Departamento de Eletrónica, Telecomunicações e Informática

45424- Introdução à Computação Móvel / Introduction to Mobile Computing -
Semester 2 - 2024-25

Flutter Homework

Last revision: 2025-04-06

The purpose of this document is to describe the Flutter homework objectives and assessment. It also provides some examples of applications that could be used as a starting point/inspiration

The homework is individual

Objective:

Use Flutter to clone existing / concept apps and, by doing so, use/explore Flutter UI widgets and basic navigation options

Demo

The homework will have a demo in class - just show the functionality and answer questions (if any)

Delivery

Code must be delivered via eLearning as it is an evaluation item, BUT the demo is an EVALUATION moment. As such, anyone is supposed to compile and run your project from your source code and the instructions in the code (e.g. in a readme file along with the code). MAKE SURE ALL DEPENDENCIES AND INSTALLATION DETAILS ARE INCLUDED

Submit them as a zip (please clean the Flutter project before "flutter clean") in the eLearning link.

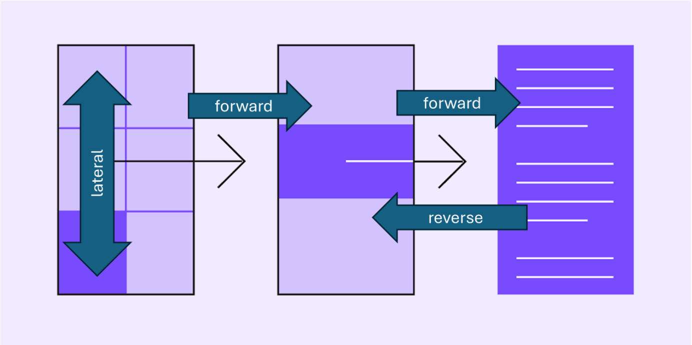
The application

The application should follow the features below. I suggest you select an existing application (e.g., Facebook) and try to mimic its UI behavior (your data/text/images can be static). Your application should have lateral, forward, and reverse navigation. The material design guidelines provide a detailed description, but here is a summary.

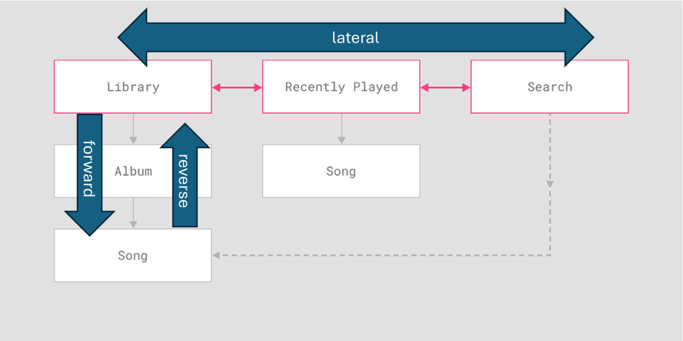
Navigation is the act of moving between screens of an app to complete tasks. It's enabled through several means: dedicated navigation components, embedding navigation behavior into content, and platform affordances.
Navigational directions
Based on your app's information architecture, a user can move in one of three navigational directions:

- *Lateral navigation refers to moving between screens at the same level of hierarchy. An app's primary navigation component should provide access to all destinations at the top level of its hierarchy.*
- *Forward navigation refers to moving between screens at consecutive levels of hierarchy, steps in a flow, or across an app. Forward navigation embeds navigation behavior into containers (such as cards, lists, or images), buttons, links, or by using search.*
- *Reverse navigation refers to moving backwards through screens either chronologically (within one app or across different apps) or hierarchically (within an app). Platform conventions determine the exact behavior of reverse navigation within an app.*

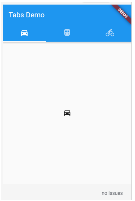
From a "screen perspective"



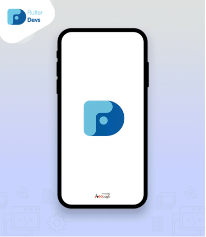
From a views/screen perspective



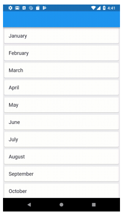
The application MUST have lateral (horizontal) navigation i.e. slide/commute between different views at the same "logical level"



Work with tabs

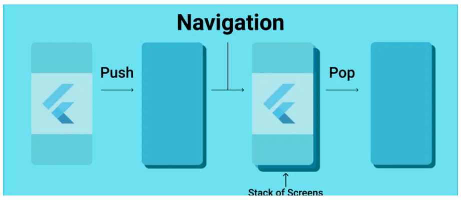


Slidable Tile In Flutter



Use listviews

The application MUST have vertical (forward and reverse)navigation, generic to detail view (pop and push views)



Navigation and routing

The application should be deployed on a device. It is accepted in the browser but may imply penalization.

Assessment

The grid may suffer changes in terms of weight, but the factors are

- Is running: something running
- Running in device: running in an actual Android phone
- Provide horizontal navigation
- Provide vertical navigation
- Realistic or basic UI: using a realistic look or a plain UI
  - e.g. based on and similar to demo counter UI
- Realistic or basic resources: fully static data? or some adaptability?
  - e.g. have a list where you can add/remove dynamic items
- Extra:
  - Unexpected integrated features - sensors? Maps? Other that may be useful in the project

Delays in the demo will automatically have a 1 point in 20 regardless of the quality of the work.

Samples

<< TO BE COMPLETED >>