

UNIVERSITY OF BERGEN

MASTER THESIS

---

**Expressive Creative Overview  
System (ECHO):  
a Creativity Support Tool**

---

*Author:*

Daniel Taranger

*Supervisor:*

Ankica Babic



*A thesis submitted in fulfillment of the requirements  
for the degree of Master of Information Science*

*in the*

Department of Information Science and Media Studies

December 1, 2017



University of Bergen

# *Abstract*

Faculty of Social Sciences

Department of Information Science and Media Studies

Master of Information Science

**Expressive Creative Overview System (ECHO):  
a Creativity Support Tool**

by Daniel Taranger

A creativity support tool called the Expressive Creative Overview (ECHO) system has been developed in this thesis. It has two main functionalities; one which supports artists to organize their creative processes and the other provides feedback from their fans. While it is not aimed at improving creativity, the system's goal is to support the creative process. In technical terms, this happens by creating musical version and enabling artists to get reviews on musical versions they post.

A focus group, in addition to the artists input, assisted in defining user requirements. Design Science and Shneiderman's principles for creativity support tools provided valuable frameworks for research into the topics of creative supports tools and implementing a web-based system as an artifact.

Final evaluation of the ECHO system showed that both usability and learnability reached satisfactory user scores. Comments from experts showed an interest for such creative tools that could be fine-tuned according to the artists needs.





## *Acknowledgements*

I would like to thank my supervisor Ankica Babic, for her guidance and motivation throughout the process. Secondly, fellow artist and friend Casey Keyworth for the early discussions about the idea of the system. Students of the University of Bergen for their engagement through interviews and evaluations. My roommates and friends at room 634 for the laughs and motivation. Last but not least, my family for inspiration and support.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Background . . . . .	1
1.1.1 The One Man Band . . . . .	2
1.1.2 Why add an extra step? . . . . .	3
1.2 Research Questions . . . . .	4
1.2.1 Research Question 1 . . . . .	4
1.2.2 Research Question 2 . . . . .	4
1.3 Chapters Overview . . . . .	5
<b>2 Literature Review</b>	<b>7</b>
2.1 Creativity . . . . .	7
2.1.1 Romantic and Non-Romantic . . . . .	8
2.1.2 Idea Finding . . . . .	8
Combination . . . . .	9
Exploration . . . . .	9
Transformation . . . . .	9
P and H-creativity . . . . .	10
2.1.3 The Creative Work Process . . . . .	11
Structuralists . . . . .	11
Inspirationalists . . . . .	11
Situationalists . . . . .	12
2.1.4 The Creative Product . . . . .	12
Originality . . . . .	13
Appropriateness . . . . .	13
Obviousness . . . . .	13
2.2 Creativity Support Tools . . . . .	14
2.3 Schneiderman's Principles . . . . .	14
2.3.1 Support Exploration . . . . .	15

2.3.2	Low threshold, high ceiling, and wide walls . . . . .	16
2.3.3	Support many paths and many styles . . . . .	16
2.3.4	Support Collaboration . . . . .	16
2.3.5	Support Open Interchange . . . . .	17
2.3.6	Make it as simple as possible and maybe even simpler . . . . .	17
2.3.7	Choose black boxes carefully . . . . .	17
2.3.8	Invent things that you would want to use yourself . . . . .	17
2.3.9	Balance user suggestions, with observation and participatory processes . . . . .	18
2.3.10	Iterate, iterate then iterate again . . . . .	18
2.3.11	Design for designers . . . . .	19
2.3.12	Evaluate your tools . . . . .	19
2.4	Versioning . . . . .	19
2.4.1	Versions . . . . .	20
2.4.2	Best practices in versioning . . . . .	20
2.4.3	The Essential Versioning Information . . . . .	21
2.5	Related Applications . . . . .	21
2.5.1	Splice . . . . .	21
2.5.2	Soundcloud . . . . .	22
2.5.3	Allihoopa . . . . .	23
<b>3</b>	<b>Methods</b>	<b>25</b>
3.1	Design Science . . . . .	25
3.1.1	Guideline 1: Design as an artifact . . . . .	25
3.1.2	Guideline 2: Problem Relevance . . . . .	25
3.1.3	Guideline 3: Design Evaluation . . . . .	26
3.1.4	Guideline 4: Research Contributions . . . . .	26
3.1.5	Guideline 5: Research Rigor . . . . .	26
3.1.6	Guideline 6: Design as a Search Process . . . . .	27
3.1.7	Guideline 7: Communication of Research . . . . .	27
3.2	Data Gathering . . . . .	27
3.2.1	Quantitative Data . . . . .	28
3.2.2	Qualitative Data . . . . .	28
3.2.3	Observation . . . . .	28
3.2.4	Gathering Participants . . . . .	28
3.2.5	User Test . . . . .	29
3.2.6	Focus Group . . . . .	29
3.2.7	System Usability Scale . . . . .	29

3.3	Kanban . . . . .	29
<b>4</b>	<b>Prototype Development</b>	<b>31</b>
4.1	Tools and Technologies . . . . .	31
4.1.1	Wamp Server . . . . .	31
4.1.2	JQuery Ajax and JSON . . . . .	31
4.1.3	HTML . . . . .	32
4.1.4	Cascading Style Sheets . . . . .	32
4.1.5	PHP and phpMyAdmin . . . . .	33
4.1.6	MySQL . . . . .	34
4.1.7	Javascript . . . . .	34
4.1.8	Git . . . . .	34
4.1.9	Photoshop . . . . .	35
4.1.10	Scalable Vector Graphics . . . . .	35
4.1.11	D3.js . . . . .	35
4.1.12	Trello . . . . .	35
4.2	Requirements . . . . .	36
4.2.1	Scenarios . . . . .	36
	Scenario 1 . . . . .	36
	Scenario 2 . . . . .	37
4.2.2	Functional requirements . . . . .	38
4.2.3	Non-Functional Requirements . . . . .	38
4.3	Iterations . . . . .	39
4.3.1	Design Iteration . . . . .	39
4.3.2	Iteration 1 . . . . .	40
	User Registration and Login . . . . .	41
	Audio Files . . . . .	43
	Project and Version Creation . . . . .	43
	Version tree . . . . .	44
	Color and contrast . . . . .	45
	Focus Group . . . . .	47
4.3.3	Iteration 2 . . . . .	47
4.3.4	Restructuring of the menu . . . . .	47
4.3.5	Review System . . . . .	48
4.3.6	Final User Evaluation . . . . .	49
<b>5</b>	<b>Results</b>	<b>51</b>
5.1	ECHO Prototype . . . . .	51
5.1.1	User Creation and Login . . . . .	52

5.1.2	Project Creation . . . . .	53
5.1.3	Uploading Audio Files . . . . .	55
5.1.4	Version Tree . . . . .	55
5.1.5	Review Functionality . . . . .	57
5.1.6	Review Feedback Functionality . . . . .	60
	Calculating Version Scores . . . . .	60
5.2	MySQL Database Tables . . . . .	61
<b>6</b>	<b>Evaluation</b>	<b>65</b>
6.1	Focus Group Session . . . . .	65
6.1.1	The Participants . . . . .	65
6.1.2	User Testing . . . . .	66
6.1.3	Focus Group Results . . . . .	66
6.2	System Usability Scale and User Test . . . . .	69
6.2.1	Participants . . . . .	69
6.2.2	Test Setup . . . . .	69
6.2.3	SUS Score . . . . .	70
6.2.4	Learnability Score . . . . .	71
6.2.5	Task Time and Completion . . . . .	72
6.2.6	Expert User Interviews . . . . .	73
<b>7</b>	<b>Discussion</b>	<b>75</b>
7.1	Theory . . . . .	75
7.2	Methods . . . . .	76
7.2.1	Design Science . . . . .	76
7.2.2	Data Acquisition . . . . .	76
	User testing . . . . .	76
	Focus Group . . . . .	77
	Expert Interviews . . . . .	77
	System Usability Scale . . . . .	77
7.3	Compliance with Shneiderman's Principles . . . . .	78
7.4	Answering Research Questions . . . . .	82
7.5	Research Limitation . . . . .	83
<b>8</b>	<b>Conclusion</b>	<b>85</b>
8.1	Future Research and Development . . . . .	86
8.1.1	Further System Development . . . . .	86
8.1.2	Login, Registration and Users . . . . .	86
8.1.3	Version Tree and Core Functionality . . . . .	86

8.1.4	Review and Feedback . . . . .	87
8.2	Further Research . . . . .	87
	<b>Bibliography</b>	<b>89</b>
A	<b>Approval from NSD</b>	<b>93</b>
B	<b>Informed Consent Form</b>	<b>95</b>
C	<b>Focus Group Tasks and Questions</b>	<b>97</b>
D	<b>System Usability Scale Questionnaire</b>	<b>99</b>
E	<b>Final Evaluation Documents</b>	<b>101</b>
E.1	Final User Test Tasks . . . . .	101
E.2	Task Time . . . . .	103





# List of Figures

2.1	Splice web application . . . . .	22
2.2	Soundcloud website . . . . .	23
2.3	Allihoopa website . . . . .	24
4.1	Scenario 1, Interaction from artist to fans . . . . .	37
4.2	Scenario 2, Interaction from fans to artist . . . . .	37
4.3	First sketch of the ECHO system . . . . .	40
4.4	Second higher-level sketch of the ECHO system . . . . .	40
4.5	High level BPMN overview of the system . . . . .	41
4.6	Initial kanban board with functionalities as cards . . . . .	42
4.7	Upload form in the ECHO system . . . . .	43
4.8	The form responsible for submitting title and description of a project . . . . .	44
4.9	D3.js example version tree structure . . . . .	44
4.10	ECHO menu layout consisting of four buttons . . . . .	45
4.11	Color codes used through the ECHO system . . . . .	46
4.12	Change theme checkbox in both on and in off position . . . . .	46
4.13	The revised placement of menu buttons . . . . .	47
4.14	The review system . . . . .	48
5.1	The ECHO system, two different color schemes . . . . .	51
5.2	Login screen screen on the left, and registration screen on the right	52
5.3	Example of the screen the user sees while entering the system .	53
5.4	Creating a new project, example input fields . . . . .	54
5.5	Error message, if the user forgets to fill one of the required fields	54
5.6	The application state after creating a project . . . . .	55
5.7	Unique icons generated for projects . . . . .	56
5.8	Uploading audio files . . . . .	56
5.9	Audio files as they appear after uploading . . . . .	56
5.10	Version buttons on the left and edit HTML form on the right . .	57
5.11	Version box on the left and create new version HTML form on the right . . . . .	57

5.12	Active review system example . . . . .	58
5.13	Date picker in the review panel . . . . .	58
5.14	Review dashboard, where other users excluding the artist can rate and comment . . . . .	59
5.15	The figure on the left shows versions ready to receive feedback, the figure on the right shows commenting and rating of the ver- sion . . . . .	59
5.16	Example of feedback in the version box . . . . .	60
5.17	Data fields for the user table in phpMyAdmin . . . . .	61
5.18	Data fields for the user settings table in phpMyAdmin . . . . .	61
5.19	Data fields for the project table in phpMyAdmin . . . . .	62
5.20	Data fields for the project icons table in phpMyAdmin . . . . .	62
5.21	Data fields for the version table in phpMyAdmin . . . . .	62
5.22	Data fields for the reviews table in phpMyAdmin . . . . .	63
5.23	Data fields for the review relations table in phpMyAdmin . . . . .	63
5.24	Data fields for the review ratings table in phpMyAdmin . . . . .	64
5.25	Review comments table storing users and their comments and timestamp . . . . .	64
6.1	Individual total SUS scores, for every user in the user test . . . . .	70
6.2	Average SUS scores (Adapted from William, 2008) . . . . .	70
6.3	Collection of individual and total SUS scores from the second evaluation . . . . .	71
6.4	Average score per question using SUS . . . . .	71
6.5	Time per task in seconds for participants . . . . .	72
6.6	Time spent for task completion . . . . .	73

# Chapter 1

## Introduction

### 1.1 Motivation and Background

For the past ten years, I have been working with electronic music, it has been my passion and hobby for over one-third of my lifetime. In recent years this has made me a semi-established artist in the progressive house and trance genre, under the production alias "Naden". This interest in building and refining ideas into music, taking strange sounds and melodies, and weaving an audiovisual experience in time, has led me to some interesting ideas of how to go through the process of making music.

I was also interested in utilizing technology in a creative process such as making music, keeping an overview of iterative versions of music was one of the tasks I focused on. The prototype system that I have developed is a web application, the system was called the ECHO system (Expressive Creative History Overview), an overview and feedback system for creative work, in this case music.

When using versioning software such as Git (GIT, 2017) for example, the goal is often to reach a final state of a shippable product. With art, it is often not the final version that is the best version. According to Fischer and Nakakoji, to do creative work means that the final design is not known, and thus the goal should be to explore the potential conceptual space of an idea (Fischer and Nakakoji, 1994). Sometimes an artist will have to build on previous musical versions, to prevent the soundtrack to appear "over produced" or "inappropriately produced". The choice of making a musical composition the best it can be, rather than just settling on a personal preferred version is something that I and many other artists value. Some artists may not value working with music in this way, and thus some of the functionalities included in the ECHO system, may not be of value to them.

### 1.1.1 The One Man Band

The creative landscape has changed tremendously the last century due to technology. One of the art forms that has changed the most over the years is music creation. Back in the early days of electronic music there still was a band structure in most electronic bands. There was a keyboardist, drummer and guitarist. Feedback between the band members was exchanged naturally between them, and thus the resulting work was a product of many minds.

In electronic music today, productions are often done in a "one man band", every virtual instrument is played and controlled by one person. With today's electronic artists, feedback is sometimes only between the artist and the fans, often the artist will only get criticism on music after it has been released. The duration of this feedback loop could start months after the music was initially created.

By implementing a system between the artist and fans that can blind test versions, it could be possible to give the artist an impression of what version is the most optimal, as well as improving the thought process behind a musical project. The feedback between band members in the old days can be simulated again.

There are commercially similar systems to the one I propose, such as Soundcloud for example (Soundcloud, 2017), users can upload their music and get feedback on their tracks with markers right on the waveform of the track. Soundcloud is focused on the listening experience of music rather than delivering valuable, detailed and unbiased feedback to the artist. At the time of writing, the system also has no rating system on the content, only a "heart" button. Splice is also a system that closely resembles Soundcloud, however it has more focus on collaboration between artists and promoting software, not so much on feedback, it also has a powerful version control system in the form of a version tree (Splice, 2017). Another notable system is Allihoopa (Allihoopa, 2017). Allihoopa is a hybrid of Soundcloud and Splice but it focuses more on the stylistic presentation of music. In comparison to Soundcloud and Splice, Allihoopa is considered to have weaker functionalities than Soundcloud and Splice. Section 2.5 describes in more depth about the differences between these platforms.

Since these three platforms are developed by large companies, it was not possible to create a system that outperformed them in terms of response time, upload speed and security. However it was possible to create a prototype that could explore the idea of organizing music versions, and testing different versions of creative work on listeners to receive feedback.

### 1.1.2 Why add an extra step?

Commonly there are not many logistical steps between the finalized production of a track, to when it is released the public. Fans trust the ability of the artist to make the right production choices, and present them the best possible version of the musical production. In this thesis, a structuralist view of working with music in creativity support tools is explored, a workflow of trying to push the musical production to the highest point technically and creatively. Personally when it comes to a project, I tend to end up with a couple of versions that are based on the same foundation of sounds but with a different feel in each one of them. What version becomes the final production is just my own intuition, and I have often been proved to be wrong in what version I thought would be the best. A real example of this, is when I released the track "Snow In Cairo", the track was far from finished, and I would not have considered it ready for release. I lost the project files of the project, and ended up having to release the unfinished version. To my surprise, the track ended up being one of my most appreciated works. In retrospect, had I developed this track through the ECHO system, I dare say that the fans would have given an indication when the track reached its peak at this point in production.

What is the point of having an extra layer of software upon an already complex work process such as making music? Since the process of music and art is so audiovisual, why not make creation process visual, as well. The proposed system that was developed, was based on ideas I had generated through my own experience and discussions with fellow artist friends. According to Paul Graham in his book "Hackers & Painters: Big Ideas from the Computer Age" (Graham, 2004), you get good design if the designer himself is an intended user of the software:

*"You're most likely to get good design if the intended users include the designer himself. When you design something for a group that doesn't include you, it tends to be for people you consider to be less sophisticated than you, not more sophisticated. That's a*

*problem, because looking down on the user, however benevolently, seems inevitably to corrupt the designer."*

Being an intended user of the software, and the designer as well, proved to be challenging. It was important to ensure that the development of the prototype followed principles and rules established in the research of creativity support tools.

## **1.2 Research Questions**

The research questions listed below encompass the two main functionalities of the system that was developed. The first one was based around the core concept of the system, visualization and organization of music versions. The second question was about the review system which is supposed to help the artist make decision on what version to choose.

1. Is it possible to develop a creativity support tool that gives content creators a better overview of their ideas?
2. To what degree can a creativity support tool help content creators make easier choices, and decide which version to finalize?

### **1.2.1 Research Question 1**

This first question addressed the issue of storing and keeping track of progress in creative projects. Often in electronic music, a project can have a large amount of versions, from my experience a number between 5 and 30 is not uncommon. Developing a overview system that visualizes the versions and branching of one's project, is something that myself and fellow artists could see potential in.

### **1.2.2 Research Question 2**

The second research question addresses the issue of knowing when to stop a project. In art forms such as painting and music making it will sometimes be commercially beneficial to stop the exploration of an idea or project. Progressing further will maybe end up making it sound "inappropriately produced". The majority of artists refer to this concept as a production that is "overproduced", meaning it has gone past a point where the musical production would appeal to more users. Inappropriately produced is the point of which a musical production becomes stuffed with too many effects or unnecessary melodic

clutter. The creation of the review functionality inside the ECHO system was a tool that could potentially help solve this.

## **1.3 Chapters Overview**

The thesis has in Chapter 1 started with a background and motivation for the creation of a system. Chapter 2 is about relevant theories and similar applications. In Chapter 3 the methods and methodologies used in the thesis were presented and explained. The development tools of the system as well as iterations and testing is written about in Chapter 4. In Chapter 5 the resulting prototype was presented. In Chapter 6, the evaluation of the prototype, the different evaluations and their results were discussed. Chapter 7 discussed the thesis as a whole: methods, theories and the prototype in relation to the research questions. In Chapter 8 conclusion and future work was presented.





## Chapter 2

# Literature Review

The first step of my thesis was to review and research the topic of creativity support systems. Google scholar and Mendeley were used to find relevant research using search words such as: "creativity", "creativity and technology", "creativity support systems" and "creativity support tools". In this literature review, definitions and concepts generated around creativity were explored and discussed. Before exploring creativity support tools, it was important to have an understanding of what creativity meant.

In addition to establishing the notion of creativity, a review of creativity support tools theories, or sometimes called creativity support systems, was conducted. The research area of creativity support tools is relatively small, but as technology advances, further research is being conducted on creativity in relation to technology. There is a large number of disciplines discussed in creativity support tools research, this thesis is focusing mostly on the art making aspects of creativity support tools.

### 2.1 Creativity

Before exploring the field of creativity support tools, the fundamental concept of creativity was first reviewed, what definitions and views currently exist was explored. An incredibly large amount of work has been written on creativity, many different definitions and explanations on the phenomenon has been proposed by researchers. The central definitions of creativity was adopted in this thesis, and how do these definitions translate into to producing artifacts such as creativity support tools.

One of the most widely known researchers on creativity is Margaret Boden. She is a research professor in the field of cognitive science, and has written a number of publications since 1972. Much of her work has been focused on

creativity, specifically the book: "The Creative Mind: Myths and Mechanisms" (Boden, 2004). In this book she presents her definition of creativity as the following:

*"Creativity is the ability to come up with ideas or artifacts that are new, surprising, and valuable."*

Boden further explains that ideas can be many things such as: concepts, poems, musical compositions, scientific theories, cooking recipes, choreography, jokes and etc. And that artifacts are objects such as paintings, sculptures, engines, vacuum cleaners, pottery, origami. After this she further argues that creativity is not a skill reserved only for the elite, it is not a special "faculty" but an integral part of human intelligence in general (Boden, 2004).

### **2.1.1 Romantic and Non-Romantic**

In the book, Boden categorizes psychologists in two types, romantic and non-romantic. The romantics have a spiritual view on creativity, viewing it as a mysterious concept which is emerging from unconscious unseen processes. To the romantics the gift of creativity is something that one is born with, talent is innate and can be squandered but not taught, she writes. In summary Boden argues that the romantic view of creativity is nothing more than suggestive and it provides no understanding of creativity. Non-romantic views of creativity often include theories that are bound in scientific results and research (Boden, 2004).

This thesis will reference definitions that are based on non-romantic views, theories and systems for creativity that have been tested and reviewed. First a look at how the idea finding aspect of creativity is divided, later the creative process and then the creative product. The three different types of idea finding, Combination, exploration and transformation describes the different ways the mind work to produce a creative idea. The idea generated is what drives the creative process that later produces the creative output or product.

### **2.1.2 Idea Finding**

In regards to creative ideas, Boden argues that ideas have to be not only new but also new to the creator for it to be considered creative. Boden also focuses

her theory on three different categories of creativity: Combination, Exploration and Transformation (Boden, 2004).

### **Combination**

This first type of creative thinking pattern in idea finding Boden presents is Combination. It involves creating unfamiliar combinations of familiar ideas, such as for example taking a politician and comparing him or her to an animal. The combined idea could have been generated through a random event, however the idea has to mean something to someone, or prove a point (Boden, 2004).

It can be argued that no creative idea is truly original, because new artifacts and ideas build upon previously recorded information or thoughts. This is the case in music creation, a musical genre is the result of taking previous ideas and standards and combining them into something new. There are different types of originality in regards to creativity, psychological and historical creativity (P-Creatvity and H-Creativity) which is discussed in Section 2.1.2.

### **Exploration**

Exploratory creativity is the second form of creativity that Boden discusses in her book, in this form of creativity, a conceptual idea structured is explored inside the mind. The conceptual space in your mind is a combination of the domain the idea is based in, and your previous experience with the concept (Boden, 2004).

For example, when an artist makes music, sometimes starting out with exploration can be very powerful method in generating a new idea. The artist will maybe start out by creating a simple melody, then expand and explore the idea in different directions, changing the internal rules set in the conceptual space around the genre.

### **Transformation**

The last form of creativity Boden discusses is transformation, conceptualizing ideas by changing one's way of thinking.

*"The deepest cases of creativity involoe someone's thinking something which, with respect to the conceptual spaces in their minds, they could not have thought before*

(Boden, 2004). "

She then describes that the impossible idea can exist if the previous thinking-style is changed in one form or another. The style of thinking is changed and transformed, resulting eventually in ideas and concepts that previously were unheard of, and unthinkable. You have to know the rules before you can break them, this is exactly how transformational creativity works to generate ideas.

All of these three types of creativity intertwine and make up what is considered the creative mind, different styles of creative thinking exist by utilizing these types of creative thinking on different levels. In the next section P-creativity and H-creativity is explained, when creative ideas are generated they can be either psychologically creative or historically creative.

### **P and H-creativity**

When discussing creativity in the sense of originality, Boden mentions two different types of originality in creative work: P-Creativity and H-Creativity. Boden distinguishes between what is called psychological creativity (P-creativity), and historical creativity (H-creativity).

Psychological creativity means the emergence of an idea, or concept that is new to the person that synthesized it. The idea might not be significant historically, but even though the idea has been conceptualized before, the idea can still be called creative. H-creativity is similar to P-creativity in that it is still new and significant to the creator, but the ideas are different from P-creative ideas in that they are historically original (Boden, 2004).

Different fields of research value these two forms of creativity differently. For example, in the field of psychology when researching personal creativity, P-creativity will be valuable. Studying the thinking pattern of a person, that led to generating an idea, could be basis for using P-creativity. In contrast, engineering research might be more interested in historical creativity since much of the research builds upon novel artifacts instead of researching origins of an idea.

### 2.1.3 The Creative Work Process

There has been done a large amount of work on the creative process through the last fifty years, however according to Todd Lubart the models are not detailed enough in how the different processes generate creative ideas (Lubart, 2001). In this section the focus is on defining the overall schools of the creative processes rather than the sub processes which is not agreed upon by many researchers.

In his paper "Creativity Support Tools: Accelerating Discovery and Innovation" (Shneiderman, 2007), Ben Shneiderman mentions that a large amount of work has been done on creativity throughout the years, especially in the cognitive sciences. But in relation to computer and information science, the research of creativity is scarce at best. Shneiderman further writes that creativity can be narrowed down to three different "schools": structuralists, inspirationalists and situationalists.

#### **Structuralists**

The structuralists view of creativity is that in order to be creative, different steps has to be taken in order to reach a creative product or idea. Preparation, incubation, illumination and verification could be some of these steps (Shneiderman, 2007). Having a structuralistic creative work process, is then to step outside of individual personal organization of ideas and rather utilize an external methodological way of thinking, such as using creativity support tools to structure ideas.

#### **Inspirationalists**

The inspirationalists view on creativity is breaking away from familiar structures and systems will elicit more creative solutions. Shneiderman also writes that the inspirationalists will encourage the change of location, traveling to scenic locations and look at random photos and structures. He also describes that inspirationalists promote techniques such as meditation, hypnosis, dreaming, playful exploration, sketching, concept mapping and visualization as well, to discover hidden ideas and to see a bigger picture. The whole goal of inspirationalists can be summarized down to breaking free of preconstructed methods of creativity so that they can break through what he calls the "aha!" or eureka moment (Shneiderman, 2007).

### **Situationalists**

The situationalists view of understanding creativity is situated on the social structure around the creative subject. They will try to understand all the social aspects behind the creative mind such as: investigating family history, their relationships with teachers, peers and mentors. They also argue the need for consultation during early stages of creativity to avoid what Shneiderman describes as being "ridiculed", "rejected", "a rip-off". In later stages of a creative project, concepts such as validation, refinement and dissemination are studied. Shneiderman summarize that situationalists try to understand the rewards and recognitions one would get as a creative person (Shneiderman, 2007).

These three views on creative processes are largely agreed upon by researchers in different fields as the three different areas of researching creativity. This thesis adopts a structuralistic view for the development of the creativity support tool prototype.

#### **2.1.4 The Creative Product**

The creative product is generally agreed upon as being either a concept or an artifact, and it differs in psychology as opposed to engineering design that a creative product can be an idea rather than a finished product. Psychologists with a romantic view see the creative product as something magical and mysterious (Boden, 2004), but the focus in this thesis is on non-romantic views, romantic views will not be discussed.

Howard et al in their paper: "Information as an input into the creative process" (Howard, Culley, and Dekoninck, 2006), present their own definition of the creative product, or as they call it "creative output". They argue in their paper that even though in the engineering design process a creative product is a solution, they acknowledge that their definition is about the creative "idea" and can thus also be used as a definition of thoughts rather than only tangible artifacts (Howard, Culley, and Dekoninck, 2006). They propose that a creative output or product is the following:

*"A novel idea that is both unobvious and appropriate. "*

They discuss three different concepts or "attributes" in regards to this definition: Originality, Appropriateness and Obviousness, these are explained in the next section.

### **Originality**

When defining originality, Howard et al reference Boden's work in regards to P-Creativity and H-Creativity (see Section 2.1.2). They discuss the concept of P and H-creativity in that if the research domain is psychological or engineering, P and H-creativity is of different importance. Howard et al further explain originality, in that unrestrained creative tasks such as working with art, originality is easier to achieve, because the end goal is not often known. In contrast, when working with a product or solution that moves towards a goal, all legal moves and operations are known, and the process goes from being creative to a more logical process, thus less ideas are generated (Howard, Culley, and Dekoninck, 2006).

### **Appropriateness**

Sometimes the result of research of creative processes leads to artifacts or ideas that are new and novel. It is then up to the researcher or the creator to evaluate the results and then add context and deem it appropriate or not. Howard et al reference the famous example of the two researchers Robert Woodrow Wilson and Arno Penzias. They had presented one of the most important discoveries in the history of cosmology, by discovering the background radiation of the big bang, completely by accident. Although they did not know the significance of their discovery at the time, they later became aware of it, when a friend of Penzias told them about a paper that was being written about the explosion that was the origin of the universe. They were awarded the Nobel Prize for Physics for understanding their data, putting it in context and judging its appropriateness to an existing field. Howard et al then argue that understanding appropriateness is coupled with the implementation of creative ideas, which in turn lead to innovative works (Howard, Culley, and Dekoninck, 2006).

### **Obviousness**

The last requirement for creativity Howard et al discusses is obviousness. For an idea to be considered creative, it has to be unobvious, Howard et al write that the concept is not easy to quantify or qualify. They describe an example with two different commercial competitors who both face the same problem,

and execute the same solution or idea, leading to an obvious idea. The resulting obvious idea is considered less creative than an idea unobvious to the creator (Howard, Culley, and Dekoninck, 2008).

## 2.2 Creativity Support Tools

In creativity research, the technological revolution led to many new fields of study, around the turn of the century the research area of creativity support tools started being discussed. Creativity support tools research is interdisciplinary, some of the disciplines are cognitive, human computer interaction and engineering design. These tools are created with an aim to augment human cognitive thinking and to build better artifacts that make creators more creative. One of the most notable researchers in the field of Creativity support tools Ben Shneiderman describe creativity support tools as following:

*“Creativity support tools extend users’ capability to make discoveries or inventions from early stages of gathering information, hypothesis generation, and initial production, through the later stages of refinement, validation, and dissemination.”* (Shneiderman, 2007).

In their paper about context awareness in creativity support tools (which was found to be outside the scope this thesis) Siellis et al present a more general definition:

*“Creativity support tools are used to simulate the creativity techniques and create environments which guide the user to become more creative”* (Siellis, Tzanavari, and Papadopoulos, 2009).

In the next section, Shneiderman’s principles in relation to creativity support tools are explained. The principles act as guidelines that must be fulfilled in order to make a system that support the creative process. These principles was followed and used as guidelines in creation of the ECHO prototype.

## 2.3 Schneiderman’s Principles

Ben Shneiderman’s principles on creativity support tools, is a set of 12 overall design principles that acts as a guide for developers making such tools. Some of the principles have appeared in earlier work on creativity support tools, but



the first formalization of the 12 principles was presented in a creativity support tools workshop report in 2005 (Resnick et al., 2005), the NSF sponsored workshop gathered 25 research leaders and graduate students to exchange experience, and discuss research on creativity support tools.

The following principles proposed by Shneiderman et al, describe loosely what functionality and methods should be implemented in an application designed to support content creators. The principles have been developed by Ben Shneiderman and many colleagues during the research of different creativity support tools. They do specify that the principles are not fully fleshed out and they are open for discussion and interpretation, however I found them to be suitable for my thesis. The principles are listed below:

1. Support exploration
2. Low threshold, high ceiling, and wide walls
3. Support many paths and many styles
4. Support collaboration
5. Support open interchange
6. Make it as simple as possible and maybe even simpler
7. Choose boxes carefully
8. Invent things that you would want to use yourself
9. Balance user suggestions with observation and participatory processes
10. Iterate, iterate then iterate again
11. Design for designers
12. Evaluate your tools

### **2.3.1 Support Exploration**

Supporting exploration in a creativity support tool means that a creativity support tool should enable the user to explore the system itself. The first requirement for exploration, is that for the users of the software, it must be easy to try out things and see the effect of the functionality, and also to be able to backtrack if necessary. Resnick et al propose that "undo" functionality can be very powerful in creativity support tools, but can be hard to implement. The second requirement proposed is that tools should be "self-revealing", meaning that what can be done within the system is clear to the user (Resnick et al., 2005).

### 2.3.2 Low threshold, high ceiling, and wide walls

With low threshold, it meant that the user should have an easy way to understand and use the application just by looking at it, it should be easy for novices to pick up the tool and get started. The tool should have a high ceiling so that expert users can potentially expand their projects and explore more complex ideas. A creativity support tool can be said to have "wide walls" if the tool have a large range of functionalities to explore and test. Concerns are discussed on this point, systems with low threshold usually have limited functionality, and in contrast, systems with high ceilings are hard to get into. The challenge is making tools that are easy to use but support powerful and rich functionality (Resnick et al., 2005).

### 2.3.3 Support many paths and many styles

The third principle, is that the tool should support different styles of creative work processes. The application structure should allow for customizable layout, and the user should be able to change for example background colors/images. This is so that the creative worker can set up the tool to match the working setup inside his or her mind. In other words the user should not be restricted to working and using the software in a specific way to reach his or her goals (Resnick et al., 2005).

A good example of a creativity support tool that support this principle is FL Studio (FL Studio, 2017). This tool supports almost infinite ways of creating music, it does this by having a set of basic powerful functionalities such as a step sequencer, piano roll and mixer. The user is free to build upon the powerful platform by using plug-ins and custom third-party tools.

### 2.3.4 Support Collaboration

Supporting collaboration is the fourth principle discussed by Resnick et al. Collaboration in this sense, meaning a collective effort to bring a creative project to a desired state. They mention that creative work is sometimes done by groups of people, and since the introduction of the Internet, it has become much easier to support collaboration over long distances. They also expand on the concept of collaboration in that it is not exclusive to just work together in a system, but can also be just exploring the techniques and ideas of others on

the same creative problem, through search engines such as Google for example (Resnick et al., 2005).

### **2.3.5 Support Open Interchange**

Often the creative process requires the introduction of external tools and transferring of project data. To encourage collaborative work, the creators need to have the ability to customize tools for exchange and control of their projects. Enabling the users to exchange their ideas and projects is crucial for the development of a creative idea (Resnick et al., 2005). This ties back the example in Section 2.3.3 where FL studio has the ability to orchestrate different plug-ins.

### **2.3.6 Make it as simple as possible and maybe even simpler**

The system should, as mentioned earlier, have a low threshold for new users and high ceiling for advanced users, this means that advanced functions should be hidden in some way, but at the same time easily accessible. Making a system simpler can sometimes result in reducing the ceiling, and lower the threshold. Resnick et al discuss that sometimes a simpler and less fleshed-out featurewise tool will lead to a better user experience, and it might not necessarily be bad for the creator to have fewer features, since it can lead to more creative solutions (Resnick et al., 2005).

### **2.3.7 Choose black boxes carefully**

Black boxes are what the user does not see when interacting with a system. It can be difficult to choose which elements the users are able to interact with, because it will directly affect what the user can do with the tool. Giving the user too much control over the inner workings of a system can lead to confusion and a high threshold (Resnick et al., 2005). By carefully planning what the user should be able to manipulate and see, it makes it easier to prevent erroneous choices.

### **2.3.8 Invent things that you would want to use yourself**

Similar to Paul Graham (Graham, 2004), Resnick et al also claim that one is more likely to do a better job as a designer of a system when the intended user is oneself. They then mention that the users should be presented a tool or system that the designers themselves would want to use, as it is more respectful to

the users. Creativity support tools do not succeed in a vacuum, having a large community built around the tools, the result is a more enjoyable experience for the content creators (Resnick et al., 2005).

### **2.3.9 Balance user suggestions, with observation and participatory processes**

Involving the users early in a project is necessary in order to specify requirements. Resnick et al argue that successful system developers and designers spend time with the users, taking notes from focus groups and individual user tests. However they also mention that users can sometimes ask for functionality that is too flexible, resulting in tools that are possible to break. They suggest that restricting the flexibility to only where it would make a difference in the experience of the user (Resnick et al., 2005).

An example of this, is the software synthesizer "Sylenth1" (Sylenth1, 2017), it has a set of parameters or "knobs" that have a maximum limit, the knobs can not turn past a certain point in all cases. The result is that the tool will be harder to break, and it ensures that the sounds produced by an artist is within boundaries of what is listenable.

### **2.3.10 Iterate, iterate then iterate again**

This principle builds on the fact that no great work comes without trying new things and rigorous experimentation. By having many iterations, the tool can be chiseled out in finer steps, resulting in a tool that is more likely to meet the expectations of the end-users. Resnick et al emphasize that the principle both apply to the user and the developers of the system, the users themselves should want to iterate on their ideas (musical versions in the case of this thesis). They highlight three simple steps that are important to the iteration of a prototype:

1. As a developer it is important to observe the users of the tool and record their experience by non-intrusive methods
2. The result of the observation is analyzed and changes to the design are applied
3. When the new design has been decided upon, implement the new design into the prototype and repeat the first step

### 2.3.11 Design for designers

Resnick et al explain that designers want to have tools that help them invent new things, they want simple tools that facilitate powerful builds and projects (Resnick et al., 2005). They also mention the example of LEGO, where individual bricks, while very simplistic, can be used as building blocks to create more complex structures. In terms of music creation, any musical digital audio work station, can be used to first create individual pieces of music, which can then be assembled into bigger pieces. The digital audio workstation is in some sense a software version of a LEGO system.

### 2.3.12 Evaluate your tools

Evaluation of creativity support tools is universally seen as a challenge to most researchers in the area, since it is not often easy to quantify if a user has become more creative with the use of a tool. However Resnick et al mention that it is possible to use metrics that ensure that the user is not hindered in creativity, meaning that the software should at least be responsive and clear to use. Usability testing, qualitative interviews and questionnaires could be some of the common methods that will ensure that the system is user friendly while not hindering creativity (Resnick et al., 2005).

These principles were utilized as guidelines for the ideation phase of the prototype, and also for structuring the development of the prototype. In terms of evaluation there is some compatibility between Design Science (see Section 3.1) and Schneiderman's principles, as they both support scientific rigor in regards to evaluation of tools.

## 2.4 Versioning

Since the ECHO system implemented version-tree visualization of versions, theory about versioning was used to formulate what information should be recorded in the system along with the music files. Jenny Brace wrote an article suggesting some techniques and implementations developers of versioning systems should take into consideration. The best practices she points out comes from The Version Identification Framework (VIF) (Brace, 2008). The VIF was created in order to help develop best practice versioning in repositories, but the practices also apply to any software that uses some form of versioning.

### 2.4.1 Versions

Even though the VIF refers to versioning in the form of creative writing, it is fair to say that the definition also applies to other forms of art such as electronic music. In the article about VIF, Brace explains versioning as following:

*"A "version" is a digital object (in whatever format) that exists in time and place and has a context that can be described by the relationship it has to one or more other objects."*(Brace, 2008).

In regards to the relationship between versions, Brace explains it as following:

*"A 'version relationship' is an understanding or expression of how two or more objects relate to each other."*(Brace, 2008).

Brace then proceeds to further explain best practices in versioning according to VIF.

### 2.4.2 Best practices in versioning

Brace points out some important factors that should be present when making a versioning system. She suggests that the best and most accurate information comes from the content providers themselves at the time of deposit, identifying the version and what relation the object has to others. The deposit and insertion of meta data about the version should "be an essential part of the ingestion process" (Brace, 2008).

She also lists a number of basic needs the system should have when submitting a new version:

- Use a numbering system that denotes major revisions.
- Make explicit the author, title and date last changed, and version status on all versions of work. This can be done:
  - descriptively within the object, for example, on a title page, title slide, first frame of film and so on.
  - by using a clear, updated and relevant filename for every different version.
  - by filling in available "properties" "details" or "ID tags".

After suggesting these essential needs when submitting a version to a system, she expands on what required information should be recorded in the system.

### 2.4.3 The Essential Versioning Information

Brace mentions that the VIF introduces a number of essential functional requirements that should be in place for a versioning system to function optimally, the ones listed below are the ones that will be relevant to the thesis:

- Defined dates
  - A date may be the only information required to differentiate between versions.
- Identifiers
  - An identifier could be a name, id or some other unique identifications such as an unique graphical icon.
- Version Numbering
  - A version number could for example be that a version has a number of 127, the following version could be named 127\_2 either manually or automatically.
- Text Description
  - Written descriptions should be one of the clearest methods for placing the version in relation to other versions.

## 2.5 Related Applications

Creative support tools span many different disciplines, this thesis is structured around the development of a creativity support tool supporting the organization, visualization and review of music projects. To position the thesis work, three similar applications that support music creation are presented: Splice (Splice, 2017), Soundcloud (Soundcloud, 2017) and Allihoopa (Allihoopa, 2017).

### 2.5.1 Splice

Splice is probably the largest of the three applications in terms of offered functionality (see Figure 2.1). Splice focuses on version control of music similar to that of Git (GIT, 2017). It managed the project files of artists through a desktop application which then synchronizes to the web application. The system has powerful visualization tools called DNA visualization, which displays changes and structures of the individual sound samples inside the music project itself.

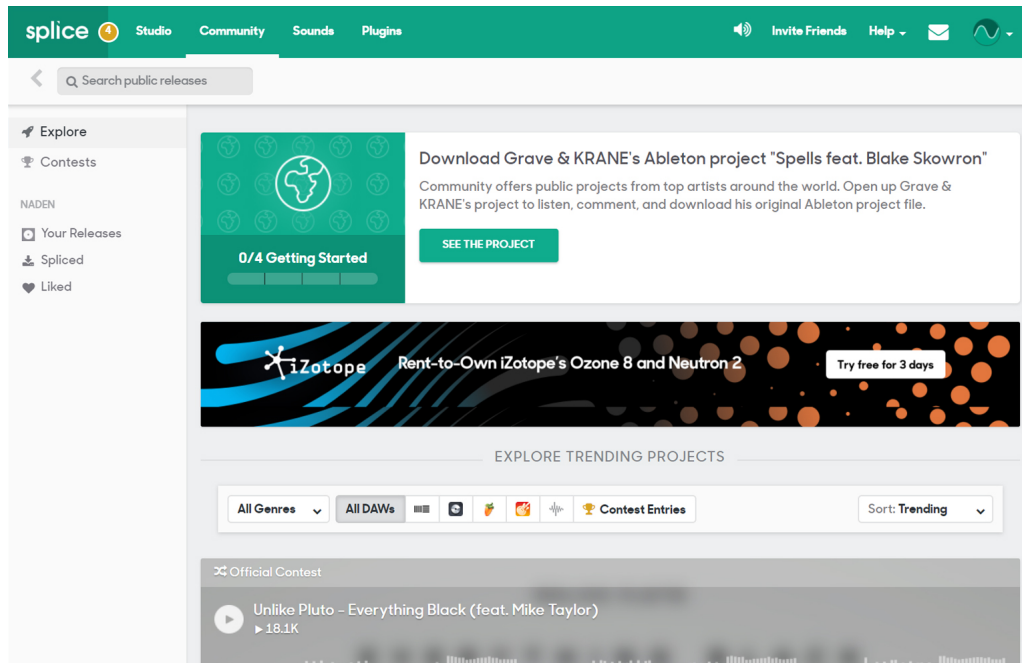


FIGURE 2.1: Splice web application

Although the base functionality of Splice is free, they have a subscription solution aimed at selling samples and audio plugins to the users. The system has tools for exploring already existing music within the Splice application, but this part of the application is of the less focus than the production and version control system.

## 2.5.2 Soundcloud

Soundcloud has the largest userbase of the three systems (Soundcloud, 2017). The system is a website that is aimed at promoting music, but also can function as a creativity support tool. It consists of different types of content providers such as musicians, labels and promoters. The content creators are free to choose how they want to present their work to the listeners, as long as it is within the rules of the website.

One of the functionalities Soundcloud has is a dashboard where new music appears, based on which channels the user is subscribed to (see Figure 2.2). Soundcloud also has a powerful integrated music player that is located at a fixed position on the bottom of the page, enabling queuing and playback of music as the user browses the platform.



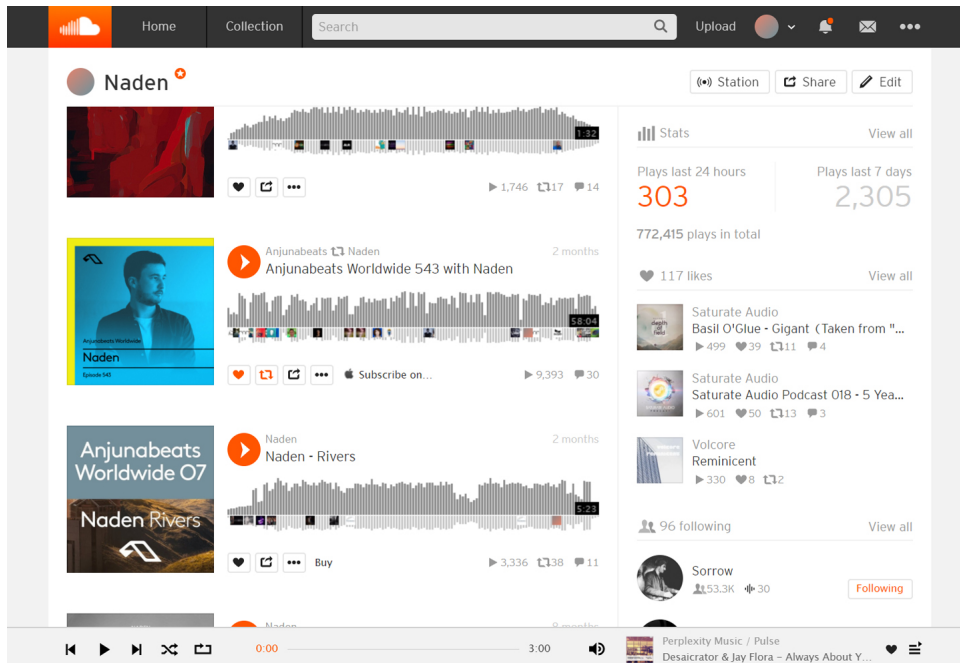


FIGURE 2.2: Soundcloud website

### 2.5.3 Allihoopa

Allihoopa (Allihoopa, 2017) is a stylistic and clean website very similar to both Soundcloud and Splice, but it has a much smaller userbase. The website focuses on collaboration between users, as well as offering some tools for creation of music. However the tools do not have direct integration with the website, and the users are required to upload the tracks to the website via mp3 files, which is similar to how Soundcloud operates.

The overall layout on Allihoopa is very similar to Soundcloud in that it has a dashboard with music relevant to the user's interest, and it also has a fixed player that can play music while the user browses the website (see Figure 2.3).

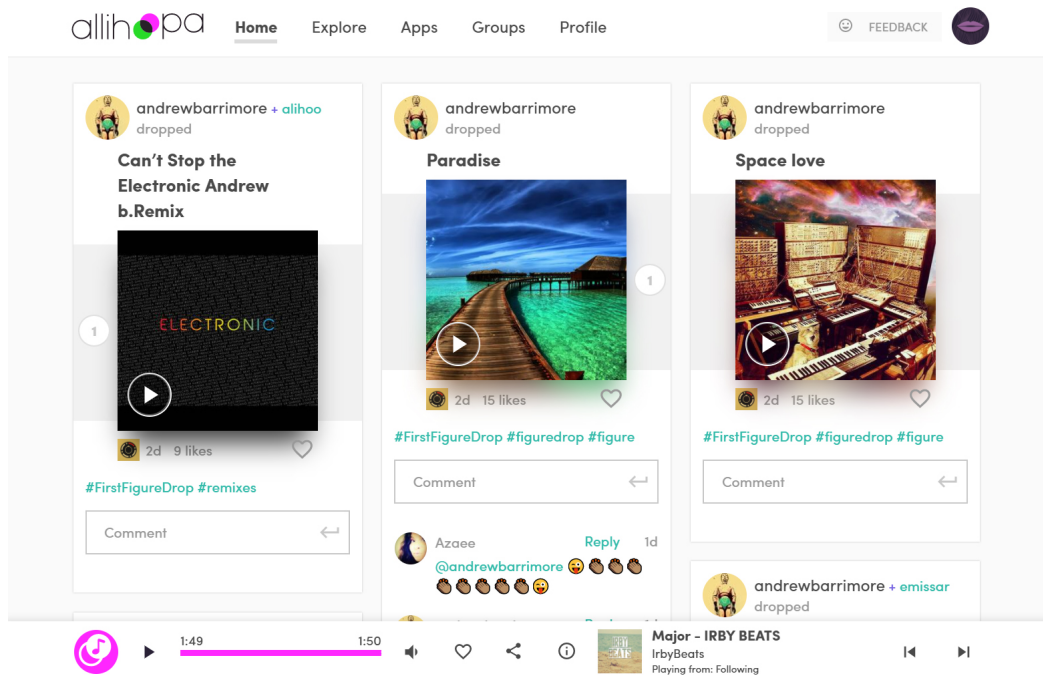


FIGURE 2.3: Allihoopa website

## Chapter 3

# Methods

In this chapter, methods used for the development of the ECHO system are presented. A combination of a framework such as Design Science, Shneiderman's principles 2.3 and Kanban was used.

### 3.1 Design Science

I have used Design Science as my theoretical framework for this study, following in the guidelines listed in the article by Hevner et al (Hevner et al., 2004). Design science is a problem solving process, and the guidelines are created to ensure the quality of research, resulting in an artifact as a novel solution to a research problem.

#### 3.1.1 Guideline 1: Design as an artifact

*"Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation."*(Hevner et al., 2004)

Research done with Design Science must bring something into the world. Either in the form of a model, construct, method or instantiation in this case. Instantiating an artifact is to demonstrate a solution to either a previous information system problem or a new problem. In this thesis the web application was expected to work as an instantiation of a creativity support tool, a solution to the research questions involving version visualization and user feedback.

#### 3.1.2 Guideline 2: Problem Relevance

The objective of Design Science is to design and develop new solutions relevant to important business problems, the artifact created must be innovative and novel in its design. Something of value must be explored and brought to the table of knowledge in information systems (Hevner et al., 2004). My

application was aimed to do this through making a system for helping music creators make an accurate choice based on fan feedback whether a version of work is final or not.

### 3.1.3 Guideline 3: Design Evaluation

*“The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well executed evaluation methods. Evaluation is a crucial component of the research process.”*(Hevner et al., 2004).

Only thorough evaluation of the system will shed light on whether or not the system has succeeded in solving the research questions.

After the individual user testing sessions, users will be asked to give feedback on the use of the software. This will lead either to prove the design effective or faulty leading to further iterations. Evaluation of the system is discussed further in Section 6.

### 3.1.4 Guideline 4: Research Contributions

In this guideline, Hevner suggest that Effective Design Science research must provide clear contributions in the areas of the design artifact, design foundation, and/or methodologies (Hevner et al., 2004). My artifact aims to contribute in the research area of creativity support tools by building a web application based on the principles provided by Shneiderman and colleagues.

### 3.1.5 Guideline 5: Research Rigor

This guideline addresses in which way research conducted is rigorous. The use of techniques and methods that are well tested and defined is important when conducting Design Science Research. Often the goal is to try to explain how well the artifact works, rather than trying to explain why it works (Hevner et al., 2004). Multiple methods have been used in this thesis to test the different iterations of the prototype: expert interviews, focus group, user tests and SUS questionnaire. All of them have been tested and used by many researchers and can be considered rigorous methods.

### 3.1.6 Guideline 6: Design as a Search Process

*“Design Science is inherently iterative, the search for the best, or optimal, design is often intractable for realistic information systems problems.”*(Hevner et al., 2004).

Hevner et al write that problem solving in design is using available means to come to an end. Through knowledge about the application domain effective design can be created. De-constructing problems into simpler problems is one of the processes Design Science uses. An example for this in relation to my application could be to make simple prototype of a rating functionality, cutting up the modules that go into each part of the module and making them function independently.

### 3.1.7 Guideline 7: Communication of Research

*“Design Science research must be presented both to technology-oriented as well as management oriented audiences”*(Hevner et al., 2004).

According to Hevner it is very important that the communication between the researcher and audience is clear and understood. Hevner discusses in this guideline that the artifact produced must be sufficiently documented in the terms of what tools and methods were used, in order for other researchers to reproduce the system and implement it. It is also of importance to sufficiently document the functionalities of the artifact. In case that the artifact has a potential to become a product, it will be important to attract funding for which a proper documentation is essential. In this thesis both technical tools used (see Section 4.1), and a description of how the system works, was presented (see Section 5.1).

## 3.2 Data Gathering

Data for this thesis was collected using a combination of both qualitative and quantitative data. Techniques used for data gathering consisted of a combination of expert interviews, focus group session, user testing and questionnaire. In addition to these techniques, digital recording tools were also used such as voice recorder and screen recorder. Norwegian Social Science Data Services (NSD) (see Appendix A) proved the handling of data for the thesis.

### 3.2.1 Quantitative Data

Data in the form of numbers, or that can be formulated with accuracy into numbers, can be considered quantitative data. For example recording temperatures over a time period and storing the data in a table for later analysis is one example of quantitative data collection. Analysis of quantitative data can be conducted using different numerical and mathematical operations, depending on what the desired result is (Preece, Sharp, and Rogers, 2015). In summary quantitative methods of collecting data are useful for studying trends in bigger datasets.

Quantitative data acquired in the thesis came in the form of numbers. In the second and final evaluation (see Section 6.2) task time was recorded for all users that tested the system. In the same section the System Usability Scale was used as a quantitative measurement of the usability of the system, this was done in the form of a questionnaire that included a likert scale.

### 3.2.2 Qualitative Data

In contrast to quantitative data, qualitative data is often not represented by numbers but by texts, descriptions, images and also quotes from interviews. Analysis of qualitative data can be a focus of the nature of something, and utilizing methods for recognizing themes, patterns or stories. (Preece, Sharp, and Rogers, 2015).

In this thesis qualitative data came in the form of recordings from interviews. The data was transcribed and analyzed, resulting in formulating potential improvement to the prototype.

### 3.2.3 Observation

"Quick and dirty" observation was done in addition to screen and audio recording. Quick and dirty observations can be conducted anywhere and consists of interacting with the user in an unobtrusive and constructive way (Preece, Sharp, and Rogers, 2015).

### 3.2.4 Gathering Participants

For gathering of participants in both focus group and expert interviews, a sampling technique known as "samples of convenience" were used (William, 2008).

It includes locating participants that are willing to participate either through networking or by advertisement.

### **3.2.5 User Test**

A simple user test was conducted using a list of tasks that was to be completed, the user expressed feelings and struggles while executing the tasks. Screen video and audio were recorded, and deemed necessary for a detailed showcase of errors and functions that the user experienced. The resulting qualitative and quantitative data from user tests can be used to improve the prototype through the next iterations.

### **3.2.6 Focus Group**

Focus group was chosen as a method in the early stages of the project, for gathering feedback and ideas. For further development of the application, i.e. after the first iteration, a focus group is considered an appropriate method of choice. For generating ideas focus groups are much more powerful than simple, one to one interviews (Breen, 2006).

Some of the pitfalls often associated with focus groups, are not using other means of data collection (Krueger and Casey, 2000). This is why combining user tests and the focus group for evaluation of the first iteration was considered appropriate.

### **3.2.7 System Usability Scale**

The System Usability Scale questionnaire is a simple likert scale measurement, for evaluating the overall usability of a system (Brooke, 2013). The questionnaire consists of ten statements that the user scores from one to five in how true the statement is to the user. According to Tom Tullis and Bill Albert, if a developer is only interested in uncovering major usability issues, a sample size of three or four should be enough to get useful feedback (William, 2008).

## **3.3 Kanban**

Kanban is an agile system development method that identifies tasks, and arranges them into the tree columns: Done, In progress, To do. Each task is

written on its own card that can be moved between the columns in accordance with the work progress. David J. Anderson in his 2010 book “Successful Evolutionary Change for Your Technology Business” writes a description of the Kanban pull system:

*“..new work is pulled into the system when there is capacity to handle it, rather than being pushed into the system based on demand. A pull system cannot be overloaded if the capacity, as determined by the number of signal cards in circulation, has been set appropriately.”(Anderson, 2010).*

Kanban encourages developers to create a context-specific process rather than blindly following a software development life cycle process definition or template (Anderson, 2010).

The set iteration periods of scrum did not fit with my work flow since it was neither clear or certain how to organize the backlog and discuss the work iterations (sprints). I was a single developer and preferred not to organize bigger iterations without a proper team to give me feedback. Kanban provided enough control and clarity that I could rely on.



## Chapter 4

# Prototype Development

This chapter first discusses what tools were utilized to create the prototype. Thereafter requirements and design choices are presented. The last sections in this Chapter go in-depth about the different prototype iterations that were designed and developed.

### 4.1 Tools and Technologies

This section will present the tools and technologies that were used for the development of the ECHO system. Some of the tools were chosen due to previous knowledge, others were chosen because of a desire to learn more about them.

#### 4.1.1 Wamp Server

The application was developed on a WAMP server locally (Wamp, 2017). WAMP is a stack of technologies used for the windows operating system, consisting of Apache web server (Apache, 2017), MySQL (MySQL, 2017) and PHP 4.1.5. The reason for choosing these tools was due to the fact that I have been developing systems previously with PHP, thus making the WAMP stack a logical choice. Using WAMP in its simplest form consists of installing the stack, putting programming code in the www directory, and then open `http://www.localhost` in a browser window while the server is running. Since I was going to use it for creating a web application, I would be using PHP together with Javascript, Ajax and a MySQL database in the WAMP stack.

#### 4.1.2 JQuery Ajax and JSON

JQuery Ajax is a Javascript framework based on Ajax (asynchronous JavaScript and XML)(Ajax, 2017). Even though Ajax has XML in its name, the most common data format used with Ajax is JSON (Javascript object notation syntax).

Ajax in my experience is a bit complicated to utilize, but using JQuery Ajax, Ajax calls are much more readable and human friendly. Almost all web applications use some form of asynchronous call to the back-end server, and JQuery Ajax is the most used one.

JSON was used extensively throughout the development of the application. Since JSON is based on Javascript objects, it can be parsed to either Javascript code or PHP, depending on where the data is sent/coming from. Parsing the objects and then extracting the data is as easy as packaging the data into an array, and then converting it to JSON.

### **4.1.3 HTML**

Of all the technologies prevalent on the web, "HyperText Markup Language" is the one most widely used (HTML, 2017). It is the most basic building block of any website.

HTML is not a programming language but markup, being a markup language means that it is only information being marked and interpreted by the browser which is then displayed to the user based on rules set by Cascading Style Sheets (CSS) (CSS, 2017). A website or web-application made with HTML typically has a HTML document presented to the user upon entering the system. The HTML document consists of different commands that instruct the web browser to load all required resources. In addition to instructing the browser to load required resources HTML also contains the content of the page itself.

HTML is often used together with CSS and Javascript to create dynamic websites. Dynamic websites presents information to the user without refreshing the page, most websites today utilize some form of dynamic content. Web applications are not very different from dynamic sites, the difference is that the a web application acts more as a real interactive application in the browser, rather than just a page with information.

### **4.1.4 Cascading Style Sheets**

Cascading style sheets CSS is an integral technology on the web, and part of the most commonly used technologies on the web: HTML, CSS and Javascript, which makes it a wise choice for development of a web application. It is used for adding style, animations and structure to a website. The syntax is very easy

to grasp, it consists of a set of "rules" that point to specific parts of the HTML document (CSS, 2017).

For example consider the following CSS code:

```
html {  
background-color: red;  
font-size: 140%;  
}
```

This code snippet tells the browser that the element "html", (which is the entire website) will have red as a background color and a font size that is 40 percent bigger than the default. By utilizing a handful of CSS rules, it is possible to create fully functional and clear static websites.

#### 4.1.5 PHP and phpMyAdmin

PHP (PHP, 2017) is a Hypertext Preprocessor, which is a dynamically interpreted programming language, used mainly for creating dynamic web pages. The language is located at server-side, and is the connecting component between front-end and the database. It can work in a number of ways, but most commonly it is utilized by receiving commands from a user either through GET or POST requests. The programming instructions are sent to the appropriate script, after which it is executed from top to bottom running functions or loops. Preprocessing in PHP happens when the PHP script has finished. The script can either print resulting HTML code onto a page, or it can package the HTML into a JSON object and return it to an Ajax handler. The data is then presented to the user in the form of lists, or other means without the page refreshing (PHP, 2017).

PhpMyAdmin is a free open source tool, intended for handling the administration of MySQL. It is used as a interface to the MySQL database on a server, making it easy to create database tables, modify or even delete tables and fields. By connecting to the database through PHP, a PHP script can construct and send MySQL queries directly to the database. Since PHP is run on the server parallel to the MySQL database, very little delay is generated between a query and a response from the database (PhpMyAdmin, 2017). The combination of PHP and phpMyAdmin is an efficient configuration for the web application.

### 4.1.6 MySQL

MySQL is an Open-source relational database management system or RDBMS for short (MySQL, 2017). It is often used in various AMP stacks such as LAMP or WAMP in this case. MySQL is written in C and C++ and interprets queries from a user, and handles data in a database located on a server. It is commonly used in small to medium projects due to its simplicity. The query is user friendly, and it reads almost as a sentence for example:

```
"SELECT * FROM products WHERE quantity >= 5000"
```

This simple query will return a table with all products that have more or equal to 5000 in quantity. Because of the relative ease of use MySQL provides, its flexibility and speed, it was a natural choice in regards to using it or not.

### 4.1.7 Javascript

Javascript developed by ECMA is a script standard, the most used script language for the web (Javascript, 2017). In a web development setting, Javascript is run locally on a users computer together with other common technologies such as CSS and HTML. Javascript works by manipulating data presented through the HTML document, either by changing animations or even the structure of the data itself. A large number of frameworks, tools and libraries have been created for Javascript by many different developers, both open-source and licensed.

In this project I have used both what is called "vanilla" Javascript, D3.js and JQuery Ajax (Ajax, 2017). One of the many benefits of using vanilla Javascript is that it can prevent bugs introduced through usage of libraries and frameworks. When a large portion of a system is based on a framework, the prevention of failure and bugs is dependent on the stability of the framework. If the framework fails, so does all the code that utilizes it. By using vanilla Javascript such risks are minimized.

### 4.1.8 Git

Git is an opensource distributed version control system (GIT, 2017). A version control system such as Git, works by controlling what is called "diffs", the difference between documents. It allows developers to work in teams due to the separation of local and distributed repositories. Whenever an error is discovered, a team member may reverse the diff that is causing the problem.

Common instructions in Git include commit, push and pull. Git was used to ensure that the system was version controlled, and that backup of all code had been stored.

### 4.1.9 Photoshop

Photoshop is a raster graphics editor developed by Adobe Systems (Photoshop, 2017). It is regarded as one of the most used image processing tools in the world. For me this was a natural choice, since I have a lot of experience with it. The most central part of Photoshop is that a user is able to create transparent layers independent of each other, that way the changes can be done to selected layers while leaving the rest of the image untouched. Photoshop was used to create some of the icons used in the ECHO system.

### 4.1.10 Scalable Vector Graphics

SVG or scalable vector graphics is a royalty free graphics format developed and maintained by W3C SVG group (SVG, 2017). It has support for both animation and interactivity, since it is based on graph theory and mathematics. Being based on numbers makes vector logos and scalable graphics very lightweight, as compared to raster image formats such as jpg and png.

### 4.1.11 D3.js

D3.js or "data driven documents" is a Javascript library for manipulating documents based on data (D3.js, 2017). It utilizes a number of technologies that have already been described such as HTML, CSS and SVG leading to powerful visualization. This library was used to create the version tree in the ECHO system.

### 4.1.12 Trello

Trello is a flexible and customizable web-based management system, often used for keeping track of software development (Trello, 2017). The user can create columns like "ongoing", "under testing" or "done" to customize different stages in the development of a system. Cards can be created and put under a backlog for example. The cards can consist of user stories or functionalities that are ready to be implemented into a system. Kanban was used in junction with Trello to keep track of features that was being developed and finished.

## 4.2 Requirements

Before the development of any prototype can be made, requirements must be acquired. Two different types of requirements have traditionally been identified in software development: functional requirements, which describe what should be done and non-functional which describe good to have features that can make the system even better (Preece, Sharp, and Rogers, 2015). An example of a functional requirement could be that an application should have below five seconds response time when loading content. In contrast, non-functional requirement could be showing the user the system status of the content-loading by using a progress bar.

Requirements for the prototype were assembled after the literature search, and after conversations with peers, fellow artists and fans. This resulted in two different scenarios.

### 4.2.1 Scenarios

A scenario is a narrative description about human activity in relation to a system in the form of small stories. It does not explain the exact functionality of the software that the scenario is describing, but it is rather a story that will keep stakeholders, or peer artists in this case, understand the use of the system (Preece, Sharp, and Rogers, 2015).

The following two scenarios were generated from discussions between fellow artist friends and myself, in order to set the basis for functional and non-functional requirements.

#### Scenario 1

With reference to Figure 4.1, scenario 1 is the interaction between the artist and fans or listeners. The artist manages and organizes his or hers music in tree-alike structure and allow the listeners to review the different versions of the music. Feedback is received and displayed inside the music management system of the artist where he or she can evaluate music based on comments and/or ratings. That helps to decide on what version to further explore. The feedback and evaluation from fans should potentially help the artist make better choices and be more creative.

Following is a story example for scenario 1:

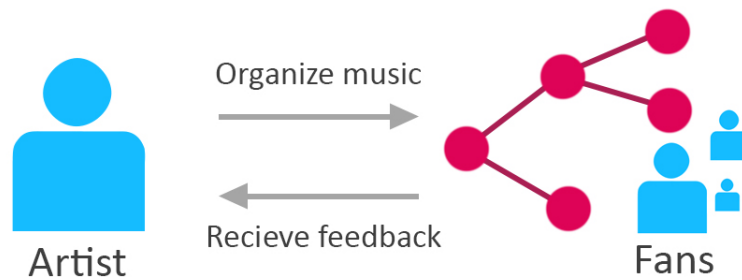


FIGURE 4.1: Scenario 1, Interaction from artist to fans

As an artist, I have a number of rendered mp3 files of musical projects that is just sitting unorganized in folders on my computer. I want to organize them somehow and get some feedback on which of the versions I should focus on. I want to be able to upload them and organize them in a tree-alike structure, add and subtract information about the tracks such as title, descriptions, notes or tags so that they can be organized and visualized in different ways. Once the files have been organized, I want to be able to present my versions to my fan base and get feedback in the form of comments, ratings or both.

### Scenario 2

In scenario 2 illustrated in Figure 4.2 is the interaction as a fan or listener. The listener should be able to enter the system and be able to find music based on preference that is open for review. The fans should be able to give feedback to the artist after the listening experience if they wish to do.



FIGURE 4.2: Scenario 2, Interaction from fans to artist

Following is a story example for scenario 2:

As a listener and fan of artist X, I want to be able to log into the system and quickly discover unreleased music. I want to be able to filter and sort music based on my specific genre preference. When discovering music that looks interesting, I would want to play the track and listen, rate and comment on the

track.

With these two scenarios in mind and with Shneidermans et al's principles for creativity support systems, I was able to create the following functional and non-functional requirements for the ECHO system.

## 4.2.2 Functional requirements

Before the design of the prototype was formalized, functional requirements had to be specified, following is a list of functional requirements generated from the scenarios and discussions about the system.

1. Fans and artists should be able to create user accounts and login to the system
2. User data will be stored in a safe database
3. A system should be in place so that the artist can upload content to the system to be organized and evaluated
4. The artist should be able to generate a hierarchical tree structure of versions of his or her songs for visualization
5. The system should have ways to identify versions easily and the projects they are attached to
6. The system should give the user freedom to use the system however he or she wants it
7. Fans should be able to easily navigate and discover new music
8. Managing of the versions should be problem-free and easy
9. If the artist desire to remove versions or projects he or she should be able to do so without much trouble
10. The system should have a solid media player for playback of music

## 4.2.3 Non-Functional Requirements

Non-functional requirements was important to the system since they could specify not necessary but good features of the system.

1. The system should be developed and run on a web server for browser and web use
2. The system should be relatively easy to use and potentially have a high ceiling of what can be done with it
3. The system should support exploration of both ideas and of functionalities within the system



4. The system should support many paths and styles, for example the users should be able to customize the visual appearance of their application
5. The system should be able to support collaboration between users
6. The system must do what the users expect it to do, and more
7. The system should facilitate the generation of more creative ideas from a structuralist perspective

These functional and non-functional requirements acted as the foundation of the prototype development.

## 4.3 Iterations

The application was created through the development of three different iterations. In the design iteration, the goal was to design the structure of the application. Following the design iteration was iteration 1, as the biggest of the iterations, the majority of the base code was created before it was evaluated by a focus group. In iteration 2, the review system was implemented into the ECHO system. Although review functionality of the system was not completely finalized according to the requirements, it was still functional on a basic level. After the second iteration the application was evaluated using a combination of user testing, expert interviews and SUS questionnaire evaluation.

### 4.3.1 Design Iteration

The design iteration started with the initial design sketches of the system. Figure 4.3 was the first basic outline of how the system could look-alike. With a central tree structure for keeping track of the versions, and a left side panel for display of other projects the artist has created.

This sketch also included a player at the bottom, similar to how the applications Soundcloud and Allihoopa have. This fixed music player idea proved later to be a technical challenge and was discarded. Instead a local music player was implemented into the version boxes.

Figure 4.4 illustrate the high level prototype sketch that was initially created before the development started. It was meant to show how the system could look like when finished, but it ended up being a direct inspiration for how the graphical user interface. In this second sketch my fascination with monochromatic color schemes started to show, some important buttons and highlights

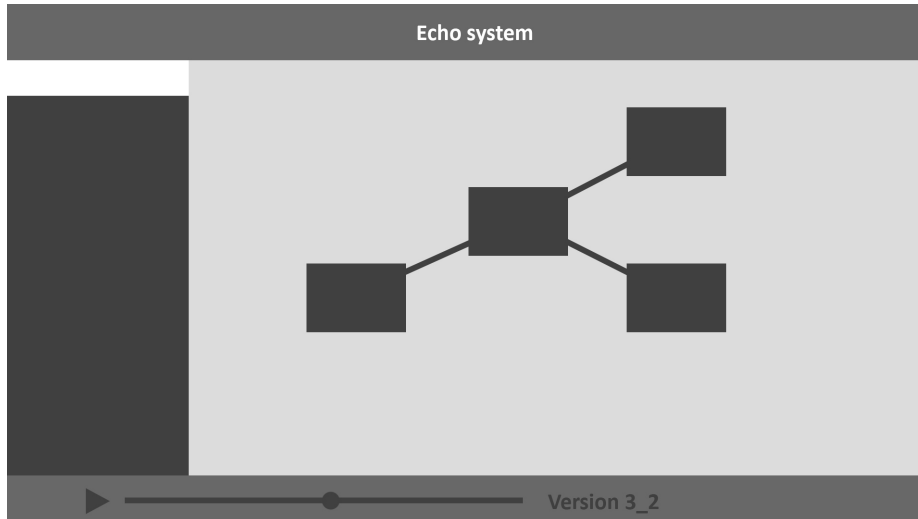


FIGURE 4.3: First sketch of the ECHO system

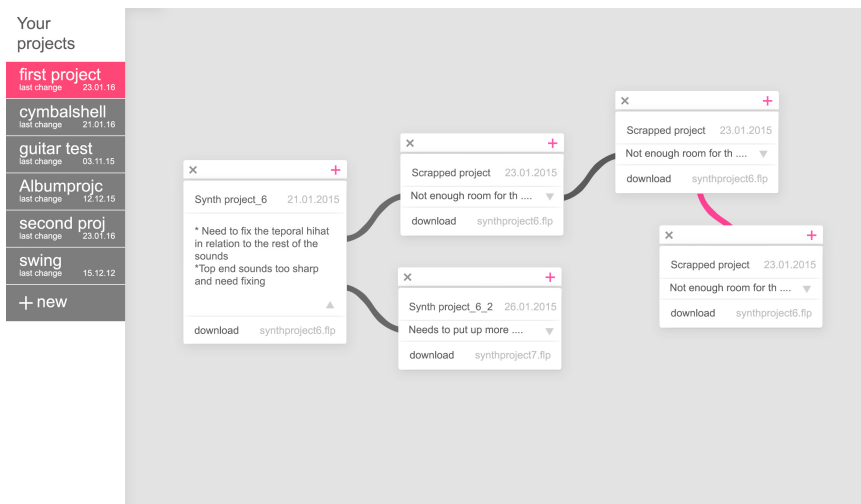


FIGURE 4.4: Second higher-level sketch of the ECHO system

were marked with a purple/redish color.

A high level BPMN (OMG, 2017) diagram was created to illustrate the flow of information in the system. BPMN or Business Process Model and Notation, is a graphical notation often used to visualize the different processes that go into a information system. It is managed and developed by the object management group (OMG). The reason for choosing this diagram, is that it is relatively easy to understand the different overall layers and processes of a system.

### 4.3.2 Iteration 1

Iteration 1 started out by generating the Kanban backlog based on the functional and non-functional requirements see Figure 4.6. Kanban cards such as

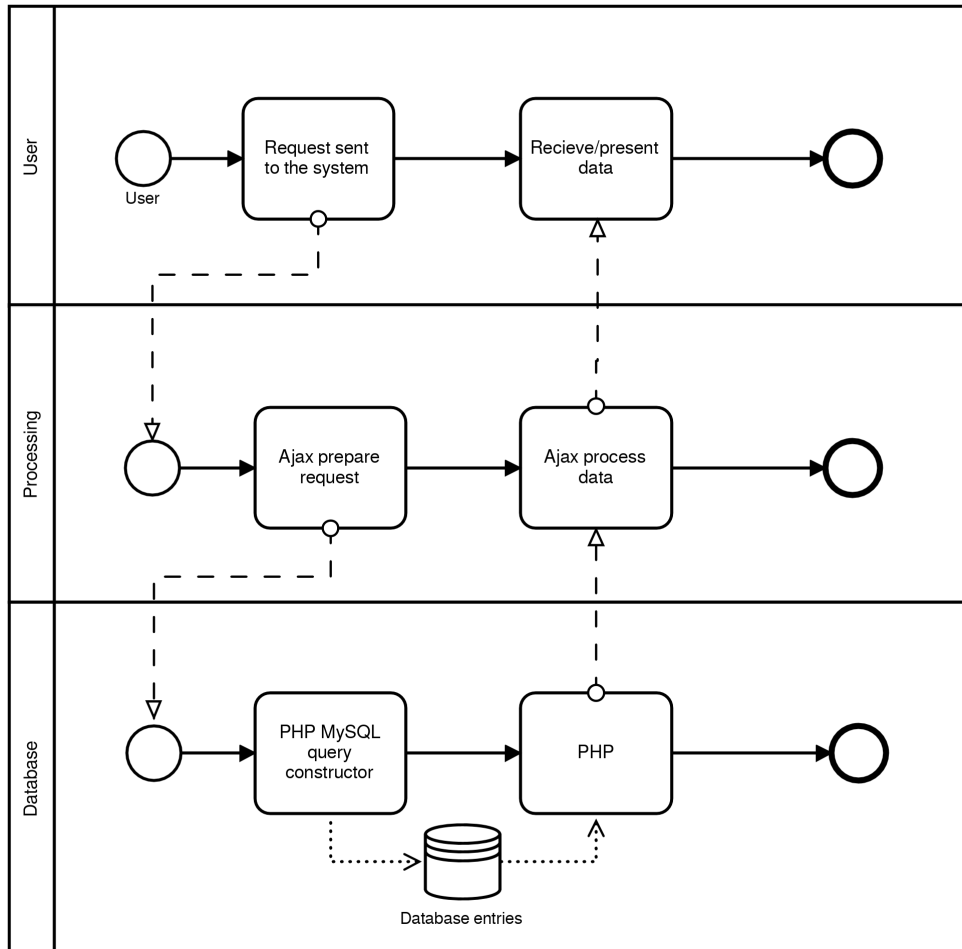


FIGURE 4.5: High level BPMN overview of the system

login, create user, upload are some of the basic functionality that were added.

The system was developed with the intention of starting with the natural entry point of a user, and then developing the successive parts of the system as they become needed. For example, the user creation and login system was created first, then when the user entered the system, simple things like changing the color theme and creating a basic project was developed. This way of development felt natural, developing the software in this way has ensured that there was a solid foundation that would allow the development of new functionalities.

### User Registration and Login

One of the functional requirements is that the user data should be stored safely (see Section 4.2.2). To manage user data, a combination of three different technologies was used: PHP (see Section 4.1.5), WAMP (see Section 4.1.1) and

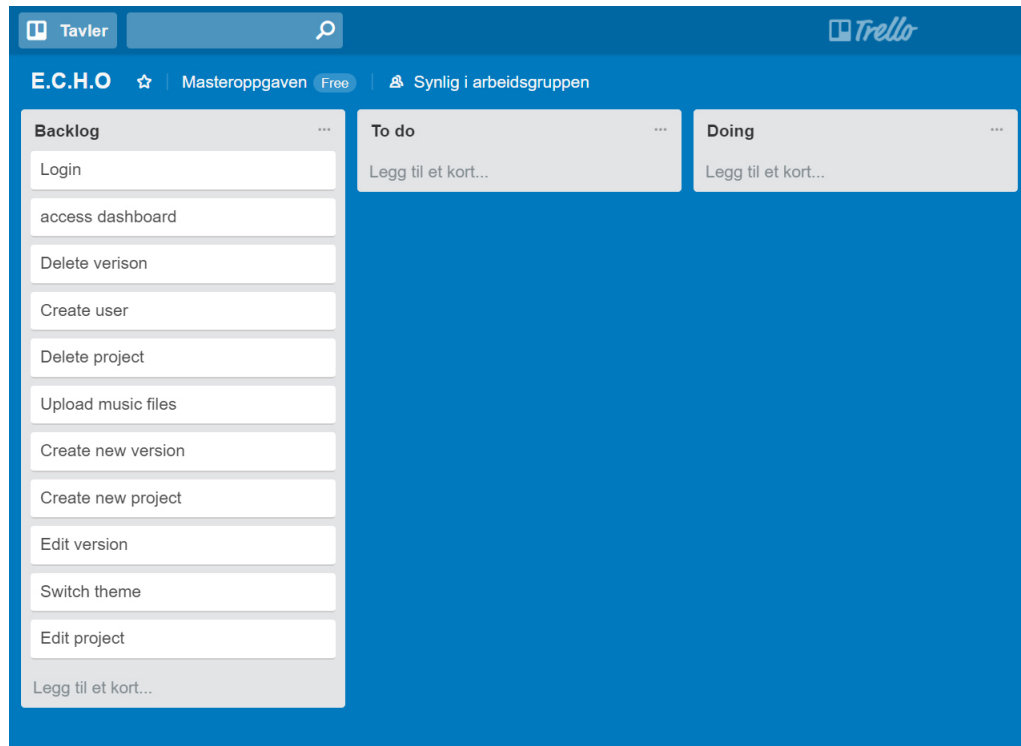


FIGURE 4.6: Initial kanban board with functionalities as cards

MySQL (see Section 4.1.6).

Following is a numbered list over how user interaction could happen during login or user creation:

1. The artist or listener creates an account
2. In the database the password and user name is stored
3. When the user attempts to login, the password is then checked against the stored password
4. Given that the password the user has provided matches the stored password, the user will be able to login into the system
5. If the password is not correct, the user will be taken back to step number three

When the handling of users and login had been completed, the next step was to begin developing the internal platform where the user would be able to create projects and versions. A separate HTML document was implemented as the scaffolding that the application was going to run on.

## Audio Files

Audio file upload and file management were mandatory functionalities necessary for the completion for the prototype. This required a separate HTML form implementation for handling the upload of multiple audio files (see Figure 4.7). File size restrictions for the audio uploads had to be added in the system, a reasonable size of 8 megabytes were set in the PHP.ini file on the server, and a restriction for only mp3 files were set in the PHP script, responsible for handling the processing of uploads. Uploaded files are connected to the project once the user has uploaded an audio file. The responsibility of naming the audio files lies with the users, so that they can remember them easier.

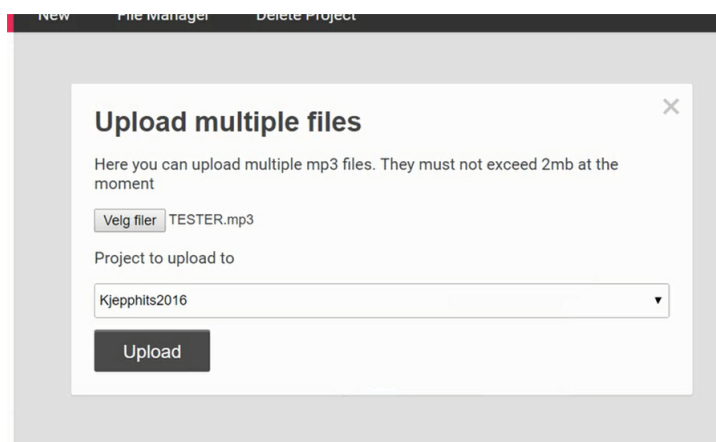


FIGURE 4.7: Upload form in the ECHO system

## Project and Version Creation

In the echo system a user can have many projects, and the projects themselves can have many versions. One of the functional requirements is making it easy to identify versions and what projects they are attached to, for that purpose two separate functionalities were developed, version and project creation. The act of creating a new project is simple, after the create project button has been triggered, the code responsible for processing (see Figure 4.5) fetches the HTML form responsible for project creation and displays it to the user see Figure 4.8. A title and description of the musical project is required before the form can be submitted. Restrictions were set on how many characters the title and description can have, 50 for the title and 250 for the description.

The process is similar when creating a new version, version title and description is required and has the same length as the project creation form. However the user has the option to choose what the parent version is, when creating

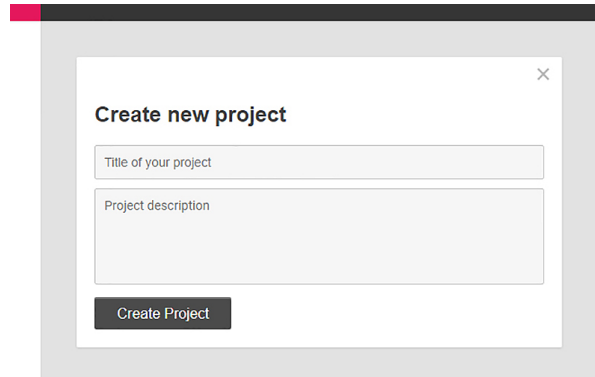


FIGURE 4.8: The form responsible for submitting title and description of a project

a new version, so that the structure of the version tree can be built. Adding audio files to the version is also an option when creating a version. Audio files that have been uploaded appear in a dropdown menu in the form. The submission of both versions and projects gets processed by the combination of the Ajax handlers, PHP and the database.

### Version tree

The version tree was one of the most important core functionalities of the ECHO system, which meant that it was important to make it visually appealing. Different solutions were reviewed, and after some time the goal of making the tree from scratch proved to be a tough problem to tackle. Due to the technical difficulties of creating such a tree, the D3.js (see Section 4.1.11) library for creating modular graphs with Javascript was adopted.

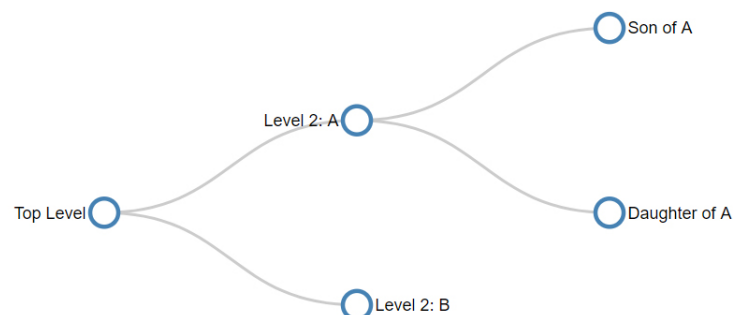


FIGURE 4.9: D3.js example tree structure, retrieved from <http://bl.ocks.org/d3noob/8375092> (accessed 2017-20-11)

The version tree was created by heavily modifying an example code from the D3.js Javascript library (see Figure 4.9). The D3.js library was found to solve most of the problems associated with creating a dynamic hierarchical structured tree from ground up. However the structuring of the versions and version tree, ended up looking differently to the initial design sketch due to limitations in the d3.js library. Initially the idea was to have version boxes directly coupled together, but instead these were replaced with clickable nodes. The version boxes were moved into a separate box under the version tree overview. If a user clicked one of the nodes, the corresponding version would load into the box. An example of how a fleshed-out version tree could look like is illustrated in Figure 5.13 in Section 5.1.5.

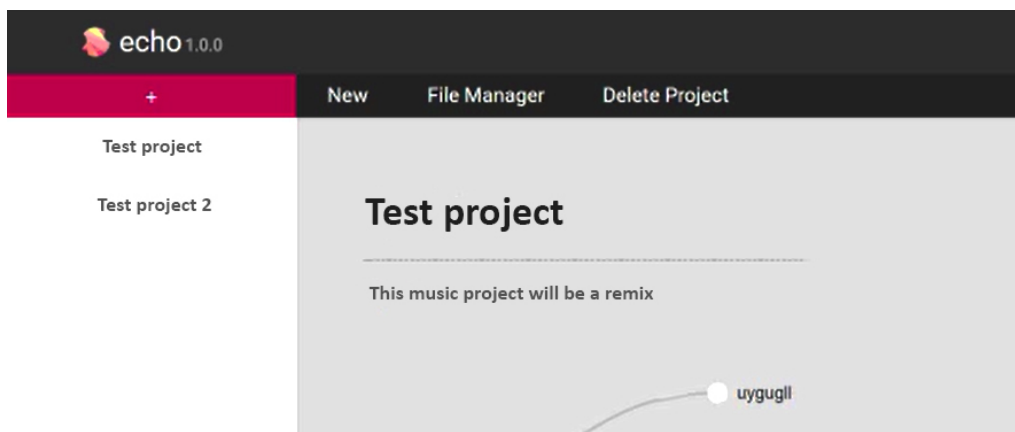


FIGURE 4.10: ECHO menu layout consisting of four buttons

Figure 4.10 show the structuring of the menu, after finishing the first iteration. The left side panel consists of a list of projects that the user currently has, and by clicking them they can be loaded into the context box in the middle of the page. There are three buttons on the top of the system: a red "+", "New", "File Manager" and "Delete Project". When the user wanted to create a new project, he or she would have to click the button with the "+" icon in the top left corner. Clicking the new button would bring up the form for creating a new version, and that would also be reflected in the version tree for that project. File Manager button is where the uploading and management of audio files were held. Delete project would simply delete the whole project.

### Color and contrast

A clear color scheme was chosen for the system, various deep shades of crimson pink and violet, as well as a combination of dark gray and lighter gray

was chosen see Figure 4.11. The goal was to keep the contrast high in most cases, as this would help to increase visibility of text. Dark and deep colored backgrounds have white text on them, and where there is light background, a darker text color is chosen. Most of the important text was kept close to black in color, using the color code "282828" seen in Figure 4.11.



FIGURE 4.11: Color codes used through the ECHO system

The choice of implementing the color theme checkbox into the system relates back to the non-functional requirement based on Schneidermans principle number two: Support many paths and many styles. Two contrasting themes were developed and implemented via a slider that gave the user a choice of either having a darker theme or a lighter theme. Adding some level of customization of the visualization of the tool was suggested by peers in the discussion of the idea. In Figure 4.12 the dark theme check box is shown for off and on position. A simple HTML check box was implemented, and using CSS a style was applied to make it look like a slider. When the dark theme is on, the check box has a dark purple/red color indicating that the dark theme is active.

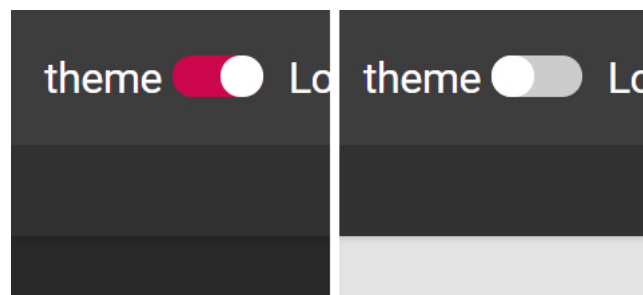


FIGURE 4.12: Change theme checkbox in both on and in off position



## Focus Group

When the prototype 1 was completed the next step was to introduce the system to a focus group for testing and generating more ideas regarding functionality. The focus group was also utilized to find potential problems with the system. A list of suggestions for features and problems was generated after the users had tested the system. More detail on how the focus group test and group interview were performed can be read in Section 6.1.

### 4.3.3 Iteration 2

After the focus group of evaluation, a number of improvements were done to the base code of the system. Menu items that were related to handling of the project, was moved to the left side panel below each project, and the buttons responsible for controlling the individual versions were moved to the version boxes.

### 4.3.4 Restructuring of the menu

In Figure 4.10 we looked at how the different menu items were structured, when the focus group concluded some structural problems became apparent. The naming scheme of the buttons was sometimes hard for the focus group participants to understand, having the simple "+" icon on the button responsible for creating a new project was confusing for two of the participants.

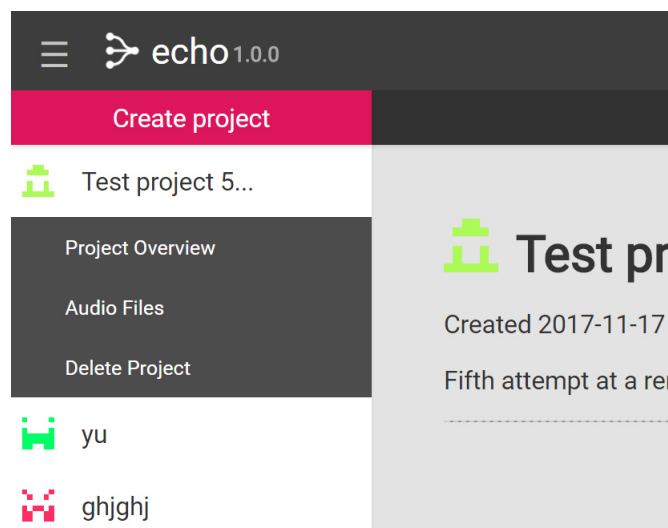


FIGURE 4.13: The revised placement of menu buttons

The restructuring of the buttons are displayed in Figure 4.13. In the new layout, menu buttons were moved to a drop down list of items below the active

project. The create new project button was also changed from a "+" sign to "Create project" as requested by the users. "File Manager" menu item was changed to "Audio Files" to remove potential confusing in regards to what kind of files could be uploaded.

### 4.3.5 Review System

When the core functionality of the system was finalized, the addition of the review system was implemented. Since one of the requirements of Shneiderman's principles is to support collaboration (see Section 2.3.4) this would be reflected in the collaboration between fans and artists. The review system's core functionality was to handle the processing and dissemination of feedback between the fans and users of the system and artists. A new panel on the right side of the application was created, options of adding versions and setting duration for the review were implemented into this part of the system.

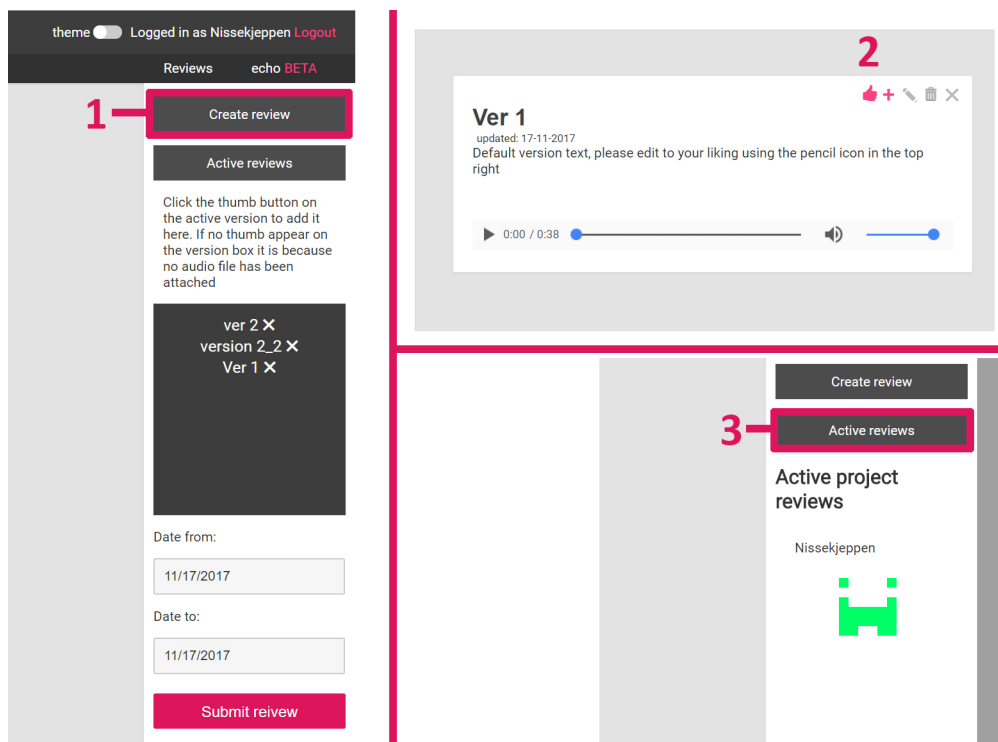


FIGURE 4.14: The review system

In Figure 4.14 and sub Figure number 1, the button "Create review" when clicked include the HTML form responsible for the submission of the review. Sub figure 2 in Figure 4.14 shows how a new thumb button appears when "Create review" is clicked. Once the review has been submitted it appears as a box below the two review menu buttons. Clicking "Active reviews" will display

the active review, as shown in sub Figure 3.

#### **4.3.6 Final User Evaluation**

Due to time restraints the prototype was deemed finished after the review section was completed. Ideally the project should have had at least one more iteration. Evaluation of iteration 2 was performed with a multitude of different techniques, user testing, expert interviews was performed along with the System Usability Scale questionnaire. The combinations of these techniques resulted in a thorough list of feedback acquired from the transcribed interviews and from the participant observation. More about the evaluation can be read in Section 6.2.



## Chapter 5

# Results

In this chapter, the resulting prototype of the system is presented. Important features and sections of the system is presented and explained from the point of view of developer and user.

### 5.1 ECHO Prototype

The resulting ECHO system prototype is a web application. Aimed at artists to organize their musical ideas in a tree-alike structure. Consisting of two main parts: a tree graph overview of individual music files, and a review system to generate feedback. Figure 5.1 shows an example from an artists point of view. The left side panel has a number of projects listed, each with individual generated icons. The center of the application contains information of the project currently open, as well as the version tree and version box connected to that project.

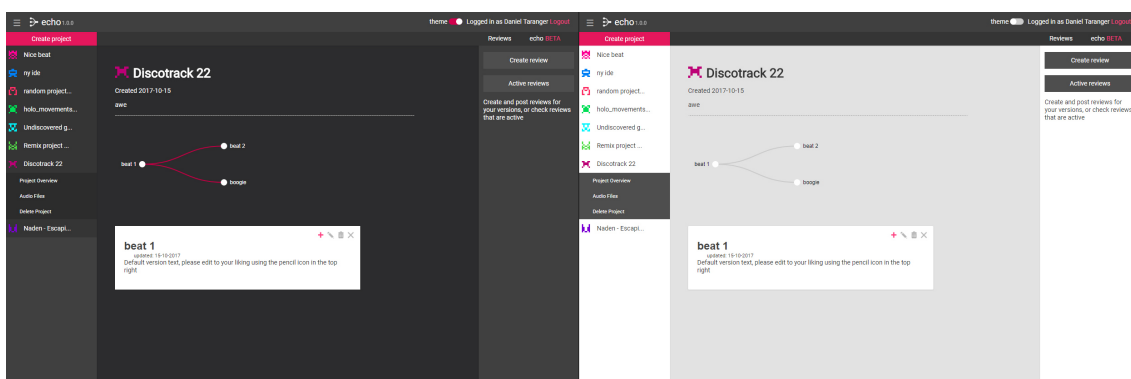


FIGURE 5.1: The ECHO system, two different color schemes

A number of functionalities were implemented and are present in the final version of the prototype. following is a list of functionalities included currently in the prototype:

1. Register and create new user

2. Project creation, description and naming
3. Add versions to the project
4. A hierarchical tree for representing the versions
5. Editing and deletion of project and versions
6. The option to upload mp3 files to the application and attach them to the version tree
7. A review system where the artist can mark versions for review to get feedback
8. A dashboard where other users can access projects that are up for review

### 5.1.1 User Creation and Login

The login page of the system is presented very minimalistically and clearly to the user (see illustration on the left in Figure 5.2). If the user tries to access parts of the system while not logged in, he or she will automatically be sent to this login page. Below the login fields is a link to the register fields, as are presented in the illustration on the right in Figure 5.2. Email registration and validation, while not hard to implement, would have taken significant time to complete. For the purpose of the prototype this simple login and registration functionality seem to be sufficient.

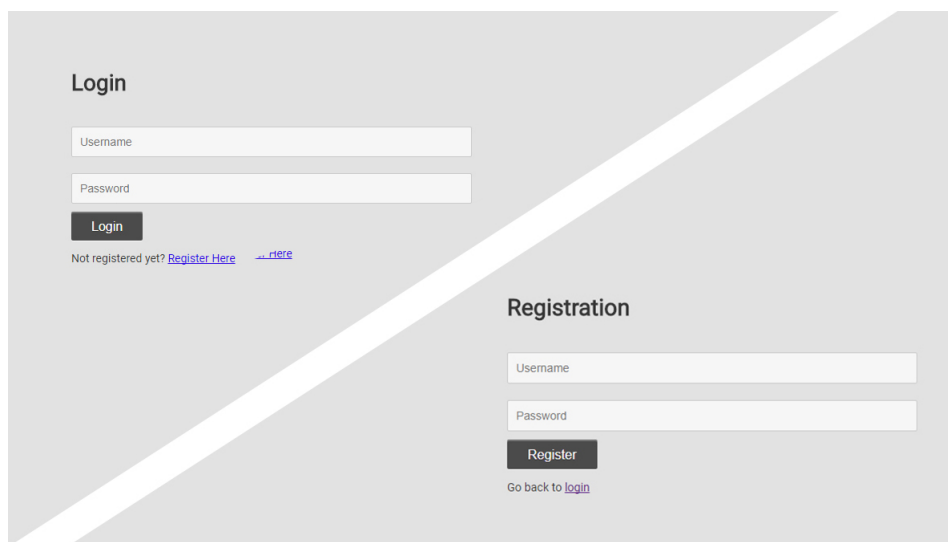


FIGURE 5.2: Login screen screen on the left, and registration screen on the right

The system uses a simple session storage to remember users after they have logged in, often recognized by the "remember me" popup that sometimes appears when logging into systems. User data is stored in a MySQL database table called "users", an unique id is generated for each user when they first

register. The id is automatically incremented and kept track of in the database. Even if a user is deleted, a new user will never have the same id as the deleted user, thus preventing potential bugs in the system.

### 5.1.2 Project Creation

When the user first enters the system, the window presented is mostly empty besides some panels and one single "Create project" button in the top left corner. Instead of keeping a heavy documentation inside the system, hiding and showing relevant functionality at different phases of a project is used. In Figure 5.3 is an example of what the user sees when entering the system. The user is encouraged to click the "Create project" button on the top left by the text that is being displayed.

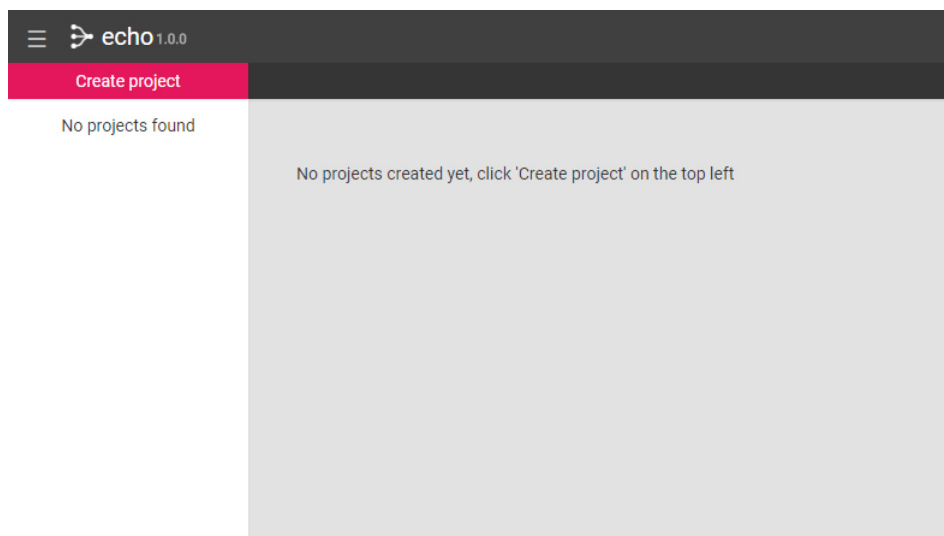
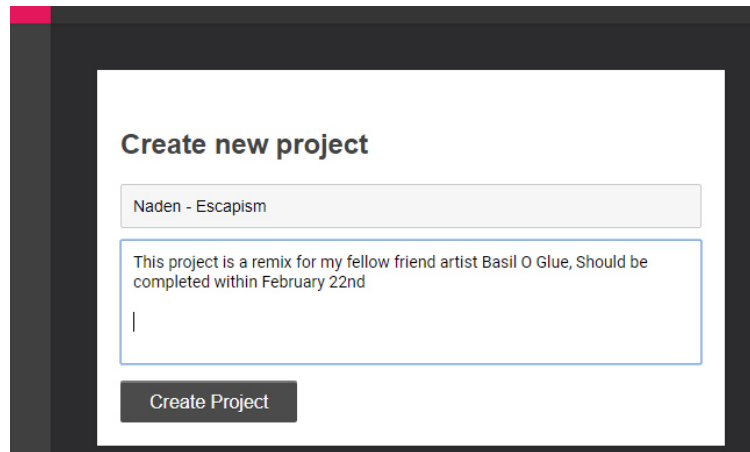


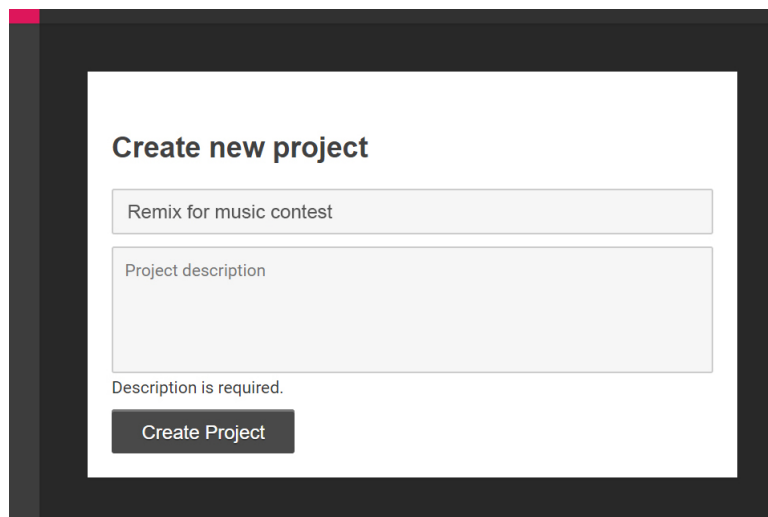
FIGURE 5.3: Example of the screen the user sees while entering the system

When the button is clicked, a window appears that prompts the user to write a title and description about a project (see Figure 5.4). This window is a HTML form that once submitted, sends an Ajax post to the server which then stores the information. The title input field has a limit of 50 characters. After reaching this limit, the user is unable to type more letters into the input field. This restraint ensures that the data submitted by the user is in line with the character limit registered in the database. If the user forgets to fill out a required field such as title or description, an error message is displayed under the appropriate HTML form input field. This helps to prevent confusion during the lifespan of the music project (see Figure 5.5).



The screenshot shows a web form titled "Create new project". It features a text input field containing "Naden - Escapism". Below it is a larger text area containing the text "This project is a remix for my fellow friend artist Basil O Glue, Should be completed within February 22nd". At the bottom of the form is a dark grey button labeled "Create Project".

FIGURE 5.4: Creating a new project, example input fields



The screenshot shows the same "Create new project" form. The text input field now contains "Remix for music contest". The larger text area is empty. Below the text area, the text "Description is required." is displayed in a smaller font. The "Create Project" button remains at the bottom.

FIGURE 5.5: Error message, if the user forgets to fill one of the required fields

After the project is submitted, it will open and fill the content of the page. Unique icons are generated for projects (see Figure 5.7). By generating unique icons it should be easier to differentiate between projects. It is not currently possible to change an icon after it has been generated. A default version is created, represented as a dot in Figure 5.6. The user is able to edit the posted version and add music files to it if she or he wants to do so. Just created projects appear in the left side project panel, and a sub menu has opened (see Figure 5.6).



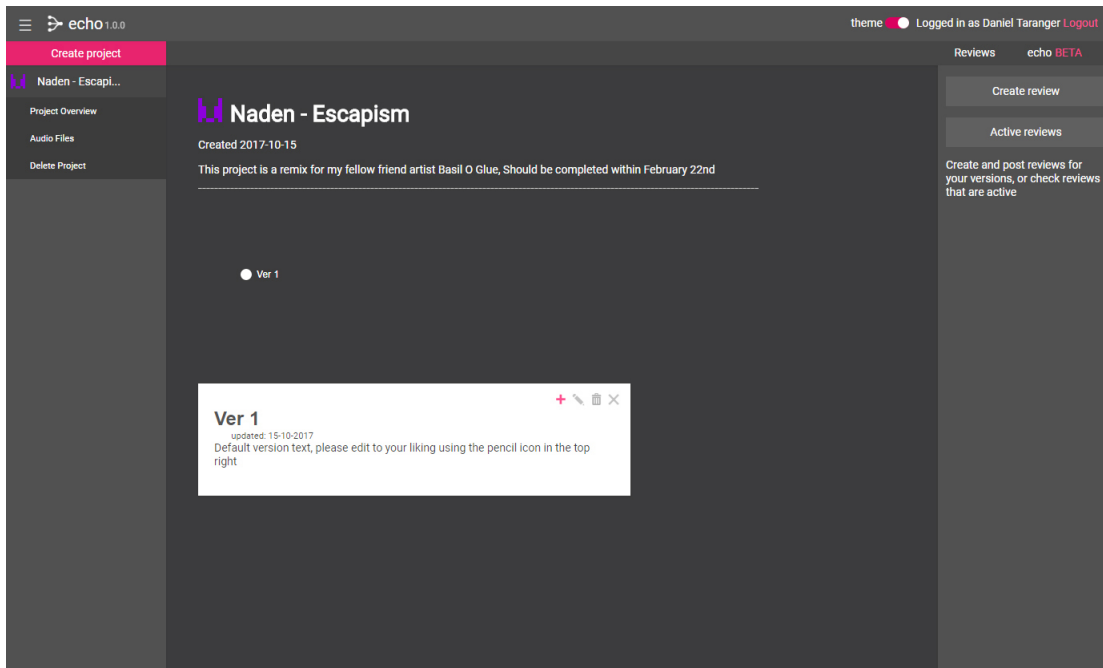


FIGURE 5.6: The application state after creating a project with dark theme enabled

### 5.1.3 Uploading Audio Files

In the ECHO system the user is encouraged to upload files, and attach them to versions, so that they can be reviewed. If the user wants to upload and uses mp3 files, the second project menu item "Audio Files" can be located on the opened project on the left side panel. After clicking it, a new box is opened and a standard upload prompt is displayed. The user selects files he or she wants to upload and clicks "upload" (see Figure 5.8).

Once the files have been uploaded, they appear on the same page they were uploaded from. The user can delete files or download them from this page (see Figure 5.9).

### 5.1.4 Version Tree

After the project has been created, and different audio file versions of a track have been uploaded to the project, the user can begin to build the version tree. Clicking the pencil icon on the version box (see Figure 5.10) will open the edit HTML form for the version, where the user can edit the title, description and attach a audio file (see Figure 5.11).

By clicking the "+" icon illustrated in Figure 5.11 a create new version box is opened, and the user can now type in the version title and description. A

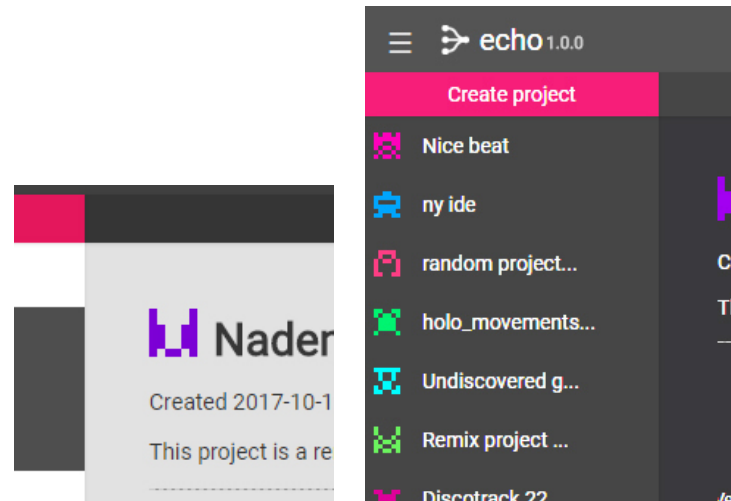


FIGURE 5.7: Unique icons generated for projects

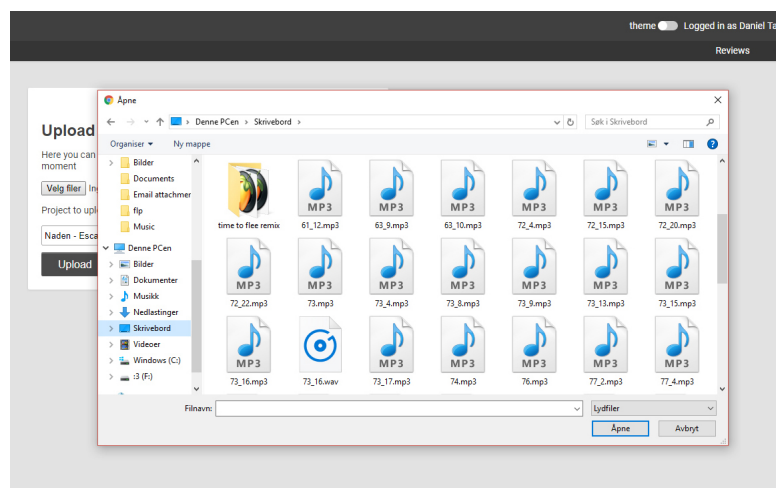


FIGURE 5.8: Uploading audio files

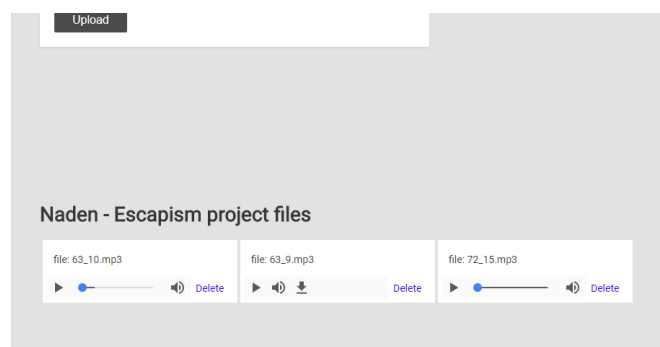


FIGURE 5.9: Audio files as they appear after uploading

dropdown menu also appears, giving the option of choosing the right parent version to start with. An audio file can be selected from those that have been uploaded to the project. In addition to the editing functionality, it is now possible for the version to be put up for review.

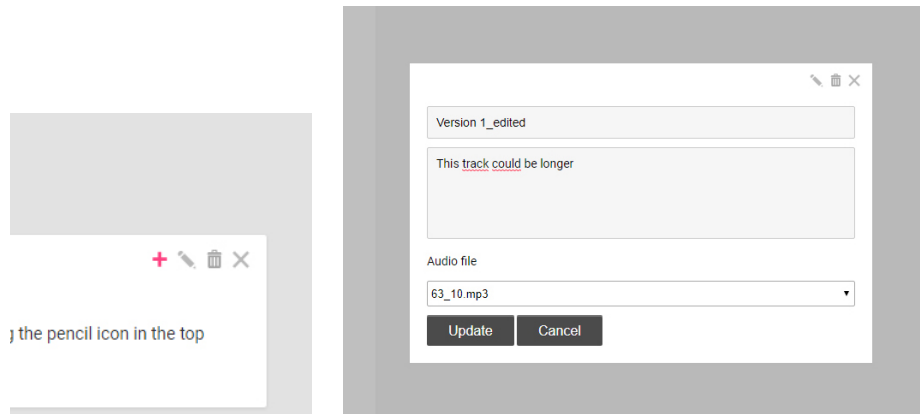


FIGURE 5.10: Version buttons on the left and edit HTML form on the right

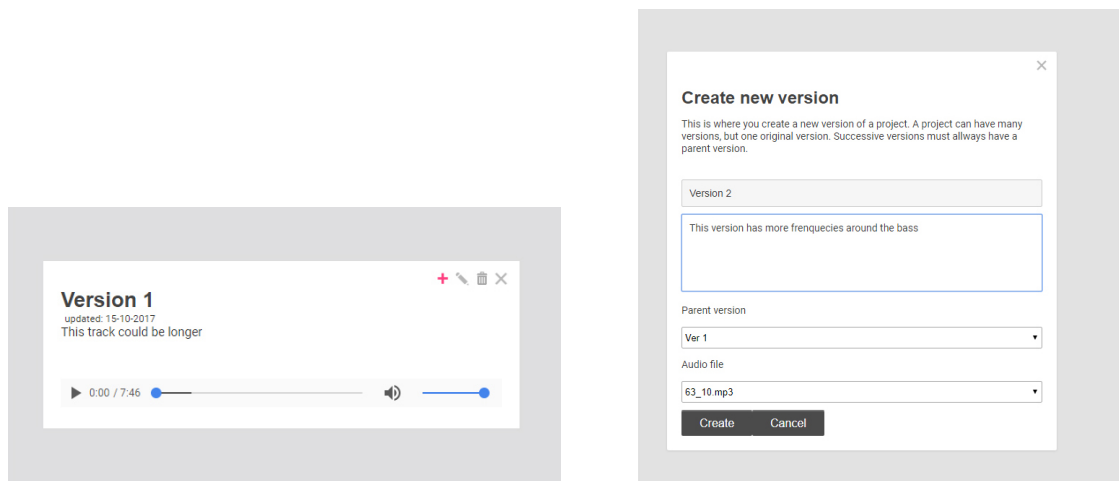


FIGURE 5.11: Version box on the left and create new version HTML form on the right

### 5.1.5 Review Functionality

At the end of the prototype development, extra functionality in regards to getting feedback was added to the system. The user could mark the version he or she wanted to be reviewed, submit the review, and other users with access to the system could rate and give comments about the music. Looking at Figure 5.12, new parts of the system have been revealed after the user has clicked the "Create review" in the top right corner. The right panel contains versions that are marked for review, which is also indicated by a red thumb button on the version box. Clicking the red thumb button adds or removes the version to the review panel. Each version displayed in the review container has a "x" button, that can remove the version from the review panel.

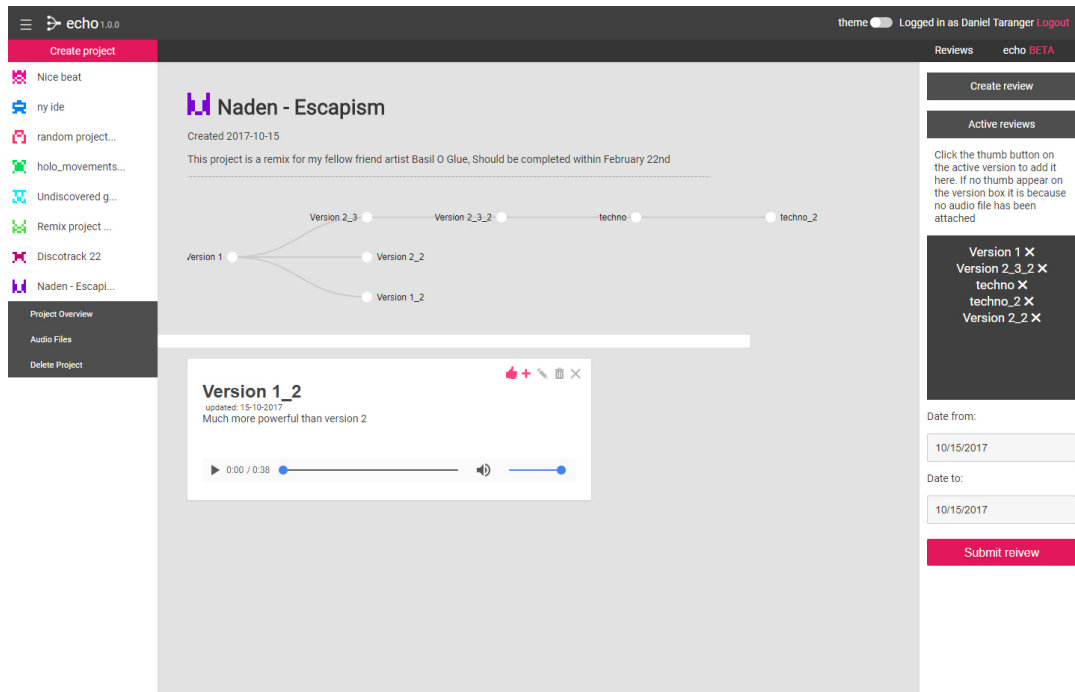


FIGURE 5.12: Active review system example

Once the versions have been selected for review, the user will move on to selecting start and end date of the review. The datepicker can be seen in Figure 5.13.

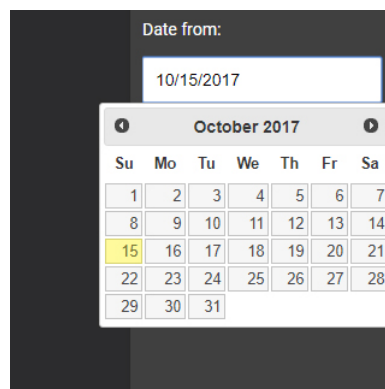


FIGURE 5.13: Date picker in the review panel

The review could be submitted after the user has chosen the versions for which he or she wants to get feedback on. Clicking the submit button will send this information through the Ajax handlers, to the PHP script and then the database. When the review has been submitted, it is revealed at the dashboard of the site (see Figure 5.14).

Giving feedback to a review is as simple as clicking one of the active review boxes on the dashboard (see Figure 5.14), the active versions will appear listed

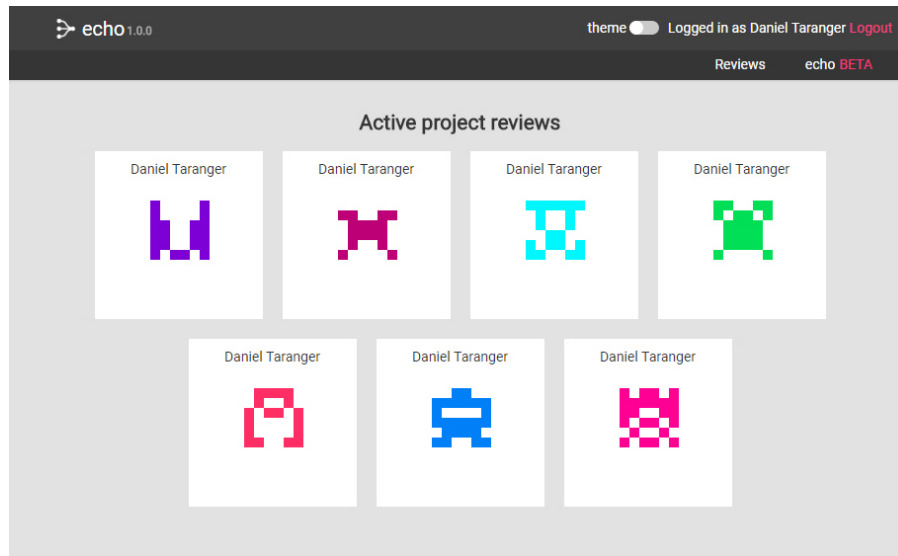


FIGURE 5.14: Review dashboard, where other users excluding the artist can rate and comment

in an anonymous order. Figure 5.15 on the left shows how it will look for the user when one of the active reviews are clicked. By default all versions that are reviewed appear in a random order. The names of the audio files are also removed in order to keep it anonymous. Previous comments and ratings from other users are hidden to prevent the listener to be influenced by other listeners.

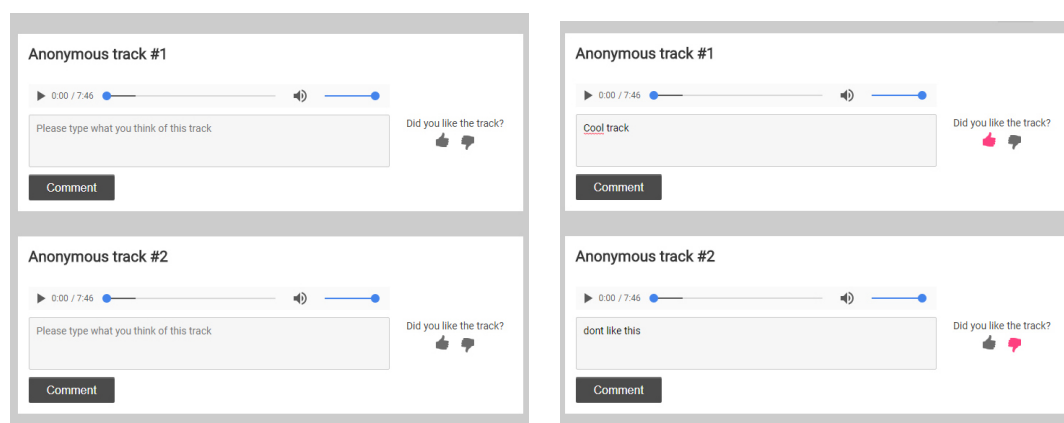


FIGURE 5.15: The figure on the left shows versions ready to receive feedback, the figure on the right shows commenting and rating of the version

On the right side in Figure 5.15 shows how the up and down thumb buttons are highlighted when they have been clicked. Rating of the version is processed separately from the comments, they are sent directly to the database once they

are clicked. Comments are submitted via HTML forms through Ajax when the user clicks the "Comment" button, and each user can only comment one version at a time, thus preventing overloading the database.

### 5.1.6 Review Feedback Functionality

After the users have given feedback to the versions, and the review has been running for the required time set by the artist, the version boxes visible to the artist should have a rating and attached comments. All of the ratings given by the users will be calculated and displayed as percentage (see Figure 5.16). This form of displaying a thumb up and down rating is used by many websites, including Youtube (Youtube, 2017).

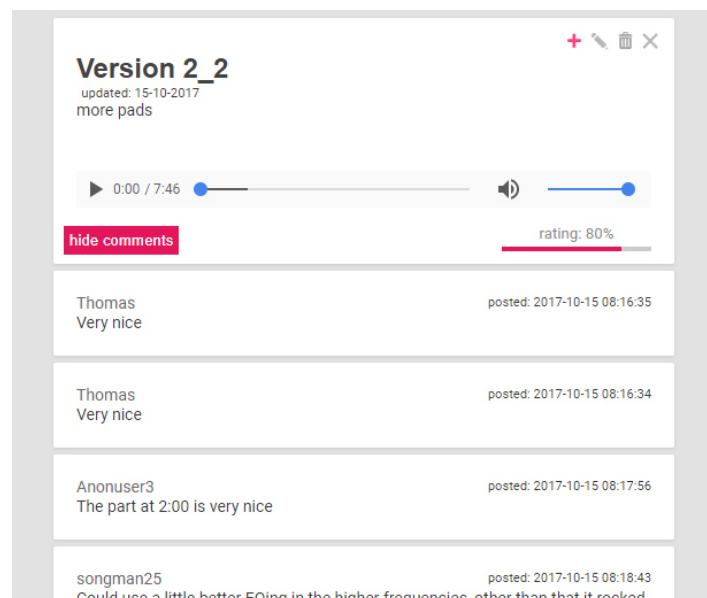


FIGURE 5.16: Example of feedback in the version box

### Calculating Version Scores

Votes from all users are stored in a MySQL database table called `review_ratings` represented by "1" and "0", "1" meaning thumbs up, and "0" meaning thumbs down. The score on each version is then calculated by adding up all of the ratings of that particular version and then divided by the total number of votes for the same version. For example if a version has 6 votes: 1,0,1,0,1,1 then the final score would be  $1+0+1+0+1+1 = 4$ , it is then divided by the total number of all 6 votes, which is results in 0,66 or 66 percent approval.

## 5.2 MySQL Database Tables

In this section the important data tables will be listed and explained. Some explanations of fields are kept minimal, for example fields such as ID's, user-names and dates are explanatory.

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
id 	int(11)			Nei	Ingen		AUTO_INCREMENT
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
password	varchar(50)	latin1_swedish_ci		Nei	Ingen		
trn_date	datetime			Nei	Ingen		

FIGURE 5.17: Data fields for the user table in phpMyAdmin


Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
id 	int(55)			Nei	Ingen		AUTO_INCREMENT
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
theme	varchar(50)	latin1_swedish_ci		Nei	Ingen		
last_project	int(25)			Nei	Ingen		
pre_last_project	int(25)			Nei	Ingen		

FIGURE 5.18: Data fields for the user settings table in phpMyAdmin

Figure 5.17 shows the structure of the users table, with an unique auto incremented id, username and password with a limit of 50 characters. It also stores the registration date of the user in trn\_date. This is the table responsible for storing all the users and their passwords.

Figure 5.18 is the user\_settings table. This table contains some settings and data related to the users of the ECHO system. When the user changes the color theme of the system, the theme field reads "on" or "off". Last\_project is used when loading a previous active project the user was working on. Pre\_last\_project is used when the last project the user was working on is not found or is deleted. Figure 5.19 gives the structure of the table with all the projects. When a user creates a project, all relevant information will be stored in this table. The title of the project and description is filled in by the user. The rest of the fields are generated through other systems. ProjectID is the unique primary key of the table and is auto incremented, this is important to ensure the data consistency. The last\_version field is used to store and load the last version that was opened in the version tree. This should not be confused with last\_project in the user\_settings table. Date field is simply a date of the creation of the project.

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
projectID	int(11)			Nei	Ingen		AUTO_INCREMENT
title	varchar(100)	latin1_swedish_ci		Nei	Ingen		
description	varchar(300)	latin1_swedish_ci		Nei	Ingen		
date	datetime			Nei	CURRENT_TIMESTAMP		
last_version	int(11)			Nei	Ingen		

FIGURE 5.19: Data fields for the project table in phpMyAdmin

Navn	Type	Sammenligning	Attributter	Null	Standard	K
projectID	int(11)			Nei	Ingen	
colorBits	varchar(150)	latin1_swedish_ci		Nei	Ingen	
color	varchar(6)	latin1_swedish_ci		Nei	Ingen	
username	varchar(50)	latin1_swedish_ci		Nei	Ingen	

FIGURE 5.20: Data fields for the project icons table in phpMyAdmin

Project\_icons table is responsible for storing the data generated for the unique project icons, illustrated in Figure 5.20. The colorBits field is the most interesting one in the table. It stores a random string of 1's and 0's generated by a PHP script which is responsible for how the project icons appear (see Figure 5.14). Each number means that a block in a grid is either colored or transparent. Only the right half of the icon is generated through this random string, and then mirrored onto the left side of the icon. The resulting icon is very lightweight to store, since it is just a simple string of numbers.

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
projectID	int(25)			Nei	Ingen		
versionID	int(25)			Nei	Ingen		AUTO_INCREMENT
title	varchar(25)	latin1_swedish_ci		Nei	Ingen		
description	varchar(300)	latin1_swedish_ci		Nei	Ingen		
parent	int(25)			Nei	Ingen		
file	varchar(50)	latin1_swedish_ci		Nei	Ingen		
timestamp	timestamp		on update CURRENT_TIMESTAMP	Nei	Ingen		ON UPDATE CURRENT_TIMESTAMP
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		

FIGURE 5.21: Data fields for the version table in phpMyAdmin

The version table has eight fields, only four of them are interesting. The title field is used to store the title of the version, this field is required by the user to type in manually. The description field is the description about the version, also required and typed manually. Parent field stores the parent versions from



the dropdown menu in the version box. Next is the file field, which is responsible for keeping track of what file is attached to the version. For example the field can have something like "track231.mp3" or "no file". "no file" is generated by default by a PHP script when a version is created.

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
reviewID	int(50)			Nei	Ingen		AUTO_INCREMENT
projectID	int(50)			Nei	Ingen		
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
datefrom	date			Nei	Ingen		
dateto	date			Nei	Ingen		
date	timestamp			Nei	CURRENT_TIMESTAMP		

FIGURE 5.22: Data fields for the reviews table in phpMyAdmin

When handling reviews in the ECHO system, a table called reviews is used to keep track of active reviews (see Figure 5.22). This table is very straight forward, as it stores start dates, end dates, date created and other necessary meta data such as username and ID's.

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
reviewRelID	int(50)			Nei	Ingen		AUTO_INCREMENT
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
projectID	int(50)			Nei	Ingen		
versionID	int(50)			Nei	Ingen		
reviewID	int(50)			Nei	Ingen		
rating	float			Nei	Ingen		

FIGURE 5.23: Data fields for the review relations table in php-MyAdmin

The review\_relations data table (in Figure 5.23) is responsible for tying together some of the parts of the review system. It stores a number of ID's together with a rating field. This rating field contains the overall score for versions that are reviewed. The overall score is not an individual user rating, but rather the total calculated score for one particular version in a review, as explained in Section 5.1.6.

The table responsible for keeping track of all users ratings on versions, is the review\_ratings table (see Figure 5.24). When a user rates a version, an Ajax call is sent to the database updating this table. This table may look similar to the review relations table but the rating field is storing what each individual user has rated, rather than the total calculated score. Storing individual rating prevent users from rating multiple times. In the event of a user clicking a

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
reviewRatingID	int(50)			Nei	Ingen		AUTO_INCREMENT
reviewID	int(50)			Nei	Ingen		
versionID	int(50)			Nei	Ingen		
rating	tinyint(4)			Nei	Ingen		
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
timestamp	timestamp		on update CURRENT_TIMESTAMP	Nei	Ingen		ON UPDATE CURRENT_TIMESTAMP

FIGURE 5.24: Data fields for the review ratings table in php-MyAdmin

thumb button, an Ajax call is sent to the server with either a "1" or a "0", which triggers updating the review rating table. The review relations "rating" field is updated accordingly each time a user rates a version.

Navn	Type	Sammenligning	Attributter	Null	Standard	Kommentarer	Ekstra
reviewCommentID	int(50)			Nei	Ingen		AUTO_INCREMENT
reviewID	int(50)			Nei	Ingen		
versionID	int(50)			Nei	Ingen		
username	varchar(50)	latin1_swedish_ci		Nei	Ingen		
comment	varchar(100)	latin1_swedish_ci		Nei	Ingen		
timestamp	timestamp		on update CURRENT_TIMESTAMP	Nei	Ingen		ON UPDATE CURRENT_TIMESTAMP

FIGURE 5.25: Review comments table storing users and their comments and timestamp

The review comments table stores a users name, comment and timestamp (see Figure 5.25). Review comments table keeps track of what version and review the comment is connected to, in the fields versionID and reviewID.

## Chapter 6

# Evaluation

This chapter presents the results from the evaluation performed after each iteration. A focus group session was conducted after the first iteration was done, and the purpose was to highlight problems and bugs. The focus group feedback was implemented into the second and final prototype of the system. When the second prototype was completed, usability tests and expert user interviews were conducted with the aim to assess usability as seen from a group of potential users. The purpose was to measure user experience using standardized methods.

### 6.1 Focus Group Session

In the focus group session, an early working prototype was presented to three users for testing and discussion. Individual user tests were conducted with the goal of uncovering problems. Using a semistructured interview guide, the focus group participants were interviewed (see Appendix C). They were asked to express freely their comments and criticism. The focus group's input was recorded, transcribed and analyzed, bullet points of feedback as well as ideas were summarized.

#### 6.1.1 The Participants

User tests was performed with information science students, all possessed a relatively high technical competence. Two of the focus group users had experience with making electronic music, and could be therefore considered target users of the system. Although one of the users had no experience with music production, being a very competent programmer, the user was expected to give valuable feedback in regards to technical aspects of the system, and ideally some suggestions for improvement.

### 6.1.2 User Testing

The user testing was performed with observer present, but only interacting on request. The tests were performed in a quiet room and natural environment with a laptop, keyboard and mouse. Together with the consent form (see Appendix B), a list of tasks (see Appendix C) was also provided to the user. The observer read the tasks aloud while the users performed the tasks. While the test was running, the users were asked to think aloud and describe problems while doing the tests. In order to have a complete picture of the user experience, a screen recorder was used. Every mouse movement on screen was recorded together with audio from both the user and observer.

Following are the tasks that was presented to the participants:

1. Register a new user
2. Login to the system
3. Try to make a new project
4. You have been given a number of versions of a song as individual files that you want to sort out in the system. Try to upload your files to the project
5. Try to make a new version and fill in information about the version
6. Try to play the uploaded file
7. Write a comment about the version
8. Try to create a new version of a project after the previous version
9. Try to fill out the system with all files
10. Try to change the color scheme of the system

### 6.1.3 Focus Group Results

After the individual tests, audio records and screen recordings were analyzed, to identify bugs and improvements suggested by users. The group was invited to discuss their overall experience with the system together as a group after the individual tests. This proved to be a rewarding activity, as it generated many ideas, as well as highlighted possible improvements.

Following is a list of the open-ended questions presented to the group for the discussion and input:

1. Explain how the login and registration went
  - (a) Is there anything that could have been done differently or changed in this regard?
2. Explain how creating a new project worked out
  - (a) Is there anything that could have been done differently or changed in this regard?
3. Explain your experiences of creating a new version of a project.
  - (a) Is there anything that could have been done differently or changed in this regard?
4. Explain how did it go trying to create a new version after the previous version.
  - (a) Is there anything that could have been done differently or changed in this regard?
5. Explain how uploading files to the system felt
  - (a) Is there anything that could have been done differently or changed in this regard?
6. How did the system feel overall?
  - (a) Were there any obstacles outside of the tasks that presented themselves?

After all the recorded data from the group and individual testing, two lists of essential improvements were created, summarizing what bugs and features for improvement.

To be improved list is as follows:

1. First version in the version tree did not allow to upload an audio file
2. When creating a new project, the first version should be auto generated
3. Project menu buttons still remaining after projects are deleted, causing confusion
4. Changing color theme would sometimes hang and not activate on first try
5. HTML injection is possible through the forms of the system, breaking the design in some parts

6. No response after uploading files
7. Fatal crash when uploading other files to the system than what are expected
8. Some text was hard to read
9. Site structure breaking when resizing browser window
10. Version tree not expanding to accommodate to the increasing number of version nodes
11. Make login and register page more clear and centered (suggested by all users)
12. Files in the dropdown list should be made less confusing by showing only current accessible versions
13. Make the file manager better and easier to use

Following is a list of features and functions that should be implemented as a result from conducting the first focus group:

1. Generate tree structure automatically
2. General meta data should be implemented
3. Visualization in the form of colors to differentiate projects
4. Appropriate icons for different functionalities

These two lists are the summarized critique of what was discussed by the group as a whole at the end, and by the individual users as well. After the focus group was conducted, these lists were used to improve the following, and last prototype of the ECHO system. Some of the improvements and features were not implemented due to time restraints, and also because they were not deemed important for the completion of the prototype. The only feature that was not implemented was the ability to generate a tree structure from a set of audio files.

Following is a list of the improvements that were not implemented:

1. Changing color theme would sometimes hang and not activate on first try
2. HTML injection is possible through the forms of the system, breaking the design in some parts (was improved but not fixed entirely)
3. No response after uploading files (this was improved, but there was still minimal response)

## 6.2 System Usability Scale and User Test

The System Usability Scale (SUS) test was chosen because of its properties to quickly give an impression of the system usability (see Section 6.2). In the same section, we saw that Tom Tullis and Bill Albert argued that with even a small sample size of three, a SUS user test can produce sufficient data to evaluate basic usability a system. Utilizing an effective combination of recording task time, Expert interviews and the SUS questionnaire (see Appendix D), it was possible to construct a clearer image of usability problems in the ECHO system.

### 6.2.1 Participants

Five users were recruited for this final user experience test gathered through samples of convenience. The three participants of the focus group session were invited to this final test along two new users. One of the two users was an expert in programming, and the other had experience in creating music. This group was expected to reflect on the features regarding of creation of music, and as well as system development and programming.

For the SUS test the users has had their names anonymized, following is a list of the codenames for each user and their experience. The letter E meaning expert user:

- SUS1E - Experience with creating music and system development
- SUS2E - Experience with creating music and system development
- SUS3E - Experience with creating music and system development
- SUS4 - Experience only with system development
- SUS5 - Experience only with system development

### 6.2.2 Test Setup

The environment where the tests took place was isolated from outside noise and distractions. The equipment used included a laptop, mouse and keyboard. Direct observation was used when performing the tests, the users were asked to perform a set of tasks. The tasks were read out loud by the observer, while users had the option of reading them from a document (see Appendix E.1). The time to completion was recorded for each user, and recorded on a document (see Appendix E.2). When the user eventually completed the task it would be

marked as completed. If the user was confused, and could not complete the task, the task was marked as not completed.

### 6.2.3 SUS Score

The final SUS score for the ECHO system prototype was 65.5 shown in Figure 6.1. This score is slightly below the average SUS score reported by Tullis and Albert to be 66 (see Figure 6.2) which is a number they siphoned after reviewing over 50 studies. The studies covered a large range of subjects and systems such as: websites, applications, mobile, hardware and voice systems. Tullis and Albert concluded that a score below 60 is relatively poor and one over 80 is considered pretty good (William, 2008).

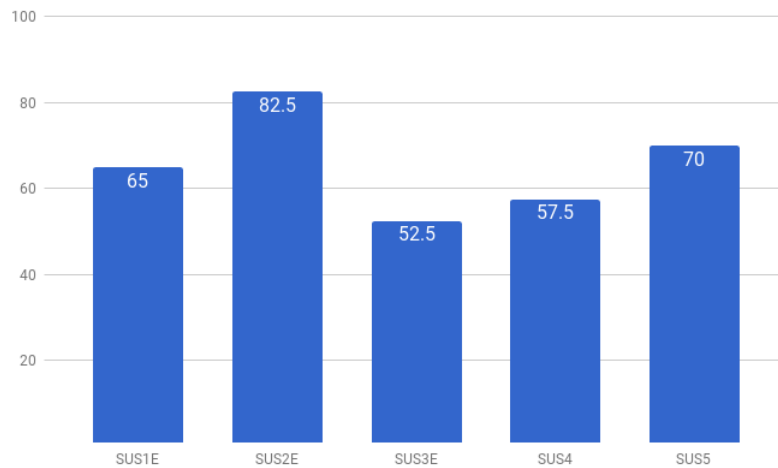


FIGURE 6.1: Individual total SUS scores, for every user in the user test

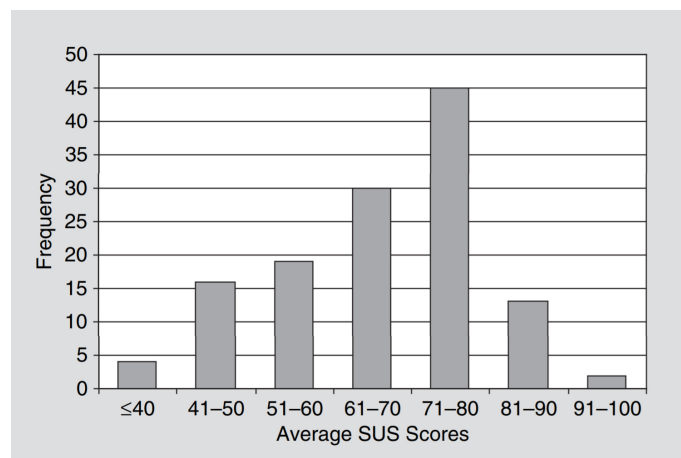


FIGURE 6.2: Average SUS scores (Adapted from William, 2008)



### 6.2.4 Learnability Score

James R. Lewis and Jeff Sauro describe learnability as being one of the key attributes in measuring usability in their book "The Factor Structure of the System Usability Scale" (Lewis and Sauro, 2009). The learnability of a system is tied to the users capacity to understand and master the mechanics of the system. If the system has few steps to accomplish a task, the system can be said to have a high degree of learnability (Lewis and Sauro, 2009). In their book they argue that a separate score of learnability of a system can be measured by scoring question 4 and 10 in the SUS questionnaire. An independent research paper by Borsci et al (Borsci, Federici, and Lauriola, 2009) have arrived at the same conclusion, combined score can measure the learnability of the system.

Question	SUS1E	SUS3E	SUS2E	SUS4	SUS5	Total
1	3	3	3	2	1	12
2	2	4	4	3	3	16
3	3	3	3	2	3	14
4	3	3	4	4	3	17
5	2	1	3	1	2	9
6	2	3	2	2	3	12
7	3	1	3	1	3	11
8	3	1	3	2	3	12
9	2	0	4	3	3	12
10	3	2	4	3	4	16
Total	26	21	33	23	28	Final SUS score
Individual SUS score	65	52.5	82.5	57.5	70	65.5

FIGURE 6.3: Collection of individual and total SUS scores from the second evaluation

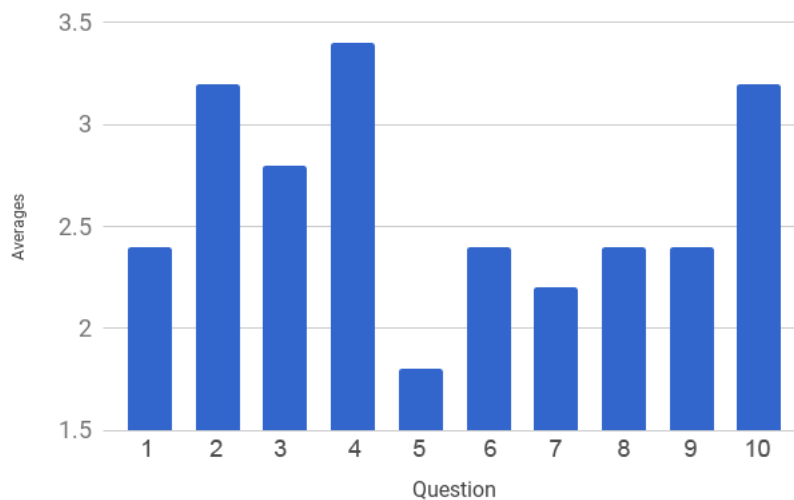


FIGURE 6.4: Average score per question using SUS

Looking at the Figure 6.4, it could be seen that question 4 and 10 were ones with the highest average scores on the questionnaire. By combining the scores of questions 4 and 10 from the table displayed in Figure 6.3, we see that the maximum score is 20 for the combined 5 users scores. Question 4 scored 17 and question 10 scored 16 resulting in 33 out of the maximum score of 40. The learnability score can then be calculated to be 82.5.

## 6.2.5 Task Time and Completion

After recording the recorded task times (see Figure 6.6), a graph was drawn to visualize the time differences between the users. Figure 6.5 suggests that the time to complete the tasks was relatively short for all the users, it was a matter of seconds. Users SUS3E and SUS4 however, experienced great trouble to locate the the thumb button that adds a version to the review window. Only two of the five users completed this task with relative ease. SUS3E experienced some fatal crashes in the software, which disabled all other functionalities. A complete reload of the application was required to get user SUS3E to resume the test.

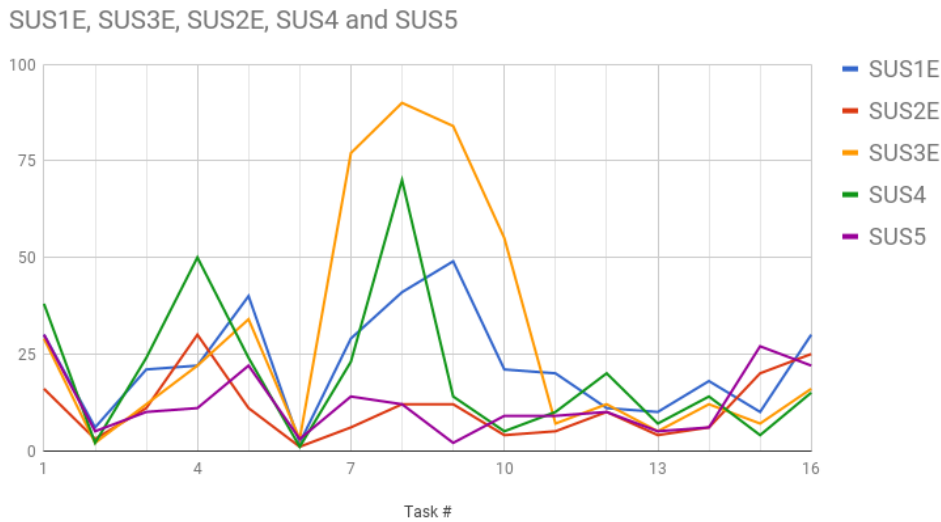


FIGURE 6.5: Time per task in seconds for participants

Looking at Figure 6.5, it is clear that the three users that had the lowest total SUS score (SUS1E, SUS2E and SUS4) had the longest time to complete task 7 through 10 (tasks can be seen in Appendix E.1). Users SUS2E and SUS5 completed task 7 through 10 in less than 15 seconds and reported that their experience with the system was satisfactory, which has likely contributed to

Task #	SUS1E	SUS2E	SUS3E	SUS4	SUS5
1	30	16	29	38	30
2	6	3	2	2	5
3	21	11	12	24	10
4	22	30	22	50	11
5	40	11	34	24	22
6	2	1	3	1	3
7	29	6	77	23	14
8	41	12	90	70	12
9	49	12	84	14	2
10	21	4	55	5	9
11	20	5	7	10	9
12	11	10	12	20	10
13	10	4	5	7	5
14	18	6	12	14	6
15	10	20	7	4	27
16	30	25	16	15	22

FIGURE 6.6: Time spent for task completion

their higher score in the SUS questionnaire. One of the more surprising results was coming from user SUS5, this user performed surprisingly fast in all of the tasks even though the user had no experience with the software beforehand (see Figure 6.5 and 6.6).

### 6.2.6 Expert User Interviews

With expert users SUS1E, SUS2E, SUS3E a brief 10-15 minute semi-structured interview were held after the tasks. The users were asked open-ended questions to avoid yes/no answers. Following are the questions asked:

1. Describe your experience with the ECHO system so far. Negative and positive
2. Describe how the work flow in the ECHO system felt
3. Did the system behavior feel surprising and unintentional?
4. What features do you miss, that you think could implemented in the system?
5. What do you think of the idea overall? Is it too cumbersome or manageable?
6. Anything extra you would like to include in the system?

All of the interviewed users reported various levels of confusion regarding the task where moving a version to the review panel. In addition to this, the placement of some panels and buttons was confusing as well. The visualization of the version tree could have been improved and made clearer. Some also

mentioned that the dropdown menus on the left side, in the project window, should have icons to reflect the function behind the menu items (see Figure 4.14 in Chapter 4).

Users also pointed out problems with the datepicker (illustrated in Figure 5.13 in Chapter 5). The date picker had no check in place to ensure that the start and end date were in correct order. As a consequence, a user could set a start date of 24th of November, and a end date on 12th of November, resulting in a review that will never be published. This bug was a large oversight that should have had to be fixed.

Regardless of the discussed bugs and flaws in the system, the expert users judged the ECHO system to be useful for its intended purpose. The expert user with the most musical experience (SUS1E) actually expressed great interest in the version tree visualization, more then the review system part. The same user pointed out that the tree would not only be of use personally, but also in a band setting when deciding different versions of riffs and melodies internally between members. Since the functionality in the system is relatively lean, the user thought that it would be quick to implement the system in to a band structure, rather than utilizing one of the larger more complex applications such as Splice. The ECHO system is not bound to any particular system so that if the user simply wants to upload a mp3 file with a voice recoding, it is possible to do so.

Another expert user (SUS2E), expressed more interest in the possibility of blind testing different versions more than the version tree. The user also mentioned that a number of options would have to be included in the review, for example the option of keeping the track names and artist name hidden or available to the listener. The user also expressed interest in seeing more filtering options that would help finding other projects up for review.

This SUS evaluation and user test concluded the final evaluation for the ECHO system. Further lists of improvements were not made at this point, but the most obvious point of focus, based on the participants feedback, would be to make adding versions to the review window easier, and also to have more options for artists on the review itself.

## Chapter 7

# Discussion

In this chapter the theory and methods used, as well as the research questions, validity and limitation of the research are discussed.

### 7.1 Theory

In the first section of Chapter 2 the thesis adopted a definition of creativity presented by Margaret Boden. In Section 2.1.2 Boden's three different thought processes of finding ideas in regards to creativity were explained: combinatorial, exploratory and transformational creativity. It was thus important to first understand how musical ideas can be generated, through the use of these three ways of thinking. Boden's work has been peer reviewed and referenced by a large amount of researchers, and therefore I chose to focus on her research in this regard, rather than multiple sources.

After establishing the basic definitions for what it means to generate creative thoughts, the creative process was looked at. In Section 2.1.3 Shneiderman presented the three different ways creators generate ideas through the creative process: structuralists, inspirationalists and situationalists. It became clear at this point that a non-romantic structuralist view of the creative process was going to be fitting for the project, utilizing tools and structured methods was tested by the development and evaluation of the ECHO system.

In addition to Boden's definition of creativity, Howard et al's definitions of appropriateness and originality in relation to the creative product, was highlighted in Section 2.1.4. These helped understand that measuring what is creative and original is an incredibly difficult task. In order to understand the meaning of data gathered when evaluating creative work, context of the data must be taken into account. And thus the ability of a person to detect potential creative ideas lies mostly on the persons ability to do so. Boden's work was

instrumental for understanding creative thought, but Shneiderman et al principles made it possible to introduce technology for building a tool in support of the structuralistic view of creativity.

Related works were also explored in Section 2.5, although the systems presented were commercial systems, they all had their benefits and limitations between them. The ECHO system's goal was to help organize versions and receive anonymous feedback quickly and organized, instead of presenting users with a bloated platform with unnecessary functionality to the user.

## **7.2 Methods**

The aim of this thesis was to develop a system based on an initial idea of visualization, and also to understand listeners preferences. There were several combined methods used in this research that produced an artifact in the form of a prototype. Following is a reflection on the methods used.

### **7.2.1 Design Science**

Since the thesis revolved around the development of an artifact "the ECHO system", Design Science Research framework was used as a guide for the research and development of the application. The seven guidelines were followed to ensure that the research was conducted in a rigorous and meaningful way. The communication of research results has been explained in accordance to guideline 7. Ensuring that both management-oriented audience, and technology-oriented audiences understand the concepts discussed.

### **7.2.2 Data Acquisition**

#### **User testing**

User testing was conducted by recording both the audio and cursor movement from the user while tasks was performed. The resulting recorded footage and audio was analyzed to pinpoint problems in the application. Both qualitative and quantitative data was collected: qualitative data in the form of expressed thoughts and feelings users had while executing tasks in the system, quantitative data by measuring task completion time in the second user test.

This combination of data acquisition was productive in the small sample size of test users. Ideally the group of potential users could have included a professional artist with longer experience, which could lead to a more critical assessment.

### **Focus Group**

A focus group evaluation was conducted after the first prototype was built. Due to the novelty of this artifact, it seemed to be important to secure a feedback and generation of ideas from the early stages of development. A combination of participants with and without experience with music creation was surprisingly good, since it gave a critical feedback, and unexpected constructive comments. If time and resources would allow the further development of the prototype, keeping the focus group would be recommended.

### **Expert Interviews**

In the second evaluation of the prototype, interviews with musical experts were conducted. The interviews with the expert users resulted in detailed and valuable feedback for the further development of the prototype. By inviting experts from different genres of music, the intention was see to how they evaluate the ECHO prototype, and not the content itself which was music. If the prototype was to be appreciated by several musicians of different genres, there is a chance that the tool has an appeal.

The result from the expert interviews in Section 6.2.6 indicated that despite obvious problems with the review system, some of the users expressed interest and thought that the system had potential. Most of the users noted that had the version tree been made better and clearer, and if the review system and dashboard had powerful filtering between genres, the ECHO system could become a valuable creativity support tool.

### **System Usability Scale**

The SUS questionnaire was used together with user tests after the second iteration. I found the resulting data from the SUS scoring to be a good indication of the usability and learnability of the system.

In Section 6.2.3 the resulting overall SUS score of 65.5 indicated that the prototype was still slightly below average level of usability. Learnability was measured to be 85.2 (see Section 6.2.4) suggesting that the users felt they could master the system with minimal effort. This confirmed the goal of having a low-threshold of entering the system, according to Shneiderman's principle.

Critical system failure with the review functionality was experienced by three of the users in the application, which was reflected in their individual scores. A score of 52.5, 57.5 and 65 were recorded for these users, suggesting that the average SUS usability score of the system could potentially have reached a significantly higher value, had these system failures not occurred for these users. The highest average score delivered by two users was 76.25, which suggests that they could use the system to a high degree. This score must be taken just for orientation, because the sample size is below the minimum of 3 participants, as suggested by Tullis and Bill Albert in Section 6.2.

Since the System Usability Scale is a standardized measure for usability of a system, getting a score such as this would mean that the system could be compared to other systems. Average good value for SUS is 66, meaning that when a score of 65.5, the ECHO system in its current state is comparable to in performance to the majority of good systems.

### 7.3 Compliance with Shneiderman's Principles

Shneiderman et al's principles (as discussed in Section 2.3) were used to both create and evaluate the ECHO system. The initial design sketch of the system was created after discussing the idea with fellow artists. Following the design sketches, the requirements were specified using the two scenarios that I refined to the best of my capacity as an artist. Hence the two main functionalities, one is versioning and the other is generating feedback. As an artist I wanted to have solid principles to guide my system development. Due to the detail, Shneiderman's principles were found to be the most suitable for the development of this kind of system.

#### **Support exploration**

The system fulfills the two requirements for supporting exploration within the system to a medium degree. Users should be able to explore and test out the



system freely to build knowledge of how to use the system, is the first requirement. The ECHO system fulfills this by having mostly clear buttons and icons indicating what the user can do next. Some of the buttons and functionality in regards to the review section of the system, were found to be hard to grasp by three of the five users in the last user test. Although most of the functions fulfill the second requirement, three of the users had problems with understanding how to proceed with the review system. Thus not meeting the second requirement for supporting exploration, which states that the system should be "self-revealing" (see Section 2.3.1).

#### **Low threshold, high ceiling, and wide walls**

The ECHO system can be said to have a relatively low threshold. It is expected that the intended users of this system have already established knowledge of how to make music and how to manage project files and audio files. All of the users that participated in the second evaluation of the prototype, expressed that the system was not especially hard to learn, which could be indicated by the Learnability measurement (see Section 6.2.4). In comparison to other creativity support tools such as FL Studio, the ECHO system can be said to have a relatively low threshold.

In regards to high ceiling and wide walls, the system could not be said to fulfill this requirement. Since only basic functionality is implemented into the system, powerful and customized review setups is not possible. The users are relatively free to use the version three as they wish, but the review system is relatively rigid. The ECHO system can be said to fulfill this principle to a medium degree.

#### **Support many paths and many styles**

The system should support many paths, in the sense that users could customize the way they use the software. In the testing phase all the users were asked perform a set of tasks, which surely influence the way they experience and worked with the system. However long term, they should be able to develop their own style of working. Users are free to explore and customize the system to fit their personal way of organizing their ideas. Uploading of music files, creation of projects and versions with custom names is possible within the system. The version tree structure, although limited, can be built in different ways by the users. Some users might explore a vertical level of their projects, others may dig deep into one specific version and then backtrack. I

will argue that the system supports custom paths to a medium degree.

The second requirement for this principle is that the application should support many styles. In the non-functional requirement (see Section 4.2.3), it was stated that the system should give the user freedom to customize the visual appearance of their application. Due to time limitations only a minimal level of customization was implemented in the prototype. The application had the option of two different color themes that the users could switch between. Further development of this feature could have been to introduce a color picker to let the user control exactly the look of the application.

### **Support collaboration**

A creativity support system should support collaboration between the users (see Section 2.3.4), the aim of the review functionality was to create a connection between the artist and listeners, so that the artist could evaluate what versions to focus on based on feedback. This could lead to a collaborative effort between listener and artist. In iteration 2 the system had implemented the basic functionality of the review system, and from user tests the review system was appreciated by two of the five users. However the majority of the users reported that it was difficult to maneuver to the section responsible for creating a review.

When the users eventually got a review of versions running, they got to see how the review looked when it was published at the dashboard. The users would also test functionality from the point of view of a listener, rating and commenting on the system. The review system is in its first completed version, just finished for the purpose of testing. More work should be done to enable collaborative functionality, which was remarked by two of the expert users (see Section 6.2.6).

I will argue that the system supports collaboration between artist and fans to a medium degree at this point, but the system has potential to fulfill the principle completely, when more collaborative options are implemented.

### **Support open interchange**

As a creativity support tool, the ECHO system can only be said to fulfill this principle to a low degree. Users do not have a possibility to use external plugins, and only minimal interchange of project data can be transmitted, such as

downloading audio files. Further development of the application could potentially support uploading of music project files rather than just audio files. In addition to this also the ability to embed music from other services such as Soundcloud for example (see Section 2.5.2).

#### **Make it as simple as possible and maybe even simpler**

Through the development process of the application, a constant level of simplicity has been kept, the second prototype was even more simplistic in how users created new projects and versions than in the first iteration. In the first iteration a user had to first create a project and then figure out how to create the first version, the user expressed some frustration since it was not clear where the version should appear. In the focus group session (see Section 6.1.3) all of the users agreed that creating a default version with no attached files would be a good option, users could then click on the version and edit the default title and description. The ECHO system can be said to fulfill this principle to a medium degree, by enabling different functionalities and making it easy to understand.

#### **Choose black boxes carefully**

Black boxes were chosen carefully in the ECHO system where the exchange of information between the users and the system took place. Where the exchange happened, most of the inner workings of the system were hidden. The user could, for example, edit a text box and click submit, after which the system would receive the data and process it further. If the PHP back-end experiences errors, a message is sent back through the same steps. The result is that the user only experience what is outside of the box. I would argue that the ECHO system have a high degree of choosing black boxes carefully.

#### **Invent things that you would want to use yourself**

Through the development of the system I have included functionality that I could see myself using as artist in addition to other user-suggested features. Using the system myself to conduct a case study with my own fans was one of my initial goals. I soon realized that it would result in an even longer iteration, since the production time of a musical work can sometimes take months. I will argue that the ECHO system fulfills this requirement to a high degree.

#### **Balance user suggestions with observation and participatory processes**

Through the development process, I sometimes found it hard to implement

some of the changes suggested by the users. For example, implementing expanding and collapsing menus in the project containing panel on the left, was one of the user suggestions. The suggestion was considered and later implemented, even though it was a demanding undertaking. I tried to tackle most of the technical challenges, while balancing user suggestions with my own ideas. The result are solutions that sometimes are not optimal but functional. Throughout the development, I will argue that the ECHO system has a high degree of balance between my own ideas, and user suggestions.

### **Iterate, iterate then iterate again**

According to Resnick et al (Resnick et al., 2005), throughout the development of a creativity support tool, several iterations must be expected. However only two iterations of the system were executed, even though the number should have been twice as many. Despite time restraints, the evaluation resulted in a good overview of the usability of the system. The development of the prototype therefore has a low degree of compliance with this principle.

### **Design for designers**

Designing for designers is hard, and in this case the designers are "musical designers", or composers. As an artist, I can already see both the potential and limitations of the system. Some of the limitations are held back by technical ability, and other limitations are due to keeping the scope of the project at a realistic level. However the development have resulted in a working prototype that acts as a creativity support tool for artists. In the current state, the prototype can be said to fulfill this principle to a medium degree.

### **Evaluate your tools**

The ECHO system have been evaluated using a combination of rigorously tested methods such as focus group, interviews, user tests, SUS questionnaire and task time measurements. These evaluations have culminated in a clear list of features that was implemented, and also system errors that need to be resolved for further development. The ECHO system can be said to fulfill this principle to a high degree.

## **7.4 Answering Research Questions**

**Research Question 1:** Is it possible to develop a creativity support tool that gives content creators a better overview of their ideas?

This research question addresses the problem of visualizing both in-progress ideas as well as finished ideas. With finished ideas, we understand that it is the versions produced and uploaded by an artist. Based on feedback from expert interviews (see Section 6.2.6), users have exclaimed that the overview functionality of the ECHO system is of personal interest to them. However, users have pointed out that more functionality and better visualization in the version tree would be desired.

The ECHO system has clearly given music creators a better way to keep an overview of their ideas. It has given them a possibility to utilize the systems functionalities, which was also shown in the evaluation.

**Research Question 2:** To what degree can a creativity support tool help content creators make easier choices, and decide which version to finalize?

This research question addresses the problem of determining if a creative product have reached an appropriate level of quality (described in Section 2.1.4). It is difficult to distinguishing between music versions, and to figure out which ones have greater appeal to the majority of users. Some artists would prefer not to answer this questions by themselves. The review system could help answering the question regarding which version of music is best liked and worth finalizing. Listeners and fans can influence the artist by giving ratings and even comments. The ECHO system is a competent solution to this question, by offering possibilities of reviewing versions. There is an expectation that users should have some creative experience, and the capability to create content and refine it. It is up to the artist to give context to the data, and to spot unobvious ideas (see Section 2.1.4) in order to fully utilize the system.

## 7.5 Research Limitation

The research has used rigorous methods to produce meaningful results, but the thesis has a number of limiting factors. User testing was performed twice throughout the development of the prototype, although ideally more iterations could have been performed. The sample size of the user group involved in the System Usability Scale evaluation, was 5, it was within the minimum recommended limit of 3. According to (Tullis and Stetson, 2004) in order to reach 80%

correct conclusion in a SUS test, a sample size of 10 would have been recommended. Thus the validity of usability and learnability scores are informative rather than fully reliable.

The research also do not make an attempt at measuring if it is possible to be more creative with the ECHO system. The system is supporting the process of creating the music, but not necessarily creativity itself. Measuring creativity is still a concept being discussed by researchers. But according to Resnick et al (see Section 2.3.12) it should still be possible to evaluate if a system can hinder creative processes. Based on usability principles, the evaluation in this thesis limits it self to only evaluating if the system support the process of creation.

## Chapter 8

# Conclusion

In this thesis the concept of creativity support tools was tested through the development of a prototype which acted as a connection between artists and listeners. How the prototype could be beneficial to artists deciding what versions of musical work to finalize, was the focus of the research. There were two functionalities explored that support the artist in creating music: one was to organize unfinished versions of music, and the other was to review and receive feedback from fans and listeners.

The development of the prototype was backed up by two standardized frameworks. One is Design Science as a research heuristics, and the other one is Shneiderman's principles for creativity support tools. These two frameworks ensured that the research was conducted in a rigorous way. Evaluation data was acquired by means of qualitative and quantitative methods. The prototype was developed through two iterations, and the final prototype was evaluated through user testing. Individual interviews and the System Usability Scale questionnaire, gave an indication that there was a potential and interest in such a system. The final prototype of the ECHO system, supports the process of creating music in a medium to high degree. The prototype however does not address the creation process itself, the artist is expected to have the prerequisite knowledge of how to create music. An artist, which is the user of the system, has all freedom to work and create the content while the system provides a sense of control and feedback from the users. Feedback captured by the system, has a potential to influence the artist through ratings and comments provided by users.

The system is not a finalized product, but in spite of certain limitations, it shows this way of working with music could be valuable to some users. There are similar commercial systems, but the novelty and advantage of this prototype is to keep the creative work process lean and without distractions.

## 8.1 Future Research and Development

### 8.1.1 Further System Development

If the system was to be used commercially, and with a larger userbase, the current dashboard where running reviews are displayed would have to be revised. Ideally filtering and sorting functionality should have been implemented, so that the users exploring and giving feedback, could also find music based on what genres they are interested in. One more important overall feature that would have to be included in the system is a help page. Documentation that describes how different parts of the system work, would have to be included to ensure that the difficulty threshold be kept at a low level.

In terms of security, functionality that prevents a user to make too many database calls would need to be implemented. At the moment it is possible to cause a distributed denial-of-service by utilizing a script that fills the database and continuously makes Ajax calls.

### 8.1.2 Login, Registration and Users

In regards to storing user information such as passwords, Md5 hashing and salting of passwords was not implemented into the current prototype. When dealing with a potential large and growing userbase, storing passwords safely in the database through md5 hashing and salting is mandatory. Since the system will be handling and controlling potential sensitive information of unreleased music, security around users is important. There are a few useful solutions to consider such as implementing password recovery through email, user validation through email and a personal "my account" section of the system.

### 8.1.3 Version Tree and Core Functionality

Customization tools and options in the version tree should be implemented to enable different ways to display the hierarchical structure of the version tree. In order to give the users a wider range of ways to work with their projects, more visual customization options on the overall system should be included as well. General sorting functionality and filters should be implemented, both in the version tree, and the project list on the left panel. For example, a user might want to list projects by the date created, instead of order created in the database.



### **8.1.4 Review and Feedback**

The Review and feedback functionality of the prototype should have more options. Artist may want the review to be as open as possible, displaying track names, ratings and comments by every user that has contributed. Currently it is only the artist who sees all the ratings and comments submitted by the users. Having more options on how to control the review should result in wider walls and higher ceiling.

## **8.2 Further Research**

The research conducted in this thesis does not address whether the ECHO system can make an artist more creative or not. The system helps artists keep an overview of their ideas and provide tools that enable a collaborative feedback from listeners or fans. The aspect of measuring creativity itself should be left to future projects and researchers in the cognitive and computer sciences. One of the future directions would be to explore whether the ECHO system in its current form, could contribute to make artists more creative.



# Bibliography

- Ajax (2017). *AJAX is the art of exchanging data with a server*. URL: <http://api.jquery.com/jquery.ajax/> (visited on 10/31/2017).
- Allihoopa (2017). *Allihoopa, music collaboration*. URL: <https://allihoopa.com/> (visited on 10/31/2017).
- Anderson, David J (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press. ISBN: 0984521402.
- Apache (2017). *Apache HTTP Server Project*. URL: [http://httpd.apache.org/ABOUT{\\\\_}APACHE.html](http://httpd.apache.org/ABOUT{\\_}APACHE.html) (visited on 10/31/2017).
- Boden, Margaret A (2004). "The Creative Mind: Myths and Mechanisms, Second Edition". In: *SciencesNew York*, pp. 1–359. ISSN: 1098-6596. DOI: 10.4324/9780203508527. arXiv: arXiv:1011.1669v3.
- Borsci, Simone, Stefano Federici, and Marco Lauriola (2009). "On the dimensionality of the System Usability Scale: A test of alternative measurement models". In: *Cognitive Processing* 10.3, pp. 193–197. ISSN: 16124782. DOI: 10.1007/s10339-009-0268-9.
- Brace, Jenny (2008). "Versioning in Repositories: Implementing Best Practice". In: *Ariadne* 56.56. URL: <http://www.ariadne.ac.uk/issue56/brace>.
- Breen, Rosanna L. (2006). "A practical guide to focus-group research". In: *Journal of Geography in Higher Education* 30.3, pp. 463–475. ISSN: 03098265. DOI: 10.1080/03098260600927575.
- Brooke, John (2013). "SUS : A Retrospective". In: *Journal of Usability Studies* 8.2, pp. 29–40. ISSN: 1931-3357. DOI: 10.1074/jbc.R115.675280. URL: [http://www.usabilityprofessionals.org/upa{\\\\_}publications/jus/2013february/brooke1.html{\\%}5Cnhttp://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html](http://www.usabilityprofessionals.org/upa{\\_}publications/jus/2013february/brooke1.html{\\%}5Cnhttp://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html).
- CSS (2017). *CSS is a mechanism for adding style web documents*. URL: <https://www.w3.org/Style/CSS/> (visited on 10/31/2017).
- D3.js (2017). *D3.js is aJavaScript library for manipulating documents*. URL: <https://d3js.org/> (visited on 10/31/2017).

- Fischer, Gerhard and Kumiyo Nakakoji (1994). "Amplifying Designers' Creativity with Domain-Oriented Design Environments". In: *Artificial Intelligence and Creativity: An Interdisciplinary Approach*. Ed. by Terry Dartnall. Dordrecht: Springer Netherlands, pp. 343–364. ISBN: 978-94-017-0793-0. DOI: 10.1007/978-94-017-0793-0\_25. URL: [https://doi.org/10.1007/978-94-017-0793-0\\_{\\\_}25](https://doi.org/10.1007/978-94-017-0793-0_{\_}25).
- FL Studio (2017). *FL Studio, digital audio workstation*. URL: <https://www.image-line.com/flstudio/> (visited on 11/27/2017).
- GIT (2017). *Git, an open source distributed version control system*. URL: <https://git-scm.com/> (visited on 10/31/2017).
- Graham, Paul (2004). *Hackers and painters - big ideas from the computer age*. O'Reilly, pp. I–XI, 1–258. ISBN: 978-0-596-00662-4.
- Hevner, Alan R et al. (2004). "Design Science in Information Systems Research". In: *MIS Quarterly* 28.1, pp. 75–105. ISSN: 02767783. DOI: 10.2307/25148625. URL: <http://dblp.uni-trier.de/rec/bibtex/journals/misq/HevnerMPR04>.
- Howard, T. J., S. J. Culley, and E. Dekoninck (2008). "Describing the creative design process by the integration of engineering design and cognitive psychology literature". In: *Design Studies* 29.2, pp. 160–180. ISSN: 0142694X. DOI: 10.1016/j.destud.2008.01.001.
- Howard, Thomas James, S J Culley, and E Dekoninck (2006). "Information as an input into the creative process". In: *DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia*.
- HTML (2017). *HTML is the most basic building block of the Web*. URL: <https://www.w3.org/html/> (visited on 10/31/2017).
- Javascript (2017). *Lightweight programming language developed by ECMA*. URL: <http://www.ecma-international.org> (visited on 10/31/2017).
- Krueger, Richard a and Mary Anne Casey (2000). "Focus groups: A practical guide for applied research". In: *Review Literature And Arts Of The Americas* 22, pp. 129–152. ISSN: 07408188. DOI: 10.1002/j.1556-6678.2007.tb00462.x. URL: <http://books.google.com/books?hl=en{\&}lr={\&}id=ySKeSZeWywMC{\&}pgis=1>.
- Lewis, James R. and Jeff Sauro (2009). "The factor structure of the system usability scale". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 5619 LNCS, pp. 94–103. ISBN: 3642028055. DOI: 10.1007/978-3-642-02806-9\_12. arXiv: 978-0-9846455702-0-1.

- Lubart, Todd I. (2001). "Models of the Creative Process: Past, Present and Future". In: *Creativity Research Journal* 13.3-4, pp. 295–308. ISSN: 1040-0419. DOI: 10.1207/S15326934CRJ1334\_07. URL: [http://www.tandfonline.com/doi/abs/10.1207/S15326934CRJ1334{\\\_}07](http://www.tandfonline.com/doi/abs/10.1207/S15326934CRJ1334{\_}07).
- MySQL (2017). *MySQL is a open-source RDBMS*. URL: <https://www.mysql.com/> (visited on 10/31/2017).
- OMG (2017). *BPMN, Business Process Model and Notation*. URL: <http://www.omg.org/spec/BPMN/2.0/PDF> (visited on 10/31/2017).
- Photoshop (2017). *Photoshop, a raster graphics editor*. URL: <http://www.adobe.com/products/photoshop.html> (visited on 10/31/2017).
- PHP (2017). *PHP Hypertext Preprocessor*. URL: <http://php.net/> (visited on 10/31/2017).
- PhpMyAdmin (2017). *phpMyAdmin is a free software tool written in PHP*. URL: <https://www.phpmyadmin.net/> (visited on 10/31/2017).
- Preece, Jenny, Helen Sharp, and Yvonne Rogers (2015). *Interaction Design: Beyond Human-Computer Interaction*. Wiley. ISBN: 1119020751.
- Resnick, Mitchel et al. (2005). "Design Principles for Tools to Support Creative Thinking". In: *Science* 20.2, pp. 25–35. DOI: -. URL: <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1822{\&}context=isr>.
- Shneiderman, Ben (2007). "Creativity support tools: accelerating discovery and innovation". In: *Communications of the ACM* 50.12, pp. 20–32. ISSN: 00010782. DOI: 10.1145/1323688.1323689. URL: <http://portal.acm.org/citation.cfm?doid=1323688.1323689>.
- Sielis, George A., Aimilia Tzanavari, and George A. Papadopoulos (2009). "Enhancing the creativity process by adding context awareness in creativity support tools". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 5616 LNCS. PART 3, pp. 424–433. ISBN: 3642027121. DOI: 10.1007/978-3-642-02713-0\_45.
- Soundcloud (2017). *Soundcloud, online audio distribution platform*. URL: <http://soundcloud.com> (visited on 10/31/2017).
- Splice (2017). *Splice, music creation and collaboration*. URL: <https://splice.com/> (visited on 10/31/2017).
- SVG (2017). *Scalable vector graphics, graphics format by W3C*. URL: <https://www.w3.org/Graphics/SVG/> (visited on 10/31/2017).
- Sylenth1 (2017). *Sylenth1, virtual synthesizer plugin*. URL: <https://www.lennardigital.com/sylenth1/> (visited on 11/27/2017).

- Trello (2017). *Trello, web-based project management application*. URL: <https://trello.com/> (visited on 10/31/2017).
- Tullis, Thomas S and Jacqueline N Stetson (2004). "A Comparison of Questionnaires for Assessing Website Usability ABSTRACT : Introduction". In: *Usability Professional Association Conference*, pp. 1–12. URL: <http://home.comcast.net/~tomtullis/publications/UPA2004TullisStetson.pdf>.
- Wamp (2017). *Wamp server, windows apache MySQL PHP stack*. URL: <http://www.wampserver.com/en/> (visited on 10/31/2017).
- William, Albert; Tullis Thomas (2008). "MeasuringThe User Experience: Collecting, Analyzing, and Presenting Usability Metrics". In: *Vasa*, p. 318. ISSN: 0896-9205. DOI: 10.1016/B978-0-12-373558-4.00012-1. arXiv: arXiv:1011.1669v3. URL: <http://medcontent.metapress.com/index/A65RM03P4874243N.pdf>.
- Youtube (2017). *Youtube online videos*. URL: <http://www.youtube.com> (visited on 10/31/2017).

## **Appendix A**

# **Approval from NSD**

Ankica Babic  
Institutt for informasjons- og medievitenskap Universitetet i Bergen  
Fosswinckelsgate 6  
5007 BERGEN

Vår dato: 02.11.2016

Vår ref: 50453 / 3 / AGL

Deres dato:

Deres ref:

## TILBAKEMELDING PÅ MELDING OM BEHANDLING AV PERSONOPPLYSNINGER

Vi viser til melding om behandling av personopplysninger, mottatt 07.10.2016. Meldingen gjelder prosjektet:

<i>50453</i>	<i>Expressive Creation History Overview system - E.C.H.O</i>
<i>Behandlingsansvarlig</i>	<i>Universitetet i Bergen, ved institusjonens øverste leder</i>
<i>Daglig ansvarlig</i>	<i>Ankica Babic</i>
<i>Student</i>	<i>Daniel Taranger</i>

Personvernombudet har vurdert prosjektet og finner at behandlingen av personopplysninger er meldepliktig i henhold til personopplysningsloven § 31. Behandlingen tilfredsstiller kravene i personopplysningsloven.

Personvernombudets vurdering forutsetter at prosjektet gjennomføres i tråd med opplysningene gitt i meldeskjemaet, korrespondanse med ombudet, ombudets kommentarer samt personopplysningsloven og helseregisterloven med forskrifter. Behandlingen av personopplysninger kan settes i gang.

Det gjøres oppmerksom på at det skal gis ny melding dersom behandlingen endres i forhold til de opplysninger som ligger til grunn for personvernombudets vurdering. Endringsmeldinger gis via et eget skjema, <http://www.nsd.uib.no/personvern/meldeplikt/skjema.html>. Det skal også gis melding etter tre år dersom prosjektet fortsatt pågår. Meldinger skal skje skriftlig til ombudet.

Personvernombudet har lagt ut opplysninger om prosjektet i en offentlig database, <http://pvo.nsd.no/prosjekt>.

Personvernombudet vil ved prosjektets avslutning, 01.06.2017, rette en henvendelse angående status for behandlingen av personopplysninger.

Vennlig hilsen

Kjersti Haugstvedt

Audun Løvlie

Kontaktperson: Audun Løvlie tlf: 55 58 23 07

Vedlegg: Prosjektvurdering

*Dokumentet er elektronisk produsert og godkjent ved NSDs rutiner for elektronisk godkjenning.*



## **Appendix B**

# **Informed Consent Form**

# Informed consent form Echo system

## Request for participation in research project “Echo system”

Institution responsible for research project: University of Bergen

Project ending: 01.12.2017

Anonymization of data: 01.12.2017

### Introduction

Echo system is an abbreviation for “Expressive Creative History Overview”, The system will be a tool for musicians and potentially other types of artists to create and manage “versions” of their art. The tool will feature a overview over a project with versions and feedback sections. In addition to this there will be a feedback system built that will give the artist feedback on which version of the artwork could possibly be the “right” one. The Echo system extends the decision making in regards to the project from the artist’s standpoint out to the listeners or viewers of the art.

### What is the purpose of the study?

The project has three research questions:

1. *“In what way can the ECHO system give content creators a better overview of their projects.”*
2. *“Will the Expressive creation history overview, ECHO (working title) help creative content creators to have an easier choice on which version of a project to choose.”*
3. *“Can the feedback, process and visualization given by the ECHO system help and give feedback to content creators in other creative fields than music.”*

### What makes one eligible to participate in this focus group?

You have general programming knowledge, and will give critical and constructive criticism towards the project. Although preferred, you will not be required to be creative yourself, objective criticism of the structure of the system is more important early on.

### Practical information

If you agree to be in this study, you will be asked to do the following:

## **Appendix C**

# **Focus Group Tasks and Questions**

# Focus group session for the Echo system

## Goal:

To test and discuss the first iteration of the echo system prototype. To propose new and better functions for the system, changes that can be made in the next iteration.

## Method:

1. Individual testing of software through specific tasks.
2. Focus group where all participants will be asked to discuss the tasks and further questions for improvement of the software.

## Tasks

1. Register a new user
2. Login to the system
3. Try to make a new project.
4. You have been given a number of versions of a song as individual files that you want to sort out in the system. Try to upload your files to the project.
5. Try to make a new version and fill in information about the version.
6. Try to play the uploaded file.
7. Try to create a new version of a project attached to the previous version
8. Try to change the color scheme of the system.
9. Try to break the system

Additional tasks may appear.

## Questions for group discussion

Questions will be asked openly for all members to discuss.

1. Explain how the login and registration went
  - a. Is there anything that could have been done differently or changed in this regard?
2. Explain how creating a new project worked out
  - a. Is there anything that could have been done differently or changed in this regard?
3. Explain your experiences of creating a new version of a project.
  - a. Is there anything that could have been done differently or changed in this regard?

## **Appendix D**

# **System Usability Scale Questionnaire**

Strongly disagree

Strongly agree

1. I think that I would like to use this system frequently

1	2	3	4	5

2. I found the system unnecessarily complex

1	2	3	4	5

3. I thought the system was easy to use

1	2	3	4	5

4. I think that I would need the support of a technical person to be able to use this system

1	2	3	4	5

5. I found the various functions in this system were well integrated

1	2	3	4	5

6. I thought there was too much inconsistency in this system

1	2	3	4	5

7. I would imagine that most people would learn to use this system very quickly

1	2	3	4	5

8. I found the system very cumbersome to use

1	2	3	4	5

9. I felt very confident using the system

1	2	3	4	5

10. I needed to learn a lot of things before I could get going with this system

1	2	3	4	5

## **Appendix E**

# **Final Evaluation Documents**

### **E.1 Final User Test Tasks**

# User test session for the Echo system

## Goal:

To test and discuss the first iteration of the echo system prototype. To propose new and better functions for the system, changes that can be made in the next iteration.

## Method:

1. Individual testing of software through specific tasks. While recording audio and screen.

## Tasks Artist

1. Register a new user and Login to the system.
2. Try to change the color scheme of the system.
3. Try to make a new project.
4. **You have been given a number of audio files on the desktop of the computer that you want to sort out in the system.** Try to upload the audio files to the project.
5. Try to edit the default version "ver 1", fill in title and description and audio file .
6. Try to play the uploaded file on the version.
7. Try to create a new version attached to the previous version you edited, fill in information and add audio file.
8. Try to create a new review using the panel to the right. Add the two versions to the review panel.
9. Pick start date and end date. Submit review and control that it existst.
10. Navigate to the review you just created and end the review.
11. Delete a version.
12. Delete audio files.
13. Delete the project.

## Tasks Feedback giver

1. Login with username "feed" and password "feed".
2. Go to the review page and pick a random review.
3. Give feedback by commenting and rating each version

## Bonus tasks

1. Login with username "bonus" and password "bonus",
2. Observe the version tree and feedeback given to each version.



## **E.2 Task Time**

## Task time and completion

Artist

Task name	time	completed?
Task 1		
Task 2		
Task 3		
Task 4		
Task 5		
Task 6		
Task 7		
Task 8		
Task 9		
Task 10		
Task 11		
Task 12		
Task 13		

Feedback

Task name	time	completed?
Task 1		
Task 2		
Task 3		

Bonus

Task name	time	completed?
Task 1		
Task 2		

