```
using System.Collections;
using System.IO;
using UnityEngine;
using UnityEngine.UI;
using DG.Tweening;


//must be placed in the main raw image GO;
public class camerademo : MonoBehaviour
{

    #region  INPUTS
    string Patientname;
     string ID;
     int date;
    float x=0.6249899f;
    float y=0.42497f;

    string gender;
     string age;
    string year;
    int months;

    string[]
monthARRAY={"JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"
};


    const string ToastClassName = "android.widget.Toast";


    [Header("ESSENTIALS")]
    public RawImage _rawimage;
    public GameObject toggle;
    public GameObject image;

    public Material material;

    Texture2D texture;


    [Header("SLIDERS")]
    public Slider BlackandwhiteSlider;
    public Slider powerSlider;
    public Slider contrastSlider;



    [Header("USER INPUTS")]
    public InputField patientnameinput;
    public InputField idinput;
    public InputField ageinput;
    public InputField yearinput;
    public Toggle maleToggle,femaleToggle,otherToggle,REDFREE_FilterToggle;
    public Toggle graphtoggle;
    public Toggle   savetoggle;
```

```
[Header("GOs")]
public GameObject uicanvasGO;

public GameObject controlsGO;
public GameObject detailsGO;
public GameObject closebuttonGO;
public GameObject PIC;
public GameObject IMPORT;
public GameObject EXPORT;
public GameObject graph;





[Header("BUTTONS AND TOGGLES")]
public Button savetodevicebtn;
public Button sharebtn;

public Toggle odToggle;
public Toggle osToggle;


public Dropdown monthdropdown;
public Dropdown datedropdown;



[Header("TEXTS")]
public Text datetimeText;


public Text odostext;


#endregion


////////////////////////////////////////////////////////////
private void Start()
{

    Vibration.Init();
    _rawimage=GetComponent<RawImage>();
    uicanvasGO.GetComponent<Canvas>().enabled=true;
    controlsGO.SetActive(false);
    detailsGO.SetActive(false);
    material.SetFloat("Redfreefilter",0);

    BlackandwhiteSlider.value=1;
    powerSlider.value=1.089286f;

    monthdropdown.value=0;
    datedropdown.value=0;

    datetimeText.enabled=false;
    odostext.enabled=false;
    savetodevicebtn.enabled=false;
    sharebtn.enabled=false;


}

/////////////////////////////////////////////////////
private void Update()
{
    date=datedropdown.value;
    date+=1;

    months=monthdropdown.value;
    months+=1;

    Debug.Log(monthARRAY[months]);


     material.SetFloat("BW",BlackandwhiteSlider.value);
     material.SetFloat("power",powerSlider.value);
     material.SetFloat("contrast",contrastSlider.value);

     if(toggle.GetComponent<Toggle>().isOn==true)
     {
         toggle.transform.DOLocalMoveX(50f,0.2f);

        controlsGO.gameObject.SetActive(true);
        material.SetFloat("graph",1);


        EXPORT.gameObject.SetActive(false);

     }

    if(toggle.GetComponent<Toggle>().isOn==false)
     {
         toggle.transform.DOLocalMoveX(-50f,0.2f);

        controlsGO.gameObject.SetActive(false);

        EXPORT.gameObject.SetActive(true);

        material.SetFloat("graph",0);
     }



     if(REDFREE_FilterToggle.isOn)
     {

         material.SetFloat("Redfreefilter",1);
     }
    else
    {
         material.SetFloat("Redfreefilter",0);
    }

    if(graphtoggle.isOn)
    {
        material.SetFloat("graph",1);
        image.gameObject.SetActive(true);
    }
    else
    {
        material.SetFloat("graph",0);
        image.gameObject.SetActive(false);
    }


    if(odToggle.isOn)
    {
        odostext.text="OD";


    }
    if(osToggle.isOn)
    {
        odostext.text="OS";


    }


    if(maleToggle.isOn)
    {

        gender="M";
    }
    if(femaleToggle.isOn)
    {

        gender="F";
    }
    else if(otherToggle.isOn)
    {

        gender="OTHER";
    }



    if(_rawimage.texture!=null)
    {
        if(savetoggle.isOn==true)
        {
            detailsGO.gameObject.SetActive(true);

            PIC.gameObject.SetActive(false);
            IMPORT.gameObject.SetActive(false);
            graph.gameObject.SetActive(false);
             closebuttonGO.gameObject.transform.DOLocalMoveX(320,1.5f);

            closebuttonGO.gameObject.GetComponent<RawImage>().DOFade(1,1.5f);



        }
        if( savetoggle.isOn==false)
```

```
            {

                 detailsGO.gameObject.SetActive(false);

                 PIC.gameObject.SetActive(true);
                IMPORT.gameObject.SetActive(true);
                graph.gameObject.SetActive(true);

                closebuttonGO.gameObject.transform.DOLocalMoveX(-140,1.5f);

            }



            if((Patientname.Length!=0) && (ID.Length!=0) && (gender.Length!=0)
                 && (year.Length!=0) && (age.Length!=0)  )
            {
                savetodevicebtn.enabled=true;
                sharebtn.enabled=true;
            }


        }





    }

    public  void clearbutton()
    {

        Vibration.VibratePop();

        patientnameinput.text="";
        idinput.text="";
        yearinput.text="";

        monthdropdown.value=0;
        datedropdown.value=0;

        maleToggle.isOn=false;
        femaleToggle.isOn=false;
        otherToggle.isOn=false;

        odToggle.isOn=false;
        osToggle.isOn=false;




    }

    public void closebutton()
    {
        detailsGO.gameObject.SetActive(false);

        savetoggle.isOn=false;
```

```
}



void myfunc()
{
    if((Patientname.Length!=0) && (ID.Length!=0) && (gender.Length!=0)
        && (year.Length!=0) && (age.Length!=0) )
    {
        savetodevice();
    }
}



public void Patientnamebutton(string s)
{
    Patientname=patientnameinput.text;
}

public void IDbutton(string s)
{
    ID=idinput.text;
}

public void yearbutton(string s)
{
    year=yearinput.text;
}
public void agebutton(string s)
{
    age=ageinput.text;
}


public void savetodevice()
{
   StartCoroutine(TakeScreenshotAndSave());
   Vibration.VibratePop();
}
public void share()
{
    StartCoroutine(TakeScreenshotAndShare());
    Vibration.VibratePop();
}
public void rotate_Y_button()
{

         y=-y;
        _rawimage.rectTransform.localScale=new Vector3(x,y,1);
}

public void rotate_X_button()
{

        x=-x;
         _rawimage.rectTransform.localScale=new Vector3(x,y,1);
    }


    private void OnDrawGizmos() {

    }

    public void sharebutton()
    {
        StartCoroutine( TakeScreenshotAndShare() );
    }
    private IEnumerator TakeScreenshotAndShare()
{

    uicanvasGO.GetComponent<Canvas>().enabled=false;
    detailsGO.GetComponent<Canvas>().enabled=false;

    datetimeText.text=System.DateTime.Now.ToString();
    datetimeText.enabled=true;
    odostext.enabled=true;

      yield return new WaitForEndOfFrame();


      Texture2D ss = new Texture2D( 710,1000  , TextureFormat.RGB24, false );
      ss.ReadPixels( new Rect( 0,605,605*2,920*2  ), 0, 0 );
      ss.Apply();



      string filePath = Path.Combine( Application.temporaryCachePath, "shared
img.png" );
      File.WriteAllBytes( filePath, ss.EncodeToPNG() );



      new NativeShare().AddFile( filePath )
            .SetSubject( "Subject goes here" ).SetText("("+gender+")" +
Patientname+ " _ "+ ID+"_"+age).SetUrl(
"https://github.com/yasirkula/UnityNativeShare" )
            .SetCallback( ( result, shareTarget ) => Debug.Log( "Share result: " +
result + ", selected app: " + shareTarget ) )
            .Share();


    datetimeText.enabled=false;
    odostext.enabled=false;


    uicanvasGO.GetComponent<Canvas>().enabled=true;
    detailsGO.GetComponent<Canvas>().enabled=true;


      // if( NativeShare.TargetExists( "com.whatsapp" ) )
      //    new
NativeShare().AddFile( filePath ).AddTarget( "com.whatsapp" ).Share();
}
```

```
    public void saveimagebutton()
      {

        detailsGO.gameObject.SetActive(true);
        myfunc();


      }

    private IEnumerator TakeScreenshotAndSave()
{

    uicanvasGO.GetComponent<Canvas>().enabled=false;
    detailsGO.GetComponent<Canvas>().enabled=false;

    datetimeText.text=System.DateTime.Now.ToString();
    datetimeText.enabled=true;
    odostext.enabled=true;

    Vibration.VibratePop();

      yield return new WaitForEndOfFrame();


      Texture2D ss = new Texture2D(  710,1000, TextureFormat.RGB24, false );


      ss.ReadPixels( new Rect( 0, 605, 605*2,920*2 ), 0, 0 );


      ss.Apply();





      NativeGallery.Permission permission = NativeGallery.SaveImageToGallery( ss,
"Retinography","("+gender+")" + Patientname+ " _ "+ ID+"_"+age+".png", ( success,
path ) => Debug.Log( "Media save result: " + success + " " + path ) );

    datetimeText.enabled=false;
    odostext.enabled=false;




    uicanvasGO.GetComponent<Canvas>().enabled=true;
    detailsGO.GetComponent<Canvas>().enabled=true;

      StartCoroutine(OnMyButtonClick());
}
    // IMAGE SAVED TOAST
    IEnumerator   OnMyButtonClick()
      {
            yield return new WaitForSeconds(0.5f);

            var toastJavaClass = new AndroidJavaClass(ToastClassName);

            const int duration = 1; // LENGTH_LONG in Android API
            const string text = "IMAGE SAVED "; // C# string is automatically
converted to java.lang.String
```

```
            var context = GetUnityActivity();
            var javaToastObject =
                  toastJavaClass.CallStatic<AndroidJavaObject>("makeText", context,
text, duration);
            javaToastObject.Call("show");

            toastJavaClass.Dispose();
      }

      AndroidJavaObject GetUnityActivity()
      {
            using (var unityPlayer = new
AndroidJavaClass("com.unity3d.player.UnityPlayer"))
            {
                  return
unityPlayer.GetStatic<AndroidJavaObject>("currentActivity");
            }
      }
    // IMAGE SAVED TOAST \\\\\\\\\\\\\\\



    public void loadbutton()
    {
        LoadImage(512);
        Vibration.VibratePop();
    }


private void LoadImage( int maxSize )
{
      NativeGallery.Permission permission =
NativeGallery.GetImageFromGallery( ( path ) =>
      {
            Debug.Log( "Image path: " + path );
            if( path != null )
            {
                  // Create Texture from selected image
                  Texture2D texture = NativeGallery.LoadImageAtPath( path,
maxSize );
                  if( texture == null )
                  {
                        Debug.Log( "Couldn't load texture from " + path );
                        return;
                  }


                _rawimage.texture=texture;
                material.SetTexture("_MainTex",_rawimage.texture);
                _rawimage.gameObject.SetActive(true);


            }
      } );


}
    public void TakePicturebutton()
    {
        TakePicture(512);
        Vibration.VibratePop();
    }


    private void TakePicture( int maxSize )
{
    NativeCamera.Permission permission = NativeCamera.TakePicture( ( path ) =>
    {

        if( path != null )
        {
            // Create a Texture2D from the captured image
             texture = NativeCamera.LoadImageAtPath( path, maxSize );
            if( texture == null )
            {

                return;
            }


            _rawimage.texture=texture;
            material.SetTexture("_MainTex",_rawimage.texture);
            _rawimage.gameObject.SetActive(true);


        }
    }, maxSize );


}
}
```