**Testing Document**

**Overall Test Plan**

Our plan for testing is to do a lot of edge case testing because that’s what children do. We will be going through normal procedure but clicking in places most people would not, to see if anything breaks. The amount of UI on the screen at one time should be minimal to prevent this from happening.

**Test Case Descriptions**

*UI1* - UI Test 1

This will test the functionality of the UI on the main menu.

This test will make sure all the main menu buttons are functioning properly.

This test will make sure the buttons load into the correct pages and can properly close the application.

Inputs: Button press.

Outputs: The application will load the correct page.

Normal

Whitebox

Functional

Integration  
Result – To test this we opened the application and went to the main menu, from the main menu we pressed each button and the application correctly went to the expected page.

*UI2 –* UI Test 2

This will test the functionality of the UI on the music page.

This test will make sure all the buttons are functioning properly.

This test will make sure the buttons will load the correct pages to return to the main menu or save the music.

Inputs: Button press.

Outputs: The application will load the correct page.

Normal

Whitebox

Functional

Integration

Result – To test this functionality we navigated to the create page and pressed each button, they all successfully tested and went or did the correct operation

*UI3 –* UI Test 3

This will test the functionality of the Pause function on the music page.

This will ensure that the music stops playing and can be returned to.

This will make sure the pause button correctly pauses the music and when resumed does not disrupt the workflow of the user.

Inputs: Button press, Key press.

Outputs: Pause menu appears.

Normal

Functional

Integration.

Result – To test this we opened the create page and started writing music, we pressed pause to make sure the music stopped scrolling and again pressed space bar to start scrolling the music again

Usage Test 1

Make sure that the time will stop after five minutes so the song cannot be made to go on forever

We will make it last five minutes and stop after that time

We need to do this because if the song can go on forever the files will be too large to save, and we do not want that

Inputs: User clicking buttons to put in a song

Outputs: Song only plays for limited time of five minutes and will not register any notes after.

Normal

Performance

Integration

Deprecated – We got around this issue by design changes

Usage Test 2

Make sure that when multiple buttons are clicked that only one is registered on the staff

We want only one note to register if multiple buttons are clicked because we do not want chords and we need to make sur that multiple clicked buttons do not break the application

We want to make it simple for us and simple for the user, so this turns into a case that needs to be tested

Inputs: User clicking multiple buttons

Outputs: One note is put onto the staff instead many, or a chord

Abnormal

Functional

Integration

Result – To test this we went into the create mode and started writing a song, from there whenever there is a place to store a note we pressed multiple buttons to make sure only one note went there

Unit Test 1

Make sure loading button works

When a song is created, we want songs to be loaded correctly with no errors

We want to make the transition between loading page and song play page flawless, so the it easy for the user

Input: Loading button clicked

Output: Correct song is loaded onto staff

Normal

Functional

Unit

Result – To test this functionality first we went into the create functionality and created a simple song, we then saved the song as a file. We went back to the main menu and loaded into the Load section. We select the file we created and ensured the playback was correct.

Unit Test 2

Make sure save button works

When a song is created, we want songs to be saved correctly with no errors

We want the user to be able to save their own works of music, so that they can be accessed later and played again

Input: save button clicked

Output: Correct song is saved with the application

Normal

Functional

Unit

Result – To test this functionality we went into the create page and created a simple song, we then test the save function to make sure there is no error. To ensure it was saved correctly we can then load the song into the Load Page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UI 1 | Normal/Abnormal | | Blackbox/Whitebox | Functional/Performance | Unit/Integration |
| UI 2 | Normal | | Whitebox | Functional | Integration |
| UI 3 | Normal | | Whitebox | Functional | Integration |
| Unit 1 | Normal | | Whitebox | Functional | Unit |
| Unit 2 | Normal | | Whitebox | Functional | Unit |
| Usage 1 | | Normal | Whitebox | Performance | Integration |
| Usage 2 | | Abnormal | Whitebox | Functional | Integration |
|  |  | |  |  |  |