Testing

1. Coverage test:

During the test the coverage. We get approximately 48.6% coverage of our total code. The code need to be simplified after finishing all the code.

1.1 The main part of the code is application package. It has half coverage. Some method maybe never used.

1.2 The database and login packages are created for Login function which is still in construction. So the coverage is nearly 30%.

1.3 Venn package is for lab mission. So it never used locally.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Venn	48.6 %	2,573	2,718	5,291
src/main/java	48.7 %	2,573	2,706	5,279
application	48.0 %	2,038	2,207	4,245
MainController.java	48.0 %	1,995	2,164	4,159
MainController	48.0 %	1,995	2,164	4,159
Main.java	50.0 %	43	43	86
Main	50.0 %	43	43	86
database	30.2 %	151	349	500
Venn.java	0.0 %	0	204	204
Category.java	0.0 %	0	83	83
Item.java	0.0 %	0	52	52
User.java	89.5 %	34	4	38
AccSys.java	96.9 %	93	3	96
SuperAcc.java	62.5 %	5	3	8
Account.java	100.0 %	19	0	19
login	76.2 %	384	120	504
ComfirmBox.java	0.0 %	0	87	87
Login.java	91.5 %	321	30	351
AlertBox.java	95.5 %	63	3	66
venn	0.0 %	0	30	30
Test.java	0.0 %	0	25	25
Library.java	0.0 %	0	5	5
src/test/java	0.0 %	0	12	12

2. Function Test

Testing our application in various ways will ensure that our program functions as it should.. There needs to be methods in which we test our program and also have properly defined expected outputs. The tests cover all our use cases and are mentioned below.

1) Text additions and info overflow

Steps:

- Repeatedly drag and drop text boxes as well as images into the areas of the venn diagram

Expected behaviour:

- Text is dynamically sized proportional to the number of elements and the average length of each element

Actual behaviour:

- The list is not dynamically sized.
- When tester tried to import a large amount of data, the list will show a roller bar that allows the customers to adjust the items.

Test result:

- App easily deals with the large amount data importance situation.
- But we need a new implementation to address the long size string problem.

2) Inputting data

Steps:

- Prepare multiple text files each containing strings of multiple lengths and formats
- Each text file should be imported individually
- After each file has been tested, import multiple files in succession

Expected behaviour:

- The text should be represented in the diagram exactly as is described in the inputted text files
- When multiple files are loaded in succession, the new data is appended to the existing diagram with no conflicts or overlaps

Actual behaviour:

- When import txt file is consistent with the standard export report. The system works properly.
- But when trying to import some invalid txt file, the system didn't work.

Test result:

- The inputting data works fine. But we need a alert box that remind the user to use a certain format input file.

3) Exporting data

Steps:

- Drag and drop multiple text strings and images to the diagram, making sure to occupy all sections of the diagram
- Click an Export button that would have the option to either export as a text file or an image

Expected Behaviour:

- Exporting the data as either a text file or image preserves all information placed by the user
- The exported text files are formatted in a clear and concise way so that users can distinguish between each set
- The exported images capture the full scope of the diagram, while also not capturing irrelevant things like menu buttons

Actual behaviour:

- Exporting the data as either a text file or image preserves all information placed by the user. The exported text files are formatted in a clear and concise way so that users can distinguish between each set
- The exported images capture the full scope of the diagram, while also not capturing irrelevant things like menu buttons
- The data in the input box is left behind. Maybe user need them to show some characteristic of Venn diagram.

Test result:

- Both exporting methods work properly.
- Need to show those items that are unclassified.

4) Menu bar

Steps:

- Use the menu bar to modify any attribute

Expected Behaviour:

- Every button on the menu bar works as they expected.

Actual behaviour:

- The color change choice is not working unless the correct text were imported.

Test result:

- The menu bar is working properly. But there should be a reminder when the user tries to change the color of any set that what kind of text they should use to change the color.

5) Venn diagram function

Steps:

- Put some data into different test.
- Test the right click to check if the move function works.
- Test if it identify the same text and remove them into intersection set.
- Test the label name can be left click to modify.

Expected Behaviour:

- All items can be move no matter how many data was input.
- When the set on the side have the same item, system remove them both and add it into intersection.
- Label can be modified directly no matter how long the text is.

Actual behaviour:

- All items can be move no matter how many data was input.
- System can detect the identical items and deal with them properly.
- When the input label name is too long. The two label may be overlapped on the screen.
- After click the label and click somewhere else, the textbox is still focused

Test result:

- Need to tell the customer that the input name has a maximum length.
- Need to fix the focus problem.