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

REQUIREMENTS DOCUMENT - GROUP 5

Purpose:

This document describes the needs of the customer as well as all required features of this application to satisfy those needs. Additionally the document provides acceptance cases, including steps to undertake and the expected behavior of the system.

Table of Contents

---

1 = Essential Features / User Needs:

The system allows the user to organize information between various ideas, products, etc in such a way that they can easily make comparisons. 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.

Feature	Description
Resizable Diagrams/Components (whenever an entry is added)	Allow the customer to change the size of the diagram circles/containers or the size of text elements
Clear Functions (clear/remove all text entries/elements)	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. Note: System displays pop up warning when clear function is clicked so that user doesn't remove all text elements by accident
Add/Change/Delete Multiple Entries in groups at once (kind of like adding an array to an array)	Allows the customer to again save time by being able to convert a premade list into elements added in the Venn diagram.

Download Venn diagram (as PDF/PNG)	Allows the user to save their Venn Diagram as a PNG image or PDF file to their computer so that they can share it with others.
Use Different Shapes, bullets, colors, fonts, etc	Gives the customer some freedom to customize the Venn Diagram to suit their appearance taste.
Sets: Intersection and Unions Functions	Allows the 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.
Ability to open existing files	Give the ability to the customer to open any existing diagrams they have made in the past. They can make any changes to the existing file if needed. The file will have its own extension.
Error Handler + **User Can Email suggestions/concerns (to the developers to add/fix features in future updates)**	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. Additionally if there is an unexpected error the developers didn't catch it can be reported (via an email to vennsolutions@gmail.com) so that it can be handled and fixed as soon as possible.

2 = 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	Comparing the similarities and differences of two different languages
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.

3 = Acceptance Test Cases:

The user can email any feedback to vennsolutions@gmail.com via to give developers the opportunity to fix any problems or update features regarding the venn diagram app. This ensures that the app is future/bullet proof for the user/customer.

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 *(Tests if Errors/Exceptions can be handled instead of crashing the program)*	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?) If it is the latter then it will be handled by its respective handler.	When a specific error occurs with the app, the system will catch and handle it. If not then the user can report that error by emailing their concern to vennsolutions@gmail.com In the future a suggestion box will be implemented to

	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.	make it easier for users to transfer their suggestions to developers.
Usability Testing (How to test if app is easy to use) **+ Error Handler** **+ User Can email suggestions/concerns to the developers to add/fix in future updates)**	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 more convenient it could be) If the user ends up experiencing issues/difficulties with the program they can click the help button.	The system is expected to notify the user of the vennsolutions@gmail.com email which the user can send emails to to report on various suggestions/concerns with the program 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 submitting their suggestions for improvement. This is also related to Error Handling as the user can point out errors with the problem.
Quality Testing (How to test that there aren't long delays between pressing of button and the action it is designed to do)	Create a separate copy of the program which has timers embedded in it. (A timer for each button) Steps: 1) Press button 2) Timer Starts 3) Action is done 4) Timer stops and is displayed	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. (Example: if Add Text is pressed then the system will calculate the time from

	Based on the task determine if the time is reasonable or not. If the time is reasonable then move on to the next button. 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. Developers will re-evaluate the button or worse case re-develop it if the delay is severely bad.	pressing that button to the Text actually being created/added)
Functional Testing/Unit Testing (How to test if text elements can be resized or if multiple entries can be added/edited/deleted?)	The user should identify all of the functions that the app is needed to perform, and create input data based on the functions specifications. The user can then determine the output based on the functions specifications and execute a test case. Finally, compare the actual and expected results. Users should first click “add text” to add multiple text elements. The users should also be able to drag out the text to wherever they like in the application (outside of the left panel area of course) Next the user when double clicking on a text element should be able to change its text as well as its font, size and color using the menu panels to the left	The system is expected to pass all JUnit tests for each function.

Resizable Diagrams/Components	The user can move the container slider (by left clicking their mouse button and dragging it to either the left or right). If the size of the containers increase when the slider is moved to the right and decrease when the slider is moved to the left then the container resizing function works	The system is expected to adjust the Venn diagram's circle size, title size, etc. according to the values in the *Add inputs* window before clicking/pressing the "Ok" button.
Reset Function (clear all text elements/entries from diagram 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	The user can click the Left Container color button and can select the desired color for it. And Or The user can click the Right Container color button and can select the desired color for it.	The system is expected to allow the user to change each container to its own independent/individual color to any of the color options provided in the color picker menu.
How to test if the user can successfully open an existing project	The user will click "Open Project". Upon doing so they will be prompted to click browse and search for an existing project. When they find it they should double left click on it (with their mouse)	The system is expected to open the existing project the user chose to open which show the containers and text elements in their same sizes, colors, positions as when the project was last saved. If this happens then the program has successfully opened an existing project.