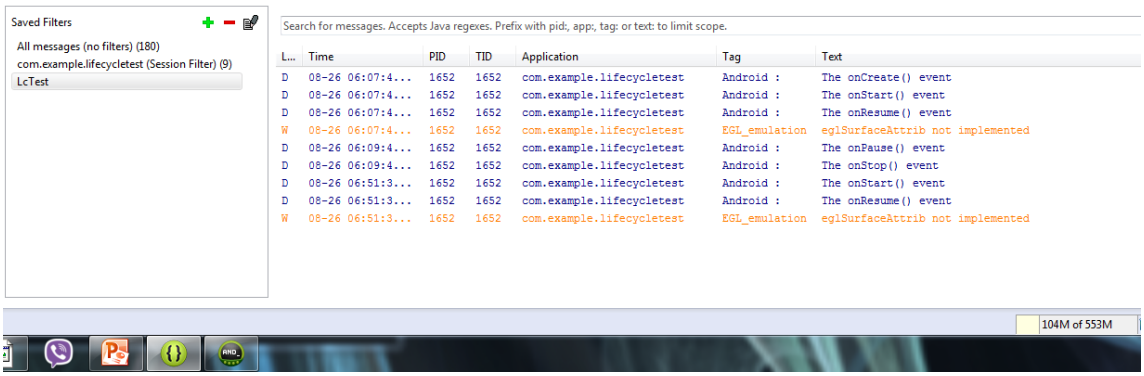
Activity Lifecycle Example:



MainActivity.java

```
package com.example.lifecycletest;

import android.os.Bundle;
import android.app.Activity;
import android.util.Log;

public class MainActivity extends Activity {
   String msg = "Android : ";

   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Log.d(msg, "The onCreate() event");
   }

   /** Called when the activity is about to become visible. */
   @Override
   protected void onStart() {
      super.onStart();
      Log.d(msg, "The onStart() event");
   }

   /** Called when the activity has become visible. */
   @Override
   protected void onResume() {
      super.onResume();
      Log.d(msg, "The onResume() event");
   }

   /** Called when another activity is taking focus. */
   @Override
   protected void onPause() {
      super.onPause();
      Log.d(msg, "The onPause() event");
   }
```

```
    /** Called when the activity is no longer visible. */
    @Override
    protected void onStop() {
        super.onStop();
        Log.d(msg, "The onStop() event");
    }

    /** Called just before the activity is destroyed. */
    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.d(msg, "The onDestroy() event");
    }
}
```

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.lifecycletest.MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/txt" />

</RelativeLayout>
```

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">LifecycleTestExample</string>
    <string name="txt">Lifecycle Test Example</string>
    <string name="action_settings">Settings</string>

</resources>
```

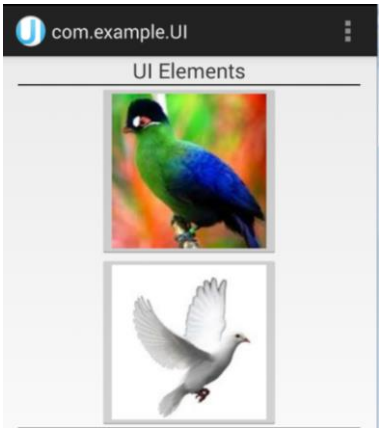
Exercise 1: Activity Lifecycle

- Extend the lifecycle example with a spinner view .
- Use string of arrays in the strings.xml file to populate the spinner.
- Select an item from the **spinner** view and observe the difference in lifecycle events with “home” and “back” buttons of the emulator or device. (Sometimes the logcat may not show the logs if the your computer is slow or you have opened many applications at the same time.)

Exercise 2: UI Development

A:

- Create an android Application with two image views with corresponding images.
- Create a TextView to assign title a the top
- Create Button to Swap the image of two image views
- Use action listener for button “click” event
- Note how drawable is used to retrieve images

B:

- Create two buttons, add images to those buttons.
- Swap the images of on their click events