UNIVERSITY OF OSLO
Department of Informatics

Crowdsourcing Subjective Quality Assessment of Multimedia Content

The Multimedia Assessment Tool

Master thesis

Ben Christopher Tomlin

15th November 2013

UNIVERSITAS OSLOENSIS
MDCCCXI

Crowdsourcing Subjective Quality Assessment of Multimedia Content

Ben Christopher Tomlin

15th November 2013

Abstract

Multimedia services are ever-increasing in popularity and are today widely accessed through countless numbers of computers and mobile devices. Service providers strive to supply users with a satisfying experience, regardless of the hardware capabilities or network environments the user is faced with. Considering that every user is different, Quality of Experience (QoE) experiments have been developed to assess what is satisfactory. With the growing potential of crowdsourcing, it is becoming more and more feasible to have an Internet crowd conduct subjective assessments on their personal computers rather than in a traditional laboratory. This opens up for a more diverse set of participants at a lower economic cost. As the main goal is to provide a satisfying end-user experience, there is a strong need for a framework that can measure the quality of multimedia content efficiently and reliably.

In this thesis, we develop and present a crowdsourceable framework for performing subjective quality assessment of multimedia content. While documenting the development process of our framework, we provide a thorough explanation of how it is built and how it works, consequently enabling researchers to run unique experiments according to their needs. The advantages of this framework compared to the traditional studies conducted in controlled environments are many, but we will also highlight the remaining challenges associated with our approach. Building on a solid theoretical framework, we aim to demonstrate that, with our application, researchers can outsource their experiments within multimedia quality assessment to an Internet crowd without risking the quality of the results. Consequently, while providing reliable evaluation, we obtain a higher level of participant diversity at a much lower cost.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors, Ragnhild Eg and Carsten Griwodz, for their valuable feedback and guidance during the work with this thesis. Thanks to the guys at the lab as well, for the moral support and a good environment to work in. Finally, I would like to thank my friends, and especially my parents, for their great support throughout my studies.

Chapter 1

Introduction

Technology in this day and age is evolving at an alarmingly fast rate. To keep up with current standards, developed software must meet users' preferences and needs. Through the last decade, one platform specifically has been formed as the de facto standard to perform many kinds of wide audience studies. The traditional paper-and-pen approach for conducting surveys takes valuable time and resources, especially when dealing with multimedia. Evidently, a more efficient and dynamic form of conducting these types of surveys is needed. The Multimedia Asessment Tool is presented as a reliable alternative to conduct user studies on the now highly accessible and diverse platform we call the Internet.

1.1 Background

Multimedia quality is typically approached in one of two manners. Either through objective metrics that consider a wide range of measured facts, like signal-to-noise ratios [22, 35], or through subjective measures that are based on the opinions of users [7]. The latter is typically referred to as Quality of Experience [33, 30], and is the method of main interest in this project. While objective metrics are powerful in their consistency, the perception and experience of multimedia quality remains highly subjective. Only human opinion can provide feedback, for instance, on which type of distortion is more distracting or how quality perception may change depending on the video content. The conventional method for collecting subjective opinions on multiple items is through user studies. These studies are often conducted using statistical surveys. Unlike a marketing survey, a statistical survey is aimed at a specific area of research. Typically, surveys provide questions to be assessed according to a range of options, frequently in the form of a scale. A rating scale can take many forms, but they all present a range of response options where one or more have to be selected. The Mean Opinion Score (MOS) rating test is an example of an assessment method which uses a typical five point rating scale [28], as seen in Table 1.1. Moreover, MOS is one of the more well-known methods for assessing the QoE of multimedia content. Originally, it was used in telephone networks to obtain the user's opinion of the quality of the network. Listeners would sit in a quiet room and score quality of telephone calls as they perceived it, as explained in detail in the ITU-T Recommendation P.800 [28]. Although it was originally intended for rating quality of telephone networks, it has become a popular test for assessing quality levels and degradations of other multimedia types. Additional methods are also commonly used in multimedia evaluation studies, which we will discuss later in the thesis.

5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 1.1: A typical rating scale, as used in MOS.

1.2 Problem Definition

As technology has evolved and become more accessible and diverse, so has the need to evaluate its quality become more imperative. Consequently, multimedia research is not only concerned with optimising solutions, but also with evaluating what is optimal. Commonly, audio and video clips are presented to a group of users in a controlled environment and participants are then instructed to rate each clip as they go. However, in itself, technology has provided a new platform for assessment studies. The traditional survey method can become cumbersome when dealing with multimedia, emphasising the need for a functional and fluent survey method that implements the media it is designed to evaluate. Moreover, this new platform also reduces the need for the presence of a researcher. Thus, this project aims to develop an online assessment tool for multimedia content where users no longer need to travel to a research facility, but instead can complete the survey when and where they please. The Multimedia Assessment Tool (MAT) therefore has the potential for reaching out to a larger and more varied group of users, providing the tools and the foundation for thorough research adapted to contemporary technology.

The thesis mainly focuses on developing MAT for assessing multimedia content. MAT should be accessible on all major web browsers and include support for running multiple experiments, or surveys, simultaneously. As with most other web applications, it is easily accessible for both experimenters and respondents, through a simple, but extensive, point-and-click graphical user interface. The software must be contained in a central experimentation server and is able to run on most standard web servers with PHP and MySQL installed. The surveys and all the underlying content, like audio/video clips, questions, instructions, rating scales and so forth, are also stored here, including responses and results gathered from users taking part in the surveys. Furthermore, related research on the topics of online statistical surveys, subjective evaluation and methodology is included. For example, why do we want to evaluate multimedia in the first place? Why is quality such an important factor? Research into these topics is essential for the project, as well as important in understanding how this framework should be developed, and to what purpose.

Following is a list of features and functionalities that are to be included in the implementation.

- Creation of new surveys, selection of multimedia content, invitation and unique logins for users, and definition of instructions prior to the commencement of surveys.
- Creation of questions and response options, specification of the type of question and, if applicable, indication of the number of items on a scale.
- Specification of the order of audio/video clips, number of repetitions, grouping, and randomisation within groups and of the groups themselves.
- Collection of responses, response times, and other relevant technical features.
- Output of response data in a comprehensible format, for example spreadsheets, and some basic statistical analyses.

The final outcome is a distributed tool with a graphical user interface, adapted to run in an online web browser, and therefore easily accessible to the vast majority of people.

1.3 Main Contributions

The evaluation framework presented in this thesis aims to provide researchers with an extensive and flexible application that can be used for running a variety of subjective assessment studies on multimedia quality, using the Internet platform. With the current lack of similar tools available to the research community, MAT may prove to be a beneficial addition in running assessment studies, both cost- and time-efficiently. These online studies can be conducted from any where at any time, greatly benefiting participants, while also allowing for a larger and more diverse group of people to take part. Moreover, it should lessen the time and effort required of participants, as well as the experimenter. Thus, MAT should offer a simple and efficient solution for running quality evaluation studies.

1.4 Research Method

Initially we performed research into subjective quality assessment, generally related to multimedia. Traditional assessment studies of this kind showed that the methodology and execution of these studies could be both time-consuming and expensive, so an online approach to this problem revealed itself as a promising alternative. Further research was done into online assessment tools that could provide similar features as traditional studies, but few could be found that provided the flexibility we were looking for.

With the Multimedia Assessment Tool, presented in detail in this thesis, we first had to evaluate the necessary requirements and possible applications of the software. The task of designing and implementing it was then undertaken, using technologies such as the LAMP software bundle [36], and the newly updated HTML and CSS standards for building web interfaces [50]. Last, but not least, a thorough discussion of the system as a whole was conducted, examining everything from benefits to possible issues.

1.5 Outline

This thesis is organised as follows. Chapter 1 provides an introduction to the thesis, explaining the background and importance of creating the described framework. Chapter 2 includes background information on relevant topics, including quality assessment, multimedia and crowdsourcing. In Chapter 3, we go into further detail about subjective quality evaluation and methodology, as well as some details on experimentation and ethical considerations. Chapter 4 follows with some important elaboration on the technologies and frameworks that are used in the implementation of the application, while Chapter 5 will discuss in detail the design and implementation of the software itself. In Chapter 6, we will discuss and evaluate the system we have developed, and finally in the last chapter, we summarise everything with a conclusion.

Chapter 2

Background

An important topic and motivation behind the work of this thesis, is what is generally known as *quality assessment*. The main purpose of the tool we are developing is exactly that, evaluating quality, more precisely assessing *multimedia* quality. This chapter will go into further detail about the subject of quality assessment, along with touching upon the topics of multimedia and crowdsourcing. These topics all play a central part in this project, and furthermore emphasise the importance and applicability of the assessment tool we are developing.

2.1 Quality Assessment

Across research institutes, in industries and in research in general, people use data in assessment and decision making. Data-based decision-making is an essential element of continuous quality improvement, and helps individuals and teams to assess the efficiency and effectiveness of current processes [45]. There are several methods for collecting data; focus groups, personal interviews, review of records, counting events, and of course surveys.

Quality assessment in itself can be divided into two related, but different categories, quality of service (QoS) and quality of experience (QoE). The two methods are mainly different in how they determine quality. QoS methods measure quality objectively, while QoE generally use subjective measures. While QoS may be an important factor in assessing critical parts of a system, QoE is often essential for providing information on how the end-user perceives the overall quality. This will be explained in further detail in the following sections.

2.1.1 Quality of Service

Quality of Service (QoS) is a term that is often used within several aspects of computer science. Originally, it was defined by the International Telecommunication Union (ITU) within the field of telephony [32], but has played an equally important part in computer networks and similar technology. The QoS concept refers to an objective system performance metric, such as the bandwidth, delay, and loss rate of a communication network. Objective methods can be divided into two categories: signal-based methods and parameter-based methods.

2.1.2 Quality of Experience

Quality of Experience (QoE) is a common term used for defining the quality of a service based on users' own individual opinions. Thus, experiments in QoE are referred to as subjective. This however is not always the case, as it is also possible to run objective QoE experiments. This is commonly done by using objective measures to detect or determine quality issues that the human user would perceive as annoying and thus lessen the experience, for example by analysing a multimedia clip and checking if unnatural noise occurs in the processed video segment. Peak signal-to-noise ratio (PSNR) is an example of an objective QoE measure which is often used in quality assessment of multimedia. However, this metric is only conclusively valid when used to compare results from the same content and codec type [22]. Thus, although objective methods are in general more convenient to use, subjective methods are often needed nonetheless. Subjective QoE experiments provide factual assessments of users' experiences, and no matter how sophisticated objective assessment methods may be, they cannot capture every QoE attribute that may affect the experiences of users [56]. Multimedia in particular has become such an essential part of our everyday lives, that QoE of multimedia content will be an especially important issue for the foreseeable future.

Experience is, obviously, highly subjective. People with different cultural background, social and economic status, and personal experiences often react differently to similar experiences. For example, just changing some colours in an interface may change the effect it has on different people. Moreover, experience is context-dependent [33]. The same multimedia content may result in a different experience by the same person depending on the context. By context, we refer to the person's understanding of the situation or experience. This, however, can be difficult to identify and, although it may be taken into consideration, it is often not practical to try to measure.

Thus, to judge which particular experience is more pleasing, or preferred, we are inclined to use QoE methodologies and experiments to evaluate users' opinions, and thereby come to a conclusion on what is perceived as the better opinion for the majority of users. In Chapter 3 we will discuss some prominent methodologies often used to determine QoE, specifically within the field of multimedia assessment.

2.2 Multimedia

Multimedia is the simultaneous use of different types of media to effectively communicate ideas or knowledge, commonly accessed by an information content processing device. Multimedia includes a combination of audio, video, text, still images, animation, or interactivity formats. It may be either live or recorded, and is often divided into two categories, linear and non-linear. Linear multimedia content progresses without any navigational control by the user, like a cinema presentation, while non-linear uses interactivity to control progress, such as with a video game [53].

The primary question we need to ask ourselves is *why do we evaluate multimedia*? Alternative answers to this question might exist depending on the field of research, but commonly, and in the case of this thesis, the answer is fairly straight forward. We evaluate multimedia to examine if today's encoding and compression technologies are acceptable and, if not, how to improve them. Basically, to find out how much encoding and compression is optimal. What we mean by optimal can be highly subjective. Commonly, the technologies can be seen as optimal or acceptable when the perceived multimedia quality of the result is good enough for the purpose it is intended to be used for. For example, a highly compressed, low resolution video might look excellent on a mobile device, but not on a High Definition TV (HDTV). This generally means little or no presence of *compression artifacts* in the final outcome of the encoding. To understand the source of artifacts in digital video, consider the schema of a typical digital video processing system presented in Figure 2.1:

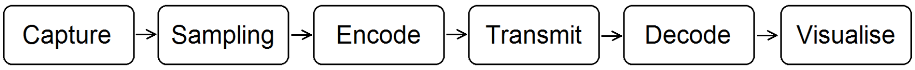


Figure 2.1: Digital video processing.

Digital video is captured, represented, processed, transmitted, and finally displayed as a sequence of still images, or frames, at a particular frame-rate. Each image consists of a rectangular array of rectangular shaped pixels, each containing colour and brightness information of a small region in a captured scene. Artifacts may occur in any frame, and different artifacts can be produced during each of the aforementioned steps. We are mainly interested in artifacts that occur in the encoding process, and to some extent the transmission and decoding processes, which we will come back to in Section 2.2.2.

2.2.1 Encoding and Compression

The advances in Internet services and applications, the rapid development of mobile communications, and the importance of video communications, are all increasingly relevant these days. Users expect high quality content with little or no delay, and limitations in networks and bandwidth make it therefore necessary to compress data in order to meet the users' expectations. Thus, encoding and compression are some of the enabling technologies for many aspects of what can be called a multimedia revolution [46].

Encoding involves representing a piece of information in another form.

For example, in hexadecimal encoding, one can represent 10 as 0xA. Compression, however, is done entirely to lessen the number of symbols (or bits) to represent a given piece of information. This is achieved with the help of specific encoding of information. Different types of encoding give different levels of compression, but encoding does not always compress data. There are two main methods of compressing data, lossy and lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy, while lossy compression reduces bits by identifying unnecessary information and removing it. As the name indicates, no information is lost in lossless compression. Both methods are equally important, but are often be used for different purposes. For example, lossless compression would be important in situations where we want the reconstruction to be identical to the original. However, in situations where this requirement is not as important and more compression would be achieved, we can use lossy compression.

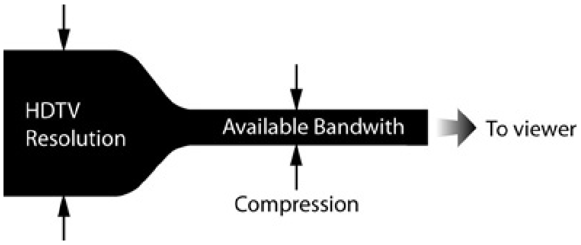
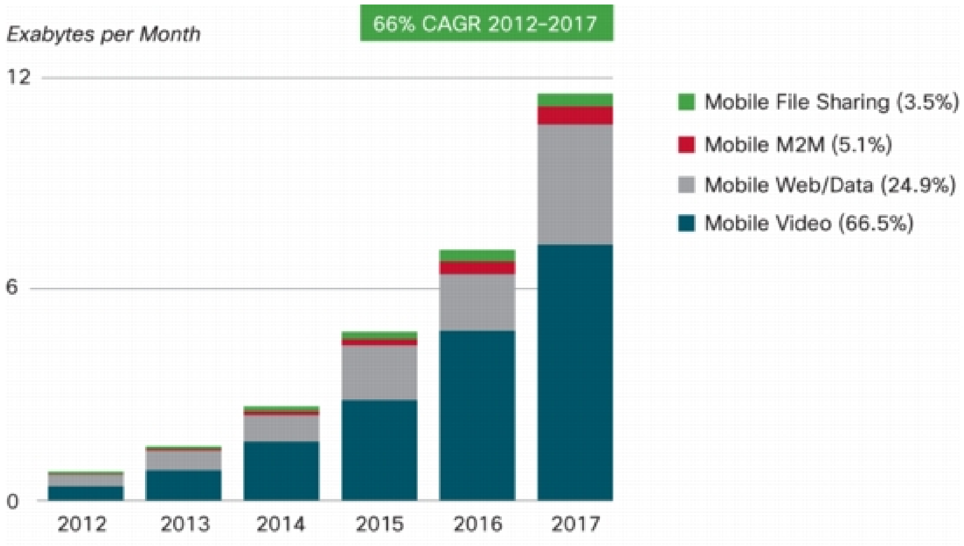


Figure 2.2: Compression and available bandwidth [12].

With the advances in technology and the expectations of the user that we mentioned initially, more and more data needs to be transferred over increasingly insufficient bandwidth. Using data compression, information can be shrunk into smaller sizes, which in turn enables larger amounts of data to be transferred at greater speeds. Moreover, compression is not only important in data communications. Today companies and people in general store a massive amount data on their computer systems. This data would take up an unnecessary large space, if it were not for compression. This is especially the case when it comes to audio and video. Development of better transmission and storage technologies to handle these vast quantities of data is ongoing, but unfortunately it is not enough. Current studies show that especially mobile data traffic is under major growth, almost doubling every year [24]. Figure 2.3 shows that video traffic accounts for over 60 percent of these numbers alone, and will continue to grow. At the same time, mobile users expect high-quality video experience in terms of video quality, start-up time, reactivity to user interaction, and so on. Consequently, compression is highly important within this and many other aspects of today's multimedia generation.

2.2.2 Compression Artifacts

Multimedia is subject to various kinds of distortions during the events of acquisition, compression, processing, transmission, and reproduction.



Figures in legend refer to traffic share in 2017.
Source: Cisco VNI Mobile Forecast, 2013

Figure 2.3: Global mobile data traffic forecast, 2012-2017 [24].

These shapes, or distortions, are what is referred to as *compression artifacts*. These artifacts are distortions that the human perception finds unnatural, and can consequently lessen the viewing experience [12]. Compression artifacts that are not related to data transmission are only present in lossy compression methods. Lossless compression does not discard any information, and therefore does not produce any artifacts of this kind. Commonly, the minimisation of perceivable artifacts is a key goal in implementing a lossy compression algorithm. Some of the most prominent artifacts that may reduce the perceived quality of multimedia sequences are summarised in the following sections.

Block Distortion

Block distortion (also known as blockiness or blocking artifacts) manifest themselves as unnatural and easily perceptible blocks within an image. It is an image distortion defined by the inherent block encoding structure becoming visible [12]. These are often seen in compression methods that use block transformation coding, like the Discrete Cosine Transform (DCT), that group blocks of pixels together. As the block distortion in Figure 2.4 shows, edges are distinctly visible along the block structures.

Mosquito Noise

One specific artifact can be seen as ringing or other edge busyness in successive still images, which may appear in sequence as a shimmering blur of dots around edges. These are generally referred to as *mosquito noise*, as they resemble mosquitoes swarming around an object. Mosquito noise is most noticeable around artificial or computer generated objects or lettering

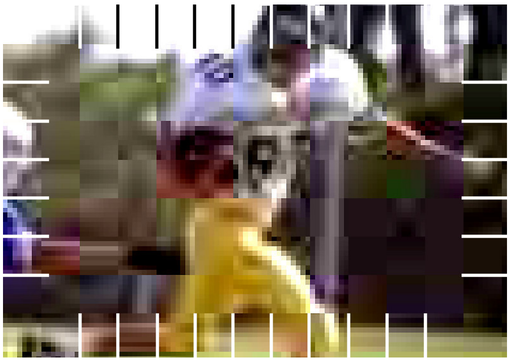


Figure 2.4: Distinct blocking artifacts in an image [12].

on a plain coloured background. Moreover, this effect is also visible around more natural shapes like a human body. It occurs when reconstructing the image and approximating discarded data by inverting the transform model. [12] [29]

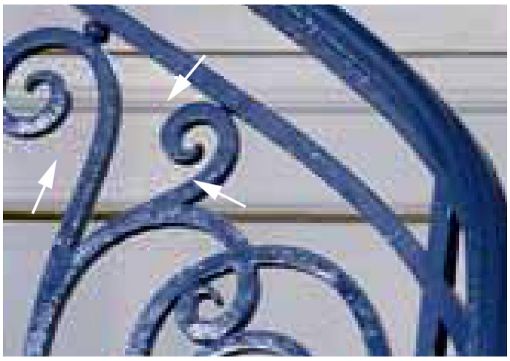


Figure 2.5: Mosquito noise around edges of objects [12].

Quantisation Noise

Quantisation noise is defined in [29] as a "snow" or "salt and pepper" effect similar to a random noise process, but not uniform over the image. Consequently, it is fairly similar to mosquito noise, although it appears randomly across an image instead of particularity around edges.

Ringing Artifacts

Ringing artifacts are spurious ring-shaped visual echoes on sharp edges, echoes of hard edges, or oscillations or shimmering along the edges of an object against a relatively uniform background. This is commonly caused by coarse quantisation and loss of high frequency components in compression [54].

Blurriness

Blurriness is commonly defined as a global distortion over an entire image, characterised by reduced sharpness of edges and spatial detail. Reduction in sharpness of edges is often due to the attenuation of the high spatial frequencies [35]. Compression algorithms that trade off bits for code resolution and motion often cause this kind of artifact [29].

Jitter/Jerkiness

Jitter, or jerkiness, is motion that was originally smooth and continuous perceived as a series of distinct snapshots. It is the result of skipping video frames to reduce the amount of video information that the system is required to transmit or process per unit of time [29].

Auditory Artifacts

As with video, there are equally many artifacts pertaining to audio. Audio encoders are similarly complex in the way the process and compress audio data, and lossy audio compression may therefore result in a wide range of artifacts which reveal themselves as weird or unnatural noises. The two most common auditory artifacts are typically referred to as band-limited artifacts and birdie artifacts [39].

Packet Loss

In transmission of video or audio, the decoder might not receive all the encoded data because of loss or delay of data packets occurring in various layers of the underlying transmission network. In turn, this may produce unwanted artifacts during reconstruction. With the use of motion prediction in compression algorithms, a single packet loss can also affect many subsequent frames (motion-compensation artifact) [40]. Consequently, the resulting reconstruction of the compressed data may produce various errors over longer periods of time in a video sequence. These artifacts occur more often in situations where bandwidth is limited or the network is prone to errors. Because the fault occurs during the transfer of data, packet loss artifacts fall under transmission artifacts rather than compression artifacts.

Asynchrony

Another transmission artifact type is asynchronous artifacts. They are, as the name indicates, noticeable asynchrony between audio and video during playback. Although asynchrony typically is not perceptible until it reaches >100 milliseconds out of sync either way [38], human subjects find synchrony issues to be especially annoying once they occur. Asynchrony may typically take place in situations where the audio and video streams are transmitted or processed separately, often with different delays.

2.3 Crowdsourcing

Crowdsourcing has emerged in recent years as a potential strategy to enlist the general public to solve a wide variety of tasks. The term itself is a combination of the two words *crowd* and *outsourcing*, a neologism that means utilising the general public's wisdom rather than the expertise of employees or contractors [56, 7]. Consequently, crowdsourcing is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, most often online using Internet crowdsourcing services. It combines the efforts of numerous self-identified volunteers or part-time workers, where each contributor of their own initiative adds a small portion to the greater result. Crowdsourcing is distinguished from outsourcing in that the work comes from an undefined public rather than being commissioned from a specific, named group. The following figure shows a typical crowdsourcing value chain, where crowdsourcers seek workers via a facilitator, and in return receives a solution to the problem at hand.

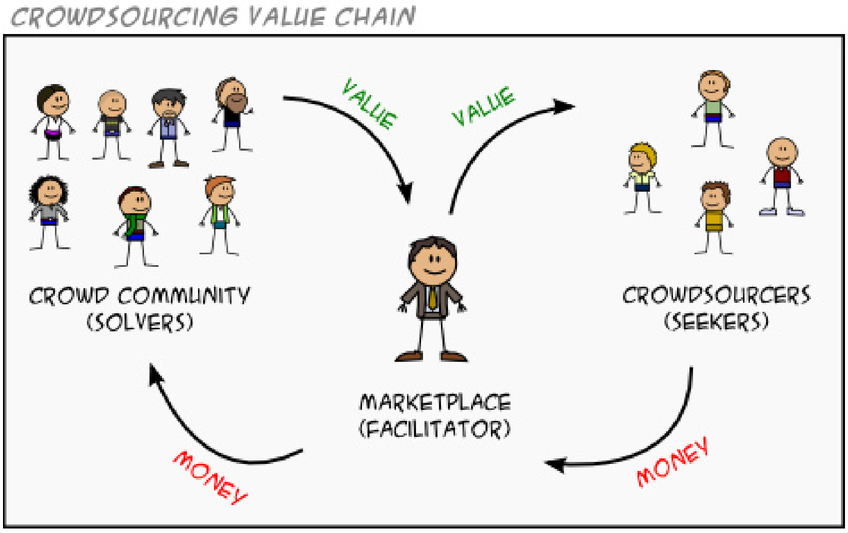

CROWDSOURCING VALUE CHAIN
VALUE
VALUE
CROWD COMMUNITY (SOLVERS)
CROWDSOURCERS (SEEKERS)
MARKETPLACE (FACILITATOR)
MONEY
MONEY

Figure 2.6: A common crowdsourcing value chain.

2.3.1 Typology

Crowdsourcing can commonly be divided into different types, depending on what problem is to be solved. For example, Daren C. Brabham has put forward a problem-based typology of crowdsourcing approaches [4]:

- Knowledge Discovery and Management - for information management problems where an organisation mobilises a crowd to find and assemble information.
- Distributed Human Intelligence Tasking - for information management problems where an organisation has a set of information in hand and mobilises a crowd to process or analyse the information.

- Broadcast Search - for ideation problems where an organisation mobilises a crowd to come up with a solution to a problem that has an objective, provable right answer.

- Peer-Vetted Creative Production - for ideation problems where an organisation mobilises a crowd to come up with a solution to a problem which has an answer that is subjective or dependent on public support.

Online QoE assessment studies seem to fall under the distributed human intelligence tasking approach, as we have a set of information in the form of multimedia and we wish to mobilise a crowd to assess, or analyse, this information.

Additionally, several categories of crowdsourcing have been indicated to define ways in which people use crowds to perform tasks [21]. This includes, although not limited to, crowdvoting, crowdfunding, microwork, wisdom of the crowd, creative crowdsourcing, and inducement prize contests. For our project, microwork is the more relevant category to bring to attention. Microwork is a platform in which users do small tasks for which computers lack aptitude, generally for a small amount in payment. Amazon's Mechanical Turk [23] may be the most popular service for this type of crowdsourcing, and would be a recommended service to use for gathering participants for research using MAT.

2.3.2 Benefits

Online surveys are possibly the most popular application of the crowdsourcing strategy for user studies [7], and there are several benefits of using crowdsourcing to gather results compared to the traditional laboratory setting. Most significantly, they are more efficient in terms of time and monetary cost, since it is relatively easy to collect responses from a large number of people within a short time-frame online. Especially the lower price, compared to the price for hiring professionals or a general public offline, draws researchers towards the crowdsourcing paradigm. Moreover, the high number of people who are ready to work for you at any time is commonly greatly beneficial. Online surveys may also be favourable for the participants, as they can respond at their own convenience. Having this option might also make participants more willing to complete the questionnaires. Additionally, online surveys do not have the "interviewer effect", where the interviewer may influence how participants answer the questions [7, 55].

While our assessment tool supports invitation of specific people for specific experiments, for example a group of experts for research within a specified field, crowdsourcing is obviously a highly valuable option for gathering participants due to the low costs and the reduced time and effort constraints it facilitates. Mass-creation of participant accounts and invitation of participants signed up through crowdsourcing services are features that are implemented in MAT.

2.3.3 Challenges

The main issue with crowdsourcing is the trustworthiness of the general internet user. Not everybody is trustworthy, unfortunately. Here we refer mainly to how the user accomplishes the crowdsourced task, if it is within a certain expected quality or not. Since crowd-workers completing tasks are paid per task, there is often a financial incentive to complete tasks quickly rather than well. Within many applications of crowdsourcing, verifying responses may be time-consuming, so having multiple workers complete the same task is often needed to correct discrepancies. However, having tasks completed multiple times increase both time and monetary costs. Consequently, an interesting affect emerges; We trust our assessors less, and it is harder to exclude outliers. Some ethical issues also arise within crowdsourcing, which we will be coming back to in Section 3.2.

When it comes to user studies conducted using online surveys, as with our project, methods may be built into applications to verify responses quickly and inexpensively. To counteract the false results that may occur due to untrustworthy participants, surveys often include multiple questions designed to tap into the same topic. This way the correlation between related items will give an indication on the consistency in responses. The correlation between related questions serves as a measure of inter-item reliability. This is what is often referred to as *consistency checking*. Moreover, within the field of quality assessment in particular, the exact same question (or test item) may be displayed more than once within the course of a survey. By using this kind of repetition, the consistency of the participants responses can be further checked, and low correlation between identical questions are easily detectable. In some cases, so-called lie detector questions are also used. These are questions people will tend to lie on if they wish to present themselves in a better light. However, this type of question is seldom used in quality assessment studies.

In traditional laboratory experiments, participants normally view multimedia content in a controlled environment that equalises experiment conditions. In crowdsourced experiments however, participants often view the content under varied conditions, such as different screen sizes, surrounding lighting and various equipment qualities. This may be a disadvantage if the goal is to measure the quality of multimedia content in a specific scenario. On the other hand, it can be considered an advantage because the users' perceptions can then be assessed in real-life scenarios.

Demographic factors also play an important role in crowdsourcing, and its issues. Some QoE assessment studies rely on a specific demographic make-up of participants [55]. However, crowdsourcing makes it difficult, if not impossible, to relate the assessment results to demographic factors, such as gender, age or location. Identifying each crowd worker, for example by asking questions about demographic elements, is unlikely to be effective as this data may not be trustworthy. Moreover, researchers cannot use sampling techniques to select candidate respondents, as they commonly do for face-to-face surveys [7, 55].

2.4 Existing Frameworks

Quality assessment has been an important topic for several decades within both business and research, and with the help of the Internet it has become an even more prominent factor. There are a large amount of software and common frameworks on the market today for running assessment surveys and studies. A question therefore arises; Why do we aim to develop yet another one? By looking at some of the alternatives out there, we reveal that there are very few tools that replicate the proposed features of MAT, and those that do are limited to a single evaluation method.

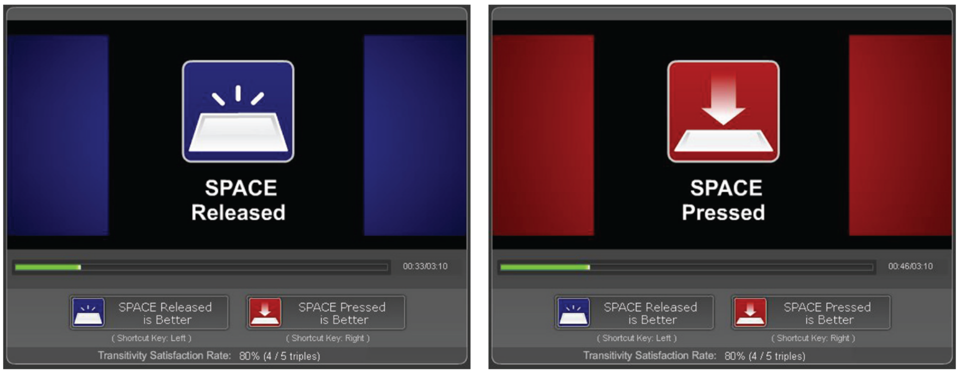
2.4.1 Online Assessment Tools

Online assessment tools, or survey applications as they are commonly referred as, are not hard to find on the Internet today. These are simple yet comprehensive tools in which experimenters can run many kinds of assessment studies. The participants commonly conduct these surveys by answering questions, one by one, by for example typing in an answer, selecting one of multiple choices, or selecting an item on a scale. A majority of the population connected to the Internet today have been through a survey of this kind, for example for evaluating a local supermarket or for measuring ones experience with a particular product or website.

Except for the presentation of multimedia sequences, our assessment tool greatly resembles that of an ordinary online survey application. This may infer that we could just use one of these tools for our purpose, however, there are quite a few issues that arise from that case. Firstly, these tools tend to be very extensive [42, 55]. This may be viewed as a positive factor, but in many cases these extra features are not necessary and in some worst cases they are just in the way. Building a framework from scratch, tailored to our specific needs, may be just as good if not better than some commercial tools. Secondly, these assessment tools are seldom free [42, 55]. For small studies or surveys, some software offer free, limited trials for commercial and private use, but as soon as one might want to scale things up, or want a few more features, the price increases accordingly. Most importantly, none of the tools we could find are specifically made for the presentation and assessment of multimedia content. Without the support for this content, together with the lack of the underlying methodology of QoE assessment of multimedia, it would be difficult to run any kind of proper assessment studies of this kind.

2.4.2 Specific Frameworks for Assessment of Multimedia

From what we have been able to find, there are actually surprisingly few frameworks available that specifically focuses on QoE assessment of multimedia content using crowdsourcing or online communications in general. One that stands out is a web-based platform facilitating QoE assessment of multimedia, called Quadrant of Euphoria [6, 7, 56]. This framework enables experimenters to create user studies on quality assessment of images, audio clips and video sequences, and has the possibility to gather participants using the previously mentioned paradigm of crowdsourcing. The procedure of conducting an experiment is based on the subjective quality assessment method Pair Comparison (PC) [30], and consists of the participant being presented with several pairs of media items under scrutiny. These pairs are presented interchangeably on the screen, and the participant may switch between them using the space-bar, as seen in Figure 2.7. The participant is then tasked with choosing with one of the items presented he or she prefers, based on the perceived quality of the items. This will ensue in giving the experimenter results on which items (and hereby systems) are generally more accepted than other.



Transitivity Satisfaction Rate: 80% (4 / 5 triples)

Transitivity Satisfaction Rate: 80% (4 / 5 triples)

Figure 2.7: Quadrant of Euphoria's experiment interface under both space-bar states [6].

Quadrant of Euphoria is thus similar in many ways to what we are developing in this project. However, some present issues still make it desirable to design and implement our own unique and different assessment tool. First of all, Quadrant of Euphoria only supports studies using the PC method. While PC is a strong method and often used in this kind of research, we aim to provide support for a number of other methodologies as well. These methods, together with PC, will be discussed further in Section 3.1. Secondly, if we were to get permission to further develop this tool, Quadrant of Euphoria seems to be partially developed in ActionScript for use with the popular browser plug-in Adobe Flash Player [26][6]. In addition to not having any previous experience with this development platform, Flash is a third-party application which unfortunately not everybody has installed. Moreover, with the entrance of HTML 5 onto the market, it is currently on an overall decrease in usage statistics on a worldwide basis [49], which indicates that it may not be supported forever.

2.5 Summary

Quality assessment has been an important topic in many areas of research for the past several decades. In this chapter, we have presented Quality of

Experience (QoE) as an important form of quality evaluation, which commonly measures user's own individual opinions or subjective experiences. QoE may be measured using both subjective and objective measures, however subjective methods often provide more factual assessments of user's experiences. Furthermore, quality assessment of multimedia content in particular is becoming more relevant, as the usage of multimedia in our everyday lives is increasing rapidly.

Multimedia is the simultaneous use of different media to effectively communicate ideas or knowledge, and is used in a majority of aspects regarding for instance entertainment and communication. The reason we are interested in evaluating multimedia content is typically to examine if current encoding and compression technologies are sufficient or optimal. However, this is highly subjective, depending on for example user's preferences and the purpose it is intended to be used for. Moreover, encoding and compression is necessary in order to transmit and store data efficiently and quickly, over increasingly insufficient bandwidth and storage. Multimedia content commonly consists of a large amount of data, making compression even more essential. Unfortunately, multimedia is subject to various types of distortions or artifacts during not only compression, but also transmission and other processing. These artifacts may compromise the quality of the content, thus lessening the user experience.

This chapter has also proposed crowdsourcing as a promising method of reaching a more diverse crowd of participants for online multimedia assessment studies. Crowdsourcing includes benefits such as time-efficiency and low monetary costs. The high numbers of people ready to work at any time, with the possibility of working from any where at any time, contributes to the popularity and usability of the crowdsourcing paradigm. However, some challenges present themselves with this method. The main issue is the trustworthiness of the typical Internet user. Environmental control and demographic factors may also play a small part within the challenges of crowdsourcing.

Furthermore, we have had a look at existing online assessment tools, pointing out the issues or shortcomings these present in the task of assessing multimedia content. Common online survey tools typically lack support for presentation and assessment methodology of multimedia, as well as generally being quite expensive. However, the web-based multimedia assessment framework Quadrant of Euphoria, is one that is similar to MAT in many ways. We are aiming to build a more flexible tool however, thus giving experimenters more freedom to design experiments according to their needs.

Chapter 3

Subjective Evaluation

As outlined, multimedia quality is typically approached by one of two methods; QoS, which assesses quality based on objective measures, and QoE, which considers the subjective opinions of assessors. In general, subjective quality assessment has no pre-established measure or standard and is thus based solely on the opinion of the evaluator, although some methods can use a point of reference to judge differences. Multimedia quality assessment relies heavily on this type of subjective evaluation to gather data on the perceived quality of experience of human observers. Subjective assessment is useful for measuring end-user acceptance, comparing alternative algorithms and finding optimal designs or configurations when it comes to encoding and compression of multimedia content.

3.1 Methodology

Several test methods for subjective quality assessment have already been researched and extensively used for many years. International recommendations, such as ITU-R Rec. BT.500 [27], ITU-T Rec. P.910 [31] and ITU-T Rec. P.911 [30], provide us with outlines of the most prominent ones. The recommendations provide instructions on how to perform these tests for the assessment of video and/or audio quality, in a controlled laboratory environment. Although, our assessment application typically does not run in a controlled environment like these recommendations describe, the outlined stringency with which to run user studies remains highly relevant. The recommended test methods are commonly known as Absolute Category Rating (ACR), Degradation Category Rating (DCR), Paired Comparison (PC) and Single Stimulus Continuous Quality Evaluation (SSCQE). Common to them all is the showing of multimedia sequences to a group of viewers, with their opinion recorded and averaged, to evaluate the quality of each audiovisual sequence. While the premises vary between tests, their outcomes contribute with mean scores for a range of quality implementations.

3.1.1 Absolute Category Rating

Absolute Category Rating (ACR), also known as the Single Stimulus (SS) method [27], is a category judgement where the test sequences are presented one at a time and are rated independently on a category (rating) scale [30, 31, 28]. The recommendations specify that after each clip, subjects are asked to evaluate the quality of the sequence presented. The presentation time may vary according to the content that is being evaluated, but the voting time should be limited to 10 seconds or less, depending on the voting mechanism used. A five point rating scale, as seen in Table 3.1, should be used. However, if a higher discriminative power is required, a larger scale may be used.

5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 3.1: ACR's recommended five point rating scale.

3.1.2 Degradation Category Rating

Degradation Category Rating (DCR), also known as the Double Stimulus Impairment Scale (DSIS) method [27], is a test method in which test sequences are presented in pairs; the first stimulus in each pair is always the source reference, while the second stimulus is the same source presented through one of the systems under test [30, 31, 28]. In this case, the subjects are asked to rate the impairment of the second stimulus in relation to the reference. The total presentation and voting times are recommended to be the same as in ACR. A five point scale is similarly to be used here, but the wording should represent a rating of impairment, as presented in Table 3.2.

5	Imperceptible
4	Perceptible but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

Table 3.2: DCR's recommended five point impairment scale.

3.1.3 Pair Comparison

The Pair Comparison (PC) method implies that the test sequences are presented in pairs, consisting of the same sequence being presented first through one system under test and then through another system [30, 31, 56, 7, 13]. Moreover, the source sequence may be included and would be treated as an additional system under test. Commonly, the systems under test are combined in all possible n(n - 1) combinations (AB, BA, CA etc.), thus to test each against all others, and in both possible orders. After presentation of each pair, the subject is tasked to choose which sequence is preferred. The voting time is similar to the previous methods, though the presentation time is recommended to be about 10 seconds.

Often in the case of large amounts of systems being tested, a huge amount of pairs of sequences could be constructed if every possible *n(n-1)* combination were to be run. Eichhorn et al. [13] presents a possible solution to this problem, named randomised pair comparison (R/PC). Using this method, each user is presented with a randomised subset of all combinations, hereby reducing the subjects time and effort significantly. However, while showing results similar to full experiments run with all pair combinations, this method requires a larger amount of participants. Additionally, the statistics in the results may become less conclusive. The importance of statistics will be explained further in Section 3.3.

3.1.4 Single Stimulus Continuous Quality Evaluation

Single Stimulus Continuous Quality Evaluation (SSCQE) is a test method that evaluates long-duration multimedia sequences, typically from 3 to 30 minutes. Subjects perform continuous subjective quality assessment, without any reference, by the means of moving sliders while looking and/or listening to a sequence [27, 30]. The results may be presented by plotting curves which indicate the percentage of time during which the subjective score was higher than a given score on a 0-100 scale. The method is consequently well suited to take into account temporal variations of quality and to make global quality assessments. The drawback however comes with having no reference, making it less suited for tests which require a high degree of discrimination [30].

3.1.5 Quality Evaluation of Long Duration Audiovisual Content

In a recent paper, Borowiak et al. [3] presents a method for multi-modal, long-term quality assessment of audiovisual content. This method, hereby referred to as QELDAC for short, differs from the previously mentioned methods in that it is based on an adjustment of the quality during playback. Assessors would adjust the quality to a desired level in the case where degradations occur, in comparison to giving a specific score which the other methodologies are based on. This eliminates the need for translating the perceived quality into a single number, which allows the subjects to focus on the content instead of directing their attention to the assessment task itself. Moreover, the research can focus more on the subjects' expectations and reactions to quality changes over longer periods of time.

3.1.6 Comparison

Each test method has its own set of advantages, and choosing which methodology to use for an assessment study may be not be as straight forward as one might imagine. An important issue in choosing a test method is the fundamental difference between methods that use explicit references (e.g. DCR) and methods that do not use any explicit reference (e.g. ACR, PC and SSCQE) [30, 31]. The latter does not test fidelity with regards to a source sequence, which is often important in evaluation of high quality systems [30]. In this case, when the viewer's detection of impairment is an important factor, the DCR method is recommended. ACR may be simple and fast to implement, and the presentation of the stimuli is similar to that of the common use of the systems under test. Thus, ACR is well suited for qualification tests [30, 31]. The PC test method takes advantage of the simple comparative judgement task in which to prioritise a set of stimuli. Because of its high discriminatory power, it is particularly valuable when several of the test items are nearly equal in quality [30]. Moreover, when using a large number of items in the test the more time consuming this procedure may be, which may be an inconvenience in some cases. The methodologies that consider long-duration sequences (e.g. SSCQE and QELDAC) are obviously more suited in situations where sequences of a longer duration needs to be assessed. The two methods vary slightly in how the method translates in regard to a final outcome of the assessment study. SSCQE would be used when the preferred outcome is a score based on the perceived quality at certain intervals throughout the test item [30]. However, QELDAC may be better in situations where the researcher would like to know what quality level is acceptable for a potential user [3].

Considering the advantages and weaknesses associated with the different assessment methods, the appropriateness of each will depend on the planned experiment. Experimenters have varying needs and they require the freedom to run assessment studies according to their particular needs or preferences. MAT aims to offer a large range of options and specifications for methodologies and presentation modes. The main limitation lies with what is feasible to implement, within the parameters of this project. With enough time and effort, all of the outlined methodologies could be implemented without any apparent challenges, even to the exact specifications of the ITU [30]. However, methods designed to assess long-duration sequences may require adjustments to the current structure of the software. This will be discussed in more detail in the design and implementation chapter. Furthermore, MAT's user interface makes it a convenient tool for experimenters of all levels of computer skills. The software can also manage large quantities of data, both on the input and the output side. Studies on multimedia quality demand that the experiment tool can handle presentation sequences of multiple audio and/or video files. In addition, response data from dozens of participating assessors need to be collected, analysed, and reported. MAT is designed to handle large and numerous data files without interrupting the flow of the experiment planning and running.

3.2 Ethical Considerations

When conducting research and administrating online assessment studies, there are several ethical dilemmas that may be necessary to take in to consideration. We will discuss in short a few of these that are relevant to the topic at hand.

Reward and Money Psychologists have found that giving rewards in the form of money or other goods commonly reduce the motivation of a participant [19]. Whether this applies for online assessment studies as well may need further research, but it is something that experimenters may need to keep in mind. An additional ethical dilemma when it comes to money or rewards, is whether or not it might influence the outcome of the results. By rewarding participants, it is possible to "pay for the right answer", thus altering the natural outcome of an experiment. Although voluntary participants may be persuaded to answer falsely, they generally have less or no motivation to do so. However, this phenomenon is unlikely to occur in QoE experiments, as researchers are ordinarily interested in finding individual preferences.

Crowdsourcing and Wages Recently, researchers have argued that the wage conditions within crowdsourcing may be unethical [44]. Crowdworkers are not guaranteed a minimum wage, because they are considered independent contractors and not employees. Moreover, no written contracts, non-disclosure agreements, or employee agreements are typically made with crowdsourced workers. This gives the requesters the final say over whether users' work is acceptable, and whether or not they will be paid. Although crowdsourcing may be viewed as slightly unethical towards the worker, the cheap labour is one of the main reasons it has become so popular.

Representativeness/Sampling Participants are essential to any experiment, and as such they need to be objectively taken into consideration when running experiments. First of all, experimenters commonly need to know if a participant is representative. Being representative means that the participant is within the demographic sample of people that the experiment is aimed towards. However, since determining demographic properties of participants is such a challenge when using crowdsourcing, this may be hard to achieve [55, 56]. Consequently, when representativeness is important, the experiment should be directed towards a known group of participants rather than an unknown crowd. Furthermore, when participant diversity is an accepted or desired property of the experiment, crowdsourcing may be especially suitable.

Fatigue Participant fatigue is a common outcome when similar tasks are expected to be performed repeatedly over long periods of time. When participants become fatigued or bored from repeating the same task over and over, as in typical assessment studies, responses may become less accurate. However, since MAT is aimed at being a flexible evaluation tool, the experimenter has control over repetitions and how long any experiment will be, thus reducing the potential of fatigue. Additionally, unlike common laboratory experiments, participants using MAT have the unique advantage of taking a break at any time he or she feels it necessary.

Anonymity and Confidentiality Anonymity and confidentiality are two significant topics when dealing with any kind of human population. Anonymity refers to concealing the identities of participants in all documents resulting from the research, while confidentiality is concerned with who has the right of access to the data provided by the participants [2]. People tend to prefer to appear anonymous when participating in online assessment surveys, especially when semi-sensitive information is requested. Associating results from assessment studies to specific people is commonly neither necessary nor useful, and anonymity is therefore a common feature for most assessment tools. Consequently, when using our tool, we provide total anonymity for participants conducting experiments and all data should be kept confidential.

Data Collection Assessment tools collect data from participants in order to produce results on the topic of the experiment. Data collection is a specifically important topic when it comes to both law and ethics. Not only are there restrictions on what kind of information can be collected and for what purpose, but also for what and how this data can be used or published subsequently. Moreover, any sensitive data collected must be stored securely so that it cannot be accessed by any unauthorised groups or individuals. Information on data collection and general disclaimers are therefore often included at the start of online surveys, to inform the participant of the purpose of the study and assure that the data will be in secure hands. In some cases, participants are asked to give consent to the planned use of the collected data, which often coincides with the possibility to withdraw from the experiment or study at any time as well.

Information on general data collection and anonymity is included in experiments created using MAT. However, since MAT allows experimenters to design several different experiments for different purposes, specific information or disclaimers on this topic should be added to the text presented prior to experiment commencement by the experimenters themselves.

3.3 Statistics

When dealing with any kind of assessment study, statistics play an essential part. Statistics is the study of the collection, organisation, analysis, interpretation and presentation of data. Consequently, assessment studies collect and organises data from its experiments for further analysis, interpretation and eventually presentation. Analysis of the data is needed to provide evidence for what is being researched in the experiment. For example, averaging user scores on each test condition as an absolute mean opinion score (MOS) is a typical approach. This is a statistical procedure, one of many different types.

Although MOS is simple to implement and use, it cannot express the confidence of a result. Thus, even a well designed experiment presented only by its MOS ratings may be rejected in a good publication. It is therefore recommended that researchers perform additional statistical analysis on the results of their experiments. For example, analysis of variance (ANOVA) is one parametric procedure that allows researchers to compute statistic outcomes including a confidence analysis.

In the results and statistics section of the experiments conducted using our assessment tool, results of a few basic statistical procedures are presented together with the experimentation data. This includes percentages, a mean opinion score and the standard deviation, together with the raw data. This is also available in a downloadable spreadsheet format, so that further analysis can be done by the researcher. More advanced statistical procedures, like ANOVA, may be implemented as well in future work.

3.4 Summary

Subjective evaluation of multimedia content is typically approached using a set of well documented and tested methodologies. In this chapter, we have discussed some of the prominent methods of quality assessment outlined by the International Telecommunication Union (ITU). Furthermore, a comparison of these methods has been presented. The methods each have their own set of advantages, and are commonly used in slightly different settings. The DCR test method for example, uses an explicit reference for detection of impairments which is often important in evaluation of high quality systems. ACR on the other hand presents sequences one at a time, where users rate each clip on an rating scale, similar to the MOS test method mentioned in the introduction chapter. The PC method displays sequences in pairs, taking advantage of the simple comparative judgement task of prioritising one stimuli over another. Finally, the SSCQE and QELDAC methods consider long-duration sequences, used in situations where evaluation of long-duration multimedia is necessary. Because experimenters often require the freedom to run assessment studies according to their particular needs and preferences, we aim to provide a large range of options and specifications for these methodologies and presentation modes in MAT.

Ethical and practical considerations are additional topics of importance when regarding online assessment studies. Several matters for consideration are discussed in this chapter. For example, giving rewards in the form of money or other goods for participation may alter the outcome of the research to some extent. Anonymity and confidentiality regarding the participants are also important factors to consider. This implies allowing participants to remain anonymous and ensuring that data collection is handled properly and securely, while keeping the data confidential.

Furthermore, we mention that statistical analysis of experiment results is necessary in any assessment study. Analysis is needed to provide evidence for what is being researched, and as such we have described briefly which statistical procedures are included and presented in MAT. Moreover, the raw data is easily downloadable for further analysis and interpretation.

Chapter 4

Technologies & Frameworks

Throughout the development of our assessment tool, we have relied on a body of well-tested and popular technologies, frameworks and programming languages. Together, these form the foundation for the design and implementation of the MAT software. In order to provide insights on the build and workings of the application, this chapter outlines the technological background for its development.

4.1 Server-side

The LAMP software bundle is a set of free, open source software that is commonly used to build a viable general purpose web server [52]. The acronym LAMP refers to the four technologies used in this bundle; Linux (Operating System), Apache (HTTP Server), MySQL (Database Software), and either PHP, Perl or Python (Scripting). The exact combination may vary, especially with the choice of scripting software, but also when it comes to the operating system. Other operating system combinations include Microsoft Windows (WAMP), Mac OS (MAMP), Solaris (SAMP), iSeries (iAMP), or OpenBSD (OAMP). Some less used variants incorporate an alternate web server, like Microsoft's Internet Information Services (WIMP), or even different database software, like PostgreSQL (LAPP). Figure 4.1 shows a graphical representation of the general LAMP architecture and illustrates how the components interact with each other. Each component is described in detail in the following sections.

The primary reason for the popularity of the LAMP combination, is the free of cost and open source of the software, which makes it easily adaptable. All the components come bundled with most current Linux distributions, which greatly improves the ease of use. Our tool uses the standard LAMP configuration as the web application server, with PHP as the chosen programming language. This setup is simple, but powerful, and covers all our requirements.

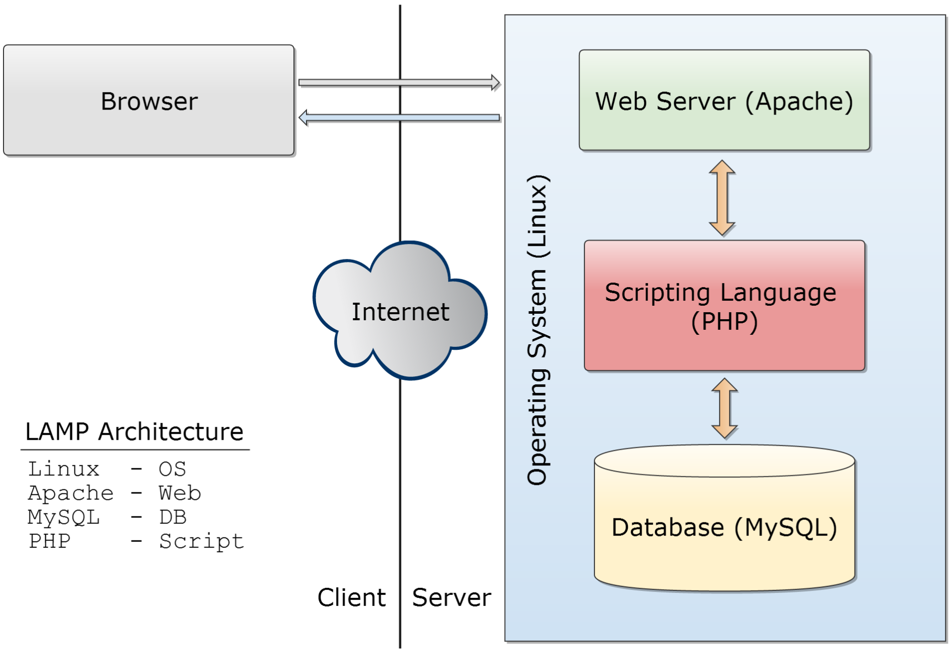


Figure 4.1: The LAMP software architecture.

4.1.1 Linux

Linux is the Operating System (OS) on which all the remaining technologies of the LAMP stack will run. Originally developed in the early 1990's as a port of UNIX to the Intel x86 processor [36], it has become one of the more commonly used operating systems. Moreover, Linux, and variants of Linux, is the most popular operating system for *servers* and other larger systems [41]. The development of Linux is one of the most prominent examples of free and open source software collaboration, contributing to its wide-spread use. Consequently, several different Linux distributions have been developed over the past two decades. The more prominent ones include Debian, Red Hat, (open)SUSE and Mandriva. MAT is designed to run on a Debian system, but is fully supported on any other Linux system with the LAMP software stack.

4.1.2 Apache

The Apache Hypertext Transfer Protocol (HTTP) Server is a highly efficient, secure and extensive web server that provides HTTP services in sync with the current HTTP standards [18]. The Apache web server is developed and maintained by an open community under the Apache Software Foundation, and has been the most popular web server since 1996 [18][41]. Consequently, it had a key role in the initial growth of the World Wide Web, and was in 2009 the first web server software to surpass the run of 100 million websites [41].

The primary function of a web server is to deliver web pages requested by clients. This service is achieved through the Hypertext Transfer Protocol (HTTP) [15], which is the foundation of data communication over the World Wide Web. Commonly, it will deliver HTML documents and any additional content that may follow the document, such as images, style sheets and scripts. If unable to deliver, the server will respond with an error message. In addition to serving content to the client, HTTP includes methods for receiving content from clients. This feature is commonly used for submitting web forms and uploading files to the server. More importantly, web servers such as Apache support server-side scripting using PHP, and a number of other scripting languages. This function can be used to create dynamic web pages, rather than simply returning static HTML pages from the server's secondary storage (See Figure 4.2). Furthermore, server-side scripting adds the possibility to retrieve and modify large amounts of data from databases. This is key for developing highly dynamic applications like the Multimedia Assessment Tool.

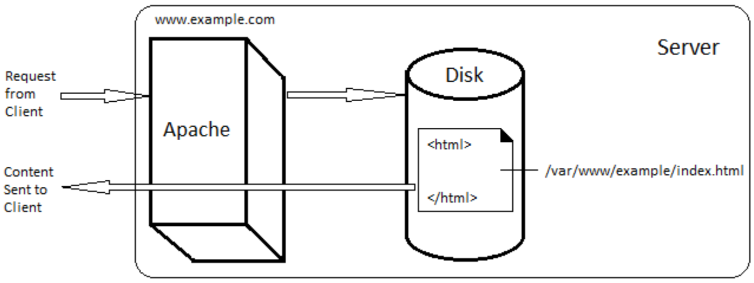


Figure 4.2: Request and delivery of static content.

4.1.3 MySQL

The most widely used open source relational database management system (RDBMS) in the world, as of May 2013, is MySQL [10]. It is an especially popular database for use in web applications, mostly due to its involvement in the LAMP software stack. Moreover, high connectivity, speed and security make it greatly suited for accessing databases on the Internet. The MySQL database software runs as a standalone server, providing multi-user access to any number of databases. In short, a database is a structured collection of data, and to access and manipulate data stored in a database, a DBMS [11] like MySQL is required. DBMSs are used in a vast number of applications and software, because they typically provide the best and most efficient way of storing large amounts of structured data. The latter part of MySQL stands for Structured Query Language, and is again the most common standardised special-purpose programming language used to access databases [11]. The language is designed specifically for managing data stored in a relational DBMS. Relational database systems, such as MySQL, store data in tables, as collections of rows and columns. In addition, these systems are responsible for providing relational operators to manipulate the data in tabular form.

The example code below shows a general SQL query for accessing specific data from a table in the database.

```
SELECT firstname, lastname
FROM people
WHERE city="Oslo"
```

MAT uses a single MySQL database to store all dynamically created data, from participant information to survey questions and answers. Details on how this data is stored, retrieved and used will be covered in Chapter 5.

4.1.4 PHP

PHP [20] works as the link between the HTTP server software (Apache) and the database management system (MySQL). Together, the software and programming language enable the construction of solid, dynamic web applications. Although several current technologies can accomplish this job, PHP is one of the most popular. Originally, this server-side scripting language was designed for exactly that purpose, web development, but today it is also used as a general-purpose language. Since its conception in 1995, PHP has evolved to become *the* most used server-side programming language in the world [48]. Similar to the other components of the LAMP software it comes free of charge, and it is constantly maintained and improved by its developers, The PHP Group. The name PHP originally stood for Personal Home Page Tools, but is now a recursive acronym for *PHP: Hypertext Preprocessor*.

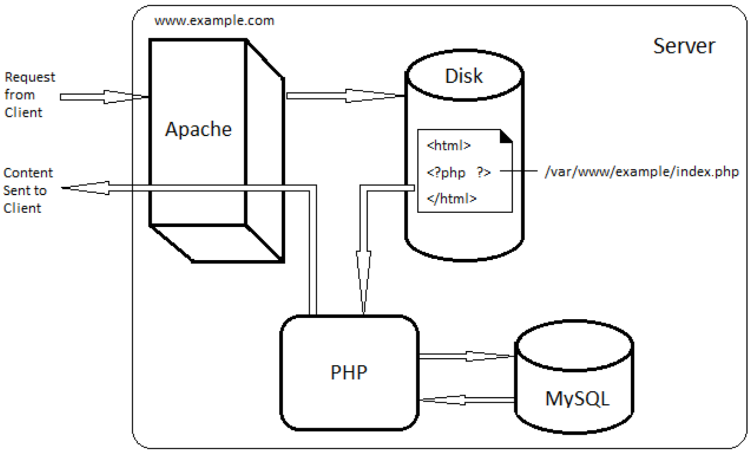


Figure 4.3: Request and delivery of dynamic content.

As seen in Figure 4.3, the PHP software acts as an intermediary between the HTTP server and the DBMS [36]. The following steps are used to exemplify a typical execution of a request [20]:

- A request for specific file (page) is sent from the client to the web server.

- The HTTP server software (Apache) dispatches the specified file from the disk to the PHP interpreter.

- The file is parsed for valid PHP code, which is then interpreted on-the-fly, and finally executed by the PHP engine.

- During execution, calls to query the DBMS with one or more SQL statements are performed as they occur in the code.

- When the execution of the code is complete, the result is typically a new dynamically created HTML document.

- This document is passed to Apache, which in turn sends it back to the client who initiated the request.

Any PHP code in a requested file is interpreted "on-the-fly" to an internal format (bytecode) that can be executed by the PHP engine. PHP acts primarily as a filter, taking input from a file or stream that contains PHP instructions and outputting another steam of data, commonly in the HTML format. Because PHP code is interpreted and executed each time a request is received, it is slower than programming languages that use pre-compiled code. However, this is rarely a problem with small to medium sized applications, like MAT. Several methods and compilers have been developed to increase the execution speed of PHP, but this is beyond the scope of this project. Another concern related to PHP is potential vulnerability to security breaches. However, the majority of these breaches can be traced back to failures to follow best-practice programming rules. To eliminate the most grave security concerns, web forms incorporate input validation that ensure user input does not contain malicious data. Furthermore, user authentication helps secure the software from unauthorised or undesired use. We will go into more detail about this in the design and implementation chapter of the thesis.

4.2 Client-side

The graphical front end of modern web sites and web applications are generally comprised of three aspects: HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (JS). The extent to which each is used varies according to the function of the site or application. A static site with no designated application might only use HTML and CSS, while larger web applications might rely heavily on JavaScript functionality to make it resemble a typical program on a personal computer. These three technologies are all rendered on the client-side, commonly in a *web-browser*, upon receiving the content from the server. Consequently, the web-browser forms the client. Everything the user can see and interact with on their side of the application is a clever combination of these technologies (See Figure 4.4).

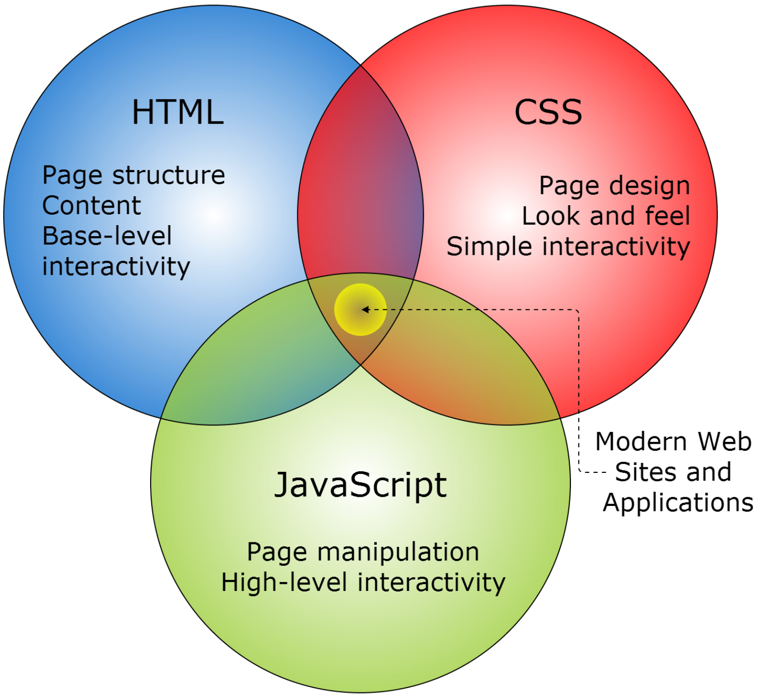


Figure 4.4: Composition of modern web sites and applications.

4.2.1 HTML and CSS

Along with graphics (images, backgrounds, etc.) and scripting (e.g. JavaScript), HTML and CSS form the basis for building web pages and web applications [50]. HTML is a *markup* language that web browsers use to interpret and compose text, images and other material into visual or audible web pages. With HTML, authors describe the structure of pages using this markup. HTML content is a form of markup that is conducive to information dissemination. Consequently, HTML is not a graphical descriptor in itself, since it is rendered by a client browser at the client's discretion. It is important to note that HTML does not dictate layout, it merely suggests it.

On the other hand, Cascading Style Sheets (CSS) are used for describing the presentation semantics (the appearance, layout and formatting) of a document written in a markup language. A document's accessibility can be greatly improved by separating its content from its presentation. Such a separation also reduces the complexity and repetition of the document's structural content, which enables multiple pages to share a specified formatting. Furthermore, this procedure makes it possible to change the presentation of a document depending on the purpose, for instance changing the layout for printing or adjusting to different screen sizes.

Both HTML and CSS are maintained by the World Wide Web Consortium (W3C) [50]. During the past few years, new versions of the two technologies have become highly mainstream. These versions are referred to as HTML 5 [1] and CSS 3 [14]. Although neither have reached full recommendation status from the W3C, the standards are to a large extent implemented in all major web browsers and are therefore in common use. MAT, for example, is implemented using several of the new features introduced in these versions.

4.2.2 JavaScript

JavaScript, or JS for short, is an extremely popular client-side scripting language that is typically used to manipulate the Document Object Model (DOM). In short, the DOM is a convention for representing and interacting with objects in HTML, XHTML and XML documents. Originally, JavaScript was implemented as part of web browsers, to enable client-side scripts to interact with the user, control the browser, communicate asynchronously with a server, and alter the document content [16]. However, recently its use outside of the browser is becoming more frequent, for example for server-side scripting. Similar to PHP, JavaScript is an interpreted programming language, where the code is immediately compiled and executed at runtime.

As previously mentioned, JavaScript enables high interactivity in a web application when used together with HTML and CSS. Although MAT does not require a great degree of interactivity in the form of these scripts, its usefulness remains relevant in a few instances. Specifically when it comes to the validation of web forms. While conducting an experiment, each and every question represents a web form. Validating the input from a user is important, not only for checking that an answer has been given, but also for verifying that the answer is a valid one. There are ways of manipulating HTML forms in order to send unintended or malicious data back to the server, which is why validation is necessary.

Several libraries are available to make JavaScript-development easier, faster, and even more robust against cross-browser compatibility issues. The most commonly used library of this type, is the free and open source jQuery [34]. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, and handle events. In addition, it adds support for *plugins* on top of the JavaScript library. This modular approach allows for the creation of powerful dynamic web pages and web applications. Consequently, jQuery is included in the MAT software and scripts are written in this specific syntax in order to make the software more easily supported on all the major browsers.

4.3 Cross-browser Issues

When developing and designing a web site or application, one major issue commonly arises: *Cross-browser compatibility*. The front end of the application, even when written in the standardised HTML, CSS and JavaScript formats, are rendered by many different web browsers all depending on the installation or user preferences. Support for specific HTML and CSS features vary from browser to browser, and some browsers render objects differently than others. These problems lie within each browser and depend on the layout engine in use. The layout engine is the rendering mechanism that a browser uses to convert HTML and CSS into a visual representation. There are currently five rendering engines in wide use [51]:

- **Trident** - The engine used by Internet Explorer.
- **Gecko** - The engine used by Firefox.
- **WebKit** - Used by Safari, older versions of Chrome, and most mobile smart phones (iPhone, Android, etc.).
- **Presto** - The engine used by older versions of the Opera browser.
- **Blink** - New (April 2013) engine made by Google, used in new versions of Chrome and Opera.

Not only can the layout engines of web browsers cause unwanted compatibility issues, but browsers may also use different engines for JavaScript interpretation. A quick and easy solution to this problem is, as outlined, to use a JavaScript library (like jQuery), which often masks these inconsistencies.

The cross-browser compatibility issues generally arise from the lack of implemented standards within the structure and layout languages (HTML and CSS). However, with the arrival of HTML 5 and CSS 3 in the last few years, each engine's compliance with the standards have greatly improved. Introducing standards opens the door for multiple rendering clients, thereby expanding web content beyond the web browser and into desktop applications and mobile phones as well.

4.3.1 Multimedia

Out of all the cross-browser compatibility issues, support for playback of multimedia in browsers is definitively one of the most cumbersome. Playing a single file of a specific format, in one specific browser, is generally an easy task. However, when this file needs to be played in multiple different browsers and versions, the task can quickly become complicated. Recently, the easiest method for playback of multimedia has been to upload content to Google's multimedia hosting service YouTube [25] and use their integrated player to display the content. However, when demands go beyond those of the average user, this may no longer be an option. For the implementation of MAT, YouTube is not a feasible option due to embedded watermarks and links to other content hosted by their service. Moreover, the quality of the content must conform to pre-defined formats that are below acceptable levels. Yet another issue is that uploaded multimedia content is accessible by anybody visiting the site.

With the arrival of HTML 5, and hereby the new <*video*> and <*audio*> tags, support for multimedia was greatly improved. Recent versions of the major browsers now support a small number of multimedia formats.

However, without third-party browser plug-ins installed, older versions have limited or no support for playback. Luckily, a major part of the clients connected to the Internet today use fairly updated browsers. Those who do not, tend to have third party plug-ins installed [47], for example the popular Adobe Flash Player. A problem still lies within the fact that different browsers support alternate distinct formats. Currently, there are three supported video formats for the <video> element: MP4 (H.264/AAC), WebM (VP8/Vorbis), and OGG (Theora/Vorbis). As seen in Table 4.1, support for each format differs from browser to browser [43].

Browser	MP4	WebM	OGG
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	NO*	YES	YES
Safari 5+	YES	NO	NO
Opera 10.6+	NO	YES	YES

Table 4.1: Video format support across primary browsers.

Similarly, there are three supported formats for the <audio> element; MP3, WAV, and OGG (Vorbis). Again, support for each type differs, as seen in Table 4.2. The asterisk on Firefox's MP3 and MP4 formats indicate that these types were not supported until recently, when they were added to the new browser versions. [43].

Browser	MP3	WAV	OGG
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	NO*	YES	YES
Safari 5+	YES	YES	NO
Opera 10+	NO	YES	YES

Table 4.2: Audio format support across primary browsers.

An efficient way to solve the issue of multimedia playback in browsers, is to convert media files into each of the specified formats and make all the converted files available to the client. In this way the client's browser can choose which file to use, depending on the format it supports. However, this does not work if the client uses an older browser without support for HTML5. Multimedia playback is essential to our assessment tool, so a solution to this problem will be described in the design and implementation chapter.

4.4 Alternatives

As we mentioned briefly in Section 4.1, there exists many different set-ups for running applications like ours. For example, several variants of the LAMP software stack have been outlined. However, LAMP and its variants are not the only approach to running efficient web applications. Oracle's Java Server Pages (JSP) [9], Microsoft's Active Server Pages (ASP) and .NET Framework [8] are also some of the more popular alternatives. With these choices available, why did we decide to use the set-up that we have? We had a lot of experience with the LAMP software and the PHP programming language from earlier, as well as the web technologies mentioned in Section 4.2. Learning new languages and web server systems may have been unnecessarily time-consuming for the purpose of this project. In addition, web servers using the LAMP software was already up and running at the intended hosting location. Moreover, the chosen system adequately fills the needs and requirements of the application we were to develop.

Another possibility was to develop a none browser-based application, or in other words, a standalone program. The problem with this approach is that it is somewhat more difficult to implement and that it defeats one of the main purposes of our application; to be easily available to the users, anywhere, at any time. Having to download and install a program is not what was initially intended.

With the recent rise in amount of people owning newer generation tablets and mobile phones, or smart-phones, yet another possibility could have been to develop this assessment platform for mobile phones, or even as a mobile application. To evaluate multimedia for mobile phones, this content has to be assessed on a mobile phone as well. Although assessment of this kind of multimedia was not a initial goal of the thesis, we can see the potential for expanding to that market in the future. HTML 5 is actually well suited and integrated in recent smart-phones and tablets, so facilitation for this feature may be in place already. However, it is not something that will be further explored in the scope of this thesis.

4.5 Summary

In this chapter we have presented the technologies and frameworks that form the foundation of the design and implementation of our assessment tool. This consists of the LAMP software for the server, and any major web-browser utilising HTML, CSS and JS for the client. The LAMP software comprises of the Linux operating system running an Apache web server, MySQL database management system, and the dynamic scripting software PHP. This is a set of simple, yet adaptable, free and open source software, which adequately meets our requirements for MAT. The front end of our application is designed to run in a web-browser, taking advantage of the recently updated HTML 5 and CSS 3 standards for structure and layout of the web interfaces. Moreover, JS is used for added interactivity and validation of user input forms.

Furthermore, the issue of cross-browser compatibility arises when using the aforementioned web standards. Since web-browsers use different rendering engines to convert HTML, CSS and JS into a visual representation, the same content may be presented differently in various browsers. Even more problematic is the support of playback of multimedia content. HTML 5 has greatly improved on this issue, but there are still some challenges with multimedia format support in different browsers that needs to be overcome.

Chapter 5

Design & Implementation

Developing the Multimedia Assessment Tool (MAT) involved a series of of steps, each with its own challenges. This chapter outlines the details of the design and implementation of the MAT software, along with an overview of the functionalities and possibilities that the newly developed tool can deliver.

5.1 System Overview

The overall system of MAT is based on the distributed application structure known as the *client-server* model. Portrayed in Figure 5.1 is an example of how HTTP, the communication protocol of the World Wide Web, works according to the client-server model. Chapter 4 describes standard frameworks that are commonly used as server or client software in this type of model, and accordingly, we have applied the same frameworks in the development of our framework. Specifically, the LAMP application stack acts as our server and provides resources. Clients, in the form of web-browsers, may then request said resources. Since the resources are commonly in the form of HTML documents and multimedia files, the client browser will use these to display the graphical representation of our multimedia assessment tool.

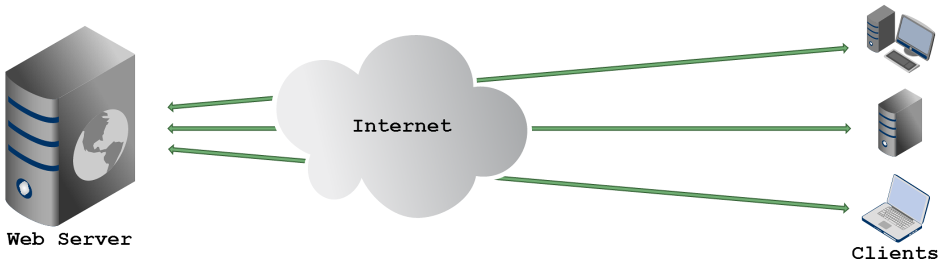


Figure 5.1: Server-client communication over the World Wide Web.

Going further into the system as a whole, we find our application, the Multimedia Assessment Tool. This is mainly developed in PHP, with elements of SQL statements that interact with the database software. The application also includes HTML templates for the front-end structure, Cascading Style Sheets for the look and layout, and JavaScripts for added interactivity. Moreover, the application is developed from scratch using the object-oriented programming (OOP) paradigm, and it complies to a specific software architecture pattern, described in 5.2. The many benefits of utilising OOP and specific architecture patterns will be explained in the following sections. Building the application from the ground up also brings its own set of advantages, most noticeably the opportunity to design the application to our exact requirements.

5.2 Software Architecture

MAT follows the software architecture pattern commonly referred to as Model-View-Controller (MVC). This architecture is special in the way it separates the representation of information from the user's interaction with it. Three components make up the architecture, namely the Model, the View, and the Controller. Furthermore, the architecture defines the interactions between these components. A representation of the MVC architecture and its interaction patterns used in web application development is outlined in Figure 5.2.

- The **Model** is the portion of the application that contains both the information and the logic to manipulate this information. In web applications, this commonly means retrieving, inserting or updating information in databases.

- The **View** displays information to the user and comprises the application's user interface. A view can be any output representation of data, however in our case it is generally rendered using HTML.

- The **Controller** is the brains of the application. It decides how to process the user's interaction, how the model needs to change as a result of that input, and which resulting view should be used.

The separation of business logic and presentation, the principal idea behind MVC, is commonly favoured in application development because it introduces a number of benefits:

- The application's look and layout can be drastically changed without altering data structures and business logic.

- The application can easily maintain different user interfaces, such as multiple languages, or different sets of user permissions.

- Reuseability of both code and content is greatly increased. Enforces the "don't repeat yourself" (DRY) principle.

- It provides a high level of semantic control and facilitates robust APIs, standards and patterns.

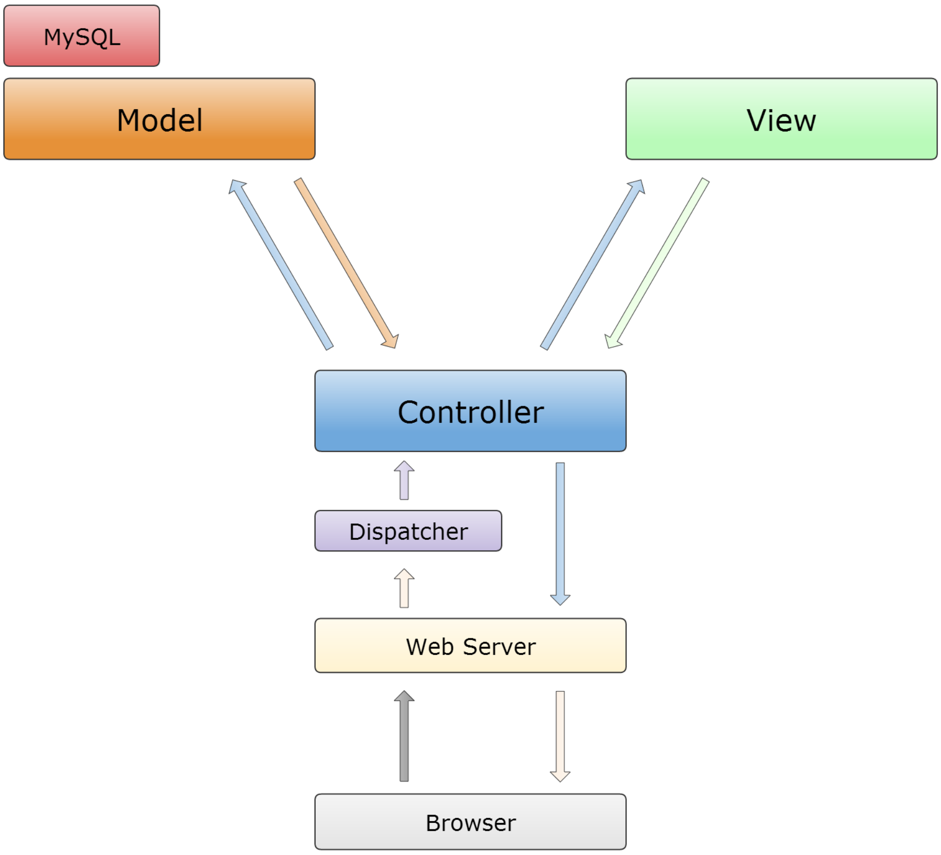


Figure 5.2: MVC components and interaction pattern.

MVC has been widely adopted as an architecture for web applications in several major programming languages, with many commercial and non-commercial application frameworks enforcing this pattern. However, in conjunction with web application development, MVC commonly varies slightly with regards to the developers' interpretation, often resembling the very similar Presentation-Abstraction-Control (PAC) design pattern. Consequently, MVC is used to describe the way changes to an applications Model are driven by a Controller that is responsible for processing user interactions and dictating logic that changes the application's overall state in response to the event created by the user's interaction. Furthermore, the Controller initiates creation of the application's new View in response to these changes in the Model [37].

5.3 File Structure

The architecture pattern makes a visible impact on the corresponding file structure used in the application, as presented in Figure 5.3. The code base itself lies within the *application* folder and is split into three parts; *controllers*, *models* and *views*. These contain the class files for the controllers, the models, and the view, respectively. With the great possibilities that follow MVC's separation of presentation logic, the views folder is also enabled to support yet another folder hierarchy for maintaining separate user interfaces. Our application currently only contains one interface, excitingly enough named *default*.

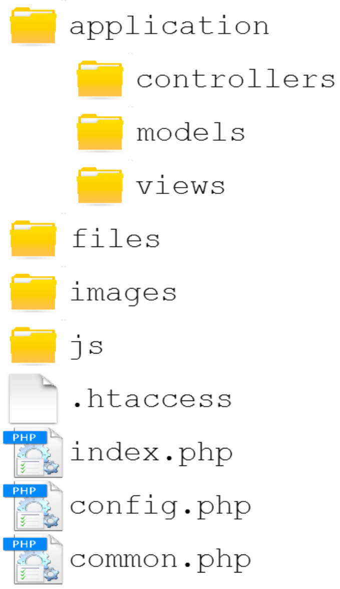


Figure 5.3: Application file structure.

Next we have the *files* folder, which contains files uploaded to the server using that feature of the application. Uploaded files will largely be comprised of multimedia content that are to be used in experiments, but the folder could technically contain any type of file one may wish to store on the server. The *images* folder is for storing resources used by the application, commonly in the form of images. Images tend to greatly increase the appeal of the user interface, and they add character to both the look and feel, and interactivity of the presentation layer. Finally there is a folder for the JavaScript files used by the application to ensure greater interactivity and control of content on the client side. The main folder also holds four essential files for overall application data flow: *.htaccess*, *index.php*, *config.php*, and *common.php*. The important functions these files provide are explained further in Section 5.6.

5.4 Database Structure

To store the data our application will produce and procure during runtime, we need an efficient database management system to interface with the scripting language that controls the whole operation. As determined in Section 4.1, MySQL is especially well suited for this task, and is consequently used in our application. MAT only requires a single MySQL database, in which several tables reside. These tables hold the information used by the application. Table 5.1 lists the incorporated tables and provides a short description of each.

Table	Description
settings	All application settings in key-value pairs.
errors	Errors that have occurred during runtime.
log_activity	Information about user's last activity.
users	All data about user accounts.
user_groups	Information about user groups.
user_in_group	Specifies user<–>group relationships.
user_done_survey	Specifies user<–>survey relationships.
surveys	All survey data, excluding questions.
question_groups	Question grouping information.
question_types	Valid question types and corresponding data.
question_options	Custom response options for certain questions.
questions	All question data, excluding response options.
answers	Every answer from every survey/experiment.

Table 5.1: Descriptions of the database tables used in the application.

There is a good reason why we have split the data into several tables. In database design a key goal is to achieve *data normalisation*, which is a technique to organise the contents of the tables for transactional databases. When normalising a database, four goals present themselves:

- Arranging data into logical groupings so that each group describes a small part of the whole.
- Minimising the amount of duplicate data stored in a database.
- Organising the data so that any changes made will be isolated to one place.
- Building a database that allows quick and efficient access to manipulate data without compromising the integrity of the data in storage.

In order to achieve all four goals, MAT groups data in separate tables, as described. Moreover, the following principles must be adhered to. Each table, except for settings, consist of a primary key column that identifies each row of data. Additionally, several tables include foreign keys, that indicate relationships with other tables. Take the *question_options* table as an example: It contains the primary key column *id_option* which identifies each option by an integer, and a foreign key column *id_question* that refers to the primary key of the *questions* table. This foreign key creates a relation between *question_options* and *questions*, in the form that each question option belongs to one question. These principles are presented in Figure 5.4, where the lines represent relationships between tables.

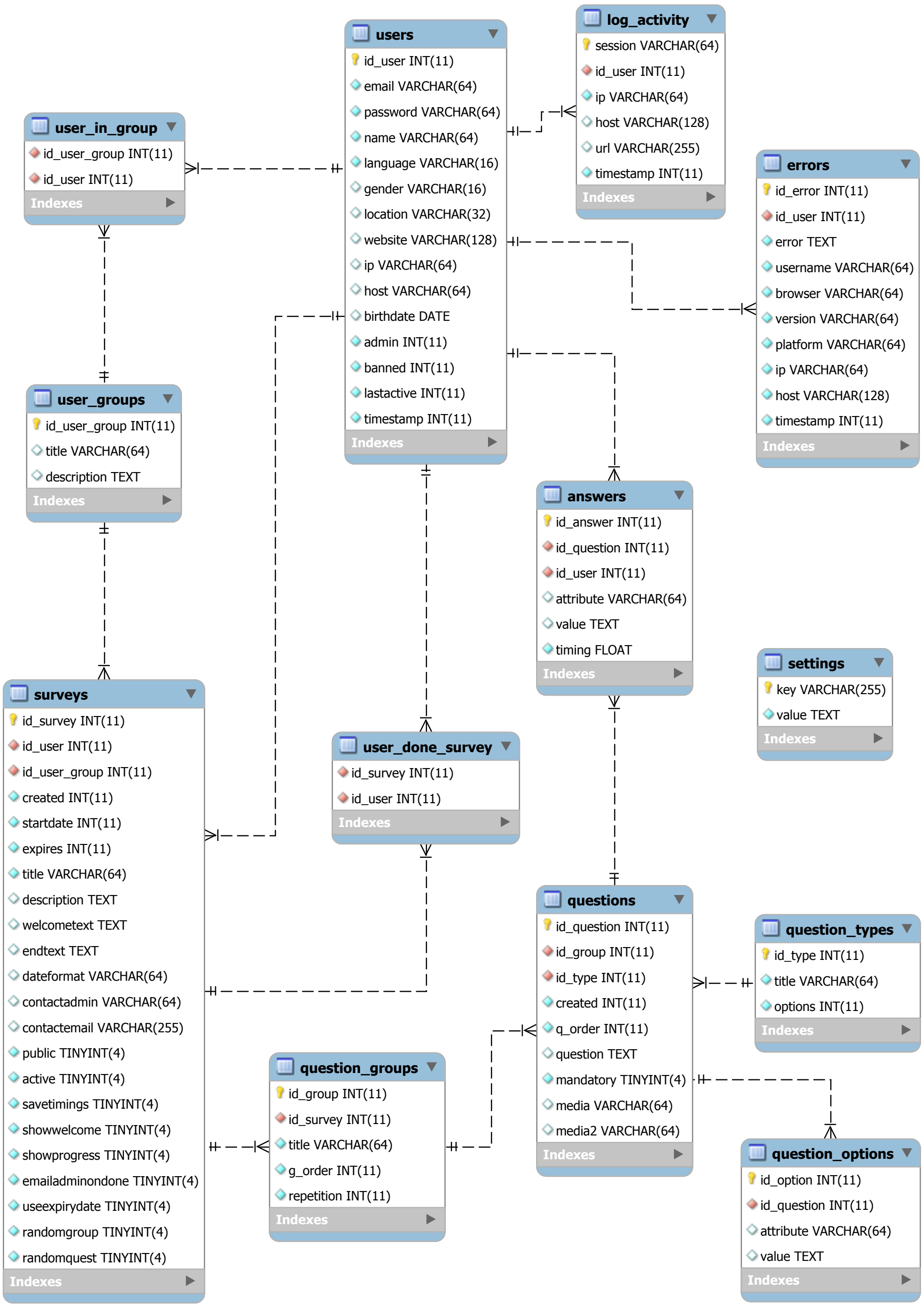


Figure 5.4: Entity-relationship model of the database.

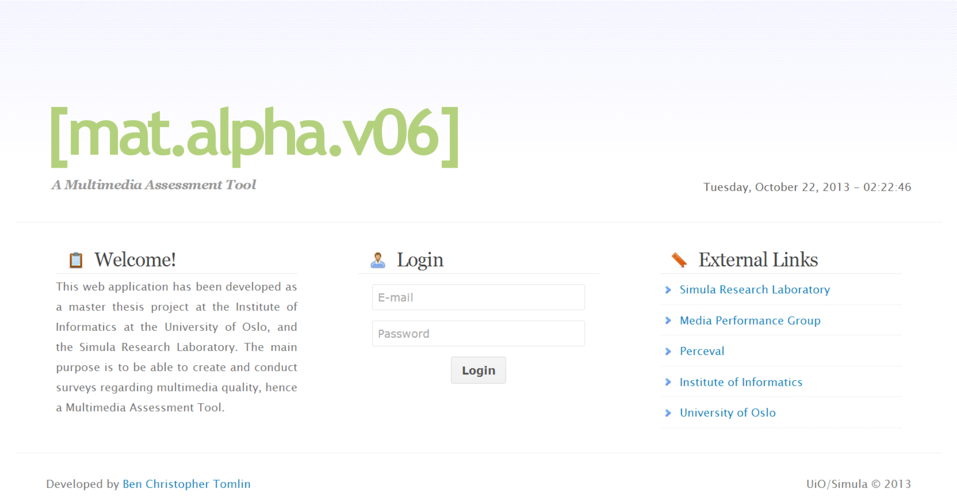
A prominent issue in the current configuration of the database relates to the *answers* table. Every single answer given by a participant, to every partaken question, is stored in this table. In turn, a massive amount of rows will be inserted in this single table. For example, if one experiment consists of 50 questions and there are a total of 100 participants, that would amount to 5000 entries for that experiment alone. Although this set-up is functional, there may be a noticeable scalability issue with the approach. The problem could be solved by having separate answer tables for each survey, however, scalability has not been deemed a critical problem in this project and is something we have set aside for possible future work.

5.5 Client Composition

In this section, the details of the client's composition are described for all steps from the first log-in screen to the more complex experiment management. In this context, clients refer to the graphical representation and interactions of the application as presented in a user's web-browser.

5.5.1 Graphical Design

When designing the look and feel of the application, the decision fell on a simple, but efficient design. This decision was not only based on the lack of graphical design experience, it was mainly inspired by a wish to develop a system that is easy to use rather than fancy to look at. Figure 5.5 shows the interface as the user would first encounter it, prior to log-in. The background is a standard white colour, with a simple blue gradient at the top. The light blue theme follows through most of the application, with a similar colour for interactive text, and dark, easily readable text elsewhere. Upon log-in, a user will find simple navigation to all relevant sections of the interface. Note that the title at the top of the interface is currently only a placeholder for an alternative title yet to be specified. Figure 5.6 includes an example of what is commonly called a link-tree. Link-trees make navigation easier and gives a clearer representation of one's current location in the application.



[mat.alpha.v06]

A Multimedia Assessment Tool

Tuesday, October 22, 2013 – 02:22:46

Welcome!

This web application has been developed as a master thesis project at the Institute of Informatics at the University of Oslo, and the Simula Research Laboratory. The main purpose is to be able to create and conduct surveys regarding multimedia quality, hence a Multimedia Assessment Tool.

Developed by Ben Christopher Tomlin

UiO/Simula © 2013

Figure 5.5: The interface's front page, including the log-in form.

5.5.2 Login & Sessions

Logging in with a user account follows the standard procedure that requires a username or e-mail address, in addition to a password, in order to securely identify the user. However, a few additional measures are needed to ensure that the correct information is entered, that the right user and corresponding data is loaded, and that the correct content is presented:

1. Check that the entered user-identifying name or e-mail address is valid, and validate that this user exists in the database.
2. Make sure the entered password, after encryption processes, matches the encrypted password stored under the selected user in the database. If this does not match, the user is presented with an error message that the information entered is invalid.
3. Following input of the correct log-in data, the application will load data specific to the user, and set permissions accordingly.
4. Depending on the user's assigned permissions (participant or experimenter), a list of navigational links is presented. For the participant, this involves navigation to their user account settings, a list of available experiments to undertake, and a logout option. Experimenters additionally have the option to enter the administration feature.

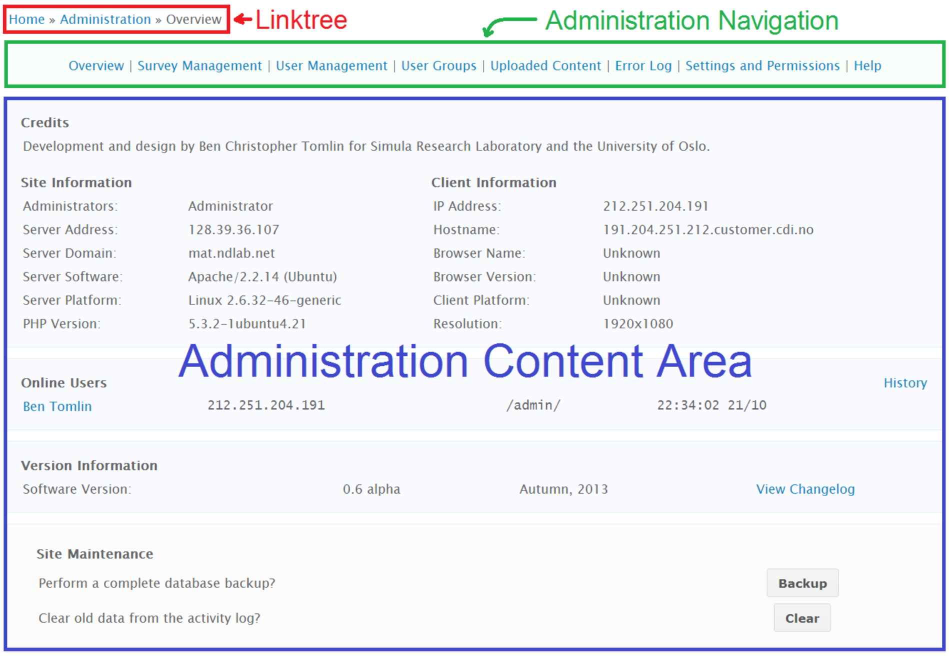
Since the HTTP communication protocol is stateless it provides no means of storing a user's data between requests. This introduces a challenge when providing a service which facilitates a user separation feature. Consequently, HTTP servers implement session management methods, that typically utilise a unique identifier in a cookie or parameter. In turn, the server is able to track requests originating from the same client, and can effectively create a stateful protocol on top of HTTP. Thus, session management is something our application will be designed to handle.

Because of such demands, PHP comes with a feature to handle sessions easily and efficiently. Upon log-in and user authentication, the application stores vital identification information in PHP's session variable, which thereafter is available upon further communication requests. However, sessions do not last indefinitely. Commonly a session expires after a set amount of time, or when a user closes the web-browser and a new session has to be started. To ensure that users can stay logged in and remain able to use the application for a longer period of time, cookies are also used to maintain the sessions. Cookies can be set by the application, but they are stored on the client, which tends to prolong the session's lifetime. As the session itself is stored on the server, it is reasonably secure; however, cookies are not. For added security, the data used in the cookies is encrypted.

5.5.3 Administration

Following a successful log-in, an experimenter with an administrator account may navigate to the administration section of the application.

This is the location for all the management of experiments (surveys), participants and the application overall. Figure 5.6 shows a representation of the administration section, where the navigation bar (green) and the area where the administration features are displayed (blue).


Home » Administration » Overview
Linktree
Administration Navigation
Overview | Survey Management | User Management | User Groups | Uploaded Content | Error Log | Settings and Permissions | Help
Administration Content Area

Credits

Development and design by Ben Christopher Tomlin for Simula Research Laboratory and the University of Oslo.

Site Information		Client Information	
Administrators:	Administrator	IP Address:	212.251.204.191
Server Address:	128.39.36.107	Hostname:	191.204.251.212.customer.cdi.no
Server Domain:	mat.ndlab.net	Browser Name:	Unknown
Server Software:	Apache/2.2.14 (Ubuntu)	Browser Version:	Unknown
Server Platform:	Linux 2.6.32-46-generic	Client Platform:	Unknown
PHP Version:	5.3.2-1ubuntu4.21	Resolution:	1920x1080

Online Users — History

Ben Tomlin	212.251.204.191	/admin/	22:34:02 21/10

Version Information

Software Version:	0.6 alpha	Autumn, 2013	View Changelog

Site Maintenance

Perform a complete database backup? Backup

Clear old data from the activity log? Clear

Figure 5.6: The administration section of the application.

- **Overview**

  The overview section displays general information about the application, the server it is running on, and the client viewing it. This includes version information with a link to the change-log file, as well as a list showing who is currently using the application, and the option to display expired activity data. Furthermore, two maintenance options are included at the bottom, one for running a full backup of the database and one to clear expired activity data.

- **Survey Management**

  Under survey management there is a list composed of every survey that has been created, including both active and expired surveys. Selecting one of these leads to the specific survey management interface, where one may edit any part of the survey. A button is included for creating a new survey or experiment. Experiment setup and management will be explained in more detail in section 5.5.4.

- **User Management**

  Similar to the survey management, a list of currently registered users are displayed for user management, which lets experimenters administrate user accounts. This involves creation of new experimenter and participant accounts, editing existing account data, and deactivation or deletion of accounts.

- **User Groups**

  The user group feature is intended for creating groups of participants that are to partake in an experiment. Each survey can offer the selection of one group of users as its participants. Thus, this section presents a list of existing user groups, with the added option to create new groups. Selecting a user group brings up the interface for editing the group and, moreover, the possibility of adding or removing participants from it. Additionally, when creating new user groups, an experimenter can include as many participants as desired in the form of their e-mail addresses. The application then adds all existing users with these addresses to the group; if they do not already exist, a new user account is created based on that e-mail address.

- **Uploaded Content**

  In this section, all files that are uploaded to the server using the application are listed. These files are generally multimedia files that are to be used in experiments. Furthermore, each file can be viewed individually or deleted if needed.

- **Error Log**

  The error log lists errors that have occurred within the application, as well as detailed information about how the error occurred. These errors can be everything from log-in failures to script malfunctions or database errors. This error-reporting feature is important for both users and developers, enabling them to easily detect and locate problems in the application's code.

- **Settings and Permissions**

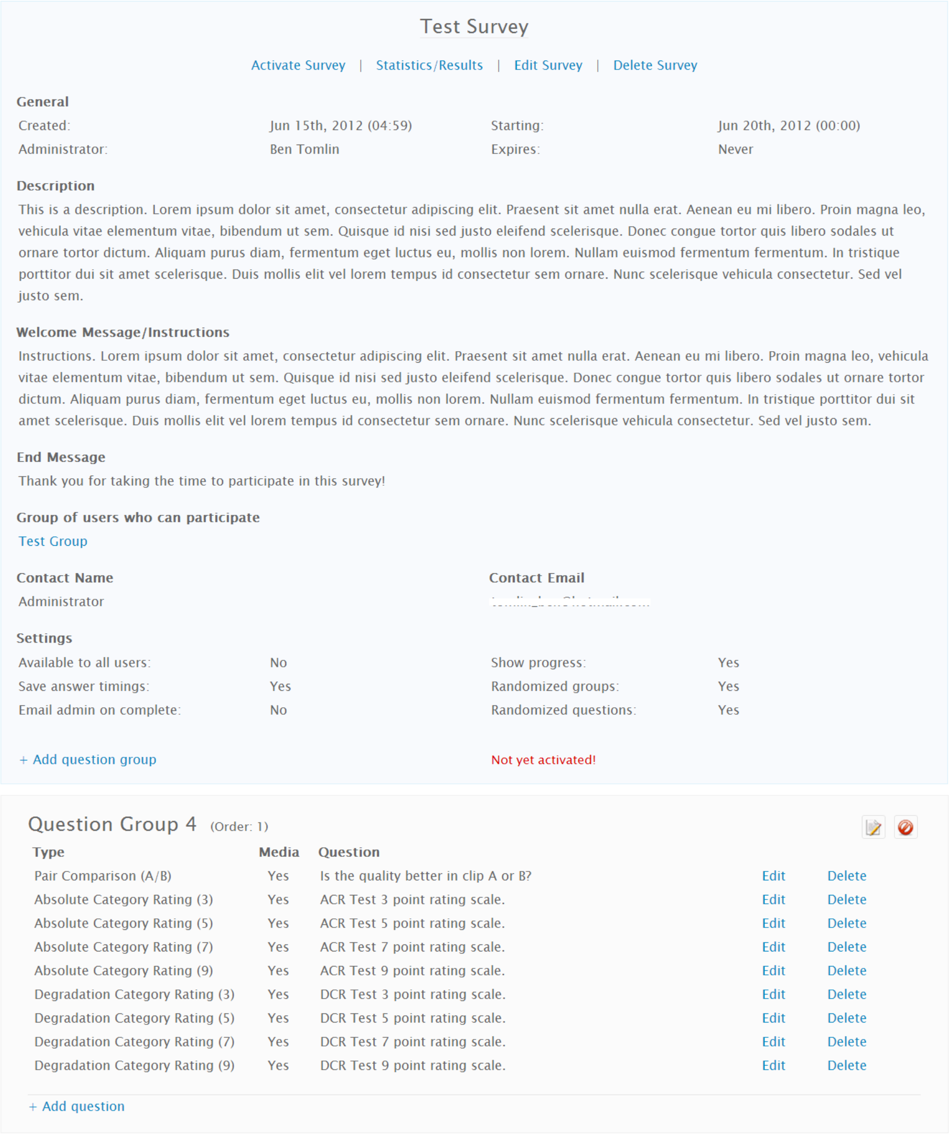
  This section lists all editable settings and permissions currently enforced throughout the web application, which can all be altered with a few simple clicks.

- **Help**

  Finally there is a help section, that provides short descriptions of how to use the various administration features.

5.5.4 Experiment Setup

Creating and setting up an experiment using our assessment application is intended to be quick and simple, even for non-technical people. As mentioned in the previous section, a new survey can be created by entering the survey management section of the administration interface. Once initiated, a form containing several fields and options must be completed. The more important fields include time and date for experimentation start and end, selection of which group of users that may participate, and



Test Survey

Activate Survey | Statistics/Results | Edit Survey | Delete Survey

General

Created:	Jun 15th, 2012 (04:59)	Starting:	Jun 20th, 2012 (00:00)
Administrator:	Ben Tomlin	Expires:	Never

Description

This is a description. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet nulla erat. Aenean eu mi libero. Proin magna leo, vehicula vitae elementum vitae, bibendum ut sem. Quisque id nisi sed justo eleifend scelerisque. Donec congue tortor quis libero sodales ut ornare tortor dictum. Aliquam purus diam, fermentum eget luctus eu, mollis non lorem. Nullam euismod fermentum fermentum. In tristique porttitor dui sit amet scelerisque. Duis mollis elit vel lorem tempus id consectetur sem ornare. Nunc scelerisque vehicula consectetur. Sed vel justo sem.

Welcome Message/Instructions

Instructions. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet nulla erat. Aenean eu mi libero. Proin magna leo, vehicula vitae elementum vitae, bibendum ut sem. Quisque id nisi sed justo eleifend scelerisque. Donec congue tortor quis libero sodales ut ornare tortor dictum. Aliquam purus diam, fermentum eget luctus eu, mollis non lorem. Nullam euismod fermentum fermentum. In tristique porttitor dui sit amet scelerisque. Duis mollis elit vel lorem tempus id consectetur sem ornare. Nunc scelerisque vehicula consectetur. Sed vel justo sem.

End Message

Thank you for taking the time to participate in this survey!

Group of users who can participate

Test Group

Contact Name	**Contact Email**
Administrator	[redacted]

Settings

Available to all users:	No	Show progress:	Yes
Save answer timings:	Yes	Randomized groups:	Yes
Email admin on complete:	No	Randomized questions:	Yes

+ Add question group

Not yet activated!

Question Group 4 (Order: 1)

Type	Media	Question		
Pair Comparison (A/B)	Yes	Is the quality better in clip A or B?	Edit	Delete
Absolute Category Rating (3)	Yes	ACR Test 3 point rating scale.	Edit	Delete
Absolute Category Rating (5)	Yes	ACR Test 5 point rating scale.	Edit	Delete
Absolute Category Rating (7)	Yes	ACR Test 7 point rating scale.	Edit	Delete
Absolute Category Rating (9)	Yes	ACR Test 9 point rating scale.	Edit	Delete
Degradation Category Rating (3)	Yes	DCR Test 3 point rating scale.	Edit	Delete
Degradation Category Rating (5)	Yes	DCR Test 5 point rating scale.	Edit	Delete
Degradation Category Rating (7)	Yes	DCR Test 7 point rating scale.	Edit	Delete
Degradation Category Rating (9)	Yes	DCR Test 9 point rating scale.	Edit	Delete

+ Add question

Figure 5.7: The experimentation setup and control interface.

definitions of instructions prior to the commencement of surveys. A set of additional options for the surveys are listed below:

- Make the survey available to any user.
- Record the amount of time it takes a participant to answer.
- Show progress during participation, e.g. questions remaining.
- Randomise the order in which the question groups are presented.
- Randomise the order of questions within each question group.

Once this information has been saved, hence the survey has been created, work can continue with adding questions and question groups. Figure 5.7

shows a test experiment completed with some arbitrary data to give an example of how the experimentation control interface looks like.

Question Groups

Question groups currently only have one purpose, which is to group certain questions together. This may be desirable if a logical order of questions is of importance, or if randomisation within blocks of questions is needed. Otherwise, grouping may be skipped all together by placing all questions within one single group.

Questions

The most essential part of any assessment study are the questions. Questions can come in several different formats, but our application is obviously aimed at supporting questions in the form specified by the subjective methodology explained in Chapter 3. Table 5.2 presents a list of the predefined question types that are currently supported in MAT. Note that the multiple choice option is somewhat special in that it allows experimenters to define their own response options as well, for example if the standard ACR/DCR scales are insufficient. Furthermore, experimenters may define the wording of any question, the order in which the question should be presented, and select which multimedia content should be displayed, if any.

Assessment Methods	
Absolute Category Rating	3-, 5-, 7- or 9-point ACR scales.
Degradation Category Rating	3-, 5-, 7- or 9-point DCR scales.
Pair Comparison	Preference on option A or B.

Miscellaneous	
Single Line	Short answer in text format.
Multi Line	Long answer in text format.
Multiple Choice	Radio button, checkbox or dropdown.
Yes/No	Positive/negative answer options.
Gender	Gender selection.
Date	Selectable day, month and year format.
Numeric	Answer has to be numerical.

Table 5.2: Available question types in the application.

Evidently, the long duration assessment methods SSCQE and QELDAC are not yet implemented, but they are definitively something that could be looked at for future work. The main reason they are not currently included is due to the complexity of implementing these methods using the technologies we have available on the client side. SSCQE will need a form of combining a slider with the media playback, in order to register slider positions at certain intervals during presentation. Similarly, QELDAC will also need a slider, though tied into the quality adjustment of the multimedia sequence.

Activation

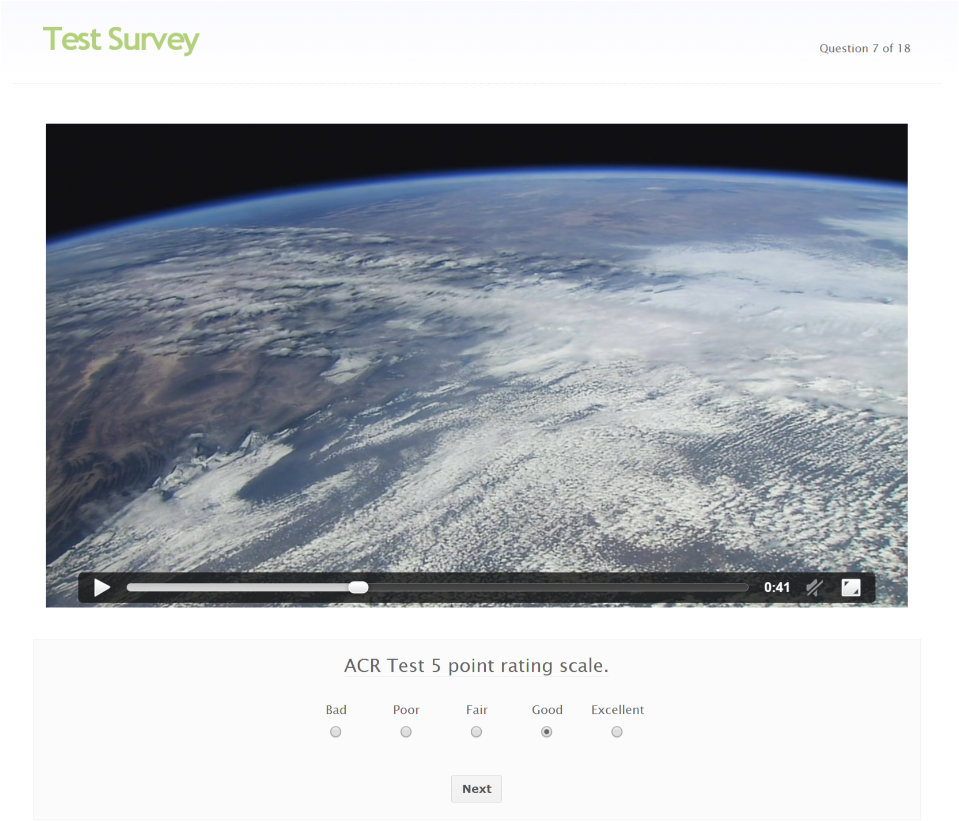
Once everything has been filled out, options have been set and a sufficient amount of questions have been added, the next step is to activate the experiment. The activation process runs through some pre-checks to see that everything is in order, after which the experiment is set to active. Once the survey start date and time has passed, the experiment will be freely available to the selected participants. The experiment control interface now also includes an option to send an invitation e-mail to the participants, asking them to take part in the survey and specifying the required user account information needed to log in and participate.

5.5.5 Conducting Experiments

As with most online survey tools, the conducting of an experiment follows a specific set of steps. Firstly, an introduction screen is shown, commonly describing the survey and addressing what the experimenters want to know. Typically, it should also include a disclaimer or consent form of some sort, depending on the type of data one may wish to gather. Specifying that all answers are given anonymously and that the data is only being used for the specific purpose described in the introduction, is also displayed. Following this introduction, the questions (or experiment test items) are presented, one by one in the order specified by the experimenter. Figure 5.8 shows the presentation of an example ACR test during an experiment. Here, the multimedia sequence under scrutiny is displayed together with the rating scale options for this test method. A participant would in this case view the sequence, select a preferred option and thereby press the appropriate button to proceed to the next test. This is repeated until all tests have been answered, and the participant arrives at end-screen, which consequently displays some text thanking the participant for the involvement in the experiment.

These steps are designed in a simple, but effective manner, making the participation as easy and convenient as possible for the user. When the experiment is started, all data needed by the application to display the aforementioned steps are loaded into the user's session variable on the server. Using the session as temporary data storage makes for some particular benefits:

- Less strain on the database, as information is only loaded once at the beginning and answers are only saved once at the end.



ACR Test 5 point rating scale.

○ Bad ○ Poor ○ Fair ● Good ○ Excellent

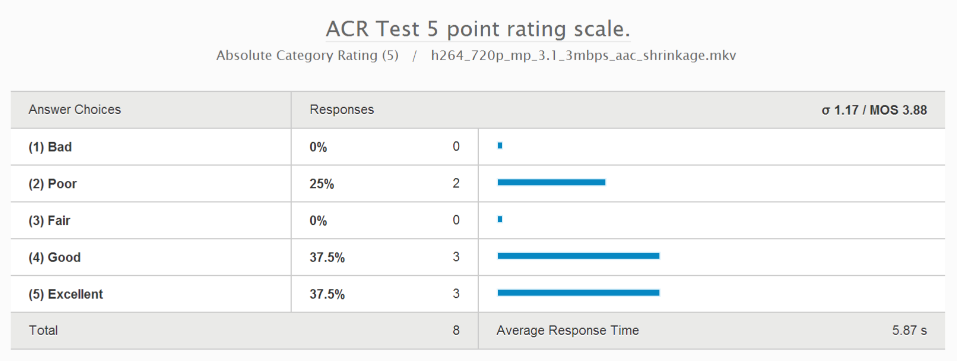
Figure 5.8: ACR test on a video sequence during an experiment.

- The session can be used to keep track of the participant's progress and, because the session lasts for a set amount of time, the participant may close the application window and at any time continue from the same point in the experiment.
- As answers are not saved to the database until the end of the experiment, answers from participants who may withdraw midway in an experiment is not incorrectly added to the set of results.

The apparent downside to using the session in this way is that it applies slightly more strain on the server software itself. Moreover, the session variables do not last indefinitely, as previously explained. However, as we presume that most participants finish the whole experiment once begun, it does not present a huge issue.

5.5.6 Results

An interface for presenting the results and statistics of an experiment is of course included in the application. This is easily accessible from the survey control interface as shown i Figure 5.7. This section displays statistics on each and every question from the experiment.



ACR Test 5 point rating scale.

Absolute Category Rating (5) / h264_720p_mp_3.1_3mbps_aac_shrinkage.mkv

Answer Choices	Responses		σ 1.17 / MOS 3.88
(1) Bad	0%	0	
(2) Poor	25%	2	
(3) Fair	0%	0	
(4) Good	37.5%	3	
(5) Excellent	37.5%	3	
Total		8	Average Response Time 5.87 s

Figure 5.9: Results/statistics of a question from a test experiment.

Figure 5.9 shows a representation of the results for a single question from a test experiment. Question wording, type and selected multimedia are displayed at the top, followed by a presentation of the data including percentages, MOS, standard deviation and an average response time. Moreover, all this data is available in a downloadable spreadsheet format for further statistical analysis.

5.6 Implementation

This section will provide a more technical approach to how the application has been developed. However, with more than 7,000 lines of code, not including comments or blank lines, we will not be going over every single aspect of the implementation process. The development has more or less followed the same coding standard throughout, providing clean and easy to read code. Every class and function is well documented, including descriptive comments for most code within the functions, which allows possible future developers to easily read and understand the code.

5.6.1 Apache Configuration

In Section 5.3 we mentioned a few files that are placed in the top-level directory of our application. One of these was *.htaccess*, a directory-level configuration file supported by several web servers, including Apache. This file consists of configuration commands that override a subset of Apache's global configuration for the directory in which it is placed, and all subdirectories. Although this file can override many configuration settings, we use it for one particular purpose; Enabling the server's rewrite engine. The rewrite engine software enables the web server to modify a web URL's appearance, hence called URL rewriting. By making URLs shorter and looking more relevant, this technique adds a layer of abstraction between the files used to generate the graphical interface and the URL that is presented to the users.

```
RewriteEngine On
```

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?url=$1 [PT,L]
```

The four lines presented above are all that is needed to provide this feature. While the first line enables the rewrite engine, the second and third lines exclude URLs pointing to existing files and directories from being rewritten. The last line does the rewriting itself, taking all URL query string parameters and redirecting them to the file *index.php* under the single parameter named *url*. In reality, this means that URLs that are commonly written in the following format

```
http://example.com/index.php?content=surveys&action=edit&id=745
or
http://example.com/surveys.php?action=edit&id=745
```

can be rewritten as

```
http://example.com/surveys/edit/745/
```

which under the abstraction will end up as

```
http://example.com/index.php?url=/surveys/edit/745/
```

Moreover, this technique allows the *index.php* file to handle all requests, making the application more streamlined and the code more structured. Since everything is run from *index.php*, it is hereby known as the *dispatcher*.

5.6.2 Dispatcher

The dispatcher, as depicted in Figure 5.2, is responsible for everything happening before and after handing operation over to the Controller. This includes loading and initiating the correct Controller, but also a few other measures of importance:

- Initialise several constants and variables used throughout the application.
- Load the *config.php* file, which includes configuration settings like the database connection details.
- Load the *common.php* file, which includes common variables and functions used throughout the application.
- Register an autoloader function on PHP's autoload stack (queue), which handles automatic loading of class files further on in the application's context flow.
- Parse the URL query string given in the *url* parameter passed by the web server, as explained in Section 5.6.1. This parsing splits the query string on every forward-slash (/) character.

- Using the parsed parameters from the previous step, it dispatches a Controller class based on the first parameter, calls a function within this Controller based on the second parameter, and passes any additional parameters to the aforementioned function. For example, lets say the web server passes the query string "/surveys/edit/745/", which is parsed into three corresponding parameters. The dispatcher would consequently initiate the Controller class *Surveys*, for then to call its *edit()* function while also passing the parameter *745*.

- After the code within the specified Controller and underlying classes and functions have been executed, the dispatcher calls the *render()* function within the Controller's instance of the View, generating the HTML content before sending it to the client.

5.6.3 MVC

As mentioned in Section 5.2, the application is designed and implemented following the Model-View-Controller (MVC) architecture pattern. This necessitates a separation of logic into different classes, spread out over several files stored in a particular hierarchical structure, as shown in Section 5.3. These classes are separated into three groups; Controllers, Models and Views. Figure 5.10 presents a condensed class diagram of the system, showing the class structure, attributes, operations and relationships. Furthermore, we will describe the implementation of each group of classes in the following sections.

Controllers

The controllers are, not surprisingly, the part of the application that controls the logic and interaction between the models and views. We have one controller super-class, and several controller sub-classes which handle specific parts of the application separately. For example, the *Surveys*-controller handle everything to do with surveys (experiments), while the *Questions*-controller manages everything to do with individual questions. On each request by a client, one controller is instantiated by the dispatcher, depending on the query string. In any case, the super-class' constructor function is called, which manage a number of highly important tasks, as listed in the following pseudo code:

```
1.  initialise a new View()
2.  initialise a new DB()
    2.1  if db connection failed or tables do not exist
         2.1.1  set fatal error
         2.1.2  run view->render()
         2.1.3  exit
3.  load settings, session, activity Models
    3.1  settings->load()
    3.2  session->load()
    3.3  activity->log()
```

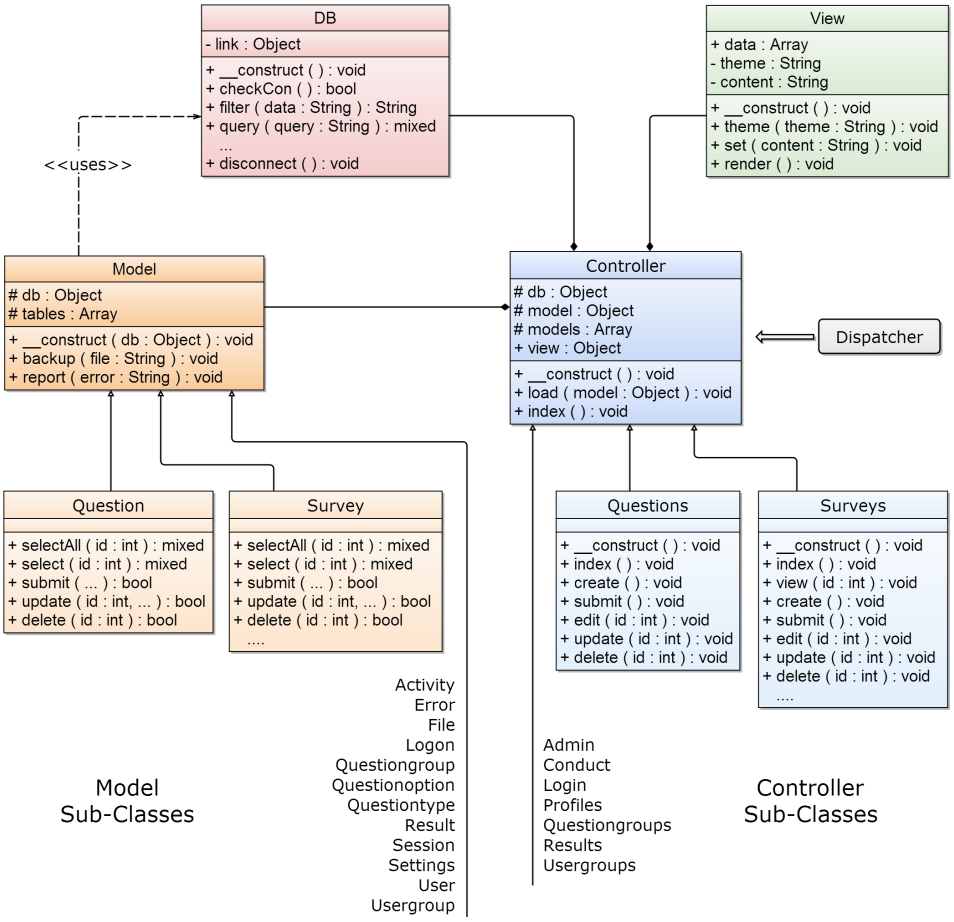


Figure 5.10: Class diagram of super-classes and some example sub-classes.

```
4.  set view->theme() from settings variable
5.  initialise main/default Model class
    5.1  pass DB object (class) to model
6.  if an error has occurred or maintenance mode is on
    6.1  run view->render()
    6.2  exit
```

All model classes are loaded using the controller's *load()* function, which not only initiates the new model while passing the DB (database) object as an argument, but also stores the resulting class object in the controller's global attribute *models* for further reference.

Once the constructors have executed, the dispatcher calls a specified operation (function) in the controller, depending on what action is requested by the client. The controllers contain more or less the same structure of functions, namely *index()*, *view()*, *create()*, *submit()*, *edit()*, *update()* and *delete()*. The *index()* function is somewhat special in that it is called whenever no action-parameter were given by the requester. Thus it varies what content is displayed when this occurs, if any. Otherwise, the aforementioned functions commonly do what their title suggests. For example, the *Surveys*-controller's *view()* function would display the data of a specified survey, while the *create()* function presents a creation form for a new survey. Once this function has successfully executed, the controllers job is essentially done and control falls back to the dispatcher.

Models

The model classes contain the logic to manipulate all persistent information, both stored in the database and in the application's code itself. Similarly to the controllers, there is one model super-class and multiple subclasses which handle data manipulation of specific parts of the information at hand. However, unlike controllers, one or more models can be loaded simultaneously, providing functions for manipulating several different groups of information. The models are loaded on an "as needed" basis, thereby not wasting any unnecessary server resources. Moreover, the super-class consists of a function for running a full database backup, a method for storing errors to the database, and an array containing the database table names used by all underlying sub-classes. Many of the model sub-classes also contain a more or less standard structure of functions. These include *selectAll()*, *select()*, *submit()*, *update()* and *delete()*, and commonly reflects the database actions SELECT, INSERT INTO, UPDATE and DELETE, respectively. While performing actions on the database using the operations of the DB class, they also include error-reporting in situations where an error would occur during the database queries.

Although the DB class is technically a stand-alone class, it typically falls under the model category as only the models interact with it. Moreover, the database class manages data manipulation and persistence, which inherently belongs to the model component of the MVC architecture. Additionally, it is worth mentioning that the DB class has been completely rewritten during the development process, from the old *mysql* PHP-extension interface to the new and improved *mysqli* interface. This is an update recommended by the PHP Group, for newer applications using recent versions of both PHP and MySQL [20]. Mysqli provides an efficient object-oriented interface, which to a greater extent corresponds to our application's programming paradigm.

Views

The purpose of the View is to be responsible for constructing the application's graphical interface before it is sent to the requester and displayed in a web-browser. This operation primarily involves building the HTML structure of the interface, including the relevant content to be displayed, as well as adding the appropriate CSS and JS code to the mix. To accomplish this, we have a single view class which controls a large set of *HTML-templates*. These templates consist mostly of HTML code, but are also interlaced with embedded PHP code. This code handles mostly output of information processed by the controller, but can also include some simple operations that lessen the need for repetition of HTML code.

The view class itself presents three variables of importance; a *theme* string representing which theme, or interface, to use, a *content* string specifying what content template to use, and a *data* array that stores information computed by the controller for use in the templates. The theme and content variables are set by the controller during its execution time, but nothing else happens in the view before its main function *render()* is called, which happens to be the last step in the execution of the application. The *render()* function carries out the following procedure, presented in pseudocode:

```
1.  if maintenance mode is activated
    1.1  output the maintenance template
    1.2  exit
2.  if a fatal error has occurred
    2.1  output the fatal error template
    2.2  exit
3.  if an error has occurred
    3.1  set content to corresponding error template
4.  if the content template file does not exist
    4.1  set content to "not found" template
5.  extract data from data array for further use
6.  output header template
7.  output specified content template
8.  output footer template
```

5.6.4 Experimentation

The implementation of the experimentation back-end and interface is quite extensive, but nonetheless fairly straightforward and transparent. The single controller class *Conduct* handles the logic for conducting all experiments, using four functions that each manage a specific part of the experimentation run-through; *start()*, *survey()*, *next()* and *finish()*.

Initialisation

The *start()* function is dispatched when a participant first initiates an undertaking of a specific experiment. This operation ensures for instance loading the necessary experimentation data, including all questions, into the session as explained in Section 5.5.5. If applicable, the order of questions is also randomised, both within groups and overall. Several pre-checks are initially performed as well, for example assuring that the participant has not taken part in this survey before, and checking that the survey is active and initiated within the set time frame. Furthermore, session variables used to keep track of experimentation progression is initialised, before issuing the view to display the experimentation start-up interface.

Looping

The two main operations of the experimentation are the *survey()* and *next()* functions. These form a particular loop throughout the conducting, which is broken once the participant has answered the last question of the experiment. Along with a few other minor tasks, *survey()* initiates a timing variable, storing the current time, before issuing the view to present an interface displaying the current question and answer options. Once the participant has given an answer and thereafter pressed the next button, the *next()* function is dispatched. The time taken to answer is firstly calculated, before validating the answer given. Furthermore, the answer value and timing are stored in the session, followed by an incrementation of the current question counter. Finally, a call to the *survey()* function is executed, hence creating the aforementioned loop.

Completion

Sufficed to say, the *finish()* function handles the last part of the experiment. This includes the task of storing all the given answers and corresponding timings in to the database, as well as registering the user as having completed the specified experiment. Lastly, the experimentation data temporarily stored in the session are cleared, making room for any other experiment to be started in the future. finishing

5.6.5 Multimedia Content

In Section 4.3 we explained briefly the difficulties in implementing presentation of multimedia content for different browsers. Currently, there are no single way of fully overcoming this issue. However, we have implemented the multimedia handling in such a way as to fully support at least some formats for each media type, as displayed in Table 5.3. With exception to H.264 video which is not supported in Opera, all formats are currently supported in HTML5 by all major browsers. To make sure that Opera users, as well as participants with older browser versions without HTML5 support, can view multimedia content, we have implemented a simple fall-back method. If the content cannot be natively displayed by the browser, it will fall back to the Adobe Flash Player. Although Flash usage is on the decrease, it is still in use by a vast majority of users [49], especially those with outdated browser versions. This use of Flash fall-back to ensure multimedia playback is a common practice today, due to the difficulties in cross-browser compatibility.

The other formats mentioned in Section 4.3 can, however, also be used in experiments run through MAT, although these should probably be targeted towards users with specific browsers. Moreover, the unsupported formats can often be played in browsers which does not support them, by the use of other third party plug-ins. Unfortunately, most people do not have these plug-ins installed and are often disinclined to do so. Support for more formats will most likely become better in the future as more browsers

Video	H.264/AVC (.mp4, .m4v, .mov)
Audio	AAC (.mp4, .m4a)
	MP3 (.mp3)
Image	PNG (.png)
	JPEG (.jpg, .jpeg)

Table 5.3: Multimedia formats with cross-browser support in MAT.

incorporate the different multimedia formats. For instance, Firefox recently added support for the H.264 format, after several years of reluctance [43].

5.6.6 Security & Validation

The log-in feature is the main source of security from unwanted intruders. Without a valid user account, and the corresponding identification name and password, unauthorised users should not be able to access the application and thus not be able to do any harm. The log-in process, as explained in Section 5.5.2, provides the security by checking that the data entered matches what is stored in the database. For increased security, a minimum password length of eight characters is enforced. Moreover, the passwords stored in the database are encrypted to prevent any attacker that has gained access to the database in some way, to see and use said passwords. The encryption method used in our application is the simple, but sufficiently effective MD5 cryptographic hash function with an additional iteration, as shown below.

```
$password = md5(md5($password));
```

Although the log-in feature may be seen as a good measure for security, it cannot prevent attackers from entering if they have gotten hold of the identification and password of users elsewhere. Consequently, as with all applications with log-in details, the users have a responsibility for keeping their password safe and secret.

A primary security issue worth mentioning, regarding most web applications using databases for persistence, is SQL-injection. This is a technique of code injection where an attacker may insert malicious SQL statements into an entry field for execution. However, SQL injection must exploit a vulnerability in the application code in order to succeed, and thus if the software is properly developed it should not be possible for the attacker to do so. MAT includes a large amount of user input fields, not only for administration features where the user is somewhat trusted, but also for example in the conducting of experiments. In either case, all input that comes from any user is run through a filtration function. This procedure utilises the *mysqli*-function *real_escape_string()*, which prepends backslashes to characters of importance in the MySQL query system, and consequently makes the data safe before sending it to the DBMS.

On top of the SQL injection prevention, we use a fair bit of input field *validation*. By validating user input we not only increase security, but also reduce errors caused by incorrect or insufficient input, hereby increasing interface usability. The validation has been implemented in three layers of our application:

- **HTML** The use of specific input field types and attributes provide some easy and basic validation. For example, if input is required, a *required* attribute would be added to the input field code, which hence notifies the user if the field is empty on submission. The below example is used in MAT to collect an answer in text format from a participant.

  ```
  <input name="answer" type="text" required="required" .../>
  ```

  This form of validation is however only supported in HTML5, and which input types and attributes are supported by different browsers vary greatly. Moreover, as HTML5 itself is not supported by older browsers, we need additional validation as well.

- **JS** With the use of Javascript, or more precisely jQuery, we can provide a much more thorough validation on any field of our choice. Upon form submission, a script of this type is run, validating all input fields that are marked for validation. This validation will for example check whether an appropriate amount of characters have been entered, or even check if an entered e-mail address is of valid format. However, Javascript can be deactivated in a client's web-browser, making it necessary for yet another layer of validation.

- **PHP** The two aforementioned validation techniques both occur on the client-side, with no interaction from the server at all. The final layer however, occurs on the server after input has been submitted and sent from the client. This is commonly an inconvenience for user, as the same page must be reloaded and the same input has to be entered once more upon validation failures. Nonetheless, it is necessary to ensure that the received data is valid before further processing. This validation works more or less in the same way as the Javascript method, although it is written in PHP along with the rest of the server.

5.7 Summary

In this chapter, we have presented the design and implementation aspects of the Multimedia Assessment Tool. Firstly, we described the distributed application structure known as the server-client model, such as HTTP, that our system's communication utilises. Furthermore, MAT has been developed based on the Model-View-Controller (MVC) architecture pattern that separates business logic and presentation, which we have shown can be highly beneficial. This architecture consequently dictated an organised and object-oriented file structure of classes and other system files. The database structure for our application has similarly been designed using well documented techniques for organisation of content in transactional databases. Data normalisation ensures that the data is stored in logical groupings, minimising duplicate data, and ascertaining consistency and integrity of the data.

Furthermore, a thorough explanation of the client composition and design has been provided. We have shown how the application handles various features, including everything from logging in, to the extensive administration section. A more technical description of the implementation of these features are also included, all the way down to MVC architectural pattern. Starting off with the local Apache configuration, we have conveyed how the dispatcher initialises specific controllers and models for each request by a client, resulting in the creation of the appropriate user interface. More significantly, we have described how the quality assessment experiments are run and handled by the application. Finally, an overview of the implementation of multimedia content and security measures have been presented.

Chapter 6

Discussion

In this chapter, the current status and potential problem areas in QoE research are discussed in the context of the Multimedia Assessment Tool.

6.1 Motivation

With the implementation of an online assessment tool for running user studies on multimedia quality, the presented work aimed to ease the experimental task of running QoE studies. As described, MAT was developed with a solid theoretical background and it takes into account a number of important topics, such as QoE assessment, methodology, and most significantly, multimedia. Considering how multimedia has infused so many aspects of everyday life, with its influence still increasing, new ways to establish what constitutes high quality multimedia are also in increasing demand. With the continuous technological advances, users have come to expect nothing less than high quality content, and they expect it instantly and on the go. As described in Section 2.2.1, compression is a key factor for supplying users with media at high speed, while maintaining acceptable quality levels. In fact, without compression, multimedia would not enjoy the presence it has in today's society. However, since compression, along with transmission and decoding, may cause both auditory and visual artifacts, quality can be compromised. As a result of the evolving technology and user expectations, we have observed a need for an adaptive assessment method that can be adjusted according to the content, research question, or quality manipulation of interest. This way, users own experiences of what is deemed as acceptable quality can be assessed across a range of scenarios.

As we have brought to attention, quality assessment can be performed by using either objective or subjective measures. Objective methods are typically more convenient to use; however, no matter how advanced these may be, they cannot tap into every factor that may affect the experiences of users. For example, an objective method may easily detect unnatural noise in a multimedia sequence, but it cannot know how this noise might affect a person. The noise could be imperceptible to humans, and if perceptible, it might not be the least distracting or annoying.

This further emphasises the need for solid subjective assessment methods which provide researchers with large sets of data to establish consistent measures of QoE. While traditional methods for this kind of assessment, run in controlled environments and under supervision of researchers, have existed for a long period of time, we set out to make use of the advancing technology to improve its own assessment. By expanding the established methodology to an online platform, we aimed to make the same assessment methods more accessible and convenient for both researchers and participants. Furthermore, with the newly emerging trend of crowdsourcing, we have introduced another solution to make user studies more simple and effective when collecting data from a large pool of participants. However, along with these advantages, the online platform brings its own set of problems and risks, as occasionally mentioned in the thesis, which we will discuss further in the following sections.

6.2 Considerations

Although running assessment studies over the Internet is both accessible and convenient, other benefits and issues have also been taken into consideration. Most notable is the lower cost compared to traditional lab-experiments. For example, in [7, 56] we see four different QoE experiments, all executed using the laboratory strategy, crowdsourced via Amazon's Mechanical Turk, and crowdsourced to an Internet community. These papers show that, of the US$191.8 spent on running these particular experiments, 89% of the total monetary cost went into the laboratory strategy. On average, the cost for each round of laboratory experiments was 4.60 cents, while the crowdsourced Mechanical Turk and community experiments were 1.00 and 0.07 cents, respectively. Although the cost of conducting an experiment depends on several factors, such as expenses associated to equipment and facilities, this set of work illustrates the cost-saving potential of the crowdsourcing and online platforms.

Online assessment studies also demand less effort and are less time-consuming, both for the experimenters and participants. Experiments are commonly more easily and quickly created, and experimenters need not facilitate nor supervise the experiments. However, this presents a problem in itself, in cases where environmental control is of importance. As we explained in Chapter 2, many parameters of the viewing conditions can influence the results, such as room illumination, display type, brightness, contrast, resolution, viewing distance, and the age and educational level of participants. This can not be controlled or monitored during an online experiment, like it would be in a laboratory experiment. Nevertheless, in other contexts this may be considered an advantage, since it allows the user's perceptions to be assessed in real-life scenarios. The online platform also eliminates the need for manual data entry. As responses are automatically stored in a database, with the ability to download the results whenever needed, the experimenter need not spend time on tedious tasks. Furthermore, participants are free to complete the experiment when and where they wish, and they do not have to spend time and money travelling to and from the experimental facility. In turn, this may facilitate better response rates, both with respect to time and quantity. A larger pool of participants should also reflect greater background diversity, and it should certainly contribute to greater statistical power in the data analysis. However, the demography of the participants can only be asked, not verified, meaning that sampling would normally be out of the question. If a situation calls for certain participant characteristics, an experiment can still be designed to target a known group of people with specific demographic attributes.

6.3 Data Validity

When performing user studies, methodological controls are enforced to ensure consistent data, as described in Chapter 3. To maintain the same consistency in data collected online from studies run on MAT, our application framework incorporates some of the most essential test methods. Through the implementation of existing and well-tested methods, assessments run on MAT will follow standardised guidelines, thereby increasing the generalisability of results. Consequently the outcomes of these studies can be interpreted in light of earlier studies, and they form the basis for future comparisons with similar experiments. The ITU recommendations for subjective quality assessments [27, 31, 30] are good examples of such standardised methods because they offer detailed instructions for designing and running experiments and they are used extensively in quality studies. Additionally, the outlined procedures are easily incorporated into our online assessment tool. Furthermore, Chen et al. [7, 56] have shown that pair comparison, one of ITU's recommended assessment methods, are particularly suitable for online and crowdsourced studies. Their work highlights PC's implementation of simple comparative judgement tasks and the ease of applying consistency checks. Similarly, ACR and DCR are two straightforward methods that may provide the same usability in online assessment studies as they do in traditional laboratory experiments. As such, these three methods are well incorporated in our assessment framework.

On the topic of data validity follows the issue of most concern with regard to the online assessment platform, namely the quality of results. In studies that rely on crowdsourcing, response data must be considered with great care, as explained in Section 2.3.3. With the anonymity that the Internet offers, online users can arguably be labelled as less trustworthy than participants that must be physically present in a facility. Results collected through a crowdsourcing scheme could reflect random response patterns or even deliberately incorrect responses, thus their accuracies are not guaranteed. Fortunately, there are two ways of countering this concern, both of which may be applied to provide a more reliable set of results. First, consistency checking works to detect and counteract false results. By including more than one repetition of a test item, the consistency in a participant's responses can be evaluated through the calculation of the inter-item correlation. Low correlations would imply that a participant is not giving the same responses across similar conditions, giving the experimenter the option to exclude the participant from further analyses. The second step also involves statistical data analysis, with measures that identify and exclude outlying scores. Furthermore, these statistics express the confidence of results, as briefly mentioned in Section 3.3. Subsequent statistical analysis is a standard research practice that can be applied using MAT; however, consistency checking has not yet been fully implemented, and is a feature of great importance for future work.

6.4 The Multimedia Assessment Tool

Chapter 5 presents the developed framework, the Multimedia Assessment Tool, and shows how it complies to the requirements put forward in the introduction chapter. For instance, we show how MAT delivers simple management of participants, invitations, multimedia content, and most significantly, experiment design and management. Moreover, experiment design and question items can be customised according to the specific requirements of a study, such as instructions, question wording, rating scales, and randomisation. An important goal in developing the MAT framework has been to provide as much flexibility as possible for experimenters. To accomplish this goal, the application has been built to support a large amount of customisability in the creation of experiments and questions. This includes multiple test methods and rating scales that can be modified. Several multimedia formats are supported, possibly with more to come depending on the future development of web-browsers. Crowdsourcing may be used to gather large, diverse groups of participants, or experiments can be issued to predetermined individuals. The benefits of being flexible will reflect highly in the outcome of the studies. While the main responsibility of designing a thorough study lies with the experimenter, MAT offers a solid platform that can alleviate the practicalities of the undertaking. As mentioned, the concerns related to the online assessment will need to be taken into consideration by the experimenter for every planned study.

Considering that the presented framework was developed within constraints related to both time and resources, full-scale testing of the application was unfortunately not feasible. However, local testing has been performed on the features of the application themselves, ensuring that every feature works as intended. From theory and observation, the framework appears ready for use in multimedia quality assessment. Preliminary testing shows that the application has an average execution time of less than 0.1 seconds. Cross-browser compatibility has been tested by accessing our application using the different major web-browsers. User-scalability has also been tested, using the HTTP server benchmarking tool ApacheBench [17]. As seen in Table 6.1, running a medium to high load of 100 requests, with 10 requests being sent concurrently, took a total of 0.411 seconds, resulting in an average request time of 40 milliseconds. Running

100 concurrent requests also yielded acceptable results, while the extreme amount of 1000 simultaneous requests showed tendencies to seriously slow down the server. A *request* includes the client connecting to the server, sending a request, waiting for the server (and our application) to execute, and receiving data in return. The results shown below may of course vary depending on where this test is executed from and what hardware and bandwidth the server has at its disposal, but it gives us an indication of the user-scalability of the application nonetheless.

Requests (#)		Time (ms)			
Total	Concurrent	Min	Avg	Max	Total
100	10	19	40	114	411
1000	100	34	341	460	3561
1000	1000	172	2860	9642	9703

Table 6.1: Scalability-/stress-test of server and application software.

Although we have described the possibilities that MAT presents, it is also meaningful to discuss the current limitations of our framework. For instance, multimedia has commonly been described in this thesis as audio (sound and music) or visual (image, video and graphics), or a combination of these. However, today's definition of multimedia has expanded to include an even larger variety of data types [5]. Therefore, it is necessary to specify what type of multimedia that can and cannot be evaluated in the present state of our assessment tool. Currently, MAT is only aimed at the evaluation of *recorded* audiovisual multimedia presentations. Multimedia presentations may also be live, and thus cannot be evaluated in MAT unless the live stream is recorded and thereafter presented for assessment. Interactive, non-linear multimedia is also a major category of multimedia that unfortunately cannot be assessed using our tool. Within the recorded audiovisual category of multimedia content, there are also limitations to which file formats, or encoding, that can be used in MAT. As explained in Section 4.3, this is due to current restrictions in the implementation of the major web-browsers. However, the most common audio, video and image formats are supported, and enabled to be presented in the various versions and types of web-browsers. Following the recent trend of updates in web-browser technology, this may change in the near future and allow presentation of a more varied set of formats.

Moreover, MAT has a certain limitation in the amount of assessment methodologies implemented. It currently supports the ACR, DCR and PC test methods, but we have put great emphasis on easy implementation of future options. Additionally, questions and response options are highly customisable, so variations can be designed by experimenters themselves. Due to the way multimedia is presented in HTML, the methodologies currently in place also differ slightly from the ITU recommendations. Multimedia playback can be controlled by the participants themselves, and there are no constraints on response time, as specified by the ITU [30]. Instead, the time taken to make an assessment is recorded, providing average response times for every question. Furthermore, as mentioned in the previous section, MAT lacks consistency checking of participant responses, which may reduce the confidence of results, especially in crowdsourced assessment studies. Consistency checking, and other future possibilities for MAT, are discussed further in Section 7.2.

Chapter 7

Conclusion

To conclude the work that has gone into developing the MAT framework, this chapter provides a brief outline of the steps we have followed. Considering past work on related topics, along with our own experiences from this process, we also present suggestions for future improvements and new directions in this line of research.

7.1 Summary

In this thesis, we have addressed the cumbersome methodology of running traditional laboratory studies to assess multimedia quality, and we have investigated and addressed the promising advantages of an online approach. Furthermore, in an era where multimedia is present nearly everywhere, the need for evaluating its subjective quality is in no way diminishing. With the new ease of social interactions across the rapidly growing Internet, an opportunity has presented itself. The same technology that is constantly in demand can also be used to assess itself. We have presented such a solution, by introducing an online platform for running quality evaluations. Moreover, we have proposed crowdsourcing as means to easily and efficiently gather a diverse population of participants for online quality assessment studies. The thesis considers established methodologies from this area of research and, with the presented framework, aims to offer a simple and cost-efficient solution for running online user studies.

The application presented in this thesis is designed to be easy to use and to provide experimenters with the flexibility they need to design their own studies. We have assessed the requirements for an online tool to facilitate many types of multimedia quality experiments. Based on these needs, we have proposed a design that fulfils the requirements and that allows researchers to run assessment studies according to their particular needs or preferences. Finally, we have been through an extensive development process, which resulted in a solid solution to the problem we faced.

In our discussion, we have shown that our assessment tool can provide a simple and flexible solution to researchers in the field of QoE. With its potential to reduce costs and increase time efficiency, it is a valid option for running user studies. With more targeted testing, we anticipate that our application can provide reliable results across several scenarios. Moreover, with the ethical and certain methodological considerations delegated to the experimenters, the discussed concerns become less relevant. In future work, we expect to negate any potential obstacles further and provide researchers with an increased amount of flexibility and improved features.

Overall, MAT offers multimedia researchers an online tool with easy-to-use and well-established methodologies. In turn, we aim for the online community to contribute with their opinions on what they expect from the multimedia they consume every day.

7.2 Future Work

For future work, we suggest running a methodological, full-scale test of the Multimedia Assessment Tool in its current state. While the implemented features of the application work according to their purposes, and should by all means operate as designed, we still wish to evaluate the tool that will be further used for evaluations. Through this testing with an appropriate demographic group, we can determine if our expectations are confirmed.

Although our assessment tool is operational and could be utilised in research studies as it is, there is always room for improvement, particularly with respect to the overall usability of the application. Certain minor features can be added to MAT to further improve its ease of use. For example, implementing an installation script that automates the setup and configuration of the software itself could prove beneficial to the initial installation. The MVC architecture facilitates good support for multiple concurrent interfaces, so additional language support or alternative graphical designs can also be implemented. Another thing we aim for, as mentioned in Section 3.1.6, is to implement the long-duration quality assessment methods QELDAC and SSCQE. Additionally, it would be interesting to explore which other methodologies could be introduced in the application, to provide even more flexibility for experimenters using MAT.

The most significant feature we propose for future work is consistency checking. We touched upon this topic in Section 2.3.3 and in the discussion, explaining that undefined crowds on the Internet can be deemed untrustworthy. Participants taking part in experiments for payment may feel an incentive to complete tasks quickly rather than well, which can lead to unreliable results. To counteract this concern, we propose to integrate consistency checking. This involves repetition of test items, where the measure of correlation between ratings for the repeated item provides an indication of consistency in participants' responses. This consistency check, along with other validity measures, may then be used to determine the reliability of the data. However, this is a fairly time-consuming feature to implement, especially when having to measure consistency between the different methodologies available to the experimenter. Meanwhile, overall reliability may be explored to an extent by applying specific statistical analyses to the results of an experiment.

It is of course always possible to make improvements and additions to any application. In its current state however, MAT should provide great assistance to the research community, and with the aforementioned additions and more, it may become an even greater resource.

Appendix A

Terms and Acronyms

MAT - The Multimedia Assessment Tool we are developing in this thesis.

QoS - Quality of Service. Objective system performance metric for assessing the quality of systems.

QoE - Quality of Experience. Quality of a service based on the opinion of individuals.

MOS - Mean Opinion Score. Well-known framework for assessing the QoE of multimedia content. Also a mean score of a set of ratings from an experiment.

ACR - Absolute Category Rating. Subjective assessment method based on rating the quality of a presented sequence on a category scale.

DCR - Degradation Category Rating. Subjective assessment method based on rating the impairment of a stimulus in relation to a reference.

PC - Pair Comparison. Subjective assessment method based on comparing the quality of two items.

R/PC - Randomised Pair Comparison. Variance of the PC method focused on reducing the amount of pairs in an assessment study by only showing a randomised subset of pairs.

SSCQE - Single Stimulus Continuous Quality Evaluation. Subjective assessment method for long duration multimedia sequences

QELDAC - Quality Evaluation of Long Duration Audiovisual Content. Similar to SSCQE, though based on adjusting the quality of the sequence during playback.

Server - Location with a defined internet address which stores files for client consumption.

Client - A software application that requests and displays content requested from a server. The most common form of client is a web browser.

LAMP - Linux, Apache, MySQL, PHP. A popular software stack for hosting and publishing interactive web-content.

PHP - PHP: Hypertext Preprocessor. Software and programming language that enables the construction of dynamic web applications.

SQL - Structured Query Language. Standardised special-purpose programming language used to access databases.

DBMS - Database Management System. A system to access and manipulate data stored in a database. E.g. MySQL.

HTTP - Hyper Text Transport Protocol. The main transport method for HTML content from server to client and visa versa.

HTML - HyperText Markup Language. The structure and content of a web site or application.

CSS - Cascading Style Sheets. Allows change to the design and "look and feel" for a site.

JS - JavaScript. Language for scripting high level interactivity on the client-side.

MVC - Model-View-Controller. Software architecture pattern used in MAT. Separates business logic from presentation.

Appendix B

Source Code

The Multimedia Assessment Tool is currently hosted at http://mat.ndlab.net/, and the source code can be accessed at http://mat.ndlab.net/release/.

Bibliography

[1] Robin Berjon, Travis Leithead, Silvia Pfeiffer, Erika Doyle Navara, Edward O'Connor, and Ian Hickson. HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C Candidate Recommendation, W3C, December 2012. http://www.w3.org/TR/2012/CR-html5-20121217/.

[2] Hazel Biggs and Chrstine Milligan. Anonymity and confidentiality. http://www.lancaster.ac.uk/researchethics/1-2-anonconf.html. [Online; accessed 23.10-2013].

[3] Adam Borowiak, Ulrich Reiter, and U. Peter Svensson. Quality evaluation of long duration audiovisual content. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 337–341. IEEE, 2012.

[4] Daren C Brabham. *Crowdsourcing*. MIT Press, 2013.

[5] Shih-Fu Chang. Chair's message | ACM SIGMM - the Special Interest Group on Multimedia. http://www.sigmm.org/news/chairs_msg_2013. [Online; accessed 04.11-2013].

[6] Kuan-Ta Chen, Chi-Jui Chang, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. Quadrant of euphoria: a crowdsourcing platform for qoe assessment. *IEEE Network*, 24(2):28–35, 2010.

[7] Kuan-Ta Chen, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. A crowdsourceable QoE evaluation framework for multimedia content. In *Proceedings of ACM Multimedia 2009*, 2009.

[8] Microsoft Corporation. ASP.NET Overview. http://msdn.microsoft.com/library/4w3ex9c2.aspx. [Online; accessed 13.08-2013].

[9] Oracle Corporation. JavaServer Pages Technology. http://www.oracle.com/technetwork/java/javaee/jsp/index.html. [Online; accessed 13.08-2013].

[10] Oracle Corporation. MySQL. http://www.mysql.com/. [Online; accessed 24.05-2013].

[11] Oracle Corporation. Mysql Reference Manual: What is MySQL? http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html/. [Online; accessed 24.05-2013].

[12] Phuc-Tue Le Dinh and Jacques Patry. How-to: Video compression artifacts and MPEG noise reduction. *Video Imaging DesignLine*, February 2006.

[13] Alexander Eichhorn, Pengpeng Ni, and Ragnhild Eg. Randomised Pair Comparison - An economic and robust method for audiovisual quality assessment. In Dick C. A. Bulterman, editor, *The 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2010)*, pages 63–68. ACM, 2010.

[14] Elika J. Etemad. Cascading Style Sheets (CSS) Snapshot 2010. W3C Note, W3C, May 2011. http://www.w3.org/TR/2011/NOTE-css-2010-20110512/.

[15] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1, 1999.

[16] David Flanagan. *JavaScript: The Definitive Guide*. O'Reilly Media, Inc., 2006.

[17] The Apache Software Foundation. ab - Apache HTTP server benchmarking tool. http://httpd.apache.org/docs/2.4/programs/ab.html. [Online; accessed 28.10-2013].

[18] The Apache Software Foundation. The Apache HTTP Server Project. http://httpd.apache.org/. [Online; accessed 22.05-2013].

[19] Alvin I Goldman. Ethics and cognitive science. *Ethics*, 103(2):337–360, 1993.

[20] The PHP Group. PHP: Hypertext Preprocessor. http://php.net/. [Online; accessed 28.05-2013].

[21] Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, 2008.

[22] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[23] Amazon.com Inc. Amazon Mechanical Turk. https://www.mturk.com. [Online; accessed 05.09-2013].

[24] Cisco Systems Inc. Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html. [Online; accessed 17.09-2013].

[25] Google Inc. YouTube. http://www.youtube.com. [Online; accessed 18.07-2013].

[26] Adobe Systems Incorporated. Adobe - Flash Player. http://www.adobe.com/software/flash/about/. [Online; accessed 14.10-2013].

[27] ITU-R. Methodology for the subjective assessment of the quality of television pictures. Recommendation BT.500-13, International Telecommunication Union, January 2012.

[28] ITU-T. Methods for subjective determination of transmission quality. Recommendation P.800, International Telecommunication Union, August 1996.

[29] ITU-T. Principles of a reference impairment system for video. Recommendation P.930, International Telecommunication Union, August 1996.

[30] ITU-T. Subjective audiovisual quality assessment methods for multimedia applications. Recommendation P.911, International Telecommunication Union, December 1998.

[31] ITU-T. Subjective video quality assessment methods for multimedia applications. Recommendation P.910, International Telecommunication Union, September 1999.

[32] ITU-T. Definitions of terms related to quality of service. Recommendation E.800, International Telecommunication Union, September 2008.

[33] Ramesh Jain. Quality of experience. *IEEE MultiMedia*, 11(1):96–95, January 2004.

[34] The jQuery Foundation. jQuery. http://jquery.com/. [Online; accessed 27.05-2013].

[35] Madhuri Khambete and Madhuri Joshi. Blur and ringing artifact measurement in image compression using wavelet transform. In *Proceedings of World Academy of Science, Engineering and Technology*, 2007.

[36] James B. Lee and Brent Ware. *Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[37] Avraham Leff and James T Rayfield. Web-application development using the model/view/controller design pattern. In *Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International*, pages 118–127. IEEE, 2001.

[38] Daniel J Levitin, Karon MacLean, Max Mathews, Lonny Chu, and Eric Jensen. The perception of cross-modal simultaneity. In *AIP Conference Proceedings*, volume 517, page 323, 2000.

[39] Chi-Min Liu, Han-Wen Hsu, and Wen-Chieh Lee. Compression artifacts in perceptual audio coding. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(4):681–695, 2008.

[40] Tao Liu, Yao Wang, Jill M Boyce, Zhenyu Wu, and Hua Yang. Subjective quality evaluation of decoded video in the presence of packet losses. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–1125. IEEE, 2007.

[41] Netcraft LTD. Netcraft Web Server Survey. http://news.netcraft.com/archives/category/web-server-survey/. [Online; accessed 27.05-2013].

[42] Rose M Marra and Barbara Bogue. A critical assessment of online survey tools. *Women in Engineering ProActive Network*, 2006.

[43] Mozilla Developer Network. Media formats supported by the HTML audio and video elements. https://developer.mozilla.org/en-US/docs/HTML/Supported_media_formats. [Online; accessed 18.07-2013].

[44] Greg Norcie. Ethical and practical considerations for compensation of crowdsourced research participants. *CHI WS on Ethics Logs and VideoTape: Ethics in Large Scale Trials & User Generated Content*, 2011.

[45] Office of Planning and Institutional Assessment. Using surveys for data collection in continuous improvement. *Innovation Insights*, 14, 2006.

[46] Khalid Sayood. *Introduction to Data Compression*. Morgan Kaufmann series in multimedia information and systems. Elsevier, 2006.

[47] StatCounter. Top 12 Browser Versions on July 2013. http://gs.statcounter.com/#browser_version-ww-monthly-201307-201307-bar. [Online; accessed 05.08-2013].

[48] World Wide Web Technology Surveys. Usage statistics and market share of server-side programming languages for websites, May 2013. http://w3techs.com/technologies/overview/programming_language/all. [Online; accessed 28.05-2013].

[49] World Wide Web Technology Surveys. Usage Statistics of Flash for Websites, October 2013. http://w3techs.com/technologies/details/cp-flash/all/all. [Online; accessed 15.10-2013].

[50] The World Wide Web Consortium (W3C). Web design and applications. http://www.w3.org/standards/webdesign/. [Online; accessed 12.02-2013].

[51] Wikipedia. Comparison of web browser engines. http://en.wikipedia.org/wiki/Comparison_of_web_browser_engines. [Online; accessed 05.06-2013].

[52] Wikipedia. LAMP (software bundle). http://en.wikipedia.org/wiki/LAMP_(software_bundle). [Online; accessed 15.05-2013].

[53] Wikipedia. Multimedia. http://en.wikipedia.org/wiki/Multimedia. [Online; accessed 22.02-2013].

[54] Wikipedia. Ringing artifacts. http://en.wikipedia.org/wiki/Ringing_artifacts. [Online; accessed 17.09-2013].

[55] Kevin B Wright. Researching internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services. *Journal of Computer-Mediated Communication*, 10(3), 2005.

[56] Chen-Chi Wu, Kuan-Ta Chen, Yu-Chun Chang, and Chin-Laung Lei. Crowdsourcing multimedia QoE evaluation: A trusted framework. *IEEE Transactions on Multimedia*, 2013.