* **Initial Design**

The initial design was done as a paper prototype for the sake of simplicity and time. It’s much easier to draw out ideas on paper than it is to find clipart approximations on the internet. It’s much easier to make adjustments as well.

The initial design was inspired by mobile applications’ simplistic and sleek User Interfaces. The side menu especially was designed to come out of the side and be able to be minimized as can be found in many mobile applications. In those applications’ cases, it is designed that way because phone screens need to save a lot of space.

The color wheel was designed to always be in the bottom corner of the screen, atop all menus, so that it could be quickly accessed when using the canvas. This was designed so in order to mimic how classical painters can easily pick different colours with a physical palette in real life. This was reimagined in the redesign as the quick panel.

* **Development Phase 3**

|  |  |
| --- | --- |
| **Features** | **Implementation** |
| Tools panel | A JTabbedPane that contains a JPanel, which in turn contains most of the canvas drawing features |
| Effects panel | A JTabbedPane that contains a JPanel, which in turn contains most of the image manipulation features |
| Rotate | A JButton that turns the degree of whatever image is on the screen by 15 degree increments |
| Brush size | A JSlider that allows you to choose the stroke width. The current brushstroke width is shown in a JLabel |
| TextBox | A ‘library’ or new class, InvisibleTextField was going to be used for this implementation, but it could not be gotten to work in time |
| Blur | A JSlider was used to control the blur of an image between values 0 and 9 (1 to 10).\* |
| Sharpness | JSlider that could not be implemented in time. (Its implementation turned out to be nontrivial) |
| Black and White | A JButton that turns colors into grayscale by getting the average RGB value for each respective pixel |
| Warm | A JButton that decreases by a percentage, the Blue and Green of each pixel in the image |
| Vivid | A JButton that changes the hue of a loaded image by 5% increments |

\* In order to make applying blurs very quick, an array of size 10 of whichever image is loaded onto the canvas is saved. Blur effects of varying degrees – from 1 to 10 – are applied to the image and stored into the array at its matching index. Therefore, changing the blur slider actually only changes the index of which image the user is accessing.