

5COSC023W - Tutorial 10 Exercises

As part of the tutorial for this week, you should complete **ALL** the tasks described in the following specifications: (**make sure that you ask questions to your tutor for anything that you do not understand or if you are stuck at any point**).

Tutorial sessions are practical sessions that you need to work towards the exercises set. They will give you the chance to practice the material learned in the lectures and learn new things as well.

You should not use these sessions to work towards the assessed assignments!

If you decide to work towards your assessed work instead, then you are not considered as part of the tutorial session. You will not get any help on the code of the assessed work by your tutor but you can ask your tutor **ONLY** about any clarifications you might need regarding the specification of the coursework.

Like all other modules, you are expected to study towards you module outside the lecture and dedicated tutorial slots for a number of hours. If you do not finish all of the exercises in the tutorial session, make sure that you finish them on your own time and by the end of the week. This is a normal process and part of your university learning.

For all the tasks you should use Jetpack Compose and NOT Views!

The Cocktails App - Another Web Services Example

Develop an Android application which the user interacts with to find and prepare cocktails. The application will use Kotlin and the `HttpURLConnection` based on what we covered in the lecture.

The application will be using the `https://www.thecocktaildb.com/api.php` Web service.

1. Implement the application so that the user can search for the names of all cocktails containing an ingredient, e.g. Vodka, by using the following url: `https://www.thecocktaildb.com/api/json/v1/1/filter.php?i=Vodka` The user should be able to see all the names of the cocktails containing vodka.
2. Extend the application so that the user can enter the name of a cocktail and receive the instructions for the preparation of the cocktail. This should be done by displaying the content of the field `strInstructions` after a request to the `https://www.thecocktaildb.com/api/json/v1/1/search.php?s=margarita` url.
3. Extend the application so that it displays a picture for the cocktail (hint: this is the url corresponding to the `strDrinkThumb` key of the JSON response in the last subquestion).

4. Extend your application so that user presses a button and he/she is presented with the full details of a random cocktail (including the name, the recipe and its picture).

Hints: you need to study the API of the website to make the relevant requests.

You can use the following function for retrieving the bitmap picture from a URL and then use a composable `Image` to display the `ImageBitmap` returned from the function:

```
// retrieve a bitmap image from the URL
fun getBitmapPicture(): ImageBitmap {
    var bitmap: Bitmap? = null

    val url = URL(picture_url)
    val con = url.openConnection() as HttpURLConnection
    val bfstream = BufferedInputStream(con.inputStream)

    bitmap = BitmapFactory.decodeStream(bfstream)

    return bitmap.asImageBitmap()
}
```