MMAD S2016
JSt vers February 10, 2016

Tingle App version 2

During the semester, we will create an app called “Tingle”. The purpose of Tingle is to register and find physical things such as books, clothing, keys etc. Many of us spend a lot of time looking for things that we cannot find and could use a Google-like search engine for things in the physical world. Something like:

During the semester you will gradually improve the user interface and add functionality to your app. A big challenge with this app is to find convenient ways of registering things. We will therefore experiment with using camera, location, bar-code reading and storing things in databases.

This week we will extend the Tingle app, so it can show all the things that been registered.

Tingle App: second step

This week we will extend version 1 of the Tingle app with a second activity that can list all things that we have stored. We will call this TingleV2.

Extending TingleV1 to TingleV2 consists of three steps:

1) Adding a second activity to show the list of things (we will call this ListActivity)
2) Create a singleton class containing all the things stored (called ThingsDB)
3) Connecting ThingsDB with ListActivity using an Adapter

Step 1 is similar (but much simpler) to the introduction of a second activity explained in Chapter 5 of the book. The two other steps are explained below.

Development of version 2 of the Tingle app

1. To do step 1 of Tingle version 2 do as explained on p. 87 – p. 97 in the book. Furthermore, add a second button ("List of all things") to TingleActivity (and its layout). When pressing this button ListActivity should be started (as explained on p. 97 of the book).

2. Create a singleton class called ThingsDB. Singletons are briefly described on p. 168 and 169 of the textbook. See also the reading part of the workplan for week3 on learnIT.

   ThingsDB is a new class that you put in the Java directory of your project. It looks as follows:

```
public class ThingsDB {
    private static ThingsDB sThingsDB;
    //fake database
    private List<Thing> mThingsDB;

    public static ThingsDB get(Context context) {
        if (sThingsDB == null) {
            sThingsDB= new ThingsDB(context);
        }
        return sThingsDB;
    }
    public List<Thing> getThingsDB() {return mThingsDB; }
    public void addThing(Thing thing) { mThingsDB.add(thing); }
    public int size() {return mThingsDB.size(); }
    public Thing get(int i){ return mThingsDB.get(i); }

    // Fill database for testing purposes
    private ThingsDB(Context context) {
        mThingsDB= new ArrayList<Thing>();
        mThingsDB.add(new Thing("Android Pnone", "Desk"));
          ... add as many as you like
        mThingsDB.add(new Thing("Big Nerd book", "Desk"));
    }
}
```

   You may use this in both TingleActivity and ListActivity by declaring a static variable:

```
private static ThingsDB thingsDB;
```

   and initializing it in OnCreate (of both activities) as follows:

```
thingsDB= ThingsDB.get(this);
```

3. In Java for Android you can use adapters to create scrollable lists. There is a nice example in the reading part of the workplan for week3 on learnIT. Do something similar to program ListActivity. You need to create a layout (list_item.xml) describing the layout of a single

element of the list. You also need to make a ListView filled by an ArrayAdapter (use *thingsDB*.getThingsDB()) to get a list of all the things stored).

Complete and run TingleV2

Fill in the missing code and try running the app from Android Studio.