**UI Requirements**

1. **Canvas**
   1. Configurable
2. **Tools**
   1. Move Tool
      1. Select
      2. Move
      3. Transform
   2. Rectangle Tool
      1. Draw rectangles. (No editing capabilities on existing rectangles)
   3. Ellipse Tool
      1. Draw ellipses. (No editing capabilities on existing ellipses)
   4. Polygon Tool
      1. Draw polygons. (No editing capabilities on existing polygons)
   5. Pen Tool
      1. Draw paths. (Capable of editing on existing paths if they have an open end)
   6. Line Tool
      1. Draw lines. (No editing capabilities on existing lines)
      2. Once the second vertex is fixed, the tool should be released from the action.

1. **Edit Box**
   1. Fill color editor
      1. Should be set to default value if no object is selected.
      2. Should show the fill color of the selected object, if any.
      3. In case if multiple objects are selected, the editor should show an uncertain symbol if the objects have different fill colors. If all of them have the same fill color, show that color in the editor.
   2. Stroke color editor
      1. Should be set to default value if no object is selected.
      2. Should show the stroke color of the selected object, if any.
      3. In case if multiple objects are selected, the editor should show an uncertain symbol if the objects have different stroke colors. If all of them have the same stroke color, show that color in the editor.
   3. Stroke width editor
      1. Should be set to default value if no object is selected.
      2. Should show the stroke width of the selected object, if any.
      3. In case if multiple objects are selected, the editor should show an uncertain symbol if the objects have different stroke widths. If all of them have the same stroke width, show that width in the editor.
2. **Layer Palette**
   1. Every element inside the canvas should be shown as a layer in the layer palette with the last element on top and the first element on the bottom.
   2. The hierarchy of the layers should be able to be rearranged, and it would get instantly reflected on the canvas.

**Vertex Constructor**

Vertex constructor creates a vertex object for each points in the path-edit-line. Every vertex object contains vertex data in the following form of an array,

[ left-control-point position , vertex-point position , right-control-point position ]

All Vertex object is stored in the pathNodeArray property of the corresponding PathObject.

**Vertex Object**

Vertex objects are created using the Vertex-constructor. It contains vertex data as represented in the array above. This objects controls the vertex position and control point position while every time an action is performed on that vertex.

**generatePoints ()**

This function generates vertex data from the ‘d’ attribute of a path element and creates a Vertex object for each point in the path using the Vertex-Constructor.

**generatePathData ()**

This function collects vertex data form all the vertex objects stored in the pathNodeArray. Then it analyze that data and generates corresponding path data which can then be directly applied to the ‘d’ attribute of a path element.