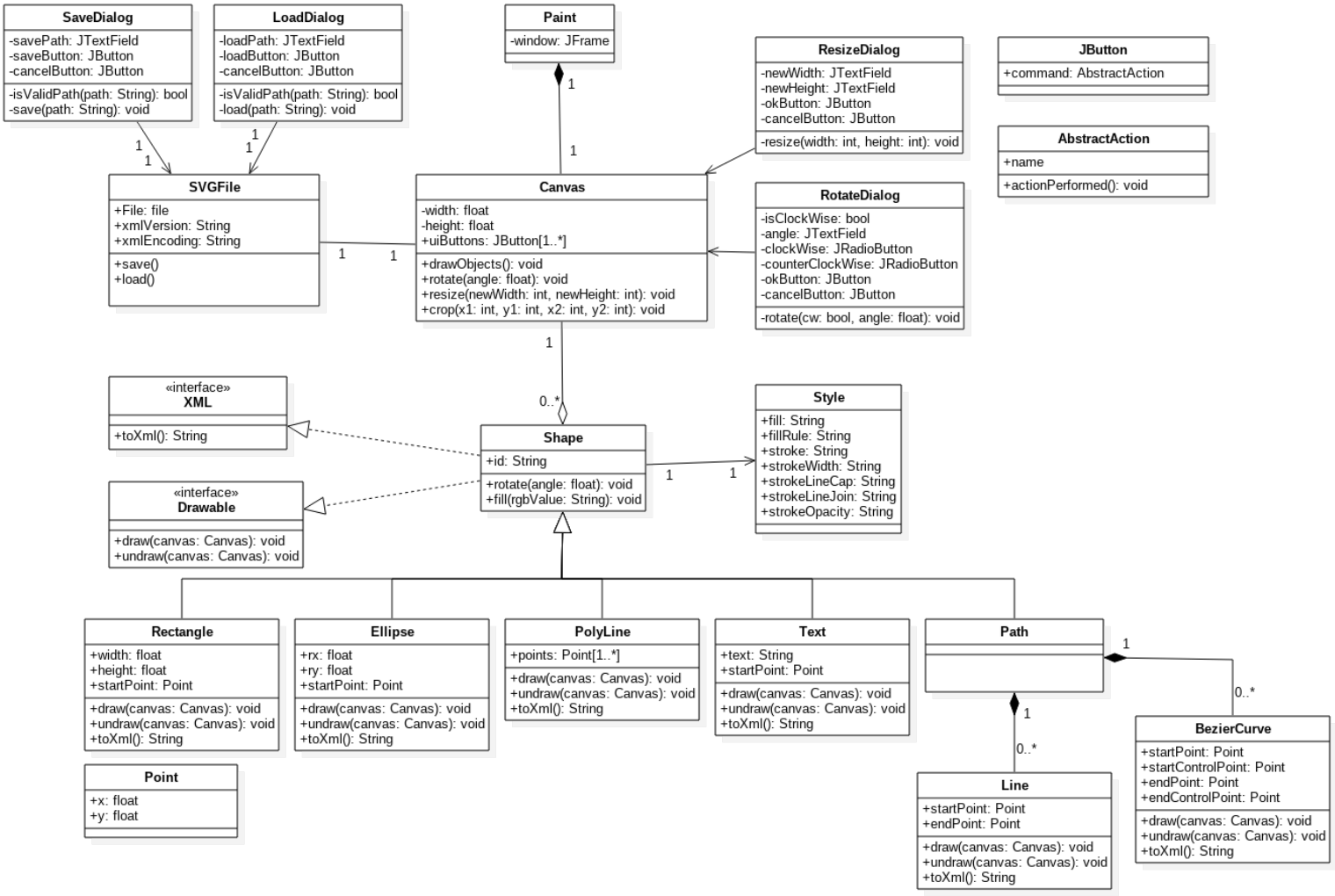
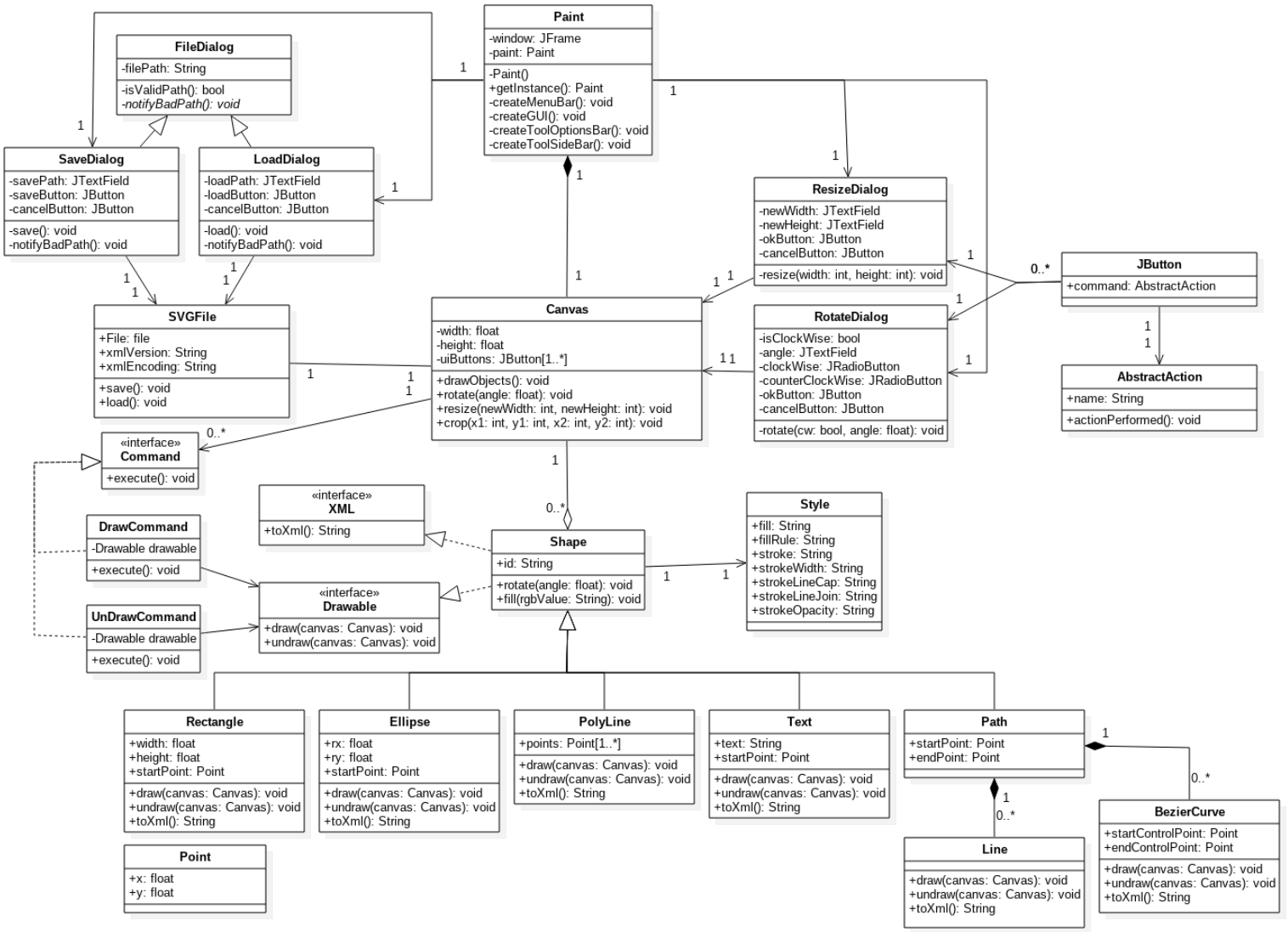
CSCI 4448 Project Part 3

Robert Allen

Old class diagram:

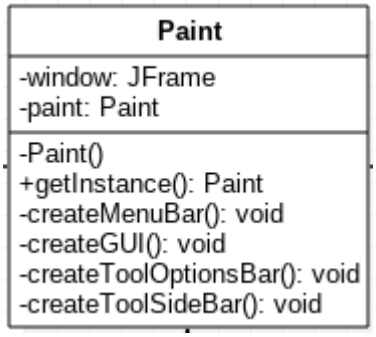


New class diagram:

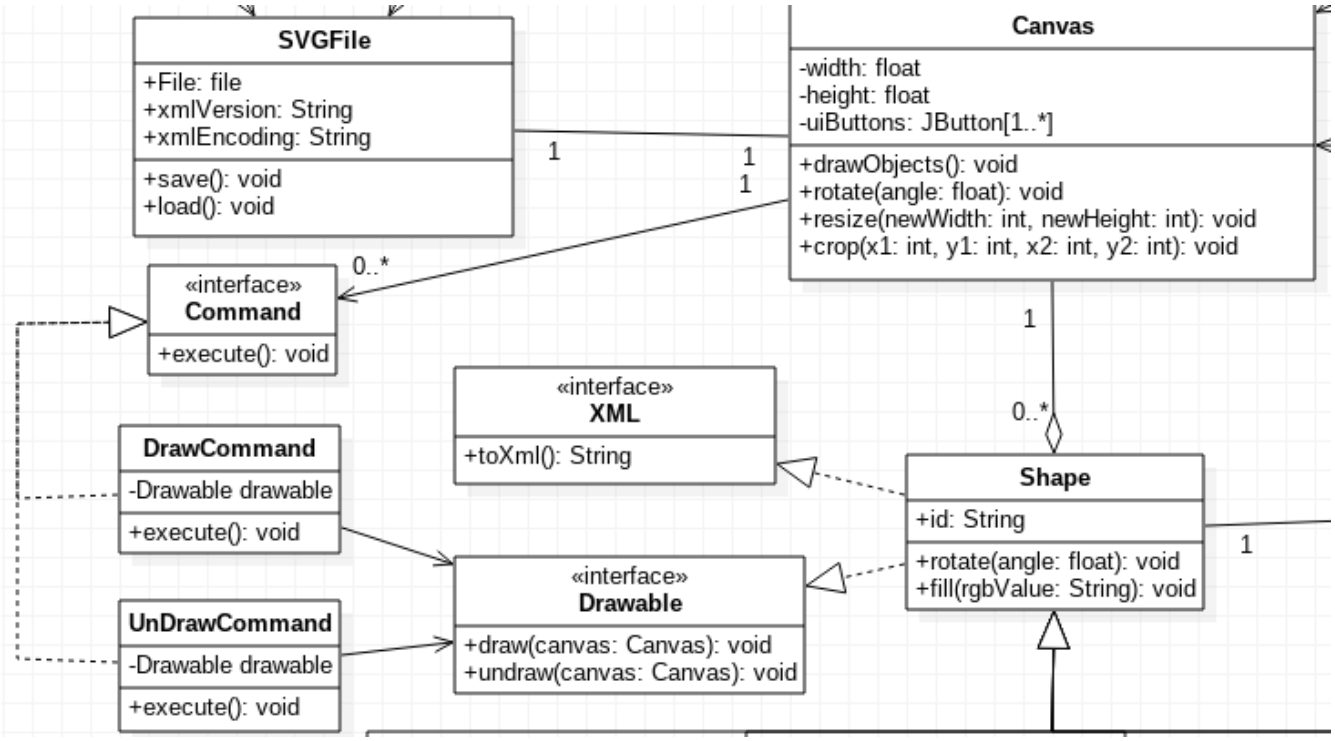


Refactoring Description:

Singleton: The paint class has been made into a singleton. There should only every be one instance of paint running at any time. To do this I made the constructor private, a private attribute paint to store itself, and the public method getInstance() to create a new instance of paint if it doesn't exist and return the old instance of paint if it already does.



Command: I've also added the command pattern. This will be used if we want to add undo and redo operations to drawing our drawables. I added an interface called command with only one method called execute(). This interface has two realizations, DrawCommand, and UndoCommand. Both hold a private instance of a drawable. In the execute method both will call draw() or undraw() respectively on their drawable.



Additional Refactoring: I attempted to fix any issues we had received from our part two feedback. JButton and Abstract Method were previously not associated with anything else in the diagram, however now they are associated with each other and Jbutton is associated with the rotate and resize dialogs where they will be used.

Path was previously a completely empty class. I've changed it so that Path now holds the start and end points as both child classes were doing this already and it made sense to move it up to the parent class.

I have not changed point. Point is used throughout the whole program as a simple way to store x and y locations. I could give it associations with everything in the program that utilizes it but I personally think that that would only make the class diagram to look much more confusing than it needs to be. I will go with clarity over UML correctness in this case.

I also fixed that some methods were missing their return types by adding them in appropriately.

The final thing I did was add methods to the paint class that would help in building out the user interface. Once I had started work on the Paint class I realized that we had not thought through some of the details of building out the user interface. These additional methods should help to address those needs.