IT UNIVERSITY OF COPENHAGEN

MMAD S2016
JSt vers March 30, 2016

# Second Mandatory Assignment: Tingle app final version

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:

## Tingle app: final version

The final version of the Tingle app should be able to:

- register new things using text and camera
- delete things
- store/retrieve things using a database and
- display things (and their properties) on the screen

You can decide what to emphasize in your final version of the Tingle app e.g. a nice user interface or extra functionality: but the app must have the minimal functionality described below.

You should make a user interface that is robust to changes and failures in the environment like user navigating to other apps and network environment changing.

You may use the final Tingle app to exemplify your presentation at the final exam. For example, if you get a question about the activity lifecycle at the exam, you could explain how you have handled lifecycle changes in your Tingle app to supplement your general answer of the exam question.

The minimal functionality of your final version of the Tingle app must include:

- Using a database to store things (Could be SQLite or a database on a server)
- Registration of things with text entry (as in the first Tingle versions)

- Registration of things with a barcode (including lookup of product info over the network)
- Searching for things including a principle/algorithm for ordering of multiple hits

You are free to design the user interface as you like (as long as it enables the minimal functionality) and to add extra functionality e.g. storing images of things.

## Submission information

*Due date*: the second mandatory assignment must be handed in before Wednesday April 27 (12:00 am) through learnIT.

*Grading:* Your assignment will be evaluated by the teachers and you will get some feedback including accept/no accept of the assignment. You need to have an accepted version of the second mandatory assignment before taking the final examination of the course.

*Team work:* You may discuss your solution with other students (or anybody else), but each student must submit their solution individually. There can be similarities between the code you submit and the code submitted by other students, but the documentation (see below) must be done individually.

Your submission must consist of two parts (files): *Code* and *Documentation* as explained below.

*Code:* You must submit a complete Android Studio project directory with a working app (no syntax errors or runtime exceptions. The project directory must be packed as a single zip file.

*Documentation:* You must submit a documentation (in pdf format) explaining your solution. The documentation must contain these sections:

> most important design choices, including functionality and user interface
> short user guide
> testing of the app – see below
> problems with your app (if there are any) e.g. if something does not work completely as you want

*Design choices*
You should describe your design choices both related to the user interface and the functionality e.g. what alternatives did you consider, and why do you prefer the solution you ended up using. Here is an incomplete list of design choices:

- layout of user interface and how it is implemented in activities/fragments
- structure of your database
- if you have a database on a server, how is it synchronized with the app
- how does key functions work (add – detection of duplicates, search – ordering of results, …)
- etc.

*Testing*

You should make a number of test sequences that covers all the functionality of your app (including error handling) and all parts of the user interface.

Example of a test sequence:

Register a new thing:

1. Start app in portrait mode
2. Scan a barcode
3. Lookup the product info and register this
4. Register the "where" information of the thing
5. Store the information about the new thing in the database
6. Check that the new thing is correctly stored

A complete test of your app will require many such test sequences. As you develop the app, write down relevant test sequences. Use these in the debugging and in the final test before submission.

Try to make the list of test sequences cover all the use cases of the app you can imagine including cases where the user makes mistakes and events outside your control like network failure.

The final documentation of your app must contain a sample of five test sequences and the results of the tests. You can decide which five to submit, but try to pick some that cover the key functions.