Forprojektrapport
E18 E6BAC-01 Forberedelse til Bachelorprojekt

Designing Multichannel Audio- and Video-playback system: Showman

Group members:

Name	Studentid.
Minik Nathanielsen Olsen	201600341

Contents

Chapter 1

Project Description

The following text is the project description pulled from the project catalog.

Development of multi-channel audio/video playback system for touring artists. This project will integrate both hardware- and software-design with emphasis on hardware design. Approx. 70-80 percent of modern Danish and international touring artists utilize audio backing tracks (prerecorded material played back on some chosen device) and video content projected on large screens as part of their performance.

However, there are a multitude of options ranging from iPod-playback to complex, customized systems operated by technical personnel.
Common for these systems is instability, making artists to purchase redundant backup-systems in order to deliver their audience a flawless and uninterrupted performance.

In a typical scenario, at least one laptop – often several laptops – with various audio- and/or videointerfaces, handles and runs the audio backing track and video-content in sync.

This project will integrate both audio- and video-handling in a single 19" rack-mounted device and provide the user with a user-friendly GUI (Graphical User Interface) while complying with international standards for audio and video and electronic devices.

For design and implementation of the device, a research is needed in:
- Circuit/hardware design
- D/A audio converters
- Standardized audio input/output formats
- Audio file formats
- Standardized video input/output formats

- Video file formats
- Time codes for synchronization
- User feedback for feasible GUI design

This project is already in the early stages of development as part of an apprenticeship in entrepreneurship (iværksætterpraktik) at Navitas Science and Innovation.
Current status on project:
- Requirements specification finished
- FURPS finished
- Market research ongoing

The project is requesting confidentiality as the device is in development intended for market release to prosumers in the music industry, thus an eventual project partner must sign a non-disclosure agreement.

Project partner profile: Electronic Engineering student or Information and Communication Technology student.

Chapter 2

Requirement Specification

Introduction

This chapter will describe the requirements needed for the multichannel audio/video playback system called Showman. This chapter contains an introductory overview of Showman in general that covers a System Description, System Overview and Usage Situation followed by Functional and Non-Functional Requirements. The chapter ends with a MoSCoW- and FURPS-analysis to organize a list of priorities needed to develop a functional prototype.

System Description

The purpose of Showman is to play back prerecorded audio- and video-material during a concert performance. Prerecorded audio- and video-playback material is colloquially called 'backing tracks'. The primary reason of backing track usage is to enhance a concert performance by play back additional material that artists on stage do not have the option or time to play themselves.

User (or artist) uploads synchronized audio files in 44.1 kHz 16-bit .WAV-format and video files in MPEG-4 format to Showman before using Showman during showtime. User creates, edits and organizes playlist with a software application Graphical User Interface (GUI) on laptop (PC or Mac). The content of the playlist is the backing track material that user produced beforehand. After playlist creation, user uploads the playlist via a USB-connection to Showman's internal memory.

Showman play back 8-tracks of audio along with video-content. The audio tracks can be connected directly into the XLR-inputs of a live sound mixer through Showman's XLR outputs, while the video-material can be connected to screens through Showman's HDMI-output. User starts each song in the playlist separately by pressing the 'Play'-button on Showman. The first song

on the playlist begins and playback stops automatically (auto-stop) after each ending of a song.

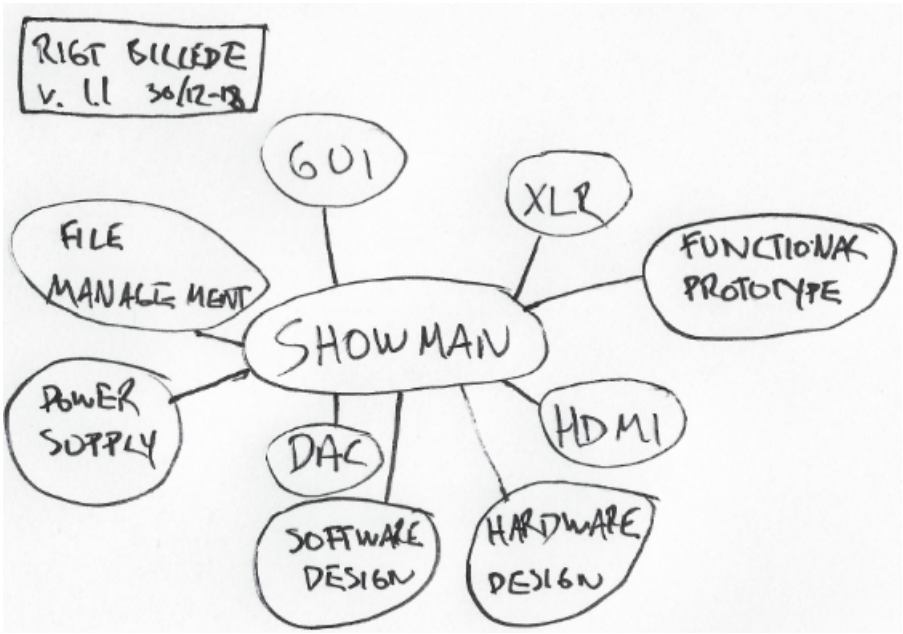

Figure 2.1 – Content of Showman as a rich picture.

Usage Situation

Showman's intended users are artists that utilize backing tracks at concert performances to enhance/augment their performance. Preconditions are essential to Showman, as user must have up to 4 stereo or 8 mono .WAV-files and a MPEG4-videofile available for each song that has backing tracks. The artist need to have produced the backing tracks beforehand.

As Showman's intended usage conditions are tough touring conditions and periodic hard handling, Showman's user interface is a 2U 19" rack mounted device with a LCD screen for user monitoring and panel buttons for navigation.

When the user have the necessary files available, the user assembles the playlist in the software application GUI and assigns the audio outputs. The GUI uploads the playlist project into specific folders in Showman's internal 500 GB flash memory.

When user need to start playback of a song, all user need to do is press 'Play' on the user panel. An auto-stop function after the end of each song in

the project is embedded in Showman.

Actor-Context Diagram

The figure below is an overview of actors that interact with Showman:

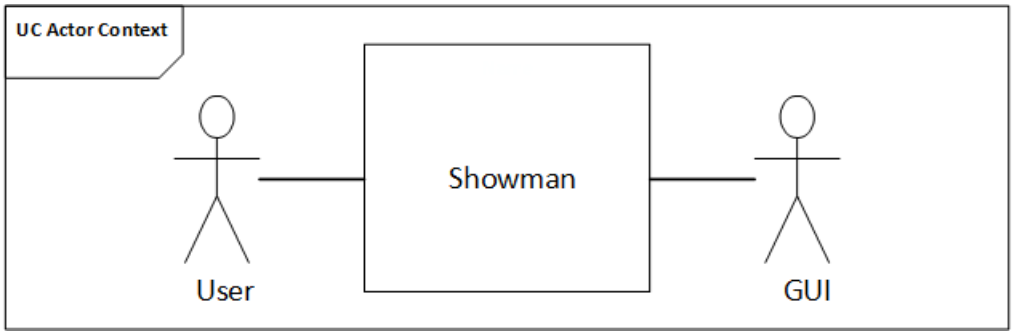

Figure 2.2 – Actor-Context diagram of Showman.

User

User is the primary actor. User creates the playlist and uploads the playlist to Showman. User operates Showman. User's interaction with Showman is outlined in detail in the specification for the individual Use Cases.

GUI

The Graphical User Interface (GUI) is the secondary actor. The GUI is the software application that handles the file management and playlist creation that is uploaded to Showman's interaction with Showman is outlined in detail in the specification for the individual Use Cases.

Functional Requirements

This section presents the functional requirements outlined for Showman. The figure below presents a Use-Case diagram that displays the actors' relations to the Use-Cases followed by Fully Dressed Use-Cases.

To keep this document brief and concise, only the first Use-Case is included. The additional Use-Cases will be explored further in the bachelor project.

Use-Case 1

Name:	Create Playlist
Scope	User wants to create a playlist
No. of concurrent events:	1
Primary Actor	User
Stakeholders	User wants to use backing tracks with Showman.
Preconditions	User has backing track files
Postconditions	User has uploaded files successfully
Main Success Scenario	1. User connects GUI to Showman via USB-connection. 2. User opens GUI. 3. User clicks on 'New Playlist'. 4. User sets desired playlist name. 5. GUI ask User to drag and drop files in Playlist Content panel. 6. GUI assist User in assigning output ports. 7. User saves playlist. 8. User clicks 'Showtime'. 9. GUI transfer playlist to Showman. 10. Showman is ready for use.
Extensions	[Extension 4a:] 1. GUI prompt displays 'Playlist already exist. Overwrite?'. 2. User prompts 'Cancel'. 3. GUI returns to Main Success Scenario step 4. [Extension 4b:] 1. GUI prompt displays 'Playlist already exist. Overwrite?'. 2. User prompts 'OK'. 3. User overwrites playlist. 4. GUI continues to Main Success Scenario 5. [Extension 5a:] 1. GUI displays 'INCORRECT FILE FORMAT'. 2. GUI returns to Main Success Scenario step 5. [Extension 5b:] 1. GUI displays 'The files exceeds memory limit. Please reduce file size.' 2. GUI returns to Main Success Scenario step 5. [Extension 6:] 1. GUI displays 'Too many files. Reduce number of files to accommodate number of available output ports. 2. GUI returns to Main Success Scenario 6.
Special Requirements	

Figure 2.3 – Use-Case diagram of Use-Case 1.

Non-Functional Requirements

Non-Functional Requirements (NFRs) are quality-demands, which are the qualities or constraints on the services of the functions offered by the system rather than a specific behaviour. Qualities are properties or characteristics of the system that its stakeholders care about and hence will affect their degree

of satisfaction with the system.
Quality demands/NFRs should satisfy two attributes:
- Must be verifiable - e.g. measurable metrics. - Should be objective.

The NFRs are drawn up using two techniques: FURPS and MoSCoW to determine system requirements and to prioritize and rank Non-Functional Requirements.

FURPS+

FURPS+ is an acronym that represents a model classify a product's properties: **F**unctionality, **U**sability, **R**eliability, **P**erformance, **S**upportability and + (Design and Physical constraints, Interfaces, Legal, Test, Reuse, Economic constraints, Aesthetics, Comprehensibility, Techology tradeoff, etc.)

A point-system from 1-5 is used to classify priorities, where 1 is the lowest and 5 the highest priority.

F - Functionality

The functionality describes the capability (size and generality of feature set), reusability (combability, interoperability, portability) and security (safety and exploitability) of a system.

Use-Case 1 - Create Playlist
UC1 has the highest priority as it is essential for Showman to create playlists with GUI in order to operate Showman.
Score: 5

U - Usability

Accessibility
Description of the ease of access to system
As the target group are touring artists and artist crew both with little to no time, the system need to be easy to use and operate with minimal learning curve.
Score: 4

User Interface
Description of the Graphical User Interface's functionality
The system need a GUI that need minimal learning curve, and therefore has to be similar to GUIs of software applications such as ProTools, Apple Logic

X, Ableton Live, Cymatic Audio uTool.
Score: 3

R - Reliability

Description of the system's stability

The reliability describes the availability (failure frequency, robustness/-durability/resilience, failure extent and time-length (recoverability/survivability), predictability (stability) and accuracy (frequency or severity of error) of the system.

P - Performance

The performance describes the speed, efficiency, resource consumption (power, RAM, cache, etc.), throughput, capacity, scalability of the system.

S - Supportability

The supportability describes the serviceability, mantainability, sustainability, repair speed, testability, flexibility (modifiability, configurability, adaptabilit, extenxsibility, modularity), installability and localizability of the system.

+ - Design constraints

The "+s" of the FURPS+ acronym allows us to specify constraints, including design, implementation, interface, and physical constraints.

Design Constraints – A design constraint, as the name implies, limits the design — for example, requiring a relational database stipulates the approach that we take in developing the system.

Implementation Constraints – An implementation constraint puts limits on coding or construction – standards, platform, or implementation language.

Interface Constraints – An interface constraint is a requirement to interact with an external item. When you develop within an enterprise, quite often you have to interact with external systems.

Physical Constraints – Physical constraints affect the hardware used to house the system – for example, shape, size, and weight.

MoSCoW

Chapter 3

Project Plan

(Udkast til) projektplan, herunder beskrivelser af hvilke eksperimenter, teknologier mm, der forventes udarbejdet i løbet af afgangsprojektet.

Chapter 4

Project Research

Undersøgelse af tilsvarende projekter og relevant litteratur.

Chapter 5

Project Expectations

Evt. aftale om forventet arbejdssted og tid.

Chapter 6

Conclusion

Konklusion på det indledende arbejde med forprojektet.

List of Figures