SQA Advanced Diploma in Computing Software development

Report- Assessment Task 3

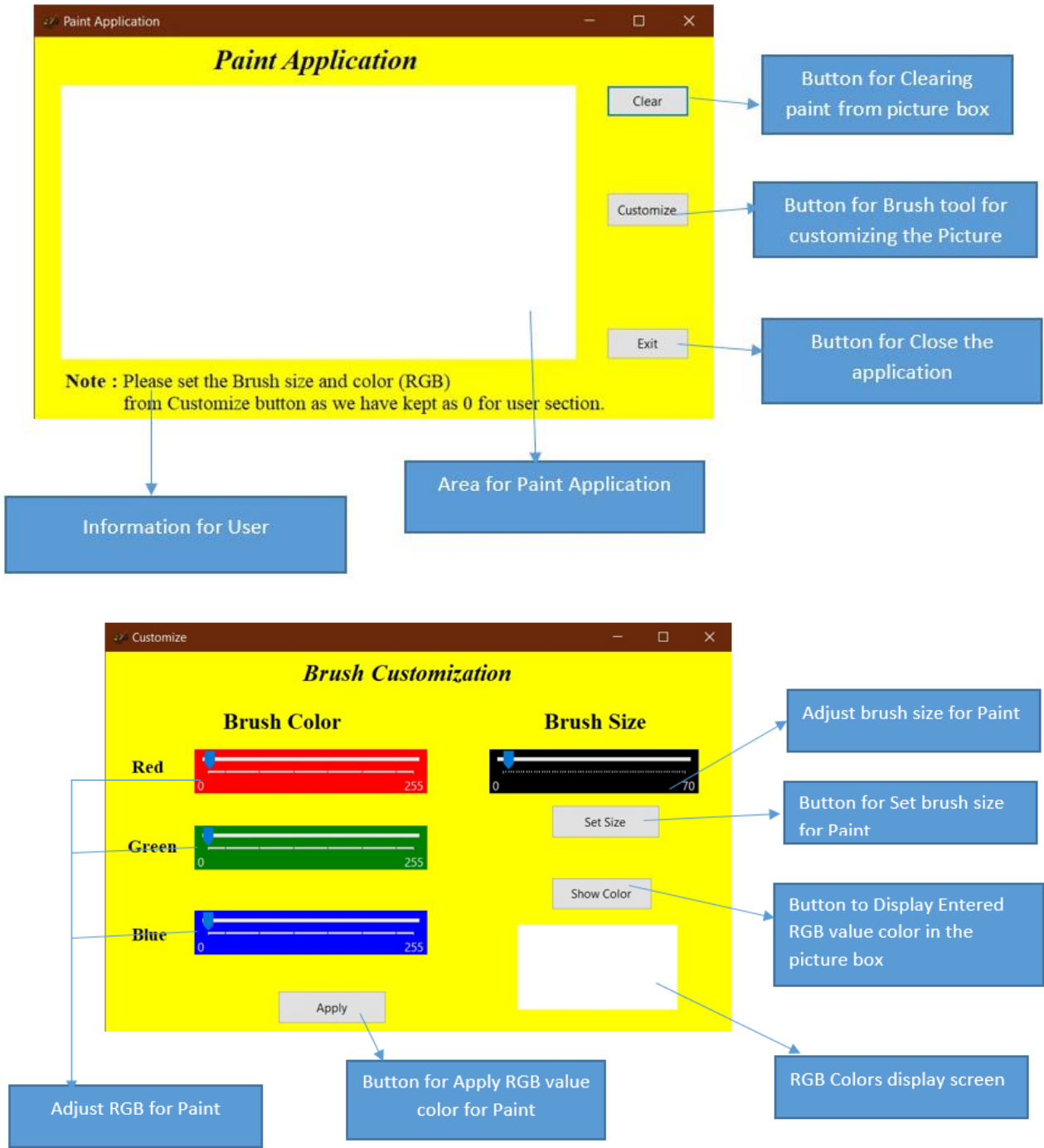**HP2N 47 – Software Development: Developing Small Scale Standalone Applications**

Candidate Name: <u>Nikhilesh Kovvuri</u>

SCN No.: <u>207066915</u>

User Interface Design

Screen layout   **Interactions**

Paint Application

Clear — Button for Clearing paint from picture box

Customize — Button for Brush tool for customizing the Picture

Exit — Button for Close the application

Note : Please set the Brush size and color (RGB) from Customize button as we have kept as 0 for user section.

Information for User

Area for Paint Application

Brush Customization

Brush Color   **Brush Size**

Red   0 — 255

Green   0 — 255

Blue   0 — 255

Apply

Set Size

Show Color

Adjust brush size for Paint

Button for Set brush size for Paint

Button to Display Entered RGB value color in the picture box

RGB Colors display screen

Adjust RGB for Paint

Button for Apply RGB value color for Paint

Code Listing

1. Drawing

```vb
Public Class Drawing
    'Initializing and declaring Variables'
    Dim down = False
    Public mybrush = Brushes.Black
    Public colordot As Integer

    ''' <summary>
    ''' By clicking on Customize button we are directed to second form which is customization
for bursh
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub Customize_button_Click(sender As Object, e As EventArgs) Handles
Customize_button.Click
        Customize.Show()
    End Sub

    ''' <summary>
    ''' this is the  picture box where user will be able to draw
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub PictureBox1_Click(sender As Object, e As EventArgs) Handles Drawing_box.Click

    End Sub

    ''' <summary>
    ''' this is the  picture box command events for mouse to be able to click and draw
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub PictureBox1_MouseDown(sender As Object, e As MouseEventArgs) Handles
Drawing_box.MouseDown
        down = True
    End Sub

    ''' <summary>
    ''' this is the  picture box command events for mouse to be able to click and draw and
    ''' and move in any direction
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>

    Private Sub PictureBox1_MouseMove(sender As Object, e As MouseEventArgs) Handles
Drawing_box.MouseMove
        If down = True Then
            Drawing_box.CreateGraphics.FillEllipse(mybrush, e.X, e.Y, colordot, colordot)
        End If
    End Sub

    ''' <summary>
    ''' this is the  picture box command events for mouse to be able to click and draw
    ''' and move in upwards direction as there would be restriction to draw up sometimes.
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub PictureBox1_MouseUp(sender As Object, e As MouseEventArgs) Handles
Drawing_box.MouseUp
        down = False
    End Sub
```

```vb
''' <summary>
''' clear box clears the done drawing done in the picture box
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>

    Private Sub Clear_button_Click(sender As Object, e As EventArgs) Handles
Clear_button.Click
        Drawing_box.Refresh()

    End Sub

''' <summary>
''' exit button exits the application
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
    Private Sub Exit_button_Click(sender As Object, e As EventArgs) Handles Exit_button.Click
        Me.Close()
    End Sub

End Class
```

2. Customization

```vb
Public Class Customize
    'Declaring and initializing variables
    Public red As Integer
    Public green As Integer
    Public blue As Integer
    Public brushsize As Integer

''' <summary>
''' The method that handles th sliders value when the user uses red trackbar
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
    Private Sub Red_trackbar_Scroll(sender As Object, e As EventArgs) Handles
Red_trackbar.Scroll
        red = Red_trackbar.Value
    End Sub

''' <summary>
''' The method that handles th sliders value when the user uses green trackbar
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
    Private Sub Green_trackbar_Scroll(sender As Object, e As EventArgs) Handles
Green_trackbar.Scroll
        green = Green_trackbar.Value
    End Sub

''' <summary>
''' The method that handles th sliders value when the user uses blue trackbar
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
    Private Sub Blue_trackbar_Scroll(sender As Object, e As EventArgs) Handles
Blue_trackbar.Scroll
        blue = Blue_trackbar.Value
    End Sub
```

```vbnet
    ''' <summary>
    ''' The method that handles the sliders value when the user uses brush size between 0 to
70
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub Brush_trackbar_Scroll(sender As Object, e As EventArgs) Handles
Brushsize_trackbar.Scroll
        brushsize = Brushsize_trackbar.Value
    End Sub

    Private Sub Brush_size_Click(sender As Object, e As EventArgs) Handles Brush_size.Click
        Drawing.colordot = brushsize
    End Sub

    ''' <summary>
    ''' The apply button is used in class customization which update the color value in the
mybrush which in drawing class
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub Apply_button_Click(sender As Object, e As EventArgs) Handles
Apply_button.Click
        Dim brush As Brush
        brush = New SolidBrush(Color.FromArgb(255, red, green, blue))
        Drawing.mybrush = brush
        Me.Hide()
    End Sub

    ''' <summary>
    ''' The display color button displays the rgb color that is choosed and is going to be
displayed
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    Private Sub Display_color_Click(sender As Object, e As EventArgs) Handles
Display_button.Click
        Preview_colorbox.BackColor = Color.FromArgb(red, green, blue)
    End Sub
    End Class
```
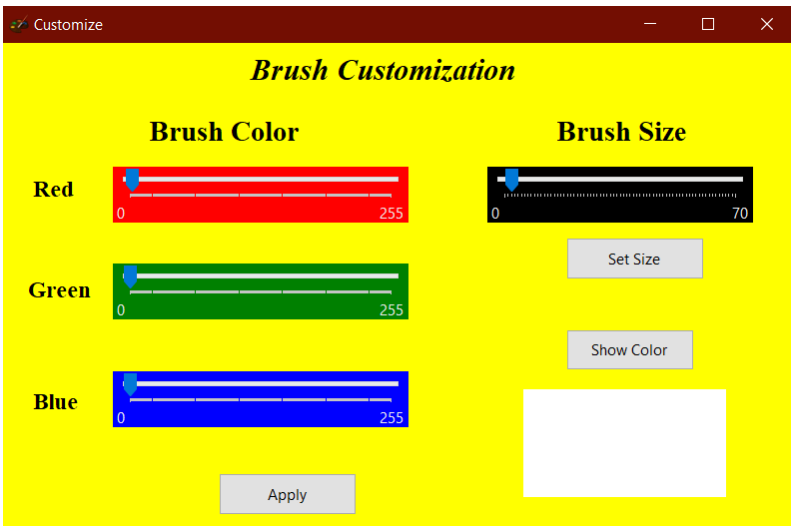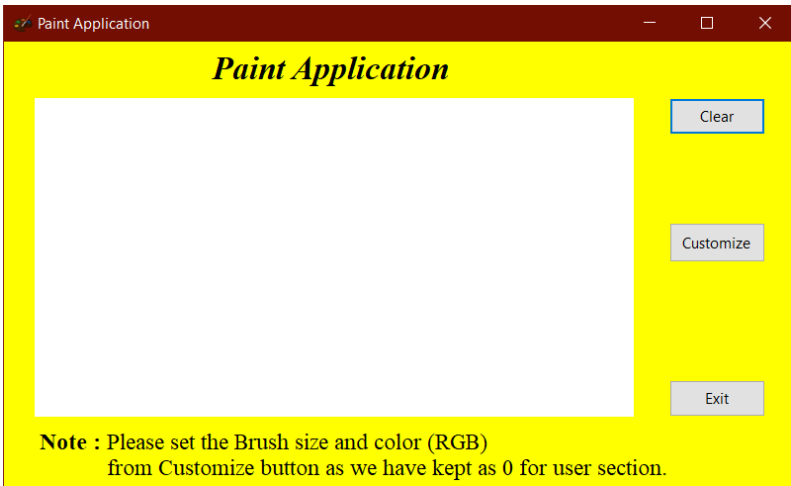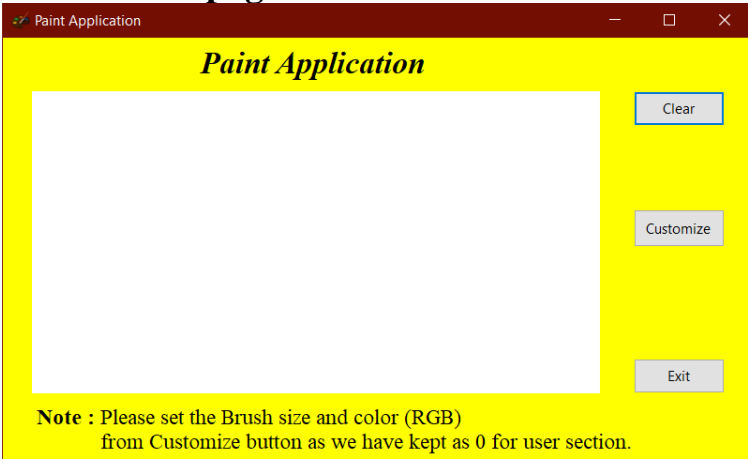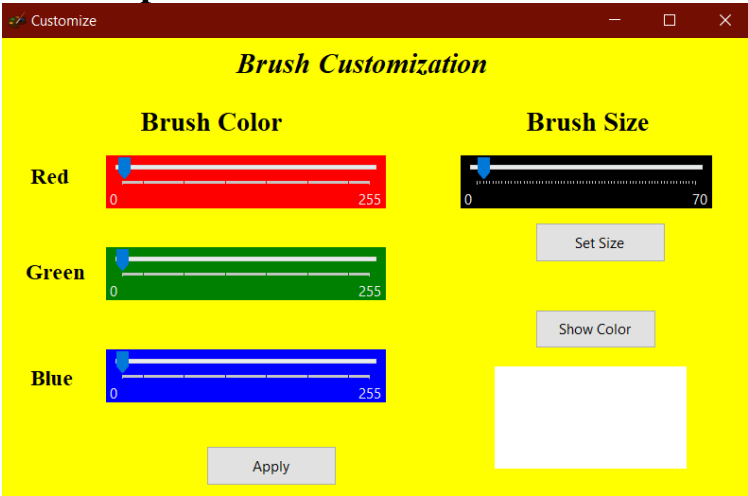
GUI Output:

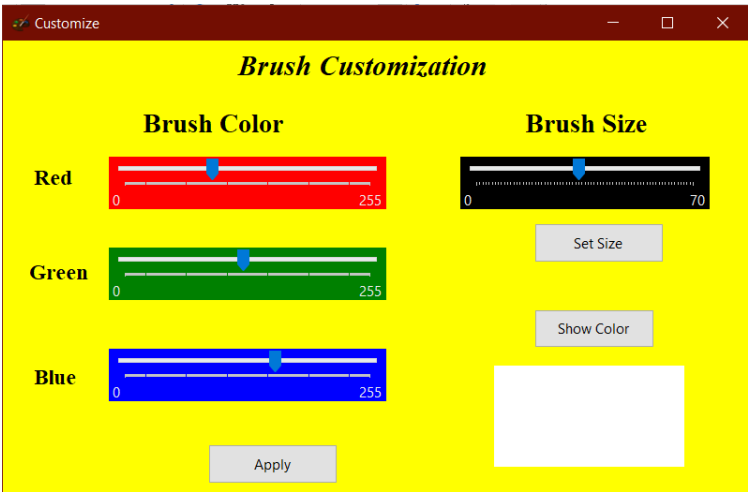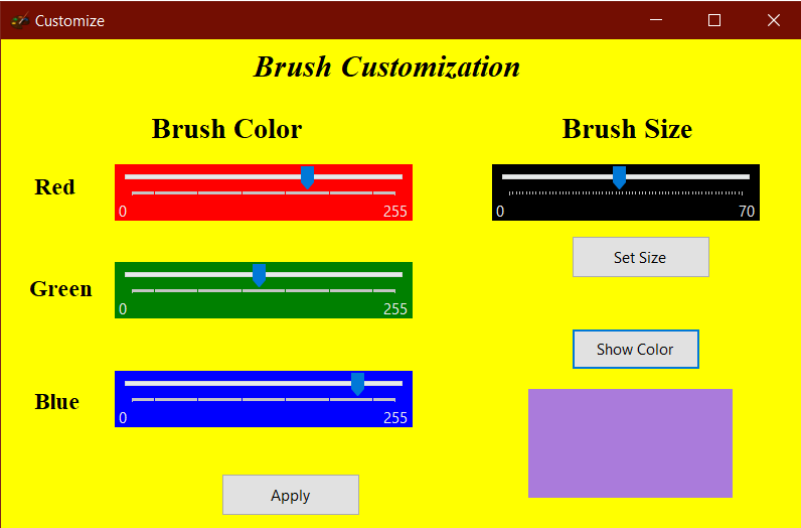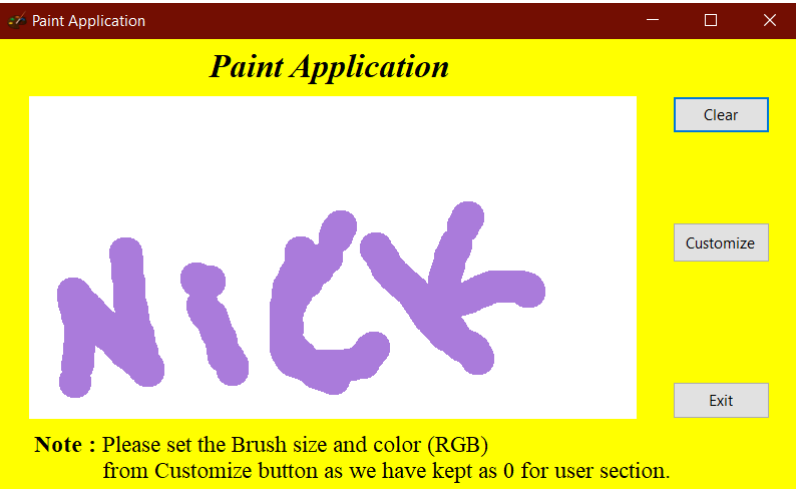# Working Program Screenshot evidence

1. Main page

2. Open Customize
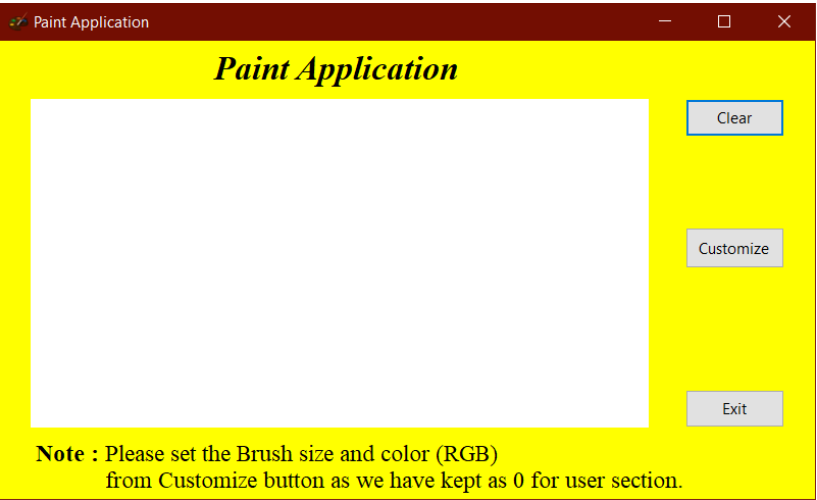

3. Select Rgb values and brush size

4. Click on Show Color to see RGB Color

5. Click on Apply and Draw

6. Click clear button

Usability Test

Questions used in usability test

No	Question	1 (Very Easy) to 5 (Very Difficult)
1	How easy was it to clear the whole paint screen ?	3
2	How easy was it to erase the painting ?	5
3	How easy was it to change RGB values and understand colors ?	3
4	How easy was the painting application ?	1
5	How easy was it to exit the Application ?	1
6	Is the brush size of the application Working and easy to use?	1
7	How easy were you able to draw on the painting screen ?	2

Usability Test Recommendations

Recommendations based on usability test

SL No	Recommendation	Reason
1	Adding Eraser in th application	Most of the users felt it difficult as they had to erase the whole screen.
2	Adding Button for storing Color	Most of the users felt it difficult as they were not able to load previous color.
3	Adding text field for RGB Color values	After the test they was a suggestion about to add text box where the user can enter RGB Values as they were unable to figure out the values from track bar fields.