Name:	Mustafa Chowdhury
Student access ID:	Ge3306
Date (MM/DD/YYYY):	03/26/2019
Group Number (if any):	2

Title of the change request	
Sources:	**Source#1: N/A** **Source#2: N/A**

1. Change Request and concepts*:* (10 points)

Change Request: “Add a slider bar to the bottom of the application for controlling zoom in/out.”

From the change request, I identified following concept:

1. Add
2. Slider bar
3. Bottom
4. Zoom In/Out

Description:

1) Add: is used to communication with programmers. So, it is a irrelevant concept.
2) Slider bar : is the new concept that need to implement, therefore this concept is not in the code and it is an external concept
3) Bottom of the application: Some form of code for this concept already exits in the old program. Therefore, this concept is a significant concept.
4) Zoom In/out: program from zoom in and out already implemented in the code. Therefore, this concept is also a significant concept.

Table 1 Significant Concepts and their details

SN#	Concept Name	Details of how your extracted this concept.
CON1	Bottom	Bottom of the application: Some form of code for this concept already exist in the old program. Therefore, this concept is a significant concept.
CON2	Zoom In/Out	program from zoom in and out already implemented in the code. Therefore, this concept is also a significant concept.

2. Functional requirements: (10 points)

Table 2 Functional Requirements

Requirement#	Functional Requirement Details
FR1: Slider Bar	A sider bar need to be appeared in the bottom of application with a bar.
FR2: Default Bar	Bar of the zoom need to preset at middle of the zoom slider with default window screen of the application
FR3: Move Slider Bar	User shall move slider bar left and right.
FR3: Zoom In	When user move the slider bar to the right, the application window screen shall be zoomed in and its keep zoom in until user stop moving slider bar to right.
FR4: Zoom Out	When user move the slider bar to the left, the application window screen should be zoomed out and its keep zoom out until user stop moving slider bar.
FR5: Move Zoom Slider Bar at Any Direction	User can move the bar with any direction they want, and the zoom bar shall need to be worked as the zoom bar moved in the zoom slider.

3. Concept Location:

.

Methodology: (5 points)

First perform the grep search by searching “Bottom” query and only one match is found. But, this match is a build in QT class. Therefore, I started dependency search to look for zoom in and out. Mainwindow have a supplier ImageArea, which have zoom function. ImageArea also have a supplier additionalTools where the zoom function is implemented. So, I figured out that, additionalTools propagating information through ImageArea to MainWindow. Therefore, I look at the MainWindow thoroughly and found that statusBar and PaletteBar is initialized in the bottom of the application, where PaletteBar used QT class. Therefore, I implemented zoomBar inside of the statusBar.

3.1 Dependency Search (Use this section if you have used dependency search) (7 points)

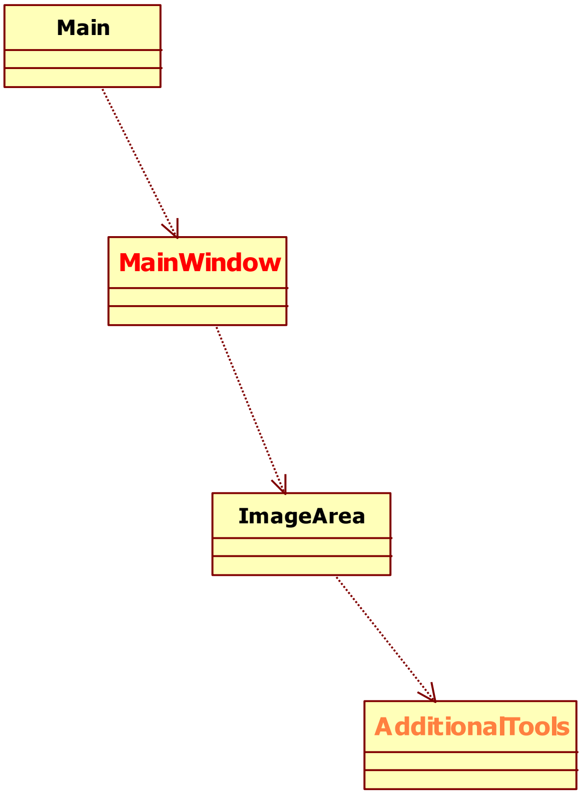
\

Table 3 if Dependency Search is used

Class/file name	Tool used	Mark	Explanation
Main	N/A	Unchanged	Top module
MainWindow	Go to definition/ Peek Definition	Located	At bottom of the application, the paletteBar and statusBar is used. Where paletteBar used addToolBar() function to show the paletteBar at the bottom. In statusBar in all the label is showed of bottom of the application. So, I added my code in status bar, which is showed in the bottom of the application.
ImageArea	Go to Definition	Unchanged	There is no function or variable that is used for bottom so the application. But, a function zoomImage() is found, which used supplier class additionalTools function.
AdditionalTools	Go to Definition	Propagating	ZoomImage() function is implemented but no code need to modify inside of this function. This

			function passing information to other classes

Partial Class Dependency Graph (UML): (3 points)



3.2 Grep Search (Use this section if you have used grep search) (10 points)

Table 4 If Grep Search is used

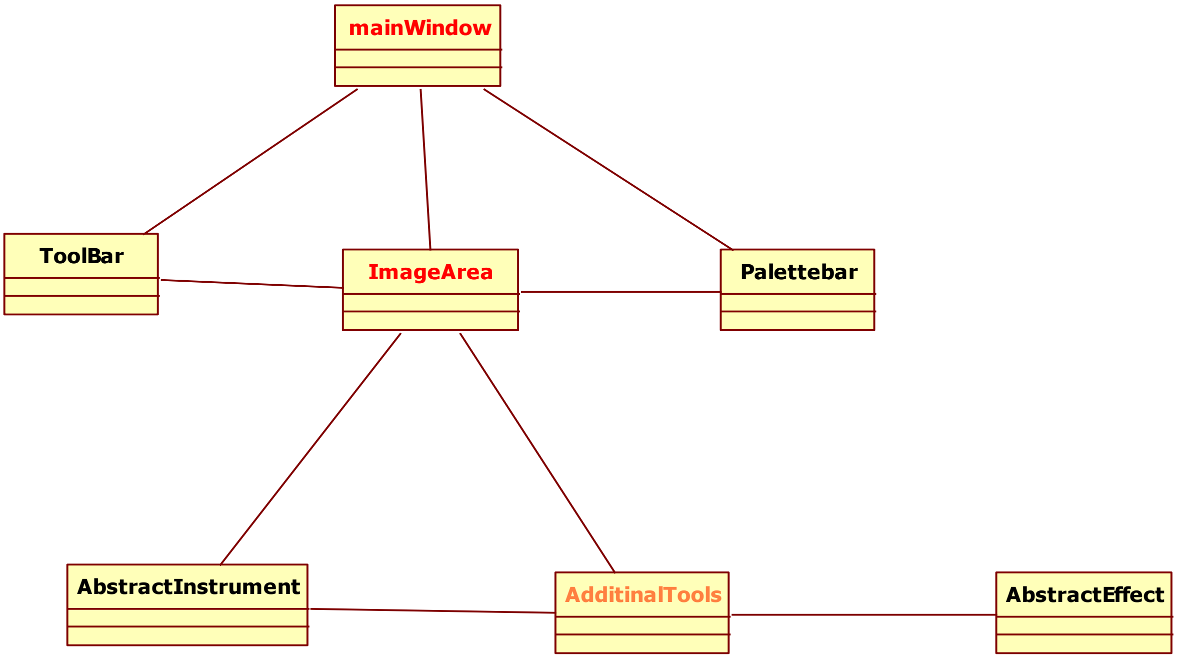
Concept	Query	#Results	Target class/file	Tool used	Explanation
Bottom of the application	*Bottom*	No Match	Main.cpp	ripgrep search	Top module
Bottom of the application	*Bottom*	One and only match found.	mainWindow .cpp	ripgrep search	Inside initializePaletteBar() function Qt:BottomToolBarArea is used

4. Impact Analysis: (10 points)

Table 5 The list of all the classes visited during impact analysis

Class name	Tool used	Mark	Explanation
MainWindow	View all reference / Go to Definition / Declaration	Impacted	Initial Impact set where the concept located.
ImageArea	View all reference / Go to Definition / Declaration / Peek Definition	Impacted	Inside of ZoomInAct(), ZoomOutAct() functions in mainWindow.cpp, supplier class ImageArea's function is called which change the zoom position of image area.
AdditionalTools	View all reference / Go to Definition / Declaration	Propagating	Inside of zoomImage() in ImageArea.cpp, supplier additionalTools function is called to update the zoom factor. So, whenever the slider moved, the zoom factor is also changed.
AbstractInstrument	Go to Declaration	Unchanged	AbstractInstrument doesn't have anything about concept.
AbstractEffect	Go to Declaration	Unchanged	AbstractEffect doesn't have anything about concept.
ToolBar	Go to Declaration	Unchanged	Toolbar doesn't have anything about concept.
PaletteBar	Go to Declaration	Unchanged	PaletteBar doesn't have anything about concept.

Partial class interaction graph (use starUML): (5 points)



5. Prefactoring: (5 points)

Table 6 Prefactoring Code Files

File Name	Refactoring Issue	Lines of Code		
		Added	Deleted	Total
N/A				

N/A

6. Actualization: (10 points)

Table 7 Actualization Summary

Code Files					
Visited#	Changed#	Added#	Propagating#	Unchanged#	Added to Changed Set#
1	1	1	2	0	1

Table 8 Actualization Code Files

File Name	Task	Lines of Code		
		Added	Deleted	Total
MainWindow.h / MainWindow.cpp	Created object of class QSlider and a function to update the zoom interval.	52	0	52

7. Postfactoring: (5 points)

Table 9 Postfactoring Code Files

File Name	Refactoring Issue	Lines of Code		
		Added	Deleted	Total
N/A				

N/A

8. Verification: (15 points)

I created a clone of the zoomSlider function, that I was created in mainWindow.cpp for controlling zoom level. Instead of void function, I make it as a return function for checking the zoomLevel is updated or not after passing a parameter.

```
int zoomSlider(int zoom)
{
       if (zoom >= 1 && zoom <= 5)
       {
              if (zoom > zoomLevel)
              {
                     //zoomInAct(); // calling function from zoom in
                     zoomLevel = zoom; // update zoomLevel by zoom
                     return zoomLevel;
              }
              else
              {
                     //zoomOutAct(); // calling function for zoom out
                     zoomLevel = zoom;// update zoomLevel by zoom
                     return zoomLevel;
              }
       }
       else // this else used to handle error for zoomLevel;
       {
              zoom = 3;
              zoomLevel = zoom;
              return zoomLevel;
       }

}
```

Table 10 Statement Verification Summary

File Name	Coverage of Application			Unit Test Failed (Just indicate the SN#)	Bugs Found
	Total statements added	Total statements covered	Statement coverage %		
Example: TestMe.cpp	7	5	71%	UT#1	0
MainWindow.cpp	20	7	35%	0	0
MainWindow.cpp	20	10	50%	0	0
MainWindow.cpp	20	8	40%	0	0

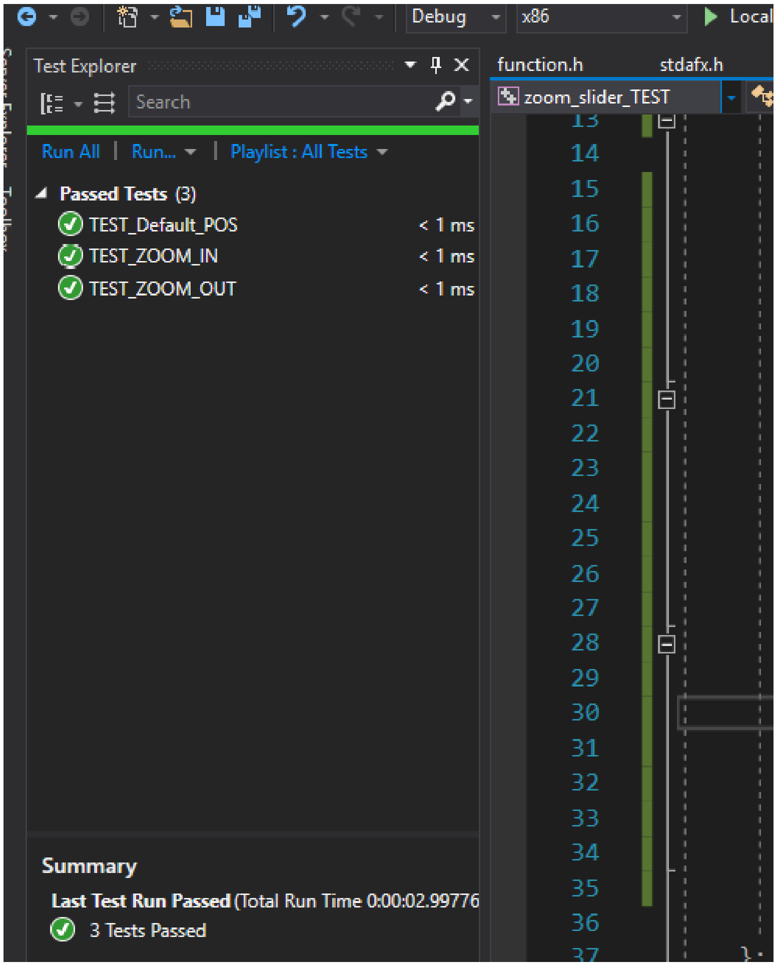
Unit Test Case Details:

```
namespace zoom_slider_TEST
{
       TEST_CLASS(UnitTest1)
       {
       public:

              TEST_METHOD(TEST_Default_POS)
              {
                     int z = 8;
                     int default = 3;
                     int level = zoomSlider(z);

                     Assert::AreEqual(level, default);
              }
              TEST_METHOD(TEST_ZOOM_IN)
              {
                     int z = 4;
                     int level = zoomSlider(z);

                     Assert::AreEqual(level, z);
              }
              TEST_METHOD(TEST_ZOOM_OUT)
              {
                     int z = 1;
                     int level = zoomSlider(z);

                     Assert::AreEqual(level, z);
              }


       };
```

This testing script created a separately because of QT library is unable to open in unit test. First I created exactly a clone of a function that I created to control zoom level and then I test all the zoom level value with the functional requirement.

Test_Default_Pos test the zoom level and bar set in the middle of the slide or not.

Test_Zoom_In test zoom level increased or not if the bar is moved to right.
Test_Zoom_Out test zoom level decreased or not if the bar moved left.



Passed Tests (3)

```
TEST_Default_POS    < 1 ms
TEST_ZOOM_IN        < 1 ms
TEST_ZOOM_OUT       < 1 ms
```

Summary

Last Test Run Passed (Total Run Time 0:00:02.99776

3 Tests Passed

Functional Test Case Details:

After implementing change request “Add a slider bar to the bottom of the application for controlling zoom in/out.” I drew a rectangle and filled the rectangle with green color. Then I move the bar on zoom slider to zoom in and zoom out and its work correctly for my test. Refactoring is not required this time.

9. Highlighted Source Code: (10 points)

```
musta@DESKTOP-ST5QVK3 MINGW64 ~/Desktop/CSC
4110/csc4111w19grp2/Project/sources (master)
$ git diff
```

```
diff --git a/A2/Football/.vs/Football/v15/.suo
b/A2/Football/.vs/Football/v15/.suo
index 616ebf6..82d6a1a 100644
Binary files a/A2/Football/.vs/Football/v15/.suo and
b/A2/Football/.vs/Football/v15/.suo differ
diff --git a/A2/Football/.vs/Football/v15/Browse.VC.db
b/A2/Football/.vs/Football/v15/Browse.VC.db
index 2e02317..e0f685d 100644
Binary files a/A2/Football/.vs/Football/v15/Browse.VC.db and
b/A2/Football/.vs/Football/v15/Browse.VC.db differ
diff --git a/Project/sources/mainwindow.cpp
b/Project/sources/mainwindow.cpp
index 3111807..fbe9884 100644
--- a/Project/sources/mainwindow.cpp
+++ b/Project/sources/mainwindow.cpp
@@ -446,8 +446,56 @@ void MainWindow::initializeStatusBar()
     mStatusBar->addPermanentWidget(mSizeLabel, -1);
     mStatusBar->addPermanentWidget(mPosLabel, 1);
     mStatusBar->addPermanentWidget(mColorPreviewLabel);
+
+    //======================================================
+    // Mustafa
+    // {{
+    mZoomSlider = new QSlider();
+    mZoomSlider->setOrientation(Qt::Horizontal); // set it as
horizontal bar
+    mZoomSlider->setTickInterval(1); // make interval set as 1
+
+    mZoomSlider->setMinimum(1); // inteval start from 1
+    mZoomSlider->setMaximum(5); // interval end at 5
+    mZoomSlider->setSliderPosition(3); // defaulf position for bar is
3
+    mZoomSlider->setPageStep(1); // when user click on slider it
doest not move the bar
+    mZoomSlider->setMaximumWidth(100);
+
+    mStatusBar->addPermanentWidget(mZoomSlider, 1);
+
+    // make a connect, so that when user move the zoom slider bar it
shoud work
+    // sliderMoved() is used because Emitted when the user drags the
slider.
+    connect(mZoomSlider, SIGNAL(valueChanged(int)), this,
SLOT(zoomSlider(int)));
+
+    //  }}
+
     mStatusBar->addPermanentWidget(mColorRGBLabel, -1);
+
+}
+
+//===================================== Mustafa
=================================
+void MainWindow::zoomSlider(int zoom)
+{
+    if (zoom >= 1 && zoom <= 5)
+    {
+        if (zoom > zoomLevel)
+        {
+            zoomInAct(); // calling function from zoom in
+            zoomLevel = zoom; // update zoomLevel by zoom
+        }
+        else
+        {
+            zoomOutAct(); // calling function for zoom out
+            zoomLevel = zoom;// update zoomLevel by zoom
+        }
+    }
+    else // this else used to handle error for zoomLevel;
+    {
+        zoom = 3;
+        zoomLevel = zoom;
+    }
+
+// ===================================== Above part added by Mustafa
======================

diff --git a/Project/sources/mainwindow.h
b/Project/sources/mainwindow.h
index de5b694..96c2db8 100644
--- a/Project/sources/mainwindow.h
+++ b/Project/sources/mainwindow.h
@@ -40,6 +40,7 @@ class PaletteBar;
 class ImageArea;
 class QLabel;
 class QUndoGroup;
+class QSlider;
 QT_END_NAMESPACE

 /**
@@ -105,6 +106,11 @@ private:
     QMenu *mInstrumentsMenu, *mEffectsMenu, *mToolsMenu,
*mSelInstruMenu;
     QUndoGroup *mUndoStackGroup;
     bool mPrevInstrumentSetted; /**< Used for magnifier */
+
+    //----------------- Mustafa ---------------------
+    QSlider *mZoomSlider; // Qslider object is created
+    int zoomLevel = 3;   // private variable zoomlevel and set it as
3;
+    //===============================
```