. A visualization that includes the time as one visualization dimension could also be implemented in future work. For the first iteration, it was assumed that this similar visualization of auditor perception and audience perception lead to minimal distraction, which is a requirement (see 2) for the first iteration.

## 5.6 Lecturer Interface

The lecturer interface should communicate the important information regarding the current perception of the audience to the lecturer as requirement 8 states. While this requirement is not considered for the first iteration, a basic lecturer interface is implemented as shown in figure 5.19 on page 5.19. This lecturer interface will give the lecturer the same information as the audience. While the lecturer interface is not yet optimized regarding meaningful presentation of information, the basic requirements regarding the workflow of the lecturer as described in section 5.3.3 are fulfilled.

On the top of the lecturer interface in figure 5.19 there is the name of the lecture. The lecturer can optionally enter the name during the creation of the lecture. The left side of the interface is about the audience perception. The audience avatar is displayed with the recent values as well as the number of online auditors. This number represents not the current voting auditors, but all auditors which are connected to the lecture also including passive auditors who are only observing the audience perception. Changes on the audience perception are also represented instantly on the audience avatar and on the dimension indicators.

On the right side of the interface in figure 5.19 the lecture code is displayed. This code, which is needed to link auditors to a lecture session (see section 5.3.3 for further explanation), needs to be communicated to the audience by the lecturer. This can be done, for example, by writing the code and the web address of the backchannel system on the blackboard. If the lecturer interface is displayed on a data projector in the lecture hall, the audience can also directly browse to the auditor interface of this lecture by scanning the QR code with a QR code reader.

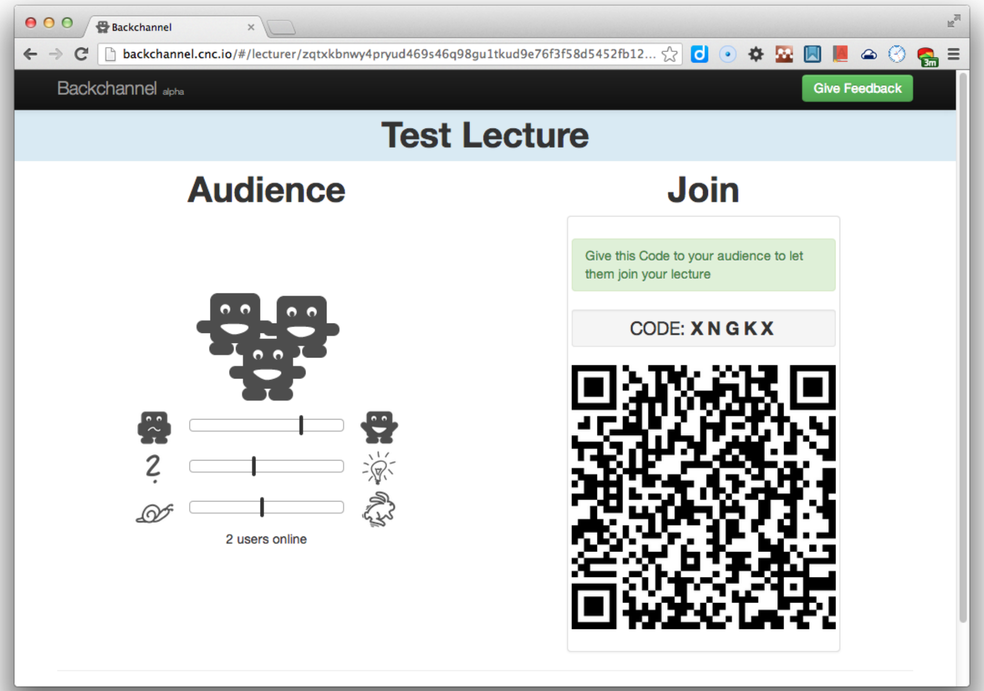


Figure 5.19: A screenshot of the lecturer interface of the first iteration

The current implementation of the lecturer interface is minimal but regarding the requirement of presenting meaningful data and not distract the lecturer (requirement 2) it is assumed that a minimal approach for the first iteration is a good foundation for further improvements.

# 6 Findings

The new quantitative backchannel system was live tested during a lecture unit of "Social Aspects of Information Technology" by Martin Ebner at Graz University of Technology. This lecture is compulsory for the bachelor programs of Telematics, Computer Science, Software Development and Business Management. It is also compulsory for the Teacher Training Program for Computer Science and Computer Science Management.

The presentation started on the 20th of March 2013 at 5:00 p.m. and ended at 5:40 p.m. The manually counted number of auditors during the presentation was 133. The system was roughly introduced in 5 minutes. The live test was stated as a performance and usability test, which does not influence grading. It is also explained that the system is in a very early stage and that during the test, the lecturer will not get feedback about the current backchannel situation in the lecture hall. During the short introduction of the system, the handling of the system was not explained. The 5-letter lecture code was told, and the URL to the system was written on the blackboard. Feedback for the new backchannel system was requested from the audience. Direct feedback could be submitted via a button on the website which links to a public Google Docs document which was opened for writing.

During the live test, a Virtual Private Network (VPN) disruption in the lecture hall occurred. This lead to an Internet connection loss of all auditors, which were connected to the Internet via the VPN of the university. Considering Google Analytics it can be assumed that at least 84.3% were connected to the backchannel website via VPN. In figure 6.2 on page 87 the outage is illustrated as straight horizontal lines in the average trendlines between 5:10 p.m. and 5:12 p.m. In this period, no votes from VPN auditors have reached the server. This complete outage of the VPN of two minutes should definitely be avoided in a production setting, but for the live test it was a highly appreciated coincidence. These two minutes of zero activity on the server, enabled the evaluation of some visualization algorithms and might also lead to some specific feedback regarding the connection state of the website.

## 6.1 Feedback

During the introduction of the backchannel system, it was tried to motivate the audience to give unadorned anonymous and open feedback. To make it as easy and motivating as possible, a different approach compared to the ticket systems or support boards, which are typically used, is tried. Assuming motivation can rise through anonymous collaboration and opinion sharing a system was needed which supports open text collaboration in real-time. Every auditor should have the possibility to instantly write open text as feedback without registering or signing in. An evaluation of various systems such as etherpad was not done because of the lack of time. Instead, it was decided to use a public available Google Docs document. Everyone with the link to it has write access. The link was integrated as a feedback button in the web application of the backchannel system. Google Docs also has version control so every change that is made is documented and can be replayed or restored. The concern that auditors might delete other auditors' feedback was groundless.

The maximum number of simultaneous auditors of the feedback document was 15. Unfortunately, there is no way to find out how many of these auditors have contributed. The raw unformatted discussion document is in German and has 194 words. In this section, the translated feedback is discussed.

| **Feedback 1: Graph over time** |
| --- |
| A graph over time for each dimension would be useful. This will make the trend of the votes over time recognizable. |

As explained in section 5.5.5 the visualization which only represents the current state is chosen because it is assumed that timely graphs would influence the voting of individual auditors in a way that auditors might support the graph in the trending direction, independent of their personal feeling and comprehension during the lecture. This behavior is not intended but cannot not be eliminated completely also in the current visualization. Considering no clear timely trend in the current visualization, it was assumed that the motivation to follow the collective trend is lower than on visualizations, which also illustrate previous values of the collective.

On the positive side of a time based visualization, is the understanding of the trends of the dimensions. Changes are more explicit over time than on a pure last-state visualization. This can result in a motivation to participate to the backchannel and adjusting the visualized trend to the currently perceived state of the lecture.

| **Feedback 2: Connection state** |
| --- |
| Provide an indicator, which shows the current connection state of the web application. There was a VPN outage, and I could not distinguish if there were no activity in the lecture or if I just lost the connection. |

Attention was paid during the architecture of the system to accordingly handle connection losses. Further, it was assumed that connection loss should be hidden from the end users to not distract them. Automatic reconnection after connection loss is established by the system. The pending votes are sent immediately after the connection is available again. A combination of automatic error handling combined with a connection indicator is an approach that should be investigated in future iterations of the system. Negative side effects in the user experience are not obvious as long as no active interaction with the indicator, or notification is necessary to continue with the normal user workflow. However, as stated before, the information itself that the connection is lost can distract the user to some extent. Usability tests need to be done to evaluate if the positive aspects of an indicator can outpace the negative ones.

| **Feedback 3: Shoutbox** |
| --- |
| I would like to have a shoutbox field from where I can send short messages and comments to the lecturer. |

This is a feature implemented for example in the Backstage system by Pohl, Gehlen-Baum, and Bry [2011]. Although it can be assumed that this feature is requested from the auditor side it can also be assumed that the lecturer has no time during active lecturing for reading, filtering, and step into questions and comments from the audience without training and reorganizing the lecture itself. One requirement of the implemented backchannel system was to distract the audience and the lecturer as little as possible. Besides this requirement, the implementation is focused on quantitative backchanneling in contrast to qualitative backchanneling. A shoutbox can be categorized as qualitative backchanneling method regarding section 2.5. Implementation of a shoutbox feature as a backchannel or feedback system is an interesting topic itself but out of the scope of this work.

| **Feedback 4: Interpretation of the Snail** |
| --- |
| For me, it is not clear if the snail stands for “lecture is too slow” or ”lecturer should slow down”. |

This problem is expected to some extent in section 5.5.3. In contrast to the other two dimensions which clearly addresses the auditor and the auditors subjective impressions of the lecture, the third dimension *speed* cannot easily be linked to a subjective distinct impression. It was tried to address this issue with the auditor avatar, which starts sweating when the slider is moving towards the rabbit symbol. This should be interpreted as ”the lecturer presentation is too fast for me”. The opposite side of the slider is symbolized with a snail, which should be associated with slowness. When the slider is moved towards the snail, the avatar starts to get tired and finally fall asleep. This should be interpreted as the lecturer is too slow for the au-

Figure 6.1: Minimum and maximum presentation speed positions of the auditor avatar.

ditor; therefore, the auditor may fall asleep because of under-engagement. Figure 6.1 presents the minimum and maximum positions of the avatar.

The other two dimensions have symbols that are more or less save against misinterpretation in the given context. The speed parameter depends on the user's point of view. The two interpretations "it is too fast" and "make it faster" cannot be distinct without communicating a reasonable context. Although it is tried to give the needed context with the sweating and sleeping of the avatar it is obviously not clear for every auditor during the live test. This is clearly an important point for future work since the three dimensions should be extendable and changeable in an easy way without losing the easily understandable environment.

A possible concept to address this issue would be a short integrated description, for example, as tooltips in the UI. These tooltips can explain the sliders and their influence. However, as already discussed in section 5.5.3 textual help needs more effort to understand and interpret as images besides the need for translation in various languages for localization.

| **Feedback 5: Optimize Loading Performance** |
|---|
| The loading of the web application took long for me on my smartphone via university VPN. Possible enhancements are GZIP compression and optimization of the website. |

This feedback is addressed right after the live test. It was planned to deploy a more optimized version but because of the needed debug options for analyzing the usage during the test, the optimization was postponed. It is tried to implement the *Best Practices for Speeding Up Your Web Site* [Yahoo, 2013a]. Not all best practices are already implemented because of different reasons. The following points are already realized in the backchannel system to address this feedback.

**Image Optimization** All images on the web applications are optimized. The image optimization is done through Yahoo Smushit [Yahoo, 2013b]. Smushit is a lossless optimization service by Yahoo. It optimizes the images by applying different third party tools and algorithms to reduce the image size. It removes unnecessary bytes of images without reducing the visual quality. All images on the backchannel web application are optimized. Twenty-seven images were processed with an average reduction in size of 33%. The maximum reduction in size of an image in the backchannel system was 63.94%, which is an image from the jQuery UI library [JQuery, 2013].

Listing 6.1: Smushit output of Grunt script. For these files smushit [Yahoo, 2013b] provides an average saving of 33%.

```
[smushit] item: ./c_min_icon.png saving: 40.36%
[smushit] item: ./h_max_icon.png saving: 32.68%
[smushit] item: ./c_max_icon.png saving: 36.06%
[smushit] item: ./h_min_icon.png saving: 33.04%
[smushit] item: ./s_min_icon.png saving: 33.05%
[smushit] item: ./s_max_icon.png saving: 35.53%
[smushit] item: ./favicon.png saving: 13.42%
[smushit] item: ./create.png saving: 39.60%
[smushit] item: ./glyphicons-halflings-white.png error: No savings
```

```
[smushit] item: ./avatar1_stripes.png saving: 22.41%
[smushit] item: ./ui-bg_flat_0_aaaaaa_40x100.png saving: 62.74%
[smushit] item: ./avatar1_audience_stripes.png saving: 26.83%
[smushit] item: ./glyphicons-halflings.png saving: 0.56%
[smushit] item: ./join.png saving: 33.85%
[smushit] item: ./ui-bg_flat_75_ffffff_40x100.png saving: 63.94%
[smushit] item: ./ui-bg_glass_55_fbf9ee_1x400.png saving: 41.79%
[smushit] item: ./ui-bg_glass_65_ffffff_1x400.png saving: 64.25%
[smushit] item: ./ui-bg_glass_75_dadada_1x400.png saving: 50.76%
[smushit] item: ./ui-bg_glass_75_e6e6e6_1x400.png saving: 50.76%
[smushit] item: ./ui-bg_glass_95_fef1ec_1x400.png saving: 41.87%
[smushit] item: ./ui-bg_highlight-soft_75_cccccc_1x100.png saving: 52.50%
[smushit] item: ./ui-icons_2e83ff_256x240.png saving: 6.31%
[smushit] item: ./ui-icons_222222_256x240.png saving: 31.80%
[smushit] item: ./ui-icons_454545_256x240.png saving: 32.48%
[smushit] item: ./ui-icons_888888_256x240.png saving: 32.50%
[smushit] item: ./ui-icons_cd0a0a_256x240.png saving: 6.31%
[smushit] item: ./home_bg_by_batmoo.jpg saving: 5.09%
```

**Javascript Concatenation and Minification** Concatenation and minification of all JavaScript files should also increase page loading time regarding to Yahoo [2013a]. The concatenation will reduce the number of HTTP requests. The minification process is done by UglifyJS [Bazon, 2013]. The complete concatenated and minified JavaScript file has a size of 510,894 bytes. Most of the code are JavaScript libraries. The real client side application code for the backchannel system is only 15,904 bytes unminified. This application code is minified with UglifyJS to 7,709 bytes, which is 48.5% of the original size.

There are other solutions of optimizing loading time, for example, RequireJS [RequireJS, 2013] which does JavaScript loading asynchronously. This is a topic for future work to evaluate further possibilities in enhancing the backchannel system loading performance.

**CSS Minification** CSS is concatenated and minified similar to the JavaScript concatenation and minification to reduce HTTP requests and minimize the size of the CSS file. To minify the CSS grunt [Grunt, 2013] uses *Clean CSS* as a backend library to minify the CSS files. The stylesheets from libraries and application have concatenated and unminified a size of

184,206 bytes. The minified version is reduced to 81.9% of the original size. The minified CSS file size is 150,932 bytes.

**Using CSS Sprites for Images** The used libraries like Twitter Bootstrap [Twitter Bootstrap, 2013] already make extensive use of CSS Sprites. CSS Sprites are used for the avatar images in the angular directives as described in section 5.5.2. The 125 possible states of the avatar are made out of 3 sprites each with 5 different states ($5^3 = 125$). These overlayed states produce the final appearance of the avatar. The use of CSS Sprites can reduce the HTTP requests and prevent flickering during dynamic image changes caused by loading latency.

**Put Scripts at the Bottom** All scripts are located at the bottom of the body to prevent a blocking of the HTML loading. In case of AngularJS, this might be of minor importance unless highly optimized and modular. Because AngularJS has a significant influence on the HTML markup content, in this application the web application is little useful without the loaded AngularJS.

## 6.2 Quantitative Data Analysis

During the live test anonymous usage data was acquired via the backchannel system itself and Google Analytics. Additionally, the presentation was recorded on video. The analysis based on the data of the backchannel system cover data accurately between 5:00 p.m. and 5:40 p.m. Data retrieved from Google Analytics are between 5:00 p.m. and 6:00 p.m. because of limitations of Google Analytics [Google, 2013c]. It can be assumed that the major data recorded from Google Analytics between 5:00 p.m. and 6:00 p.m. are relevant for the analysis. This assumption is because the website was not announced publicly and by the fact that 84.3% of the visitors, based on Google Analytics in this time period, are accessing the website through the university network. It can further be assumed that some auditors are accessing the website via UMTS devices like smartphones or tablets which increase the relevant data above 84.3%. Another strong indicator to assume that most of the auditors were in the lecture hall, is the VPN outage mentioned earlier. The VPN outage leads to a connection loss of the Internet for all students connected via university VPN. In these 2 minutes, where VPN was not working, no votes were incoming on the server for this particular lecture. During these 2 minutes, the server was online and reachable from the Internet. This proofs that during these 2 minutes absolutely no vote was sent from outside the lecture hall. It is assumed that these 2 minutes are representative for the participation of users from outside the lecture hall. It can then be further deduced that if users from outside the lecture hall were actively participating in the voting, they do not significantly influence the result because of their minority.

The raw dataset consists of the backchannel systems internal database and the dataset provided by Google Analytics. The backchannel system dataset consists of a record for every vote. Every vote is related to an auditor. A client-side cookie distinguishes auditors. The cookie value is an Universally Unique Identifier (UUID). Introducing auditors makes it possible to distinguish multiple votes from one auditor against single votes from multiple auditors. A vote record is stored as shown in listing 6.2.

Listing 6.2: Vote structure in JSON (MongoDB)

```
{
  "_id" : ObjectId( "5149d2028f7f42379812c6c8" ),
  "timestamp" : 1363792386.271212,
  "value" : 100,
  "type" : "comprehension",
  "auditor_id" : "5149cf228f7f44579812c6ba",
  "lecture_id" : "5149bdb08f7f44672840995a"
}
```

### 6.2.1 Raw Data

During the live test, 1,675 votes from 117 different auditors were recorded. If one person in the audience votes with two different devices the backchannel system recognizes this person as two auditors, one on each device. In contrast, Google Analytics might count them as one person since Google

| | |
|---|---|
| Number of valid votes: | 1675 |
| Counted auditors in lecture hall: | 133 |
| Active auditors counted by backchannel system: | 117 |
| Unique visitors counted by Google Analytics: | 83 |
| Number of vote types: | 3 (happiness, comprehension, speed) |
| Vote scaling: | linear from -100 to +100 |

Table 6.1: Data analysis facts

Analytics also uses session information of other services to remember unique visitors. For example, if the person is signed in to Google Mail with the same account on both devices Google Analytics can count the person as one visitor across devices. This might be one reason Google Analytics counts only 83 visitors. Another reason might be the opt-out option for Google Analytics [Google, 2013d]. This plug-in prevents data capturing by Google Analytics for browsers which have the plugin installed. In a lecture with an audience probably aware of privacy considerations related to Google, it can be assumed that at least a minor part of the audience is using the opt-out plugin provided by Google. Another option for the difference might be manual deletion of cookies by the auditors to test the reaction of the web application. In such a case, when the cookie got deleted, the backchannel system believes that the visitor is a new auditor and creates a new unique auditor token. Therefore, the auditor is counted twice. A summary of these base numbers from the datasets is listed in table 6.1 on 86.

In figure 6.2 on page 87 all valid data points captured by the backchannel system over the 40 minutes presentation time can be seen. The y-axes represent the value of the votes. For example, if an auditor drags the slider of type happiness to the right (the most happy smiley) and drops it there, a vote with the value +100 of dimension happiness is sent to the backchannel server. Every vote in this figure represents a single data point. Auditors are not considered in this representation. The raw data points are hard to interpret, and because auditors are not considered in this representation, the information value of this figure for a lecturer is low. It is necessary to create easier to interpret visualizations for the lecturer. Even in non-lecture environments this representation gives us only an idea how active the audience was.

For easier interpretation and visualization of the data, some diagrams are

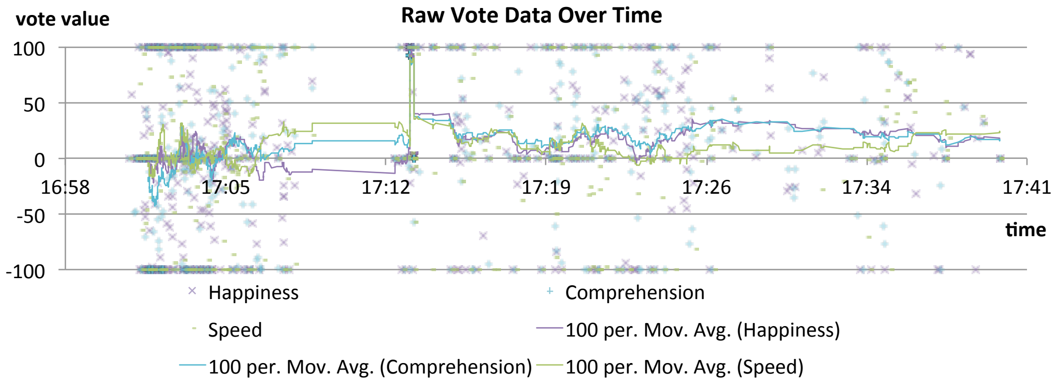


Figure 6.2: Raw votes data over time. 1675 data points over 40 minutes.

averaged on a per minute base. This means if an auditor votes for three times in a minute for happiness these three votes are combined in one vote. The values are averaged with the arithmetic mean before the diagram preparation algorithm processes them. The normalized diagrams are explicitly labeled as time normalized diagrams.

### 6.2.2 Activity Over Time

Considering that it is the first time the audience use such a backchannel system it could not be expected to get representative data similar to a production use case. Besides, the curiosity of the audience to explore the web application, interesting data for the activity distribution over time were acquired. This is important for the requirement continous interaction 1.

Figure 6.3 on page 88 shows the timely trend of the number of votes. Every vote is counted no matter if it is from different auditors or not. Auditors are not considered in this diagram. A high activity in the first few minutes can be seen. Considering the network outage, it is not exactly clear if the collapsing network causes the decrease in votes, starting from minute three. The peak in minute 13 let us assume that the decrease is strongly influenced by the network outage. The peak is a result of the automatic

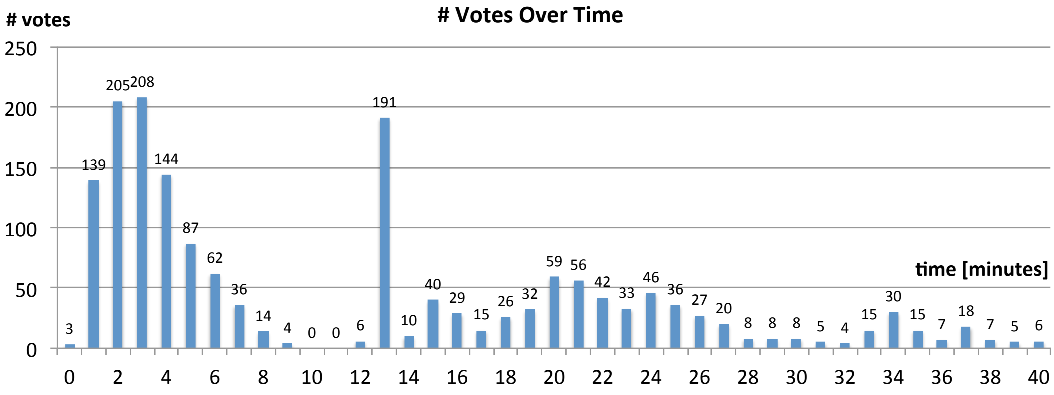


Figure 6.3: Number of audience votes over time

re-connection from the auditor interface in the browser, which tries to send the queued votes, which were pending since minute 10. The system handles connection losses in the background and re-sends the queued votes on minute 13 when the network is online again. Therefore, all votes were incoming to the server simultaneously. Supposing the network outage would not have happened, it could be assumed that the votes, which generate the peak, would be distributed over the last 3 to 9 minutes. This would have absorbed the strong decrease of votes during this period. Apart from this, high interaction in the beginning and a continuous decrease in votes can be seen.

Possible reasons for the decrease of the votes can be, for example, a too large effort to handle the backchannel during participation in the lecture or the motivation to use the backchannel might decrease over time for several reasons.

The average number of votes over the 40 minutes is 42.7 votes per minute. This can be assumed *active*. Auditors can be taken into account to get a different perspective. Visualizing activity based on the auditors, which were active per minute give us another perspective (see figure 6.4 on page 89). The distribution of the number of active auditors over time is quite similar to the distribution of votes. The average is 7.7 active auditors per minute. It should be mentioned that there is no distinction in the counting if it is the

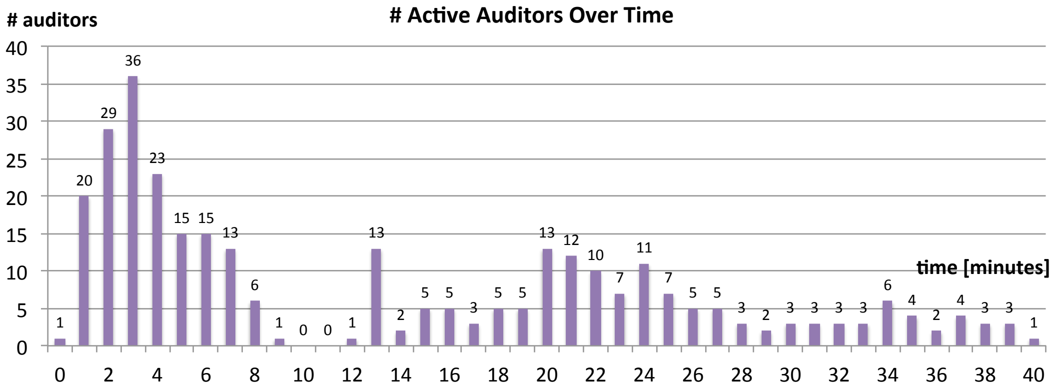


Figure 6.4: Number of voting auditors over time

same auditor in distinct minutes or different auditors.

Interpreting figure 6.3 and 6.4 leads to the question if the activity per auditor will stay constant over the lecture. Figure 6.5 on page 90 shows the average votes per active auditor over the lecture time. This activity is also decreasing but far less significant than per auditor and per vote. Based on this information it can be assumed that an average auditor will not stop to actively participate in the backchannel during the lecture.

### 6.2.3 Voting frequency

In figure 6.6 on page 90 the number of auditors with a specific amount of votes is presented as a histogram. In this detailed chart, there is a peak of 14 auditors who are just passively watching the backchannel and not actively participating by voting. The next peak is at three votes, 13 auditors voted three times during the lecture. The number of auditors is decreasing quite fast until 30 votes per auditor. Only three auditors have voted more than 50 times. The top "voter" has 99 votes in 40 minutes, this results in 2.5 votes per minute. This histogram gives a quite detailed view of the number of given votes from an auditor.

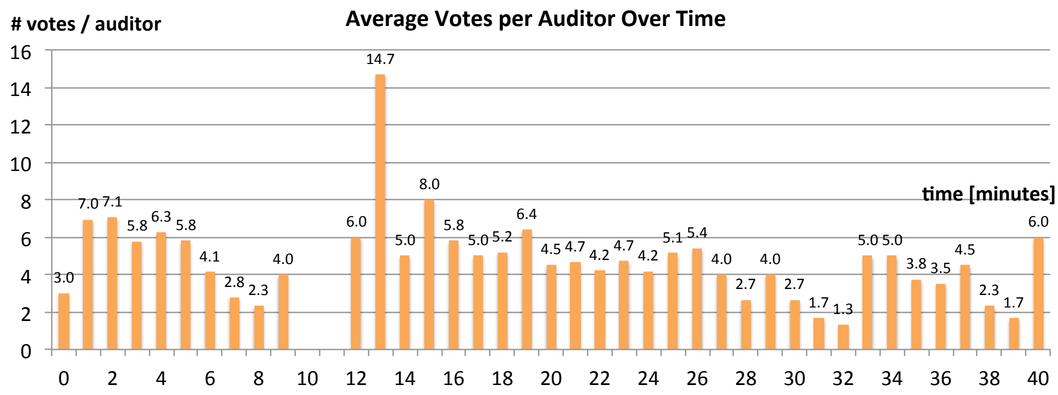


Figure 6.5: Average number of votes per auditor over time

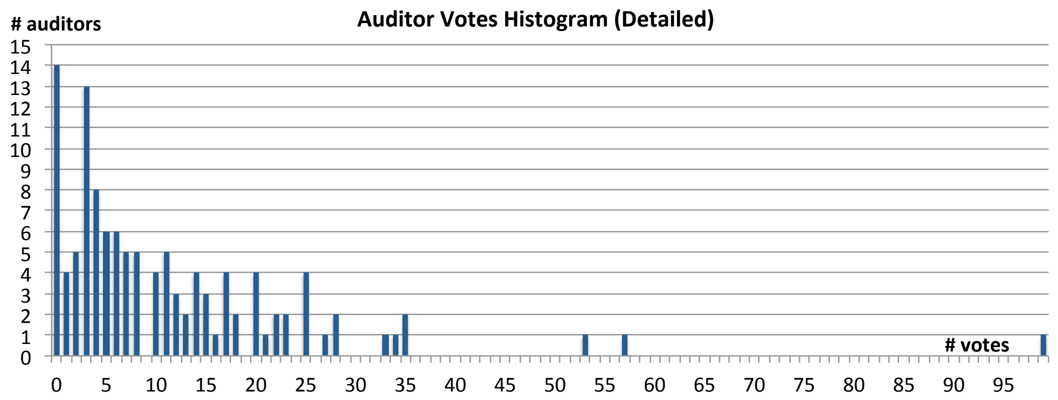


Figure 6.6: Auditor votes histogram over a 40 minutes presentation. There is a peak at zero votes, passive users who just observe the backchannel. The next peak is at 3 votes during this lecture.

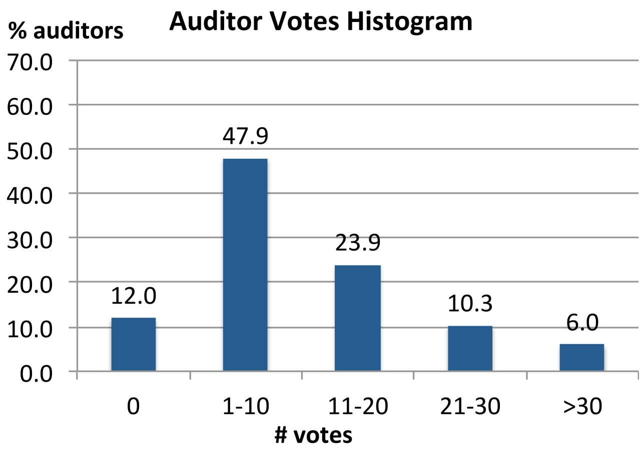


Figure 6.7: Auditor votes histogram over a 40 minutes presentation. The majority of auditors are voting between 1 and 10 times during this lecture. The non-voters are a minority.

To deduce further information especially for the area from 0 to 30 information needs to be summarized further as shown in figure 6.7 on page 91. This histogram has the same data, but condensates it to five value ranges to get a better image of the typical number of votes that can be expected. In this figure 6.7 it can be deduced that most of the auditors are voting 1 to 10 times during a lecture. 11-20 votes during the lecture are sent by a significant amount of auditors. In contrast, the non-voters are a minority.

### 6.2.4 Vote Type Distribution

For the first live test, three dimensions are implemented. An auditor can vote dimensions independently as explained in section 5.4. These three dimensions are *happiness*, *comprehension* and *speed*. This section discusses the distribution of these three types, for example, if one dimension has been voted more often than others.

Figure 6.8 on page 92 presents a timely distribution over the number of votes for a distinct dimension. A clear favorite dimension of the auditors

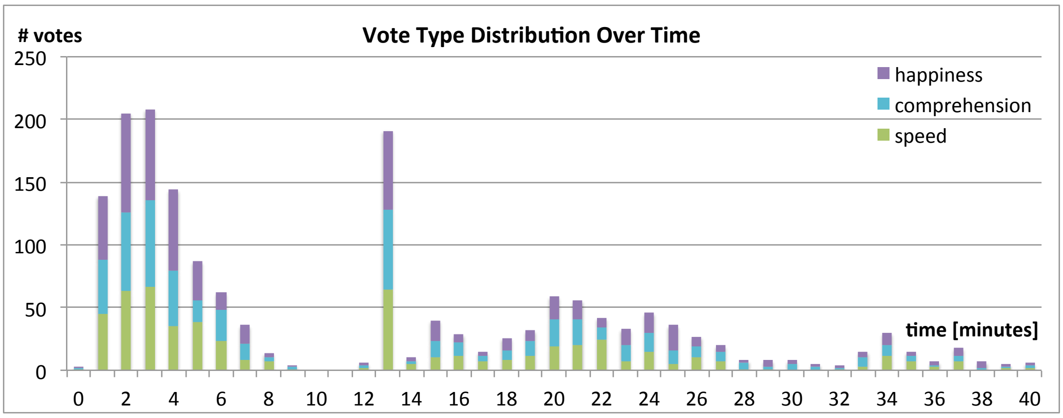


Figure 6.8: Auditor vote dimension distribution over time

cannot be identified. It looks like all three dimensions are almost identical popular for voting. This might be come from the misconception that all three types must be voted to achieve a valid vote. It might also come from the auditors' motivation to produce a complete picture of their current situation by adjusting all three sliders of their avatar.

In figure 6.9 on page 93 the accumulated data are shown. It is explicit that there is no significant difference in the distribution over the three types. This result can be interpreted positively as it can be assumed that three dimensions do not distract the auditors. The auditors seem to have the willingness to vote for all dimensions. Interesting questions for future work regarding this evaluation could be: Will the number of total votes decrease significantly when the number of dimensions increase? How many dimensions are acceptable before a significant drop of votes can be measured?

### 6.2.5 Vote Value Distribution

Compared to other backchannel systems [NK Labs Inc, 2013; Lecture Tools, 2013] the implemented system has a range of 200 values for each dimension as explained in section 5.5.1. For example, a vote for the dimension *happiness* can be in between the value -100, which means "unhappy" to

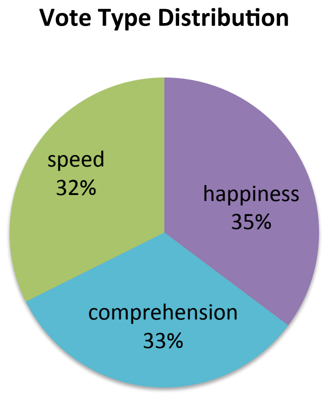


Figure 6.9: Auditor vote type distribution

+100 which means "happy". The same is applied for comprehension and speed. It will be discussed if these increased choices result in using these choices or if only minimum and maximum values are voted. Figure 6.10 on page 94 illustrates the distribution of the vote values of single votes. Three significant peaks can be identified, the largest is the neutral position followed by the minimum and maximum values. This visualization also displays all votes in between, which are low compared to the neutral and min/max areas.

To conclude if the auditors also use this fine-grained control, a different visualization is needed. The visualization must address the questions: *how much votes are in the neutral position, the min/max position, and in the area between?* This question is faced in figure 6.11 on page 95. This diagram states that approximately one-third (excact 35%) of all votes is in between of minimum, neutral, and maximum. This is not obvious in figure 6.10. This indicates that the auditors significantly use intermediate choices. Therefore, sliders seem a suitable UI element for choosing a distinct value in a large range fast and efficient. The auditors use the full range of resolution, which is provided. It can also be seen that slider elements are not suitable for fast and precise positioning with a high resolution. In figure 6.10 the area around the neutral position is normally distributed, which

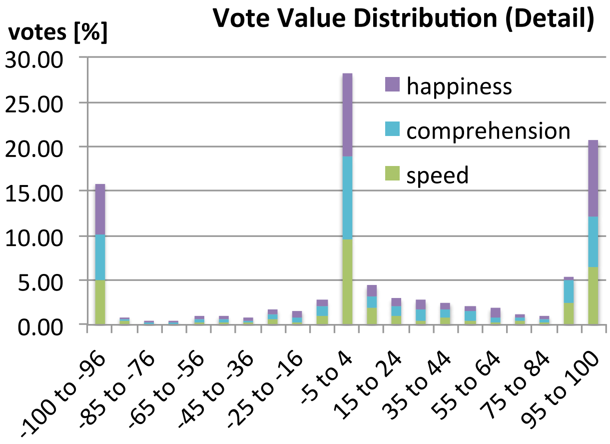


Figure 6.10: Auditor vote value distribution detailed

can be interpreted as auditors that missed the neutral point during slider positioning.

### 6.2.6 Results for the Lecturer

The lecturer interface during the live test has the same information presentation as the auditor interface as described in section 5.6. The lecturer interface in the first iteration visualizes only recent data without a time dimension. The recorded data are visualized subsequently with a time dimension as shown in figure 6.12 on page 95. This chart is based on the algorithm used for the live data visualization as described in section 5.5.4 extended with a time dimension. This chart is also time normalized as described in section 6.2.1.

$$Result_{\%} = \frac{\sum_{i=1}^{n} vote\_value_i}{\sum_{i=1}^{n} auditor\_influence_i} \tag{6.1}$$

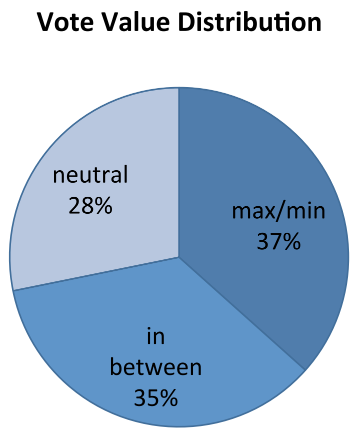


Figure 6.11: Auditor vote value distribution

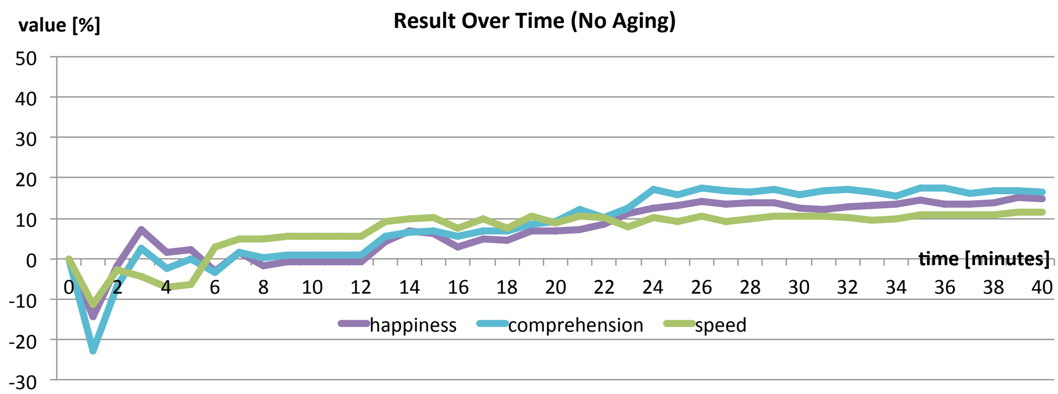


Figure 6.12: The average results of the audience voting over time. Votes from the audience are valid indefinitely. No aging is applied. This leads to an offset in the end where voters from the beginning have stopped engaging, and the active voters in the end cannot influence the result significantly.

Equation 6.1 is the arithmetic mean calculation to produce an average result in percent for each dimension. The $vote\_value_i$ represents the value of the vote which the auditor submits. In the first iteration this value is in the range from -100 to 100. For the comprehension dimension, a $vote\_value_i$ of 100 would represent *100% understood*. The $auditor\_influence_i$ represents the effect of the vote compared to other votes. The $auditor\_influence_i$ is 1 and is lowered only for aging reasons.

In this chart shown in figure 6.12 votes from the audience are valid indefinitely according to equation 6.1 on page 94. This can also be clearly seen during the network outage between minute 10 and 11. Because no aging is applied the graphs have constant values during this time. This leads to an offset in the end of the lecture. It is assumed that this happens because voters from the beginning have stopped engaging and the active voters in the end cannot influence the result significantly because of the large number of total auditors.

This approach is not satisfying the requirements for the auditors because their influence is decreasing significantly over time. This might lead to a decrease of motivation, which should not be aspired. For the lecturer, this approach is not meaningful, as it does not represent recent changes to the audience perception in a clear way. The large influence of voters, which are getting passive over time, tampers the data. Even though, auditors and lecturer have different requirements for their information presentation as stated in section 5.3.3 this approach is not a good solution neither for auditors nor lecturers.

The problem is the tampered data by auditors that turn passive over time. This could be addressed by introducing an aging factor. The influence of the aging factor which is calculated according to equation 6.2 on page 97 is shown in figure 6.13 on page 97. The aging factor should reduce the influence of a vote and its value over time. Assuming an $aging\_decay$ of 0.5 that will reduce the influence and the vote value by one half per minute. For example, a vote of 100% comprehension by one auditor will usually have indefinitely the value 100% and the influence value of 1 for the arithmetic mean calculation as shown in equation 6.1 on page 94. Using the aging factor as shown in equation 6.3 on page 97 it will decrease the value and

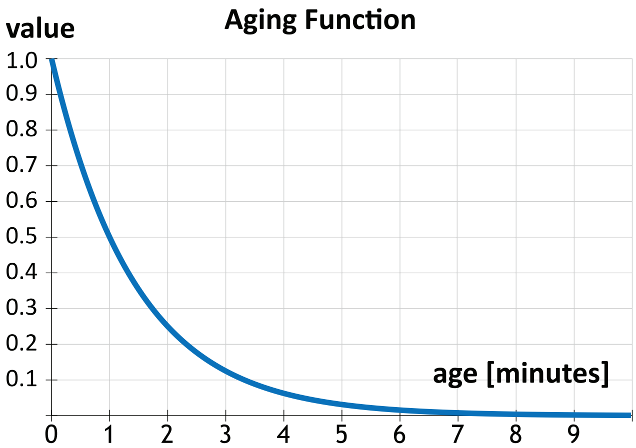


Figure 6.13: Aging factor over time based on equation 6.2 on page 97 with an $aging\_decay = 0.5$.

influence by half compared to other votes on each minute the vote gets older.

$$aging\_factor(age) = \frac{1}{(\frac{1}{aging\_decay})^{age}} \tag{6.2}$$

$$Result_{\%} = \frac{\sum_{i=1}^{n} vote\_value_i \times aging\_factor_{i,50\%}}{\sum_{i=1}^{n} auditor\_influence_i \times aging\_factor_{i,50\%}} \tag{6.3}$$

Applying the modified equation 6.3 with aging and a *aging_decay* of 0.5 the chart is changing as shown in figure 6.14 on page 98. The graphs have larger variations over time compared to figure 6.12. More recent votes influence the graphs more than older votes. A special case appears in the area where the network outage occurred. During this network outage from minute 10 to 12, all three dimension graphs are staying constant. Although aging is introduced this can be explained with the lack of new votes in this time. The averaged values will stay the same because the aging factor decreases the

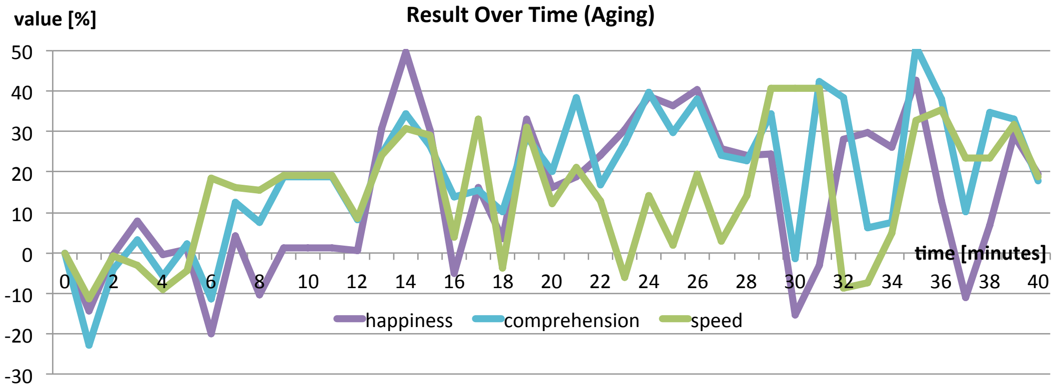


Figure 6.14: The average results of the audience voting over time. In contrast, to figure 6.12 on page 95 aging is applied to the auditor interaction. Aging of 50% per minute is applied to vote values, and the to influence of auditors. This leads to higher influence of active auditors with recent votes.

value and the influence equally as seen in equation 6.2. Without new votes, the dimensions will stay constant beginning from the last vote. When a new vote occurs the old votes, which are now weakened by the aging factor, will lose its influence, and the graph will change dramatically according to the new vote. This phenomenon can be seen in figure 6.14 in minute 12. After the connection was established again, the previous constant dimensions are starting to change according to the new votes.

$$Result_{\%} = \frac{\sum_{i=1}^{n} vote\_value_i \times aging\_factor_{i,50\%}^2}{\sum_{i=1}^{n} auditor\_influence_i \times aging\_factor_{i,50\%}} \tag{6.4}$$

This modification (equation 6.3) improves the influence of more recent votes. This is an improvement for the auditors concerning recentness of information as well as for the lecturer who needs meaningful information, which is also related to recent data. However, the effect that appeared during the network outage is not desirable. It is assumed dimensions should converge to the neutral value over time if no recent votes are incoming.

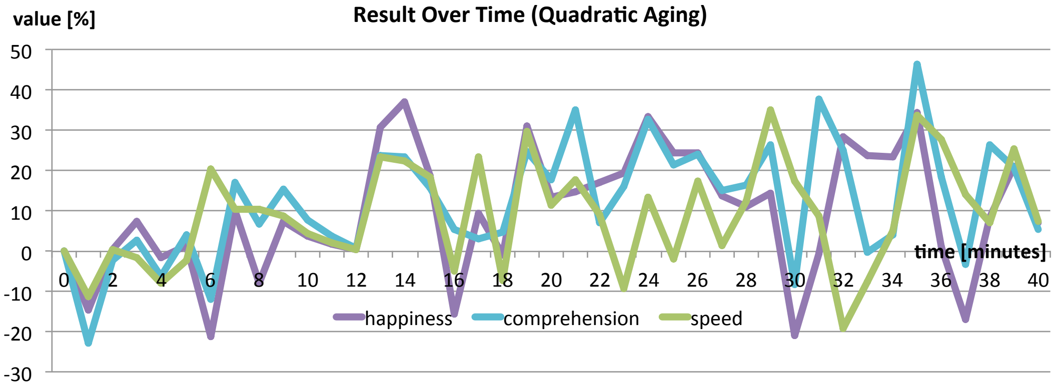


Figure 6.15: Average results of the audience voting over time. In contrast to figure 6.14 on page 98 aging is applied differently. Aging of 50% per minute is applied to vote values and to the influence of auditors, additionally aging of another 50% is applied to vote values to countervail the equally influence of aging on vote values and auditor influence (see equation 6.4) With this aging equation result converge to zero over time without new audience votes.

This is not yet achieved with the aging equation 6.3. To address this issue, the equation is adapted to decrease the voting value more than the voting influence. This is implemented by applying a quadratic aging factor for the vote value. The modified equation 6.4 on page 98 is applied to the data and the resulting chart is shown in figure 6.15 on page 99.

In figure 6.15 the variety of the dimension values improved compared to 6.14. The dimension values during the network outage from minute 10 to 12 were no votes were recognized are converging as expected to zero. After the connection is established again, the votes have high influence because of the recentness and accordingly change the dimension values.

In figure 6.16 on page 100 a comparison is made across the different approaches. The data from the speed dimension are used to illustrate the differences. In the beginning of the chart, all three graphs are quite congruent. In minute six the graphs start to differ. The aging algorithm is causing this effect. The aging graphs have a higher variety because recent votes have more effect than older votes. The blue non-aging graph has a low variety. This is caused by the indefinite validity of votes, which leads to less effect

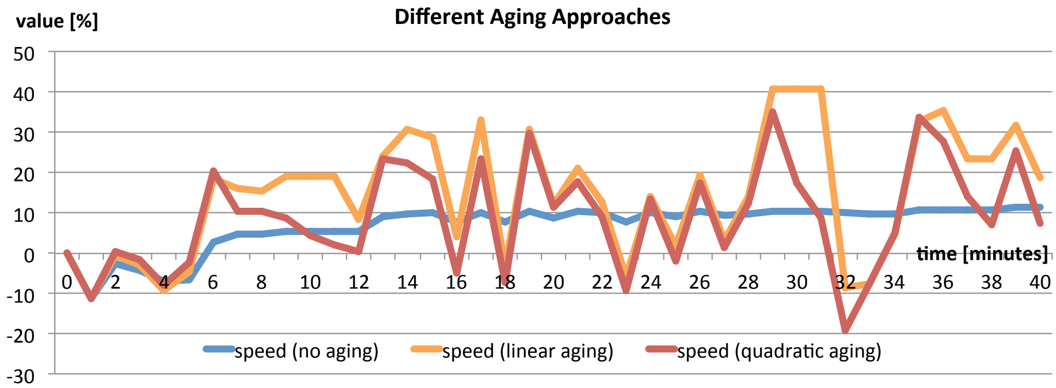


Figure 6.16: Comparison of different aging approaches illustrated on the presentation speed dimension. The graph speed (no aging) is based on equation 6.1. The graph speed (linear aging) uses equation 6.3. The graph speed (quadratic aging) uses equation 6.4

of a single vote. Between minute 10 and 12 the difference of the linear and quadratic aging can be observed. The linear aging equation 6.3 lowers influence and value over time similarly which results in a constant graph as long as no new votes are incoming. In contrast, the quadratic aging equation 6.4 lowers the vote value quadratically and the influence only linear. This leads to a decrease of the result graph over time even if no new votes are incoming.

Equation 6.4 is assumed as an improvement and will be evaluated in more detail in future work. The evaluation of a proper *aging_decay* can also be addressed in future work.

### 6.2.7 Auditor Device Characteristics

Google Analytics acquires many data dimensions to analyze. Only a few dimensions, which are important for the evaluation of the first iteration are analyzed. As stated in the requirement 6 the backchannel system is implemented as a web application with responsive design. The responsive design optimizes the information presentation and interaction methods to the

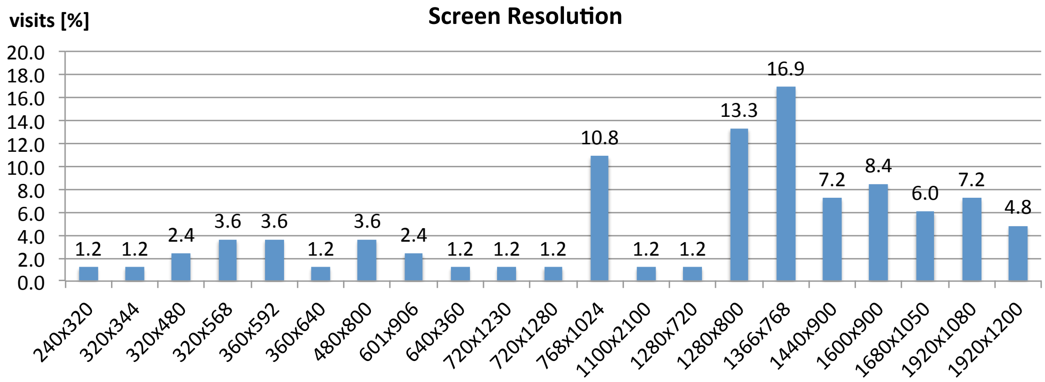


Figure 6.17: Screen resolutions used by the audience devices based on Google Analytics.

available device properties. This is useful to target the broad range of different devices starting from smartwatches over Google Glasses [Google, 2013f] which will be available in the upcoming future to smartphones, tablets, and laptops.

## Screen resolution

Based on the data of screen resolutions captured during the live test, clusters might be identified which could give a good starting point for optimizing screen representation for future iterations of the UI. Figure 6.17 on page 101 illustrates the screen resolution distribution of the auditors. For the 83 unique visitors based on Google Analytics data, 21 different screen resolutions can be distinguished. The range starts on the lower end with just 76,000 pixels and ends at 2.3 million pixels at the upper end. Even tough, the screen resolution compared to the aspect ratio is not the most urgent optimization task it is still important. Especially all pixel-graphics needs to be optimized for high resolutions otherwise, rough edges are visible on high pixel density displays like the Apple Retina display. These optimizations are only needed for pixel graphics like PNG or JPEG images among others. Vector graphics like SVG or EPS are scaling without quality loss automatically. This is also the case for text.

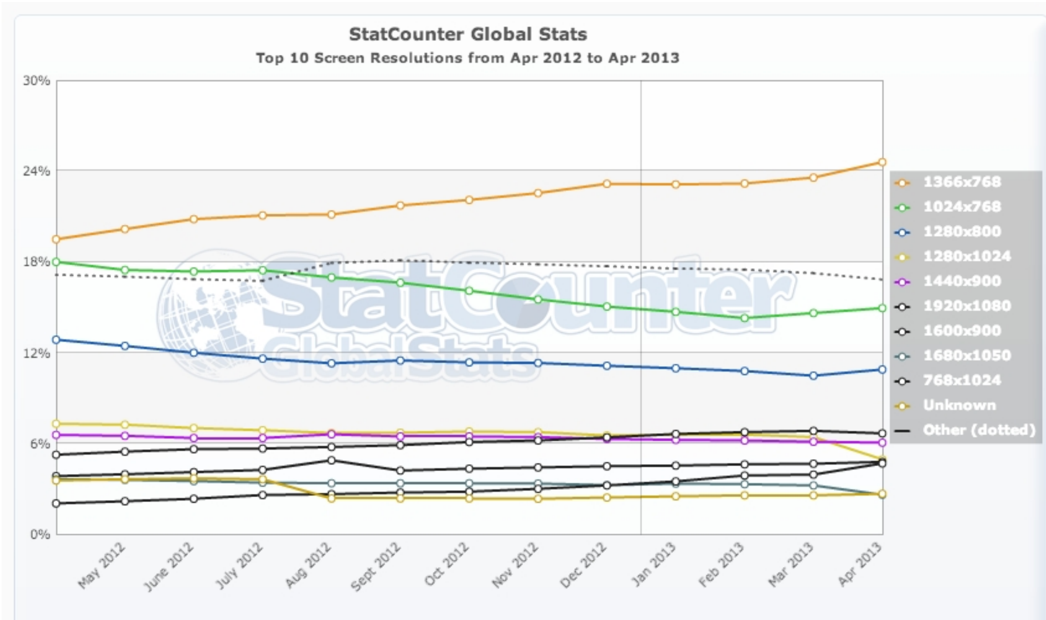


Figure 6.18: Worldwide screen resolutions trend by StatCounter for the last year [Statcounter, 2013]. The most popular resolution in this chart is also the most used resolution in our live test illustrated in figure 6.17

The aspect ratio is more relevant to the responsive design. It defines the width to height ratio of the screen resolution. The aspect ratio can even change instantly during the use of the device by rotating the device. The web application UI must also instantly change to fit the new properties for a good user experience. In figure 6.17 the trend to higher resolution displays can be seen. A cluster is at 1366x768 which is also the worlds most used screen resolution at the time of writing [Statcounter.com, 2013] (see figure 6.18) followed by 1280x800 in the live test. The diversity of the screen resolution, even with this relative small number of auditors, is a good indicator that testing the UI for all kinds of screen resolutions is an important task for a great user experience.

### Browser

To improve the user experience, it is necessary to test the system extensively on various browsers. The implemented backchannel system uses HTML5 greatly. Some functionality like HTML5 WebSockets, which is used for the live interaction is unfortunately not supported by all major browsers; for example, Android 4.2 and Opera Mini 7.0 [Caniuse, 2013]. Wrapper libraries are used to provide backward compatibility and fallback solutions for older browser while maintaining the new HTML5 feature set. An example of such a backward compatible wrapper that is used in the first iteration is Socket.IO. It encapsulates the WebSocket API to a higher level and transparently integrates fallback solutions like long polling [IETF, 2013] and Flash[1] fallback. The analysis of the browsers used during the live test will show which browsers are needed to focus on for the implementation of the system.

Figure 6.19 on page 104 illustrates the distribution of different browsers used by the auditors during the live test. Almost 50% of the audience used chrome to access the backchannel system followed by Firefox and Safari. Internet Explorer was only used by one auditor. Chrome as the most used browser in the live test correlates with the world wide statistics by Statcounter.com [2013] see figure 6.20 on page 105. In contrast, the Internet Explorer is on the lower end of the usage statistics for the live test, but in

[1] Adobe Flash

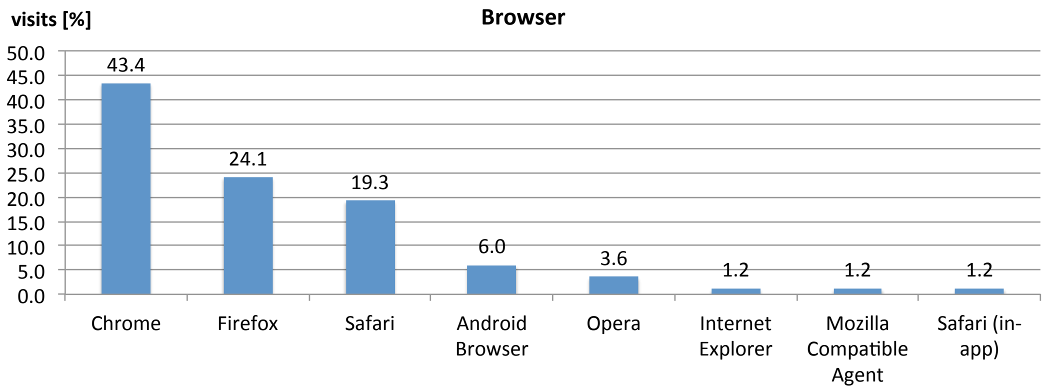


Figure 6.19: Browsers used by the audience based on google analytics.

the worldwide browser usage statistics it is on the second place. A probable reason for this large discrepancy might be the audience who is related to computer science because of their study programs. These numbers are supported the decision for Chrome as the primary development browser for the backchannel system.

### Operating System

The distribution of the operation system usage is interesting for future developments. For a web application, the OS on which the browser is running is of minor interest. Future work might implement native equivalents of the current web application for different OS platforms. When developing native application it is a difficult decision which platform to support first. This OS analysis of the live test should indicate which OS are important to support.

Figure 6.21 on page 6.21 illustrates the distribution of different OS used by the auditors in the live test to access the backchannel web application. One third of the audience is using Microsoft Windows. Followed by Apples Macintosh OS with a usage share of 19.3%. Closely followed by Android the Google mobile OS and iOS the mobile OS of Apple. On the lower

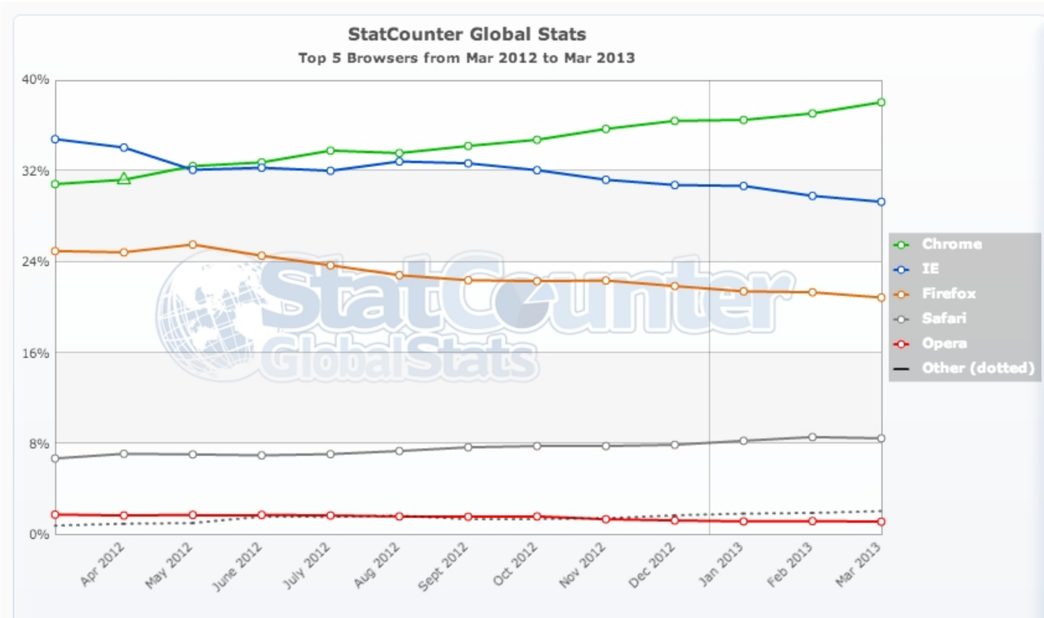


Figure 6.20: Worldwide browser usage trend by StatCounter for the last year [Statcounter.com, 2013]. The most popular browser Chrome in this chart is also the most used browser in our live test illustrated in figure 6.19

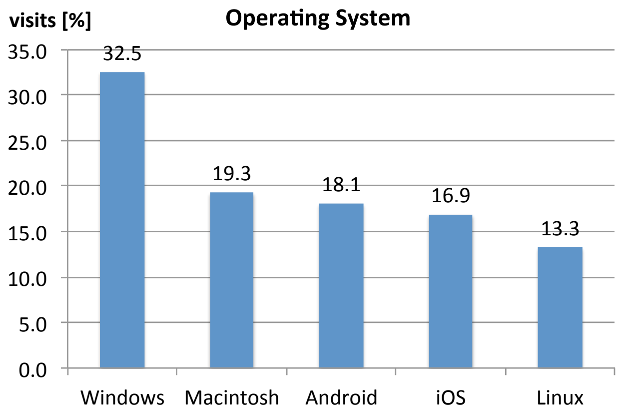


Figure 6.21: Operating systems used by the audience based on Google Analytics.

end is Linux still with 13.3%. Comparing the global statistics from Statcounter.com [2013] (see figure 6.22 on page 107) the differences can be analyzed. Windows is the most used OS worldwide and during the live test. The distance to others OS is much lower in the live test. This might again be related to the computer science audience. Focusing on future work in mobile development, analyzing usage of mobile operating systems is important. While Blackberry OS and Symbian OS are not present in the live test, Android and iOS are almost equally shared.

Development of native apps for mobile platforms is strongly dependent on the OS version. Mobile OS versions are released more often than, for example, new Windows versions. The OS API changes or improves with every version. It is crucial to know which version is the minimum version, which the application should support and on which version the focus is set.

Figure 6.23 on page 108 illustrates the usage of the different OS versions. Focusing on the mobile OS in the live test, the recent Android version Jelly Bean 4.2.x has the main share of Android with 46.7%. Statistics for the iOS Platform cannot be compared because of the lack of official usage data from Apple. Therefore, the Android usage statistics were analyzed.

Compared to the provided usage statistics by Google [Google, 2013a] in

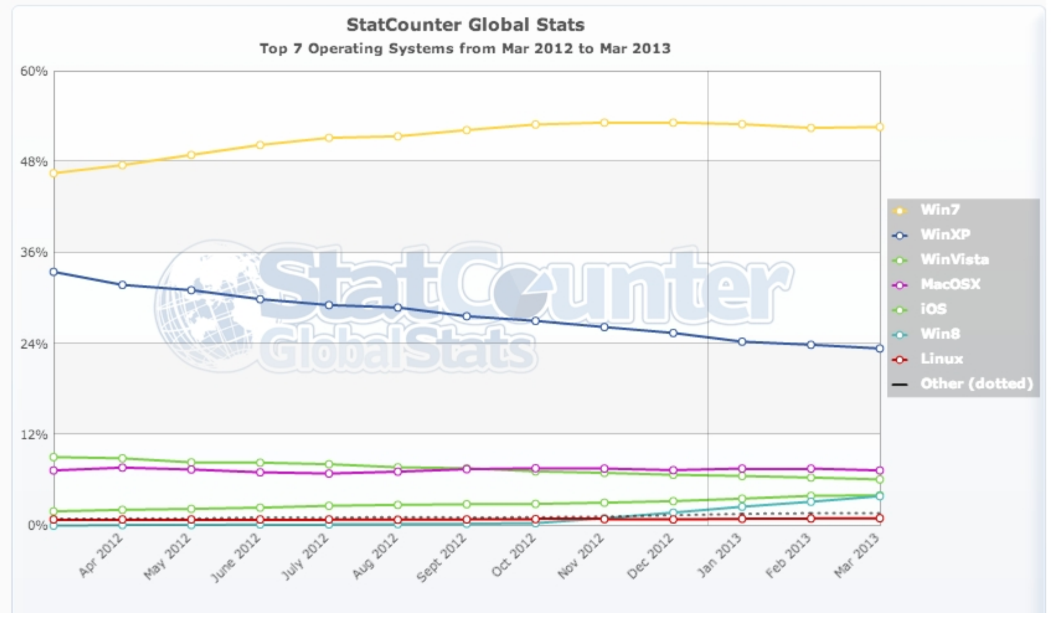


Figure 6.22: Worldwide operating system usage trend by StatCounter for the last year [Statcounter.com, 2013].

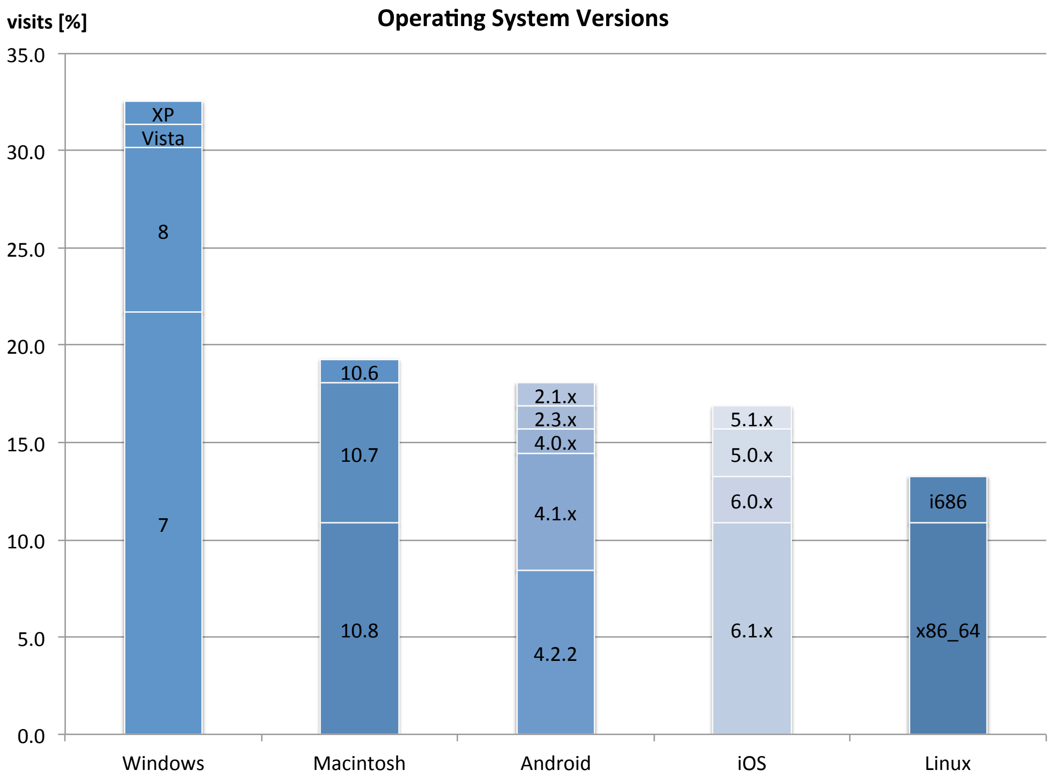


Figure 6.23: Operating system versions used by the audience based on Google Analytics.

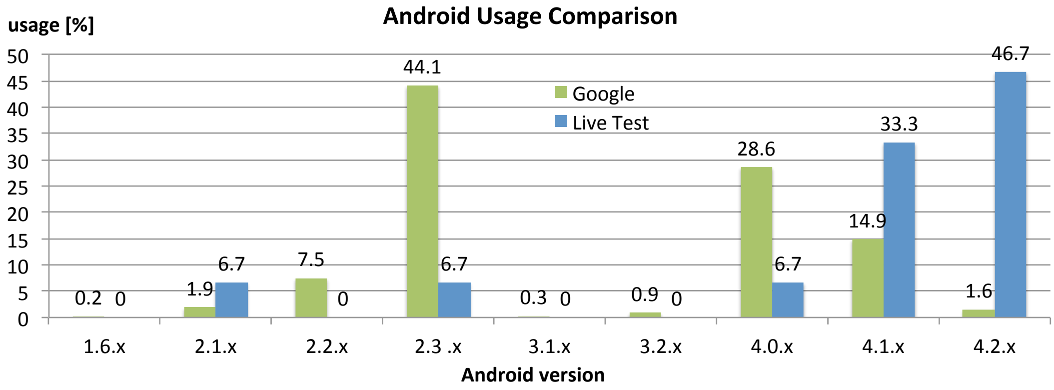


Figure 6.24: Official Android version usage statistics as of 4/2/2013 [Google, 2013a] compared to the live test android statistics.

figure 6.24 there is a significant difference. Jelly Bean has a share of just 1.6% in the statistics by Google. That is a factor of approximately 30 compared to the live test. It can be assumed that the distribution of the OS and the version of the OS is highly dependent on the user base, which is in the live test case, the audience which is dependent on the lecture content.

## Device

Another interesting indicator is the type of devices that the audience is using to participate in the backchannel. Based on the datasets two types of devices are distinguished. One category is *mobile devices* which includes pocket-sized devices mostly handled by touch input. For example, tablets and smartphones fit in this category. The other category is *laptops*, which also includes Ultrabooks and similar devices. Google Analytics also includes desktop devices in the raw data, which were not present during the live test in the lecture hall.

As shown in figure 6.25 on page 110 around two-thirds of all devices are laptops and around one third are mobile devices. These numbers might also be related to the audience in this lecture, which is highly related to computer science and is used to bring a laptop to university courses. It can

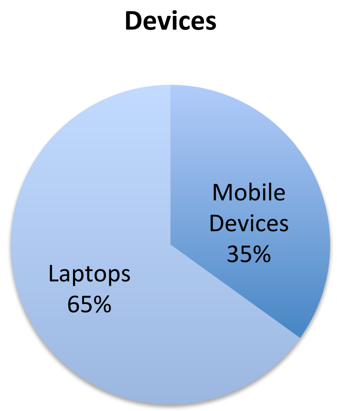


Figure 6.25: Devices used by the audience based on google analytics.

be assumed that in other lectures, the number of laptops might be significantly lower, and the mobile devices will increase its share. Even tough there is a statistic from Statcounter.com [2013] (see figure 6.26 on page 111) comparing the usage of mobile vs. desktop computers these two comparisons cannot be set in reasonable relation. The reason for this is, even though StatCounter uses the same device categorization as stated here, they are capturing website access to acquire their data. The situation in the lecture hall is different, desktop systems are usually not available and taking a laptop to the lecture is in Austria nowadays at least common for computer science related university programs. Therefore, the mobile devices as they are defined here are having a larger share compared to the StatCounter analysis and might even grow on non-computer science related lectures.

As it can be seen in the previous section, global statistics might not be representative for the use cases of the backchannel system. Further, analysis and test lectures need to be done to get an impression how the behavior and characteristics are different for the audience. Especially the lecture situation makes the use of web applications different. For example, thinking-aloud tests under laboratory conditions might differ extensively to the real lecture situation. A main difference to laboratory conditions is the needed attention during lecture. Backchannel systems need to be non-distracting

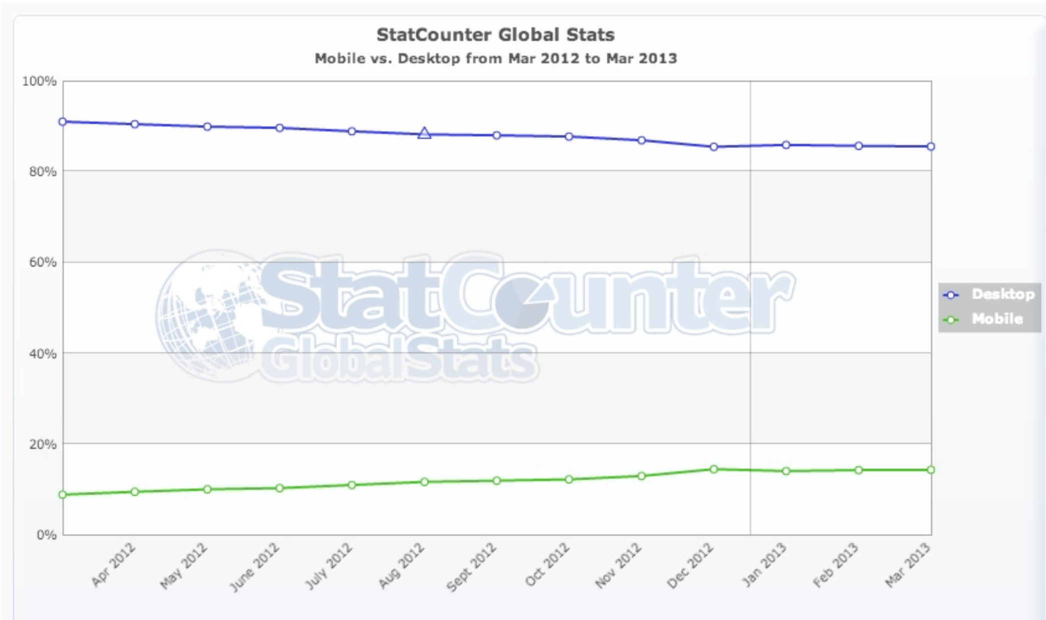


Figure 6.26: Comparison of the worldwide usage of mobile vs. desktop/laptop devices for the last year by Statcounter.com [2013].

as they should support the lecture not undermine it by distracting the audience.

# 7 Conclusion

This work presented an approach how to develop a state of the art quantitative digital backchannel system in the context of agile teaching. This system should support the BYOD policy to satisfy the trend of today where students and lecturers have their own mobile devices with them.

Because quantitative backchannel systems are less researched than qualitative ones, most of the foundation for this work is based on the evaluation of the few existing services and products, which support quantitative backchanneling methods. Based on this evaluation, the requirements, for the first iteration of the new state of the art quantitative backchannel system, were formulated.

- Constant and continuous activity
- Reduce distraction
- Usability
- Simplicity
- Support BYOD policy
- Responsive Design
- Auditor impact
- Reduce information
- Cross-platform capabilities
- Internationalization
- Maximize meaningful information

These requirements lead to the architecture of the new system. A client-server architecture is chosen. Different cross-platform approaches are discussed to fulfill the requirements regarding BYOD and cross-platform capabilities. A web-based approach is chosen to maximize the supported devices and platforms with the least effort. The technology stack uses HTML5 and JavaScript extensively on the client side. The server side is driven from a WSGI web application implemented in python. Data storage is realized with MongoDB and Redis. Socket.IO, a WebSocket JavaScript library, was used for instant and continuous communication between the client and the server side.

The quantitative dimensions for the backchannel system are inspired by the existing solutions. Three dimensions are implemented in the first iteration: *happiness*, *comprehension*, and *presentation speed*.

A different approach is used for the auditor interface. Considering the requirements *auditor impact*, *internationalization*, and *maximization of meaningful information* leads to an image based avatar approach. The auditor and the audience are represented as stickmen, which adapt their facial expressions based on the perception. The lecturer interface is not the focus in this work and is implemented similar to the auditor interface.

The implementation of the first iteration is tested during a 40-minute lecture unit at the Graz University of Technology. Feedback from the audience addressed the visualization of the perception, the loading performance, and the connection state. One feature request addresses a qualitative shoutbox module for sending comments to the lecturer.

The acquired quantitative data during this lecture unit are analyzed and interpreted. The focus is on usage characteristics to verify the requirements. Around 100 from 133 auditors (approx. 75%) in the lecture hall participated in the backchannel. While 12% of the participating auditors were passive, 80% have voted more than 3 times. The activity over time decreases significantly while the average is 42.7 number of votes per minute. The activity per auditor only slightly decreases, which might lead to the conclusion that if an auditor keeps participating over time, the activity of the auditor only slightly decreases. Three dimensions are available for continuous voting. The distribution across these three dimensions is almost equal. This leads to the assumption that if auditors vote, they vote for all dimensions.

Considering the chosen slider UI input element to improve the dimension resolution, the following results were measured. While 28% percent of all vote values are in the neutral area and 37% percent are minimum or maximum values, more than one-third (35%) voted in the value range between minimum, neutral, and maximum. This supports the requirement that also a higher input resolution than binary is used by the auditors.

Although, the lecturer visualization of the data was not focused for the implementation, different charts were discussed during the data analysis. It is concluded that aging of the data improves the visualization regarding the recentness of votes.

Based on the Google Analytics dataset technical characteristics such as device type, screen size, OS and browser of the used auditor devices are analyzed and compared to global statistics. While some characteristics such as screen size are similar distributed as in the global statistics, some characteristics differ highly. This might come from the specific environment in the lecture hall.

# 8 Future perspectives

While the first iteration is already implemented and approved the first live test many ideas and improvements came up during this work.

The lecturer interface supports the basic use case but has room for improvements. The visualization as discussed could be implemented to visualize the time dimension. The discussed aging algorithm needs to be implemented and tested. The aging algorithm can generate a chart, which improves the visualization of the recentness of information for the lecturer.

While the *non-distracting* requirement is also important for the lecturer, implementation of the lecturer interface on devices such as Google Glass and Smartwatches should be evaluated.

Regarding the auditor interface, the legend for the *presentation speed* dimension should be improved. Feedback suggests the implementation of a time-based visualization of the audience perception for the auditor. A connection indicator might also improve the usability if a connection loss occurs as happened during the live test.

Performance optimizations for the whole systems can be addressed in future iterations. While some improvements are already implemented, there is room for further optimization.

Different new dimensions could be tested if they can generate important information for the lecturer. The number of dimensions can also be considered as an open question. While the implemented three dimensions seemed reasonable for the audience, how many dimensions can be added without distracting or demotivating the audience?

The collective interpretation of dimensions might also be of interest for the lecturer. In the first iteration, all dimensions are processed separately.

Processing the collective information of all dimensions might generate additional important information for the lecturer.

The web application seemed a reasonable approach for the requirements and the first iteration. Additional native applications could be implemented to improve the user experience through native UI elements on the main used platforms. The hybrid approach might also be considered for future iterations.

# Acronyms

**API** Application Programming Interface. 34–36, 38, 103, 106
**ARS** Audience Response System. 1, 2, 5–9, 11, 13–15, 17, 19, 21, 24, 27, 28, 34

**BYOD** Bring Your Own Device. 5, 13, 14, 19, 27, 30, 32, 34, 39, 48, 55, 69, 113

**CSS** Cascading Style Sheets. 60, 62, 83, 84

**DOM** Document Object Model. 56

**HTTP** Hypertext Transfer Protocol. 40, 41, 83, 84

**MVC** Model View Controller. 40

**OS** Operating System. 13, 14, 32, 34–38, 104, 106, 109, 115

**UI** User Interface. 30, 37, 38, 40, 48, 49, 52, 54, 55, 64, 103, 114, 118
**UUID** Universally Unique Identifier. 85

**VPN** Virtual Private Network. 77, 79, 82, 85

**WSGI** Web Server Gateway Interface. 39

# Bibliography


Abrahamson, Louis (Feb. 2006). "A Brief History of Networked Classrooms: Effects, Cases, Pedagogy, and implications." English. In: *Audience Response Systems in Higher Education: Applications and Cases*. Ed. by David Banks. IGI Global. ISBN: 9781591409472. DOI: 10.4018/978-1-59140-947-2 (cit. on pp. 1, 14, 17).

Adobe Systems (2013). *Phone Gap*. URL: http://phonegap.com/, visited on 04/04/2013 (cit. on p. 38).

Alexander, Cara J. et al. (2009). "Assessing the integration of audience response system technology in teaching of anatomical sciences." In: *Anatomical sciences education* 2.4, pp. 160–6. ISSN: 1935-9780. DOI: 10.1002/ase.99 (cit. on pp. 8, 17).

Atkinson, Cliff (2009). *The Backchannel: How Audiences are Using Twitter and Social Media and Changing Presentations Forever*. New Riders, p. 240. ISBN: 0321659511 (cit. on pp. 1, 8, 9, 19).

Banks, David A. (2006). *Audience Response Systems in Higher Education: Applications and Cases*. Information Science Publishing. ISBN: 1591409470 (cit. on p. 1).

Baumgart, Daniel and Alexander Pohl (2011). "Providing guidance on Backstage, a novel digital backchannel for large class teaching." In: *Education in a technological world: communication current and emerging research and technological efforts*, pp. 364–371 (cit. on p. 1).

Bazon, Mihai (2013). *UglifyJS*. URL: https://github.com/mishoo/UglifyJS, visited on 04/02/2013 (cit. on p. 83).

Beatty, Ian D., William J. Gerace, et al. (2006). "Designing effective questions for classroom response system teaching." In: *American Journal of Physics* 74.1, p. 31. ISSN: 00029505. DOI: 10.1119/1.2121753 (cit. on p. 5).

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*, pp. 1–19. ISBN: 1591409470 (cit. on pp. 5, 7).

Beck, Kent (2006). "Extreme Programming Explained." In: *Embrace change* (cit. on p. 30).

Beck, Kent et al. (2001). *Agile Manifesto*. URL: http://agilemanifesto.org/, visited on 04/04/2013 (cit. on p. 28).

Black, Paul and Dylan Wiliam (1998). "Assessment and classroom learning." In: *Assessment in Education* (cit. on p. 6).

Boud, David (Nov. 2000). "Sustainable Assessment: Rethinking assessment for the learning society." In: *Studies in Continuing Education* 22.2, pp. 151–167. ISSN: 0158-037X. DOI: 10.1080/713695728 (cit. on p. 6).

Bransford, John D., Ann L. Brown, and Rodney R. Cocking (2000). *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. National Academies Press, p. 347. ISBN: 0309070368 (cit. on p. 6).

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. 1, 6, 7).

Caniuse (2013). *CanIUse Websockets*. URL: http://caniuse.com/websockets, visited on 04/02/2013 (cit. on p. 103).

Carrot & Company GmbH (2013). *Backchannel*. URL: http://backchannel.cnc.io/, visited on 04/16/2013 (cit. on p. 27).

Chesneau, Benoit (2013). *Gunicorn*. URL: http://gunicorn.org/, visited on 04/16/2013 (cit. on p. 40).

Chun, Andy Hon Wai (2004). "The agile teaching/learning methodology and its e-learning platform." In: *Advances in Web-Based Learning–ICWL 2004* 3143, pp. 11–18 (cit. on p. 17).

Collins, Jannette (Sept. 2008). "Audience response systems: technology to engage learners." In: *Journal of the American College of Radiology : JACR* 5.9, pp. 993–1000. ISSN: 1558-349X. DOI: 10.1016/j.jacr.2008.04.008 (cit. on p. 17).

CSS Tricks (2013). *CSS Sprites*. URL: http://css-tricks.com/css-sprites/, visited on 04/06/2013 (cit. on p. 60).

Digia (2013). *QT*. URL: http://qt.digia.com, visited on 04/16/2013 (cit. on p. 35).

Ebner, Martin (2009). "Introducing live microblogging: how single presentations can be enhanced by the mass." In: *Journal of research in innovative teaching*, pp. 91–101 (cit. on pp. 1, 17, 19).

Ebner, Martin (2011). "Is Twitter a Tool for Mass-Education?" In: *International Conference on Student Mobility and ICT* (cit. on p. 10).

Ebner, Martin, Walther Nagler, and Martin Schön (2012). "Have They Changed? Five Years of Survey on Academic Net-Generation." In: *World Conference on Educational Media and Technology* 8, pp. 21–31 (cit. on p. 13).

Elliman, D. G. (2006). *A System to support Interactive Teaching in the Lecture Theatre*. Tech. rep., p. 4 (cit. on p. 17).

Emery, Scott (2012). *Factors for Consideration when Developing a Bring Your Own Device (BYOD) Strategy in Higher Education*. Tech. rep. July 2012 (cit. on p. 13).

Flint, Michele S. (2012). "Formative Assessment to Support Agile Teaching and Active Learning : A Case in Auditing Education Using the I-clicker Classroom Response System." In: *Journal for Excellence in Business Education* 1.1 (cit. on p. 29).

Freij, Nabil (2012). *Do's and Don'ts in Software Development before Internationalization (i18n)*. URL: http://software.intel.com/en-us/blogs/2012/03/12/dos-and-donts-in-software-development-before-internationalization-i18n, visited on 04/04/2013 (cit. on p. 32).

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. 17).

Furfero, David (2013). *jQuery UI Touch Punch*. URL: http://touchpunch.furf.com/, visited on 04/06/2013 (cit. on p. 55).

Garg, Devendra (1975). *Experiments with a Computerized Response System [microform] : A Favorable Experience*. Distributed by ERIC Clearinghouse, [Washington, D.C.] : 17 p. (Cit. on p. 17).

Gartner (2012). *Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012*. URL: http://www.gartner.com/newsroom/id/2335616, visited on 04/04/2013 (cit. on p. 35).

Gedye, Sharon (2003). "Formative assessment and feedback: a review." In: *Planet* 23, pp. 40–45 (cit. on p. 6).

Gehlen-Baum, Vera, Alexander Pohl, and François Bry (2011). "Assessing Backstage — A Backchannel for Collaborative Learning in Large Classes A Formative Study on its Usability and Influence on Students'

Questioning." In: *Interactive Collaborative Learning (ICL)*. September, pp. 154–160. ISBN: 9781457717475 (cit. on pp. 1, 10).

Gehlen-Baum, Vera, Alexander Pohl, Armin Weinberger, et al. (2012). "Backstage – Designing a Backchannel for Large Lectures Computer-Supported Learning Activities in Large Backstage – An Environment to Foster Social Learning." In: *European conference on Technology Enhanced Learning*, pp. 459–464 (cit. on p. 1).

Goal Smashers Dev, Team (2013). *Clean CSS*. URL: https://github.com/GoalSmashers/clean-css, visited on 04/02/2013 (cit. on p. 83).

Google (2013a). *Android Usage Statistics*. URL: http://developer.android.com/about/dashboards/index.html\#Platform, visited on 04/02/2013 (cit. on pp. 106, 109).

Google (2013b). *AngularJS*. URL: http://angularjs.org/, visited on 04/07/2013 (cit. on p. 40).

Google (2013c). *Google Analytics Accuracy Discussion*. URL: http://productforums.google.com/forum/\#!topic/analytics/vek8cNEyusw, visited on 04/01/2013 (cit. on p. 84).

Google (2013d). *Google Analytics Opt-out*. Visited on 04/01/2013 (cit. on p. 86).

Google (2013e). *Google Earth*. URL: earth.google.com, visited on 04/06/2013 (cit. on p. 69).

Google (2013f). *Google Glass*. URL: http://www.google.com/glass/start/, visited on 04/14/2013 (cit. on p. 101).

Groh, Fabian (2012). "Gamification: State of the Art Definition and Utilization." In: *Research Trends in Media Informatics* (cit. on p. 29).

Grunt (2013). *Grunt*. URL: http://gruntjs.com/, visited on 04/17/2013 (cit. on p. 83).

Holzinger, Andreas et al. (2001). "TRIANGLE: A Multi-Media test-bed for examining incidental learning, motivation and the Tamagotchi-Effect within a Game-Show like Computer Based Learning." In: *World Conference on Educational Media and Technology*. Imi, pp. 2–7 (cit. on p. 57).

Ideaworks3D Limited (2013). *Marmalade*. URL: http://www.madewithmarmalade.com/, visited on 04/16/2013 (cit. on p. 35).

IETF (2013). *RFC6202*. URL: http://tools.ietf.org/html/rfc6202, visited on 04/16/2013 (cit. on p. 103).

JQuery (2013). *jQuery UI*. URL: http://jqueryui.com/, visited on 04/02/2013 (cit. on pp. 55, 82).

JQuery Foundation (2013). *jQuery Mobile*. URL: http://jquerymobile.com/, visited on 04/05/2013 (cit. on p. 39).

Kay, Robin H. and Ann LeSage (2009). "A strategic assessment of audience response systems used in higher education." In: *Australasian Journal of Educational Technology* 25.2, pp. 235–249 (cit. on p. 1).

Krasner, Glenn E. and Stephen T. Pope (1988). "A description of the model-view-controller user interface paradigm in the smalltalk-80 system." In: *Journal of Object Oriented Programming* 1 (cit. on p. 40).

Lam, Shun Leung et al. (2011). "Classroom communication on mobile phones–first experiences with web-based 'clicker'system." In: *Australian Society for Computers in Learning in Tertiary Education* 1993, pp. 763–777 (cit. on p. 17).

Larman, Craig (2003). *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, p. 368. ISBN: 0131111558 (cit. on p. 32).

Lecture Tools (2013). *Lecture Tools*. URL: http://www.lecturetools.com/, visited on 04/16/2013 (cit. on pp. 19, 21–23, 27, 33, 92).

Lehni, Juerg and Jonathan Puckey (2013). *Paper.js*. URL: http://paperjs.org/, visited on 04/06/2013 (cit. on p. 69).

Logicalis (2012). *BYOD : an emerging market trend in more ways than one*. Tech. rep., pp. 1–12 (cit. on pp. 13, 14).

Marcotte, Ethan (2011). *Responsive web design*. ISBN: 9780984442577 (cit. on p. 31).

McManus, Dean A. (2001). "The two paradigms of education and the peer review of teaching." In: *Journal of Geoscience Education*, pp. 423–434 (cit. on p. 5).

Miller, George A. (Apr. 1994). "The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956." In: *Psychological review* 101.2, pp. 343–52. ISSN: 0033-295X (cit. on pp. 11, 29, 31).

MyTU (2013). *MyTU Freiberg*. URL: http://mytu.tu-freiberg.de, visited on 04/05/2013 (cit. on pp. 19, 20, 27, 33, 45).

NK Labs Inc (2013). *UnderstoodIt*. URL: https://understoodit.com/, visited on 04/03/2013 (cit. on pp. 19, 21, 25–27, 33, 46, 47, 65, 92).

Oxford University Press (2013). *Oxford Dictionaries*. URL: http://oxforddictionaries.com/, visited on 04/16/2013 (cit. on p. 58).

Pichler, Karin M. (2013). "Relevant Information and Information Visualizations for Lecturers in Web-Based Audience Response Systems." Master's Thesis. Graz University of Technology (cit. on p. 27).

Pigeon Lab Pte Ltd (2013). *Pigeonhole Live*. URL: http://www.pigeonhole.sg/, visited on 04/05/2013 (cit. on p. 27).

Pohl, Alexander, Vera Gehlen-Baum, and François Bry (2011). "Introducing Backstage – a digital backchannel for large class lectures." In: *Interactive Technology and Smart Education* 8.3, pp. 186–200. ISSN: 1741-5659. DOI: 10.1108/17415651111165410 (cit. on pp. 1, 10, 19, 27, 80).

Pohl, Alexander, Vera Gehlen-Baum, and François Bry (2012). "Enhancing the Digital Backchannel Backstage on the Basis of a Formative User Study." In: *International Journal of Emerging Technologies in Learning* 7, pp. 33–41 (cit. on p. 2).

Purgathofer, Peter (2008). "Massive" Multiplayer" E-Learning." In: *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (cit. on pp. 1, 17, 19).

Pylons Project (2013). *Pyramid*. URL: http://www.pylonsproject.org/, visited on 04/16/2013 (cit. on p. 40).

Reinhardt, Wolfgang et al. (2009). "How people are using Twitter during conferences." In: *Edumedia Conference*. March (cit. on p. 9).

RequireJS (2013). *RequireJS*. URL: http://requirejs.org/, visited on 04/02/2013 (cit. on p. 83).

Ricardo Cabello, Miguel (2013). *Three.js*. URL: http://threejs.org/, visited on 04/06/2013 (cit. on p. 69).

Richardson, Leonard and Sam Ruby (2008). *RESTful Web Services*. O'Reilly Media, Inc., p. 454. ISBN: 0596554605 (cit. on pp. 34, 39).

Roschelle, Jeremy (2003). "Keynote paper: Unlocking the learning value of wireless mobile devices." In: *Journal of Computer Assisted Learning* 19.3, pp. 260–272. ISSN: 02664909. DOI: 10.1046/j.0266-4909.2003.00028.x (cit. on p. 17).

Shambles.net (2013). *Shambles Backchannel List*. URL: http://www.shambles.net/pages/staff/BCtools/, visited on 04/05/2013 (cit. on p. 19).

Sharp, Helen, Yvonne Rogers, and Jenny Preece (2007). *Interaction Design: Beyond Human-Computer Interaction*. Wiley, p. 800. ISBN: 0470018666 (cit. on p. 30).

Spraylight GmbH (2013). *Murl Engine*. URL: http://murlengine.com/, visited on 04/16/2013 (cit. on p. 35).

Statcounter (2013). *Statcounter Screen Resolution*. URL: http://gs.statcounter.com/\#resolution-ww-monthly-201204-201304, visited on 04/02/2013 (cit. on p. 102).

Statcounter.com (2013). *Statcounter.com*. URL: http://gs.statcounter.com/, visited on 04/02/2013 (cit. on pp. 32, 35, 36, 103, 105–107, 110, 111).

Strauch, Christof (2011). *NoSQL databases*. Tech. rep. (cit. on p. 41).

Sysoev, Igor (2013). *NGINX*. URL: http://nginx.org/, visited on 04/16/2013 (cit. on p. 40).

Thomson, Gordon (2012). "FEATURE BYOD : enabling the chaos." In: *Network Security* 2012.2, pp. 5 –8 (cit. on p. 13).

TU Freiberg (2013). *TU Freiberg*. URL: http://tu-freiberg.de, visited on 04/05/2013 (cit. on p. 19).

Twitter (2013). *Twitter*. URL: https://twitter.com/, visited on 04/05/2013 (cit. on pp. 8, 9, 27).

Twitter Bootstrap (2013). *Twitter Bootstrap*. URL: http://twitter.github.com/bootstrap/, visited on 04/02/2013 (cit. on pp. 39, 84).

Varnish Software (2013). *Varnish*. URL: https://www.varnish-cache.org, visited on 04/16/2013 (cit. on p. 41).

W3C (2013). *HTML 5 Spec*. URL: http://www.w3.org/html/wg/drafts/html/master/, visited on 04/04/2013 (cit. on p. 36).

Wright, Stephen C., Donald M. Taylor, and Fathali M. Moghaddam (1990). "Responding to membership in a disadvantaged group: From acceptance to collective protest." In: *Journal of Personality and Social Psychology* 58.6, pp. 994–1003. ISSN: 0022-3514. DOI: 10.1037//0022-3514.58.6.994 (cit. on p. 31).

Yahoo (2013a). *Best Practices for Speeding Up Your Web Site*. URL: http://developer.yahoo.com/performance/rules.html, visited on 04/02/2013 (cit. on pp. 82, 83).

Yahoo (2013b). *SmushIt*. URL: http://www.smushit.com/ysmush.it/, visited on 04/02/2013 (cit. on p. 82).

Yardi, Sarita (2006). "The role of the backchannel in collaborative learning environments." In: *international conference on Learning sciences* October 2004, pp. 852–858 (cit. on pp. 1, 7, 8, 17, 19).