Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Tehnologii Informaţionale

RAPORT

Lucrarea de laborator : 5

la Medii Interactive de Dezvoltare a Produselor Soft

Tema: Dezvoltarea unei aplicatii mobile

A efectuat: Cheptanaru Anatolie
st.gr. TI – 143

A verificat: Cojocaru Svetlana
Lect.Univer..

Chişinău 2016

Scopul lucrării:

Dezvoltarea unei aplicatii mobile

Visual Studio
Xcode
Android Studio
Eclipse
NetBeans

Prerequisites:

IDEs: Visual Studio, Xcode, Android Studio, Eclipse, NetBeans
Limbaje de programare: C#, JavaScript, Objective C, Java, Swift
Tehnologii si Frameworks: Windows Mobile, iOS, Android

Obiective:

Cunostinte de baza privina arhitectura unei aplicatii mobile
Cunostinte de baza ale platformei SDK

Conditii Generale:

Se considera ca ai trecut cu succes laboratorul daca ai urmat toti pasii din:

1. Submission Process
2. Trebuie sa elaborezi un program prototip care il vei arata in timpul laboratorului
3. Ai respectat DL (data limita)

Technical Prerequisites:

Your application must be developed and tested in SDK included Emulator. You probably would like to run your application on real device.
Your application must support multiple screen resolutions.

Laboratory Requirements:

Basic Level (nota 5 || 6) :

- Realizeaza o aplicatie simpla "Hello world" care va contine 2 butoane care vor afisa 2 pagini diferite, folosing 2 elemente diferite de interactiune

Normal Level (nota 7 || 8):

- Implimenteaza un simplu ceas sau stopwatch

Advanced Level (nota 9 || 10):

- Realizeaza o aplicatie care va implimenta tehnica *Pomodoro* **SAU**
- O alta aplicatie sofisticata la alegere
  - Game

Bonus Point

- Foloseste libraria cross platform pentru a realiza o apliacatie cross platform (aplicatia poate fi compilata atit pe Android, cit si pe iOS)
- Folosirea Facebook/Twitter/Google Maps API

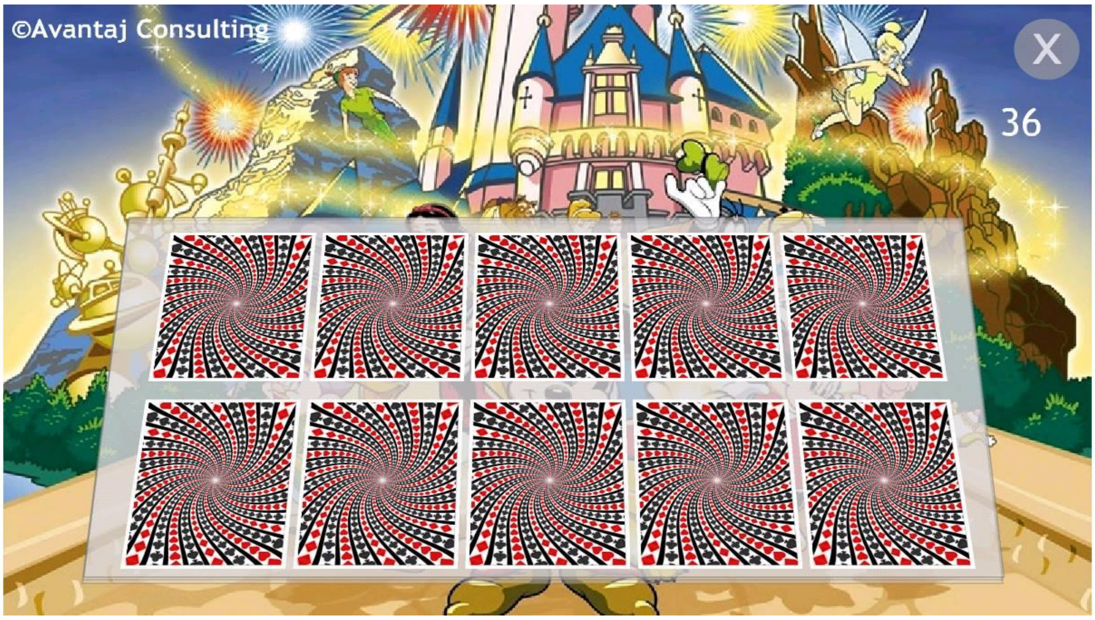
Cod Sursa:

```
package air.jocmemotimer;

import android.R.id;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.app.Dialog;
import android.content.ActivityNotFoundException;
import android.content.ComponentName;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnCancelListener;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.content.ServiceConnection;
import android.content.pm.PackageManager;
import android.content.pm.PackageManager.NameNotFoundException;
import android.content.res.Configuration;
import android.content.res.Resources.Theme;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Process;
import android.util.AttributeSet;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuInfo;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager.LayoutParams;
import dalvik.system.DexClassLoader;
import java.io.File;
import java.lang.reflect.Method;
import java.net.URISyntaxException;

public class AppEntry extends Activity
{
  private static final String LOG_TAG = "AppEntry";
  private static String RUNTIME_PACKAGE_ID;
  private static Object sAndroidActivityWrapper;
  private static Class<?> sAndroidActivityWrapperClass;
  private static boolean sDexLoaded = false;
  private static DexClassLoader sDloader;

  static
  {
    sAndroidActivityWrapper = null;
    RUNTIME_PACKAGE_ID = "com.adobe.air";
  }

  private Object InvokeMethod(Method paramMethod, Object[] paramArrayOfObject)
  {
    if (!sDexLoaded)
      return null;
    Object localObject1 = null;
    if (paramArrayOfObject != null);
    try
    {
      localObject1 = paramMethod.invoke(sAndroidActivityWrapper,
paramArrayOfObject);
      break label47;
      Object localObject2 = paramMethod.invoke(sAndroidActivityWrapper, new
Object[0]);
      localObject1 = localObject2;
    }
    catch (Exception localException)
    {
    }
    label47: return localObject1;
  }

  private void InvokeWrapperOnCreate()
  {
    try
    {
      Method localMethod = sAndroidActivityWrapperClass.getMethod("onCreate", new
Class[] { Activity.class, [Ljava.lang.String.class });
      Boolean localBoolean1 = new Boolean(false);
```

```
   Boolean localBoolean2 = new Boolean(false);
   String[] arrayOfString = new String[5];
   arrayOfString[0] = "";
   arrayOfString[1] = "";
   arrayOfString[2] = "-nodebug";
   arrayOfString[3] = localBoolean1.toString();
   arrayOfString[4] = localBoolean2.toString();
   InvokeMethod(localMethod, new Object[] { this, arrayOfString });
   return;
  }
  catch (Exception localException)
  {
  }
 }

 private static void KillSelf()
 {
  Process.killProcess(Process.myPid());
 }

 private void launchAIRService()
 {
  try
  {
   Intent localIntent = new Intent("com.adobe.air.AIRServiceAction");
   localIntent.setClassName(RUNTIME_PACKAGE_ID, "com.adobe.air.AIRService");
   bindService(localIntent, new ServiceConnection()
   {
    public void onServiceConnected(ComponentName paramComponentName, IBinder
paramIBinder)
    {
     AppEntry.this.unbindService(this);
     AppEntry.this.loadDexAndCreateActivityWrapper();
     if (AppEntry.sDexLoaded)
     {
      AppEntry.this.InvokeWrapperOnCreate();
      return;
     }
     AppEntry.access$400();
    }

    public void onServiceDisconnected(ComponentName paramComponentName)
    {
    }
   }
   , 1);
   return;
  }
  catch (Exception localException)
  {
  }
 }

 private void loadDexAndCreateActivityWrapper()
 {
  try
  {
   if (!sDexLoaded)
   {
    Context localContext = createPackageContext(RUNTIME_PACKAGE_ID, 3);
    sDloader = new DexClassLoader(RUNTIME_PACKAGE_ID,
getFilesDir().getAbsolutePath(), null, localContext.getClassLoader());
    sAndroidActivityWrapperClass =
sDloader.loadClass("com.adobe.air.AndroidActivityWrapper");
    if (sAndroidActivityWrapperClass != null)
     sDexLoaded = true;
   }
   sAndroidActivityWrapper =
sAndroidActivityWrapperClass.getMethod("CreateAndroidActivityWrapper", new Class[]
{ Activity.class }).invoke(null, new Object[] { this });
   return;
  }
  catch (Exception localException)
  {
  }
 }

 public void BroadcastIntent(String paramString1, String paramString2)
 {
  try
  {
      startActivity(Intent.parseUri(paramString2,
0).setAction(paramString1).addFlags(268435456));
      return;
    }
    catch (ActivityNotFoundException localActivityNotFoundException)
    {
      return;
    }
    catch (URISyntaxException localURISyntaxException)
    {
    }
  }

  public void finishActivityFromChild(Activity paramActivity, int paramInt)
  {
    super.finishActivityFromChild(paramActivity, paramInt);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Activity.class;
      arrayOfClass[1] = Integer.TYPE;
      Method localMethod = localClass.getMethod("finishActivityFromChild",
arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramActivity;
      arrayOfObject[1] = Integer.valueOf(paramInt);
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void finishFromChild(Activity paramActivity)
  {
    super.finishFromChild(paramActivity);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("finishFromChild", new
Class[] { Activity.class }), new Object[] { paramActivity });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean isRuntimeInstalled()
  {
    PackageManager localPackageManager = getPackageManager();
    try
    {
      localPackageManager.getPackageInfo(RUNTIME_PACKAGE_ID, 256);
      return true;
    }
    catch (PackageManager.NameNotFoundException localNameNotFoundException)
    {
    }
    return false;
  }

  public void launchMarketPlace(String paramString)
  {
    String str = "market://details?id=" + paramString;
    try
    {
      BroadcastIntent("android.intent.action.VIEW", str);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onActivityResult(int paramInt1, int paramInt2, Intent paramIntent)
  {
    try
    {
      if (sDexLoaded)
      {
        Class localClass = sAndroidActivityWrapperClass;
        Class[] arrayOfClass = new Class[3];
        arrayOfClass[0] = Integer.TYPE;
        arrayOfClass[1] = Integer.TYPE;
        arrayOfClass[2] = Intent.class;
        Method localMethod = localClass.getMethod("onActivityResult", arrayOfClass);
        Object[] arrayOfObject = new Object[3];
        arrayOfObject[0] = Integer.valueOf(paramInt1);
        arrayOfObject[1] = Integer.valueOf(paramInt2);
        arrayOfObject[2] = paramIntent;
        InvokeMethod(localMethod, arrayOfObject);
      }
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onApplyThemeResource(Resources.Theme paramTheme, int paramInt,
boolean paramBoolean)
  {
    super.onApplyThemeResource(paramTheme, paramInt, paramBoolean);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Resources.Theme.class;
      arrayOfClass[1] = Integer.TYPE;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onApplyThemeResource",
arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = paramTheme;
      arrayOfObject[1] = Integer.valueOf(paramInt);
      arrayOfObject[2] = Boolean.valueOf(paramBoolean);
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onAttachedToWindow()
  {
    super.onAttachedToWindow();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onAttachedToWindow",
new Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onBackPressed()
  {
    super.onBackPressed();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onBackPressed", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onChildTitleChanged(Activity paramActivity, CharSequence
paramCharSequence)
  {
    super.onChildTitleChanged(paramActivity, paramCharSequence);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onChildTitleChanged",
new Class[] { Activity.class, CharSequence.class }), new Object[] { paramActivity,
paramCharSequence });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onConfigurationChanged(Configuration paramConfiguration)
  {
    super.onConfigurationChanged(paramConfiguration);
    try
    {

InvokeMethod(sAndroidActivityWrapperClass.getMethod("onConfigurationChanged",
new Class[] { Configuration.class }), new Object[] { paramConfiguration });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onContentChanged()
  {
    super.onContentChanged();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onContentChanged", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean onContextItemSelected(MenuItem paramMenuItem)
  {
    boolean bool1 = super.onContextItemSelected(paramMenuItem);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = MenuItem.class;
      arrayOfClass[1] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onContextItemSelected",
arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramMenuItem;
      arrayOfObject[1] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public void onContextMenuClosed(Menu paramMenu)
  {
    super.onContextMenuClosed(paramMenu);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onContextMenuClosed",
new Class[] { Menu.class }), new Object[] { paramMenu });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onCreate(Bundle paramBundle)
  {
    super.onCreate(paramBundle);
    if (!isRuntimeInstalled())
    {
      showDialog();
      return;
    }
    loadDexAndCreateActivityWrapper();
    if (!sDexLoaded)
    {
      launchAIRService();
      return;
    }
    InvokeWrapperOnCreate();
  }

  public void onCreateContextMenu(ContextMenu paramContextMenu, View paramView,
ContextMenu.ContextMenuInfo paramContextMenuInfo)
  {
    super.onCreateContextMenu(paramContextMenu, paramView,
paramContextMenuInfo);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateContextMenu",
new Class[] { ContextMenu.class, View.class, ContextMenu.ContextMenuInfo.class }),
new Object[] { paramContextMenu, paramView, paramContextMenuInfo });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public CharSequence onCreateDescription()
  {
    CharSequence localCharSequence1 = super.onCreateDescription();
    try
    {
      CharSequence localCharSequence2 =
(CharSequence)InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateDesc
ription", new Class[] { CharSequence.class }), new Object[] { localCharSequence1 });
      return localCharSequence2;
    }
    catch (Exception localException)
    {
    }
    return localCharSequence1;
  }

  protected Dialog onCreateDialog(int paramInt)
  {
    Dialog localDialog1 = super.onCreateDialog(paramInt);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = Dialog.class;
      Method localMethod = localClass.getMethod("onCreateDialog", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = localDialog1;
      Dialog localDialog2 = (Dialog)InvokeMethod(localMethod, arrayOfObject);
      return localDialog2;
    }
    catch (Exception localException)
    {
    }
    return localDialog1;
  }

  protected Dialog onCreateDialog(int paramInt, Bundle paramBundle)
  {
    Dialog localDialog1 = super.onCreateDialog(paramInt, paramBundle);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = Bundle.class;
      arrayOfClass[2] = Dialog.class;
      Method localMethod = localClass.getMethod("onCreateDialog", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramBundle;
      arrayOfObject[2] = localDialog1;
      Dialog localDialog2 = (Dialog)InvokeMethod(localMethod, arrayOfObject);
      return localDialog2;
    }
    catch (Exception localException)
    {
    }
    return localDialog1;
  }

  public boolean onCreateOptionsMenu(Menu paramMenu)
  {
    boolean bool1 = super.onCreateOptionsMenu(paramMenu);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Menu.class;
      arrayOfClass[1] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onCreateOptionsMenu", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramMenu;
      arrayOfObject[1] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public boolean onCreatePanelMenu(int paramInt, Menu paramMenu)
  {
    boolean bool1 = super.onCreatePanelMenu(paramInt, paramMenu);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = Menu.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onCreatePanelMenu", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramMenu;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public View onCreatePanelView(int paramInt)
  {
    View localView1 = super.onCreatePanelView(paramInt);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = View.class;
      Method localMethod = localClass.getMethod("onCreatePanelView", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = localView1;
      View localView2 = (View)InvokeMethod(localMethod, arrayOfObject);
      return localView2;
    }
    catch (Exception localException)
    {
    }
    return localView1;
  }

  public boolean onCreateThumbnail(Bitmap paramBitmap, Canvas paramCanvas)
  {
    boolean bool1 = super.onCreateThumbnail(paramBitmap, paramCanvas);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Bitmap.class;
      arrayOfClass[1] = Canvas.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onCreateThumbnail", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = paramBitmap;
      arrayOfObject[1] = paramCanvas;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public View onCreateView(String paramString, Context paramContext, AttributeSet
paramAttributeSet)
  {
    View localView1 = super.onCreateView(paramString, paramContext,
paramAttributeSet);
    try
    {
      View localView2 =
(View)InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateView", new
Class[] { String.class, Context.class, AttributeSet.class, View.class }), new Object[] {
paramString, paramContext, paramAttributeSet, localView1 });
      return localView2;
    }
    catch (Exception localException)
    {
    }
    return localView1;
  }

  public void onDestroy()
  {
    super.onDestroy();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onDestroy", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onDetachedFromWindow()
  {
    super.onDetachedFromWindow();
    try
    {

InvokeMethod(sAndroidActivityWrapperClass.getMethod("onDetachedFromWindow",
new Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean onKeyDown(int paramInt, KeyEvent paramKeyEvent)
  {
    boolean bool1 = super.onKeyDown(paramInt, paramKeyEvent);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = KeyEvent.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onKeyDown", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramKeyEvent;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public boolean onKeyLongPress(int paramInt, KeyEvent paramKeyEvent)
  {
    boolean bool1 = super.onKeyLongPress(paramInt, paramKeyEvent);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = KeyEvent.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onKeyLongPress", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramKeyEvent;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public boolean onKeyMultiple(int paramInt1, int paramInt2, KeyEvent paramKeyEvent)
  {
    boolean bool1 = super.onKeyMultiple(paramInt1, paramInt2, paramKeyEvent);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[4];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = Integer.TYPE;
      arrayOfClass[2] = KeyEvent.class;
      arrayOfClass[3] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onKeyMultiple", arrayOfClass);
      Object[] arrayOfObject = new Object[4];
      arrayOfObject[0] = Integer.valueOf(paramInt1);
      arrayOfObject[1] = Integer.valueOf(paramInt2);
      arrayOfObject[2] = paramKeyEvent;
      arrayOfObject[3] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public boolean onKeyUp(int paramInt, KeyEvent paramKeyEvent)
  {
    boolean bool1 = super.onKeyUp(paramInt, paramKeyEvent);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = KeyEvent.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onKeyUp", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramKeyEvent;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public void onLowMemory()
  {
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onLowMemory", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean onMenuItemSelected(int paramInt, MenuItem paramMenuItem)
  {
    boolean bool1 = super.onMenuItemSelected(paramInt, paramMenuItem);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = MenuItem.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onMenuItemSelected", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramMenuItem;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public boolean onMenuOpened(int paramInt, Menu paramMenu)
  {
    boolean bool1 = super.onMenuOpened(paramInt, paramMenu);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[3];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = Menu.class;
      arrayOfClass[2] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onMenuOpened", arrayOfClass);
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramMenu;
      arrayOfObject[2] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  protected void onNewIntent(Intent paramIntent)
  {
    super.onNewIntent(paramIntent);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onNewIntent", new
Class[] { Intent.class }), new Object[] { paramIntent });
      return;
    }
    catch (Exception localException)
    {
    }
  }
  public boolean onOptionsItemSelected(MenuItem paramMenuItem)
  {
    boolean bool1 = super.onOptionsItemSelected(paramMenuItem);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = MenuItem.class;
      arrayOfClass[1] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onOptionsItemSelected",
arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramMenuItem;
      arrayOfObject[1] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public void onOptionsMenuClosed(Menu paramMenu)
  {
    super.onOptionsMenuClosed(paramMenu);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onOptionsMenuClosed",
new Class[] { Menu.class }), new Object[] { paramMenu });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onPanelClosed(int paramInt, Menu paramMenu)
  {
    super.onPanelClosed(paramInt, paramMenu);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = Menu.class;
      Method localMethod = localClass.getMethod("onPanelClosed", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramMenu;
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onPause()
  {
    super.onPause();
    try
    {
      if (sDexLoaded)
        InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPause", new Class[0]),
new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onPostCreate(Bundle paramBundle)
  {
    super.onPostCreate(paramBundle);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPostCreate", new
Class[] { Bundle.class }), new Object[] { paramBundle });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onPostResume()
  {
    super.onPostResume();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPostResume", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onPrepareDialog(int paramInt, Dialog paramDialog)
  {
    super.onPrepareDialog(paramInt, paramDialog);
    try
    {
      Method localMethod = sAndroidActivityWrapperClass.getMethod("onPrepareDialog",
new Class[] { R.id.class, Dialog.class });
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramDialog;
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onPrepareDialog(int paramInt, Dialog paramDialog, Bundle
paramBundle)
  {
    super.onPrepareDialog(paramInt, paramDialog, paramBundle);
    try
    {
      Method localMethod = sAndroidActivityWrapperClass.getMethod("onPrepareDialog",
new Class[] { R.id.class, Dialog.class, Bundle.class });
      Object[] arrayOfObject = new Object[3];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramDialog;
      arrayOfObject[2] = paramBundle;
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean onPrepareOptionsMenu(Menu paramMenu)
  {
    boolean bool1 = super.onPrepareOptionsMenu(paramMenu);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Menu.class;
      arrayOfClass[1] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onPrepareOptionsMenu",
arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramMenu;
      arrayOfObject[1] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }
  public boolean onPreparePanel(int paramInt, View paramView, Menu paramMenu)
  {
    boolean bool1 = super.onPreparePanel(paramInt, paramView, paramMenu);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[4];
      arrayOfClass[0] = Integer.TYPE;
      arrayOfClass[1] = View.class;
      arrayOfClass[2] = Menu.class;
      arrayOfClass[3] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onPreparePanel", arrayOfClass);
      Object[] arrayOfObject = new Object[4];
      arrayOfObject[0] = Integer.valueOf(paramInt);
      arrayOfObject[1] = paramView;
      arrayOfObject[2] = paramMenu;
      arrayOfObject[3] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public void onRestart()
  {
    super.onRestart();
    try
    {
      if (sDexLoaded)
        InvokeMethod(sAndroidActivityWrapperClass.getMethod("onRestart", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onRestoreInstanceState(Bundle paramBundle)
  {
    super.onRestoreInstanceState(paramBundle);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onRestoreInstanceState",
new Class[] { Bundle.class }), new Object[] { paramBundle });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onResume()
  {
    super.onResume();
    try
    {
      if (sDexLoaded)
        InvokeMethod(sAndroidActivityWrapperClass.getMethod("onResume", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public Object onRetainNonConfigurationInstance()
  {
    Object localObject1 = super.onRetainNonConfigurationInstance();
    try
    {
      Object localObject2 =
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onRetainNonConfigurationIns
tance", new Class[] { Object.class }), new Object[] { localObject1 });
      return localObject2;
    }
    catch (Exception localException)
    {
    }
    return localObject1;
  }

  protected void onSaveInstanceState(Bundle paramBundle)
  {
    super.onSaveInstanceState(paramBundle);
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onSaveInstanceState",
new Class[] { Bundle.class }), new Object[] { paramBundle });
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean onSearchRequested()
  {
    boolean bool1 = super.onSearchRequested();
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[1];
      arrayOfClass[0] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onSearchRequested", arrayOfClass);
      Object[] arrayOfObject = new Object[1];
      arrayOfObject[0] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public void onStart()
  {
    super.onStart();
  }

  public void onStop()
  {
    super.onStop();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onStop", new Class[0]),
new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onTitleChanged(CharSequence paramCharSequence, int paramInt)
  {
    super.onTitleChanged(paramCharSequence, paramInt);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = CharSequence.class;
      arrayOfClass[1] = Integer.TYPE;
      Method localMethod = localClass.getMethod("onTitleChanged", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramCharSequence;
      arrayOfObject[1] = Integer.valueOf(paramInt);
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public boolean onTouchEvent(MotionEvent paramMotionEvent)
  {
    boolean bool1 = super.onTouchEvent(paramMotionEvent);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = MotionEvent.class;
      arrayOfClass[1] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onTouchEvent", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramMotionEvent;
      arrayOfObject[1] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public boolean onTrackballEvent(MotionEvent paramMotionEvent)
  {
    boolean bool1 = super.onTrackballEvent(paramMotionEvent);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = MotionEvent.class;
      arrayOfClass[1] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onTrackballEvent", arrayOfClass);
      Object[] arrayOfObject = new Object[2];
      arrayOfObject[0] = paramMotionEvent;
      arrayOfObject[1] = Boolean.valueOf(bool1);
      boolean bool2 = ((Boolean)InvokeMethod(localMethod,
arrayOfObject)).booleanValue();
      return bool2;
    }
    catch (Exception localException)
    {
    }
    return bool1;
  }

  public void onUserInteraction()
  {
    super.onUserInteraction();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onUserInteraction", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  protected void onUserLeaveHint()
  {
    super.onUserLeaveHint();
    try
    {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onUserLeaveHint", new
Class[0]), new Object[0]);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onWindowAttributesChanged(WindowManager.LayoutParams
paramLayoutParams)
  {
    super.onWindowAttributesChanged(paramLayoutParams);
    try
    {

InvokeMethod(sAndroidActivityWrapperClass.getMethod("onWindowAttributesChanged
", new Class[] { WindowManager.LayoutParams.class }), new Object[] {
paramLayoutParams });
```

```
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void onWindowFocusChanged(boolean paramBoolean)
  {
    super.onWindowFocusChanged(paramBoolean);
    try
    {
      Class localClass = sAndroidActivityWrapperClass;
      Class[] arrayOfClass = new Class[1];
      arrayOfClass[0] = Boolean.TYPE;
      Method localMethod = localClass.getMethod("onWindowFocusChanged",
arrayOfClass);
      Object[] arrayOfObject = new Object[1];
      arrayOfObject[0] = Boolean.valueOf(paramBoolean);
      InvokeMethod(localMethod, arrayOfObject);
      return;
    }
    catch (Exception localException)
    {
    }
  }

  public void showDialog()
  {
    AlertDialog.Builder localBuilder = new AlertDialog.Builder(this);
    localBuilder.setTitle(2130968578);
    localBuilder.setMessage(2130968579);
    localBuilder.setPositiveButton(2130968576, new DialogInterface.OnClickListener()
    {
      public void onClick(DialogInterface paramDialogInterface, int paramInt)
      {
        AppEntry.this.launchMarketPlace(AppEntry.RUNTIME_PACKAGE_ID);
        System.exit(0);
      }
    });
    localBuilder.setNegativeButton(2130968577, new DialogInterface.OnClickListener()
    {
      public void onClick(DialogInterface paramDialogInterface, int paramInt)
      {
        System.exit(0);
      }
    });
    localBuilder.setOnCancelListener(new DialogInterface.OnCancelListener()
    {
      public void onCancel(DialogInterface paramDialogInterface)
      {
        System.exit(0);
      }
    });
    localBuilder.show();
  }
}
```

Aplicatia:



Concluzie:

In urma efectuarii acestei lucrari de laborator am capatat experienta in lucrul cu Android Studio. Am realizat un simple joc memo time unde trebuie sa asociezi imaginile intrun timp anume.Aplicatia a fost testata cu succes pe un telefon cu sistemul de operare Android. Modul de utilizare al aplicaţiei Memotime este simpla, prin gesturi pe ecran se pot deschide imagini pentru a fi asociate.