

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### OBJECTIVES

- To understand Speech to Text
- Understanding Fragments
- Understanding Adapters
- Understanding ListView
- Practice Activities

### OBJECTIVE 1: Understanding Speech to Text

- Android's feature to translate Speech into Text
- Supported since Android 2.2 Android Froyo
- The most important intent is `RecognizerIntent.ACTION_RECOGNIZE_SPEECH` with only one required extra data source, `RecognizerIntent.EXTRA_LANGUAGE_MODEL`.
- To change the language, if we don't want to use default language we can add one more extra data with the intent `RecognizerIntent.EXTRA_LANGUAGE`.
- Before starting recognizer Intent it's a good habit to check whether the voice to text API is supported by your smartphone.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    if (!SpeechRecognizer.isRecognitionAvailable(context: this)) {
        updateResults(s: "\nNo voice recognition support on your device!");
    } else {
        // Do your work for listening speech
    }
}

void updateResults(String s) {
    mTextView = (TextView) findViewById(R.id.textView);
    mTextView.setText(s);
}
```

- To start recognizing there are a few things necessary to do.
- Create an Intent object and assign a `RecognizerIntent` to it.
- Then add a few extra input parameters to it as discussed above

### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

```
private void startSpeechRecognizer() {
    Intent intent = new Intent
        (RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "Hi! How can I help you?");
    startActivityForResult(intent, requestCode: 1234);
}

void updateResults(String s) {
    mTextView = (TextView)findViewById(R.id.textView);
    mTextView.setText(s);
}
```

- To fetch the output, we will have to override one of the built-in methods of the ActivityClass (onActivityResult).

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
    Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

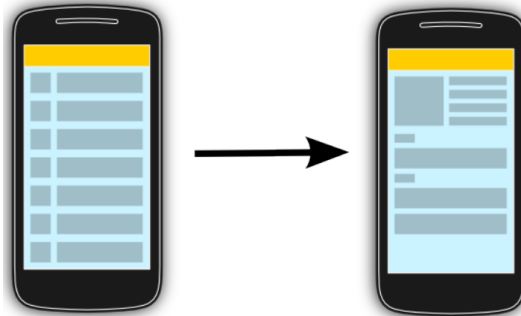
    if (requestCode == 1234) {
        if (resultCode == RESULT_OK) {
            List<String> results = data.getStringArrayListExtra
                (RecognizerIntent.EXTRA_RESULTS);
            String mAnswer = results.get(0);
        }
    }
}
```

### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

### OBJECTIVE 2 – Understanding Fragments

- A *fragment* is an independent Android component which can be used by an activity. A fragment encapsulates functionality so that it is easier to reuse within activities and layouts.
- A *fragment* runs in the context of an activity, but has its own life cycle and typically its own user interface. It is also possible to define fragments without an user interface, i.e., headless fragments.
- The issue of different sizes of Android Devices can be solved using fragments and making different layouts for different Android Device.
- Fragments help us in creating multi-pane layouts.
  - o A *panel* or *pane* represents a part of the user interface. The term pane is a general term used to describe the concept that multiple views are combined into one compound view depending on the actual available space
- If not enough space is available one panel is shown. This is typically called single-pane layout.



- If space is available multiple panels will be shown.



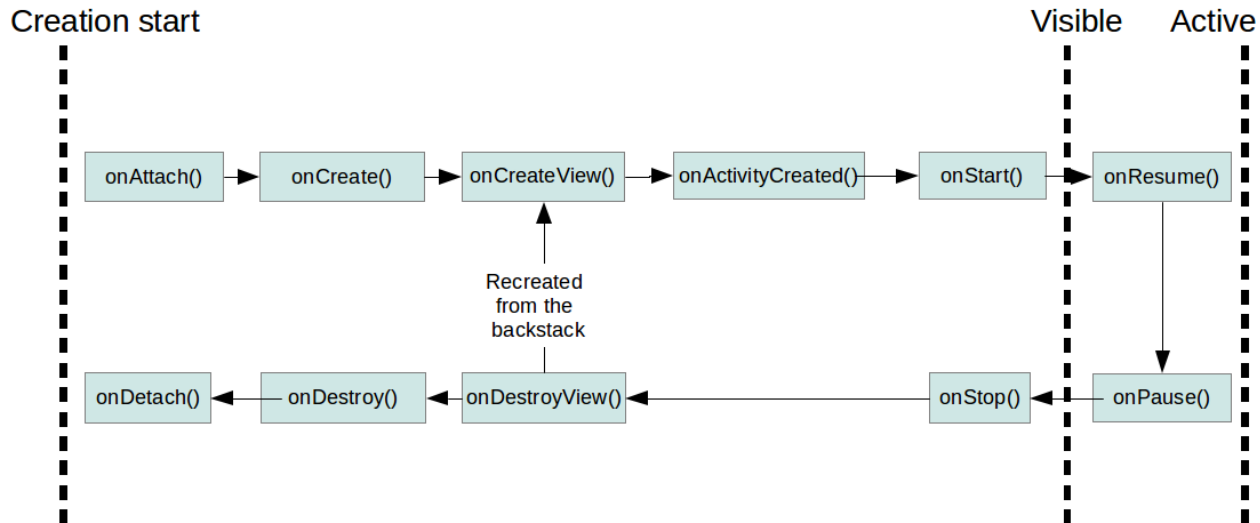
- There are two type of fragments
  - o Static Fragments
    - Static fragments are declared and defined at compile time and won't change even after you run the applications. We won't be replace it with other fragments.
  - o Dynamic
    - Dynamic fragments can be loaded and replaced at runtime on user choice.

#### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### - Fragment Life Cycle



### Importance of Fragments

- **Reusing View and Logic Components** - Fragments enable re-use of parts of your screen including views and event logic over and over in different ways across many disparate activities. For example, using the same list across different data sources within an app.
- **Tablet Support** - Often within apps, the tablet version of an activity has a substantially different layout from the phone version which is different from the TV version. Fragments enable device-specific activities to reuse shared elements while also having differences.
- **Screen Orientation** - Often within apps, the portrait version of an activity has a substantially different layout from the landscape version. Fragments enable both orientations to reuse shared elements while also having differences.

### Communicating with Fragments

- **Bundle** - Activity can construct a fragment and set arguments
- **Methods** - Activity can call methods on a fragment instance
- **Listener** - Fragment can fire listener events on an activity via an interface

In other words, communication should generally follow these principles:

- Activities can initialize fragments with [data during construction](#)
- Activities can pass data to fragments [using methods on the fragment instance](#)
- Fragments can communicate up to their parent activity [using an interface and listeners](#)
- Fragments should pass data to other fragments only routed through their parent activity
- Fragments can pass data to and from [dialog fragments as outlined here](#)
- Fragments can contain [nested child fragments as outlined here](#)

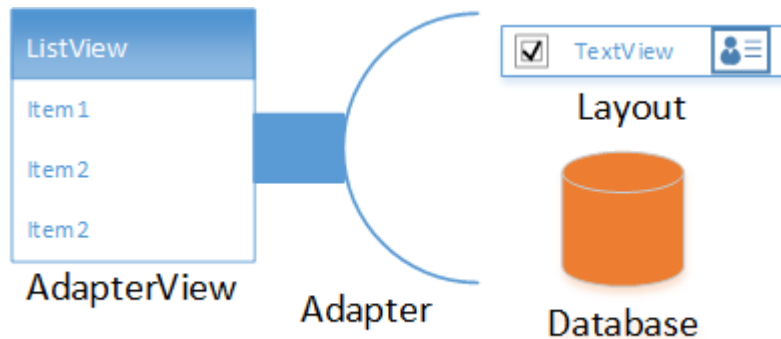
Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

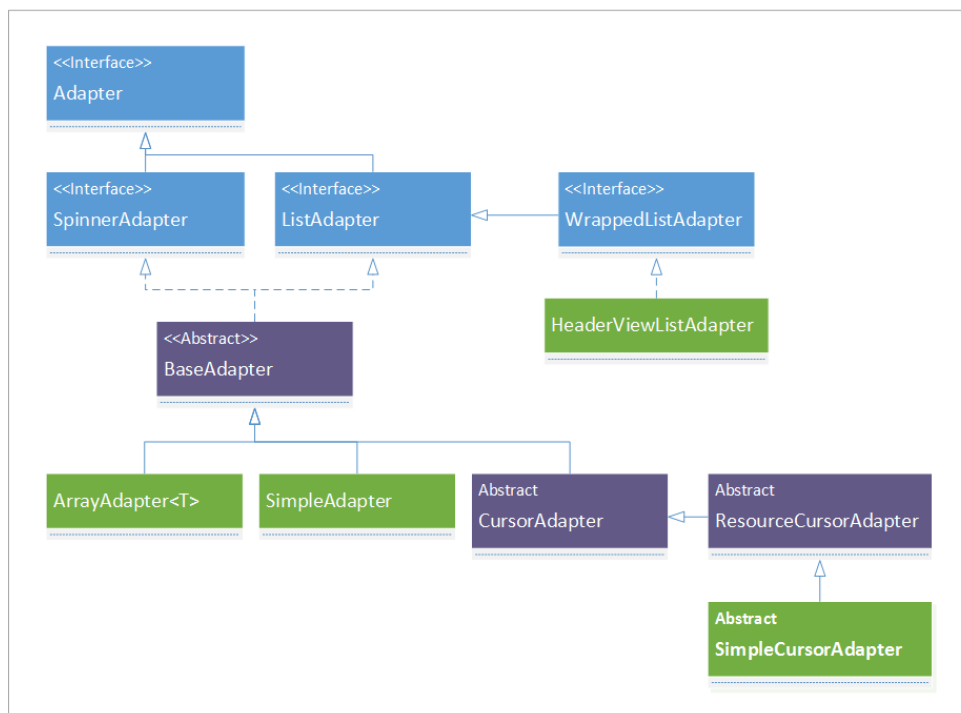
## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### OBJECTIVE 3: Introduction to Adapters.

- A bridge between an AdapterView and the underlying data for that view.
- An AdapterView is a group of widgets (aka view) components in Android that include the ListView, Spinner, and GridView.
- AdapterView also provides the layout of the underlying data for the view



### Types of Adapters



### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### Adapter View Methods

- **getCount( ):** indicates to Android how many items (or rows) are in the data set that will be presented in the AdapterView.
- **getItem(int pos):** get the data item associated with the item (or row) from the AdapterView passed as a parameter to the method. This method will be used by Android to fetch the appropriate data to build the item/row in the AdapterView.
- **getItemId(int pos):** This method returns the data set's id for a item/row position of the AdapterView. Typically, the data set id matches the AdapterView rows so this method just returns the same value.
- **getView(int position, View convertView, ViewGroup parent):** This method creates the View (which may be a single View component like a TextView or a complex set of widgets in a layout) that displays the data for the specified (by position) item/row in the AdapterView

### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### OBJECTIVE 4: Introduction to ListView

- **ListView** is a view group that displays a list of scrollable items. The list items are automatically inserted to the list using an Adapter that pulls content from a source such as an array or database query and converts each item result into a view that's placed into the list.



- To display a more custom view for each item in your dataset, implement a **ListAdapter**. For example, extend **BaseAdapter** and create and configure the view for each data item in **getView(...)**:

```
private class MyAdapter extends BaseAdapter {  
  
    // override other abstract methods here  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup container) {  
        if (convertView == null) {  
            convertView = getLayoutInflater().inflate(R.layout.list_item, container, false);  
        }  
  
        ((TextView) convertView.findViewById(android.R.id.text1))  
            .setText(getItem(position));  
        return convertView;  
    }  
}
```

#### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### OBJECTIVE 5 – PRACTICE ACTIVITIES

#### Activity 1 – Adding fragments statically (Steps)

- Add fragment tags in the layout file.
- Create fragment classes
- Pass fragment tag any unique id along with the name of the class you want to attach the fragment with.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:baselineAligned="false"
    android:orientation="horizontal" >

    <fragment
        android:id="@+id/listFragment"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="match_parent"
        class="com.example.android.rssreader.MyListFragment"
        tools:layout="@layout/fragment_rsslist_overview">
    </fragment>
    <fragment
        android:id="@+id/detailFragment"
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="match_parent"
        class="com.example.android.rssreader.DetailFragment"
        tools:layout="@layout/fragment_rssitem_detail">
    </fragment>

</LinearLayout>
```

View Name	View Value	Event
Two Fragments		
One TextView in each Fragment layout		

#### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)



## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### Activity 2: Dynamic Fragments (Steps)

- Add the containers (FrameLayouts) in the main Activity Layout instead of adding the fragment tags.

```
<?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="horizontal" >

    <FrameLayout
        android:id="@+id/listcontainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <FrameLayout
        android:id="@+id/detailscontainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:visibility="gone" />

</LinearLayout>
```

- Create 5 fragment classes (*BlankFragment*).
- Instead of mentioning in xml file that which fragment we will run, we do that with using fragment manager.
- Create the object of *FragmentManager* and assign it *getFragmentManager()*, if you are using support support manager then assign it *getSupportFragmentManager()*.
- Start *FragmentTransaction* using *FragmentManager* object.

### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

- We can add fragments, replace fragments, and remove fragments using that *FragmentManager* object.

```
// get fragment manager
FragmentManager fm = getFragmentManager();

// add
FragmentManager ft = fm.beginTransaction();
ft.add(R.id.your_placeholder, new YourFragment());
// alternatively add it with a tag
// trx.add(R.id.your_placeholder, new YourFragment(), "detail");
ft.commit();

// replace
FragmentManager ft = fm.beginTransaction();
ft.replace(R.id.your_placeholder, new YourFragment());
ft.commit();

// remove
Fragment fragment = fm.findFragmentById(R.id.your_placeholder);
FragmentManager ft = fm.beginTransaction();
ft.remove(fragment);
ft.commit();
```

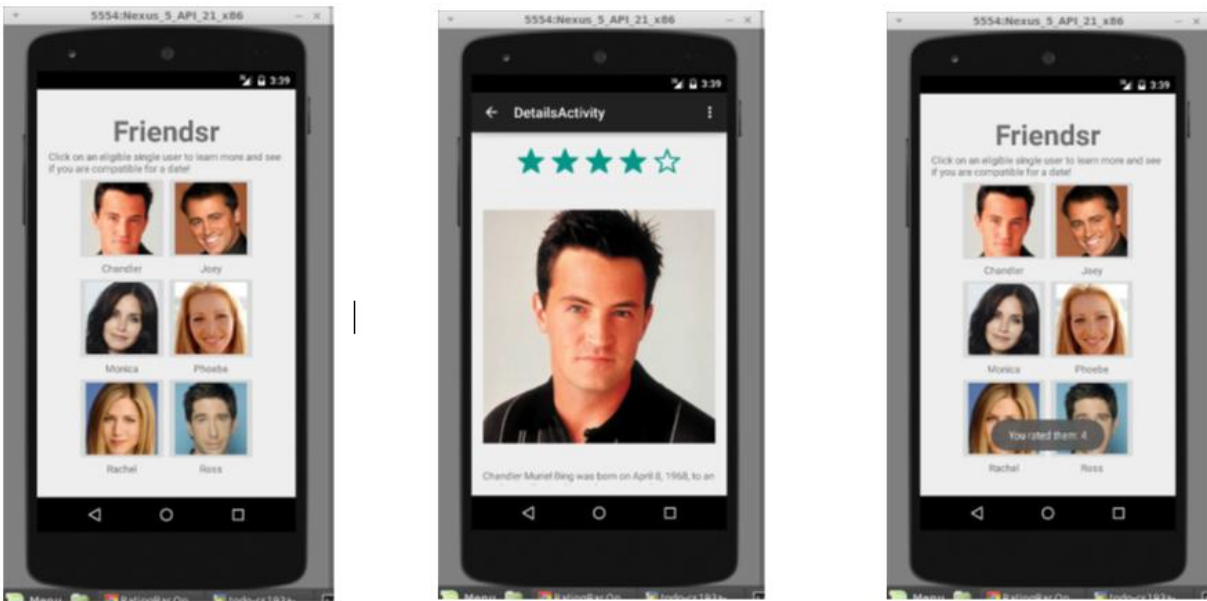
### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### Activity 3: Friendsr App (Activity 2, Lab 4).

- Add fragments instead of activities and update it accordingly.



- Achieve the same thing using fragments, instead of using multiple activities.
- To share information between fragments we can rely on MainActivity and share it with other fragments from there on.

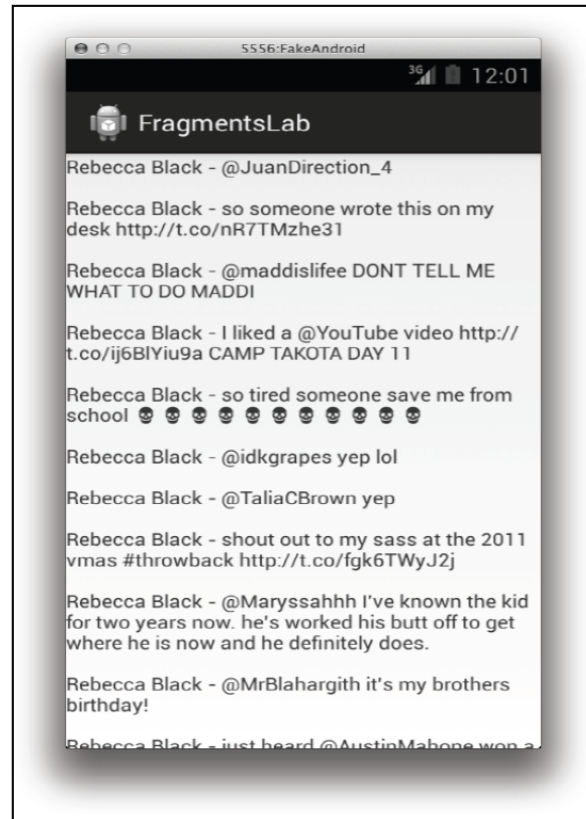
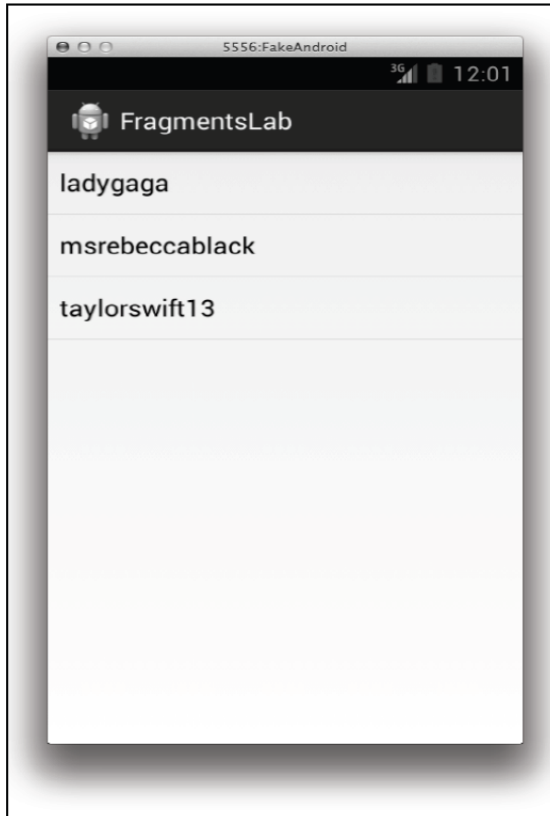
#### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### Activity 4 – Fake Twitter (Steps)

- Add two fragments (List Fragment and Details Fragment)
- When user clicks on any of the name in the list, it opens the tweets of that person.
- For tweet data add a few random tweets as array with each person's name in strings.xml file and read them from there and show them in textView.



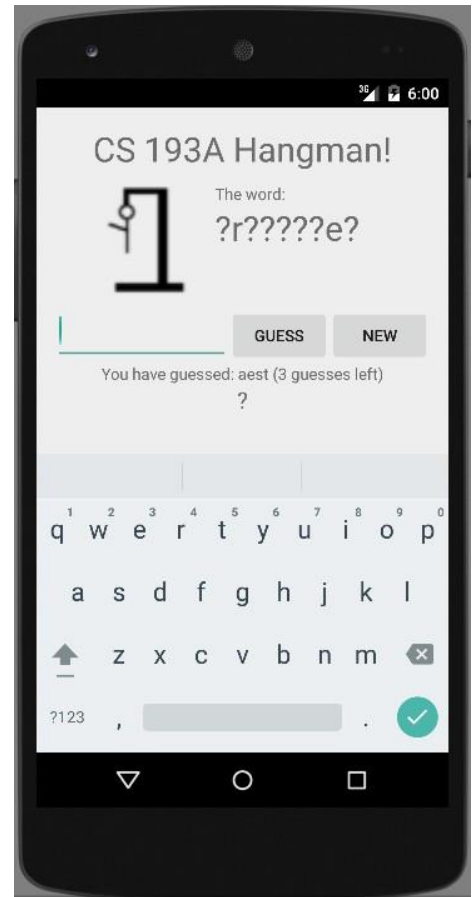
### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)

## MOBILE APPLICATION DEVELOPMENT (MAD) – LAB 7

### Activity 5 – Modify Lab 1 – Activity 5 (Hangman)

- Add Text To Speech and Speech to Text Function
- User not only can type a letter, but user can also speak a letter
- The application should remind the user about remaining number of lives by speaking back.



#### Reference Links:

1. <https://academy.realm.io/posts/michael-yotive-state-of-fragments-2017/>
2. <https://guides.codepath.com/android/creating-and-using-fragments>
3. <https://www.javatpoint.com/android-listview-example> (Listview + Adapter)