

# Design Document

## EECS2311

Group 10

Thanh Nguyen 213 507 231

Mahamed Hassan - 216536476

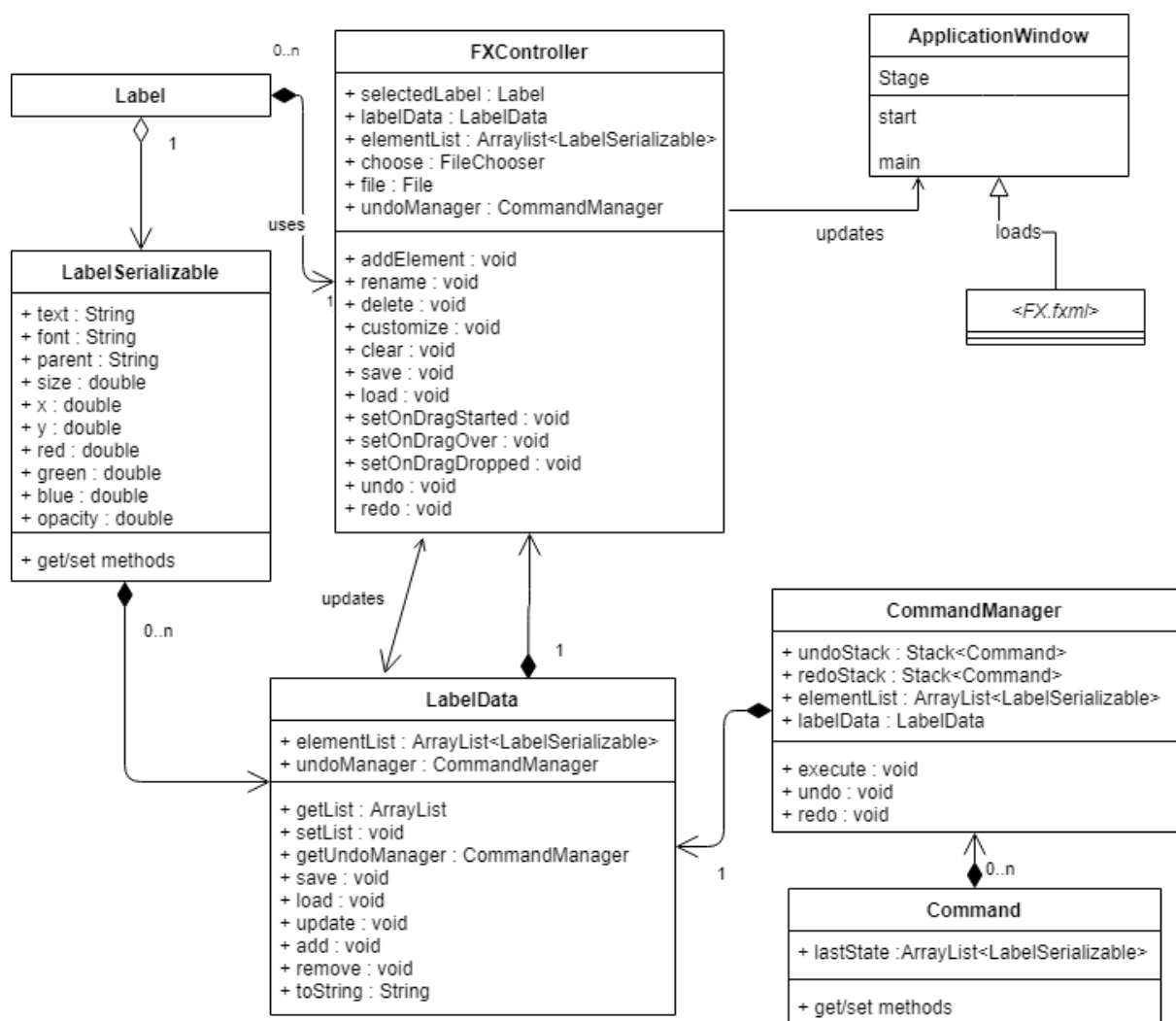
<b>Class Diagram</b>	<b>3</b>
<b>Sequence Diagrams</b>	<b>4</b>
Add	4
Delete	4
<b>Maintenance Scenarios</b>	<b>5</b>
Adding Additional Circles	5
Opening Hyperlinks in Elements	5

# Class Diagram

This diagram is a representation of what classes are used and how they interact with each other. The ApplicationWindow is initialized by the FXML file, then is updated with the FXController. The controller creates an instance of LabelData, which stores labels as LabelSerializable.

This allows us to save and load the state of the ApplicationWindow.

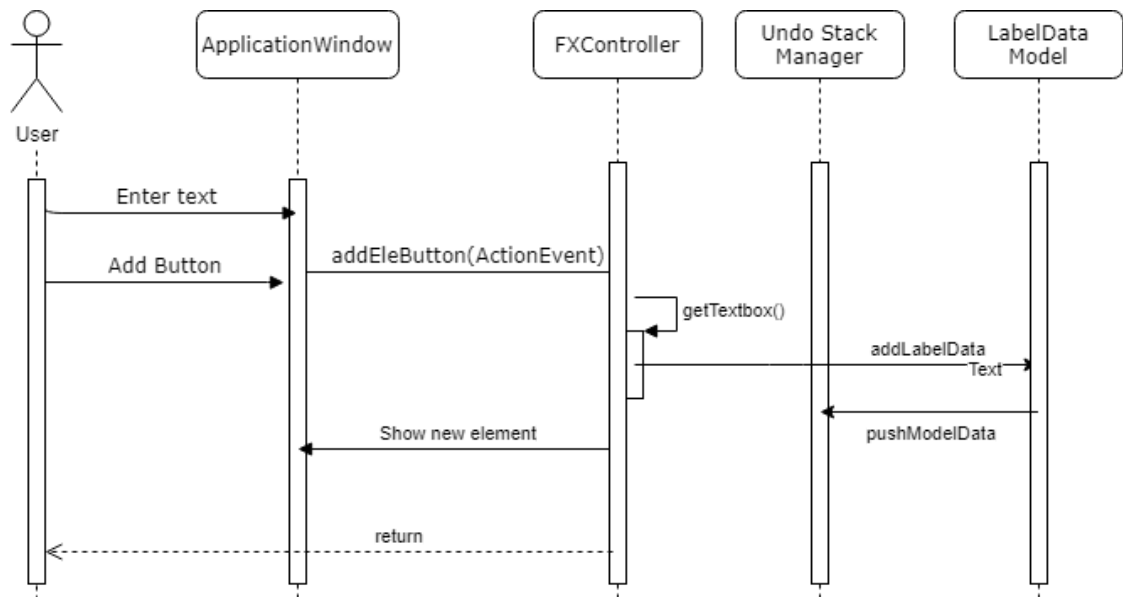
Every change done to LabelData is saved as a Command, which is handled by CommandManager. CommandManager will push a new Command onto its undo stack, and every undo operation both pop that Command and push it onto the redoStack.



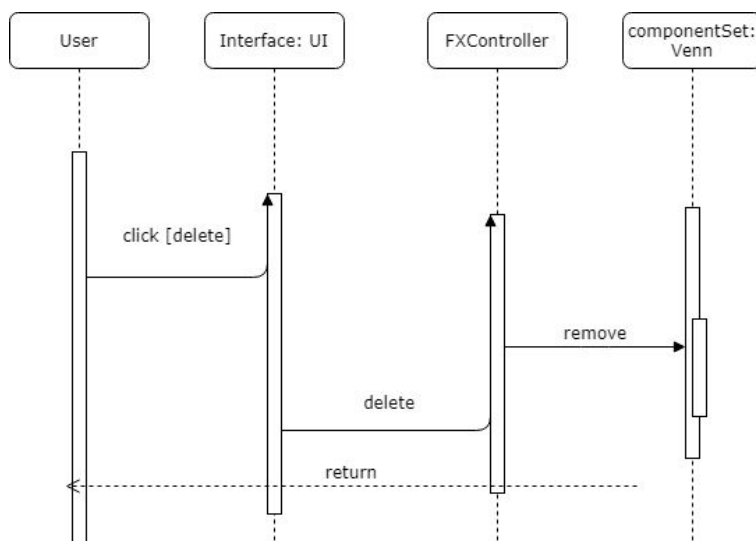
# Sequence Diagrams

These diagrams are a representation of how the end user can interact with the program, and how the program responds accordingly. Each diagram represents an action performed by the user, and the methods involved with that action.

## Add



## Delete



# Maintenance Scenarios

## Adding Additional Circles

If we want to add more circles to the diagram in the future, a simple way would be to make a copy of the two-circle FXML file, modify it to add a third or fourth circle.

Then we would load each additional FXML in the application class as an individual scene.

Finally, we would add a button to the application and accompanying method that would switch scenes to the desired number of circles, and reload the elements created by the user.

## Opening Hyperlinks in Elements

In the future, we may want to allow the program to accept hyperlinks in elements, then allow the user to click them to open it in their browser.

This could be achieved by creating a hyperlink attribute for each element, and creating a method that will open the link in their default browser using the Hyperlink class. Then we would add a button in the elementContextMenu associated with that method.