Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diploma Thesis Nr. 3495

Investigating Conflicts of Interests between Stakeholders of Public Display App Stores

Miriam Greis

Course of Study:	Software Engineering
Examiner:	Prof. Dr. Albrecht Schmidt
Supervisor:	Prof. Dr. Florian Alt, Prof. Dr. Niels Henze
Commenced:	June 3, 2013
Completed:	November 28, 2013
CR-Classification:	H.4.3, H.5.m

Acknowledgements

I would like to thank my adviser Prof. Dr. Albrecht Schmidt for offering me a diploma thesis in the context of PD-Net and for introducing me to the topic. Furthermore, I would like to express my sincere gratitude to my supervisors Prof. Dr. Florian Alt and Prof. Dr. Niels Henze for their valuable suggestions, patience, and immense knowledge that helped me write this diploma thesis. I am sure it would have been impossible without their support.

I would also like to express my gratitude to Prof. Dr. Nigel Davies and M.Sc. Sarah Clinch from Lancaster University for the great cooperation regarding the PD App Store. In particular, I would like to thank Mateusz Mikusz for the great collaboration and support during the development phase of the PD App Store. Special thanks are also given to all other members of the PD-Net project and the members of the HCI group in Stuttgart who always supported my work, the development of the PD App Store and the development of the UniDisplay application.

Also, I would like to thank all survey participants and users of the UniDisplay application for the willingness to participate in the survey and to share their time for the interviews.

Finally, I would like to thank my partner, my family, and my friends who always supported me in reaching my goals and accept me as the person I am.

Abstract

Public displays showing static content such as news and advertisement are very common, but are often ignored of users expecting them to show uninteresting content. Although applications showing dynamic content which could reduce the display blindness exist, distribution channels and concepts for successfully moderating user-generated content on public displays are still missing. We developed an app store for public displays and analyzed six potential moderation strategies for, in particular, user-generated content. To do this, we conducted an online survey. Survey participants mostly specified that content moderation is necessary and preferred a reactive moderation strategy. Moreover, they expected their messages to appear instantly. Therefore, we developed the UniDisplay application and did an in-the-wild-deployment to investigate the impacts of moderation delays on user behavior. We conducted data and content analyses, observations, a survey and interviews with users of our application. Hence, we can show that already a short delay time of messages (90 seconds) may confuse users and that the delay time is not the only strong factor for posting to a display. On this basis, we built a theoretical concept for integrating content moderation in public display app stores. In the long term, understanding the impacts of moderation strategies for public displays will help to build trust between different stakeholders such as display owners and users. In addition, it will cause public display content to become more dynamic and therefore more interesting for the users.

Kurzfassung

Displays im öffentlichen Raum, die statischen Inhalt wie z.B. Nachrichten oder Werbung anzeigen, sind weit verbreitet. Sie werden jedoch sehr oft von Benutzern ignoriert, da diese erwarten, dass nur uninteressanter Inhalt angezeigt wird. Obwohl bereits Anwendungen existieren, die dynamischen Inhalt anzeigen und damit die Aufmerksamkeit der Benutzer erhöhen könnten, fehlen Plattformen zur Verteilung der Anwendungen und Konzepte für die erfolgreiche Moderation von dynamischem Inhalt auf öffentlichen Displays. Wir entwickelten einen App Store für öffentliche Displays und analysierten sechs potentielle Moderationsstrategien, insbesondere für nutzergenerierten Inhalt. Wir führten außerdem eine Online-Umfrage durch. Die Teilnehmer gaben mehrheitlich an, dass Inhaltsmoderation notwendig ist und bevorzugten eine reaktive Moderationsstrategie. Außerdem erwarteten sie, dass Nachrichten sofort auf dem Display erscheinen. Des Weiteren entwickelten wir das UniDisplay und führten Daten- und Inhaltsanalysen, Beobachtungen, eine Umfrage und Interviews mit den Benutzern unserer Anwendung durch. Wir zeigen, dass bereits eine kurze Verzögerungszeit der Nachrichten (90 Sekunden) für Verwirrung bei den Benutzern sorgen kann und dass die Verzögerungszeit nicht der einzige Faktor ist, der beim Veröffentlichen einer Nachricht relevant ist. Auf dieser Basis entwickelten wir ein theoretisches Konzept wie Inhaltsmoderation in App Stores für öffentliche Displays integriert werden kann. Auf lange Sicht wird das Verständnis über die Auswirkungen verschiedener Moderationsstrategien auf Displays dazu beitragen, dass das Vertrauen zwischen den verschiedenen Interessengruppen wie Displaybesitzern und Benutzern wächst. Zusätzlich hilft das Verständnis dabei, dass der Inhalt von öffentlichen Displays dynamischer und damit auch interessanter für die Benutzer wird.

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

A public display is a big screen which is situated in public space, e.g., in train stations, airports or universities. Most people know these screens as platforms for advertisements, news, and weather information. This content shown on current public displays is so called static content. The PD-Net project[1] is a European research project which focuses on large scale networks of pervasive public displays. This thesis has partly been conducted in the context of PD-Net. The vision of the project is to have a network of public displays where the applications and the content are open and come out of different sources. In the context of the project, an app store for public displays was developed, because so far a distribution channel for the applications was missing. The app store is a well-known concept for mobile phone applications and can be adopted to public displays. Nevertheless, the development of an app store for public displays needs additional consideration, because there are known differences between the two fields.

In the environment of a mobile phone app store a customer normally buys an application for his or her device which is only used by that person. The stakeholder relationships are therefore easy to identify. In contrast, the stakeholder relationships of an app store for public displays are much more complicated. Not only content providers (e.g., application developers) and users belong to the stakeholders, also display owners are included because they will in the end buy the applications for their displays. In addition to those three groups of stakeholders also space owners can have a strong interest to influence the content shown on a display. We can furthermore differentiate between users and viewers. While users are persons that interact with a public display, viewers only pass it and might look at the content without interacting.

The large number of stakeholders causes conflicts between those. As display and space owners are mainly interested in advertisements, promoting their café, bar, shop or wherever the display is located, viewers and users are interested in viewing content which is related to them. It is very difficult for content providers to fulfill the needs of all stakeholders. User-generated content would be a solution to make the content of public displays more interesting for users, but it is a problematic topic because display and space owners seldom have the possibility to control the content. Thus, they are afraid of offensive or inappropriate content showing up on their displays which could harm their reputation. Users could in contrast gain a lot from showing user-generated content on public displays, because the content is adapted to their needs. In research, applications showing user-generated content were already developed (for particular applications see section 2.3), but concepts for using moderation strategies on public displays are still missing, even if moderation strategies are often used and well-known in online communities and web forums.

[1] PD-Net project, http://pd-net.org/ (accessed November 26, 2013)

In the context of PD-Net, we developed the front end of an app store for public displays and, in particular, concentrated our further research on user-generated content. We conducted an online survey about authentication mechanisms and content moderation on public displays to get insights in the users' view of the topics. We aimed at understanding the problems of several moderation strategies and the impacts of the choice of a moderation strategy. None of the presented authentication mechanisms in the online survey was able to convince the participants. Most participants saw a necessity for content moderation, but they would expect their content to be displayed instantly. Reactive moderation was rated the best moderation strategy and correlated with space owners and users as potential groups of moderators. Additionally, participants showed a high information demand about the moderation process of a display and the point their content is shown. On this basis, we developed the application UniDisplay to analyze data which was collected during the deployment process. The analysis of the data showed that users are already confused when applying a moderation delay of 90 seconds without informing them about moderation. In addition, a higher moderation delay caused a decrease in the number of posts, but the delay of the first post of each user had no effect on whether or not they posted again. These insights generated through the deployment were used to develop a theoretical concept for a moderation framework included in the app store to help prevent conflicts between the groups of stakeholders.

Our work shows that future public display applications have to address issues such as privacy, delays caused by moderation, and the information demand of the users. A moderation strategy has to be chosen based on its impacts, in regard to possible conflicts between stakeholders, and in regard to different user behavior. Finally, we propose to include a moderation framework in public display app stores to prevent conflicts between stakeholders and standardize moderation.

Outline

The thesis is structured as follows:

Chapter 2 – Related Work: In this chapter, we discuss related work in the field of public displays, in particular, about app stores and applications for public displays. We also present related work about moderation strategies.

Chapter 3 – Stakeholder Analysis for Public Display App Stores: In this chapter, we describe potential reasons for conflicts between stakeholders of public display app stores.

Chapter 4 – Implementing a Public Display App Store: In this chapter, we describe the front end implementation of the PD App Store.

Chapter 5 – Online Survey about Public Displays: In this chapter, we describe the online survey about public displays including topics such as authentication mechanisms and content moderation strategies. We also present the results of the survey.

Chapter 6 – UniDisplay: An Application for Displaying User-Generated Content on Public Displays: In this chapter, we describe the design, implementation and deployment of the UniDisplay application.

Chapter 7 – Analyzing Deployment Data of the UniDisplay: In this chapter, we analyze the data collected during the deployment phase of the UniDisplay application including observations and interviews with users of the application.

Chapter 8 – Preventing Conflicts between Stakeholders with the Help of the PD App Store: In this chapter, we describe a theoretical concept for a moderation framework that could be included in public display app stores to prevent conflicts between stakeholders.

Chapter 9 – Conclusions: In this chapter, we summarize conclusions that we can draw from our work and discuss future work.

2 Related Work

The topic of our research is highly grounded on former research and based on the knowledge about public display applications that were already developed. To understand the state of the art and build on this former research, this chapter discusses related work about public displays with a focus on app stores and applications for public displays. Furthermore, we discuss related work about different moderation strategies in the real and the digital world.

2.1 Public Displays

Public displays are a common research topic and different interaction techniques such as a mouse [GMS+13], touch [OKL+10], [AKB+11], [PKS+08], mobile phones and Bluetooth [MMH07], [CDF+05], [JOIH08], [AKB+11], web interfaces [TCF+07], [TC09], [MRR11], gestures [SOC08], [RML11], or batches [MCL01], [RDS02] have been used and discussed in related literature. Particular applications often offer more than one interaction technique. Alt et al. [ASKS13], e.g., also conducted a user study that compares different interaction techniques: direct touch, mobile phone-based interaction and paper-based techniques. They show that the preferred interaction technique depends on its simplicity and the age, the current situation, and the privacy concerns of the users.

To better understand the usage model of digital bulletin boards, Taylor et al. [TCM09] and Alt et al. [AME+11] looked at paper-based noticeboards in communities and shops to extract design guidelines for digital noticeboards. Digital noticeboards should contain initial content and never be empty to ensure that users know what to do with it. In addition, they also propose to offer more than one interaction technique to give users the possibility to choose the technique they prefer.

Looking at the content of public displays, they often show static advertisement, news or weather information. Müller et al. [MWE+09] investigated the effect of display blindness, which occurs because users expect that a public display shows uninteresting content. To overcome display blindness, Huang et al. [HKB08] present design recommendations for displays people really look at. They found that the positioning, the size, the location and the content of the display are important to consider. Thinking more apart from static content, Davies et al. [DLJS12] describe the vision of open display networks beyond advertising which use content from many sources. Alt. et al [AMS12] share this vision and are convinced that open public displays create a benefit which also reduces display blindness.

Memarovic et al. [MCL+13] investigated the design space for accessing and uploading content to public displays. They distinguish between two types of content: ’tethered’ to a public display or 'free to roam'. 'Tethered' content is viewed and uploaded directly at the display, opposed to content that is 'free to roam' and can be viewed and uploaded without being physically present in front of a display. Their work supports application developers in deciding how to allow accessing and uploading of content.

As discussed, much research on public displays was done evaluating different interaction techniques, understanding the usage model of noticeboards, and understanding when people look at the content of public displays. On the basis and in the context of this work, a multitude of applications for public displays were developed.

2.2 App Stores

App stores are already widely used for selling mobile phone applications and are well-known by mobile phone owners and companies. The first app stores for mobile phones were launched in 2008 with the Apple App Store[1] and Google Play[2]. They started with a small number of applications, but today offer over 800.000 applications[3]. But not only mobile phone companies use an app store model to sell applications; e.g., Renault started to sell applications for cars with their R-Link Store[4]. Clinch et. al [CDKS12] propose to adopt a similar concept for public displays.

First of all, Clinch et al. [CDFE11] developed the e-Channel system to distribute content for public displays such as images and videos with a drop box approach. The channels are created by content providers. Display owners can subscribe to channels via a web interface to show the content of the channel on their public displays. Davies et al. [DFCS10] address the problems and challenges of the e-Channel system. The e-Channel system is not suitable for distributing interactive applications, which leads to the conclusion that an app store for public displays is needed to simplify the development and distribution of applications.

To get ideas on how to construct an app store for public displays, Clinch et al. [CDKS12] conducted focus groups with application developers for public displays. Furthermore, they presented an exemplary API for submitting and requesting applications. They also discussed the influences of different stakeholders: content providers, display owners, space owners, users and viewers. A detailed multi-stakeholder study was also done by Valkama et al. [VO11] for an application deployment in Oulu.

As experienced with the mobile phone industry, app stores can revolutionize the usage of a device. Building on the previous work presented, we were able to build an app store for public displays considering the existence of different stakeholders during the development.

[1] Apple App Store, http://www.apple.com/iphone-5s/app-store/ (accessed November 26, 2013)

[2] Google Play, https://play.google.com/store/apps (accessed November 26, 2013)

[3] Graph showing number of applications in Google Play, http://www.appbrain.com/stats/number-of-android-apps (accessed November 26, 2013)

[4] Renault R-Link Store, https://ie.rlinkstore.com/RLink (accessed November 26, 2013)

2.3 Existing Applications for Public Displays

Applications showing dynamic content with different purposes have been developed for public displays. Langheinrich et al. [LMEA11] investigated autopoiesic content and based on this, Memarovic et al. [MEL11] developed FunSquare, an application that shows fun facts by connecting information from the environment of a display with facts of a database. This is an approach to show self-generative content on public displays. Other applications allow the personalization of the content shown on the display, but most applications rely on social networks or user-generated content which is either directly created at the display or can be uploaded.

2.3.1 Applications Showing Personalized Content

Applications which show dynamic content do not conclusively have to show user-generated content. Personalizing content is also an approach to show content related to the user in front of the display. In the following, we discuss two applications offering personalization.

Rusell et al. [RDS02] describe an application called BlueBoard that allows persons to show their personal content on a public display by wearing a specific badge.

José et al. [JPSM13] developed Instant Places and the Pins and Posters application. Users can select a pin they like and checking in at a certain place will show content belonging to their pin. A moderation strategy is not needed for this application because the content associated with the pins is assumed to come from trusted sources. For each location, pins associated with appropriate content for this specific location can be picked. In contrast, posters are images that can be uploaded by users. A pre-moderation strategy is used to avoid inappropriate content. Space owners have to approve the posters that they want to show on their displays.

No content moderation seems to be needed for personalized content coming from trusted sources in contrast to using a pre-moderation strategy for posters created by users.

2.3.2 Applications Using Twitter

Hazlewood et al. [RHM08] developed Twitterspace, an application that shows Tweets on a public display, but offers no interaction.

Munson et al. [MRR11] developed a public display application that shows Twitter messages sent to a specific user and a Thank You Board that allows people to post thank you messages. 196 and 251 posts were received on the displays in a period of five to six months. They mostly had textual posts not encouraging the posting of photos.

Both applications are limited in the settings of their deployment which seems to be semi-public. It is also not clear if a particular moderation strategy was used.

2.3.3 Applications Using Flickr

McCarthy et al. [MCH08] developed a C3C system which shows Flickr images. Users can input a Flickr search query to select the images shown on the display. The application allows users to report content as inappropriate which will be removed from the application and can then be checked by a moderator on a web page.

CityWall [PKS+08], [PSJ+07] also fetches images from Flickr which are tagged with a selected keyword. Users can not change this keyword, but interact with the public display via touch to move, grab, rotate, or scale images.

While McCarthy et al. used a reactive moderation strategy in their C3C system, it is again not clear if any moderation strategy was applied to the CityWall content. If users did not know that the images were fetched from Flickr with a selected keyword, posting inappropriate content deliberately was not possible at all.

2.3.4 Applications Showing User-Generated Content Created at the Display

Memarovic et al. [MEM+13] developed the Moment Machine, an application that shows images taken through the display. The application also supports posting those images to Facebook, but does not fetch images from Facebook. During the trial of the application, people were invited to participate from present researchers. An expansion of the Moment Machine is the Moment Gallery [MEM+13] which additionally allows to show Instagram photos on the public display.

Brignull et al. [BR03] developed the Optionizer. On a laptop, besides the wall display showing the application, people could type in views and comments which were then displayed. During their deployment, they observed a honey pot effect. This effect describes the behavior that people were interested in the display when others interacted with it. They point out that at the beginning, it is very important to encourage people to interact with an application.

For both applications, it is not entirely clear if a certain moderation strategy was used, but researchers seemed to be present all time during the deployment. Thus, they were able to control the content, interfere and talk to people.

2.3.5 Applications Allowing to Upload User-Generated Content

Taylor et al. [TCF+07], [TC09] developed the Wray Photo Display which is installed in an urban village. Users can upload images and videos using either Bluetooth or a web interface and can then browse the uploaded content on the display. All content items are manually checked and approved before showing them on the display to avoid negative effects for the relationship between the university and the village. Building on the Wray Photo Display, Taylor et al. also developed the WrayDisplay [TC11] which allows users to additionally post news, which were also manually pre-moderated before showing them.

Churchill et al. [CND$^+$03a], [CND03b] developed the Plasma Poster Network, an application which shows content from two distinct sources: content can be explicitly posted by authenticated users or is picked from selected web pages. They created a minimal content moderation policy, but the system was deployed at their research group and not open to other users, so they had a common sense about appropriateness of content.

Two applications with a more specific usage were developed by Brignull et al. and Hosio et al. The application Dynamo [BIF$^+$04] allows users to share, exchange and show digital media on an interactive public display. Brignull et al. also conducted a study with a small group of students where two researchers were present all the time. Hosio et al. [HKK$^+$12] developed Ubinion which consists of large public displays that are used to collect personalized feedback from young people also allowing interaction with Facebook and Twitter. It is not clear if any content moderation strategies were used.

Four applications were developed to serve as digital bulletin boards. First of all, Greenberg et al. [GR01] developed the notification collage, a digital bulletin board where users can posts notes including text messages, images and videos.

WebWall [FV02], which was developed at the University of Linz by Ferscha et al., also allows users to send classified digital media, such as text, images, and videos to a display via several input channels.

MacColl et al. [MR08] developed IWALL, a digital community noticeboard where people can post content via SMS. As content moderation strategy, they used a self-moderation strategy as for physical noticeboards.

Alt et al. [AKB$^+$11] developed the application Digifieds, which allows users to post classified advertisements. With their mobile phones, users are able to put content on the display or take content away from the display. Digifieds uses a reactive moderation strategy where users can mark posts that are inappropriate. The posts are then banned from the display right away and later manually checked by moderators who decide if the content has to be removed permanently.

More applications showing user-generated content were also developed in the content of CSCW for small groups. An example is SharedNotes [GBL99] which allows people to exchange notes between personal devices and a public display in a meeting area. Moderation is normally not necessary for such a fixed setting which includes a small group of people that always use the display together.

As discussed, a variety of applications that allow users to upload user-generated content exist. Focusing on the moderation strategy, one of the applications used pre-moderation and two used a reactive moderation strategy. For all other six applications, it is not clear if a moderation strategy was applied and again settings of studies normally included a small number of users, because none of the applications focused on examining the topic of content moderation on public displays.

2.4 Content Moderation

Discussing related work about public display applications showed that content moderation is a topic which is not very often addressed in previous work regarding public displays. Nevertheless content moderation is also done in the real world and is highly addressed for web communities and forums. In the following, we discuss related work about content moderation in different fields.

2.4.1 Content Moderation in the Real World

Alt et al. [AME$^+$11] investigated paper-based notice areas (PNAs) and their implications for public display networks by observing PNAs in 29 different locations over a period of four weeks. In addition, the authors also conducted interviews with the persons responsible for the PNAs. In the interviews, Alt et al. found three different approaches for doing content management and moderation on PNAs. The most used approach was to remove outdated content by giving each content item an expiration date. Other persons just did a complete cleanup and regularly removed all posts from the PNAs. The third type of PNA only had curated content which had to be approved by a specific instance before publishing it.

2.4.2 Content Moderation in the Digital World

In the digital world, content moderation is well-known from web communities and forums. Surprisingly, to the best of the authors' knowledge, there is only little work about summarizing different moderation strategies or presenting a general concept for content moderation. Most previous work focuses on a specific moderation strategy. We will discuss some of these strategies in the following sections.

2.4.2.1 Classification for Content Moderation

In a blog post from 2003[5], the author distinguishes between user-level and post-level moderation strategies. A user-level moderation strategy rates and bans users completely, in contrast to a post-level moderation system that moderates each content item independent from the user. The author describes four strategies for post-level moderation in web communities: pre-moderation, post-moderation, reactive moderation, and distributed moderation.

[5] Four types of moderation, http://www.everythinginmoderation.org/2003/10/on_four_types_of_moderation.shtml (accessed November 26, 2013)

Pre-Moderation When pre-moderating user-generated content, every content item is checked before showing it to other users. It is the most secure moderation strategy for operators of a platform showing user-generated content but also the most time-consuming one and therefore the moderation strategy with the highest costs. Additionally, the delay time between a user submitting a content item and other users viewing the content item can be high.

Post-Moderation Post-moderation of user-generated content allows users to get an instant feedback when posting a content item. The content item is directly shown to other users, but is manually checked by a moderator afterwards. Thus, the strategy is also time-consuming, because still all posts have to be checked. The delay time between a user submitting a content item and other users viewing the content item is minimal. However, inappropriate content is shown to all users.

Reactive Moderation A reactive moderation strategy allows users to complain about content items of other users by informing a moderator. The strategy is less time-consuming, because moderators only have to check those content items reported by users, but the control about the content is even weaker than with post-moderation. The operators have to trust the users to report inappropriate content to be able to remove it.

Distributed Moderation A distributed moderation strategy can have different forms. A self-moderation approach where the community of users decides what is appropriate is one option. Other options include specific rating systems that allow users to rate content items. The control of content with such a distributed moderation strategy is weaker than with pre-moderation or post-moderation. The operators also have to trust the users to rate other content items reasonably.

We think that the discussed moderation strategies do not only apply for online communities and web forums, but for all situations where user-generated content is involved. Therefore, those strategies could also be applicable for public displays.

2.4.2.2 Online Commmunities

In this section, we discuss two particular moderation strategies of online communities with different purposes to show how those moderation strategies are used in practice.

Wright [Wri09] discussed content moderation in government online forums and outlines the necessity of moderation by focusing on related work. He also highlights that moderation can be contra-productive, e.g., if moderators do not behave fair. Next, he presents roles for moderators such as greeter, conversation stimulator, conflict resolver, problem solver, censor, or cleaner which includes different tasks associated with a role. Further on, he proposes two moderation models: silent moderation and interactive moderation. Silent moderation includes the fact that users get no feedback for their posts while interactive moderation includes the users.

A special and rather sophisticated distributed moderation strategy is used on the page slashdot.org [LR04]. Slashdot provides technical news, in particular, about open source software. Users can read stories and add comments, which could also be posted anonymously. Comments are rated for their worth by all users of the page. Users can also earn reputation which will help them to start their comments with a higher score. Moderators can then use a limited number of points to rate comments up or down and a meta-moderation system is used to control if the ratings were fair.

In addition to describing specialized moderation strategies, algorithms for automatic content moderation were developed. E.g., Lou et al. [LCL09] present several web moderation strategies. System moderation which is done by official moderators is outlined to be very labor-intensive. Social moderation includes users reporting content which will then be inspected by official moderators. Normally, such systems only allow the down-rating of posts which can also encounter innocent users. To decrease labor intensity, they propose an automation of social moderation and present an approach that analyzes relations between users to detect attacks against innocent users.

Wise et al. [WHT06] conducted an experiment simulating an online community with and without moderation. They show that the users have a greater intent to participate in a moderated online community than in an unmoderated online community.

Nevertheless, Gurzick et al. [GL09] determine that the effect of moderation strategies is not investigated very well. They did an experiment in an online community and changed the behavior of the moderators by instructing them differently. With the change of the behavior of the moderators, also the activity of the community changed and the number of posts increased or decreased, but they were not able to give a concrete conclusion about the impacts.

As discussed, moderation strategies are widely used in online communities and forums, but there is still a lack of classifications and understanding the full impacts of such a moderation strategy, which also depends on the users and purpose of an online community.

2.4.2.3 Public Displays

There is only little work about content moderation on public displays. Elhart [Elh12] describes the control and selection of content on a network of public displays. He stipulates four main factors that influence the control and selection of content: owner interests, content availability, viewer preferences, and the situation of the display.

Elhart et al. [EMLR13] also highlight disadvantages of pre-moderation strategies and present two post-moderation strategies which are content monitoring and removing through a web page or content moderation directly at the displays by using the RFID cards of staff and students.

Schroeter et al. [Sch12] developed Discussion in Space using a pre-moderation strategy. Moderators accept or decline new posts in a sort of online chat room. The average delay time of a post was 22 seconds. The authors also report that moderation was appreciated by the users. During the deployment, the moderators declined 26 % of the posts because they were offensive or inappropriate.

2.5 Summary and Discussion

As shown in section 2.3, previous work mainly focused on interaction techniques and community building. The applications are often deployed in small settings with a fixed number of users where no content moderation is required, because it is common sense what content will be appropriate. Studies of user-generated content on public displays were also very often restricted, e.g., researchers were present all the time or only researchers used the application. If applications use a moderation strategy, they often use pre-moderation or a reactive moderation strategy, but none of the authors discussed the impacts of the chosen moderation strategy for related stakeholders and the application itself. We want to build a basis for the future development of public display applications which helps developers choose a suitable moderation strategy for their application with considering the impacts on all related stakeholders.

3 Stakeholder Analysis for Public Display App Stores

As mentioned in the introduction, five different groups of stakeholders of an app store for public displays exist: display owners, space owners, content providers, users and viewers. In terms of user-generated content, the term “content providers” may be mistakable; we will therefore call them “application developers” to make clear that they provide the application running on the public display and do not generate the content.

In the following, we will discuss the interests of the groups of stakeholders and also identify potential factors and topics which may lead to conflicts between them.

3.1 Interests of Stakeholders

The interests of the related stakeholders for public display app stores strongly vary and conflicts are likely to occur due to different interests.

Display owners and space owners are mainly interested in advertising the location of the display or their products e.g. in a café, bar or shopping center. They want to use the display to attract new customers and get monetary benefits from it, but at the same time they do not want to loose their reputation due to inappropriate or offensive content on their displays. Although space owners and display owners seem to have the same point of view, it is not guaranteed that they really agree with each other about the usage of the display and the approach to get a monetary benefit.

Application developers are interested in distributing their applications to a big community. It is a very labor-intensive work to discuss concerns of different display and space owners convincing all stakeholders related to a public display installation of the benefits of a particular application. If such a time-consuming process is necessary for every application and every public display installation, it is not possible to develop a general application which can be used for a multitude of displays. Thus, less applications can be developed and less users interact with a particular application. The vision should clearly be to develop an application which will be bought by a large number of display owners, therefore shown on a large number of public displays, and also frequently used. The same development was observed with applications for mobile phones before app stores existed.

Users and viewers of public display applications want to benefit from the content. The content has to add some value to their daily life when passing a display regularly. They already highly ignore public displays because they do not expect them to show interesting content which reduces the profits of public displays. A display is much more valuable for users if they are able to interact with it, personalize it or even post user-generated content to it. The content is then no longer static, but dynamic, and adapts to their needs.

3.2 Conflicts between Stakeholders

The reasons for conflicts between stakeholders of a public display app store could be manifold. The level of conflicts may depend on three factors, which are highly related and also influence each other.

The first factor is the number of display and space owners which influences the number of different points of view. It is more likely to find a common point of view with less people. Conflicts between display owners and space owners cannot occur if they are combined in the same person.

The second factor is the number of users and viewers. Public displays are not always situated in public space, but also in semi-public spaces like office environments or classrooms. Such displays are normally bound to a fixed number of users which are also personally known and a common sense about how to use the public displays can be assumed.

The third factor is the location of the display. A prominent or delicate location, as for example a public display located next to a kindergarten, will increase the number of considerations and discussions between stakeholders to avoid negative consequences for their reputations.

In the following, we look at three potential topics which can cause conflicts between different stakeholder groups: the purpose and placement of a public display, the content shown on a public display, and trust between stakeholders. Further conflicts can emerge if the roles of the stakeholders are not well-defined.

3.2.1 Purpose and Placement

All stakeholders have to agree about the purpose of the display. If there is no common purpose, a display may remain switched off or simply be ignored.

Display Owners vs. Space Owners Both groups of stakeholders, display owners and space owners, want to benefit from a display installation which is only possible if they agree on a common purpose. Conflicts around the purpose of the display can occur when display owners and space owners do not share the same interests., e.g., if they want to promote different products. In addition, space owners may not always agree with display owners about the placement of the display.

Display Owners vs. Application Developers Display owners are highly interested in a monetary benefit of running a public display which has to be considered by application developers when designing their applications. Application developers are more interested in designing an application for a multitude of displays. If the developer does not allow customization of an application, it may fail to convince display owners to show the application.

Display and Space Owners vs. Viewers Conflicts can also arise when viewers do not understand or tolerate the placement or purpose of a display, which could be for example situated on their way to work.

3.2.2 Content

Conflicts about the content on public displays are most likely to occur. Controversial subjects are the personalization of content, the scheduling of available applications, and the appropriateness of the content shown on the display.

Display Owners vs. Application Developers Display owners may want to schedule different applications on their displays and do not want to show one application all the time. In contrast, application developers want their application to be shown for a long period. Allowing the scheduling of applications by design is important to prevent this conflict.

Display and Space Owners vs. Users Users are highly interested in interactive content and personalization of content, but the content is often dictated by display and space owners. The problem arising with user-generated content is the possibility that inappropriate content is shown on the display.

Users vs. Viewers Users and viewers can have a completely different understanding of appropriate content. Applications which show user-generated content will most probably be used by younger people who own a Facebook or Twitter account and are used to social networks. This experience makes them more liberal regarding the content of a public display. In contrast, it is likely that viewers who are not involved and maybe do not own a social network account, will complain about content that they do not like. Often viewers also do not know the intent or context of users posting content, which makes it even more difficult for them to decide whether or not content is appropriate. In the end, that can also lead to conflicts between display owners and viewers or space owners and viewers, if viewers complain about the content of a public display regularly.

3.2.3 Trust

Another topic causing conflicts between groups of stakeholders is missing trust in each other.

Display Owners vs. Application Developers Display owners can get applications from developers and have to trust their information about the content of the application. In particular for applications showing user-generated content, application developers can not guarantee for the content shown and display and space owners have no possibility to look at the content or get an idea of the content before using the application.

Display Owners vs. Space Owners It is difficult to establish trust between display and space owners, because space owners may want to control the work of the display owners.

Display and Space Owners vs. Users Display and space owners are, in particular, focused on their reputation. They often do not trust users to provide appropriate content on a public display showing user-generated content.

3.3 Summary and Discussion

We presented interests of different stakeholders and potential conflicts between them. We named three factors which influence the level of the conflicts: the number of display and space owners, the number of users and viewers, and the location of a public display. We also presented three potential topics which can cause conflicts: the purpose and placement of a public display, the content, and the trust between stakeholders. In chapter 8, we present ideas how to prevent and solve these conflicts with the help of the PD App Store.

4 Implementing a Public Display App Store

The concept for a public display app store was developed at Lancaster University in the context of the PD-Net project. Some workflows and sketches were already designed and served as a starting point for the development of the PD App Store. The work was split on two students being responsible for either back or front end. This thesis will only describe the front end work which was done by the author. All back end and API work was done by another student and is described in the thesis “Building and Using an Application Store to Support Public Display Users” [Mik13].

The vision of the app store is to provide a platform for application developers and display owners. Application developers can add and manage their applications to distribute them to the display owners. Display owners cannot only buy applications to show them on their public displays, the app store also provides a platform to manage all displays and the playlists for the displays which allow scheduling of applications.

In the following, we first of all present an overview on the architecture of the app store and the navigation structure. Next, we describe the front end of the app store providing screen shots of the important pages.

4.1 Architecture Overview

The architecture diagram in Figure 4.1 shows the back end of the app store. The back end was developed with Python[1] and the Django framework[2]. The Django models contain Python classes which specify data models and APIs. A MySQL database is used to store the instances of classes. Django has an integrated database API which manages the communication with the database automatically. With a direct mapping to each model, the REST API was created to interact with the back end of the app store. Third party tools and public display applications can use the REST API to communicate with the app store.

Based on the back end, we built a web front end that accesses the REST API and servers as user interface for the app store. The code of the front end is also located in the Django framework, because we use the templating language of Django. Besides using the Django templating language, we used HTML, CSS, and JavaScript to develop the front end code.

[1] Official Python Website, http://www.python.org/ (accessed November 26, 2013)
[2] Official Django Website, https://www.djangoproject.com/ (accessed November 26, 2013)

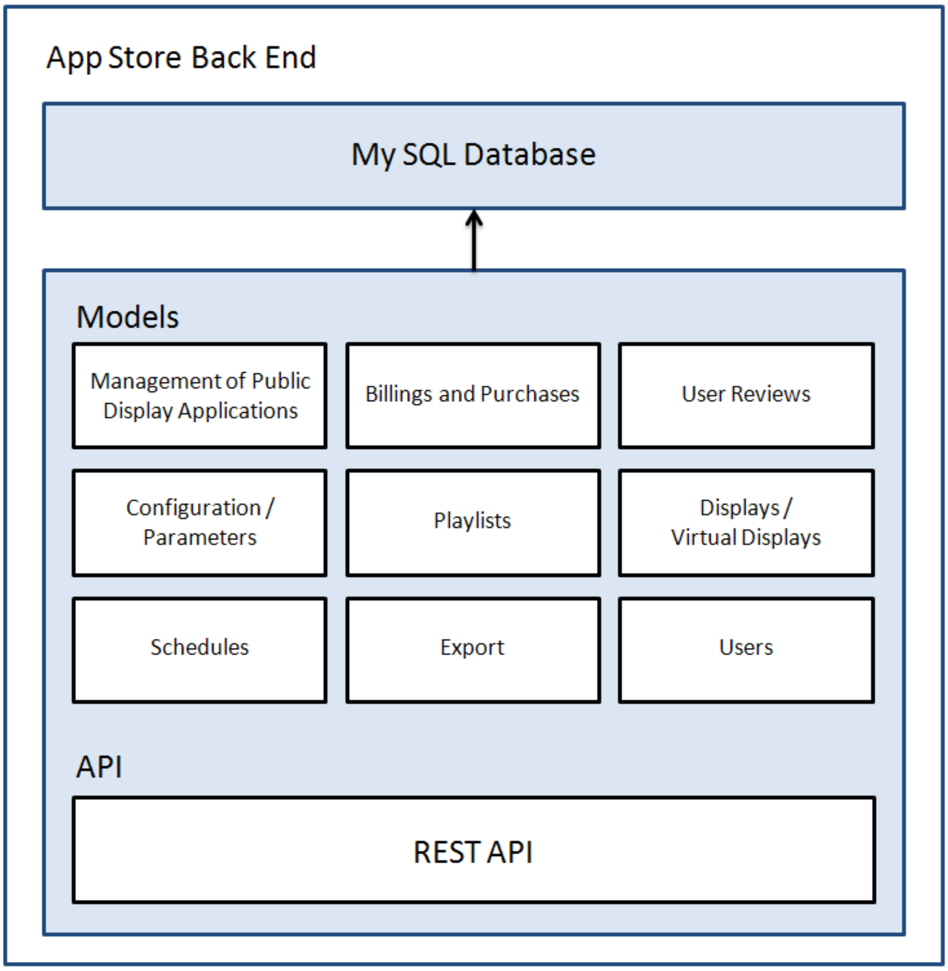


Figure 4.1: Architecture of the back end of the PD App Store including Django models.

Additionally, we used the JavaScript template library EJS[3], the JavaScript library JQuery[4], and the JQuery plug-in Raty[5]. As front end framework, we used Bootstrap[6] in combination with Font Awesome[7].

Display owners and application developers can access the front end of the app store with a web browser to create a PD App Store account or log in with an existing Google or Facebook account (see section 4.3.3). Application developers can use the PD App Store to create and manage public display applications. Display owners can purchase these applications for their displays. They can also review and configure purchased applications. Additionally, display

[3] EJS website, `http://embeddedjs.com/` (accessed November 26, 2013)
[4] Official JQuery website, `http://jquery.com/` (accessed November 26, 2013)
[5] Raty on GitHub, `https://github.com/wbotelhos/raty` (accessed November 26, 2013)
[6] Official Bootstrap website, `http://getbootstrap.com/` (accessed November 26, 2013)
[7] Official Font Awesome website, `http://fontawesome.io/` (accessed November 26, 2013)

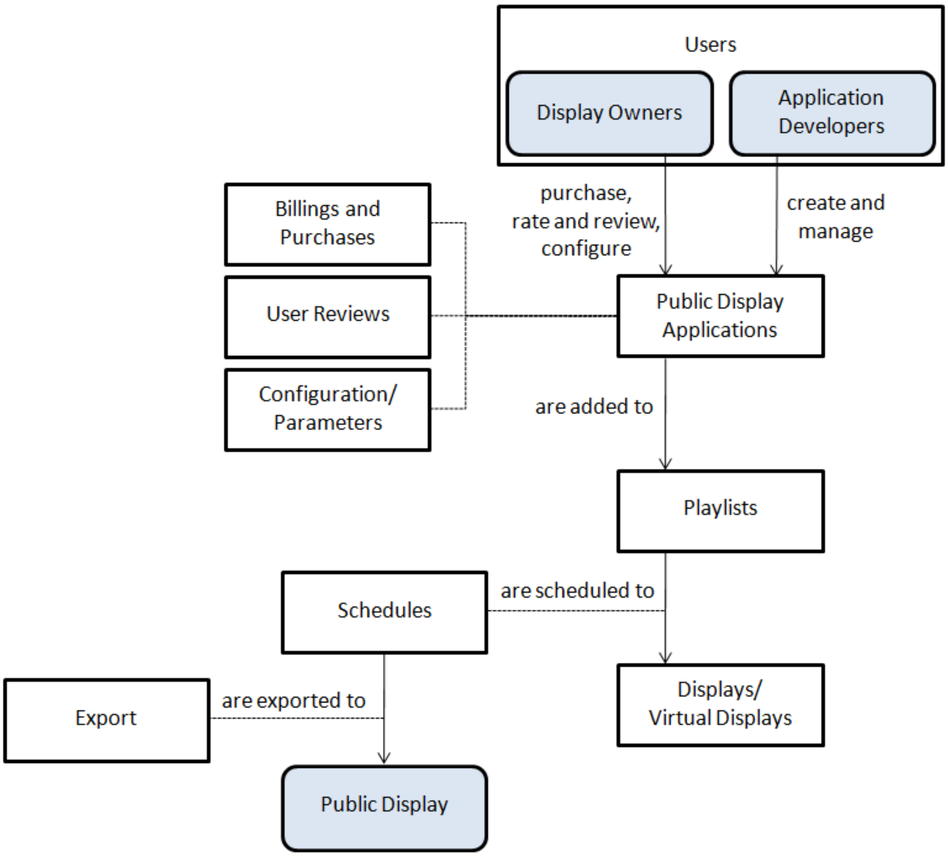


Figure 4.2: Context of Django models in the interaction flow of the PD App Store. Display owners can purchase, review and configure applications, add applications to playlists, schedule playlists to displays, and export the schedules for their public displays. Application developers can create and manage applications.

owners can add applications to playlists which can then be scheduled to displays. The app store therefore does not only serve as a distribution medium for the applications, but also allows display owners to manage their displays. Scheduled playlists can then be exported to be displayed on the public display. An overview on the described interaction flow with the associated Django Models is shown in Figure 4.3.

4.2 Front End

As described in section 4.1, the front end was developed with HTML, JavaScript and EJS. In the following, we first of all present the navigation hierarchy of the PD App Store to give and overview on the navigation items and pages developed. Next, we describe the most important pages and provide screenshots.

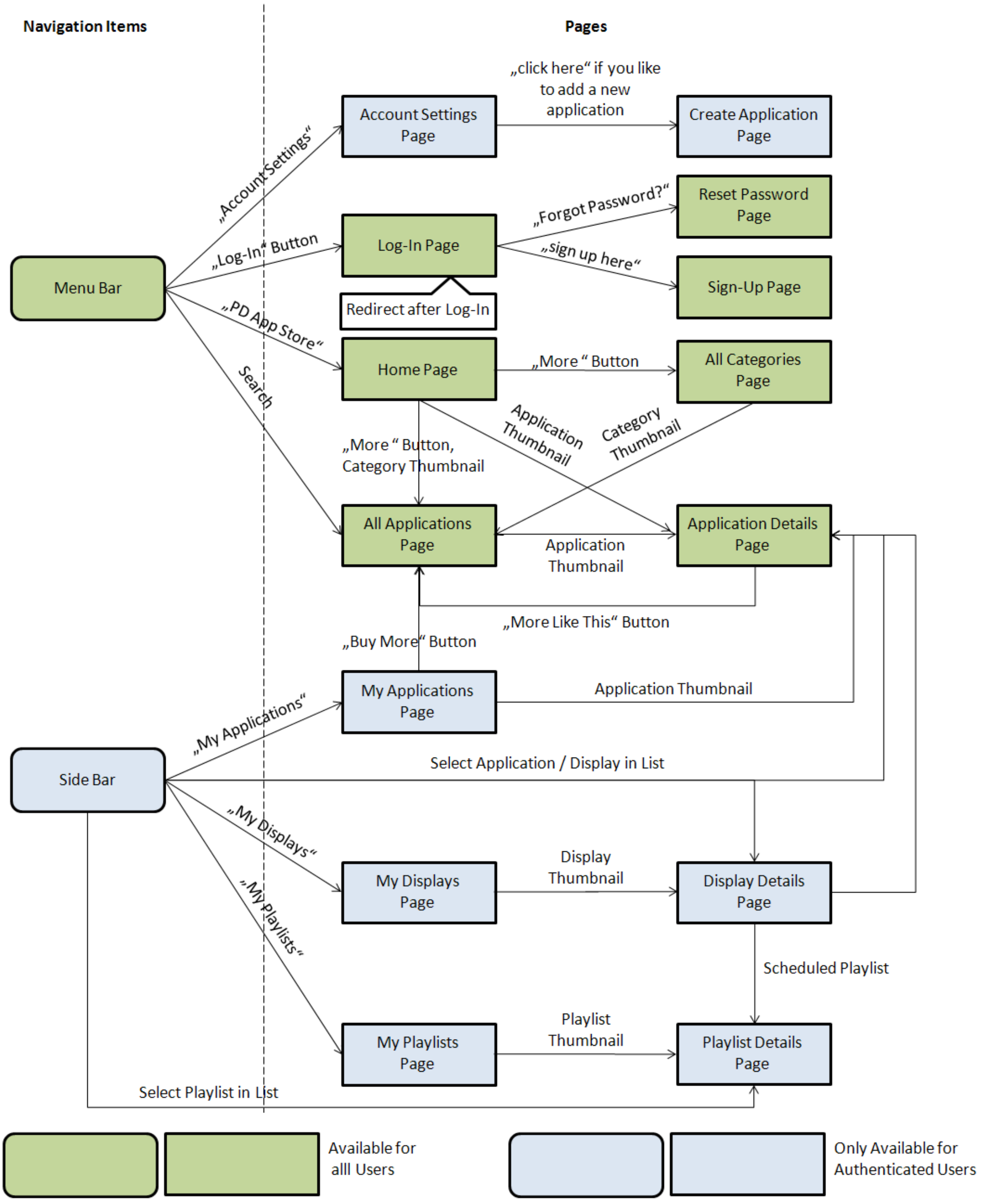


Figure 4.3: Navigation hierarchy of the PD App Store with all navigation items and pages. Arrows show the connection between pages.

4.3 Navigation Hierarchy

The PD App Store provides two navigation items, a menu bar (see section 4.3.1.1) and a side bar (see section 4.3.1.3). The menu bar allows unauthenticated users to access all general pages which include the login page, the home page and the pages with all applications. The side bar can only be accessed by authenticated users and allows to navigate to the personal applications, playlists, and details. An overview on the navigation hierarchy is provided in Figure 4.3. It shows the direct links from the navigation items to the pages and from one page to another page. The diagram shows that the all applications page and the application details page are essential pages of the app store. The reset password page and the sign-up page are not discussed in the following, because they are irrelevant for the functionality of the PD App Store. All other pages are discussed in the following sections.

4.3.1 Basic Page Layout

The basic page layout consists of five components (see Figure 4.4): the menu bar and the breadcrumbs navigation at the top of the page, a side bar at the left part of the page, a status bar at the bottom of the page, and the main content filling the rest of space.

4.3.1.1 Menu Bar

The menu bar provides shortcuts for navigating through the PD App Store. The brand name of the app store (PD App Store) is shown at the left and a click on it brings the user back to the home page of the PD App Store (see section 4.3.2). Next to the brand name, the users find a search box where they can enter a term to search in names, descriptions and reviews of all

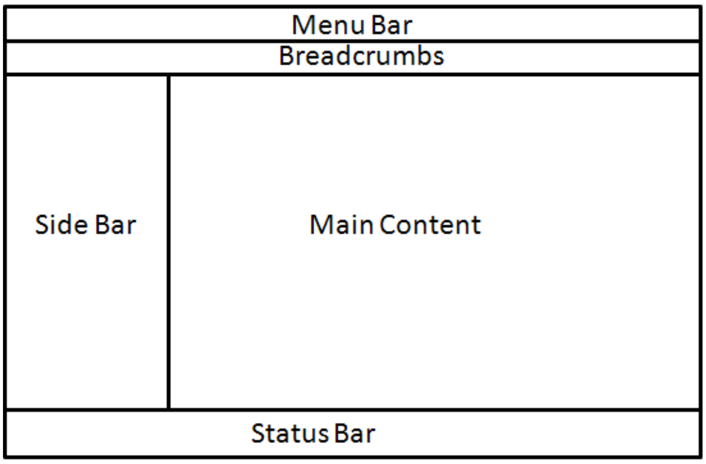


Figure 4.4: The basic page layout consists of a menu bar and breadcrumbs at the top, a side bar for further navigation at the left and a status bar at the bottom.

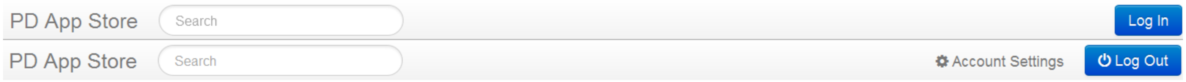


Figure 4.5: At the top, the menu bar when not authenticated as a user. At the bottom, the menu bar when authenticated as a user of the app store.

applications. The listeners for the search functionality are implemented in the basic JavaScript file. The right part of the menu bar depends on whether or not the user is authenticated (see Figure 4.5). If the user is not authenticated, a login button is shown which directs the user to the login page. The login button is hidden on the login page to avoid confusing the users. When the user is authenticated, a log-out button is shown instead of a login button. Additionally, authenticated users can reach their account settings (see section 4.3.4) by clicking on the corresponding menu item.

4.3.1.2 Breadcrumbs Navigation

The breadcrumbs navigation is a known pattern in web user interface design. It should help users to be aware of their location when navigating through a page with hierarchical content. We used the breadcrumbs in a very common way by just adding textual breadcrumbs below the menu bar, because we only have the side bar as main navigation.

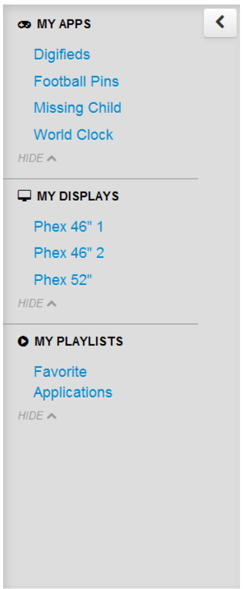
4.3.1.3 Side Bar

The side bar (see Figure 4.6) is the main navigation of the app store. It is only shown if a user is authenticated because it contains all purchased applications, displays and playlists associated with the user. The user is therefore able to navigate to one of the overview pages or to navigate directly to a particular application, display or playlist by expanding the menu items in the side bar. The side bar itself can be hidden at any time if the user, e.g., needs more space for the main content but can also be expanded again.

The side bar is initialized in the basic JavaScript file by calling the REST API to load the purchased applications, displays and playlists of the authenticated user.

4.3.1.4 Status Bar

The status bar contains the PD-Net logo to associate the app store with the project.



MY APPS

Digifieds
Football Pins
Missing Child
World Clock

MY DISPLAYS

Phex 46" 1
Phex 46" 2
Phex 52"

MY PLAYLISTS

Favorite Applications

Figure 4.6: The side bar is the main navigation item where the authenticated users can get to applications, displays and playlists. The particular applications, displays and playlists can be hidden and also the side bar itself can be hidden by clicking on the button at the top right corner.

4.3.2 Home Page

The starting point of the app store is the home page which contains four different boxes as main content (see Figure 4.7). At the top of the main content is a carousel which contains rectangular icons of applications picked by the editors. The carousel switches between these icons regularly, but the user is also able to switch the icons by clicking on the provided arrows. Below the carousel, the four recently added applications and the four applications which are hot right now are shown as application thumbnails in two boxes. The more buttons direct the user to the overview page of all applications (see section 4.3.7). The box at the bottom contains six category thumbnails that directly link to the overview page of all applications filtered by the picked category. The more button links to the overview page of all categories.

All information shown on the page are loaded via the REST API for applications. Three different actions are used for editors' pick applications, recently added applications, and applications which are hot right now. All information of the applications are inserted in an EJS template for the application thumbnail. Additionally, the information about the categories are also loaded via the REST API for applications. Up to six category thumbnails are shown by filling in the information into the EJS template for category thumbnails which are then shown in the corresponding box.

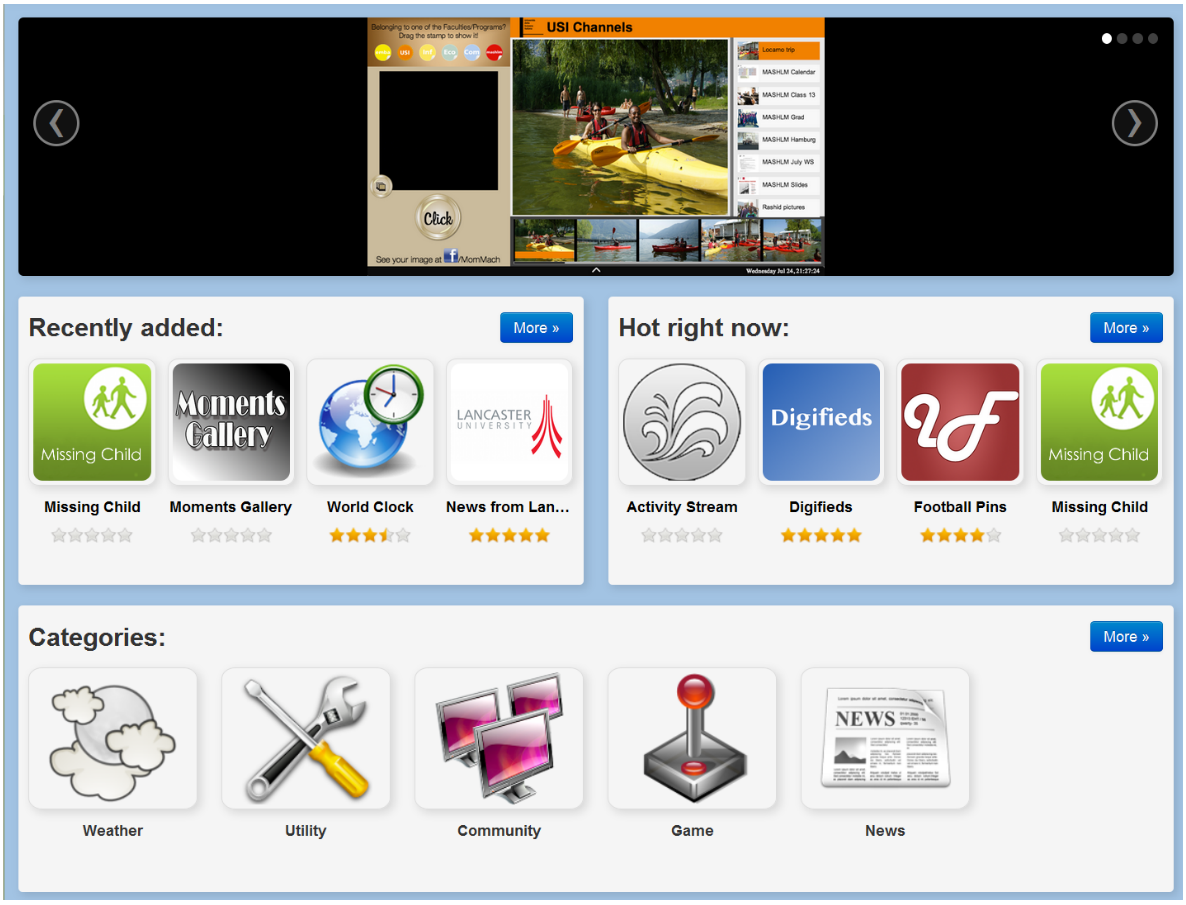


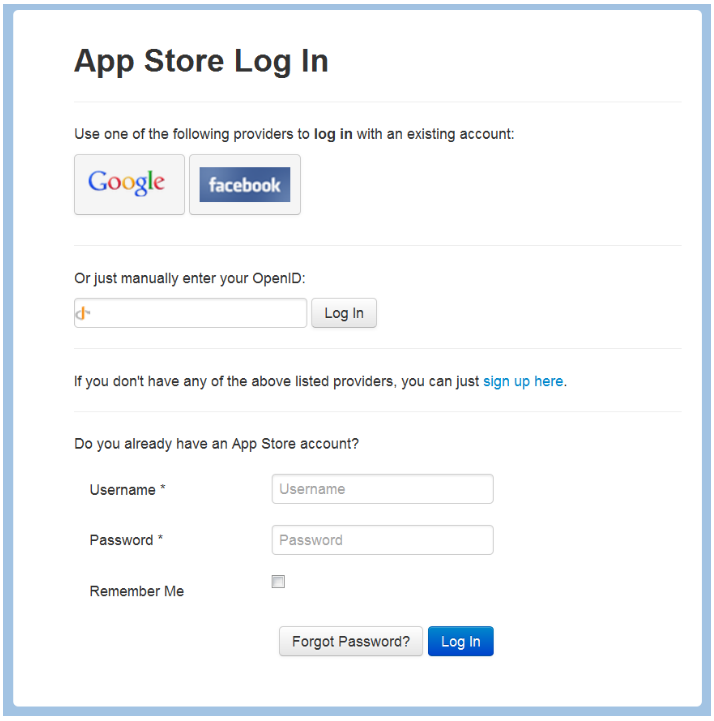
Figure 4.7: The home page of the app store shows a carousel with applications picked from the editors, the most recently added applications, the applications which are hot right now, and up to six categories.

4.3.3 Login Screen

The login screen (see Figure 4.8) of the app store provides four different possibilities to log in to the app store. The users can either use a third-party account (Google or Facebook), an OpenID, or an account specifically created for the PD App Store. If the user does not already own a account, he or she can decide to sign up. An e-mail address, a username and a password are needed for the sign-up process.

4.3.4 Account Settings

The account settings page of the app store (see Figure 4.9) allows the users to see their username and the associated e-mail address. On this page, users can also connect their account to third party accounts. If users want to register as application developers, they can specify their development company and afterwards add their applications through this page. The



App Store Log In

Use one of the following providers to **log in** with an existing account:

Or just manually enter your OpenID:

If you don't have any of the above listed providers, you can just sign up here.

Do you already have an App Store account?

Username * Username

Password * Password

Remember Me □

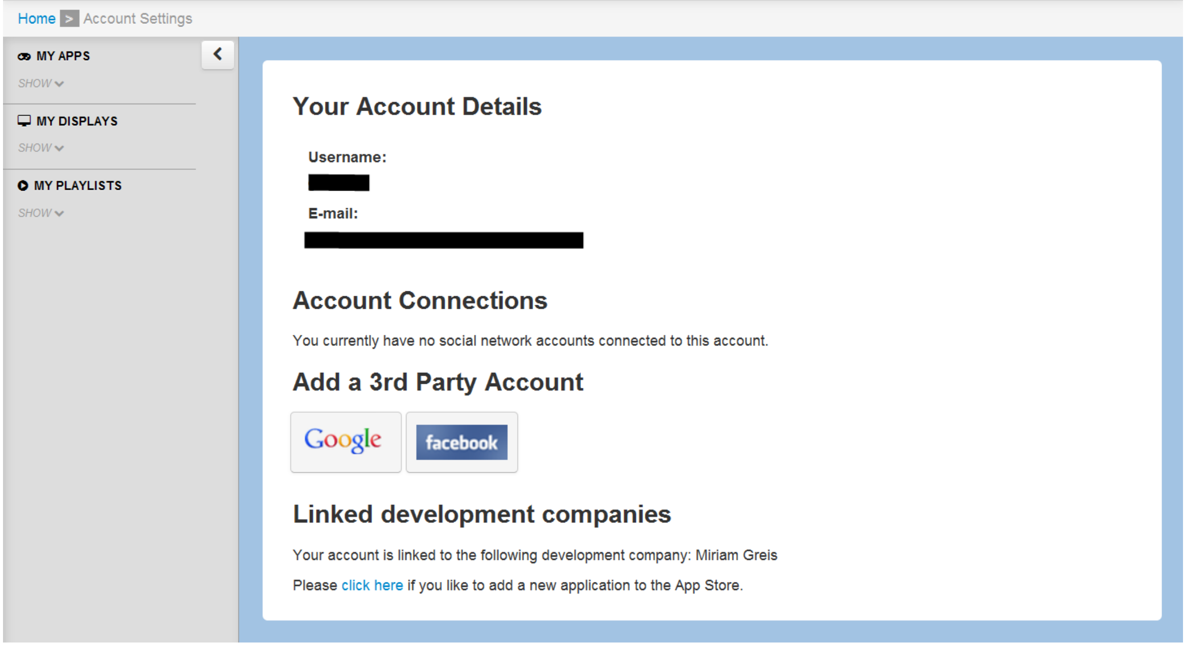
Figure 4.8: The login screen of the PD App Store offers four different authentication mechanisms. Users can log in with an existing Google or Facebook account, an OpenID, or a PD App Store account.

functionality for application developers was not further developed in the first version of the PD App Store, because the support of functionality for the display owners had priority. In a second version of the PD App Store, the functionality for the application developers should be improved.

The REST API is used to submit the sign-up form as developer and to load the development companies a user is linked with.

4.3.5 Categories

The all categories page (see Figure 4.10) gives an overview on all categories available for the classification of applications in the app store. Categories are shown in a grid with four category thumbnails in a row. The information for the category thumbnails are loaded via the REST API and filled in the EJS template for category thumbnails. Currently, five categories are used to classify applications: weather, utility, community, game, and news. A click on the icon of a particular category directs the user to the all application page with an applied filter of the selected category (see section 4.3.7).



Your Account Details

Username:
[redacted]

E-mail:
[redacted]

Account Connections

You currently have no social network accounts connected to this account.

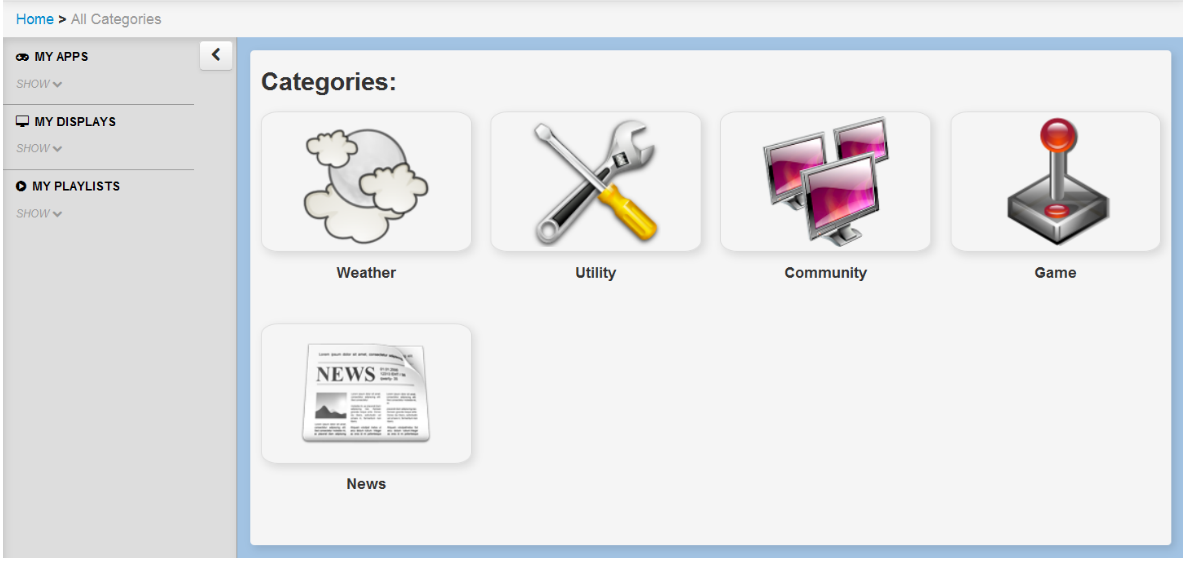
Add a 3rd Party Account

Linked development companies

Your account is linked to the following development company: Miriam Greis

Please click here if you like to add a new application to the App Store.

Figure 4.9: The account settings page allows the user to connect with a third party account and to sign up as an application developer. If already signed up, the user can add applications through the page.



Categories:

Weather

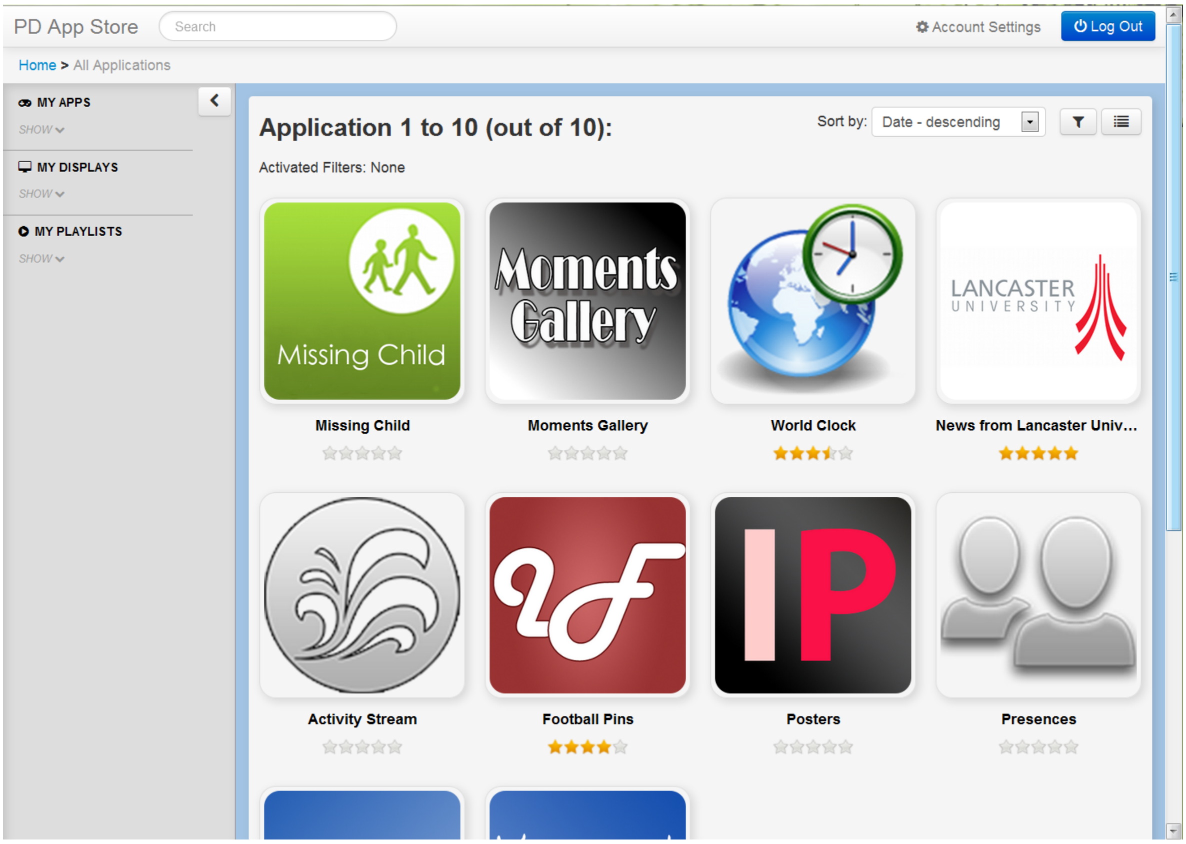
Utility

Community

Game

News

Figure 4.10: The all categories page shows all available application categories. When creating an application, the application developer has to choose one of the categories to classify the application.



Home > All Applications

Application 1 to 10 (out of 10):

Sort by: Date - descending

Activated Filters: None

Figure 4.11: The all applications page provides and overview on all applications. Twenty applications are shown as application thumbnails. A pagination feature allows to navigate through all applications. Ordering and filtering can also be applied.

4.3.6 Applications

Application developers can create applications in the PD App Store. Display owners can then look at all applications on the all applications page which also supports ordering and filtering of applications. In contrast to the all applications page, the my application page only shows the purchased application of the authenticated user. New applications can be purchased on their application details page which shows a description, screenshots and the reviews of the application. After purchasing an application, the users are also able to rate and configure the application.

In the following, we present screenshots and describe the implementation of all pages related to applications.

4.3.7 All Applications

The all applications page provides an overview on all applications available for purchase in the app store. 20 applications are shown on one page in a 4 × 5 grid of application thumbnails. An application thumbnail consists of a rectangular icon of the application, the name of the application and five stars that show the current average rating of the application. A click on the icon or the name of the application directs the user to the details page of the selected application (see section 4.3.7.3). A pagination feature allows the user to move forwards and backwards between different pages all containing 20 applications. The headline in the top left above the application thumbnails always shows the current number of the applications and the total application count. At the top right corner of the main content sorting, filtering, and changing the view from icon to list view are offered to the user. The sorting mechanism can be selected in a drop down menu. Applications can be sorted by name, date or score in either ascending or descending order. Filtering is implemented for categories and development companies.

The applications are loaded via the REST API and an EJS file for either the list or the icon view of the application thumbnail is used to show them on the page.

4.3.7.1 Adding New Applications

Application developers can add new applications through the link on the account settings page (see section 4.3.4) when already registered as a developer. The link leads to a wizard with five steps that have to be completed to successfully create a new application in the PD App Store.

The first step of the wizard offers two application types: URL application and dropbox slideshow application. The application developer has to choose one of the types for the new

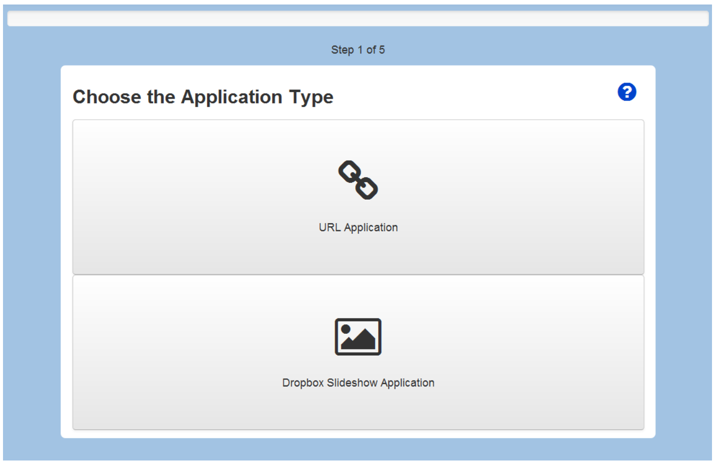


Figure 4.12: First step of the wizard to create a new application. The application developer can choose the type of the application. Types provided: URL application or dropbox slideshow application.

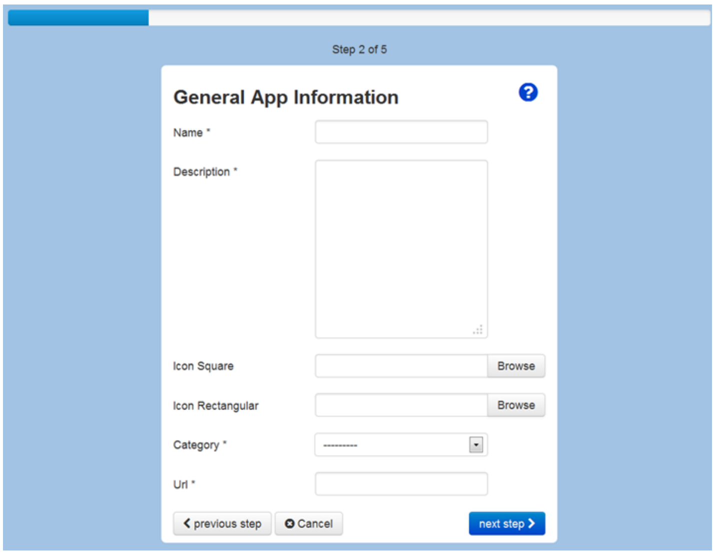


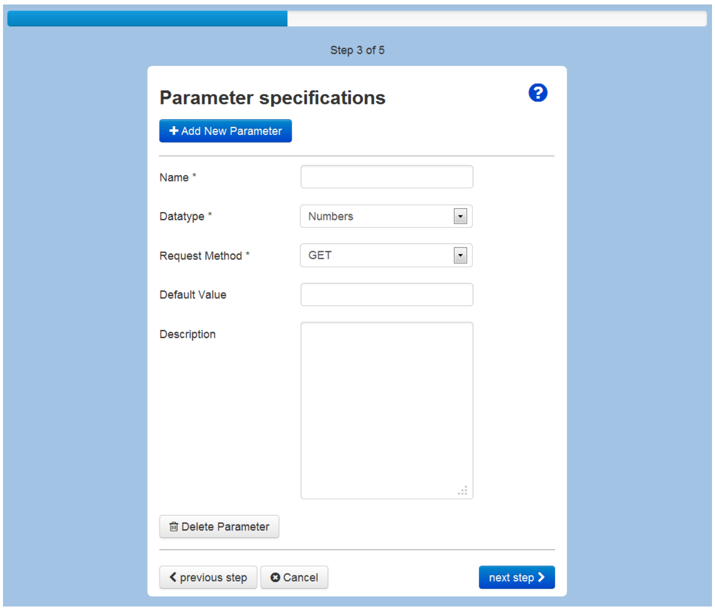
Figure 4.13: Second step of the wizard to create a new URL application. The application developer has to enter basic information about the application. Name, description, category, and URL are required fields. Optionally, icons for the application could be added.

application. The PD App Store currently only offers those two types. Adding a dropbox slideshow application includes adding a dropbox API key which has to be provided by the application developer. The developer can either add a new key, use the default key or reuse an already existing key. Further on, the developer has to specify information about the application, can add screenshots and submit the application as draft or for moderation. For the following steps, we focus on the creation of a URL application.

The second step of the wizard shows a form for general information about the application. The application developer has to enter a name and a description for the application. Additionally, one of the existing categories has to be chosen in the drop down menu. The developer also has to specify the URL pointing to the application. Optionally, the application developer can upload a square and a rectangular icon for the application.

With the navigation buttons at the bottom of the wizard, the developer can either return to the previous step of the wizard, cancel the creation of the application or proceed to the next step.

The third step of the wizard offers the application developer to add parameter specifications. A new parameter could be added by clicking on the button at the top and requires a unique name. The application developer also has to choose the data type and the request method for the parameter. Five data types are offered for the choice of the developer in a drop down menu: numbers, text, choices, multiple choices, and date. As request method GET, PUT, or POST can be selected. The application developer can optionally specify a default value for the parameter and write a description about the purpose of the parameter. The description could help display owners to understand the functionality of the parameter when configuring



Name *

Datatype * Numbers

Request Method * GET

Default Value

Description

Figure 4.14: Third step of the wizard to create a new URL application. The application developer can add parameter specifications. A parameter consists of a unique name, a data type, and a request method. Optionally, a default value and a description can be added.

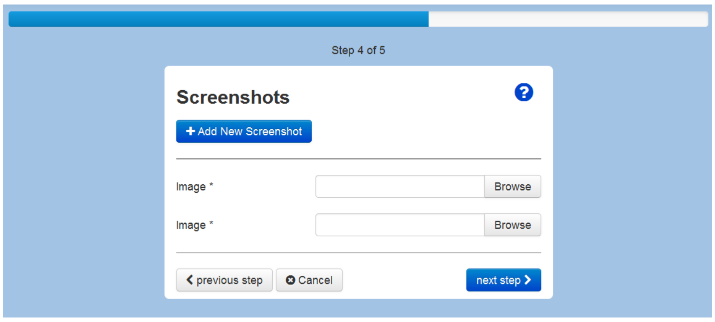


Image *

Image *

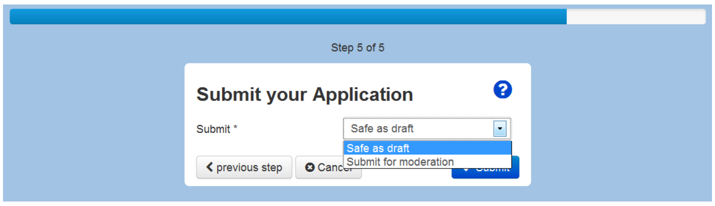
Figure 4.15: Fourth step of the wizard to create a new URL application. The application developer can add an arbitrary number of screenshots of the application.

an application. Application developers can specify an arbitrary number of parameters and add or delete them.

In the fourth step of the wizard for creating an URL application, the application developer can add an arbitrary number of screenshots of the application.

The fifth step of the wizard saves the created application either as a draft or submits it for moderation.

The app store does not support editing of submitted applications at the moment. The functionality is planned for a second version of the app store.

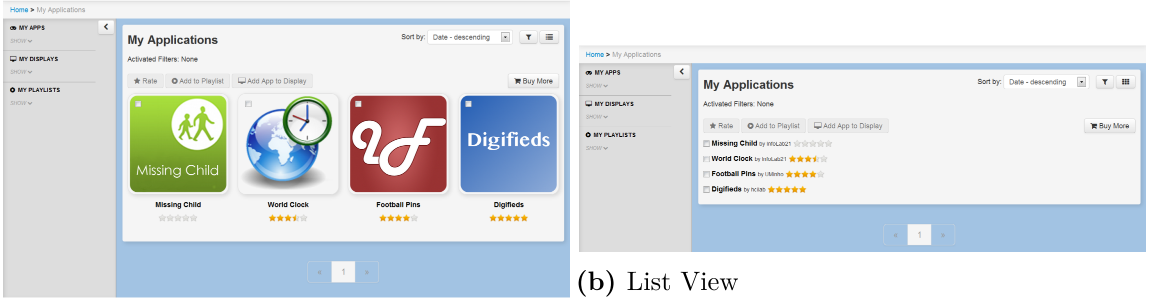


Step 5 of 5

Submit your Application

Submit *: Safe as draft

Figure 4.16: Fifth step of the wizard to create a new URL application. The application developer can either save the application as draft or submit it for moderation.



(b) List View

(a) Icon View

Figure 4.17: The my applications page gives an overview on the purchased applications of the authenticated users. Applications can be rated and added to playlists or displays.

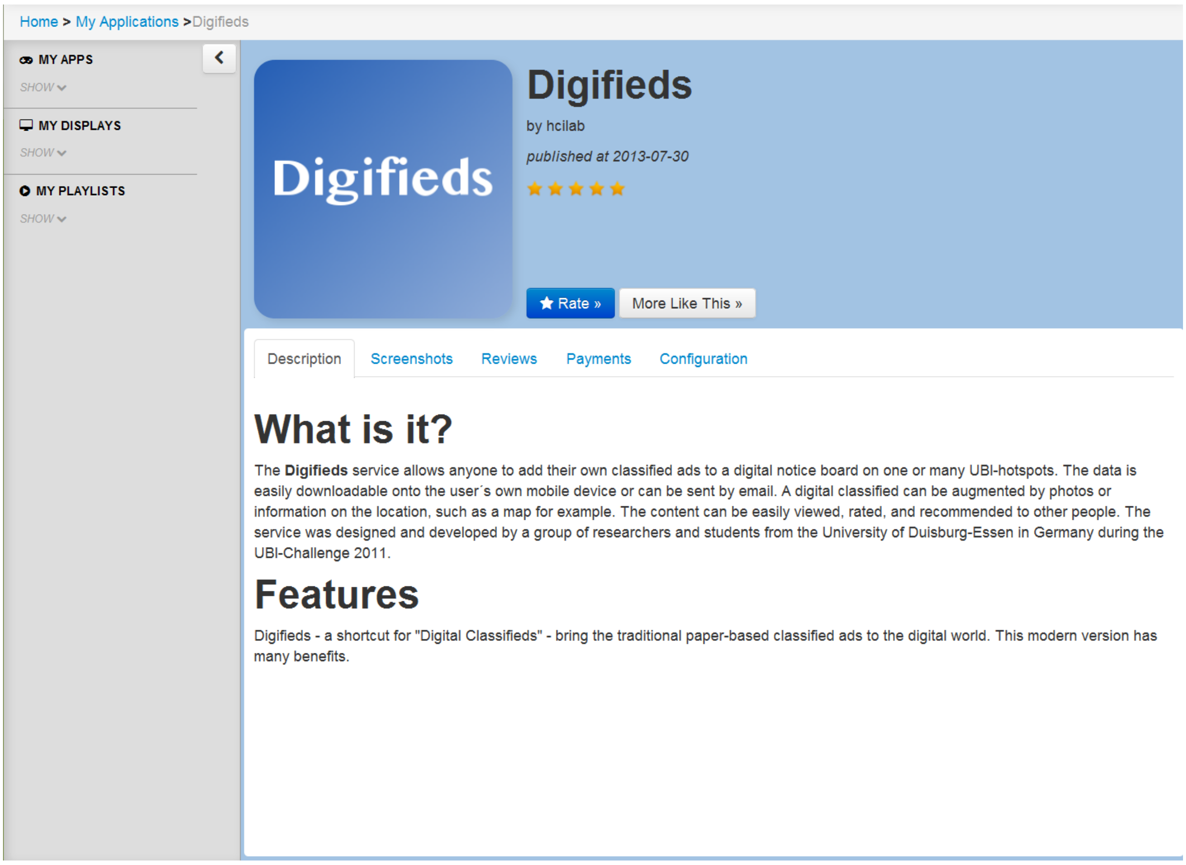
4.3.7.2 My Applications

The my applications page uses the same concept and skeletal structure as the all applications page (see section 4.3.7), but the page contains additional parts which are not necessary to be provided on the all applications page. The applications shown on the my applications page were purchased by the authenticated user. The user can therefore rate this applications and add them to playlists and displays. To do this, the user has to tick the box in the left top corner of one or multiple application thumbnails. For rating the application, a dialog will open which allows the user to choose a number of stars for each application and additionally write an optional review comment. Adding applications to playlists or displays is also done by opening a dialog where the user can choose the playlist or display in a drop down menu. The user can also add applications to more than one playlist or display at the same time.

All applications and also the displays and playlists for the dialogs are loaded via the REST API. All inputs of the user are also saved via the REST API.

4.3.7.3 Application Details

The details page of an application (see Figure 4.18) provides general information at the top of the page: the square icon and the name of the application, the development company, the publication date and the average rating. In the tab panes below the general information, all



What is it?

The **Digifieds** service allows anyone to add their own classified ads to a digital notice board on one or many UBI-hotspots. The data is easily downloadable onto the user´s own mobile device or can be sent by email. A digital classified can be augmented by photos or information on the location, such as a map for example. The content can be easily viewed, rated, and recommended to other people. The service was designed and developed by a group of researchers and students from the University of Duisburg-Essen in Germany during the UBI-Challenge 2011.

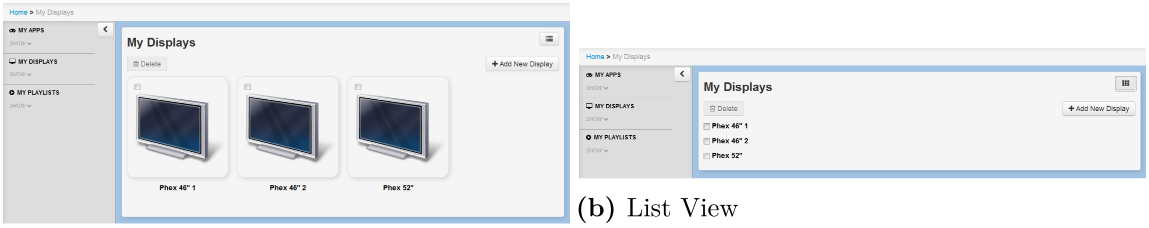
Features

Digifieds - a shortcut for "Digital Classifieds" - bring the traditional paper-based classified ads to the digital world. This modern version has many benefits.

Figure 4.18: The application details page shows all information about an application and the purchase of an application. Authenticated Users who purchased the application can rate and configure the application on this page.

users can read the description of the application, look at associated screenshots, and look at the reviews of other users. The users also have the possibility to search for applications which are similar to the application they are looking at. Users, who are not authenticated, can choose to log in to buy the application, whereas authenticated users can directly buy it. Users who are authenticated and already purchased the application can either choose to rate the application, check their current payments or configure the application for their needs. The configuration of an application is done with the parameter specified from the application developer during the creation process of the application. The display owner is able to selector insert a value for each parameter.

All information about the application are loaded via the REST API. Different actions have to be used for reviews, screenshots, purchases, and the configuration.



(a) Icon View

(b) List View

Figure 4.19: The my displays page offers the authenticated users to manage their displays. They can create and delete displays.

4.4 Displays

Display owners can create and organize their displays in the app store. Therefore, they can create displays and attach hardware items to them. The URL to the exported configuration of a display can be shown for each display.

In the following, we present screenshots and describe the implementation of the my displays and the display details page.

4.4.0.4 My Displays

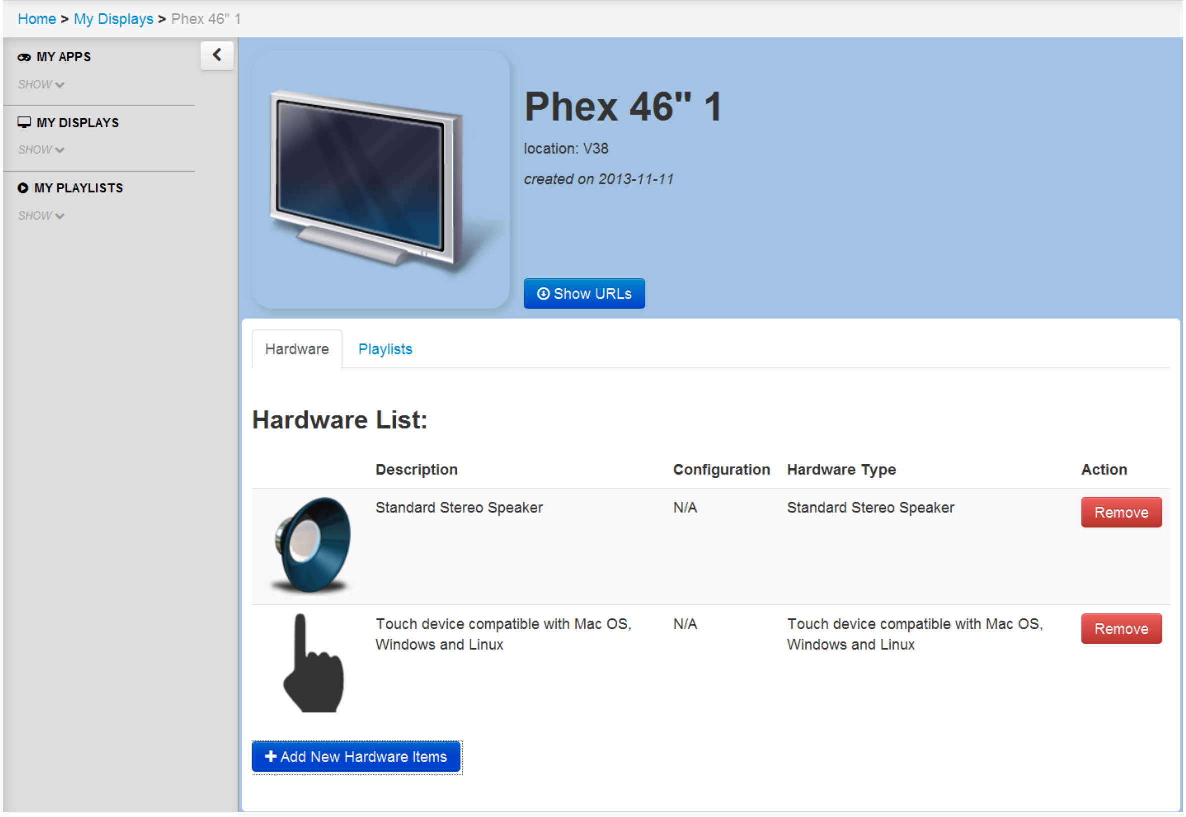
The my displays page (see Figure 4.19) gives an overview on the displays of the authenticated user. All displays are shown in a grid with four display thumbnails in a row. Each display thumbnail shows a default icon and the name of the display below the icon. A click on the icon or the name of the display directs the user to the display details page (see section 4.4.0.5). The page also allows to switch between icon and list view. A user can add new displays by inserting a name and a location for a new display. By ticking the box in the left corner of the display thumbnails, displays could be selected and deleted.

All displays are loaded via the REST API and added to the page with an EJS template for either the list or icon view of the display thumbnail. Creating and deleting displays is also done via the REST API.

4.4.0.5 Display Details

The display details page (see Figure 4.20) shows the default icon, the name, the location, and the creation date of the display. The user can also look at the URL necessary to export the display's playlists and applications. In the tab pane, hardware items such as speaker or touch devices can be added and removed from the display. Additionally, the user can look at all playlists scheduled to the display and remove them.

The associated JavaScript file loads the display information, the associated hardware items and the playlists via the REST API to fill in all information on the page.



Phex 46" 1

location: V38

created on 2013-11-11

Hardware List:

	Description	Configuration	Hardware Type	Action
	Standard Stereo Speaker	N/A	Standard Stereo Speaker	Remove
	Touch device compatible with Mac OS, Windows and Linux	N/A	Touch device compatible with Mac OS, Windows and Linux	Remove

Figure 4.20: The display details page shows all information about a display. The user can add or remove hardware items, show the URL to the XML export or look at the playlists scheduled to the display.

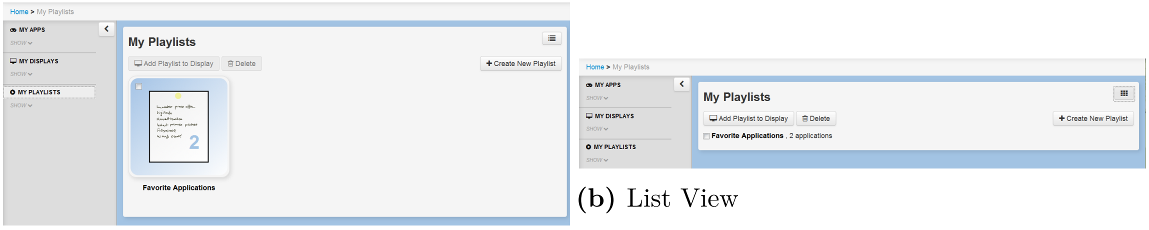
4.4.1 Playlists

The app store allows display owners to organize their applications in playlists. A display owner has to first create a playlist. Next, the display owner can add applications to the playlist and then add the playlist to displays for scheduling the applications in the playlist to the displays.

In the following, we present screenshots and describe the implementation of the my playlists page and the playlist details page.

4.4.1.1 My Playlists

The my playlists page (see Figure 4.21) gives an overview on the playlists of the authenticated user. All playlists are shown in a grid with four playlist thumbnails in a row. Each playlist thumbnail has a default icon showing the number of playlist elements in the bottom right corner and the name of the playlist below the icon. A click on the icon or the name of the



(b) List View

(a) Icon View

Figure 4.21: The my playlists page offers the authenticated user to manage playlists with applications. Playlists can be created, deleted and added to displays.

playlist directs the user to the playlist details page (see section 4.4.1.2). The page also allows to switch between icon and list view. Three specific actions are offered to the user on the page: the user can add a playlist to a display, delete a playlist or create a new playlist. To select playlists for the first two actions, the user can just tick the box in the top left corner of the playlist thumbnail. The dialog for adding playlists to a display allows to choose one or more displays. To create a new playlist, the user has to enter a name for the new playlist.

The associated JavaScript file loads all playlists via the REST API and creates playlist thumbnails with an EJS template for either the list or the icon view. All actions are carried out via the REST API.

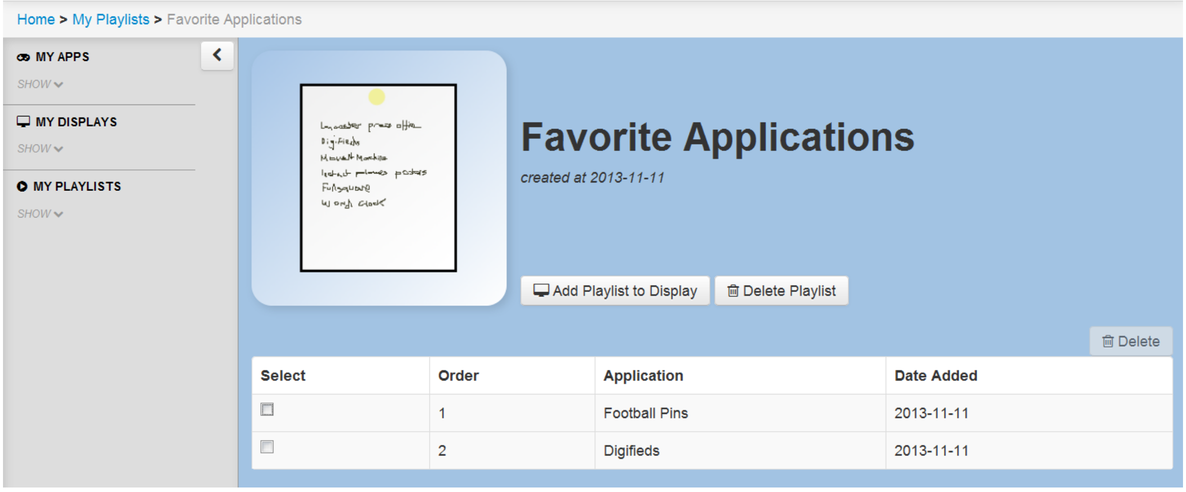
4.4.1.2 Playlist Details

The playlist details page (see Figure 4.22) shows the default icon, the name, and the creation date of the playlist. As on the my playlists page, the users can add the playlist to the display or delete it. The page also provides a table that includes all applications which are contained in the playlist. Applications can be selected in the table to delete them from the playlist. In the next version of the PD App Store, the user should also get the possibility to order the applications in a playlist.

The associated JavaScript file loads the playlist via the REST API and extracts and fills in all important information. The element of the playlist are also loaded via the REST API. Other functionality as adding playlists to displays or deleting a playlist use the code implemented for the my playlists page.

4.5 Summary and Discussion

In this chapter, we described the implementation of the front end of the PD App Store. Application Developers can create their applications to be sold in the app store. Display owners can purchase an application, rate and review the application and configure the application for their needs. Applications can then be added to playlists which can be scheduled to displays. The schedules can be exported in an XML format for the real display.



Home > My Playlists > Favorite Applications

Favorite Applications

created at 2013-11-11

Select	Order	Application	Date Added
□	1	Football Pins	2013-11-11
□	2	Digifieds	2013-11-11

Figure 4.22: The playlist details page shows all applications added to the playlist. The playlist can be added to a display or can be deleted. Particular applications can also be deleted from the playlist.

The web front end was developed with HTML, JavaScript, and EJS. The framework Bootstrap was used for the design of the front end. The front end provides to navigation items:a menu bar and a side bar. The side bar and all pages containing applications, playlists or displays of a particular user are only accessible if the user is authenticated. The user can either login with a third party account (Google or Facebook), an OpenID or a PD App Store account.

Currently, the PD App Store does mainly support display owners. More support for application developers should be added in the next version.

5 Online Survey about Public Displays

Little work has been done so far about the users' view of content moderation on public displays, but understanding the impacts of moderation strategies and the users' view can help display owners and developers to pick appropriate moderation strategies for their applications. We conducted an online survey about public displays to gain insight into the public display users' view and their expectations towards content moderation of public displays. Our aim was to identify a suitable moderation concept for public display applications, in particular, those displaying user-generated content. We asked survey participants for basic demographic data, as well as their prior experience with public displays. The survey itself consisted of questions and statements about the topics: public display content, authentication mechanisms, and moderation strategies including identification of potential content moderators.

In this chapter, we describe in detail the questionnaire, its preparation, and distribution. We also summarize the demographic data of our participants and describe the two groups that we created: inexperienced and experienced participants. Next, we present the survey results for the three subjects, subject by subject. Then we discuss the optional comments inserted by participants. Finally, we summarize and discuss all results and show the implications for future development of public display applications.

5.1 Questionnaire

A first version of the questionnaire was created and then reviewed in a meeting with three participants. Then the questionnaire was revised, improved, and tested by filling out the online survey. The final version of the questionnaire with all explanations and questions can be found in the appendix (see section A.1).

The questionnaire is four pages long. Since questions that appear later in the survey depend on answers to previous questions, only forward navigation is permitted. Participants must answer all mandatory questions on a given page before moving on to the next page. They are free to interrupt the survey and complete it at a later time. They may also choose to exit the survey, which deletes all of their given answers. Nearly all questions are mandatory. There are only two optional questions, which allow participants to enter free text comments.

Besides the optional questions with free text answers, all other questions build on given answers. Two of them offer a comment field for the participants to give the reasons for their choice. However, mostly the participants have to indicate their level of agreement with several statements on a five-level Likert scale. The Likert scale allows the participants to choose one of five levels of agreement with a specified statement in a range from "totally disagree" to "totally agree". Intermediate steps represent the levels "disagree", "neutral" and "agree". The levels are consecutively numbered from one ("totally disagree") to five ("totally agree").

The exact questions, the preparation and deployment of the questionnaire, and its distribution are described more detailed in the following sections.

5.1.1 Questions

The survey itself starts with an introduction and a short explanation. The explanation should help participants to get into the topic and fill in potential knowledge gaps. Hence, a public display is explained to be a (large) screen in public space, e.g., at universities, train stations, or airports often showing static content as advertisements, weather forecasts, or news. Further on, the example of a digital bulletin board is used to explain the term user-generated content.

Starting the survey with a click on the "Next" button, participants get to the first page of the questionnaire where they have to enter demographic data, which is needed for the analysis. Besides gender, age, and profession, participants are also asked if they already posted content on a public display or even operate, maintain or work with public displays. Comment fields allow participants to specify their answers. This information about their prior experience could be used to split up participants into two distinct groups, one group of inexperienced and one group of experienced participants (see section 5.2 for more details).

The second page of the questionnaire covers the topic of authentication. Participants have to indicate their level of agreement on the five-level Likert scale with multiple statements covering different authentication mechanisms. First of all, they are asked if they would post a message on a public display when there is no authentication process at all. The other statements specify three different authentication mechanisms: authentication with a verified mail account, a social network account or a personal ID, e.g., a student ID or ID card. These statements are further specialized distinguishing between two different contexts which are authenticating directly at the display or authenticating remotely, e.g., on a laptop or a mobile phone.

The third page of the questionnaire contains questions about potential public display content. Again, participants are asked to indicate on the five-level Likert scale if they agree with the statement that they expect a message to appear on the display instantly. Additionally they have to choose the maximal delay time that would be acceptable for them. Participants can choose between given periods, which are "1 second", "1 minute", "10 minutes", "1 hour", "24 hours" or "It wouldn't matter for me how long it takes until my message is displayed." Optionally, participants can specify content that they would post on a public display such as a digital bulletin board.

The last page of the questionnaire covers the topic of content moderation. Again, participants have to indicate their level of agreement with multiple statements on the five-level Likert scale. First of all, they have to rate if a moderation or control process for public display is necessary to avoid misuse. Furthermore, the participants have to rate statements about using different moderation strategies. A classification of moderation strategies in the web environment is presented in section 2.4.2.1. The following moderation strategies are presented to participants:

Pre-Moderation: All content should be manually checked and approved before showing it on the display.

Post-Moderation: All content should appear on the display right away and should be checked regularly afterwards.

Reactive Moderation: The participant himself or herself should be able to report inappropriate or offensive content.

Community-Based Moderation: A community should check the content (e.g., if five people mark a post as inappropriate, it will be deleted automatically).

Community Moderators: A group of moderators should check the content regularly.

White List: The content should be checked before showing it on the display. Regular posters will be added to a white list which allows them to post without pre-moderation.

The next statements that are presented to participants refer to the stakeholders who could possibly check or moderate the content on a public display. The following stakeholders are presented to participants:

Display Owners: The persons who own the displays.

Space Owners: The persons who owns the location of the displays.

Viewers: Everyone that can see and possibly interact with the display.

Users: All persons that posted content to the display.

Uninvolved Third Parties: Every person not involved in any of the other specified roles.

Participants are then again asked to choose a delay time which is acceptable for them knowing that a public display is moderated. The choices are exactly the same as in the second question on the third page of the questionnaire and also range from “1 second” to “It wouldn’t matter for me how long it takes until my message is displayed.”

The last statements that have to be rated by participants cover the informational aspect. Participants have to rate if they would like to be informed when their message was received and would like to know when their message will be displayed, if there is a moderation process going on, and who moderates the content.

At the end of the questionnaire, participants are able to enter any questions or comments about the topic or the survey itself.

5.1.2 Preparation and Deployment

The questionnaire was prepared with Lime Survey[1] and deployed on the Lime Survey Server of the HCI group. The survey was an open survey, which means that everyone clicking on the link to the questionnaire was able to participate. Furthermore the survey was anonymous, which was also communicated to participants before starting the survey.

We prepared two different versions of the questionnaire, an English and a German one, to make sure that participants are able to choose one of the two languages and therefore able to understand the questions and statements. Participants were able to switch the language at any time whilst reading the questions and editing their answers. For the analysis of the data all German answers were translated to English.

5.1.3 Distribution

The link to the questionnaire was distributed across social networks such as Facebook, the university mailing list and the PD-Net project mailing list. The PD-Net project mailing list was chosen to get more participants who already have experience with public displays whilst the main audience on Facebook was suspected to be inexperienced.

5.2 Participants

In total 114 participants, 64 males and 50 females, completed the online survey. The average age of the participants was 24.10 (SD: 4.54). The participants were mainly students (83 out of 114) with different subjects, one of the participants was a pupil, thirteen participants were academics (lecturers and PhD students), and the rest of the participants worked as employees in various professions, e.g., as teacher, doctor, mailman or project manager.

The question about the prior experience of participants with public displays showed that most people never posted content to a public display. One participant even stated that he never saw a public display. However, ten participants already posted content to a public display, e.g., messages on a Twitterwall, happy birthday wishes on a festival or photos on photo displays. Seven participants stated that they maintain, operate or work with public displays in research or for their bachelor thesis. We split the participants in two groups, one with 101 participants who never posted content to a display and one with thirteen already experienced participants that posted content to a public display or even work with displays because we expected to get different answers depending on the experience of participants. In addition, most of the experienced participants were included in the PD-Net project and are aware of the stakeholders which could potentially influence their answers.

[1] Survey application Lime Survey, http://www.limesurvey.org (accessed November 26, 2013)

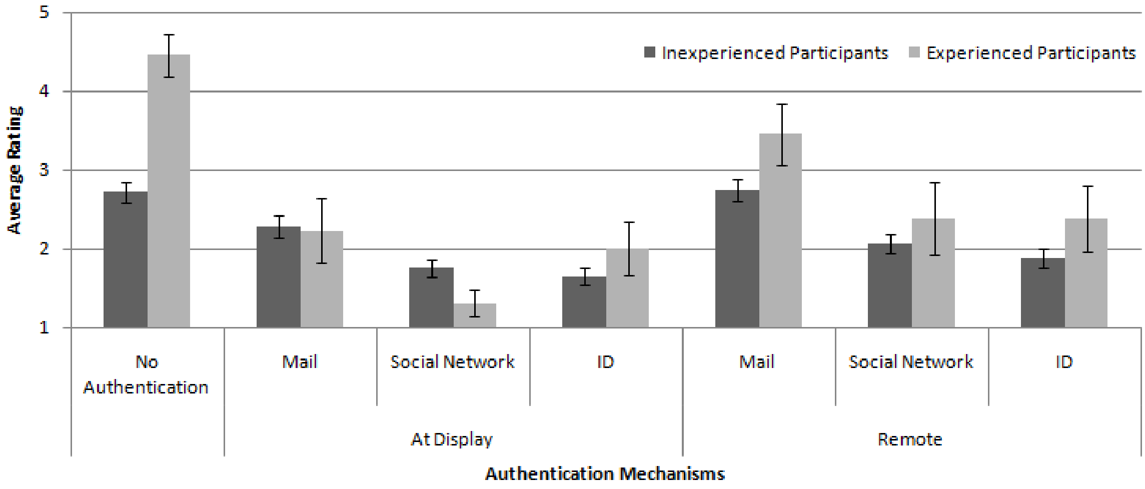


Figure 5.1: Average ratings on a five-level Likert scale from "totally disagree" (1) to "totally agree" (5) for the usage of different authentication mechanisms distinguishing between inexperienced and experienced participants. Error bars show the standard error.

5.3 Results

In the following, we present the results of the online survey for the topics authentication, content on public displays and content moderation with the help of diagrams and statistical evaluations.

5.3.1 Authentication

Suggesting six different authentication mechanisms (see section 5.1.1) and no authentication at all, none of the mechanisms got an average agreement over 3 on our five-level Likert scale (see Figure 5.1) when looking at the group of inexperienced participants. Only 56.44 % of the inexperienced participants agreed partly with at least one of the suggested authentication mechanisms, the other 43.56 % did not agree partly with at least one of the mechanisms. Nine participants even totally disagreed with all of the presented mechanisms.

In both scenarios, authenticating at the display and authenticating remotely, the average agreement of the inexperienced participants decreases from mail to social network and from social network to id. Nevertheless, the average agreement with an authentication mechanism is slightly higher when authenticating remotely.

The Friedman test shows a statistically significant difference in agreeing with different authentication methods for inexperienced participants, $\chi^2(6) = 119.160$, $p < 0.001$. Post-hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied on $p < 0.05$, resulting in a significance level of $p < 0.002$. For exact results see Table 5.1. No

	At Display			Remote		
	Mail	Social Network	ID	Mail	Social Network	ID
No Authentication	$p = 0.005$	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$p = 0.832$	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$
Mail	–	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$p = 0.140$	$p = 0.005$
Social Network	–	–	$p = 0.282$	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$p = 0.342$
ID	–	–	–	$\boldsymbol{p < 0.001}$	$p = 0.003$	$\boldsymbol{p = 0.001}$
Mail	–	–	–	–	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$
Social Network	–	–	–	–	–	$p = 0.125$

Table 5.1: Results of the post-hoc analysis with a Wilcoxon signed-rank test for different authentication mechanisms and an applied Bonferroni correction, resulting in a significance level of $p < 0.002$. Statistically significant values are highlighted in bold font.

authentication is significantly different from authenticating with social network or ID either at the display or remotely. No statistically significant difference can be discovered between no authentication and authentication via mail. Comparing the authentication on the display with the remote authentication, each authentication mechanism such as mail, social network account and personal ID is rated significantly higher with remote authentication than with authentication on the display.

The small number of experienced participants only allows us to show trends, but no statistically significant differences. In contrast to inexperienced participants, experienced participants highly agree with the statement that they would post a message if there is no authentication at all (average: 4.46, SD: 0.97). When authenticating at the display, experienced participants also prefer authenticating via mail. They would use social network accounts less than inexperienced participants and rate the authentication with a personal ID about 0.69 higher than the authentication with a social network account. For the remote authentication the average values of the experienced participants are also higher than the average values for authenticating directly at the display and also higher than the average values of the inexperienced participants. Authenticating remotely via mail is rated highest with 3.46 (SD: 1.39). The average values for authenticating remotely with a social network account or a personal ID are exactly the same.

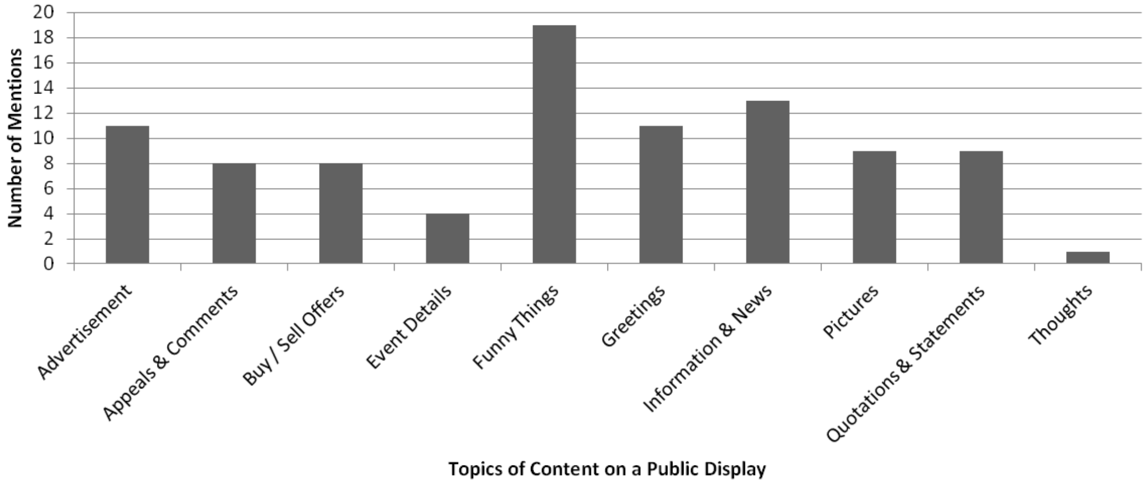


Figure 5.2: Topics of potential content mentioned from participants in the online survey.

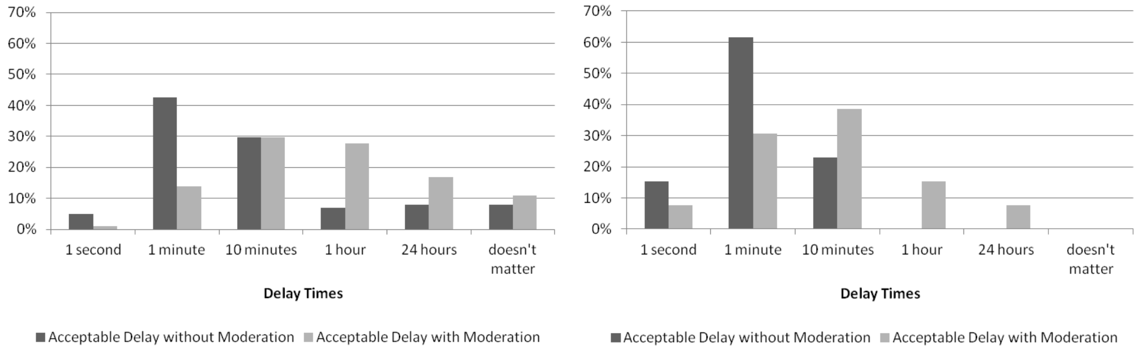
5.3.2 Public Display Content

Regarding public display content, we first analyzed topics of potential content. The topics were extracted from answers of all participants. Additionally, we examined the time expectations of participants towards content publication with and without moderation.

5.3.2.1 Topics of Potential Content

70 participants answered to the optional question about what content they would post to a public display, e.g., a digital bulletin board. Half of them included more than one type of content. Eight participants would not post any content to a public display at all and three participants do not know what they would post. The other participants regarded the question in a different way and, e.g., stated that the content would depend on already existing content or the location of the public display. Three participants would only post content that cannot be traced back to them and one participant even explicitly stated that he would try to misuse and exhaust the display for fun.

The other 53 answers can be assigned to ten topics (see Figure 5.2). The topics were created from the participants' answers by going through them and adding a new topic if none of the existing topics fitted for the current answer. Our categorization shows that 19 participants would post funny things on the display, e.g., jokes. 13 participants would post political or economical information or news, tips for cheap offers and restaurants, travel information, or information about room changes at the university. Eleven participants named advertisement such as advertisement for own projects or a youth group along with social and political advertisement. Eleven participants would post greetings to friends and colleagues, e.g., congratulations, love messages or even a proposal of marriage. Less than eleven participants



(a) Inexperienced Participants **(b)** Experienced Participants

Figure 5.3: Acceptable delay times for messages with and without moderation.

named pictures, quotations and statements, comments as motivational messages or feedback, offers, event details, and thoughts.

5.3.2.2 Time Expectations

In total, 81.58 % of the participants partly or totally agree with the statement that they would expect a message to appear instantly. The average rating is 3.89 (SD: 1.08).

Asking the participants for the delay time they would mostly accept, 47.52 % of the inexperienced participants and 76.92 % of the experienced participants would only accept delay times up to one minute. Asking the same question again later on specifying that there is a moderation process running only 14.85 % of the inexperienced participants and 38.46 % of the experienced participants would still mostly accept a delay up to one minute. In total, 57.89 % of the participants are therefore willing to wait longer knowing that there is an on-going moderation process. Fore more details, see Figure 5.3.

5.3.3 Content Moderation

The next part of the questionnaire covers the topic of content moderation. The first statement indicates that content moderation or a control process on public displays is necessary to avoid misuse. 80.20 % of the inexperienced participants choose a 4 or 5 on the Likert scale and therefore agree with the statement and see a necessity for content moderation. Surprisingly only 61.54 % of the experienced participants agree with the statement. However, with an overall average rating of 4.15 (SD: 1.10) most participants see a necessity for moderation.

5.3.3.1 Moderation Strategies

We also asked participants to state their agreement with different moderation strategies (see section 5.1.1). The group of inexperienced participants mostly disagrees with the concept

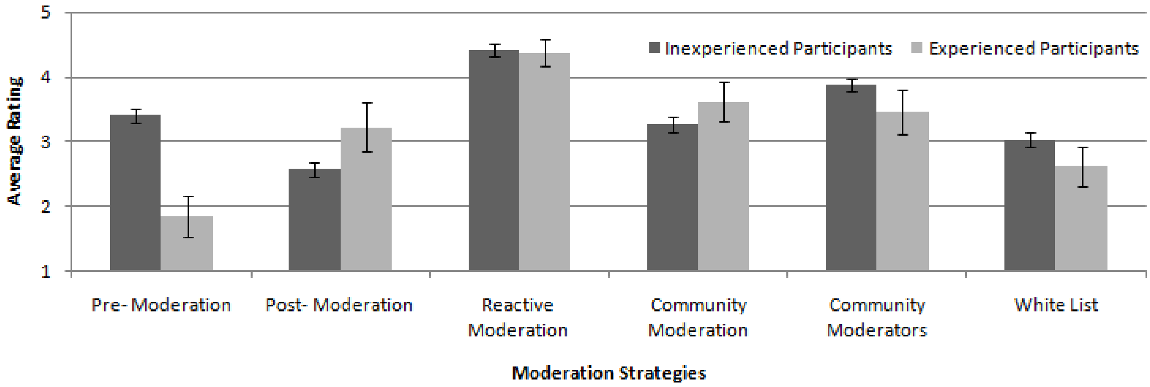


Figure 5.4: Average ratings on a five-level Likert scale from “totally disagree” (1) to “totally agree” (5) for the usage of different moderation strategies distinguishing between inexperienced and experienced participants. Error bars show the standard error.

	Post-Moderation	Reactive Moderation	Community Moderation	Moderators	White List
Pre-Moderation	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$p = 0.291$	$\boldsymbol{p < 0.001}$	$p = 0.019$
Post-Moderation	–	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$p = 0.003$
Reactive Moderation	–	–	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$	$\boldsymbol{p < 0.001}$
Community Moderation	–	–	–	$\boldsymbol{p < 0.001}$	$p = 0.151$
Moderators	–	–	–	–	$\boldsymbol{p < 0.001}$

Table 5.2: Results of the post-hoc analysis with a Wilcoxon signed-rank test for agreeing with the usage of different moderation strategies and an applied Bonferroni correction, resulting in a significance level of $p < 0.002$. Statistically significant values are highlighted in bold font.

of post-moderation having an average agreement of only 2.57 (SD: 1.09). In contrast, they mostly prefer the reactive moderation strategy with an average of 4.42 (SD: 0.96). All other moderation strategies receive an average agreement from 3.03 to 3.88 (see also Figure 5.4).

We performed a Friedman test which shows a statistically significant difference in agreeing with the usage of different moderation strategies for inexperienced participants, $\chi^2(5) = 141.287$, $p < 0.001$. Post-hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied on $p < 0.05$, resulting in a significance level of $p < 0.003$. For exact results see Table 5.2. The tests show that the average agreement with a reactive moderation strategy is significantly higher than the average agreement with every other moderation strategy, followed

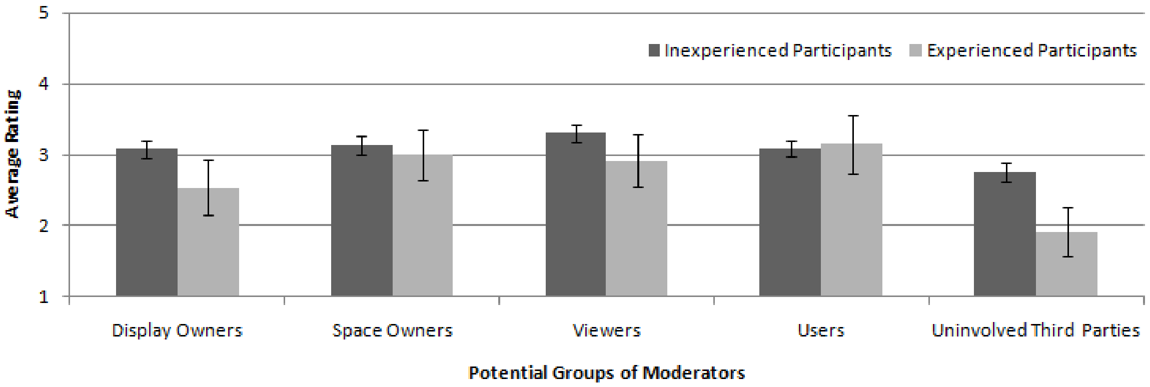


Figure 5.5: Average ratings on a five-level Likert scale from "totally disagree" (1) to "totally agree" (5) for the stakeholders as potential groups of moderators distinguishing between inexperienced and experienced participants. Error bars show the standard error.

by the community moderators strategy. No statistically significant difference can be found between pre-moderation, community moderation and the white list moderation strategy, but the average agreement with post-moderation is significantly lower than the average agreement with all other moderation strategies expect for the white list moderation strategy.

The group of experienced participants also mostly prefers the reactive moderation with an average of 4.38 (SD: 0.77) followed by the community-based strategy and the community moderators. Strikingly, the majority of the experienced participants disagrees with the pre-moderation strategy which is only rated with an average of 1.85 (SD: 1.14).

5.3.3.2 Groups of Moderators

Asking the group of inexperienced participants to indicate their level of agreement with statements about potential groups of moderators, no strong tendency in agreeing or disagreeing with any of the given groups can be discovered. All possible stakeholders get an average agreement between 2.75 and 3.31. Thereby the viewers are rated highest and the uninvolved parties are rated lowest. Figure 5.5 shows all average ratings.

The Friedman test nevertheless shows a statistically significant difference in agreeing with different groups of moderators $\chi^2(5) = 15.365$, $p = 0.004$. Post-hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied on $p < 0.05$, resulting in a significance level of $p < 0.005$. For exact results see Table 5.3. A statistically significant difference can only be discovered between uninvolved third parties and viewers. Uninvolved third parties are rated lower than viewers.

	Space Owners	Viewers	Users	Uninvolved Third Parties
Display Owners	$p = 0.688$	$p = 0.285$	$p = 0.906$	$p = 0.081$
Space Owners	–	$p = 0.314$	$p = 0.764$	$p = 0.078$
Viewers	–	–	$p = 0.018$	$\boldsymbol{p = 0.004}$
Users	–	–	–	$p = 0.041$

Table 5.3: Results of the post-hoc analysis with a Wilcoxon signed-rank test for agreeing with different groups of moderators and an applied Bonferroni correction, resulting in a significance level of $p < 0.005$. Statistically significant values are highlighted in bold font.

In the group of experienced participants the uninvolved third parties only score 1.92 on average (SD: 1.26). The display owners are only rated with 2.54 in average (SD: 1.39) and all other stakeholders range between an average of 2.92 to 3.15.

5.3.3.3 Correlations between Moderation Strategies and Groups of Moderators

Looking at the correlations between moderation strategies and groups of moderators by calculating Pearson's r, eight correlation are statistically significant. The findings indicate that moderation strategies and groups of moderators are coupled. Table 5.4 shows the correlation matrix with all highly significant correlations for the group of inexperienced participants.

The correlation matrix itself shows a weak positive correlation with the display owners and a strong positive correlation with uninvolved third parties for the pre-moderation strategy. The correlations for all other groups of moderators are not significant. Regarding post moderation there is no significant correlation with any of the groups of moderators at all because all correlation coefficients only show no or a negligible relationship. However, for reactive moderation the correlation coefficients show a statistically significant moderate positive correlation with the space owners and a weak positive correlation with the users. In contrast, the community-based moderation has a strong correlation with the viewers. The strongest correlation can be discovered between community moderators and display owners, but there is also a moderate positive correlation with the space owners. The last one of the presented moderation strategies, the white list moderation strategy, correlates moderately positive with the users. Surprisingly, there is no statistically significant negative correlation at all and all groups of moderators correlate at least with one moderation strategy.

The correlation coefficients of the experienced participants look very different to the ones of the inexperienced participants. However, a sample of thirteen participants is not very large and therefore only two significant correlations were detected. The method of community moderators has a very strong positive relationship with the space owners ($r = 0.71$, $p < 0.01$) and the method of community-based moderation has a strong negative relationship with the uninvolved third parties ($r = -0.61$, $p < 0.05$). All other correlation coefficients are not significant, but

	Display Owners	Space Owners	Viewers	Users	Uninvolved Third Parties
Pre-Moderation	**0.29****	0.07	−0.01	−0.11	**0.43***
Post-Moderation	−0.04	0.04	0.03	0.18	−0.15
Reactive Moderation	0.19	**0.33***	0.23	**0.27****	0.21
Community-based Moderation	−0.16	0.02	**0.45***	0.15	−0.09
Community Moderators	**0.51***	**0.35***	0.03	0.15	0.25
White List	0.17	0.06	0.16	**0.36***	0.03

Table 5.4: Pearson's r for correlations between moderation strategies and potential moderators for inexperienced participants. Statistically significant values are highlighted in bold font. $^{**}p < 0.01$, $^{*}p < 0.001$

may show a tendency which could be proved with a larger group of experienced participants. It is nevertheless remarkable that the display owners only get negligible correlation coefficients.

5.3.3.4 Information Demand

The last statements of the questionnaire focused on the information demand of the participants. First we wanted to know if the participants would like to be informed that their messages were received by the system. With an average of 4.12 (SD: 1.05), most inexperienced participants agreed with the statement. Only 7.92 % rated the statement with 1 or 2 and therefore do not want to be informed. Additionally, the participants would like to know when their messages are displayed. With an average of 4.19 (SD: 1.06), the majority of participants agreed with the statement. Another important point is to know if there is a control or moderation process going on. The average rating was also very high (average: 4.18, SD: 1.11). The information about the persons or institutions who control the content is rather not that important. The average agreement was at 3.91 (SD: 1.22).

The group with experienced participants did not agree with the statements on the same level as the inexperienced group of participants. Their agreement about being informed when the message was received was rated with an average of 3.45 (SD: 1.39). Their agreement about knowing when the message will be displayed is slightly higher (average: 3.85, SD: 1.28), but knowing if there is a control or moderation process of the content is as important for experienced participants as for inexperienced participants (average: 4.15, SD: 0.99). The information about the persons or institutions who moderate the content is also less important. (average: 3.31, SD: 1.32).

5.3.4 Comments of Participants

Twelve participants included comments at the end of the survey. Three participants missed questions about the usefulness of public displays because they do not really think that they are useful. In addition, two other participants stated that they do not know how such a display with personal messages could look like. Two participants also stated that it is very difficult to answer the questions if there is no given location for the public display. Presumably they did not read the introduction and explanation with the example locations. After all, one participant was fascinated by the idea of showing user-generated content on public displays and said that it would be the only reason for him to use such a display.

5.4 Summary and Discussion

In this chapter, we described the design of an online survey with in total 114 participants divided into two group: inexperienced and experienced participants.

Asking participants about their agreement level with six authentication mechanisms showed that authentication is an important issue to address. None of the presented authentication mechanisms was able to get a positively agreement of inexperienced participants. Considering privacy, it is no surprise that the average agreements for authenticating on the display are lower than the ones for authenticating remotely. We think that data security and privacy are topics that have to be considered when developing future applications for public displays.

When asking participants for the type of content that they would post to a public display, again three participants stated that they are concerned about their privacy and would only post content that cannot be traced back to them. One of 114 participants explicitly stated that he would try to misuse the display, but the majority of participants does not seem to have this goal.

Time expectations of users showed that the majority expects messages to be displayed instantly. It is therefore crucial for public display applications to either really display messages instantly or to inform users about the moderation process because they are then willing to wait longer.

Most participants also saw a necessity for content moderation. Reactive moderation and community moderators were the strategies mostly preferred by participants. Pre-moderation was rated very low by experienced participants because they presumably are aware of disadvantages such as high costs and delay times. Summed up, participants do not see a necessity for manually checking every message before or after displaying it.

We did not discover a big difference in agreeing with six potential groups of content moderators, but we were able to correlate moderation strategies with moderators. The following combinations should be used in future public display applications: pre-moderation should be done by display owners or uninvolved third parties, reactive moderation by space owners and users, community moderation should be done by viewers, display and space owners should be community moderators, whereas white list moderation should be done by users. There was no correlation for manual post-moderation.

Additionally, we found that potential users of public display applications have a high information demand and want to be informed about the moderation process and the display time of their messages.

6 UniDisplay: An Application for Displaying User-Generated Content on Public Displays

As described in section 3.2, user-generated content is one of the main sources for conflicts between different stakeholders. Especially display and space owners are skeptical of displaying user-generated content, because they do not want to harm their reputation whilst users do not want to look at advertisements and prefer to see interesting content on displays [MWE+09]. We focused on this conflict by developing an application to further generate insight and experience with user-generated content on public displays. In order to satisfy the needs of display and space owners, a pre-moderation concept could be included in an application to avoid inappropriate content being displayed. Our online survey showed that pre-moderating content manually is a concept that is not at all preferred by experienced users (see section 5.3.3.1), presumably because of the amount of work it requires and the delay that it causes. We designed and implemented an application for public displays to examine this conflict focusing on the users to understand their needs and behavior to examine the problems of user-generated content on public displays. Users were able to post Twitter messages to the display including short text messages and images. The content was not pre-moderated at all, but some mechanisms for post-moderation to delete posts that were already displayed were implemented to be used in emergency cases, e.g., if someone complains about the content. More practical pre-moderation strategies and post-moderation strategies were then simulated by delaying posts regularly.

Below, we describe the vision and the requirements for our application, its design, the hardware which was available and the implementation of the application including the software components that were developed.

6.1 Vision

Our application was build to fulfill ten requirements that we defined during the design, when we did evaluations by looking at APIs of social networks and developing prototypes to test the possibilities of different approaches.

1. The application should display at least text messages and images, potentially also short videos together with a timestamp and the author of the message. The messages should possibly be fetched from a social network such as Facebook or Twitter to get more users and use the authentication mechanisms of the social network. On the display, the users are always able to look at the recent messages sent to the application.

2. The application has to adapt to the size and orientation of the display because the university owns displays with portrait and displays with landscape mode (see section 6.3.1).
3. The time from sending the message to the point that the message is shown on the display should be exactly the same for each user. The minimal time period should be one second to be able to display messages instantly. In addition, it should be possible to include an artificial delay between sending and showing messages which also has to be exactly the same amount of time for all users and should be changeable in fixed intervals.
4. The messages should not be pre-moderated before showing them on the display. Received messages do not have to be approved in any way.
5. When a new message is received in the system, the administrator should be informed via e-mail. The administrator can then keep track of the content on the display and remove any inappropriate content promptly.
6. The view should only reload if new messages were received to avoid a high loading rate of the system.
7. The administrator has to be able to easily delete messages that are shown on the display to remove inappropriate content or delete content that others complained about.
8. After restarting the application, the former messages should still be shown on the display to avoid an empty display after a restart.
9. Each user of the display should get a personalized link to a questionnaire after posting the first message to the display.
10. The most important information of the messages have to be logged and the images have to be stored on the server to be able to reconstruct the content of the display for analysis tasks later on.

These requirements were used to create a design for the application trying to fulfill as many requirements as possible.

6.2 Design

During the design phase, we first looked at Facebook and Twitter as social networks and providers of user-generated content. After a short evaluation of the Facebook API, it was considered to be too complex and too cumbersome for our small application. Additionally, Facebook as social network was not able to fulfill all our requirements. In contrast, Twitter offers three simple APIs and Twitter users are able to post short text messages with 140 characters and images mentioning other users and including hashtags, which perfectly fits to the content that we wanted to show on the displays. Twitter has a crucial drawback which is the small amount of people having and actively using a Twitter account in Germany. According to a survey of BITKOM [BIT13], only 13 % of the Germans own a Twitter account and only 6 % use it actively in contrast to 64 % having a Facebook account of whom 56 % use it actively. Short of other social networks that could be suitable for our task, sending messages via mail was also considered as alternative and tested in a prototype. Unfortunately the prototype was too slow and caused an irregular minimum delay of ten seconds between sending a message via mail and showing the message on the display. Thus, one of the most important requirements which included to show messages instantly was not fulfilled and the alternative was dropped. We also considered using a web interface to send messages to the display but that would have complicated the development of the application and the distribution of the questionnaire which was important to get participants for our survey. Despite the drawback of the small amount of people owning a Twitter account, Twitter was chosen as provider for user-generated content because all of the important requirements were fulfilled.

6.2.1 Twitter APIs

Twitter offers three different APIs, which are the REST API, the search API and the streaming API. The APIs had to be examined to find the best way to show the posts on the display instantly. First of all, we considered loading messages with the REST API which was easy to implement, but the rate limit of the API did not allow to update the messages on the display instantly. In the following, it turned out that the search API was also not suitable for our scenario because it just randomly picks messages with the search terms and it is not guaranteed to get all messages back that fulfilled the search criteria. The API which finally turned out to be usable was the streaming API which allows to listen on a Twitter user stream. The user stream receives all Tweets, Retweets and Mentions of the specified Twitter User. Additionally, the user stream provides deletion notices, disconnect messages, friends lists and events such as new followers or favored tweets. Choosing the API also meant to choose the concept for sending messages, because the user streams do not provide messages with specific hashtags which would have been only provided from the search API. Hence, a Twitter account for the displays had to be created. Mentioning the name of the account in a Tweet ("@unidisplay") is the key to send that message to the display. The REST API is used in combination with the streaming API to reload former messages at a restart of the application.

6.2.2 Application

The application was designed as a web application using Node.js[1] for its server-side. Node.js itself was recommended by several colleagues that already have experience in developing web applications and was considered to be suitable for the display application after a short inspection. The client-side was designed using HTML, JavaScript and the template language EJS (Embedded JavaScript)[2].

[1] Node.js, http://nodejs.org/ (accessed November 21, 2013)

[2] Embedded JavaScript, http://embeddedjs.com/ (accessed November 21, 2013)

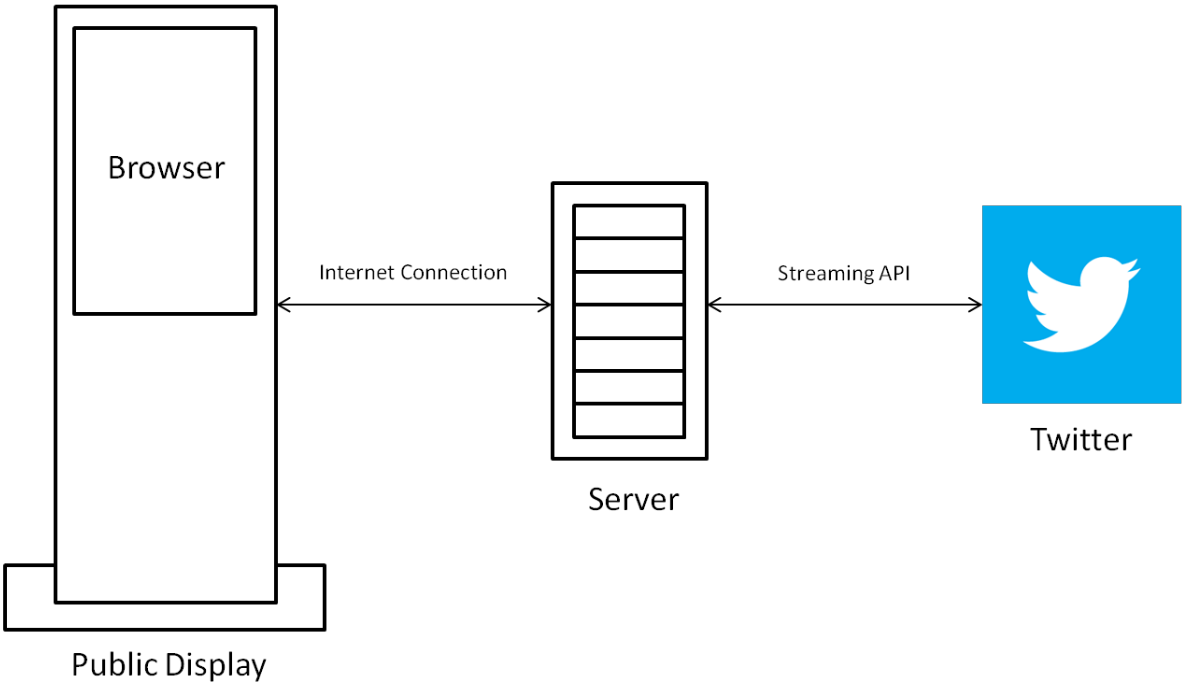


Figure 6.1: Connection between one display, the server and Twitter.

6.2.3 Structure of the System

A structure of the system is shown in Figure 6.1. The public display only needs a browser showing the web application. The browser should be in full-screen mode. An Internet connection is needed to communicate with the server where Node.js is running. The server-side of the application directly connects to Twitter using the REST API and the streaming API to fetch the messages which should be shown on the display. The system is not bound to one public display. More displays can communicate with the server via an Internet connection and get all relevant information, but will also show the same messages.

6.3 Hardware

Five displays which are connected to computers and a server are available for the development and the deployment of the application at the university. In the following, a short summary of the hardware is provided.

6.3.1 Displays

Five displays were available for the deployment of our application. Three of them are portable and two are mounted stationary. The portable displays are all in portrait mode, whereas

Quantity	**Company**	**Type**	**Display Size**	**Display Orientation**	**Touch Screen**
2	eKiosk	Phex Stand 46”	46”	Portrait Mode	yes
1	eKiosk	Phex Stand 52”	52”	Portrait Mode	yes
1	Sasmung	LE32R51B	32”	Landscape Mode	no
1	Panasonic	TH-50PF20	50”	Landscape Mode	no

Table 6.1: Basic information about the available displays for the deployment of our application.

the stationary displays are in landscape mode. Table 6.1 summarizes the most important information about the displays.

Each display is connected to a computer with the operating system Windows 7 and an installed browser. The computers either have an integrated WLAN card or a WLAN stick to connect to the WLAN of the university.

6.3.2 Server

The available server is an Ubuntu server running version 12.04. The Node.js version v0.8.21 is installed and running on the server.

6.4 Software

The Node.js web application consists of a server-side part and a client-side part. Those two parts communicate via an API. The server-side part of the application communicates with Twitter and has access to the file system of the server to store log files and images. The architecture of the system is shown in Figure 6.2 including all important files belonging to the different parts of the application.

In the following, the server-side part and the client-side part of the application are presented with more details.

6.4.1 Server-Side

The server-side of the application was built with Node.js and consists of four parts which are the initialization of the application, establishing the Twitter connection, processing the Tweets and the API for communicating with the client-side. All four parts are described more detailed in the following sections.

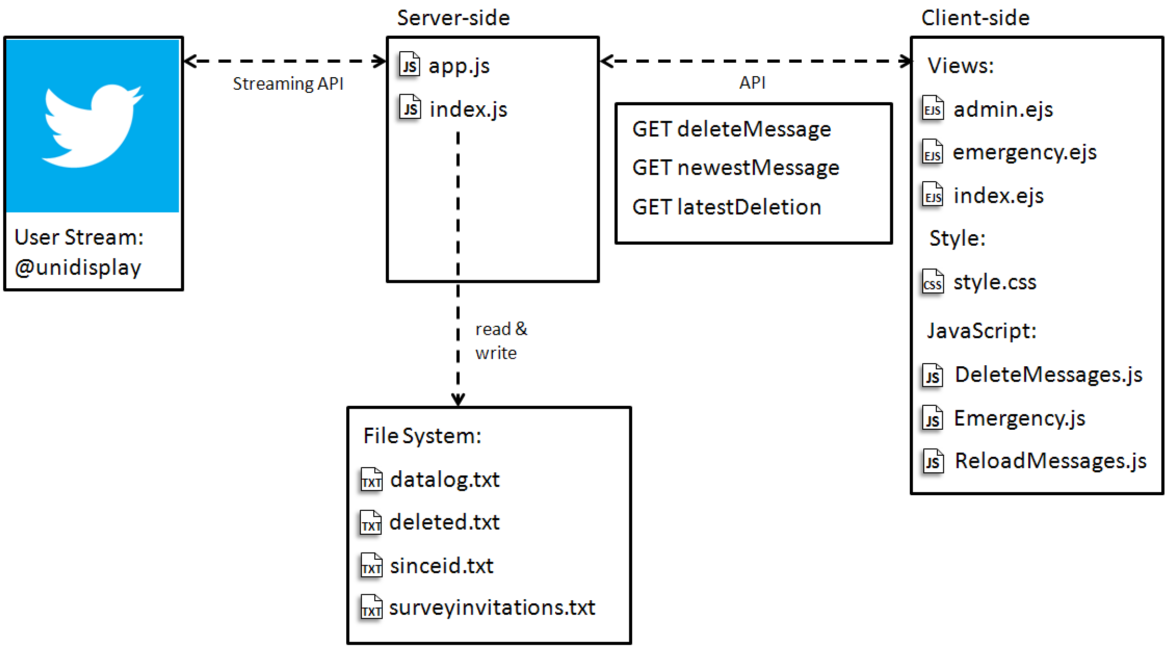


Figure 6.2: The system consists of a server-side and a client-side part which communicate via an API. The server-side part connects to the Twitter streaming API and stores important information of the application in the file system.

6.4.1.1 Initialization

When starting Node.js and running the application, an initialization process takes place. The posts which will be shown on the display are stored in an array which is initialized with the message “This could be your message!” for the number of messages that should be shown on the display. More initialization steps have to take place if the application was already started before and just stopped and restarted in between. The application then reads three log files from the file system to get all necessary information for a restart of the system. The first file read is the file *deleted.txt* which contains the Tweet IDs of all formerly deleted posts. The IDs are separated by semicolons. All IDs are read from the file and stored in a list. Another file (*surveyinvitations.txt*) contains all names of Twitter users that already received their personal link to the questionnaire to avoid sending the link to the same user twice. The names are also stored in a list. The third file only contains one Tweet ID. This ID identifies the last Tweet that was saved to the log. Due to Internet connectivity problems or Twitter related issues there is a chance that Tweets may not be received via the streaming API and have to be logged at a restart. To be sure that only the missing Tweets are logged, the ID of the last Tweet received, is always stored in a variable in the application and also in the *sinceid.txt* file.

After loading the information from the files, a timeout for the artificial delay of messages is set. The number of seconds for the delays are stored in an array which can be easily adapted in the code of the application. Additionally, a time interval is set to switch between the different delay times regularly. The specific choices of delay times and the interval used during our deployment phase of the application are described in section 6.5.

6.4.1.2 Twitter Connection

As mentioned in section 6.2, we decided to use the Twitter streaming API as connection to Twitter. We created a Twitter account named “@unidisplay” and also had to create a Twitter application to get the necessary key information as there are a consumer key, a consumer key secret, an access token key and an access token key secret. The Node.js plug-in user-stream[3] supports establishing a connection to the streaming API of Twitter. All key information is necessary to create a stream and listen to the data of the stream (see Listing 6.1).

```
var Stream = require(’user-stream’);
var myUser = "@unidisplay";

// Twitter key information
var consumer_key = ’...’;
var consumer_secret = ’...’;
var access_token_key = ’...’;
var access_token_secret = ’...’;

// Creates new Twitter user stream
var stream = new Stream({
  consumer_key: consumer_key,
  consumer_secret: consumer_secret,
  access_token_key: access_token_key,
  access_token_secret: access_token_secret
});

// Starts streaming Twitter data
stream.stream();

// Listens to the user stream
stream.on(’data’, function(json) {
  ... // processing data
});
```

Listing 6.1: Establishing a Twitter connection using the streaming API through the Node.js plug-in called user-stream with all relevant key information.

The listener gets a variety of information from Twitter which do not only contain the mentions that we are interested in. Therefore all notifications, events, and messages from the user itself have to be carefully excluded from further processing.

A problem of the streaming API is that former messages cannot be pulled with it. In case of a system restart or a loss of the Internet connection it is possible to miss some Tweets which

[3] Node.js plug-in user-stream on GitHub, https://github.com/aivis/user-stream (accessed November 26, 2013)

were not received by the system. The old messages could possibly be loaded from the log file to make sure that the display always shows former messages but that does not solve the problem of missing Tweets designated to the display. Therefore the application loads former messages when restarting the whole application by using the Twitter REST API once. With the help of the ID of the last logged Tweet, all new Tweets could also be added to the log file and inserted in the system. We use the Node.js plug-in OAuth to communicate with the REST API (see Listing 6.2 for more details).

```
var OAuth = require(’oauth’).OAuth;

// Creates oauth authentification
var oa= new OAuth(
  "https://twitter.com/oauth/request_token",
  "https://twitter.com/oauth/access_token",
  consumer_key, consumer_secret,
  "1.0", null, "HMAC-SHA1"
);

// Loads former messages with Rest API
oa.get("https://api.twitter.com/1.1/statuses/mentions_timeline.json?screen_name=unidisplay",
access_token_key, access_token_secret, function(err, data) {
  if (err) {
    console.log(err);
  } else {
    var arrData = JSON.parse(data);
    ... // processing data
  }
});
```

Listing 6.2: Loading former messages using the Twitter REST API through the Node.js plug-in OAuth.

6.4.1.3 Processing Tweets

When a new Tweet is received through the streaming API or loaded with the REST API after a loss of connection, the key data is extracted from the Tweet which consists of the ID, the message, the media link, the user and the user's ID, and the timestamp. This information is necessary to be able to analyze the content and the users' behavior. A line in the log file is written for every message following the pattern: *tweet ID; message; media link; timestamp; user ID; user name; delay condition; status.* The status indicates if a message was published or deleted, because both actions are stored in the log file. Additionally, the images are stored on the disk of the server using the Node.js module Request. No other information is stored during the process.

After storing the key data of the Tweet, the application checks if the Twitter user already received a survey invitation. If not, the application creates a personal link for the user encoding the ID of the Tweet, the user itself and the delay condition. With the help of the Twitter REST API, a Tweet is posted mentioning the user containing the request to participate at the survey and the personal link to the survey. We also considered sending direct messages to the Twitter users, but this was not possible because Twitter did not allow sending direct messages to everyone but your followers at that point.

6.4.1.4 API

For the communication between server-side and client-side of the Node.js application, a small API was implemented. The API allows three different actions.

Newest Message The newestMessage action returns the ID of the latest Tweet that was received from the Twitter API.
`REQUEST`: GET newestMessage/
`RESPONSE`: e.g., "325175288411586560"

Latest Deletion The latestDeletion action returns the ID of the Tweet that was deleted lastly.
`REQUEST`: GET latestDeletion/
`RESPONSE`: e.g., "325175288411586560"

Delete Message The deleteMessage action allows to specify the Tweet which should be deleted by adding the ID as data to the request. A specific combination also allows to clean the display in cases of emergencies and posts twelve new Tweets to override the entire existing content on the display. The API just responds with "OK".
`REQUEST`: e.g., GET deleteMessage?id=325175288411586560
`RESPONSE`: "OK"

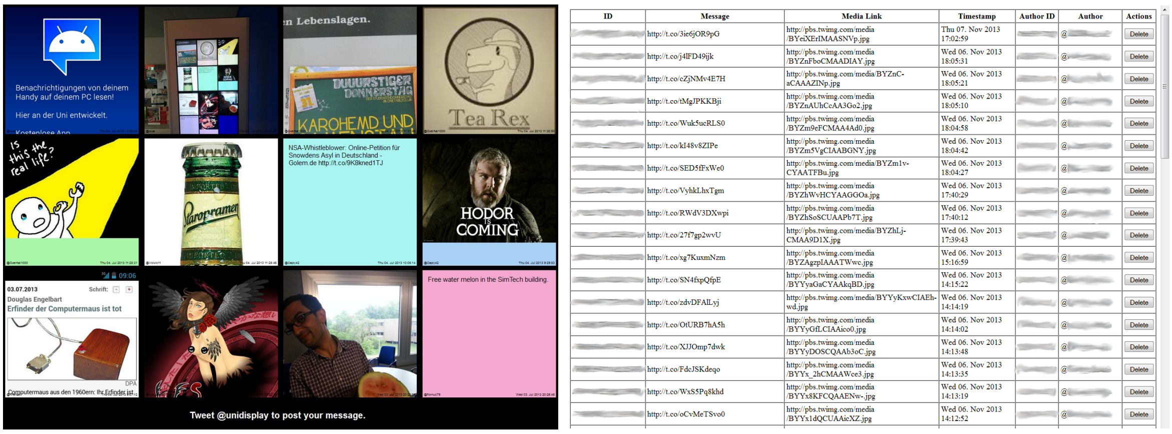
6.4.2 Client-Side

The client-side supports three different views: the display view that is shown on the public displays, an administrator view and an emergency view which are both only known by application developers and selected moderators. Additionally, the client-side part of the application is responsible for updating the page when new Tweets are received on the server-side. The API is used as communication medium to be able to do the actualizations.

In the following, we describe the parts of the client-side more detailed.

6.4.2.1 Views

The client-side part of the application supports three different views. The display view is shown on all public displays and is the only view that the users can see. In contrast, the administrator view and the emergency view offer control mechanisms for display or space owners which allow them to delete or override posts on the display.


Benachrichtigungen von deinem Handy auf deinem PC lesen!
Hier an der Uni entwickelt.
Kostenlose App
Is this the real life?
DUUURSTIGER DONNERSTAG
KAROHEMD UND
Tea Rex
NSA-Whistleblower: Online-Petition für Snowdens Asyl in Deutschland - Golem.de http://t.co/9K8kned1TJ
HODOR IS COMING
03.07.2013
Douglas Engelbart
Erfinder der Computermaus ist tot
Computermaus aus den 1960ern: Ihr Erfinder ist...
Free water melon in the SimTech building
Tweet @unidisplay to post your message.

ID	Message	Media Link	Timestamp	Author ID	Author	Actions
[redacted]	http://t.co/3ie6jOR9pG	http://pbs.twimg.com/media /BYeiXErIMAASNVp.jpg	Thu 07. Nov 2013 17:02:59	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/j4lFD49ijk	http://pbs.twimg.com/media /BYZnFboCMAADIAY.jpg	Wed 06. Nov 2013 18:05:31	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/cZjNMv4E7H	http://pbs.twimg.com/media/BYZnC-aCAAAZINp.jpg	Wed 06. Nov 2013 18:05:21	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/tMgJPKKBji	http://pbs.twimg.com/media /BYZnAUhCcAA3Go2.jpg	Wed 06. Nov 2013 18:05:10	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/Wuk5ucRLS0	http://pbs.twimg.com/media /BYZm9eFCMAA4Ad0.jpg	Wed 06. Nov 2013 18:04:58	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/kI48v8ZIPe	http://pbs.twimg.com/media /BYZm5VgCIAABGNY.jpg	Wed 06. Nov 2013 18:04:42	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/SED5fFxWe0	http://pbs.twimg.com/media/BYZm1v-CYAATFBu.jpg	Wed 06. Nov 2013 18:04:27	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/VyhkLhxTgm	http://pbs.twimg.com/media /BYZhWvHCYAAGGOa.jpg	Wed 06. Nov 2013 17:40:29	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/RWdV3DXwpi	http://pbs.twimg.com/media /BYZhSoSCUAAPb7T.jpg	Wed 06. Nov 2013 17:40:12	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/27f7gp2wvU	http://pbs.twimg.com/media/BYZhLj-CMAA9D1X.jpg	Wed 06. Nov 2013 17:39:43	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/xg7KuxmNzm	http://pbs.twimg.com/media /BYZAgzpIAAATWwc.jpg	Wed 06. Nov 2013 15:16:59	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/SN4fxpQfpE	http://pbs.twimg.com/media /BYYyaGaCYAAkqBD.jpg	Wed 06. Nov 2013 14:15:22	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/zdvDFAlLyj	http://pbs.twimg.com/media/BYYyKxwCIAEh-wd.jpg	Wed 06. Nov 2013 14:14:19	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/OtURB7hA5h	http://pbs.twimg.com/media /BYYyGfLCIAAico0.jpg	Wed 06. Nov 2013 14:14:02	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/XJJOmp7dwk	http://pbs.twimg.com/media /BYYyDOSCQAAb3oC.jpg	Wed 06. Nov 2013 14:13:48	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/FdcJSKdeqo	http://pbs.twimg.com/media /BYYx_2hCMAAWce3.jpg	Wed 06. Nov 2013 14:13:35	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/WxS5Pq8khd	http://pbs.twimg.com/media /BYYx8KFCQAAENw-.jpg	Wed 06. Nov 2013 14:13:19	[redacted]	@[redacted]	Delete
[redacted]	http://t.co/oCvMeTSvo0	http://pbs.twimg.com/media /BYYx1dQCUAAicXZ.jpg	Wed 06. Nov 2013 14:12:52	[redacted]	@[redacted]	Delete

(a) Display View

(b) Administrator View

Figure 6.3: The UniDisplay application supports different views. The display view is shown on the public displays whilst the administrator view allows administrators to delete messages. Tweet IDs and user information were removed for privacy reasons.

Display View The display view shows twelve colored divs adapting their layout to portrait or landscape mode with a 3 × 4 or a 4 × 3 grid. Each colored block contains a message or an image posted on the display. The blocks also contain the author and the timestamp of the message which are displayed at the bottom left respectively bottom right corner of the block. The explanation of the application and how to send messages is shown at the bottom of the display. Figure 6.3a shows the display view.

Administrator View The administrator view consists of a simple table with all entries shown on the display and offers the possibility to delete a specific entry. Clicking on the delete button calls the deleteMessage action of the server-side is with the tweet ID of the message that should be deleted. The server-side then deletes the message (see section 6.4.1.4). Figure 6.3b shows the administrator view.

Emergency View The emergency view only contains one button which could be pressed in case of the display showing inappropriate content. With a click on the emergency button, the deleteMessage API of the server-side is called. A specific ID is used for the emergency override when calling the API (for the server-side see section 6.4.1.4).

6.4.2.2 Updating Views

When a new message arrives via Twitter, the page of the web application has to be reloaded to show the new message. An easy solution is to just reload the page in regular intervals. These intervals have to be very short to be able to show all messages nearly instantly. To

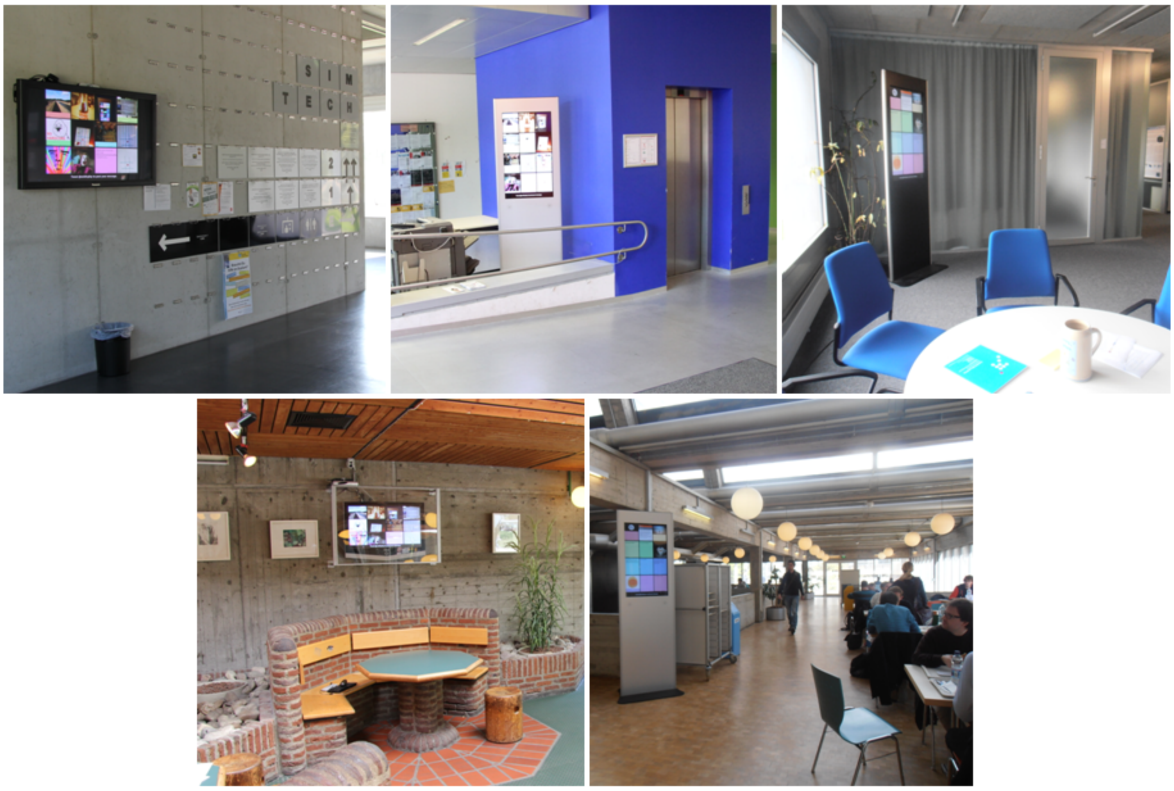


Figure 6.4: All display locations in the entrance areas of faculty buildings, a coffee kitchen and two cafeterias.

avoid flickering when reloading the page every second without knowing if it is necessary, the client-side was implemented to call the API of the server-side to check whether the page has to be reloaded. Two IDs are stored on the client side, which are the ID of the newest message that is shown at that moment and the ID of the latest message that was deleted. Calling the newestMessage action and the latestDeletion action, the client-side JavaScript receives the ID of the newest message on the server and the ID of the latest message that was deleted on the server. Comparing these IDs makes it possible to find out whether the client-side view is up-to-date or the page has to be reloaded. The API is called every 1000 milliseconds, which could be adapted in the code.

6.5 Deployment

The application was deployed on five displays across campus. Display locations are shown in Figure 6.4. Two displays are situated in the entrance areas of faculty building next to lecture rooms, one is situated in a coffee kitchen for employees and two displays are situated in the main cafeterias on the campus.

The deployment started in June with a test deployment on one of the displays until it was spread out to the other displays. The application is still running on four of the five displays, but all content analyzed in this diploma thesis was generated between the 22nd of June and the 16th of August. Later interaction with the public displays and our application is not part of the analysis.

To be able to observe the effect of pre-moderation, delays were inserted to the app. Messages were shown 0 seconds, 30 seconds or 90 seconds after they were posted to Twitter. The 0 seconds condition stands for a display with no pre-moderation or automatic checks (e.g., bad word check) which does not take up to one second. The 30 seconds condition could be an employee of a display owner who only reads and manually approves messages for the display and directly decides if they are shown or not. 90 seconds were predicted to be feasible for a crowd-based concept as Amazon's platform Mechanical Turk or another community based approach.

The delay conditions changed every two hours while lectures took place, because we expected most of the students to post messages during the lecture breaks and at lunch time.

6.6 Summary and Discussion

We implemented the UniDisplay, an application for showing user-generated content on public displays. During the design process, we looked at Facebook, Twitter, mail and a web interface as input possibilities. In the end, we decided to fetch the messages from Twitter using the streaming API. The UniDisplay itself is a web application developed with Node.js, JavaScript, HTML, and EJS and consists of a server-side part and a client-side part. The server-side part initializes the application, establishes the Twitter connection, and processes all incoming Tweets. Furthermore, the server-side stores important data on the file system, coordinates sending links to the questionnaire, and offers a REST API for communicating with the client-side part. The client-side part offers three views: display, administrator and emergency view. The display view is shown in the web browser on the displays.

After finishing the implementation, we deployed the UniDisplay on five displays at different locations on campus for a period of eight weeks. We choose delay times of 0 seconds, 30 seconds and 90 seconds to simulate different moderation strategies and changed the delay conditions every two hours. In the next chapter, we analyze the data captured during the deployment.

7 Analyzing Deployment Data of the UniDisplay

After implementing the application, we deployed it on five displays across campus for a period of eight weeks from 22nd of July to 16th of August. During the deployment phase, we collected data of all posted Tweets. Additionally, we observed users when posting content, analyzed the content which was posted, sent out a questionnaire, and conducted interviews.

In this chapter, we present the results from our data analysis with the distribution of the posts across days and time of days and the effect of the delay conditions. Next, we present a content categorization schema for all posts on the display and report interesting interactions between users knowing or even not knowing each other. Additionally, we gained insight from observations of users, a survey and semi-structured interviews with seven users of the application.

7.1 Data Analysis

We analyzed 519 posts from 95 different authors which were posted during the time span of eight weeks. 82 % of the posts were images. All other posts contained text. Multiple Tweets in a row of the same user were summarized as one post with multiple Tweets if the Tweets did not make any sense on their own, but only together, because users constructed big images out of single images or words out of single characters. Counting those as one post, 346 posts remain for analysis. The data analysis does not concentrate on the content of the posts, because a detailed analysis of the content is provided in section 7.3. In contrast, the aims of the data analysis are to identify typical user behavior and discover the influences of the simulated moderation strategies by delaying messages.

7.1.1 Distribution of Posts

Looking at the distribution of the posts during the time of deployment, we found a high peak of posts in the second week with 166 posts. Removing multiple Tweets of the same person in a row reveals that 89 posts remain (see Figure 7.1). The number of posts then decreases with the beginning of the semester break at the end of July to four posts a week. Afterwards the number increases again to 25 up to 35 posts per week.

We also looked at the posts per day of the week (see Figure 7.2). The number of posts on the display is very high on week days such as Monday, Tuesday, Wednesday and Thursday, but already decreases on Friday and is very low on Saturday and Sunday which could be explained by the fact that nobody is able to look at the displays on weekends because university is closed.

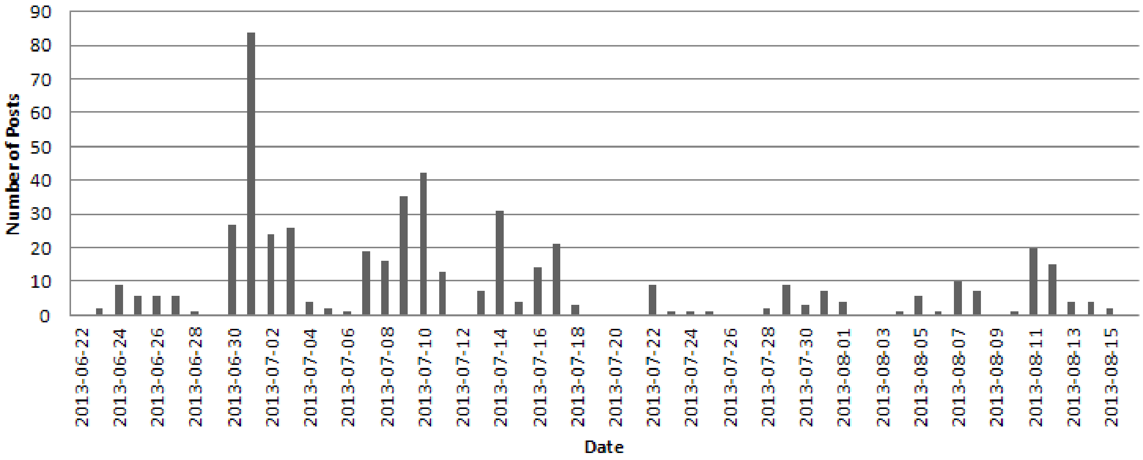


Figure 7.1: Number of posts per day during the deployment phase of eight weeks without multiple Tweets.

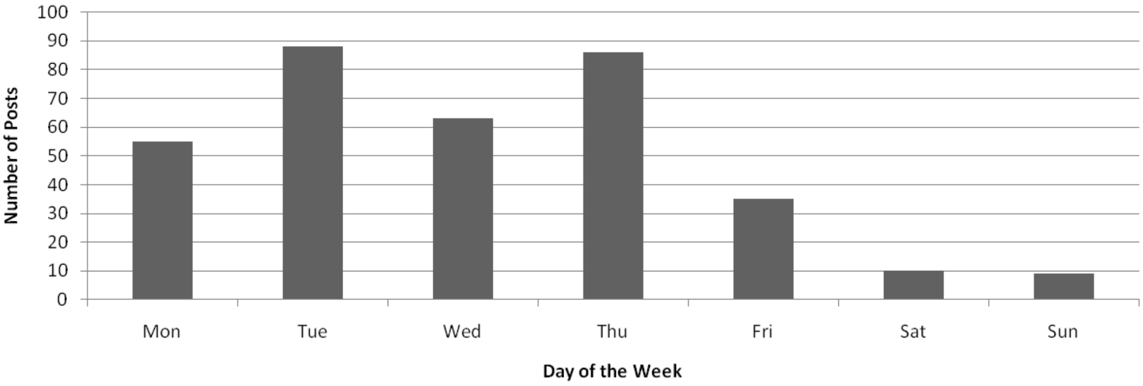


Figure 7.2: Number of posts per day of the week during the deployment phase of eight weeks without multiple Tweets.

Presumably also less students are at university on Fridays because they prefer to keep that day free of lectures or stay at home. The highest number of posts on Tuesday could be explained by the fact that people started posting multiple Tweets on a Tuesday. Although we removed multiple Tweets for the analysis, it is possible that the high number of multiple Tweets also increased the total number of posts because more users were interested in interacting with the display when they saw multiple Tweets of the same person in a row.

Looking at the number of posts per time of day (see Figure 7.3) reveals what we already expected before. Most posts were written at lunch time and during the lecture breaks when most of the students passed the displays because they left the lecture buildings or entered the cafeteria. The lecture breaks are between 9 a.m. and 10 a.m., 11 a.m. and 12 p.m., 3 p.m. and 4 p.m., and 5 p.m. and 6 p.m. The lunch break takes place between 1 p.m. and 2 p.m.

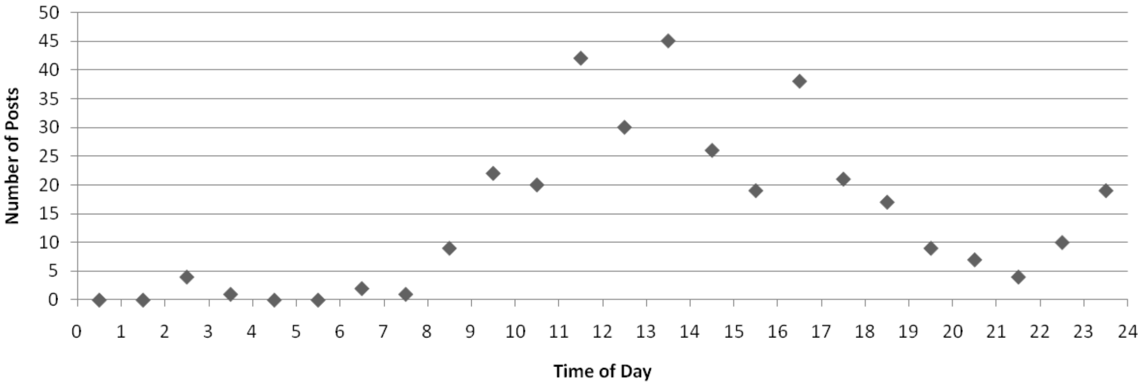


Figure 7.3: Number of posts per time of day during the deployment phase of eight weeks without multiple Tweets.

Nevertheless, posts were also published during lectures or late at night, which is an indicator that people also posted content without being in front of one of the displays.

7.1.2 Delay of Posts

We also had a look at the effects of delaying the posts. Figure 7.4 shows the number of posts per delay condition. 147 posts were posted during the 0 seconds condition, compared to 126 posts during the 30 seconds condition and only 72 posts in the 90 seconds condition. A chi-squared test reveals that the difference between the number of posts is statistically significant ($\chi^2(2) = 27.62$, $p < 0.001$) which indicates a strong effect of the moderation delay on the number of posts.

We also examined the influence of the delay time for the first post of each user. We expected that users with a longer delay time would less often post again than users with a shorter delay time, but we did not find a significant association between the delay times and whether a user posted again or not ($\chi^2(2) = 0.73$, $p = 0.76$). Figure 7.5 shows that under the 0 seconds condition 43.24 % did not post again compared to 51.72 % in the 30 seconds condition and 41.38 % in the 90 seconds condition.

7.2 Observations of Users and Content

During the time of deployment we also observed users in front of the display even listening to their conversations about the display and the application.

Observing people in front of the display revealed a lot of insights in their behavior and thoughts. People posting in the 90 seconds condition often wondered why the content is not displayed instantly or faster. When the application, e.g., was deployed on an open day of the university people posting images asked presentors next to the display how long it will take until the images

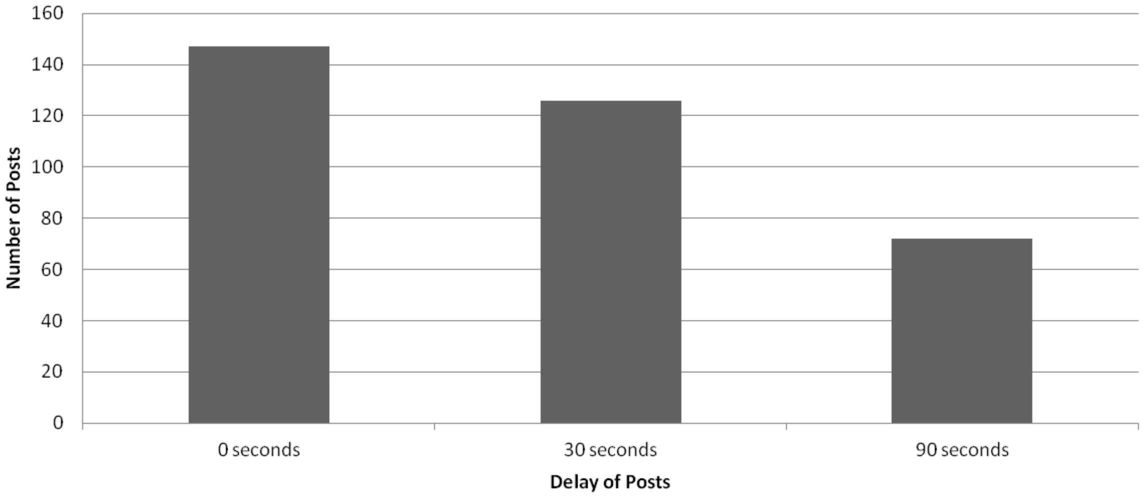


Figure 7.4: Total number of posts per delay condition.

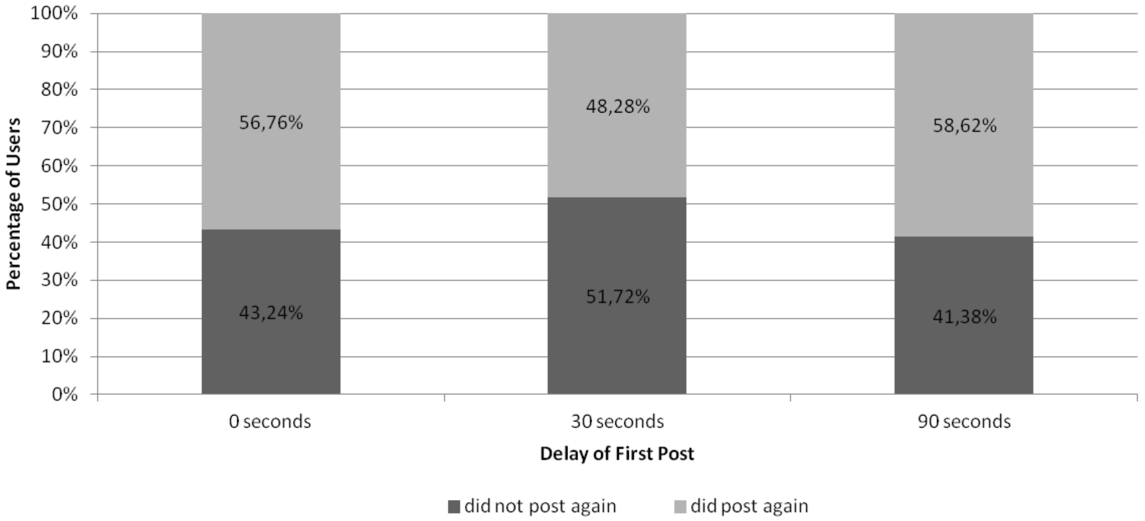


Figure 7.5: Percentage of users who posted again depending on the delay of their first post.

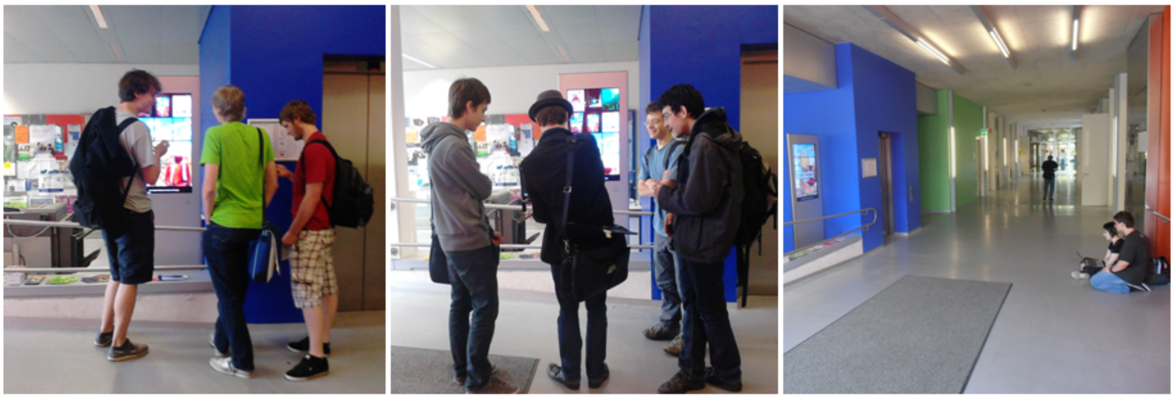


Figure 7.6: Observing users in front of the display who use different devices and posting strategies.

are displayed. They seemed to be impatient and even posted an image again to check if it just did not work for the first time. We also observed users who have never used Twitter before and posted under the 90 seconds condition. In front of the display, they discussed together if the display is working and were unsure if they did everything right when posting. After checking everything on Twitter they decided that it had to be the display's fault. In contrast, other users thought that there is a kind of moderation or content check in the background which takes some time. One person even posted a message on the display asking if the content is moderated.

During our observations we discovered that users post with different devices and have different posting strategies (see Figure 7.6). Most users posted content when passing with their smartphone, others carried a laptop to post their content and two users were even observed sitting on the floor in front of the display with their laptop to be able to post content and look at the display at the same time.

We also observed a honey pot effect when larger groups of people stand in front of the display which also attracted other passersby.

An extraordinary case happened when the display in the coffee kitchen lost the Internet connection once so that it was not able to update the content anymore. One person posted content to the display during this period which was not displayed. The person became angry about the delay and posted a negative opinion about the application. A similar situation happened when a user tried to post an animated GIF image which did not show up properly. The user also complained on the displays about the fact that it is not working. These situation show that users getting angry about the the display installations have a high potential of complaining about the situation by insulting the display or the application running on the display. In the end, the posted messages were not too bad, but clearly showed that the users were not satisfied with the mode of operation on the display.

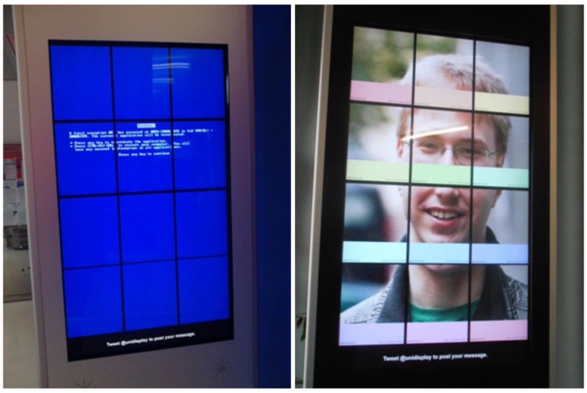

Tweet @unidisplay to post your message.
Tweet @unidisplay to post your message.

Figure 7.7: Big images which consist of twelve single Tweets.

A couple of users tried to use the display in a way that was not intended from us. They posted twelve Tweets in a row to form a big image with their Tweets (see Figure 7.7). We did not ban or prevent this behavior.

7.3 Content Analysis

Besides the data analysis and the observations, we decided to do a content analysis of all content posted to the displays. Therefore we grouped and categorized the posts developing a categorization schema. The aim of the content analysis was to identify types of content on public displays and to figure out if a fear of offensive or inappropriate content is justified.

In the following, we describe the categorization process and all categories that we found during the process naming examples for each category. Further on, we identify interactions between different groups of users to show how the application can encourage users to communicate with each other.

7.3.1 Categorization

In order to understand the users and the generated content, we decided to do a content categorization of all posts. The categorization helps understand what content has to be expected on public displays and whether or not offensive content is a real problem for open public displays and the related stakeholders.

7.3.1.1 Process of Categorization

During the deployment phase all messages and images were stored on the disk of the deployment server. Afterwards it was then possible to reconstruct all states of the display. A small script put all posts on an HTML page which was then converted to a PDF file and printed leading to 34 pages with small posts on it. Three people had a look at these printed posts on their

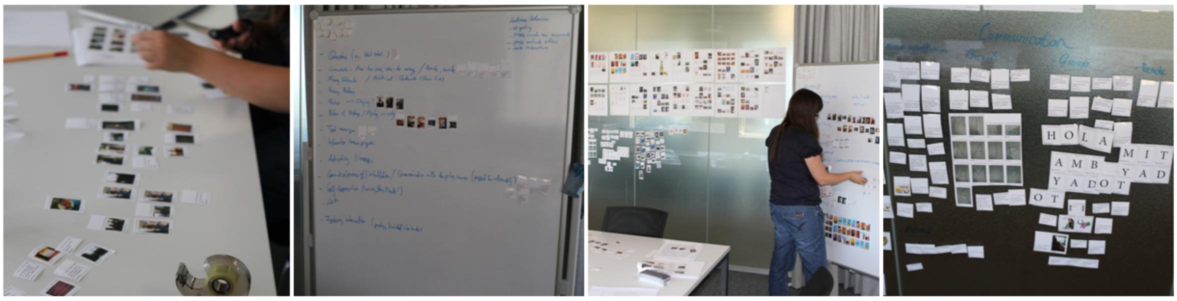


Figure 7.8: Process of assigning posts to categories.

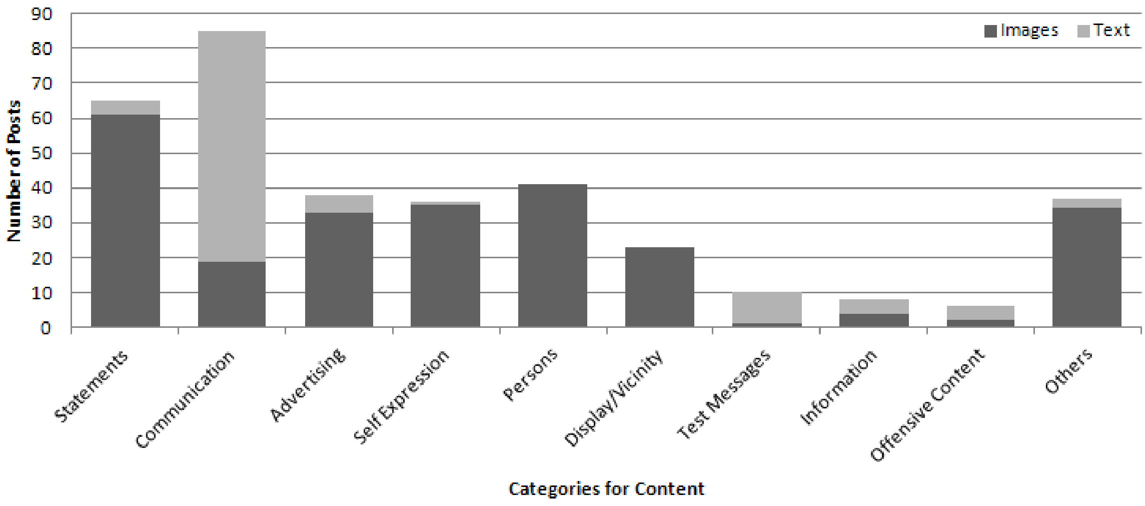


Figure 7.9: Number of posts per category for all content posted during the deployment phase of eight weeks. Images and text are represented with different colors.

own and developed a categorization schema. In a meeting, we then discussed all possible categories assigning the posts to the appropriate categories. Therefore we cut them all out and pinned them onto a glass wall. This process of assigning posts to categories is shown in Figure 7.8. Pinning the posts onto the wall gave us a good overview on all the content that was displayed during the period of eight weeks just by looking at the wall which helped during further analysis.

After assigning all posts to our initial categories, we started to group or reassign them and redefined the categories building on our insights, e.g., at the beginning we started with separated categories for text and images matching them together later. At last, we ended up with ten different categories. In the following, each category will be discussed and explained showing examplary posts. An overview over all categories and the number of posts in each category is shown in Figure 7.9.

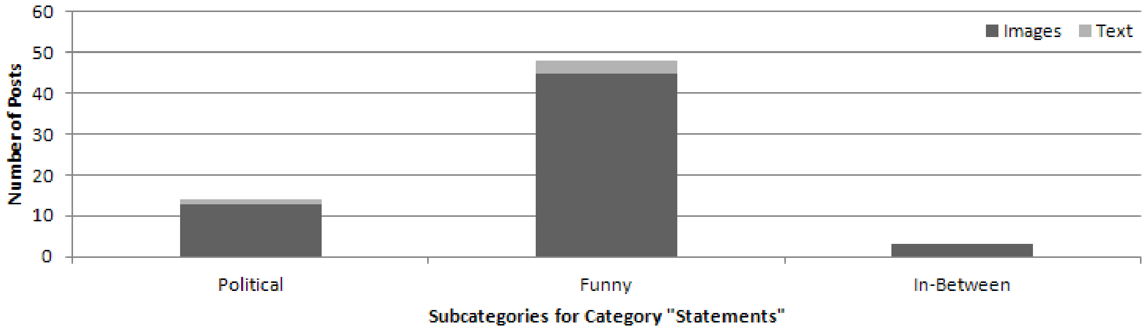


Figure 7.10: Number of posts per subcategory for the category “Statements”.

7.3.1.2 Statements

The category “Statements” covers all posts that make a statement. We subdivided the 65 posts in two major subcategories, which are “Political Statements” and “Funny Statements”. Three posts could not be placed in any of the two subcategories because they are somehow funny but also critical or reflective without being political. Those posts were therefore classified as in-between. In total, 48 posts were classified as funny and 14 posts as political (see Figure 7.10).

Political Statements The subcategory “Political Statements”, e.g., contains caricatures of German politicians such as Angela Merkel and international politicians such as Barack Obama with related current topics as the NSA affair. Users also posted religious caricatures, e.g., showing the pope.

Examples for images:

Wann
ist der Mensch
HIRNTOT?

BABE COME TO BED...
NO! I MUST CONTROL THE
INTERNET.

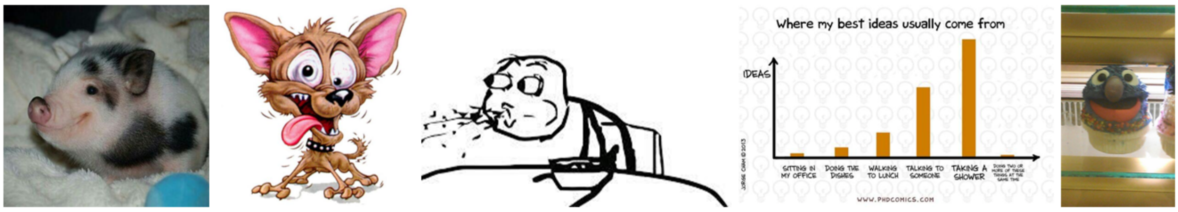
BIG BROTHER
IS WATCHING
YOU

Funny Statements The subcategory of “Funny Statements” mostly consists of funny images and only three textual messages. The images show cartoons, animals, memes, funny statistics, funny signs or even funny scenes from movies.

First example: *I am on TV!*[1]

Second example: *"I have evolved beyond jokes. I am now a robot. Beep Boop."*

Examples for images:



7.3.1.3 Communication

The category "Communication" contains 85 posts and is therefore the category with the highest number of posts. It includes all posts that generally cover any aspect of communication. 66 posts in this category were textual posts and images were only rarely used for communication. We split the posts in six different subcategories (see Figure 7.11) to differentiate between topics and addressees of the posts. First of all we extracted all posts about the installation of the display and the application to be in an own subcategory because of the special topic. We also identified messages that fostered interaction and answers to messages. All other communicative posts were divided by their addressees.

About Installation The first subcategory contains 19 posts about the installation of the displays. The posts comment the displays or the application itself and ask questions about it, because the users were really interested in how the application works and what can be done

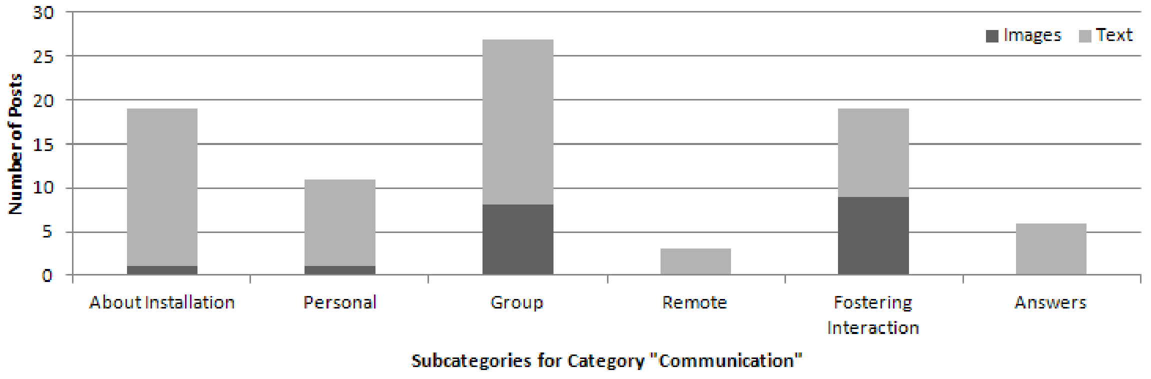


Figure 7.11: Number of posts per subcategory for the category "Communication".

[1] This message was translated. Original Message: "Ich bin im Fernsehen!"

with it. We replied to some of the questions on the display which are also posts belonging to this subcategory.

First example: *Display, what are you doing here?*[2]

Second example: *"This display is cool."*

Third example: *Does someone control the content?*[3]

Personal Communication The next subcategory of the category "Communication" is "Personal Communication". In eleven Tweets users communicated with one other person, e.g., a friend or university employee. They often mentioned the name of the person in their posts and sometimes even another Twitter user's profile. The posts varied from simple greetings up to wishing the best for specific events as presentations or exams.

First example: *Best wishes to Harald :)*[4]

Second example: *"Davids Final Presentation. Wishing all the best."*

Group Communication All posts that were not specifically directed to one person, but a multitude of persons, were placed in the subcategory "Group Communication". The 27 posts in this category were not really personal but more general and addressed to the community.

First example: *Good luck in the written exam.*[5]

Second example: *We are already going to grab food!*[6]

Third example: *"HOLA"*[7]

Remote Communication Three users even posted remotely. We were able to identify their messages because they mentioned their current location. Those were mainly researches from other universities who heard about our project and wanted to test the system.

Example: *"Greetings to the hciLab from sunny Barcelona."*

[2] This message was translated. Original Message: "Display, warum stehst du hier?"
[3] This message was translated. Original Message: "kontrolliert jemand was da angezeigt wird?"
[4] This message was translated. Original Message: "Liebe Grüße an Harald :)"
[5] This message was translated. Original Message: "Viel Glück in der Scheinklausur."
[6] This message was translated. Original Message: "Wir sind schon mal essen!"
[7] Spanish word for hello.

Fostering Interaction The subcategory "Fostering Interaction" contains posts that animated other users to interact with the display and answer to the posts. In total, the subcategory contains 19 Tweets, but not all attempts of increasing the interaction were successful. We describe the interactions between users more detailed in section 7.3.2.

First example: *"keep tweeting on till @Uni_Stuttgart feels like a birdcage. Use it for art and protest, leaks and fun! (o:"*

Second example: *Show your ankles.*[8]

Third example: *"When you read this tweet stand up and clap your hands."*

Answers Short answers to Tweets were grouped in the category "Answers". Those answers consist of smileys or short comments on further Tweets.

First example: *":-)"*

Second example: *":-D"*

Third example: *"awesome tweet!"*

7.3.1.4 Advertising

The category "Advertising" covers 38 posts that presented offers or advertised anything, e.g., an application developed at university or a web page. This category is the category with the largest number of repeated posts. All other categories only contain up to two repeated posts, whereas in the category "Advertising" 14 out of 38 posts were just a reposting of former posts by the same user.

Examples for images:

Benachrichtigungen von deinem Handy auf deinem PC lesen!
Hier an der Uni entwickelt.
Kostenlose App für Android:
http://tiny.cc/notif

[8] This message was translated. Original Message: "zeigt eure knoechel"

7.3.1.5 Self Expression

35 images and one textual message that express the personality or feelings of the users are grouped in the category “Self Expression”. Mostly, the users posted logos from soccer teams or their favorite brands, e.g., of beverages or computers and images of cars, movies, or actors.

Example: *“Vive la France!!”*

Examples for images:

7.3.1.6 Persons

We also encountered 41 images of persons on the display. Those were mainly images that the users took from themselves or their friends. We therefore divided the category into three subcategories: “Staging”, “Friends/Family”, and “Other Persons”. We needed the subcategory “Other Persons” to classify three images of university employees which were posted by students and therefore did not fit in the other two subcategories. An overview of the category “Persons” is shown in Figure 7.12.

Staging The subcategory “Staging” covers fifteen images that users did from themselves trying to draw attention to themselves, e.g., by doing funny poses or wearing funny clothes.

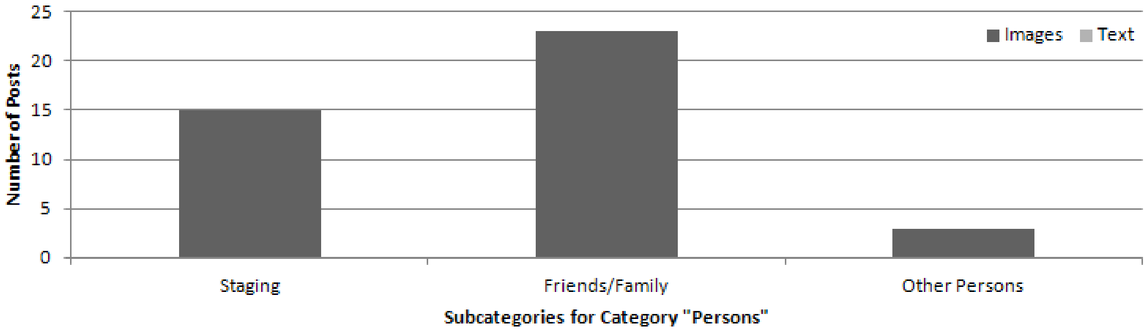


Figure 7.12: Number of posts per subcategory for the category “Persons”.

Examples for images:

Friends/Family 23 Tweets showed groups of people which were friends or family members of the users. More than half of them were taken at the university, e.g., in lecture rooms, during lunch time and while working or relaxing during the lecture breaks.

Examples for images:

7.3.1.7 Display/Vicinity

In total, the users posted 23 images of the displays and the vicinity of the displays. Often those messages were the first message of a user and seemed to be used as a test message.

Examples for images:

7.3.1.8 Test Messages

In addition to images from the displays and the vicinity, ten Tweets were clearly identifiable as test messages. Mostly, users posted short textual messages to get an idea on how the display works. The messages in this category were all posted as first message.

First example: *"Hello world :-)"*

Second example: *"Teeest!!"*

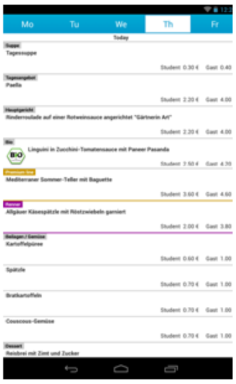
Third example: *"hi"*

7.3.1.9 Information

Eight posts were classified as information posts. These were for example the menu of the cafeteria or posts informing about current news.

First example: *"isotonic (Greek ίσος ísos 'equal' and τόνος tónos 'stretching')"*

Examples for images:



Suppe
Tagessuppe
Student 0.30 € Gast 0.40

Tagesangebot
Paella
Student 2.20 € Gast 4.00

Hauptgericht
Rinderroulade auf einer Rotweinsauce angerichtet "Gärtnerin Art"
Student 2.20 € Gast 4.00

Bio
Linguini in Zucchini-Tomatensauce mit Paneer Pasanda
Student 2.50 € Gast 4.30

Premium line
Mediterraner Sommer-Teller mit Baguette
Student 3.60 € Gast 4.60

Renner
Allgäuer Käsespätzle mit Röstzwiebeln garniert
Student 2.00 € Gast 3.80

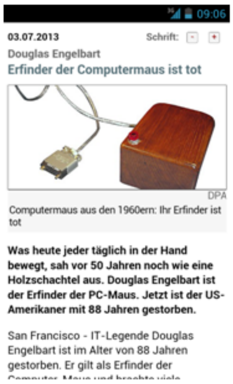
Beilagen / Gemüse
Kartoffelpüree
Student 0.60 € Gast 1.00

Spätzle
Student 0.70 € Gast 1.00

Bratkartoffeln
Student 0.70 € Gast 1.00

Couscous-Gemüse
Student 0.70 € Gast 1.00

Dessert
Reisbrei mit Zimt und Zucker



03.07.2013

Douglas Engelbart

Erfinder der Computermaus ist tot

Computermaus aus den 1960ern: Ihr Erfinder ist tot

Was heute jeder täglich in der Hand bewegt, sah vor 50 Jahren noch wie eine Holzschachtel aus. Douglas Engelbart ist der Erfinder der PC-Maus. Jetzt ist der US-Amerikaner mit 88 Jahren gestorben.

San Francisco - IT-Legende Douglas Engelbart ist im Alter von 88 Jahren gestorben. Er gilt als Erfinder der

7.3.1.10 Offensive Content

During our eight week deployment, we only classified six posts as offensive, which were in total less than 1.5 % of all the Tweets that were posted to the display. We expected a higher amount of offensive or inappropriate content and were surprised by the small number of posts in this category. Furthermore, in our eyes the posts were not really bad. Four of the posts contained swearwords which were therefore removed. We also removed an image of a person which was manipulated to avoid bullying on the display. The last image we removed was a photo of a woman at the beach wearing a bikini, because a viewer complained about the image.

During the deployment phase we discovered that the opinions about offensive or inappropriate content differ very much. The best example was a student posting photos of other students which were converted to zombies. Viewers complained about the images because they did not like them. Talking to the user who posted the images revealed that he did not had in mind to

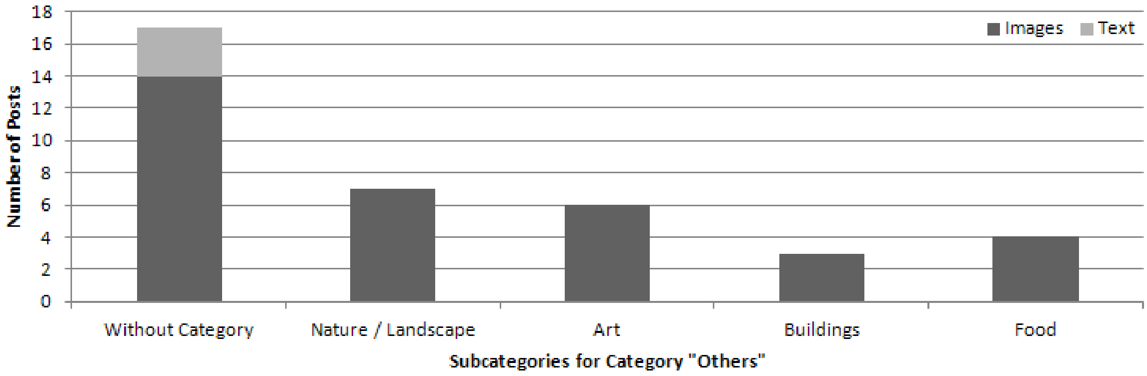


Figure 7.13: Number of posts per subcategory for the category “Others”.

post offensive content. He sat in a boring lecture when downloading an application for his smartphone turning photos of friends into photos of zombies and even asked them if he is allowed to post them to the UniDisplay. Other viewers seeing the images displayed did not know the background and misinterpret the purpose of the images.

7.3.1.11 Others

37 posts were categorized in the category “Others” because only seven or less Tweets had the same topic. We split the posts in the category in five small subcategories, because images were identified to show nature, food, art or buildings. We were not able to categorize the other images and text messages because they had no particular or common topic at all.

First example: *“WAAAAAAAAAAH”*

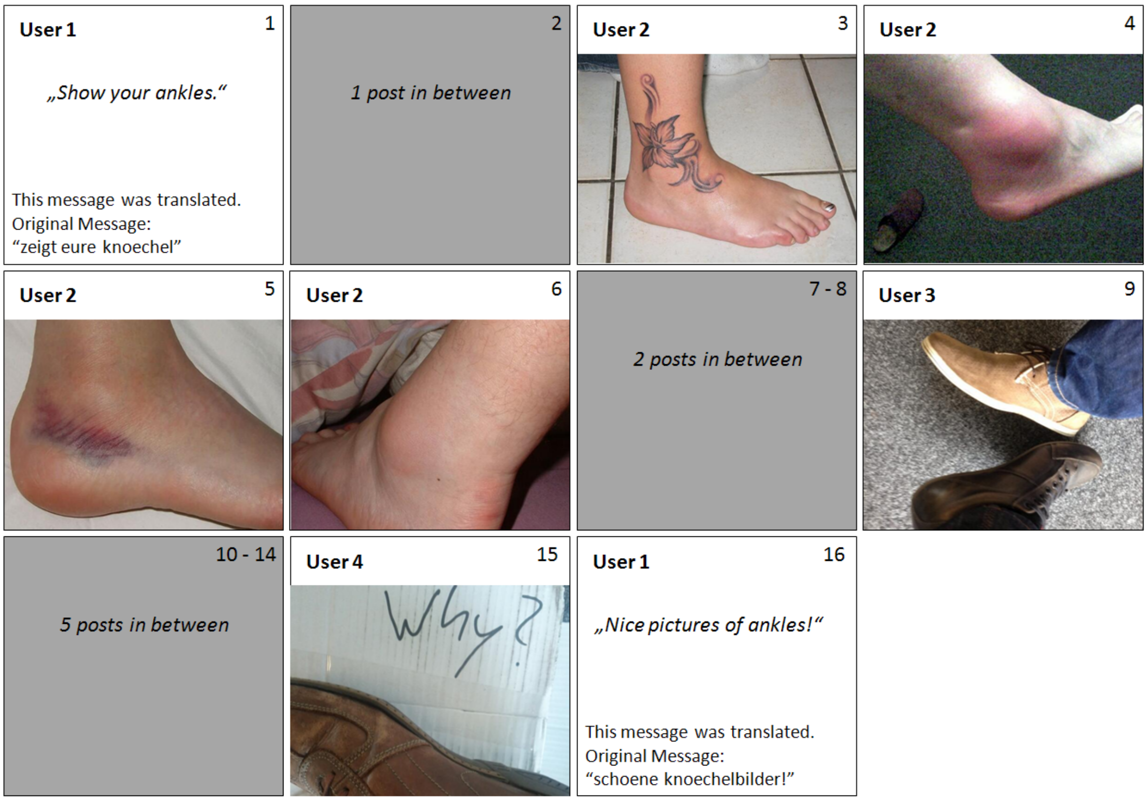
Second example: *“ottt :D”*

Examples for images:

7.3.2 Interactions between Users

During the analysis of the content we also tried to identify relations between content on the display to analyze the interactions of the users with each other. We were able to identify reactions on former posts due to related topics. First of all, we observed a kind of interactive self-regulation. A photo was for example posted to the display twice. After some days it appeared on the display again, because someone did a Retweet of it on Twitter. Another user reacted with a caricature saying “not that shit again”. One user also expressed that he did not like parts of the content by for example posting a caricature containing the text “there is the door”. Users posting big pictures that consisted of several Tweets also complained about others destroying their pictures, because they tried to keep it on the display as long as possible. Besides reacting to posts they did not like, users also showed that they liked specific content posting smileys and related images.

One statement fostering interaction on the displays was answered by three users. The interaction took place on the 10th of July between 11 a.m. and 6 p.m. One user started the conversation by posting a request to post photos of ankles. Two users followed the request and posted photos of ankles. When the first message with the request was no longer shown a fourth user, knowing the second user that reacted, posted a photo which shows that he was wondering why the other users posted photos of ankles. All messages and images are shown in Figure 7.14. The users are named user 1 up to user 4. This is a good example for communication between a



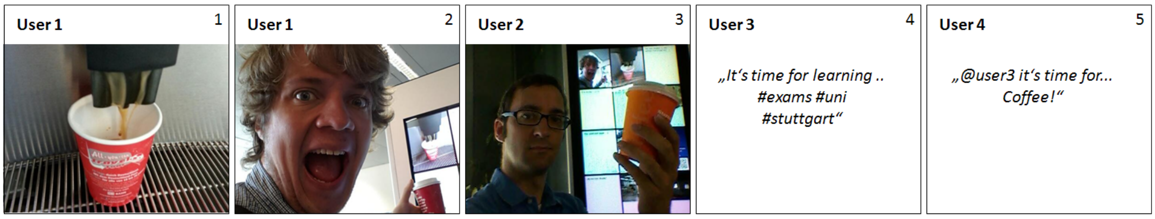
„Show your ankles.“

This message was translated.
Original Message:
“zeigt eure knoechel”

„Nice pictures of ankles!“

This message was translated.
Original Message:
“schoene knoechelbilder!”

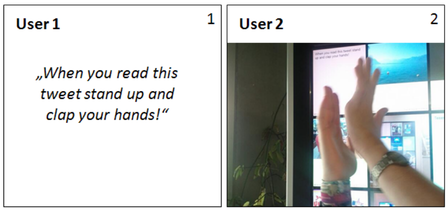
Figure 7.14: Interaction between four users on the displays. User 3 and user 4 know each other.



„It's time for learning .. #exams #uni #stuttgart"

„@user3 it's time for... Coffee!"

Figure 7.15: Interaction between four users regarding coffee.



„When you read this tweet stand up and clap your hands!"

Figure 7.16: Fostering interaction on the display with a Tweet that requests viewers to clap their hands.

group of users that do not know everyone else in the group and just reacted to the content on the display, because they either thought that it was funny or did not understand the content.

Another interaction took place between two friends using the UniDisplay to communicate with each other about the display. The messages were clearly addressed to the other user by including the Twitter name of the other person. The users are also named user 1 and user 2.

User 1 posts several messages to the UniDisplay.

User 2: "*@user1 you are slowly becoming the biggest fan of unidisplay[.]*"

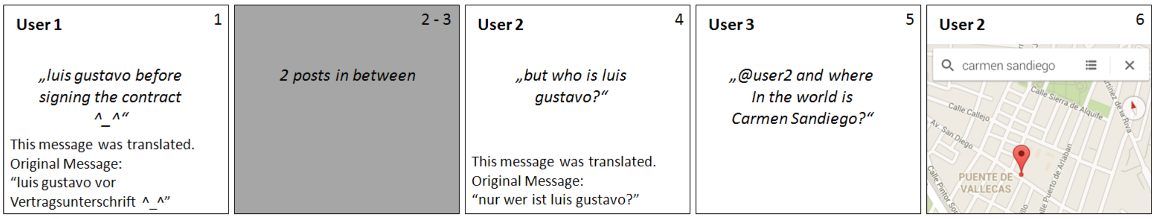
User 1: "*@user2 now it's your turn to publish something at unidisplay, go ahead.*"

User 2 posts a photo of a beverage to the UniDisplay.

The posts were not displayed on the displays at the same time. Other users posted more than 20 Tweets between the parts of the conversation. Nevertheless the two users communicated through the displays and user 1 even animated user 2 to use the display more often.

Another interaction was caused by a user posting one image of his coffee and an image of himself with the coffee in his hand. Another user imitated the second image and posted it to the UniDisplay as well. Coffee was again addressed in a further Tweet answering to a Tweet of a person that was learning for exams at university. The Tweets are presented in Figure 7.15.

One user also tried to get some interactivity on the displays and posted a Tweet that requested all viewers to clap their hands when they read the Tweet. Two other users made a photo together, one user standing in front of the display clapping his hands with the Tweet of the first user in the background and posted the photo to the UniDisplay (see Figure 7.16).



User 1 1

„luis gustavo before signing the contract ^_^"

This message was translated.
Original Message:
"luis gustavo vor Vertragsunterschrift ^_^"

2 - 3

2 posts in between

User 2 4

„but who is luis gustavo?"

This message was translated.
Original Message:
"nur wer ist luis gustavo?"

User 3 5

„@user2 and where In the world is Carmen Sandiego?"

User 2 6

Figure 7.17: Users reacting on a Tweet addressing the signing of a contract by Luis Gustavo.

Another reaction on former Tweets evolved when a user posted a Tweet about Luis Gustavo. The answers of two other users are shown in Figure 7.17.

The examples for interactions show that users who know each other or do not know each other communicate on a public display with user-generated content. The UniDisplay application could therefore help to build and support a community.

7.4 Survey

We conducted a survey about the UniDisplay application with all users that used the application during our deployment phase of eight weeks. In total, 31 users participated and filled in our online questionnaire where we asked them about their thoughts on the application and their experience when using it.

In the following, we describe the questionnaire, the participants, and the results of the survey.

7.4.1 Questionnaire

The questionnaire was initially created and then reviewed in several meetings about the meaning of the questions and the potential output we could get from it. The original version of the questionnaire with all explanations and questions can be found in the appendix of the document (see section A.1).

The questionnaire has four pages with in total 18 questions. As the questionnaire of the online survey described in chapter 5, the questionnaire only allows forward navigation, so participants are not able to change answers given on a prior page. Participants must answer all mandatory questions on a given page before moving on to the next page. They are free to interrupt the survey and complete it at a later time. They may also choose to exit the survey, which deletes all of their given answers. There are in total three optional questions in the questionnaire which offer the entry of free text and enable the participants to include suggestions, ideas and comments about the application, topic, and survey.

Besides the optional questions, most of the other questions use preset answers. Three questions use the five-level Likert scale used in our first online survey and specified in section 5.1.

The participants have to indicate their level of agreement with a given statement in a range from “totally disagree” to “totally agree”. Intermediate steps represent the levels “disagree”, “neutral” and “agree”. The levels are consecutively numbered from one (“totally disagree”) to five (“totally agree”).

In the following, the questions, the preparation and deployment, and the distribution of the questionnaire will be described in detail.

7.4.1.1 Questions

The survey starts with an introduction explaining participants that the survey is part of a diploma thesis. The introduction also explains why participants are getting the link to the survey which is the fact that they posted content to the UniDisplay application.

The first page asks participants to enter demographic data such as gender, age, and profession. They are also asked to enter their Twitter user name to check whether they really fill in their personal questionnaire. Further on, they are asked if they ever interacted with a public display before using our application. A comment field allows them to specify their interaction. We then provide several preset answers such as “curiosity”, “communication with friends”, “communication with others”, “to spread news”, “to present an offer”, and “promotion/advertising” from which the participants can choose all answers that apply to specify their reasons for using the UniDisplay. They also have the possibility to choose “other” as an option and insert their reason manually. Lastly, participants are asked if they stood in front of the display when posting their message and if so, they are also asked if they waited in front of the display until their message was displayed. Both questions could only be answered with “yes” or “no”.

The second page of the questionnaire contains questions about the concrete usage of the UniDisplay application. Participants first have to rate their overall experience with the UniDisplay on a scale from one (poor) to five (brilliant). In the following they have to state if they would use the UniDisplay again and explain their choice in a comment field. Optionally, they have the possibility to enter any comment about the usage of the application or any suggestions for improvement.

The third page of the questionnaire contains questions very similar to those of the online survey asking participants if they expected their message to appear instantly, if the message appeared right in time and what delay would be mostly acceptable for them in a range from one second to “It doesn’t matter for me how long it takes until my message is displayed” (see section 5.1.1).

The last page of the questionnaire covers the topic of misuse of the UniDisplay. Participants are asked if it makes sense that messages get checked before showing them on the public display and what delay would be acceptable knowing that messages will be checked before displaying them (see section section 5.1.1). Optionally, participants could suggest their ideas how to prevent misuse of a public display and can enter any further suggestions, remarks or comments about the application and the survey.

7.4.1.2 Preparation and Deployment

The questionnaire was prepared with Lime Survey and deployed on the Lime Survey Server of the HCI group. The survey was not anonymous and every participant got a personal link to the survey encoding all necessary information.

7.4.1.3 Distribution

The personal link to the questionnaire was sent to the participants via Twitter by mentioning them in a Tweet (see section 6.4.1.3). The link to the questionnaire encoded the tweet ID, the user name and the delay condition which were filled in hidden fields when clicking on the link to the survey. Asking participants for their Twitter name allowed us to control that every person really filled out the questionnaire with their personal link which was necessary to match the filled in questionnaires with the Tweets and the delay conditions.

7.4.2 Participants

In total 31 participants, 29 males and 2 females, completed the survey about the UniDisplay. The average age of the participants was 23.71 (SD: 4.70). The participants were all students or researchers in a technical oriented field such as computer science, software engineering, aerospace engineering, or information technology.

Nine participants specified that they already interacted with public displays before, e.g., with the application Digifieds, with an application that captured pictures, with displays from companies or for advertisement.

7.4.3 Results

In the following, we present the results of our survey with users of the UniDisplay.

7.4.3.1 Usage of the UniDisplay

We asked participants for their reasons for posting a message to the UniDisplay (see Figure 7.18). They were able to select multiple answers out of a list of preset answers. 90.32 % selected curiosity as one of their reasons for posting, followed by communication with friends reaching only 29.03 %. Five participants used the others option to specify their reasons. Four of them mentioned fun as one of their reasons and one participants wanted to be seen. 12.90 % or less of the participants selected each one of the other reasons as there are to spread news, promotion/advertising, communication with others and to present an offer.

In contrast to our expectation that most users would only post messages when being in front of the display, only 64.52 % of the participants posted their first message when being in front

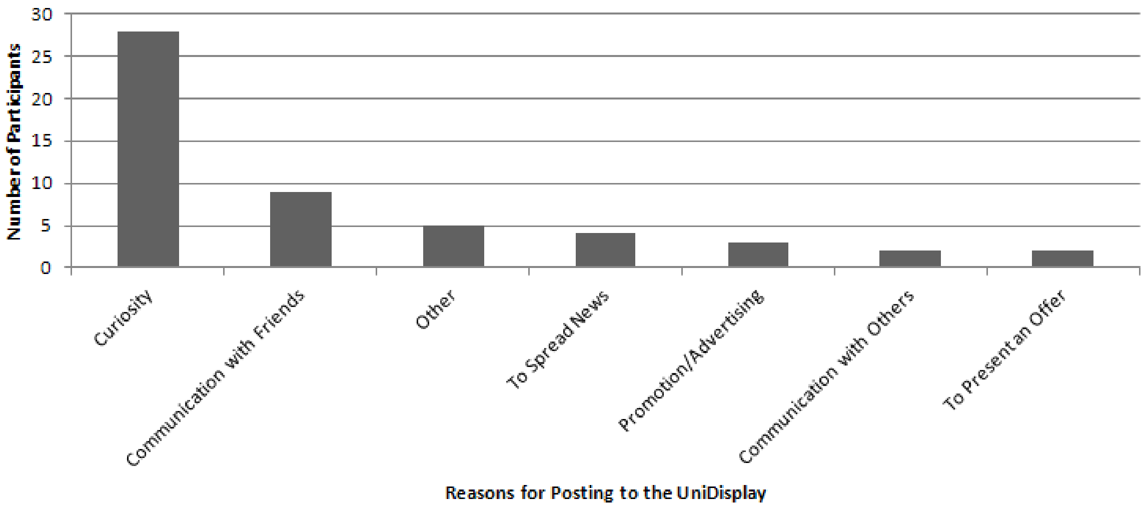


Figure 7.18: Reasons for posting to the UniDisplay mentioned by survey participants who were able to select multiple reasons.

of the display. 25 % of those participants did not even wait in front of the display until their message was displayed.

Participants rated their overall experience using the UniDisplay with an average rating of 4.16 on a five-level scale from poor to brilliant (SD: 0.90). For the 0 seconds condition we got an average of 4.00 (SD: 0.96) compared to an average rating of 4.55 (SD: 0.69) for the 30 seconds condition and 3.83 (SD:0.98) for the 90 seconds condition. Only three participants stated that they would not use the UniDisplay again, the other participants said that they would use it again because it is fun to use and one participant in the 0 seconds condition also mentioned that he would post again because the content appeared instantly. In contrast, one user posting under the 30 seconds condition asked in a further comment field why messages are not displayed instantly.

7.4.4 Delay Times

In total, 54.84 % of the participants partly or totally agree with the statement that they would expect a message to appear instantly. The average rating is 3.35 (SD: 1.56).

With an average agreement of 4.33 (SD: 1.11), participants agreed with the statement that the appearance of their message was right in time. We excluded participants who were not in front of the display when posting because they cannot know when their message appeared.

Asking participants for the delay time they would mostly accept, 70.97 % would only accept delay times up to one minute. Asking the same question again later on specifying that there is a moderation process running, only 35.48 % of participants would still mostly accept a delay

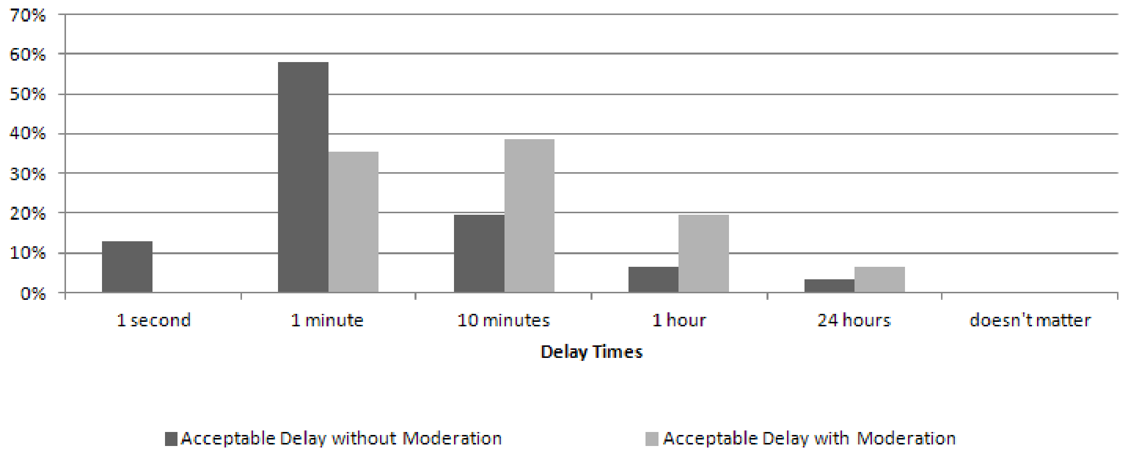


Figure 7.19: Acceptable delay times for messages with and without moderation.

up to one minute. In total, 51.61 % of participants are therefore willing to wait longer knowing that there is an on-going moderation process. Fore more details, see Figure 7.19.

7.4.5 Content Moderation

48.39 % of the participants partly or totally agreed with the statement that it makes sense to check messages before they are displayed to avoid misuse. The average rating is 3.35 (SD: 1.56).

Asking participants for further ideas to prevent misuse, seven participants stated that they would use automatic methods such as a bad word check and image recognition. Three participants would automatically check Twitter for the reputation of the user and three others would block or ban users completely if they misused the display. Further on, two participants suggest a community control and a third participant suggests to use Amazon's Mechanical Turk. Three participants do not think that any methods are necessary because it should be common sense what to post on the display. Two participants would only post the author of a message and think that social pressure will hold people off misusing the display.

7.4.5.1 Comments of Participants

Only four participants added optional comments at the end of the survey. Two participants wanted to have a better user interface for the application. The other two participants wanted either more displays on campus or a bigger display outside the campus displaying the application.

7.5 Interviews

After the deployment phase we conducted interviews with seven users of our application. The interviews were semi-structured, open-ended and covered multiple topics such as the usage of the UniDisplay, authentication mechanisms, display locations, time expectations, offensive content, and implications of the application. The interviews lasted took between 40 minutes and 1 hour depending on the number of ideas and comments of participants. Participants were given 10 € for taking part.

The interviews were recorded with the built-in microphone of a laptop and important parts of the interviews were afterwards transcribed, summarized, and ordered by topic. In the following, each topic that emerged from the interview is discussed in detail. Six interviews were conducted in German. All answers were translated to English by the author of this document.

7.5.1 Participants

We invited all users of our application to take part in our interviews, but in particular tried to choose users from different user group to get a wide range of opinions. We interviewed seven male users. Six of them were students of computer science, software engineering and the international master program infotech and the other one was an employee at university. Participants posted between three and forty-three Tweets to the UniDisplay and participant 7, in contrast to all other participants, was not situated at Stuttgart when he posted content to the UniDisplay and therefore posted remotely.

7.5.2 First Contact with the UniDisplay

All participants told us that they work or attend lectures in one of the building where the displays are situated. When first passing the display, three participants ignored the display and did not look at the content.

> *"At the beginning I didn't really look at the display."* - Participant 3

The others looked at the content but all participants told us that it took a while until they interacted with the display. Five of them were not even able to post anything in the first place because they did not own a Twitter account.

> *"At that point I didn't post anything because I had no Twitter account."* - Participant 7 about the moment when he saw the display for the first time

Four of the participants created a Twitter account later on to be able to post a message on the display whilst one other participant used the Twitter account of his girlfriend.

Two main reasons triggered the interviewees to start looking at the display, post content, and create a Twitter account. One of the reasons was simple curiosity and interest in the functionality of the application. Thus, they posted a first message to find out what it does and how it works.

> *"What's that? Who did that? What is it doing?"* - Participant 5 about his reasons for posting a message

Participants also admitted that they posted their first message just for fun or for testing, so they picked random images or greetings to see what happens.

> *"First of all only for testing."* - Participant 4 about his first post

The other reason for participants to start posting content on the display was to watch other people, in particular friends posting content on the display. When they saw that it worked and that the messages were displayed, they started to use the UniDisplay themselves.

> *"Wow, it worked! Because I didn't really think it would be so fast. [...] It was nice to see something happen."* - Participant 2 describing his reaction when a friend posted his first message

7.5.3 Posting Behavior

Further messages of the participants included advertisements for projects, communication with friends, photos of friends and trips, and interactive posts. Three participants mentioned that they never reacted to posts from other users because *"someone else will answer on it"* or they just had no time to answer. In contrast, participant 7 deliberately posted interactive stuff to see if someone reacts and hoped that a sort of conversation could be developed on the display. Participant 6 mentioned that he changed his posting behavior from just posting for fun and curiosity to a more interactive and reactive way of posting.

In contrast to our expectations before deploying the UniDisplay, most of the participants did not post the content while being in front of the display. They either posted the content being in another room at university and afterwards went to the display to look if their message was displayed or even posted the content on their way to university to see their message when they arrive for their lectures.

> *"I usually post without being in front of the display, because I want to arrive and see how it looks."* - Participant 2

7.5.4 Authentication Mechanisms

The topic of authentication was also discussed during the interview. As already mentioned, five of the participants did not own a Twitter account. One of the participants used the account of his girlfriend, all other four participants created the Twitter account only to be able to post messages on the displays.

> *"I created the Twitter account especially for the display. Prior to that I didn't find it fascinating."* - Participant 7

Talking about other authentication mechanisms, one participants mentioned that he would like to use Facebook or Instagram, but none of the participants was in favor to use the university account because they found it to complicated and normally do not use their university account very often. Another reason for being against the university account was that other persons outside the university would not be able to post messages to the display, e.g., companies or students and researchers from other universities.

> *"I think it is better to have something more public, not bound to the university accounts."* - Participant 2 about authenticating with the university account

Additionally other authentication mechanisms were considered too complex for the application and all participants agreed that the usage of Twitter is very easy and comfortable for posting images or even using hash tags and user references in your messages. Participants did not see the creation of a Twitter account as a barrier for the application and also stated that the authentication mechanism has to be as easy as possible to attract a large number of users.

> *"The less you have to do, the more it will be used."* - Participant 1 about authentication

7.5.5 Time Expectations

The interviewees had different time expectations. Participant 1 hoped that messages will be displayed very fast and even discussed the topic with his friends. Participant 3 did not expect his Tweets to be displayed instantly, in contrast to Participant 6 who expected a delay time around ten seconds. Only one participant noticed that some message were delayed, because he observed other users when posting. All other participants expect Participant 7 who was not able to look at the display stated that the messages appeared very fast.

7.5.6 Influence of Display Locations

The interviewees knew that we deployed the content on several displays. Asking them if they would post the same content knowing that there are several displays across the campus, four participants stated that they would nevertheless post the same messages again even if more people could see them. Only one of the participants declined that he would not post the same content if there are more viewers.

Participant 7 was not even at Stuttgart when posting content to the UniDisplay, but he saw the UniDisplay on a visit at the university. He had no possibility to check whether his messages were really displayed, but he posted messages anyway trusting in the system trying to follow the posts on Twitter by searching for our twitter user.

> *"I had no idea if the messages really appeared on the display."* - Participant 7

7.5.7 Offensive Content and Content Moderation

All participants agreed that there was no offensive content on the display and that they would not have deleted any of the posts they have seen so far. Anyway, they admitted that they saw posts that were uninformative or just spam.

> *"So far, from what I've seen, the content is fine. It's not offensive."* - Participant 2

Participants also agreed with each other that they would not post any offensive content. They had different reasons for their opinion, e.g., that the account name is visible, that they use their Twitter account properly or that they do not know what will be stored by the application because it could be possible to store IP addresses which would lead to the user posting offensive content.

> *"Of course my account name is visible [...], therefore I wouldn't post any offensive content."* - Participant 3

Additionally, participants even told us that they would generally not delete content and just leave the content as it is.

> *"I would generally delete quite little."* - Participant 5

They were also not bothered by any posts and were relaxed regarding the content on the display, because they assumend persons who do not like the content not to look at it.

> *"Actually the posts from other users don't face me and I don't have to look at them."* - Participant 4

Nevertheless, they stated that it would be nice to be able to request the deletion of a specific post.

> *"Maybe if someone complains about some posts it would be nice to be able to send a request for its removal."* - Participant 2 about the deletion of posts

In total, participants were very liberal in the context of what content should be posted to the UniDisplay.

> *"Everyone could post whatever he wants to."* - Participant 5

7.5.8 Implications for the Real World

Another topic during the interviews were the implications for the real word. All participants besides participant 7, who was not situated at Stuttgart when he posted, specified that they look at the display when they pass it and that they often discuss the content with friends or colleagues. Additionally, the posts of the application were able to, e.g., influence their plans for lunch when they saw photos of food. Three participants also stated that they were addressed by others to talk about their posts, while three other participants stated that they addressed other users to talk to them about their content.

> *"I saw a photo from you on the UniDisplay."* - Participant 1 about another person addressing him

Additionally one participant told us that he was very interested in one of the posts and started an Internet search to find out what it is all about.

> *"I found that interesting and did some research on the Internet to find out what it was."* - Participant 1 about a specific post

7.5.9 Opinions about the UniDisplay

Asking participants for their thoughts on the application, all participants liked the idea of the application and the concept behind it. It was described as *"a kind of modern black board"* which is fun to use.

> *"We liked the idea very much. It's fun."* - Participant 1 about the application

Participant 7 additionally mentioned the drawback that you never know how long your Tweet will be shown on the display and how many people you will reach with it.

In the end, four participants told us that the UniDisplay could stay as it is.

> *"I like it as it is. It could stay like it is."* - Participant 3

In general, all participants wanted the application to stay because they found it still interesting. Three participants also had concrete suggestions for improving the application which were mostly more interactions by being able to virtually like or vote for posts, a new user interface, conducting special events on the display (such as topics for images), show YouTube videos, or more information on when Tweets will appear.

7.6 Summary and Discussion

In this chapter, we analyzed the data collected during the Deployment of the UniDisplay.

The data analysis showed that most of the users posted images. The distribution of posts increased at the beginning and then decreased to a constant level, possibly because users tried out how the application works and then only used it for a certain purpose. We also discovered an effect of the moderation delay on the number of posts. As the moderation delay increases, the number of posts decreases. This could be explained by a digital honey pot effect which happens when new content items appear on the display. We observed that it is more likely that other users post a message when new content items appeared. When the delay increases, the trigger time to post a new message also increases. Additionally, we found that the delay condition of a first post does not influence further posting behavior. In combination of what the observations and the interviews revealed, users were often watching others posting messages to the UniDisplay before posting a message themselves. They tried to learn how the system works and when they found out that messages are displayed relatively fast after posting it, the delay was not crucial anymore.

The observations clearly showed that informing users is one of the most important tasks to address. Even if users observed the system and therefore knew that the messages will be displayed after a certain delay, other users that did not observe the system were confused. They tend to think that the display is either broken or reposted their content because it was not displayed. Additionally, if the functionality of the application is not clear or just not provided as expected, users will start complaining about the display and the application. It is therefore very important that the display works without major breakdowns such as losing the Internet connection or to again inform users about such breakdowns right away on the display, e.g. by showing a message that it lost its Internet connection. Unintended usage cannot be totally prevented, but does not need to be inappropriate or offensive.

The observations also showed that users individually prefer different devices for posting their content to the display. Therefore posting with different devices and also different input methods should be provided by a public display application.

In our content analysis, we showed that only a small amount of offensive or inappropriate content was posted to the display. It is nevertheless problematic that opinions about what is inappropriate or offensive vary greatly. Developing guidelines for a display application including all stakeholders could help to get to a common sense about the appropriateness of content. A few times, we also were able to discover self-regulation when users complained about others on the display or showed that they liked or disliked the content by answering to former posts.

The results of our survey indicate that one third of the users do not stand in front of the display when posting a message. This has to be considered in the design of future display applications, because it has impacts on how applications are used. Additionally, 51.61 % of the survey participants are willing to wait longer if informed about the moderation process. This is another indicator that informing users about content moderation is important.

The interviews revealed that many of users do not post directly at the beginning when looking at a public display. They first observe other users in front of the display before they start posting themselves. Participants mentioned the reasons for not posting in front of the display; they posted before arriving at university to see their messages when they arrive or from their laptop being in a seating area, looking at the display later on when they pass it. Additionally, we discovered that our public display application also had implications in the reald world, because users talked with each other about the content.

The insights gained during the deplyoment of the UniDisplay could be used to develop a concept that helps preventing conflicts between different stakeholders.

8 Preventing Conflicts between Stakeholders with the Help of the PD App Store

We developed the PD App Store (see chapter 4) as a distribution medium for public display applications. Besides being a distribution medium, an app store has great potential to help prevent or moderate conflicts between stakeholders. The online survey and the deployment of the UniDisplay helped us understand problems and impacts related to content moderation on public displays and highly influenced this part of the work.

In the following, we discuss how an app store can prevent conflicts along the topics described in chapter 3.

8.1 Purpose and Placement

The PD App Store cannot prevent conflicts about the placement of a public display, but conflicts about the purpose of the display could possibly be prevented.

Display Owners vs. Space Owners: Display owners and space owners who disagree about the purpose of the display could use the playlist feature of the app store to schedule multiple applications to a display. The scheduling functionality of the app store could even be extended in the future to support more options. If the time of showing applications chosen by the display owners and applications chosen by the space owners is equal, it should be much easier for them to find a compromise.

Display Owners vs. Application Developers: The app store allows application developers to specify parameters for an application. Thus, display owners can configure an application to their needs. Additionally, a large number of available applications and scheduling more than one application to a display via a playlist allow the display owners to choose applications matching their purpose.

Display and Space Owners vs. Viewers: The app store could also prevent conflicts between display or space owners and viewers. With an extended scheduling interface and applications which allow the personalization of content, it is much easier for display owners and space owners to provide applications that are accepted by users and viewers, because they also match their interests.

8.2 Content

Appropriateness of content is probably the main source of conflicts between stakeholders of public display app stores. Including content moderation in the app store can help prevent these conflicts, because content moderation is no longer a problem of each application. Otherwise a display owner has to moderate content separately for every application, which is time-intensive and maybe also includes switching between different tools. A general moderation framework integrated in the app store can provide standards about how to moderate content, how to inform users and how to inform moderators about moderations still due. A general framework helps display owners to accelerate and coordinate moderation and also supports application developers who do not have to implement a content moderation strategy on their own. In addition, a general framework is much more flexible and allows display owners to adapt content moderation and choose the strategy that matches best to their interests or even switch their strategy due to changing interests.

8.2.1 Including Content Moderation in the PD App Store

Moderation could be included in the app store by adding a moderation API. An application can then call the API with four required parameters. The first parameter is a unique ID, which is a combination of the display ID and the application ID. The application can read the ID from the XML file used to configure a display. The second parameter is the ID of the content item which does not have any meaning for the app store, but allows to identify a content item later on in the application. The third parameter has to specify the type of the content item, e.g., text, image, a URL to an image or video, depending on the types supported by the app store. The fourth parameter has to contain the content item in the format matching the content type.

8.2.1.1 Moderation Strategies

The app store could support different moderation strategies: automatic moderation, manual pre-moderation, manual post-moderation, and reactive moderation. These strategies could even be combined and distributed to one or more moderators.

Automatic Moderation For the automatic moderation, a bad word check and an image recognition algorithm could be implemented. If there are different approaches on how to do this, the app store could also support multiple approaches. A display owner could enable this moderation strategy and can also be billed for using it. Users could also request new features, e.g., new words that should be included in the bad word check. In addition, users could potentially customize the filters and add new words by paying a higher fee.

Manual Pre-Moderation The display application has to send all content items to the app store before showing them on the display. All content items could then be shown in a moderation view, where the display owner could decide whether to approve or decline them. The decision could be communicated to the application in the response to the request or via a callback method that has to be implemented by the application.

Manual Post-Moderation To use a manual post-moderation strategy, the application also has to send all content items to the app store, but the content items can already be displayed. All content items are again shown in a moderation view, where the display owner can delete content items. The display application has to provide an API or a callback method to delete a content item by its ID.

Reactive Moderation Reactive moderation would require the application to send each content item to the app store that users complained about. The display owner can use a moderation view to delete the content item if it is offensive or inappropriate.

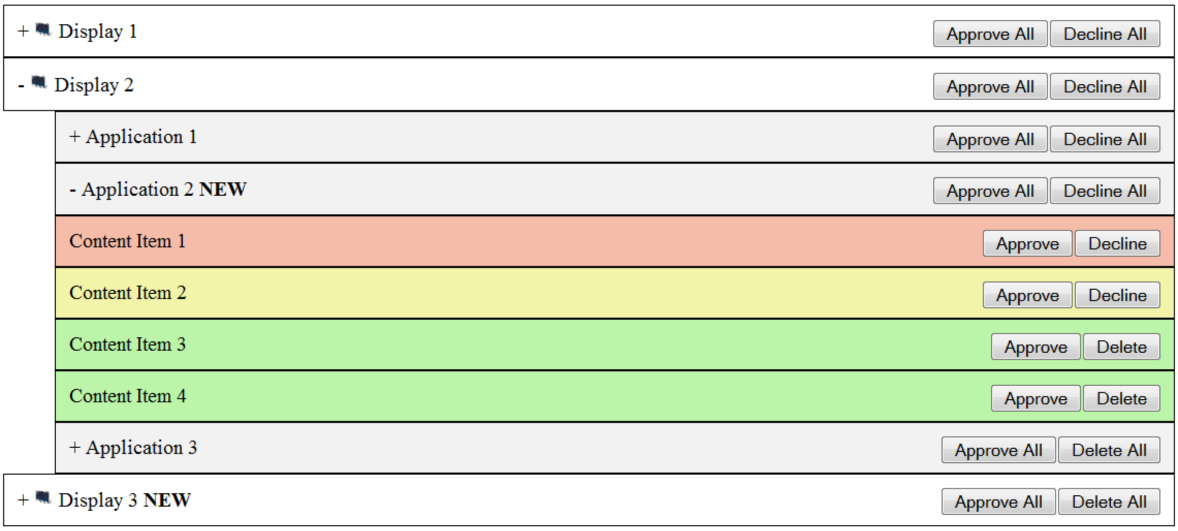
8.2.1.2 Moderators

To distribute the work and prevent conflicts between display and space owners, a display owner should be able to add moderators for specific displays and applications. A moderator can be assigned to one display including all applications on the display, to one application on all displays or even to particular applications on particular displays. The display owner therefore has to link the app store account of the moderator in the moderation view. The moderator could then be informed by sending an e-mail.

8.2.1.3 Prototype for the User Interface

The user interface could provide a moderation view for all moderators showing only the displays and applications that they moderate. Figure 8.1 shows a prototype of this view. Color codes could be used to distinguish between content items failed to pass a certain moderation process. Additionally, the number of moderators per display or application could also be shown and the display owner should be able to add moderators via a dialog.

Another view that has to be supported is a table listing all moderators. An example view is shown in Figure 8.2, where display owners could remove and reassign moderators to applications and displays.



+ Display 1

- Display 2

  + Application 1

  - Application 2 **NEW**

    Content Item 1

    Content Item 2

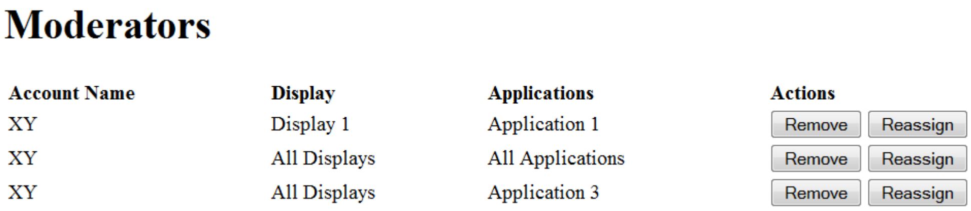
    Content Item 3

    Content Item 4

  + Application 3

+ Display 3 **NEW**

Figure 8.1: Prototype of a moderation view included in the app store. Content items can be approved or declined each by one, per application or per display. Red colored content items were removed because users of the application marked them as inappropriate or complained about it, yellow content items did not pass the automatic moderation and green content items are available for post-moderation and are already shown on the display.



Moderators

Account Name	Display	Applications	Actions
XY	Display 1	Application 1	Remove Reassign
XY	All Displays	All Applications	Remove Reassign
XY	All Displays	Application 3	Remove Reassign

Figure 8.2: Prototype of an overview on all moderators for the display owner. The display owner can remove or reassign moderators.

8.2.1.4 Notification Features

The app store could also provide notification features. Display owners and moderators should be able to receive e-mail notifications if new content has to be moderated. Notification settings should allow to customize the number and reasons for notifications per moderation strategy and for different applications and displays.

The app store could also help to send notifications to users about the reason that a post of them was declined or deleted. Therefore, content guidelines could be supported by the app store. Display owners could then write guidelines for appropriate content and link them to displays or applications. These guidelines could be automatically sent to the API of the display application when a content item was declined or deleted. Alternatively, when declining or deleting posts, moderators could insert a reason into a text field which could also be sent to the application. The application can then contact the user and provide this information.

Another interesting point is about how to inform users when their messages will be shown on the display. In particular, a manual pre-moderation strategy adds a delay to content items. The deployment of the UniDisplay showed, that also short delays of 90 seconds can confuse users. The app store could therefore possibly store the minimum time, the average time and the maximum time from sending a content item to moderating the same content item. These numbers could be sent to the application when sending a content item for pre-moderation. The application could then also contact the users to inform them about the delay and even give an estimation about how long it will take until the content item is displayed.

8.2.2 Monitoring Display Content

Another approach would be to provide a live view or a current screenshot of all displays for the display owner. This could either be shown on every display details page or even on one page where the display owner has an overview about the content on all displays.

8.3 Trust

The PD App Store tries to build trust between the related stakeholders by offering a rating system. Furthermore, the moderation framework presented in section 8.2 could also help to build trust.

Display Owners vs. Application Developers: Trust between display owners and developers emerges through the rating system applied in the app store. Display owners can rate the applications they purchased and therefore give other display owners recommendations for good applications. Additionally, they could write optional reviews, which also help other display owners to decide whether or not to buy and trust an application and the associated developer. If display owners then trust one application of a developer, they might decide to use another application of the same developer, which in turn enhances trust between the two stakeholders. A drawback of the rating system is that no ratings are available for new applications. To get display owners to buy such applications, developers should try to specify the content of their application as good as possible in the description and suggest suitable locations for the application. The app store could encourage application developers to do that by creating extra fields to fill in this information when creating a new application in the app store.

Display Owners vs. Space Owners: Regarding the moderation framework we present in section 8.2, trust between display owners and space owners is much easier to build if display owners invite space owners as moderators of the display. Space owners are then able to influence the content on the display and in particular delete inappropriate content. Display owners and space owners could also agree about guidelines on how to moderate a certain display depending on its location and viewers.

Display and Space Owners vs. Users: Trust between display owners or space owners and users cannot be directly built through the app store. Nevertheless, the app store could support trust building. Informing users about the deletion of their content and the reason for the deletion may lead to a common sense about the appropriateness of content for a particular display. If also guidelines for content are available, they could be included in the information which helps users understand the expectations of display and space owners regarding appropriateness.

8.4 Summary and Discussion

In this chapter, we presented solutions to prevent conflicts between stakeholders of public display app stores. Conflicts about the purpose of a display could be prevented by the scheduling functionality of the app store which allows the adjustment of the content to all stakeholders. Additionally, display owners can configure the applications for their needs.

Conflicts about the appropriateness of static and user-generated content on public displays could possibly be prevented by a moderation framework included in the app store. A moderation framework can standardize and simplify the process of content moderation with a standardized way of checking and approving content items across applications. Adding notifications for users about content guidelines and the delay of a content item could add new values to applications showing user-generated content.

The rating system of the app store tries to build trust between display owners and application developers by allowing display owners to rate and review applications. Trust between display owners and space owners could be encouraged with a moderation framework enabling display owners to include space owners in the moderation process, which could improve the relationship between them. Trust between display or space owners and users could be supported by the app store when communicating guidelines for the content of displays and informing users about the reasons to delete their content.

Technical aspects were not covered in this theoretical concept, but it is clear that topics as authentication and security are also important and have to be addressed when designing the moderation framework.

9 Conclusions

Although app stores are a well-known concept and widely used by mobile phone owners, developing an app store for public displays needs additional consideration. Involved stakeholders such as display owners, space owners, application developers, users, and viewers are likely to get in conflict with each other. We presented three factors that influence the level of conflicts: the number of display and space owners, the number of viewers and users, and the location of a public display. Considering the conflicts, we developed the PD App Store in cooperation with Lancaster University. The PD App Store supports application developers by providing a distribution channel for their applications and also display owners who could purchase applications that they want to show on their displays.

Previous work only seldom addressed content moderation when developing applications for public displays, although this is a crucial topic for display owners and space owners.

Besides developing the front end of the PD App Store, we conducted an online survey about public displays to gain more insight into the users' view of content moderation on public displays. We aimed at understanding the expectations of potential users. Regarding those expectations of survey participants, we furthermore developed the UniDisplay, an application showing user-generated content on public displays via Twitter. We applied artificial delays of 0, 30, and 90 seconds to the messages of the users, simulating content moderation, without informing them about the delay. Aiming to understand the impacts of the choice of a moderation strategy which includes particular delays, we analyzed content, survey data, and interview data collected during the deployment of the application. Finally, we developed a theoretical concept for including content moderation in public display app stores, applying what we learned from the online survey and the deployment of the UniDisplay.

We found authentication and privacy to be topics which have to be addressed when developing applications for public displays. In addition, the choice of the moderation strategy causing a possible delay of content items has to be carefully considered. Survey participants preferred a reactive moderation strategy with users and space owners as moderators. This finding meets their expectation that a message would be displayed instantly. A delay of 90 seconds applied to the messages during the deployment of the UniDisplay already confused users. Nevertheless, the delay was not a strong factor for the posting behavior after finding out how the application works by observing other users. Besides the impacts of the delay, a public display application should also address the information demand of the users. Survey participants were willing to accept longer delays when knowing that a moderation process takes place.

Public display applications and displays have to be as robust as possible or inform users about breakdowns to avoid upsetting users. This could easily happen if displays lose their Internet connection or do not provide the functionality that users presume them to provide. Another effect that has to be considered is that people post content when not being in front of the display. Application developers have to think about the implications when developing an application and make their decision clear for display owners.

During our deployment and analysis, we found that inappropriate content is not always a problem when displaying user-generated content on public displays. The percentage of inappropriate or offensive content on the UniDisplay was below 1.5 %. Guidelines about the appropriateness of content could additionally help to establish a common sense and boost self-regulation.

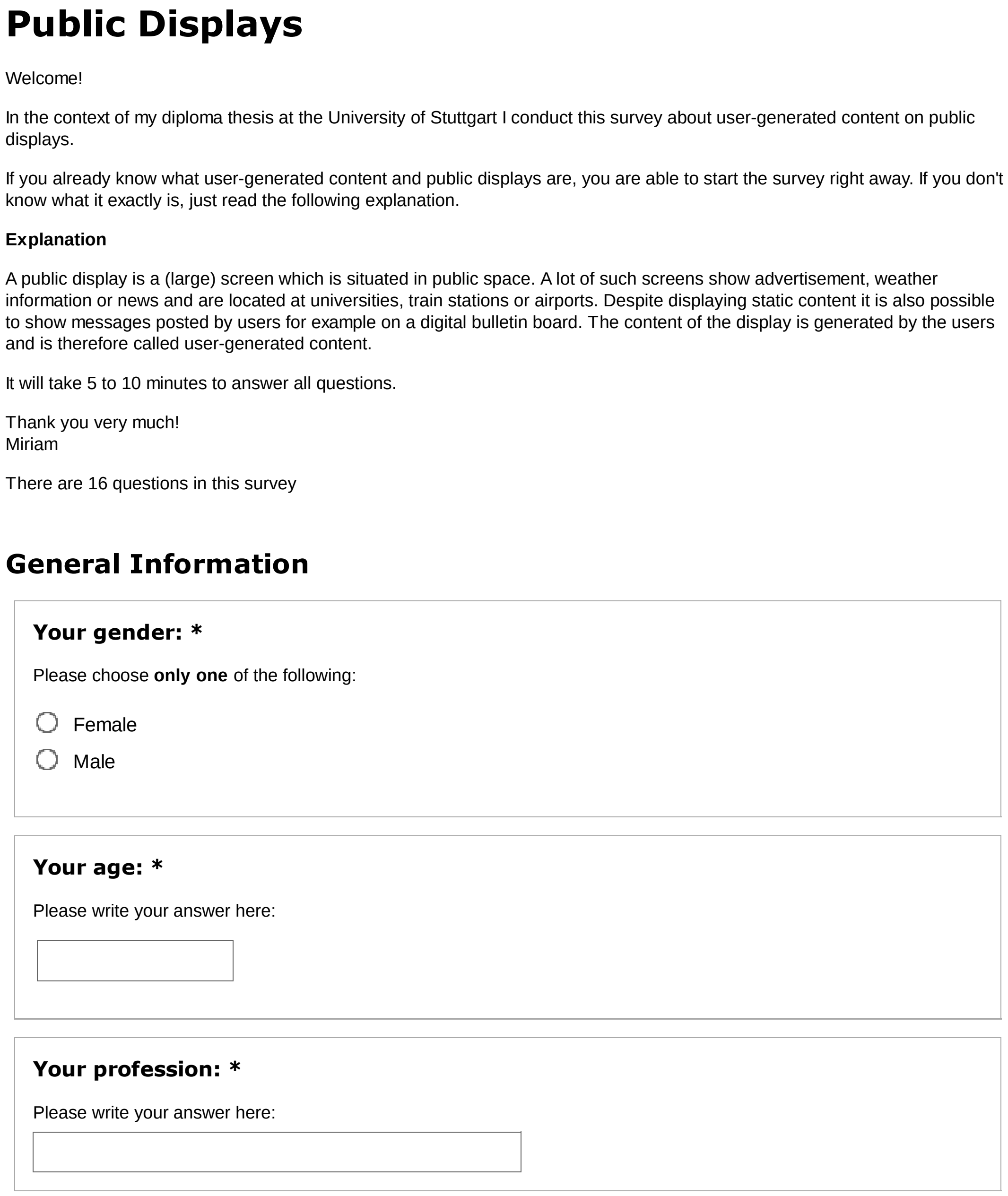
An interesting finding of our interviews with users of the UniDisplay application is that the UniDisplay influenced their real world conversations and sometimes even their decisions for lunch. It is the aim of display owners and space owners to promote their products or locations. If a public display application showing user-generated content leads to users talking about the display and the content of the display, animating other people to use it, or even decisions of them getting affected from the content, advertising will be much more effective.

In the long term, the PD App Store could help to prevent conflicts between stakeholders with its rating and review system, an extended scheduling functionality, the possible configuration of applications and an included moderation framework. A standardized and consistent way of moderating content across applications could make user-generated content more attractive for display and space owners. Display owners additionally would have the possibility to switch between moderation strategies or use a combination of different strategies. Notification features could fulfill the users' information demand by estimating the delay caused by moderation. In summary, the PD App Store could reduce conflicts and build trust between stakeholders.

Future work should investigate different strategies to inform users about a moderation process to determine the optimal strategy. Informing users could, e.g., be done by putting up a poster next to the display, showing messages on the display, or contacting users via e-mail. Furthermore, the digital honey pot effect we observed, could also be further investigated. Rotating the content on a public display may help to simulate new content appearing to create an artificial digital honey pot effect, which will possibly attract more users and viewers to a public display. Future work should also include the development of further scheduling, personalization, and moderation features for public display app stores. The moderation framework should be technically designed, developed and evaluated with display and space owners. Evaluating the usage of the app store itself and the relationships between stakeholders using the app store should also sever as a starting point for measuring the amount of trust created between stakeholders by providing public display app stores. Finally, a more sophisticated classification concept for moderation strategies not only analyzing the impacts of the delay and considering all stakeholders could be developed to support future work in all fields.

A Appendix

A.1 Surveys



Public Displays

Welcome!

In the context of my diploma thesis at the University of Stuttgart I conduct this survey about user-generated content on public displays.

If you already know what user-generated content and public displays are, you are able to start the survey right away. If you don't know what it exactly is, just read the following explanation.

Explanation

A public display is a (large) screen which is situated in public space. A lot of such screens show advertisement, weather information or news and are located at universities, train stations or airports. Despite displaying static content it is also possible to show messages posted by users for example on a digital bulletin board. The content of the display is generated by the users and is therefore called user-generated content.

It will take 5 to 10 minutes to answer all questions.

Thank you very much!
Miriam

There are 16 questions in this survey

General Information

Your gender: *

Please choose **only one** of the following:

○ Female

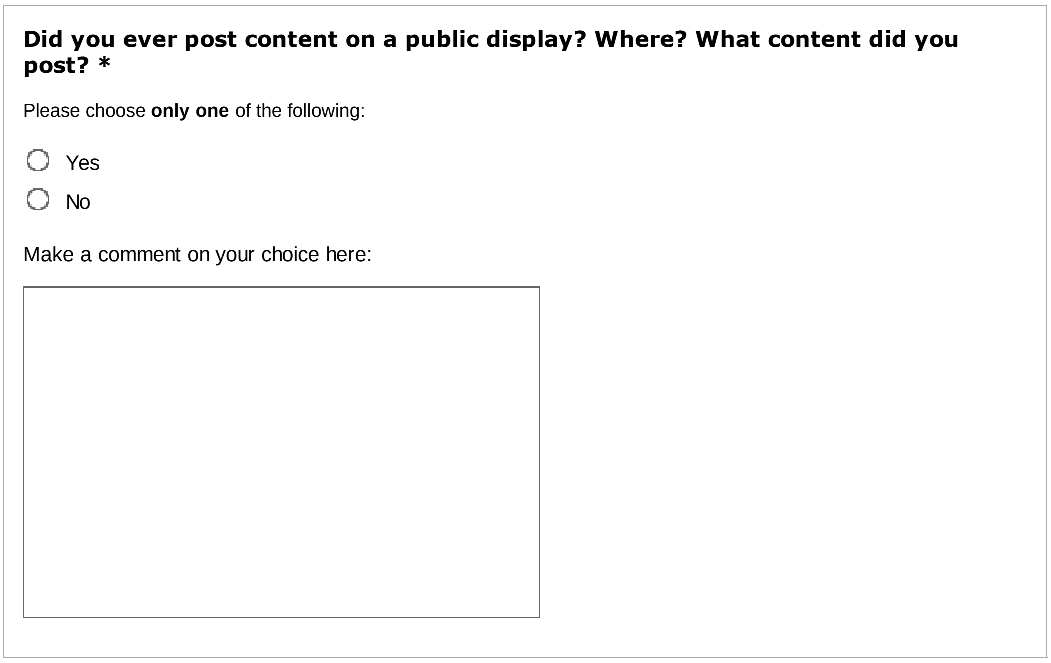
○ Male

Your age: *

Please write your answer here:

Your profession: *

Please write your answer here:



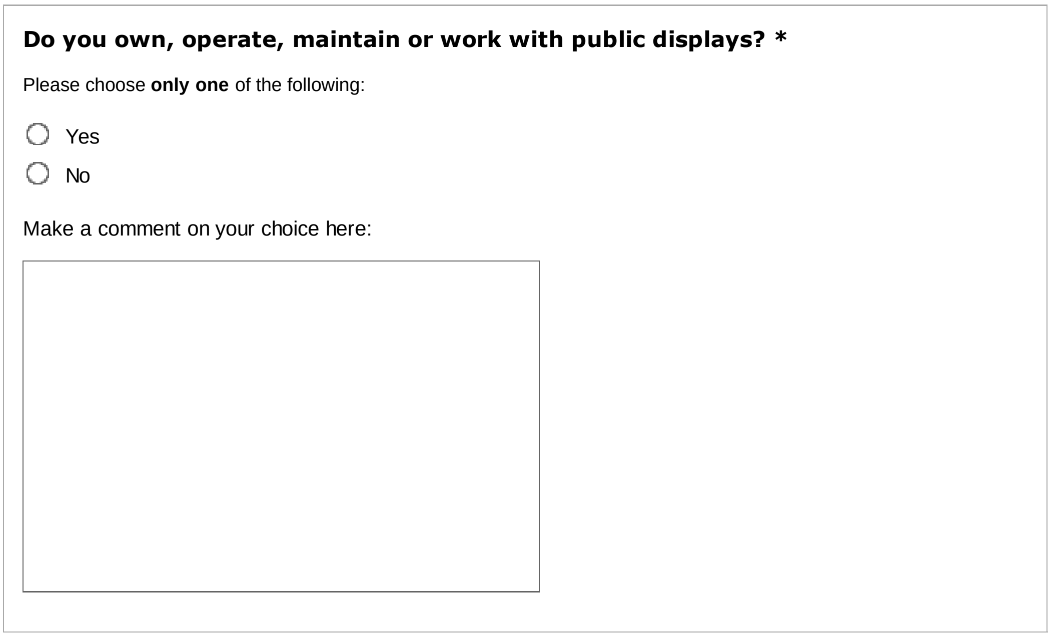
Did you ever post content on a public display? Where? What content did you post? *

Please choose **only one** of the following:

○ Yes

○ No

Make a comment on your choice here:



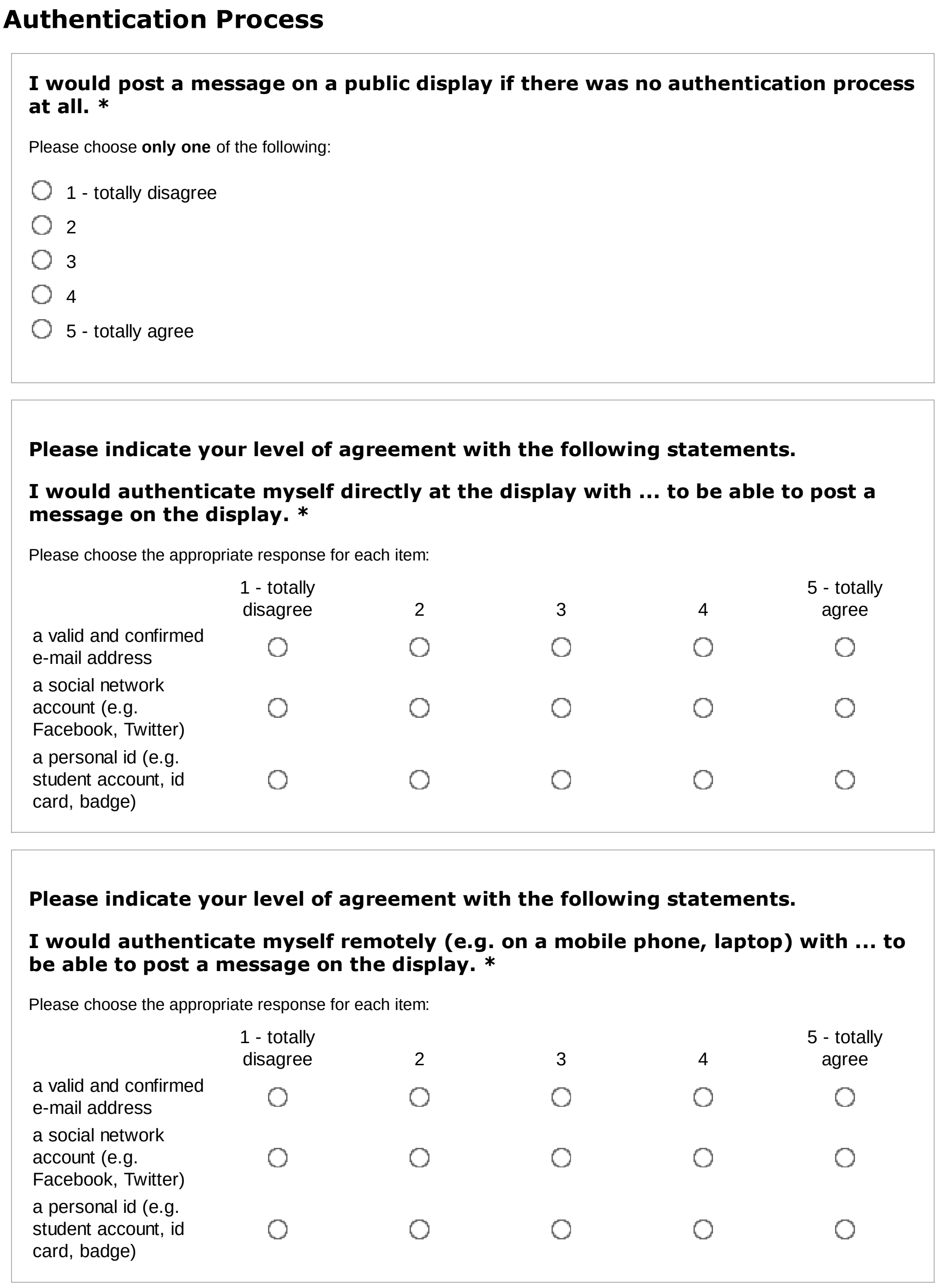
Do you own, operate, maintain or work with public displays? *

Please choose **only one** of the following:

○ Yes

○ No

Make a comment on your choice here:



Authentication Process

I would post a message on a public display if there was no authentication process at all. *

Please choose **only one** of the following:

○ 1 - totally disagree
○ 2
○ 3
○ 4
○ 5 - totally agree

Please indicate your level of agreement with the following statements.

I would authenticate myself directly at the display with ... to be able to post a message on the display. *

Please choose the appropriate response for each item:

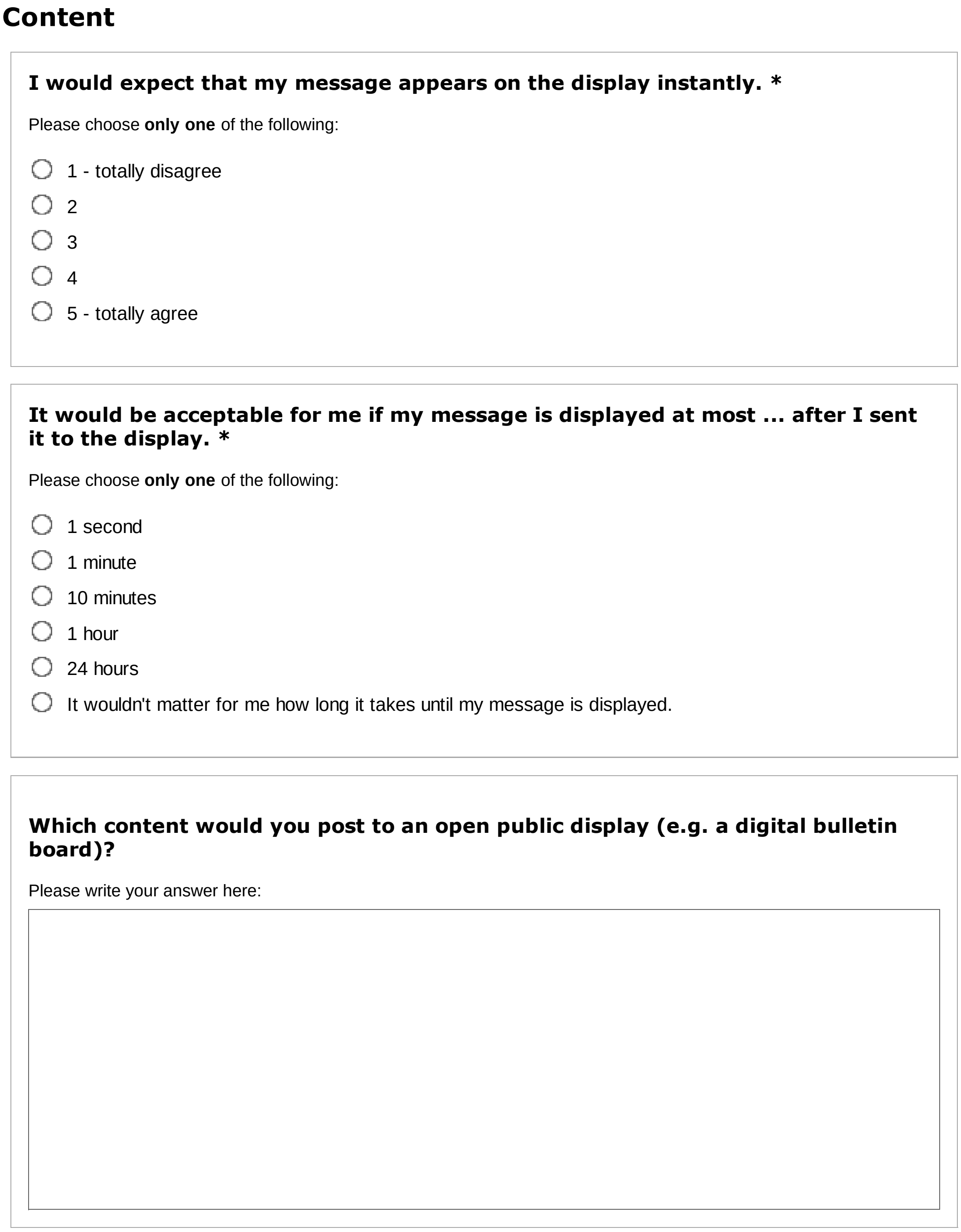
	1 - totally disagree	2	3	4	5 - totally agree
a valid and confirmed e-mail address	○	○	○	○	○
a social network account (e.g. Facebook, Twitter)	○	○	○	○	○
a personal id (e.g. student account, id card, badge)	○	○	○	○	○

Please indicate your level of agreement with the following statements.

I would authenticate myself remotely (e.g. on a mobile phone, laptop) with ... to be able to post a message on the display. *

Please choose the appropriate response for each item:

	1 - totally disagree	2	3	4	5 - totally agree
a valid and confirmed e-mail address	○	○	○	○	○
a social network account (e.g. Facebook, Twitter)	○	○	○	○	○
a personal id (e.g. student account, id card, badge)	○	○	○	○	○



Content

I would expect that my message appears on the display instantly. *

Please choose **only one** of the following:

○ 1 - totally disagree
○ 2
○ 3
○ 4
○ 5 - totally agree

It would be acceptable for me if my message is displayed at most ... after I sent it to the display. *

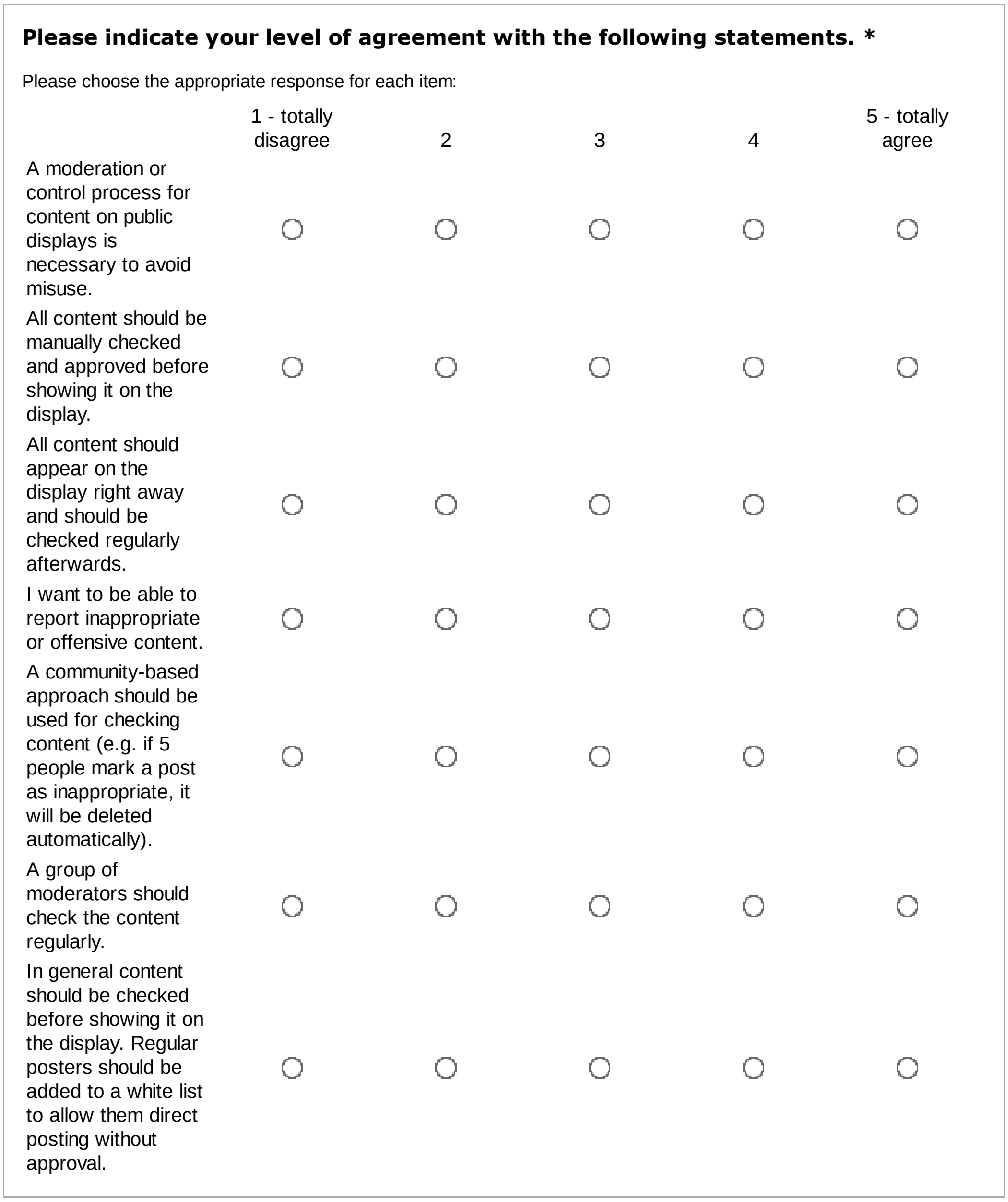
Please choose **only one** of the following:

○ 1 second
○ 1 minute
○ 10 minutes
○ 1 hour
○ 24 hours
○ It wouldn't matter for me how long it takes until my message is displayed.

Which content would you post to an open public display (e.g. a digital bulletin board)?

Please write your answer here:

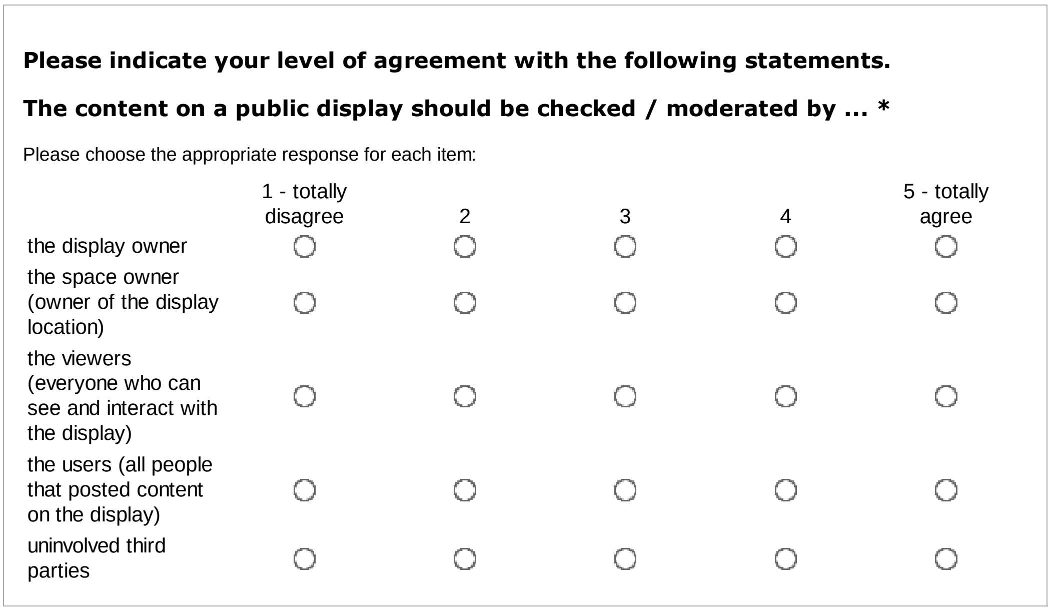
Content Moderation



Please indicate your level of agreement with the following statements. *

Please choose the appropriate response for each item:

	1 - totally disagree	2	3	4	5 - totally agree
A moderation or control process for content on public displays is necessary to avoid misuse.	○	○	○	○	○
All content should be manually checked and approved before showing it on the display.	○	○	○	○	○
All content should appear on the display right away and should be checked regularly afterwards.	○	○	○	○	○
I want to be able to report inappropriate or offensive content.	○	○	○	○	○
A community-based approach should be used for checking content (e.g. if 5 people mark a post as inappropriate, it will be deleted automatically).	○	○	○	○	○
A group of moderators should check the content regularly.	○	○	○	○	○
In general content should be checked before showing it on the display. Regular posters should be added to a white list to allow them direct posting without approval.	○	○	○	○	○

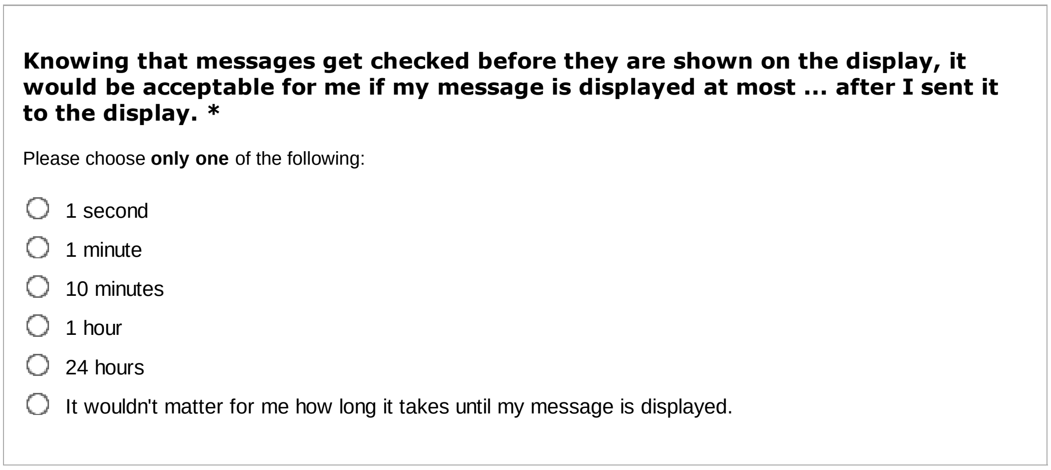


Please indicate your level of agreement with the following statements.

The content on a public display should be checked / moderated by ... *

Please choose the appropriate response for each item:

	1 - totally disagree	2	3	4	5 - totally agree
the display owner	○	○	○	○	○
the space owner (owner of the display location)	○	○	○	○	○
the viewers (everyone who can see and interact with the display)	○	○	○	○	○
the users (all people that posted content on the display)	○	○	○	○	○
uninvolved third parties	○	○	○	○	○



Knowing that messages get checked before they are shown on the display, it would be acceptable for me if my message is displayed at most ... after I sent it to the display. *

Please choose **only one** of the following:

○ 1 second

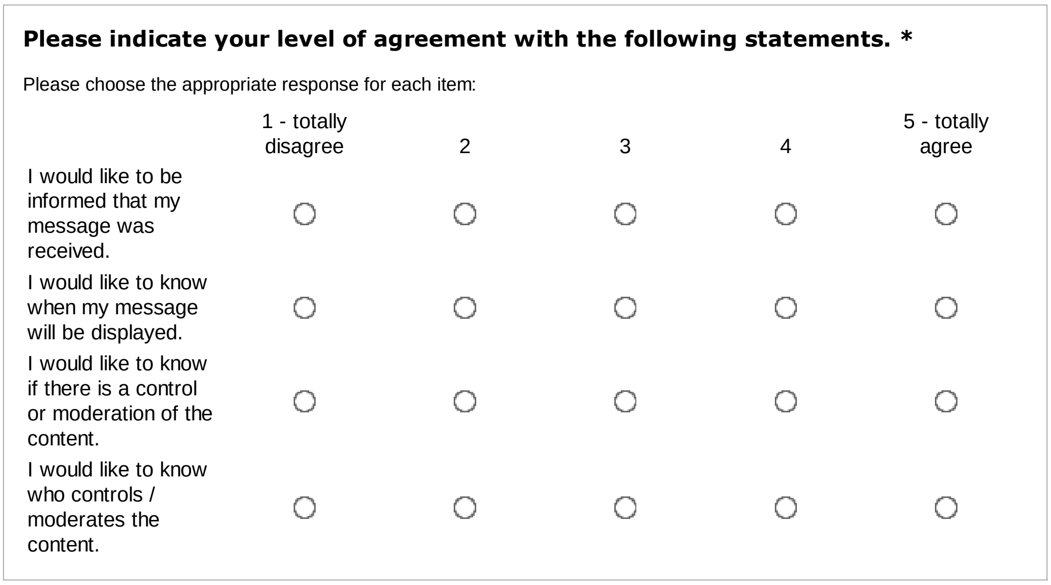
○ 1 minute

○ 10 minutes

○ 1 hour

○ 24 hours

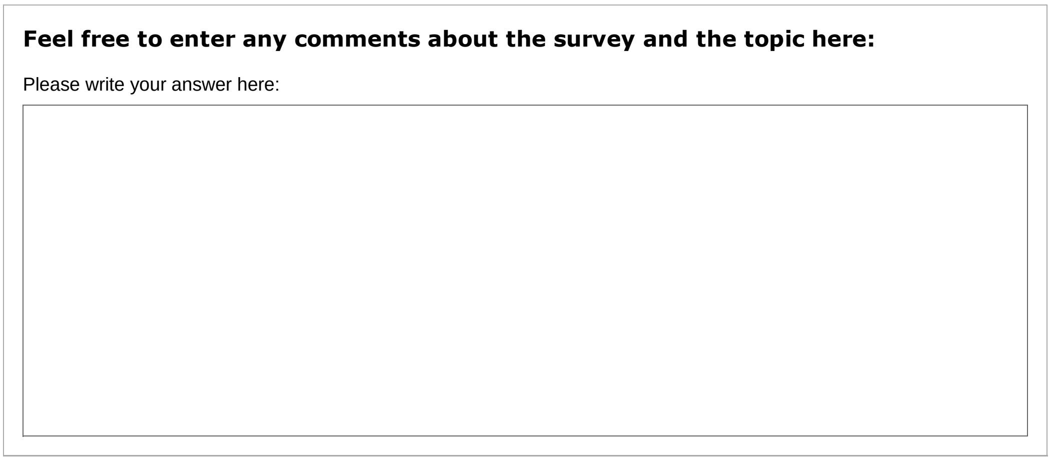
○ It wouldn't matter for me how long it takes until my message is displayed.



Please indicate your level of agreement with the following statements. *

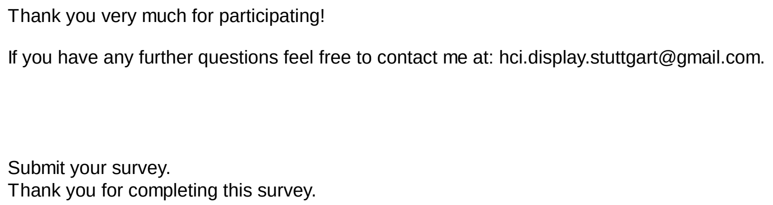
Please choose the appropriate response for each item:

	1 - totally disagree	2	3	4	5 - totally agree
I would like to be informed that my message was received.	○	○	○	○	○
I would like to know when my message will be displayed.	○	○	○	○	○
I would like to know if there is a control or moderation of the content.	○	○	○	○	○
I would like to know who controls / moderates the content.	○	○	○	○	○



Feel free to enter any comments about the survey and the topic here:

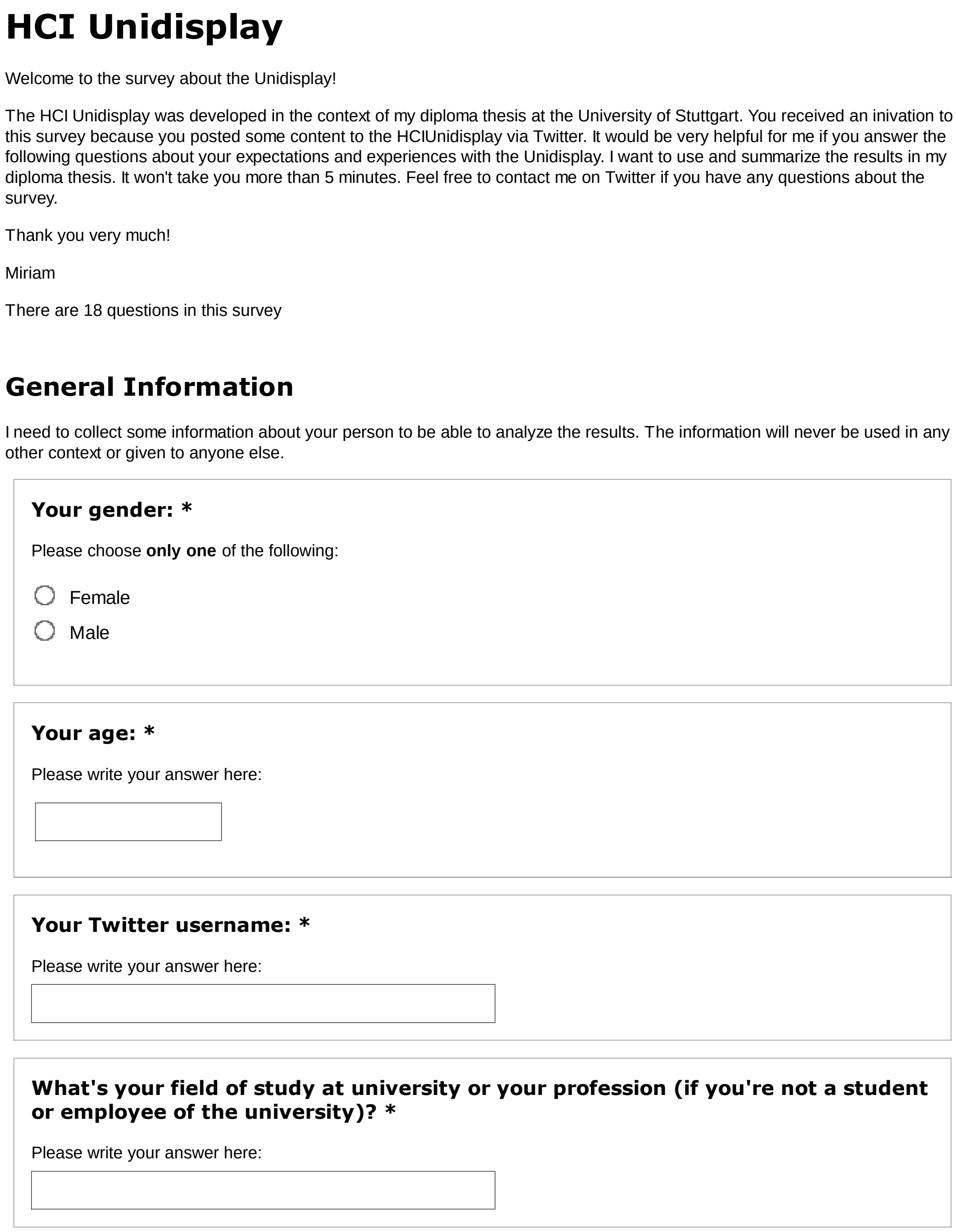
Please write your answer here:



Thank you very much for participating!

If you have any further questions feel free to contact me at: hci.display.stuttgart@gmail.com.

Submit your survey.
Thank you for completing this survey.



HCI Unidisplay

Welcome to the survey about the Unidisplay!

The HCI Unidisplay was developed in the context of my diploma thesis at the University of Stuttgart. You received an inivation to this survey because you posted some content to the HCIUnidisplay via Twitter. It would be very helpful for me if you answer the following questions about your expectations and experiences with the Unidisplay. I want to use and summarize the results in my diploma thesis. It won't take you more than 5 minutes. Feel free to contact me on Twitter if you have any questions about the survey.

Thank you very much!

Miriam

There are 18 questions in this survey

General Information

I need to collect some information about your person to be able to analyze the results. The information will never be used in any other context or given to anyone else.

Your gender: *

Please choose **only one** of the following:

○ Female

○ Male

Your age: *

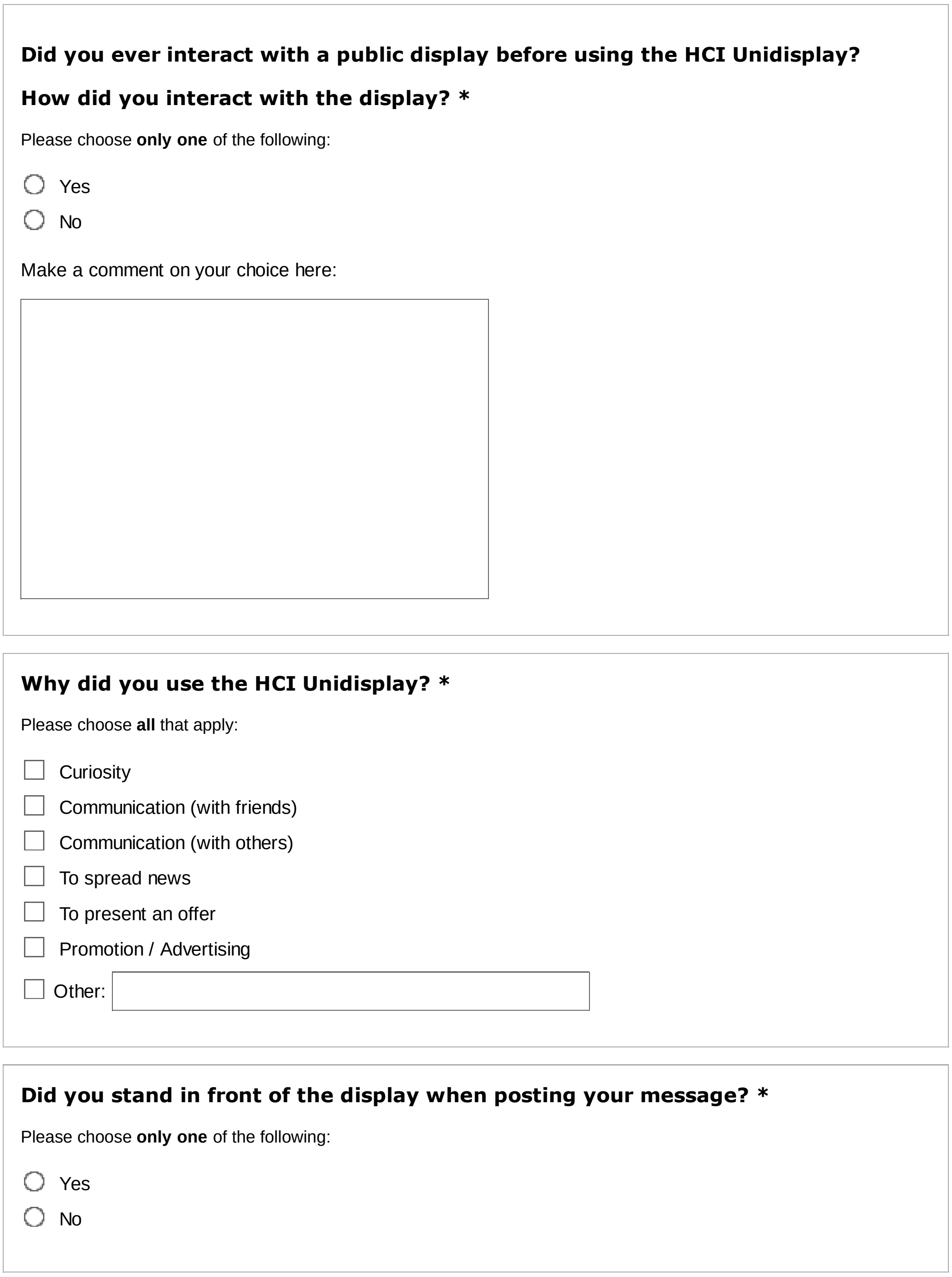
Please write your answer here:

Your Twitter username: *

Please write your answer here:

What's your field of study at university or your profession (if you're not a student or employee of the university)? *

Please write your answer here:



Did you ever interact with a public display before using the HCI Unidisplay?

How did you interact with the display? *

Please choose **only one** of the following:

○ Yes

○ No

Make a comment on your choice here:

Why did you use the HCI Unidisplay? *

Please choose **all** that apply:

□ Curiosity

□ Communication (with friends)

□ Communication (with others)

□ To spread news

□ To present an offer

□ Promotion / Advertising

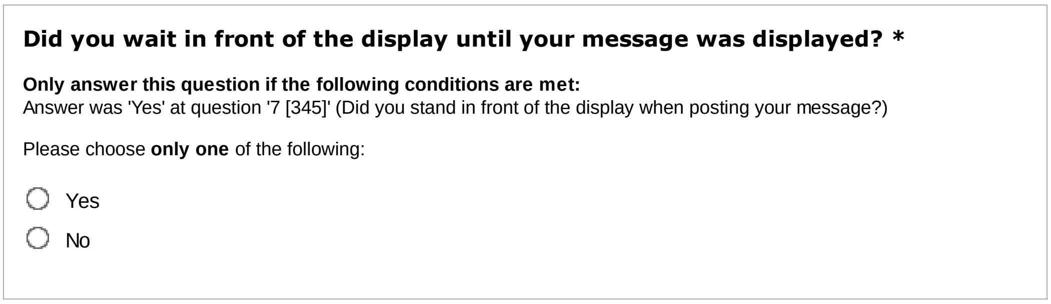
□ Other:

Did you stand in front of the display when posting your message? *

Please choose **only one** of the following:

○ Yes

○ No



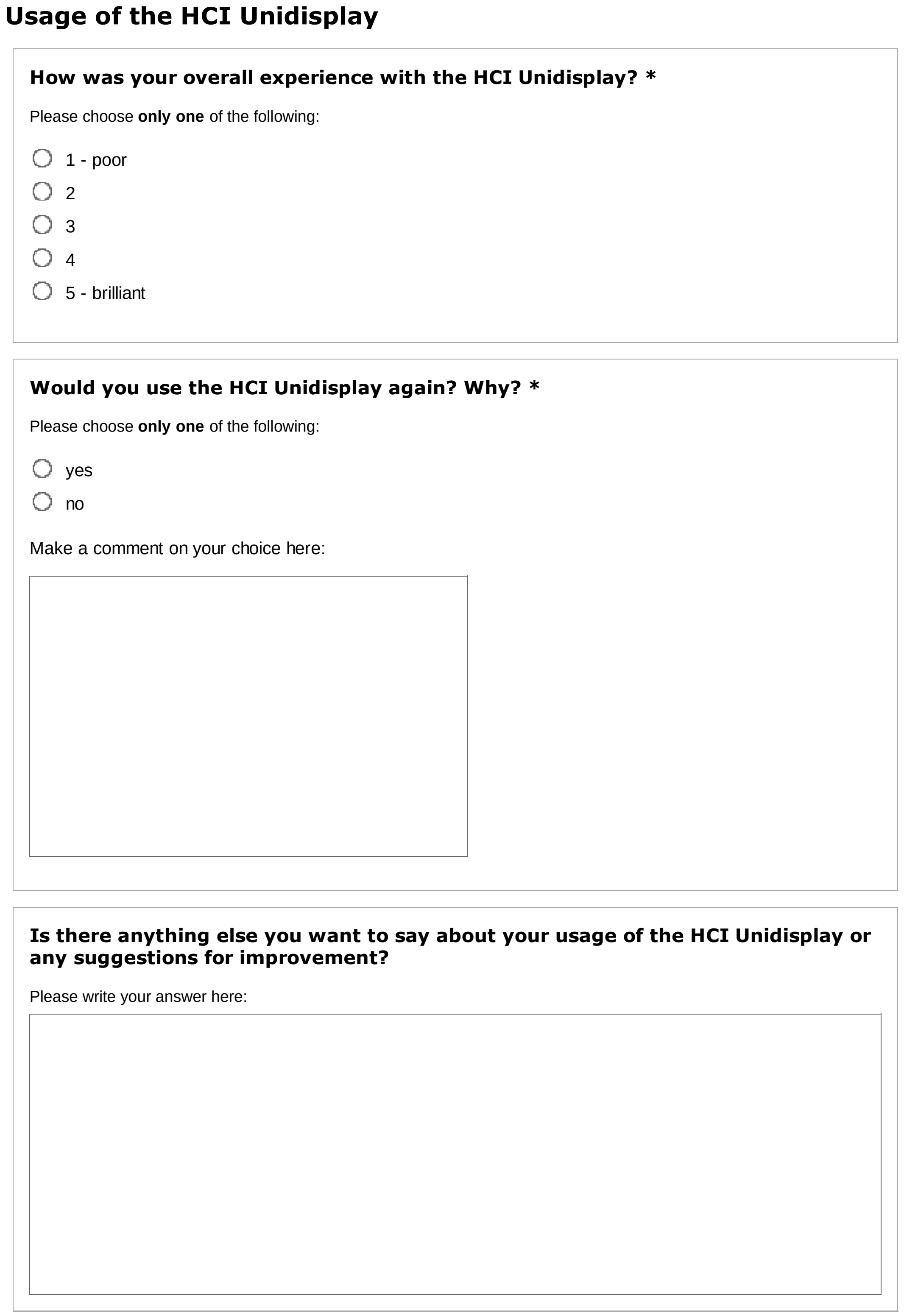
Did you wait in front of the display until your message was displayed? *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '7 [345]' (Did you stand in front of the display when posting your message?)

Please choose **only one** of the following:

○ Yes

○ No



Usage of the HCI Unidisplay

How was your overall experience with the HCI Unidisplay? *

Please choose **only one** of the following:

○ 1 - poor
○ 2
○ 3
○ 4
○ 5 - brilliant

Would you use the HCI Unidisplay again? Why? *

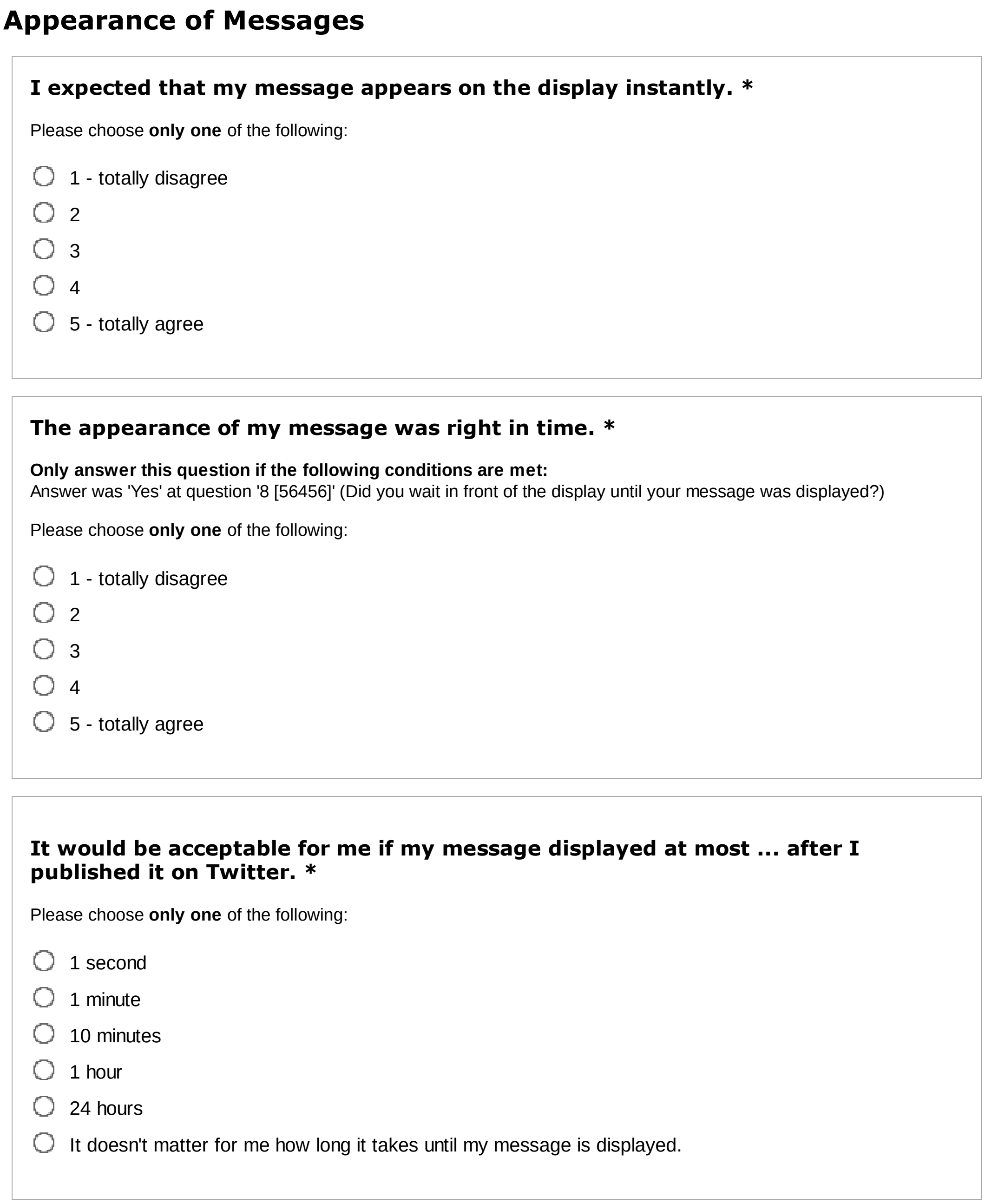
Please choose **only one** of the following:

○ yes
○ no

Make a comment on your choice here:

Is there anything else you want to say about your usage of the HCI Unidisplay or any suggestions for improvement?

Please write your answer here:



Appearance of Messages

I expected that my message appears on the display instantly. *

Please choose **only one** of the following:

○ 1 - totally disagree
○ 2
○ 3
○ 4
○ 5 - totally agree

The appearance of my message was right in time. *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '8 [56456]' (Did you wait in front of the display until your message was displayed?)

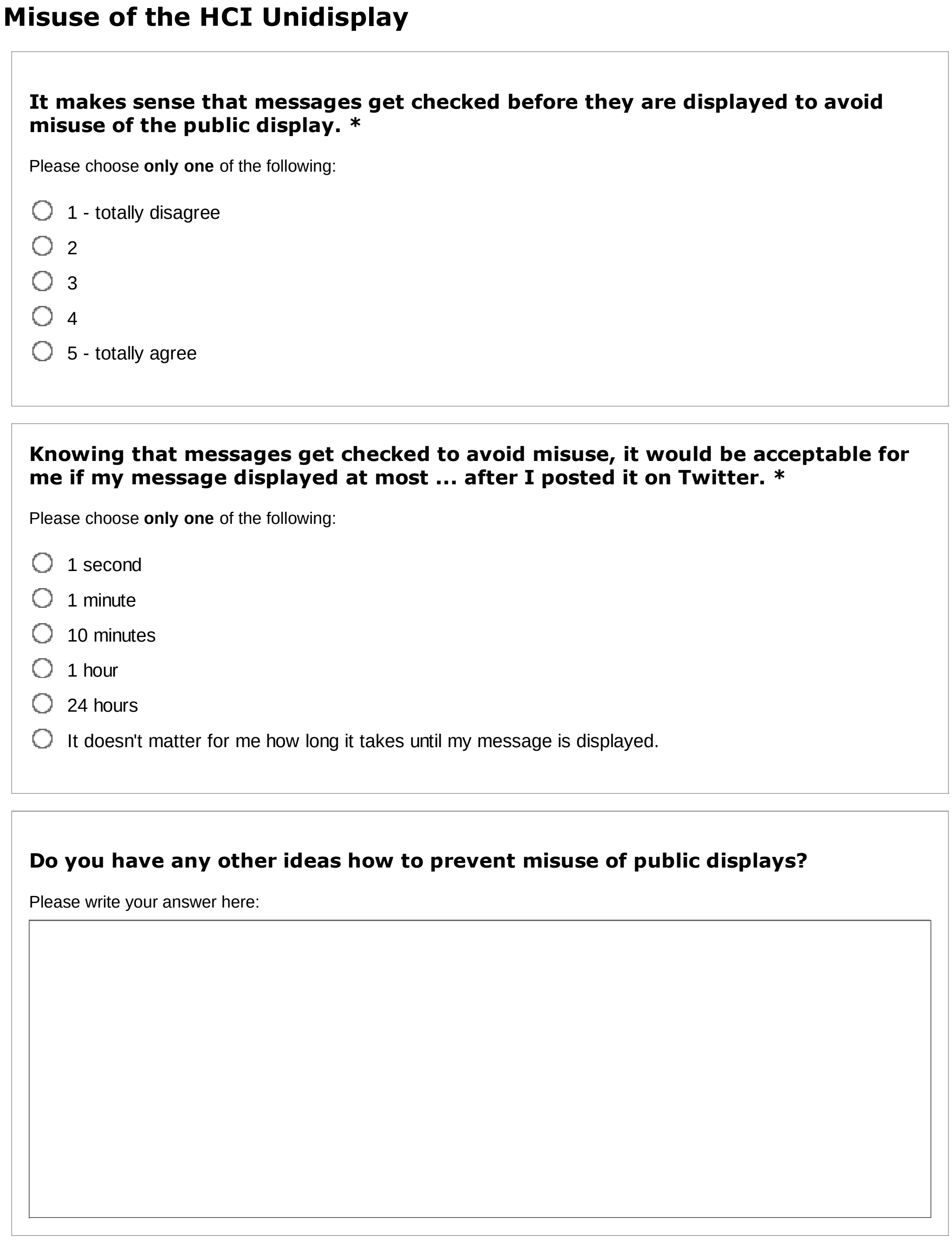
Please choose **only one** of the following:

○ 1 - totally disagree
○ 2
○ 3
○ 4
○ 5 - totally agree

It would be acceptable for me if my message displayed at most ... after I published it on Twitter. *

Please choose **only one** of the following:

○ 1 second
○ 1 minute
○ 10 minutes
○ 1 hour
○ 24 hours
○ It doesn't matter for me how long it takes until my message is displayed.



Misuse of the HCI Unidisplay

It makes sense that messages get checked before they are displayed to avoid misuse of the public display. *

Please choose **only one** of the following:

○ 1 - totally disagree
○ 2
○ 3
○ 4
○ 5 - totally agree

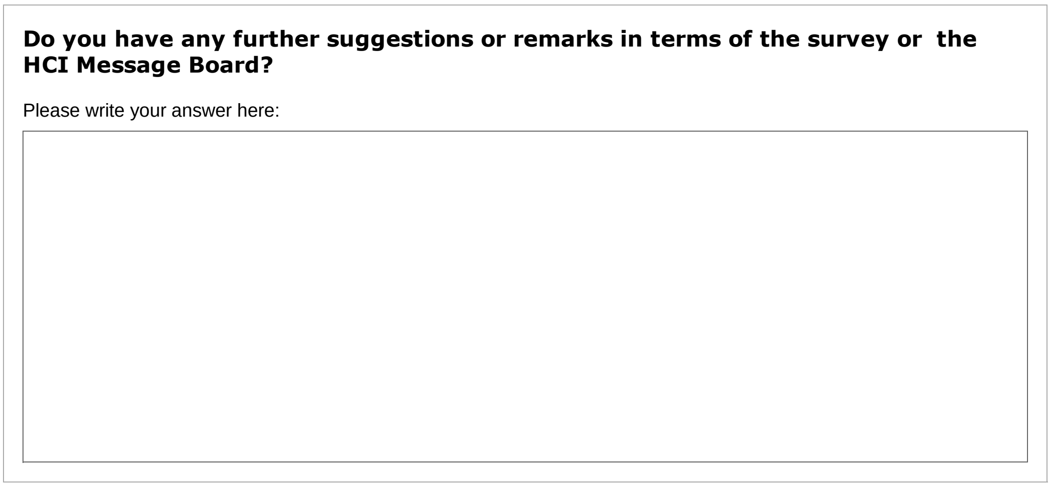
Knowing that messages get checked to avoid misuse, it would be acceptable for me if my message displayed at most ... after I posted it on Twitter. *

Please choose **only one** of the following:

○ 1 second
○ 1 minute
○ 10 minutes
○ 1 hour
○ 24 hours
○ It doesn't matter for me how long it takes until my message is displayed.

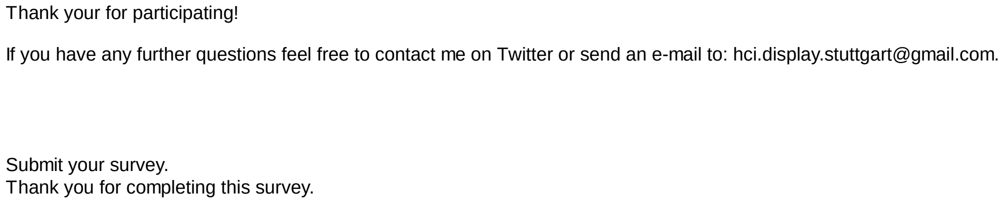
Do you have any other ideas how to prevent misuse of public displays?

Please write your answer here:



Do you have any further suggestions or remarks in terms of the survey or the HCI Message Board?

Please write your answer here:



Thank your for participating!

If you have any further questions feel free to contact me on Twitter or send an e-mail to: hci.display.stuttgart@gmail.com.

Submit your survey.
Thank you for completing this survey.

Bibliography

[AKB+11] F. Alt, T. Kubitza, D. Bial, F. Zaidan, M. Ortel, B. Zurmaar, T. Lewen, A. S. Shirazi, A. Schmidt. Digifieds: insights into deploying digital public notice areas in the wild. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, pp. 165–174. ACM, 2011.

[AME+11] F. Alt, N. Memarovic, I. Elhart, D. Bial, A. Schmidt, M. Langheinrich, G. Harboe, E. Huang, M. P. Scipioni. Designing shared public display networks–implications from today's paper-based notice areas. In *Pervasive Computing*, pp. 258–275. Springer, 2011.

[AMS12] F. Alt, J. Müller, A. Schmidt. Advertising on public display networks. *Computer*, 45(5):50–56, 2012.

[ASKS13] F. Alt, A. S. Shirazi, T. Kubitza, A. Schmidt. Interaction techniques for creating and exchanging content with public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1709–1718. ACM, 2013.

[BIF+04] H. Brignull, S. Izadi, G. Fitzpatrick, Y. Rogers, T. Rodden. The introduction of a shared interactive surface into a communal space. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp. 49–58. ACM, 2004.

[BIT13] BITKOM. Soziale Netzwerke 2013, 2013. URL http://www.bitkom.org/files/documents/SozialeNetzwerke_2013.pdf.

[BR03] H. Brignull, Y. Rogers. Enticing people to interact with large public displays in public spaces. In *Proceedings of INTERACT*, volume 3, pp. 17–24. 2003.

[CDF+05] K. Cheverst, A. J. Dix, D. Fitton, C. Kray, M. Rouncefield, G. Saslis-Lagoudakis, J. G. Sheridan. Exploring Mobile Phone Interaction with Situated Displays. In *PERMID*, pp. 43–47. 2005.

[CDFE11] S. Clinch, N. Davies, A. Friday, C. Efstratiou. Reflections on the long-term use of an experimental digital signage system. In *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 133–142. ACM, 2011.

[CDKS12] S. Clinch, N. Davies, T. Kubitza, A. Schmidt. Designing application stores for public display networks. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, p. 10. ACM, 2012.

[CND+03a] E. Churchill, L. Nelson, L. Denoue, P. Murphy, J. Helfman. The plasma poster network. In *Public and Situated Displays*, pp. 233–260. Springer, 2003.

[CND03b] E. F. Churchill, L. Nelson, L. Denoue. Multimedia fliers: Information sharing with digital community bulletin boards. In *Communities and technologies*, pp. 97–117. Springer, 2003.

[DFCS10] N. Davies, A. Friday, S. Clinch, A. Schmidt. Challenges in developing an app store for public displays–a position paper. In *Research in the Large: Using App Stores, Markets and other Wide Distribution Channels in UbiComp Research Workshop, UbiComp*. 2010.

[DLJS12] N. Davies, M. Langheinrich, R. José, A. Schmidt. Open display networks: A communications medium for the 21st century. *Computer*, 45(5):58–64, 2012.

[Elh12] I. Elhart. Control and Selection of Content on a Network of Public Displays. In *PhD Colloquium, Pervasive*. 2012.

[EMLR13] I. Elhart, N. Memarovic, M. Langheinrich, E. Rubegni. Control and scheduling interface for public displays. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 51–54. ACM, 2013.

[FV02] A. Ferscha, S. Vogl. Pervasive web access via public communication walls. In *Pervasive Computing*, pp. 84–97. Springer, 2002.

[GBL99] S. Greenberg, M. Boyle, J. LaBerge. PDAs and shared public displays: Making personal information public, and public information personal. *Personal Technologies*, 3(1-2):54–64, 1999.

[GL09] D. Gurzick, W. G. Lutters. Towards a design theory for online communities. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, p. 11. ACM, 2009.

[GMS+13] S. Gianluca, M. Milano, J. Saldivar, T. Nasir, M. Zancanaro, G. Convertino. Agora2. 0: enhancing civic participation through a public display. In *Proceedings of the 6th International Conference on Communities and Technologies*, pp. 46–54. ACM, 2013.

[GR01] S. Greenberg, M. Rounding. The notification collage: posting information to public and personal displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 514–521. ACM, 2001.

[HKB08] E. M. Huang, A. Koster, J. Borchers. Overcoming assumptions and uncovering practices: When does the public really look at public displays? In *Pervasive Computing*, pp. 228–243. Springer, 2008.

[HKK+12] S. Hosio, V. Kostakos, H. Kukka, M. Jurmu, J. Riekki, T. Ojala. From school food to skate parks in a few clicks: using public displays to bootstrap civic engagement of the young. In *Pervasive Computing*, pp. 425–442. Springer, 2012.

[JOIH08] R. José, N. Otero, S. Izadi, R. Harper. Instant places: Using bluetooth for situated interaction in public displays. *Pervasive Computing, IEEE*, 7(4):52–57, 2008.

[JPSM13] R. José, H. Pinto, B. Silva, A. Melro. Instant Places: Using pins and posters as paradigms for content publication for situated displays. *Computer Graphics and Applications, IEEE*, 33(2):64–72, 2013.

[LCL09] J.-K. Lou, K.-T. Chen, C.-L. Lei. A collusion-resistant automation scheme for social moderation systems. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pp. 1–5. IEEE, 2009.

[LMEA11] M. Langheinrich, N. Memarovic, I. Elhart, F. Alt. Autopoiesic Content: A Conceptual Model for Enabling Situated Self-generative Content for Public Displays. In *Proc. of the 1st Int. Workshop on Pervasive Urban Applications at Pervasive.* 2011.

[LR04] C. Lampe, P. Resnick. Slash (dot) and burn: distributed moderation in a large online conversation space. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 543–550. ACM, 2004.

[MCH08] J. F. McCarthy, B. Congleton, F. M. Harper. The context, content & community collage: sharing personal digital media in the physical workplace. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pp. 97–106. ACM, 2008.

[MCL01] J. F. McCarthy, T. J. Costa, E. S. Liongosari. Unicast, outcast & groupcast: Three steps toward ubiquitous, peripheral displays. In *Ubicomp 2001: Ubiquitous Computing*, pp. 332–345. Springer, 2001.

[MCL$^+$13] N. Memarovic, K. Cheverst, M. Langheinrich, I. Elhart, F. Alt. Tethered or free to roam: the design space of limiting content access on community displays. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*, pp. 127–132. ACM, 2013.

[MEL11] N. Memarovic, I. Elhart, M. Langheinrich. FunSquare: First experiences with autopoiesic content. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, pp. 175–184. ACM, 2011.

[MEM$^+$13] N. Memarovic, I. Elhart, A. Michelotti, E. Rubegni, M. Langheinrich. Social networked displays: integrating networked public displays with social media. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 55–58. ACM, 2013.

[Mik13] M. Mikusz. *Building and Using an Application Store to Support Public Display Users.* Diploma thesis, Universität Stuttgart, 2013.

[MMH07] A. Maunder, G. Marsden, R. Harper. Creating and sharing multi-media packages using large situated public displays and mobile phones. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, pp. 222–225. ACM, 2007.

[MR08] I. MacColl, I. Richardson. A cultural somatics of mobile media and urban screens: Wiffiti and the IWALL prototype. *Journal of Urban Technology*, 15(3):99–116, 2008.

[MRR11] S. A. Munson, E. Rosengren, P. Resnick. Thanks and tweets: comparing two public displays. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pp. 331–340. ACM, 2011.

[MWE$^+$09] J. Müller, D. Wilmsmann, J. Exeler, M. Buzeck, A. Schmidt, T. Jay, A. Krüger. Display blindness: The effect of expectations on attention towards digital signage. In *Pervasive Computing*, pp. 1–8. Springer, 2009.

[OKL$^+$10] T. Ojala, H. Kukka, T. Lindén, T. Heikkinen, M. Jurmu, S. Hosio, F. Kruger. UBI-hotspot 1.0: Large-scale long-term deployment of interactive public displays in a city center. In *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on*, pp. 285–294. IEEE, 2010.

[PKS$^+$08] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, P. Saarikko. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1285–1294. ACM, 2008.

[PSJ$^+$07] P. Peltonen, A. Salovaara, G. Jacucci, T. Ilmonen, C. Ardito, P. Saarikko, V. Batra. Extending large-scale event participation with user-created mobile media on a public display. In *Proceedings of the 6th international conference on Mobile and ubiquitous multimedia*, pp. 131–138. ACM, 2007.

[RDS02] D. M. Russell, C. Drews, A. Sue. Social aspects of using large public interactive displays for collaboration. In *UbiComp 2002: Ubiquitous Computing*, pp. 229–236. Springer, 2002.

[RHM08] W. Ryan, W. R. Hazlewood, K. Makice. Twitterspace: A co-developed display using Twitter to enhance community awareness. In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, pp. 230–233. Indiana University, 2008.

[RML11] E. Rubegni, N. Memarovic, M. Langheinrich. Talking to Strangers: Using Large Public Displays to Facilitate Social Interaction. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, pp. 195–204. Springer, 2011.

[Sch12] R. Schroeter. Engaging new digital locals with interactive urban screens to collaboratively improve the city. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp. 227–236. ACM, 2012.

[SOC08] J. Scheible, T. Ojala, P. Coulton. MobiToss: A novel gesture based interface for creating and sharing mobile multimedia art on large public displays. In *Proceedings of the 16th ACM international conference on Multimedia*, pp. 957–960. ACM, 2008.

[TC09] N. Taylor, K. Cheverst. Social interaction around a rural community photo display. *International Journal of Human-Computer Studies*, 67(12):1037–1047, 2009.

[TC11] N. Taylor, K. Cheverst. Rural Communities and Pervasive Advertising. In *Pervasive Advertising*, pp. 269–286. Springer, 2011.

[TCF+07] N. Taylor, K. Cheverst, D. Fitton, N. J. Race, M. Rouncefield, C. Graham. Probing communities: study of a village photo display. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces*, pp. 17–24. ACM, 2007.

[TCM09] N. Taylor, K. Cheverst, J. Müller. Affordances and signifiers of community noticeboards. In *Pervasive 2009 Workshop on Pervasive Advertising.* 2009.

[VO11] V. Valkama, T. Ojala. Stakeholder value propositions on open community testbed of interactive public displays. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pp. 107–113. ACM, 2011.

[WHT06] K. Wise, B. Hamman, K. Thorson. Moderation, response rate, and message interactivity: Features of online communities and their effects on intent to participate. *Journal of Computer-Mediated Communication*, 12(1):24–41, 2006.

[Wri09] S. Wright. The role of the moderator: problems and possibilities for government-run online discussion forums. *Online deliberation: Design, research, and practice*, pp. 233–242, 2009.

All links were last followed on November 26, 2013.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature