Tweedback: Online Feedback System for Large Lectures

Martin Garbe, Jonas Vetterick and Clemens H. Cap

Department of Computer Science
University of Rostock
Albert-Einstein-Str. 22
18059 Rostock, Germany
{martin.garbe,jonas.vetterick,clemens.cap}@uni-rostock.de

Abstract: The possibility to give feedback is required to improve quality of lectures. Tweedback is an online Classroom Response System enabling feedback especially for large audiences. Lecturers can choose between three different types of feedback. First, Peer Instruction is implemented for asking multiple choice questions. Second, on a Chatwall the audience asks questions and votes on them. Third, Speech Parameters of the lecturer's talk can be rated. Example parameters are "speed" and "understanding". Users access Tweedback using their mobile devices. The system was designed as a scalable mobile web application. Parts of Tweedback are already actively used in various courses.

1 Motivation

Lecturers getting feedback from the audiences can advance the quality of their courses. In small groups feedback can be given by every person directly to the teacher. In large groups verbal feedback is often restricted to a small part of participants. The reason for that is that on the one hand people are afraid of talking in front of a large auditorium and on the other hand requesting verbal feedback from every person is a very time consuming task.

Techniques to support feedback in such scenarios are called classroom response systems (CRS). One form of feedback are multiple choice questions. Here a teacher explains some situation or facts, e.g. a diagnosis of patient's symptoms in medicine. At the end of lecture the teacher can present a multiple choice question and ask the students to vote for a solution. Some vendors have specialized on this particular case and offer voting devices. Examples for these hardware solutions are "QwizdonQ5" [Qwi12], "Powervote" [Pow13] and "voting4life" [Art13]. But after each lecture the devices have to be collected and charged in one place. This administrative task is additional work for lecturers.

Due to fast technological development in the area of mobile devices most of the students own a smartphone or other mobile device which can be connected to the Internet. This leads to the idea of integrating user's mobile devices into the feedback system. One advantage of this idea is the saving of administrative costs compared to dedicated hardware devices.

Our Tweedback system[1] [VGC13] enables feedback for large audiences (more than 100 participants). The system is implemented as a web application, so mobile device users can easily connect to this system. Mainly three types of feedback are supported. First, Peer Instruction (PI) which is similar to the multiple choice question scenario above. Second, a Chatwall (CW) is implemented to enable users to ask question and vote on them. Third, Speech Parameters (SP) allow users to rate the lecturer's speed of talk and the understanding.

2 State of the Art

This section presents an overview of existing classroom response systems which either implement one kind of live feedback or a composition of two or more. We introduce first the implementations of the Peer Instruction concept and then solutions for giving feedback based on a lecturer's Speech Parameters.

The University of Paderborn realized the Peer Instruction concept as a closed-source web application called PINGO [RSM+12]. They built an application that is accessible on any device, mobile or not. Using this application a lecturer, who must have an account at the University of Paderborn, can add surveys during the lecture. A survey thereby exists of a multiple choice question. The participants, mostly students, do not need to have an account and answer anonymously to the lecturer's questions. It is not possible to add free text questions and the source code is not open.

One solution, that also handles just a single kind of live feedback has been developed at the University of Freiberg and is called myTU [Tec12b]. It is available for the mobile operating systems Android and iOS. It is more a manager for student's life than an instrument for live feedback, but students can use it to rate the speed of lecturer's speech.

Combining a Chatwall and the ability to rate the lecturer's Speech Parameter is the idea behind the web application Backstage [GBPWB12], developed and researched by F. Bry. Backstage was designed to investigate the use and the design of a digital backchannel by implementing a prototype that looks and acts like twitter, but is used for a lecture. The students may ask questions to the whole audience, including the lecturer. The lecturer also may ask questions to the students using Backstage. Backstage is focused on large classes, but is used only with a small number of students for the experimental studies (approx. 20 students). Due to its design using a synchronous Java web server as Jetty, it may be slow under heavy use with multiple large lectures (more than 100 participants) in parallel.

A tool developed by the University of Freiburg is SMILE 2.0 [FB12]. It allows students to ask free questions and rate the lecturer's Speech Parameters using a social platform. Furthermore it has the functionality of a Peer Instruction application. Unfortunately it is not open, neither to use nor the source code, so we could not evaluate it.

Most similar to our project is ARSnova [Tec12a] from Technische Hochschule Mittelhessen. One part of the software is lecture assistance which include rating speech param-

[1] http://www.tweedback.de

eters. Here students can rate the speed of talk from “too fast” to “too slow”. It also has a chatwall and an electronic voting system. The software is also open source. This open behavior simplifies cooperation with other similar projects. From the functionality point of view our system is very similar. Differences can be found in implementation and many user interface features.

3 Tweedback System

3.1 Requirements

Feedback can be given in different ways. We concentrate on three categories. The first functional requirement is Peer Instruction (PI). Teachers should be able to start surveys ad-hoc in lecture or start prepared ones. A survey is implemented as a multiple choice question. Results of these surveys should be accessible for the lecturer at all time.

Besides Peer Instruction, students have additional possibilities to give feedback regarding the content of course and form of presentation. Speech Parameters (SP) can be rated throughout the lecture. Example categories are “speed” and “understanding”. A threshold regulates how many votes are needed in a given period of time to notify the lecturer. The teacher will be notified on a small screen about the current situation. Besides the other feature, rating SP is not implemented right now.

The third functional requirement, called Chatwall (CW), is feedback in form of asking particular questions, rating and answering them. For lecturers it is hard to give a presentation and read questions from the audience at the same time. Therefore an unfiltered forwarding of audience questions will not be convenient. The system gives the audience the possibility to ask questions on a Chatwall-like platform. At the beginning questions are proposed and displayed only on students’ devices. After questions have been proposed others can vote for them. This can result in high rated questions concerned by many others. A threshold decides which question will be presented to the lecturer. A log of the Chatwall can also be used by the lecturer afterwards to evaluate his presentation.

Besides the functional features as described above, Tweedback has many other requirements which aim at the best possible user experience. First of all, interaction with the system has to be fast and responsive. Waiting while interacting with the application is decreasing acceptance of users. A second requirement with direct impact on acceptance is an intuitive and minimalistic user interface, that even fits on very small mobile devices. We designed Tweedback having especially laypersons in mind. According to (N)ONLINER Atlas 2012 [Ini12], which represents the situation in Germany, 96.9% of young people (20–29 years) have Internet experience. So we can assume that they are able to use even complicated websites. According to the lecturer’s view, it is assumed that teachers have less experience with the Internet (e.g. 60–69 years, 60.4%). We do not know what people will use our system in the future, how old they are or what preferences they have. So we cannot suppose that teachers are familiar with complex websites and their navigation features. According to the students’ side it can be assumed that there is more experience in dealing with complex websites due to the age of students.

Students use their mobile devices to access Tweedback. It is not possible to control which device will be used. This is a design decision. The advantage is that users maintain their own devices. As a disadvantage much work has to be invested to design the user interface of a live feedback system as flexible as possible to fit any students' device.

With a growing number of users the system needs to be scalable. This is especially a matter of concern when many surveys start at the same time. In these cases many students take part at a survey simultaneously. This causes an access peak. Situations with thousands of connections need to be handled.

For users we want to make the first step of participation in Tweedback as easy as possible, which leads to an anonymous access for students. We do not want users to pass an initial account creation procedure. We believe in success of open systems.

To generate additional value compared to "offline" feedback, users cannot be blamed for wrong answers. It is a common phenomenon that people do not participate in surveys because they are afraid to say wrong answers in the front of others. But these people are able to participate in a feedback system when users are anonymous. This is the second reason why anonymous access is preferred.

Finally the system has to be modular to easily implement new forms of audience interaction in the future.

3.2 Architecture

In order to provide an application for many mobile devices, it is sensible to build a web application that simply serves HTML. Because most mobile devices are able to render HTML5, Javascript and CSS, there is no need to develop a native app for all the mobile operating systems (namely the most common used: Android, iOS and Windows Phone) [KG12]. By using just one browser-related client interface, we do not have to manage and maintain multiple different platform variants. Even if there are frameworks that promise to handle these issues, there would always be multiple different code repositories. This would unnecessarily increase the development effort and support.

Furthermore the user interface shipped by the web application should not only fit on most kinds of screens (for example on a mobile device and/or a notebook), but it should be easy to use, too. While the first can be assumed by using a well-known framework for web user interfaces, the last is hard to achieve. Our approach is to use the bootstrap framework from twitter. It provides a set of common user interface elements, such as a header bar or a button, and organizes the way it is visually structured. Additionally we want to subdivide the user interface into different views, in such a way that each feature has its own view. Offering the lecturer a choice of features, students are able to see which feature is enabled by recognizing which view is currently activated. A feature implements one of the previously explained functional requirements, namely the Peer Instruction, the Chatwall and Speech Parameter.

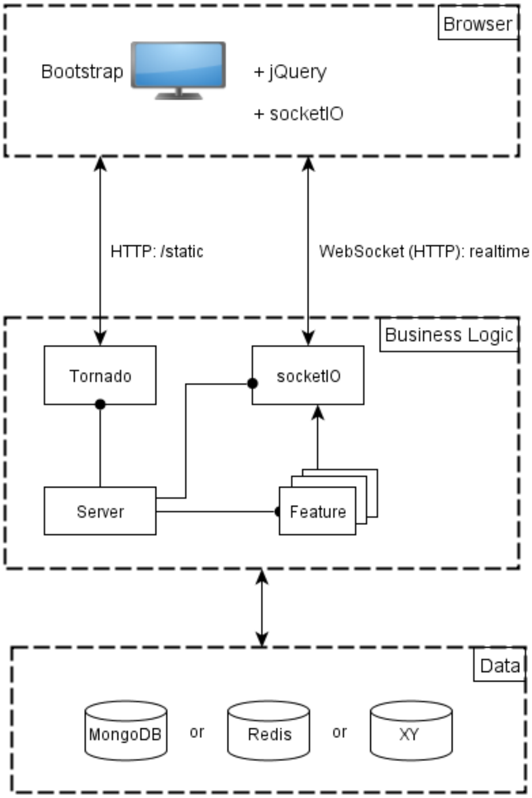

Figure 1: Technology Stack

To implement anonymity there has to be a concept to ensure that a lecturer cannot trace back the students' id. Technically we solve this issue by allowing everybody to participate. Nobody has to register herself or login with her university account. Another fact arises with the ability to be anonymous. It is maybe possible to manipulate the system by overusing a function or trying to distract the lecturer. We prevent such manipulations by defining certain thresholds and time locks. A student for example cannot ask multiple questions at a short time (<1 second), so we disable her functionality to ask questions right after a question is asked.

Responsiveness and scalability are difficult to achieve, so we need sophisticated technologies to solve these requirements. Fortunately both can be handled at once by using an asynchronous web server working with two-way communication channel in tandem. Investigating numerous solutions for asynchronous web servers, we decided to use the tornado web server, developed by facebook. It is event-based, well-known for stability and performs even on thousands of concurrent connections. It is possible to handle more than 5000 connections simultaneously on commodity hardware. A two-way communication

Figure 2: Audience's view of multiple choice questions.

channel is necessary, because both, the server and the client, have to send messages to each other. The client hereby is the browser, of the lecturer as well as of the students. Imagine for example the use of the Chatwall, where one student asks a question. To show this message to all the other students, it has to be delivered to their browsers. Here the student's browser sends the question to the server. Then the server has to push this message to all connected browsers. Without the use of a two-way communication this would not be possible.

We use the library socketIO as the two way communication layer. It makes use of WebSockets, defined in HTML5. Furthermore, it is well implemented for Javascript and for the tornado web server. Additionally, it has several fallbacks for browsers not supporting Websockets. These fallbacks make use of AJAX to emulate a two-way communication over HTTP 1.1.

Figure 1 summarizes the technology stack previously presented. All permanent data will be stored in a non-relational database, because there is a flexible scheme necessary to develop this web application.

3.3 Feedback from Users

At the moment, as of June 2013, Peer Instruction and Chatwall are implemented. The way of implementing Peer Instruction is very clear. An example view of our system is presented in Figure 2 and 3. The audience view with a multiple choice dialog is shown in Figure 2. Lecturers see results of the voting as show in Figure 3. On the top of the screen, teachers can easily switch between Peer Instruction (Quiz) and Chatwall.

In contrast a Chatwall, especially for classroom scenarios, can be designed in many ways. Our first presentation of this feature in front of didactics raised much distrust in usefulness. They argued that many parallel communication channels in a lecture are reducing the attention of the audience. Therefore we made the decision to restrict the Chatwall to a "question wall". People can ask questions and all others can see them. Students cannot answer questions. This is a design decision. They can only vote for questions. An example Chatwall is depicted in Figure 4. The figure shows a situation with anonymous users that asked three questions. These questions were voted by others which find them useful. The input text is limited to 140 characters as it is done by Twitter, too. This limit was chosen

Figure 3: Teacher's view of multiple choice questions.

Figure 4: Audience's view of the Chatwall with voted questions.

because teachers have to get an overview of this Chatwall in a small amount of time. Long descriptions of problems would make this difficult.

The Chatwall is the best tested feature at the moment. It is successfully used by two lecturers of computer science. A lecturer of medical science tested it in one course and plans to use it for a larger one. From now on we start testing the Peer Instruction feature.

There is much feedback regarding our software and the communication infrastructure in general. Feedback which is related to the software features can be summarized as follows:

- Ordering of entries in Chatwall seems to be an issue. Users demand different kinds of ordering. There has to be an option for "order by newest entry" and "order by most voted entry".
- Users also require to hide entries for a better overview.
- There is a need by some users for marking questions as solved.

While testing we had to handle bugs in the software and problems with the communication infrastructure. This second point is a big issue. Classes in computer science are held in rooms with good WLAN infrastructure. Here, no problem regarding connecting to the Tweedback website occurred. Lectures of medical science are given in rooms of the hospital. The hospital has its own WLAN for managing patients data which have high security requirements. Only in some parts of the hospital students can access the Internet using an open WLAN. At the moment we are working together with the local IT administration of the hospital to find a feasible solution.

4 Summary & Outlook

Many people have mobile devices for communicating with others or simply browse the Internet. These devices can be used to give feedback within large classes. The presented live feedback system Tweedback uses these mobile devices. People can give feedback in one of three ways: Peer Instruction, Chatwall and Speech Parameters. Besides these functional requirements, the feedback system has to be fast and responsive, easy to use by laypersons, scalable and compatible to many mobile devices. These requirements are accomplished with user interface frameworks for mobile web which assure compatibility to most user devices. Event-based asynchronous communication is used for scalability. Anonymous access is allowed to assure an easy participation and to give users the chance to give feedback which they would have refused in front of many other people.

The Chatwall is implemented and actively used by two lecturers of computer science. In medical science the system was tested in one course. First tests of the Peer Instruction feature for asking multiple choice questions where done in one computer science lecture. We got much good feedback from students and lecturers. This feedback enables us to further improve our system. A final version is planned for October 2013.

References

[Art13] Art4live GmbH. voting4live: http://www.voting4live.de/ [23.04.2013], 2013.

[FB12] L. Feiten and B. Becker. SMILE smartphones in lectures: Initiating a smartphone-based audience response system as a student project. 4th International Conference on Computer Supported Education (CSEDU 2012), 4 2012.

[GBPWB12] Vera Gehlen-Baum, Alexander Pohl, Armin Weinberger, and François Bry. Backstage Designing a Backchannel for Large Lectures. In Andrew Ravenscroft, Stefanie Lindstaedt, CarlosDelgado Kloos, and Davinia Hernández-Leo, editors, *21st Century Learning for 21st Century Skills*, volume 7563 of *Lecture Notes in Computer Science*, pages 459–464. Springer Berlin Heidelberg, 2012.

[Ini12] Initiative D21 e.V. *(N)ONLINER Atlas 2012. Eine Topographie des digitalen Grabens durch Deutschland.* TNS Infratest, München, 2012.

[KG12] V. Kamboj and H. Gupta. Mobile Operating Systems. *International Journal of Engineering Innovation & Research*, 1(2):115–120, 2012.

[Pow13] Powervote. Powervote: http://www.powervote.com [23.04.2013], 2013.

[Qwi12] Qwizdom Ltd. Qwizdom Q5: http://www.qwizdom.co.uk/q5.php [23.04.2013], 2012.

[RSM+12] Wolfgang Reinhardt, Michael Sievers, Johannes Magenheim, Dennis Kundisch, Philipp Herrmann, Marc Beutner, and Andrea Zoyke. PINGO: Peer Instruction for Very Large Groups. In Andrew Ravenscroft, Stefanie Lindstaedt, CarlosDelgado Kloos, and Davinia Hernández-Leo, editors, *21st Century Learning for 21st Century Skills*, volume 7563 of *Lecture Notes in Computer Science*, pages 507–512. Springer Berlin Heidelberg, 2012.

[Tec12a] Technische Hochschule Mittelhessen. ARSnova: http://blog.mni.thm.de/arsnova/ [01.05.2013], 2012.

[Tec12b] Technische Universität Bergakademie Freiberg. myTU App: http://mytu.tu-freiberg.de/ [23.04.2013], 2012.

[VGC13] Jonas Vetterick, Martin Garbe, and Clemens Cap. Tweedback: A Live Feedback System for Large Audiences. In *CSEDU 2013: 5th International Conference on Computer Supported Education*, 2013.