

Hasan Al-Habbobi 216315428
Sharujan Rajakumar 216376410
Alain Ballen 216341703
Ahmed Hagi 215043896

REQUIREMENTS DOCUMENT - GROUP 5

Purpose:

The system allows the user to organize information between various ideas, products, etc in such a way that they can make comparisons more easily. This in turn satisfies the client's/ customer's need of putting abstract ideas into the picture. This need can be broken up into many needs which are outlined and described below as essential features the system will do to satisfy the customer.

Essential Features:

Feature	Description
Resizable Diagrams/Components (whenever an entry is added)	<p>Allow the customer to change the size of the whole diagram or individual components (e.g. titles, shapes, number of circles, etc)</p> <p>We incorporated this feature by having a JButton called <i>Add Inputs</i>, which enables the customer to enter the dimensions of the circles for the venn diagram. This allows the client to change the size according to their preference.</p>
Clear Functions (clear all and put all entries back into the field)	<p>Allow the customer to reset their work; this is so that the customer does not need to remove each component of the diagram one at a time. They have the ability to reset the whole diagram at once and start from scratch.</p> <p>We created a JButton called <i>hide</i>, to hide the venn diagram that was initially made.</p>
Add/Change/Delete Multiple Entries in groups at once (kind of like adding an array to an array)	<p>Allows the customer to again save time by being able to convert a premade list into elements added in the Venn diagram.</p>
Save/Print Venn diagram (as PDF, PNG, etc)	<p>Allows user to save their Venn Diagram to their computer so that they can share it to others.</p>
Use Different Shapes, bullets, colors, fonts, etc	<p>Gives the customer some freedom to customize the Venn Diagram to suit their appearance taste.</p>
Sets: Intersection and Unions Functions	<p>Allows user to take two lists and automatically assign common denominators between the lists in the middle. This goes hand in hand with the Add Multiple Entries feature explained above.</p>
Ability to open existing files	<p>Give the ability to the customer to open any existing diagrams they have made in the past.</p>

	They can make any changes to the existing file if needed. The file will have its own extension.
User suggestion box (for user to comment suggestions for developers to add features in updates) + Error Handler	<p>Allows customers to suggest features and ideas the developers may have missed or not considered as essential features. Ultimately no one is a better judge than the customer with regards to what they want. This gives customers a voice/say into how they shape the Venn Diagram app to conform to their needs rather than the customer having to conform to the demands of the system.</p> <p>Additionally if there is an unexpected error the developers didn't catch it can be reported so that it can be handled and fixed as soon as possible.</p>

Important Use Cases:

The app can be used to compare or find similarities between products, strategies and plans in a variety of different use cases (outlined below).

Use Case	Explanation & Specific Examples
Math and Science	Venn Diagrams in general are used to enable students to visualize and organize information to see various relationships among sets, or groups of objects.
Statistics and Probability	A visual comparison of probabilities of different events occurring based on the probability of their factors being met, etc
Logic	It can be used to determine the validity of various arguments and conclusions by comparing their deductive reasoning.
Linguistics	Planning between two different Trip Routes. Which one is the most efficient vs. most fun
Computer Science	Comparing multiple algorithms and seeing which one will satisfy more of the users important needs. Also, to determine which algorithm is efficient and works best.
Business	Comparison between different products/services to determine which one of the two is the better option.

Acceptance Test Cases:

The user can implement a feedback forum to give customers the opportunity to fix any problems or updates features regarding the venn diagram app. This ensures that the customer is comfortable and can easily interact with the app.

Below are key important acceptance cases addressing questions including but not limited to:

- How do we indicate whether the app can see the user needs outlined above in Essential Features?
- How can you determine if requirements are met?

These acceptance cases are also useful to have a standardized method of dealing with errors/issues so that they can be better understood and solved by the developers.

Acceptance Case	Steps to Undertake	Expected Behavior of the System
Error/Exception Handling <i>(Tests if Errors/Exceptions can be handled instead of crashing the program)</i>	<p>The handler determines what kind of error it is by (is it caught by an exception handler? Or is it an error that has never before seen/anticipated?)</p> <p>If it's the latter then it will be handled by its respective handler.</p> <p>If it's the former then it will automatically display a message to the user to report the error to the development team so that it can be handled and fixed by the developers. This way an update can be issued so that other customers don't get the same error or that the exception can be better handled.</p>	<p>When a specific error occurs with the app, the system will catch and handle it. If not then the user can report that error in the feedback forum</p>
Usability Testing (How to test if app is easy to use)	<p>First the user tries out the app and will discover obstacles (things that make app tedious to use or at least relative to how much smoother and seamless or</p>	<p>The system is expected to allow the user to use the application version so that the customer can give qualitative feedback as to whether the app is friendly and easy to</p>

<p>User Emails Suggestions (suggestions for developers to add features in updates) + Error Handler</p>	<p>more convenient it could be)</p> <p>Next the user reports those obstacles and the testing (strategy) they did to reach that problem.</p> <p>Developers get the report and treat it similar to an exception handling function.</p>	<p>use.</p> <p>The system is also expected to allow the user to develop a testing strategy to ensure that all functions of the app work. These include navigation and content. Hence, the user can analyse the results and improve the app accordingly by submitted their suggestions for improvement.</p> <p>This is also related to Error Handling as the user can point out errors with the problem.</p>
<p>Quality Testing (How to test that there aren't long delays between pressing of button and the action it is designed to do)</p>	<p>Create a separate copy of the program which has timers embedded in it. (A timer for each button)</p> <p>Steps:</p> <ol style="list-style-type: none"> 1) Press button 2) Timer Starts 3) Action is done 4) Timer stops and is displayed <p>Based on the task determine if the time is reasonable or not.</p> <p>If the time is reasonable then move on to the next button.</p> <p>If the time is not reasonable then look into the action performed. For example, are there too many calls being made. Is the algorithm run time too long and inefficient.</p> <p>Developers will re-evaluate the button or worse case re-develop it if the delay is severely bad.</p>	<p>Upon a button being pressed the system will start a timer. The timer is expected to only stop when the action a particular button is supposed to do is completed.</p> <p>(Example: if Create a title is pressed then the system will calculate the time from pressing that button to the title actually being created)</p>
<p>Functional Testing/Unit Testing</p>	<p>The user should identify all of the functions that the app</p>	<p>The system is expected to pass all JUnit tests for each</p>

(How to test if items can be resized or if multiple entries can be added/deleted?)	is needed to perform, and create input data based on the functions specifications. And, then determine the output based on the functions specifications and execute a test case. Finally, compare the actual and expected results.	function.
How to test if user suggestion box is working	<p>Sample User clicks button to make complaint/suggestion</p> <p>User types their name, email and complaint or suggestion then clicks send button</p> <p>Send button triggers the sending the above information in the form of an email to the developers</p> <p>If the Developers receive the email then the suggestion box is working. Otherwise it needs to be fixed</p>	The system is expected to be able to pass on all user suggestions/comments to the developers email inbox.
Resizable Diagrams/Components	The user enters the input for the dimensions of the venn diagram by using the <i>Add Inputs</i> button that was implemented. Then, they click the <i>create the Diagram</i> button and it resizes the diagram for the client.	The system is expected to adjust the Venn diagram's circle size, title size, etc. according to the values in the <i>Add inputs</i> window before clicking/pressing the "Ok" button.
Clear Functions (clear all and put all entries back into the field)	The user has clicked/pressed the clear button.	The system is expected to allow the user to clear the entire diagram so that the customer doesn't have to manually delete everything one by one.
How to test if user can save/print Venn diagram (as PDF, PNG, etc)	The user can click a "Save" button which when clicked downloads a pdf or png (users choice) of the venn diagram.	The system is expected to allow the user to download a pdf or png file of the Venn Diagram they have created. By doing so the user can then share or store that file should

		they wish.
How to test if user can Use Different Shapes, bullets, colors, fonts, etc	<p>The user selects a unique shape from the "Shape" dropdown menu.</p> <p>The user selects a unique font style from the "Font" dropdown menu.</p> <p>The user selects a unique color for either left and or right circle from the "Color" dropdown menu.</p>	The system is expected to have the same shape, font style and or colors of the circles in the Venn diagram according to the selected items that user inputed in the "Shape", "Font" and "Color" dropdown menus.
How to Test if User can upload lists which automatically get their common elements combined in	<p>The User selects "Upload List(s)"</p> <p>The user pastes the list(s) in the textbox then clicks upload</p> <p>The system converts each list into an item and follows the sets protocol below if multiple lists are uploaded</p>	The system is expected to allow the user to upload multiple lists which then get automatically converted to elements on the diagram.
Sets: Intersection Functions	<p>Any elements that are found in multiple lists are placed in the middle intersection part of the venn diagram.</p> <p>This is done by scanning the elements in a list then scanning the elements in another list comparing each element in list 2 to each element in list 1.</p> <p>When a match is found the system will remove the element from both lists and insert it into an intersection column of the two lists.</p>	The system is also expected to intelligently figure out if lists share common elements in which case they should be placed in the middle of the venn diagram.
How to test if the user can successfully open an existing file	The user will be given the option to open up the specific files with a certain extension. This option will be given through a button.	The system is expected to save their venn diagram as a certain file type. This file type will be the only editable file.