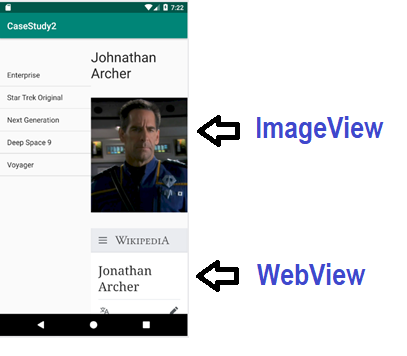
**Week 1**

**Case Study   
“Using Fragments with images and WebView”**

**Step 1:** Setup main layout with two fragments and add click action to the list selection.

Start with a new Android project (default) using Android Studio. Make sure to have working device or emulator ready for your project. To natively support fragments in this case study, you may use a smartphone **(1)** or a tablet **(2)**:

****

**(1)**

****

**(2)**

1. Start with “*Hello, world*” basic application. First of all, make sure to refactor-> rename the name of the startup activity to **Main.kt** (extended from ***AppCompatActivity*** as shown below:

***File : Main.kt – for the Main activity:***

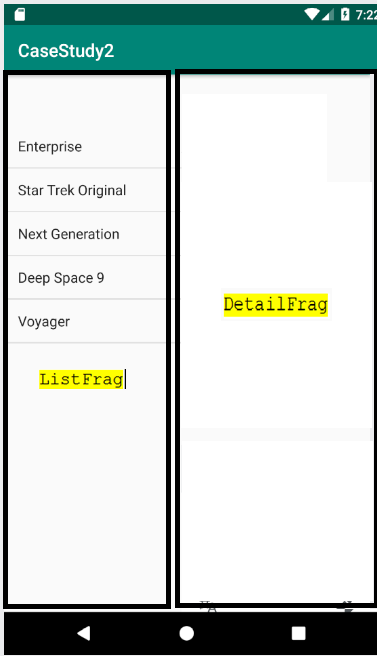
**import** androidx.appcompat.app.AppCompatActivity  
**import** android.os.Bundle  
  
**class** Main : AppCompatActivity() {  
 **override fun** onCreate(savedInstanceState: Bundle?) {  
 **super**.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 }  
}

**Note:** you may need to change main activity name in AndroidManifest.xml to avoid future run-time errors!

For this project, we will use the following layout:

***activity\_main.xml*** - to represent our main container which holds the following two fragments:

1. On the left side - **a standard list layout** to define the list of choices from ListFrag.kt;
2. On the right side - **detail\_fragment.xml** to define a detailed fragment, according to the selection.

v 

***File : activity\_main.xml – for the Main activity:***

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="horizontal"**>  
  
 <**fragment  
 android:id="@+id/frag\_series"  
 class="abc.sheridancollege.casestudy2.ListFrag"  
 android:layout\_width="200dip"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="?android:attr/actionBarSize"** />  
  
 <**fragment  
 android:id="@+id/frag\_capt"  
 class="abc.sheridancollege.casestudy2.DetailFrag"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
</**LinearLayout**>

**Note:**It’s ok to have two errors with missing fragment classes at this point.

***Classes ListFrag.kt*** and ***DetailFrag.kt*** will be created below.

1. The list fragment (**ListFrag.kt**), we will define in code by extending the handy **ListFragment** class:

***File : ListFrag.kt***

**import** android.os.Bundle  
**import** android.widget.ArrayAdapter  
**import** androidx.fragment.app.ListFragment  
  
**class** ListFrag : ListFragment() {  
  
 **override fun** onActivityCreated(savedInstanceState: Bundle?) {  
 **super**.onActivityCreated(savedInstanceState)  
 **val** values = *arrayOf*(**"Enterprise"**, **"Star Trek Original"**, **"Next Generation"**, **"Deep Space 9"**, **"Voyager"**)  
 **val** adapter = ArrayAdapter(*activity*!!, R.layout.*support\_simple\_spinner\_dropdown\_item*, values)  
 *listAdapter* = adapter  
  
 } // method closed  
} // class closed

As you can see above, we initialised array of choices from Start Trek and connected it to list adapter. We will complete a full functionality of this class later, on Step 5 below.

1. Next, add the following layout to your /res/layout folder:

***File: detail\_fragment.xml – layout for DetailFrag.kt   
  
(Right click on layout folder -> New -> XML -> Layout XML File:***

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

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:orientation=*"vertical"* >

<TextView

android:id=*"@+id/captain"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"match\_parent"*

android:layout\_gravity=*"center\_horizontal|center\_vertical"*

android:layout\_marginTop=*"20dip"*

android:text=*"Johnathan Archer"*

android:textAppearance=*"?android:attr/textAppearanceLarge"*

android:textSize=*"30dip"* />

</LinearLayout>

As you can see above, layout has only one textView field to display a name.

4. Move back to a source folder to create ***DetailFrag*** class. Its implementation is nearly as simple; it inflates a view and exposes a setText() method.

***File: DetailFrag.kt***

**// you have to import all classes**

**import** android.os.Bundle  
**import** android.view.LayoutInflater  
**import** android.view.View  
**import** android.view.ViewGroup  
**import** android.widget.TextView  
**import** androidx.fragment.app.Fragment

**class** DetailFrag : Fragment() {  
  
 **override fun** onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {  
 **val** view = inflater.inflate(R.layout.*detail\_fragment*, container, **false**)  
 **return** view  
 }

*// this method update the captain text field***fun** setText(item: String) {  
 captain.text = item // *UPDATE text for the captain name in detail\_fragment.xml*  
}

}

5. The last update for an implementation will be for the ListFrag.java file.   
  
As it was mentioned above, the list fragment we will define in code by extending the ListFragment base class. This class that displays a list of items by binding to a data source such as an array or Cursor, and exposes event handlers when the user selects an item.

So, we create a list view, implement the “on click” event, and make use of the fragment manager to reference one fragment from another according to the click.

In the onListItemClick() method, notice the check for whether the target fragment exists in this particular layout — that allows a developer to implement a separate layout for smaller screen devices like phones.

***File: ListFrag.kt – upfdate, to display a list of items***

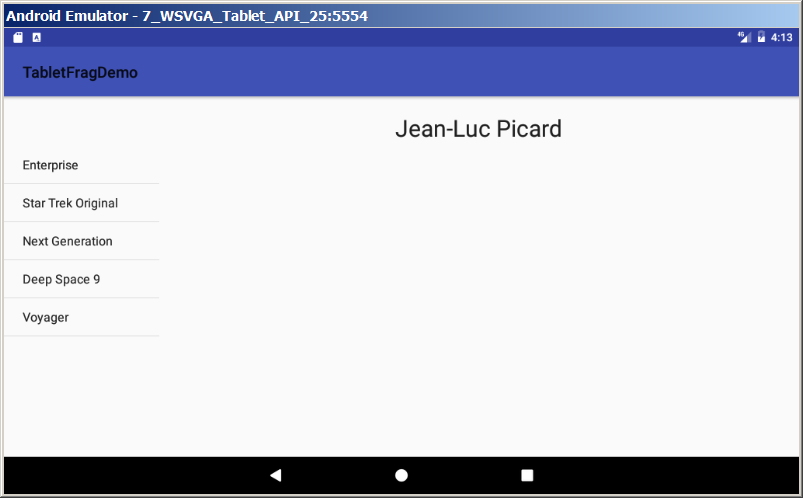
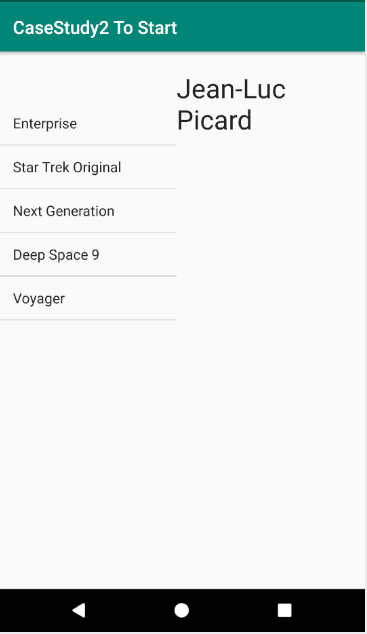
// add 2 more methods to this class you have already created above on step 2

**class** ListFrag : ListFragment() {  
  
 **override fun *onActivityCreated***(savedInstanceState: Bundle?) {  
 **super**.onActivityCreated(savedInstanceState)  
 **val** values = *arrayOf*(**"Enterprise"**, **"Star Trek Original"**, **"Next Generation"**, **"Deep Space 9"**, **"Voyager"**)  
 **val** adapter = ArrayAdapter(*activity*!!, android.R.layout.*simple\_list\_item\_1*, values)  
 *listAdapter* = adapter  
 }  
  
 **private fun** getCapt(ship: String): String {  
 **if** (ship.*toLowerCase*().*contains*(**"enterprise"**)) {  
 **return "Johnathan Archer"** }  
 **if** (ship.*toLowerCase*().*contains*(**"star trek"**)) {  
 **return "James T. Kirk"** }  
 **if** (ship.*toLowerCase*().*contains*(**"next generation"**)) {  
 **return "Jean-Luc Picard"** }  
 **if** (ship.*toLowerCase*().*contains*(**"deep space 9"**)) {  
 **return "Benjamin Sisko"** }  
 **return if** (ship.*toLowerCase*().*contains*(**"voyager"**)) {  
 **"Kathryn Janeway"** } **else "???"** }

**override fun *onListItemClick***(l: ListView, v: View, position: Int, id: Long) {  
 **val** item = *listAdapter*!!.getItem(position) **as** String  
 **val** frag = *fragmentManager*!!.findFragmentById(R.id.*frag\_capt*) **as** DetailFrag?  
 **if** (frag != **null** && frag.*isInLayout*) {  
 frag.setText(getCapt(item))  
 }  
 }

}

6. Finally, we are ready to execute the demo! If you own a tablet, device, go ahead and connect your device. If not, you can easily emulate a tablet or device using the **Android Virtual Device Manager**:

**Step 2: Individual work – adding Images**

**After completing Step1 above, follow with the individual work instructions below.**

You probably notice a lot of free space available on the right screen panel, so we may upgrade it to add some more important information. For this purpose, you have to collect some actor’s images from “Star Trek” – a famous American science fiction entertainment franchise.

When you click on the list item on the left, a right panel will be populated with not only appropriate text, but also with the specific image, as shown on a sample below:

****

**Step 3: Project Upgrade – adding Web content (WebView field)**

And one simple upgrade left to implement! When you click on the list item, the right panel will be populated with not only appropriate text and image, but also with the appropriate web-content of a web-page, as shown below (“Deep Space 9” activates image of Ben, and open web-page about Benjamin Sisko). Basic use of WebView is presented by link for Android developers:

<https://developer.android.com/guide/webapps/webview.html>

****

**Note1:** before make it works, the application must request [INTERNET](https://developer.android.com/reference/android/Manifest.permission.html#INTERNET) permission in AndroidManifest file.   
**Note2:** If you run the project and get a run-time error: “*XML error:* *Error inflating class android.webkit.WebView* “ , just add those new dependencies to build.gradle(Module: app) file:

*implementation 'androidx.appcompat:appcompat:1.2.0-beta01'  
implementation 'androidx.appcompat:appcompat-resources:1.2.0-beta01'*

P.S.: If you are not a big fan of “Star Trek”, feel free to replace it with any other specific theme.

**Prepare your Project Demo for the final submission on-line:**

1. Add your ***Full name*** and ***student ID*** to the app title;
2. Add your programming comments with your ***Full Name*** and ***Student ID*** to the top area of each of three Kotlin files;
3. Export your project to zip. Select ***File*** -> ***Export To Zip File*** -> ***YournameProjectName.zip***, and submit zip – project on-line;
4. Add a document (DOC or PDF) with your ***personal reflections*** and screen shotS to demo THE BEST full functionality;
5. Be ready to demo it to your instructor during the next class.