Vrije Universiteit Amsterdam

Master Thesis

From Sensor Data to Audience Feedback: A Web-based Visualisation Tool for Performing Artists

Author:
Thomas Röggla

Supervisor:
Prof. Dr. D.C.A. Bulterman
PhD Candidate Chen Wang

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science

in the

Internet and Web Technologies Specialisation
Department of Computer Science

August 2014

VRIJE UNIVERSITEIT AMSTERDAM

Abstract

Faculty of Science
Department of Computer Science

Master of Science

From Sensor Data to Audience Feedback: A Web-based Visualisation Tool for Performing Artists

by Thomas Röggla

This work reports the implementation and evaluation of a platform for visualising GSR (Galvanic Skin Response) sensor data from audiences. This platform is especially targeted at performing artists to provide them with a way to gain deeper insight into how audiences perceive their performances by combining the sensor data with video recordings and traditional ratings.

To this end, this document provides a short overview of the current state of the art on physiological computing and GSR sensors. Then it outlines the requirements for such a platform as gathered with the help of potential end users. A major part of this work provides a thorough insight into the back-end and front-end of the final product.

This final product is then evaluated with the help of potential end users with methods coming from academic research and publicly available products to assess the quality and helpfulness of it. In the results, the testers perceived the idea of the platform well, but also some issues emerged. These were mostly concerned with some visualisations not being approachable enough at first sight. Moreover, the way the evaluation procedure was structured was not optimal.

Finally, some issues with the platform in general are pointed out and ideas for future developments are given.

Acknowledgements

This thesis is the product if an internship which I carried out from February to August 2014 at the *CWI (Centrum Wiskunde & Informatica)* in Amsterdam. There I worked in the *Distributed & Interactive Systems* group to conclude my master studies in the *Internet & Web Technologies* specialisation at the *Vrije Universiteit Amsterdam.*

First and foremost, I would like to thank my supervisor, Prof. Dr. Dick Bulterman for sparking my interst in the topic. Moreover I would like to thank Chen Wang for her help in the initial phases of the thesis and for setting up the final evaluation. Another big thanks goes to Dr. Pablo Cesar for his continued support and his guidance throughout all phases of this project and also thanks to all the members of the *Distributed Interactive Systems* group at CWI for an interesting and enjoyable six months of internship.

- Thomas Röggla

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The need for entertainment has always been a cornerstone of human society. Actors and directors have been staging the plays and people have been watching these performances for their entertainment. In all this time, a round of applause at the end of the performance has been the universally accepted method for the audience to show their appreciation for the performance and provide the artists with feedback. However, with digital media having an influence on the classic disciplines of performing arts, there might be other means to elicit and assess an audience's response to a theatrical performance.

The problem with applause is not that it is not honest or does not properly convey the audience's emotion: A good performance will get vibrant applause and the audience might even be standing up to show their appreciation even more, while a bad one will merely get a few cautious claps. The final round of applause only gives the audience the possibility to assess the overall performance. But what if we want to know how a specific part of a performance was perceived by the audience or how an individual audience member reacts to it on a subconscious level? How can we know whether people are clapping because they thoroughly enjoyed the performance or they only clap because people around them clap? In these cases we have to go beyond that, we need tools that can capture a person's enjoyment and immersion on a subconscious level and tools that allow us to interpret the data we get from it.

One of the most promising approaches to achieve this comes from a field called physiological computing [1], which makes use of physiological sensors to gain insight into a person's emotional state. For performances in particular, each audience member, or a subgroup of the audience is wearing some sort of sensor which measures their body's response to the performance they are experiencing. This way, one can abstract and quantify the subconscious response of a person in terms of numbers and figures. While this may be very useful for the researchers conducting the experiments, which can revert to their scientific background and the intuitions they have developed by working with this data, the people who might actually profit from these readings, namely artists and directors, are left out of the equation. What this work is trying to do is to take the data collected by physiological sensors and process it in order to present it in such a way, that people with no or little background in computer science, such as performing artists, can easily make sense of it and thus hopefully gain a deeper insight on how their performances are perceived by the audience.

To this end, this work describes all the phases from the gathering of requirements, architectural design and implementation to the final evaluation of the finished platform with potential end users.

1.2 Overview

The main focus of this work is is to develop a platform for visualising audience feedback. The term audience feedback not only includes biometric sensor data, but also more traditional rating measures such as questionnaires, where people are asked to rate certain aspects on a numerical scale. As mentioned in [2], it is useful to combine measurements from different sources to obtain a more accurate picture of an person's state of mind.

While there are different kinds of physiological sensors for capturing audience response, this work focuses on the data gathered from GSR (*Galvanic Skin Response*) sensors. These sensors are attached to a person's hands and measure how well the skin conducts electricity. The conductivity of the skin is related to the level of excitement this person perceives and can be used as a method to assess audience response [3]. The issue with this is that a person's response is highly dependent on factors one has usually no control over. Among the factors which affect the baseline of the readings are a person's current state of mind and their general sense of wellbeing at the moment of measurement. Moreover, these factors also have an influence on the strength of fluctuations of the response data.

To transform this data in a way such that artists can use it, we first needed to understand what data and in which form the data might be useful to performers and directors. To this end, an initial basic prototype was developed which was shown to performing artists, representing potential end users, in a focus group discussion. The results of this interview were used to obtain a more formal set of requirements on the basis of which the final product was developed. This final product was also used in the evaluation phase, which included testing of the product by potential end users. In order to quantify these tests, the testers were asked to fill out a questionnaire based on the *Creativity Support Index* (CSI), described in [4]. Moreover the tester's click and scroll actions were recorded to generate heat maps, which provide a way to visualise where the users clicked and how often. Finally, to obtain a more general overview of user behaviour, *Google Analytics* was used.

To elaborate, this work essentially takes data from three different sources. First and foremost sensor-data from physiological sensors, then questionnaire data. Lastly, the component that ties this all together, are video recordings from the performances which are presented in a synchronised way to create new and deeper insights into art performances.

An important distinction for this work is the one between *on-line analysis* and *off-line analysis*. The former taking the sensor data as it is being recorded, processing and visualising it in real time, while the latter is using the data which has been recorded previously (and possibly already processed in some way). This work describes the development of an *off-line* tool. Off-line analysis gives us a few more metrics to work with, such as the average global sensor response and also allows for the interrelation of the sensor data with questionnaire data collected from the participants before and/or after the experiment. Moreover knowing the data for the entire performance beforehand gives more possibilities in performing simple analysis of the audience response at a particular moment without having to resort to machine learning algorithms or other more involved approaches.

Another advantage of an *off-line* tool is the potential for collaboration and easier dissemination of the analyses. As mentioned in [5], directors and especially choreographers like to annotate specific parts of the performance with their own thoughts. This requirement has also come to light in interviews with actors during the requirement gathering phase. Thus, the platform provides an easy way to add and modify textual annotations temporally synchronised with the video.

Synchronisation is another important issue in the context of this work. As already mentioned in the previous paragraph, the user has the possibility to add textual annotations to arbitrary moment of the video stream and assign a temporal duration to it. It goes without saying that these temporal relations between the video file and the annotations need to be preserved over the course of subsequent visits and for different user sessions. Another point where this is an important issue is the synchronisation of the sensor data, which is a numeric value for a participant assigned to a specific point in time during the performance, with the video stream displayed alongside it. Only if they are synchronised in a reasonable manner, the end-user can actually see which parts of the performance elicit which reaction in the audience members.

1.3 Features

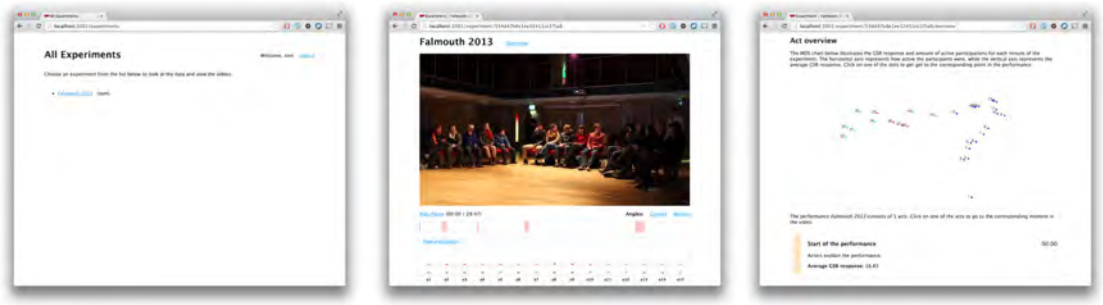


FIGURE 1.1: Collection of screen shots of the final application

All summed up, the developed platform enables end users, which are performing artists and directors, to visualise audience feedback in an easy and intuitive way. The platform is web based to ensure maximum compatibility and accessibility.

The central component of the platform is a video player, intended to play back video recordings from performances where audience members were fitted with biometric sensors. The data gathered from these sensors shall be visualised in an intuitive and easy to understand manner to the end users.

Apart from sensor data, the platform also visualises data obtained using more traditional means of collecting audience feedback data, such as questionnaires.

Another functionality, which is often found in the context of tools which feature multimedia players is the possibility of making textual annotations. For instance, see services like *YouTube*[1] (video annotations) or *SoundCloud*[2] (annotations for audio tracks), but also [5] and [6]. Thus it is natural for this platform to have a similar functionality, which allows users to annotate temporal segments of the video stream.

Finally, a paradigm which is very common in the performing arts, or media in general, is the segmentation of a piece into several parts, or acts in the context of theater performances. With this the platform offers the possibility to create such a segmentation, each with a short description and a brief overview of relevant metrics.

1.4 Contribution

To reiterate the main goals this work tries to achieve and the questions it tries to answer:

Q1 *Can physiological sensor data be interrelated with data from other sources to produce other meaningful information?* This encompasses a review of related literature from the field as well experimentation with the data which has been obtained in experiments. Herein we well also discover whether the sensor data is even a meaningful measure to quantify audience experience.

Q2 *What are performing artists specifically looking for in such a tool and accordingly would they even want to make use of it?* By interviewing potential end-users in a requirement gathering phase before any concrete implementation, we try to gain insight into the needs and expectations that the artists have towards such a tool. The final product shall be evaluated by potential end users as well to gauge whether our endeavour has been successful.

Q3 *How well are first-time users engaged with the platform and how do they navigate the pages?* This research objective mostly concerns the question of usability and shall be answered by running tests with potential end users on the finished platform. Possible

[1] `http://www.youtube.com`
[2] `http://www.soundcloud.com`

tools to evaluate this are questionnaires, click heat-maps and other navigation-based metrics.

Q4 *What ways are there to process the sensor data and in which ways can it be represented in a visually appealing and intuitive way?* The final research question deals with finding ways to represent the data and visualise it. This touches factors such as data normalisation, diagramming or user interface design. The constraint here is imposed by the programming language used, potential availability of related libraries and constraints imposed by the user's web browser while maintaining maximum compatibility.

The order of these research questions roughly corresponds to the order of their appearance in the chapters of this document. The following section provides an outline of the chapters, their contents and which chapters answer which research questions.

1.5 Outline

This document will be a guide through the process of gathering requirements, analysing them, analysing the sensor data, designing and implementing the platform and evaluating it with the end-user. Furthermore, it will provide an overview of the necessary background information in order to get fully acquainted with the matter at hand and offer conclusions and suggestions for future work and improvements. More specifically, the work follows the structure outlined below:

Chapter 1 - Introduction A brief introduction into the subject at hand and a rough overview of the layout of this work.

Chapter 2 - State of the Art An overview of existing approaches in measuring audience feedback and physiological sensors which can be found in the literature. This overview of literature shall answer question **Q1** by investigating possible relationships between different methods of assessing audience feedback.

Chapter 3 - Requirements This chapter looks into the requirement gathering process and their analysis. The findings of this chapter shall serve as a base for the application architecture and the results of the requirement gathering process shall answer research question **Q2** and partially also **Q4**.

Chapter 4 - Application Architecture This chapter formalises the requirements gathered in the previous chapter and transforms them into an application architecture. The results shall provide material to answer additional parts of **Q4**.

Chapter 5 - User Evaluation Using the application developed from the architecture in the previous chapter, this section will present the results, which were gathered from a series of tests with potential end users. This will answer research question **Q3** and should also serve to finally answer **Q4** by assessing the testers' reactions to specific interface components.

Chapter 6 - Conclusions Final wrap-up of the results that came from developing and testing the platform. This chapter will also highlight issues encountered during each of the phases and give suggestions for potential future work.

Chapter 2

Overview of the State of the Art

Before delving into the process of developing a system to assess response data from physiological sensors, it makes sense to get more acquainted with the matter and the underlying technology. This chapter will take a brief look at the variety of sensors which are in use today. It will then go into more detail about GSR (Galvanic Skin Response) sensors, the type of sensor which actually provides the data which the final platform processes. Finally, we will take a look at ways in which audience feedback is be assessed and interpreted and how one might visualise the data on a graphical user interface.

2.1 Sensors

Over past few years, the field of physiological computing and experimental psychology has come up with a wide variety of physiological sensors, which can be used to assess a user's emotional state and quantify it. In [1] Allanson and Fairclough provide a very thorough overview of possible approaches. In their work, they identify the following types of sensors:

Electroencephalogram (EEG) Measures the electrical activity of the surface of the brain by means of electrodes attached to a person's scalp.

Electromyogram (EMG) Measures the tension of a muscle by means of electrodes placed on the skin covering a particular muscle.

Pupillometry This approach measures the speed and direction of eye movements. This can be achieved by means of electrodes or video-based eye tracking.

Electrocardiogram (ECG) Measures contractions of the heart muscles as heart rate or in a frequency spectrum.

Respiratory patterns Analysis of respiratory patterns under the assumption that these change when the user is encountered with a demanding task.

Electrodermal activity/galvanic skin response Two electrodes apply a small current to the skin and measure resistance of the skin, which changes in response to emotional stimuli.

Blood pressure Common and easy to implement measure which measures the force the heart is pumping blood through the body. One has to take a lot of outside factors into account, as they might have an influence on the readings.

All of these techniques vary in usefulness and applicability. While the EEG measurements might provide a very good insight into a person's emotional state by observing stimulation of different parts of the brain, the setup can be rather complicated and the devices for getting good measurements are rather intrusive, as observed by Carroll and Latulipe in [2]. Another issue is that the measurements recorded by some of the sensors might be affected by outside factors such as the user's physiology. A good example for this is to measure blood pressure, which readings are heavily affected by the user's general health and lifestyle. Pupillometry, on the other hand, is completely independent of the user's physiology and provides good insight on the focus of a user's attention, but does not say much about the way they are feeling. Still eye-tracking finds useful applications in user interface evaluation and market research [7].

Another issue for sensors in general is scalability. While it may not be a big problem for most of these approaches to fit a single user with such a sensor, it can get progressively worse the more users are studied. Next to invasiveness and setup, for this issue, one also has to factor in the price. In this regard, galvanic skin response sensors have the advantage of being inexpensive, easy to apply and scale, while at the same time giving reasonable results. As Lang observed in [8], there is a linear relationship between the conductivity of the skin and user arousal.

Apart from the type of sensor, Latulipe et al. make a distinction between *implicit* and *explicit* measurement methods in [9]. In their work, physiological sensors are considered *implicit* methods, since they simply report biometric data, while their interpretation is left up to the researcher. Apart from the types of sensors already identified by Allanson and Fairclough, the authors suggest thermal imaging, computer vision algorithms to detect facial expressions or *body posture detection* as other possible means for measuring engagement implicitly. However, they question the scalability and general applicability of these approaches.

They also note that one could use measurements from different sources, implicit or explicit, and triangulate them in order to gain more meaningful results. This is also confirmed by Bardzell et al. in [10].

2.1.1 Galvanic Skin Response Sensors

After having taken a general look at physiological sensors, we now direct the main focus to GSR sensors. *Galvanic Skin Response* or *Electrodermal Activity* sensors are sensors which are attached to a person's skin. Two electrodes then apply a small current to the skin and measure the skin's electrical resistance according to *Ohm's Law*. The resistance of the skin changes when the eccrine sweat glands in the skin start emitting fluid in response to emotional stimuli (e.g. stress, excitement, happiness, . . .). Because this fluid is an electrolyte solution, it affects the conductivity of the skin. According to [11], the human body possesses about three million of these glands, with soles, palms and the forehead having the highest density, while at the same time not being (or only minimally) covered by hair. Thus, these areas are the preferred places to attach the sensors to. Usually the hand is chosen, as it is least likely to cause an inconvenience for the wearer.

It is natural to assume that the response value of a galvanic skin response sensor is affected by the body temperature. This assumption is accurate, as the sweat glands will emit more fluid as the body gets warmer to regulate the body's temperature, which is indeed their primary function [11]. However, this only affects the baseline of the measurements and the eccrine sweat glands of the palms have been identified to be especially susceptible to emotional arousal [12]. This *emotional sweating* occurs independently from body temperature, as stated by [13].

Apart from body temperature, the measurements may also be influenced by other events, such as change of mechanical pressure on the electrodes (i.e. forcing them tighter onto the skin), loose electrodes, body movements or even speech [14]. Thusly, one needs to take these factors into account by either a more careful setup of the experiment or normalisation/smoothing of data in the analysis phase. A suggested approach for this can be found in [15].

2.1.2 Arousal vs. Valence

To complete the foray to physiological computing and sensor technology, we need to introduce one final concept which is crucial to understand the subject matter. Namely it is the distinction between *arousal* and *valence*. These are two very different concepts and physiological sensors are usually only able to measure the former. Arousal can be interpreted as the strength of an emotion, whereas valence assigns a positive or negative quality to the emotion. In an experiment carried out by Latulipe et. al. in [9], artists were shown arousal data from physiological sensors and asked if they were also interested in valence data. The artists noted that they would not find it interesting since valence data is highly subjective and thus more difficult to measure accurately.

Nevertheless, in order to get a complete picture on a person's emotional state, we need both arousal and valence. Many authors note that these two factors can be seen as two orthogonal dimensions of emotion as noted by Picard in [16]. Moreover, [17] puts them in a two dimensional grid to assess a player's reaction to challenges in different video games. This grid is loosely based on the *affect model* by Russell in [18]. A simplified version of this model with a selected set of so-called *affect words* is outlined in Figure 2.1.

One can see from the figure that on the vertical axis the feelings range from passive and sleepy to very active and engaged. The horizontal axis on the other hand assigns a positive or a negative quality to it. Together they form a complete description of the emotion at hand. As mentioned previously, GSR sensors only measure arousal, i.e. the vertical axis. For assessing theater audiences this is sufficient, since a director is more interested in how attentive an audience is than how positive or negative their emotions are, since the goal of the performance might be to invoke either of them, depending on the type of the performance. Just as noted by interviewees in [9].

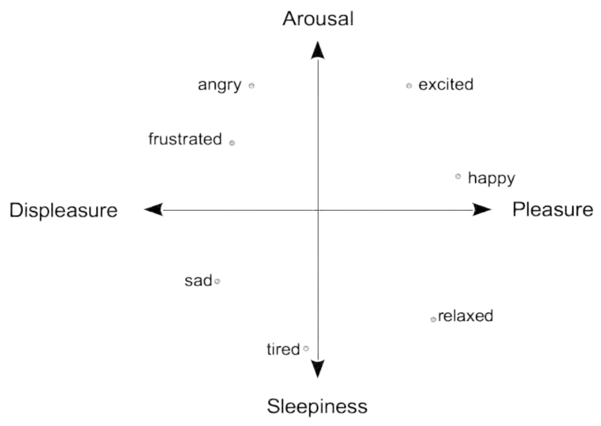


Figure 2.1: Grid with arousal and valence as orthogonal dimension and associated affect words

2.2 Assessing Audience Feedback

One major and one of the earliest applications for GSR sensors has been as part of lie detectors. This again highlights the simplicity and applicability of these sensors. Even though their performance as lie detectors may not be entirely accurate, they have found their applications as a tool for gauging the interest level of people interacting with different types of media. So for instance, Mandryk et al. in [15] used GSR sensors, among other measures to assess people's emotional reactions to video games. Their goal was it to develop a objective and quantitative framework for judging a user's emotion solely based on sensor technology.

Similarly Latulipe et al. in [9] used GSR sensors combined with self-report measures to assess participants' reactions to a dance piece. Their main goal was not the analysis of the sensor data, but the presentation of the data on a graphical user interface.

What both of these solutions have in common is their use of a commercial set of sensors and maybe more importantly, that their trials were conducted on a smaller scale. So in order to be able assess an entire audience's reaction, multiple people, potentially an entire audience, need to be fitted with the sensors and their data needs to be collected. An example of how this can be achieved can be found in [3] and [19]. The authors built a wireless sensor network in which groups of five sensors feed their data into an *Arduino* board, which in turn communicates wirelessly with a computer collecting all the data. This way, the authors were able to collect data in parallel from 15 participants.

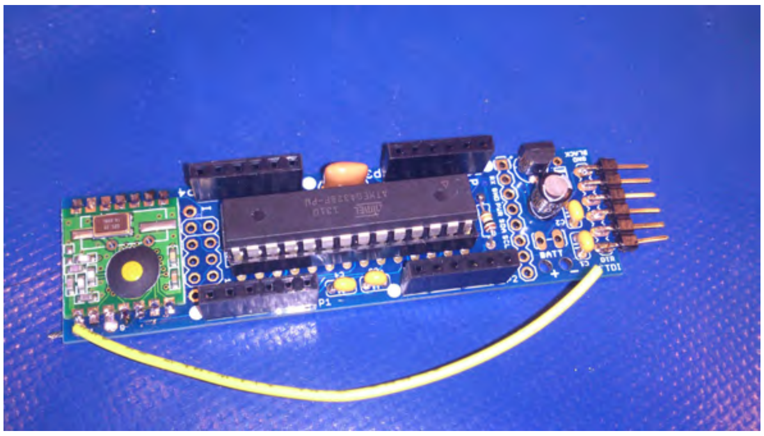


FIGURE 2.2: GSR Sensor

In this experiment, three Arduino boards had each five hand straps attached to them, which were worn by the audience members. The hand straps contained the electrodes which enabled the skin conductance measurements. An updated version of such a GSR sensor without the hand strap attached to it is depicted in Figure 2.2. With this version, each sensor has a wireless module which communicates with a central sink node plugged into the USB port of a computer. This solution is more scalable and can in theory be used with any number of sensors, not just groups of five. Experiments have been carried out with 20 sensors per sink node.

2.3 Available Tools

This final section of this review of literature is dedicated to other tools that are intended for performing artists or that use physiological sensors. There seems to be some effort being put into the creation of user interfaces for choreographers. In [20], the authors describe the evolution of a piece of software aimed at choreographers to aid them in the conception of dance pieces. While this is not really software for assessing audience feedback, it is a good example for the development process of a software project with the involvement of performing artists.

Similarly the authors in [5] and [6] also describe user interfaces and software applications for choreographers and dancers. They both put their focus on support for different types of annotations which can be drawn on video recordings of choreographies. These annotations can be textual or simple sketches for indicating certain movements of a dance piece.

Finally, there is a limited amount of literature about software, which directly visualises sensor data. Unrelated to performing arts, the authors of [21] have used GSR sensor data for personalised music selection based on the wearer's GSR level and developed an interface for PDAs to monitor the data on a line chart. But again, with this, the user itself does not analyse the data. The data is merely used as input to an algorithm which then decides on the music selection to play.

The only application which really visualises data from physiological sensors to be analysed by artists is described in [9]. The authors developed a simple interface playing back a recording of a dance performance temporally synchronised with GSR data that was recorded from people watching the performance. The GSR data is displayed as a simple line chart which can be segmented into chunks in order to reveal average aggregate responses over the selected time frame. The authors of this paper also made extensive studies with artists about the usefulness of such a platform. The interviewed artists noted that they were interested in a way to see how an audience reacts to different parts of performances and use it as a tool to improve their performances.

2.4 Beyond the State of the Art

The remaining question is now what is beyond the state of the art. While there are different types of physiological sensors in use which try to measure a person's emotional state, there is little work done on making this data accessible to people in the performing arts. We have seen that there are some tools to help especially choreographers to *clean* their choreographies by using video playback and annotations, but are not used to assess audience feedback.

The only work that has really been done on visualising sensor data is still rather small-scale and uses comparably simple visualisations. This is where we find our research gap. We aim at a tool, which offers multiple visualisations to analyse sensor data intuitively, temporally synchronised with video recordings and fosters collaboration through annotation support for the performing arts.

// Chapter 3

Requirements

3.1 Basic Requirements

For the implementation of the system a few *basic requirements* have been laid out. The purpose of these requirements is it to lay out a framework on top of which the platform is to be implemented. They include a series of fundamental features which are to be included and on top of which the requirements specified by potential end users are added. Table 3.1 gives a formal outline of these requirements.

ID	**Description**
1	The user shall be able to view a video recording of the performance
2	The user shall be able to see the performance from different camera angles
3	The user shall be able to have full control over the video playback. This includes playing, pausing, stopping and seeking the video
4	The user shall be able to visualise GSR data synchronised with the video recording
5	The user shall be able to log into the page with a personal user account
6	The user shall be able to view view questionnaire data related to the performance

Table 3.1: Fundamental requirements for the system

From the very beginning of the project, it was clear that video playback is central to the functionality of the platform. That is why so many of these basic requirements either directly or indirectly relate to it. Also seamlessly changing the camera angles is found under these basic requirements because previous experiments had been recorded from different angles and it was found useful to be able to see the same situation from a different perspective, as to maybe once see an event happening on stage and then directly switch to a camera angle facing the audience to observe the audience's reaction to the event.

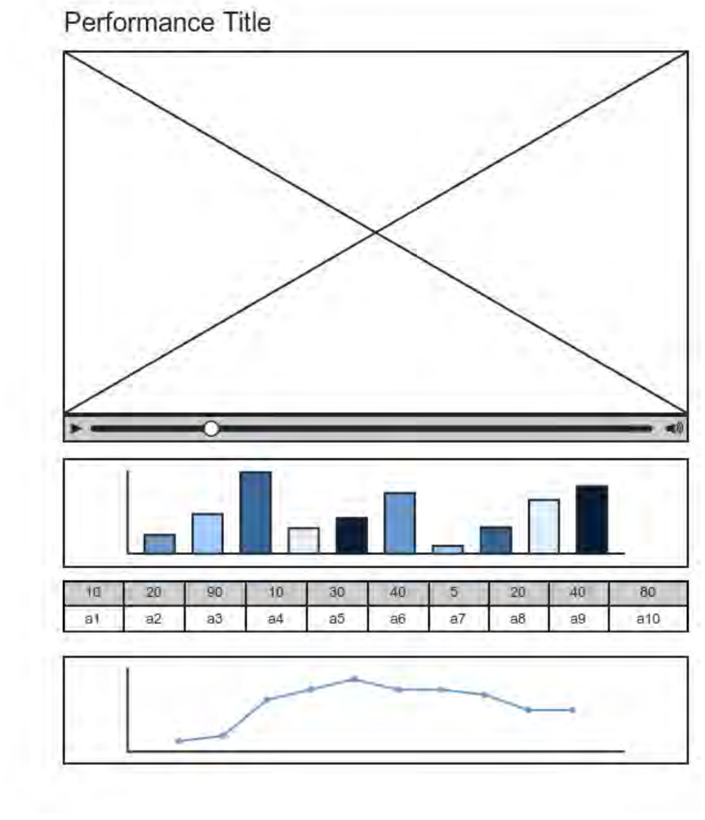
The second major functionality that the system should be able to provide is the visualisation of the gathered data alongside video playback. From the initial specification this includes (1) data collected from biometric GSR sensors attached to the hands of audience members and (2) data collected from questionnaires filled out before and after the performance by the audience members wearing the sensors. The sensor data consists of a series of numbers divided by participants ranging form 0 to 1023. Ideally each of the numbers in the series represents one second in real time and is in sync with the video file. The questionnaire data, on the other hand, represent an overall rating of the performance by each audience member. They were asked to rate the strength of different feelings before and after the performance on a graphical scale. These ratings were then transformed into a numeric value by measuring the location of the mark the participant had drawn on the graphical rating scale.

3.2 Basic Prototype

Equipped with some initial ideas for visualisations and the basic requirements, a basic prototype of the final platform was developed. This was done with the intention to give potential end users an idea of what is possible and what can be done with the data and how it can be visualised. Figure 3.1 shows a basic mock-up of the first prototype.

This basic prototype contains a video player with playback control functionality and a seek bar for jumping to arbitrary points in the video stream. Moreover, it already provides a way to visualise the averaged GSR response of the entire audience on a line plot. The data for each individual participant of the experiment is displayed in a table directly below the video. Moreover, to make the visualisation of differences between each participant easier, the table also contains a bar chart with a bar for each participant representing their current GSR response value on a scale from 0 to 100.

This basic prototype was then implemented in order to be used in the requirement gathering process with the artists. This implementation proved very useful, since it provided the artists with an idea of what was actually possible. For screen shots and



10	20	90	10	30	40	5	20	40	80
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10

FIGURE 3.1: Wireframe illustration of the first prototype

implementation details of this first prototype, refer to Chapter 4. The next sections will go into the requirement gathering process involving artists that lead the implementation of the final version of the system.

3.3 Requirement Gathering

FIGURE 3.2: Interview with artists

The main requirements were gathered in an interview with a focus group with three performing artists and two research staff at the *University of Falmouth* in Falmouth, UK. The platform was also shown to other faculty members of the university and they were asked about their opinion. One of them, after being shown the first prototype, noted:

> *"This platform could be very useful to performers and it might even have enough features the way it is now. I feel one of the most important things for a website like this is that it is very simple."*

This comment was reassuring, since this person was also involved in all the experiments and had worked with sensor data previously. Nevertheless, we wanted to implement more sophisticated visualisations to make sure we got the sensor data's full potential. Also, the first prototype still contained a lot of numeric data, which did not necessarily convey a lot of meaning.

When asking a choreographer about some specific visualisations he wanted to see for each participant in the experiment, he suggested:

> *"I think it would be the most intuitive if the current response value for each person is represented by either a smiling or frowning face."*

While this is in essence a good idea, we felt that is fails to accurately convey the meaning of the sensor data, since a high GSR response value does not necessarily corresponds to happiness or positive emotions. Correspondingly, a low value does not necessarily equal negative emotions. GSR sensors can only measure *arousal* and as already mentioned in Chapter 2, we would need to somehow measure *valence* to assign a positive or negative quality to the measurements. In the end, arrows were chosen in favour of faces to represent whether a participant's response curve increases or decreases. The next chapter will offer a more complete description of this.

Apart from these one-to-one discussions with staff of the University of Falmouth, a more structured focus group discussion was set up. The interview lasted for about 45 minutes and was recorded on video. In the interview the artists were first presented with the preliminary prototype and asked about their opinions. Moreover, they were encouraged to think about features they would want to see in the platform or features that they think would be useful.

The overall response of the artists was positive and they said that they could see themselves using the platform to gain insight into their performances. One of the artists noted:

> *"I think what would be more interesting for me, would be to not have the overall levels there at the bottom but the individual ones."*

This expresses the intention of seeing the data at different levels of granularity. Thus in the final implementation, we allot some space for overall response and but also for individual response curves.

FIGURE 3.3: The three artists in the interview

In general, the artists felt that visualisations should be simple and expressive. One of them noted, after we pointed out the possibility to aggregate the data in various ways:

> *"Showing the data in a more explicit way would be better."*

Finally, the artists were presented with the results found in the paper [3], which were obtained from an experiment which had been carried out at the same university one year earlier. One of the artists present in the interview had been playing in that performance (on the very right in Figure 3.3), so the results were especially interesting to him.

The paper contained some more advanced way of visualising the sensors data, such as MDS (*Multidimensional Scaling*) charts and the artists required some explanation before understanding the concepts they express. They noted that this kind of visualisation would be useful, but should definitely include some explanations alongside it, as they are otherwise hard to understand. In particular one of them said about the MDS chart representing the minute-by-minute audience response in [3]:

> *"So that's actually different scenes and how the audience responds to different scenes? [...] This is actually really useful for any theater maker I'd say, or any performer or even musicians."*

As a final question in the interview the artists were asked about annotation support. The artists unanimously responded that annotation support would be important. When being asked about the types of annotation the platform should support, they noted that textual annotations which can be assigned to a certain time frame of the video recording would be enough.

To sum it up, the artists responded positively to the first prototype but wanted more explicit visualisations at different levels of granularity. They liked the idea of showing MDS results but had some trouble understanding it immediately. Overall, they responded very well to the data synchronised in time with the video recording, as expressed by one of the artists:

> *"That timeline just makes such a big difference to understanding it. Because with those graphs [the MDS charts], I don't understand the time, I don't understand the dimension 1, dimension 2."*

After the interview, a more formal set of requirements was compiled. These requirements are outlined in Table 3.2. Together with the basic requirements from previous sections, they form a complete set of requirements on top of which the system was implemented.

These requirements were used to improve the basic prototype. An improved mock-up image can be seen in Figure 3.4. The bar charts were dropped in favour of arrows to visualise whether an individual's response curve increases or decreases. Also, another line chart has been added. One of them displaying the global response and the other one displaying the response curve of an individual audience member.

ID	**Description**
1	The user shall be able to view the overall GSR response data
2	The user shall be able to view individual GSR response data
3	The user shall be able to add textual annotations
4	The user shall be able to remove textual annotations
5	The user shall be able to click on the timeline so see a specific annotation
6	The user shall be able to visualise clustering of audience members
7	The user shall be able to visualise an overview of the entire performance

Table 3.2: Requirements gathered in interviews with potential end users

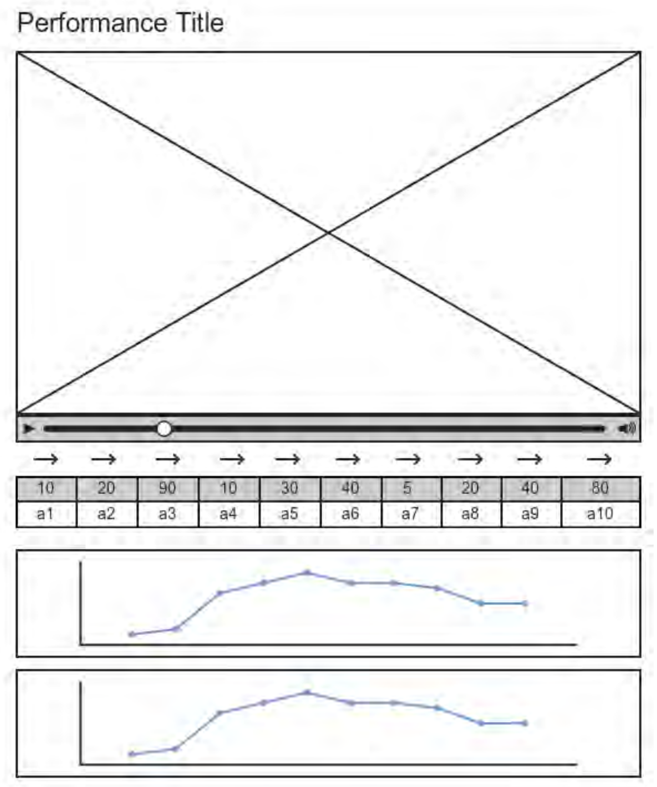


Figure 3.4: Wireframe illustration of the platform after the interviews

Chapter 4

Application Architecture

4.1 Technology Overview

This chapter will provide an overview of the architecture of the final application. Being a web-based application, it is structured into multiple layers, roughly following the *Model-View-Controller* (MVC) design pattern to make sure that concerns can be separated easily. In order to achieve this, this work takes advantage of *JavaScript*. Developed in 1995 by Brendan Eich at Netscape, it has taken a huge upturn in the past few years under the banner of *Web 2.0*. From initially only being used to perform mundane tasks, with the advent of AJAX and new APIs contained in the new HTML5 standard, it has evolved into a fully-fledged solution for implementing rich and interactive web applications.

Initially, JavaScript was only meant to run inside the user's browsers, it has found its way onto the server-side and to databases. Projects like *Node.js* [22] have gained a following because of their different way of approaching web development and allowing the developer a more event-driven view on web applications. Also the database world was influenced by JavaScript. It is used at the heart of many of the *schemaless* or *document-oriented* databases, or *NoSQL* databases to use a common umbrella term. One of the more popular of these NoSQL databases is *MongoDB* [23]. It allows for storage and efficient retrieval of data, grouped together in arrays, dictionaries, strings and numbers, which can be queried using a JavaScript based query language. After all this, it should come to no surprise that JavaScript has been dubbed *The assembly language of the Internet* [24].

This work takes advantage of all of these technologies to reach the end goal. It will use *MongoDB* as a data-storage engine and *Node.js* for retrieving and pre-processing the data. The main bulk of the work, the heavy lifting so to speak, is performed on the client side by the JavaScript engine of the user's browser itself. It is responsible for orchestrating video playback control, data processing, asynchronous communication with the server and rendering of the data. To this end, several libraries are consulted, making tasks such as DOM manipulation, AJAX interaction and SVG rendering much easier and thus preventing us from reinventing the wheel. The following sections provide a technical guide through all the layers of the application and highlight specific design decisions and architectural patterns.

4.2 Preliminary Data Analysis

Before attempting any sort of implementation of the platform, it was proven useful to perform some preliminary analysis on existing data gathered from GSR sensors using statistical software. This helped to gain an initial intuition about the form of the data and how it can be transformed. The analyses presented in this section have been performed using *RStudio*[1], an IDE for the *R programming language.*

The data used in this process comes from an experiment carried out in 2013 at the *University of Falmouth* during a specially staged theater performance. The results of this experiment are presented in [3]. However, a brief description of it will be provided hereafter.

During the performance of 28 minutes in length, 15 participants were fitted with the GSR sensors and their response was collected at one sample per second, yielding a total of 1680 data points for each participant. The performance was segmented in four parts and specifically staged in a way that required active participation from the audience during certain parts. Moreover, the performance contained elements which elicited sudden spikes in the GSR response, such as the popping of a balloon.

[1]http://www.rstudio.com

In the original paper, the authors identified four clusters of participants, which are outlined in Table 4.1. One of the goals of this preliminary data analysis was it to make sure that the visualisation techniques accurately reflect the differences between the clusters. Figure 4.1 shows how the authors identified the clusters outlined in the table. One should refer to the paper for a detailed explanation of the underlying process.

Name	**Description**	**Participants**
Engaged	Participants which felt engaged with the performance	a1, a5, a7, a9, a10, a11, a12, a13, a14, a15
Not engaged	Participants which did not feel engaged with the performance	a3
Got distracted	Got distracted at some point during the performance	a4, a6
Took a while	Took some time to get immersed into the performance	a2, a8

TABLE 4.1: Clusters of participants of the experiment

The final data was collected in a spreadsheet document after being cleaned from noise using statistical software. These spreadsheets contain the 1680 data points (one per second) for each participant in a range from 0 to 1023. The R code, which was used to generate these results and plots can be found in Appendix A.

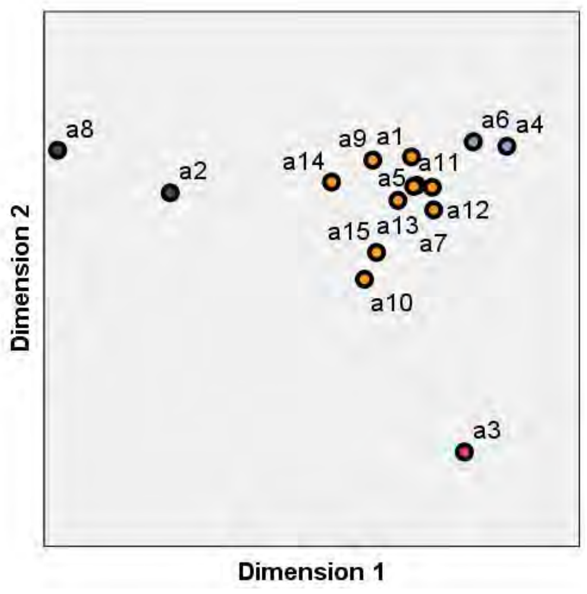


FIGURE 4.1: MDS chart with clusters of audience members identified by different colours

4.2.1 Data Normalisation

As already mentioned in an earlier section, the baseline for the GSR readings varies from person to person and also depends on other factors, such as the person's mood as well as how tight the sensor is attached to the hand. These facts make the normalisation of the data necessary before they can be compared visually in any way. One approach for this comes from [17], where the authors find a global minimum and maximum for all participant and use it to normalise the data using the formula in Equation 4.1.

$$\left(\frac{GSR(i) - GSR_{min}}{GSR_{max} - GSR_{min}}\right) \times 100 \tag{4.1}$$

While this approach accurately projects the data onto a scale from 0 to 100, it has a problem, which is evident in Figure 4.2, which illustrates two data sets normalised with the mentioned formula. The figure depicts two participants of the experiment: One of them, marked with the label *a3*, showing very little variation over the course of the performance, or in numbers, a standard deviation of 0.326 but a very high baseline close to 100. The other one, marked with the label *a4* has with a standard deviation of 10.46 a much higher standard deviation and shows thusly more variation.

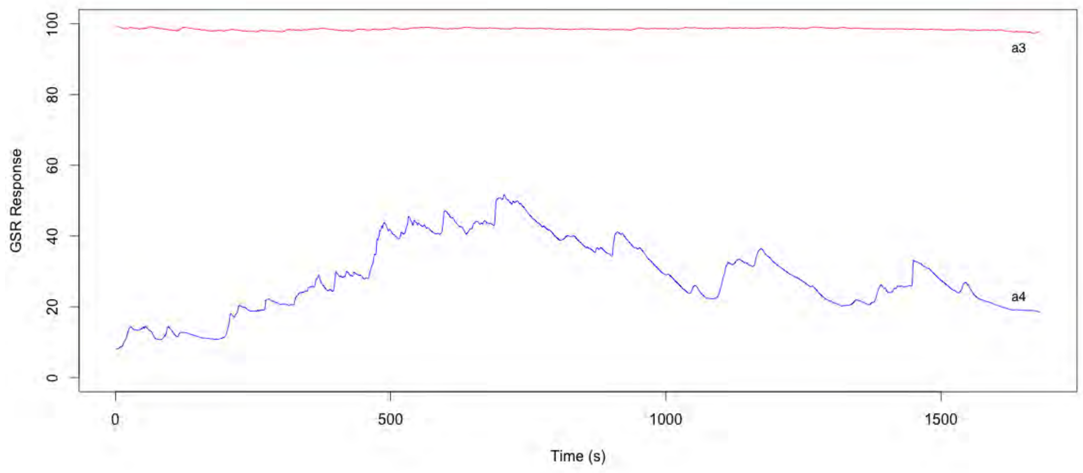


FIGURE 4.2: Two datasets normalised with the naive approach

This might lead one to take the false conclusion that in fact participant *a3* was more engaged with the performance on the grounds of their response value being higher. However from the post-performance questionnaire, which was conducted in the scope of the experiment, this participant stated that they did not enjoy the performance, while the rating of enjoyment given by participant *a4* was much more positive.

Thusly, a normalisation technique which takes these factors into account had to be found. Equation 4.2 outlines the new approach. Similar to Equation 4.1, the formula makes use of a global maximum as normalising factor. The difference here is that it uses the minimum of the data to be normalised (instead of the global minimum) to determine the baseline. This way, we make sure that every response curve eventually touches the x-axis at its minimum, while the overall maximum of all participants serves as a measure to compare the strength of response among participants.

$$\left(\frac{GSR(i) - GSR(i)_{min}}{GSR_{max}}\right) \times 100 \tag{4.2}$$

Visually, the results of this normalisation process are depicted in Figure 4.3. The shapes of the curves stay the same, which is crucial because it allows us to visualise the variation of the response over time, but their position on the y-axis is shifted downwards such that they both touch the x-axis.

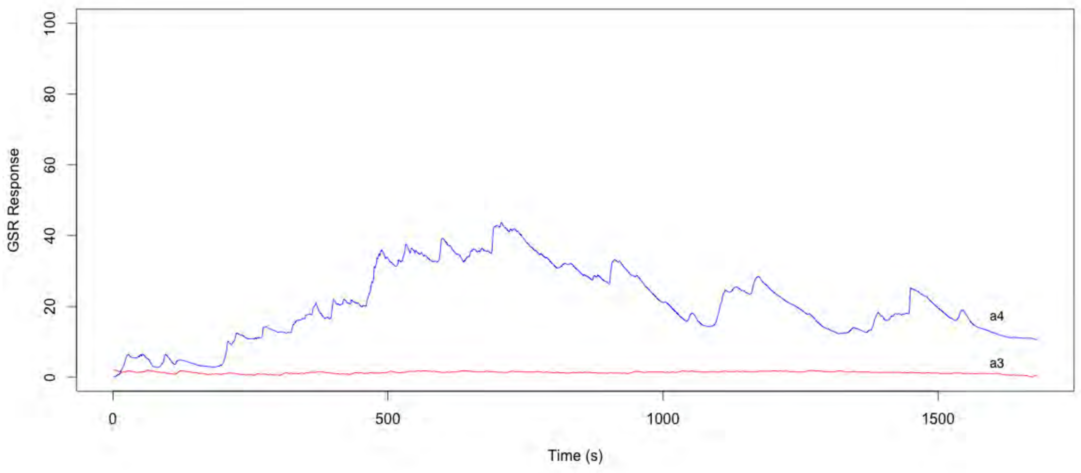


FIGURE 4.3: Two datasets normalised with the proposed approach

This satisfies the requirement that participants from different clusters be represented visually different in the resulting visualisations.

4.2.2 Obtaining a Global Average

In order to see how all participants of an experiment reacted to it, it feels natural to take an average of all the data points and visualise them in some way. To achieve this, we simply compute the row averages of all the participants over the entire time frame. This way we obtain an average value for each second of the experiment. Again, this data is normalised on a scale ranging from 0 to 100 using the same formula that has been used for normalising the individual response values. The resulting curve, as illustrated in Figure 4.4, depicts the average GSR response of the entire audience. It provides a fast way to visualise the audience's response to certain events. So for instance the sudden popping of a balloon can be seen as a maximum in the data at $x = 1463$ (24 minutes and 23 seconds).

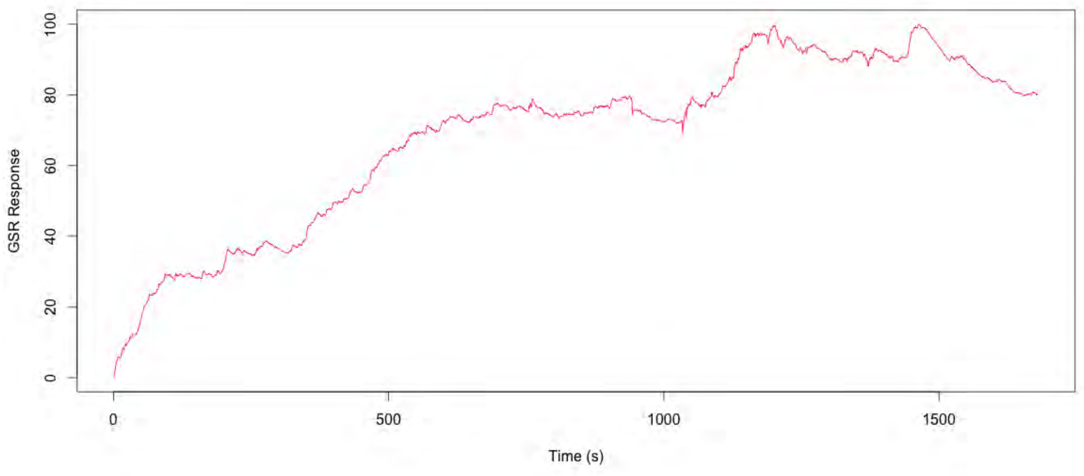


FIGURE 4.4: Average audience response

Note here that when taking the average the shape of the curve stays the same regardless of computing the average over the normalised or the raw data. The only difference is the scale, which bears no significance, since the data is normalised from 0 to 100 anyway.

4.2.3 Advanced Visualisations

While being simple and easy to understand, visualisations such as line charts can only convey a limited set of ideas. In the examples from the past sections they were used to visualise data on a temporal axis, which they are well suited for. Moreover this way of representing data on a temporal axis is almost universally understood.

But if we want to visualise more advanced concepts, we consequently also need to access more advanced methods of visualisation. One such method, which has been used for several purposes in the paper which describes the experiment this data comes from [3], is MDS (*Multidimensional Scaling*). MDS is a technique which allows us to visualise data as dots on a n-dimensional map with the distances between the dots representing their similarities, or in other words, the closer together the dots, the more similar are the underlying values. In order to compute these distances, the MDS algorithm makes use of a so-called *distance matrix*. This distance matrix contains a value representing the similarity - or distance - of each element to each other element. This value is obtained using a fixed *distance metric*, which can be simply the euclidean distance or, in this case, the absolute value of Pearson's product moment correlation coefficient between all possible combinations of data sets of participants. A more complete and formal introduction to the MDS algorithm can be found in [25].

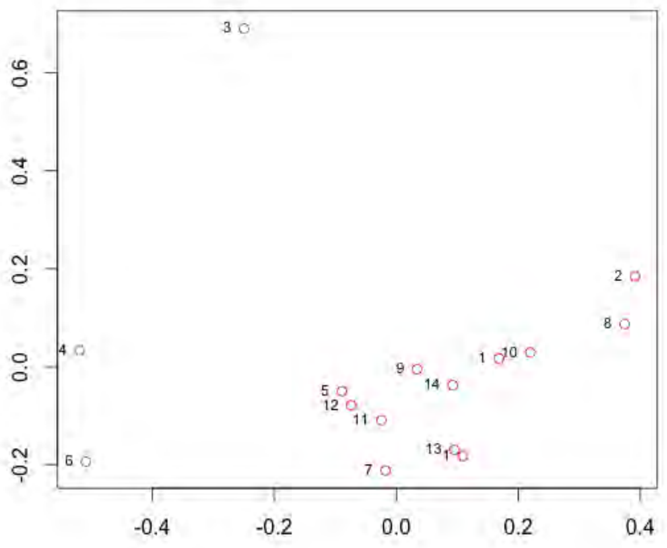


FIGURE 4.5: MDS chart depicting similarity of response curves for each participant

Such a MDS-chart, which depicts the similarity between the response curves of each participant is illustrated in Figure 4.5. In this case, the number of dimensions has been fixed to two, which allows us to visualise the data as a two-dimensional map. The distance metric used is 1 minus the absolute value of Pearson's product moment correlation coefficient (ρ) and is outlined again in Equation 4.3.

$$1 - |\rho(x, y)| \quad \forall x, y \in X \tag{4.3}$$

Here we take the absolute value of the correlation between two sets of observations and subtract it from 1 for each possible combination of observations from a set X. The absolute value is taken because positive and negative correlation should be interpreted as the same degree of similarity. Moreover, the subtraction from one is necessary because the correlation gives a measure for *similarity*, whereas the MDS algorithm requires a measure of *dissimilarity*. The resulting square matrix of dissimilarity values is then fed into the actual MDS algorithm to generate the two-dimensional map. This technique allows us to directly visualise the audience clusters.

The same technique has been applied in [3] to each minute of the performance instead of participants. To this end, one needs to compute the average response values per minute for each participant and transpose this input matrix. The remaining process is analogous to the previous example.

Besides MDS, other techniques for advanced data transformation or visualisation have been explored as well, one of which was *Fourier Transformation*. Fourier transformation takes a signal from a time domain and transforms it into the corresponding frequency domain. This is often used in audio applications, where the technique is used to visualise the strength of specific parts of the frequency spectrum of the signal under investigation.

With the GSR response curves existing in a time domain, Fourier transformation might reveal some interesting properties of the signals. However, looking at the frequency spectrums for three participants in Figure 4.6, there is an evident difference between *a1* and *a3* (*engaged* and *not engaged*) and *a4* and *a3* (*distracted* and *not engaged*), but spotting a significant visual difference between *a1* (engaged) and *a4* (distracted) is very difficult.

There is no immediate visual benefit coming from looking at the frequency spectra of the participant as they appear to be very similar, Fourier transformation as a means of visualisation has been discarded in favour of simpler techniques.

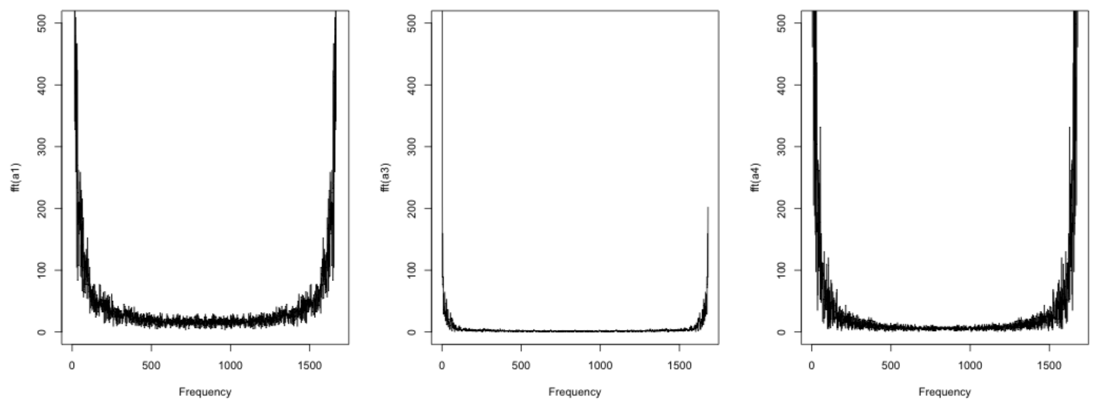


FIGURE 4.6: Frequency spectrum for three participants from different clusters

4.2.4 Real-time Visualisation

While the visualisation techniques described in the previous sections all pertain to the entire performance, it is useful to find some techniques which allow the user to visualise the performance and associated readings at a specific point in time. For this purpose, one could simply display the GSR response value corresponding to one moment in time. However, since the numbers bear little significance, it should be much more appropriate to convey them in a visual way.

One such technique is to numerically compute the slope of a segment of one of the time series and use the resulting number as an indicator whether the value is on the rise, decline or stays approximately the same. A simple means to visually express this is the use of arrows pointing either up to the left for a significantly positive slope, pointing down to the right for significantly negative slope or being straight to the right for a slope value around zero. Trough a process of trial and error a sample size of 10 seconds around the current point in time has proven to be relatively stable while still being able to illustrate a change in slope.

Slope	**Description**	**Corresponding Arrow**
$dy < -0.1$	Curve is on the decline	$\nearrow$
$-0.1 \leq dy \leq 0.1$	No or only small change	$\rightarrow$
$dy > 0.1$	Curve is on the rise	$\searrow$

TABLE 4.2: Thresholds for the slope and corresponding visualisations

Table 4.2 illustrates the chosen thresholds and the corresponding arrows. These thresholds were also determined in a trial and error process and were chosen in way that they change if the underlying curve changes, without being too sensitive, i.e. jumping back and forth erratically. An artist suggested to use smiling, frowning and neutral faces to express the corresponding change in the slope of the curve. While probably intuitive, it might lead one to take false conclusions, since a rise in the GSR value does not necessarily correspond to positive feelings as noted by [9].

The final real-time visualisation, which will be included in the platform displays the normalised GSR response value for each participant as a line on a horizontal axis, in order to visualise the similarity of participants. This is in line with one of the alternatives mentioned by the artists in the focus group interview. One of the artists mentioned that they wanted to see the clustering of audience members at any point in the timeline, but they also felt that the MDS chart was too difficult to grasp. Moreover, testing with real-time MDS charts also revealed some issues, as the chart would not stay stable, i.e. the points would at certain points in the timeline move erratically about the chart area.

4.3 Back-end Architecture

4.3.1 Data Modelling

Given that the platform uses *MongoDB* as a means of storing the data, it does not need any fixed data *per se*. But nevertheless, it makes sense to come up with a basic structure for organising the data. The following paragraphs will describe in detail how the data is organised and stored in the back-end. To do so it will make use of the common tools which are also used for modelling relational data whenever possible.

Figure 4.7 illustrates the basic layout of the data in the database. The central entity, or *document*, as it is called in the world of schemaless databases, is *Experiment*. It harbours an experiments name and all other data associated to it. From the illustration we can see that all other entities are *weak entities* (i.e. they cannot exist on their own) and have a *1-to-n* relationship with the entity *Experiment*. While in the relational world, the data for all of these entities would exist in separate tables, here they actually only exist as JavaScript data structures (e.g. strings, numbers or arrays) nested within an *Experiment* object. The following paragraphs will describe the purpose and selected attributes of each entitiy in greater detail.

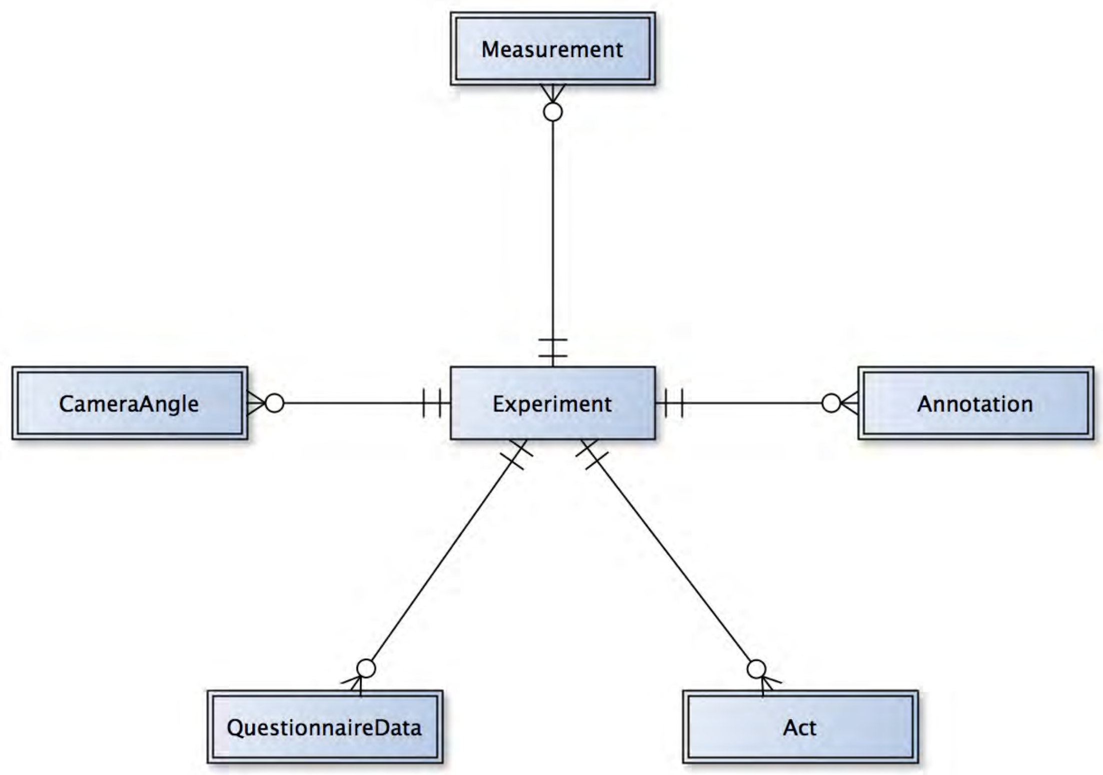


FIGURE 4.7: Basic data model with relationships

Act Theater plays (or arts performances in general) are typically segmented into different *acts*. In this context each act has a timestamp, which indicates its starting time, a name and a short description. These acts are used in the final platform to segment the performance in smaller pieces and compute some more focused metrics. Table 4.3 outlines the exact fields and their data types.

Field Name	**Data Type**
`name`	`String`
`description`	`String`
`timestamp`	`Integer`

TABLE 4.3: Field names and corresponding data types for *Act*

Annotation An annotation is a piece of text, which is associated with a part of the performance at a certain point in time. In the platform, annotations are displayed on the screen at a certain point on the timeline of the video recording and are hidden again after a duration specified in advance. To this end, they contain the piece of text to be displayed, have a starting timestamp and a duration, both specified in seconds from the start of the video. Table 4.4 outlines the exact fields and their data types.

Field Name	Data Type
text	String
timestamp	Integer
duration	Integer

TABLE 4.4: Field names and corresponding data types for *Annotation*

Camera Angle One of the central basic requirements of the platform it is to be able to view a performance from different camera angles. For this reason, the data model includes this entity. It stores a name and path to a video file for each camera angle associated with an experiment. Table 4.5 outlines the exact fields and their data types.

Field Name	Data Type
name	String
url	String

TABLE 4.5: Field names and corresponding data types for *CameraAngle*

Questionnaire Data This entity is responsible for managing the questionnaire results associated with an experiment. This entity is somewhat particular, as it cannot be as easily modelled in a relational setting as the previous entities. It contains two fields *before* and *after*, each of which are optional. These fields contain an array of questions in which the participants rated different feelings before and after the performance. These questions are represented as JavaScript objects with the keys *name*, representing the name of the feeling and *value* which is the rating for the feeling on a scale from 0 to 100. Table 4.6 outlines the exact fields of the topmost level of the data and their types. An example of the JavaScript objects contained in the arrays for *before* and *after* are illustrated in JSON format in Listing 4.1. A formal specification for this format can be found in [26].

```json
{
    "name": "cheerful",
    "value": 59.89
}
```

LISTING 4.1: Example of one of the JavaScript objects representing a questionnaire response

Field Name	**Data Type**
`before`	`Array[Object]`
`after`	`Array[Object]`

TABLE 4.6: Field names and corresponding data types for *QuestionnaireData*

Measurements Finally, the entity *Measurements* is responsible for holding a participants sensor measurements. While in the example experiment, the measurements come from GSR sensors, the database can essentially any kind of measurements as long as it is represented in a numeric format and temporally synchronised with the video recordings. Each entry for measurements holds a unique identifier for each participant as a string and an array of numeric values representing the sensor data. The temporal resolution of the data is assumed to be one second. Table 4.7 outlines the exact fields and their data types.

Field Name	**Data Type**
`name`	`String`
`data`	`Array[Numeric]`

TABLE 4.7: Field names and corresponding data types for *Measurement*

While these entities suffice to describe and model the data as far as the visualisation of sensor data, questionnaire data and associated video recordings goes, there is one more entity to be implemented in order to meet the basic requirements for the platform. This is the abstraction of the concept of the *user*. According to the basic requirements, the platform should support user management and also the main pages of the platform should only be accessible by means of a user account. The upcoming section will describe the entire process of user management and its implementation in greater detail.

4.3.2 User Management

User management and authentication is central to web services. In case of this particular tool, the basic requirements state that the main pages of the platform shall only be accessible via a registered user account. This necessitates the implementation of a simple account management system, at the basis of which lies a database entity called *User*. Unlike all the other entities of the data model (with the exception of *Experiment*), this is not a weak entity and can exist on its own. Table 4.8 shows the field of a user entry and the corresponding data types.

Field Name	Data Type
`username`	`String`
`password`	`String`

TABLE 4.8: Field names and corresponding data types for the *User* entity

For the sake of simplicity, the user entity only contains a user's name and the password, which is associated with this particular user account. Important to note here is that the password is not stored in plain text, but is hashed using *BCrypt*[2].

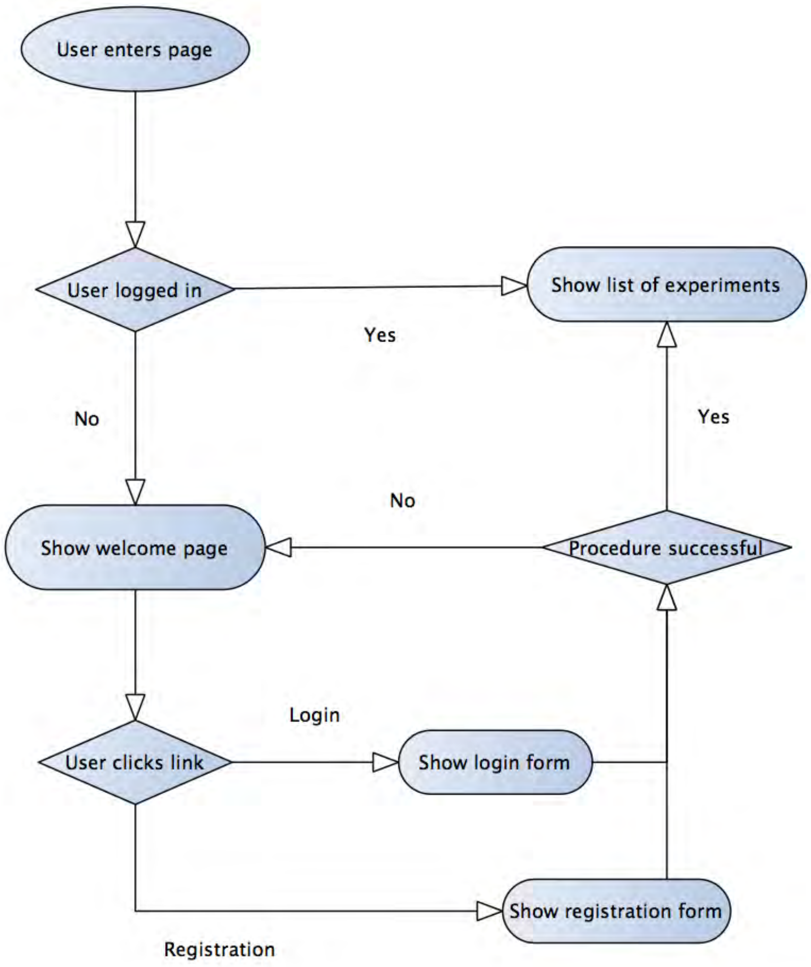


FIGURE 4.8: Behaviour flow for accessing the main pages of the platform

Figure 4.8 shows the behaviour flow for the for log-in and registration. As can be seen from the flow chart, upon entering the page, the system checks whether the current user is logged in. If so, they are immediately presented with a page displaying a list of available experiments (or performances). Should the user not be logged in, they are

[2]`http://bcrypt.sourceforge.net/`

presented with a welcome page, where they can either choose to *log in* to *register* a new account.

If the user chooses to log in, they are presented with a login form, where they can insert their user name and password. The password is hashed and checked against the one stored in the database for the given user name. Upon successful login, the user is redirected to the list of experiments.

Should the user on the other hand choose to register a new account, he is presented with the registration form, where they can choose a new user name and password. Upon successful completion of the registration procedure, the user is logged into their new account and redirected to the list of experiments, where he can start using the main pages of the platform.

4.3.3 MVC Architecture

The *Model View Controller* (MVC) pattern is a common pattern for the development of web application. It allows for the clear separation between user-facing content, data processing and data storage. For this purpose, the platform uses the MVC framework *Express.js*[3], one of the most prominent MVC frameworks for *Node.js*.

The MVC framework groups the files into three categories: *models* for providing an object-oriented abstraction of the database, *controllers* for responding to incoming requests and fetching the data and *views* for rendering the data and presenting it to the user in the web browser.

Models In general, every entity (or document in this case) in the database has its own model class. Given that most entities for the platform are weak entities, the number of models is two: *Experiment* and *User*. They do not have any advanced responsibilities and merely outline the schema of each of the fields of the document alongside their data types. Only the user model contains special methods for hashing and comparing passwords.

[3]http://expressjs.com/

Controllers A bit more involved is the implementation of the controllers for the project. As already with the models, we also have two controllers serving the platform, one for users and one for the experiments. In general each of the methods of a controller corresponds to a request URI. Moreover a controller method may call a routine to render a view, but some of them may not render a view but instead return the data directly in a format like JSON. Table 4.9 shows the request URIs, HTTP methods and data returned for *ExperimentController* class.

Verb	**URI**	**Return**	**Ctrl. Method**
GET	`/experiments`	View	index()
GET	`/experiment/:id`	View	show()
GET	`/experiment/:id/overview`	View	overview()
GET	`/experiment/:id/sensordata`	JSON	getSensorData()
GET	`/experiment/:id/questionnaire`	JSON	getQuestionnaireData()
GET	`/experiment/:id/annotations`	JSON	getAnnotations()
GET	`/experiment/:id/acts`	JSON	getActs()
POST	`/annotation`	Empty	addAnnotation()
DELETE	`/annotation`	Empty	removeAnnotation()

TABLE 4.9: Methods and URI routes for *ExperimentsController*

In the table, there are three methods which render a view. First of all, pointing a web browser to `/experiments` will call the `index()` method of the controller. This method fetches a list of available experiments from the database and renders the list in a HTML view. Once the user selects a specific experiment, they are redirected to `/experiments/:id`, where `:id` is to be replaced with the unique identifier of an experiment as generated by the database. This method also renders a view, which displays the GSR response data for the selected experiment alongside the video recording. The final request which would render a view is `/experiment/:id/overview`. This page calculates some overall statistics for the selected experiment and renders them in a view using for example MDS charts.

The methods in the table which return data in the JSON format are all methods which are being called via AJAX. They return additional data associated to experiments, such as the raw sensor data, questionnaire data, annotations for the experiment and data about acts which the performance is segmented into. Again the `:id` is replaced with the unique identifier for the experiment as generated by the database.

All of the previously described methods respond to HTTP GET requests. However, there are two more methods for creating and destroying annotations. They require a

POST or a DELETE request correspondingly. These two methods expect the data in the request body and return an empty response on success. On failure the methods set a HTTP error code and return an error message.

Verb	**URI**	**Return**	**Ctrl. Method**
GET	`/login`	View	login()
POST	`/login`	Empty	
GET	`/register`	View	register()
POST	`/register`	Empty	
GET	`/logout`	Empty	logout()

TABLE 4.10: Methods and URI routes for *UsersController*

As Table 4.10 shows, the implementation of the controller for users is much less involved. It contains three methods responding to GET requests. Two of which, `/login` and `/register` simply render the login or registration form correspondingly. The third one, `/logout` handles the logout process. It destroys the user's session and redirects them to the root page of the platform. The actual login and registration process is handled by an authentication middleware package for Node.js called *Passport*[4]. It is invoked by invoking the URLs `/login` or `/register` via a POST request with the login or registration data in the request body. This process is handled automatically by the user's browser.

Views Finally, the views rendered by some of the controller methods are simple HTML files containing special annotations in the form of `<%= varname %>`, where `varname` is the name of a variable, but also other constructs such as loops and conditionals are available. For a more complete reference, refer to the documentation of *Embedded JS* available at [27]. These annotations are picked up by the framework and the variable names are replaced by their contents. Once the views are fully rendered, the resulting HTML file is sent to the user's browser where it is displayed.

4.4 Front-end Architecture

After having highlighted various aspects of the back end of the platform, we are now going to take a look at the front end. The front end is everything that the end user sees in their browser. This includes the login and registration process and obviously

[4]http://passportjs.org/

also the main pages of the platform, which display the video recordings, sensor data and resulting visualisations.

Figure 4.9 outlines the login and registration process using screen shots from the implementation of the platform. A user which is not logged in is directed to the welcome page and can then either log in with an existing account or register a new one. With that procedure completed successfully, the user is directed to a list of available experiments to investigate.

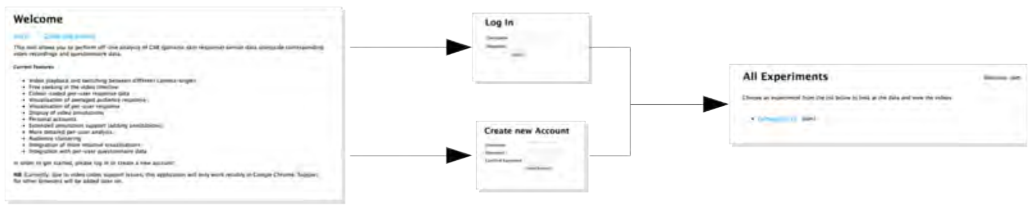
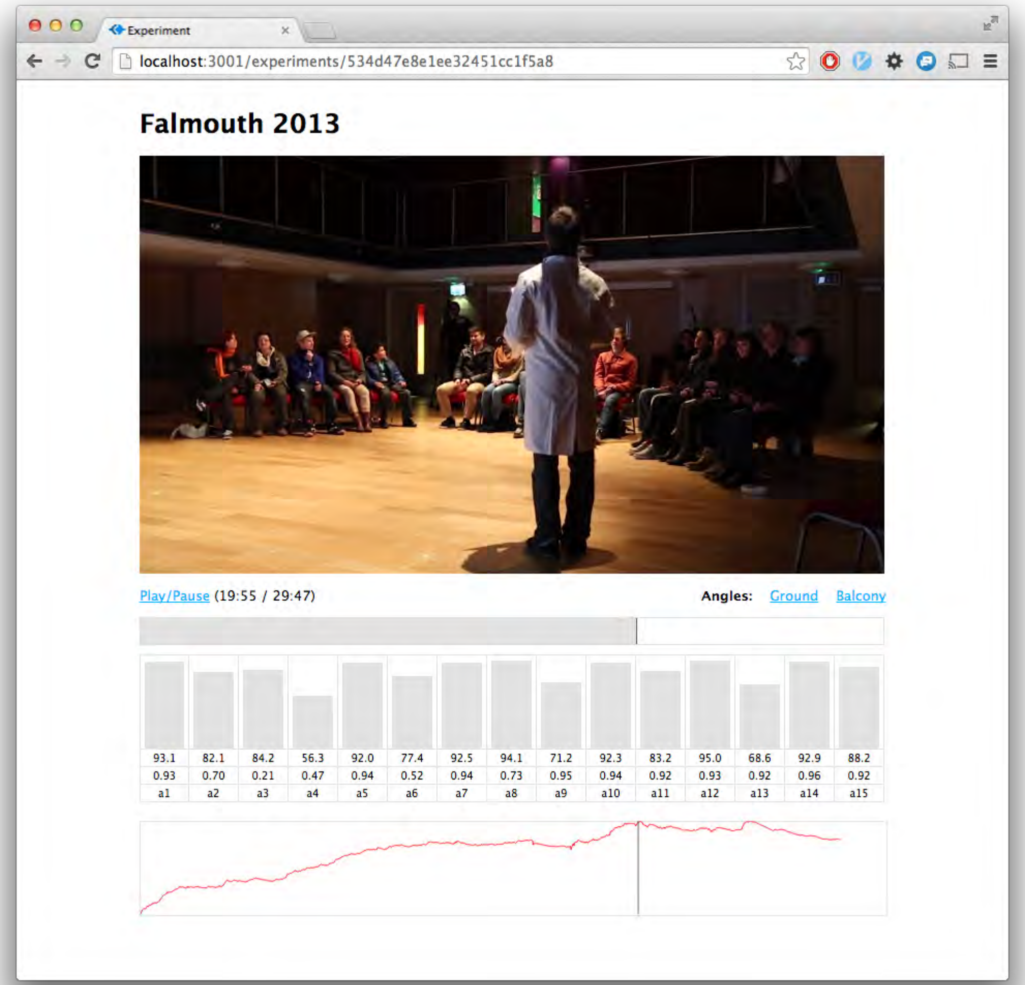


FIGURE 4.9: Behaviour flow in the actual implementation of the platform

Before going into the final version of the platform, we take a look at the basic prototype, which was presented to the artists in the interview for gethering further requirements. Figure 4.10 shows the actual implementation of the basic prototype outlined in Chapter 3 running in the *Google Chrome* browser on *Mac OS X*. The default HTML5 video controls have been disabled and their functionality has been moved to custom elements below the video area to make sure the video area is unobstructed at all times. The interface offers basic playback control, indication of current playback times, a means to switch camera angles and a simple timeline to seek the video.

The interface also displays the current GSR response for each participant in the experiment as a number and on a bar chart. Moreover, each participants' correlation with the global average response is displayed. The bottom of the page is taken up by a line chart displaying the global GSR response. This line chart can also be used as a seek bar.

For the final version of the platform, this structure stayed roughly the same. However, the bar charts, individual response values and correlations were dropped. This is due to the fact that the response values themselves did not express much and also the resulting bar charts did not represent the state of audience members accurately. Moreover, also the correlations were dropped since they did not provide much value and were not intuitive to understand. A screen shot of the final version of the platform can be seen in Figure 4.11. Note that this screen shot displays the content *above the fold*, which is



Falmouth 2013

Play/Pause (19:55 / 29:47)

93.1	82.1	84.2	56.3	92.0	77.4	92.5	94.1	71.2	92.3	83.2	95.0	68.6	92.9	88.2
0.93	0.70	0.21	0.47	0.94	0.52	0.94	0.73	0.95	0.94	0.92	0.93	0.92	0.96	0.92
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15

FIGURE 4.10: First prototype running in an actual web browser

the content which is visible to a user with normal screen resolution without having to scroll.

The video area, playback controls and links for switching camera angles are still the same as in the basic prototype, but the table displaying each of the audience members received a major overhaul. The red stripes on the time line below the video playback controls mark annotations, which are displayed below the timeline whenever the current playback position touches one of the read areas. New annotations can be added by clicking the *New annotation* link. Clicking it will reveal a text field for inserting the annotation text and a number input field, where one can type in the duration in seconds the new annotation shall be displayed.

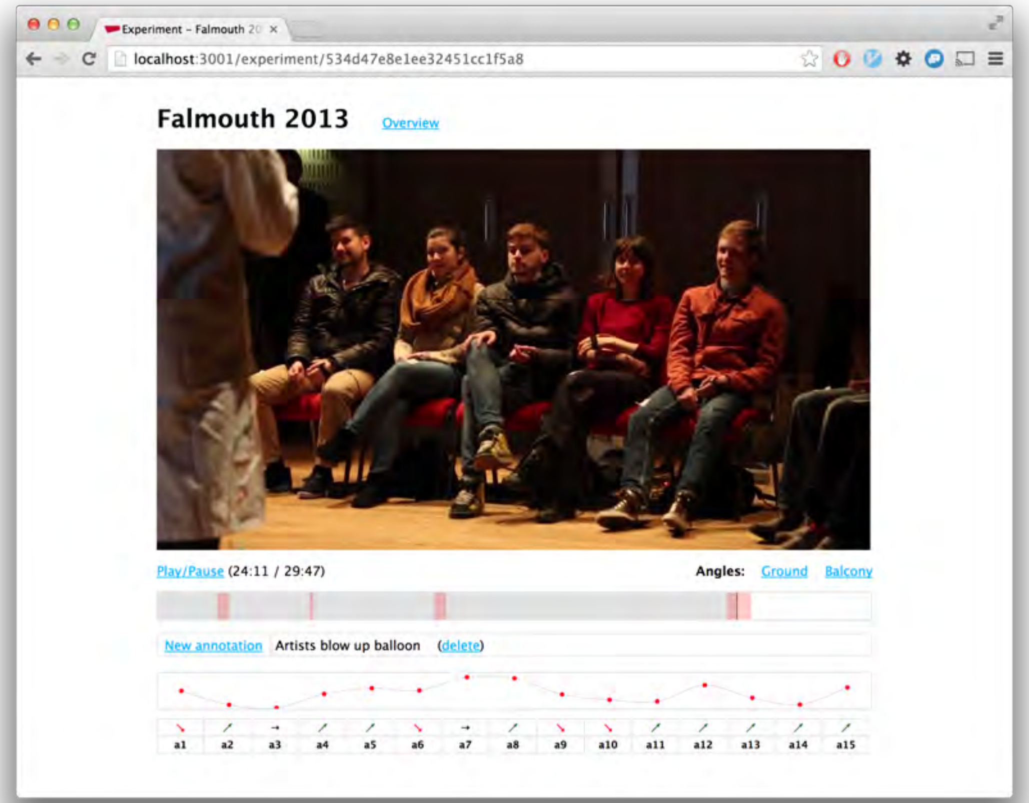
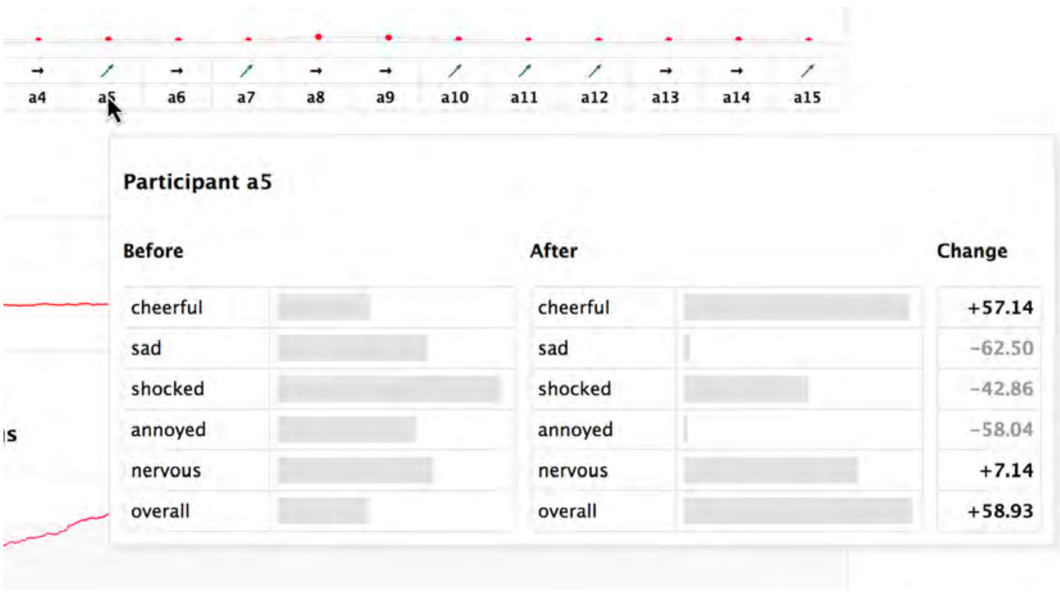

Experiment – Falmouth 20
localhost:3001/experiment/534d47e8e1ee32451cc1f5a8
Falmouth 2013
Overview
Play/Pause (24:11 / 29:47)
Angles: Ground Balcony
New annotation
Artists blow up balloon
(delete)
a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15

FIGURE 4.11: Final version of the platform running in an actual web browser

Bar charts, numeric response values and correlations have been dropped from the table displaying the audience members. Instead, arrows indicating the slope of the response curve of each audience member are displayed. As a replacement for for the bar charts, a chart with red dots connected by *Catmull-Rom splines*[5] has been added. In essence this is similar to the bar charts, but is much more compact and the splines make it easier to spot whether one audience member influences the ones around them by looking at the movement of the dots. While this works fine for this particular experiment where the audience was sitting in a single row, it may not be as useful for different arrangements of the audience.

Figure 4.12 shows how the questionnaire values for an audience member are displayed when hovering over the user's ID. A window appears containing the *pre-performance* and the *post-performance* ratings of different feelings for the selected audience member.

[5]http://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline



Participant a5

Before		**After**		**Change**
cheerful	▨	cheerful	▨	+57.14
sad	▨	sad	▨	-62.50
shocked	▨	shocked	▨	-42.86
annoyed	▨	annoyed	▨	-58.04
nervous	▨	nervous	▨	+7.14
overall	▨	overall	▨	+58.93

FIGURE 4.12: Questionnaire values are displayed when hovering over a user ID

Moreover a third column displays the change in value from pre-performance to post-performance with decreasing values being printed in grey. When clicking the ID of an audience member in the table, the window containing the questionnaire disappears and the participant's GSR response curve is displayed above the overall GSR response curve. The response curves and the chart for closeness of participants can be seen in Figure 4.13.

All the charts generated by the platform are dynamically generated SVG (Scalable Vector Graphics) images. The library *Raphaël.js*[6] has been used for generating these SVGs. The sensor data itself is loaded via AJAX from the server after the video element fires the `loadedmetadata` event. The questionnaire data for each participant is also loaded dynamically via AJAX once the user hovers their mouse over a participant's ID in the table.

Moreover, in the final version, an *overview* page has been added. While all the visualisations described in the previous paragraphs show what is happening at the current moment in the video recording, this page displays charts and graphs which describe the entire performance. Figure 4.14 shows a screen shot of this overview page.

It contains the MDS charts outlining clustering of audience members and the similarity of the GSR response for each minute of the performance. In the latter chart, each of the dots can be clicked to be redirected to the video page at that very moment. Moreover,

[6]http://raphaeljs.com/

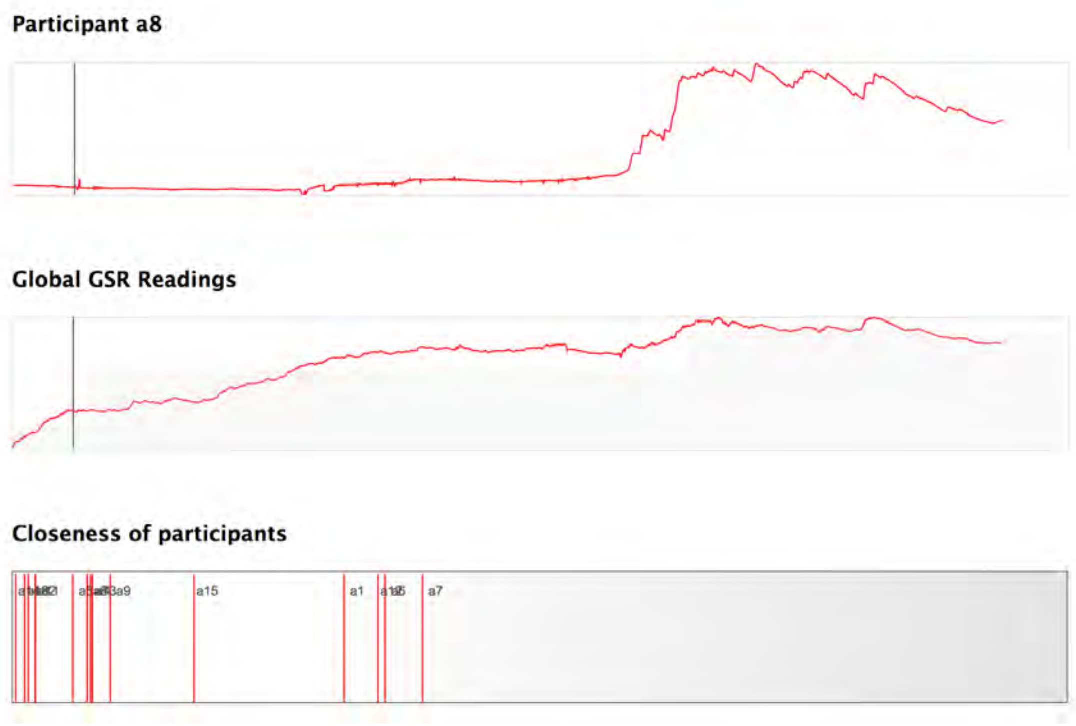


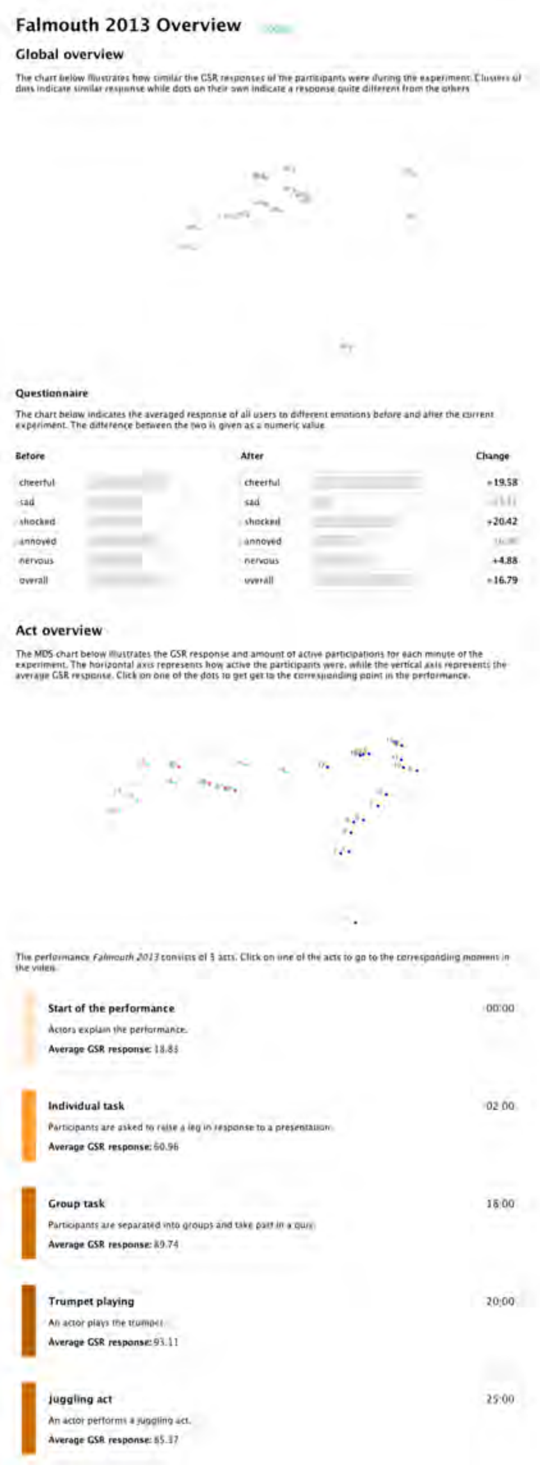
FIGURE 4.13: Individual and overall response curves and closeness of participants

groups of dots have different colours to express them belonging to different acts of the performance.

The overview page also contains the aggregated questionnaire response values over all participants as bar charts. Again, these bar charts are separated in *pre-performance* and *post-performance* questionnaires with a third column displaying the difference between before and after, with negative values being highlighted in grey. Finally, at the bottom of the overview page, there is a list of all the acts of the performance. Each of the acts displays the name, a short description, average GSR response value over the time frame of the acts and a colour indicating the strength of this GSR value, with darker hues indicating stronger response values. Also here, clicking on one of the acts in the list, redirects the user to the corresponding moment in the video recording.

Analogous to the video page, the charts on this page are generated in the user's browser on page load as SVGs using Raphaël.js. The MDS algorithm uses Pearson's product moment correlation as distance metric and for some more advanced matrix algebra a special library called *Numeric.js*[7] has been used.

[7]http://numericjs.com/



Falmouth 2013 Overview

Global overview

The chart below illustrates how similar the GSR responses of the participants were during the experiment. Clusters of dots indicate similar response while dots on their own indicate a response quite different from the others

Questionnaire

The chart below indicates the averaged response of all users to different emotions before and after the current experiment. The difference between the two is given as a numeric value

Before	After	Change
cheerful	cheerful	+19.58
sad	sad	[illegible]
shocked	shocked	+20.42
annoyed	annoyed	[illegible]
nervous	nervous	+4.88
overall	overall	+16.79

Act overview

The MDS chart below illustrates the GSR response and amount of active participations for each minute of the experiment. The horizontal axis represents how active the participants were, while the vertical axis represents the average GSR response. Click on one of the dots to get get to the corresponding point in the performance.

The performance *Falmouth 2013* consists of 5 acts. Click on one of the acts to go to the corresponding moment in the video.

Start of the performance Actors explain the performance. **Average GSR response:** 18.83	00:00
Individual task Participants are asked to raise a leg in response to a presentation. **Average GSR response:** 60.96	02:00
Group task Participants are separated into groups and take part in a quiz. **Average GSR response:** 89.74	16:00
Trumpet playing An actor plays the trumpet. **Average GSR response:** 93.11	20:00
Juggling act An actor performs a juggling act. **Average GSR response:** 85.37	25:00

Figure 4.14: Overview page of the platform

Chapter 5

Evaluation

After a successful implementation and deployment, the platform was evaluated together with potential end users. This was done in three ways: (1) using a questionnaire based on [4] which the testers had to fill out, (2) a JavaScript plug-in which recorded the testers' click locations and scrolls in order to generate heat maps and (3) simple analytics including session times and behaviour flow using *Google Analytics*.

For the evaluation of the platform 30 users with background in performing arts where invited to test it. Each of them had a unique user account numbered from *a1* through *a30* so that we were able to identify them uniquely. The following sections will give a short introduction to these techniques and present the relevant results. For this evaluation process only the *Video* and *Overview* page are considered relevant. Note that at the time of this writing, only five of the testers reported any results.

5.1 Creativity Support Index

5.1.1 Background

The *Creativity Support Index* is a metric specifically designed to evaluate the ability of a tool to support a user's creative process. The author provides a Java-based implementation of the relevant questionnaire at [28]. A detailed explanation of the metric can be found in [4] and [29]. Moreover, a concrete application of it and the results are outlined in [2].

The testers of the platform were sent the Java application alongside some basic instructions, asked to test the platform and fill out the questionnaire contained in the Java application. The questionnaire asked the testers to rate different aspects of their experience. The aspects rated by the users are as follows:

- Results Worth Effort
- Exploration
- Collaboration
- Immersion
- Expressiveness
- Enjoyment

The results for each of these factors are presented hereafter alongside some explanation. The presented results come from five different users, three of which work on theater and the other two involved with music. So all of them could be potential end users. The exact questions and screen shots of the Java application that was sent to the testers can be found in Appendix B.

5.1.2 Results

In the evaluation of the video interface, the testers rated the factors of *Exploration* and *Collaboration* the highest, whereas the remaining factors did not seem as important to them. The average score for the video page is 57.6 with a standard deviation of 23.14, whereas for the overview page we have 44.9 and a standard deviation of 17.14. This indicates that the video page is slightly better at supporting the creative process.

Each of the factors is evaluated with an average factor count, meaning how many times a tester chose the factor as being important to the task at hand, an average factor score, which indicates how well the platform supports the task for this factor and finally a weighted score, which is simply the product of the scores mentioned before.

5.1.2.1 Video Page

For the video page, the testers rated *Exploration* and *Collaboration* the highest as factor count. For the average factor score, the factors *Enjoyment*, *Exploration* and *Results Worth Effort* were rated highest. For the weighted score, this results in *Exploration* and *Collaboration* having the highest scores. The exact results are again outlined in Table 5.1.

Factor	**Avg. Count**	**Avg. Score**	**Avg. Weighted Score**
Results Worth Effort	1.4	12.42	21.4
Exploration	5.2	13.88	73.2
Collaboration	3.8	9.4	33.26
Immersion	1.8	7.14	14.22
Expressiveness	1.6	6.84	9.9
Enjoyment	1.4	14.72	20.9

Table 5.1: CSI scores for the video page

5.1.2.2 Overview Page

For the overview page, the overall score is lower than for the video page. The average factor score shows that the testers rated *Results Worth Effort*, *Exploration* and *Enjoyment* the highest and again, the wieghted scores rate *Exploration* (54.94) and *Collaboration* (26.28) the highest. The exact results can be seen in Table 5.2.

Factor	**Avg. Count**	**Avg. Score**	**Avg. Weighted Score**
Results Worth Effort	1.4	9.48	19.38
Exploration	5.2	10.68	54.94
Collaboration	3.8	6.88	26.28
Immersion	1.8	4.8	8.72
Expressiveness	1.6	5.98	10.3
Enjoyment	1.4	10.9	15.08

Table 5.2: CSI scores for the overview page

5.1.2.3 Comparison

The average factor count for the factor *Results Worth Effort* is rather low, indicating that this factor is not very important to the testers. However, the score for the video page is higher, indicating that users felt it worth exploring more than the overview page. This is also mirrored in the heat maps, which are explored in the next section.

For the factor *Exploration* both the video and the overview page score the highest, indicating that users felt this factor to be very important. This indicates that the platform allows the user to explore different ideas, options and outcomes.

The score *Collaboration* was rated by the testers as being of moderate importance. Again, the video page receives a higher score, possibly also because it includes the option to add annotations and view other people's annotations.

Finally the remaining factors *Immersion*, *Expressiveness* and *Enjoyment* were rated rather low, with none of them going above a score of 2. All of these factors might indicate that the users did not immediately feel at home when using the interface and were slightly confused by some of the elements. Overall, we can say that the platform is able to help artists to explore new ideas and designs and collaborate with others. The video page was felt to be more useful to the artists, probably also because it allows for more interactivity. In the next section we will explore the clicks and scroll events for a selected group of testers in more detail to gain a deeper insight into their behaviour.

5.2 Click Heat Maps

5.2.1 Background

For the evaluation process, the platform was also fitted with a JavaScript plug-in, which recorded click and scroll actions and reported them to another web service via a *WebSocket* connection which aggregated the data.

Each click and scroll action generates an event in the browser. The plug-in catches each of those events and compiles a *JSON* message containing the user's account name, IP address, port, size of the browser window and the document size as well as the current URL and session times. The JSON messages are sent to another web service which listens for these messages and stores them in a *MongoDB* database. The same web service is also used to extract the resulting click heat maps for specific users and URLs. These heat maps are then fused with screen shots of the running platform obtained using an automated headless instance of the *WebKit* browser engine called *CasperJS*[1]

[1]http://casperjs.org/

to be able to replicate the tester's browser window size. In the upcoming sections, we will take a look at some interesting heat maps.

5.2.2 Results

The results are presented separately for *Video* and *Overview* page. Table 5.3 illustrates the number of click events done by testers on the video page and Table 5.4 illustrates the scroll events. This includes all events generated by the testers using their personal user accounts on the live version of the platform.

Page	**Number of Events**
Video	717
Overview	58

Table 5.3: Click events per page

One can immediately see that there are a lot more click events on the video page of the platform, but this is also due to the fact that there are a lot more clickable items on this page. Moreover, the fact that the video page is the first page the users see and the overview page is only reachable *via* the video page may come into play.

Page	**Number of Events**
Video	779
Overview	305

Table 5.4: Scroll events per page

A similar pattern can be seen for for the scroll data. More scroll events were registered for the video page, but every user visited both pages and tried out most of the components.

5.2.2.1 Video Page Heat Maps

Much more interesting than the number of clicks and scrolls are the actual heat on each of the pages in question. Figure 5.1 shows the click heat map for tester *a1* on the video page.

One can see that the tester interacted with more or less all the elements of the user interface. He investigated the individual response curves for all the experiment's participants and also used the curves to seek to specific points in the video. However, it seems that he did not use the global response curve for seeking the video as much. This may

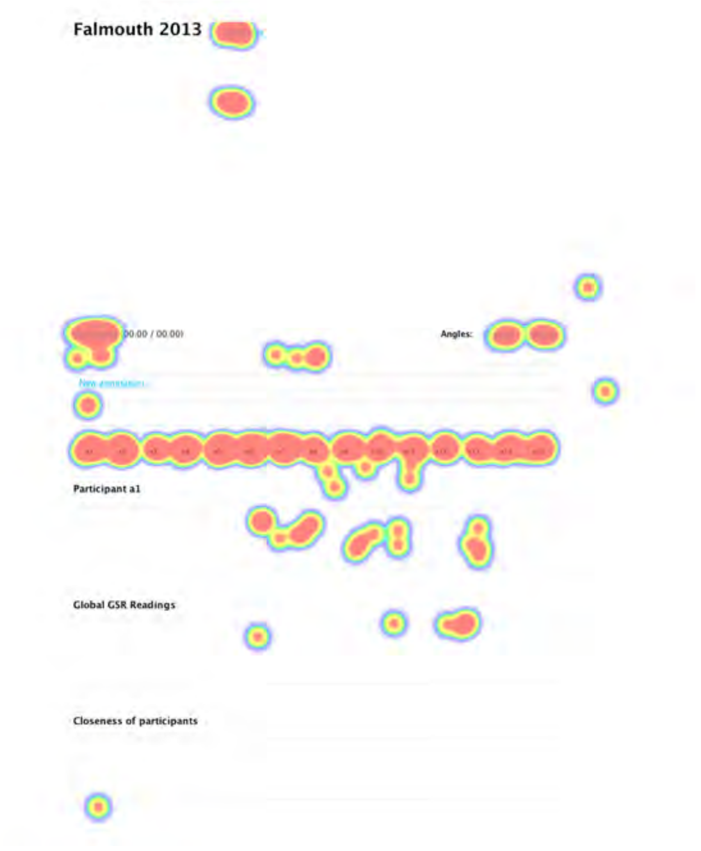


FIGURE 5.1: Click heat map for tester *a1* on the video page

be due to the tester's browser window size which, according to the data was 1263 pixels in width and 668 pixels in height. A height of 668 pixels can barely contain the video and necessitates a lot of scrolling when interacting with the page. This is also mirrored by the number of scroll events, which is with 102 for this tester rather high.

Much more interaction with the line charts containing the global response data can be seen by tester *a2* in Figure 5.2. The tester also looked at all the individual response curves and used them for seeking. The tester also used the annotation bar for seeking. Interestingly, the clicks on the global GSR response chart are concentrated towards the end of the timeline, where the curve reaches its peak. This tester's browser window was with 1663 times 925 pixels about 300 pixels than the window of the previous tester, thus covering the video area till down to the individual response chart. With 31 scrolls, there are also significantly less scroll events for this user. It is also interesting to see that the tester tried to click the chart titled *Closeness of participants* several times, even though it is not clickable. This could mean that users are slightly confused by it, since the other charts, having the same format are clickable.

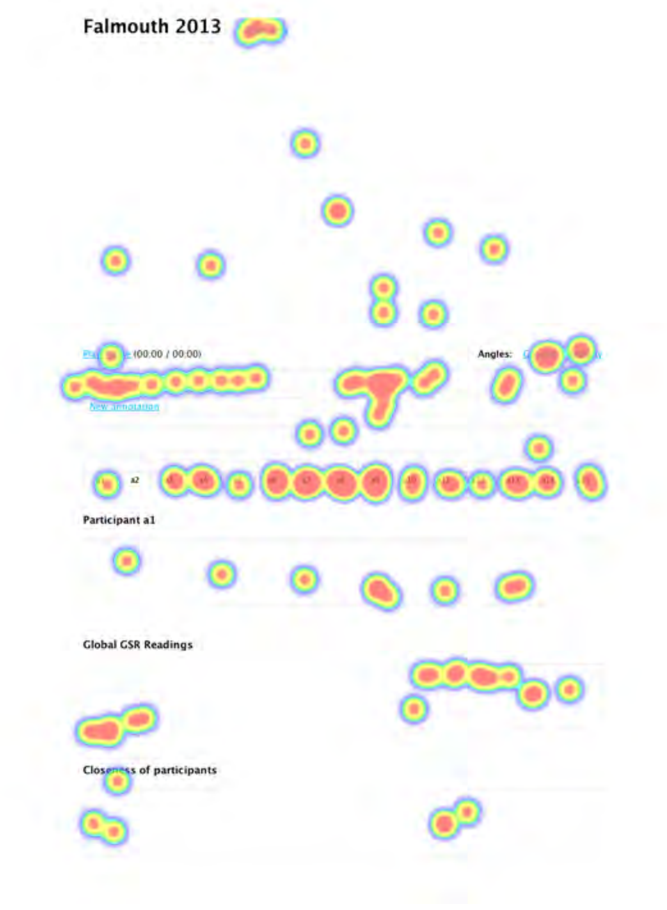


FIGURE 5.2: Click heat map for tester *a2* on the video page

Finally, Figure 5.3 displays the click heat map for tester *a3*. What is interesting to note here is that the tester seemed to have clicked at seemingly random places. There is certainly no patterns which is as clear as for the previous testers, but we can say that the tester did use the seek bar containing the annotations, investigated some of the individual individual response curves and tried to switch camera angles. One possible reason for the erratic pattern of clicks may me that the tester was using a mobile device such as a tablet and touch and swipe events were interpreted as clicks by the tracking plug-in. However, this is difficult to say as the browser's user agent cannot be traced by the plug-in.

5.2.2.2 Overview Page Heat Maps

Finally, we will take a look at the heat maps for the *Overview* page. As already mentioned earlier, this page does not contain as many clicks as the video page. But is probably also due to fact that there are not as many clickable items on the overview

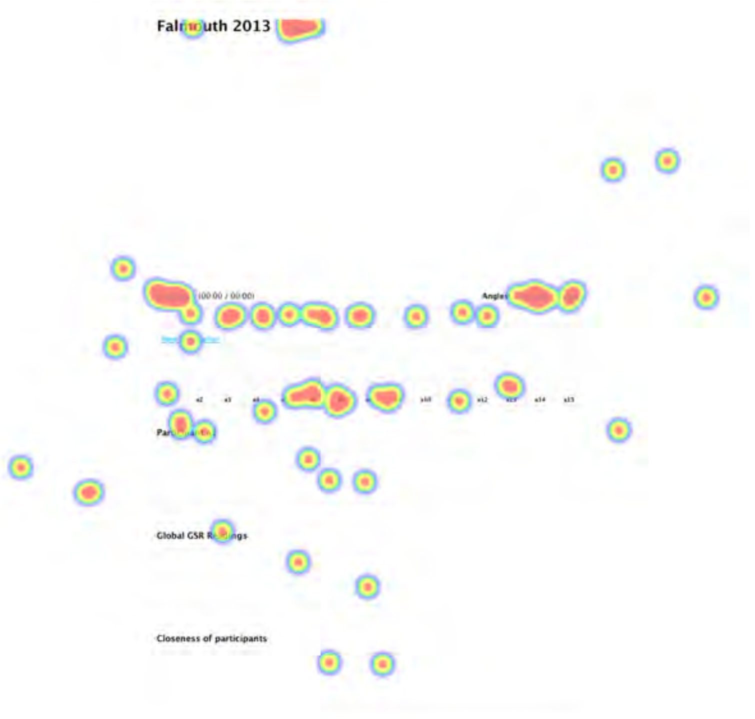


FIGURE 5.3: Click heat map for tester *a3* on the video page

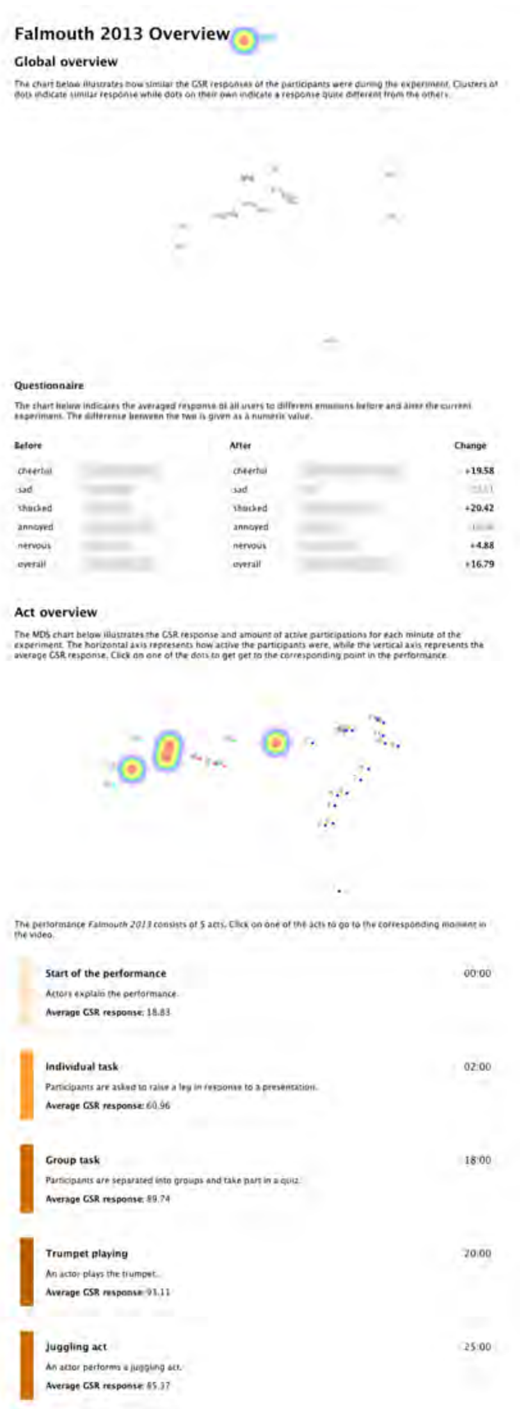
page. In this section we will take a look at where the users actually clicked during their interaction.

In Figure 5.4, we see the click heat map for tester *a1* on the overview page. Immediately noticeable is the significantly decreased number of clicks on this page. The tester only briefly interacted with the MDS chart displaying the aggregated GSR response for each minute of the performance. They did not interact with the act overview.

A similar pattern can be seen for tester *a2* in Figure 5.5. This tester also briefly interacted with the act overview and tried to click on the MDS chart displaying the participant clusters, even though this element is not clickable.

Slightly more interaction with the act overview can be seen from tester *a3* in Figure 5.6. They clicked on several of the act elements but also tried to click on the questionnaire overview, which is itself not clickable. This tester did not interact with the MDS charts at all and we observe the same, seemingly erratic click patterns on the sides. Again, this may be touch events from a mobile device or tablet faultily interpreted as clicks.

All in all, judging by the number of clicks and the resulting heat maps, we can say that there was much more interaction with the video page. However there is also much



Falmouth 2013 Overview

Global overview

The chart below illustrates how similar the GSR responses of the participants were during the experiment. Clusters of dots indicate similar response while dots on their own indicate a response quite different from the others.

Questionnaire

The chart below indicates the averaged response of all users to different emotions before and after the current experiment. The difference between the two is given as a numeric value.

Before	After	Change
cheerful	cheerful	+19.58
sad	sad	[illegible]
shocked	shocked	+20.42
annoyed	annoyed	[illegible]
nervous	nervous	+4.88
overall	overall	+16.79

Act overview

The MDS chart below illustrates the GSR response and amount of active participations for each minute of the experiment. The horizontal axis represents how active the participants were, while the vertical axis represents the average GSR response. Click on one of the dots to get get to the corresponding point in the performance.

The performance *Falmouth 2013* consists of 5 acts. Click on one of the acts to go to the corresponding moment in the video.

Start of the performance 00:00
Actors explain the performance.
Average GSR response: 18.83

Individual task 02:00
Participants are asked to raise a leg in response to a presentation.
Average GSR response: 60.96

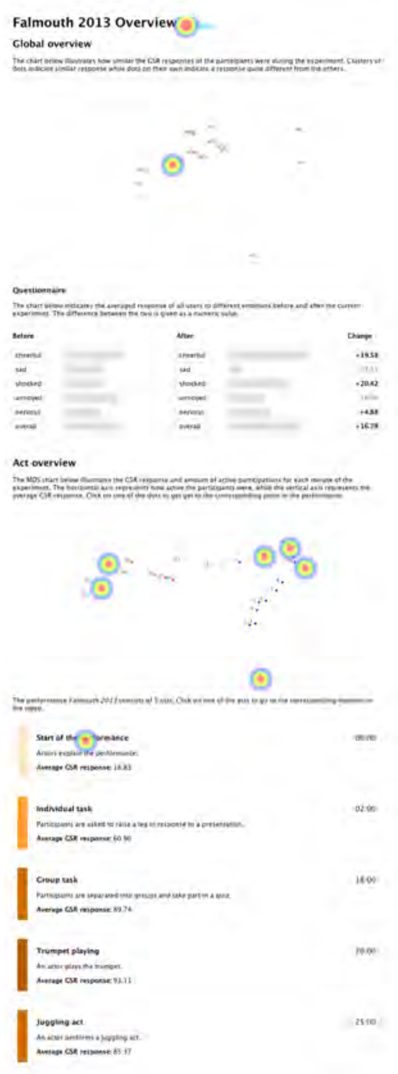
Group task 18:00
Participants are separated into groups and take part in a quiz.
Average GSR response: 89.74

Trumpet playing 20:00
An actor plays the trumpet.
Average GSR response: 93.11

Juggling act 25:00
An actor performs a juggling act.
Average GSR response: 85.37

FIGURE 5.4: Click heat map for tester *a1* on the overview page

more to interact with on that page compared to the overview page. We have also seen that some users tried to click on elements which are not clickable, which might be an indicator that some patterns of the user interface are not immediately clear.


Falmouth 2013 Overview

Global overview

The chart below illustrates how similar the GSR responses of the participants were during the experiment. Clusters of dots indicate similar response while dots on their own indicate a response quite different from the others.

Questionnaire

The chart below indicates the averaged response of all users to different emotions before and after the current experiment. The difference between the two is given as a numeric value.

Before	After	Change
cheerful	cheerful	+19.58
sad	sad	[illegible]
shocked	shocked	+20.42
annoyed	annoyed	[illegible]
nervous	nervous	+4.88
overall	overall	+16.78

Act overview

The MDS chart below illustrates the GSR response and amount of active participations for each minute of the experiment. The horizontal axis represents how active the participants were, while the vertical axis represents the average GSR response. Click on one of the dots to get get to the corresponding point in the performance.

The performance *Falmouth 2013* consists of 5 acts. Click on one of the acts to go to the corresponding moment in the video.

Start of the performance 00:00
Actors explain the performance.
Average GSR response: 16.83

Individual task 02:00
Participants are asked to raise a leg in response to a presentation.
Average GSR response: 60.90

Group task 18:00
Participants are separated into groups and take part in a quiz.
Average GSR response: 89.74

Trumpet playing 20:00
An actor plays the trumpet.
Average GSR response: 93.11

Juggling act 25:00
An actor performs a juggling act.
Average GSR response: 85.37

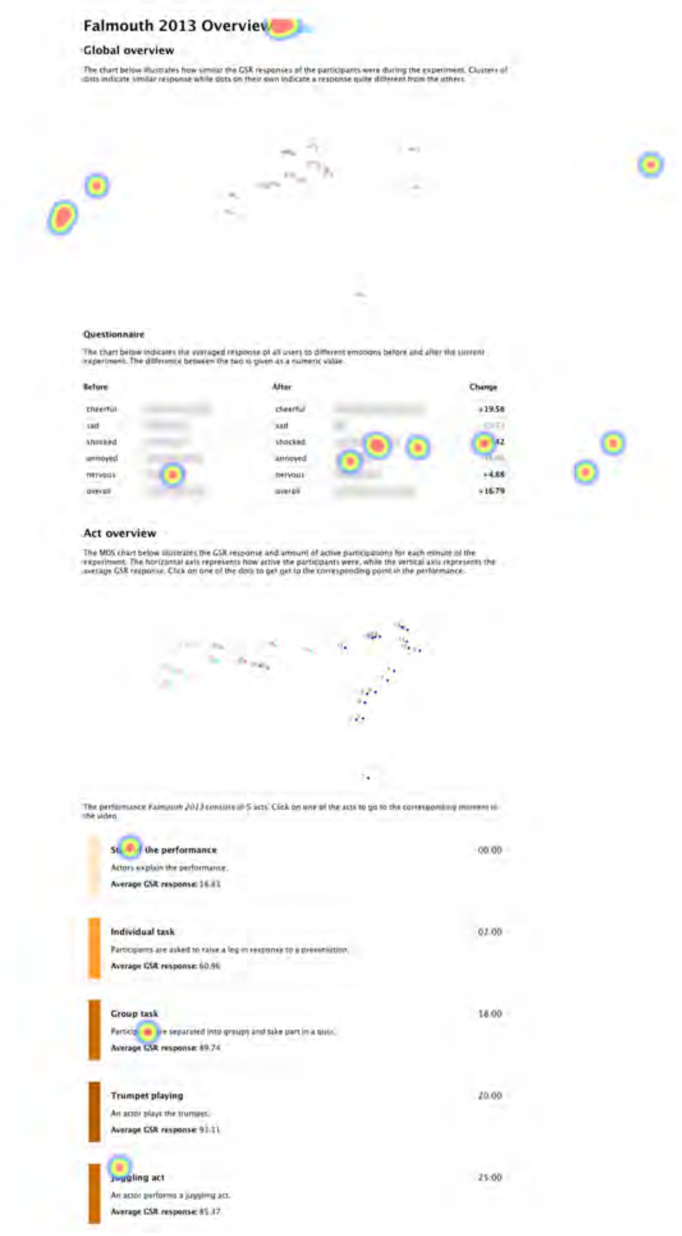
FIGURE 5.5: Click heat map for tester *a2* on the overview page

5.3 Google Analytics

Finally, the platform was also fitted with *Google Analytics*[2] to gain some more general information about the users such as browser, operating system, demographics and general behaviour.

During the testing period, the platform handled 147 user sessions from 111 users and a total of 654 page views. The testers visited 4.45 pages per session, likely going back and forth between video and overview page several times. The average session time was 2 minutes and 34 seconds. However this includes login, registration, welcome page and

[2]http://www.google.com/analytics



Falmouth 2013 Overview

Global overview

The chart below illustrates how similar the GSR responses of the participants were during the experiment. Clusters of dots indicate similar response while dots on their own indicate a response quite different from the others.

Questionnaire

The chart below indicates the averaged response of all users to different emotions before and after the current experiment. The difference between the two is given as a numeric value.

Before	After	Change
cheerful	cheerful	+19.58
sad	sad	[illegible]
shocked	shocked	[illegible]42
annoyed	annoyed	[illegible]
nervous	nervous	+4.88
overall	overall	+16.79

Act overview

The MDS chart below illustrates the GSR response and amount of active participations for each minute of the experiment. The horizontal axis represents how active the participants were, while the vertical axis represents the average GSR response. Click on one of the dots to get get to the corresponding point in the performance.

The performance *Falmouth 2013* consists of 5 acts. Click on one of the acts to go to the corresponding moment in the video.

St[illegible] the performance Actors explain the performance. **Average GSR response:** 16.83	00:00
Individual task Participants are asked to raise a leg in response to a presentation. **Average GSR response:** 60.96	02:00
Group task Particip[illegible]re separated into groups and take part in a quiz. **Average GSR response:** 89.74	18:00
Trumpet playing An actor plays the trumpet. **Average GSR response:** 93.11	20:00
[illegible]ggling act An actor performs a juggling act. **Average GSR response:** 85.37	25:00

FIGURE 5.6: Click heat map for tester *a3* on the overview page

the page containing the list of experiments. The upcoming sections will provide some more insights into the data gathered from *Analytics*.

5.3.1 Results

Figure 5.7 show the *behaviour flow* for the page that the users tested. This means which path through the interface they took and were they dropped off. From the image we see that, unsurprisingly, most people enter the site through the login page and go from there to the list of experiments or the page of the only experiment which was available to the

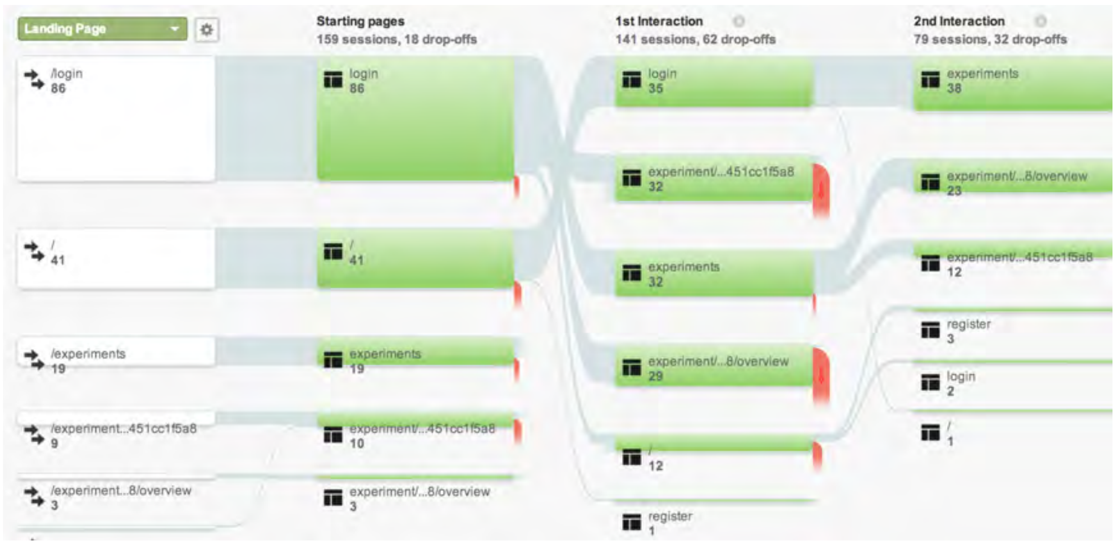

Landing Page
Starting pages
159 sessions, 18 drop-offs
1st Interaction
141 sessions, 62 drop-offs
2nd Interaction
79 sessions, 32 drop-offs
/login 86
login 86
login 35
experiments 38
experiment/...451cc1f5a8 32
experiment/...8/overview 23
/ 41
/ 41
experiments 32
experiment/...451cc1f5a8 12
register 3
/experiments 19
experiments 19
experiment/...8/overview 29
login 2
/experiment...451cc1f5a8 9
experiment/...451cc1f5a8 10
/ 12
/ 1
/experiment...8/overview 3
experiment/...8/overview 3
register 1

FIGURE 5.7: Behaviour flow as gathered by Google Analytics

testers (second column, first row). A bit more surprising is the fact that there is a high drop off rate at the experiment page (third column, second row). This means that users on the video page are more likely to leave the page than to navigate to the overview page. This behaviour is also confirmed by the heat maps in the previous section. Figure 5.8 highlights this behaviour in greater detail again: The traffic from the experiment page drops either off (44.4%) or goes back to the list of experiments (22.2%), while only one third (33.3%) of the sessions visit the overview page. This could indicate that the link to the page is not visible enough or that the users simply were not interested in it.

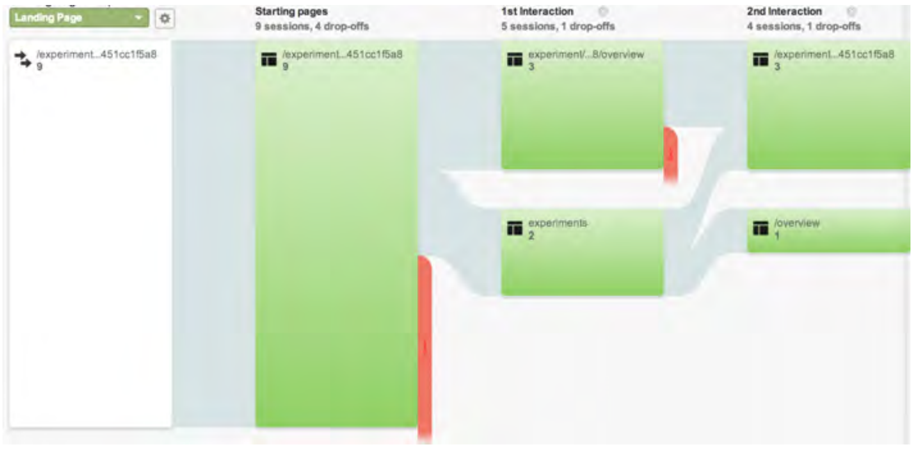


FIGURE 5.8: Behaviour flow specifically for the experiment page

Next, some general statistics about the page: Table 5.5 shows the top pages and the share of page views each of them received. Unsurprisingly, the experiment page received with 21.43% the most page views, the list of experiments with 19.5% as a close second.

This makes sense since a user needs to go the list of experiments first before seeing the experiment itself. After that, the experiment should be in the tester's browser history and can be accessed directly. Also a reload of the page triggers a new page view.

#	Page	# Views	% Views
1	`/experiments/534d47e8e1ee32451cc1f5a8`	155	21.43%
2	`/experiments`	141	19.5%
3	`/login`	132	18.26%
4	`/`	99	13.69%
5	`/experiments/.../overview`	72	10%

Table 5.5: Ranking of page views for the final platform

Also the login page was with 18.26% visited rather often. This makes also sense since a user needs to log in before they are able to view any experiments. After that they only need to log in again after the session cookie in the browser expires.

Rather far off are the welcome page with 14.69% and the overview page with 10.0% of views. This is also mirrored by the behaviour flow diagrams in the previous section. Also note that pages with less page views than those displayed in the table have been omitted as they add no value to the results.

Session Duration (s)	# Sessions	# Pageviews
10-30	7	31
31-60	6	42
61-180	4	64
181-600	9	78
601-1800	14	198
1801+	3	37

Table 5.6: Distribution of session durations from Google Analytics

Finally, it is worth looking at some engagement statistics as gathered by *Analytics*. First of all, Table 5.6 displays the session duration for the test pages. While the average session duration over all the pages is with roughly two minutes rather low, this table provides a bit more insight. We can actually see that a majority of sessions spent something between 10 and 30 minutes on the page and only 13 sessions spent less than one minute on the page.

Similarly, Table 5.7 illustrates the distribution of the page depth (i.e. how many distinct pages were visited). From the table we can see that during most sessions users actually visited two pages. In line with the other results this is most likely the list of experiments and the video page of one of the experiments. The seconds largest share is

Page Depth	**# Sessions**	**# Pageviews**
1	17	17
2	61	122
3	33	99
4	17	68
5	5	25
6	4	24
7	4	28

Table 5.7: Distribution of page depth from Google Analytics

taken by session which visited three pages, with 33 sessions. This is most likely the list of experiments, the video page of the available experiment and lastly the overview page of the experiment.

Perhaps also interesting are the values for page depth one. These sessions likely never got past the login page. Also the number of sessions for page depth four is surprisingly high. These are most likely users which navigated multiple times between the video page and the overview page.

5.4 Issues

This section outlines some issues with the evaluation process. First and foremost, the testers were not given any instructions on how to evaluate the platform. They were given an account name and a password and told to explore the pages and then fill out the questionnaire.

The results might have been more conclusive if the testers had been given more specific tasks, such as trying to add and delete annotations or try to identify some performance highlights using the MDS charts on the overview page. This process could have been supported by more specific questionnaires in addition to the CSI survey to assess how easy the tasks were to accomplish.

There were also some minor issues with the heat maps. First of all, the system exhibited a couple of instabilities, but these can be fixed given more time and some testing. Moreover, the heat map generation plug in did not collect some data which would have been interesting, such as data about the user's browser and operating system. But also these issues are easy to address in a future version of the heat map generation plug in.

Chapter 6

Conclusion

To conclude, in this session we will highlight a couple of issues which exist with the system, either coming from features which are not yet implemented or general issues related to system or performance constraints. Furthermore, this section provides some outlook as to where this platform could go in the future, how it could be improved or how it could be used in an actual environment.

6.1 Lessons Learned

Video Codec Support One issue, which was present from the very start of the development was with the video recordings. The platform uses the HTML5 `<video>` tag, which on the one hand provides a standardised way of displaying video on a website which works in any reasonably modern browser. The problem is that different browsers support different video encoding formats and there is hardly any overlap between them. Table 6.1 shows an outline of different video formats supported by different browsers. This table alongside further explanations can be found at [30].

For this final implementation of the platform the *H.264* codec with MP4 has been chosen. From the table we can see that Google Chrome, Apple Safari and Microsoft Internet Explorer support this codec. However, Safari only supports it through QuickTime. But given that Internet Explorer and Chrome are the two browsers with the highest market share at the moment[1], choosing a format that they both support makes the most sense.

[1] `http://netmarketshare.com/browser-market-share.aspx?qprid=0&qpcustomd=0`

	Theora	**H.264**	**WEBM**
Google Chrome	5.0+	5.0+	5.0+
Mozilla Firefox	3.5+		4.0+
Internet Explorer		9.0+	9.0+
Safari		3.0+	
Opera	10.5+		10.6+

Table 6.1: Video codecs supported by different browser vendors

The most obvious solution to this problem is given by encoding the video files in different formats and offering them to the browser in the video tag. The user's browser will then select the video encoding most appropriate for it. The problem with this approach is that the files for each camera angle need to be exported several times, which could cause issues with storage space and also are the paths where the video files are stored hard coded in the database. So the schema would have to be changed in that case as well. However, the main problem with multiple versions of each video file is the switching of camera angles.

During some tests the switching procedure did not work as well as it does now, since the browser's JavaScript engine does not know which video format to switch to. This issue could be circumvented by implementing the entire switching and video loading procedure in the browser and perform some detection of supported video formats. This would however involve a substantial rewrite of the code base related to video loading and switching.

Scalability Another issue is the one of scalability. The platform has only really been tested with a performance containing data points from 15 audience members over a time frame of 28 minutes. Given that all the calculations are performed in the user's browser and that the entirety of the data set needs to be transferred from the server to the client every time, this might cause some problems as the number of participants increases. For the experiment currently available for testing the sensor data for the 15 participants is 270 KB in size, which amounts to 18 KB per user for 28 minutes or $\sim 640B$ per user per minute. Considering for instance a movie performance of 2 hours and 40 audience members connected to GSR sensors with a sample rate of 1 seconds, we will get roughly 3 MB of sensor data. While the size of the data is still manageable, given modern broadband connections and also taking into account that the video files are up to two orders of magnitude larger than this data, it is most likely the processing times on the client side which will cause problems.

This is especially grave on the overview page, where the MDS algorithm requires some more involved matrix algebra and calculation of other metrics which with the current experiment of 15 participants take $\sim 147.15ms$, which is more than 50% of the entire page load time. These issues can either be circumvented by the implementation of more sophisticated algorithms to reduce complexity or preprocessing on the server. Preprocessing and caching on the server side are likely to have the most payoff, as it reduces the amount of data that needs to be shipped to the client side. Moreover the server could cache the results for an experiment in an in-memory key-value store like *Redis*[2] as they are not going to change anyway.

6.2 Future Work

During the test phase with potential end users we have discovered a couple of issues with the tool. Even though the amount of data from the testers may not be sufficient yet, we can say that the user interface in general could use some improvements. Some of the testers had a small screen resolution and were not able to see the most important elements of the page at the same time. Also did they try to click elements which are not clickable. Moreover, the overview page was not visited as much as the video page and received lower ratings in the CSI score too. This implies that the testers did not feel as much engaged with the overview page. Some more and different tests need to be run in order to improve this issue.

Another thing which can be done includes the addition of *smarter* visualisations. One could for example train a machine learning algorithm on the shape of the curve of an audience member and provide thusly more meaningful conclusions on these curves, i.e. one could say more easily whether an audience member enjoyed the performance or if they got distracted at some point.

Finally, in order to be used by performing artists to assess actual performances, the platform requires one more feature, which has not been implemented due to time constraints. Namely, the possibility for the artists to add new experiments themselves. The

[2]`http://www.redis.io`

artists would upload the video files, the sensor data and the questionnaire data and the platform would generate all the pages in the same way as they are available right now for the sample experiment. This however is a more involved process and especially the video upload might cause some issue due to file sizes and file formats. The artists would also have to make sure the sensor readings are perfectly synchronised with the video recordings and are formatted in the right way.

6.3 Conclusion

In conclusion, this work was a guide through the implementation process of a platform which allows performing artists to visualise their performances in an enhanced way by adding sensor and questionnaire data, which allow them to gain a better insight into how the audience perceives the performance and how engaged they are. In a review of relevant literature we have seen different types of physiological sensors and how they are used. Special attention has been given to GSR sensors, which are the sensors which were used in previous experiments with audiences and performances.

We have used the data from these previous experiments to get a feel for the data and what can be done with it. Moreover, based on this data and some video recordings a basic prototype was developed, which was presented to artists, which acted as end users. From these interviews, we gathered a more formal and complete set of requirements on top of which the final platform was developed.

The resulting platform makes use of modern web technologies to enable the artists to gain deeper insight into their performances. For evaluation purposes, the platform was tested by potential end users and their behaviour was tracked. The resulting data gave us some insight how the artists explored the platform and gave us some hints how it could be improved.

We can say that in general the platform was perceived well by the artists and they saw themselves using such a tool for the purpose of improving their performances. But obviously, there is also a negative side to it. If we are able to assess the quality of every minute of a theater performance, someone could use it as a means to decide whether a play is worthwhile playing or not, based on the misconception that a low response value implies a bad performance.

Lastly, we highlighted some issues, mostly related to video encoding and scalability. Moreover, we suggested some ways to improve the platform and make it more intuitive for artists to use and suggested some missing features, which could still be implemented to make the platform more complete.

With some ideas in mind, on how the platform can be improved to make it more usable and most importantly also make it usable in a real environment, we conclude this work and come back to reflect on the opening paragraph. Even though the emotional response can be quantified using technology, the final say about the quality of the performance should be judged by the strength of the final applause.

Appendix A

R Source Code

This section contains all the *R* source code which was written in the preliminary data exploration phase. The code normalises the raw input data and generates several plots.

```r
sensors = read.csv('sensors.csv')

max = max(sensors)
min = min(sensors)

snorm = sensors

# data normalisation
snorm$a1 = ((sensors$a1 - min(sensors$a1)) / (max)) * 100
snorm$a2 = ((sensors$a2 - min(sensors$a2)) / (max)) * 100
snorm$a3 = ((sensors$a3 - min(sensors$a3)) / (max)) * 100
snorm$a4 = ((sensors$a4 - min(sensors$a4)) / (max)) * 100
snorm$a5 = ((sensors$a5 - min(sensors$a5)) / (max)) * 100
snorm$a6 = ((sensors$a6 - min(sensors$a6)) / (max)) * 100
snorm$a7 = ((sensors$a7 - min(sensors$a7)) / (max)) * 100
snorm$a8 = ((sensors$a8 - min(sensors$a8)) / (max)) * 100
snorm$a9 = ((sensors$a9 - min(sensors$a9)) / (max)) * 100
snorm$a10 = ((sensors$a10 - min(sensors$a10)) / (max)) * 100
snorm$a11 = ((sensors$a11 - min(sensors$a11)) / (max)) * 100
snorm$a12 = ((sensors$a12 - min(sensors$a12)) / (max)) * 100
snorm$a13 = ((sensors$a13 - min(sensors$a13)) / (max)) * 100
snorm$a14 = ((sensors$a14 - min(sensors$a14)) / (max)) * 100
snorm$a15 = ((sensors$a15 - min(sensors$a15)) / (max)) * 100

sd(snorm$a4)

# basic line charts
par(mfrow=c(1,1))
plot(snorm$a3, type="l", ylim=c(0,100), xlab="Time (s)", ylab="GSR Response", col="red")
lines(snorm$a4, type="l", ylim=c(0,100), col='blue')

# MDS distance matrix
c = 1 - abs(cor(sensors))
fit = cmdscale(c, eig=T, k=2)

# plot MDS chart
plot(fit$points[,1], fit$points[,2], col='red', xlab="", ylab="")
text(fit$points[,1], fit$points[,2], labels = row.names(x), cex=0.7, pos=2)
```

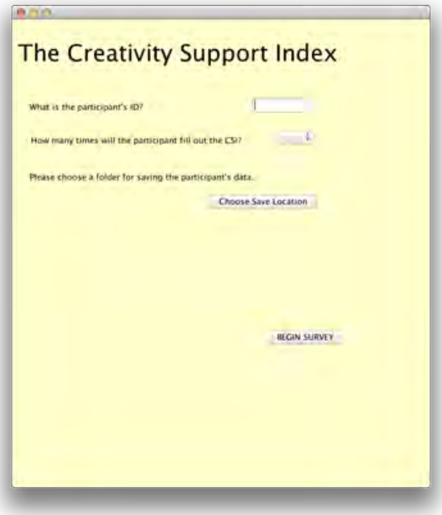
```r

# plot Fourier analysis charts
par(mfrow=c(1,3))

fourier = fft(snorm$a1)
plot(abs(fourier), type='l', ylim=c(0, 500), xlab="Frequency", ylab="fft(a1)")

fourier = fft(snorm$a3)
plot(abs(fourier), type='l', ylim=c(0, 500), xlab="Frequency", ylab="fft(a3)")

fourier = fft(snorm$a4)
plot(abs(fourier), type='l', ylim=c(0, 500), xlab="Frequency", ylab="fft(a4)")
```

Appendix B

CSI Questionnaire

This section contains screen shots of the application which was given to testers to fill out the CSI (Creativity Support Index) questionnaire. The application itself and its source code can be downloaded from http://www.erincherry.net/csi.html (accessed 2014-08-07).

NB: At the time of this writing, this version of the application contains a minor bug, which causes problems when choosing the location for saving the questionnaire results under *Microsoft Windows*. The bug has been fixed and a patch has been submitted to the original author.

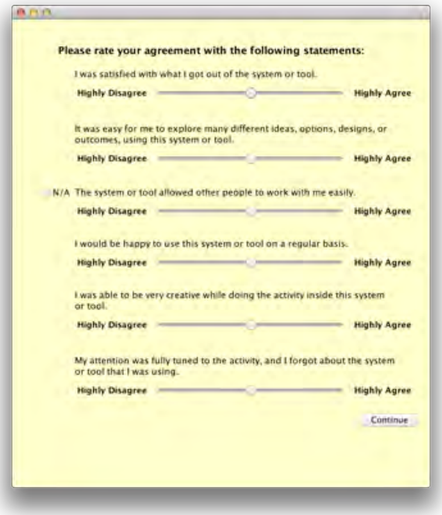


The Creativity Support Index

What is the participant's ID?

How many times will the participant fill out the CSI?

Please choose a folder for saving the participant's data.



Please rate your agreement with the following statements:

I was satisfied with what I got out of the system or tool.
Highly Disagree — Highly Agree

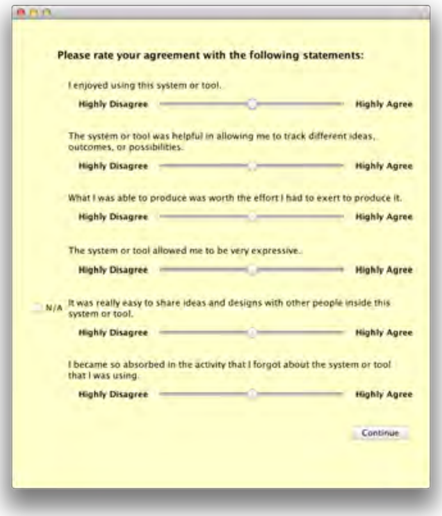
It was easy for me to explore many different ideas, options, designs, or outcomes, using this system or tool.
Highly Disagree — Highly Agree

N/A The system or tool allowed other people to work with me easily.
Highly Disagree — Highly Agree

I would be happy to use this system or tool on a regular basis.
Highly Disagree — Highly Agree

I was able to be very creative while doing the activity inside this system or tool.
Highly Disagree — Highly Agree

My attention was fully tuned to the activity, and I forgot about the system or tool that I was using.
Highly Disagree — Highly Agree


Please rate your agreement with the following statements:

I enjoyed using this system or tool.
Highly Disagree — **Highly Agree**

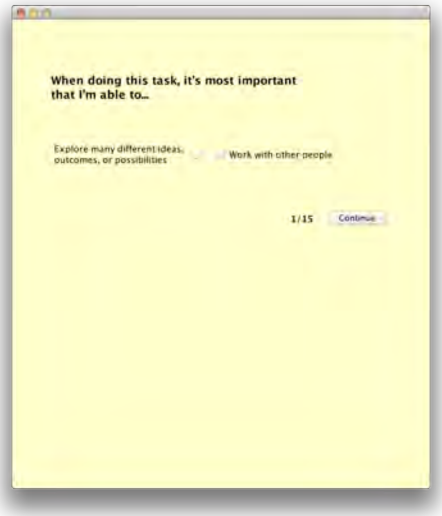
The system or tool was helpful in allowing me to track different ideas, outcomes, or possibilities.
Highly Disagree — **Highly Agree**

What I was able to produce was worth the effort I had to exert to produce it.
Highly Disagree — **Highly Agree**

The system or tool allowed me to be very expressive.
Highly Disagree — **Highly Agree**

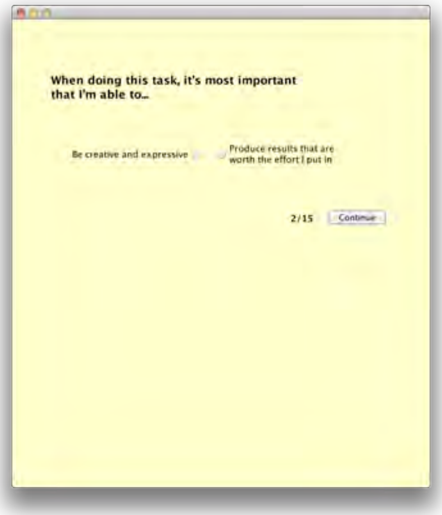
N/A It was really easy to share ideas and designs with other people inside this system or tool.
Highly Disagree — **Highly Agree**

I became so absorbed in the activity that I forgot about the system or tool that I was using.
Highly Disagree — **Highly Agree**


When doing this task, it's most important that I'm able to...

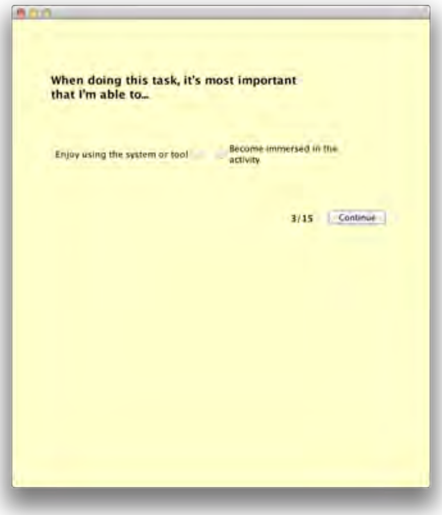
Explore many different ideas, outcomes, or possibilities — Work with other people

1/15


When doing this task, it's most important that I'm able to...

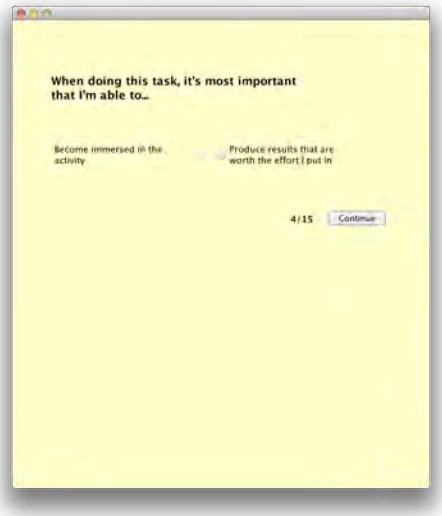
Be creative and expressive — Produce results that are worth the effort I put in

2/15


When doing this task, it's most important that I'm able to...

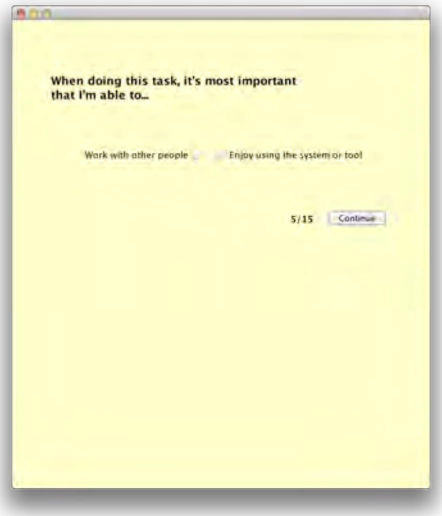
Enjoy using the system or tool — Become immersed in the activity

3/15


When doing this task, it's most important that I'm able to...

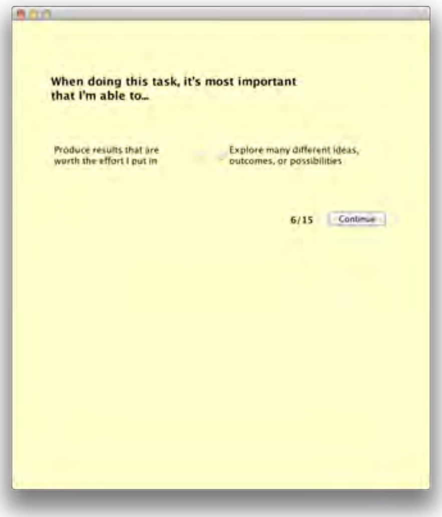
Become immersed in the activity — Produce results that are worth the effort I put in

4/15


When doing this task, it's most important that I'm able to...

Work with other people — Enjoy using the system or tool

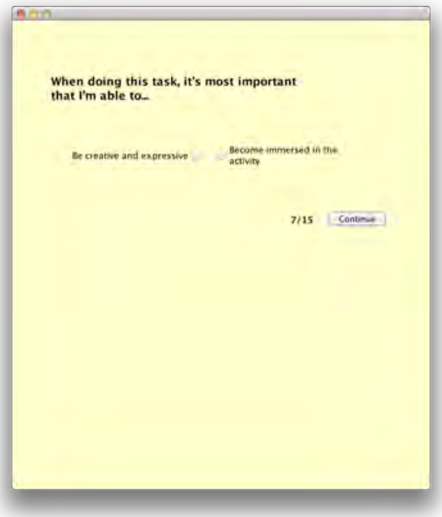
5/15



When doing this task, it's most important that I'm able to...

Produce results that are worth the effort I put in | Explore many different ideas, outcomes, or possibilities

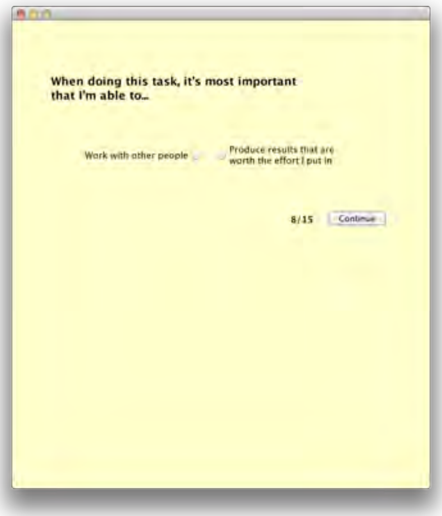
6/15 Continue



When doing this task, it's most important that I'm able to...

Be creative and expressive | Become immersed in the activity

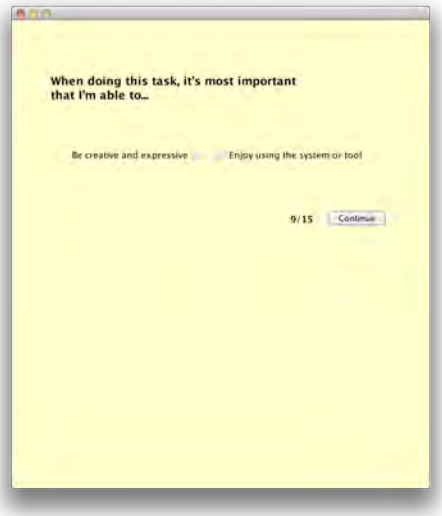
7/15 Continue



When doing this task, it's most important that I'm able to...

Work with other people | Produce results that are worth the effort I put in

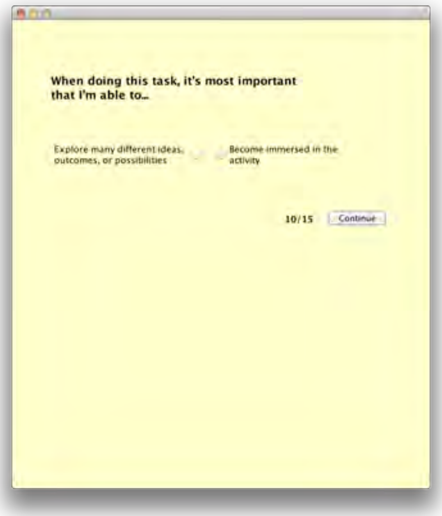
8/15 Continue



When doing this task, it's most important that I'm able to...

Be creative and expressive | Enjoy using the system or tool

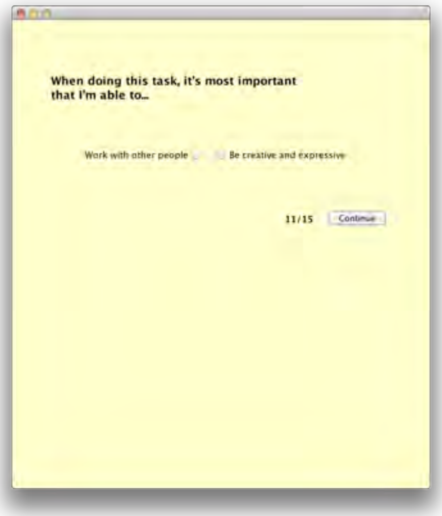
9/15 Continue



When doing this task, it's most important that I'm able to...

Explore many different ideas, outcomes, or possibilities | Become immersed in the activity

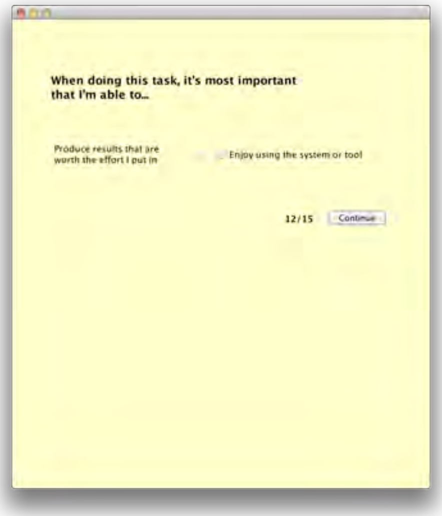
10/15 Continue



When doing this task, it's most important that I'm able to...

Work with other people | Be creative and expressive

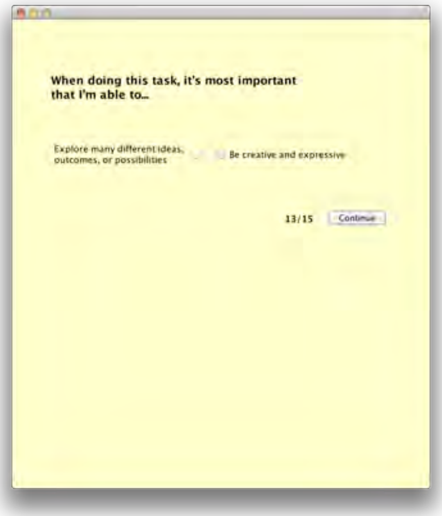
11/15 Continue


When doing this task, it's most important that I'm able to…

Produce results that are worth the effort I put in

Enjoy using the system or tool

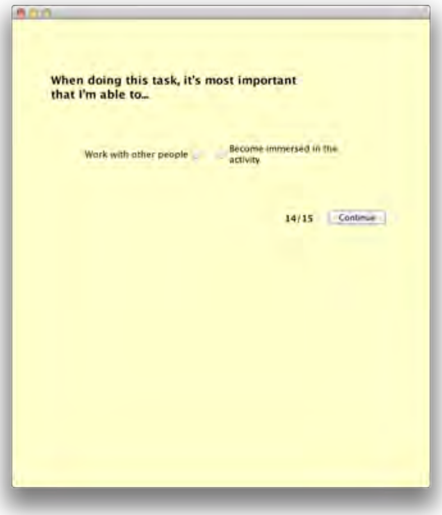
12/15


When doing this task, it's most important that I'm able to…

Explore many different ideas, outcomes, or possibilities

Be creative and expressive

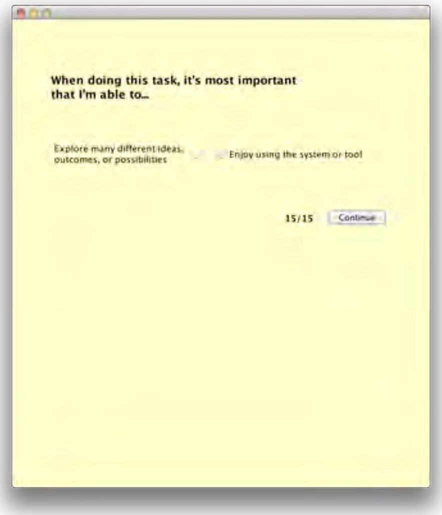
13/15


When doing this task, it's most important that I'm able to…

Work with other people

Become immersed in the activity

14/15


When doing this task, it's most important that I'm able to…

Explore many different ideas, outcomes, or possibilities

Enjoy using the system or tool

15/15

Bibliography

[1] Jennifer Allanson and Stephen H Fairclough. A research agenda for physiological computing. *Interacting with computers*, 16(5):857–878, 2004.

[2] Erin A Carroll and Celine Latulipe. Triangulating the personal creative experience: self-report, external judgments, and physiology. In *Proceedings of Graphics Interface 2012*, pages 53–60. Canadian Information Processing Society, 2012.

[3] Chen Wang, Erik N. Geelhoed, Phil P. Stenton, and Pablo Cesar. Sensing a live audience. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1909–1912, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557154. URL http://doi.acm.org/10.1145/2556288.2557154.

[4] Erin A. Carroll and Celine Latulipe. The creativity support index. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4009–4014, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520609. URL http://doi.acm.org/10.1145/1520340.1520609.

[5] Vikash Singh, Celine Latulipe, Erin Carroll, and Danielle Lottridge. The choreographer's notebook: A video annotation system for dancers and choreographers. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, C&#38;C '11, pages 197–206, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0820-5. doi: 10.1145/2069618.2069653. URL http://doi.acm.org/10.1145/2069618.2069653.

[6] Diogo Cabral, João G. Valente, Urândia Aragão, Carla Fernandes, and Nuno Correia. Evaluation of a multimodal video annotator for contemporary dance. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*,

AVI '12, pages 572–579, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1287-5. doi: 10.1145/2254556.2254663. URL http://doi.acm.org/10.1145/2254556.2254663.

[7] What is eye tracking? http://www.tobii.com/en/about/what-is-eye-tracking/. Accessed: 2014-06-11.

[8] P.J. Lang. The emotion probe: Studies of motivation and attention. *American psychologist*, 50:372–372, 1995.

[9] Celine Latulipe, Erin A Carroll, and Danielle Lottridge. Love, hate, arousal and engagement: exploring audience responses to performing arts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1845–1854. ACM, 2011.

[10] Shaowen Bardzell, Jeffrey Bardzell, and Tyler Pace. Understanding affective interaction: Emotion, engagement, and internet videos. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–8. IEEE, 2009.

[11] Wolfram Boucsein. *Electrodermal activity*. Springer, 2012.

[12] Galvactivator. frequently asked questions. http://www.media.mit.edu/galvactivator/faq.html. Accessed: 2014-06-13.

[13] K Wilke, A Martin, L Terstegen, and SS Biel. A short history of sweat gland biology. *International journal of cosmetic science*, 29(3):169–179, 2007.

[14] Don C Fowles, Margaret J Christie, Robert Edelberg, William W Grings, David T Lykken, and Peter H Venables. Publication recommendations for electrodermal measurements. *Psychophysiology*, 18(3):232–239, 1981.

[15] Regan L Mandryk, M Stella Atkins, and Kori M Inkpen. A continuous and objective evaluation of emotional experience with interactive play environments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1027–1036. ACM, 2006.

[16] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[17] Regan L Mandryk and M Stella Atkins. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International Journal of Human-Computer Studies*, 65(4):329–347, 2007.

[18] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[19] Chen Wang. Unfolding audience response by using physiological sensors. Unpublished, 2014.

[20] Tom W. Calvert, Armin Bruderlin, Sang Mah, Thecla Schiphorst, and Chris Welman. The evolution of an interface for choreographers. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 115–122, New York, NY, USA, 1993. ACM. ISBN 0-89791-575-5. doi: 10.1145/169059.169113. URL http://doi.acm.org/10.1145/169059.169113.

[21] Frank Dabek, Jennifer Healey, and Rosalind Picard. A new affect-perceiving interface and its application to personalized music selection. In *Proc. from the 1998 Workshop on Perceptual User Interfaces*, 1998.

[22] Node.js homepage. http://www.nodejs.org. Accessed: 2014-06-14.

[23] Mongodb homepage. http://www.mongodb.org. Accessed: 2014-06-14.

[24] Scott Hanselmann. Javascript is assembly language for the web: Sematic markup is dead! clean vs. machine-coded html. http://www.hanselman.com/blog/JavaScriptIsAssemblyLanguageForTheWebSematicMarkupIsDeadCleanVsMachinecodedHTML.aspx, 2011. Accessed: 2014-06-14.

[25] JB KRUSKAL, M WISH, Inc Sage Publications, and United States of America. Multidimensional scaling. 1978.

[26] Json specification. http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf. Accessed: 2014-08-05.

[27] Embedded js. http://www.embeddedjs.com. Accessed: 2014-08-08.

[28] Erin Cherry. Creativity support index. http://www.erincherry.net/csi.html. Accessed: 2014-08-01.

[29] Erin A. Carroll, Celine Latulipe, Richard Fung, and Michael Terry. Creativity factor evaluation: Towards a standardized survey metric for creativity support. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, C&#38;C '09, pages 127–136, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-865-0. doi: 10.1145/1640233.1640255. URL http://doi.acm.org/10.1145/1640233.1640255.

[30] Dive into html5 - video codecs. http://diveintohtml5.info/video.html#video-codecs. Accessed: 2014-08-09.