

Practical 3: Designing Layouts II

Learning Outcomes

The goal of this practical lab is to provide you with the opportunity to replicate the design of an existing app, which was originally developed within the University, by considering the design components involved in the creation and implementation of the UI for the app.

In this lab, you will:

- Gain further experience in understanding the component makeup of a UI.
- Combine *ViewGroups* to achieve a more complex UI.
- Execute an instance of your app within the emulator.

Task

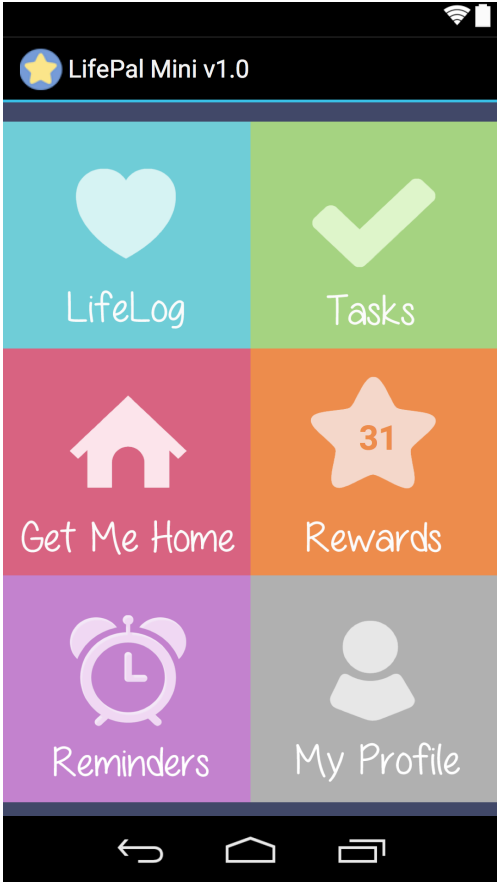
Consider the **Sample App UI** presented as an image in **Appendix A** (below). Attempt to recreate this layout within Android Studio. The primary artwork required to replicate the solution are provided for you within Blackboard (Practical3Artwork.zip). You should approach this task by considering the structure of the app UI and sketching the relevant components on paper. You may need to use a relevant image-editing tool, such as Photoshop or GIMP, in order to determine the exact colors used for the background, etc., all of which should be determined during the planning phase. Consequently, you should then reference the relevant developer support on User Interfaces¹ to help you progress with the implementation.

Save your example project and make sure to discuss your progress with one of the lab tutors before leaving the lab.



¹ <http://developer.android.com/guide/topics/ui/overview.html>

Appendix A: Sample App UI

Sample UI	Helpful Hints
	<ul style="list-style-type: none"> • Use <i>Theme.holo</i> to enable <i>ActionBar</i> by default. • Use a combined hierarchy of <i>LinearLayouts</i> to achieve the structure. • Use <i>ImageViews</i> passing in the relevant <i>drawable</i> resource (image).

Further Work

If you have finished this task early, then why not try to implement `onClick()` listeners for each of the *ImageViews* created and display a *Toast* message onscreen that details the title of the view that was clicked and handled by the relevant *onClick()* listener.

You could also consider redesigning the layouts using the more contemporary *ConstraintLayout* and noting the optimizations and challenges you meet.

