**MBARARA UNIVERSITY OF SCIENCE AND TECHNOLOGY**

****

**FACULTY OF COMPUTING AND INFORMATICS**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**COURSE: SOFTWARE ENGINEERING INDUSTRIAL MINI PROJECT II**

**COURSE CODE: SWE 4106**

**INSTRUCTOR: DR. ADONES RUKUNDO**

**STUDENT: DAVID NYAMUTALE**

**REGISTRATION NUMBER: 2020/BSE/057/PS**

**TEST CASE DOCUMENT FOR THE MULTIMODAL CONTENT CREATION LIBRARY**

**Introduction**

This document gives a list of the test cases that have been peformed on the Audio library along with the corresponding java project and python project where this library has been imported as module that is part of the multimodal content creation Library. This module is focused only on the audio part where the implementations made are to facilitate content creation in form of sound /audio recording, storing content and retrieving content. Other implementations are on being able to import this library in other java projects and also being able to import it in python project.

The attached files include all the module files along with their Test cases from which some of the results included in this document have been obtained.

**Test Cases**

**Test Case ID: AL-TC-001**

**Test Case Description**: Record audio using AudioLibrary's recordAudio method.

**Pre-Conditions:** None

**Test Steps:**

1. Call recordAudio method with a specified duration (e.g., 10 seconds).
2. Verify that the returned audio data is not null.
3. Verify that the length of the audio data is greater than 0.

**Test Data:** Duration = 10 seconds

**Expected Result:** The method should successfully record audio, and the returned audio data should not be empty.

**Post Condition:** Notification of successful completion of the recording

**Actual Result:** Successful

**Status:** Pass

**Test Case ID: AL-TC-002**

**Test Case Description:** Save audio to the database using AudioLibrary's saveAudioToDatabase method.

**Pre-Conditions:** Must be connected to a test database.

**Test Steps**:

1. Call saveAudioToDatabase method with a sample name and audio data.
2. Verify that the audio recording is saved to the test database.
3. Test Data: Sample name, audio data

**Expected Result:** The method should save audio data to the database without errors.

**Post Condition:** Audio recording saved in the test database.

**Actual Result:** Successful

Status: Pass

**Test Case ID: AL-TC-003**

**Test Case Description:** Play audio from the database using AudioLibrary's playAudioFromDatabase method.

**Pre-Conditions:** Connected to a test database, sample audio recording in the database.

**Test Steps:**

1. Call playAudioFromDatabase method with the sample name.
2. Verify that the audio recording is played successfully.

**Test Data**: ABCD Song

**Expected Result:** The method should play the audio recording from the database without errors.

**Post Condition:** None

**Actual Result:** Successful

**Status:** Pass

**Test Case ID: AL-TC-004**

**Test Case Description**: Integration test for AudioRecorderPlayerModule.java and AudioLibrary.

**Pre-Conditions:** None

**Test Steps:**

1. Run AudioRecorderPlayerModule.java.
2. Choose the option to record audio and provide a name and duration.
3. Choose the option to play recorded audio.
4. Verify that the recorded audio is played successfully.

**Test Data:** Name, duration

**Expected Result:** Integration test should pass, and recorded audio should be played without errors.

**Post Condition**: None

**Actual Result:** Successful

**Status**: Pass

**Test Case ID: AL-TC-005**

**Test Case Description:** Integration test for AudioRecorderPlayerModule.py and AudioLibrary.

**Pre-Conditions:** None

**Test Steps:**

1. Run AudioRecorderPlayerModule.py.
2. Choose the option to record audio and provide a name and duration.
3. Choose the option to play recorded audio.
4. Verify that the recorded audio is played successfully.

**Test Data:** Sample name, duration

**Expected Result:** Integration test should pass, and recorded audio should be played without errors.

**Post Condition:** None

**Actual Result:** Successful

**Status:** Pass