HowlonDemand Black Box Test Plan

Document Author(s): Kavit Patel

Date: Fall 2017

Introduction

HowlonDemand is an on-demand music streaming service from North Carolina State University. Streaming media is multimedia that is being continuously provided to the user in small data files. Different from downloading (where the user only has access to the media file upon completion), streaming delivers smaller data files that can be played before while another data file collected to be played later (known as buffering). Buffering allows the system to experience drops in network connectivity without pausing the music file. Likewise, musical tracks are often organized into playlists, or stations, to allow for easy transitioning from one song to the next. This simplified version of the steaming media handling process is modeled using a Finite State Machine.

The program is started by running the HowlonDemandUI class (as shown in the Design Proposal). If using Eclipse, first create a Run Configuration for the project by right clicking on HowlonDemandUI and selecting Run > Run Configuration. Select the new configuration button in the upper left, Browse for the HowlonDemand project. The main class is edu.ncsu.csc216.HowlonDemand.HowlonDemandlUI. Under the Arguments tab in the Run Configuration window. The assumption is that all files will be stored under the top level of the Eclipse project. If so, you only need to enter the file name into the Program arguments text field.

Test ID	Description	Expected Results	Actual Results

loadGUI (Kavit Patel)	**Preconditions: Click on HowlOnDemand GUI.**
	
	Right click on the HowlOnDemand GUI and then press the correct project and select the Run button.
	
	Check results.
	
	Close GUI
		HowlOnDemand loads
		
		`Following 9 stations will load:`
		`Rainy Day`
		`Sunny Day`
		`Driving`
		`Studying`
		`Friday Night`
		`Wake up Groove`
		`Office DJ`
		`Sing-a-longs`
		`Something Different`
			`Following 9`
			`stations will load:`
			`Rainy Day`
			`Sunny Day`
			`Driving`
			`Studying`
			`Friday Night`
			`Wake up Groove`
			`Office DJ`
			`Sing-a-longs`
			`Something Different`

Select Station (Kavit Patel)	**Preconditions: Test 1 passed.** Select a specified Station from the HowlOnDemandUI and 9 different choices. Select a Station. Station plays music.	Console output: **Loads the selected Station and plays music.** Station loads after executing given command.	**Loads the selected Station and plays music.** Station loads after executing given command.

audioStopped (Kavit Patel)	**Preconditions: Tests 1 & 2 have passed.** Loaded audio stopped due to unable to load (buffer) or if user clicked on stop. Message: Unable to play Check output Close	Decrypted Text: `Unable to play` Program ends execution.	Decrypted Text: `Unable to play`
stopSong (Kavit Patel)	**Preconditions: Tests 1, 2 & 3 Pass** Press stop button while playing a song to halt .	Console output: Stop Program ends execution	Program ends Console output: Stop Program ends execution

nextSong (Kavit Patel)	**Preconditions: Tests 1, 2, 3 & 4 pass** Press next to play the next song in the current station. Plays next song	Expected: Next song in the station is played.	Next song in the station is played.

Document Revision History

Date	Author	Change Description
11/2/2017	Kavit Patel	• Wrote loadGUI
11/2/2017	Kavit Patel	• Added Selection State, audioStopped, stop song, next song