EECS 2311

VENN DIAGRAM APPLICATION

Software Testing Document

By: Mahamed Hassan - 216536476

Thanh Nguyen - 213507231

Esra Kastrati - 215507205

Group 10

Testing Document

Test cases implemented:

The documentation had numerous circumstances for the client to have the option to edit and change data. Due to the needs of our client, we had to modify our Venn app to their likings, and extra strategies had to be made to reach them. Our major modifications include giving the client an option to name the sets whatever they desire and being able to remove items from the set, in case they decide to edit an item out.

Most of the UI was unable to be tested with JUnit tests, so we focused our testing of the backend of the application. We tested LabelSerializable to ensure the right data was being saved. The tests on Command and CommandManager ensure that Undo/Redo operations perform successfully every time.

Why test cases are sufficient:

The test cases were used to make the application user-friendly and satisfy the client's needs. For adding elements to a set case, the client needs the ability to add items to the set of their choice. Secondly, the client wanted an option to rename the sets to anything they wanted. Finally, the client mainly wanted to remove elements from the sets in case they changed their mind. For these test cases to be sufficient enough to fulfill the clients' needs it must be able to compile with the program. It is important to check that the test cases worked to make sure the program functions as expected.

The test cases done ensure that when any new element is added, or an existing element is modified or deleted, that change in the application will reflected in the backend. Therefore, we are able to save and reload to any previous state.

How Test Cases were derived:

Test Case #	Test Case Description	Test Data	Expected Result
1	Adding elements to a set, we tested to check if the element would be added to the correct set as commanded. Below the sets, there is a textbox to type and then click on add to add the item to whatever set you have chosen.	Click on the right circle. The set should be highlighted with a bold line. Enter “Basketball” into the textbox. Press Enter or click Add.	Right set should now include “Basketball” with a message written on the console.
2	Renaming sets, we tested to see if the sets would be renamed once it was instructed to. Right-click on the title of a set and an option to rename the set should appear	Right-click on “Right Set”. Click rename. A small window will open. Enter “Set1” into the textbox. Press Enter or click Ok.	The set’s label should now change from “Right Set “, to “Set1”
3	Removing an element from a set, we tested to see if the element would be removed once commanded to. Right-click on the element and click on remove, the element should now be removed from the set as instructed.	Right-click on “Basketball”. Click “Delete. A small confirmation window will open. Press Enter or click Ok.	“Basketball” should now be removed from the right set.

Test coverage conclusion:

The most important thing about these test cases is to describe the modifying abilities that the client can do. The main modifications, for now, is to give the client the ability to add elements, change the label's name and remove elements from the desired set. Tests are implemented and we measured coverage of more than 70% using "EclEmma". JavaFXMain.java class (94.5% coverage) and FXController.java class (76.4% coverage). Since we could not test the GUI automatically, this was done via manual testing.

The JUnit tests that we were able to perform, show good coverage for those tests. Since most of the code is written in the controller class for the GUI, it lowers the overall overage.