Alexandria University
Faculty of Engineering
Computer and Systems Engineering

Lab 3

Web Based Drawing App

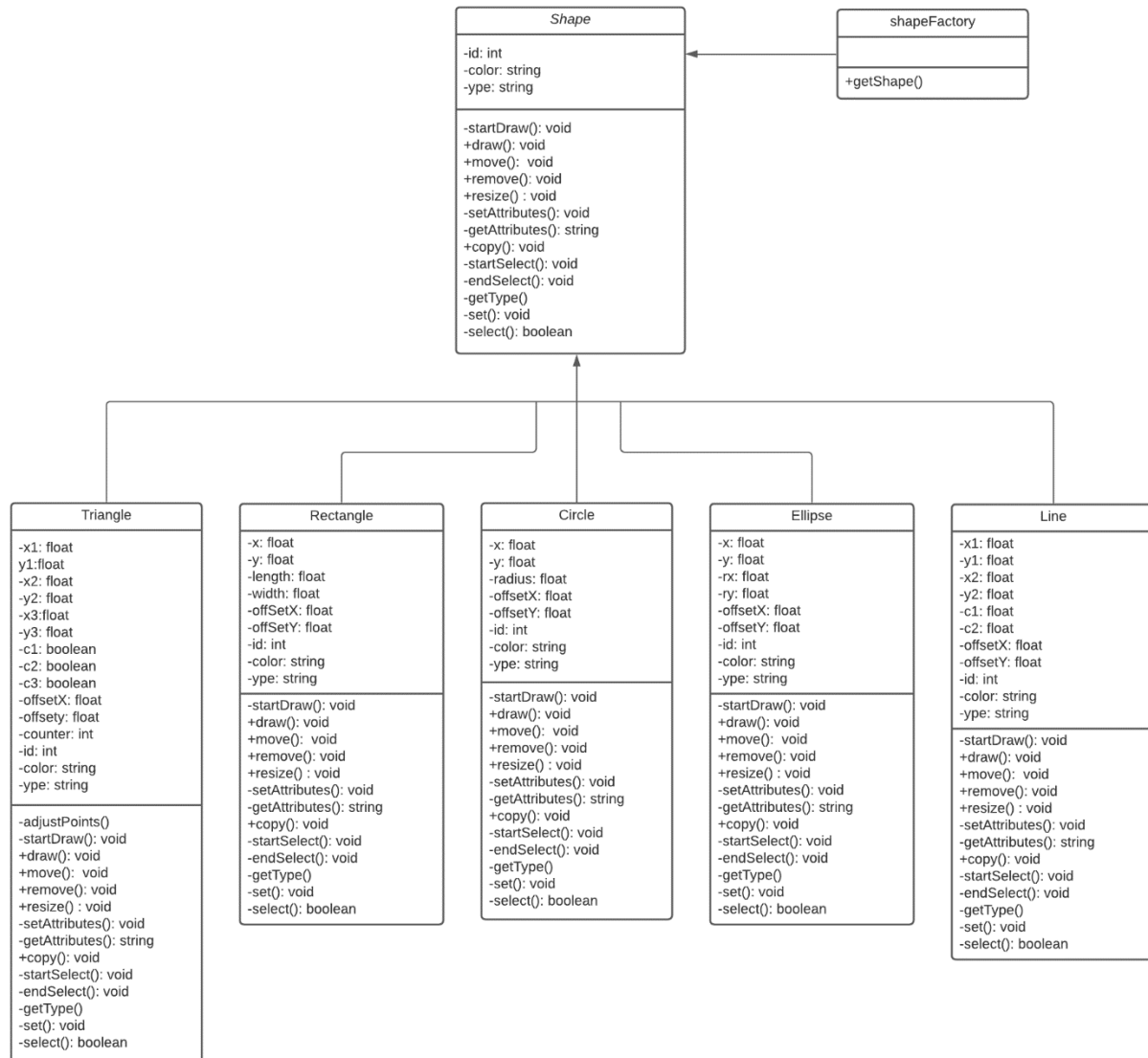| Name	ID
Mariam Mohammed Ahmed	19017274
Yara Hossam Abdelaziz	19016871
Nourhan Ahmed Arafa	19016812
Nada Mohamed Ibrahim	19016782

How to run the code

You need to import some packages, components, and decorators to run the program in a full phase.

- Front-End
  - HttpClient: to allow sending requests to the server.
  - Renderer2: to allow operations on the svg element.
  - ViewChild: to look for the first element that matching the selector in the DOM.
  - ElementRef: to permit direct access to the DOM.
  - Injectable: to mark a class as available to be provided and injected as a dependency.
  - FormModule: to allow extracting inputs from forms.
- Back-End
  - java.util.Arraylist: to allow dealing with the objects in more familiar way than the array.
  - org.springframework.boot.SpringApplication
  - org.springframework.boot.autoconfigure.SpringBootApplication: to allow running spring application

UML Diagrams

- Factory Pattern in the Front End

Shape

```
-id: int
-color: string
-ype: string
```

```
-startDraw(): void
+draw(): void
+move():  void
+remove(): void
+resize() : void
-setAttributes(): void
-getAttributes(): string
+copy(): void
-startSelect(): void
-endSelect(): void
-getType()
-set(): void
-select(): boolean
```

shapeFactory

```
+getShape()
```

Triangle

```
-x1: float
y1:float
-x2: float
-y2: float
-x3:float
-y3: float
-c1: boolean
-c2: boolean
-c3: boolean
-offsetX: float
-offsety: float
-counter: int
-id: int
-color: string
-ype: string
```

```
-adjustPoints()
-startDraw(): void
+draw(): void
+move():  void
+remove(): void
+resize() : void
-setAttributes(): void
-getAttributes(): string
+copy(): void
-startSelect(): void
-endSelect(): void
-getType()
-set(): void
-select(): boolean
```

Rectangle

```
-x: float
-y: float
-length: float
-width: float
-offSetX: float
-offSetY: float
-id: int
-color: string
-ype: string
```

```
-startDraw(): void
+draw(): void
+move():  void
+remove(): void
+resize() : void
-setAttributes(): void
-getAttributes(): string
+copy(): void
-startSelect(): void
-endSelect(): void
-getType()
-set(): void
-select(): boolean
```

Circle

```
-x: float
-y: float
-radius: float
-offsetX: float
-offsetY: float
-id: int
-color: string
-ype: string
```

```
-startDraw(): void
+draw(): void
+move():  void
+remove(): void
+resize() : void
-setAttributes(): void
-getAttributes(): string
+copy(): void
-startSelect(): void
-endSelect(): void
-getType()
-set(): void
-select(): boolean
```

Ellipse

```
-x: float
-y: float
-rx: float
-ry: float
-offsetX: float
-offsetY: float
-id: int
-color: string
-ype: string
```

```
-startDraw(): void
+draw(): void
+move():  void
+remove(): void
+resize() : void
-setAttributes(): void
-getAttributes(): string
+copy(): void
-startSelect(): void
-endSelect(): void
-getType()
-set(): void
-select(): boolean
```

Line

```
-x1: float
-y1: float
-x2: float
-y2: float
-c1: float
-c2: float
-offsetX: float
-offsetY: float
-id: int
-color: string
-ype: string
```

```
-startDraw(): void
+draw(): void
+move():  void
+remove(): void
+resize() : void
-setAttributes(): void
-getAttributes(): string
+copy(): void
-startSelect(): void
-endSelect(): void
-getType()
-set(): void
-select(): boolean
```

- Factory Pattern in the Back End

<<Interface>> IShape

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setColor(String):void
+ setID(int):void

implements

Line

~ type:String = line
~ parameters:String
~ color:String
~ outline:String
~ posx1:int
~ posy1:int
~ posx2:int
~ posy2:int
~ ID:int

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setID(int):void
+ setWidth(int):void

Circle

~ type:String = circle
~ parameters:String
~ color:String
~ outline:String
~ radius:int
~ posx:int
~ posy:int
~ ID:int

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setID(int):void

Ellipse

~ type:String = ellipse
~ parameters:String
~ color:String
~ outline:String
~ r1:int
~ r2:int
~ posx:int
~ posy:int
~ ID:int

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setID(int):void

Triangle

~ type:String = triangle
~ parameters:String
~ color:String
~ outline:String
~ posx1:int
~ posy1:int
~ posx2:int
~ posy2:int
~ posx3:int
~ posy3:int
~ ID:int

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setID(int):void

Rectangle

~ type:String = rectangle
~ parameters:String
~ color:String
~ outline:String
~ posx:int
~ posy:int
~ width:int
~ length:int
~ ID:int

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setID(int):void

Square

~ type:String = square
~ parameters:String
~ color:String
~ outline:String
~ posx:int
~ posy:int
~ width:int
~ length:int
~ ID:int

+ getAttributes():String
+ getType():String
+ getColor():String
+ getOutline():String
+ getID():int
+ setAttributes(String):void
+ setType(String):void
+ setColor(String):void
+ setOutline(String):void
+ setID(int):void

Controller

+ getInstances(String):ArrayList
+ changeColor(String):ArrayList
+ resize(String):ArrayList
+ move(String):ArrayList
+ delete(String):ArrayList
+ copy(String):ArrayList
+ undo(String):void
+ save(String):void
+ load(String):void
+ newFile():void

Asks

ShapeFactory

~ shapeType:String
+ shapes:Arraylist

+ setShape(String):void
+ createShape(String, String[ ], String)

creates

Operations

+ updateShapeStatus(int, String):ArrayList
+ delete(int):ArrayList
+ copy(int, int):ArrayList
+ getLastID():int

Design Patterns

- For the Front-End
  We used the factory pattern to allow producing new shapes which are independent from each other, each one with its own attributes and behaviors.
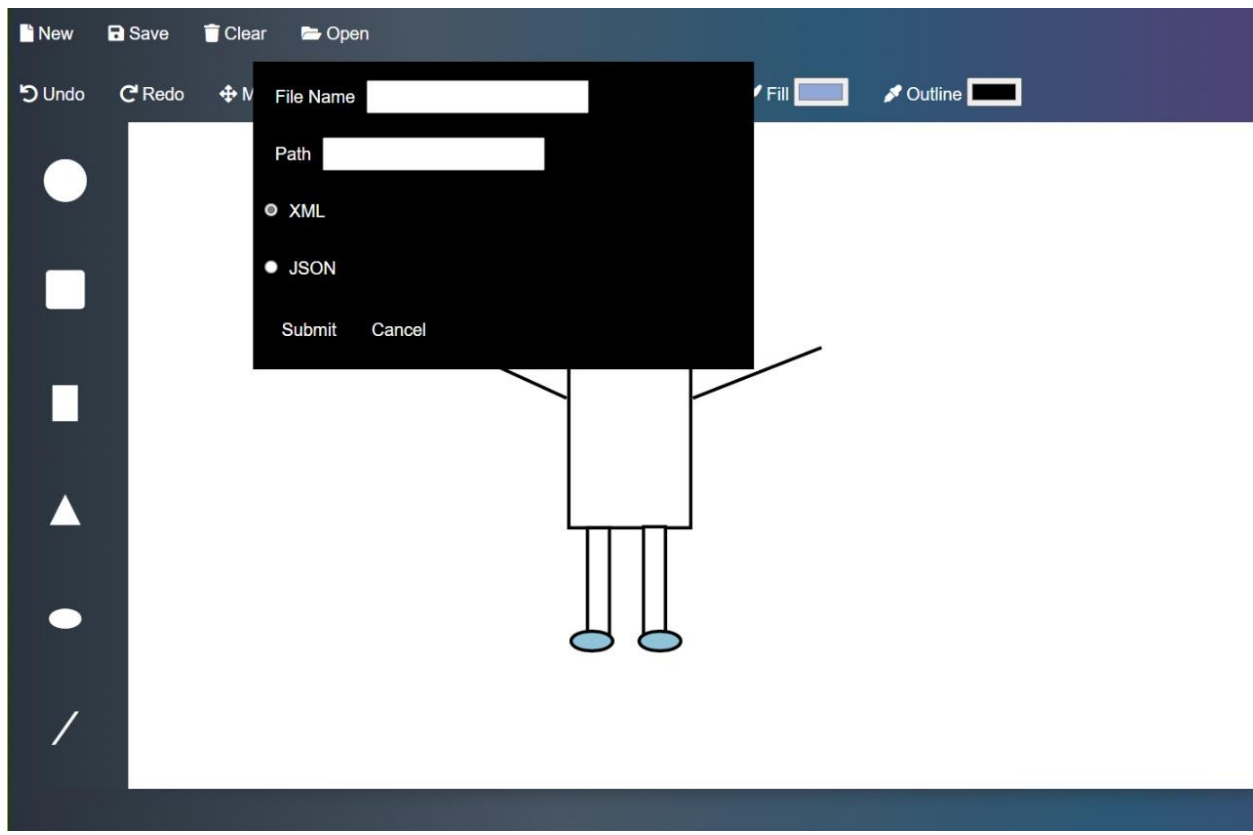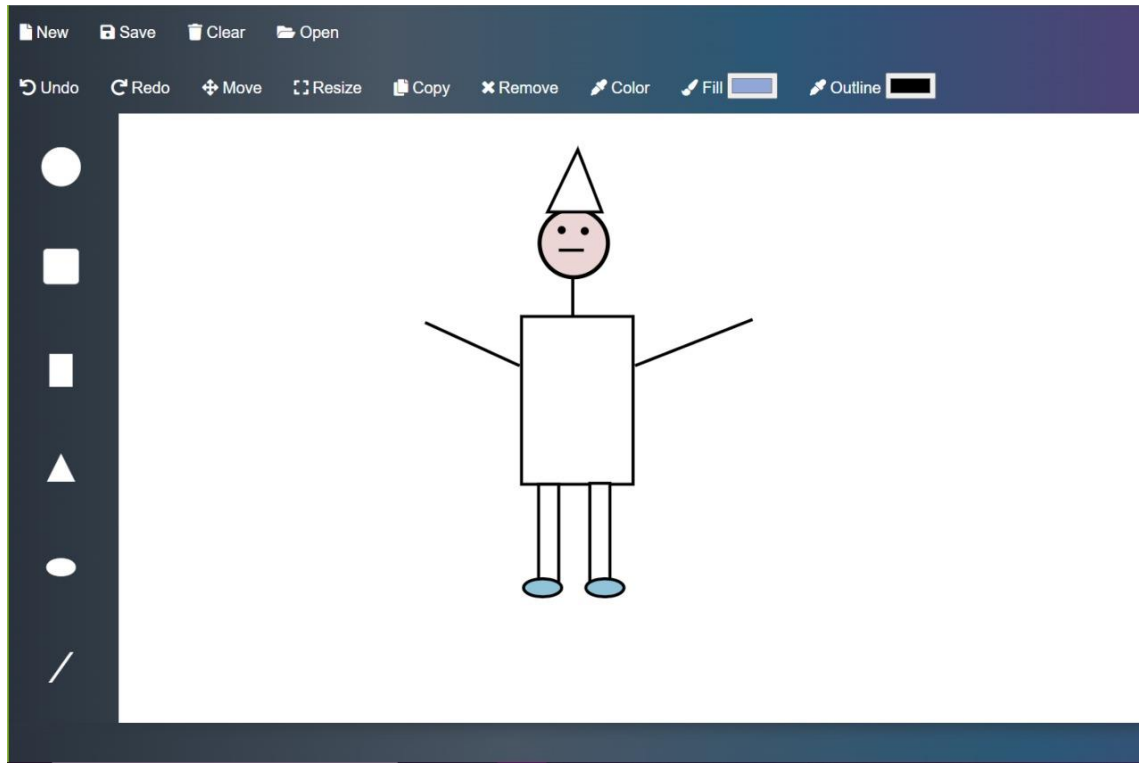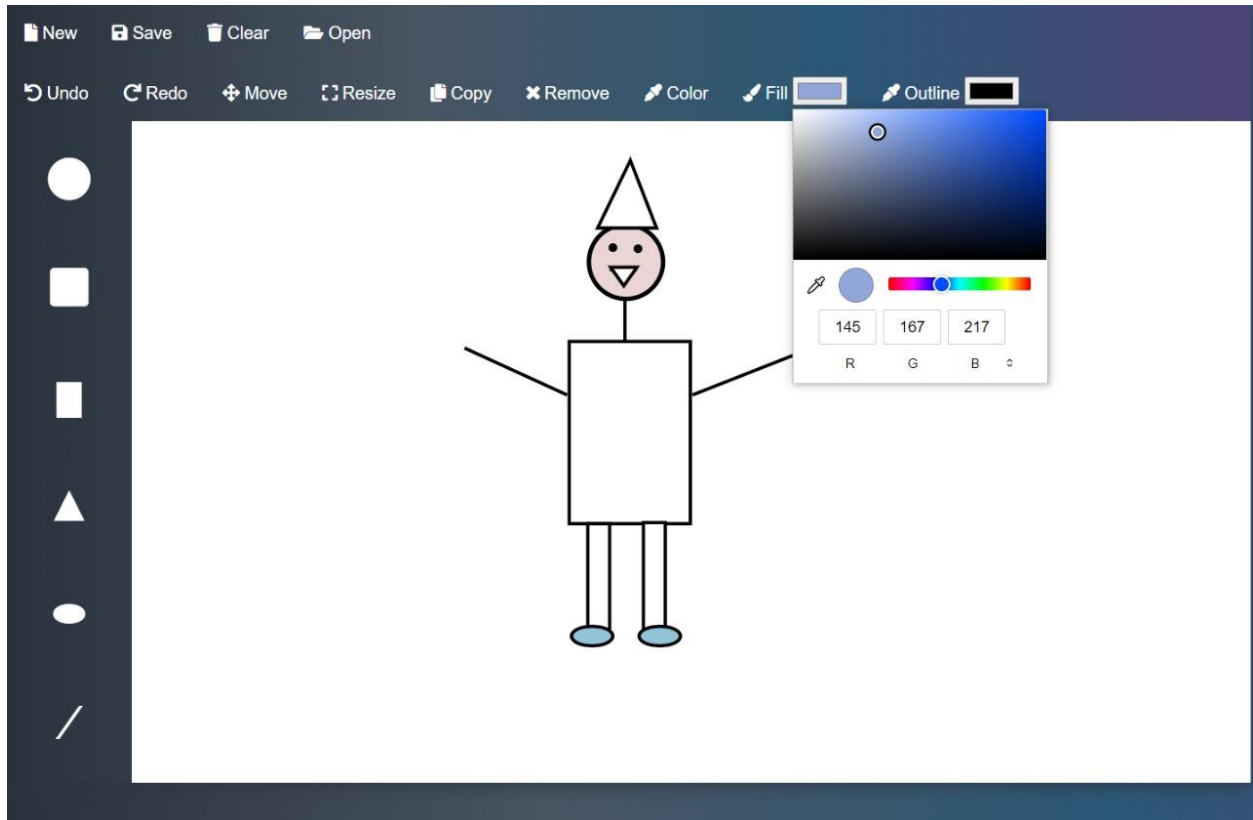- For the Back-End
  The factory pattern was used to create new shapes of different classes.

Design Decisions

- The triangle's resizing takes place at one point at a time.
- The triangle and the line can be selected from the terminal points only.
- When loading a file, undo/redo operations can be applied only to the new modified operations.

Snapshots of the UI

How-To-Use

- To Draw

  You need to click on the icon of the shape to be drawn, then follow the next instructions:

  - For all shapes except triangle
    - You need to select 2 points.
    - Select the first point on the board.
    - Drag and select the second point.
  - Triangle
    - You need to select 3 points.
    - Select the first point on the board to start drawing
    - Move the line and select the second point.
    - Move the line and select the third point.

- For selecting a shape in the following operation, you need to click on any point on the shape to be selected except the triangle and the line you need to select a terminal point.
- To Move
  - Click on the move icon.
  - Move the required shape by dragging and dropping it.
- To Resize
  - Click on the resize icon
  - Drag the shape until reaching the required size.
- To Copy
  - Click on the copy icon
  - Select the required shape.
  - The copy will appear on the select shape.
- To Change the color

  The following 2 operations are set together:
  - To change the fill color
    - Click on the fill icon.
    - Choose a color
    - Select a shape to change its color.
  - To change the border color
    - Click on the border icon
    - Choose a color
    - Select a shape to change its color.
- To create a new file
  - Press the icon new
- To Upload a file
  - Click on the open icon.
  - Enter the file name, the path.
  - Choose the file type either json or xml.
  - Press submit.

- To clear the board
  - Press the icon clear.
- To save a file
  - Click on the open icon.
  - Enter the file name and the path.
  - Choose the file type either json or xml.
  - Press submit