
Software Requirements Specification

EECS 2311

VennEver

Version 2.0

Prepared by Group 19

Muneer Nekiar

Sidharth Sudarsan

Areeba Abidi

Indermohan Gill

11/4/2020

1. Introduction

1.1 Purpose

Venn diagrams display logical relations between a collection of sets. This software will be displaying the relationship between different sets of data. The software displays the union and the intersection between the datasets. This software is expected to organize information visually for the users.

1.2 Intended Audience and Reading Suggestions

The application is intended for organizing data. The users for this application can be children in school, managers, instructors, and any personnel that is required to visually represent relations of different sets. They should read features of the application for optimal use. The developers that want to expand on this application should read about the system features along with interface requirements. Testers should investigate interface requirements and other non-functional requirements.

1.3 Product Scope

The functionality of the software is to create a Venn diagram for the user, upon receiving required input by the user. As the user enters in different sets of data, the data that is shared between the sets will appear in the overlapped shape. The goal is to create a user friendly and easy to use application. Further features are available such as being able to share the data with other individuals. Users can draw and change the color of the text and customize their diagrams. The user is also able to edit the diagram once the data has been added, change the size of the circles, and customize the text.

2. Overall Description

2.1 Product Functions

- FE-1: Customizable shapes - can be resized
- FE-2: Text customization - font, size, alignment
- FE-3: Intersection set automatically generate
- FE-4: Undo/Redo changes
- FE-5: Custom color schemes
- FE-6: VennTest Mode

2.2 Design and Implementation Constraints

CO-1: User interface shall be composed using javaFX library. Currently the application has two scenes. The main window that appears upon initially running the application, this is technically the home screen of the application. It contains the Venn diagram and has vertical and horizontal toolbars.

(This portion will have more information added to it upon further development of application.)

3. External Interface Requirements

3.1 User Interfaces

There is a vertical toolbar and a horizontal tool menu on the top-left side of the screen, both roughly resembling a shape that looks like an upside down 'L'. The vertical toolbar consists of features for sizing the diagram, adding text labels, and adding data to the sets. While the horizontal toolbar which is currently still being updated will contain information about undo/redo, import and export. After an option is chosen on the vertical toolbar, a second horizontal toolbar may appear depending on the option. This toolbar contains the main functions related to the option chosen. For example, if the user were to select "Customize Circles" on the vertical toolbar, the new horizontal toolbar contains the size slider and the colour picker options for either circle.

Horizontal toolbar: Text dropdown menu containing Import and Export, file options.

Vertical toolbar: option for adding data, VennTest mode, labels, formatting text, customizing shapes, undo, redo and clear all.

Second Horizontal toolbar: Main functions of the options chosen from the vertical toolbar.

The Diagram will appear in the middle of the screen, on a canvas.

A **second window** can appear upon clicking "Data". This window contains two text areas, labeled respectively "First Set" and "Second Set". On the top right of the window a toggle button exists that changes the layout of the application depending on the option picked.

4. System Features

In this section we will describe key features of the application.

4.1 Generating a Venn Diagram

4.1.1 Description and Priority

This is the main feature of the application which allows users to generate a Venn diagram including a visualization of the intersection and union of two sets of elements. As this is the main feature of the app it has high priority.

4.1.2 Stimulus/Response Sequences

The application will provide a basic template/visual of a basic Venn diagram upon initial start up of the application. It will have two circles joined together giving a basic layout of a Venn diagram.

4.1.3 Functional Requirements

REQ 1: This feature will be able to differentiate between the set elements.

REQ 2: It will be able to create new sets of the union of entered sets and a new set of the intersection of entered sets.

REQ 3: Feature will be able to create a Venn diagram visualization of elements of sets, users can edit the diagram at any point. Each textbox can be deleted by pressing 'ESC.' There are undo/redo options and at any point users can access the previous screen by pressing the back button.

4.2 Text Customization

4.2.1 Description and Priority

This feature will allow users to choose from a wide range of fonts and colors for their text used in the Venn diagram. This feature has med priority as we want users to have many custom options but as it doesn't affect the overall functionality of the application.

4.2.2 Stimulus/Response

The feature has a button that will have an editable textbox appear for the use. This textbox is draggable and can be placed anywhere on the canvas. The text is customizable, users can change the font, font size, bolden or italicize and can add a description to the textbox if needed.

4.2.3 Functional Requirements

REQ 1: This feature will be easy to use

REQ 2: This feature will have all fonts and colors available

REQ 3: This feature will apply customizations within rules of Venn diagram and not negatively affect the structure of the diagram.

4.3 Custom color schemes

4.3.1 Description and Priority

This feature will allow the user to customize the color of the shapes used in their Venn diagram. The priority level of this feature is medium due to the importance of having this feature in an application that creates fully customizable Venn diagrams but does not affect overall functionality of the application.

4.3.2 Stimulus/Response

Sizing of the circles along with the colors of the circles are customizable for the user. This is so that the user has more control over what the Venn diagram looks like.

4.3.3 Functional Requirements

REQ 1: Feature will be easily accessible and easy to use.

REQ 2: Feature will not affect overall structure of diagram.

REQ 3: Feature will have all wide range of colors available for the user to choose from.

4.4 Export as PDF/JPEG

4.4.1 Description and Priority

This feature will allow users to export their Venn diagram as a pdf or jpeg file. This feature has high priority as we want users to be able to save their Venn diagram onto their computer and use it how they feel fit, without this export feature the Venn diagram will remain only on the app.

4.4.2 Stimulus/Response

This feature will have a button that the user can click to export the file. A mini window will open on the screen with the available options for the user to fill in. Options include; name the file, choose file in which to save the diagram, choose between pdf or jpeg. The response of this action will be having a saved pdf/jpeg file of the created Venn diagram saved in the chosen file of the user.

4.4.3 Functional Requirements

REQ 1: Feature will have a button.

REQ 2: Feature will not affect overall application functionality

REQ 3: Feature will allow users to save file on name of their choosing in a file of their choosing

REQ 4: Feature will save file as is at the time the button is clicked

4.5 Large data set turn into numbers

4.5.1 Description and Priority

This feature will allow customers to know the number of elements in each set they provide which is useful when a user has many elements in their set. This feature has medium priority as it is important for users to know how many elements they have but this does not affect overall functionality of sets and Venn diagrams.

4.5.2 Stimulus/Response

This feature will have a toggle button which will open a mini window giving the user the option to use “number of elements” on their diagram instead of having every element as a single entry on the diagram. The response of the button will be to replace the elements in the visual diagram to the total number of elements in each set and vice versa.

4.5.3 Functional Requirements

REQ1: This feature will have a counter that counts every element

REQ2: Counter will differentiate between sets and count elements of sets separately

REQ3: This feature will allow users to easily switch between “Text View” and “Count View” Venn diagram mode.

5. Use Cases

Name of Use Case: Comparison
Description: Comparing two sets
Actors: User

Preconditions:

1. The union of the two shapes must be shown as an intersection of the two shapes.
2. The diagram must be able to identify if two texts are the same.
3. The diagram must be able to place one of the two same text boxes in the intersection of the shape and delete the extra textbox.

Postconditions:

1. Must be able to return the number of entries in A, B or the intersection.

Flow:

1. Two overlapping circles appear on the canvas.
2. "Data" button will take the user to a new window. With two text-areas, "First Set" and "Second Set", the user can add the values in each set.
3. Changes can be made to the diagram at any point.
4. The size and the color of the circles is editable as well.

Alternative Flows:

After Step 4

1. The user can add labels or titles to their shapes using the Label Button.

During Step 4

1. To avoid typing values the user knows will be in the intersection set two times, the user has the option to change the layout via the “intersection set input” button at the top of the window and choosing “enter explicitly”. This will change the layout so that there is a separate column for intersection. To change back, the user can click “compute from two sets” from the “intersection set input” button.

2. The user will be able to customize their text if inclined. The user will be able to customize font, colour, size and be able to highlight their text as well.

At any point

The user can undo their work and/or be able to delete the textboxes.

Requirements:

The following requirements must be met before execution of the use case:

1. All buttons, menu items and toolbar items must do their required function.
2. Must be able to customize the shapes (change colour or resize).
3. Must be able to customize text (change font, colour, size, bolden or italicize).
4. Must have an undo and redo button to fix changes.

Name of Use Case:	Comparing Phones and Laptops		
Created By:		Last Updated By:	
Date Created:		Last Revision Date:	
Description:	Comparing Smartphones and Laptops to figure out what is the better choice for the user using a Venn diagram with 2 circles.		
Actors:	User		
Preconditions:	<ol style="list-style-type: none"> 1. The shape (circle) appears duplicated and creates a diagram with both shapes that snap together. 2. The union of the two shapes must be shown as an intersection of the two shapes. 3. The diagram must be able to identify if two texts are the same. 		

Postconditions:	<ol style="list-style-type: none"> 1. Window must quit after clicking “Quit”. 2. Window must allow the user to clear the diagram they have made, so the user can create another diagram if required.
Flow:	<ol style="list-style-type: none"> 1. User will open up the application and will see two circles already generated. 2. User clicks on “Label” and editable labels pop-up, allowing the user to label the two circles, “Smartphones” & “Laptops”. The labels appear above the circles. 3. User clicks “Data” and a window pops up with 2 text-areas. 4. User sets values in the text-areas of the 2 sets. 5. Press done upon completion. 6. A diagram will be generated with the values in the appropriate locations for the user. 7. User repeats steps 3,4,5,6 until they are satisfied. 8. User is finished with the Venn diagram and closes the window.
Alternative Flows:	<p>Step 2 can be done at any point.</p> <ol style="list-style-type: none"> 1. User can click on “Label” and add labels to their Venn diagram to make it easier to follow which circle represents what. <p>After Step 3</p> <ol style="list-style-type: none"> 2. User have the option to add a third text-area for values that are known to be intersections of the two sets by changing to the “enter explicitly” from “intersection set input”.
Requirements:	<p>The following requirements must be met before execution of the use case</p> <ol style="list-style-type: none"> 1. All buttons, menu items and toolbar items must do their required function. 2. Must be able to customize the shapes (change colour or resize). 3. Must be able to customize text (change font, colour, size, bolden or italicize).

Name of Use Case:	Comparing Airbnb and Hotels		
Created By:		Last Updated By:	
Date Created:		Last Revision Date:	
Description:	Comparing Airbnb and Hotels to figure out what is the better choice using a Venn diagram with 2 circles.		
Actors:	User		
Preconditions:	<ol style="list-style-type: none"> 1. The circle must be able to duplicate and create a diagram with both shapes that snap together. 		

	<ol style="list-style-type: none"> 2. The union of the two shapes must be shown as an intersection of the two shapes. 3. The diagram must be able to place one of the two same text boxes in the intersection of the circles and delete extra text boxes.
Postconditions:	<ol style="list-style-type: none"> 1. Window must quit after hitting close on the application. 2. Window must allow the user to clear the diagram they have made, so the user can create another diagram if required.
Flow:	<ol style="list-style-type: none"> 1. User will open up the application and will see two circles already generated. 2. User does not like the colour scheme, so User clicks on the circle they want and chooses the color from the menu button. 3. User clicks “Data”. New window with two text-areas appears, the user can input set values. 4. Repeat step 3 until content. 5. User has finished adding entries to the Venn diagram and wants to export the file so they can send it to another person. User clicks on “Export” from the menu and then has a window pop up. User chooses to save the file as a pdf and exports. 6. User closes application.
Alternative Flows:	<p>After Step 2</p> <ol style="list-style-type: none"> 1. Users can click on “Label” and add labels to their Venn diagram to make it easier to follow which circle represents what. <p>After Step 3</p> <ol style="list-style-type: none"> 2. Users can click on “Format Text” and customize text and the diagram.
Requirements:	<p>The following requirements must be met before execution of the use case</p> <ol style="list-style-type: none"> 1. All buttons, menu items and toolbar items must do their required function. 2. Must be able to customize the shapes (change colour or resize). 3. Must be able to customize text (change font, colour, size, bolden or italicize).

Name of Use Case:	Comparing Houses and Apartments		
Created By:		Last Updated By:	
Date Created:		Last Revision Date:	
Description:	Comparing Apartments and Houses to figure out what is the better choice for the user using a Venn diagram with 2 circles.		
Actors:	User		
Preconditions:	<ol style="list-style-type: none"> 1. The shape chosen must be able to duplicate and create a diagram with both shapes that snap together. 		

	<ol style="list-style-type: none"> 2. The union of the two shapes must be shown as an intersection of the two shapes. 3. Diagram must be able to identify if two texts are the same
Postconditions:	<ol style="list-style-type: none"> 1. Window must quit after the “Quit” button is pressed from the menu. 2. Window must allow users to clear the diagram they have made, so users can create another diagram if required.
Flow:	<ol style="list-style-type: none"> 1. User will open up the application and will see two circles already generated. 2. User clicks on “Label” and editable labels pop-up. allowing the user to label the two circles, “Apartments” & “Houses”. The labels appear above the circles. 3. User clicks “Data” and a window pops up with 2 text-areas. 4. User sets values in the text-areas of the 2 sets. 5. Press done upon completion. 6. A diagram will be generated with the values in the appropriate locations for the user. 7. User repeats steps 3,4,5,6 until they are satisfied. 8. User is finished with the Venn diagram and closes the window
Alternative Flows:	<p>After any Step</p> <ol style="list-style-type: none"> 1. Users can click on “Label” and add labels to their Venn diagram to make it easier to follow which circle represents what. 2. Users can customize the circles and make changes they would like such as resize and/or set a different background for the shape. <p>After Step 3</p> <ol style="list-style-type: none"> 3. Users can click on “Format Text” and customize text and the diagram.
Requirements:	<p>The following requirements must be met before execution of the use case</p> <ol style="list-style-type: none"> 1. All buttons, menu items and toolbar items must do their required function. 2. Must be able to customize the shapes (change colour or resize). 3. Must be able to customize text (change font, colour, size, bolden or italicize).

6. User Acceptance Test Cases

6.1: UAT Scope:

6.1.1 UAT – In Scope:

- Generating a Venn diagram
- Customization of Venn diagram
- Undo/Redo
- Exporting diagram
- Importing existing file
- Converting large data sets into numbers

6.2: UAT Assumptions and Constraints:

6.2.1 UAT Assumptions:

- The desktop application functions as expected

6.2.2 UAT Constraints:

- Time frame
- Vague instructions from client

6.3 UAT Risks:

Description of Risk	Probability	Impact	Mitigation
Not properly trained testers	Medium	Medium	Ongoing training for testers
Incomplete test environment <i>(Still in development phase)</i>	High	High	Realistic time and resource planning
Acceptance test case failure	Medium	High	Completing development phase before UAT testing

6.4 UAT Entry Criteria:

- The development of the features that are being tested is completed
- All bugs fixed
- Test environment is ready
- Test plan is available

6.5 UAT Requirements-Based Test Cases:

ID	Test Cases
6.5.1	Generating Venn Diagram: <ul style="list-style-type: none"> • Launch the application • Add data to the diagram Expected result: <ul style="list-style-type: none"> • Venn diagram generated with data • Common data between sets inserted into the intersection of the diagram
6.5.2	Customization of Venn Diagram: <ul style="list-style-type: none"> • Launch the application • Add data to the diagram • Add titles using textbox • Select the size of the circles • If the color is to be changed, use the color picker to display the shape in the desired color • If the text to be edited, select the textbox and use the text formatting options on the top bar to edit the font, color, or alignment Expected Result: <ul style="list-style-type: none"> • Customized Venn diagram with the appropriate color or text is displayed on the editor
6.5.3	Undo/Redo <ul style="list-style-type: none"> • Launch the application • Add data to the diagram • Perform some actions on the diagram • Use the undo/redo option on the sidebar to undo or redo the changes that were just made Expected Results: <ul style="list-style-type: none"> • The changes made to the diagram were undone or redone according to the users choice