

Theresa Pham

CSC 415: Software Engineering

Assignment 4

Due. 11/4/16

### Test Case Design

#### *Functional Testing:*

In order to ensure the program is developed successfully, it is important to perform all types of functional testing throughout the development lifecycle to make sure individual modules and their interfacing are able to function properly. For unit testing, statement testing will be used as I will be learning many new languages and tools for the first time, so making sure each line works will be important. Input domain testing will also be done to ensure that the parameters a module needs can be passed as an input, otherwise the application will not be able to accomplish its intended functionalities. since albums, countries, and lists are essential parameters that must be passed between classes in order for most controls to work in Spotify WorldView.

For integration testing, the top-down approach will be used as for this project, as some of the user interface already exists and I have a clear idea of where I want different functionalities to go in the system. I may add or change some of the functionalities as I encounter challenges or find new ways to make the program better, but in general, I will be able to work from the main page down to the individual use cases within the system. This will also ensure that the map is always

able to function, as the map is the biggest proponent of the application and is at the top of the system.

For system testing, the black box method is used most often, and will work for this project as well, as once the application is complete I will be able to test the application as if I am a user and test all possible cases that could be encountered. All the controls available in the system are fairly easy to reach, and users will not be straying much from the main page at all. A list of test cases that I will be testing is listed below in the table.

*Test Cases*

<b>Functionality Tested</b>	<b>Inputs</b>	<b>Expected Outputs</b>	<b>Actual Outputs</b>
Find album by URI	Album URI	Album found + album name & artist displayed	
Search album by keyword	Keywords (strings)	List of possible albums the user wants to search	
Display highlighted countries on map	Album (URI) or clicking link after searching by keyword	Highlighted countries where album is available	
Show tracklist of a searched album	Keyword search + Album selection	List of songs in album	
Error message display	Invalid album	Map grayed + error message displayed	
Drag album from Spotify app to map interface	Album cover dragged	Album found + name, artist, and highlighted countries displayed	

I also plan to have other people attempt to use to program to see if they are able to understand the program without assistance and perform any functions that they can think of that I may not have been able to. It will help me account for new cases I may have missed or adjust the user interface based on feedback.

#### *Testing/Debugging Tools:*

Because Eclipse includes a way to import GitHub project, I will use JUnit to run various types of unit and integration testing. JUnit will give me a way to view my individual modules and interfaced modules as I develop my application and will also provide a method of debugging as Eclipse provides feedback based on the tests done.