

Design Document

Group 16

Varuhn Ruthirakuhan - 215634140

Abdalah Yusuf-216516718

Uchechukwu Madu - 214507800

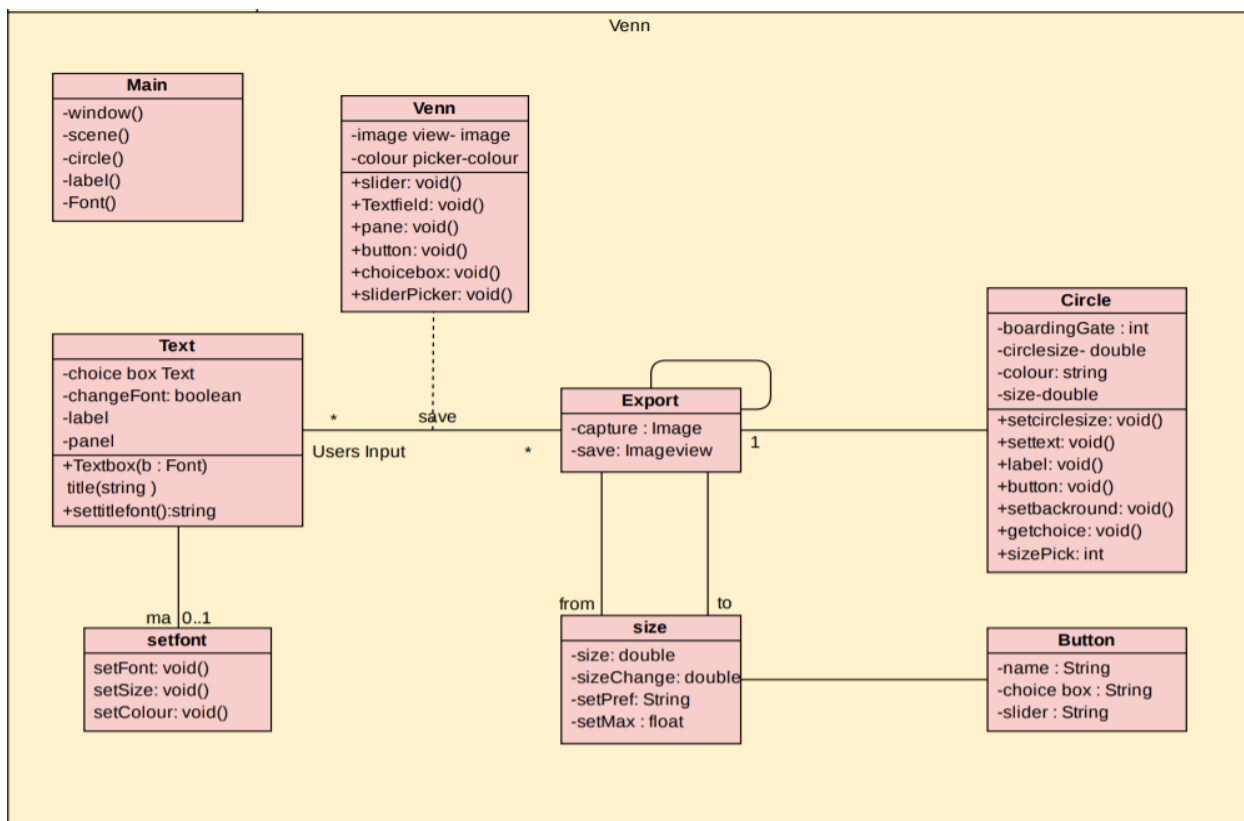
Overview

This program operates by user inputting data to our Venn diagram. Users can very well modify their diagram to their needs by selecting and customizing their diagram. When users are satisfied with their diagram, users can export the diagram.

High-level Class Diagram

This class diagram shows how objects in the program interact with each other. The main class is where the program starts. From the main class, our program constructs the User Interface. Our User Interface handles all of the users interaction such as, users inputting data, select and delete, resizing, and much more. It is from there we have multiple classes that extend main, such as our Venn class. Our Venn class handles all of the background and image panels. The Circle class holds all of our circle customizing panels. Also, we can highlight that many of our classes depended on each other. For example, the setFont is dependent on the text. The text class allows for the User Interface to work so the correct class works.

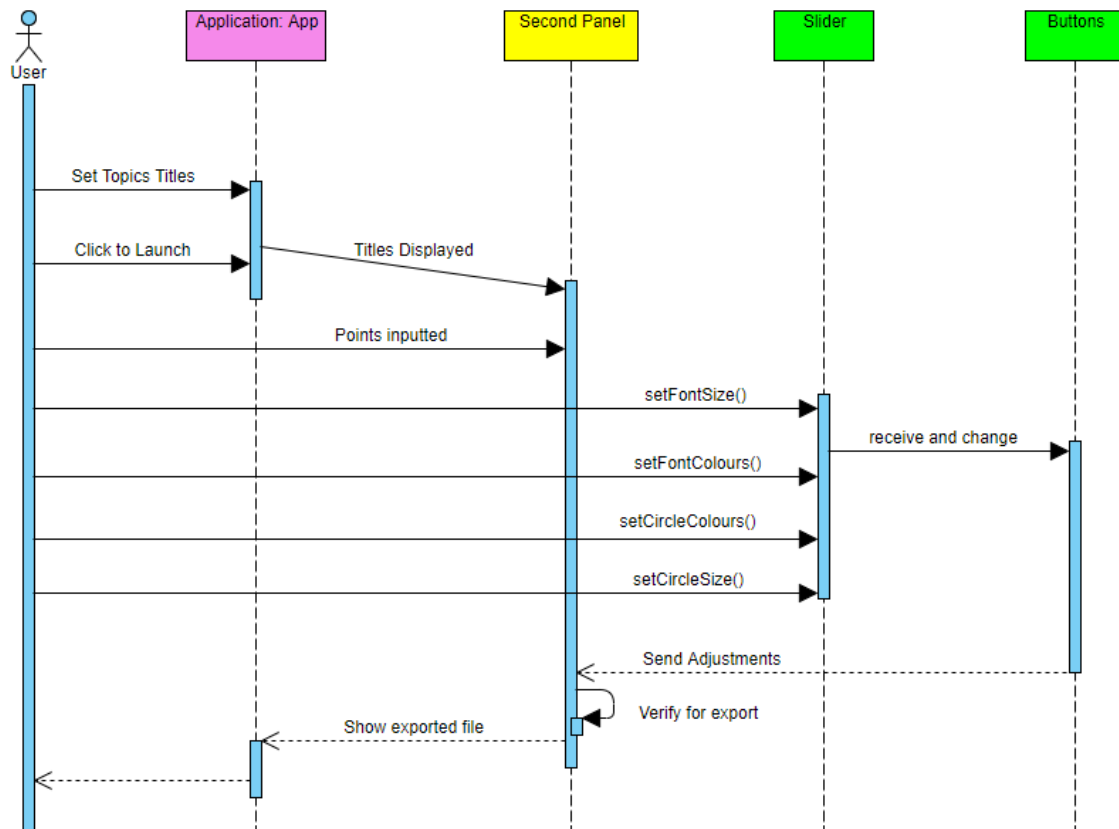
Here below is Our High-level Class Diagram highlights how our classes interact with each other, within our system to achieve High level success.



Sequence Diagrams:

Brief overview in Runtime:

This sequence diagram displays and explains what happens from when the user opens the file for the first time, to when they want to export it.



To start, the user will be prompted with a primary window when launching the Venn diagram application. This diagram will provide the user with options for user input to enter topic one title, topic two title, and Venn Diagram title. Once these fields are entered the user can activate and launch the application into the second panel.

This second panel will display the inputted titles for topic one and topic two above their respective Venn diagram circles with The main title centered and above them. The user will also

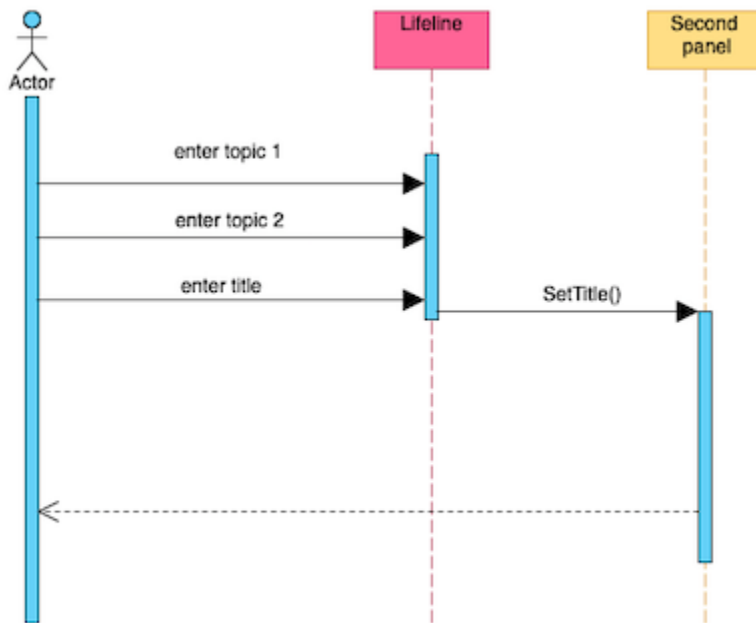
have the ability to input as many points as they wish into the diagram using the allocated text field.

After doing this the user can apply many customizations of their choice from the wide range of modifications. These modification abilities include changing the font sizes, changing the font colours, changing the circle sizes, and changing the circle colours. These modifications will be done by using the slider and set using their respective buttons. After these values have been received and changed, once the user is happy, they can export their Venn diagram at its current status.

A recursive method will call all the attributes that have been allocated and save them for the export. The export function will deliver two files to the user. One is a .jpeg which will save the file as a diagram so the user can display and show their work. The second file will be a text file which will easily display every individual component line-by-line. This second file is mainly used if the user would like to re-open the file and add further modifications at a later time or date.

Focus On Input From First Panel To Second Panel:

This sequence diagram's main focus is on the input from the first panel to the second panel, showcasing what happens when the application is launched, and topic and title are entered by the user.

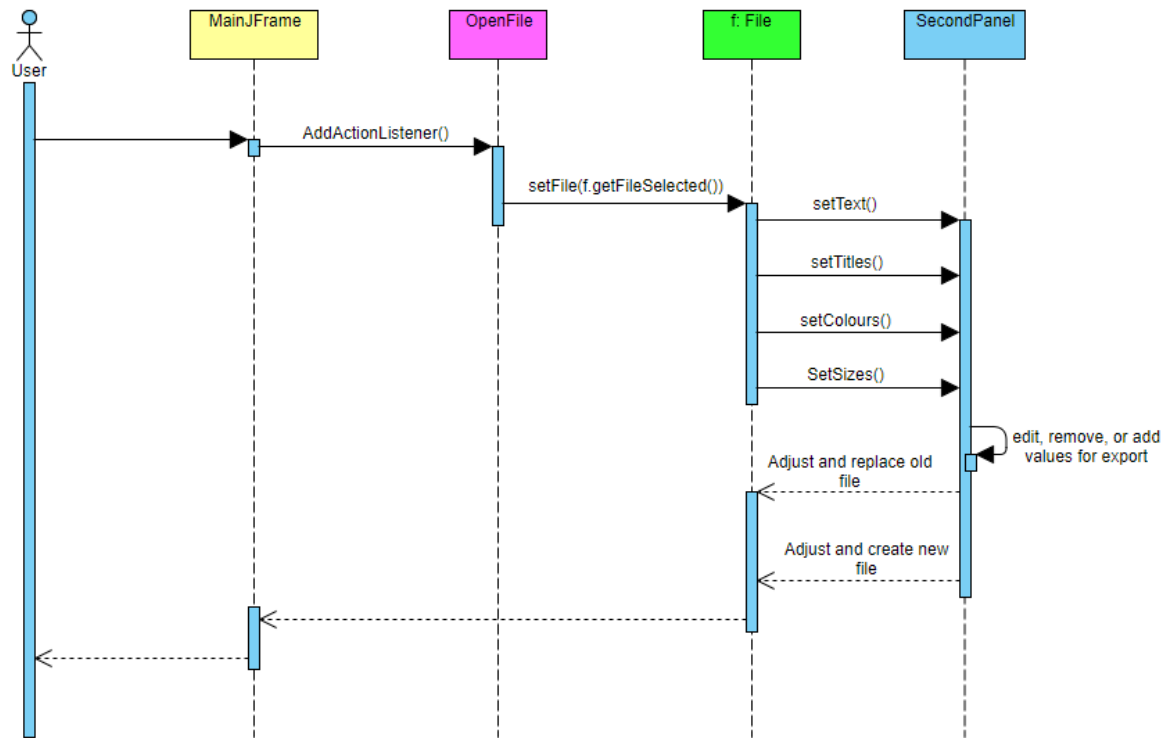


Initially, when the application is launched the user has the option to choose what the title and topics for the Venn diagram would be. When this is done, the user clicks on a button which says, "Construct Venn Diagram."

The second panel comes up with the title which was chosen by the user and is displayed on top of the Venn diagram as well as the topics on top of the circles. This displays a better understanding of the structure of the Venn diagram.

Focus on import and export of Venn Diagram App and Files:

This diagram displays and explains



When the user launches the application, they will have the ability to import pre-existing files that they have already worked on back for modifications. If the user wishes to import a previously worked on project then the ActionListener() will be called to open the file. Then the file will be set and opened on the second panel where the application attributes can be modified further. Multiple methods get called to allocate the current text, set the Venn diagram sizes, and Venn diagram colours. Now the user can add or modify these attributes for a later import if they wish to further work on the project.

Once the user is happy with the modifications they can export the file. This time when the user wishes to export the file, they will have the option to create new files, or replace the old files. This adds to the user preferences.

Maintenance Scenarios

Our code was implemented to be very maintainable. We focused on organizing our code to multiple packages that implement different features. We focused on maintaining a high cohesion level when designing our packages and classes. Below we will mention multiple typical maintained scenarios, with brief explanation.

Titles do not line up where user wants above Venn diagram:

A scenario can exist where the user enters long titles for their specific topics and Venn Diagram. Since the locations of the topic titles and application titles are set to their location, they may not line-up or look unorganized when displayed. To fix this, the developer would have to set the text fields allocated for the title and topic titles to a draggable node. This is simply done in javafx using the current `makeDraggable()` method. This method is already used in the application for the textboxes where the user would like to input text. If they wish to interchange and move texts from one area to another they would simply drag that text box to the new location they would like. This functionality can be added to the titles text fields to allow the user to move their specified titles to wherever they would like. Even if they would like to add the titles below the diagrams, they will now have the capability to do it.

Venn diagram does not save as wanted:

When the Venn diagram does not save as wanted this could be due to various reasons. One scenario could be where the user wants to save a file by replacing a currently saved version. The developer would have to add a method to the export button so when the button is called it can replace the old file. Currently, the user may experience issues where every version saves as a new file. This can simply be fixed by the developer in the `export()` method. Sub methods can be included to check for pre-existing files with the same name. If the file exists, the user will be prompted with a pop-up asking them to replace the old file or create a new file to satisfy their needs.

Adding an object to the application:

In a scenario where a user wants to import an object, for example, a picture to the diagram where the user does not currently have the ability to do so. The developer would highlight users' important experience such as importing pictures or other diagrams. We also prioritize user saving experiences, to allow users to save images and import them as well. In our final release we also allow users to share their Venn Diagram. The Developer can add these new features, by adding these new options to our code base. Developer would use `addListener()` to call the imported images into the application. Developers would also add to the `makeDraggable()` method to make the image movable. The image would first be set to a node so that the method can make use of it.