SWE 205: Introduction to Software Engineering

Term 191

Software Design-Document Structure
Project:
Painting Software called “Mypaintshop”

Group 11 Names:

- **ABDULLAH ALFAIFI (Leader)**
- FAWAZ ALESAYI
- WALEED ALFAIFI
- YAZEED ALJOHANI

Table of Contents

1. Introduction

- **Introduction of the product:**

This project is a drawing program which allows the user to draw shapes, change their color, export canvas, and load canvas.

This document is a description for the design aspect for this project which will help in the programming step. The document is based on requirements mentioned in the SRS document.

- **Purpose of this document:**

The document purpose is to put a guideline for the developers from the Software Architecture till the description of the functionalities. This document also will help in the process of upgrading and modifying the project.

- **Overview of the remainder of the document:**

The remaining section will goes like that:

- Section 2:

Will give a brief about the architecture of the system using Block Diagrams, with the justification.

- Section 3:

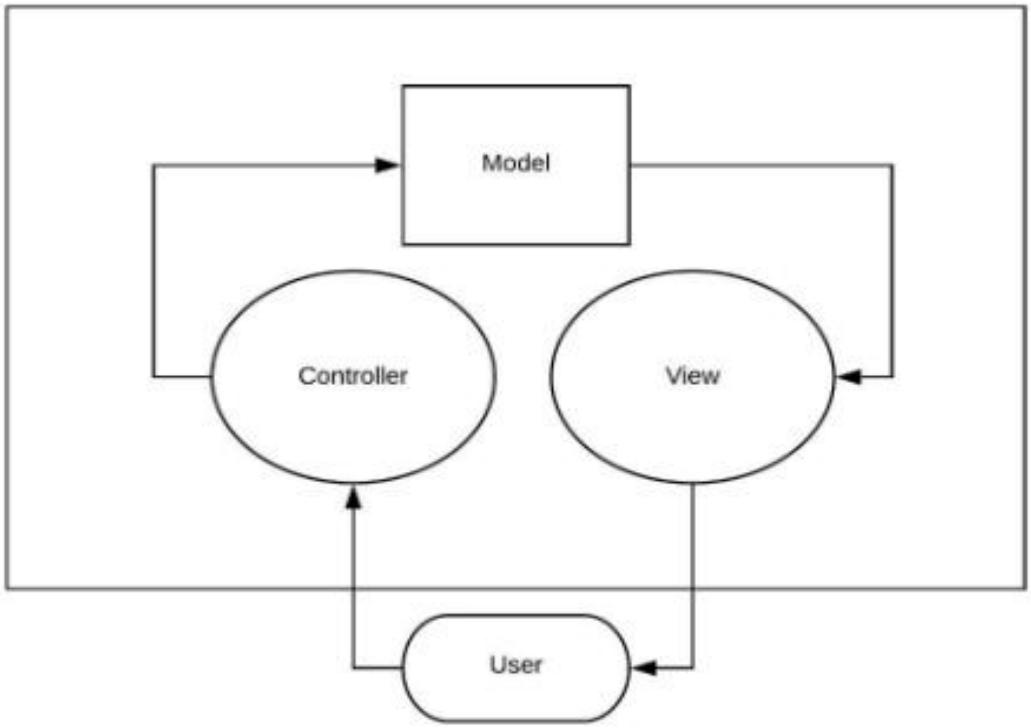
Will show the UI and how the user should interact with the major functionalities.

- Section 4:

Will show the structural design of the system using Class Diagrams.

2-Software architecture

- Block diagram of the architecture:



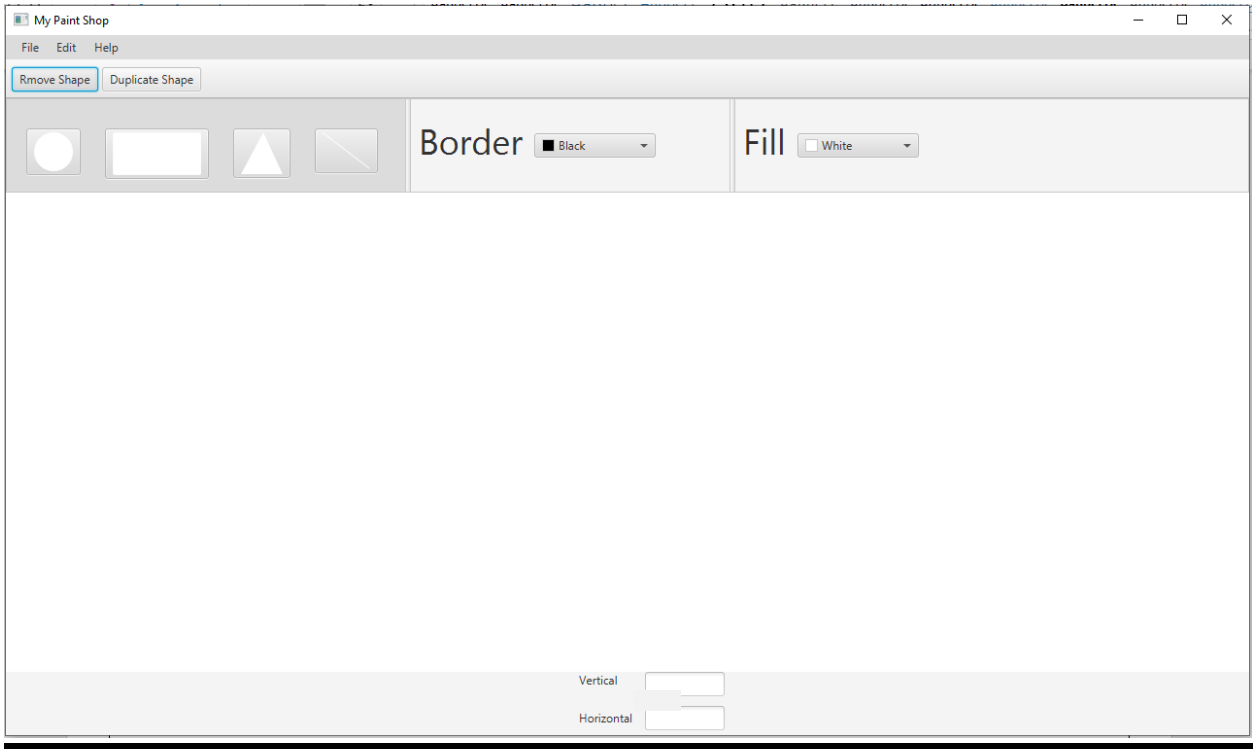
The Model has been chosen for this system is the Model-View-Controller (MVC) architecture due to the following reasons:

- All the processes can be written in the model sub-system as pure code.
- The graphical user interface deals only with the view sub-system.
- The MVC make the software easier to maintain and more organized.

My-Paint-Shop program consists of 3 sub-systems:

1. **Model sub-system:**
   This subsystem consists of the method that does the functionality of the application without representing it to the user.
2. **View sub-system:**
   This subsystem is presenting the model's data to the user without involving in the process of showing it to the user or how the user interacting with it will be handled.
3. **Controller sub-system:**
   This subsystem is handling the interactions that has been done by the user in the view such as clicking by mouse or entering data by keyboard. Then, it calls the proper methods that implemented in the Model to process the user interactions.

3-User interface design:



a. Menu bar
   i. File:
      1. New: To create a new page for drawing.
      2. Open: To open pictures and old project and edit on them.
      3. Close: To close the canvas.
      4. Save: To save the work that has been done on the file:
      5. Save as: To save the work that has been done in different file.
      6. Quit: To close the program.

ii. Edit:
    1. Undo: To remove the last change that the user has done.
    2. Delete: To delete the file.
    3. Cut: To cut a shape.
    4. Copy: To copy a shape.
    5. Paste: To paste what user has copied.
    6. Clear: To clear the canvas.

iii. Help:
    1. About: To see the program details.

b. Tool bar:

i. Shape list which contains 4 shapes.
    1. Ellipse which also can be modified to be a circle by equating the X and Y radius.
    2. Rectangle which also can be modified to be a square.
    3. Triangle.
    4. Straight line.

ii. Border which contains all colors to change the color of the shapes borders.

iii. Fill which contains all colors and it is function to change the color of the shape itself.

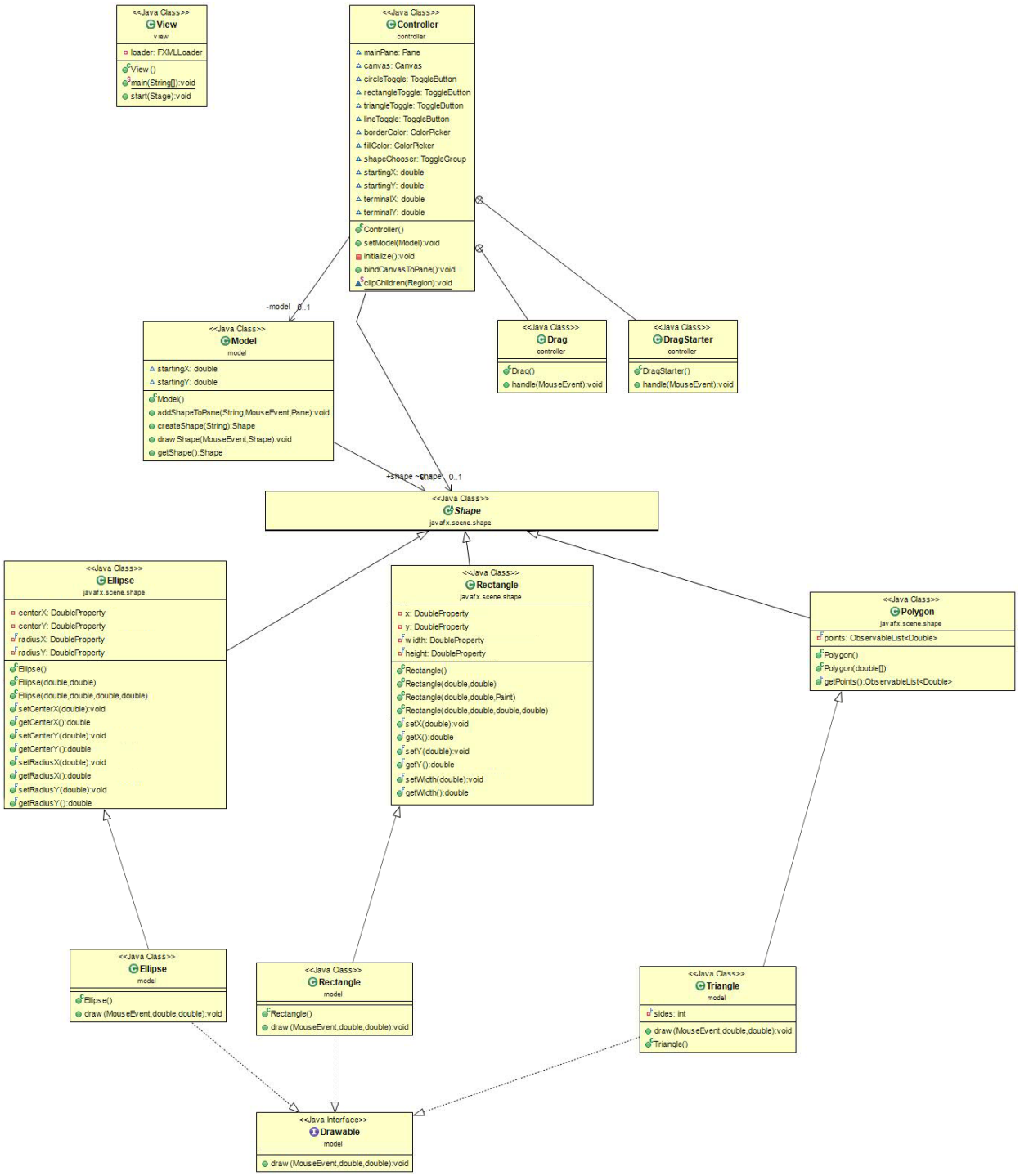
c. Properties panel:

i. Horizontal: To show the width of the shape.

ii. Vertical: To show the length of the shape.

Components design:

a. Class diagram for “myPaintShop”:



b. Classes description:

- Abstract classes:
  - Shape:
    - Which provided by JavaFx and has the following method that we used:
      - setFill: which Sets the value of the property fill that fill the shape color.
      - setStroke: which Sets the value of the property stroke to change the color of the border.
- Concrete classes:
  - Model:
    - Which define the logic of adding/removing/selecting and other operation on the shapes and has the following methods and attributes:
      - startingX: The X coordinate of the starting point.
      - startingY: The Y coordinate of the starting point.
      - CreateShape: creates a shape with width and length of 0
      - AddShapeToPane: Adds shape to the parent node (pane.)
      - DrawShape: the methods that draws the shape according to the

starting location of the mouse and current location of the mouse.
- getShape: returns the shape.

- Controller:
  - Which handle the user interaction with the view:
    - mainPane: the node that contains the shapes.
    - canvas: used for free-drawing.
    - EllipseToggle: button to select the ellipse
    - rectangleToggle: button to select the rectangle
    - tringleToggle: button to select the triangle
    - borderColor: the color of the border of the shape.
    - fillColor: the fill color of the shape.
    - shapeChooser: the toggle group for the shape toggles, so that one shape can be selected at any given time.
    - startingX: The X coordinate of the starting point.
    - startingY: The Y coordinate of the starting point.
    - terminalX: The X coordinate of the terminal point.

- terminalY: The Y coordinate of the terminal point.
- setModel: method to bind the model to the controller.
- clipChildren: helper method used for ensuring that shapes will not come out of the pane
- bindCanvasToPane: helper method used for resizing the canvas relative to the pane.

- View:
  - Which contains GUI:
- Rectangle:
  - which inherited from javafx.scene.shape.Rectangle and has the following attributes and methods:
    - X: Defines the X coordinate of the starting point of drawing from the upper-left corner.
    - Y: Defines the Y coordinate of the starting point of drawing from the upper-left corner.
    - Width: the width of the rectangle
    - Height: the height of the rectangle.
    - getWidth: which gives you the width of the rectangle.

- getHeight: which gives you the height of the rectangle
- get/set X: gives you the position of X coordinate/ set the value of X coordinate.
- get/set Y: gives you the position of Y coordinate/ set the value of Ycoordinate.
- darw: it draws the shape on the pane.

- Ellipse: which inherited from javafx.scene.shape.Ellipse and has the following attributes and methods:
  - centerX: the X-coordinate of the ellipse center.
  - centerY: the Y-coordinate of the ellipse center.
  - radiusX: the horizontal distance from the center.
  - radiusY: the vertical distance from the center.
  - draw: draws the ellipse on the pane.

- Tringle:
  - which inherited from javafx.scene.shape.Polygon and has the following attributes and methods:
    - points: the X and Y coordinates of the triangle vertices.
    - sides: the number of the triangle sides which is equal to 3.
    - getPoints: return the points.
    - draw: draws the triangle on the pane.

- Interfaces:
  - Cloneable:
    - Which allows duplication of the shape.
  - Drawable:
    - Which has the following method:
      - Draw.
- Exceptions:
  - NegativeValues: which is responsible for handling the exception of negative values width, height and negative values of x and y coordinates.

References:

a. Sommerville, I. (2018). Software Engineering. Hallbergmoos/Germany: Pearson.

b. MVC Design Pattern. (2018, February 8). Retrieved from https://www.geeksforgeeks.org/mvc-design-pattern/.

c. Savitch, W. J., & Mock, K. (2017). *Absolute Java*. Brantford, Ontario: W. Ross MacDonald School Resource Services Library.

Appendix

The Group Meetings:

1. Monday, 18 Nov 2019

We went over the design document template and samples we have and connected them with our project. Also, we determined the model that will be used and how will we implement it. Finally, we discussed how the work will be distributed.

2. Tuesday 19 Nov 2019

We started writing the document starting by the introduction up to the Class diagram and assigned some tasks to the group members.

3. Wednesday 27 Nov 2019

We gathered the members work and added the description for the class diagram components.

In addition to all of these meetings we are communicating with each other through the social apps to track our process in the project.

Percentage of contribution of each member (Leader is in red)

Member	Percentage
Abdullah Alfaifi	**25%**
Fawaz Alesayi	**25%**
Waleed Alfaifi	**25%**
Yazeed Aljohany	**25%**