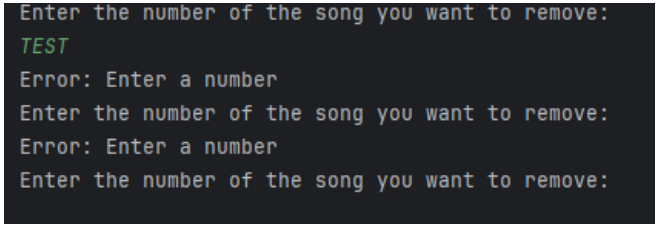
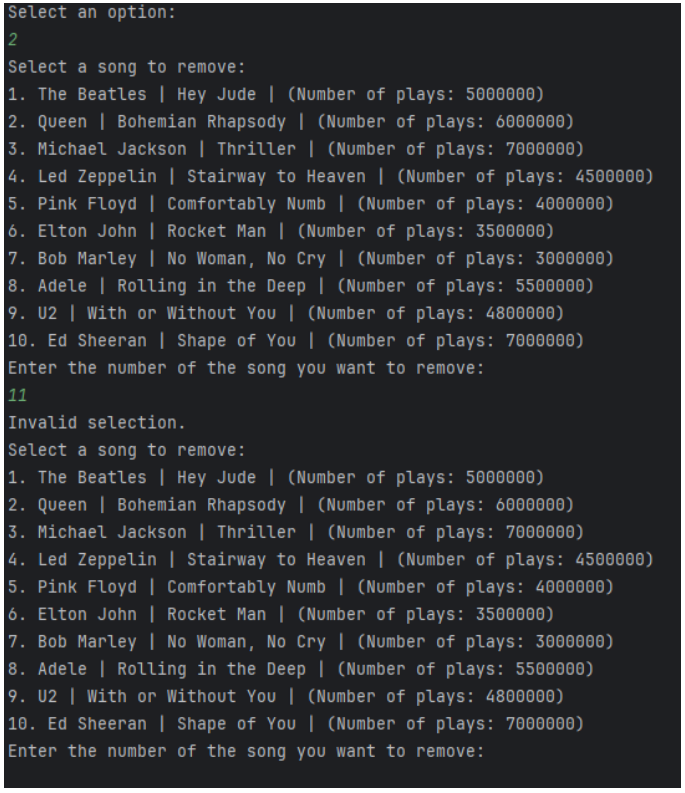
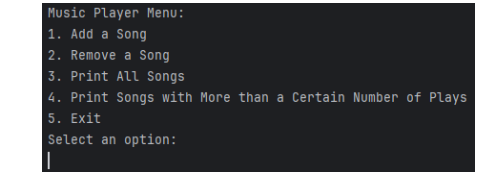
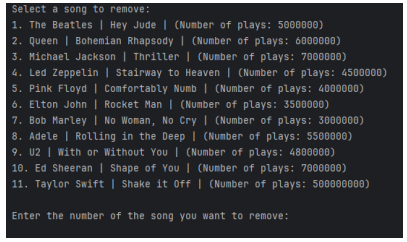
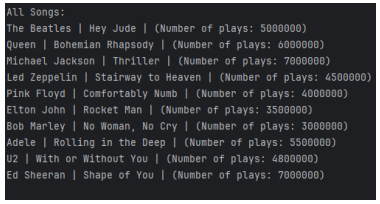
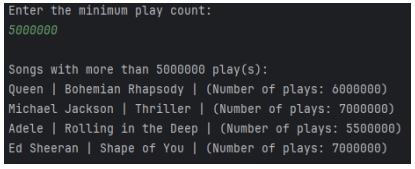
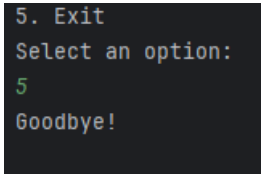
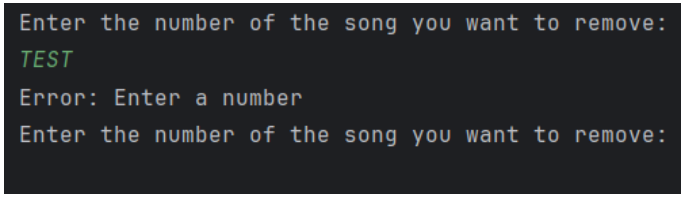
							IF OUTCOME ISN'T WHAT EXPECTED	
Test Case #	**Related Test Case #**	**Test Case Type**	**Description**	**Test Value**	**Expected Outcome**	**Pass / Fail**	**Screenshot**	**Comments**
#01	#05	Input Validation	In this test I will be testing each value displayed from the menu to the user	1 - 5	This should **PASS**, I have set up a "switch" with only 5 cases. If the user input doesn't fall into any of these, it will default to the error message	**PASS**		
#02		Input Validation	In this test I will be testing the number input for number of plays when the user wants to add a new song to the playlist	TEST	I expect this to **FAIL**, as the validation I have in place should prevent anything apart from an integer being inputted	**PASS**		
#03	#11	Input Validation	In this test I will be testing the input for when the user wants to remove songs from the playlist. If they try typing the name of the song, they should get an error saying "Error: Enter a number"	TEST	I expect this to **FAIL**, as the validation I have in place should prevent anything apart from an integer being inputted	**PASS / FAIL**		The program picked up that it was not an integer and presented the error message, however it repeated the error message twice before letting the user try again
#04		Functionality	In this test I will be testing the basic functionality of the program. I	N /A	I expect this to **PASS**	**PASS**		

			will run through the entire program and use it in a way to avoid any errors.					
#05	#01	Error Handling	In this test, I am testing how the program would react if the user were to input a different data type	testError	I expect this to **PASS**	**PASS**		
#06	#07	Error Handling	In this test, I am going to test how the program reacts to when the user tries removing a song number that is not there (If there is only 10 songs and the user enters 11)	11	I expect this to **PASS**	**FAIL**	This didn't actually fail, it just did not behave how I wanted it to. When the user inputs an invalid number, it returns them to the main menu. I want the program to ask the user again which song they want to remove.	
#07	#06	Error Handling	This is an extension off of Test Case #06. I have added a loop into case 2 and moved the break statement. So now the program should keep asking the user to remove a song until they enter a correct number	11	I expect this to **PASS**	**FAIL**	This gave me the same result as before. Looking at the code now, I can see that I was missing a break statement to come out of case 2. I will add that in now and test.	

#08	#06	Error Handling	Another extension from Test Case #06. I have added in an extra break statement to see if that will loop case 2 the way I want it to.	11	I expect this to **PASS**	**PASS**		This now behaves the way I intended it to do so in the first place.
#09		Compatibility	The IDE that I used was IntelliJ, so I will test the program in NetBeans as well.	N / A	I expect this to **PASS**	**PASS**		

#10		Usability	In this test I will run through every menu that can be displayed to the user, and make sure that they are all easy to understand and information is clear.		I expect all menus to be clear and for this to **PASS**	**PASS**	    	All menus here can easily be understood and information is clearly stated.

#11	#03	Input Validation / Error Handling	This is an extension from Test Case #03. I have added a new line that I hope will fix my issue of repeating the error message multiple times	TEST	I expect this to **PASS**	**PASS**		The program now behaves exactly as I want it to.