# MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 5

OBJECTIVES

1. Understanding text to speech
2. Understanding Fragments
3. Practice Activities

Objective 1 – Understanding Text to Speech

- Synthesizes speech from text for immediate playback or to create a sound file
- Android text to voice was introduced when Android 4.2.2 Jelly Bean

```
ttobj=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
   @Override
   public void onInit(int status) {
   }
});
```

- In this listener, you have to specify the properties for `TextToSpeech` object, such as its language, pitch e.t.c. Language can be set by calling `setLanguage`() method
- Once you have set the language, you can call speak method of the class to speak the text

```
ttobj.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null);
```

Objective 2 – Understanding Fragments

- A Fragment represents a behavior or a portion of user interface in a `FragmentActivity`.
- A fragment must always be hosted in an activity and the fragment's lifecycle is directly affected by the host activity's lifecycle
- When you add a fragment as a part of your activity layout, it lives in a `ViewGroup` inside the activity's view hierarchy and the fragment defines its own view layout

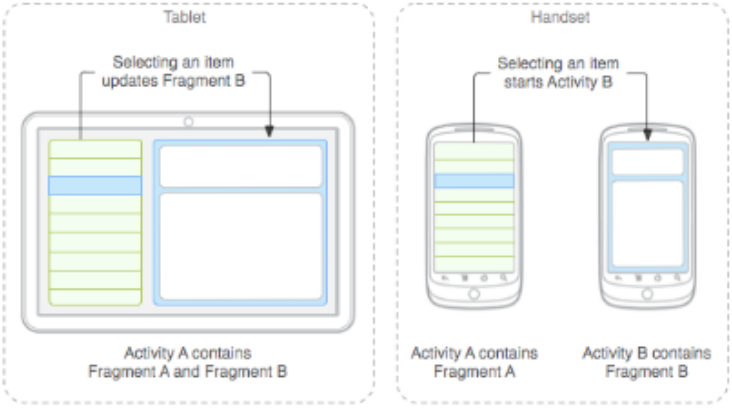


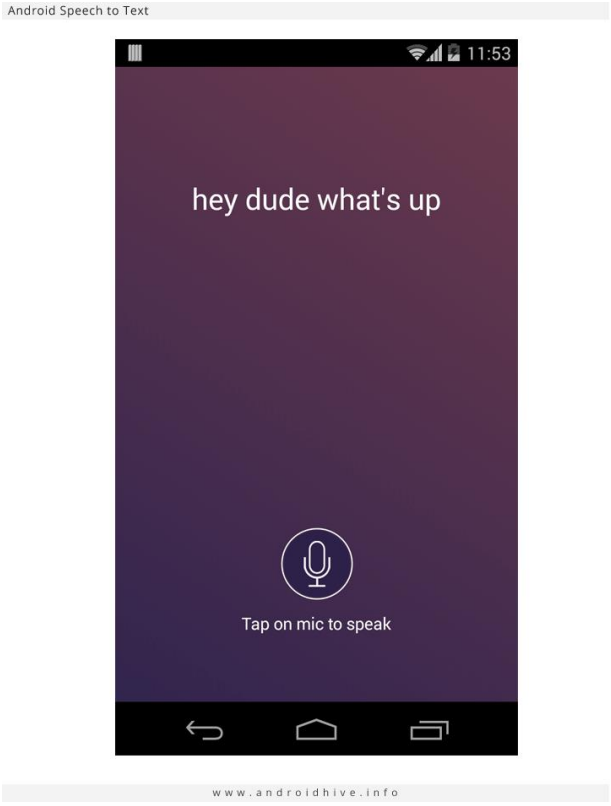

**Figure 1.** An example of how two UI modules defined by fragments can be combined into one activity for a tablet design, but separated for a handset design.

# MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 5

Activities

Activity 1: Command Listener

- Make an application which listens to user's command and performs an action. For example if user asks it to open camera, it opens the camera. (use it to open any one application)




To launch any app using package name

```
Intent launchIntent = getPackageManager().getLaunchIntentForPackage("com.example.abc");
startActivity( launchIntent );
```

If you don't know the package name

```
PackageManager pm;
pm = getPackageManager();
//  get a list of installed apps.
packages = pm.getInstalledApplications(0);
```

# MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 5

Activity 2 : Improve the Application HangMan from Lab 3 (Lab 3 – Task 3)

- Add two more functionalities, one it will speak the letter typed by user in the textbox
- Second it will speak the remaining lives after every letter user pressed in.
- Finally, user can also put text by speaking (add one button near the text box for user to speak)

Activity 3 : ToDo List

Write a simple to-do list app that has a `ListView` of tasks that the user needs to complete. Initially the app is empty and has nothing in the to-do list. But if the user types text into a bottom `EditText` and clicks an Add button, the new item will be added to the top or bottom of the list.

It's also good to have a way to **remove items** from the list. You could achieve this by attaching a listener to the list that removes an item when that item is clicked on by the user. Or if you want to try something slightly different, try making it remove an item when the user performs a "long click" (pressing and holding the mouse on an item). You can do this by calling the `setOnItemLongClickListener` method of your list and passing an anonymous `AdapterView.OnItemLongClickListener` class. Android Studio can help you auto-generate the skeletons of these anonymous listener classes if you press Ctrl-Space in the editor at the right place in the code.

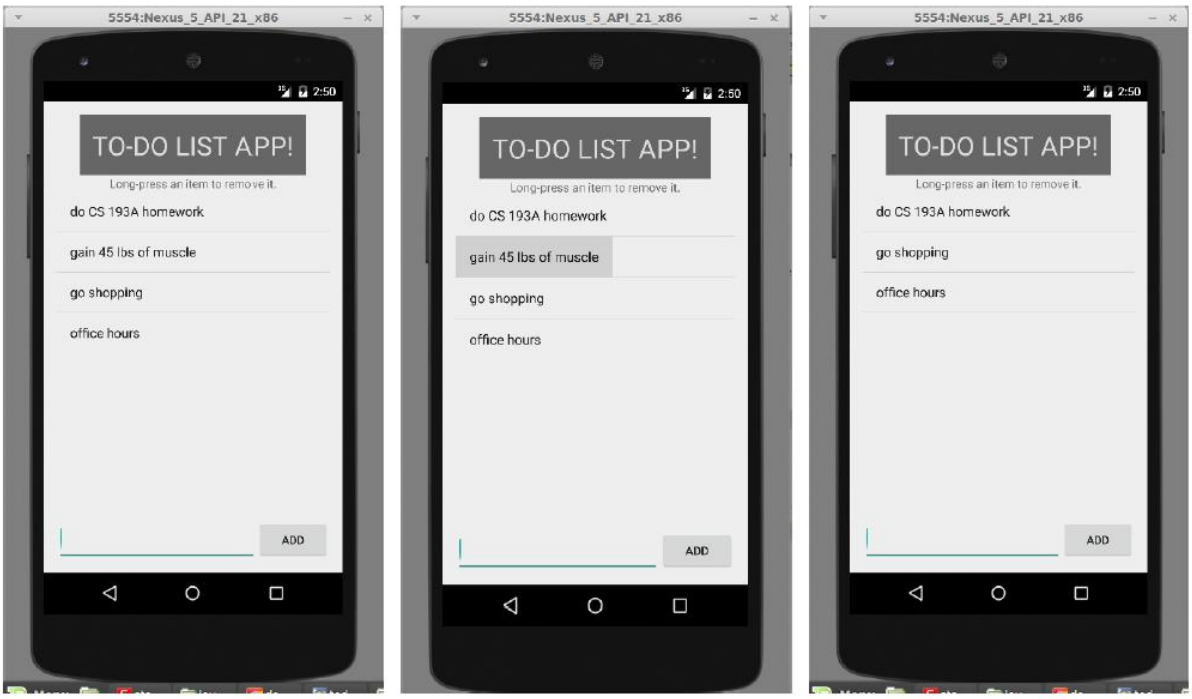



*Figure: User long-clicking on second list item to delete it*

If the items in your to-do list are stored into an `ArrayList`, by default the app's GUI won't notice when you add or remove an item from the list. That is, you'll modify the ArrayList state but the graphical list on the screen won't update to match. To fix this, you have to call the method `notifyDataSetChanged()` on your `ArrayAdapter` to tell it that the underlying array list has changed. To be able to do this, of course, you'll have to save your `ArrayList` and your `ArrayAdapter` as private fields inside your activity.