Digital Audio Workspace

Analysis and Design Document

Student: Cioban Dumitru-Darius

**Group:30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/03/2020 | 1.0 | Project Deliverable 1. Project Specification, Elaboration Iteration I, Construction and Transition sections added. | Cioban Dumitru-Darius |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# Digital Audio Workspace is an application which lets a user edit and make an audio file.

# To do this, the user will create one or more MIDI objects attached to an instrument.

# After this, the objects may be arranged and aligned based on a time dimension to produce a music piece.

# The application has only one user, which can be called the producer.

# Elaboration – Iteration 1.1

# Domain Model

# Architectural Design

## Conceptual Architecture

## Package Design

## Component and Deployment Diagrams

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

The integration tests are performed based on the use-case diagram and they will indicate that the system functions properly.

Create pattern: The user will create a pattern by putting notes of various duration, pitch and timestamp.

Edit pattern: The user may edit an already existing pattern.

Assign instrument: Assign an instrument to a pattern, which will play the melody described by the pattern.

Delete pattern: The user may delete an existing pattern.

Arrange workspace: The user may use existing patterns to place them in the main workspace, thus creating the music piece.

Export: The user may export the workspace as an audio file.

# Future improvements

The application can be improved in many ways, namely including more instruments, giving the ability to import one’s own instrument providing the correct file, adjusting various settings through mixing and mastering, and using a MIDI-style input to create patterns.

# Bibliography