CMPS 312 Mobile Application Development
[**Imbedded Lab**]

✔ Copies of Assignments

HOMEWORK-1

1. Install Android Studio on your Laptops by following the steps in "PART A"
2. Create an application and name it "My First App-<XX> Replace "XX" with the initials of your first name and last name" Eg. "My First App-AH"
3. Domain Name: cmps312.qu.edu.qa
4. Min SDK 4.2
5. Select Empty Activity
6. Change the Activity Name to "MyMainActivity" then Finish
7. Go to Your "MyMainActivity" and add the Six System Log Messages that are shown in PART C.

   Example: Log."**X**"(**TAG**, "**MESSAGE**"); [Replace the **X** by the Log type {v,w,I,wtf,e or d} , **TAG** = "MyMainActivity" and **MESSAGE** ="The type of the Log message" eg. In log.e the message will be **"this is an error message"**

8. Create Two Virtual Devices
   a. **Phone**: Nexus 6P API 23
   b. **Tablet**: Nexus 9 API 23

***NOTE**: If the device Image is not downloaded, first download the emulator Image then create the virtual device.*

9. Install Your Application on both devices (Phone and Tablet)
10. Install the Application on your physical device (**Optional**)
11. Take a Screenshot of your application on both devices

HOMEWORK-2

The Homework2 app stores the number of times each lifecycle stage of an Android Activity is triggered and displays the counts in the current Activity as in Fig 1. As your app gets in the background of another activity, or destroyed, these numbers should reflect the times the lifecycle stages are triggered.

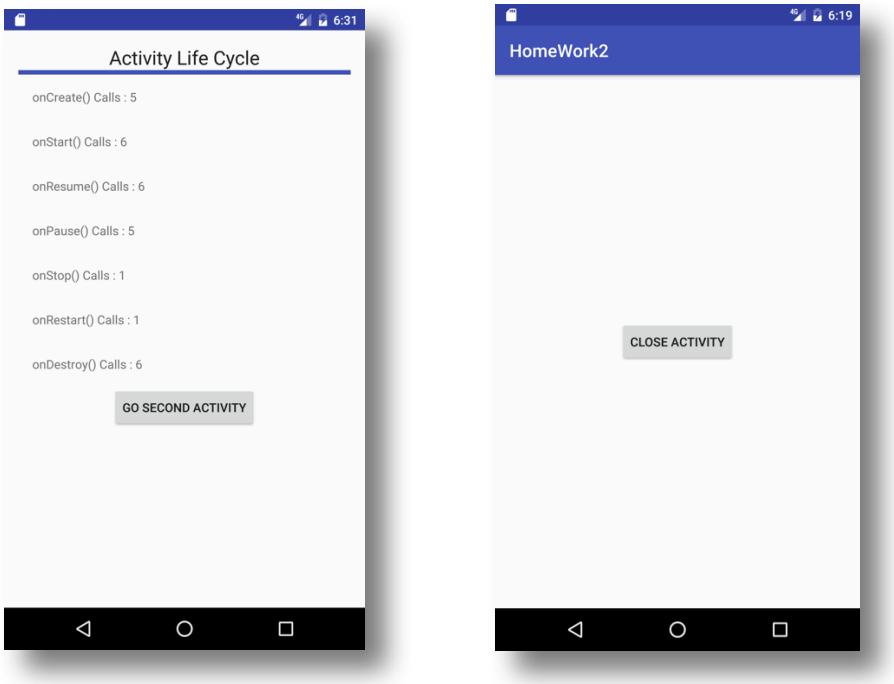


Figure 1 : Main Activity Screen

Important note: When you rotate your device (CTRL + F12 in windows or CTRL + FN + F12 in Mac in the emulator), android destroys and recreates the activity by default in Android. It is your job to make sure that these counters do not get reset when a screen rotation happens and instead, the counts should increment as if they were never reset!

You will need to use the **onSaveInstanceState()** and **onRestoreInstanceState()** (or use the bundle parameter in **onCreate())** to save and restore the counter values when a screen rotation happens, or when Android decides to destroy your activity when it is in the background due to memory constraints.

HOMEWORK 3

The home work has two parts. A part that focus on intent filters and second part which focus on the implicit intents and getting results from activities. Screenshots of the sample applications are shown below. However, it does not mean you should exactly do the same App. If you want to be more creative and add more features, please do so. But the minimal requirement is to achieve the listed features. Please make sure to name your applications as follows **YOUR_NAME**. homework2..cmps312.**APP_NAME.**

Example

1. abdulahihassen. homework2.cmps312.cameraapp
2. abdulahihassen. homework2.cmps312.emailhandlingapp

Part 1: Intent Filters

1. You need to implement an Email handling application using Intent filters that handles the implicit intents with SENDTO action.
2. To test your Email handling APP, you can use the Lab application. Once you press on the email Icon in the "solar" app , then your application should show as one of the email handling application as shown in the figure 1.
3. If the user selects your application, you should display back the content of the email sent through the implicit intent as shown in figure 2.

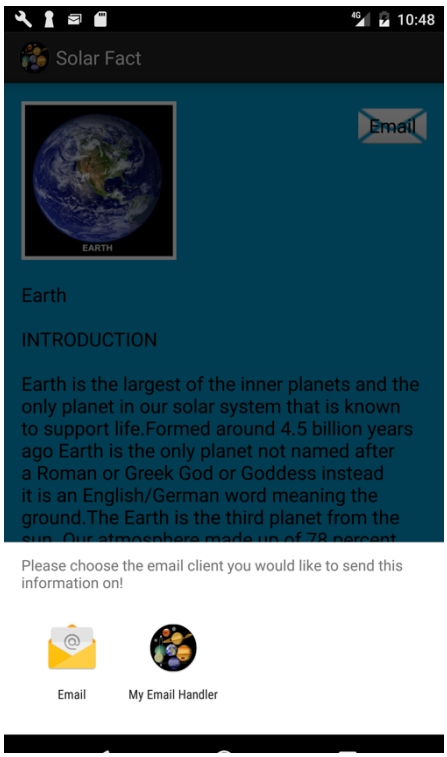


Figure 3. Listening to the SENT_TO ACTION

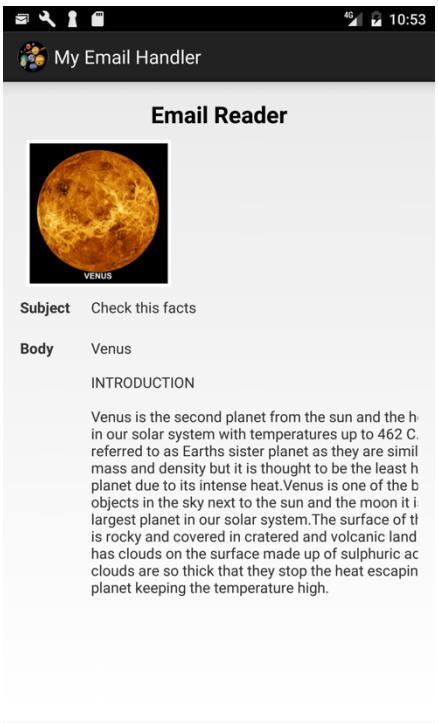


Figure 2. Displaying back the email content

Part 2: Implicit Intents and getting a result from an Activity

1. You need to create a photo taking application.

2. In the app when the user presses on the take photo button in figure 3 and 4 then you should launch the camera as shown in figure 3 and display the user picture back on your activity as shown in figure 4.
3. You have to have all the runtime permissions handled correctly

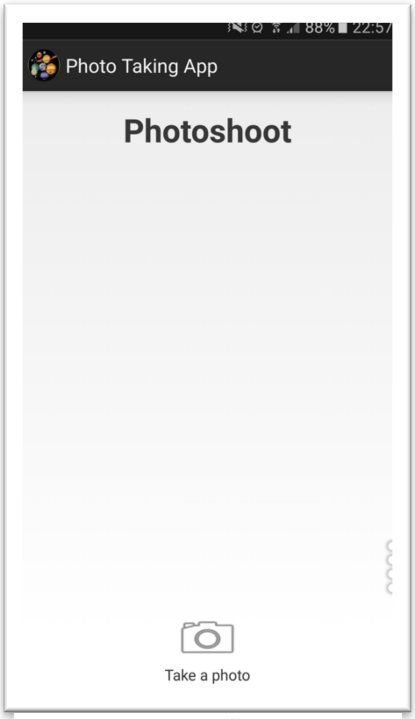


Figure 6. Camera app home screen

Figure 4. THE PHONE CAMERA APP

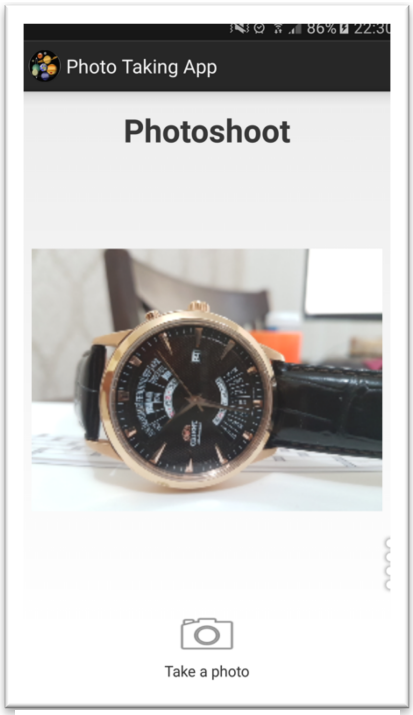


Figure 5. After taking a picture. Your application displaying it back

What you should submit

1. Screenshots of both applications.
2. The code for both the applications. Make sure you save them with the same package structure that is requested above [**YOUR_NAME**. homework2.cmps312.**APP_NAME**].

Note: After submission you will be asked to demo your app and explain the code.

HOMEWORK 4

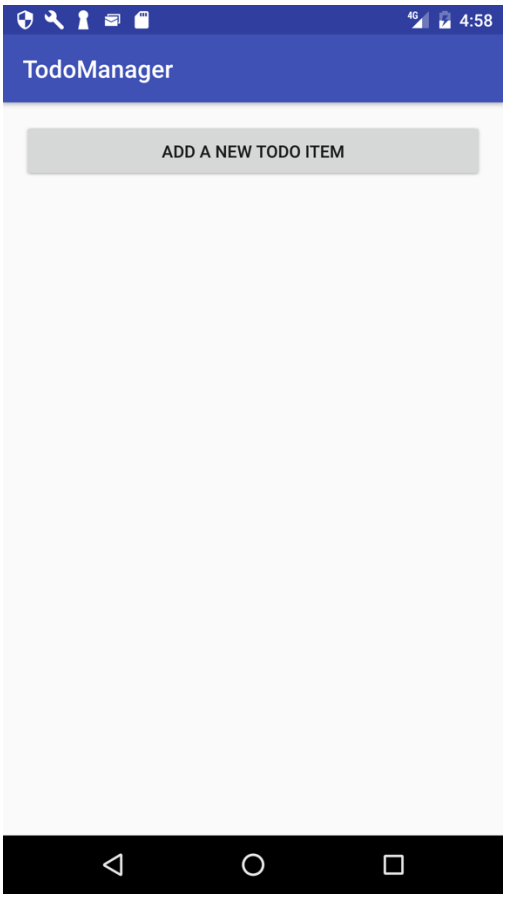
In this homework, you will create a ToDo manager application. The application creates and manages Todo List Items. You will design this application's UI, including its layout and resource files. You will also need to implement the application's features by yourself.

PART A: A BASIC TODO MANAGER APPLICATION

The first time the application runs it will have no ToDo Items and therefore its initial UI will look something like this:

This UI contains a ListView for displaying existing Todo Items. As ToDo Items are created, they will be added to this ListView.

Initially, the application can either start with empty list or you can create some dummy ToDo Items. As you work through the Homework, you'll make ToDo Items persist across sessions.



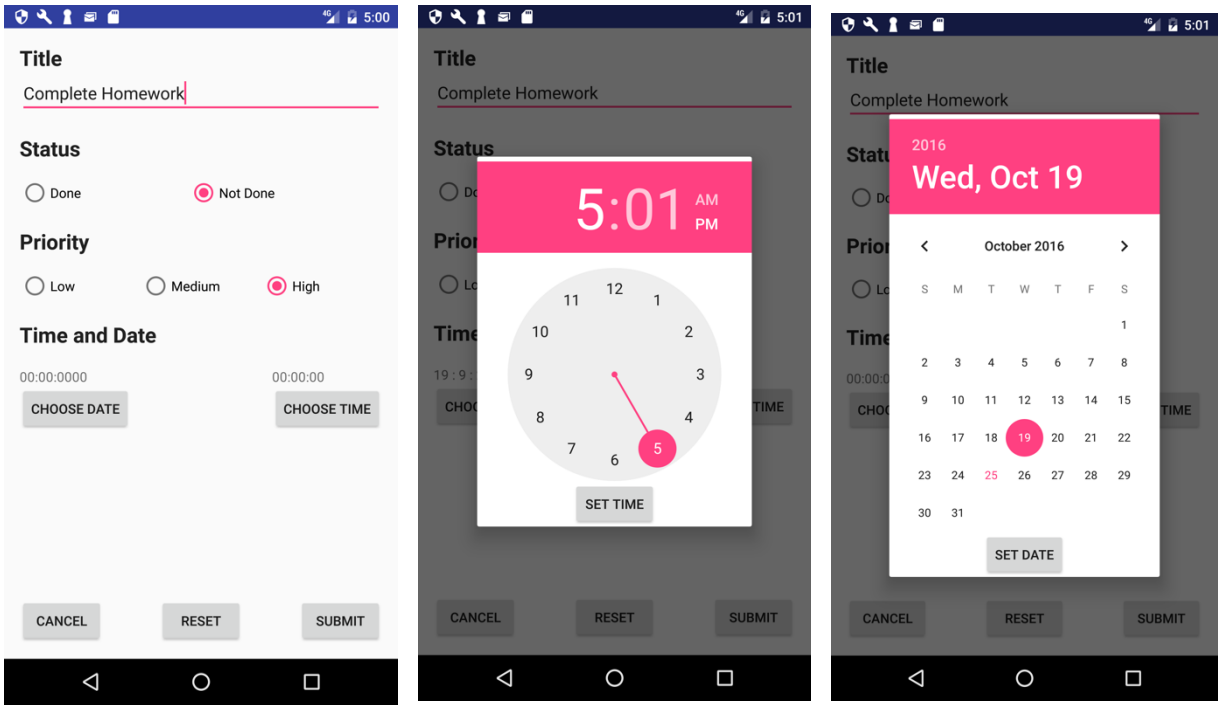
The ListView, shown above, always displays a special item, labeled "**Add New ToDo Item**" in its last position. This position is the called the "**footer**." When the user clicks on the **ListView**

footer, a new Activity will be opened that allows the user to create a **new ToDo Item**.

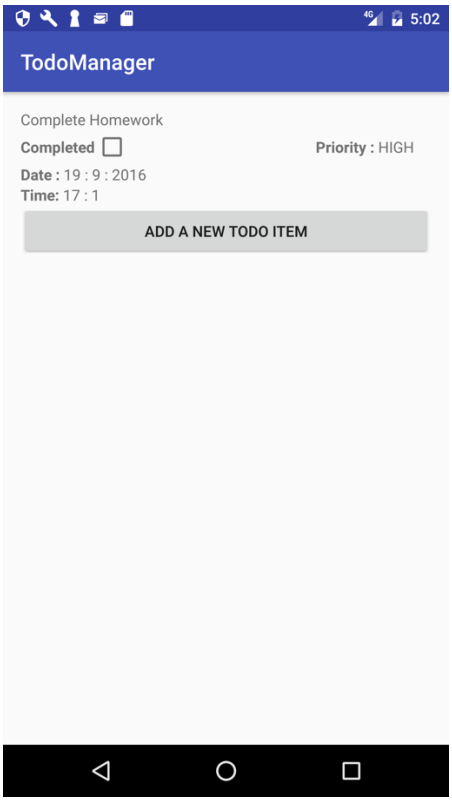
You are free to define your own user interface for this new Activity, but it must include the fields shown in the figure below.

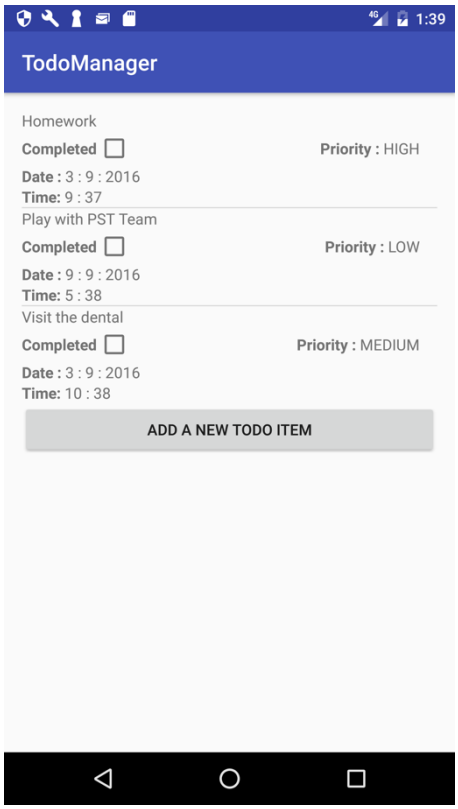
Specifically, **ToDo items** have the following fields:

1. Title: A user-provided String
2. Status: {Done, Not Done}
3. Priority: {Low,Med,High}
4. Time & Date: A deadline for completing the underlying ToDo Item. The "Add New ToDo Item" Activity's UI should also includes several buttons:
5. Cancel – finish the Activity without creating a new ToDo Item.
6. Reset –Resets the fields of the Todo Item to their default values.



• Submit – Create a new Todo Item with the user selected data fields & return to main Activity. When the application returns to the main Activity, the new ToDo Item should appear in the main Activity's Listview. For example, if the user adds a new ToDo Item to an empty ToDo list, as shown below, then the main Activity's ListView should update to display the new ToDo Item.



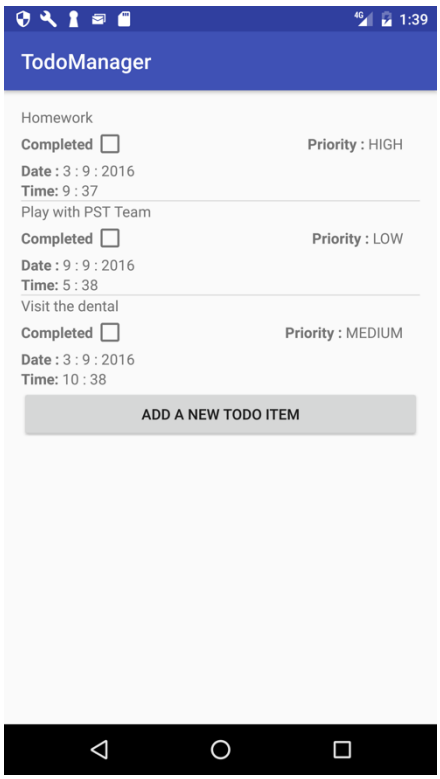


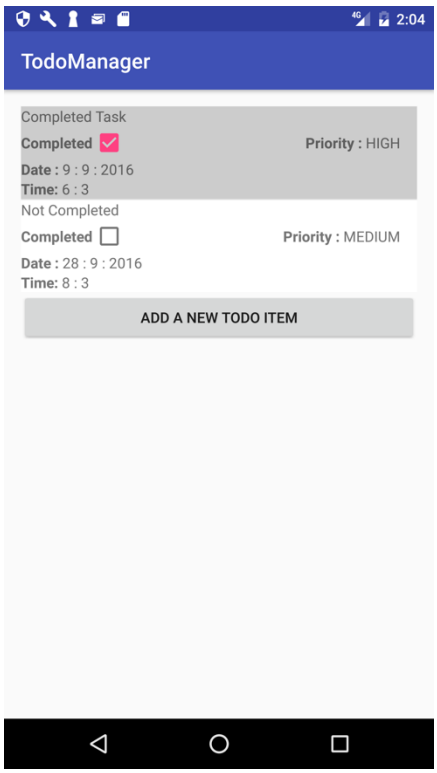
Back in the Main Activity, the user should be able to toggle the Done checkbox to indicate that the ToDo Item is Done or Not Done.

To make Todo Items persist across sessions you can use an Array/ List, you can write Todo items to a file and read existing Todo items from the file at application startup. Since you have not learned about file I/O, you are required to do this for the moment. However, if you are curious on how to implement this you can find more information about FileInputStream and FileOutputStream can be found from here: http://developer.android.com/reference/android/content/Context.html. You can also persist using SharedPreferences or SQL database.

Part B: An improved ToDo manager application

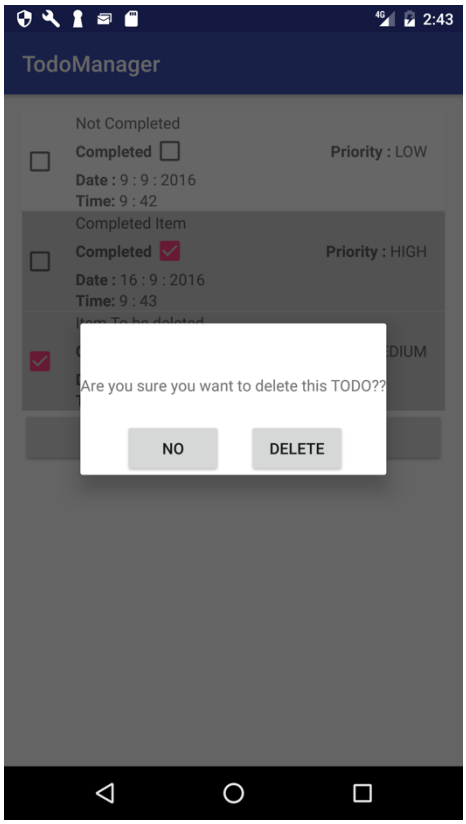
In this part, modify your application so that **ToDo Items** that are Not Done are displayed in the Main Activity with a different colored background than those that are Done. In addition, when the user toggles the Done checkbox, the background color should change as appropriate.

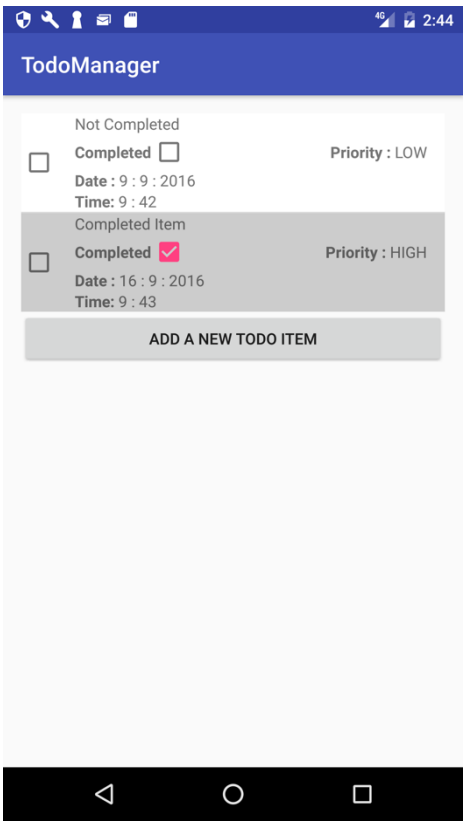




Part C: A ToDo manager application you might actually use

In this part, modify your application so that if the user selects a ToDo Item in the Main Activity's ListView, a dialog pops up, allowing the user to delete the selected ToDo Item.



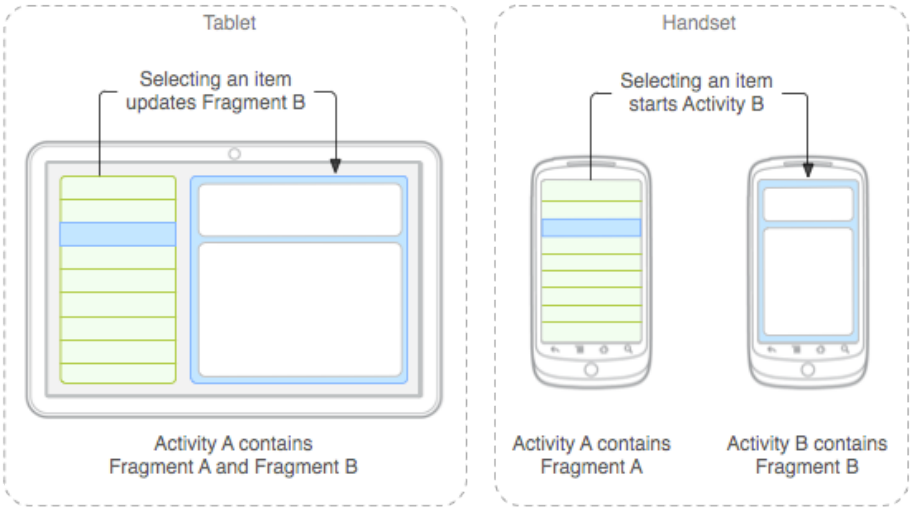


Deliverables:

Submit your project via the submit BB. Your submission should include: a. A zip file containing the source code Android Studio project.

HOMEWORK 5

In this homework, you are given the freedom to create an application of your choice which applies the fragment concepts that we took in the Lab. So be creative ☺ . If you need some app ideas, see the list given in page two.



Important: The application that you develop should meet the following requirements.

REQUIREMENTS

1. You should have a minimum of **four fragments or more**
2. At least one of your fragment should be a **list fragment**
3. You should use both **dynamic and static fragments** inside your application
4. You should also use the following transaction methods (**replace/add**) fragments in order to load the different UI layouts in your application.
5. Your application needs to have a **communication between fragments** with the use of "**interfaces**" (at least once).
6. In tablet, your application should render at **least two or more fragments** side by side to take advantage of the extra space as shown in the figure above.

Deliverables:

Submit your project via the submit BB. Your submission should include: a. A zip file containing the source code Android Studio project.

Some app ideas for the homework

1. QU computer science website
2. Restaurant app
3. Quiz app
4. Car rental
5. Weeding planner
6. Hotel reservation
7. Fashion

NOTE: Beside the above you can use your own app idea.