

Bottom Navigation

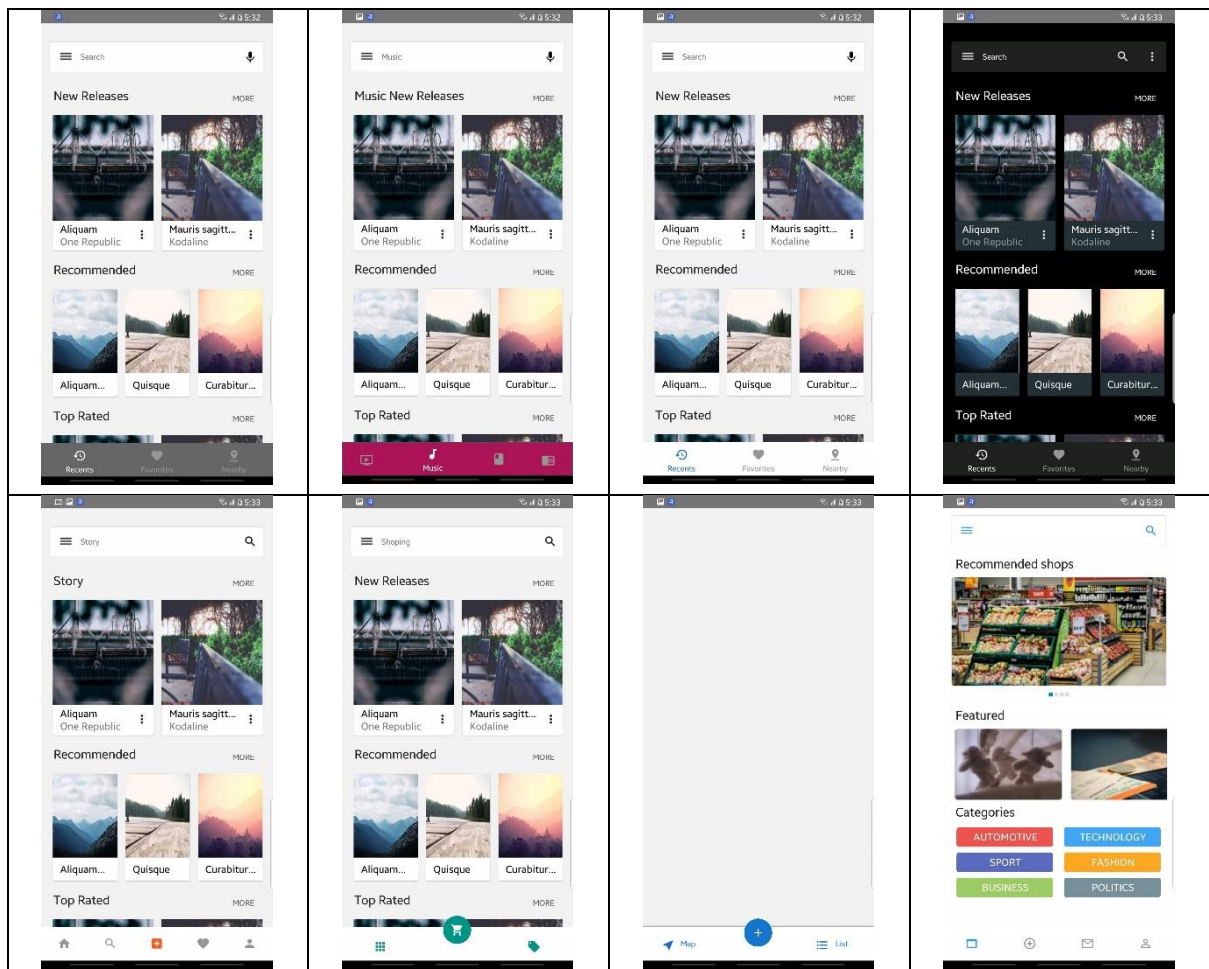
FireMonkey Material Design Template

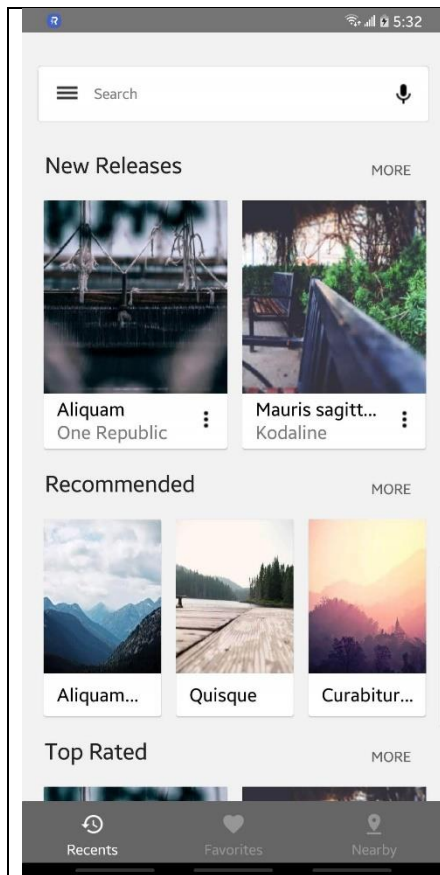
The Bottom Navigation template is designed to embrace the Material Design user interface and user experience philosophy. It looks great on both iOS and Android, and work on Windows for testing.

The template has eight different variations within the application as different forms.

1. Basic - ufrmBasic
2. Shifting - ufrmShifting
3. Light - ufrmLight
4. Dark - ufrmDar
5. Icon - ufrmIcon
6. Primary - ufrmPrimary
7. Map Blue - ufrmMap
8. Light Simple - ufrmLightSimple

Most of the forms contain multiple tabs that you move between as you tap the buttons at the bottom. The tabs content looks mostly similar, so it may not be obvious that you are moving between tabs.





Basic Template

The basic variation is found in the **ufmBasic** form and unit.

It is the simplest template that most of the others are based on.

When you tap on the three buttons in the bottom, they display a subtle colored animation as the view changes to another tab.

At the top of the template is a search box containing a menu button to open the multi-view, and a search button.

Contained within the client area is a scrolling card view with images and captions.

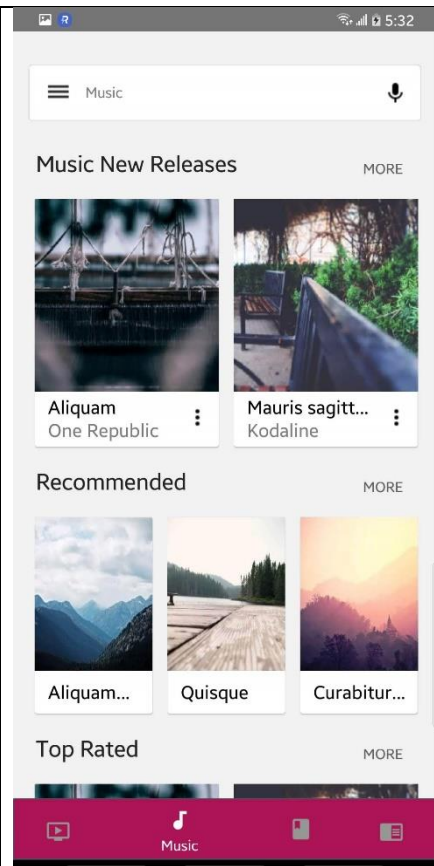
Shifting Template

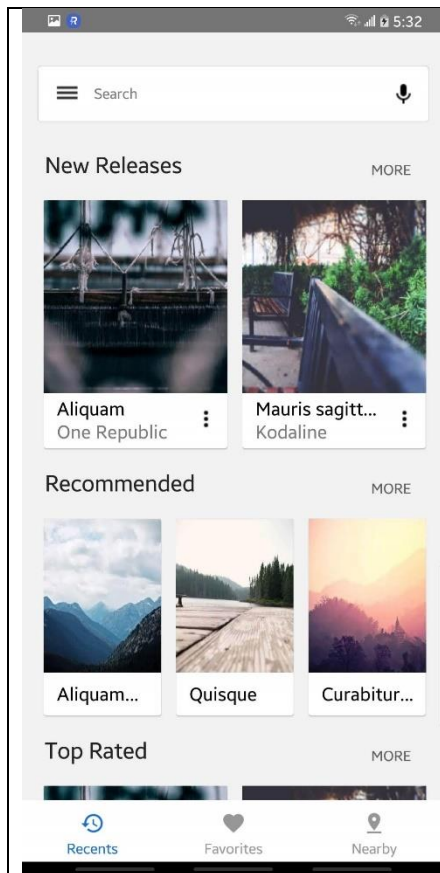
The shifting variation is found in the **ufmShifting** form and unit.

This one builds on the basic template, but as the user taps on the different icons the other icons shift away giving the selection more space to display a caption.

At the top of the template is a search box containing a menu button to open the multi-view, and a search button.

Contained within the client area is a scrolling card view with images and captions.





Light Template

The light variation is found in the **ufmLight** form and unit.

It is a variation on the basic template with a lighter bar at the bottom.

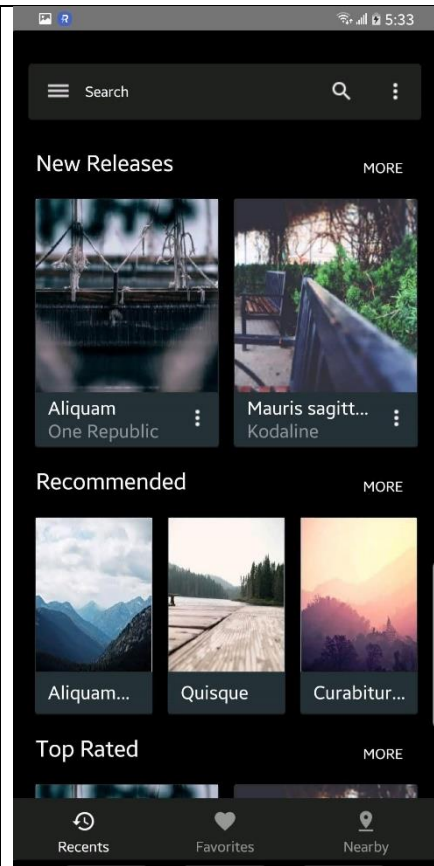
The rest of the content and behavior is the same.

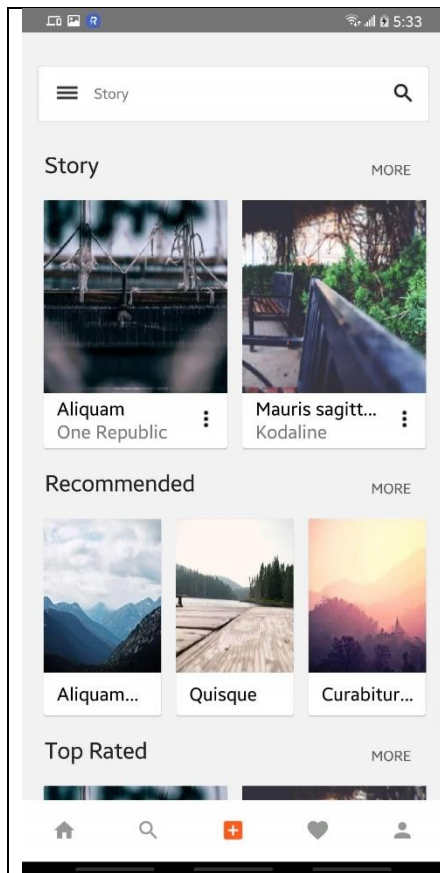
Dark Template

The dark variation is found in the **ufmDark** form and unit.

Much like how the Light template is has a lighter theme, the Dark template is a dark theme with white text on a black background.

The behavior and content are the same otherwise.





Icon Template

The icon variation is found in the **ufmIcon** form and unit.

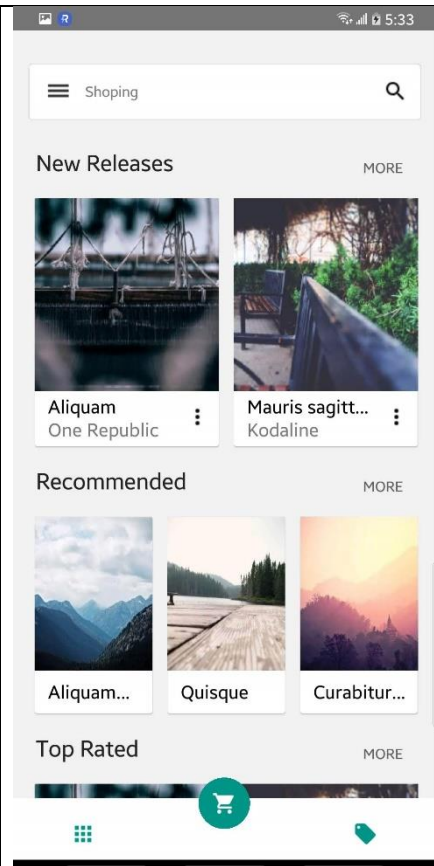
To make space for more icons along the bottom, this template only uses icons without captions.

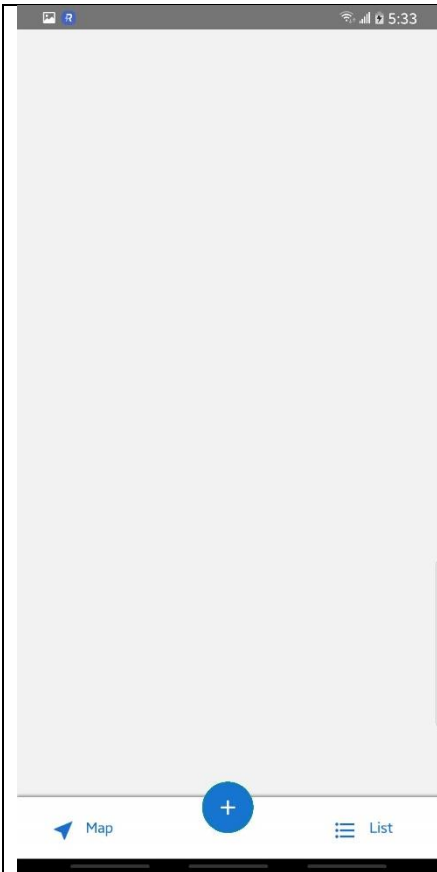
Primary Template

The primary icon variation is found in the **ufmPrimary** form and unit.

This template only has three icons the bottom, but the "primary" icon is in the center and on a larger circle. This works for situations where perhaps the other two secondary icons are navigation or other secondary items, while the user is typically in the primary icon.

Or the buttons could be used for a non-navigation purpose where the user uses the three icons to invoke different actions.





Map Blue Template

This variation is found in the **ufmIcon** form and unit.

It is like the Primary template but includes a caption on the two secondary items. It is designed for a map interface.

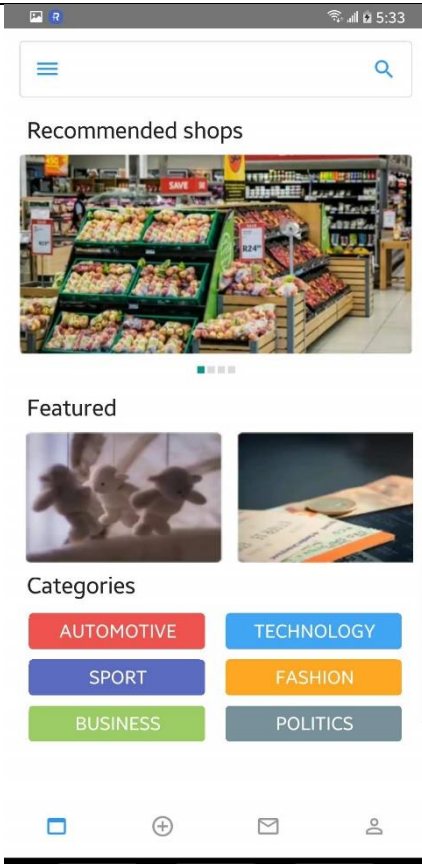
Put the TMapView in the client area, and then use the buttons on the bottom navigation to interact with the map.

Light Simple Template

And now for something completely different in the **ufmLightSimple** form and unit.

The big difference here is in the client area and the search bar at the top.

The bottom navigation template is changed to a light bar that is the same color as the background.



A few things to notice on the templates:

- The search box and navigation bar slide off the screen when scrolling down
- The search box and navigation bar reappear when scrolling back up
- The client area scrolls beneath the search box
- Tapping in the search box causes the icons to slide off the sides with a smooth animation
- Tapping on the icons along the bottom invokes a subtle animation

There are a lot of subtle design elements that you may want to replicate in other applications, even if you don't use these templates completely.

To use these templates in your project just copy the unit and form file to your project. You can further customize them and remove the TListView from them as well.

For the TListBox visual elements to display your data, rather than test data, you need to load your data into the TFDMemTable located on the form or make changes to fill method when initializing the InsertTestDataToTheMemTables () form.

InsertTestDataToTheMemTables () method fills with the test data three TFDMemTable components located on the template form. These are FDMemTable1, FDMemTable2 and FDMemTable3. According to the data from these Memory Tables, the TListBoxes on the template form are also filled, in order from top to bottom.

Also, using the same method, to display test images in TListBox, images from TImageLists are loaded into Memory Tables.

In templates, only three TFDMemTablets components are used to fill in the visual elements of a form with test data, regardless of how many tabs and buttons are implemented in the bottom menu. This is done in order not to confuse you with unnecessary test information and not to overload the example with extra information. Also, unlikely that you will use TFDMemTables in a real project, probably you will use, for example, TFDQuery to fetch data and fill in the visual elements of a form based on this data.

Filling TListBox with data from TFDMemTables is implemented in universal methods:

- LoadTopListboxItems (aListBox: TListBox)
- Load3PartsListboxItems (aListBox: TListBox)
- LoadBottomListboxItems (aListBox: TListBox)

Also, you can change these methods as you like, or write your own methods for populating TListBox with data. The methods in the template only show how to create TListBoxItems elements and how to assign touch processing methods for

TListBoxItems or buttons located inside TListBoxItems. After analyzing these methods, you will notice that the Tag property is used to pass the identifiers of a record or item.

In any case, each of the templates in the current implementation implements only style, graphic and animation effects, and the test data is presented only for understanding the processes of filling TListBox with data, and how the style elements of items are called and how to transfer certain data to them or assign methods.

This template application includes icons and images from the following sources:

Pexels.com:

Visit <https://www.pexels.com/photo-license/> to learn more about Pexel's image licensing rights.

Material.io:

Icons from Material.io (<https://material.io/icons>) are used throughout the sample application.

Embarcadero has modified all of the icons hereby used by changing the size and/or fill color ("Derivative Work"). Embarcadero grants you the right to use, reproduce, distribute and modify the Derivative Work by changing the color using RAD Studio. To the extent there are unmodified icons used by Embarcadero, those unmodified icons still governed by the Apache License v.2.0.

Copyright © 2019 Embarcadero Technologies, Inc. You are free to adapt, modify, and reuse this code and template as you see fit under your existing Embarcadero license agreement.